Washington Journal of Law, Technology & Arts

Volume 6 | Issue 4 Article 6

4-1-2011

Jacobsen Revisited: Conditions, Covenants and the Future of Open-Source Software Licenses

Yamini Menon

WASHINGTON JOURNAL OF LAW, TECHNOLOGY & ARTS
LAW, TECHNOLOGY & ARTS GROUP FEATURE ARTICLE
VOLUME 6, ISSUE 4 SPRING 2011

JACOBSEN REVISITED: CONDITIONS, COVENANTS AND THE FUTURE OF OPEN-SOURCE SOFTWARE LICENSES

Yamini Menon[*]

CITE AS: 6 WASH J.L. TECH. & ARTS 311 (2011)
http://digital.law.washington.edu/dspace-law/bitstream/handle/1773.1/1054/6WJLTA311.pdf?sequence=1

ABSTRACT

Open-source software licensing has become mainstream in the field of software development. Nowhere is this more evident than in the 2008 Federal Circuit decision Jacobsen v. Katzer, *where the court first interpreted the terms of an open-source software license. The* Jacobsen *decision offers an important first step in how to interpret the terms of an open-source license, though it does not address how to interpret licenses other than the Artistic License. This Article explores how* Jacobsen*'s reasoning can be used to interpret the terms of other open-source licenses, particularly the GPL v.2, GPL v.3, Apache License v.2, BSD License, and the Mozilla Public License. After examining the Federal Circuit's discussion of "conditions" and "covenants," this Article suggests how to draw a principled distinction between these terms when interpreting an open-source license. This Article also*

[*] Yamini Menon, Johns Hopkins University, B.A. 1998; American University–Washington College of Law, J.D. 2003; University of Washington School of Law, LLM 2010. A special thanks to Professor Robert W. Gomulkiewicz for his feedback, guidance, and advice throughout the writing process. I extend my deepest gratitude to my parents, Unni & Usha Menon, to my sister, Yamuna Menon, and to my husband Ashesh Bakshi. Thank you to Signe Naeve and the staff of the Washington Journal of Law, Technology & Arts for their efforts and feedback in preparing this Article for publication.

examines policy considerations that arise when interpreting the terms of an open-source license and offers proposals for reducing exposure to potential copyright infringement liability when using software licensed under an open-source software license.

TABLE OF CONTENTS

INTRODUCTION

While software developers have been sharing code since the first days of the software industry, they did not begin using open-source licenses to produce collaborative works until the late 1990s.[1] Since then, open-source licensing has become mainstream in the field of software development.[2] Open-source licensing even remains relevant for those who prefer proprietary licensing to open-source licensing. Large players in the software industry (e.g., Novell, Sun, IBM, Cisco, and Microsoft) either offer or support these licenses.[3] Even those who do not use open-source licenses may become exposed to them through third-party vendors.[4] Understanding the legal implications of using open-source software is therefore vital, even in professions that do not involve writing and developing code.[5]

These implications are evident in the recent Federal Circuit decision *Jacobsen v. Katzer*,[6] in which the Federal Circuit issued the first judicial decision to address remedies available to a litigant when a defendant has violated the terms of an open-source software license.[7] In *Jacobsen*, the United States District Court for the Northern District of California held that breaching the terms of an open-source license (in this case, the Artistic License) did not rise to the level of copyright infringement.[8] On appeal, the Federal Circuit reversed the lower court's decision and found such a breach nonetheless limits the scope of the license, even without purely

[1] *See* Amanda Albrecht Earl, Case Note, *Copyright Infringement And Open Source Public Licenses:* Jacobsen v. Katzer*, 535 F.3d 1373 (Fed. Cir. 2008)*, 77 U. CIN. L. REV. 1605, 1605 (2009).

[2] ALAN STERN & A. CLIFFORD ALLEN, OPEN SOURCE LICENSING, 985 Prac. L. Inst. 321, 327 (PLI 2009).

[3] *Id.*

[4] *Id.*

[5] *See id.* (explaining that the consequences of ignorance or apathy about open source licenses include bad publicity, litigation, and being outmaneuvered by competitors, among other things).

[6] Jacobsen v. Katzer, 535 F.3d 1373,(Fed. Cir. 2008).

[7] DENIS T. RICE, DEVELOPMENTS IN LICENSING OF INTELLECTUAL PROPERTY IN THE UNITED STATES, 984 Prac. L. Inst. 605, 616 (PLI 2009).

