

NOTE

SMOTHERED BY JUDICIAL LOVE: HOW *JACOBSEN V. KATZER* COULD BRING OPEN SOURCE SOFTWARE DEVELOPMENT TO A STANDSTILL

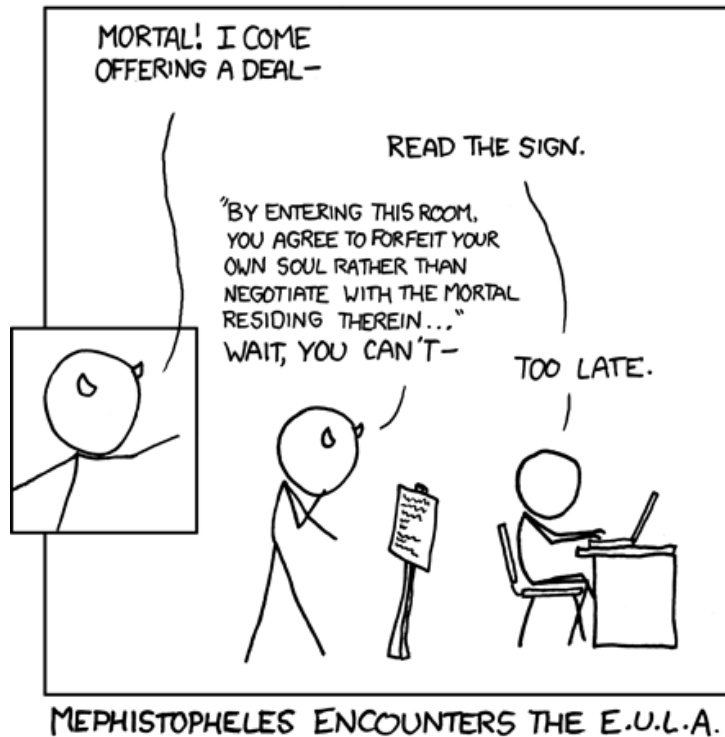
*Benjamin I. Narodick**

I. INTRODUCTION	265
II. BACKGROUND	268
A. <i>Copyright Protection & Software Programming</i>	268
B. <i>Emergence of Software Use Licenses</i>	270
C. <i>Development of Open Source Licensing</i>	271
III. <i>JACOBSEN V. KATZER</i>	273
A. <i>District Court Proceedings</i>	273
B. <i>Federal Circuit Proceedings</i>	274
C. <i>New Issues on Remand</i>	276
D. <i>Jacobsen, Comparable Precedent, and Future Effects</i>	277
1. <i>Equal Treatment for Open Source Software Distributors</i>	277
2. <i>Recognition of Intellectual Property Rights Outside of the Copyright Act</i>	278
IV. STATE OF REVERSE ENGINEERING	282
A. <i>Pre-Jacobsen Law of Reverse Engineering</i>	283
B. <i>Reverse Engineering and a Strict Reading of Jacobsen</i>	285
C. <i>Jacobsen, Reverse Engineering and the DMCA</i>	286
D. <i>Comparative Effects of Jacobsen on Commercial and Open Source Software</i>	287
V. ANALYSIS OF LEGAL ALTERNATIVES	288
A. <i>EULAs, Reverse Engineering and Unconscionable Contracts</i>	288
1. <i>The U.C.C. Test for Unconscionability</i>	288
2. <i>The ALI Test for Unconscionability</i>	289
B. <i>Effect of Antitrust Law on Reverse Engineering Prohibitions</i>	292
VI. CONCLUSION	293

* J.D., Boston University School of Law, 2010; B.A. Political Science & Rhetoric, University of California, Berkeley, 2007.

2010]

OPEN SOURCE SOFTWARE



I. INTRODUCTION

In the ever-changing world of high-tech intellectual property law, a recent legal dispute over model train sets – a prevalent hobby in American culture going as far back as the middle of the 19th century – has the potential to dramatically reshape copyright protection for software.² Judges and legislatures have struggled to adapt copyright law to software since Congress's original decision to extend the realm of copyright protection to cover such material.³ This struggle stems from the challenge of the copyright regime to protect original expression without restricting public access to an idea, and "[t]he essentially utilitarian nature of a computer program further complicates

¹ Randall Munroe, *xkcd - A Webcomic of Romance, Sarcasm, Math, and Language: Faust 2.0*, <http://www.xkcd.com/501> (last visited Mar. 6, 2010). Reproduced with permission of author.

² Christopher Lehmann-Haupt, *Books of the Times: The Trains on Avenue de Rumine*, N.Y. TIMES, Jan. 25, 1984, at C23.

³ See *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 430-32 (1984).

the task of distilling its idea from its expression.”⁴

Related to the issue of clarifying the ramifications of the idea-expression dichotomy is another pivotal issue in copyright protection – the inherent uniqueness of the process for creating software. One of the predominant, and also most controversial, methods of software production and design is reverse engineering.⁵ This method is particularly useful in the software industry as a way to ensure cross-compatibility between different programs, hardware platforms, and operating systems.⁶ Legal protection for reverse engineering of software relies on a case-by-case application of the fair use doctrine, primarily because the factors used to determine fair use are unordered, non-exclusive, and include a mixture of fact and law.⁷

Many popular open source software programs owe their existence to the reverse engineering of previous commercial programs performing the same or similar functions.⁸ Legal protection for reverse engineering is particularly important for small-scale software developers and those who distribute their works under open source licenses. Open source software encompasses original programs dedicated for cost-free public use under certain terms and conditions, typically as parts of public collaborative projects.⁹ The popularity of open source programming continues to increase with technology advances that make the programs more accessible to the public. As many as one hundred million works, ranging from web browsers and operating systems to advanced server programs, are currently licensed under various Creative Commons licenses.¹⁰ These licenses allow for efficient completion and debugging of new programs by any number of programmers and conditionally allow redistribution of copyrighted software in a controlled manner.¹¹

In August 2008, the Federal Circuit recognized the legally binding nature of the Artistic License, one form of an open source license, in *Jacobsen v. Katzer*.¹² Katzer, a commercial software maker, copied literal elements of

⁴ Computer Assocs. Int’l v. Altai, 982 F.2d 693, 704 (2d Cir. 1992).

⁵ Andrew Johnson-Laird, *Reverse Engineering of Software: Separating Legal Mythology from Actual Technology*, 5 SOFTWARE L.J. 331, 353 (1992).

⁶ Karen E. Georgenson, Comment, *Reverse Engineering of Copyrighted Software: Fair Use or Misuse?*, 5 ALB. L.J. SCI. & TECH. 291, 291-92 (1996).

⁷ Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520-22 (9th Cir. 1992).

⁸ Noah Clements, *Be Careful What You Wish For: How the Jacobsen v. Katzer Decision Could Hurt the Free Software Movement*, Aug. 25, 2008, <http://blog.actonline.org/2008/08/be-careful-what.html>.

⁹ *Jacobsen v. Katzer*, 535 F.3d 1373, 1378-79 (Fed. Cir. 2008).

¹⁰ *Id.*

¹¹ *Id.*

¹² *Id.* at 1380-83.

program code from Jacobsen's competing open source program.¹³ Katzer then patented the new program and offered it for sale without following the terms of use of the original program's Artistic License.¹⁴ Overturning the decision of the district court, the Federal Circuit held that software license conditions, which were traditionally dismissed due to a lack of immediate economic consideration, may be legally valid and enforceable if the conditions have even remote economic value to the copyright holder, regardless of whether the original material is being sold for profit.¹⁵

In protecting the rights of open source copyright holders, the Federal Circuit continues a trend of greatly increasing the potential power of all licensing agreement conditions, with potentially serious consequences for the open source community and the software development community at large. This Note argues that these increasingly powerful software license conditions may not just endanger the reverse engineering practices that have sustained the rapid growth of the open source movement, but also the software industry in general. The holding in *Jacobsen* illustrates many of the inherent conflicts between the freedom of contract associated with software licenses and fair use protection for reverse engineering. Software producers institute and enforce license provisions on a private basis and without the support of statutory prohibitions like those that doomed restrictive use conditions in the early 1990s. Judicial enforcement of modern software licenses may lead to a point where the American judicial system fails to preserve the health of the software industry by overprotecting copyright holders.

Part II of this Note briefly explores the rough history of copyright protection for software and its evolution over the past two decades, as well as the concurrent rise of open source software licenses. Part III analyzes the specific facts leading up to the *Jacobsen* decision, the key points in the district court decision originally suppressing the effectiveness of the Artistic License, and identifies the key policy changes. Part IV addresses the current legal state of reverse engineering under a strict reading of cases similar to *Jacobsen* as well as in the context of the Federal Copyright Act ("Copyright Act"), the Digital Millennium Copyright Act ("DMCA"), the history of reverse engineering as fair use, and policy reasons for potentially holding license conditions unenforceable in the context of the goals of the patent and copyright process. Part V discusses alternative perspectives of legal analysis that may provide more balanced and appropriate legal judgments when addressing issues of open source software and reverse engineering.

¹³ *Id.* at 1379.

¹⁴ *Id.* at 1379-82.

¹⁵ *Id.* at 1380-83.

