1995

# Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms

Maureen A. O'Rourke
*Boston University School of Law*

Duke Law Journal

VOLUME 45 DECEMBER 1995 NUMBER 3

# DRAWING THE BOUNDARY BETWEEN COPYRIGHT AND CONTRACT: COPYRIGHT PREEMPTION OF SOFTWARE LICENSE TERMS

MAUREEN A. O'ROURKE†

## INTRODUCTION

The Copyright Act of 1976[1] (the "Act") sought to clarify the boundary between federal and state enforcement of proprietary rights in works of authorship by specifically addressing federal preemption of state law causes of action in § 301 of the Act. Unfortunately, § 301 is not a model of clarity, and its legislative history is also cloudy. Consequently, the courts have had some difficulty in formulating consistent decisional guidelines in preemption cases. This difficulty has perhaps been most evident in cases in which the particular preemption issue is based not on a state statute but on state enforcement of private contractual rights. Courts have not clearly stated whether the Act preempts contracts between authors and users of copyrighted works that purport to vary the rights and obligations set forth in the Act.

To a certain extent, however, all copyright contracts vary the rights and obligations set forth in the Act. The Act grants the

† Associate Professor of Law, Boston University School of Law; J.D., Yale Law School; B.S., Marist College. Thanks to Randy Barnett, Joe Brodley, Robert Bone, Ron Cass, Carmela Correale, David Dana, Dan Freehling, Wendy Gordon, Anne Gowen, Keith Hylton, Elly Leary, Rob Merges, Fran Miller, Stuart Moskowitz, David Seipp, Manuel Utset and all the participants in the Boston University Faculty Workshop series for their comments, support and advice. Special thanks to Mary, Tom, and Patty O'Rourke as well as Trish and Eliseo Pena. Finally, thanks to the Boston University School of Law library and copy center staffs, and to my cadre of research assistants: Evan Berg, Michael Bowse, Stacy Jacob, Marie Flore-Johnson, Mark Kern, Janaki Komanduri, Simone Lonigan, and Eric Silberberg.

[1] 17 U.S.C. §§ 101–810, 1001–1010 (1994).

copyright owner certain exclusive rights.[2] Parties negotiate over the allocation of those rights, and the result of that negotiation is reflected in their contract. The exclusive rights thus resemble "default" rules in the sense that the parties are free to allocate them by private contractual agreement. At the same time, the Act also contains certain mandatory rules, like those on transfers of copyright ownership, around which individual parties may not contract.[3]

Additionally, the Act also contains certain background provisions, such as those governing "fair use,"[4] which are not clearly identified as default or mandatory rules.[5] The Act confers fair use rights nonexclusively on the public without explicitly indicating whether or not the public or its members are free to contract away those rights. Historically, it seems as if contracts restricting a licensee's fair use rights were rare.[6] This fact suggests that the Act's background rules generally have succeeded in striking the balance between rights of copyright owners and licensees in an acceptable manner, whether or not the parties actually bargained over contractual terms.[7]

This success, however, is not as evident in the electronic world. Electronic publishers customarily use license agreements to alter the bargain struck by the copyright regime. These license

[2] *See infra* note 20.

[3] 17 U.S.C. § 204(a) (1994) ("A transfer of copyright ownership . . . is not valid unless an instrument of conveyance . . . is in writing and signed by the owner of the rights conveyed . . . ."). Note that contract law, like copyright law, contains both default and mandatory rules. *See, e.g.*, U.C.C. § 1-102(3) (1995) ("The effect of provisions of this Act may be varied by agreement, except as otherwise provided in this [UCC] . . . ."). An example of where the UCC "otherwise provides" is in the Statute of Frauds provision, which is analogous to 17 U.S.C. § 204(a). *See* U.C.C. § 2-201(1) (1995) ("Except as otherwise provided in this section a contract for the sale of goods for the price of $500 or more is not enforceable . . . unless there is some [sufficient] writing . . . .").

[4] *See infra* text accompanying notes 96-98.

[5] *See generally* Wendy J. Gordon, *An Inquiry into the Merits of Copyright: The Challenges of Consistency, Consent, and Encouragement Theory*, 41 STAN. L. REV. 1343 (1989) (discussing alternatives to copyright, and how copyright may affect contractual agreements).

[6] The lack of evidence of such contracts may be due, in part, to the fact that contracts restricting the use of copyrighted materials may be confidential in nature and thus disclosed, if at all, only in litigation.

[7] *Cf.* Pamela Samuelson & Robert J. Glushko, *Intellectual Property Rights for Digital Library and Hypertext Publishing Systems*, 6 HARV. J.L. & TECH. 237 (1993) (noting that traditional copyright law has been effective in meeting parties' expectations in hard copy world).

agreements may alter the copyright balance in several ways.[8] For example, most software is distributed in object code (machine readable) form[9] with license agreements containing provisions against decompilation.[10] These provisions prohibit the purchaser/licensee from uncovering the human readable version of the software. However, at least three courts have held that under the Act's "fair use" doctrine, software purchasers/licensees may be entitled to decompile the software to access that human readable version.[11] The widespread use of license agreements including such terms suggests that copyright owners are dissatisfied with the background provisions of the Act when applied to the distribution of electronic information. Because the critical question in this context is whether private parties may contract around the fair use provision of the Act,[12] the preemption issue, long dormant, is likely to be-

[8] This Article discusses only one such alteration: prohibiting decompilation. However, license agreements also may attempt to confer copyright-type rights on noncopyrightable data, *see infra* note 313 and accompanying text, or contract around the Act's first sale doctrine, *see infra* note 201. In recent years, most mass-market licenses do not prohibit further transfer of the software but purport to bind the transferee to the same terms as those that bound the original licensee. *See generally* Thomas M.S. Hemnes, *Restraints on Alienation, Equitable Servitudes, and the Feudal Nature of Computer Software Licensing*, 71 DENV. U. L. REV. 577, 579, 580–81 (1994).

[9] *See infra* notes 42–43 (defining source code and object code).

[10] *See, e.g.*, INT'L BUS. MACHINES CORP. (IBM) CUSTOMER AGREEMENT, at 14 (on file with author) ("You agree not to do any of the following: . . . reverse assemble, reverse compile, or otherwise translate any Program."). The IBM Customer Agreement (ICA) is signed by the customer and generally is used to market software running on mid- to large-range systems. For example, the ICA is used for the licensing of AIX Version 3, which runs on IBM's UNIX workstations; of OS/400, which runs on IBM's AS/400 mid-range commercial systems; and of MVS (OS/390) and VM, which run on IBM's mainframes. IBM PROGRAM LICENSE AND LIMITED WARRANTY AGREEMENT FOR IBM® PS/1® COMPUTER, at 1 (on file with author) ("You may not: . . . reverse assemble or reverse compile any Program . . . ."). This agreement accompanies high-volume products like IBM's PS/1 personal computers, for which it is impractical to require the customer to sign an agreement. MICROSOFT LICENSE AGREEMENT—MICROSOFT MS-DOS 6.2 STEP-UP (on file with author) ("You may not reverse eugineer, decompile, or disassemble the SOFTWARE."). IBM and Microsoft are two of the largest software providers in the world. See *infra* text accompanying notes 54–62 for a detailed description of the various types of license agreements used by software vendors; text accompanying notes 42–44 for an explanation of the decompilation process; and text accompanying notes 118–21 for an explanation of the differences between the terms "decompilation," "disassembly," and "reverse engineering."

[11] *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520 (9th Cir. 1992); Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 843 (Fed. Cir. 1992); *see also* DSC Comm. Corp v. DGI Technologies, Inc., No. 3:94-CV-1047, 1995 WL 526429, at *7 (N.D. Tex. Sept. 1, 1995). *See infra* Part II for a detailed discussion of the *Atari* and *Sega* cases.

[12] Some licensors have begun explicitly limiting licensees' fair use rights. *See, e.g.*,

come critically important as works of authorship are increasingly distributed in an electronic rather than hard copy medium.

This Article argues that there are many circumstances in which the law should not preempt parties' agreements to surrender decompilation rights, despite the fact that such agreements contract around the Act's background rules on fair use; that is, it argues that fair use rights should be alienable. The Article then goes on to explore the factors that should be considered in deciding whether circumstances exist in which these otherwise alienable fair use rights should be held inalienable, preempting contractual provisions to the contrary.

The Article begins in Part I by discussing the differences between the hard and soft copy worlds. Without a basic understanding of these differences, one cannot perceive why electronic publishers have sought to contract around the copyright law to a greater extent than their counterparts in the hard copy world. In Part II, the Article focuses on judicial authority in the decompilation context. Part III then considers the decompilation term in greater detail, concluding that it generally should not be preempted in cases in which the parties have explicitly negotiated it. It further considers the argument that the case for preemption may be stronger when the term is contained in a boilerplate agreement. It contends, however, that courts should analyze relevant market evidence in both contexts rather than holding as a matter of law that the term should always be preempted or nonpreempted. Part IV sets forth a proposal for an integrated preemption analysis that places the Copyright Act within the larger federal legal context to help courts understand where, within that scheme, private contract may legitimately operate. The Article concludes with the application of that proposal by analogy to electronic databases.

WESTLAW® SUBSCRIBER AGREEMENT, § 2 (on file with author) (providing that fair use rights are accorded to a licensee only "if not otherwise expressly prohibited" by agreement).

## I. THE BACKGROUND—COPYRIGHT LAW IN THE HARD AND SOFT COPY WORLDS

### A. *An Overview of Copyright Law*

The congressional power to enact copyright legislation is found in Article I, Section 8, Clause 8 of the Constitution, which states: "The Congress shall have the power . . . to promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries; . . . ." While the Supreme Court has interpreted the constitutional grant of copyright power to permit Congress to protect only original works,[13] other details were left for legislative resolution. For example, congressional judgment as expressed in the Act determined the nature and scope of the exclusive rights to be granted as well as the duration of those rights.[14]

In exercising its judgment through implementing legislation, Congress has been guided by the policy goals underlying the constitutional grant of power. Specifically, American law historically has cited an economic rationale as the theoretical underpinning of the copyright clause set forth in the Constitution.[15] In its simplest

[13] Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 345 (1991).

[14] *See, e.g.*, 17 U.S.C. § 106 (1994) (enumerating exclusive rights of copyright owner); *id.* §§ 107–120 (setting forth limitations on copyright owner's exclusive rights); *id.* § 302 (setting statutory term of copyright protection as, generally, life of author plus 50 years).

[15] *See, e.g.*, Mazer v. Stein, 347 U.S. 201, 219 (1954) ("The economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors in 'Science and useful Arts.' "). The legislative history of the 1909 Copyright Act states that

> [t]he enactment of copyright legislation by Congress under the terms of the Coustitution is not based on any natural right that the author has in his writing . . . but upon the ground that the welfare of the public will be served and progress of science and useful arts will be promoted by securing to authors for limited periods the exclusive rights to their writings.

H.R. REP. NO. 2222, 60th Cong., 2d Sess., pt. 1, at 7 (1909); Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2391 n.332 (1994) (quoting Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 558 (1985) ("[I]t should not be forgotten that the Framers intended copyright itself to be the engine of free expression. By establishing a marketable right to the use of one's expression, copyright supplies the economic incentive to create and disseminate ideas.")). This economic approach of American law contrasts with the moral rights approach of some European countries. However, as global harmonization proceeds through terms, this rationale may be described as a response to market imperfections caused by a public goods problem.[16] A public good such as the national defense is one from which everyone may benefit whether or not they contribute to its production (the nonexcludability problem) and one which is characterized by relative inexhaustibility of supply.[17] Because of the nonexcludability problem, no *one* party has an incentive to contribute to the particular public good's production.[18] Thus, such a good will be underproduced in the absence of some mechanism by which producers may recoup their investments.[19]

Traditional literary works such as books resemble public goods in that an author is unlikely to make the investment to create the book if all may copy it without fee upon its publication. Copyright law seeks to solve this problem by conferring on the author a limited statutory monopoly through the grant of certain exclusive rights.[20] Although this monopoly is meant to address

multilateral agreements such as the Berne Convention, certain aspects of moral rights have made inroads into American law. *See, e.g.*, 17 U.S.C. § 106A (1994) ("[r]ights of certain authors to attribution and integrity").

[16] U.S. CONGRESS, OFFICE OF TECHNOLOGY ASSESSMENT, FINDING A BALANCE: COMPUTER SOFTWARE, INTELLECTUAL PROPERTY AND THE CHALLENGE OF TECHNOLOGICAL CHANGE 185-86 (1992) [hereinafter OTA REPORT]; William M. Landes & Richard A. Posner, *An Economic Analysis of Copyright Law*, 18 J. LEGAL STUDIES 325, 326 (1989). *See generally* Wendy J. Gordon, *Fair Use as Market Failure: A Structural and Economic Analysis of the Betamax Case and Its Predecessors*, 82 COLUM. L. REV. 1600 (1982).

[17] ROBERT COOTER & THOMAS ULEN, LAW AND ECONOMICS 46-48 (1988) (identifying nonrivalrous consumption and nonexcludability problem as distinguishing characteristics of public goods); *see also* OTA REPORT, *supra* note 16, at 185 ("In economic terms, a 'public good' is one that has the property of *nonexclusivity*: once the good has been produced, it is impossible (or prohibitively costly) to exclude any individual from benefiting from it, whether or not he or she pays."); David A. Rice, *Public Goods, Private Contract and Public Policy: Federal Preemption of Software License Prohibitions Against Reverse Engineering*, 53 U. PITT. L. REV. 543, 544-45 (1992) (summarizing public goods aspects of software).

[18] OTA REPORT, *supra* note 16, at 185 ("Indeed, individuals have an incentive not to pay for the good, or to undervalue it, in hopes of getting access as 'free riders.' "); COOTER & ULEN, *supra* note 17, at 46-48.

[19] OTA REPORT, *supra* note 16, at 185 ("The inability to exclude free riders distorts market signals and is thought to result in inefficient allocation of resources to nonexclusive goods and underproduction of them, relative to socially optimal quantities.").

[20] 17 U.S.C. § 106 (1994) (providing copyright owner with exclusive right to reproduce copyrighted work in copies, to prepare derivative works and publicly to distribute, display and perform copyrighted work, and to authorize others to do so).

> Establishment of a system of defined "intellectual property rights" can help alleviate . . . [the public goods] difficulty. . . . In granting a limited monopoly through copyright or patent, government attempts to compensate for distortions

the market imperfections inherent in the public goods model, it, in turn, by its very nature, introduces its own market imperfections.[21] The copyright law attempts to counteract those imperfections by placing certain limitations on the copyright owner's exclusive rights, including the fair use limitation codified in § 107 of the Act.[22] Additionally, to preserve the free exchange of ideas, copyright protection does not extend to the ideas embodied in a work, but only to the work's expression.[23] The law thus seeks to strike a delicate balance between providing an incentive to create while maintaining the free flow of the information on which such creativity is built.[24]

> arising from nonexclusivity. According to this rationale, without the counterbalancing grants of monopoly power bestowed through copyright and patent, the inability of authors and inventors to appropriate economic returns from their labors would result in the underproduction of new works and inventions.

OTA REPORT, *supra* note 16, at 185 (footnotes omitted). *But see* COOTER & ULEN, *supra* note 17, at 48 (noting that because government may choose to produce public goods or use tax subsidies to encourage private production and correct incentive structure, the grant of exclusive rights is not the sole means to correct the public goods problem).

[21] OTA REPORT, *supra* note 16, at 186 (listing detriments of monopoly, including (1) lower production at higher price; (2) creation of excessive incentives for investment in activities accorded monopoly status resulting in inefficient allocation of resources; (3) production of "spillover" effects—externalities in other markets; and (4) costs of intellectual property regime); COOTER & ULEN, *supra* note 17, at 38–40.

[22] 17 U.S.C. § 107 (1994); *see* Gordon, *supra* note 16, at 1601 (arguing that courts and Congress use fair use to effectuate socially desirable uncompensated transfers otherwise not occurring because of market failure); David A. Rice, *Sega and Beyond: A Beacon for Fair Use Analysis . . . At Least as Far as It Goes*, 19 U. DAYTON L. REV. 1131, 1178 (1994) ("Section 107, particularly, embodies a doctrine developed to check the application of copyright law in a manner that entrenches or expands rightholder protection at the expense of achieving the paramount objectives of expanding, and broadening use of, the public domain of knowledge and ideas."); *see also infra* text accompanying notes 96–98 and 152; 17 U.S.C. §§ 108–120 (enumerating further limitations on copyright holder's exclusive rights).

[23] 17 U.S.C. § 102(b) (1994) ("In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."); *see* Charles R. McManis, *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community*, 8 HIGH TECH. L.J. 25, 40–41 (1993) (noting corollary doctrines of (1) "merger," which states that where ouly one or a limited number of ways to express an idea exists, idea and expression merge, rendering copyright protection uuavailable; and (2) denial of protection for facts). *See generally* Jessica Litman, *The Public Domain*, 39 EMORY L.J. 965 (1990) (stating in part that ideas are reserved to public domain to enable authors to express basic ideas uniquely, without fear of infringing another's copyright).

[24] "The primary objective of copyright is not to reward the labor of authors, but '[t]o promote the Progress of Science and useful Arts.' . . . To this end, copyright as-

The public goods problem addressed by copyright law is particularly acute in the case of software. Millions of dollars may be invested in software design and coding.[25] However, once the software is distributed on disk or made electronically accessible, it is easily copied and distributed to others.[26] Thus, the copyright approach, which had worked so well in addressing similar problems in the hard copy world, seemed to be at least an appropriate starting point for protection of computer software.[27]

sures authors the right to their original expression, but encourages others to build freely upon the ideas and information conveyed by a work." Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 349–50 (1991) (citing Harper & Row, Publishers, Inc. v. Nation Enters., 471 U.S. 539, 556–57 (1985)); *see* Samuelson et al., *supra* note 15, at 2330–31 (noting that innovation in software is both incremental and cumulative in nature, often deriving from existing ideas in the hard and soft copy worlds).

[25] It is difficult to obtain reliable data on the actual cost of developing and maintaining particular software because companies regard this information as proprietary. Some commentators, though, have ventured estimates. *See, e.g.*, Joel Dreyfuss, *Struggle for Dominance: Operating Systems: Software's Crucible*, INFORMATION WEEK, Oct. 24, 1994, at 2 (stating that IBM spent at least $1 billion developing OS/2). *See generally* Lisa Picarille, *IBM Drags Feet on OS/2 Pitch*, COMPUTERWORLD, Apr. 17, 1995, at 1, 12 (estimating that IBM has spent more than $2 billion on OS/2 development and marketing since 1987); Samuelson et al., *supra* note 15, at 2321 n.39 (citing Edward Bride, *Software Magazine: A Look Back*, 11 SOFTWARE MAG. 89 (1991)) (noting that IBM's AS/400 operating system had 6.9 million lines of source code). For the fiscal year ending June 30, 1994, Microsoft claimed a research and development expense of $610 million. MICROSOFT CORP., 1994 ANNUAL REPORT, *available in* WESTLAW, SEC-ONLINE database, at *38.

[26] *See infra* note 35.

[27] However, it is a fundamental axiom of copyright law that mere effort alone—or dollars expended in creation of a work—is insufficient to justify the grant of a copyright in a particular work. *See generally Feist Publications, Inc.*, 499 U.S. at 340 (interpreting Copyright Act as rejecting "sweat of the brow" doctrine). Rather, a work must meet the threshold requirements set forth in § 102(a) of the Act. Moreover, under § 102(b), copyright protection does not extend to ideas, procedures, or methods of operation. The courts have had some difficulty in applying copyright principles, particularly § 102(b), to new technology that is different from traditional hard copy works but still requires a substantial investment in its development. *See infra* text accompanying notes 51–52 (defining scope of protection for software is difficult because § 102(b) idea/expression distinction is difficult to draw); *cf.* Maureen A. O'Rourke, *Proprietary Rights in Digital Data*, 41 FED. B. NEWS & J. 511 (1994) (arguing that copyright should serve as appropriate model for proprietary rights in most data traveling over networks but that access providers may seek to contract around copyright law in certain circumstances).

### B. *Distribution Models Under the Copyright Act*

1. *The Nature of the Licensing Transaction and Its Standard Terms.* Although the copyright model may, at first glance, seem to provide adequate protection for software, software providers frequently attempt to alter the relationship created by that model by contracting with parties who receive copies of the software, denominating such parties as copyright licensees rather than purchasers.[28] This approach is in stark contrast to the general approach of publishers of hard copy materials. Such publishers usually simply place a copyright notice on works of authorship, using the Copyright Act's set of enumerated rights and limitations as, essentially, a boilerplate contract. The rationale behind these different approaches helps to illuminate the shortcomings of the copyright regime in dealing with software.

Typically, the purchaser of a hard copy publication is likely to be a consumer who is—or is perceived to be—an uninformed and perhaps even incompetent party.[29] The provider, a publishing house, is usually thought of as both informed and competent. Thus, one might expect providers to put consumers on notice of the applicable law by including license agreements with their publications, even if the licenses simply restated the basic provisions of copyright law. Yet license agreements generally do not accompany hard copy publications.[30]

[28] *See, e.g.*, license agreements cited *supra* uote 10. This Article uses the term "licensee" to encompass all "purchasers" of software. However, the licensee is likely to be a true "licensee" only when it actually negotiates with the software provider (licensor), but is more likely to resemble a product purchaser when it merely buys a software package off the shelf. See *infra* Part III for a discussiou of the relevance of the distinction between uegotiated and nonnegotiated contracts.

[29] The consumer is "incompetent" to the extent that even if he were informed about the relevant law, he would be unable to understand its substantive consequences.

[30] Many publishers simply place the statutory notice on the works they distribute. *See* 17 U.S.C. § 401 (1994) (Optional notice of copyright must have symbol ©, word "Copyright", or "Copr."; year of first publication; name of copyright owner; and must be positioned in such a manner and location as to give reasonable notice of copyright claim.). Others will include a brief statement with the copyright notice that primarily addresses the copyright holder's exclusive right of reproduction. For example, some publishers use statements such as "This book, or parts thereof, may not be reproduced in any form without permission." LILIAN JACKSON BRAUN, THE CAT WHO CAME TO BREAKFAST 6 (1994). Other publishers are beginning to include copyright statemeuts that explicitly address the electronic medium: "All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means including informa-

Several explanations may be offered for this seemingly anomalous publisher behavior. First, the provisions of the Copyright Act may embody the reasonable expectations of even the uninformed purchaser. In other words, perhaps the "uninformed" purchaser is not so uninformed after all. This explanation is not as farfetched as one might think. After all, the Copyright Act, in one form or another, has been with us since 1790,[31] and the availability of hard copy publications predates even that Act. Purchasers thus may be aware of the relevant legal regime, perhaps regarding it not so much in legal terms but more as a matter of intuition or common sense. Second, hard copy publishers can detect large-scale infringement and bring a few highly publicized cases that both further drive that awareness home and act as a deterrent against others' copying.[32] Finally, because of the fairly widespread common understanding of copyright law, a purchaser likely knows that the price charged for a hard copy publication does not include the right to copy and distribute it in competition with the publisher.

What is so different in the soft copy world that compels providers to include license agreements with their software? First, in the earliest days of software distribution, it simply was not clear that software was protected by copyright law.[33] Thus, software

tion storage and retrieval systems without permission in writing from the publisher, except by a reviewer who may quote brief passages in a review." JESSE DUKEMINIER & JAMES E. KRIER, PROPERTY at iv (1988).