[8] Jacobsen v. Katzer, 2007 WL 2358628 (N.D. Cal. 2007), *vacated in part*, 535 F.3d 1373 (Fed. Cir. 2008).

economic harm arising from a breach of the conditions in an open-source license.[9] The court held that exceeding the scope of an open-source license constitutes copyright infringement.[10]

Although the open-source community heralded the *Jacobsen* decision for providing legal legitimacy for open-source licenses,[11] the case still left several unanswered questions. First, how does *Jacobsen* apply to open-sources licenses other than the Artistic License? Second, because the Federal Circuit's decision in *Jacobsen* hinges on the distinction between "conditions" and "covenants" in an open-source license, how is this distinction drawn when interpreting the language of an open-source license?

This Article attempts to answer these questions by examining how *Jacobsen* applies to open-source licenses including the GPL v.2, GPL v.3, Apache License v.2, BSD License, and the Mozilla License. Part I summarizes the *Jacobsen* decision and highlights several gaps in its reasoning. Part II provides an overview of the open-source licenses that will be examined through the lens of *Jacobsen*. Part III attempts to fill the gaps left by the *Jacobsen* decision by offering a principled distinction between conditions and covenants when reading the language of an open-source license. By using this distinction to interpret the language of the various open-source licenses listed above, this Article will provide guidance to attorneys, software developers, and anyone else who may be affected by the use of open-source code. It will also illustrate how the Federal Circuit's reasoning can be extended to licenses other than the Artistic License litigated in *Jacobsen* and will offer proposals to help avoid potential copyright infringement liability pursuant to *Jacobsen*.

I. AN OVERVIEW OF *JACOBSEN V. KATZER*

A. Factual and Procedural Background

The story behind *Jacobsen* is a familiar one in the open-source

[9] *Jacobsen*, 535 F.3d at 1382; *see also* Earl, *supra* note 1, at 1606.

[10] Earl, *supra* note 1, at 1606.

[11] *See id.* at 1624 (noting that the *Jacobsen* decision was a victory for open source advocates because the Artistic License, which is viewed as being broad and vague, was found enforceable under copyright law).

community. The plaintiff, Robert Jacobsen, and the defendant, Matthew Katzer, had each developed competing open-source software applications to control model trains.[12] Jacobsen and his open-source group, Java Model Railroad Interface Project (JMRI), developed the Decoder Pro application and licensed it under an open-source license (the Artistic License) that contained a number of preconditions for use.[13] Matthew Katzer and his company sold Decoder Commander.[14]

Katzer had obtained a patent for Decoder Commander in March 2005 and then accused Jacobsen of infringing his patent for Decoder Commander.[15] Jacobsen responded by filing a complaint for declaratory judgment, seeking a finding of non-infringement, invalidity, and unenforceability of Katzer's patent.[16]

While preparing his complaint for declaratory judgment, Jacobsen discovered that there were similarities between the code contained in his Decoder Pro software and the code contained in Katzer's Decoder Commander software.[17] Jacobsen also realized that Katzer had used portions of the open-source Decoder Pro code in Decoder Commander, but had failed to adhere to the requirements of the Artistic License.[18] Under the terms of this license, a user had to comply with certain preconditions before redistributing or modifying the software.[19] One of these conditions required that the downstream user keep the author's name and describe or disclose any changes made to the software program.[20]

Jacobsen amended his complaint for declaratory judgment to

[12] *See* PAUL H. ARNE, *JACOBSEN V. KATZER*: OPEN SOURCE LICENSE VALIDATION – HOW FAR DOES IT GO?, 961 Prac. L. Inst. 133, 137 (PLI 2009).

[13] *See* Hersh R. Reddy, Note, Jacobsen v. Katzer*: The Federal Circuit Weighs In On The Enforceability Of Free And Open Source Software Licenses*, 24 BERKELEY TECH. L.J. 299, 314-15 (2009).

[14] Earl, *supra* note 1, at 1614.

[15] *Id.*

[16] *Id.*

[17] *Id.*

[18] *Id.* Jacobsen learned that the Decoder Commander files did not contain any reference to the authors' names, the JMRI copyright notices were not referenced in the "COPYING" file or the original source location of the file, and the files did not keep track of the changes made from JMRI's original source code.

[19] Reddy, *supra* note 13, at 315.