II. BACKGROUND

A. *Copyright Protection & Software Programming*

American courts have had repeated and recorded difficulties in applying intellectual property and copyright law to computer software, with one court referring to the practice as “reflect[ing] the courts’ attempt to fit the proverbial square peg in a round hole.”¹⁶ Many of these difficulties led to the initial denial of copyright protection for software even while those programs became more popular and publicly accessible.¹⁷ The major conceptual problem with the current relationship between software and copyright law is that copyright protection generally extends to creative works of authorship while computer software is generally utilitarian in nature. Much like other copyrighted utilitarian works, there are questions and limits as to how far, if at all, courts can extend copyright protection.¹⁸

In order to partially resolve this problem, federal courts have gradually incorporated the idea-expression dichotomy as it applies to software.¹⁹ The idea-expression dichotomy dictates that copyright law may protect only specific expressions of an idea – the idea itself cannot be withheld from the public.²⁰ Courts generally apply this principle by analyzing the computer software program at different levels of abstraction – from the literal code to the way in which modules of code interact and function – and filtering out *scène à faire* material – familiar themes that are staples of the industry and do not receive copyright protection.²¹ Additionally, courts filter out software elements that are dictated by efficiency or basic need.²² However, this analysis occurs on a case-by-case basis, unbounded by any bright-line rules.²³

Giving software copyright protection presents an additional problem – that software is not generally held to be communicative to humans in the same way as traditional works of authorship.²⁴ Early copyright cases on software protection reflected the idea that

¹⁶ *Computer Assocs. Int’l v. Altai*, 982 F.2d 693, 712 (2d Cir. 1992).

¹⁷ *Data Cash Sys., Inc. v. JS&A Group, Inc.*, 480 F. Supp. 1063, 1066 (N.D. Ill. 1979).

¹⁸ MARK A. LEMLEY, PETER S. MENELL & ROBERT P. MERGES, *INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE* 987 (rev. 4th ed. 2007).

¹⁹ *Atari Games Corp. v. Nintendo of Am. Inc.*, 975 F.2d 832, 842 (Fed. Cir. 1992).

²⁰ *Id.*

²¹ *Cavalier v. Random House, Inc.*, 297 F.3d 815, 823 (9th Cir. 2002) (citing *Berkic v. Crichton*, 761 F.2d 1289, 1294 (9th Cir. 1985)).

²² *Computer Assocs. Int’l v. Altai*, 982 F.2d 693, 703-09 (2d Cir. 1992).

²³ *Id.* at 715.

²⁴ Brett Frischmann & Dan Moylan, *The Evolving Common Law Doctrine of Copyright Misuse: A Unified Theory and Its Application to Software*, 15 *BERKELEY TECH. L.J.* 865, 903 (2000).

[p]rograms are profoundly different from the various forms of ‘works of authorship’ secured under the Constitution by copyright. Works of authorship have always been intended to be circulated to human beings and to be used by them – to be read, heard, or seen, for either pleasurable or practical ends. Computer programs, in their mature phase, are addressed to machines.²⁵

Congress has since legislated out this criticism and most courts do not acknowledge this as a valid obstacle to copyright protection, but it serves as another reminder of the unique and awkward relationship that software and copyright law continue to share.²⁶

Finally, the very nature of software use inherently requires making copies, dampening the potential for any sort of absolute copyright protection.²⁷ “[T]o use a computer you have to copy. You have to copy to screen. You make a backup copy to start. Everything you do is copying. So it seems ridiculous to have a copyright law that is applied to something in which you really want to encourage copying.”²⁸ Modern statutes provide some accommodations for the nature of software use, but it again represents the modifications and special tailoring that the Copyright Act requires to even come close to adequately protecting software programming.²⁹

Despite these and other functional problems, Congress acted on the recommendations of the National Commission on New Technological Uses of Copyrighted Works (“CONTU”) and passed legislation in 1980 amending the Copyright Act to explicitly protect software.³⁰ The reasons for the decision to focus on copyright protection, rather than alternatives such as patent or trade secret protection, are unclear and some scholars have anecdotally attributed the position to political and rhetorical posturing rather than substantive reasoning.³¹ Rather than clarify how the unique nature of software would fit in the rigid regime of pre-existing copyright law, the legislation by Congress reflected CONTU’s recommendation that copyrightable portions of software

²⁵ *Id.* at 903 n.165 (citing NAT’L COMM’N ON THE NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 70 (1980) (Hersey, Comm’r, dissenting)).

²⁶ *See* Frischmann, *supra* note 24, at 903-05.

²⁷ *See* Anne W. Branscomb, Senior Scholar in Residence, University of Pennsylvania, Panelist Remarks at From Conduit to Content: The Emergence of Information Policy and Law Symposium (Mar. 3, 1995), in *Panel One: Information Issues: Intellectual Property, Privacy, Integrity, Interoperability, and the Economics of Information*, 48 FED. COMM. L.J. 5, 16 (1995).

²⁸ *Id.*

²⁹ *See* 17 U.S.C. § 117 (2006).

³⁰ *Computer Assocs. Int’l v. Altai*, 982 F.2d 693, 703-04 (2d Cir. 1992).

³¹ *See* Branscomb, *supra* note 27, at 16-17.

be limited “to the extent that they embody an author’s original creation.”³² Congress effectively left the resolution of these issues to the judicial branch, which, over the past three decades, has been hesitant to take sweeping action out of fear of unjustly expanding the powers of copyright holders or upsetting the balance of the competing interests of the public and private spheres.³³

B. Emergence of Software Use Licenses

In order to reduce legal uncertainties and provide a firmer basis for proving copyright violations, many software developers and distribution companies include user agreements with their copyrighted products. These agreements are often referred to as shrink-wrap, browse-wrap, or click-wrap agreements, depending on at which point in the purchasing or installation process the user encounters the licensing agreement.³⁴ The agreements may also be referred to as End User License Agreements (“EULAs”), a more encompassing term. Drafters originally wrote these agreements with the intention that they would be used by specific customers.³⁵ However, as the market for software evolved, the function of the agreements changed, and EULAs now explicitly state the terms by which any individual can use and make copies of the protected software.³⁶ One user license may affect thousands of software users, regardless of the particular circumstances of their uses.³⁷ Some distributors have drafted these agreements with the intent to circumvent federal and state laws limiting copyright protection to retain as many rights as possible.³⁸

Some of these licenses have been challenged on the basis that the Copyright Act preempts them as legal contracts.³⁹ Section 301 of the Copyright Act states:

all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright . . . are governed exclusively by this title . . . no person is entitled to any such right or equivalent right in any such work under the common law or statutes of any State.⁴⁰

³² See Frischmann, *supra* note 24, at 904 (citing NAT’L COMM’N ON THE NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1980)).

³³ Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 429-31 (1984).

³⁴ Robert A. Hillman & Jeffrey J. Rachlinski, *Standard-Form Contracting in the Electronic Age*, 77 N.Y.U. L. REV. 429, 464-65 (2002).

³⁵ Matthew D. Stein, Comment, *Rethinking UCITA: Lessons From The Open Source Movement*, 58 ME. L. REV. 157, 171 (2006).

³⁶ *Id.*

³⁷ *Id.*

³⁸ Step-Saver Data Sys., Inc. v. Wyse Tech., 939 F.2d 91, 96 n.7 (3d Cir. 1991).

³⁹ See, e.g., ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1453 (7th Cir. 1996).

⁴⁰ 17 U.S.C. § 301(a) (2006).

But, in 1996, the Seventh Circuit held that “whether a particular license is generous or restrictive, a simple two-party contract is not ‘equivalent to any of the exclusive rights within the general scope of copyright’ and therefore may be enforced.”⁴¹ Federal and state courts across the country have generally adopted this legal principle as settled law.⁴²

C. Development of Open Source Licensing

In the midst of the booming development of the commercial software industry and its licensing policies in the late 1980s and 1990s, an alternative method of software production and distribution known as open source emerged.⁴³ As the commercialization of the software industry increased and intellectual property restrictions began to limit access to source code in the 1980s, interest grew in some circles to make access to software free in order to increase creative liberties, cooperation, and mutually beneficial competition.⁴⁴ Originally, the audience for open-source programs consisted mostly of software developers and website administrators.⁴⁵ However, in 1991, the operating system popularly known as Linux was first released to the public, expanding the appeal of open-source programming to anyone with a personal computer.⁴⁶ Today, some estimates hold that over one hundred million open source programs are registered with Creative Commons, serving functions ranging from server network programming to Internet browsers to word processors.⁴⁷

The emergence of legitimate open source licensing fueled the growing popularity of open source programming. Since the beginning of the free software movement, the concept of free addressed the issue of availability, not price.⁴⁸ The adoption of the term “open source” rather than “free software”

⁴¹ *ProCD*, 86 F.3d at 1455.

⁴² *See, e.g.*, *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1324-25 (Fed. Cir. 2003); *Lipscher v. LRP Publ’ns., Inc.*, 266 F.3d 1305, 1318-19 (11th Cir. 2001); *Xpel Techs. Corp. v. Am. Filter Film Distribs.*, No. SA-08-CA-175-XR, 2008 U.S. Dist. LEXIS 60893, at *21-23 (W.D. Tex. Aug. 11, 2008); *JZK, Inc. v. Weaver*, No. C06-5477 FDB, 2006 U.S. Dist. LEXIS 75523, at *9-10 (W.D. Wash. Oct. 17, 2006); *Salco Distribs. LLC v. Icode, Inc.*, No. 8:05-CV-642-T-27TGW, 2006 U.S. Dist. LEXIS 9483, at *7-8 (M.D. Fla. Feb. 21, 2006); *Green v. Hendrickson Publishers, Inc.*, 770 N.E.2d 784, 789-90 (Ind. 2002).

⁴³ Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 BERKELEY TECH. L.J. 443, 445-50 (2005).

⁴⁴ David S. Evans & Anne Layne-Farrar, *Software Patents and Open Source: The Battle Over Intellectual Property Rights*, 9 VA. J.L. & TECH. 10, at para. 4-5 (2004).

⁴⁵ *Id.* at para. 6.

