

December 2008

Game Over for Reverse Engineering?: How the DMCA and Contracts Have Affected Innovation

Craig Zieminski

Follow this and additional works at: <https://scholarship.law.ufl.edu/jtlp>

Recommended Citation

Zieminski, Craig (2008) "Game Over for Reverse Engineering?: How the DMCA and Contracts Have Affected Innovation," *Journal of Technology Law & Policy*. Vol. 13: Iss. 2, Article 6.
Available at: <https://scholarship.law.ufl.edu/jtlp/vol13/iss2/6>

This Article is brought to you for free and open access by UF Law Scholarship Repository. It has been accepted for inclusion in *Journal of Technology Law & Policy* by an authorized editor of UF Law Scholarship Repository. For more information, please contact kaleita@law.ufl.edu.

GAME OVER FOR REVERSE ENGINEERING?: HOW THE DMCA AND CONTRACTS HAVE AFFECTED INNOVATION

Craig Zieminski*

I.	INTRODUCTION	290
II.	THE LEGAL LANDSCAPE OF REVERSE ENGINEERING DURING THE 1990S	292
	A. <i>Reverse Engineering Is Accepted and Perceived as Beneficial to Society</i>	292
	B. <i>Copying for Purposes of Reverse Engineering Is a Fair Use Under Copyright Law</i>	294
	C. <i>Contracts Limiting the Right to Reverse Engineer Are Either Preempted by Copyright Law or Constitute Misuse of Copyright</i>	301
III.	THE MODERN LEGAL LANDSCAPE OF REVERSE ENGINEERING	305
	A. <i>The DMCA</i>	305
	B. <i>How the DMCA Affects Reverse Engineering</i>	307
	C. <i>Recent Decisions in Contract Law</i>	309
	D. <i>Reverse Engineering Takes a Step Back</i>	312
III.	ANALYSIS OF <i>BOWERS</i> AND <i>BNETD</i>	316
	A. <i>Contractual Burdens</i>	316
	B. <i>DMCA Burdens</i>	317
IV.	SHOULD SOCIETY PREFER THE <i>SEGA</i> ERA OR THE CURRENT STATE OF THE LAW?	319
	A. <i>An Empirical Impossibility</i>	319

* Craig Zieminski, Stanford J.D. '08. Craig is an associate at Vinson & Elkins LLP in Dallas, TX. He may be reached at craig.zieminski@stanfordalumni.org. This Article is intended for educational and informational purposes only and does not constitute legal advice or services. If legal advice is required, the services of a competent professional should be sought. These materials represent the views of and summaries by the author. They do not necessarily reflect the opinions or views of Vinson & Elkins LLP or of any of its other attorneys or clients. They are not guaranteed to be correct, complete, or current, and they are not intended to imply or establish standards of care applicable to any attorney in any particular circumstance.

B.	<i>The Costs and Benefits of Reverse Engineering, in General</i>	320
C.	<i>An Attempt to Balance the Scales</i>	322
V.	WARNING SIGNS	324
A.	<i>Copyright Law Is Being Used as a Substitute for Patent Protection</i>	324
B.	<i>Copyright Law Is Being Used to Capture Secondary Markets</i>	326
C.	<i>The Copyright Act Has Become a "Default Rule," Valid Only in the Absence of a Contract</i>	330
VI.	CONCLUSION	339

I. INTRODUCTION

When Hide Nakajima and Dennis Wood, President and Vice-President of Atari Games, approached Nintendo in 1987 with the desire of licensing Atari's video game titles for the phenomenally popular Nintendo Entertainment System, they expected to receive star treatment.¹ Atari, having founded the industry in the late 1970s, had substantial name recognition, and only Nakajima and Wood could offer Nintendo access to Atari's impressive library of popular arcade games.² As such, Nakajima and Wood asked Nintendo to waive terms in its standard licensing agreement that limited licensees to publishing five new titles per year and restricted licensees from releasing these titles on other platforms for two years.³ Nintendo's top executives, Minoru Arakawa and Howard Lincoln, saw no need to negotiate: the Nintendo Entertainment System had a market share between eighty-six and ninety-three percent during 1987.⁴ Arakawa "shrugged as if to indicate it was not personal"—the licensing terms were a quality control mechanism that Nintendo had to apply evenly to all licensees.⁵ Atari relented and signed a licensing agreement that included both of the unwanted conditions.⁶

1. STEVEN KENT, *THE ULTIMATE HISTORY OF VIDEO GAMES* 372 (2001); DAVID SHEFF, *GAME OVER: PRESS START TO CONTINUE* 243 (2d ed., CyberActive Publ'g 1999).

2. KENT, *supra* note 1, at 372; SHEFF, *supra* note 1, at 246.

3. KENT, *supra* note 1, at 372.

4. *Id.* at 360.

5. SHEFF, *supra* note 1, at 243; KENT, *supra* note 1, at 351.

6. SHEFF, *supra* note 1, at 243.

The license came amidst a computer chip shortage that prevented Nintendo from releasing adequate quantities of its video games.⁷ Licensees received only twenty-five percent of the blank cartridges—the storage medium which contained the video game software—they had ordered, and despite the popularity of Atari's titles, it was treated the same as other licensees.⁸ Atari, estimating that it was supplying only ten percent of the demand for its top titles, found alternate sources for the scarce chips, but Nintendo rejected the idea, claiming that the alternates were of unacceptable quality.⁹ Dennis Wood was incensed: "We're talking about chips for games, not for a Cray computer."¹⁰

Sensing the frustration of his new licensee, Nintendo's president invited Atari's executives to his decadent home in Seattle for dinner in August 1988.¹¹ After an exquisite feast, Arakawa invited his guests outside for drinks.¹² Then, as he has a strange tendency to do at awkward moments, he fell asleep—mid-conversation with Atari's Nakajima.¹³

The faux pas had a symbolic element. As Arakawa snoozed on his deck that evening, Atari was engaged in a full-scale effort to reverse engineer Nintendo's technology in order to publish unlicensed games for the popular video game hardware.¹⁴ A year before the chip shortage and the licensing negotiations, Atari had commissioned engineer Donald Paauw to reverse engineer the security system that prevented unauthorized cartridges from functioning on the Nintendo Entertainment System. A year after the fateful dinner party, a similar engineer working for a similar software company, Accolade, would complete a similar reverse engineering feat on a similar piece of hardware, the Sega Genesis.¹⁵

The results of these efforts and the ensuing litigation would determine the ability of innovators to engage in the reverse engineering of computer programs during the decade that saw the proliferation of the Internet and countless other technological marvels. Engineers today face a diminished ability to reverse engineer, due to the Digital Millennium Copyright Act

7. *Id.* at 244.

8. *Id.*

9. *Id.* at 244-45.

10. *Id.* at 245.

11. SHEFF, *supra* note 1, at 237.

12. *Id.*

13. *Id.*

14. *Id.* at 246.

15. *Sega Enters. v. Accolade, Inc.*, 785 F. Supp. 1392, 1395 (N.D. Cal. 1992), *aff'd in part, rev'd in part*, 977 F.2d 1510 (9th Cir. 1992).

and developments in contract law. This Article will illustrate and explore the shift by comparing software lawsuits from the 1990s and 2000s.¹⁶

In an effort to facilitate efficient research by practitioners, I have endeavored to separate my normative discussion from an objective overview of the legal landscape. Part II lays out this legal landscape, while Part IV provides commentary.

II. THE LEGAL LANDSCAPE OF REVERSE ENGINEERING DURING THE 1990S

A. Reverse Engineering Is Accepted and Perceived as Beneficial to Society

Reverse engineering is the practice of “starting with the known product and working backward to divine the process which aided in its development or manufacture.”¹⁷ More broadly, it is “the process of extracting know-how or knowledge from a human-made artifact.”¹⁸

The Supreme Court gave a resounding endorsement for reverse engineering in *Bonito Boats v. Thunder Craft Boats*, which was argued around the same time that *Accolade* was reverse engineering the *Sega Genesis*.¹⁹ The *Bonito* Court struck down a Florida law prohibiting the application of the “direct molding process” to boat hulls.²⁰ The process, which the *Bonito* Court considered a form of reverse engineering, allowed boat manufacturers to quickly and easily replicate the designs of their competitors.²¹ The statute was enacted to protect innovations in boat hull design, as it requires massive effort and expense to design an innovative boat hull, but only hours to replicate it using the direct molding process.²² The Court struck the law as preempted by federal patent law, but in doing so, it extolled the virtues of reverse engineering. First, federal policy dictates “that all ideas in general circulation be dedicated to the common good unless they are protected by a valid patent.”²³ Second, “[r]everse engineering of chemical and mechanical articles in the public domain often

16. The choice of video game-related reverse engineering is not arbitrary. Many of the leading cases on the topic of reverse engineering fall within this industry.

17. *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 476 (1974).

18. Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 *YALE L.J.* 1575, 1577 (2002).

19. *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141 (1989).

20. *Id.* at 144.

21. *Id.* at 160.

22. *Id.* at 144, 158.

23. *Id.* at 159-60 (quoting *Lear, Inc. v. Adkins*, 395 U.S. 653, 668 (1969)).

leads to significant advances in technology.”²⁴ Third, “the competitive reality of reverse engineering may act as a spur to the inventor, creating an incentive to develop inventions that meet the rigorous requirements of patentability.”²⁵

Given the facts of the case, these commendations are particularly indicative of the Court’s extremely high esteem for reverse engineering. Market-based, non-legal protections against reverse engineering (e.g., incentives to innovate) are strongest when (a) an initial innovator has a long ‘lead time,’ or duration of time before competing products enter the marketplace, and (b) reverse engineering is costly.²⁶ Thus, if society benefits from “promot[ing] the Progress of Science and useful Arts,” it might prefer protections against reverse engineering where innovative advancements can be quickly and cheaply reverse engineered.²⁷ Where reverse engineering is costly and time-consuming, however, protection is less necessary in order for innovators to make an adequate return. Further, those parties that *do* reverse engineer in these markets are more likely to be innovators themselves (second generation innovation), as the gain from mere replication is lower where lead time is high and reverse engineering is costly.

In *Bonito*, defendant Thunder Craft Boats was scarcely engaging in any sort of reverse engineering at all. The direct molding process is closer to photocopying than reverse engineering.²⁸ Few would consider a programmer photocopying a computer program to be reverse engineering. In fact, the *Atari* court exempted any use of photocopied Nintendo program code from Atari’s reverse engineering defense.²⁹ Boat hull design is the quintessential case for restrictions on reverse engineering: a costly innovation could be cheaply and immediately reverse engineered, thus providing almost no incentive to innovate.³⁰ From a purely legal standpoint, the Florida law was almost certainly preempted by patent law, but in a normative debate about the prudence of the statute, the defendant would have struggled to prevail. Still, the Rehnquist court held reverse engineering in such esteem at the dawn of the 1990s that O’Connor’s unanimous opinion waxed poetic about its virtues for two pages.³¹

24. *Bonito*, 489 U.S. at 160.

25. *Id.*

26. Samuelson & Scotchmer, *supra* note 18, at 1586.

27. U.S. CONST. art. 1, § 8, cl. 8.

28. *See, e.g.,* *Interpart Corp. v. Italia*, 777 F.2d 678, 684-85 (Fed. Cir. 1985) (describing the direct molding process).

29. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 842-43 (Fed. Cir. 1992).

30. *Contra Bonito*, 489 U.S. at 160.

31. *Id.* at 159-60.

B. Copying for Purposes of Reverse Engineering Is a Fair Use Under Copyright Law

Until computer programmers sought copyright protection for their programs, copyright law had little need for a reverse engineering doctrine.³² After all, paintings and novels are not often the target of reverse engineering in the narrow sense of the term.³³

A trio of cases establishes that reverse engineering of software components in video game hardware is permissible, even if intermediate copying occurs. The most often cited is *Sega Enterprises v. Accolade, Inc.*³⁴ Sega developed and marketed the Sega Genesis video game console.³⁵ Accolade produced software titles for IBM-compatible personal computers and wanted to make its products available on the Genesis console.³⁶ However, Sega, like Nintendo, only licensed the initialization code and interface protocols necessary to produce games for its console to software developers that agreed to release titles exclusively for the Genesis.³⁷ This initialization code, called the trademark security system (TMSS), contained four bytes of data that were included in the video game software.³⁸ When a user inserted a game into the console and turned on the power, the Genesis would search the software and initialize the game only upon finding the TMSS.³⁹

Unwilling to agree to the exclusivity contract, Accolade “‘reverse-engineered’ Sega’s video game programs in order to discover the requirements for compatibility with the Genesis console.”⁴⁰ Its engineers decompiled (or “disassembled”) the object code of a Genesis console and software titles, printed copies of the resulting source code, and analyzed the printouts.⁴¹ They also experimented with the console and programs in

32. Samuelson & Scotchmer, *supra* note 18, at 1585.

33. *Id.*

34. *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

35. *Id.* at 1514.

36. *Id.*

37. *Id.*

38. *Id.* at 1515. The four bytes of data spelled “S-E-G-A.”

39. *Sega*, 977 F.2d at 1515.

40. *Id.* at 1514.

41. *Id.* at 1515. Object code is the machine-readable signal code of a program, represented by a long string of zeros and ones. *Id.* Source code is the textual code of a program, written in a programming language, such as C++. Source Code Definition, The Linux Information Project, http://www.linfo.org/source_code.html (last visited Nov. 18, 2008). In the typical order of computer programming, a programmer writes source code in a programming language and uses a program, called a compiler, to translate the source code into a string of zeros and ones that the computer can “understand.” *Id.* Compilation and decompilation create copies of a work. *See id.*

order to discern the Genesis' interface specifications.⁴² Accolade memorialized this knowledge in an internal development manual and used it to release several unlicensed games for the Genesis.⁴³ Sega sued for copyright and trademark infringement, and the district court entered a preliminary injunction against Accolade.⁴⁴

The Ninth Circuit reversed, holding that "disassembly of copyrighted object code is, as a matter of law, a fair use of the copyrighted work if such disassembly provides the only means of *access* to those elements of the code that are not protected by copyright and the copier has a legitimate reason for seeking such *access*."⁴⁵

The first fair use factor, "purpose and character of the use," favored Accolade.⁴⁶ Although Accolade made reproductions of Sega's works for commercial purposes, it merely sought to discover the unprotected functional elements of the Genesis system in order to release its own creative works on a new platform.⁴⁷ The fourth factor, effect on the market, also favored Accolade.⁴⁸ Though acknowledging that Accolade's copying facilitated the release of software that competed with Sega's software, the court concluded, "it [does not] 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."⁴⁹ The third factor, amount copied, favored Sega, as Accolade copied entire works.⁵⁰

42. *Sega*, 977 F.2d at 1515.