[20] *Id.*

include copyright infringement and filed a motion for a preliminary injunction to prevent Katzer from distributing Decoder Commander.[21] Interestingly, Katzer later admitted to copying portions of the Decoder Pro software.[22]

The district court denied and dismissed Jacobsen's motion for a preliminary injunction on the basis that Katzer's use of portions of Jacobsen's source code in Decoder Commander did not rise to the level of copyright infringement, but merely constituted a violation of Jacobsen's contractual right.[23] According to the district court, the only way Jacobsen could establish that Katzer's copying constituted copyright infringement was to show that Katzer had exceeded the scope of the Artistic License.[24] The court found that Katzer's actions were within the scope of the Artistic License under its interpretation of the license terms.[25] Consequently, the district court held that Katzer had not committed copyright infringement.[26]

In its reasoning, the court found that the scope of the license was intentionally broad and that the attribution requirement did not limit the scope of the license.[27] The court further held that an alleged violation of the license conditions may constitute a breach of the license, but that does not create liability for copyright infringement where it would not otherwise exist.[28] Therefore, the court interpreted the restrictions contained in the Artistic License as covenants rather than as preconditions.[29] Thus, Jacobsen needed to pursue a cause of action for breach of contract,[30] not for copyright infringement.[31] But in order to obtain a preliminary injunction pursuant to a contract law theory, Jacobsen needed to demonstrate both a likelihood of success on the merits of the case (i.e., prove every element of a contract and a

[21] Earl, *supra* note 1, at 1614.
[22] *Id.*
[23] Jacobsen v. Katzer, 2007 WL 2358628 at *7 (N.D. Cal. 2007).
[24] *Id.*
[25] *Id.*
[26] *Id.*
[27] *Id.*
[28] *Id.*
[29] *Jacobsen*, 2007 WL 2358628 at *7.
[30] *Id.*
[31] *Id.*

breach thereof) and irreparable harm.[32] Because the court had found that Katzer's use did not exceed the scope of the Artistic License, it held that Jacobsen had failed to demonstrate a likelihood of success on the merits and was not entitled to the presumption of irreparable harm.[33] Consequently, the court denied Jacobsen's motion for a preliminary injunction, resulting in Jacobsen's appeal to the Federal Circuit.[34]

When the Federal Circuit decided the *Jacobsen* appeal, it became the first court to recognize and support the terms of an open-source license.[35] The court's decision turned on whether the terms of the Artistic License were conditions of the license subject to copyright protection or covenants subject to contract remedies.[36] In its reasoning, the Federal Circuit expressly stated that if the terms of the Artistic License were both covenants and conditions, then this would limit the scope of the license and copyright law would apply; if, however, the terms were only covenants, contract law would apply exclusively.[37]

The Federal Circuit never actually explained the difference between conditions and covenants in its opinion. Instead, the court simply pointed to the language of the Artistic License.[38] The court concluded that taken at face value, the language of the Artistic License creates conditions.[39] Relying on the Preamble as well as Section Three of the Artistic License, the court noted that the intent of the license was to state the conditions under which it is permissible to copy a software package.[40] Moreover, the court noted that users have certain rights "provided that" they adhere to the stated

[32] *See id.* at *5.

[33] A breach of contract does not carry with it a presumption of irreparable harm, but rather requires separate proof of irreparable harm. *See* Earl, *supra* note 1, at 1615.

[34] *Jacobsen*, 2007 WL 2358628 at *7; *see also* Earl, *supra* note 1, at 1615.

[35] ARNE, *supra* note 12, at 141.

[36] *See* Jacobsen v. Katzer, 535 F.3d 1373, 1380 (Fed. Cir. 2008).

[37] *Id.*

[38] *Id.* at 1381.

[39] *See id.* (relying on the language of the Artistic License stating that "[t]he intent of this document is to state the *conditions* under which a Package may be copied.").

[40] *Id.* at 1381.

conditions.[41]

The court cited only one California Supreme Court case, *Diepenbrock v. Luiz*, in support of its proposition that the language of the Artistic License created a condition and not merely a covenant.[42] The court noted that under California contract law, the words "provided that" traditionally denote the existence of a condition.[43] Therefore, the court found that the Artistic License used "the traditional language of conditions by noting that the rights to copy, modify, and distribute are granted '*provided that*' the conditions are met."[44]

In February 2010, Jacobsen and Katzer settled the litigation by agreeing to an injunction.[45] Now that the *Jacobsen* case has ended, the far-reaching implications of the decision are becoming apparent to the open-source community.

B. The Implications of Jacobsen

While the Federal Circuit decision provides limited guidance on how one should interpret the terms of an open-source license, the decision raises more questions than it answered. For instance, the court indicated that the distinction between a condition and a covenant is what determines whether or not copyright infringement liability will attach when someone violates the language of an open-source license.[46] However, the Federal Circuit did not provide much guidance as to how to make such a distinction.

In making this determination, should courts look at the intent of

[41] *Id.* Although the Federal Circuit stated that users retain the rights to copy, modify, and distribute the software "provided that" they adhere to certain conditions listed in the Artistic License, the court did not specifically list or explain these conditions.