⁴⁶ *Id.*

⁴⁷ *Jacobsen v. Katzer*, 535 F.3d 1373, 1378 (Fed. Cir. 2008).

⁴⁸ John Tsai, Note, *For Better or Worse: Introducing the GNU General Public License*

shifted the emphasis away from issues of cost and towards collaboration and development concepts in an attempt to encourage commercial applications.⁴⁹ Community standards dictate that open source programming fulfill certain requirements ranging from guaranteed distribution of the source code to prohibitions against discriminatory behavior.⁵⁰ Open source licensing strikes a balance between ensuring public access to software programs while keeping the benefits of open source collaborations, including faster debugging and compiling, as well as increased compatibility, that make it such a successful method of programming.⁵¹

From the early stages of the open source movement, its pioneers recognized the ability of copyright licensing to protect the freedoms of the software while ensuring that future users who modified or distributed the programming would be bound by the terms of the license.⁵² Part of the strength of the General Public License (“GPL”), the predominant and most commonly used open source license, is its “copyleft” structure – it mandates that future works derived from or using the source code of GPL-licensed software include the GPL when re-distributed.⁵³ The third and most recent version of the GPL was released in June 2007, sixteen years after the last major revision to the license.⁵⁴ While the GPL is the most common open source license, there are a number of popular variations as well as specific licenses for more popular programs, including the Apache HTTP Server and Mozilla Firefox.⁵⁵

Initial attempts to enforce open source licensing in court were ineffective despite the growing popularity of the GPL.⁵⁶ Many aspects of the GPL were ambiguous and led to debates amongst lawyers and academics.⁵⁷ No public record exists of any case that, prior to *Jacobsen*, directly interpreted an open source license as part of an intellectual property or copyright enforcement case.⁵⁸ Instead, open source developers used non-legal enforcement mechanisms to protect their products, usually agreeing to settlements out of

Version 3, 23 BERKELEY TECH. L.J. 547, 549 (2008).

⁴⁹ *Id.* at 551-52.

⁵⁰ OPEN SOURCE INITIATIVE, THE OPEN SOURCE DEFINITION (ANNOTATED), available at <http://www.opensource.org/docs/definition.php> (last visited Mar. 1, 2010).

⁵¹ See Carver, *supra* note 43, at 447-50.

⁵² See Tsai, *supra* note 48, at 550.

⁵³ See Carver, *supra* note 43, at 455-56.

⁵⁴ See Tsai, *supra* note 48, at 547.

⁵⁵ *Id.* at 552.

⁵⁶ See Carver, *supra* note 43, at 464-81.

⁵⁷ See Tsai, *supra* note 48, at 553.

⁵⁸ See Stein, *supra* note 35, at 192 (citing Dennis M. Kennedy, *A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture*, 20 ST. LOUIS U. PUB. L. REV. 345, 368 (2001)).

2010]

OPEN SOURCE SOFTWARE

court.⁵⁹ This arrangement still left both original creators and infringing programmers operating in a business environment of relative uncertainty.⁶⁰ The primary issue determining the enforceability of open source licenses, including the GPL, is whether the agreements are true licensing agreements (bilateral contracts) or bare licenses (unilateral granting of rights in which the user has no obligations to the software producer) that effectively yield copyright protection.⁶¹

III. *JACOBSEN V. KATZER*

In 2000, Robert Jacobsen and a group of associates formed the Java Model Railroad Interface Project (“JMRI”), an endeavor intended to produce software for model train hobbyists.⁶² This software programs the controlling hardware for the model trains and is known as a set of “decoder definition files.”⁶³ The software products were formally copyrighted and distributed under the Artistic License, an open source license similar but distinct from the GPL.⁶⁴ By 2005, Matthew Katzer, a commercial software maker, was distributing a patented program called Decoder Commander.⁶⁵ The Decoder Commander program contained code definitions identical to those found in JMRI’s project without fulfilling any requirements of the Artistic License, most notably a lack of attribution for any of JMRI’s work.⁶⁶ When Katzer filed patent infringement charges against Jacobsen, Jacobsen counter-filed against Katzer in the Northern District of California in 2006 to quiet Katzer’s patent claims and claim damages for breach of the Artistic License.⁶⁷

A. *District Court Proceedings*

Jacobsen quickly moved for a preliminary injunction against Katzer to stop him from further infringing the copyrighted work.⁶⁸ The preliminary injunction is traditionally a particularly potent weapon for software developers as it helps keep control over the copyright intact, encourages the private generation of licensing fees, and makes both infringement and redistribution

⁵⁹ See Stein, *supra* note 35, at 192.

⁶⁰ *Id.*

⁶¹ *Id.* at 193.

⁶² Amended Complaint at 1-2, Jacobsen v. Katzer, No. C-06-1905-JSW, 2007 WL 2358628 (N.D. Cal. Aug. 17, 2007).

⁶³ *Id.* at 7.

⁶⁴ *Id.* at 2-3.

⁶⁵ *Id.* at 15.

⁶⁶ *Id.*

⁶⁷ *Id.* at 17, 24-26.

⁶⁸ Jacobsen v. Katzer, No. C-06-01905-JSW, 2007 WL 2358628, at *5 (N.D. Cal. Aug. 17, 2007).

less economically viable for those looking to pirate software.⁶⁹ A plaintiff is only entitled to a preliminary judgment once it demonstrates:

- (1) that it has suffered an irreparable injury; (2) that remedies available at law, such as monetary damages, are inadequate to compensate for that injury; (3) that, considering the balance of hardships between the plaintiff and defendant, a remedy in equity is warranted; and (4) that the public interest would not be disserved by a permanent injunction.⁷⁰

At the time that Jacobsen filed his claim, a plaintiff who demonstrated a likelihood of success on the merits of a copyright claim was traditionally entitled to a presumption of irreparable harm.⁷¹

After briefly addressing the facts of the case, the District Court acknowledged that Katzer's actions would typically "constitute infringement of plaintiff's copyright and exclusive rights under the Copyright Act."⁷² But, in its assessment of the Artistic License, the District Court recognized that the license does not require copying parties to do anything beyond give the original author attribution and to place disclaimers about the copied content in the source form.⁷³ Rather than recognize the Artistic License as a user agreement or click-wrap agreement, the District Court determined that the Artistic License was a nonexclusive bare license to use, distribute, and copy the files.⁷⁴ The scope of the bare license is intentionally left broad in favor of the copying party and, accordingly, the District Court declined to find any grounds for a copyright infringement.⁷⁵ Jacobsen's remedies were limited to those provided under contract law, making an injunction a very unlikely remedy given the circumstances of the case and general guiding principles of contract law.⁷⁶

B. Federal Circuit Proceedings

Jacobsen appealed the denial of a preliminary injunction and the Federal Circuit asserted jurisdiction due to the patent claims in the initial case.⁷⁷ The Circuit Court unanimously vacated the decision of the Northern District of

⁶⁹ *eBay Inc. v. MercExchange, L.L.C.*, 547 U.S. 388, 395-97 (2006) (Kennedy, J., concurring).

⁷⁰ *Id.* at 391.

⁷¹ *Jacobsen*, 2007 WL 2358628, at *5.

⁷² *Id.* at *6.

⁷³ *Id.*

⁷⁴ *Id.*

⁷⁵ *Id.* at *7.

⁷⁶ *Id.*

⁷⁷ *Jacobsen v. Katzer*, 535 F.3d 1373, 1377 (Fed. Cir. 2008).

California and remanded the case for further action.⁷⁸ Like the District Court, the Federal Circuit acknowledged that Jacobsen's copyright was valid and he had made out a prima facie case of copyright infringement so long as Katzer's use of the material was outside of the scope of Jacobsen's license.⁷⁹ However, in a tone-setting manner and in a departure from the district court, the Federal Circuit announced its intention to treat open source licenses in a similar manner to traditional, for-profit user agreements:

The lack of money changing hands in open source licensing should not be presumed to mean that there is no economic consideration There are substantial benefits, including economic benefits, to the creation and distribution of copyrighted works under public licenses that range far beyond traditional license royalties.⁸⁰

The recognition of Jacobsen's indirect economic interests is particularly important because Katzer's rebuttal argument relied on the assertion that since the Artistic License reserved no economic rights, and non-economic rights are not a valid cause of action for copyright law, a breach of the Artistic License could not lead to a copyright infringement action.⁸¹

The Federal Circuit then addressed the central argument on appeal – whether the Artistic License acts as a nonexclusive license with contractual covenants or a binding contract with conditional use of a copyright between the user and copyright holder.⁸² Here, the Federal Circuit again found in favor of Jacobsen. “Copyright holders who engage in open source licensing have the right to control the modification and distribution of copyrighted material.”⁸³ If “a license is limited in scope and the licensee acts outside the scope, the licensor can bring an action for copyright infringement.”⁸⁴ The Federal Circuit found ample evidence in the terms of the Artistic License that its explicit intent is to create binding conditions on the user if he or she chooses to copy, modify or distribute the software program.⁸⁵ Most notably, the Federal Circuit identified the use of the phrase “provided that” where the Artistic License grants rights to copy the program – a term of art that, under California contract law, usually signals a binding condition.⁸⁶ Therefore, if Katzer had copied and

⁷⁸ *Id.* at 1376.

⁷⁹ *Id.* at 1379.

⁸⁰ *Id.*

⁸¹ *Id.* at 1380-81.

⁸² *Id.* at 1380.

⁸³ *Id.* at 1381.

⁸⁴ *Id.* at 1380.

⁸⁵ *Id.* at 1381.