43. *Id.*

44. *Id.* at 1516-17. The trademark claims arose from a byproduct of the initialization code. After locating the TMSS initialization code in the cartridge, the Genesis would automatically display a startup screen that read, "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD." This function was added to the console to give Sega a cause of action against Taiwanese pirates, as Taiwan recognizes foreign trademarks but not copyrights. *See id.* at 1515.

45. *Id.* at 1518 (emphasis added). The court had first rejected Accolade's other arguments: (1) intermediate copying does not infringe the reproduction right unless the end product of the copying infringes the reproduction right; (2) decompilation of object code to understand functional elements is permissible under the idea and expression dichotomy of 17 U.S.C. § 102(b); (3) disassembly is authorized by 17 U.S.C. § 117, which exempts from copyright scrutiny those copies made "as an essential step in the utilization of the computer program . . ." *Id.* at 1517; 17 U.S.C. § 117(a)(1) (2008).

46. *Sega*, 977 F.2d at 1521-22 (citing 17 U.S.C. § 107(1)).

47. *Id.* at 1522. The court also noted the public benefit from increased availability of independently designed video games. *Id.* at 1523.

48. *Id.* at 1522.

49. *Id.* at 1523.

50. *Id.* at 1526-27.

The Ninth Circuit's discussion of the second fair use factor, nature of the copyrighted work, has the most substantial connection to the DMCA anti-circumvention provision and reverse engineering.⁵¹ Copyright analysis of computer programs is complicated because programs are intertwined with copyrightable expression and unprotected functional elements:

To the extent that there are many possible ways of accomplishing a given task or fulfilling a particular market demand, the programmer's choice of program structure and design may be highly creative and idiosyncratic. However, computer programs are, in essence, utilitarian articles—articles that accomplish tasks. As such, they contain many logical, structural, and visual display elements that are dictated by the function to be performed, by considerations of efficiency, or by external factors such as compatibility requirements and industry demands.⁵²

Sega's TMSS initialization code was deemed functional, as "[t]here is no showing that there is a multitude of different ways to *unlock* the Genesis III console."⁵³ As such, the only code that Accolade directly transcribed from Sega's software was not protected by copyright.⁵⁴ The nature of the work also substantiated Accolade's reverse engineering tactics. Because a computer program's protected expression is intertwined with unprotected elements, and because programs are distributed in object code, "humans often cannot gain *access* to the unprotected ideas and functional concepts contained in object code without disassembling that code—i.e., making copies [of the entire program, including any protected expression]."⁵⁵ If the court did not find Accolade's copying for reverse engineering purposes to be a fair use, Sega would have a "*de facto* monopoly over . . . ideas and functional concepts" without "satisfy[ing] the more stringent standards imposed by the patent laws."⁵⁶

Although Atari's console coup was completed a full year before Accolade began its endeavor to unlock the Genesis, the Federal Circuit decided *Atari Games Corp. v. Nintendo of America, Inc.* about one month after the Ninth Circuit decided *Sega*.⁵⁷ The situation was nearly identical. To recap: Nintendo produced the Nintendo Entertainment System (NES),

51. *Sega*, 977 F.2d at 1522.

52. *Id.* at 1524.

53. *Id.* at 1524 n.7 (emphasis added).

54. *Id.* at 1525.

55. *Id.* (emphasis added).

56. *Sega*, 977 F.2d at 1527, 1526.

57. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992).

a predecessor to Sega's Genesis, and Atari produced software for the NES.⁵⁸ Nintendo had a licensing agreement similar to Sega's, and Atari sought a method of producing unlicensed games for the NES.⁵⁹ Nintendo had developed a program, called 10NES, that "controls access to the NES" by requiring a computerized "lock" in the console to receive a coded message from a "key" within the software cartridge before initializing the game.⁶⁰

Atari had less success in reverse engineering the 10NES program than Accolade did with the simpler Genesis lockout mechanisms. Donald Paauw could not fully analyze the object code by microscopically examining the chips themselves.⁶¹ Atari's engineer, Pat McCarthy, could not adequately "listen" to the communication between the lock and key chips to produce an alternative key, and he told Atari's executives to abandon the project.⁶² Unfazed, Atari hired a Virginia law firm to defend it in a copyright infringement suit against Nintendo.⁶³ The only catch: no such lawsuit had been filed (this suit was not filed for another two years).⁶⁴ Without bothering to verify that a lawsuit had been filed, the law firm filed an affidavit with the U.S. Copyright Office, requesting a copy of the 10NES object code for the purpose of defending the lawsuit.⁶⁵ Ten days after Atari and Nintendo had signed their licensing agreement, "[a]n employee of the law firm headed up to room 402 [of the Copyright Office] and waltzed out with a copy of the 10NES copyright in his briefcase," which he promptly delivered to Atari.⁶⁶

Perhaps in anticipation of future (non-imagined) litigation, Atari resumed its efforts to read the source code from the 10NES chips, using the ill-gotten copy of the code only to correct errors in this transcription.⁶⁷ Having quasi-reverse-engineered 10NES, Atari developed its own program, called Rabbit, which produced the same coded message as the

58. *Id.* at 835-36.

59. *Id.* at 836. Atari, unlike Accolade, had signed a licensing agreement when it could not successfully reverse engineer the 10NES program. *Id.*

60. *Id.*

61. *Id.*; see also SHEFF, *supra* note 1, at 247. A computer chip embodies the zeros and ones of object code through a series of microscopic switches. Basically, the chip has the object code's string of zeros and ones inscribed onto it. Chip, PC Magazine Encyclopedia, http://www.pcmag.com/encyclopedia_term/0,2542,t=chip&i=39636.asp (last visited Nov. 18, 2008).

62. SHEFF, *supra* note 1, at 247; *Atari*, 975 F.2d at 836.

63. SHEFF, *supra* note 1, at 248.

64. *Id.* at 248-49.

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

66. SHEFF, *supra* note 1, at 248.

67. *Id.* (The Copyright Office copy facilitated Atari's replication of the 10NES object code.).

10NES key using a different programming language.⁶⁸ Nintendo sued, alleging (among other claims) that Atari infringed the 10NES copyright by making intermediate copies during the reverse engineering process.⁶⁹ The Federal Circuit first determined that the 10NES did contain protected expression: "Nintendo incorporated within the 10NES program creative organization and sequencing unnecessary to the lock and key function."⁷⁰

The *Atari* court then determined that intermediate copying for reverse engineering was a fair use.⁷¹ Rather than mechanically considering the four factors, like the *Sega* court did, the *Atari* court took a broader, rights-based approach:

Under the [Copyright] Act, society is free to exploit facts, ideas, processes, or methods of operation in a copyrighted work. To protect processes or methods of operation, a creator must look to patent laws. An 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.⁷²

Two caveats tempered this fair use finding. First, "[t]he fair use reproductions of a computer program must not exceed what is necessary to understand the unprotected elements of the work."⁷³ The court did not elaborate on this point, ostensibly content that Atari had engaged in no more intermediate copying than was necessary. Second, "[t]o invoke the fair use exception, an individual must possess an authorized copy of a literary work," and as such, intermediate copying of the ill-gotten Copyright Office reproduction was not a fair use.⁷⁴

Atari's victory was not complete, however. Even if reproductions made during the *process* of reverse-engineering were a fair use, reproductions in the *end product* of the reverse engineering were not necessarily

68. *Atari*, 975 F.2d at 836-37 ("The Rabbit gave Atari access to NES owners without Nintendo's strict license conditions.").

69. *Id.* at 840, 842.

70. *Id.* at 840.

71. *Id.* at 843.

72. *Id.* at 842 (citing *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 348-50 (1991); *Bonito Boats v. Thunder Craft Boats*, 489 U.S. 141 (1989) (additional citations omitted)).

73. *Atari*, 975 F.2d at 843-44 ("Atari could not use reverse engineering as an excuse to exploit commercially or otherwise misappropriate protected expression.").

74. *Id.* at 843. Atari's misrepresentation to the Copyright Office also prevented the court from considering whether Nintendo was abusing its copyright in the lockout program to monopolize the unprotected elements of the NES. Because Atari came to the court with unclean hands, it could not seek equity. *Id.* at 845-46.

permissible. Nintendo alleged that Atari's Rabbit program contained expression that was substantially similar to 10NES, thereby violating the reproduction right.⁷⁵ The trial court "detected similarities between the programs beyond [those] . . . necessary to embody the unprotectable idea, process, or method of the 10NES program," and the Federal Circuit agreed, citing instructions present in both programs that were unnecessary for "Atari's stated purpose—unlocking the NES console."⁷⁶ The superfluous instructions related to tangential functions, such as shutting down the console if the "key" receives an error message.⁷⁷ Atari was still on the hook for Rabbit, even if it was somewhat exonerated for its reverse engineering.⁷⁸

Eight years later, the Ninth Circuit revisited the issue of reverse engineering gaming consoles in *Sony Computer Entertainment, Inc. v. Connectix Corp.*⁷⁹ Sony, like the two plaintiffs before it, produced a video game console—the Playstation.⁸⁰ Unlike Accolade and Atari, however, Connectix was not a would-be game software producer. Connectix produced The Virtual Game Station (VGS), a software program for Apple operating systems that allowed users to play Playstation games on their Apple computers.⁸¹

To produce the VGS, Connectix engineers developed hardware emulation software that allowed the components of a Macintosh computer, such as the graphics card, to mimic the components of the Playstation.⁸² Connectix then extracted the Playstation's BIOS ("basic input-output system"), a program that initializes and coordinates the various functions

75. *Id.* at 840.

76. *Id.* at 844-45 ("The unnecessary instructions in the Rabbit program suggest copying, not independent creation."). One wonders whether the outcome would have differed if Atari had stated its purpose as "imitating the complete functionality of the 10NES program."

77. *Id.* at 845.

78. *Atari*, 975 F.2d at 845. Atari had slightly more success in its countersuit against Nintendo, which alleged anticompetitive practices. Fueled by American distaste for Japanese businesses, the suit flourished into an FTC investigation and a lawsuit by the attorneys general of every state. To chill the accusations of anticompetitive practices, Nintendo abandoned the exclusivity clause of its licensing agreement and began to loosen its monopoly on the manufacturing process. However, only top licensees were allowed to manufacture games, they could only do so with chips provided by Nintendo, and a high royalty was required. The FTC investigation was settled when Nintendo agreed to send five dollar coupons to five million NES purchasers. Critics scoffed at the settlement, as the coupons served to boost Nintendo sales and covered only a fraction of the price of any Nintendo merchandise. See SHEFF, *supra* note 1, at 256-279.

79. *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000).

80. *Id.* at 599.

81. *Id.*

82. *Id.* at 601.

of the hardware, from a legally purchased Sony console.⁸³ The engineers then loaded the BIOS onto a computer running the hardware emulation software and observed the signals sent between the two programs.⁸⁴ The process required countless reproductions of the copyrighted BIOS.⁸⁵ Connectix announced its VGS at the 1999 MacWorld Expo, and three weeks later, Sony sued for, among other causes, copyright infringement.⁸⁶

Relying heavily on *Sega*, the Ninth Circuit held that “Connectix’s intermediate copying and use of Sony’s copyrighted BIOS was a fair use for the purpose of *gaining access* to the unprotected elements of Sony’s software.”⁸⁷ Discussion of the ‘nature of the copyrighted work’ fair use factor again focused on the inherently intertwined nature of software and the resulting need to copy protected expression to access unprotected elements. Reiterating that “*Sega* expressly sanctioned disassembly,” the court saw “no reason to distinguish observation of copyrighted software in an emulated computer environment.”⁸⁸ Although the context was slightly different, Sony was attempting the same trick as *Sega*: “If Sony wishes to obtain a lawful monopoly on the functional concepts in its software, it must satisfy the more stringent standards of the patent laws.”⁸⁹

Analysis of the ‘purpose and character of use’ factor was markedly different than in *Sega*, as the Supreme Court had, in the interim, endorsed the transformative use doctrine and rejected the principle that a commercial use creates a presumption of unfairness.⁹⁰ By developing its own original object code and creating a new platform on which to enjoy Playstation games, the court found the VGS “modestly transformative.”⁹¹

83. *Id.*; see also BIOS, PC Magazine Encyclopedia, http://www.pcmag.com/encyclopedia_term/0,2542,t=BIOS&i=38653,00.asp (last visited Nov. 18, 2008).

84. *Connectix*, 203 F.3d at 601.

85. *Id.*

86. *Id.*

87. *Id.* at 602 (emphasis added).

88. *Id.* at 603-04.

89. *Connectix*, 203 F.3d at 605 (citing *Bonito*, 489 U.S. at 160-61; *Sega*, 977 F.2d at 1526).

90. See *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 579, 583-84 (1994).

The central purpose of this [factor] is . . . whether the new work merely “supersede[s] the objects” of the original creation . . . or instead adds something new, with a further purpose or different character, altering the first with new expression, meaning, or message; it asks, in other words, whether and to what extent the new work is “transformative.”

Id. at 579 (internal citations omitted).

91. *Connectix*, 203 F.3d at 606.

Further, Connectix's use was commercial, but only indirectly so, as no copying was alleged in the final product.⁹²

The 'effect on the market' factor was handled more directly than in *Sega*, and without supposition as to gamers' purchasing habits: "Sony understandably seeks control over the market for devices that play games Sony produces or licenses. The copyright law, however, does not confer such a monopoly."⁹³ Thus, Connectix was the overwhelming victor.⁹⁴

C. Contracts Limiting the Right to Reverse Engineer Are Either Preempted by Copyright Law or Constitute Misuse of Copyright

Nor could software companies contractually limit the ability to reverse engineer during the *Sega* era, though many companies attempted such a limitation by including a "shrinkwrap" license with the software.⁹⁵ Professor Mark Lemley concluded in 1995 that "virtually no reported decisions have actually enforced shrinkwrap license provisions as written, especially where those provisions modify federal law."⁹⁶

The most frequently cited of these decisions is *Vault Corp. v. Quaid Software Ltd.*, in which the Fifth Circuit held that federal contract law preempted state contract law explicitly permitting software vendors to contractually forbid reverse engineering.⁹⁷ Vault developed "PROLOK," a software program that prevented unauthorized copying of the contents of floppy disks.⁹⁸ The company included "a license agreement with every PROLOK package that specifically prohibits the copying, modification, translation, decompilation or disassembly of Vault's program."⁹⁹ Quaid

92. *Id.* at 607

93. *Id.*

94. Sony soon turned to a more direct strategy. Shortly after losing its legal battle, the company purchased the VGS from Connectix and immediately discontinued the product. A few years later, Connectix sold its virtual PC application, a Windows emulator for Mac computers, to Microsoft. Connectix is no longer in business, its employees having either taken positions at Microsoft or (perhaps) retired to a tropical destination with large checks from Sony and Microsoft. See Jim Dalrymple, *Microsoft Acquires Virtual PC from Connectix*, MACWORLD, Feb. 19, 2003, <http://www.macworld.com/news/2003/02/19/macbu/>.