[42] *Id.* (citing Diepenbrock v. Luiz, 115 P. 743 (1911)).

[43] *See id.* Interestingly, *Diepenbrock* is not a contract case, but a real property case involving the interpretation of language contained in a lease. *See Diepenbrock*, 115 P. at 743.

[44] *Jacobsen*, 535 F.3d at 1381.

[45] *See* Posting of Andy Updegrove to The Standards Blog, *A Big Victory for F/OSS:* Jacobsen v. Katzer *is Settled*, http://www.consortiuminfo.org/standardsblog/article.php?story=201002190850472 (Feb. 19, 2010, 09:45 PST)..

[46] *See Jacobsen*, 535 F.3d at 1381.

the parties as courts normally do when interpreting a contract?[47] Should the courts permit drafters of an open-source license to choose whether their license provisions are conditions or covenants?[48] If so, what "magic words" should drafters of open-source licenses incorporate to ensure the terms are enforceable? One key issue is whether it is appropriate, as a matter of public policy, to allow drafters the freedom to craft license language in such a way as to change a covenant into a condition to hold parties liable for copyright infringement.[49] It other words, is it permissible to use contract law to create a cause of action for copyright infringement that otherwise would not exist? While the traditional notion of freedom of contract allows for this, the opposing view is that only conditions touching upon the exclusive rights explicitly provided under copyright law should qualify as license conditions.[50]

Before attempting to answer these questions left open by *Jacobsen*, it is important to provide some background on the spectrum of existing open-source licenses that will be at issue going forward.

II. OVERVIEW OF OPEN-SOURCE LICENSES TO WHICH *JACOBSEN* WILL BE APPLIED

This section examines five representative open-source software licenses of the sixty-plus licenses certified by the Open Source Initiative (OSI): General Public License (GPL) versions 2 and 3, Apache License version 2, Berkeley Software Distribution (BSD) License, and the Mozilla License.[51]

A. GPL v.2

[47] *See* Robert W. Gomulkiewicz, *Conditions and Covenants in License Contracts: Tales From a Test of the Artistic License*, 17 TEX. INTELL. PROP. L.J. 335, 356 (2009).

[48] *See id.* at 351.

[49] *See id.* at 353-54.

[50] *See* Gomulkiewicz, *supra* note 47, at 354.

[51] Open Source Initiative, *Licenses By Name*, http://www.opensource.org/licenses/alphabetical (last visited Apr. 29, 2011); *see also* STERN & ALLEN, *supra* note 2, at 333.

The GNU GPL may be the most common free open-source license.[52] Version 2 of the GPL (GPL v.2) appeared in 1991 and is the license that governs the Linux kernel.[53] It is also one of the most controversial licenses because of its "strong copyleft" terms and the debate over how these terms apply in specific cases.[54] The "strong copyleft" terms of GPL v.2 state that when a user distributes the original software, he or she must also license the "work as a whole" under those same terms.[55]

While it is evident what is meant by "work as a whole" in some contexts, it is unclear how one can satisfy this requirement in the context of software when most software applications interact with other programs, libraries, drivers, and components of operating systems as a routine way of functioning.[56] Thus, there is much debate over where to draw the line with respect to this requirement.[57] Federal copyright law regarding derivative works is an important consideration and makes drawing this line even more difficult in the context of computer software.[58]

GPL v.2 addresses the "work as a whole" issue by indicating that the aggregation of two works in the same medium for the purposes of distribution does not trigger the provision named above.[59] It states that this provision does not affect an independent and separate work that does not derive from original software code when that work is distributed as a separate work.[60] The license further states that the intent of the license drafter was merely to control distribution of "derivative or collective works based on the [original software]."[61] Finally, GPL v.2 states that the intent of the drafters is merely to

[52] RICE, *supra* note 7, at 615.

[53] STERN & ALLEN, *supra* note 2, at 350.

[54] *Id.*

[55] GNU Operating System, *What Is Copyleft*, http://www.gnu.org/copyleft/ (last visited Apr. 29, 2011) (arguing that the licensing of all copyrights, along with the requirement that others do the same, is generally viewed as making the license a "copyleft" license); *see also* STERN & ALLEN, *supra* note 2, at 350-51.

[56] STERN & ALLEN, *supra* note 2, at 350-51.

[57] *Id.*

[58] *Id.* at 351.