[31] Act of May 31, 1790, ch. 15, 1 Stat 124 (1790). Major revisions to the first Copyright Act were enacted in 1831, 1870, 1909, and 1976. *See generally* WILLIAM F. PATRY, LATMAN'S THE COPYRIGHT LAW 2-15 (6th ed. 1986) (outlining roots and history of U.S. copyright laws).

[32] *See* Basic Books, Inc. v. Kinko's Graphics Corp., 758 F. Supp. 1522 (S.D.N.Y. 1991) (holding that sale of packets containing copyrighted readings to students without permission of copyright owner amounted to copyright infringement); *see also Copyrights: Infringements*, 59 U.S.L.W. 2592, 2592 (1991) (discussing Kinko's case); Maryellen O'Shaughnessy, *Campus Chaos After Kinko's Loses Suit*, BUS. FIRST-COLUMBUS, Sept. 16, 1991, at 1. In recent years, high profile cases have been brought involving electronic information as well. *See generally* Michael Meyer, *Crimes of the 'Net,'* NEWSWEEK, Nov. 14, 1994, at 46, 47 (noting proliferation of unauthorized copyrighted information available on Internet and listing recent such incidents and indictments).

[33] *See* NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1978).

> [I]t became apparent that problems raised by the use of the new technologies [such as] . . . computers . . . were not dealt with by the then pending [copyright] revision bill . . . . CONTU was created to provide . . . recommendations concerning those changes in copyright law or procedure needed both to assure public access to copyrighted works in conjunction with computer[s] . . . and to respect the rights of owners of copyright in such works, while considering the concerns of the general public . . . .

providers perceived a need for contracts to serve as a type of "private legislation" of copyright-type rights in the absence of certainty that copyright protection extended to this new subject matter.[34] Second, even now that the availability of copyright protection is more certain, licensees' expectations in the electronic context are less clear. Software is still a relatively new technology and its widespread distribution began only in the last ten or twenty years, while the availability of hard copy works predates the enactment of the first Copyright Act. Also, by its very nature, software is more easily copied and distributed than hard copy works,[35] with this copying and distribution generally conducted without much fear of detection by the copyright owner.[36] Thus, in the soft copy world, there is no longstanding, widely understood legal regime to

*Id.*; *see also* Hemnes, *supra* note 8, at 578 n.3 (noting that 1976 Act extended copyright protection to computer programs to unspecified extent, while 1980 amendments explicitly applied 1976 Act to computer programs); Samuelson et al., *supra* note 15, at 2348 n.146. Samuelson notes that the U.S. Copyright Office initially expressed doubts about the copyrightability of object code:

> The Office recognized that the functionality of programs in machine-executable form might disqualify them from copyright protection under principles from the Supreme Court decisions in White-Smith Music Publishing Co. v. Apollo Co., 209 U.S. 1, 17 (1908) (concerning unreadability) and *Baker* [*v. Selden*], 101 U.S. at 104–05 (concerning functionality). . . . [The Office] decided, however, in the mid-1960s, to issue registration certificates for programs, although the certificates bore witness to the Office's doubts as to the validity of claims to copyrights in machine-executable programs.

*Id.* (citation omitted); *see also* McManis, *supra* note 23, at 41–42 (discussing first generation of computer cases in which commentators argued against extending copyright protection to publicly distributed object code).

[34] *But see* Hemnes, *supra* note 8, at 577–81 (citing need to protect trade secrets embodied in source code as primary reason for software licensing); Rice, *supra* note 17, at 553 (noting that contract became important means of strengthening trade secret protection). However, note that most mass-distributed object code today is not distributed with confidentiality terms alluded to by Hemnes.

[35] *See* OTA REPORT, *supra* note 16, at 176 (While "[u]nauthorized copying and distribution is not a problem unique to digital information," limitations exist in the distribution of hard copy materials that do not in the electronic forum. For example, hard copy quality degrades with each copy, photocopying is time-consuming, and a copied document is still in the same form as the original and therefore is often identifiable as the copyrighted work. In the electronic world, the quality of copies does not degrade, copying can be achieved by merely pressing a few keys, and the copyrightable nature of the original may be disguised by modifications. Moreover, copies multiply: One copy uploaded to a bulletin board may be disseminated to hundreds or thousands of subscribers.).

[36] *Cf. id.* at 97–98 (discussing software piracy and industry attempts to overcome difficulties of detection through industry organizations such as Software Publishers Association (SPA)).

help provide licensees with, minimally, an intuitive appreciation of the applicable law. Moreover, because of software copyright owners' difficulties in detecting large-scale infringement, there are fewer opportunities for them to publicize major cases highlighting the applicability of copyright law to the electronic context.[37] Therefore, software providers have concluded that, in a soft copy world, it is critically important to put licensees on notice of the copyright law.[38]

To a certain extent, that is exactly what software license agreements do. Generally, with few exceptions, their terms track those of the Copyright Act.[39] The exceptions, however, are troubling from the Act's perspective. For example, software license agreements typically prohibit decompilation by the licensee,[40] even though that decompilation may constitute permissible fair use under the Act.[41] Decompilation is a method of translation. Generally, software is written by a programmer in source code form.[42] The source code is processed through another program called a compiler and then distributed in the resulting object code, which is comprehensible to the computer on which it runs but not to the human user.[43] Decompilation is a process through which a

[37] *But see* Meyer, *supra* note 32, at 46-47 (noting that copyright enforcement in the electronic context is beginning to increase).

[38] *Cf.* Samuelson et al., *supra* note 15, at 2373 n.261 (describing advertising by software vendors, activities of SPA, and legal means used to put customers on notice of copyright law).

[39] *See* license agreements cited *supra* note 10. Also, note that license agreements address matters in addition to those relating to rights in the software. For example, most licenses address warranties, limitations of liability, and choice of law. *See* Hemnes, *supra* note 8, at 581 (noting that warranty disclaimers, limitations on liability, noncompetition agreements, and indemnification clauses are often included in license agreements); Rice, *supra* note 17, at 564 ("The purpose and effect of most mass market contracts . . . is to allocate product failure and performance risks.").

[40] *See supra* note 10.

[41] *See* discnssion *infra* Section II(A).

[42] Source code is the human (computer programmer) readable version of the program. See generally Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984), which described source code as follows:

> High level language, such as the commonly used BASIC or FORTRAN, uses English words and symbols, and is relatively easy to learn and understand . . . . A somewhat lower level language is assembly language, which consists of alphanumeric labels . . . . Statements in high level language, and apparently also statements in assembly language, are referred to as written in "source code."

[43] Andrew Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 U. DAYTON L. REV. 843, 859 (1994) ("Object code consists of numeric codes specifying each of the computer instructions that must be executed, as well as the locations in

user takes the object code version of the program, processes it through another program called a decompiler, and receives output in the form of an assembler or higher-level source code version of the licensed program. This additional, higher-level language version of the program is readily understood by a programmer.[44] However, courts have held that this "intermediate" copy of the program may be an infringing copy of the original object code version from which it was derived.[45]

Although most standard software license agreements merely echo the provisions of the Act, the frequently included prohibition against decompilation may be inconsistent with judicial interpretations of the Act. At least three courts have interpreted § 107 of the Act, the fair use section, to allow decompilation under certain limited circumstances.[46] Software is usually distributed in object code form. While the user may be able to discover some of the

memory of the data on which the instructions are to operate."). Computer hardware consists of electronic circuits/switches that understand only electronic impulses in one of two states—on or off. Object code sets those states and is thus like machine language. It is usually represented by strings of 1's and 0's that signify the on/off state of a particular switch. *See* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 698 (2d Cir. 1992) ("Object code is the binary language comprised of zeros and ones through which the computer directly receives its instructions.").

[44] However, some commentators contend that the decompiled code is of limited usefulness. For example, Johnson-Laird argues that decompilation is a "mythical process." When source code is compiled into object code, programmer comments written into the source code are deleted, the source code is rearranged to optimize execution speed, and symbolic names are stripped. Thus, decompilation cannot reveal the original source code complete with commentary. Johnson-Laird, *supra* note 43, at 843 n.4, 859, 899. Professor Samuelson and her colleagues also argue that

> [d]ecompiled code is not as useful as source code. Decompiling cannot, for example, restore the mnemonic names for variables and procedures chosen by the programmer. Programmers select those names to help them keep track of their own code, by making clear in the name what each piece of code is doing. Hence they greatly assist anyoue else who wishes to understand the code. But those names are lost in the translation from source to object code and cannot be recreated during reverse analysis.

Samuelson et al., *supra* note 15, at 2336 n.90.

[45] Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1519 (9th Cir. 1992) ("[W]e hold that intermediate copying of computer object code may infringe the exclusive rights granted to the copyright owner in section 106 of the Copyright Act regardless of whether the eud product of the copying also infringes those rights."); *see infra* Section II(A). In the hard copy context, the expression is readily available to the purchaser—for example, if one buys a book, he expects to read the book's text. In contrast, the licensee of software, as a general rule, buys its functionality, not its underlying text. *See infra* text accompanying note 162. Thus, the analogy between hard and soft copy works is somewhat strained.

[46] *See infra* note 76 and Section II(A).

software's ideas merely by running the program, many of the ideas—as well as their expression—are embedded in the underlying source code. These embedded ideas are not protected by copyright, although the embedded expression is.[47] Thus, courts have allowed decompilation as a means to obtain the ideas in software that are otherwise inaccessible because they are hidden in incomprehensible object code.[48] This is consistent with the Act's policy of promoting the free flow of ideas through society as building blocks for further innovation.[49] At the same time, because decompilation also reveals expression protected by copyright, the courts have attempted to define narrowly the circumstances under which decompilation is allowed so as not to have an adverse impact on the creative incentive of copyright.[50] This judicial balancing act is necessitated by the unique nature of software, which requires adjusting the principles traditionally applied to hard copy works.

In the hard copy context, although separating idea and expression to define the scope of a plaintiff's copyright may be difficult,[51] at least the ideas are sufficiently apparent for a court to understand. For example, distinguishing between idea and expression may be accomplished simply by looking at the table of contents or reading the material. Thus, courts can effectuate the purposes of the Act without too much difficulty. Moreover, because copyright principles are well established and time-tested in the hard copy world, copyright owners and purchasers of copyrighted materials in hard copy form can conduct their business in light of a legal regime that provides some degree of certainty regarding the results that will obtain if a copyright dispute reaches the courts.

[47] 17 U.S.C. § 102(b) (1994). See *supra* note 23 for the text of § 102(b).

[48] *See infra* Section II(A).

[49] *See supra* notes 23–24 and accompanying text.

[50] Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 843 (Fed. Cir. 1992) ("[R]everse engineering object code to discern the unprotectable ideas in a computer program is a fair use. . . . The fair use reproductions of a computer program[, however,] must not exceed what is necessary to understand the unprotected elements of the work.") (citation omitted); *see also infra* Section II(A).

[51] *See* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931) (L. Hand, J.) ("Nobody has ever been able to fix that boundary [between idea and expression], and nobody ever can."); *see also* Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487, 489 (2d Cir. 1960) (L. Hand, J.) (noting that decisions in this area "must . . . inevitably be *ad hoc*.").

The distinctions and policies of the Act are more difficult to make and enforce in the soft copy world because the courts have not yet been able to develop a reliable test for drawing the line between idea and expression.[52] As a result, software providers are never sure, *ex ante*, exactly what part(s) of their program will be protected and what will not. Thus, they often seek to prevent access to the *entire* program through contractual provisions against decompilation. Yet judicial enforcement of these provisions would prevent access to those embedded ideas that are not protected by copyright. Such enforcement, therefore, may seem at first glance to undercut fundamentally the copyright policy of maintaining the free flow of ideas in society in an effort to encourage further creative activity.[53]

2. *The Contracting Process.* Software providers generally contract with end users in one of three ways. At one end of the spectrum are individually negotiated transactions, usually between two competent, informed parties.[54] Often these transactions reflect a long-term relationship between the parties and involve customized software tailored to fit the licensee's needs.[55] In these cases, the license agreement usually contains a decompilation prohibition.[56]

[52] The first case to attempt to make that distinction was Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1236 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987) ("[T]he purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea.") (emphasis omitted). However, *Whelan* met with a mixed reception, with some courts opining that its rule was technically simplistic and overbroad, protecting too much of a program. *See* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 705–06 (2d Cir. 1992) (collecting cases after *Whelan* and discussing *Whelan's* flaws). *Altai* adopted a three-part abstraction-filtration-comparison test for distinguishing idea from expression. *Id.* at 706–11. *See infra* text accompanying notes 138–45 for further elaboration of the *Altai* test.

[53] *See supra* text accompanying notes 23–24.

[54] *See, e.g.*, Glen Rifkin, *Wang and Microsoft: Foes Turn into Allies*, N.Y. TIMES, Apr. 13, 1995, at D8 (describing agreement providing that Microsoft is to receive a license to Waug's imaging software and that Microsoft is to incorporate software in future versions of Windows95 and WindowsNT operating systems as standard features); *Apple Computer and Independence Technologies Announce Agreement for Data Access Language*, PR NEWSWIRE, Jan. 24, 1994.

[55] *See, e.g.*, *General Electric Chooses PDA Engineering Software for Computer-Aided Engineering*, Business Wire, Jan. 20, 1993, *available in* WESTLAW, BUSWIRE database; Peter H. Lewis, *Microsoft's Next Move Is On Line*, N.Y. TIMES, Jan. 13, 1995, at D1 (discussing Microsoft's payment of an undisclosed sum to Spyglass Inc. to license the rights to Mosaic, a software product that lets users with computer modems access the Internet easily).

[56] This assertion is difficult to prove because most of these contracts are confiden-

However, in some circumstances, software providers also license the source code to customers, usually for maintenance purposes and under strict confidentiality terms.[57]

tial. It is based on the author's observation of industry practice while working for three years in IBM's legal department. *See also* Marshall Leaffer, *Engineering Competitive Policy and Copyright Misuse*, 19 U. DAYTON L. REV. 1087, 1103 (1994) ("Contractual provisions restricting reverse engineering are the most obvious and traditional method to obstruct reverse engineering. They are quite common and are found in both negotiated licenses and those characterized as 'shrink wrap licenses.' "). *But see* David A. Rice, *Licensing the Use of Computer Program Copies and the Copyright Act First Sale Doctrine*, 30 JURIMETRICS J. 157, 169 (1990) (noting that licensing practices differ).

[57] In the last few years, there has been an industry move to "open systems." While the term "open" means different things to different people, some have interpreted it to mean that customers should be able to license source code for their own internal business purposes. This approach has been most prevalent in the UNIX software market in which complex licensing arrangements often allow vendors to license the source code of UNIX-based operating systems to customers. *See* Lawrence M. Fisher, *New Crusader in Software's Holy War*, N.Y. TIMES, Oct. 3, 1993, § 3 (Business), at 7 ("Over the years, the UNIX operating system became the lingua franca of computing's 'open systems' movement, which is spurring competition by allowing customers to mix and match different vendors' hardware and software."); IBM CUSTOMER AGREEMENT ATTACHMENT FOR AIX PROGRAM SOURCE CODE 1–2 (on file with author). Customers may obtain selected AIX files in source code form, provided they have prerequisite source code licenses from (1) AT&T or UNIX System Labs., Inc. for certain UNIX software; (2) the University of Berkeley for certain Berkeley Software Distribution software; and (3) the Open Software Foundation, Inc. for OSF/1 software. There may be further charges for licensing the third party software. For example, OSF charges for source code licenses. OSF's fee is $125,000 for a full distribution rights source code license for OSF/1 on one CPU; each additional CPU source code license ranges between $3,000 and $4,000. OSF/1 1.3 PRICE LIST at 1 (on file with author) For a discussion of source versus object code license fees generally, see *infra* notes 226–29 and accompanying text. Moreover, the source code is provided under strict confidentiality provisions. See IBM CUSTOMER AGREEMENT ATTACHMENT FOR AIX PROGRAM SOURCE CODE, *supra* at 2, which states:

> All Information (including ideas, concepts, know-how, and techniques contained in the [AIX source code]) shall be confidential and used solely in accordance with [the license granted] . . . and the prerequisite third-party licenses. . . . You agree to: 1. use the same degree of care as required by the prerequisite third-party licenses to avoid disclosure, publication or dissemination of the AIX Program source code; and 2. utilize the security functions implemented in AIX Program object code or their functional equivalent. Such security functions include password controlled access, secured dial up facility, and file permission access. You agree to provide secure lockable work facilities where all work using AIX Program source code will be performed and where all copies will be kept.

*See also* OSF/1 OPERATING SYS. COMPONENT STANDARD SUPPLEMENT § 9.1, at 8 (on file with author):

> Licensee shall not disclose any or all such Confidential Information (including methods or concepts utilized therein) to anyone, except to employees and contractors of Licensee to whom such disclosure is necessary under this Agreement. . . . Licensee shall appropriately notify each employee to whom any such disclosure is made that such disclosure is made in confidence and shall be kept in confidence by such employee.

At the other end of the licensing spectrum are the familiar shrink wrap agreements.[58] In the case of mass market software, usually distributed for use with high-volume hardware like personal computers (PCs), licensors cannot practically incur the huge transaction costs that would be involved if they attempted to negotiate with every licensee.[59] The shrink wrap thus serves as a shorthand for essentially those terms negotiated in the custom software context, in which transaction costs are low enough to facilitate the face to face bargaining that is impractical in the mass market.[60]

In the middle of the spectrum are contracts that are signed by the licensee, but are generally not negotiated.[61] This approach characterizes high-volume software that has not yet reached the commodity status of mass-market PC software, but which is fairly standardized and somewhat more expensive than shrink wrap products.[62]

To the extent that software license agreements track the provisions of the Act, they may simply provide another enforcement mechanism for an aggrieved licensor. Should a licensee breach the agreement, a licensor could choose to sue for breach of contract instead of or in addition to copyright infringement.[63] In choosing

[58] Shrink wrap licensing has evolved over the years. Generally, shrink wrap agreements accompany mass market software such as WordPerfect, Windows, and Lotus 1-2-3. Originally, shrink wrap licenses were visible to the customer before they purchased the product. The product would come in a box wrapped in plastic and the license would be visible beneath the plastic. *See* Samuelson et al., *supra* note 15, at 2318 n.26, 2373. Now, such licenses often are not visible, but rather are part of the terms listed inside the box. Thus, the customer does not become aware of the licensing terms until after purchase. Finally, some software providers distributing object code on networks like the Internet, as well as many on-line database providers, have begun using electronic license agreements.

[59] *See generally* Robert P. Merges, A Comparative Look at Property Rights and the Software Industry 3 (1995) (unpublished manuscript, on file with author) (stating prepackaged software accounts for about 70% of industry revenues, but estimates vary widely).

[60] *See supra* note 56. *But see* Rice, *supra* note 56, at 177-78 (arguing that software vendors use shrink wraps in an attempt to avoid the Act's first sale restrictions and achieve market protection not existing if program copies are sold, and noting that custom agreements are not necessarily characterized by the inclusion of prohibitions against reverse engineering). *See also infra* note 201 and accompanying text (discussing first sale doctrine in more detail).

[61] *See, e.g.*, IBM CUSTOMER AGREEMENT, *supra* note 10.

[62] For example, the *IBM Customer Agreement, id.*, is used to license such high volume software as AIX Version 3 and OS/400, which run on mid-range systems, as well as IBM's mainframe software.

[63] *But see infra* Part III (noting that breach of contract action alleging same facts as copyright infringement action may be preempted under Act); 3 MELVILLE B. NIMMER

between a suit in contract or copyright, the licensor's primary consideration likely would be an economic one—to maximize its monetary return.[64] However, a breach of contract action may be problematic if it attempts to enforce contractual provisions that are more restrictive than and potentially at odds with the policies of the Act.[65]

3. *Licensors' Rationale for Inclusion of Restrictive Terms.* In assessing whether breach of contract actions based upon these particular provisions are preempted by the Copyright Act, it is necessary first to consider why software providers include license terms that are more restrictive than the rights that the Act would

& DAVID NIMMER, NIMMER ON COPYRIGHT, §10.15[A], at 10–122 to -23, §14.02[B], at 14–23 n.53.1 (1995) (plaintiff may be required to elect contract or copyright remedy). Note also (1) that by choosing solely a breach of contract remedy, the plaintiff could avoid federal court jurisdiction; and (2) that many breach of contract actions involving software are based on traditional contract doctrines such as breach of warranty or the validity of limitations of remedy rather than on copyright-type claims. *See, e.g.*, Step-Saver Data Sys., Inc. v. Wyse Tech., 939 F.2d 91, 105–07 (3d Cir. 1991) (holding that warranty disclaimer and limitation of remedy on shrink wrap not part of parties' agreement but evidence insufficient to show breach of implied warranty of merchantability); Arizona Retail Sys., Inc. v. Software Link, Inc., 831 F. Supp. 759, 761–66 (D. Ariz. 1993) (discussing disclaimer of implied warranties); Colonial Life Ins. Co. of Am. v. Electronic Data Sys. Corp., 817 F. Supp. 235, 239–44 (D.N.H. 1993) (addressing limitation of damages and limitation of warranties provisions).

[64] Most simply stated, if the probability of success (P) in bringing a suit based on a single claim of either breach of contract or copyright infringement multiplied by the expected damage award (D) is greater than the cost of suit (C), then a risk-neutral plaintiff will choose to bring suit: (P X D) > C. In choosing between copyright and contract causes of action, the plaintiff will opt for the cause of action with the highest expected return: (P X D) - C. The decision becomes more complex as the plaintiff considers adding additional causes of action to the complaint. For example, in deciding whether or not to join a breach of contract action with a copyright infringement one or vice versa, the plaintiff would examine the marginal cost of bringing that additional claim and the marginal expected gain, bringing the additional claim if the marginal gain exceeds the marginal cost. One commentator argues that a "reverse engineering prohibition makes breach of contract an independent basis of potential liability" and characterizes breach of contract as a strict liability alternative to trade secret or copyright litigation because proof of breach may be treated as some evidence of misappropriation or infringement; and the use of breach of contract as evideuce of misappropriation or infringement opens the door to "statutory damages and attorneys' fees under the Copyright Act, and makes available *ex contractu* remedies, including punitive damages, for trade secret misappropriation." Rice, *supra* note 17, at 559–61. However, as a practical matter, for a breach of contract action in itself to be viable, the plaintiff must prove damages. Thus, it would be irrational for a software vendor to incur litigation costs in a breach of contract action based on reverse engineering when no damage may be shown as, for example, when the software is reverse-engineered to uncover ideas.

[65] *See supra* text accompanying notes 39–53.

otherwise confer. Essentially, software is fundamentally different from hard copy works in that a user can receive the benefit of software's functionality simply by running the object code.[66] Additionally, the software provider has an interest in protecting its usually substantial investment in the source code[67] as well as the trade secrets contained therein.[68]

At one extreme, if courts were to interpret copyright law to allow a blanket decompilation right, software providers would be extremely unlikely to rely solely upon copyright law to protect their investments. Such a legal rule could encourage large-scale infringement because decompilers could take modules of decompiled code and place them in other products without the copyright owner ever knowing of the infringement.[69] The courts, however,

[66] *Cf.* Jerome H. Reichman, *Electronic Information Tools—The Outer Edge of World Intellectual Property Law*, 17 U. DAYTON L. REV. 797 (1992) (discussing the need to conceptualize electronic information processing as a special kind of industrial tool that may not fit within the classical intellectual property scheme); Samuelson et al., *supra* note 15, at 2315–20, 2364–70 (describing the unique nature of software and suggesting it could better be handled by a legal system that takes a market-oriented approach to protection of technological know-how).