⁸⁶ *Id.* *But see* CFM Commc'ns, LLC v. Mitts Telecasting Co., No. CVF046111RECDLB, 2005 WL 2089836, at *5 (E.D. Cal. 2005) (citing Diepenbrock v. Luiz, 115 P. 743, 743-44 (Cal. 1911)) (the term “provided” is an indicator of a condition but

translated Jacobsen's code without conforming to the conditions of the Artistic License, Katzer was acting outside of the scope of the license and a copyright infringement action was not preempted.⁸⁷

C. New Issues on Remand

On remand, the District Court again addressed the issue of whether Jacobsen was entitled to a preliminary injunction against Katzer's copying of the open source licensed product.⁸⁸ Again, the District Court rejected Jacobsen's motion for injunctive relief.⁸⁹ Shortly after the Federal Circuit addressed the substantive factual matters governing Jacobsen's copyright violation claim, the Supreme Court issued a decision that changed the procedural requirements for a preliminary injunction.⁹⁰ Previously, a plaintiff who demonstrated a likelihood of success on the merits of a copyright claim was entitled to a presumption of irreparable harm.⁹¹ However, under the new standard the Supreme Court articulated in *Winter v. Natural Resources Defense Council*, a plaintiff is required to independently demonstrate that irreparable injury is "likely in the absence of an injunction" in order to obtain equitable relief.⁹² Since Jacobsen did not proffer evidence to meet this burden, and there was no evidence in the record to support more than the potential of irreparable injury, the district court rejected the preliminary injunction motion and granted Jacobsen the opportunity to submit an amended complaint.⁹³

The January 2009 *Jacobsen* decision has very interesting ramifications for the future of equitable relief and the procedural rights of all copyright holders.⁹⁴ However, the 2009 portion of the *Jacobsen* proceedings focused on the procedural requirements of preliminary injunction law, not the merits of the arguments regarding the enforceability of the license agreements.⁹⁵ Despite the recent procedural developments of *Jacobsen*, the Federal Circuit's

does not necessarily impose a condition and that the intent of the contracting parties must be interpreted from the words and purposes established throughout the entire document).

⁸⁷ See *Jacobsen*, 535 F.3d at 1382.

⁸⁸ *Jacobsen v. Katzer*, 609 F. Supp. 2d 925, 935-38 (N.D. Cal. 2009).

⁸⁹ *Id.* at 938.

⁹⁰ *Winter v. Natural Res. Def. Council*, 129 S. Ct. 365, 374 (2008); see also *Jacobsen*, 609 F. Supp. 2d at 937.

⁹¹ *Jacobsen v. Katzer*, No. C 06-01905 JSW, 2007 WL 2358628, at *5 (N.D. Cal. Aug. 17, 2007).

⁹² *Jacobsen*, 609 F. Supp. 2d at 936 (citing *Winter*, 129 S. Ct. at 375).

⁹³ *Jacobsen*, 609 F. Supp. 2d at 937-38.

⁹⁴ See Benjamin I. Narodick, Legal Update, *Winter v. Natural Resources Defense Council: Going Into the Belly of the Whale of Preliminary Injunctions and Environmental Law*, 15 B.U. J. SCI. & TECH. L. 332, 343 (2009).

⁹⁵ *Jacobsen*, 609 F. Supp. 2d at 935-36 (citing *Winter*, 129 S. Ct. at 375).

discussions on the merits of the 2007 and 2008 *Jacobsen* proceedings still have independent value as legal precedent for future copyright cases.⁹⁶ Regardless, such future cases may be complicated by the new procedural issues introduced in *Winter* and *Jacobsen* which deserve further independent study.

D. *Jacobsen, Comparable Precedent, and Future Effects*

1. Equal Treatment for Open Source Software Distributors

Several changes of judicial policy and precedent are identifiable in *Jacobsen*. First, *Jacobsen* signifies a victory for the equal treatment for open-source and for-profit software programming. The decision answers two primary questions about the GPL which have plagued the license since its inaugural release – whether open source licenses are legally enforceable and whether the same licenses receive equal treatment with commercial software.

Jacobsen settles, in part, an uncertainty regarding the general enforceability of open source licenses including the GPL, which has more stringent requirements for use and distribution than the Artistic License.⁹⁷ The specific reliance of the Federal Circuit on the terms of art and conditions as written in the Artistic License suggests a concession that open source licenses may be properly structured and enforced.⁹⁸ The *Jacobsen* decision is consistent with other recent software licensing cases, to be discussed *infra*, in which courts have upheld the ability of the copyright holder to dictate conditions on use and copying of programs that are not based in the provisions of the Copyright Act.⁹⁹ Ultimately, with *Jacobsen* on the books as the first recorded instance of judicial interpretation of an open source licensing agreement, it may immediately serve as a legal guide for the drafting and enforcement of any future open source software license.¹⁰⁰

The Federal Circuit decision in *Jacobsen* also sets a standard that, regardless of the typical economic system in which open source programmers operate, the Copyright Act offers standard copyright protection to software licensed using open source and copy-left principles. The Federal Circuit justified this position by identifying the potential economic value of the benefits that can be reaped from the spread of open source software. These benefits include an

⁹⁶ See *Jacobsen*, 609 F. Supp. 2d at 937 (“The Federal Circuit court’s list of potential harms that a copyright holder may face in the open source field are just that – *potential* harms. There is no showing on the record before this Court that *Jacobsen* has actually suffered *any* of these potential harms.” (emphasis in original)).

⁹⁷ See Stein, *supra* note 35, at 192.

⁹⁸ See *Jacobsen v. Katzer*, 535 F.3d 1373, 1381 (Fed. Cir. 2008).

⁹⁹ *Davidson & Assocs. v. Jung*, 422 F.3d 630, 638-39 (8th Cir. 2005); *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1324-25 (Fed. Cir. 2003).

¹⁰⁰ See Stein, *supra* note 35, at 192.

increased market share, reduced programming costs, and future programming projects due to an increase in reputation and public exposure.¹⁰¹ That the requirements for licensing are non-monetary does not contradict the essential nature of copyright protection – the right to exclude.¹⁰² This is not a large leap of intellectual faith for the Federal Circuit, which does not hesitate to, almost in passing, “recogniz[e] the economic motives inherent in public licenses, even where profit is not immediate.”¹⁰³ Additionally, the Federal Circuit says little about the potential for open source software to speed up the debugging, compilation, testing, and other programming phases – a commercial potential already realized by many mainstream technology companies including Sun Microsystems, Hewlett Packard, Dell Computers, and IBM.¹⁰⁴

2. Recognition of Intellectual Property Rights Outside of the Copyright Act

The Federal Circuit decision in *Jacobsen*, while granting a long-awaited recognition of the rights of open-source software distributors, also includes an important and overlooked expansion of the legal idea that licensing agreements can confer intellectual property rights upon the copyright holder that might not be supported by the Copyright Act. This position has been present in the federal judiciary since the *ProCD* decision in 1997.¹⁰⁵ Before *ProCD*, courts had not conclusively determined whether contractual obligations could be formed using shrink-wrap licenses, let alone click-wrap licenses, or which particular conditions would be unenforceable.¹⁰⁶ Soon after that decision, the shrink-wrap license format was widely recognized as part of the vendor’s invitation of buyer acceptance by limited means of performance.¹⁰⁷ Part of the basis for this rationale was that the buyer could not use the software program without indicating acceptance of the license.¹⁰⁸ Courts soon recognized the validity of click-wrap licensing agreements as well, comparing them to shrink-wrap agreements which work under approximately the same legal principles.¹⁰⁹

Just as breaking the shrinkwrap seal and using the enclosed computer

¹⁰¹ See *Jacobsen*, 535 F.3d at 1379.

¹⁰² See *id.* at 1381.

¹⁰³ *Id.* at 1379 (citing *Planetary Motion, Inc. v. Techsplosion, Inc.*, 261 F.3d 1188, 1200 (11th Cir. 2001)).

¹⁰⁴ See Stein, *supra* note 35, at 183-87.

¹⁰⁵ See Maureen A. O’Rourke, *Copyright Preemption After The ProCD Case: A Market-Based Approach*, 12 BERKELEY TECH. L.J. 53, 71-90 (1997).

¹⁰⁶ *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1452 (7th Cir. 1996); O’Rourke, *supra* note 105, at 58 n.23.

¹⁰⁷ *Specht v. Netscape Commc’ns Corp.*, 150 F. Supp. 2d 585, 592 (S.D.N.Y. 2001).

¹⁰⁸ *Id.*

¹⁰⁹ *Id.* at 593-94.

program after encountering notice of the existence of governing license terms has been deemed by some courts to constitute assent to those terms in the context of tangible software, so clicking on a webpage's clickwrap button after receiving notice of the existence of license terms has been held by some courts to manifest an Internet user's assent to terms governing the use of downloadable intangible software.¹¹⁰

ProCD also sets an important standard by defining what behavior a click-wrap agreement can cover. "[L]icenses are enforceable unless their terms are objectionable on grounds applicable to contracts in general (for example, if they violate a rule of positive law, or if they are unconscionable)."¹¹¹ In *ProCD*, the Court did not discuss the specific terms of the license, and the Seventh Circuit did not elaborate further on the standard that courts would use to determine what license terms are objectionable.¹¹² Since then, amendments by Congress to the Federal Copyright Act have more clearly established the standard:

The Copyright Act preempts state laws that attempt to protect rights exclusively protected by federal law. Conversely, the Copyright Act does not preempt state law from enforcing non-equivalent legal or equitable rights. A state cause of action is statutorily or expressly preempted if: (1) the work at issue is within the subject matter of copyright as defined in §§ 102 and 103 of the Copyright Act, and (2) the state-law-created right is equivalent to any of the exclusive rights within the general scope of copyright as specified in § 106.¹¹³

The relevant section of the Copyright Act, § 301(a), prevents states from giving special protection to works of authorship that Congress has decided should be in the public domain.¹¹⁴ The preemption requirements of the Copyright Act leave it as "the exclusive source of protection for 'all legal and equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106' of the Copyright Act."¹¹⁵ However, *ProCD* disputed that contract rights, enforced primarily by common and state law, are equivalent to those rights found in the Copyright Act.