[59] *Id.*

[60] *Id.*

[61] *Id.*

control the distribution of "derivative or collective works based on the [original software]."[62]

Unfortunately, no American court has yet to interpret the substantive terms of the GPL v.2, which has resulted in debates about its terms among attorneys, software engineers, and the Free Software Foundation (FSF).[63] With no bright-line rule available for interpreting the terms of the GPL v.2, companies and the open-source community have developed their own norms and interpretations.[64] To further complicate matters, the question of the enforceability of copyleft provisions under the copyright laws of various jurisdictions also remains unclear.[65] Copyright law provides copyright holders the legal right to prevent others from using their program except by complying with the copyright holders' license terms.[66] However, an action under copyright law can also raise enforceability issues regarding strong copyleft provisions that try to supersede the copyright ownership of works linked to the copyleft work.[67] Thus, these aspects of the GPL v.2 can make interpreting the terms of this license challenging.

B. GPL v.3

Version 3 of the GPL (GPL v.3) was released on June 29, 2007, after a lengthy period of drafting and discussion.[68] Since its release,

[62] STERN & ALLEN, *supra* note 2, at 350-51.

[63] *Id.*

[64] For example, Linus Torvalds (the software engineer best known for the development of the Linux kernel operating system) has stated that he does not believe that the so called "viral effects" of GPL v.2 extend to the applications that run on top of Linux, but rather that these applications are considered separate works that are untainted by the copyleft requirement of GPL v.2 if they access the Linux operating system's functions using its own standard interfaces. Torvalds' interpretation may be considered dispositive with respect to GPL software that he owns and may even be persuasive vis-à-vis other Linux components (such as the Linux kernel), but it is still unclear whether this interpretation applies to other code licenses under GPL v.2. *Id.*

[65] *Id.* at 351-52.

[66] *Id.*

[67] *Id.* at 352.

[68] Brett Smith, *A Quick Guide to GPL v3*, GNU OPERATING SYSTEM, http://www.gnu.org/licenses/quick-guide-gplv3.pdf; STERN & ALLEN, *supra* note 2,

thousands of open-source projects have adopted this version of the license.[69] GPL v.3 was drafted with the goal of addressing some of the areas that GPL v.2 had failed to address, such as patent indemnity, internationalization, and remedies for inadvertent license infringement.[70] The newer version of the license (drafted by Richard Stallman of the FSF and Eben Moglen of the Software Freedom Centre) reflects the drafting process that led to its creation — a "broad, consensus-driven process" that involved seeking feedback from four separate committees and broad comment from the public during the course of an eighteen month period.[71] The purpose behind GPL v.3, however, is the same as that of GPL v.2 — to "ensure the preservation of users' freedom to 'run, copy, distribute, study, change and improve the software.'"[72] Although this newest version of the GPL adds some new features and resolves some of the ambiguities contained in the previous version, the ambiguity surrounding the copyleft provisions of the license remains unresolved.

The GPL v.3 contains an "internationalization" provision, which reflects a change in terminology found in the GPL v.3.[73] Previously, critics claimed that the language contained in GPL v.2 as having too much of an American focus rather than an international one.[74] Consequently, the drafters of GPL v.3 changed the phrasing to reflect a more neutral approach, as seen in the use of terms such as "propagate" and "convey," rather than "copy" and "distribute."[75] Although GPL v.3 defines these terms in more detail, it still refers to very broad copyright ideas that many people believe are still too vague to apply to software code.[76] Also, different countries may still reach other conclusions as to what comprises a derivative work of a

at 354.

[69] STERN & ALLEN, *supra* note 2, at 354.

[70] Liz Laffan, *GPL v2 vs. GPL v3: The Two Seminal Open Source Licenses: Their Roots, Consequences, and Repercussions - Analysing, Understanding, and Interpreting the GPL v2 and GPL v3 Licenses*, VISION MOBILE (2007), www.linuxfordevices.com/files/misc/GPLv2_vs_GPLv3.pdf.

[71] *Id.*

[72] *Id. at 4.*

[73] *See* STERN & ALLEN, *supra* note 2, at 354.

[74] *Id.*

[75] *Id.*

[76] *See* STERN & ALLEN, *supra* note 2, at 354.

copyleft software program and whether or not the terms contained in GPL v.3 apply to works that are not derivative works.[77]

There are three features of GPL v.3 in particular that have caused controversy. The first is the patent provision.[78] Unlike GPL v.2, GPL v.3 contains a clear and explicit royalty-free patent grant from all contributors to the software code under all "essential patent claims" that are owned or controlled by the contributor.[79] The license also prohibits attempts at controlling downstream distribution through a discriminatory patent license**.**[80] While there may be general agreement that the adoption of an express patent grant provision is beneficial to those who use software covered by GPL v.3, there is still much debate regarding the specific wording used for the patent grant.[81] Some feel that the wording is too broad because it covers future patents that may be filed or invented with respect to an area of technology similar to that which a user contributes or licenses under GPL v.3.[82] In addition, the use of the phrase "knowingly rely" in the patent grant has sparked several controversies.[83]

A second feature of GPL v.3 that has caused debate is the section of the license entitled "Protecting Users' Legal Rights From Anti-Circumvention Law,"[84] which is a section of the license that intends to prevent code distributed under GPL v.3 from being included in technology or products that would be used to enforce the Digital Millennium Copyright Act (DMCA).[85] This provision restrains the type of product in which the software code can be used, which many believe is not appropriate for free and open-source software (FOSS)

[77] *Id.* at 354-55.