95. See Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 S. CAL. L. REV. 1239, 1241 (1995). The term "shrinkwrap license" comes from the clear plastic wrapping on the outside of the software package. Some very early shrinkwrap licenses were affixed to the outside of the package, such that they could be viewed before opening the package. However, companies soon began including the license inside the box, so that the license could only be read after removing the shrinkwrap and opening the package. The ramifications of this modification will be discussed in Part IV. *Id.* at 1241-42.

96. *Id.* at 1259.

97. *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 270 (5th Cir. 1988).

98. *Id.* at 256.

99. *Id.* at 257.

reverse engineered PROLOK to develop a software program that would create backup copies of programs stored on PROLOK-protected disks.¹⁰⁰ Vault filed suit, “contending that Quaid breached its license agreement.”¹⁰¹

The contract, which the court considered “a contract of adhesion,” was governed by the Louisiana Software License Enforcement Act, which provides the conditions for creating a binding software license agreement.¹⁰² If these conditions are met, the Act then defines the limits of acceptable contract terms:

Terms of which shall be deemed to have been accepted under R.S. 51:1963, if included in an accompanying license agreement which conforms to the provisions of R.S. 51:1965, may include any or all of the following:

* * *

100. *Id.* (“Quaid copied Vault’s program into the memory of its computer and analyzed the manner in which the program operated.”).

101. *Id.* at 258.

102. *Vault*, 847 F.2d at 268-69 (quoting *Vault Corp. v. Quaid Software Ltd.*, 655 F. Supp. 750, 761 (E.D. La. 1987)). The Louisiana Act, still unchanged since its passage, requires of a license agreement:

Any person who acquires computer software or a copy thereof shall be conclusively deemed to have accepted and agreed to all the terms of the license agreement for such software or copy thereof, including any applicable provisions contained in [§ 51:1964], if:

- (1) A written legend or notice is affixed to or packaged with the software or copy thereof in such a manner that the legend or notice is clearly and conspicuously visible upon cursory examination of the software and related packaging; and
- (2) The legend or notice is prominently displayed in all capital letters and in language which is readily understandable to a person of average literacy; and
- (3) The legend or notice states clearly that:
 - (a) Any use of the software or copy thereof will constitute acceptance of the terms of the accompanying license agreements; or,
 - (b) Any opening of a sealed package, envelope, or container in which the software or copy thereof is contained will constitute acceptance of the terms of the accompanying license agreement; and
- (4) The legend or notice states clearly that anyone who receives the software or a copy thereof and does not accept and agree to the terms of the accompanying license agreement may, within a reasonable time, return the unused, unopened software or copy thereof to the party from whom it was acquired, or to some other identified party, for a full refund of any consideration paid; and
- (5) The person acquiring the software or copy thereof takes such action as is stated in the legend or notice to constitute acceptance of and agreement to the terms of the accompanying license agreement.

(3) If title to the copy of computer software has been retained by the licensor, provisions for the prohibition or limitation of rights to modify and/or adapt the copy of the computer software in any way, including without limitation prohibitions on translating, reverse engineering, decompiling, disassembling, and/or creating derivative works based on the computer software.¹⁰³

The Fifth Circuit quoted the Supreme Court's guiding principle for the application of the preemption doctrine: "[w]hen state law touches upon the area of [patent or copyright statutes], it is 'familiar doctrine' that the federal policy 'may not be set at naught, or its benefits denied' by the state law."¹⁰⁴ It then held, "[t]he provision in Louisiana's License Act, which *permits* a software producer to prohibit the adaptation of its licensed computer program by decompilation or disassembly . . . is preempted by federal law, and thus that the restriction in Vault's license agreement against decompilation or disassembly is unenforceable."¹⁰⁵

The Louisiana statute at issue in *Vault* is often misquoted. It is not, as the Federal Circuit has claimed, "a state law prohibiting all copying [or reverse engineering] of a computer program."¹⁰⁶ Louisiana's license act is a regulation of contracts, setting the guidelines and boundaries of contracts between two parties. The Louisiana law does not create any society-wide prohibitions; it merely clarifies the extent to which parties may bind each other in software licenses. The Fifth Circuit's holding, then, states that no contract may incorporate the provision deemed acceptable-but-optional by the Louisiana License Act.

Two years later, the Fourth Circuit reached a similar conclusion with a different doctrine in *Lasercomb America, Inc. v. Reynolds*.¹⁰⁷ Both sides of this dispute manufactured devices used to cut cardboard so that it could be folded into boxes.¹⁰⁸ Lasercomb developed a software program, entitled *Interact*, to aid in this manufacturing endeavor and licensed four prerelease copies to defendant's company.¹⁰⁹ The license agreement forbade licensees

103. *Vault*, 847 F.2d at 269 n.28 (quoting LA. REV. STAT. ANN. § 51:1964 (2008)) (emphasis added). Section 51.1964, which is still on the books, also contains a provision allowing the licensor to prohibit any and all copying. See LA. REV. STAT. ANN. § 51:1964(2).

104. *Vault*, 847 F.2d at 269 (quoting *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225, 229 (1964)) (bracketed text in original).

105. *Id.* at 270 (emphasis added).

106. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1325 (Fed. Cir. 2003) (citing *Vault*, 847 F.2d at 268-69).

107. *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990).

108. *Id.* at 971.

109. *Id.*

from “writ[ing], develop[ing], produc[ing], or sell[ing]” competing software.¹¹⁰ Defendants made unauthorized copies of Lasercomb’s software, and they developed and marketed “a software program . . . which was almost entirely a direct copy of Interact.”¹¹¹ Lasercomb sued, and defendants made the affirmative defense of copyright misuse.¹¹²

The Fourth Circuit began its analysis by noting the uncertain status of the copyright misuse defense: “Although the patent misuse defense has been generally recognized since *Morton Salt*, it has been much less certain whether an analogous copyright misuse defense exists. . . . The few courts considering the issue have split on whether the defense should be recognized.”¹¹³ After establishing that the public policy interests behind patent and copyright laws are analogous, the court determined that the reasoning behind the adoption of the patent misuse doctrine applied equally well in the copyright context.¹¹⁴ The court held that such a defense should apply to copyrights, interjecting copyright terminology into a famous passage from *Morton Salt*:

The grant to the [author] of the special privilege of a [copyright] carries out a public policy adopted by the Constitution and laws of the United States, “to promote the Progress of Science and useful Arts, by securing for limited Times to [Authors] . . . the exclusive Right . . .” to their [“original” works]. But the public policy which includes [original works] within the granted monopoly . . . forbids the use of the [copyright] to secure an exclusive right or limited monopoly not granted by the [Copyright] Office and which it is contrary to public policy to grant.¹¹⁵

Having determined that such a defense is available to the defendants, the Fourth Circuit found that Lasercomb had misused its copyright: “Its standard licensing agreement . . . attempts to suppress any attempt by the licensee to independently implement the idea which Interact expresses.”¹¹⁶ As the Fourth Circuit explained further, “[t]he misuse arises from

110. *Id.* at 973 (“Licensee agrees during the term of this Agreement and for one (1) year after the termination of this Agreement, that it will not write, develop, produce or sell or assist others in the writing, developing, producing, or selling computer assisted die making software. . . .”).

111. *Id.* at 971.

112. *Lasercomb*, 911 F.2d at 973.

113. *Id.* at 976 (citing *Morton Salt Co. v. G.S. Suppiger Co.*, 314 U.S. 488 (1942) (upholding the patent misuse doctrine); W. HOLMES, INTELLECTUAL PROPERTY AND ANTITRUST LAW § 4.09 (1989)).

114. *Id.*

115. *Id.* at 977 (quoting *Morton Salt*, 314 U.S. at 492 (quoting U.S. CONST. art. I, § 8, cl. 8)).

116. *Id.* at 978.

Lasercomb's attempt to use its copyright in a particular expression, the Interact software, to control competition in an area outside the copyright, *i.e.*, the idea of computer-assisted die manufacture."¹¹⁷

III. THE MODERN LEGAL LANDSCAPE OF REVERSE ENGINEERING

A. *The DMCA*

Ten months after the *Connectix* decision was published, the Digital Millennium Copyright Act (DMCA) went into effect.¹¹⁸ The sections of the DMCA relevant to this analysis were enacted to provide copyright holders with additional protections against piracy seen as necessary in the digital age, where perfect copies of copyrighted works could be effortlessly and infinitely disseminated.¹¹⁹ Legal support for electronic copyright protections was a prerequisite for copyright holders to offer their works in digital form and in Internet marketplaces.¹²⁰

To this end, the DMCA states that “[n]o person shall circumvent a technological measure that effectively controls access to a work protected under [the Copyright Act].”¹²¹ Trafficking in devices that facilitate circumvention is also prohibited:

No person shall manufacture, import, offer to the public, provide, or otherwise traffic in any technology, product, service, device, component, or part thereof, that—

(A) is primarily designed or produced for the purpose of circumventing a technological measure that effectively controls access to a work protected under [the Copyright Act];

(B) has only limited commercially significant purpose or use other than to circumvent a technological measure that effectively controls access to a work protected under [the Copyright Act]; or

(C) is marketed by that person or another acting in concert with that person with that person's knowledge for use in circumventing a

117. *Lasercomb*, 911 F.2d at 979.

118. 17 U.S.C.A. § 1201 (West 2007).

119. S. REP. NO. 105-190, at 8 (1998) (“Due to the ease with which digital works can be copied and distributed worldwide virtually instantaneously, copyright owners will hesitate to make their works readily available on the Internet without reasonable assurance that they will be protected against massive piracy.”).

120. *Id.* (predicting that the DMCA would “launch[] the global digital on-line marketplace for copyrighted works . . . [and] facilitate making available quickly and conveniently via the Internet the movies, music, software, and literary works that are the fruit of American creative genius.”).

121. 17 U.S.C. § 1201(a)(1)(A) (2007).

technological measure that effectively controls access to a work protected under [the Copyright Act].¹²²

“Circumventing a technological measure” is defined as “to descramble a scrambled work, to decrypt an encrypted work, or otherwise to avoid, bypass, remove, deactivate, or impair a technological measure, without the authority of the copyright owner,” and “a technological measure ‘effectively controls access to a work’ if the measure, in the ordinary course of its operation, requires the application of information, or a process or a treatment, with the authority of the copyright owner, to gain access to the work.”¹²³

The DMCA includes several exceptions, one of which directly pertains to this discussion of reverse engineering:

(1) Notwithstanding the provisions of subsection (a)(1)(A), a person who has lawfully obtained the right to use a copy of a computer program may circumvent a technological measure that effectively controls access to a particular portion of that program for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs, and that have not previously been readily available to the person engaging in the circumvention, to the extent any such acts of identification and analysis do not constitute infringement under this title.

* * *

(4) For purposes of this subsection, the term “interoperability” means the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged.¹²⁴

The Senate Report corresponding to this subsection explicitly states that the reverse engineering sanctioned by the *Sega* line of cases was to remain lawful:

The objective the DMCA’s exception provisions is to ensure that the effect of current case law interpreting the Copyright Act is not changed by enactment of this legislation for certain acts of

122. *Id.* § 1201(a)(2).

123. *Id.* § 1201(a)(3).

124. *Id.* at § 1201(f); *see also* S. REP. NO. 105-190, at 33 (1998) (“tools that programmers use in developing computer programs, such as compilers, trace analyzers and disassemblers . . . are not prohibited by this section.”).

identification and analysis done in respect of computer programs. [sic] *See, Sega Enterprises Ltd. v Accolade, Inc.*, 977 F.2d 1510, 24 U.S.P.Q.2d 1561 (9th Cir. 1992.). The purpose of this section is to foster competition and innovation in the computer and software industry.¹²⁵

B. *How the DMCA Affects Reverse Engineering*

Despite the assurances of the Senate Report, the post-DMCA ability to reverse engineer is weaker than in the *Sega* era. First, the DMCA is much more restrictive in defining the circumstances under which software reverse engineering is acceptable. Under *Sega*, reverse engineering is permissible when (1) it is necessary to access unprotected information within a computer program and (2) the reverse engineers have a legitimate purpose.¹²⁶ Under the DMCA, which applies whenever efforts to reverse engineer require the circumvention of an access control mechanism, reverse engineers may only (1) identify and analyze (2) elements related to program-to-program interoperability (3) for the purpose of achieving interoperability (4) without committing copyright infringement while identifying and analyzing.¹²⁷ Instead of having the ability to reverse engineer for any legitimate purpose, programmers in the DMCA world may reverse engineer only in order to make their programs work with other programs (“interoperability”).

The narrowed window of acceptable purposes is burdensome for innovation industries (except, of course, those parties that have *already* innovated). *Accolade*, *Atari*, and *Connectix* would arguably be safe under § 1201(f). Their reverse engineering efforts were dedicated solely to making their software interoperate with the software of *Sega*, *Nintendo*, and *Sony*.¹²⁸ Of course, *Sega* and *Nintendo* might argue that plaintiffs were attempting software-to-hardware interoperability, which may or may not be included in the safe harbor for program-to-program interoperability. Further, imagine that *Google* had reverse engineered *AltaVista*’s web crawler in order to develop its world-beating search engine. If *AltaVista*’s code were protected by technological access control measures, then *Google* would have violated the DMCA, as the purpose of its reverse engineering was innovation, as opposed to interoperability. Under *Sega*, on the other hand, *Google* would be free to learn as much as possible from

125. S. REP. NO. 105-190, at 13 (1998).

126. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1518 (9th Cir. 1992).

127. 17 U.S.C. § 1201(f) (2007). These numbers are for parsing purposes only and do not reflect any enumeration in the actual statutory text.

128. *See supra* text accompanying notes 36, 59, 81.