[67] Of course, it is commonplace for all authors to contend that they must have copyright protection as a mechanism to recoup their investments. Software authors are no different. As already mentioned, *see supra* note 27, the mere fact of investment is not sufficient to confer copyright status on a work. However, given the ease with which software may be duplicated and distributed without detection, *see supra* note 35, software providers often seek to buttress the protection that may be afforded by copyright or to create such protection when none is granted by law. *See, e.g., infra* notes 313–16 and accompanying text.

[68] *See* Samuelson et al., *supra* note 15, at 2342–43 (noting that trade secrecy law cannot protect behavior or other information evident on face of mass-marketed prodnct and that trade secrecy law regards reverse engineering as fair means of acquiring trade secrets). It is an open question whether trade secrets are preserved in mass-distributed products as the breadth of their distribution may undercut the licensor's argument that it has made reasonable efforts to maintain secrecy. *See* Barr-Mullin, Inc. v. Browning, 424 S.E.2d 226, 229 (N.C. Ct. App. 1993); *see also* Trandes Corp. v. Guy F. Atkinson Co., 996 F.2d 655, 663–64 (4th Cir.), *cert. denied* 114 S. Ct. 443 (1993); ISC-Bunker Ramo Corp. v. Altech, Inc., 765 F. Supp. 1310, 1323 (N.D. Ill. 1990).

[69] *See* Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 978, 1026–27 (1993). Miller notes that

> [t]he simple truth is that permitting decompilation allows a second comer to create a market substitute and reap the benefits of a successful program after others have incurred the risk and expense of its development—an especially inappropriate result given the extraordinary discrepancy between the cost of creating the software and the cost of duplicating it. If an exemption from copyright is permitted, the decompiler will be able to *reproduce* the entire program of a competitor—appropriating in one relatively simple procedure what may represent years of creative effort and investment—and then electronically mas-

have not adopted this extreme position, but rather have adopted a narrow rule as to when decompilation is permitted.[70] Licensors, though, continue to include provisions against decompilation in their license agreements. These provisions seem overbroad because there are often legitimate, noninfringing reasons wliy one might seek access to tle underlying source code. Sucli access may be required to understand underlying ideas, to produce a compatible product, or to fix bugs.[71]

---

> sage the copy until every trace of that illicit reproduction is obscured. . . . Freedom to decompile . . . eliminates any incentive to produce an innovative or creative expression of one's own, thereby debilitating one of the basic objectives of the copyright regime.

*Id.* (citations omitted). *But see* Jolinson-Laird, *supra* note 43, at 843–44, 900–01. Johnson-Laird contends that because software reverse engineering is difficult and time-consuming, it represents a remedy of last resort to obtain infonnation, and therefore such reverse engineers would not represent a real threat to copyright owners:

> If a thief merely shuffles the original source code, the resulting object code will still contain numerous indicia of the software's origin . . . . If the thief . . . both shuffles and heavily modifies the source code to disguise it, almost certainly the program either will no longer work at all, or it will be so unreliable that the thief will not find many buyers. . . . A software thief wlio lives by reverse engineering will die a death in the marketplace because of reverse engineering. The costs of reverse engineering, taken across the product's entire life, usually five to seven years, will invariably be higher than software written *de novo.*

*Id.* at 900-01; *see also* Leaffer, *supra* note 56, at 1096 (stating that reverse engineers are unlikely to have "free ride" and the risk of massive infringement is low since (1) decompilation is technically difficult and costly; and (2) developers are likely to find technical ways in which to increase decompilers' costs by making decompilation process more difficult); Rice, *supra* note 22, at 1188–89 (contending that time and effort to decompile object code and reconstruct source code approaches investment of copyright owner). Finally, Professor Samuelson and her colleagues argue that

> [d]ecompilation and reimplementation of an interface are sufficiently arduous processes that they necessarily entail significant delay, leaving the firm whose interface infonnation is being appropriated a nontrivial head-start in the marketplace. As a consequence, we do not perceive that market failure is presently occurring when software developers appropriate interface infonnation.

Samuelson et al., *supra* note 15, at 2403–04. However, prohibitions against reverse engineering guard against future technological advances tliat may make decompilation easier. *See id.* at 2404 ("However, if decompilation technology improves substantially, there may be a need to consider whether use of another firm's internal interface information should be blocked for a market-preserving period of time, or perhaps be subjected to a right of compeusation.").

[70] *See infra* Section II(A).

[71] *See* McManis, *supra* note 23, at 30–31 (listing (1) learning or teaching "new programming techniques, either for purely academic reasons or in order to create a commercially-marketable software product"; (2) debugging; (3) customizing, including modifying a program to run with another prograin or with particular hardware; and (4) developing competing or noncoinpeting products as possible purposes underlying decoinpilation). A "compatible" product may simply interoperate with the decompiled product, requiring the user to have the decompiled product installed in order to run the new product. Often,

The tension among these three alternatives—blanket decompilation rights under the Act, limited decompilation rights under the Act, and contractual prohibitions against decompilation—arises because providers of mass-market software are unable to distinguish between purchasers who would decompile for a legitimate reason and those who would do so for infringing purposes.[72] Thus, the provider includes a blanket prohibition against decompilation in all cases in an attempt to ensure a return on its investment in the development of the code. The contractual prohibition lessens the provider's need to (1) monitor the market for decompilation tools to assess the probability that decompilation of its software is technologically feasible; (2) expend resources on making decompilation of its program more difficult; and (3) expend resources monitoring legislative efforts to address decompilation.[73] However, in light of such an enforceable contractual provision, the party who wishes to decompile is then forced to contact and negotiate with the software provider for permission.

Depending on the parties involved, negotiating with the software provider to modify standard terms may be expensive from a transaction-cost perspective or simply an impractical alternative from a business perspective. The ultimate issue then might best be framed as whether courts, in addressing breach of contract issues that directly implicate the policies of the Act, should formulate a

the new product may augment the demand for the decompiled product (e.g., the availability of WordPerfect for Windows may enhance the demand for Windows). See the *Atari* and *Sega* cases, discussed *infra* Section II(A), for an example of this type of interoperable compatibility. Another type of compatible product is one that supplants market demand for the decompiled product by being functionally equivalent (for example, if OS/2 could run all Windows applications, it might decrease the demand for Windows). The copyright owner is more likely to object to this latter type of compatibility—often called cloning—as it decreases demand for its product and is more likely to have involved appropriation of protected expression. Theoretically, however, a decompiler could use only functional specifications obtained through decompilation (i.e., not code) to create the clone.

[72] Of course, the same assertion could be made with respect to hard copy publications. Publishers cannot know in advance which of the purchasers of a book will copy and resell it or otherwise infringe the exclusive copyright rights. However, large-scale hard copy infringement is much more easily detected and policed than large-scale soft copy infringement. *See supra* note 35.

[73] *See infra* note 164 and accompanying text; Samuelson et al., *supra* note 15, at 2404 (noting that if decompilation technology improves, there may be a need to consider legislation to protect market lead time or provide compensation to copyright holder).

legal rule that places particular transaction costs on the licensor or licensee.[74]

## II. The Judicial Response to Decompilation Provisions

Although there has been some academic commentary on the subject,[75] the caselaw addressing the relationship between copyright and contract is fairly sparse, especially in the electronic context. Two courts, in ground-breaking cases, have directly addressed the extent to which a licensee may decompile a copyrighted work,[76] while another has attempted to draw a boundary line between contract and copyright.[77]

[74] Another way to frame the issue would be whether or not the public's limited right to decompile should be alienable or inalienable.

[75] McManis, *supra* note 23 (arguing that federal copyright and patent law preempt contractual provisions against reverse engineering of publicly distributed programs); Rice, *supra* note 17, at 1151 (arguing that contractual reverse engineering prohibitions are preempted in both negotiated and nonnegotiated contracts); Rice, *supra* note 22 (repeating contentions contained in *id.* and responding to arguments set forth in Miller, *supra* note 69); Samuelson et al., *supra* note 15, at 2390 n.329 (noting that debate exists in United States as to when developers can legitimately enter into contracts prohibiting decompilation or disassembly); S. Carran Daughtrey, Note, *Reverse Engineering of Software for Interoperability and Analysis*, 47 VAND. L. REV. 145, 181–86 (1994) (suggesting that Congress either amend Copyright Act or create *sui generis* law regarding reverse engineering to alleviate legal uncertainty); Ramona L. Paetzold, Comment, *Contracts Enlarging a Copyright Owner's Rights: A Framework for Determining Unenforceability*, 68 NEB. L. REV. 816, 831–33 (1989) (suggesting that contracts eliminating fair use rights be evaluated for enforceability under rule of reason standard and opining that "contract provisions prohibiting decompilation or reverse engineering of mass-marketed computer software should be preempted"); *cf.* Allan M. Soobert, *Legitimizing Decompilation of Computer Software Under Copyright Law: A Square Peg in Search of a Square Hole*, 28 J. MARSHALL L. REV. 105 (1994) (proposing amendment to copyright law to provide for limited decompilation right); Christopher W. Hager, Note, *Apples & Oranges: Reverse Engineering as a Fair Use After* Atari v. Nintendo *and* Sega v. Accolade, 20 RUTGERS COMPUTER & TECH. L.J. 259 (1994) (discussing statutory codification of and congressional action on the extension of the common law doctrine of "fair use" to copyright law).

[76] Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992); Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832 (Fed. Cir. 1992); *see also* DSC Comm. Corp. v. DGI Technologies, Inc., No. 3:94-CV-1047, 1995 WL 526429, at *4 (N.D. Tex. Sept. 1, 1995) (adopting *Sega* reasoning in holding disassembly of firmware may be fair use). This Article concentrates on the *Atari* and *Sega* cases because they set forth the principles on which other courts, such as the *DSC* court, are beginning to rely.

[77] Vault Corp. v. Quaid Software Ltd., 847 F.2d 255 (5th Cir. 1988); *see infra* Section III(B)(1).

### A. *Decompilation as Fair Use:* Atari *and* Sega

*Atari Games Corp. v. Nintendo of America, Inc.*[78] and *Sega Enterprises Ltd. v. Accolade, Inc.*[79] are the seminal cases addressing the extent to which one party may decompile the code of a competitor in order to produce an interoperating product. Both cases arose in the home video game context. Most home video games are sold on cartridges that are inserted into a console hooked up to a video display screen.[80] A user normally purchases one console and then a variety of game cartridges for it.[81] Game cartridges are produced by both console providers and independent developers. Cartridge sales, a lucrative part of the home entertainment industry, are likely to generate more revenue if the cartridges run on the most popular console as well as on other manufacturers' consoles. The dominant console provider that is also a game cartridge producer has a revenue interest in controlling both the number of games available for use on its console and the identity of the providers of those games. Less dominant console providers and independent game developers have a revenue interest in assuring that their games run on the console of the dominant manufacturer.

In *Atari*, Nintendo, which had wrested control of the console market from Atari,[82] had written a lock (the 10NES program) into its Nintendo Entertainment System (NES) consoles.[83] The lock prevented the NES from accepting unauthorized game cartridges, and could only be opened by a softkey contained in a game cartridge.[84] Nintendo would license third-party game devel-

[78] 975 F.2d 832 (Fed. Cir. 1992).

[79] 977 F.2d 1510 (9th Cir. 1992).

[80] More specifically, the cartridge containing the game program is inserted into a base unit console. The console is connected to the television, making the television into a video screen and audio speaker for the game's instructions.

[81] Popular games designed for the Sega Genesis system include Ecco the Dolphin, Sonic the Hedgehog, Aliens 3, and Aladdin; for Nintendo, they include Zelda, Mario Bros., and Donkey Kong. Many, if not most, games run on both systems (e.g., Mortal Kombat).

[82] *Cf.* Neil Gross, *Here Comes the Super Mario Bros.*, BUS. WK., Nov. 9, 1987, at 138 (stating Nintendo holds 90% of home market and 70% of U.S. video game market).

[83] Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 836 (Fed. Cir. 1992).

[84] The Court described the softkey mechanism as follows:

> The console contains a "master chip" or "lock." Authorized game cartridges contain a "slave chip" or "key." When a user inserts an authorized cartridge into a console, the slave chip in effect unlocks the console; the console detects a coded message and accepts the game cartridge. When a user inserts an unau-

opers to develop cartridges for the NES, but closely guarded the specification and code for unlocking the console.[85] Atari, after failing to break the code by analyzing the chips in Nintendo's console and cartridges, finally became a Nintendo licensee.[86] However, under the license, Atari still did not have access to the 10NES code: "Nintendo would take Atari games, [put] them in cartridges containing the 10NES [code], and [then] resell them to Atari for distribution to the general public."[87] Additionally, the license limited Atari to production of five new games for the NES per year.[88] It also prohibited Atari from marketing those games on other consoles for two years.[89]

Atari sought to escape the license restrictions by finding an alternative means of access to the 10NES. By making false representations to the Copyright Office, it obtained a copy of the 10NES source code.[90] It used this copy of the source code to correct errors in the transcription of the code it had made through decompilation.[91] Next, it developed a program called Rabbit to unlock Nintendo's console.[92] Despite the fact that the line by line instructions of Rabbit were different from those of the 10NES, the court held that Atari had likely infringed copyrightable elements of the 10NES.[93]

> thorized cartridge, the console detects no unlocking message and refuses to operate the cartridge.

*Id.*

[85] *Id.* (Prior to becoming a Nintendo licensee, Atari attempted to analyze the code by (1) "monitoring the communication between the master and slave chips . . . [; and (2)] analyzing the chips themselves . . . [by] chemically peel[ing] layers from the NES chips to allow microscopic examination of the object code.").

[86] *Id.*

[87] *Id.*

[88] *Id.*

[89] *Id.*

[90] *Id.* (Atari claimed to be a defendant in a pending infringement case and in need of a copy for litigation; in fact, no such case existed.).

[91] *Id.* (Atari transcribed 10NES object code into a handwritten listing of ones and zeros, and used the Copyright Office copy to correct errors in transcription. "The Copyright Office copy facilitated Atari's replication of the 10NES object code.").

[92] *Id.*

[93] *Id.* at 845. The Rabbit program contained elements of 10NES unnecessary to unlocking the console, including unnecessary instructions found in an early version of 10NES:

> The unnecessary instructions in the Rabbit program suggest copying, not independent creation. . . . Atari's efforts to reverse engineer the 10NES chip to learn the ideas in the program will not alone support a copyright infringement claim. To the extent, however, Nintendo is likely to show misappropriation and copying of the unauthorized Copyright Office copy, it is likely to succeed on

However, the court, after citing the traditional prohibition against granting copyright protection to ideas, noted in dicta that "[a]n author cannot acquire patent-like protection by putting an idea, process, or method of operation in an unintelligible format and asserting copyright infringement against those who try to understand that idea, process, or method of operation."[94] The court also cited legislative history stating that Congress intended "that courts should adapt the fair use exception to accommodate new technological innovations."[95]

Fair use is an equitable defense to an infringement claim, nnder which public policy grounds excuse conduct that would otherwise be considered infringing.[96] In assessing a clain of fair use, a court weighs the nonexclusive statutory fair use factors:

> (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes; (2) the nature of the copyrighted work; (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and (4) the effect of the use upon the potential market for or value of the copyrighted work.[97]

Although the Supreme Court has emphasized the necessity of weighing all four factors in light of the Act's statutory purpose be-

---

> the merits of its infringement claim. Alternatively, Nintendo is likely to prove substantial similarity between the Rabbit and 10NES programs sufficient to support its infringement claims.

*Id.*

[94] *Id.* at 842.

[95] *Id.* at 843; *see also infra* text accompanying notes 152-53 (discussing theoretical underpinning of fair use).

[96] *See* 3 NIMMER & NIMMER, *supra* note 63, § 13.05, at 13-152 to -153; *see also* 17 U.S.C. § 107 (1994) (listing "purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research" as examples of types of uses that may be privileged as fair).

[97] 17 U.S.C. § 107 (1994). Historically, the first and fourth factors have been viewed as the most important. *See* Stewart v. Abend, 495 U.S. 207, 238 (1990) (stating that "[t]he fourth factor is the 'most important . . . fair use factor'" (quoting 3 NIMMER & NIMMER, *supra* note 63, § 13.05[A], at 13-81)); Original Appalachian Artworks v. Topps Chewing Gum, 642 F. Supp. 1031, 1035 (N.D. Ga. 1986) (asserting that first and fourth factors are closely related); Roxana Badin, Comment, *An Appropriate(d) Place in Transformative Value: Appropriation Art's Exclusion from* Campbell v. Acuff-Rose Music, Inc., 60 BROOK. L. REV. 1653, 1678-79 (1995) (stating that the Court weighs the first and fourth factors "heavily in its fair use determinations"). As noted in the text, these factors are nonexclusive. Market evidence, *see infra* text accompanying notes 151-52, should therefore be admissible as relevant to the analysis of the first and/or fourth factors or as relevant in its own right.

fore making a fair use determination,[98] the *Atari* court focused only on the second: "When the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying."[99] The court suggested a relatively narrow rule as to when decompilation would be considered fair use: "The Copyright Act permits an individual in *rightful* possession of a copy of a work to undertake *necessary* efforts to understand the work's ideas, processes, and methods of operation. This permission appears in the fair use exception to copyright exclusivity."[100] Moreover, according to the court, a fair use reproduction could "not exceed what is necessary to understand the unprotected elements of the work."[101] Even though Atari may have breached its license agreement with Nintendo, there ostensibly was no breach of contract action involved in the case, obviating the need for the court to address the issue of whether particular license provisions would be enforceable or preempted by the Act.[102]

*Sega*[103] presented its court with a fact pattern somewhat similar to that in *Atari.* Accolade, an independent game cartridge manufacturer, sought to enable its cartridges to run on Sega's Genesis console. Because Sega's standard terms required that it be the exclusive manufacturer of all games, Accolade did not want to license the console specifications.[104] Instead, it purchased a Genesis console and three Sega game cartridges.[105] It decompiled and experimented with the console and cartridges to discover the interface specifications for the Genesis console.[106] The end result of

[98] Campbell v. Acuff-Rose Music, Inc., 114 S. Ct. 1164, 1170–71 (1994).

[99] *Atari,* 975 F.2d at 843.

[100] *Id.* at 842 (emphasis added).

[101] *Id.* at 843. The court also stated that "[f]air use to discern a work's ideas, however, does not justify extensive efforts to profit from replicating protected expression. . . . [F]air use in intermediate copying does not extend to commercial exploitation of protected expression." *Id.*

[102] Despite the dicta cited *supra* text accompanying notes 94–95 and 99–100, Atari still lost the case. *See supra* note 93 and accompanying text.

[103] Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).

[104] *Id.* at 1514.

[105] *Id.* at 1514–15.

[106] Specifically,

> [Accolade] wired a decompiler into the console circuitry, and generated printouts of the resulting source code. Accolade engineers studied and annotated the printouts in order to identify areas of commonality among the three game programs. They then loaded the disassembled code back into a computer, and experimented to discover the interface specifications for the Genesis console by

this process was a development manual containing the interface specifications.[107] Accolade used this manual to modify a number of its existing programs that already ran on other manufacturers' consoles to be compatible with the Genesis system.[108] Later, it wrote a new game program to run on the Genesis system.[109]

The *Sega* court held that the intermediate copy which was necessarily created in the decompilation process infringed Sega's exclusive rights under the Act.[110] However, the court found this infringement to be excused under the fair use doctrine.[111] In contrast to *Atari*, this court analyzed exhaustively the four statutory fair use factors. It found that although Accolade's ultimate goal was the commercial release of Genesis-compatible games, its direct purpose in copying Sega's code was to study the functional requirements for Genesis compatibility.[112] Accolade's purpose was thus held to be only indirectly commercial, enabling it to enter, rather than usurp, the market for Genesis compatible games.[113]

modifying the programs and studying the results.

*Id.* at 1515.

[107] *Id.* ("According to the Accolade employees who created the manual, the manual contained only functional descriptions of the interface requirements and did not include any of Sega's code.").

[108] *Id.* at 1515-16. For example, in 1990 Accolade released Ishido, a game for use with the Genesis console, which it had "originally developed and released for use with the Macintosh and IBM personal computer systems." *Id.* at 1515. Accolade released five more games for use with the Genesis III system in 1991, four of which had been written originally to run on other hardware systems. *Id.* at 1516.

[109] *Id.* at 1516 (noting Mike Ditka Power Football was developed initially for use with Genesis system).

[110] *Id.* at 1519 ("[W]e hold that intermediate copying of computer object code may infringe the exclusive rights granted to the copyright owner in section 106 of the Copyright Act regardless of whether the end product of the copying also infringes those rights.").

[111] *Id.* at 1520 ("Where there is good reason for studying or examining the unprotected aspects of a copyrighted computer program, disassembly for purposes of such study or examination constitutes a fair use.").

[112] *Id.* at 1522.

[113] *Id.* at 1523. The court did not enunciate a test for making a principled distinction between "direct" and "indirect" commercial uses. However, it did note that

> [t]here is no basis for assuming that Accolade's "Ishido" has significantly affected the market for Sega's "Altered Beast," since a consumer might easily purchase both; nor does it seem unlikely that a consumer particularly interested in sports might purchase both Accolade's "Mike Ditka Power Football" and Sega's "Joe Montana Football," particularly if the games are, as Accolade contends, not substantially similar.

*Id.* Video game users may disagree with this statement. Particularly in the case of standardized games like baseball, football, and basketball, a user is unlikely to purchase more

Thus, the first and fourth statutory factors weighed in favor of Accolade.[114] The second statutory factor, the nature of the copyrighted work, also weighed in Accolade's favor because the unprotected ideas contained in object code may not be comprehensible in the absence of decompilation.[115] Finally, despite the fact that Accolade engaged in wholesale copying of Sega game cartridges, tipping the third factor in Sega's favor, the court sustained the fair use defense.[116]

As in *Atari*, the *Sega* court enunciated a seemingly narrow fair use decompilation exception:

> We conclude that where disassembly is the *only* way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a *legitimate reason* for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law.[117]

### B. *An Evaluation of* Sega *and* Atari

Both the *Atari* and *Sega* courts used the terms "reverse engineering," "decompilation," and "disassembly"[118] interchangeably. Arguably, however, this usage is incorrect.[119] As a general rule, reverse engineering refers to the process of "starting with the known product and working backward to divine the process which

than one version.

[114] *See supra* note 97 (first and fourth factors often stated to be most important in fair use inquiry).

[115] *Sega*, 977 F.2d at 1525-26.

[116] *Id.* at 1526. The court explained that the third factor weighed against Accolade because "Accolade disassembled entire programs written by Sega," but went on to state that "[t]he fact that an entire work was copied does not, however, preclude a finding of fair use." *Id.*

[117] *Id.* at 1527-28 (emphasis added). *See generally* Rice, *supra* note 22, at 1181-95 (criticizing *Sega* limitations and advocating broader fair use exemption).

[118] Disassembly refers to the process of decompiling object code to obtain assembler code that is closer to machine language than source code. Decompilation is a broader term, encompassing translations into assembler as well as other higher level languages. This Article will continue to use the term decompilation.