[78] Laffan, *supra* note 70, at 8.

[79] STERN & ALLEN, *supra* note 2, at 355.

[80] *Id.*

[81] Laffan, *supra* note 70, at 8.

[82] *See id.*

[83] *Id.*

[84] *Id.*

[85] *Id.* The DMCA makes it a crime to produce and disseminate technology, devices, or services that are used to circumvent processes that control access to copyrighted materials. *Id.* This is known as "Digital Rights Management," (DRM), which refers to access control technologies that can be used to impose limits and restrictions on the usage of digital content and devices). *See* Electronic Frontier Foundation, *Digital Rights Management*, www.eff.org/isues/drm (last visited Apr. 29, 2011).

licensing.[86] When someone distributes code under the terms of GPL v.3, this waives the right to enforce any legal remedy under the DMCA and any other laws that implements Article 11 of the World Intellectual Property Organization (WIPO) copyright treaty.[87] Thus, it is impracticable to use code covered under GPL v.3 as part of any digital rights management mechanism.[88]

The third controversial feature of GPL v.3 is the "TiVoisation" feature, which refers to a feature contained in TiVo, a digital video recorder and consumer device that enables consumers to record multiple television programs at once and view them later.[89] TiVo runs a small Linux operating system licensed under GPL v.2, and under that license, hardware manufacturers are required to disclose the source code to other users.[90] While TiVo makes the source code available to other users and users are able to alter and compile the source code licensed under GPL v.2, modified software will not run because a special mechanism in the Tivo device shuts down the system when it detects changes to the code.[91]

Thus, while TiVo technically adheres to the requirements of GPL v.2, it violates the four freedoms established by the FSF.[92] In order to address this circumvention by TiVo, GPL v.3 includes new terms and obligations that are "intended to ensure that entities using GPL v.3 licensed software for any user product also provide any and all additional information necessary to ensure installation and running of the software."[93] These new terms and conditions are especially significant since know-how contained in the installation methods may provide value to the entity that uses the GPL v.3 licensed software.[94] In addition, if a user or third party has the ability to update software licensed under GPL v.3 that is stored in ROM, this triggers the obligation to provide source code.[95]

[86] *Id.*
[87] STERN & ALLEN, *supra* note 2, at 355.
[88] *See id.*; *see also* Laffan, *supra* note 70, at 8.
[89] *See* Laffan, *supra* note 70 at 8-9.
[90] *Id.*
[91] *Id.* at 8.
[92] *Id.* at 8-9.
[93] *Id.*
[94] *Id.*
[95] *Id.*

C. Apache 2.0

The Apache License was adopted by the Apache Software Foundation in January 2004 and favors a model of software development that is centrally managed, where many contributors provide improvements to the software code directly to the original licensor.[96] The Apache License requires users to maintain the copyright notice and disclaimer, but is not necessarily a copyleft license.[97] This is because although a contribution that is deliberately submitted to the original licensor is presumed to be licensed under the same license terms as the original code, a contributor can overcome the presumption by stating different terms.[98] The license also permits users to use source code for developing both closed-source software projects as well as open-source software projects.[99]

Like other open-source licenses, the Apache License allows users to use the software for any purpose, including distribution, modification, or distribution of modified versions of the software under the license terms.[100] Users need not distribute modified versions of the software using the same license, but all license files must retain any original patent, copyright, trademark, or attribution notices in the redistributed code.[101] The license also contains an express patent grant and a patent peace provision.[102] In addition, while a licensee may distribute a derivative work under different license terms, a copy of the Apache License must be included every time a work is distributed.[103] Moreover, the licensee must describe

[96] STERN & ALLEN, *supra* note 2, at 348.

[97] *Id.*

[98] *Id*.

[99] *See* OSS Watch – Open Source Software Advisory Service, *The Apache License (v2) – An Overview*, www.oss-watch.ac.uk/resources/apache2.xml (last visited Apr. 29, 2011).