AltaVista's programs, as long as Google's ultimate program did not merely copy code from AltaVista.¹²⁹

Second, reverse engineering tools are more likely to be outlawed by the DMCA by requiring courts to scrutinize an individual component or part of a device.¹³⁰ Under contributory infringement, the doctrine that controls the legality of such devices in the non-DMCA world, it is an open question whether a court should analyze the product-as-a-whole or individual components of that device. The seminal case on contributory infringement, *Sony v. Universal City Studios*, declares that a device does not violate the copyright of others "if the *product* is widely used for legitimate, unobjectionable purposes[,] or, in other words, if it is "capable of substantial noninfringing uses."¹³¹

This Supreme Court instruction refers to the product-as-a-whole and makes no mention of analyzing individual components. On the other hand, the *Sony* Court analyzed only the 'record' function of the Betamax in determining whether the Betamax machine contributorily infringed the reproduction right of television content owners by allowing Betamax users to record television programs.¹³² One could argue that the Supreme Court was implicitly approving an individual function analysis, because the Betamax had a much less complicated non-infringing use: watching purchased, rented, or camcorder-recorded movies.¹³³ This flexibility allows courts to apply a "sniff test," condemning a blatantly infringing device duct taped to a non-infringing device but exculpating a device primarily designed for non-infringing purposes that has, by necessity, an infringing component.

Consider, for example, a computer program that allows users to analyze and manipulate other pieces of software that are access-protected by technological measures. By necessity, this program has a function to circumnavigate technological measures, such as decrypting encrypted software. The program, assume further, consists primarily of functions that assist engineers in performing the type of activities undertaken by the *Sega* plaintiffs—perhaps the ability to log messages sent between programs or hardware components, a decompiler, or a code debugger.

129. See *supra* text accompanying note 47.

130. 17 U.S.C. § 1201(a)(2) (2007).

131. *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 442 (1984) (emphasis added).

132. *Id.* at 422-23, 442.

133. Twentieth-Century Fox began releasing home videos in 1975. See The Museum of Broadcast Communications, *Home Video*, available at <http://www.museum.tv/archives/etv/H/htmlH/homevideo/homevideo.htm> (last visited Sept. 15, 2008); Total Rewind, *Format War*, http://www.totalrewind.org/fmt_war.htm.

Without the DMCA, a software company could sue the designer of this program for contributory infringement, but a court would apply the more flexible *Sony* rule and potentially view the program-as-a-whole for substantial non-infringing uses.¹³⁴ This program has non-infringing uses that are arguably “substantial,” as a programmer might use it on *her own* independently-written software during the development and debugging stages of production. Under *Sega*, the programmer could also use the program on the software of *another* party for legitimate, fair-use reverse-engineering.¹³⁵

Under the DMCA, however, the reverse-engineering program is almost certainly doomed. Despite its many non-infringing uses, the statute obliges a court to ask whether the program has a “component[] or part” that “is primarily designed . . . for the purpose of circumventing a technological measure that effectively controls access to a work.”¹³⁶ The component of this program that unlocks access-protected software is undoubtedly designed to circumvent technological measures. At the very least, a court would require the program’s designers to remove the circumvention function, an injunction that would weaken the ability to reverse-engineer in an increasingly access-controlled world.

C. Recent Decisions in Contract Law

A software license with an anti-reverse-engineering provision, similar to those in *Vault* and *Lasercomb*, was upheld in *Bowers v. Baystate Technologies, Inc.*, highlighting a potential shift in courts’ views on the interaction between copyright and contract laws.¹³⁷ Bowers produced an add-on template for CAD software.¹³⁸ His template program included a shrink-wrap license that “prohibited, *inter alia*, all reverse engineering of [the] software, protection encompassing but more extensive than copyright protection, which prohibits only certain copying.”¹³⁹ Baystate, having purchased a copy of Bowers’ program, designed a program that incorporated many of its features.¹⁴⁰ Bowers sued for, among other claims, breach of contract, and Baystate defended on the grounds that “the Copyright Act preempts the prohibition of reverse engineering embodied in Mr. Bowers’ shrink-wrap license agreements.”¹⁴¹

134. *Sony*, 464 U.S. at 442.

135. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 110, 1527-28 (9th Cir. 1992).

136. 17 U.S.C. § 1201(a)(2) (2000).

137. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317 (Fed. Cir. 2003).

138. *Id.* at 1320-21.

139. *Id.* at 1327.

140. *Id.* at 1322.

141. *Id.* at 1323.

The Federal Circuit analyzed the claim based on the principle that no preemption has occurred under the Copyright Act if “a state cause of action requires an extra element, beyond mere copying, preparation of derivative works, performance, distribution or display.”¹⁴² The Federal Circuit acknowledged that “not every ‘extra element’ of a state law claim will establish a qualitative variance between the rights protected by federal copyright law and those protected by state law.”¹⁴³ Noting that “most courts to examine this issue have found that the Copyright Act does not preempt contractual constraints on copyrighted articles,” the Federal Circuit held that the contract elements of assent and consideration were extra elements that foreclosed preemption.¹⁴⁴

The Federal Circuit also noted that its decision did not affect its holding in *Atari* or the reverse engineering exception of the DMCA, § 1201(f).¹⁴⁵ It then distinguished its holding from *Vault*: “Moreover, while the Fifth Circuit has held a state law prohibiting all copying of a computer program is preempted by the Federal Copyright Act, no evidence suggests the First Circuit would extend this concept to include private contractual agreements supported by mutual assent and consideration.”¹⁴⁶

Judge Dyk dissented. He argued that the test for copyright preemption should match the test for patent preemption: “whether the state law ‘substantially impedes the public use of the otherwise unprotected’ material.”¹⁴⁷ Using this test, Judge Dyk found that copyright holders were using state contract law to eliminate a fair use defense that is necessary to prevent authors from gaining protection over unprotectable ideas.¹⁴⁸

Next, Dyk argued that Bowers’ contract failed the “extra element” test employed by the majority.¹⁴⁹ While a fully-negotiated contract might limit fair use without being preempted by Copyright law, a shrink-wrap license is an adhesion contract, giving “the copyright holder the ability to

142. *Bowers*, 320 F.3d at 1324 (quoting *Data Gen. Corp. v. Grumman Sys. Support Corp.*, 36 F.3d 1147, 1164 (1st Cir. 1994)) (citation omitted).

143. *Id.* (quoting *Data Gen.*, 36 F.3d at 1164) (“For example, the First Circuit observed that ‘a state law misappropriation claim will not escape preemption . . . simply because a plaintiff must prove that copying was not only unauthorized but also commercially immoral.’” (quoting *Data Gen.*, 36 F.3d at 1165)).

144. *Id.* at 1324-25.

145. *Id.* at 1325.

146. *Id.* (citing *Vault*, 847 F.2d at 255).

147. *Bowers*, 320 F.3d at 1335 (quoting *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 157 (1989); citing *Sears, Roebuck & Co. v. Stiffel Co.*, 376 U.S. 225, 231-32 (1964)) (Dyk, J., dissenting).

148. *Id.* at 1336 (Dyk, J., dissenting).

149. *Id.*

eliminate the fair use defense in each and every instance at its option.”¹⁵⁰ He likened the use of state contract law in this respect to a hypothetical “black dot” law: “If state law provided that a copyright holder could bar fair use of the copyrighted material by placing a black dot on each copy of the work offered for sale, there would be no question but that the state law would be preempted.”¹⁵¹

Dyk pointed out that the majority had substantially misread *Vault*:

the Fifth Circuit held that the specific provision of state law that authorized contracts prohibiting reverse engineering, decompilation, or disassembly of computer programs was preempted by federal law because it conflicted with a portion of the Copyright Act and because it “‘touched upon an area’ of federal copyright law.” From a preemption standpoint, there is no distinction between a state law that explicitly validates a contract that restricts reverse engineering (*Vault*) and general common law that permits such a restriction (as here).¹⁵²

Dyk distinguished the cases relied upon by the majority in that the cases upheld contract terms that did not restrict a right under the copyright law.¹⁵³ For instance, in *ProCD, Inc. v. Zeidenberg*, which, at the time, was the only other court of appeals decision regarding shrinkwrap licenses, the software developer sold non-commercial and commercial versions of its software.¹⁵⁴ The non-commercial version, which was offered at a reduced price, had a “no commercial use” restriction in its shrinkwrap license.¹⁵⁵ As opposed to the reverse-engineering prohibition in his case, Dyk found this contract term acceptable under preemption doctrine: “The Copyright Act does not confer a right to pay the same amount for commercial and personal use. It does, however, confer a right to fair use, which we have held encompasses reverse engineering.”¹⁵⁶

150. *Id.* at 1337 (Dyk, J., dissenting).

151. *Id.* at 1336-37 (Dyk, J., dissenting).

152. *Bowers*, 320 F.3d at 1337 (quoting *Vault*, 847 F.2d at 269-70 (quoting *Sears, Roebuck*, 376 U.S. at 229)) (Dyk, J., dissenting).

153. *Id.* (Dyk, J., dissenting).

154. *Id.* at 1337 (citing *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996)) (Dyk, J., dissenting).

155. *Id.*

156. *Id.* at 1338 (citing 17 U.S.C. § 107) (Dyk, J., dissenting).

D. Reverse Engineering Takes a Step Back

Curtailments of reverse-engineering abilities in both contract and copyright doctrine came to a head in *Davidson & Associates, Inc. v. Internet Gateway, Inc.* (also known as “*bnetd*”). *Bnetd* offers a perfect opportunity to view the shifting legal landscape because it pits a successful video game company (Blizzard Entertainment) against a producer of interoperating software, just as in *Sega, Nintendo*, and *Connectix*. Blizzard produces extremely popular computer games, including the WarCraft, StarCraft, and Diablo series.¹⁵⁷ In January 1997, Blizzard launched Battle.net, a free online service that allows gamers to play alongside or against each other in certain Blizzard titles.¹⁵⁸ Many users were disappointed by system bugs, unscrupulous players who cursed or cheated, and the apparently unconscionable quid-pro-quo of having to view small advertisements in order to use the free service.¹⁵⁹ Led by Ross Combs, Rob Crittenden, and Jim Jung, a loose alliance of programmers formed the “*bnetd* project” to develop an alternative method of playing multiplayer Blizzard games over the internet.¹⁶⁰

To create the *bnetd* service, the programmers needed to learn the “language” that Blizzard software spoke to the Battle.net service.¹⁶¹ Without this language, called “protocol” in programming parlance, the *bnetd* service could not communicate with the end-users’ software, and the multiplayer system could not function.¹⁶² The programmers, by necessity, performed reverse engineering operations to learn the protocol:

Combs used reverse engineering in the process of developing the *bnetd* server, including a program called “*tcpdump*” to log communications between Blizzard games and the Battle.net server. Crittenden used reverse engineering in the process of developing the *bnetd* server, including using a program called “*Nextray*.” Crittenden also used a program called “*ripper*” to take Blizzard client files which were compiled together in one file and break them into their component parts.¹⁶³

157. See Blizzard Entm’t Web Site, <http://www.blizzard.com/> (last visited Oct. 20, 2008).

158. *Davidson & Assocs. v. Internet Gateway, Inc.*, 334 F. Supp. 2d 1164, 1168 (E.D. Mo. 2004), *aff’d*, 422 F.3d 630 (8th Cir. 2005); see also Battle.net Introduction, <http://www.battle.net/intro.shtml> (last visited Oct. 20, 2008).

159. *Davidson*, 334 F. Supp. 2d at 1171-72.

160. *Id.* at 1172 (“The *bnetd* program was intended as a functional alternative to the Battle.net service.”).

161. *Id.*

162. *Id.*

163. *Id.* at 1172-73.

Even Blizzard, through its expert report, acknowledged that the protocol language required for a multiplayer server could not be learned another way.¹⁶⁴ The programmers reverse engineered legally acquired versions of Blizzard software, which, when installed, required users to accept the following End User License Agreement and Terms of Use:

BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE PROGRAM YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT.

* * *

[S]ubject to the grant of license hereinabove, you may not, in whole or in part, copy, photocopy, reproduce, translate, *reverse engineer*, *derive source code*, *modify*, *disassemble*, *decompile*, create derivative works based on the Program, or remove any proprietary notices or labels on the program without the prior consent, in writing, of Blizzard.

* * *

You are entitled to use Battle.net for your own personal use, but you shall not be entitled to. . .(ii) copy, photocopy, reproduce, translate, *reverse engineer*, *modify*, *disassemble*, or *de-compile* in whole or in part any Battle.net software; (iii) create derivative works based on Battle.net; (iv) host or provide matchmaking services for any Blizzard software programs or *emulate or redirect the communication protocols* used by Blizzard as part of Battle.net, through protocol emulation, tunneling, modifying, or adding components to the Program, use of a utility program, or any other technique now known or hereafter developed for any purpose. . . .¹⁶⁵

Such licensing agreements are a recent development in the realm of gaming software. Diablo, the oldest title at issue in the suit, lacked an agreement, and no mention of similar terms is found in the published opinions of *Sega*, *Nintendo*, or *Connectix*.¹⁶⁶

One important feature of Battle.net that bnetd lacked was CD key authentication.¹⁶⁷ All Blizzard software after Diablo includes a CD key, a unique string of alphanumeric characters (generally printed on the jewel case

164. Plaintiff's Motion for Summary Judgment at 6, *Davidson*, 334 F. Supp. 2d at 1168 (No. 4:02CV498 (AS)), 2003 WL 243 09750.

165. *Davidson*, 334 F. Supp. 2d at 1170-71 (emphasis added).

166. *Id.* at 1182 n.13. The author has also examined his personal collection of video games produced during the 1980s and 1990s and found no end user license agreements.

167. *Id.* at 1173.

holding the game media) that the user must enter to install the game.¹⁶⁸ In an attempt to curb piracy, Blizzard created an authentication process to restrict access to Battle.net.¹⁶⁹ Each time a user logs onto Battle.net, the user's CD key is encrypted and transmitted to Battle.net, where the service authenticates that the CD key is valid and not being used by another active Battle.net participant.¹⁷⁰ The bnetd server, on the other hand, received the encrypted key but performed no authentication process, simply sending back the "authentication successful" message necessary for the software to initialize online play.¹⁷¹ The bnetd developers offered to implement functions that would prevent users from playing on bnetd with pirated copies, but Blizzard refused, ostensibly wanting to maintain the confidentiality of its data pertaining to CD keys.¹⁷² Instead, Blizzard filed suit, alleging (among other claims) copyright infringement, violation of the DMCA, and breach of the End User License Agreements (EULA) and Battle.net Terms of Use (TOU).¹⁷³

Having settled the copyright infringement claim, Blizzard alleged that the bnetd programmers breached the EULA and TOU by reverse engineering Blizzard's software.¹⁷⁴ The defendants argued that these contracts were unenforceable because they prohibited a well-established fair use.¹⁷⁵ Thus, the issue of reverse engineering was entirely contractual, and the district court resolved the matter with only a passing reference to *Connectix* and no discussion of the *Sega* doctrine.¹⁷⁶ Despite acknowledging that "[r]everse engineering was necessary in order for the defendants to learn Blizzard's protocol language and to ensure that bnetd worked with Blizzard games" and that "[i]t would not have been possible to create a workable bnetd server without reverse engineering Blizzard's software and protocols[.]" the district court found the contractual agreement to be determinative.¹⁷⁷ The district court gave a brief overview of *Vault* and *Bowers* and, without explanation, concluded, "[t]he Court finds the reasoning in *Bowers* persuasive."¹⁷⁸ It then held, "[t]he

168. *Id.* at 1169.