Rights "equivalent to any of the exclusive rights within the general scope

¹¹⁰ *Specht v. Netscape Commc'ns Corp.*, 306 F.3d 17, 22 n.4 (2d Cir. 2002) (citations omitted).

¹¹¹ *ProCD*, 86 F.3d at 1449.

¹¹² *See id.*

¹¹³ *Davidson & Assocs. v. Jung*, 422 F.3d 630, 638 (8th Cir. 2005) (citations omitted).

¹¹⁴ *ProCD*, 86 F.3d at 1453.

¹¹⁵ *Nat'l Car Rental Sys., Inc. v. Computer Assocs. Int'l.*, 991 F.2d 426, 428 (8th Cir. 1993).

of copyright” are rights established by law – rights that restrict the options of persons who are strangers to the author. Copyright law forbids duplication, public performance, and so on, unless the person wishing to copy or perform the work gets permission; silence means a ban on copying. A copyright is a right against the world. Contracts, by contrast, generally affect only their parties; strangers may do as they please, so contracts do not create “exclusive rights.”¹¹⁶

Under this doctrine, federal courts have recognized the legally binding nature of shrink-wrap and click-wrap licenses, even if those licenses impose more restrictive use and copy conditions than those already found in the Copyright Act.¹¹⁷

ProCD has been gradually accepted by the federal judiciary up through and including the decision in *Jacobsen*.¹¹⁸ However, some critics have challenged the underpinnings of the *ProCD* decision. One notable critique targets the Seventh Circuit’s perception of the nature of the process that forms the contractual conditions found in the use licenses.¹¹⁹ The court in *ProCD* distinguished copyright as “a right against the world . . .” and recognized that contracts, which only affect the contracting parties, fail to create exclusive rights which would make them equivalent to those found in the Copyright Act.¹²⁰ Competing companies could therefore choose to offer different contractual terms, and market forces would then encourage differing levels of rights and a variety of cost/benefit options for consumers.¹²¹ “Terms of use are no less a part of ‘the product’ than are the size of the database and the speed with which the software compiles listings. Competition among vendors, not judicial revision of a package’s contents, is how consumers are protected in a market economy.”¹²²

But the Seventh Circuit also recognized in *ProCD* that the user license under review was included with all copies of the product and is a standard of the software industry.¹²³ The license then serves as a form contract, applying to all individuals who wish to purchase and use the software, and conferring exclusive rights to the producer of the program.¹²⁴ When the form contract

¹¹⁶ *ProCD*, 86 F.3d at 1454 (emphasis in original).

¹¹⁷ See *Davidson*, 422 F.3d at 637-39; *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1324-25 (Fed. Cir. 2003).

¹¹⁸ *Bowers*, 320 F.3d at 1324-25.

¹¹⁹ Robert P. Merges, *Intellectual Property and the Costs of Commercial Exchange: A Review Essay*, 93 MICH. L. REV. 1570, 1611-12 (1995).

¹²⁰ *ProCD*, 86 F.3d at 1454.

¹²¹ *Id.* at 1453.

¹²² *Id.*

¹²³ *Id.* at 1450-52.

¹²⁴ See Merges, *supra* note 119, at 1609-13.

becomes standardized throughout the industry, this creates an effect referred to as private legislation – transactions are normalized on a universal basis without public oversight and with only the industry’s interests being served.¹²⁵ Indiscriminate judicial enforcement of EULAs encourages a system where large firms benefit from an imbalance of negotiating power and allows for the employment of discriminatory customer practices under the guise of freedom of contract.¹²⁶ Because this private legislation regime has “the same generality of scope as the state legislation that is often the target of federal preemption . . . [and has] the same effect as offending state legislation: wholesale subversion of an important federal policy,” pervasive licensing provisions that are contrary to federal copyright law should be preempted as a creation of equivalent rights.¹²⁷

Additionally, judicial enforcement of end user license agreements may give de facto copyright protection to non-copyrightable material.¹²⁸ In *ProCD*, the end user agreement prohibited the user from sharing the telephone listings that were part of the purchased software.¹²⁹ The informational aspects of telephone listings – as opposed to arrangement and organization – are considered non-protectable matter by copyright due to a lack of originality, which is a key requirement for any protectable subject matter.¹³⁰ However, dissemination of this type of useful information is strictly prohibited, creating a virtual copyright on the material, controlling access to the listings, and using that restricted access as consideration for financial gain.¹³¹ By disassociating the protective functions of copyright law from the incentives designed to promote the creation of original material, legally recognized contract provisions circumvent the purposes of copyright provisions and may defeat the public policy reasoning for copyright protection of software.¹³²

Despite these and other concerns, judicial criticism of *ProCD* and its progeny has often been limited and overlooked, and this group of contract cases has provided the judicial principals that courts have reaffirmed over the past decade as the controlling precedent for *Jacobsen*.¹³³ Additionally, click-wrap end user agreements, due to freedoms in the format of presentation, have

¹²⁵ *Id.*

¹²⁶ *Id.*

¹²⁷ *Id.* at 1613.

¹²⁸ Niva Elkin-Koren, *Copyright Policy and the Limits of Freedom of Contract*, 12 BERKELEY TECH. L.J. 93, 106-08 (1997).

¹²⁹ *Id.* at 107.

¹³⁰ *Id.*

¹³¹ *Id.* at 107-08.

¹³² *Id.* at 109-11.

¹³³ See *Davidson & Assocs. v. Jung*, 422 F.3d 630, 639 (8th Cir. 2005); *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1324-25 (Fed. Cir. 2003).

a greater record of successful judicial recognition than shrink-wrap licenses.¹³⁴ The decision does not represent a fundamental shift in judicial policy, but the Federal Circuit's rationale in *Jacobsen* may justly concern the very open source software engineers who want to produce more programs in the future.¹³⁵ This expansion of intellectual property rights effectively stacks the deck in favor of any software producer already in the market. The court-recognized ability of using click-wrap licensing to privately strengthen copyright protection has the potential to have its greatest effect on a tool of software protection used by commercial and open source developers alike: reverse engineering.

IV. STATE OF REVERSE ENGINEERING

Reverse engineering is the “general process of analyzing a technology specifically to ascertain how it was designed or how it operates.”¹³⁶ For software programs, this frequently involves translating a copy of the object code, which operates in binary, into human-readable source code as part of processes referred to as disassembly and decompilation.¹³⁷ These translations and reproductions are explicit copies for the purposes of copyright law.¹³⁸ Both processes can be accomplished by using a variety of publicly available software programs, and different methods and tools for these processes are widely used by the computer industry in general.¹³⁹ Reverse engineering offers an important, and sometimes the only, way by which programmers can learn how to design new programs that will be compatible with different operating systems and processes.¹⁴⁰ Reverse engineering is also important for allowing software engineers to investigate existing programs and find ways to make those programs more efficient, useful, and effective – an important aspect of the competition in the software industry.¹⁴¹ It provides an effective tool in determining whether another program has copied previously existing work.¹⁴² Many popular open source programs, including Linux, are the products of reverse engineered commercial software.¹⁴³ While this is not an

¹³⁴ See LEMLEY ET AL., *supra* note 18, at 881.

¹³⁵ See Clements, *supra* note 8.

¹³⁶ *FAQ about Reverse Engineering—Chilling Effects Clearinghouse*, SAMUELSON LAW, TECHNOLOGY, AND PUBLIC POLICY CLINIC, <http://www.chillingeffects.org/reverse/faq.cgi> (last visited Mar. 6, 2010).

¹³⁷ *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1514 (9th Cir. 1992).

¹³⁸ *Id.* at 1518.

¹³⁹ *Id.* at 1514 n.2.

¹⁴⁰ *Id.* at 1525-26.

¹⁴¹ See SAMUELSON LAW, TECHNOLOGY, AND PUBLIC POLICY CLINIC, *supra* note 136.

¹⁴² *Id.*

¹⁴³ See Evans, *supra* note 44, at para. 6.

exhaustive list of uses for reverse engineering of software, it does indicate a number of common practices essential to the open source software community that could be endangered if other courts blindly recognize modern software end user licenses in their entirety.

A. *Pre-Jacobsen Law of Reverse Engineering*

In the two decades before the Federal Circuit decision in *Jacobsen*, the legal treatment of the practice of reverse engineering had undergone as much of a legal evolution as that regarding open source software production in general.¹⁴⁴ Many of the complications regarding the practice, however, also stem from the decision in *ProCD*, which further complicated judicial treatment of copyrighted materials through interpretations of contract law. Shrinkwrap licenses, clickwrap licenses, and other EULAs could now unpredictably redefine the rights of a user of copyrighted material.¹⁴⁵ As a result, most commercial software licensing agreements now feature language limiting or prohibiting users from reverse engineering the product.¹⁴⁶ In federal jurisdictions where state laws prohibiting reverse engineering were not upheld, courts recognized the right of software distributors to sell products with their own expansive licensing terms and use state contract laws as tools for the enforcement mechanism.¹⁴⁷ In 2003, the Federal Circuit, ruling in *Bowers v. Baystate Technologies*, reaffirmed the precedent from *ProCD* and noted that a contract formed by mutual consent and consideration is distinguishable from the precedent of general copyright infringement cases and there is no express preemption of equivalent rights.¹⁴⁸

Notably, the *Bowers* court recognized that the formation of the subject software license was privately negotiated by the two parties, unlike many mass-consumer software licenses, constituting an extra element that separated *Bowers* from other cases protecting reverse engineering as fair use.¹⁴⁹ The Eighth Circuit, however, directly applied this precedent from the *Bowers* decision to a reverse engineering prohibition in a mass-consumer software license without distinguishing between individualized and commercial licensing agreements.¹⁵⁰ In 2005, the Eighth Circuit ruled in *Davidson & Associates v. Jung* that merely “[b]y signing the TOUs and EULAs, Appellants

¹⁴⁴ *Sega*, 977 F.2d at 1520-21.