[100] *See id.*

[101] *See id.*

[102] The license contains a provision stating that the grant of patent rights is withdrawn if the licensee initiates legal action against the licensor over patent infringements of the covered software. *Id.*

[103] *See* STERN & ALLEN, *supra* note 2, at 349.

any modifications made to the code with a notice.[104]

D. Berkeley Software Distribution

The Berkeley Software Distribution (BSD) group of licenses grew out of a 1970's movement at the University of California at Berkeley, where a group of computer programmers developed a version of the Unix operating system.[105] This led to the creation of software known as the Berkeley Software Distribution of Unix or BSD Unix, as well as several variants.[106]

The ideas that led to BSD Unix were similar to the ideas of the FSF.[107] For example, BSD Unix made source code readily available and permitted programmers to make derivative works to fix bugs and improve the software, although there was no formal requirement for doing so.[108] Berkeley charged a small fee intended to cover the cost of copying the source code to a medium that licensees could use.[109] As the software program became more popular, a short and simple version of a license was created that enabled licensees to work with the source code and to make derivative works.[110]

BSD Licenses are a group of licenses that grew from the original license created by Bill Joy (a well-regarded programmer who later co-founded Sun Microsystems) when he developed Unix.[111] In contrast to the GPL Licenses, these licenses are not copyleft licenses.[112] In fact, they are among the least restrictive open-source software licenses.[113]

For derivative works, the BSD Licenses permit, but do not

[104] *Id.* These modifications need not be made available in source form, however.

[105] *See* Dennis M. Kennedy, *A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture*, 20 ST. LOUIS U. PUB. L. REV. 345, 351 (2001).

[106] *Id.* at 352.

[107] *Id.*

[108] *Id.*

[109] *Id.*

[110] *Id.*

[111] *Id.* at 351-52.

[112] *See* Kennedy, *supra* note 105, at 352.

[113] *Id.* at 363.

require, distribution of source code.[114] This makes it possible to combine programs under the BSD Licenses with proprietary software.[115] Examples of code that users have distributed under BSD licenses include commercial software found in Windows NT and the Macintosh operating system, OS X.[116]

The BSD Licenses are much simpler in language and style than typical commercial software licenses and permit distribution and use of source and object code, either with or without modification.[117] The redistribution of source code, however, must retain the required copyright and other notices, the disclaimer of warranties, and the limitation of liability clauses.[118] Many consider the BSD Licenses to be much more “free” than the GPL License because they allow developers to release derivative works under any license they want (including licenses that contain different terms than the BSD Licenses that applies to the original code).[119]

As a result of this freedom, commercial developers of software code are more apt to use the BSD Licenses.[120] The BSD Licenses are a good illustration of the fact that open-source licenses do not need to be copyleft in nature to qualify under the definition of open-source.[121]

E. Mozilla Public License

The Mozilla Public License (MPL) is sponsored by the Mozilla foundation and is most commonly known as the license for the Firefox web browser.[122] As with the Apache License, the drafters of the MPL intended to create a community model of software development.[123] The MPL contains a copyleft provision that requires all modifications to Mozilla source files be made available in source

[114] *Id.*
[115] *Id.*
[116] *Id.*
[117] *Id.*
[118] *See* Kennedy, *supra* note 105, at 364.
[119] *Id.* at 363.
[120] *See id.* at 365.
[121] *See* Kennedy, supra note 105, at 365.
[122] STERN & ALLEN, *supra* note 2, at 349.
[123] *Id.*

code form under the same license.[124] One can avoid this requirement, however, by placing modifications to software in separate files, which has caused some people to describe the MPL as a "weak copyleft" license.[125] Moreover, the language of the MPL states that Mozilla files may be combined with non-Mozilla files to create larger works.[126]

The MPL also contains a grant by the original author of the code and separate sections granting similar rights by subsequent contributors, explicit patent grants, and a patent peace provision.[127] The patent peace provision is a complex provision that comes into effect when a licensee makes a claim against any contributor to the code (even if the claims are not related to the software), but the revocation is limited to the patent grants of contributors named in the patent action.[128] The licensee's patent grant may also be revoked retroactively.[129] Also, a licensee may distribute executable versions of the code under different license terms, and modifications to the code must be described in a file distributed with the code.[130]

Each derivative work must include a "LEGAL" file describing any known intellectual property claims to the code, any known patents required to implement an application programming interface (API) that includes the code, and any statutes, regulations, or orders which prohibit full compliance with the license.[131] Due to the difficulty in complying with this latter provision, many licensees do not adhere to its requirements.[132]

Many view the MPL as a model for open-source licensing of commercial software entities because it appears and reads like a standard commercial software license.[133] This is because the license drafters solicited external comments, including comments from