169. *Id.*

170. *Davidson*, 334 F. Supp. at 1169.

171. *Id.* at 1173.

172. Cindy Cohn, EFF Letter to Blizzard/Vivendi, available at <http://www.eff.org/pages/eff-letter-blizzard-vivendi> (last visited Oct. 21, 2008).

173. *Davidson*, 334 F. Supp. 2d at 1167.

174. *Id.* at 1174.

175. *Id.* at 1180.

176. *Id.*

177. *Id.* at 1172.

178. *Davidson*, 334 F. Supp. at 1180-81.

defendants in this case waived their ‘fair use’ right to reverse engineer by agreeing to the licensing agreement.”¹⁷⁹

Blizzard next alleged that the bnetd programmers violated the DMCA by circumventing the CD key system, a technological measure that controls access to Battle.net mode.¹⁸⁰ The programmers argued that 17 U.S.C. § 1201(f)(1) protects their circumvention, as it was done “for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs. . . .”¹⁸¹ This exception is only available “to the extent any such acts of identification and analysis do not constitute infringement under this title.”¹⁸² The district court quickly established that the bnetd team had “extended into the realm of copyright infringement” based on the following: (1) bnetd lacked CD key authentication, (2) “[u]nauthorized copies of the Blizzard games were played on bnetd servers,” (3) bnetd was free, (4) bnetd was open source, (5) bnetd was distributed in binary form (as opposed to uncompiled source code), and (6) bnetd performs the same function as battle.net.¹⁸³ Based on these facts, the district court found the bnetd programmers liable for circumvention and trafficking a circumvention device.¹⁸⁴

The Eighth Circuit affirmed the contractual waiver of the bnetd programmers’ fair use defense and the inapplicability of the DMCA’s interoperability exception.¹⁸⁵ On the contract issue, the court first distinguished *Vault*: “the state law at issue here . . . [does not] restrict [] rights given under federal law.”¹⁸⁶ Quoting Judge Dyk’s dissent in *Bowers*, the appeals court determined that the bnetd programmers had “expressly relinquished their rights to reverse engineer” by entering into a “freely negotiated” contract.¹⁸⁷ Its reasoning for the inapplicability of the interoperability exception closely paralleled that of the district court: “Appellants’ circumvention in this case constitutes infringement. . . . [a]s a result [of the bnetd project], unauthorized copies of the Blizzard games were freely played on bnetd.org servers.”¹⁸⁸

179. *Id.* at 1181.

180. *Id.* at 1183.

181. *Id.* at 1184 (quoting 17 U.S.C. § 1201(f)(1)).

182. *Id.* (quoting 17 U.S.C. § 1201(f)(1)).

183. *Davidson*, 334 F. Supp. at 1185.

184. *Id.* at 1185-87.

185. *Id.* at 642.

186. *Id.* at 639.

187. *Id.* (quoting *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1337 (Fed. Cir. 2003) (Dyk, J., dissenting)).

188. *Davidson*, 334 F. Supp. at 642.

III. ANALYSIS OF *BOWERS* AND *BNETD*

From both doctrinal and policy perspectives, the recent curtailment of the *Sega* era freedom to reverse engineer is questionable. Doctrinally, the *Bowers* and *bneta* decisions misinterpret statutes and prior case law.

A. Contractual Burdens

Although the *bneta* courts do not misquote *Vault* as blatantly as the Federal Circuit in *Bowers*, all of these courts were heavily influenced by the fact that the Louisiana license act (in *Vault*) does explicitly what contractual common law (in *Bowers* and *bneta*) does implicitly. In *bneta*, the Eighth Circuit interpreted the state law at issue in *Vault* as restricting rights granted under federal law while determining that general contract law does not restrict such rights.¹⁸⁹ The Federal Circuit determined that the licensing regulation at issue in *Vault* was nothing less than "a state law prohibiting all copying of a computer program. . . ."¹⁹⁰ However, none of the state laws at issue in *Vault*, *Bowers*, or *bneta* directly restrict rights granted under federal law. Rather, they allow private parties to restrict other private parties' rights granted under federal law. The law at issue in *Vault* was no more a state-wide prohibition on reverse engineering than a common law contract doctrine stating that 'parties may bind each other to agreements.' Still, the *Vault* court invalidated the no-reverse-engineering clause because the law was being *used* to restrict rights granted under copyright law. Similarly, the common law doctrine that 'parties may bind each other to agreements' was *used* in *Bowers* and *bneta* to restrict rights granted under federal law.

If *Vault* had relied on Louisiana's general contract law instead of a specific software licensing statute, the *Bowers* and *bneta* courts probably would have deemed it persuasive authority. In this regard, they have created a distinction without a difference, as the *Vault* court correctly recognized that it was dealing with a contract provision and not a state-wide ban.¹⁹¹ Thus, it would almost assuredly have reached the same holding without the presence of the licensing statute. By distinguishing

189. *Id.* at 639 ("Unlike in *Vault*, the state law at issue here . . . [does not] restrict[] rights given under federal law.")

190. *Bowers*, 320 F.3d at 1325.

191. *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 268-69 (5th Cir. 1988) ("Louisiana's License Act *permits* a software producer to impose a number of *contractual terms* upon software purchasers provided that the terms are set forth in a license agreement which *comports* with La. Rev. Stat. Ann. §§ 51:1963 & 1965 . . . Enforceable *terms* include the prohibition of . . . reverse engineering, decompilation or disassembly.") (emphasis added).

Vault on the grounds that its law directly impedes copyright law by imposing a state-wide ban, the *Bowers* and *bned* courts failed to adequately analyze the defendants' arguments.

Judge Dyk would be horrified to learn that his dissent in *Bowers* was cited as support for the *bned* holding. Dyk proposed that courts permit parties to waive their fair use rights in contracts that are "freely negotiated."¹⁹² The contract in *bned*, like the contract in *Bowers*, was anything but freely negotiated. The *bned* programmers first learned the terms of the license agreement after purchasing the software and inserting the game disks into their computers.¹⁹³ The license agreement was not "negotiated" in any sense of the term, as the programmers had no input regarding the terms of the contract. Instead, the programmers had but two choices: accept all of Blizzard's terms or return the software for a refund.¹⁹⁴ Of course, almost no retail store allows customers to return opened software, and returning the software to Blizzard could take weeks.¹⁹⁵ Nevertheless, this level of bargaining strength was sufficient for the Eighth Circuit to determine that the *bned* programmers had "freely negotiated" their licenses with Blizzard.¹⁹⁶

B. DMCA Burdens

The *bned* district and appeals courts seriously mishandled their analysis of § 1201(f), the reverse engineering exception to the anti-circumvention provision. A court should only bar defendants from invoking § 1201(f) if the "acts of identification and analysis" infringe the copyright of another.¹⁹⁷ In *bned*, the courts focused on the *result* of the acts of identification and analysis by determining that the *bned* emulator itself infringed Blizzard's copyrights.¹⁹⁸ The district court opinion makes only a passing reference to the various tools and methods employed by the programmers, and it completely foregoes any analysis of whether the programmers' actions in learning Blizzard protocols would constitute

192. *Bowers*, 320 F.3d at 1336 (Dyk, J., dissenting).

193. *Davidson*, 422 F.3d at 633 n.2.

194. *Id.* at 634 n.4.

195. See Store Return Policy, bestbuy.com, available at <http://www.bestbuy.com/> (follow "Customer Service" link at top; then follow "Returns" link; then follow "Store Return Policy" link) (last visited Oct. 21, 2008) ("Opened computer software, movies, music and video games can be exchanged for the identical item but cannot be returned for a refund.").

196. *Davidson*, 422 F.3d at 639.

197. 17 U.S.C. § 1201(f)(1) (2007) (emphasis added).

198. *Davidson*, 422 F.3d at 640.

infringement. The Eighth Circuit followed suit by focusing on the bnetd emulator instead of the acts undertaken to develop the emulator.¹⁹⁹

The applicability of § 1201(f) should have turned on an analysis of the methods and tools used by the bnetd programmers. The bnetd team employed two of the most common methods for protocol analysis, neither of which is likely to result in copyright infringement. First, the bnetd programmers decompiled Blizzard's games to analyze these protocol messages sent by the games to the Battle.net servers.²⁰⁰ This method, which parallels the actions of Accolade and Atari, is expressly sanctioned by the *Sega* doctrine.²⁰¹ Second, the bnetd programmers used programs that "listen" to the information transmitted between Blizzard's games and its Battle.net system.²⁰² This method is analogous to learning a foreign language by listening to two individuals speaking the language.²⁰³

Even if the courts were supposed to determine whether the bnetd program infringed Blizzard's copyright for purposes of the § 1201(f) exception, their analysis was lackluster. The infringement portion of the district court opinion lacks a single reference to 17 U.S.C. §§ 101-107 or the substantial similarity doctrine, which are hallmarks of direct infringement analysis. Instead, the Eighth Circuit noted that, among other facts, bnetd was free and distributed in binary code.²⁰⁴ If these factors are critical to a determination of copyright infringement, then every piece of software in the world is likely to violate the Copyright Act, as the Eastern District of Missouri will be hard-pressed to locate an end-user program that is *not* distributed in binary code. The Eighth Circuit, perhaps attempting to establish some cursory form of secondary infringement, noted that pirated copies of Blizzard games were played using the bnetd emulator.²⁰⁵ While these facts may help the court establish that "[t]he defendants' purpose in developing the bnetd server was to avoid the anti-circumvention restrictions of the game and to avoid the restricted access to Battle.net," they are completely out of place in an analysis of whether the bnetd program infringed any copyrights.²⁰⁶ The courts may have been attempting to nullify the *Sega* doctrine by establishing an illegitimate purpose behind the reverse engineering, but neither opinion expresses this line of reasoning or cites to *Sega*.

199. *Id.*

200. *Id.* at 642.

201. *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 110, 1527-28 (9th Cir. 1992).

202. *Davidson*, 422 F.3d at 636.

203. Thanks to Fred von Lohmann for this analogy.

204. *Davidson & Assocs. v. Internet Gateway, Inc.*, 334 F. Supp. 2d 1164, 1185 (E.D. Mo. 2004).

205. *Davidson*, 422 F.3d at 642.

206. *Davidson*, 334 F. Supp. 2d at 1186.

The bnetd emulator may well have infringed Blizzard's copyright (or materially contributed to other infringements). However, this situation was delineated more properly by the Federal Circuit in *Atari*, which held the defendants liable for the Rabbit program but not for any permissible acts of reverse engineering they employed in developing the Rabbit.²⁰⁷

IV. SHOULD SOCIETY PREFER THE SEGA ERA OR THE CURRENT STATE OF THE LAW?

A. *An Empirical Impossibility*

Determining the "appropriate" level of freedom to reverse engineer is an impossible endeavor. The first step, selecting criteria on which to assess "appropriateness," is not difficult, as the U.S. Constitution provides helpful guidance. The second step, measuring the effects of a policy choice using the selected criteria, is infinitely harder. After detailing the most influential attempt to measure the effects of policy choices, this article will argue that we have shifted too far in the restrictive direction by discussing "warning signs" instead of engaging in any sort of measurement.

The Framers of the Constitution set forth the guiding principle for intellectual property law when they permitted Congress to pass copyright and patent laws "[t]o promote the Progress of Science and useful Arts[.]"²⁰⁸ Thus, an intellectual property doctrine's effect on the "Progress of Science and useful Arts," which this Article refers to as "innovation," measures the appropriateness of the doctrine. A law that promotes innovation is favored over one that stifles innovation.

However, "innovation" is worthless unless society benefits from it. Thus, the endeavor to select an appropriate level of intellectual property protection, including the freedom to reverse engineer, runs into an intractable dilemma: "without monopoly protection, not enough intellectual property will be developed, but, with a legal monopoly, too little will be used."²⁰⁹ Society is better off if *everyone* benefits from inventions and artistic works, but unless inventors and artists are able to reap adequate returns for their efforts, society will not receive these developments in the first place. Because the law cannot pull a "bait and

207. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 844-45 (Fed. Cir. 1992).

208. U.S. CONST., art. I, § 8, cl. 8.

209. Lawrence D. Graham & Richard O. Zerbe, *Economically Efficient Treatment of Computer Software: Reverse Engineering, Protection, and Disclosure*, 22 RUTGERS COMPUTER & TECH. L.J. 61, 71 (1996) (citing ROBERT COOTER & THOMAS ULEN, *LAW AND ECONOMICS* 37-41 (1988)).

switch” on inventors and artists (e.g., promising extensive protection before the moment of creation, and then, delivering no protection after creation), it must provide just enough protection to incentivize creation without preventing society from making adequate use of the creation.²¹⁰

This dilemma is particularly nefarious in the context of reverse engineering because innovation is on *both* sides of the equation. Society hopes to incentivize “first-level” innovators with intellectual property protections, but in doing so through reverse engineering prohibitions, it decreases “second-level” innovation.²¹¹ To take an earlier example, society wants AltaVista to create its search engine, but it also wants Google to improve upon AltaVista’s design. Keep this dilemma in mind while considering the costs and benefits of reverse engineering.

B. *The Costs and Benefits of Reverse Engineering, in General*

Reverse engineering has untold value to society, especially in the computer industry. First, reverse engineering increases the speed with which technology advances by allowing second-level innovators to build upon proprietary first-level innovations, a process known as sequential (or incremental) innovation. Second, reverse engineering strengthens the computer industry by increasing the number of interoperable programs. Each new program that interoperates with an existing program makes that existing program more valuable, a benefit known as “network effects.”