¹⁴⁵ See *Merges*, *supra* note 119, at 1609-13.

¹⁴⁶ Annalee Newitz, *Dangerous Terms: A User's Guide to EULAs*, ELECTRONIC FRONTIER FOUNDATION (Feb. 2005), <http://www.eff.org/wp/dangerous-terms-users-guide-eulas> (last visited Mar. 6, 2010).

¹⁴⁷ *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1324-26 (Fed. Cir. 2003).

¹⁴⁸ *Id.*

¹⁴⁹ *Id.* at 1325-26.

¹⁵⁰ *Davidson & Assocs. v. Jung*, 422 F.3d 630, 639 (8th Cir. 2005).

expressly relinquished their rights to reverse engineer.”¹⁵¹

There are two types of potential preemption when interpreting licensing agreements. *Jacobsen* and *Bowers* each deal with cases of express preemption, in which state claims of action would be preempted by the Supremacy Clause of the Constitution and § 301 of the Copyright Act, whereas many other cases, including *Davidson*, address the issue of conflict preemption.¹⁵² Conflict preemption occurs when the adjudicating court does not find any express preemption but finds that either “it is impossible for a party to comply with both the state and federal law” or “the state law stands as an obstacle to the accomplishment and execution of the full purposes and objectives of Congress.”¹⁵³ Currently, some software license terms fulfill the procedural requirements for this second test. The adverse software licensing terms which companies seek to enforce, including terms which control reverse engineering, are a nearly universal and unavoidable aspect of computer use.¹⁵⁴ The majority of traditional software users adhere to these terms, but many of them do not fully understand the rights they have signed away if they are even fully aware of the existence of the agreements themselves.¹⁵⁵

Predominant cases dealing with the conflicts of fair use and software licenses – including *Davidson*, which speaks broadly to conflict preemption cases – do not find that contractual provisions constitute state restrictions of federally given rights.¹⁵⁶ Regardless of its standing as a protected fair use, “[p]rivate parties are free to contractually forego the limited ability to reverse engineer a software product under the exemptions of the Copyright Act”¹⁵⁷ While *Jacobsen* does not address this type of preemption specifically, it does broaden the scope of potential licensing rights and the ability to control the modification and distribution of copyrighted material.¹⁵⁸ After over thirty years of congressional and state legislation, as well as decisions on all levels of both the federal and state court systems, the judicial status of reverse engineering for copyrighted software has apparently come full circle. Any stability protecting the functioning ability to reverse engineer has been fleeting – the state of the law is best defined as having returned to its pre-CONTU

¹⁵¹ *Id.*

¹⁵² *Id.* at 638.

¹⁵³ *Id.*

¹⁵⁴ Jennifer S. Bisk, *Software Licenses Through the Bankruptcy Looking Glass: Drafting Individually Negotiated Software Licenses That Protect the Client’s Interests in Bankruptcy*, 17 *FORDHAM INTELL. PROP. MEDIA & ENT. L.J.* 611, 613 (2007).

¹⁵⁵ See Newitz, *supra* note 146.

¹⁵⁶ *Davidson*, 422 F.3d at 637-39.

¹⁵⁷ *Id.* at 639 (quoting *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1325-26 (Fed. Cir. 2003)).

¹⁵⁸ *Jacobsen v. Katzer*, 535 F.3d 1373, 1382-83 (Fed. Cir. 2008).

status of effective full control by the commercial software industry. If the intent of CONTU was to clearly delineate the rights of both consumers and producers of software through federal law and mandate consistent protections of those rights, then that mission has been lost over the past thirty years. The work of the Commission has effectively become as outdated and obsolete as punch card computers.

B. Reverse Engineering and a Strict Reading of Jacobsen

Bowers and *Davidson* reflect a growing judicial trend giving greater levels of deference to the parties involved in negotiating and accepting contracts in the form of EULAs. However, a closer look at *Jacobsen* suggests that this doctrine is approaching a new benchmark. The Federal Circuit noted that the violated terms of the Artistic License in *Jacobsen* do not affect licensing royalties, differentiating the case from more traditional copyright license claims.¹⁵⁹ However, the court went further, recognizing the terms of public licenses even where economic motives and direct profit are not immediate.¹⁶⁰ This provides even more leniency for copyright-holders to write restrictive EULA terms than even the low bar set in *Bowers*.¹⁶¹

Jacobsen, while not addressing commercial EULAs, extends the precedent of *Bowers* and *Davidson* with seemingly even more relaxed restrictions for all license terms.¹⁶² It does so without explicitly distinguishing between open source licenses and traditional commercial EULAs.¹⁶³ Under a strict reading of *Jacobsen*, it does not appear that there are any substantial impediments to the licensing powers of commercial vendors as long as there is some link between the subject term of the license and a less-than-immediate economic motive. Reverse engineering still stands as an important way to make software cross-compatible and interoperable with other platforms. Allowing reverse engineering also facilitates competition and encourages the proliferation of more efficient and effective products in the software market.¹⁶⁴ But if new programs, reliant on the fruits of reverse engineering, are going to compete with the originally licensed products, then the economic motives for prohibiting reverse engineering are not just immediate but almost self-evident. It would be hard to envision a situation where prohibiting a direct competitor from entering an existing market would not be to a company's direct and immediate economic benefit.

¹⁵⁹ *Id.* at 1379.

¹⁶⁰ *Id.*

¹⁶¹ *Bowers*, 320 F.3d at 1325.

¹⁶² *Jacobsen*, 535 F.3d at 1381-82.

¹⁶³ *Id.*

¹⁶⁴ Elizabeth M. N. Morris, *Will Shrinkwrap Suffocate Fair Use?*, 23 SANTA CLARA COMPUTER & HIGH TECH. L.J. 237, 248-50 (2007).

C. Jacobsen, *Reverse Engineering and the DMCA*

The extent of the power of the *Jacobsen* decision and the recognized powers of licensing parties may not be limited to civil actions. The Digital Millennium Copyright Act, passed by Congress in 1998, institutes substantial criminal penalties for the circumvention of digital security measures that software developers designed to make copying and distributing copyrighted software more difficult.¹⁶⁵ The legislation was designed to help strengthen copyright law against the growing specter of internet piracy and to protect the growing commercial software industry.¹⁶⁶ However, the DMCA also included measures to protect the future use of software for development and research.¹⁶⁷ To that end, reverse engineering was granted specific exemptions from prosecution under the new federal legislation.¹⁶⁸ To claim the defense, appellants must show:

- (1) they lawfully obtained the right to use a copy of a computer program;
- (2) the information gathered as a result of the reverse engineering was not previously readily available to the person engaging in the circumvention;
- (3) the sole purpose of the reverse engineering was to identify and analyze those elements of the program that were necessary to achieve interoperability of an independently created computer program with other programs; and
- (4) the alleged circumvention did not constitute [copyright] infringement.¹⁶⁹

Recent court decisions leading up to and including *Jacobsen*, however, have trended towards treating the DMCA exemption for reverse engineering similar to the fair use exemption in other copyright actions.¹⁷⁰ These issues are most predominant in *Davidson*, where a defendant invoked the interoperability exemption and was promptly rejected in both the district and appellate courts.¹⁷¹ There, the court struck down affirmative defenses based on the first (lawful use of the program) and fourth (circumvention does not constitute infringement) requirements of the DMCA exemption.¹⁷² The Eighth Circuit based these decisions on the tenuous and sparsely discussed grounds that the reverse engineered server program used by the defendants allowed

¹⁶⁵ See Digital Millennium Copyright Act, 17 U.S.C. §§ 1201-1205, 1301-1332 (2006); Morris, *supra* note 164, at 254-57.

¹⁶⁶ Morris, *supra* note 164, at 254-55.

¹⁶⁷ 17 U.S.C. § 1201(d)-(j); *id.* at 255-57.

¹⁶⁸ 17 U.S.C. § 1201(f); Morris, *supra* note 164, at 257.

¹⁶⁹ Davidson & Assocs. v. Jung, 422 F.3d 630, 641-42 (8th Cir. 2005).

¹⁷⁰ *Id.*

¹⁷¹ *Id.*

¹⁷² *Id.*

unauthorized users of the software to use all of its protected features.¹⁷³ This assertion ignores the fact that the actual individuals responsible for the construction of the emulator program had presumably purchased and used the product legally.¹⁷⁴ The *Davidson* court also refrained from discussing any of the aspects of a potential contributory infringement claim.¹⁷⁵ Regardless, *Davidson* set a high standard, perhaps impossibly high, for what is expected for the interoperability exception to the DMCA. More importantly, it perpetuates a challenging legal environment for software engineers who reverse engineer software, diminishes the status of reverse engineering as a legally protected activity, and encourages licensing parties to include increasingly restrictive licensing terms with their products.