[119] Academic commentary, as well as the caselaw, often has used the terms "reverse engineering," "decompilation," and "disassembly" interchangeably. However, IBM, in its license agreements, has included provisions against decompilation and disassembly, not reverse engineering. *See* IBM CUSTOMER AGREEMENT, *supra* note 10. This may reflect its understanding of the distinctions among the three terms and its belief that it may not prohibit all forms of reverse engineering, but simply two of those forms—decompilation and disassembly.

aided in its development or manufacture."[120] Decompilation is merely one method of reverse engineering; it is not synonymous with "reverse engineering." This is not a matter of mere semantics because there may be a relevant legal distinction between license agreements that prohibit reverse engineering and those that prohibit decompilation. The former absolutely prohibit the licensee from attempting to understand how the product works, while the latter merely remove one method a licensee could employ to reverse-engineer.[121]

Both courts also seemed to assume that the only context in which their enunciated limited fair use defense would shelter decompilation would be when the alleged infringer decompiled an operating system[122] to understand its interfaces.[123] An applica-

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

[121] *See, e.g.*, Samuelson et al., *supra* note 15, at 2317–18, 2393 (describing "black box" technique in which programmer runs software with variety of inputs, noting its behavior so that second programmer may write new code producing same behavior as first program); *see also* Johnson-Laird, *supra* note 43, at 846 (describing the four ways to perform reverse engineering: "(1) read about the program; (2) observe the program in operation by using it on a computer; (3) perform a static examination of the individual computer instructions contained within the program; or (4) perform a dynamic examination of the individual computer instructions as the program is being run on a computer"). However, note that Johnson-Laird argues that documentation is always incomplete, and running the program and watching what it does is nothing more than using it. *Id.* But see *infra* text accompanying note 263, quoting *Bonito Boats* for the principle that the availability of alternative means of reverse engineering did not justify the state's banning of one particular means. Note that under the proposal set forth *infra* Section IV(A), the distinction between the terms "reverse engineering" and "decompilation" have substantive relevance. The licensor's business justification for its particular licensing provisions would be relevant to the question of predatory conduct, with a provision against decompilation less likely to be evidence of predation than one against reverse engineering.

[122] IBM DICTIONARY OF COMPUTING 399 (10th ed. 1993) [hereinafter IBM DICTIONARY] (Operating system is defined as "[s]oftware that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible."); Samuelson et al., *supra* note 15, at 2376–77 ("Operating systems are the foundational layer on which applications programs must run, and with which they must be compatible."). Examples of some operating systems include MS-DOS, MVS (OS/390), VM, AIX, and OS/400.

[123] *See, e.g.*, Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520 (9th Cir. 1992) ("The need to disassemble object code arises, if at all, only in connection with operations systems, system interface procedures, and other programs that are not visible to the user when operating . . . ."). The *Atari* court was less specific, but noted that the fair use right "did not give Atari more than the right to understand the 10NES program and to distinguish the protected from the unprotected elements of the 10NES program." Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 844 (Fed. Cir. 1992). The 10NES program was basically the interface between the console and the cartridge.

tion[124] programmer must understand those parts of the operating system that the application must utilize in order to execute successfully on the particular operating system.[125] Although neither court undertook to define an "interface,"[126] both assumed, *Atari* implicitly and *Sega* explicitly,[127] that interfaces are functional as-

[124] IBM DICTIONARY, *supra* note 122, at 23 (Application is defined as "(1) The use to which an information processing system is put; for example, a payroll application, an airline reservation application, a network application. (2) A collection of software components used to perform specific types of user-oriented work on a computer . . . ."); *see also* Samuelson et al., *supra* note 15, at 2377 (The line between operating system and applications program is unclear, and chauges with time. Recently, the trend has been to include more functionality in operating systems: "[F]or example, the widespread utility of and acceptance of graphical user interfaces led to their migration into operating systems, such as Microsoft's Windows."). Examples of application programs include WESTLAW, WordPerfect, and Lotus 1-2-3.

[125] The need to decompile for compatibility also may arise with respect to applications—e.g., developers likely would be unable to write enhancements/speed-ups to existing applications without utilizing application functions that can be ascertained only through access to a published interface specification or the code itself. Thus, although this Article concentrates on operating systems, the problem also may occur with applications, and the principles enunciated herein may be extended to that context.

[126] The courts may have avoided defining an interface because of the lack of any accepted definition within the computer scieuce community. Generally, an iuterface may be defined as "1: a surface forming a common boundary of two bodies . . . 2 a: the place at which independent systems meet and act on or commuuicate with each other . . . ; *broadly*: an area in which diverse things interact . . . b: the means by which interaction or communication is effected at an interface." WEBSTER'S NINTH NEW COLLEGIATE DICTIONARY 602 (1984). More specifically, in the computer context, an interface may be defined as "(1) [a] shared boundary between two functional units, defincd by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions . . . (2) Hardware, software, or both, that links systems, programs, or devices . . . ." IBM DICTIONARY, *supra* note 122, at 294. From the industry's perspective, the problem with any definition of interface is that it threatens to swallow the whole—each line of code in some sense "connects" to something else, opening the door to the argument that the program itself is nothing more than the sum of unprotected interfaces.

[127] Both cases involved communication between the cartridge and console, which seems to fit within both definitions of "interface" set forth *supra* note 126. Discussing the distinction between protected and unprotected elements of a program, the *Sega* court stated that

> disassembly of the object code in Sega's video game cartridges was necessary in order to understand the functional requirements for Genesis compatibility. The interface procedures for the Genesis console are distributed for public use only in object code form, and are not visible to the user during operation of the video game program. . . . If disassembly of copyrighted object code is *per se* an unfair use, the owner of the copyright gains a *de facto* monopoly over the functional aspects of his work—aspects that were expressly denied copyright protcction by Congress.

Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1525-26 (9th Cir. 1992); *see also* Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 838-39 (Fed. Cir. 1992) (discussing process of distinguishing between idea and expression and types of subject matter covered pects of a system and thus not protected by copyright.[128]

Additionally, both courts implicitly recognized that software providers have legitimate reasons for seeking to prevent decompilation. Both stated that a fair use defense would not shelter decompilation, even when such an activity was carried out solely to discover ideas, if alternative means of obtaining the information were available.[129] But if the rationale for the fair use exception were simply to allow access to ideas, it seems irrelevant whether or not alternative means of obtaining those ideas existed. It seems more likely that the courts recognized that because decompilation discloses much more information—the underlying source code expression or a variant thereof—than merely the ideas contained in the program, software providers have a legitimate interest in trying to prevent or at least limit this activity.[130] Assuming that this is so, the question then becomes one of defining the limits of that interest and determining whether it is best protected through legal rules, market transactions, or some combination of both.

*Sega* and *Atari* steer a middle course in defining those limits. The courts stated that fair use would shelter one (1) in rightful possession of a copy of a computer program who (2) engages in decompilation to gain access to ideas for a "legitimate reason" (3) when no other means of access exists.[131] Thus, when the market has functioned to provide alternative means of access to ideas, no right to decompile exists, and when it has not, the law will intervene to protect the policy of fostering the dissemination of ideas.[132] The contours of this legal solution are still unclear. This uncertainty, the high costs of enforcing the legal rule by distinguishing between legitimate and illegitimate cases of decompilation, and the high risk of error involved in that assessment suggest that an alternative approach might be preferable.

by copyright and patent).

[128] Samuelson et al., *supra* note 15, at 2402 ("[I]nternal interface compilations [are] unprotectable by copyright law because they are industrial compilations of applied know-how, information equivalents to the gears that allow physical machines to interoperate.").

[129] *Sega*, 977 F.2d at 1527-28; *Atari*, 975 F.2d at 843-44.

[130] *See supra* text accompanying notes 66-69.

[131] *Sega*, 977 F.2d at 1527-28; *Atari*, 975 F.2d at 842-44.

[132] *Cf.* Gordou, *supra* note 16, at 1601 (stating that doctrine of fair use "permit[s] uncompensated transfers that are socially desirable but not capable of effectuation through the market").

Simply put, the *Sega/Atari* rule invites abuse. While *Atari* indicated that decompilation could only be excused if it were limited to those parts of the program containing unprotected elements,[133] *Sega* did not contain such a limitation. Technically, it is questionable whether such a limitation is feasible. It is extremely difficult to know what area of the object code has to be decompiled to obtain the required information.[134] Thus, the decompiler is likely to access substantial protected expression during the decompilation process.

Of course, under the approach of *Sega* and *Atari*, the decompiler may use only the ideas and must discard the protected expression obtained during the decompilation process. It is questionable, however, whether this is in fact what a decompiler actually does. By its nature, software is readily adaptable to a variety of purposes, and copying is difficult to detect.[135] It is a reasonably simple matter for a decompiler—particularly a large software firm with a variety of applications—to misappropriate protected expression by adapting it for use in another program.[136] At the same time, it would be extremely difficult for the copyright holder to detect such infringement.

These effects are exacerbated by the decision of the U.S. Court of Appeals for the Second Circuit in *Computer Associates International, Inc. v. Altai, Inc.*[137] In *Altai*, which involved allegations of infringement of nonliteral elements of the plaintiff's software,[138] the court acknowledged the conceptual difficulty in applying the idea/expression distinction to software.[139] After review-

[133] *Atari*, 975 F.2d at 843 ("The fair use reproductions of a computer program must not exceed what is necessary to understand the unprotected elements of the work.").

[134] *See generally* Johnson-Laird, *supra* note 43.

[135] Of course, the same could be said of hard copy publications. The difference is one of degree. The ease with which software may be modified and replicated without retaining any indicia of original origin is an order greater than that of hard copy works. *See supra* note 35.

[136] *See supra* note 69.

[137] 982 F.2d 693 (2d Cir. 1992); *see also* Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 (1st Cir. 1995) (limiting *Altai* to cases involving nonliteral infringement and holding menu command hierarchy of Lotus 1-2-3 uncopyrightable as "method of operation"), *aff'd per curiam by an equally divided Court*, 64 U.S.L.W. 4059 (U.S. Jan. 16, 1996) (No. 94-2003).

[138] Nonliteral elements are those parts of the program that are not the written code, such as the structure of the modules and parameter lists. *See generally Altai*, 982 F.2d at 696–98, 702–03.

[139] *Id.* at 705–06; *see also supra* note 52 and accompanying text.

ing authority in both the hard and soft copy contexts, the court adopted a three-part test for distinguishing idea from expression. The first step, based on hard copy caselaw, is called abstraction. During this phase of the analysis, the court is to

> dissect the allegedly copied program's structure and isolate each level of abstraction contained within it. This process begins with the code and ends with an articulation of the program's ultimate function. Along the way, it is necessary essentially to retrace and map each of the designer's steps—in the opposite order in which they were taken during the program's creation.[140]

The next step is called filtration.

> This process entails examining the structural components at each level of abstraction to determine whether their particular inclusion at that level was "idea" or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is nonprotectable expression.[141]

The filtration process defines the scope of the plaintiff's copyright.[142] The final part of the test involves an assessment of whether the defendant copied any of the "core of protectable expression" remaining after the prior two steps and "an assessment of the copied portion's relative importance with respect to the plaintiff's overall program."[143]

The *Altai* court emphasized that many ideas underlie a computer program: Programs are comprised of subprograms, sub-subprograms, and so on, each of which may have a separable, unprotected idea.[144] This dissection of the program into its modular components, coupled with the filtering process, shrinks the scope of the plaintiff's copyright to a size smaller than that afforded by any prior court.[145] Under the test, almost any decompiler could

[140] *Altai*, 982 F.2d at 707.

[141] *Id.*

[142] *Id.*

[143] *Id.* at 710.

[144] *Id.* at 705–11 ("[A] computer program's ultimate function or purpose is the composite result of interacting subroutines. Since each subroutine is itself a program, and thus, may be said to have its owu 'idea,' *Whelan*'s general formulation that a program's overall purpose equates with the program's idea is descriptively inadequate." The court then noted that its three-part test was designed to recognize this reality by breaking a program into its parts and filtering out unprotectable elements from each part.).

[145] The court admitted that it was narrowing copyright protection for computer pro-

justify wholesale decompilation as necessary to get to those low-level sub-subprogram ideas that are not documented by the provider in publications accompanying the software or other publicly available documentation.

Arguably, this was the precise situation that the *Atari* and *Sega* cases hoped to settle. Their underlying rationale appears to be that if software providers are forced to allow decompilation as a way of allowing others access to embedded ideas, they might choose instead to disclose those ideas by, for example, publishing their interface specifications.[146] But because the line between idea and expression is unclear, particularly in the software context,[147] a provider can never be sure that it has satisfied the *Atari* and *Sega* disclosure standard and thus has protected its program against decompilation. Moreover, while it is an established industry practice to publish interface specifications, those specifications are generally incomplete for a number of reasons.[148] Application programs usually use some operating system functions and must be written to fit—or interface—with the particular operating system.[149] However, not all parts of the operating system are tested or supported. Thus, published operating system interface specifications are incomplete because they do not include those unsupported or untested functions—for good reason.[150] If an ap-

grams: "If the test we have outlined results in narrowing the scope of [copyright] protection, as we expect it will, that result flows from applying, in accordance with Congressional intent, long-standing principles of copyright law to computer programs." *Id.* at 712.

[146] *Cf.* Samuelson et al., *supra* note 15, at 2378 ("The desire for interoperability means that there are natural incentives for many companies to share information (and for some, to monopolize such information) and that consumers have an interest in seeing that information shared.").

[147] *See supra* notes 52 and 138–45 and accompanying text.

[148] *See generally* Johnson-Laird, *supra* note 43, at 860 (finding documentation inaccurate, incomplete, and out of date).

[149] Operating system vendors usually define an application programming interface (API). An API may be defined as "[a] functional interface supplied by the operating system . . . that allows an application program written in a high-level language to use specific data or functions of the operating system . . . ." IBM DICTIONARY, *supra* note 122, at 24. The API provides application programmers with information to allow them to write applications running on the particular operating system. Note that an application written for one operating system, such as AIX, usually will not run on another, such as HP-UX (Hewlett Packard's version of UNIX), without change because the AIX and HP-UX APIs are different. The process of making modifications to an application to allow it to run on another operating system is called "porting."

[150] Admittedly, one reason why providers fail to publish complete API specifications is that a complete specification can form the basis for another party's cloning of the orig-

plication developer were to utilize such operating system functions, the application's performance would be unreliable. End users would likely have difficulty identifying whether the operating system or application provider were responsible for any bugs, and both software providers would likely lose some goodwill.

Nonetheless, it is possible to offer an additional justification for a fair use approach: It may encourage creative expression that would otherwise not have occurred. As noted previously, the statutory grant of copyright provides a limited monopoly that may introduce market imperfections.[151] The fair use provision of the Act helps to counteract those imperfections by sheltering copying and encouraging newcomers to build freely on earlier works when (1) transaction costs are such that the copyright owner and copier are unlikely to reach a licensing agreement; and (2) the particular activity in which the copier engages is likely to redound to the public benefit without unduly impacting the incentive of authors to create.[152] Thus, it may be appropriate to shelter decompilation under the fair use doctrine if situations exist in which (1) a license is unlikely to be concluded either because transaction costs are too high or the particular parties are unwilling to deal with each other;[153] and (2) allowing decompilation would increase creative ex-

inal system to compete directly with it. Thus, providers generally try to provide enough high level information to satisfy the requirements of application writers without disclosing low-level kernel detail that discloses much about the implementation of the operating system but is of little use to the application provider. *See also infra* note 158 (noting allegations that Microsoft intentionally hides supported, fully functional parts of its Windows interface to give its own applications a competitive advantage).

[151] *See supra* note 21.

[152] *See* Gordon, *supra* note 16, at 1601. *See generally,* Landes & Posner, *supra* note 16, at 357–61 (arguing for narrow construction of fair use doctrine).

[153] A good illustration of parties who may not deal with each other, even though it would be economically advantageous to both and socially beneficial to the public is provided by Microsoft and IBM, one-time collaborators who are now arch-enemies. Currently, IBM and Microsoft are bitter competitors in the market for PC operating systems. Microsoft has the largest market share with its Windows product, while IBM has been playing catch-up with OS/2. *See infra* note 154. It might be socially beneficial if Microsoft were to license its Windows interface specifications to IBM on a contiuuing basis so that IBM could adapt its OS/2 product to run Microsoft applications on an ongoing basis. This would offer customers more choice in operating systems. In fact, however, the two companies parted ways and announced that the exchange of interface specifications would cease in September 1993. *See, e.g.,* Stuart J. Johnston & Doug Barney, *IBM Fights for Visitation Rights to Windows API,* INFOWORLD, Sept. 6, 1993, at 1, 22 (discussing how source code exchange agreement involving IBM's and Microsoft's respective systems software ends September 17, 1993).

pression on the part of the decompiler without unduly impacting the incentives of the original developer.

The transaction cost argument is generally not borne out in industry practice and the impact of the fair use defense on creative incentives is unclear. The value and the market success of an operating system lie in the number and quality of applications running on it rather than just on the technical proficiency of the system.[154] Thus, in most cases, operating system providers have an incentive to overcome the transaction cost problem by helping application providers adapt their applications—a process called "porting"—to run on the particular operating system. For example, IBM has regional porting centers and 800 numbers set up for just such a purpose. Microsoft, the dominant operating system provider in the personal computer market,[155] has a developer's program to help others develop applications running on Microsoft systems.[156] Companies have even paid application developers to write on their operating systems.[157] Thus, in practice it seems rare that an ap-

[154] Compare IBM's OS/2 and Microsoft's Windows. OS/2 is a 32-bit multi-tasking operating system, while for a long time Windows was a 16-bit system with no multi-tasking (Windows95, the Windows follow-on product that became generally available in August 1995, has limited multi-tasking and is a 32-bit platform). *See* THE MICROSOFT SYSTEMS STRATEGY: WINDOWS (1994) (on file with author). However, despite the fact that Windows may not have been technically superior to OS/2, it consistently had available significantly more applications that OS/2, *see* Michael J. Miller, *Who Shot OS/2?*, PC MAGAZINE, Apr. 11, 1995, at 75, helping it to gain an approximate 80% market share, while OS/2 has approximately a 10% share. Alan Goldstein, *IBM Launches Hostile Bid for Software Maker Lotus*, DALLAS MORNING NEWS, June 6, 1995, at 1A; *see also* Samuelson et al., *supra* note 15, at 2375 (noting "there is considerable value in having users familiar with one's software"); *infra* text accompanying notes 276–77.

[155] *See* United States v. Microsoft, 159 F.R.D. 318, 322 (D.D.C.), *rev'd*, 56 F.3d 1448 (D.C. Cir. 1995) (stating that Microsoft's share of market for PC operating systems is consistently well above 70%); *supra* note 154.

[156] Telephone Interview with John Briggs, Senior Product Manager, SmartSuite (July 17, 1995); *see also* MICROSOFT DEVELOPER NETWORK (on file with author) (The *Microsoft Developer Network* describes various levels of development program membership, each with associated prices and information; for example, a Level 2 annual membership is $495 with members receiving updates to the Development Library, issues of the Developer Network News, a 20% discount on Microsoft Press books, a $20 credit on CompuServe connect charges, invitations to special events at trade shows, a membership kit, and "all the operating systems and Windows-related SDKs and DDKs on the Development Platform" with quarterly npdates thereto.). *But see infra* note 158 (discussing allegations that Microsoft favors its own developers).

[157] *See* Beth Freedman, *IBM Offers Cold Cash to OS/2 Developers: Borland, Lotus Among Firms Wooed*, PC WK., Mar. 23, 1992, at 129 (noting that IBM is coaxing software developers to write applications for OS/2 2.0 by offering a "smorgasbord of incentives").

plication developer has difficulty obtaining the required information.[158] Moreover, competitive pressures will likely force deals even between parties who would generally rather not reach agreement.[159] Even if that were not the case and management were to subordinate shareholder interests to grievances against other computer companies, refusing to deal with such companies when dealing would increase the value of the firm, the corporate law of fiduciary duty may intervene to give the shareholders a remedy and may, in fact, act as a check on this type of behavior. One then must consider whether the fair use doctrine should be used as another mechanism to force parties to deal. Finally, as noted, in situations in which large companies are bargaining, transaction

[158] *But see Microsoft*, 159 F.R.D. at 334 (stating that allegations against Microsoft include charges that Microsoft "manipulates its operating systems so competitors' applications software are inoperable or more difficult for the consumers to utilize effectively"); Response of the United States to Public Comment Concerning the Proposed Final Judgment and Notice of Hearing, 59 Fed. Reg. 59,426, at 59,427 (1995) (stating that Federal Trade Commission "considered whether Microsoft had deliberately created incompatibilities between its Windows operating system software product and its competitors' disk operating system products in order to disadvantage those competing products"); Rice, *supra* note 22, at 1152 (noting that operating system and application developers often provide information or access thereto to selected third parties but arguing that the privilege to gain access to a program without the agreement of the copyright owner is necessary as such a "privilege corrects for market failures or externalities that result in information access not being provided"); Amy C. Page, Note, *Microsoft: A Case Study in International Competitiveness, High Technology, and the Future of Antitrust Law*, 47 FED. COMM. L.J. 99, 104 (1994) (describing allegations that Microsoft gives product information to other applications' developers but gives its own developers a more complete version of such information first); Jared Sandberg, *Inftghting Unravels Alliance Seeking Standard to Protect Internet Purchases*, WALL ST. J., Sept. 28, 1995, at B10 (reporting that the chairman of Netscape, a small software firm, alleged that Microsoft "had demanded a 20% stake [in Netscape] and a seat on [Netscape's] board . . . in return for giving Netscape important technical data on Microsoft's new operating system"). See *infra* Section IV(A) for a proposal on how the law should respond in such situations.

[159] Despite the competitive relationship between IBM and Microsoft, IBM announced that it would license and support Windows NT. *See* Don Clark, *IBM to Promote Use of Microsoft System on PowerPC Chip in Challenge to Intel*, WALL ST. J., Nov. 11, 1993, at B6. Windows NT is Microsoft's operating system for distributed computing across many different types of hardware. Windows NT runs on a wide range of hardware architectures,

> deliver[ing] workstation power through advanced operating system features . . . . With Windows NT, business-critical applications that used to be available only on expensive workstations and mini-computers are now available for the PC platform. . . . Windows NT also runs existing applications for Windows and MS-DOS, so that information can be integrated with these new 32-bit applications for Windows NT.

THE MICROSOFT SYSTEMS STRATEGY: WINDOWS, *supra* note 154.

costs are low enough for the parties to strike a mutually agreeable deal.[160]

The transaction cost argument has more force in the end-user context. The average end user is unlikely to contact the operating system provider to ask for information on how the program works. Even if she were to do so, it is unlikely that she would obtain any meaningful information.[161] However, it is questionable whether the end user wishes to purchase anything more than the functionality that is obtained by running the object code.[162] A program that includes the *Atari/Sega* right to decompile would probably cost the end user more than one that did not.[163]

In sum, the *Atari* and *Sega* standard, while narrowly drawn in theory, may in practice constitute a broad invitation to decompile. *Altai* expanded the universe of ideas to be found within a computer program, and it is doubtful, given current industry practice, that providers will publish specifications replete with all of those ideas. *Atari* and *Sega* support a fair use defense for decompilation to obtain those unpublished ideas, albeit only for a "legitimate reason," with the "legitimacy" of the reason ostensibly determined *ex post* by the court.

In the decompilation process, the programmer is likely to obtain a significant amount of protected expression, particularly because the "hidden" ideas are most likely to be operating system kernel functions. While *Atari* and *Sega* do not permit the decompiler to copy that protected expression, in practice it is very difficult for the copyright holder to discover and police such copying. According to this analysis, the impact of the *Sega/Atari* rule on the

[160] *See supra* notes 54–60 and accompanying text.

[161] For example, when asked about Windows interface specifications, Microsoft referred the casual, nonprogrammer caller to bookstores where one may purchase publicly available documentation. When asked by the same type of caller about the possibility of obtaining a source code license, Microsoft indicated that it does not license its source code as that is what "makes Microsoft competitive." Telephone Interview with Michael Heald, Microsoft Corp. employee (June 29, 1995). Note, however, that at least one commentator contends that publicly available information is, in fact, obtained through reverse engineering. Johnson-Laird argues that software vendors provide insufficient documentation and states that "[a] quick inspection of any worthwhile technical bookstore will reveal numerous books that augment well-respected software products whose documentation is . . . wretched. All such books, to varying degrees, are borne of reverse engineering." Johnson-Laird, *supra* note 43, at 847.