Sequential innovation and network efforts are particularly important in the software industry, where “innovation typically proceeds via a mix of new coding, modifications to some existing modules and subroutines, and either literal or functional reuse of others” and “patterns of improvements are constrained to a substantial degree by the need to preserve

210. Neil Winstock Netanel, *Copyright and a Democratic Civil Society*, 106 YALE L.J. 283, 285.

Copyright law strikes a precarious balance. To encourage authors to create and disseminate original expression, it accords them a bundle of proprietary rights in their works. But to promote public education and creative exchange, it invites audiences and subsequent authors to use existing works in every conceivable manner that falls outside the province of the copyright owner’s exclusive rights. Copyright law’s perennial dilemma is to determine where exclusive rights should end and unrestrained public access should begin.

Id.

211. And, of course, “third-level” innovation, “fourth-level” innovation, and beyond. In this Article, I will refer only to two iterations of innovation.

interoperability between program, system, and network components.”²¹² In the patent context, Professors Cohen and Lemley cite these benefits in arguing for a limited right to reverse engineer patented software and a more restrictive application of the doctrine of equivalents in patent infringement suits.²¹³ These benefits are equally applicable in a determination of the proper level of reverse engineering copyrighted software.

The dawn of the personal computer revolution highlights the value of both sequential innovation and network effects. First-level innovator IBM released the first personal computer on August 12, 1981.²¹⁴ IBM protected its proprietary BIOS code as a copyrighted work, and as a result, the PC market was fragmented by “clones,” which were not completely compatible with software written for MS-DOS/IBM systems:

A good PC knockoff running DOS might be able to mimic the IBM 99 percent of the time, but that other 1 percent would be left as a seed of doubt in the minds of corporate managers, who could be counted on to remember the old saying, “Nobody was ever fired for buying IBM.”²¹⁵

Along came a group of reverse-engineers from Texas working for a then-unknown startup named Compaq.²¹⁶ Over the course of fifteen months (and at a cost of one million dollars), one group of engineers developed an incredibly detailed catalog of the various functions performed by the IBM BIOS.²¹⁷ A second team, which had never seen the code for the IBM BIOS, then wrote its own independent BIOS based on the first team’s specifications.²¹⁸ The result of these efforts was the first 100% IBM-compatible personal computer that did not infringe the copyright of IBM’s BIOS.²¹⁹ Compaq’s computer was itself innovative, as it was the first

212. Julie Cohen & Mark Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CAL. L. REV. 1, 41 (2001) (“Software innovation is by nature largely incremental. It is rare for programs to be rewritten entirely from scratch . . .”).

213. *Id.*

214. Jim Battley, *Big Blue Birthday: IBM Turns 20*, CNN.com, <http://archives.cnn.com/2001/TECH/ptech/08/10/IBM.open.arch.idg/> (last visited Nov. 18, 2008).

215. Revenge of the BIOS, Startup Gallery, <http://www.startupgallery.org/gallery/story.php?ii=57> (last visited Oct. 21, 2008).

216. *Id.*

217. *Id.*

218. *Id.*

219. See *supra* text accompanying notes 215-18.

“portable” PC—a twenty-eight pound precursor to the modern laptop.²²⁰ Also, it shifted the personal computer industry from one of fragmentation to one of phenomenal “network effects.” The “100% IBM-Compatible” market (as opposed to the “clone” market) benefited from software and hardware that could work with any system, a reduced need for innovations to be repeated in distinct markets, and a more fertile environment for sequential innovation.

On the other hand, first-level innovation is negatively affected by reverse engineering. The incentive to develop innovative new products is lowered when reverse engineers can quickly produce competing products. The boat hull designs discussed earlier are a perfect example: few would invest large sums of money and time crafting an innovative boat hull without reasonable expectations of an adequate return, and this return is weakened when an innovation can be immediately and effortlessly copied.²²¹

Also, reverse engineering is often a costly and complicated process that drains resources from other potentially innovative research. When a product is reverse engineered for educational purposes, this cost is minimized, as the process itself is valuable to the engineer. However, when the engineer seeks only to make a competing product, she is spending resources to recreate an innovation that already exists. If this competing product is not independently innovative, then no “progress in the useful arts” has occurred, though society may benefit from increased access to the innovation.

C. *An Attempt to Balance the Scales*

With these costs and benefits to innovation in mind, anybody may attempt to “read the scales,” that is, to conclude whether reverse engineering ought to be encouraged more or less strongly. Two Berkeley professors, Pamela Samuelson and Suzanne Scotchmer, recently performed such a feat in “The Law and Economics of Reverse Engineering,” and their results were given the imprimatur of the *Yale Law Journal*.²²² They cautiously laid the groundwork for their analysis:

[t]he economic effects of reverse engineering depend on a number of factors, including the purpose for which it is undertaken, the industrial context within which it occurs, how much it costs, how

220. Compaq I Portable Computer, <http://oldcomputers.net/compaqi.html> (last visited Nov. 18, 2008).

221. See *supra* Part II.A.

222. Samuelson & Scotchmer, *supra* note 18.

long it takes, whether licensing is a viable alternative, and how the reverse engineer uses information learned in the reverse engineering process.²²³

Then, the professors evaluated the effects of reverse engineering for purposes of enabling software interoperability.²²⁴ The professors employed the social welfare criterion discussed above: “incentives to develop platforms,” (a.k.a. first-level innovation), “incentives to develop applications” (a.k.a. second-level innovation), “system prices” (a.k.a. the public’s access to the invention), and “wasted costs.”²²⁵ However, even with the cognitive firepower of two Berkeley professors, and even in an article worthy of the *Yale Law Journal*, neither empirical nor precise conclusions were possible. Instead, the professors painted well-reasoned but unfortunately broad strokes in reaching a theoretical conclusion that reverse-engineering should be encouraged for purposes of enabling interoperability:

It is difficult to integrate these disparate welfare effects into an unassailable view as to whether reverse engineering for interoperability purposes should be legal. On balance, we believe that consumers benefit from interoperability because it encourages the development of a larger variety of software applications from a wider array of software developers with fewer wasted application development costs. Incentives to develop platforms are generally adequate owing to the high costs and difficulties of reverse-engineering software. Furthermore, interoperability lessens the potential for tipping into monopoly. Reverse engineering to achieve interoperability may also lessen a monopoly platform provider’s market power by providing application developers with an alternative means of entry if the monopolist’s licensing terms are unacceptable.²²⁶

Samuelson and Scotchmer’s scorecard consisted, by necessity, of ratings like “Worse (Adequate?)” and “Better (Too High?).”²²⁷ This Article cannot “read the scales” with any greater level of precision. However, several

223. *Id.* at 1585.

224. *Id.* at 1607 (discussing reverse engineering in several other contexts, such as the semiconductor industry).

225. *Id.* at 1621.

226. *Id.* at 1625-26.

227. Samuelson & Scotchmer, *supra* note 18, at 1621.

warning signs indicate that too much weight has been placed on the “restrictive” end of the reverse engineering scales.

V. WARNING SIGNS

A. Copyright Law Is Being Used as a Substitute for Patent Protection

Copyright law states that “[i]n 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. . . .”²²⁸ This protection is afforded only by patent law, which has stricter requirements and a more limited time frame.²²⁹ When copyright is used to award any party sole use of an idea or function, lawmakers have distorted copyright law.

Such a distortion occurs when software cannot be reverse engineered to examine the program’s unprotected elements, as the appeals courts in all of the *Sega*-era cases noted.²³⁰ However, because of the intertwined nature of idea and expression within software, and because of the gatekeeper function performed by access control measures, new restrictions on reverse engineering allow parties to monopolize ideas using the Copyright Act. Under the DMCA, a software access control measure may only be circumvented for purposes of enabling interoperability.²³¹ Thus, the ideas and functions embodied in a computer program that is encrypted or otherwise locked are given near-monopoly protection for the life of the copyright.

The “100% compatible PC” might never have taken flight in the DMCA era, assuming that IBM fashioned some form of access control measure to protect its BIOS. Compaq’s reverse engineering efforts were undertaken to develop a competing BIOS with functionality identical to

228. 17 U.S.C. § 102(b) (2007).

229. 35 U.S.C. § 101 (2006); *Walton v. United States*, 80 Fed. Cl. 251, 264 n.21 (Fed. Cl. 2008).

230. *Sony Computer Entm’t, Inc. v. Connectix Corp.*, 203 F.3d 596, 605 (9th Cir. 2000) (“If Sony wishes to obtain a lawful monopoly on the functional concepts in its software, it must satisfy the more stringent standards of the patent laws.”); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 842 (Fed. Cir. 1992) (“An 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.”); *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1526 (9th Cir. 1992) (“In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator of the work must satisfy the more stringent standards imposed by the patent laws.”).

231. 17 U.S.C. § 1201(f)(1) (2007).

the IBM BIOS, not to achieve interoperability with the IBM BIOS. Under the well-established boundaries of copyright law, Compaq should be permitted to copy the unprotected functionality of IBM's BIOS, so long as Compaq does not copy the expression with which IBM articulated these functions. However, under the DMCA, Compaq violates the Copyright Act unless it circumvents the access control measure "for the sole purpose of identifying and analyzing those elements of the program that are necessary to achieve interoperability of an independently created computer program with other programs."²³² Even if Compaq could convince a court that the interoperability exception is not limited to interoperability with the circumvented program, Compaq would also have to argue that hardware devices fall within the DMCA's meaning of "computer program[s]," as a BIOS enables interoperability between various pieces of hardware.²³³ A court is likely to reject one—or both—of these arguments.

It may seem unfair or unwise that copyright law does not protect the ideas embodied in expression, especially in the computer software context. Perhaps the principle is best illustrated in a more traditional copyright environment. Antonio Vivaldi's "Four Seasons" is one of the most cherished musical compositions of all time. If written today, it would have received copyright protection as a musical work.²³⁴ However, the *idea* of a violin concerto that mimics the sounds of nature as the seasons progress from spring to winter would be completely unprotected. Another composer would be free to "reverse engineer" Vivaldi's composition by listening to a recording or studying a printed score of the work in order to learn about the work. Then, the composer would be free to write a competing violin concerto that "sounds" like spring or winter, so long as she does not copy the precise expressions Vivaldi used to portray his idea. Similarly, society would not maximize progress in the arts by granting Matt Groening, creator of *The Simpsons*, a monopoly over an animated comedy featuring a dysfunctional family led by a dim-witted father, and as such, copyright law is concerned with reproductions of the expression, such as bootleg DVDs, and *not* reproductions of the idea (however blatant), such as *Family Guy*.

Yet, in Congress's attempt to curtail expression-reproduction (which was the rationale for the DMCA), it has severely curtailed idea-reproduction.²³⁵ As it stands, copyright law now grants absolutely no idea protection to copyrighted works lacking technological access control measures and robust idea protection to copyrighted works protected by

232. *Id.*

233. *Id.* § 101.

234. *Id.* § 102(a)(2).

235. S REP. NO. 105-190, at 8 (1998).

technological access control measures. This demarcation is baffling from a public policy perspective: an idea is no more deserving of monopoly protection merely because it is distributed with a technological access control measure. Patent law spells out the criteria for ideas that “deserve” monopoly protection: utility, novelty, and non-obviousness.²³⁶ One might argue that the presence of the access control measure indicates that the author so valued his creation that he took the effort to protect it, which leads to the conclusion that the idea is indeed “worth” protecting. However, a computer program embodies the same ideas and functions at the moment before and after the access control measure is implemented; its objective value to society has not changed, even if there is an indication of increased subjective value given by its creator.

Copyright law, even as it relates to computer programs, has never before offered monopoly protection for ideas, processes, and functions. This inconsistency is a warning sign that lawmakers have over-restricted reverse engineering.

B. Copyright Law Is Being Used to Capture Secondary Markets

Aside from protecting the idea of a product, which falls under the domain of patent law, certain parties are using reverse engineering restrictions to monopolize ancillary markets, a practice that is unacceptable under patent and copyright law.

The *Sega*-era plaintiffs were not able to use copyrights in game consoles and first-party software titles to control the market for third-party software titles, as was most pithily stated in *Connectix*: “Sony understandably seeks control over the market for devices that play games Sony produces or licenses. The copyright law, however, does not confer such a monopoly.”²³⁷

The DMCA has pushed us dangerously close to conferring such a monopoly. In two well-publicized cases, manufacturing companies attempted unsuccessfully to use copyright law to gain monopoly power over the markets for replacement parts. In *Lexmark Int’l, Inc. v. Static Control Components, Inc.*, a printer manufacturer attempted to control the market for replacement ink cartridges, and in *Chamberlain Group, Inc. v. Skylink Technologies, Inc.*, a garage door manufacturer attempted to control the market for garage door openers.²³⁸ In each case, the main component would not operate with the ancillary component unless the

236. 35 U.S.C. §§ 101-103 (2006).

237. *Sony*, 203 F.3d at 607.

238. *Chamberlain Group, Inc. v. Skylink Techs., Inc.*, 381 F.3d 1178 (Fed. Cir. 2004); *Lexmark Int’l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004).

latter performed a “secret handshake” to gain access to the main component.²³⁹ Defendants reverse-engineered the various components to create third-party ink cartridges and garage door openers.²⁴⁰ The respective plaintiffs sued, claiming that defendants had infringed copyrights in the authentication programs and violated the DMCA by circumventing technological measures that controlled access to the copyrighted computer programs that operated the machines.²⁴¹ Professor Jane Ginsburg was highly critical of these arguments:

The extraordinary consequence of plaintiffs’ reasoning is that any useful object whose workings are controlled by computer programs—and today, that means an endless variety of consumer and industrial goods—can come within the scope of section 1201 if the object’s producer makes access to those programs subject to an authentication sequence. As a policy matter, this result is inconceivable.²⁴²

“Happily,” according to Professor Ginsburg, plaintiffs’ arguments were rejected.²⁴³ Unhappily, however, the cases were not as open-and-shut as one might expect, and the courts required some creativity to reach their holdings.

The *Lexmark* court rejected the DMCA circumvention claim because the access control measure controlled access to the unprotected functionality of the program, not the copyrightable expression of the program itself. The court held, “[i]t is not Lexmark’s authentication sequence that ‘controls access’ to the Printer Engine Program. It is the purchase of a Lexmark printer that allows ‘access’ to the program.”²⁴⁴ The *Lexmark* court found determinative that the secret handshake was not necessary to view the object code of the printer software, only to *use* it.²⁴⁵ In other words, the technological device did not prevent access to a work

239. *Lexmark*, 387 F.3d at 530 (“If the code calculated by the [toner cartridge’s] microchip matches the code calculated by the printer, the printer functions normally. If the two values do not match, the printer returns an error message and will not operate, blocking consumers from using toner cartridges that Lexmark has not authorized.”)

240. *Id.* at 529; *Chamberlain*, 381 F.3d at 1183.

241. *See supra* text accompanying note 238.