D. Comparative Effects of Jacobsen on Commercial and Open Source Software

With the expanded scope of EULAs and the growing restrictions on the right to reverse engineer copyrighted software, it is important to determine whether certain groups of software developers will be more affected by these shifts in legal policy. On a cursory glance, open source developers seem to receive the most direct legal impact. The Federal Circuit in *Jacobsen* is the first court to fully recognize an open source license as legally binding, protecting the rights of holders of the Artistic License if not the more reasonably-based General Public License.¹⁷⁶ *Jacobsen*, in many ways, validates the open source process as legitimate and may encourage a greater share of developers to distribute software in this manner. Additionally, many open source software developers may be fully dependant on reverse engineering for the development of competing products.¹⁷⁷ As such, restricted rights of use might hamper development and slow progress in the software market, or even create market forces which would encourage the formation of commercial software monopolies that would make open source development infeasible.¹⁷⁸

Despite the unfortunate impact of *Jacobsen* on open source developers, commercial software makers may suffer from a disparately stronger impact. First, strong restrictions on reverse engineering encourage a “race to the market” scenario, with commercial copyright holders given monumental market advantages by having their products in the market first with a legal guarantee that the features of those products cannot be analyzed and then used

¹⁷³ *Id.*

¹⁷⁴ *Id.*

¹⁷⁵ *Id.*

¹⁷⁶ *Jacobsen v. Katzer*, 535 F.3d 1373, 1381-82 (Fed. Cir. 2008).

¹⁷⁷ *See Morris*, *supra* note 164, at 249-252.

¹⁷⁸ *Id.* at 268-69.

against them in competing products.¹⁷⁹ This greatly affects the market incentives for the creation of new software programs and features, which may increase profit shares for developers. Alternatively, these changes may lead to more software in the public market with varying EULA terms to give developers a greater range of control over the distribution of their products – a benefit originally foreseen by the Seventh Circuit in *ProCD*.¹⁸⁰ Regardless of the precise impact, stronger protection for copyrights will increase the amount of copyright litigation. If the rights held by copyright holders become more lucrative, software developers may have greater incentives to file legal claims against competing products.¹⁸¹ Even a small amount of legal ambiguity about the merits of such claims could dramatically slow the software development process as commercial producers would need to try to find a clear legal position and follow a more evident course of action without liability. Under any number of these scenarios, commercial software producers could be the group most negatively affected by the assertion of the cases leading up to and including *Jacobsen* that EULA terms need only have some economic value, even indirectly, for the copyright holder to seek enforcement in court.

V. ANALYSIS OF LEGAL ALTERNATIVES

The varying course of American courts in dealing with the issue of reverse engineering and the strength of licensing agreements has been addressed from many angles inherent to the study of copyright law. This section will address some of the more complicated issues from the perspective of contract law and unconscionable contracts as well as the potential applicability of antitrust law prohibitions to EULA terms.

A. *EULAs, Reverse Engineering and Unconscionable Contracts*

1. The U.C.C. Test for Unconscionability

One of the points repeatedly mentioned by courts addressing copyright licensing issues is that licenses effectively operate as contracts, since they are private agreements negotiated by two parties with proper consideration and assent.¹⁸² Since the Seventh Circuit ruled on the merits of the process of assent in *ProCD*, EULAs have been effectively treated in the same legal framework as those originally envisioned under the Uniform Commercial Code (“U.C.C.”). One primary power of the judiciary in contract cases, found in the

¹⁷⁹ *Id.*

¹⁸⁰ *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1454-55 (7th Cir. 1996).

¹⁸¹ *See Newitz, supra* note 146.

¹⁸² *Davidson & Assocs. v. Jung*, 422 F.3d 630, 639 (8th Cir. 2005) (citing *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1325-26 (Fed. Cir. 2003)).

U.C.C. as well as many state codes, allows a court to find a contract or contract clause to be unconscionable at the time it was made. If the court makes such a ruling, it may refuse to enforce or limit the enforcement of that contract.¹⁸³

The widespread application of the rules of unconscionability to software user license cases may be problematic, as the U.C.C. clearly delineates an effort to prevent oppression but not disruption of risk allocation based on a relative bargaining position.¹⁸⁴ In commercial cases, rights of implied warranty imposed by law could not be negated by a contractual clause excluding all warranties for the seller.¹⁸⁵ The parallels between these legally imposed rights and the fair use provisions for reverse engineering are striking, and they both reflect public policy intents to help shape a fair and competitive market.

2. The ALI Test for Unconscionability

A legal position mandating that contracts not negate any rights imposed by federal law may be too broad a judicial rule. However, there are alternative rubrics for analyzing the unconscionability of EULA terms. The American Law Institute's ("ALI") Principles of the Law of Software Contracts, adopted in 2009, identify three key points of analysis that line up with the purposes of intellectual property law as established in the United States Constitution and serve as a strong base to start an analysis for overturning unconscionable software license terms:

- (i) whether federal law renders the restriction unenforceable because it upsets the intellectual property balance between exclusionary rights and creating a rich public domain; (ii) whether the transferee had sufficient notice and opportunity to read the term restricting rights; and (iii) whether the restriction runs afoul of public policy or unconscionability norms.¹⁸⁶

The first of these factors, the balance between the public interest and the private incentive system, is particularly important when dealing with reverse engineering due to the structure and incentives of copyright law.¹⁸⁷ The Constitution mandates that Congress's powers regarding intellectual property must be aimed at "promot[ing] the Progress of Science and useful Arts."¹⁸⁸ Somewhat counter-intuitively, though, the means provided to promote this public interest grant exclusive rights to the originators of that information and

¹⁸³ U.C.C. § 2-302(1) (2004).

¹⁸⁴ U.C.C. § 2-302 cmts. 1-2.

¹⁸⁵ *Bekkevold v. Potts*, 216 N.W. 790, 791-92 (Minn. 1927).

¹⁸⁶ PRINCIPLES OF THE LAW OF SOFTWARE CONTRACTS § 1.06 cmt. b (2009).

¹⁸⁷ *See Morris*, *supra* note 164, at 238.

¹⁸⁸ U.S. CONST. art. I, § 8, cl. 8.

reduce the flow of information to the public.¹⁸⁹ In a step to balance these interests, Congress and the courts have established fair use exceptions, among the most prevalent of which is the act of reverse engineering.¹⁹⁰ With the public interest in reverse engineering established by legislative action, the interest in promoting reverse engineering as an unconscionable software licensing term is strongly based in public policy concerns, regardless of the judicial hesitation to alter the powers of copyright holders without an explicit command by Congress.¹⁹¹

The second factor of the ALI test, a requirement for sufficient notice and opportunity of information, is fairly well-established in modern jurisprudence and was part of the heart of the dispute in *ProCD*.¹⁹² Sufficient notice rules are very similar to traditional contract rules in that they require a fair drafting of the agreement and the ability to understand the terms of the contract before finalizing assent.¹⁹³ The requirement of finalization of assent is an important distinction, primarily because many licensing agreements include a thirty-day period in which a customer can return the software for a full refund.¹⁹⁴ Publication and use of these licensing agreements has become so common that many companies have been embarrassed by the use of EULAs that have not been updated to reflect current market realities.¹⁹⁵ However, it is unlikely that this would provide a basis (or, for that matter, a preventing factor) for overturning a reverse engineering clause in a click-wrap agreement.

Finally, reverse engineering restrictions in EULAs may run afoul of public policy or unconscionability norms – the third required factor in the ALI test. The ALI, citing U.C.C. § 2-302, notes that most findings of unconscionability require both substantive (unfair surprise or oppression of one party) and procedural (defective formation process) elements.¹⁹⁶ However, given the nature of form click-wrap licenses and the degree to which reverse engineering licensing restrictions may be particularly offensive, procedural element

¹⁸⁹ *Id.*

¹⁹⁰ *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1520-21 (9th Cir. 1992); *see also Morris*, *supra* note 164, at 257.

¹⁹¹ *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 429 (1984).

¹⁹² *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1452-53 (7th Cir. 1996).

¹⁹³ *Id.*

¹⁹⁴ *Id.* at 1453; *see also Davidson & Assocs. v. Jung*, 422 F.3d 630, 634 n.4 (8th Cir. 2005).

¹⁹⁵ Jeff Hinman, *I'm a EULA. I'm a Contract. Apple Fumbles, Exposes EULA Dangers, THE LEGALITY (U. OF OR. SCH. OF LAW)* (Apr. 30, 2008), <http://www.thelegality.com/2008/04/30/i'm-a-eula-i'm-a-contract-apple-fumbles-exposes-eula-dangers/> (last visited Mar. 6, 2010).

¹⁹⁶ *See PRINCIPLES OF THE LAW OF SOFTWARE CONTRACTS*, *supra* note 186, at § 1.11 cmts. a-b.

requirements may be downplayed or eliminated.¹⁹⁷ A number of federal courts have recognized that “[t]he two elements operate on a sliding scale such that the more significant one is, the less significant the other need be.”¹⁹⁸ The question remains whether reverse engineering restrictions, as a matter of public policy, actually oppress software users.

While most previously identified unconscionable software licensing terms involved terms of arbitration, which are generally clearer for courts to assess, terms that grant excessive general power to software vendors also can be seen as unconscionable.¹⁹⁹ “In the light of the Constitution’s stated goal of ‘promot[ing] the Progress of Science and useful Arts’ courts should be especially careful to protect the right to reverse engineer to create interoperable software products.”²⁰⁰ Particularly where Congress has already granted specific rights to software users and restricted the ability of states to tamper those rights, courts have a duty to recognize the public policy initiatives set by Congress. This includes prohibiting private parties from restricting those rights through form contracts that, even if most consumers read them, stand unopposed as the uncontested norm in the commercial software industry.²⁰¹ Protecting reverse engineering rights in boilerplate EULAs also qualifies under the suggested conditions for unenforceability identified by the ALI in its Principles of the Law of Software Contracts:

- (1) whether the agreement effectively expands the scope of the transferor’s rights or contracts the scope of the transferee’s rights to its own creations under federal law; (2) whether the agreement was negotiated and the parties’ relative bargaining power; (3) the degree to which enforcement of the provision is likely to affect competition adversely; and (4) the degree to which enforcement of the provision is likely to affect innovation adversely.²⁰²

While the benefits of this unenforceability standard may be inherent in its subjectivity, the ALI test may have some pragmatic drawbacks. Federal courts have very rarely overruled the validity of any contract on public policy grounds.²⁰³ But encouraging federal courts to use the ALI factors to determine the legal viability of software licensing terms may be the only way for judges

¹⁹⁷ *Id.* at § 1.11 cmt. c.