[162] *See* Samuelson et al., *supra* note 15, at 2318–19 (noting that consumers value behavior rather than program text).

[163] *See infra* text accompanying notes 224–30.

incentives of the copyright system is far from clear, casting a good deal of doubt on *Sega's* benign view that its rule would encourage a growth in creative expression.

After *Atari* aud *Sega*, providers who are concerned about abuse of the limited decompilation permission that those cases grant are left with the alternatives of (1) publishing substantially more information about their systems than they ordinarily would in order to satisfy the *Atari/Sega* disclosure standard; (2) channeling resources into developing methods to make decompilation difficult;[164] or (3) contracting around the *Atari/Sega* disclosure standard through the use of a decompilation prohibition in their license agreements.

The evidence indicates that most providers continue to include provisions against decompilation in license agreements, thus contracting around the fair use exception set forth in *Atari* and *Sega*.[165] Assuming, however, that one accepts the essential premise of those cases—that there are ideas embodied in object code that are not protected by copyright and that should be accessible for the purpose of furthering creative expression—the question then becomes whether these private contracts that nullify the judicial interpretation of the Act should be enforced or preempted by the Act.

## III. THE PREEMPTION ANALYSIS

### A. *Statutory Preemption Under the Act*

A logical starting point for a copyright preemption discussion is the statutory preemption provision contained in § 301 of the Act. As originally drafted, § 301(b)(3) would have included exam-

[164] *See* Leaffer, *supra* note 56, at 1096 ("In response to a threat of reverse engineering, software owners may increase their efforts by placing both legal and technological hurdles in the path of those trying to understand the function of their program. Companies will engage in technological strategies to render the task of reverse eugineering more costly.").

[165] *Id.* at 1097 (suggesting that reverse engineering prohibitions in license agreements, "already common in the industry, will become ever more popular after *Sega*"). Both Microsoft and IBM continue to include provisions against decompilatiou in their license agreements. However, IBM does not prohibit reverse engineering. *See supra* note 119. *But see* PROGRAM LICENSE AGREEMENT FOR U.S.A. (on file with author) (The license accompanying Compaq's Contura 400C laptop provides: "Pursuant to this Agreement, you may . . . (b) copy the programs into any computer in readable or printed form for backup or modification purposes in support of your use of the programs.").

ples of causes of action that were not preempted. These examples were "intended to illustrate rights and remedies that are different in nature from the rights comprised in a copyright and that may continue to be protected under State common law or statute."[166] The bill included breach of contract as one of these examples and the legislative history noted that "[n]othing in the bill derogates from the rights of parties to contract with each other and to sue for breaches of contract."[167] However, the list of examples was deleted from the final, enacted version of § 301 because of the Justice Department's concern that state law misappropriation actions should not be included.[168] Although it is always speculative to infer congressional intent, both the original inclusion of breach of contract as an example of a cause of action that would survive preemption and the reason for deletion of the list of examples suggest that Congress did not intend, as a general rule, to preempt breach of contract actions under the Act.

Under the enacted version of § 301, a cause of action will be preempted when (1) the subject matter at issue is within the subject matter of copyright; and (2) the action involves a legal or equitable right equivalent to any of the § 106 exclusive rights given to a copyright holder.[169] Because it is now well settled that computer programs are within the subject matter of copyright,[170] the preemption inquiry should focus on whether or not a breach

[166] H.R. REP. NO. 1476, 94th Cong., 2d Sess. 132 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659, 5748.

[167] *Id.* Other examples of causes of action that were not to be preempted by the Act included "breaches of trust, invasion of privacy, defamation and deceptive trade practices such as passing off and false representation." NIMMER & NIMMER, *supra* note 63, § 1.01[B], at 1-17 (citing § 301(b)(3), H.R. 4347, 89th Cong., 2d Sess (1966)).

[168] McManis, *supra* note 23, at 88 & n.307; *see also* NIMMER & NIMMER, *supra* note 63, § 1.01[B], at 1-26 to -30 (documenting Justice Department objections and detailing legislative debate as well as congressional confusion as to correct interpretation to place on deletion of § 301(b)(3)).

[169] 17 U.S.C. § 301(a) (1994) provides:

> On and after January 1, 1978, all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106 in works of authorship that are fixed in a tangible medium of expression and come within the subject matter of copyright as specified by sections 102 and 103, whether created before or after that date and whether published or unpublished, are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State.

[170] *See, e.g.*, Apple Computer Inc. v. Formula Int'l Inc., 725 F.2d 521, 524 (9th Cir. 1984); Apple Computer Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

of contract action based on breach of a contractual provision against decompilation "involves a legal or equitable right equivalent to any of the exclusive rights given to a copyright holder."

The courts have had some difficulty in interpreting this second preemption requirement. In general, they assess whether the non-copyright cause of action contains an element that makes it qualitatively different from a claim of copyright infringement; if so, the action is not preempted.[171] However, they have not held uniformly that a breach of contract action is always sufficiently qualitatively different from one in copyright infringement such that it survives preemption.

For example, in *Wolff v. Institute of Electrical and Electronics Engineers, Inc.*, the court held that when the act breaching the contract was the same as the act constituting copyright infringement, the breach of contract action was preempted.[172] The court rejected the reasoning set forth above regarding the congressional intent to be inferred from the deletion of § 301(b)(3), stating that "[w]here Congress includes limiting language in an earlier version of a bill, but deletes it prior to enactment, it may be presumed that the limitation was not intended."[173]

However, *Wolff* has not been well received by either the commentators[174] or other courts. In *National Car Rental System, Inc. v. Computer Associates International, Inc.*, Computer Associates (CA) had licensed software to National Car Rental System, Inc. (National).[175] The license agreement contained language limiting

[171] *See, e.g.,* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 719 (2d Cir. 1992) (Trade secret action was not preempted by the Act because "where the use of copyrighted expression is simultaneously the violation of a duty of confidentiality established by state law, that extra element renders the state right qualitatively distinct from the federal right, thereby foreclosing preemption under section 301.").

[172] 768 F. Supp. 66, 69 (S.D.N.Y. 1991) (holding that defendant's use of copyrighted picture violated use restriction in license agreement).

[173] *Id.* (citing Russello v. United States, 464 U.S. 16, 23-24 (1983)).

[174] See, e.g., 1 NIMMER & NIMMER, *supra* note 63, §1.01[B] at 1-19 to -20 (footnotes omitted), which argues that the court's logic in relying on *Russello* is faulty:

> [I]t could be argued that under the bill as originally drafted, enumerated activities such as breach of contract would be deemed nonpreempted, and nonenumerated causes of action could be held preempted, whereas following the amendment removing the limiting language, even nonenumerated causes of action could be held preempted. The conclusion does not follow, by contrast, that breach of contract . . . and all of the other originally enumerated causes of action are all to be deemed subject to blanket preemption under the law as enacted. The conclusion of [*Wolff*] must therefore be rejected.

[175] 991 F.2d 426 (8th Cir.), *cert. denied*, 114 S. Ct. 176 (1993).

National to using the programs "only for the internal operations of [National] and for the processing of its own data."[176] Later, the parties agreed that National could permit Electronic Data Systems Corporation (EDS) to use the software solely "to process data of [National] and in no event for the processing of data . . . of any third party other than [National]."[177] National breached the licensing agreement by using the programs to process third-party data and authorizing EDS to do the same.[178]

National sought a declaratory judgment that its use of the programs did not constitute breach of contract or copyright infringement.[179] CA, in turn, counterclaimed for both breach of contract and copyright infringement based on National's making an unauthorized copy of the program.[180] The district court held that the breach of contract action, as pled, was " 'equivalent' to the exclusive copyright right of distribution of copies of the work," and was therefore preempted.[181] The district court further noted that the mere "presence of a contract promise did not create a right qualitatively different from copyright, and stated: '[I]n essence, CA alleges National breached the license agreement by infringing CA's copyrights in the licensed software.' "[182]

The Eighth Circuit reversed the district court's finding of preemption. The court construed CA's pleadings as alleging that "National breached their contract by using the program itself, or through EDS, to process data for [third parties.]"[183] The court then framed the issue as whether a "limitation on the uses to which a licensee may put a licensed work are preempted even though those uses do not involve the exclusive copyright rights."[184] It held that in such a case, the breach of contract action is not preempted because "the alleged contractual restriction on . . . use of the licensed programs constitutes an extra element in addition to the copyright rights making this cause of action

[176] *Id.* at 427.
[177] *Id.* at 428.
[178] *Id.*
[179] *Id.*
[180] *Id.*
[181] *Id.*
[182] *Id.* at 430 (quoting mem. op. at 10).
[183] *Id.* at 429 (rejecting possible district court view that pleadings alleged that Natioual actually distributed copy of program).
[184] *Id.* at 431.

qualitatively different from an action for copyright."[185] The court stated:

> CA does not claim that National is doing something that the copyright laws reserve exclusively to the copyright holder, or that the use restriction is breached "by the mere act of reproduction, performance, distribution or display." Instead, on this posture, CA must be read to claim that National's or EDS's processing of data for third parties is the prohibited act. None of the exclusive copyright rights grant CA that right of their own force. *Absent the parties' agreement, this restriction would not exist.* Thus, CA is alleging that the contract creates a right not existing under the copyright law, a right based upon National's promise, and that it is suing to protect that contractual right.[186]

Finally, the *National* court both limited *Wolff's* holding[187] and disagreed with its interpretation of congressional intent.[188]

*National* stands for the proposition that a breach of contract action is qualitatively different from a claim in copyright infringement when the right or obligation being enforced would not exist but for the parties' agreement. There is some disagreement as to whether a breach of contract action in which the breaching act is the same as one that would infringe a copyright is preempted under the Act. *National* did not reach this issue, but cited with approval district court cases that have held such types of actions preempted.[189] However, in their seminal treatise on copyright law, Nimmer & Nimmer note that since a contract is predicated on a promise and copyright infringement is not, breach of contract actions are, by their nature, qualitatively different from copyright because they necessarily allege a breach of promise.[190] Nimmer therefore seems to imply that copyright preemption is inappropriate as a general rule in breach of contract cases.[191]

At first glance, it would seem that prohibitions against decompilation contained in license agreements would be preempt-

[185] *Id.*

[186] *Id.* at 433 (emphasis added).

[187] "[T]he *Wolff* case stands at most for the proposition that a breach of contract claim alleging nothing more than an act of infringement is preempted." *Id.* at 432.

[188] *Id.* at 433–34.

[189] *Id.* at 434 n.6.

[190] 1 NIMMER & NIMMER, *supra* note 63, § 1.01[B], at 1–18.

[191] *See National*, 991 F.2d at 434 n.6 (citing earlier edition of *Nimmer*). It is unclear from the latest edition of the treatise, *supra* note 63, whether *Nimmer* still has this view.

ed under § 301 and the *National* reasoning. The same act that would breach the contract—decompiling—would infringe the copyright unless excused by fair use.[192] Thus, unless the courts were to adopt the approach attributed to Nimmer, such a breach of contract action would not be "qualitatively" different from a copyright infringement one. Under Nimmer's approach, on the other hand, a court might conclude that the breach of contract 'action would not be preempted. This Article argues, however, that posing the problem in this manner—as an all or nothing preemption or nonpreemption of decompilation provisions—is too simplistic. Rather, when one assesses separately the validity of such provisions in the context of negotiated licenses, there is a good argument that breach of contract actions are neither statutorily nor constitutionally preempted. Moreover, there are compelling policy reasons that support this result. In contrast, when such provisions are contained in nonnegotiated, standard form licenses, the argument for nonpreemption of breach of contract actions is more tenuous.[193] However, in both contexts, a court should make a deep-

[192] *See* Rice, *supra* note 17, at 608–09 (stating that while at first glance, differing elements proving breach and infringement suggest nonequivaleuce, "[r]everse engineering prohibitions functionally exclude copying or preparation of a derivative work by exacting an agreement to uot utilize the sole independent means of access to source code"). Since decompiling involves the making of a copy, the right to reproduce in § 106(1) and the right to prepare derivative works in § 106(2) are implicated.

[193] But see Rice, *supra* note 22, at 1201, which asserts that

> [i]t does not, and it should not make a difference whether differences in the nature of programs and their markets dictate or justify differing outcomes in preemption analysis. Patent law and policy do not distinguish between ideas and processes embodied in products or computer programs distributed in high volume in the mass market and those distributed in limited volume in special use or niche markets. Any difference, if one exists, is more fundamental to competition law, which independently would make competition-restricting use restrictions contained in mass market, standard form contracts vulnerable to invalidation as impermissible restraints of trade. One ought not, in short, simply assume that classic notions of freedom to contract assure the enforceability of restrictions contained in more fully negotiated commercial market contracts.

This Article takes issue with some of the foregoing statement. Specifically, it attempts to demonstrate that there is a qualitative difference between negotiated and nonnegotiated contracts that justifies a different preemption result. Moreover, it suggests that even in the nonnegotiated contract context, nonpreemption may be the correct result. It is true that neither patent nor copyright law makes substantive distinctions between negotiated and nonnegotiated contracts for purposes of preemption analysis. However, this distinction is helpful in understanding the nnderlying policy issues, particularly copyright law's "extra element" test. Additionally, if different results are, in fact, justified in either of the two contexts or under existing intellectual property law generally, society should consider adjusting that law rather than affording it presumptive legitimacy. Finally, this Article attempts to explain the rationale for certain provisions and the importance of freedom of er inquiry into market conditions before making the preemption decision.

1. *The Statutory Preemption Analysis Applied to Negotiated Contracts.* As noted, at one end of the licensing spectrum is the negotiated contract.[194] Such contracts generally are bargained for agreements in which both the licensor and licensee are knowledgeable parties. They understand the nature of the rights they are granting and obtaining, respectively, including the rights that the licensee has agreed to forego. Neither party would enter the agreement if it did not think it were receiving something worthwhile in exchange. Unless some overriding policy justification may be asserted, there is no persuasive reason to preempt particular provisions of these deals.[195]

Under § 301, breach of contract actions in this context should survive preemption whether or not the claim alleges any act in addition to that of breaching the contract through the process of making copies in the decompilation process. Simply stated, there *is* an extra element in such breach of contract actions that renders them qualitatively different from an action in copyright infringement. That extra element is a breach of promise—which also might be labeled a breach of trust or confidential relationship—that would not exist but for the parties' agreement.[196]

Nonpreemption seems to be the right result as a policy matter as well. Given the level of sophistication of the parties involved, it is reasonable to conclude that the rights that the licensee receives

contract in maintaining the viability of the computer software industry.

[194] *See supra* text accompanying notes 54–57.

[195] *See infra* Part IV (suggesting that contractual provisions may be preempted when enforcement conflicts with antitrust laws); *cf.* Paetzold, *supra* note 75, at 831–32 (suggesting that agreements contracting around fair use should be evaluated under rule of reason standard).

[196] *See* Rice, *supra* note 22, at 1200–01 (citing premarketing distribution of beta test copies under contracts prohibiting reverse engineering as consistent with *Kewanee Oil v. Bicron* as true confidential relationship is involved). Rice contends, however, that *Bonito Boats*

> makes it clear that a state legislature, let alone a private party seeking to contractually "legislate" maximum protection against competition, may not secure "protection beyond that available under the law of unfair competition or trade secret, without any showing of consumer confusion, or breach of trust or secrecy." Merely declaring the existence of such an obligation of trust or secrecy ordinarily is not itself sufficient to establish a relationship characterized by such an obligation.

*Id.*

are closely connected to the price it pays. If the licensee were to bargain for a right to decompile, the price would likely increase. It makes sense to conclude that the knowledgeable licensee has decided that the right to decompile is not worth the corresponding price increase. Moreover, it is possible that the licensor would refuse—at any price—to license the program under terms including a right to decompile. In both cases, the transaction would not go forward in the absence of an enforceable contractual provision against decompilation. Thus, preempting these provisions would not necessarily strike the copyright balance correctly, and may, in fact, ultimately undermine the goals of copyright. If the transaction does not go forward, the licensee not only may lack the opportunity to learn the ideas inherent in the program through means other than decompilation, but also may be deprived of access to the software's functionality altogether.[197] Preemption, then, rather than increasing the flow of ideas and encouraging creative activity may, in fact, dampen both that flow and activity.

The argument against preemption is also addressed, albeit implicitly, by others grappling with the problems created by the intersection of copyright and contract. The National Conference of Commissioners on Uniform State Laws (NCCUSL) has approached the problem in its proposed revision of Article 2 ("Sales") of the Uniform Commercial Code (UCC). The NCCUSL has issued a discussion draft of revisions that would bring licensing transactions within Article 2's scope.[198] The NCCUSL starts its commentary

[197] This result suggests that the law should recognize the distinction between prohibitions against reverse engineering and those against decompilation. If a licensee agrees to a prohibition against decompilation, it still may engage in other forms of reverse engineering; if it agrees to a prohibition against reverse engineering, it has no opportunity to access the ideas in the software. *See generally supra* notes 118–21 and *infra* Part IV.

[198] U.C.C. §§ 2-2101 to -2521 (Proposed Official Draft 1994) [hereinafter REVISED UCC]. At present, the *Revised UCC* lacks any substantive force and has "not been passed upon by the National Conference of Commissioners on Uniform State Laws . . . . [It does] not necessarily reflect the views of the Committee, Reporters or Commissioners. Proposed statutory language . . . may not be used to ascertain legislative meaning of any promulgated final law." *Id.* at title page. This draft recently has been abandoned as a result of the Committee's decision to add a new article to the UCC to deal with licensing, rather than to use the "hub and spoke" drafting approach embodied in the *Revised UCC.* For a description of the "hub" and "spoke" approach, see Raymond T. Nimmer, *Intangibles Contracts: Thoughts of Hubs, Spokes, and Reinvigorating Article 2*, 35 WM. & MARY L. REV. 1337 (1994); Raymond T. Nimmer et al., *License Contracts Under Article 2 of the Uniform Commercial Code: A Proposal*, 19 RUTGERS COMPUTER & TECH. L.J. 281 (1993). However, the new drafting committee is likely to use the *Revised UCC* as a starting point for the new article. Telephone Interview with John McCabe, Legal Counsel

by reiterating the broad principle of freedom of contract which has historically been a premise of Article 2.[199] This philosophy is reflected in proposed § 2–2107, which states: "All of the rules expressed in this chapter can be varied by, and are subject to, contrary agreement of the parties . . . ." This broad freedom of contract principle is limited by proposed § 2–2103, which notes that license agreements may be subject to federal law. The comments to that section, however, note that as a general rule, federal copyright laws contain property rules affecting certain aspects of license agreements but very few contract rules:

> These [contract rules] include provisions regarding registration and writing requirements for effective creation of exclusive licenses and assignments, provisions that may limit rules on the termination at will of copyright licenses, and provisions about the rights received by an owner of a copy of software with reference to making back up and adaptive copies for its own use. . . . If these rules apply, federal law preempts state rules and, under the terms of this exclusion, covers that aspect of the transaction. . . . This is a limited preemption both in terms of what contract rules are covered by federal law and what areas of intangibles contracting are included. The case law consistently holds that contract law issues are state law matters regarding intellectual property unless there is a specific preemptive provision with which the state law conflicts. This chapter applies to the extent that the federal statutes do not, but obviously must defer to federal law when it is applicable. . . . Federal law does not govern most aspects of the contract law for intangibles, even with respect to intellectual property.[200]

Further, proposed § 2–2301(c)(1) states that "a licensee who becomes an owner of [a] copy has all of the rights of an owner under applicable federal law." The comments state:

> Current case law does not clarify whether a contract supersedes or can exclude the rights that a first sale[201] would create for

& Legislative Director, NCCUSL, Sept. 7, 1995. This Article is unaffected by the Committee's decision because it is the substance of the provisions, rather than the drafting approach, that is presented here to provide another perspective on the problem. This perspective remains helpful for comparative purposes, whether or not the revisions eventually are adopted in the current substantive form or not at all.

[199] REVISED UCC, *supra* note 198, at 1.

[200] *Id.* § 2–2103 cmts. 4–6 (citations and emphasis omitted).

[201] The first sale doctrine is codified in § 109(a) of the Act, which states that

> the owner of a copy. In part this is because the agreement of the parties in itself not only establishes license terms, but also determine [sic] whether or not a sale occurs. The better view is that a valid contract can condition property rights. Subsection (c)(1) states as a default rule the premise that the copy owner's rights are those granted in the agreement and any rights created under the applicable federal law unless otherwise agreed thus allowing for the possibility that contrary agreement may convey away those federal property rights, but not mandating that this be the case.[202]

Other sections of the proposed revision also seem to support the notion that decompilation terms should withstand a federal preemption analysis. For example, in § 2-2204(b), which deals with license interpretation, the drafters have proposed the following language: "Contract terms dealing with the scope of the grant and the subject matter are construed under ordinary principles of contract interpretation and the commercial context if possible." This seems analogous to the approach in existing Article 2, in which, unless the parties otherwise agree, the commercial context is always ad-

> [n]otwithstanding the provisions of section 106(3) [the copyright owner's exclusive right to distribute copies to the public by sale, rental, lease or lending, and to authorize others to do so], the owner of a particular copy or phonorecord lawfully made under this title, or any person authorized by such owner, is entitled, without the authority of the copyright owner, to sell or otherwise dispose of the possession of that copy or phonorecord.

17 U.S.C. § 109(a) (1994). Essentially, the first sale doctrine limits the copyright holder's monopoly in the sense that the copyright owner may only "control" the first sale. The first purchaser is free to do with the copyrighted work as he wants, subject to the copyright owner's § 106 rights. That freedom includes the right to resell or lease the copyrighted work and receive compensation for that rental or lease. Commentators have cited the first sale doctrine as one of the driving forces behind shrink wrap licensing. *See, e.g.*, Mario L. Baeza, *Acquisition and Exploitation of Mass Market Software*, *in* COMPUTER SOFTWARE AND CHIPS 1986: PROTECTION AND MARKETING 515, 529 (Practicing Law Institute, 1986); Gary W. Hamilton & Jeffrey C. Hood, *The Shrink Wrap License—Is It Really Necessary*, COMPUTER LAW., Aug. 1993, at 16. Standard shrink wrap terms attempt to avoid the first sale doctrine by denominating the transaction as a license rather than a purchase and by purporting to bind not only the direct purchaser/licensee to its terms, but also any subsequent transferee. Software providers feared that original purchasers/licensees would lend the program to others who would make copies and return the original to the first purchaser. If this practice were widespread, it would adversely impact the software provider's revenue. This fear led to the Computer Software Rental Amendments Act of 1990, which amended § 109 to prohibit "any person in possession of a particular copy of a computer program . . . [from,] for the purposes of direct or indirect commercial advantage, dispos[ing] of, or authoriz[ing] the disposal of, the possession of that . . . computer program . . . by rental, lease, or lending." 17 U.S.C. § 109(b) (1994).