242. Jane C. Ginsburg, *The Pros and Cons of Strengthening Intellectual Property Protection: Technological Protection Measures and Section 1201 of the U.S. Copyright Act*, at 4 (Feb. 1, 2007 version), available at <http://ssrn.com/abstract=960724>.

243. *Id.*

244. *Lexmark*, 387 F.3d at 546.

245. *Id.* at 546-47.

protected by the Copyright Act, as required by the DMCA, because it did not prevent access to any expression.²⁴⁶

The *Chamberlain* court interpreted the DMCA as requiring a link between circumvention and infringement.²⁴⁷ The *Chamberlain* court focused on the purpose of the circumvention, exculpating those who circumvent for legitimate reasons—such as creating an interoperable program or device—and inculcating those who circumvent for illegitimate reasons—such as piracy.²⁴⁸

Unfortunately, the reasoning of *Lexmark* may not apply in the context of more advanced computer programs, where technological protection measures control access to both functional and expressive content. *Lexmark*'s computer software had very thin copyright protection: only the creative elements of the source code, such as its layout and comments. All of this copyrighted expression was indeed accessible without using the software.

Video game software, on the other hand, has multiple layers of potentially copyrightable expression: the code itself, as well as the sounds and images created by the game. If, as *Lexmark* held, DMCA applicability turns on whether a technological protection measure controls access to copyrightable expression, then reverse engineers in the area of advanced software must proceed with caution.²⁴⁹

One could argue that the copyrightable expression that is viewed and heard during use, such as the musical score, is just as accessible without using the program as the object code, because anyone who purchases the software can read the portion of the program's binary code that translates into music. Under this interpretation of *Lexmark*, however, it is hard to think of any "technological [protection] measure" other than encryption of binary code, a limitation that the definitions of the DMCA seem to

246. 17 U.S.C. § 1201(a)(1)(A)(2007) ("[n]o person shall circumvent a technological measure that effectively controls access to a work protected under [the Copyright Act].").

247. *Chamberlain Group v. Skylink Techs.*, 381 F.3d 1178, 1204 (Fed. Cir. 2004).

A copyright owner seeking to impose liability on an accused circumventor must demonstrate a reasonable relationship between the circumvention at issue and a use relating to a property right for which the Copyright Act permits the copyright owner to withhold authorization—as well as notice that authorization was withheld. A copyright owner seeking to impose liability on an accused trafficker must demonstrate that the trafficker's device enables either copyright infringement or a prohibited circumvention.

Id.

248. *Id.* at 1198.

249. *Lexmark*, 387 F.3d at 548.

contradict.²⁵⁰ Also, while the musical score of a video game is typically a pre-determined string of binary code on the game's disc, the visual images created by a game are not.

A game displays images not by loading a pre-rendered series of images on the screen but by "painting" the screen based on complicated rendering software that employs thousands of variables. As such, the three-dimensional image one sees when Mario hurtles through the clouds is not itself captured in binary code in the software. The Ninth Circuit implicitly recognized as copyrighted expression visual images created by gaming software though it noted that any individual frame of an animated work is "an insignificant portion of the complex copyrighted work as a whole."²⁵¹ Therefore, copyrighted expression can truly be accessed only by *using* a video game or other audiovisual piece of software, which significantly limits the applicability of *Lexmark* in this area.

As for *Chamberlain*, Ginsburg argues that a court's interpolation of copyright violation into the requirements for circumvention liability may run counter to legislative intent:

[s]ome activities subject to access controls do not implicate traditional copyright owner rights such as reproduction and public performance. For example, an access control may limit the number of viewings of a motion picture distributed on a DVD. . . . The legislative history indicates that the DMCA was designed in part specifically to foster a variety of business models offering the public a diversity of levels of access, for a diversity of prices.²⁵²

If the DMCA intends to prohibit access control circumvention that does not result in copyright infringement, then the reasoning behind the *Chamberlain* holding is fallacious.

Thus, situations like *bned* are now possible, where Blizzard was able to capture the market for platforms on which to play its software (the Battle.net service) with the copyright protection of the software itself, even though Sega, Nintendo, and Sony failed in similar attempts only a decade earlier.²⁵³

The next great battle over the DMCA, reverse engineering, and ancillary market capture is likely to involve the Apple iPod. Through the use of a technological protection measure called FairPlay, Apple has been able to ensure that songs downloaded using its iTunes program will

250. 17 U.S.C. § 1201(a)(3) (2007).

251. *Sony Computer Entm't Am., Inc. v. Bleem, LLC*, 214 F.3d 1022, 1028 (9th Cir. 2000).

252. Ginsburg, *supra* note 242, at 6 (citing H.R. Rep. No. 105-551, pt. 2, at 23 (1998)).

253. *See supra* Part II.B.

operate only on Apple devices, such as the iPod.²⁵⁴ A computer scientist reverse engineered FairPlay in such a way that iTunes tracks are playable on any device but without sacrificing its piracy-prevention functionality.²⁵⁵ Apple has yet to file suit, but if it does, the strength of the DMCA's reverse engineering exception will once again be scrutinized. Section 1201(f) exempts only program-to-program interoperability, and this case would involve song-to-device interoperability.²⁵⁶ As such, Apple may be able to use an anti-piracy provision in copyright law to retain a monopoly in hardware devices that are ancillary to the copyrighted software.

Of course, interpretations of the DMCA with regards to reverse engineering are of no consequence if parties have an unlimited ability to contractually restrict use of software, a topic discussed in the next section.

C. The Copyright Act Has Become a "Default Rule," Valid Only in the Absence of a Contract

The increasing ability to expand protection for copyrighted works through the use of non-negotiable "contracts" is troublesome for two reasons. First, justifications for this contract-based approach are based on assumptions about contract formation that are not present in the modern software industry. Second, this approach is a threat to the delicate balance of public policy that supports intellectual property law.

Freedom of contract is an admirable value for lawmaking bodies. The ability to forge agreements based on private preferences is central to the concept of liberty:

it must not be forgotten that the right of private contract is no small part of the liberty of the citizen, and that the usual and most important function of courts of justice is rather to maintain and enforce contracts[,] than to enable parties thereto to escape from their obligation on the pretext of public policy, unless it clearly appear that they contravene public right or the public welfare.²⁵⁷

254. See How Fair Play Works: Apple's iTunes DRM Dilemma, Roughly Drafted Media, Feb. 26, 2007, <http://www.roughlydrafted.com/RD/RDM.Tech.Q1.07/2A351C60-A4E5-4764-A083-FF8610E66A46.html> (last visited Nov. 19, 2008).

255. *Id.*; see also Robert Levine, *Unlocking the iPod*, FORTUNE, Oct. 30 2006, available at http://money.cnn.com/magazines/fortune/fortune_archive/2006/10/30/8391726/index.htm (last visited Oct. 1, 2008).

256. 17 U.S.C. § 1201(f)(1) (2007).

257. *Baltimore & O.S.R. Co. v. Voigt*, 176 U.S. 498, 505 (1900).

Of course, freedom of contract is no longer granted the same lofty status as during the *Lochner* era,²⁵⁸ but the principles of America are inexorably linked to the economic philosophies of Adam Smith, John Stuart Mill, and Jeremy Bentham, who “insisted [that] freedom of bargaining [is] the fundamental and indispensable requisite of progress.”²⁵⁹

Even Lemley, an opponent of allowing parties to expand copyright protections through contract, admits, “[t]he enforcement of agreements between private parties is an intrinsic part of our system of laws, and is not to be discarded lightly.”²⁶⁰ Tom Bell and Maureen A. O’Rourke agree, concluding that copyright law should be treated as a default rule, valid only in the absence of private agreement.²⁶¹

One argument advanced in favor of copyright law as a default rule is the Coase Theorem. According to the Coase Theorem, bargaining achieves an optimal allocation of resources compared to government regulation in the absence of transaction costs.²⁶² Efficient market-driven outcomes require the presence of knowledge and assent in contract formation:

Two fundamental requirements of the neoclassical model of social ordering through private exchange are knowledge of contract terms and meaningful (i.e., voluntary and fully informed) assent. Both are necessary (though not sufficient) requirements for an “unregulated” market to reach the efficient equilibrium point; the absence of either

258. *Lochner v. New York*, 198 U.S. 45, 53 (1905) (“The general right to make a contract in relation to his business is part of the liberty of the individual protected by the 14th Amendment of the Federal Constitution.”), *overruled by* *W. Coast Hotel Co. v. Parrish*, 300 U.S. 379, 391-92 (1937).

The Constitution does not speak of freedom of contract . . . [Rather,] the liberty safeguarded is liberty in a social organization which requires the protection of law against the evils which menace the health, safety, morals, and welfare of the people . . . This essential limitation of liberty in general governs freedom of contract in particular.

Id.

259. Samuel Williston, *Freedom of Contract*, 6 CORNELL L.Q. 365, 366 (1921) (“Jeffersonian democracy finds its cardinal tenet in restricting government [] activities and allowing the individual free play.”).

260. Lemley, *supra* note 95, at 1284.

261. See Tom W. Bell, *Fair Use vs. Fared Use: The Impact of Automated Rights Management on Copyright’s Fair Use Doctrine*, 76 N.C. L. REV. 557 (1998); Maureen A. O’Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 DUKE L.J. 479 (1995).

262. DAVID BESANKO & RONALD BRAEUTIGAM, *MICROECONOMICS: AN INTEGRATED APPROACH* 747-49 (2002).

or both may signal a market failure justifying some form of adjustment.²⁶³

However, in the modern consumer arena, where terms are dictated by the seller and standardized across sellers, and where buyers have questionable means to assent or decline, the requirements for efficient outcomes via private bargaining are arguably absent.²⁶⁴

First, the contracts included with mass-produced software are dictated entirely by the seller, who selects all the terms and offers no opportunity to negotiate. Aside from the policy ramifications of this inequity, which are discussed below, the complete lack of actual negotiations substantially decreases the likelihood that the buyer understands the terms of the contract. To the typical purchaser of software, the terms of a boilerplate software license are as informative, understandable, and attention-grabbing as Judge Dyk's hypothetical black dot.²⁶⁵ For better or worse, the

263. Julie E. Cohen, *Lochner in Cyberspace: The New Economic Orthodoxy of "Rights Management,"* 97 MICH. L. REV. 462, 482 (1998); see also Mark A. Lemley, *Beyond Preemption: The Law and Policy of Intellectual Property Licensing,* 87 CAL. L. REV. 111, 169-70 (1999).

264. Lemley, *supra* note 95, at 1286-87.

The arguments for contract, moreover, lose much of their force in the case of shrinkwrap licenses governing software transactions. Contract law is at its strongest where there is an actual agreement between the parties. That is, after all, the basis of a contract. Unfortunately, the ideal assumptions behind a "bargained contract"—relatively equal bargaining power, actual discussion and agreement as to individual terms, and joint drafting—often do not reflect reality. In practice, contracts are often entered into between large corporations and consumers. The corporation drafts the contract and sets the price. The consumer merely decides whether or not to sign the contract. In this circumstance, there has been no meaningful "bargaining" over the contract at all, and certainly not over particular terms contained in the form drafted by the large corporation.

Id.

265. Robert A. Hillman & Jeffrey J. Rachlinski, *Standard-Form Contracting in the Electronic Age,* 77 N.Y.U. L. REV. 429, 446 (2002).

Reading and understanding boilerplate terms is difficult and time consuming for consumers. Consumers recognize that they are unlikely to understand the lengthy and complicated legal jargon in the boilerplate. To make matters worse, consumers commonly encounter standard forms when they are in a hurry. Businesses also can create boilerplate that is difficult to read by using small print, a light font, and all-capital lettering and by burying important terms in the middle of the form.

Id.

consumer is unlikely to have knowledge of a contract's terms.

Second, no "meeting of the minds" occurs between buyer and seller with respect to the terms included in most software licenses. Until recently, assent to a license was indicated by simply keeping or using the software, and the license was included as a paper document amidst the slew of documentation that came with a commercial software package.²⁶⁶ Today, buyers are given a slightly better opportunity to recognize and assent to license terms, as the license is typically displayed on the computer screen, and the program is not installed until the user clicks an "I agree" button.²⁶⁷

Still, one may argue that the terms of the license agreement are first presented to the buyer after the point of purchase, which is the moment the contract is formed.²⁶⁸ As such, the license is merely an offer to modify whatever agreement was reached when the user accepted the offer to purchase the software. This argument has met with mixed results in the context of license terms included in the packaging of computer hardware.²⁶⁹

266. *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 257 n.2 (5th Cir. 1988) (the license at issue stated the following: "BY USING ANY OF THE ENCLOSED DISKETTE(S), YOU AGREE TO THE FOLLOWING PROVISIONS. IF YOU DO NOT AGREE WITH THESE LICENSE PROVISIONS, RETURN THESE MATERIALS TO YOUR DEALER, IN ORIGINAL PACKAGING WITHIN 3 DAYS FROM RECEIPT, FOR A REFUND.").

267. *See, e.g., Davidson & Assocs. v. Jung*, 422 F.3d 630, 634-35 (8th Cir. 2005). To some extent, the older method for registering assent was necessitated by limits on disk space that prevented programs from displaying the license agreement on the screen. For instance, the license for EA Sports Madden 2007 software program requires eleven kilobytes of disk space. By comparison, the entire Super Mario Brothers Nintendo program requires forty-one kilobytes of space.

268. Lemley, *supra* note 263, at 120.

269. *Klocek v. Gateway, Inc.*, 104 F. Supp. 2d 1332, 1341 (D. Kan. 2000).

Gateway argues that plaintiff demonstrated acceptance of the arbitration provision by keeping the computer more than five days after the date of delivery. Although the Standard Terms purport to work that result, Gateway has not presented evidence that plaintiff expressly agreed to those Standard Terms. Gateway states only that it enclosed the Standard Terms inside the computer box for plaintiff to read afterwards.

Id. Compare Hill v. Gateway 2000, Inc., 105 F.3d 1147, 1149 (7th Cir. 1997).

Cashiers cannot be expected to read legal documents to customers before ringing up sales. If the staff at the other end of the phone for direct-sales operations such as Gateway's had to read the four-page statement of terms before taking the buyer's credit card number, the droning voice would anesthetize rather than enlighten many potential buyers.