¹⁹⁸ *Comb v. Paypal, Inc.*, 218 F. Supp. 2d 1165, 1172 (N.D. Cal. 2002); *see also Bragg v. Linden Research, Inc.*, 487 F. Supp. 2d 593, 605 (E.D. Pa. 2007).

¹⁹⁹ *See* PRINCIPLES OF THE LAW OF SOFTWARE CONTRACTS, *supra* note 186, at § 1.11 cmt. c and illus. 5.

²⁰⁰ *See* *Morris*, *supra* note 164, at 268.

²⁰¹ *Id.* at 268-69.

²⁰² *See* PRINCIPLES OF THE LAW OF SOFTWARE CONTRACTS, *supra* note 186, at § 1.09 cmt. c.

²⁰³ *Id.* at § 1.10 cmt. b.

to avoid throwing the proverbial baby out with the bathwater. Software licensing agreements exist within a specific subset of contract law due to the policy groundings of federal copyright law and deserve treatment not normally reserved for other contract cases. Directly analyzing the substance of software licenses allows courts to distinguish between aggressive reverse engineering prohibitions and open source licensing terms that serve to promote the spread of ideas and information, preserving the intent of federal copyright law and the original policy findings of CONTU.

Another alternative, which the ALI identifies as a way to circumvent a spectrum or subjective test, would be to explicitly identify particular suspect terms as unconscionable in form contracts and EULAs.²⁰⁴ As the case law and statutory history suggest, reverse engineering is a particularly problematic aspect of licensing law and is subject to greater attention due to its formal designation as a fair use exception to copyright use.²⁰⁵ Designating broad prohibitions of reverse engineering as explicitly unconscionable, or even instituting a rebuttable presumption of unconscionability, would protect fair use exceptions in copyright law, encourage future software development by allowing for interoperability studies, and strengthen the general positions of copyright holders by removing one of the major practical impediments of the DMCA without impeding on the more general rights and abilities of software developers, particularly those of the open source community.²⁰⁶

B. Effect of Antitrust Law on Reverse Engineering Prohibitions

Another lens through which to analyze the reverse engineering conflict is to study the extent to which EULA provisions constitute antitrust violations. Antitrust laws are primarily designed to protect commerce from artificial restraints and monopolies in the interest of maintaining competition and promoting an efficient market.²⁰⁷ Monopoly power is generally defined as “the power to control prices or exclude competition.”²⁰⁸ While there are logistical problems applying antitrust law to software and technology cases, enforcement

²⁰⁴ See *id.* at § 1.09 cmt. c.

²⁰⁵ *Id.*

²⁰⁶ See Morris, *supra* note 164, at 268.

²⁰⁷ Burton D. Garland, Jr. & Reuven R. Levary, *The Role of American Antitrust Laws in Today's Competitive Global Marketplace*, 6 U. MIAMI BUS. L.J. 43, 43-44 (1997) (arguing that antitrust laws are aimed at promoting competitive conduct, protecting consumer welfare, and ensuring that economic resources are dedicated to the most beneficial and efficient uses). *But see* John B. Kirkwood & Robert H. Lande, *The Fundamental Goal of Antitrust: Protecting Consumers, Not Increasing Efficiency*, 84 NOTRE DAME L. REV. 191, 192-97 (2008) (arguing that the “Chicago School’s efficiency view is not only incorrect on the merits; it has not triumphed in the courts”).

²⁰⁸ *United States v. E.I. du Pont de Nemours & Co.*, 351 U.S. 377, 391 (1956).

of the law can prevent manipulation of the public and preserve the consumer benefits of market competition.²⁰⁹ In order to prove an antitrust claim under the Sherman Act, a plaintiff must:

- (1) define a “relevant market;”
- (2) show that defendant possesses “monopoly power” within this market; and
- (3) demonstrate that this monopoly power was acquired or maintained by anticompetitive “willful” acts, “as distinguished from growth or development as a consequence of a superior product, business acumen, or historic accident.”²¹⁰

The Supreme Court has established that copyright holders do not have the right to fix prices or commit any other violations of the Sherman Act that are not reasonably necessary to the maintenance and use of that copyright.²¹¹ But there are suggestions that copyright misuse through the leveraging of market power would be sufficiently unreasonable to qualify for review under antitrust law.²¹² In *Lasercomb America, Inc. v. Reynolds*, the Fourth Circuit “held that it was misuse for the plaintiff to have included in its software license a clause prohibiting its licensee from participating in the development of any competitive software for a period of ninety-nine years.”²¹³ Reverse engineering functionally performs in the same way as a non-competition clause – by banning reverse engineering, overly broad protection is given to the first innovator in the field to freeze out competitors from the market.²¹⁴ However, in a key differentiation from intellectual property law, the rights of contracting parties to consent to illegal activity and create antitrust safe havens are non-existent as a matter of general antitrust law.²¹⁵ In many ways, antitrust law might be a better avenue than conventional intellectual property law to protect the fair use provisions of intellectual property law and promote the public interest by creating a market for new technological innovations and encouraging future creativity without eliminating the incentives of copyright holders and software developers.

VI. CONCLUSION

Supporters of the open source software community should, by all means,

²⁰⁹ Michele Kelber, *Living in America: Antitrust Law and the Software Industry; What Can We Do?*, 9 U. BALT. INTELL. PROP. L.J. 155, 163-68 (2001).

²¹⁰ Robert H. Lande & Sturgis M. Sobin, *Reverse Engineering of Computer Software and U.S. Antitrust Law*, 9 HARV. J.L. & TECH. 237, 251 (1996) (citing *United States v. Grinnell Corp.*, 384 U.S. 563, 570-71 (1966)).

²¹¹ *Broad. Music, Inc. v. Columbia Broad. Sys.*, 441 U.S. 1, 19 (1979).

²¹² See Lande & Sobin, *supra* note 210, at 248-50.

²¹³ *Id.* at 249 (citing *Lasercomb Am., Inc. v. Reynolds*, 911 F.2d 970 (4th Cir. 1990)).

²¹⁴ *Id.* at 270.

²¹⁵ *Id.* at 272-73.

celebrate the Federal Circuit's decision in *Jacobsen v. Katzer*. The decision gives long overdue recognition to a highly productive and under-rewarded group of software developers that have driven key innovations in the computer industry for over three decades. However, a very wary eye needs to be cast toward the growing trend of courts to enforce end user license agreements based on potential economic value. A uniform application of the economic value standard will restrict the freedom of programmers to develop new programs, regardless of whether their motives are competitive, productive, educational, or even mere curiosity. When viewed in line with cases like *Bowers* and *Davidson*, *Jacobsen* could derail the efforts of the open source community and its allies to promote the technological liberties, including the important right to reverse engineer existing software, which these groups so adamantly champion. Without the proper efforts in courts, state legislatures, and Congress, the open source community may win this key battle to establish itself as a legitimate and legally recognized institution in the American technological environment while losing the war to keep computing open and free. And it is not just the open source community that stands to lose as courts give more strength to EULA terms; commercial developers would also be crippled by explicit restrictions prohibiting reverse engineering.

There are, however, ways for the open source community to have its cake and eat it too. If courts choose to continue to use copyright law and the DMCA to protect the intellectual property of software developers, then there needs to be a proper place for licenses and copyright law that compliments the will of Congress, not expands beyond and supersedes it. First, it would be appropriate for courts to adopt the ALI Principles of the Law of Software Contracts as interpretational guidelines. The guidelines reflect the complicated nature of the American software industry and would allow for courts to distinguish between licensing clauses that encourage the distribution and sharing of information, like those found in open source "copyleft" licenses, and terms that stand against the principles of intellectual property law as enumerated in the Constitution, including blanket restrictions of reverse engineering. Additionally, select applications of antitrust law and the Sherman Act would serve to prevent the monopolization of information and encourage innovation and growth in the software market to the benefit of the public interest in future creative efforts by developers and programmers.

Finally, there is a distinct need for Congress to summon a second CONTU. When Congress enacted the recommendations of the first CONTU thirty years ago to construct a copyright regime for computer programs, the participating legislators and policy experts were working under the auspices of an era of computer technology that is radically different from the environment in which software is programmed and sold today. While Congress updated these rules in part by the passage of the DMCA in 1998, the challenges facing small and open source programmers, as well as the judicial challenges to reverse

2010]

OPEN SOURCE SOFTWARE

engineering, reflect that the actual state of the law is drifting farther and farther away from the intent of Congress and the Constitution to promote the arts and sciences for the betterment of American society. There is no indication that CONTU contemplated the type of mass-consumer software licensing agreements that are currently the industry standard. This has left holes in federal copyright policy and, in the case of reverse engineering, the state of the law has seemingly returned in full to the pre-CONTU status of full industry control over existing products in the market. Only once intellectual property protection for software is wholly reexamined with the original purposes of the legislation in mind will judges and Congress have the ability to effectively institute a cohesive, comprehensive, and efficient regime for intellectual property use control.