[202] REVISED UCC, *supra* note 198, § 2-2301, cmt. 7 (emphasis omitted).

missible in the form of usage of trade, course of dealing, and/or course of performance to supplement even a fully integrated contract.[203] Because decompilation provisions clearly are a part of the usage of trade, § 2-2204 seems to argue that they should be permitted.[204] Further buttressing this contention is comment 6 to proposed § 2-2204, which states:

> Other than for exclusivity issues, this section rejects reserving in the licensor any rights not expressly granted to the licensee as a basic interpretation approach. If a federal policy to the contrary exists relating to copyrights or patents, that policy preempts ordinary commercial standards in appropriate cases. In most cases, however, ordinary interpretation rules do not violate federal policy and the better reading of modern case law involving licenses applies ordinary rules in most cases. Contract law should enforce the choices of the parties.[205]

Yet the revision intentionally displays some ambivalence.[206] For example, in § 2-2206, the proposed revision states that "[a]n intangibles contract conveys no right to the licensee to receive underlying code," but indicates that the parties are free to contract around this provision. At first glance, this seems consistent with the contention that the parties are free to bargain to a decompi-

[203] U.C.C. § 2-202 (1995). The drafters of the original UCC proceeded under the assumption that the terms to which commercial parties normally agree are likely to be efficient and are found in usage of trade, course of dealing, and course of performance. Thus, the drafters favored broad admissibility of this evidence as a contractual cost-saving device. That is, if parties may contract against a commercial backdrop of shared assumptions that the UCC automatically adds to their agreement, there is no need for the parties to negotiate standard terms, thus saving time and transaction costs. *See* ALAN SCHWARTZ & ROBERT E. SCOTT, COMMERCIAL TRANSACTIONS PRINCIPLES AND POLICIES 49-50, 60 n.1 (2d ed. 1991) (discussing importation of and intentional negation of contextual evidence). The drafters of the revision seem to be trying to achieve the same effect in proposed § 2-2204.

[204] *But see* U.C.C. § 1-205 cmt. 6 (1995):

> The very fact of commercial acceptance makes out a prima facie case that the usage is reasonable, and the burden is no longer on the usage to establish itself as being reasonable. But the anciently established policing of usage by the courts is continued to the extent necessary to cope with the situation arising if an imcouscionable or dishonest practice should become standard.

[205] REVISED UCC, *supra* note 198, § 2-2204 cmt. 6.

[206] In fact, the drafters of the revision have drafted it "both ways" because they have not yet decided the preemption issue. Because the drafters have abandoned the "hub and spoke" approach, *see supra* note 198, the drafting committee will analyze this issue and may or may not adopt the approach embodied in the *Revised UCC* cited herein. Telephone Interview with Deborah Perlman, NCCUSL Legislative Counsel, Sept. 7, 1995.

lation prohibition. However, comment 6 to proposed § 2-2206 states: "Federal policy supports a transferee's right to reverse engineer and discover secrets in technology, but a transferee's right to reverse engineer depends on it obtaining the relevant information in a fair manner."[207] The relationship between the text of § 2-2206 and its comments is unclear.

Despite the ambivalent language noted, the UCC's historical underpinnings and much of the revised language are consistent with a freedom of contract approach that would militate against the preemption of decompilation restrictions in negotiated licenses and for a view of fair use as an alienable right. Negotiated license agreements generally are the result of arm's length bargaining between informed parties—the classical contract characterized by true consent. This consent, whether characterized as a promise, a trust, or a confidential relationship, should be enough to render an action for breach of a negotiated license agreement qualitatively different from a copyright infringement action. However, this analysis and the UCC's approach is more difficult to support in the context of nonnegotiated license agreements.

2. *The Statutory Preemption Analysis Applied to Standard Form Contracts.* The consensual bargaining model observed with respect to negotiated license agreements does not hold for standard form license agreements.[208] These agreements are presented to the licensee on a take-it-or-leave-it basis. The licensor is usually a software company—an informed and competent party—while the licensee may be neither informed about the content of the license, its meaning, or the applicable law, nor competent to understand the license terms or law even if informed. Moreover, the licensee may not even learn what terms accompany the software until after the product is purchased and the program is loaded and run for the first time.[209] The issue then becomes one of deciding whether the differences in the bargaining model between the negotiated context and the standard form context compel a different conclusion under § 301.

[207] REVISED UCC, *supra* note 198, § 2-2206 cmt. 6 (citing *Sega* and *Atari* with ostensible approval).

[208] This discussion includes shrink wrap agreements and signed contracts that are not negotiated.

[209] *See supra* note 58.

At first glance, it would seem that only expressly negotiated licenses between knowledgeable and informed parties would survive a § 301 analysis. It is the critical additional element of informed consent that qualitatively distinguishes the breach of contract and copyright infringement actions. While this argument may be comforting because of its simplicity, it ignores both legal and practical market realities that suggest that even in the case of standard form contracts in most cases, it would be best to allow a breach of contract action based on breach of a decompilation provision to proceed.

Under contract law, a contract is not necessarily unenforceable merely because it is standard in form.[210] Nonetheless, there is legitimate concern that when one party to a transaction is uninformed and/or incompetent, the transaction itself may be both inefficient[211] and unfair. Consequently, courts and legislatures have tried to formulate rules that correct these types of market imperfections by mandating disclosure of information and evaluating transactions for unconscionability.[212]

The enforceability of standard form contracts within the computer industry—let alone specific terms of those contracts—has been a matter of substantial debate.[213] In its proposed revision of

[210] Standard form contracts are not necessarily unenforceable contracts of adhesion. *See* 2 RONALD A. ANDERSON, ANDERSON ON THE UNIFORM COMMERCIAL CODE § 2-302:52 (1982). *Cf.* Rice, *supra* note 17, at 563 (Shrink wrap contracts are a "rational and expedient response to the evolution of market conditions and arrangements; neither contemporary theory nor actual practice would render either the contracts or their specific terms unenforceable solely because they are contracts of adhesion.").

[211] "Economic efficiency is generally defined as the allocation of scarce resources to their highest valued uses at minimum cost." SCHWARTZ & SCOTT, *supra* note 203, at 25.

[212] *See, e.g.*, U.C.C. § 2-316 (1995) (setting forth requirements for procedurally effective warranty disclaimers). These UCC requirements—conspicuousness, standardized language, etc.—generally go to trying to inform the party against whom a disclaimer is to operate. Some courts, however, have held that mere compliance with § 2-316 is insufficient to ensure the enforceability of a disclaimer, and that disclaimers must also be evaluated under § 2-302 for unconscionability. *See* U.C.C. § 2-302 (1995) (stating that contract provisions may be unenforceable if unconscionable at the time the contract was made). The unconscionability inquiry is concerned with ensurmg that the party against whom a disclaimer is to operate is competent or at least treated fairly. *See* Martin v. Joseph Harris Co., Inc., 767 F.2d 296, 299-302 (6th Cir. 1985) (holding that disclaimers, even those complying with the procedural requirements of § 2-316, may nevertheless be ineffective because they are unconscionable, particularly in a situation involving standard form contract, latent defect, and change from standard warranty practice).

[213] *See, e.g.*, Ronald L. Johnston & Allen R. Grogan, *Trade Secret Protection for Mass Distributed Software*, COMPUTER LAW., Nov. 1994, at 1, 5 (stating that "the law is far from settled with respect to the enforceability of shrinkwrap license agreements"). For

Article 2, the NCCUSL has generally advocated the enforceability of standard form license agreements[214] as long as

> (i) the licensee has an opportunity to review the terms of the agreement whether or not he actually does so;[215] and
> (ii) the licensee expressly consents or otherwise manifests his assent to the terms. A party expressly consents to a license term if the term is conspicuous[216] and the party engages in conduct

discussions regarding the enforceability of shrink wrap licenses, see generally Page M. Kaufman, *The Enforceability of State "Shrink-Wrap" License Statutes in Light of* Vault Corp. v. Quaid Software, Ltd., 74 CORNELL L. REV. 222 (1988). *See also* David A. Einhorn, *Box-Top Licenses and the Battle-of-the-Forms*, 5 SOFTWARE L.J., 401, 406-12 (1992) (discussing validity of shrink wrap contracts in light of UCC Article 2); Michael G. Ryan, *Offers Users Can't Refuse: Shrink-Wrap License Agreements as Enforceable Adhesion Contracts*, 10 CARDOZO L. REV. 2105, 2125-35 (1989) (endorsing shrink wrap licensing as a valid response and solution to mass-market software piracy); *supra* note 58. One of the most recent cases dealing with shrink wrap licenses is Arizona Retail Sys., Inc. v. Software Link, Inc., 831 F. Supp. 759 (D. Ariz. 1993). This case held that where a licensee engaged in a number of purchases from a licensor, the shrink wrap terms applied solely to the initial purchase, and the contract was formed only after the licensee opened the shrink wrap with notice that such action would result in an enforceable contract. *Id.* at 763.

[214] The theoretical underpinnings of the drafters' approach are set forth in comment 3 to § 2-2203.

> The basic issue in standard form contracts lies with what presumption one makes about a contract. One model holds that contract terms consist only of those terms that the parties actually agree to in fact. This would invalidate many provisions of standard contracts because in many transactions neither party reviews or bargains about every term of the agreement. This model hinges on *actual* agreement. Its chief flaw is that it anticipates or requires conduct that is not common in commercial or consumer practice. In fact, many commercial transactions are conducted with written agreements that are not fully negotiated or read. The alternative of enforceability is followed in most cases. It assumes that assent to the contract implies assent to all terms. In many court opinions, this is explained in terms of a duty to read and understand the contract or in terms of an objective assessment of contract terms. The contract contains the terms to which you objectively, not subjectively, assented. The *Restatement* uses this latter model with restrictions designed to avoid unfair surprise.

REVISED UCC, *supra* note 198, § 2-2203 cmt. 3.

[215] Section 2-2203(d) of the REVISED UCC provides:

> A party has an opportunity to review the terms of a license if: (1) the license is made available to the party (i) prior to its acquisition of a copy of the intangibles and in a manner designed to call the license terms to the attention of the party assenting to the form, or (ii) provided to the party in a manner so that the terms of the license will be conspicuous in the normal course of its initial use or preparation to use the intangibles; and (2) in a mass market license, the party assenting to the form is authorized to obtain a refund of all license fees paid by returning the copy of the intangibles or discontinuing use following its opportunity to review the terms of the license.

[216] Under REVISED UCC § 2-2101(4),

> "Conspicuous" with reference to a term or clause, means so written or displayed that a reasonable person against whom it is to operate ought to have

> signifying acceptance of the term, after having an opportunity not to engage in such conduct.[217] Similarly, a party manifests assent to a license if, after having the opportunity both to review the license and to decline to engage in conduct constituting acceptance under the license terms, it engages in such conduct.[218]

Thus, despite the fact that there is no true consent, trust, or confidential relationship in the standard form contract, the proposed revision would infer such a relationship under certain circumstances—circumstances that encompass most mass-market software contracting today.

The revision is less clear on whether it would sanction all of the terms of those standard form contracts. It does state that it is intended to "[validate] most terms of most shrink wrap licenses."[219] On the other hand, proposed § 2-2203(c) states that "a term becomes part of the contract if the term: (1) transfers rights or creates restrictions on the behavior of the licensee which give the licensee no fewer rights than would be given to a purchaser at a first sale under federal intellectual property law."[220] If the license were characterized as a sale, the purchaser would, by default, have the limited *Sega/Atari* right to decompile. Thus, the proposed revision seems to mandate preemption of any mass-market license term prohibiting decompilation. However, the comments to § 2-2203 indicate that, in the context of copyright law, § 2-2203(c) was intended only to ensure that licensees have rights to

> noticed it or, in the case of an electronic message intended to evoke a response without the need for review of the message by an individual, in a form that would enable the recipient or the recipient's computer to take it into account or reacte [sic] to it without review of the message by an individual. A term or clause is conspicuous if it is (i) a writing or display and is a printed heading in capitals; (ii) language in the body of a form or a display and is in larger or other contrasting type or color than other language; (iii) a term or clause referenced in the body of a form or a display by conspicuous language and the term or clause can be readily accessed.

*See also* U.C.C. § 1-201(10) (1995):

> A term or clause is conspicuous when it is so written that a reasonable person against whom it is to operate ought to have noticed it. A printed heading in capitals (as: NON-NEGOTIABLE BILL OF LADING) is conspicuous. Language in the body of a form is "conspicuous" if it is in larger or other contrasting type or color. But in a telegram any stated term is "conspicuous." Whether a term or clause is "conspicuous" or not is for decision by the court.

[217] REVISED UCC, *supra* note 198, § 2-2203(f).

[218] *Id.* § 2-2203(e).

[219] *Id.* § 2-2203 cmt. 4.

[220] *Id.* § 2-2203(c); see also *supra* note 201 for a description of "first sale" rights.

make backup and archival copies under § 117 of the Act.[221] The text and comments are largely silent on the topic of decompilation.[222]

While the draft is indeterminate on the issue of the enforceability of decompilation provisions, it offers some arguments that might help such a provision survive § 301 preemption in the shrink wrap context. For example, its general approach of inferring consent from the conduct of the licensee suggests that even in the standard form context, there may be an extra element that would make a breach of contract action qualitatively different from a copyright infringement one. Of course, decompilation provisions still must be evaluated for unconscionability.[223] If one accepts the implied consent theory, there is no procedural unconscionability in the manner of contract formation. Thus, the only remaining relevant consideration is whether the term itself is substantively unconscionable.

The question of substantive unconscionability is best answered by reference to policy considerations. Among these considerations is what the parties' reasonable expectations were regarding the terms of the bargain, given the price attached to the particular "bundle of rights" granted. As noted, mass-market licensing sprang up because transaction costs made negotiating with every licensee impractical.[224] Yet, the same decompilation provisions that appear in these mass-market standard form licenses often appear in negotiated agreements.[225] These facts lend support to the contention that if, in the mass-market context, transaction costs were such that the parties would bargain, they might agree to a decompilation prohibition.

[221] REVISED UCC, *supra* note 198, § 2-2203 cmt. 11.

[222] *See supra* notes 206–07 (noting that draft refers to *Sega* and *Atari* only once and that drafters have not yet decided preemptive effect of federal law on provisions against reverse engineering).

[223] The *Revised UCC* does not contain an unconscionability provision. Under the "hub and spoke" drafting approach, the licensing provisions discussed herein constituted a "spoke," while unconscionability provisions and other universal provisions applicable to both sales and licensing transactions would be contained in the "hub." *See supra* note 198. The current Article 2 provisions on unconscionability are discussed herein. *See* U.C.C. § 2-302 cmt. 1 (1995) (Code does not define unconscionability but indicates that "[t]he principle is one of the prevention of oppression and unfair surprise.").

[224] *See supra* text accompanying notes 58–60.

[225] *See supra* text accompanying note 60.

There is also some empirical evidence to support this contention. The price that a licensee pays for software is a function of the rights granted by the licensor. A customer may license an object code copy of IBM's AIX Version 3.2.5 for between $650 and $12,100, with the price dependent upon the number of users and the size of the machine for which the object code is licensed.[226] This license includes a prohibition against decompilation[227] that prevents the licensee from accessing the source code version of the program.[228] A customer may, however, license selected portions of the source code of AIX from IBM directly. The cost of a source code license is $60,000, and the license requires agreement to strict confidentiality provisions.[229] This example suggests that if courts were not to enforce contractual prohibitions against decompilation, there would be some upward pressure on the price of software. It is difficult to assess the magnitude of that increase, but one can conclude that all licensees—those who wish to decompile as well as those who do not—would be paying more.

Perhaps a better approach would be to enforce these provisions to keep the price of software lower for all[230] and put the

[226] IBM, RISC SYSTEM/6000, at 35 (Spring 1995) (on file with author) (listing license fees, ranging from $800-$14,700, for AIX Version 4.0 object code); *see also* OSF/1 1.3 PRICE LIST, *supra* note 57 (setting forth $125,000 charge for initial CPU source code license of OSF/1 and $65 royalty for each object code copy distributed).

[227] The licensee must sign the IBM Customer Agreement. See *supra* note 10 for relevant provisions.

[228] *See supra* text accompanying notes 42–44.

[229] IBM, PROGRAMMING ANNOUNCEMENT 291-335, IBM AIX VERSION 3 SOURCE CODE FOR RISC SYSTEM/6000 3 (July 2, 1991); see also *supra* note 57 for an outline of the applicable confidentiality provisions.

[230] But see Rice, *supra* note 17, at 599, who argues that expansive software copyright in the form of agreements prohibiting reverse engineering

> erects formidable barriers to interbrand competition and the development of compatible products. The predictable consequence is that supracompetitive pricing will persist rather than being a short-term phenomenon. The prevalence of this market condition also is indicated by increasing industry concentration and the aggressive use of copyrights to limit competition and to exact royalties or cross-licensed use of competitors' technology.

However, the trend in software prices has generally been downward. *See, e.g.*, Jeffrey M. O'Brien, *The New Pricing Model*, MARKETING COMPUTERS, Jan. 1, 1996, at 36 (describing market forces contributing to a "precipitous" drop in software prices); *see also* Charles A. Radin, *Psst, Want Some Software Cheap?*, BOSTON GLOBE, Feb. 28, 1995, at 61 (stating that software prices are coming down as a result of software piracy). Of course, a downward trend in price may indicate that prices are moving from a high supra-competitive price to a lower one that is still supra-competitive. However, note that merely because an industry shows a trend toward consolidation does not mean that it is noncompetitive. In fact, consolidation can indicate an efficient market as small, entrepreburden on licensees who have a need to decompile to contact the vendor and disclose to it the reason for the desired decompilation. The parties could then negotiate the appropriate terms for their transaction and price it accordingly. Admittedly, most consumers are unlikely to be able to bargain effectively to such an agreement.[231] However, this does not necessarily dictate the conclusion that standard form prohibitions against decompilation should not be enforced. While § 301 on first analysis does seem to preempt these terms in the standard form context, there is a tenable argument based on contract law and market realities suggesting that § 301 preemption does not apply given the implied consent of the licensee. Yet, other policies must be examined to complete the preemption analysis as there is a strong presumption that licensees of publicly distributed products are in fact really "purchasers" of a product who should be free to do with that product as they please, as long as they do not infringe any applicable intellectual property right. This leads directly to a consideration of preemption based on constitutional concerns.

### B. *Constitutional Preemption*

In addition to statutory preemption, preemption under the Supremacy Clause of the Constitution must be examined. Even if a particular cause of action survives a § 301 preemption analysis, it still must be evaluated for consistency with constitutional concerns because it still may be preempted if it "stands as an obstacle to the accomplishment of the full purposes and objectives of Congress."[232] The preceding analysis identifies that objective as striking a balance between providing an incentive to create through the grant of a limited statutory monopoly in the form of copyright and maintaining the free flow of the information on which such cre-

neurial firms are weeded out in a Darwinian survival of the fittest. Moreover, some commentators have suggested that at least a minimally concentrated market is one that maximizes innovation. *See, e.g.*, MORTON I. KAMIEN & NANCY L. SCHWARTZ, MARKET STRUCTURE AND INNOVATION 53 (1982) (asserting that intermediate degree of industry concentration produces highest volume of research and development).

[231] *See supra* note 161.

[232] 1 NIMMER & NIMMER, *supra* note 63, § 1.01[B], at 1–44.12 (quoting Hines v. Davidowitz, 312 U.S. 52, 67 (1941), although *Hines* predated 1976 Act which incorporated specific preemption section); *see also* Wendy J. Gordon, *On Owning Information: Intellectual Property and the Restitutionary Impulse*, 78 VA. L. REV. 149, 156 n.22 (1992) (inquiring whether state law that interferes with congressional intentions should survive the existence of § 301).

ativity is built.[233] The fair use doctrine is an essential part of this balancing approach. It attempts to correct market inefficiencies inherent in the grant of a monopoly by sanctioning uses that otherwise might be infringing when doing so is necessary to strike the appropriate balance between the rights of authors and those of the public. Therefore, regardless of the applicability of § 301, a court might set aside as constitutionally preempted a licensing scheme that frustrates the Act's overall objectives by contractually prohibiting the licensee from engaging in uses that would otherwise be sanctioned by the fair use provisions of the Act.

Again, it may be instructive to examine this issue from the two perspectives of (1) the negotiated license, and (2) the non-negotiated standard form license. In the case of the negotiated license, the inquiry would proceed in the same manner as noted under § 301 and should result in the same finding, namely that such provisions are not preempted in the negotiated agreement. The constitutional inquiry regarding the standard form license also will resemble that under § 301. However, it is also likely to draw by analogy on patent preemption principles that state a stronger case for preemption.

1. Vault v. Quaid. In *Vault Corp. v. Quaid Software Ltd.*, the Fifth Circuit held that a decompilation provision in a standard form license agreement was constitutionally preempted.[234] Vault marketed an anticopying software program called PROLOK.[235] With each copy of PROLOK, Vault included a standard form shrink wrap agreement that contained a prohibition against decompilation.[236] Quaid developed a program called "CopyWrite"

[233] *See supra* note 24 and accompanying text.

[234] 847 F.2d 255, 269–70 (5th Cir. 1988).

[235] *Id.* at 256–57. Software providers were Vault's primary customers. The providers would use Vault's program to prevent their customers from making unauthorized functioning copies of their program. For example, assume Company A markets a spreadsheet called 4-5-6. Company A could license PROLOK from Vault and run it in conjunction with 4-5-6. The disks that Company A then marketed to its customers with the PROLOK feature installed could be copied. However, a computer would not execute the 4-5-6 software unless the original 4-5-6 diskette purchased from Company A were in the drive. Thus, customers could not purchase one copy of a program and then make unauthorized functional copies for others. *Cf. supra* notes 17–19 and accompanying text (describing public goods problem).

[236] *Vault*, 847 F.2d at 257 n.2 ("Vault's license agreement reads . . . [1] You may not transfer, sublicense, rent, lease, convey, copy, modify, translate, convert to another programming language, decompile or disassemble the Licensed Software for any purpose

that defeated the anticopying features of PROLOK.[237] Vault brought a number of claims against Quaid, including a breach of contract claim based on Quaid's decompilation and disassembly of the PROLOK code.[238]

Vault's breach of contract claim was brought under Louisiana law, including the Louisiana Software License Enforcement Act ("Louisiana Act").[239] The Louisiana Act sought to clarify the enforceability of shrink wrap agreements by setting forth acceptable terms.[240] The district court opined that the shrink wrap was a "contract of adhesion which could only be enforceable if the [Louisiana Act] is a valid and enforceable statute."[241] The district court found a number of conflicts between the Louisiana Act and the Copyright Act,[242] and concluded that because the Louisiana Act "touched upon the area" of federal copyright law, its provisions were preempted and Vanlt's license agreement was unenforceable.[243]

without VAULT's prior written consent.").

[237] CopyWrite contained a feature to unlock the protection provided by PROLOK. With the PROLOK feature defeated, the original diskette containing the software manufacturer's program no longer had to be in the computer for the program to run. Without PROLOK protection, users could make unlimited fully functional copies of disks. *Id.* at 257.

[238] Vault's other claims included copyright infringement and trade secret misappropriation. *Id.* at 258.

[239] *Id.*

[240] As described by the court,

> Enforceable terms include the prohibition of (1) any copying of the program for any purpose; and (2) modifying and/or adapting the program in any way, including adaptation by reverse engineering, decompilation or disassembly. LA. REV. STAT. ANN. § 51:1964. The terms "reverse engineering, decompiling or disassembling" are defined as "any process by which computer software is converted from one form to another form which is more readily understandable to human beings . . . ." LA. REV. STAT. ANN. § 51:1962(3).

*Vault*, 847 F.2d at 268-69 (footnote omitted).

[241] Vault Corp. v. Quaid Software Ltd., 655 F. Supp. 750, 761 (E.D. La. 1987), *aff'd*, 847 F.2d 255 (5th Cir. 1988).