Further, assent is not fully voluntary in the absence of an adequate means of rejecting the contract terms. The End User License Agreement for Microsoft Windows XP contains the following provision: "IF YOU DO NOT AGREE, DO NOT INSTALL, COPY, OR USE THE SOFTWARE; YOU MAY RETURN IT TO YOUR PLACE OF PURCHASE FOR A FULL REFUND, IF APPLICABLE."²⁷⁰ A newer Blizzard Entertainment program, entitled World of Warcraft, contains a similar provision.²⁷¹ This contract gives little choice to the buyer, as no major retailer offers refunds for opened software.²⁷² By the time the buyer sees the contract terms, he can either accept the terms or throw away the software without hope of refund. Windows Vista provides an option to return the software directly to Microsoft for a refund (plus \$7 in shipping expenses), provided that the user completes a list of information, is willing to wait up to four weeks for "processing," and has a physical address ("sorry, no P.O. boxes").²⁷³ "In these circumstances," Professor Lemley argues, "it is simply unrealistic to believe that a failure to return the software for a refund constitutes any form of 'agreement' to the terms of the form license."²⁷⁴

Id.

270. Microsoft Windows XP Home Edition (Retail) End-User License Agreement for Microsoft Software, *available at* <http://www.microsoft.com/korea/windowsxp/home/eula.msp> (last visited Oct. 21, 2008).

271. World of Warcraft End User License Agreement, *available at* <http://www.wow-europe.com/en/legal/eula.html> (last visited Oct. 23, 2008) ("If you do not agree with the Terms of Use, then (i) you should not register for an Account to play the Game, and (ii) you should arrange to return the Game to the place where you purchased the Game within thirty (30) days of the original purchase.").

272. *See* Returning Software, amazon.com, <http://www.amazon.com/gp/help/customer/display.html?ie=UTF8&nodeId=901928> (last visited Oct. 23, 2008) ("Software sold in New condition which has been opened, and Software products which are sold as Used and are not materially different than the item purchased, cannot be returned."); Returns Policy—Shipping and Returns—Help—Walmart. com, <http://www.walmart.com/returns> (last visited Oct. 23, 2008) ("Computer software: Must be returned unopened."); Store Return Policy, bestbuy.com, *supra* note 195 ("Opened computer software, movies, music and video games can be exchanged for the identical item but cannot be returned for a refund.").

273. Microsoft North American Retail Product Refund Guidelines, *available at* <http://www.microsoft.com/mscorp/productrefund/refund.msp> (last visited Oct. 23, 2008).

274. Lemley, *supra* note 95, at 1289.

Even if we accept the notion of blanket assent uncritically, a significant percentage of the consumers purchasing the software may be unable to return it for a refund. A second group may choose not to return the software because they believe the shrinkwrap license is unenforceable—a commonly held belief among software consumers and one with significant support in the current case law. A third group may choose not to return the software because it is simply too much

Purchasers of retail software have uncertain knowledge of contract terms, and assent is often the only practicable choice. In these respects, the software market fails to satisfy the requirements of the neoclassical model of social ordering through private contracting.²⁷⁵ Still, many argue that boilerplate contracts are socially desirable because negotiation is unwanted and inefficient with respect to the minor details underlying a retail transaction.²⁷⁶ Indeed, sophisticated companies are usually better-equipped to craft many contract terms (such as risk allocation), the typical consumer does not even want to *read* a contract, let alone negotiate each and every term, and retail software would be prohibitively expensive if producers had to employ an army of agents to handle negotiations.

However, permitting boilerplate contracts to avoid transaction costs is most prudent where “[c]ompetition in the market for the goods or services can provide courts with some assurance that businesses will not supply exploitative terms.”²⁷⁷ The level of competition in many software markets is less than reassuring. Three companies—Nintendo, Sony, and Microsoft—control 100% of the market for new video game consoles.²⁷⁸ Microsoft Windows has a 90.29% share of the operating system market.²⁷⁹ Google has a 79.90% share of the search engine market.²⁸⁰ On the other hand, distribution is less lopsided in applications markets, though there are notable exceptions, such as office software (Microsoft Office) and sports games (Electronic Arts).²⁸¹ As such, there is room to debate whether

effort, or because they need the software right away and cannot afford to wait and select a different brand. A final group may object to particular terms in a license, but may be unwilling to reject the software as a whole, perhaps because they cannot find an equally attractive alternative that does not require a shrinkwrap license.

Id.

275. See *supra* text accompany notes 260–63.

276. See Hillman & Rachlinski, *supra* note 265, at 435–37 (“The consumer, engaging in a rough but reasonable cost-benefit analysis . . . understands that the costs of reading, interpreting, and comparing standard terms outweigh any benefits of doing so and therefore chooses not to read the form carefully or even at all.”).

277. *Id.* at 442.

278. See Video Game Chartz, <http://www.vgchartz.com/> (last visited Oct. 23, 2008).

279. See Operating System Market Share for September 2008, <http://marketshare.hitslink.com/report.aspx?qprid=8> (last visited Oct. 23, 2008).

280. See Search Engine Market Share for September 2008, <http://marketshare.hitslink.com/report.aspx?qprid=4> (last visited Oct. 23, 2008).

281. Steve Hamm, *More to Life Than the Office*, BUSINESS WEEK, July 3, 2006, http://www.businessweek.com/magazine/content/06_27/b3991412.htm (last visited Nov. 19, 2008); Matt Richtel, *Electronic Arts and ESPN Sign 15-Year Deal to Sell Games*, N.Y. TIMES, Jan. 18, 2005, <http://www.nytimes.com/2005/01/18/technology/18games.html> (last visited Nov. 19, 2008).

boilerplate software contracts lead to an efficient allocation of resources; the answer may depend on the sub-market in question.

Second, and more damning, the increased ability to impose software licenses completely ignores the underpinnings of intellectual property law, which seek to balance the interests of society and artist/inventor by providing whatever limited set of rights is necessary to induce intellectual creation.²⁸² Instead, the software industry is leaning toward a world of “private legislation,” in which parties who are in a position to write contracts can jointly impose uniform terms that no one can escape.”²⁸³ Where the creator defines the terms of “copyright” protection, and competition is not robust, the incentive level will be set too high, and society will not make optimal use of the creation.

Bowers, the leading glimpse into the world of private copyright legislation, illustrates just how much creators may shift the precarious balance of intellectual property policy with contract law. The contract in *Bowers* “unambiguously prohibits ‘reverse engineering,’” which the court defined as “to study or analyze (a device, as a microchip for computers) in order to learn details of design, construction, and operation, perhaps to produce a copy or an improved version.”²⁸⁴ What if the *Bowers* defendant had merely *used* the plaintiff’s program? When an individual uses a program, she is arguably analyzing the program in order to learn details of its design and operation, at least at a semi-subconscious level. The first time she operates the program, she analyzes the layout of the screen. When she first needs to execute a function, such as “Tools -> Word Count” (in Microsoft Word), she analyzes the menus to deduce the location of the function. As a gamer uses Blizzard’s WarCraft, she analyzes the actions of the computer-controlled opponents to learn how to best combat her fictional enemies. In the broader sense of reverse engineering (“the process of extracting know-how or knowledge from a human-made artifact”),²⁸⁵ everybody reverse engineers when using any program, in violation of most end user license agreements. While software creators are unlikely to sue customers for learning how to use their programs, a creator has a colorable contract action against a user that, having deduced the central idea of the creator’s program *by mere use of that program*, develops a competing software title. Absurd? Certainly. But the *Bowers* majority offers no indication that it would draw a line between the actual scenario and this hypothetical.

282. See *supra* Part IV.

283. Lemley, *supra* note 263, at 148.

284. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1326 (Fed. Cir. 2003) (quoting Random House Unabridged Dictionary (1993)).

285. Samuelson & Scotchmer, *supra* note 18, at 1577.

Nor is competition likely to offer a panacea to the policy dilemma. Even accepting the dubious assumption that various software packages offer materially different terms,²⁸⁶ a free-rider problem and inadequate information will almost certainly prevent consumers from favoring software titles that have innovation-friendly terms, and as such, software producers will feel no pressure to include these terms. For instance, few buyers would likely prefer a software program whose contract does not modify the *Sega* right to reverse engineer. Almost all purchasers of software have no intention of engaging in reverse engineering and are indifferent to this license term, whereas commercial software producers are highly interested in including a no-reverse-engineering clause. Thus, the market-based outcome dictates that such a clause is optimal, as it gives benefit to one party without apparent cost to the other. However, all buyers—as members of society—benefit from the right to reverse engineer, as it leads to innovations that are beneficial to society.²⁸⁷

Even if consumers were cognizant of this benefit, free-rider theory suggests that an individual consumer would hesitate to alter her purchase decision unless the competing packages were otherwise identical. If Software X (reverse engineering acceptable) lacked one feature found in Software Y (reverse engineering prohibited) or was identical but cost one additional dollar, we might prefer Software X as a society, but Software Y is almost certain to “win” in the marketplace. If Software X is produced by a different company than Software Y, Company X would cease to exist in a perfectly competitive market. If Software X and Software Y are produced by the same company—that is, the company is merely employing price discrimination by charging a higher price (almost certainly more than our hypothetical one dollar) to those who wish to reverse engineer—then the outcome is equivalent to granting full idea protection. Here, the company is offering to license the unprotected

286. Hillman & Rachlinski, *supra* note 230, at 439 (“most businesses will offer terms similar to those offered by their competitors. Less experienced businesses simply copy their senior counterparts.”).

287. See Netanel, *supra* note 183, at 296 (“Robust public debate, the spread of knowledge, and the questioning of cultural hierarchy are of paramount importance to a democratic society. To the extent that these activities bolster democratic institutions, all citizens benefit from their occurrence.”); Lemley, *supra* note 95, at 1278-79 (noting that copyright law’s “allocation of rights reflects a ‘delicate balance’ between many different interests, and not all of those interests are represented in licensing contracts. Allowing private parties to avoid the effects of intellectual property law would undermine that balance[,]” and arguing that the enactment of a U.C.C. code that would increase software producer’s ability to impose licenses “would represent a complete victory for the forces of contract over the forces of federal law—and for licensors over licensees, consumers, and the public.”).

elements of its program, much like a patentee licensing the use of his or her patented design.

This “price discrimination” world is an improvement over a world in which contracts completely obliterate the right to reverse engineer. After all, those individuals who engage in reverse engineering are likely to reap the commercial benefits from their efforts, so they should be willing to pay a disproportionate share of society’s cost for obtaining the innovation.

However, *any* purely market-based system is likely to suffer from externalities—third-party interests that are not considered by the contracting parties leading to a societally undesirable price/quantity level. In the world where the right to reverse engineer must be purchased, parties are unlikely to capture all the societal benefit from innovation—a positive externality that leads to an insufficient demand for reverse engineering. The impossibility of including every member of society in these negotiations is a transaction cost that impedes the Coasian efficient outcome.

Copyright law is an attempt to carefully balance creators’ incentives versus the public good, in accordance with Congress’s constitutional charge,²⁸⁸ and the multitude of interests involved in setting this public policy cannot be served by “private legislation.” Just as the proverbial fox is not allowed to guard the chicken coop, copyright owners cannot be trusted to grant themselves *just enough* monopoly power to earn a return adequate to incentivize their work, which is the amount of monopoly power authorized by the Constitution.²⁸⁹ Nor can we trust the market to reach this equilibrium, thanks to a free-rider problem, a too-imperfect bargaining arena, and inadequate competition. Without limitations on the ability to contract around certain tenets of copyright law, society is unlikely to continue benefiting from the idea-expression dichotomy, a multitude of fair uses (including the right to reverse engineer), and the limited term of protection.²⁹⁰ As such, copyright law cannot be thought of

288. Netanel, *supra* note 183, at 385.

289. U.S. CONST. art. I, § 8, cl. 8.

290. Lemley, *supra* note 263, at 128-32. For the opposite viewpoint, see Raymond T. Nimmer, *Breaking Barriers: The Relation Between Contract And Intellectual Property Law*, 13 BERKELEY TECH. L.J. 827, 828-29 (1998).

As I read some of the commentary about the future of copyright in the information industries on the Internet, it often appears as if some believe that we are facing an impending big bang as the fields of copyright and contract head toward some unclearly defined, but cataclysmic conflict . . . Nothing could be further from the truth and, indeed, the fundamental premise is flawed.

as a “default rule,” valid only in the absence of alternate terms. Cases like *Bowers* and *ProCD*, which advocate for unbridled freedom of contract in intellectual property, are warning signs that the freedom to reverse engineer is in peril.

VI. CONCLUSION

These changes in copyright and contract law are most likely linked to the fight against piracy in the digital age.²⁹¹ Preventing piracy is a laudable goal for intellectual property. The exponential growth in piracy enabled by the internet is a frightening enemy for the artists, creators, and innovators who “promote the Progress of Science and useful Arts. . . .”²⁹² Nowhere is this devastation more evident than in the music industry, where sales have plummeted while a billion songs are illegally traded online per month.²⁹³ However, in their attempt to stem the tide of piracy, legislators and courts should not allow reverse engineering to become collateral damage. The American economy is built on innovation, and our leadership in technological industries is under constant challenge by brilliant engineers and scientists around the world.²⁹⁴ The unprecedented shifts in copyright doctrine—idea protection, ancillary market protection, and contract law’s usurpation of copyright law—are warning signs that the ability to reverse engineer is less robust than in the *Sega* era. If America is to continue encouraging innovation, these warning signs must not be ignored.

291. See S. REP. NO. 105-190, at 8 (1998).

292. U.S. CONST. art. I, § 8, cl. 8.

293. Ethan Smith, *Sales of Music, Long in Decline, Plunge Sharply*, WALL ST. J., Mar. 21, 2007, at http://online.wsj.com/article_email/SB117444575607043728-lMyQjAxMDE3NzI0MTQyNDE1Wj.html.

294. See The Task Force on the Future of American Innovation, *The Knowledge Economy: Is the United States Losing Its Competitive Edge?* (Feb. 16, 2005), <http://www.futureofinnovation.org/PDF/Benchmarks.pdf> (noting the importance of innovation to the American economy and detailing the various statistical measures by which America’s leadership is declining).

LexisNexis® Rewards of Working Smarter



Over **17,000**
Cash Winners

\$100
weekly

\$1000
monthly

\$100,000*
grand prize: law school tuition

Plus, daily cash winners for using LexisNexis!

It's simple...

Just go to www.lexisnexis.com/lawschool to view the official rules and register. After that, each time you use the LexisNexis™ research service, you're entered to win!

Use LexisNexis...
Where it really pays to research!

*Grand prize is based on actual student's tuition rate as of 9/02/01 at the school in which student is enrolled. Up to a maximum of \$100,000.

LexisNexis and the Knowledge Burst logo are trademarks of Reed Elsevier Properties Inc., used under license. ©2001 LexisNexis, a division of Reed Elsevier Inc. All rights reserved.



LexisNexis™