[242] The conflicts identified included:

> (1) while the License Act authorizes a total prohibition on copying, the Copyright Act allows archival copies and copies made as an essential step in the utilization of a computer program, 17 U.S.C. § 117; (2) while the License Act authorizes a perpetual bar against copying, the Copyright Act grants protection against unauthorized copying only for the life of the author plus fifty years, 17 U.S.C. § 302(a); and (3) while the License Act places no restrictions on programs which may be protected, under the Copyright Act, only "original works of authorship" can be protected, 17 U.S.C. § 102.

*Id.* at 269 (citing *Vault*, 655 F. Supp. at 762-63).

[243] *Id.* (quoting *Vault*, 655 F. Supp. at 763).

Interestingly, the appeals court did not address § 301 preemption, but instead proceeded directly to consider constitutional preemption, drawing on patent preemption cases to assist its analysis.[244] The analysis itself was quite sparse, simply stating that

> [s]ection 117 of the Copyright Act permits an owner of a computer program to make an adaptation of that program provided that the adaptation is either "created as an essential step in the utilization of the computer program in conjunction with a machine," § 117(1), or "is for archival purpose only," § 117(2). The provision in Louisiana's License Act, which permits a software producer to prohibit the adaptation of its licensed computer program by decompilation or disassembly, conflicts with the rights of computer program owners under § 117 and clearly "touches upon an area" of federal copyright law. For this reason, and the reasons set forth by the district court, we hold that at least this provision of Louisiana's License Act is preempted by federal law, and thus that the restriction in Vault's license agreement against decompilation or disassembly is unenforceable.[245]

*Vault*, then, would seem to have resolved the issue of constitutional preemption by holding that decompilation provisions in standard form licenses are preempted. However, developments since *Vault* suggest that its holding is of limited precedential value. First, the statement that shrink wrap agreements would not be enforceable in the absence of an enabling statute is not necessarily correct. In addition to the fact that the NCCUSL is considering revising the UCC to provide explicitly for the enforceability of standard form agreements, at least one court has upheld a shrink wrap agreement under common law contract and the current UCC.[246] Thus, the preemption issue is more likely to arise in the context of state enforcement of private contractual rights rather than state enforcement of specific statutory rights.[247]

[244] *Id.* at 269-70 (citing patent preemption cases such as *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225 (1964), and *Compco Corp. v. Day-Brite Lighting, Inc.*, 376 U.S. 234 (1964)).

[245] *Id.* at 270 (footnote omitted).

[246] Arizona Retail Sys., Inc. v. Software Link, Inc., 831 F. Supp. 759, 763 (D. Ariz. 1993); see *supra* note 213 for a summary of the holding of the case.

[247] This distinction may have only a limited substantive effect on the preemption analysis. On the surface, it seems absurd to suggest that parties could achieve, through private contract, the same end resnlt that the state itself could not achieve through legislation. For example, no one would seriously suggest that individual states could repeal the fair use provision of the Copyright Act. Yet, by enforcing provisions against reverse

Second and perhaps more importantly, *Sega* rejected *Vault*'s reliance on § 117 as sanctioning a right to decompile.[248] *Sega*'s holding was grounded in a more sophisticated understanding of computer technology as well as an examination of the history behind the enactment of the section.[249] Thus, since *Vault*, the preemption conflict has been recharacterized not as a clash between § 117 and private contractual provisions, but between the limited *Sega/Atari* fair use rights and private contract.

2. Bonito Boats *and Patent Preemption.* *Vault*, however, still may be useful for its reliance on patent preemption cases. As the number of software patents has increased,[250] the collision between private contract and public intellectual property law in the form of patent has become increasingly apparent.[251] Unlike copy-

engineering through state contract law, whether common law or the UCC, the state would essentially allow private parties to legislate a repeal of fair use. However, there are some relevant differences between public and private acts. A state legislative act that repealed fair use would have an impact on all contracts within the state and would proceed from the authority granted it by the general consent of the governed. A private contract in which one party surrenders its fair use rights implicates other contracts only in the sense that state enforcement would signal to other parties that such a provision is acceptable—it would not, of its own accord, compel others to surrender fair use rights. Additionally, the private contract is based on specific consent to a particular transaction. Because of this specific consent and based on the policy considerations identified above, these private contracts do not present the same conflict with federal policy as state legislation would. However, it is easy to see why the shrink wrap, which has characteristics of private legislation, has a tougher time surviving a preemption analysis than a negotiated contract.

[248] Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520 (9th Cir. 1992) ("Section 117 does not purport to protect a user who disassembles object code, converts it from assembly into source code, and makes printouts and photocopies of the refined source code version.").

[249] For example, the *Sega* court recognized that

> [s]ection 117 was enacted on the recommendation of CONTU, which noted that "[b]ecause the placement of any copyrighted work into a computer is the preparation of a copy [since the program is loaded into the computer's memory], the law should provide that persons in rightful possession of copies of programs be able to use them freely without fear of exposure to copyright liability." CONTU Report at 13. We think it is clear that Accolade's use went far beyond that contemplated by CONTU and authorized by section 117.

*Id.*; *see also* McManis, *supra* note 23, at 83–85 (setting forth criticisms of *Vault* and noting that some have called case "rogue elephant" in construction of § 117).

[250] Flex H. Kent & Lawrence M. Hertz, *Establishing a Foothold in Cyberspace*, N.Y. L.J., Apr. 21, 1995, at 3, 30 (describing increase in issuance of software patents from 3,600 in 1993 to 4,500 in 1994 to an expected 5,500 in 1995); *see also* OTA REPORT, *supra* note 16, at 55.

[251] This Article focuses on copyright preemption, addressing patent preemption briefly in the context of the publicly distributed product. *See infra* text accompanying notes

right preemption, patent preemption is based solely on the Constitution; there is no section of the Patent Act setting forth statutory preemption.[252] Thus, it may be helpful to analyze the principles underlying patent preemption cases and use them as a guide in discussing constitutional copyright preemption of provisions contained in licenses accompanying mass-marketed software.

While Supreme Court decisions discussing patent preemption have not always been a model of consistency,[253] *Bonito Boats Inc. v. Thunder Craft Boats, Inc.*[254] addresses facts most analogous to those involved in the case at hand—mass-market distribution of patentable material. Bonito Boats had developed a hull design for a boat.[255] The manufacturing process involved creating a mold that then served to produce the hull to be used on the finished product.[256] Bonito did not file a patent application for either the molding process or the utilitarian or design aspects of the hull.[257] Under a Florida statute that prohibited the use of a direct molding process to duplicate unpatented boat hulls,[258] Bonito filed an action against Thunder Craft. Bonito alleged that Thunder Craft had violated the statute by "using the direct molding process to duplicate the Bonito . . . fiberglass hull."[259]

254–67. However, the same considerations that support copyright nonpreemption in the context of negotiated agreements do so with respect to patent preemption. Moreover, market evidence supports patent nonpreemption of negotiated agreements. The number of software patents has consistently increased, *see supra* note 250, so there should be little concern that software providers are foregoing patent protection in favor of trade secret protection. Therefore, the conflict between patent and trade secret law is minimal in the negotiated license agreement context and enforcement of those agreements would be fully consistent with patent policies. Some commentators, however, disagree with this assessment. *See* Rice, *supra* note 17, at 577–91. *But see* Mark I. Koffsky, Note, *Patent Preemption of Computer Software Contracts Restricting Reverse Engineering: The Last Stand?*, 95 COLUM. L. REV. 1160, 1161 (1995) (contending that, as a general rule, patent law should not preempt state enforcement of private contracts).

[252] Rice, *supra* note 22, at 1198 ("Although not based in statute, patent law preemption of state law intrusion into its domain is well established.").

[253] Rice, *supra* note 17, at 574–75 (setting forth chronology of important Supreme Court cases).

[254] 489 U.S. 141 (1989).

[255] *Id.* at 144.

[256] *Id.*

[257] *Id.*

[258] *Id.* at 144–45 ("The statute makes '[i]t . . . unlawful for any person to use the direct molding process to duplicate for the purpose of sale any manufactured vessel hull or component part of a vessel made by another without the written permission of that other person.' ").

[259] *Id.* at 145.

The Supreme Court held that the Florida statute was preempted, noting that "[f]rom the Patent Act of 1790 to the present day, the public sale of an unpatented article has acted as a complete bar to federal protection of the idea embodied in the article thus placed in public commerce."[260] A state may not, by statute, "prevent[] the exploitation of the design and utilitarian conceptions embodied in [a] product itself"[261] because such state protection "conflicts with the federal policy 'that all ideas in general circulation be dedicated to the common good unless they are protected by a valid patent.' "[262] Furthermore, the fact that the Florida statute removed only one means of reverse engineering from Bonito's competitors and the public did not save it:

> In essence, the Florida law prohibits the entire public from engaging in a form of reverse engineering of a product in the public domain. . . . If Florida may prohibit this particular method of study and recomposition of an unpatented article, we fail to see the principle that would prohibit a State from banning the use of chromatography in the reconstitution of unpatented chemical compounds, or the use of robotics in the duplication of machinery in the public domain.[263]

*Bonito Boats* strongly suggests that prohibitions against decompilation accompanying mass-marketed products, whether enforced as a matter of state statute or contract law, are preempted under patent law.[264] The relationship between patent and copyright preemption is unclear.[265] However, given that the case under the Copyright Act for upholding decompilation provisions in standard form contracts is tenuous at best,[266] it seems likely that patent policy would influence the analysis and that the provisions would be constitutionally preempted in the copyright context as well.[267]

[260] *Id.* at 148–49.

[261] *Id.* at 158.

[262] *Id.* at 159–60 (citing Lear, Inc. v. Adkins, 395 U.S. 653, 668 (1969)).

[263] *Id.* at 160.

[264] *See* McManis, *supra* note 23, at 94–95; Rice, *supra* note 17, at 595.

[265] *See* McManis, *supra* note 23, at 95 (citing commentators suggesting that patent system is traditionally thought of as preempting copyright).

[266] *See supra* Section III(A)(2).

[267] *But see* Koffsky, *supra* note 251, at 1161. Despite the *Bonito Boats* holding, the case for patent preemption in the context of nonnegotiated license agreements should be stronger with respect to prohibitions against reverse engineering than those against decompilation. *See supra* text accompanying notes 118–21 and *infra* note 278; *see gener-*

Since the policies underlying the Copyright Act cut both ways in this debate, even in the matter of mass-marketed software, this approach of wholesale preemption or nonpreemption may not be the best. Rather, a more nuanced analysis that places the Copyright Act within the context of other federal law should be conducted. This analysis suggests that a more detailed inquiry is appropriate and that copyright policy could best be served by upholding the provisions unless doing so would allow the software provider to expand its limited copyright monopoly beyond the market to which that monopoly was intended to apply.

## IV. A Proposal for an Integrated Preemption Analysis

As a purely empirical matter, it is difficult to assess whether or not there would be an increase in creative expression if contractual provisions against decompilation were held preempted and unenforceable. Software providers may be less willing to develop and market products if they fear that those products may not be protected sufficiently to be profitable. Also, if providers may not rely on private contract to establish the minimum level of protection they desire, they may divert resources into making decompilation technically more difficult, potentially slowing the availability of new technology.[268]

On the other hand, if these provisions are held enforceable, there may be some decline in the availability of competing products—although software providers may be more willing to invest in development. New and potentially improved expression may be foregone because other developers are unable to obtain the source code of their competitors either because (1) such competitors refuse to negotiate a license at all or (2) they will only grant a license that effectively contracts around the Act's *Atari/Sega* rule by prohibiting decompilation. Without such source code, competitors will likely be unable either to compete directly with another developer's product[269] or to produce a product that runs with the

*ally infra* Part IV.

[268] Of course, new ideas and technologies, such as advances in cryptography, could be discovered in the process of developing new programs to make decompilation more difficult. However, the rate of availability of end-user applications may slow.

[269] Essentially, without the complete interface specification of a software product, which a provider is unlikely to publish, *see supra* text accompanying notes 148–50, the competitor would be unable to "clone" the system and thereby replace it in the market.

other developer's.[270] It is impossible to know where the balance between creativity and competition would settle if either rule—preemption or nonpreemption of decompilation provisions—were adopted.[271]

However, it *is* possible to identify those situations in which the Act's policy concerns are primarily implicated and, based on that identification, to propose an alternative preemption analysis to maximize the availability of creative activity in the form of new products.[272] The software market may be analogized to the video game market at issue in *Atari* and *Sega*. Operating system software that drives computer hardware is similar to the device and locking code that Nintendo and Sega had written into their consoles. Operating system software provides access to the functionality of the hardware in much the same way as Nintendo's and Sega's device and locking codes provide access to their consoles. In the same manner that Atari's and Accolade's video game cartridges had to contain code to unlock the Nintendo and Sega consoles, respectively, any application that runs on an operating system must utilize certain operating system functions in order to execute success-

For example, it is extremely difficult to develop a product that runs all Microsoft Windows applications in the absence of a detailed design specification for Windows, the Windows source code, or assistance from Microsoft.

[270] *See, e.g., supra* Section II(A) (describing *Sega* and *Atari* cases in which video game competitors were unable to develop games running on dominant console in absence of ability to decompile); *see also supra* note 158 (discussing allegations that Microsoft intentionally hides interfaces to give its application developers an advantage over competitors in writing programs to run on Windows).

[271] One commentator discusses the *Sega* court's view that too broad intellectual property protection would slow innovation in the software industry, which advances by building on earlier technologies, but then asserts that "[i]f *Sega* results in attempts by manufacturers to hinder reverse engineering, then consumer welfare will suffer." Leaffer, *supra* note 56, at 1095–97.

[272] Generally, copyright law's goal could be viewed as promoting the free exchange of ideas or developing and marketing new works. Usually, these two goals go hand in hand. *See, e.g.*, Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1523 (9th Cir. 1992) (stating that copyright law promotes "growth in creative expression, based on the dissemination of other creative works and the unprotected ideas contained in those works"); Ramsey Hanna, Note, *Misusing Antitrust: The Search for Functional Copyright Misuse Standards*, 46 STAN. L. REV. 401, 419–20 (1994) ("The Supreme Court views the *dissemination* of creative works as a concomitant goal of eopyright law, recognizing that the creation of innovative works does little to further the public interest if those works are not widely accessible to potential beneficiaries."). The proposal suggested here focuses primarily on increasing the availability of new products, but it should also have a salutary effect on total idea dissemination generally because ideas that may be foregone by an inability to reverse engineer are replaced by ones eontained in new products that otherwise would not have been available.

fully on it.[273] Moreover, just as Accolade had to modify its games to run on Sega's Genesis system, an application developer must make changes to its software to allow it to run on an operating system other than the one for which it was designed.[274]

The success of a particular operating system is, in large part, determined by the quality and quantity of applications available to run on it. This gives operating system providers an incentive to disclose enough information to allow application developers to write functioning code for that operating system.[275] Thus, under most circumstances, the market functions well enough to provide the requisite information to further both creativity and competition.[276]

However, the market may not function to effectuate the Act's goals when a particular operating system provider has obtained near monopoly power in the relevant market.[277] When an operating system provider obtains near monopoly power, it has the ability to restrain competition in the operating system market through a variety of exclusionary practices, such as refusing to deal with competitors[278] or engaging in predatory innovation.[279] Addition-

[273] *See supra* notes 124–125 and accompanying text.

[274] *See supra* note 149.

[275] *See supra* text accompanying notes 154–59.

[276] However, note that providers will withhold information that would allow cloning of their system. *See supra* note 150.

[277] Note that while the textual statement is simple, defining "monopoly power" and "relevant market" is not. Monopolization claims often turn on the definition of the relevant product and geographic markets. *See* 3 JULIAN O. VON KALINOWSKI, ANTITRUST LAWS AND TRADE REGULATION § 19.02[2], at 19-27 to -28 (2d ed. 1995). Both market share, which varies depending on the "relevant market" examined, and direct evidence of the use of market power to raise prices or exclude competition are used in determining whether a defendant has monopoly power. *Id.* § 19.02[3], at 19-56 to -60. However, the Supreme Court has never found a defendant with a market share less than 75% to possess monopoly power. *Id.* at 19–60 to -61.

[278] *Id.* § 19.05[3], at 19–100 to -09 (Generally, a manufacturer is free to choose parties with whom it will deal unless such conduct is anticompetitive; in assessing refusal to deal, one should examine the business reasons that may explain such a refusal.).

> The Supreme Court has established that the monopolist has no duty to cooperate with others and that in the vast majority of cases, a monopolist may "deal with whom he pleases." Nevertheless, "[t]he absence of an unqualified duty to cooperate does not mean that every time a firm declines to participate in a particular cooperative venture, that decision may not have evidentiary significance or that it may not give rise to liability in certain circumstances." Thus, in some circumstances, a monopolist's refusal to deal with a party seeking business ties may constitute, or be evidence of, illegal monopolization.

Page, *supra* note 158, at 109 (footnotes omitted); *see also infra* text accompanying notes 290–96 (discnssing particular type of refusal to deal where refusal involves essential facilially, an operating system provider often also competes in the application market—in much the same way that Nintendo and Sega both competed in not only the game console market but also the game cartridge market. For example, Microsoft markets both operating systems such as MS-DOS and Windows and applications like Word for Windows that run on those operating systems. In the same way that Nintendo and Sega could "lock out" competitors by refusing to grant access to the console, such software providers may "lock out" competition in the application market by refusing to provide information to allow application developers to write functioning code. Alternatively, they may provide sufficient information to permit development of functional applications, but withhold certain information to give their own applications a competitive advantage.[280]

When a particular operating system has attained a certain market share, other operating system providers are unable to compete effectively with the dominant system because most applications will be written to run on that dominant system.[281] Applica-

ty). Here the difference in meaning between "reverse engineering" and "decompilation" becomes relevant. *See supra* notes 118–21 and accompanying text. The monopolist who provides software only with provisions against reverse engineering bars the licensee from all means of understanding how the monopolist's software works, while the monopolist who includes a provision against decompilation has removed only one means through which the licensee might understand the program. As a general rule, the latter monopolist seems less blameworthy and less likely to have engaged in predatory behavior than the former.

[279] 3 VON KALINOWSKI, *supra* note 277, § 21.04[2], at 21–69 to -74 (Predatory innovation is rarely successful in demonstrating liability; liability is most likely found when a dominant firm makes design changes to drive out competition.). This doctrine supports the argument, *infra* note 297 and accompanying text, that once liability is established, the court could order disclosure of interface changes to ensure the competitor access to the market. This would eliminate some of the lead time the dominant developer has and also prevent it from intentionally disabling competitive products through design changes.

[280] *See supra* note 158 (discussing allegations against Microsoft of such activity).

[281] This essentially describes the position in which IBM found itself with its OS/2 operating system product. Because so many applications were available for Microsoft's Windows, customers were unwilling to switch to OS/2, and thus its market share remains minimal. *See supra* note 154. IBM undertook a campaign to entice application developers to write OS/2 applications, but was not particularly successful. *See supra* note 157. OS/2 Version 2.0 is touted as running Windows 3.1 applications but, since the exchange of interface specifications between IBM and Microsoft has stopped, *see supra* note 153, it is unclear whether future versions of OS/2 will be able to run Windows applications designed to run on Windows versions subsequent to Version 3.0 such as Windows95. This fact in and of itself does not demonstrate any wrongdoing on Microsoft's part. However, if it is true that Microsoft error messages appear in Windows when operating systems other than MS-DOS are used, this may indicate some predatory intent because it discour-

tion providers, to maximize revenue, must make their applications available on the dominant system. However, the dominant operating system provider that has achieved such market share may, if it also writes applications, block the access of other application providers to the hardware if it is permitted contractually to restrict others from accessing code to obtain the necessary interface specifications.[282]

## A. *The Proposal*

These market realities suggest that although freedom of contract should prevail as a general rule, when certain market imperfections exist, courts should be more willing to examine contracts for consistency with the Act's goals and to find decompilation provisions preempted. In short, the Act itself is premised on a market analysis because it attempts to adjust for certain market imperfections.[283] Yet the Act does not exist in a vacuum, but rather is just one part of federal law aimed at correcting market failures. Perhaps the most comprehensive federal scheme addressing market imperfections is that of antitrust law. Thus, when informing the copyright analysis by examining market factors, courts should focus on long-standing antitrust principles that may be applied to the software market by analogy to inform the preemption inquiry. In particular, antitrust doctrine on monopolies, essential facilities, tying, and leveraging are instructive in analyzing the software market and in formulating guidelines to assist the courts in the preemption analysis.

Monopoly power in and of itself is not an antitrust offense. Under section 2 of the Sherman Act,[284] the Supreme Court has identified two elements comprising the offense of monopoly: (1) the possession of monopoly power in the relevant market;[285] and

ages customers from buying other operating systems. *See* Page, *supra* note 158, at 104.

[282] *Cf.* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1523–24 (9th Cir. 1992) ("[A]n attempt to monopolize the market by making it impossible for others to compete runs connter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine.").

[283] *See supra* text accompanying notes 15–24.

[284] 15 U.S.C. § 2 (1994) ("Every person who shall monopolize, or attempt to monopolize, or combine or conspire with any other person or persons, to monopolize any part of the trade or commerce among the several States, or with foreign nations, shall be deemed guilty of a felony . . . .").

[285] Mere possession of a copyright or patent is not enough for a court to presume

(2) the willful acquisition or maintenance of that power as distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident.[286] Thus, an element of deliberateness is necessary to sustain successfully an action sounding in monopolization.[287] Moreover, the particular conduct demonstrating willfulness does not have to be illegal: Conduct that otherwise would be acceptable may be evidence of illegality when engaged in by a monopolist.[288]

For example, manufacturers generally are free to choose the parties with whom they deal, but antitrust law may impose a duty to deal when the manufacturer controls an "essential facility." Because of the singular importance that knowledge of the interface specifications of the dominant operating system has to the ability to enter the software applications market, the "essential facilities" doctrine is particularly helpful in informing the preemption analysis. The essential facilities doctrine, although an oft-used but seldom successful antitrust claim,[289] perhaps most closely describes the circumstances of the software market and their potential for harm to competition.

Most simply put, "the 'essential facilities' approach under Section 2 of the Sherman Act finds a violation when a monopolist refuses to provide a competitor with reasonable, non-discriminatory access to some 'facility' that is essential to effective competition."[290] Under the essential facilities doctrine, a defendant will be held liable when the plaintiff proves the following:

> (1) Control of an essential facility by a monopolist;
> (2) A competitor's inability reasonably or practically to duplicate the essential facility;
> (3) Denial of the use of the facility to a competitor; and

monopoly power in a § 2 case. *See generally* Data Gen. Corp. v. Grumman Sys. Support Corp., 36 F.3d 1147, 1185–86 (1st Cir. 1994) ("[The] Supreme Court has suggested that an otherwise reasonable yet anti-competitive use of a copyright should not 'be deemed a *per se* violation of the Sherman Act' . . . . [And] [o]ther courts have specifically held that a monopolist's unilateral refusal to license a patent is ordinarily not properly viewed as exclusionary conduct.")

[286] 3 VON KALINOWSKI, *supra* note 277, § 19.01[2], at 19–4 (citing United States v. Grinnell Corp., 384 U.S. 563, 570–71 (1966)).

[287] *Id.* at 19–5.

[288] *See, e.g., supra* note 278.

[289] 3 VON KALINOWSKI, *supra* note 277, §19.05[3], at 19–124 (discussing 38 "essential facilities" cases, of which only eight resulted in verdicts for plaintiff).

[290] Page, *supra* note 158, at 109.

(4) Providing the competitor access to the facility is feasible.[291]

It is appropriate to use this doctrine to address the market distortions caused by the dominant operating system provider in both the operating systems and applications markets.[292] By virtue of the operating system provider's monopoly power, its interface becomes an essential facility because access to it is necessary for others to compete.[293] Additionally, the competitor will be unable to practically duplicate the interface because documentation is incomplete; even if the competitor can obtain a license to the operating system, it will contain a decompilation prohibition.[294] Finally, providing interface access to competitors is feasible.[295]

If the operating system interface is an essential facility, competitors should be afforded access to it. In such situations, a contractual provision against decompilation should not be enforced.[296] Such enforcement harms competition by blocking other

[291] 3 VON KALINOWSKI, *supra* note 277, § 19.05[3], at 19-111.

[292] Although the essential facilities doctrine is normally used when a manufacturer possesses monopoly power, it may be appropriate to employ the doctrine in the software industry when the operating systems provider has attained a near-monopoly share—enough so that it has power to force licensees to accept restrictive terms. Minimally, the operating systems provider should have market share sufficient to support a tying claim. See *infra* note 300 for a discussion of what market share supports a tying claim.

[293] *See* 3 VON KALINOWSKI, *supra* note 277, § 19.05[3], at 19-116 to -118 (Facility is not essential if (1) equivalent ones exist; or (2) the benefits of access are obtainable in another manner; or (3) access is not vital to ability to compete.). While these are stringent requirements, they are likely to be met in the factual situation outlined in the text. The information could only be obtained by decompilation, which is forbidden under the license agreement (if the competitor is able to license the operating system at all).

[294] *Cf. id.* at 19-119 to -20 (noting that plaintiff need not prove that duplication is impossible, only economically infeasible). Theoretically, using reverse engineering techniques other than decompilation, a programmer might be able to clone the system by trial and error. This process would, however, likely be prohibitively expensive. *See supra* note 69.

[295] *But see* Page, *supra* note 158, at 109 (contending that while essential facilities charge seems worthwhile, Microsoft could probably assert legitimate business justification—"the desire to reap the benefits from its technological innovation and research and development investment"). This point is debatable and does not seem particularly compelling if the facility is truly essential. Moreover, nothing in the essential facilities doctrine says that the party controlling access has to grant others access free of charge. Thus, it may make its return on investment from access fees, provided that they are not so high as effectively to continue to deter competition. It would be relevant to look at the behavior of the alleged offender in assessing such a defense to evaluate any seemingly predatory conduct.

[296] *See supra* notes 118-21 and accompanying text (suggesting that courts should

developers from access to the facility. Moreover, the courts might consider additional remedies to be appropriate. For example, a court could impose a compulsory disclosure order on the offender, requiring it to disclose interface changes in advance of their implementation[297] and enjoining it from changing its interface in a manner that would render a large number of existing applications incompatible with the revised version.

While the essential facilities doctrine could assist either the competitor who wishes to clone *or* be compatible with the dominant provider's operating system, the antitrust doctrines of leverage and tying may provide additional assistance to application developers seeking compatibility with the' dominant operating system. The concepts of leverage and tying are designed to assure that a party with power in one market does not use that power to gain an advantage in another. It is appropriate to apply these concepts to the preemption analysis because the Copyright Act grants a *limited* monopoly that is not intended to permit the copyright owner to leverage its statutory monopoly into another market.[298]

A firm should not be permitted to use its power in the market for the copyrighted work, the operating system, to gain an advantage in another market—the application program market. This conduct could be termed leveraging or tying.[299] Generally, a firm is engaged in illegal tying if

> (a) [it] has conditioned the purchase of one product on the sale of another, (b) a substantial volume of commerce is foreclosed, (c) [it] has sufficient market power in the tying [operating systems] market to force purchases in the tied [applications] mar-

recognize the difference between provisions against reverse engineering and those prohibiting only decompilation).

[297] Withont such an order, the mere right to decompile is likely to be meaningless because competitors would always lag behind the first-to-market software provider who controls the essential facility. *Cf.* Page, *supra* note 158, at 121 (arguing that Antitrust Division "should require Microsoft to test its MS-DOS with all applications software and give applications software developers the information they need to make their applications compatible").

[298] Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 429 (1984) (stating that copyright may not be used to extend monopoly power beyond that granted by statute).

[299] *See* 3 VON KALINOWSKI, *supra* note 277, § 19.05[3], at 19-125 to -33 (noting conflict about whether "leveraging" conduct must provide advantage in second market that is close to monopoly, and suggesting that tying may be more appropriate cause of action).

> ket,[300] and (d) the products involved are really two separate products.[301]

One commentator suggests that

> tying charges would have addressed the allegations that Windows can only be run on MS-DOS. Microsoft's competitors essentially allege that the company uses its dominance to obtain a monopoly in the applications software market. Because most PC consumers use Windows, they are forced to buy software that is compatible with it, which is usually also marketed by Microsoft.[302]

This analysis may represent a slight expansion of present tying law. Generally, to sustain a tying claim the sale of the tying product (the operating system) must be conditioned on purchase of the tied product (the application). This is not normally the case in the software market. For example, Microsoft does not force its Windows licensees to purchase its spreadsheet, Excel. There is no coercion in the sense that the licensee is free to use only Windows. As a practical matter, however, there may be "market" coercion given the limited number of spreadsheets available running on Windows and Microsoft's popularity within the spreadsheet market. Excel's popularity, standing alone, should not be conclusive evidence that the sale of Windows is conditioned on the sale of Excel for the purposes of a tying analysis. Only if Excel's popularity is a result of Microsoft's exercise of market power to refuse spreadsheet developers access to its Windows interface

[300] Courts have held that possession of a patent or copyright on the tying product may be sufficient to generate the presumption of requisite economic power. United States v. Loew's, Inc., 371 U.S. 38, 45 n.4 (1962). However, this presumption has been criticized. *See* J. Dianne Brinson, *Proof of Economic Power in a Sherman Act Tying Arrangement Case: Should Economic Power Be Presumed When the Tying Product is Patented or Copyrighted?*, 48 LA. L. REV. 29 (1987). Generally, possession of monopoly power in the market for the tying product is not required to sustain a tying action, but market share of 30% or less will probably be held insufficient. Plaintiffs generally attempt to show that

> (1) the defendant has sufficient market share to force buyers to purchase the tied product; (2) a high percentage of the seller's customers have accepted the tying arrangement, for which there is no valid explanation; or (3) the defendant has a competitive advantage due to special characteristics of the tying product or to legal barriers, such as a trademark, copyright, or patent.

2 VON KALINOWSKI, *supra* note 277, § 6J.02[4], at 6J-35; *see also id.* at 6J-29 to -43 (collecting and synthesizing Supreme Court authority in tying context and attempting to abstract consistent principles therefrom).

[301] Page, *supra* note 158, at 108.

[302] *Id.*

specifications should it be evidence of "forcing" as part of a tying analysis.

While antitrust principles seem useful in informing the preemption analysis, any attempt to apply antitrust concepts to what is essentially a copyright or contract case runs the risk of making litigation much more complex and expensive than it otherwise would be. The copyright misuse doctrine may prove helpful to avoid transforming breach of contract litigation based on a licensee's breach of a decompilation provision into full blown antitrust litigation. Historically, copyright misuse has been used as a defense to infringement actions "where there is an attempt to extend the exclusionary power granted by copyright beyond the protected work itself."[303] It is somewhat unclear whether a copyright misuse defense grounded in an antitrust violation requires the usual detailed antitrust proof of that violation, but it seems that the quantum of proof is somewhat less.[304] In a case in which a particular licensor has obtained a dominant position in the operating system market and is seeking through exclusionary practices to extend that power into the application market, copyright misuse should be found based on an essential facilities, leveraging, or modified tying analysis.[305] In such a case, any decompilation pro-

[303] 3 NIMMER & NIMMER, *supra* note 63, § 13.09[A], at 13-300 (citing Bellsouth Adv. & Pub. Corp. v. Donnelley Info. Pub., Inc., 719 F. Supp. 1551, 1562 (S.D. Fla. 1988), *aff'd*, 933 F.2d 952 (11th Cir. 1991), *rev'd en banc*, 999 F.2d 1436 (11th Cir.), *cert. denied*, 114 S. Ct. 943 (1993)); *see also* Lasercomb Am., Inc. v. Reynolds, 911 F.2d 970 (4th Cir. 1990) (holding that licensor misused copyright by including non-compete agreement with 99-year duration in license). *See generally* Leaffer, *supra* note 56, at 1098-99 (explaining copyright misuse doctrine). Some commentators have argued that inclusion of a reverse engineering prohibition in a license agreement constitutes copyright misuse:

> If the [*Lasercomb v. Reynolds*] precedent holds and it is further concluded that the Copyright Act preempts state law enforcing contractual prohibition of reverse engineering, Lasercomb powerfully implies that the copyright in a program distributed subject to the contract restriction is wholly and universally unenforceable until the misuse is purged.

Rice, *supra* note 17, at 550-51.

[304] *See* Leaffer, *supra* note 56, at 1099-1102 (discussing differences between copyright misuse doctrine and autitrust laws).

> Most courts have declared that the misuse defense does not require proof of an antitrust violation. . . . [N]either proof of market power, nor competitive injury is necessary to prove misuse. . . . [T]he defendant in a misuse claim must prove only that the plaintiff has extended his property right beyond the patent or copyright.

*Id.* (footnotes omitted).

[305] Historically, the copyright misuse defense has been most concerned with leveraging the statutory monopoly into another market. When the operating system provider seeks to prevent cloning, it is attempting to obtain monopoly power within the same

hibition should be struck because it is anticompetitive, barring application developers from competing. The term would then be unenforceable and software licensees would be free to decompile the operating system under the limited *Atari/Sega* rule.

In sum, a court faced with a breach of contract action grounded in a licensee's breach of a decompilation prohibition should analyze relevant market evidence, whether the contract is negotiated or not. This analysis should be informed by reference to antitrust principles because the grant of a copyright monopoly is still tempered by general antitrust doctrine and the goals of copyright law are best effectuated when market forces are considered. Decompilation provisions should be preempted[306] if the operating system provider has sufficient power effectively either (1) to engage in exclusionary practices to prevent cloning; or (2) to leverage that power into the application market.[307] This approach seems to strike a reasonable balance between the legitimate needs of operating system providers and application developers as well as the copyright concerns of encouraging creativity while maintaining competition by allowing fair use rights to be alienable except in those cases in which inalienability is appropriate to protect the public's interest in maintaining competition.

market as that of the copyrighted product. However, copyright misuse still should be available in this situation despite the lack of leveraging into another market. *See, e.g.*, *Lasercomb*, 911 F.2d at 970 (involving copyright misuse where licensor sought to prevent competition in market for copyrighted product by including 99-year non-compete agreement in license).

[306] This preemption could be either constitutionally or statutorily based. Enforcement of decompilation provisions under the circumstances described in the text would stand as "an obstacle to the accomplishment of the full purposes and objectives of Congress" as embodied in the Copyright Act. *See supra* note 232 and accompanying text. Alternatively, a finding of copyright misnse could support a preemption finding under § 301, although that section may have to be amended to provide specifically for the analysis suggested here.

[307] Of course, one might assert that this entire inquiry could be avoided if decompilation provisions were per se preempted up front. Such provisions only assist a developer in obtaining marketing power. However, in and of themselves, they are not insidious and in fact encourage investment and creative effort by helping to assure a return on that investment. It is only when they are used in conjunction with exclusionary practices by one with sufficient market power to exclude competition that their effects are truly harmful. This is fully consistent with both the policies underlying the Copyright Act and the antitrust principle that conduct that is generally acceptable may later become unacceptable when undertaken by a monopolist.

### B. *An Evaluation of the Proposal*

1. *The Pros.* The primary benefit of the foregoing proposal is that it integrates business, technical, and legal realities in a sensible manner, striking a balance between business needs and policy concerns. It allows licensors and licensees to contract around the fair use provision of the Act except in those cases in which such contracts unduly harm competition. Moreover, it offers an alternative to the per se preemption or nonpreemption of such provisions in license agreements. While a per se rule may provide *ex ante* certainty, it does not provide the correct result in most cases. The integrated and nuanced analysis is more likely to encourage both creativity and competition, rather than one at the expense of the other.

In addition, the proposal places the Copyright Act within both the federal and state contexts within which it operates. Parties remain free under state contract law to contract around the Act unless the consent that characterizes that contract is flawed. An example of flawed consent would be that obtained by a licensor using the monopoly granted under the Copyright Act coupled with state contract law to obtain a result that contravenes the broader antitrust principles within which the Act operates.

Finally, it recognizes that software providers may have legitimate business reasons for seeking to prevent decompilation. However, these justifications must be balanced against the federal policy of maintaining free access to ideas. Therefore, the proposal seeks to formulate a legal regime in which decompilation provisions are, as a general rule, enforced, except in those circumstances in which enforcement will hinder the competition enabled by the free flow of ideas. This scheme does not penalize a particular provider for its success, but rather recognizes the market reality that a dominant position in the software operating system market can be parlayed into a dominant position in the application market. The operating system interface then becomes an "essential facility" without which others are not able to compete in the application market—a market to which the original copyright monopoly did not extend.

2. *The Cons.* However, there are some drawbacks to the proposal. Chief among these may be that it would put the United States at odds with the European Union (EU). Software, as a series of electronic impulses, flows relatively freely over borders,

which suggests the need for global uniformity of its governing law. The European Economic Community recently adopted a Software Directive that sanctions decompilation under circumstances similar to those enunciated in *Atari* and *Sega*.[308] Moreover, the Directive explicitly provides that parties may not contract around the limited decompilation right.[309] It may be costly and impractical for software providers to offer and enforce one set of contractual terms in the United States and a different set in the EU. However, just because Europe has adopted this approach does not mean that it is the correct one. The Directive also may, in fact, indirectly offer support for the approach suggested in this Article. Europe may have adopted the Directive in response to U.S. software companies' dominance of the European market.[310] The Directive's limited decompilation right may be one tool for EU countries to use in growing a domestic software industry in a market dominated by foreign firms that refuse to license information that would allow European firms to compete. The European response thus supports the argument that the right to decompile is necessary in markets distorted by monopolistic behavior. Under the above analysis, however, the EU should be willing to forego the decompilation right once the software market becomes more competitive.

[308] Council Directive No. 91/250, art. 6, 1991 O.J.(L 122) 42, 45 (allowing for limited decompilation right when "indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs," provided that the information is not otherwise available and the decompilation is limited to those parts of the program necessary for interoperability).

[309] *Id.* at 43 ("Whereas, however, any contractual provisions contrary to Article 6 . . . should be null and void.").

[310] See OTA REPORT, *supra* note 16, at 94–97, which describes the dominant role of the United States in the global market:

> U.S. producers are increasingly challenged by competition from developing software industries abroad, particularly in Europe. Europe has been a very important market for U.S. firms, which dominated their European rivals. In the late 1980s, U.S. software producers held almost half of the European software market, with IBM being the largest single software vendor in the European market. . . . In 1990, according to [the Software Publishers Association], U.S. companies had more than 70 percent of the European PC-software market. . . . U.S. software is so pervasive, in part, because of the head start the U.S. industry enjoyed and the large size of the domestic U.S. market. Our large domestic market has given the U.S. industry significant advantages: a nation's domestic software market is an important base for developing the expertise and experience that are necessary to compete successfully (through exports) in the international marketplace.

*Id.* at 96–97 (citations omitted).

Another drawback to the proposal is that it may increase the cost of litigation if it transforms copyright infringement/breach of contract cases into full blown antitrust litigation. Antitrust cases are notoriously expensive. Moreover, the software market has historically been prone to rapid changes in market share. Thus, by the time a case came to trial, the window of opportunity during which the decompilation right would have been meaningful may have closed. However, this may no longer be the case. Once an operating system vendor is firmly established, customers are reluctant to switch to another system—even if it is superior—given the costs of learning another system and the probable lack of applications for the new system. Additionally, the costs of antitrust litigation may be avoided by use of the copyright misuse doctrine.[311]

Finally, it may, at first glance, seem difficult to reconcile the proposal with the policy concern of maintaining the free flow of ideas. The aim of federal intellectual property laws has been not just to increase creativity and the number of products available but also the general store of information available to the public at large. Preventing a consumer from decompiling through the "private legislation" of the standard form contract seems to contradict that policy. No one would seriously contend that the purchaser of a bottle of Coca-Cola is not free to attempt to analyze it chemically; why then should consumers be prevented from analyzing the programs that they purchase? An answer to that question would require a recapitulation of all that has been set forth above—from the origins of the reverse engineering prohibition to its market effects. Essentially, the argument is that given the software industry's structure, enforcing these provisions except in certain circumstances may, in fact, encourage the dissemination of ideas such that no conflict with overriding policy concerns is presented.

The argument suggests either that software is in some way fundamentally different from other works that have been protected by copyright or that, in the case of software, courts should inquire into business justifications for particular contract terms—an inquiry they do not normally make in copyright cases. The extensive literature on the difficulties of applying traditional intellectual property principles to software and setting forth proposals for new approaches[312] suggests that software is, in fact, markedly different

[311] *See supra* note 304 (noting that copyright misuse assertion traditionally does not require proof of antitrust violation).

[312] This literature culminated in the seminal work of Samuelson et al., *supra* note

from works traditionally covered by the federal intellectual property scheme. Thus, rather than clinging to intellectual property principles and procedures that worked well in the hard copy world, society should be willing to consider alternatives and experiment to discover the one that will best encourage continued innovation. Depending on one's perspective, and perhaps also on one's political orientation, market-based contracting usually implements efficient innovation. Rather than simply adopting a knee-jerk approach that sets aside market-based contracts when they seem to conflict with traditional intellectual property principles, society should be willing to make a deeper inquiry, despite its nontraditional nature, and consider changes to the system to deal with these new technologies.

## CONCLUSION

As new technologies emerge, the collisions between copyright and contract are likely to increase. The proposal set forth herein should provide a useful framework for addressing preemption issues in other contexts as well. For example, database vendors routinely condition access to data on the licensee's agreement to treat the data as if it were copyrighted when, in fact, it may not be.[313] The considerations in this case are somewhat different than

[15] For a survey of the earlier literature, see *id.* at 2310 n.1.

[313] For example, a subscriber to the WESTLAW database may use the data available to it solely in the regular course of legal and other research and related work. The subscriber has the right to download and temporarily store insubstantial portions of data to a storage device under the subscriber's exclusive control, solely to display the data internally and to quote from it—appropriately cited and credited—in memoranda, briefs, and similar work product. The subscriber also may create printouts of data for internal use and distribution to third parties, provided that such third parties do not further distribute them. The subscriber may not sell or license data to third parties, or use data as a component of or basis for any material offered for sale or license. WEST PUBLISHING CORPORATION, WESTLAW® SUBSCRIBER AGREEMENT, *supra* note 12, §§ 1-2. WESTLAW's database combines both copyrighted and public domain information, but the same terms ostensibly apply to each and charges for downloading apply regardless of what type of information is downloaded. It seems then that WESTLAW has attempted to restrict the use of even the factual, public domain information contained in its database through the license grant, while also limiting the "fair use" of copyrighted material. *See supra* note 12; *see also* MEAD DATA CENTRAL, SUBSCRIPTION AGREEMENT—LEXIS-NEXIS § 2 (on file with author) (granting research license to subscriber, but also providing that subscriber may make copies of printouts and distribute printouts and copies to extent permitted by copyright law; however, subscriber has no right to forward information electronically, although it may display such information to not more than one user at a time).

those set forth above, but illustrate how the preceding proposal can be applied to other new technologies that present novel copyright questions. The essential question in the database context is whether or not parties may, by contract, confer copyright status on material that Congress has determined is not protected by copyright.

Again, the analysis includes an examination of market conditions and a comparison of negotiated agreements with standard form ones. Such an analysis would likely reveal that many contracts allowing access to database content do so under terms requiring the licensee to treat the data as if copyrighted whether or not it, in fact, is.[314] The use restriction in this context is different from that discussed above. Here the same concerns about the licensee's inability to uncover ideas are not implicated because the data itself—and its ideas—are not hidden from the licensee's view. However, the contractual use restrictions that set up "private" copyright law seem fundamentally at odds with the policy of promoting the free flow of information.

Market conditions and the unique characteristics of electronic databases give rise to these restrictions. If a purely factual database receiving little or no copyright protection[315] is distributed

[314] *See supra* note 313.

[315] This "thinner" scope of protection is a direct result of Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 347–48 (1991) (addressing copyrightability of alphabetical white pages telephone directory and holding that selection, coordination, and arrangement of data failed to meet originality standard described by Court as constitutionally required).

> "No one may claim originality as to facts." . . . This is because facts do not owe their origin to an act of authorship. . . . Factual compilations, on the other hand, may possess the requisite originality. The compilation author typically chooses which facts to include, in what order to place them, and how to arrange the collected data so that they may be used effectively by readers. These choices as to selection and arrangement, so long as they are made independently by the compiler and entail a minimal degree of creativity, are sufficiently original that Congress may protect such compilations through the copyright laws.

*Id.* (citations omitted). Some commentators have criticized the decision as discouraging information providers from making electronic databases of factual information available. *See* Priscilla A. Walter & Eric H. Sussman, *Protecting Commercially Developed Information on the NREN*, COMPUTER LAW., April 1993, at 1, 4 (*Feist's* requirements of creative expression for copyright protection in information "may ultimately lead to making the creation and access to databases more expensive and less efficient;" thus, "commercial information providers may not make their rich offerings of information available [on a database] over the new [National Research and Education Network (NREN)]."); *see also* Jane C. Ginsburg, *No "Sweat"? Copyright and Other Protection of Works of Information After* Feist v. Rural Telephone, 92 COLUM. L. REV. 338, 387–88 (1992) (arguing that the *Feist* creative originality standard "may require information providers to restrict access to without a license agreement, any licensee could compete with the database provider, and undercut that provider's price because the second-comer would not incur the large start-up costs that the first-comer must recoup. Moreover, the effort to make a copy would be minimal. Most licensees probably would not engage in such conduct but, in a mass market, database providers cannot distinguish between those who would copy and resell from those who would not.[316] Thus, they enter into license agreements to ensure a return on their investment because the copyright law does not provide a means through which they may recoup it.

Applying the reasoning set forth above in this different context, these licenses should not be preempted unless, by creating and enforcing "private copyright," the licensor gains near monopoly power in the market for the particular information. If this were to occur, a party would have achieved, by private contract, that which the copyright law denied it—the benefits of a copyright monopoly in the absence of copyrightable data.

The Copyright Act has stood the test of time relatively well, but new technologies threaten to render it a nullity as providers of software and other electronic information denominate their transactions as licenses and contract around the Act's background provisions. While a *sui generis* regime might be a better way with which to deal with the novel questions electronic technology poses, even if Congress were inclined to consider such legislation, it still would take years to implement. Thus, principles must be established today to deal with the inevitable conflicts between these license agreements and the Act. This Article has identified one such conflict and suggested a detailed model to use in its resolution. While more such clashes are likely to arise, the analytical framework put forth herein should provide a useful starting point for addressing them. Although it does depart from traditional intellectual property paradigms, this Article proposes that such a departure may be warranted by the novel questions presented by emerging technologies. Nor is such a departure inappropriate, so

compilations in order to maintain a contractual or technological hold on the material . . . [possibly making] access to information both more burdensome and more expensive than had copyright protection been available"). For a listing of the extensive literature that the *Feist* case has spawned, see Paolo Cerina, *The Originality Requirement in the Protection of Databases in Europe and the United States*, 24 IIC: INT'L REV. OF INDUS. PROPERTY & COPYRIGHT L. 579, 589 n.71 (1993).

[316] *Cf. supra* text accompanying note 72.

long as it is conducted with full awareness of the policy concerns inherent in the intellectual property system as well as those implicated in the enforcement of private contracts.