[124] *Id.*
[125] *Id.*
[126] *Id.*
[127] *Id.*
[128] STERN & ALLEN, *supra* note 2, at 349.
[129] *Id.*
[130] STERN & ALLEN, *supra* note 2, at 349-50.
[131] STERN & ALLEN, supra note 2, at 350.
[132] *Id.*
[133] *See* Kennedy, *supra* note 105, at 365.

attorneys, as part of the drafting process.[134] The MPL is also a model for future releases of commercial software into open-source.[135] The MPL is now in the process of undergoing a revision and is once again open for public comment as part of the revision process.[136]

III. APPLYING *JACOBSEN* TO OPEN-SOURCE LICENSES

A. Defining "Condition" versus "Covenant"

As previously noted, the Federal Circuit's decision in the *Jacobsen* case turns primarily on the interpretation of the open-source license terms as "conditions" or as "covenants." The *Jacobsen* court interpreted the terms of the Artistic License and reached the conclusion that the terms of the license were in fact conditions and not covenants. This distinction between conditions and covenants is important because if a court interprets the terms of an open-source license to be conditions, then a licensee who violates such conditions can be held liable for copyright infringement. To the contrary, if a court interprets the license terms to be covenants, a licensee cannot be held liable for copyright infringement under *Jacobsen*.

In applying *Jacobsen* to other open-source licenses, the first step is to define what constitutes a condition and what constitutes a covenant. Since the Federal Circuit did not define either term or provide any guidance as to how to draw the distinction, lawyers and open-source scholars are now left trying to find a principled way to differentiate between covenants and conditions. As implied by the Federal Circuit opinion in *Jacobsen*, the best way to do this is to look at the law of contracts.[137]

As a preliminary matter, there is some debate as to whether open-source licenses are contracts.[138] Nevertheless most scholars assert

[134] *See id.*

[135] *See* Kennedy, *supra* note 105, at 365–66.

[136] *See* Mitchell Baker, *Updating the Mozilla Public License*, LIZARD WRANGLING MITCHELL ON MOZILLA & MORE, http://blog.lizardwrangler.com/2010/03/10/updating-the-mozilla-public-license (Mar. 10, 2010).

[137] *See Jacobsen*, 535 F.3d 1373, 1381 (Fed. Cir. 2008).

[138] *See* Robert A. Hillman & Maureen A. O'Rourke, *Rethinking Consideration In the Electronic Age*, 61 HASTINGS L.J. 311, 325 (2009). This article explains how

that such licenses should be treated as contracts, despite arguably lacking consideration. [139]

Although an examination of whether open-source licenses are contracts is outside the scope of this Article, it is worth noting that Professor Hillman and Dean O'Rourke have provided compelling reasons to support their argument that open-source licenses are in fact contracts.[140] They suggest that contract and intellectual property law work well together in supporting the FOSS model and provide a useful source of remedies for failing to comply with a license.[141]

This argument not only supports the view of those in the open-source movement who wish to have the remedy of injunctive relief available at their disposal when an open-source license is violated, but also dovetails with the Federal Circuit's own reasoning in *Jacobsen*. The Federal Circuit was very clear in *Jacobsen* that its interpretation of the terms contained in the Artistic License was based on state contract law.[142] Moreover, Hillman and O'Rourke also believe that contract law is the best source of law to accurately determine whether a contractual provision is a pure condition or merely a promise.[143] Therefore, this Article will define condition and covenant and draw the distinction between these two terms by looking at several contract law treatises and the state common law of contracts.

Although several contract law scholars have defined condition and covenants lightly differently in their treatises, the meaning of these terms is relatively uniform. For instance, Professor Corbin

Eben Moglen, the attorney for the FSF, has argued that GPL v.2 is not a contract because: (1) the term "license" denotes a specific technical meaning in property law that denotes unilateral permission to use someone else's property, whereas a "contract" is an exchange of obligations (i.e., either promises for promises or promises of future performance for present performance or payment); and (2) Richard Stallman has provided policy reasons why FSF prefers that copyright law govern GPL rather than contract law. *Id.*

[139] For example, Professor Robert A. Hillman of Cornell Law School and Dean Maureen A. O'Rourke of Boston University School of Law have each asserted that despite the lack of consideration in open-source licenses, this should not prevent these licenses from being treated as contracts. *Id.*

[140] *Id.* at 328-35.

[141] *Id.* at 333.

[142] *See Jacobsen*, 535 F.3d at 1381.

[143] *See* Hillman & O'Rourke, *supra* note 138, at 334–35.

defines the term “condition” as:

> [A]n “operative” fact or event. This means that it is a fact or event that affects legal relations; it is a cause of some change in those legal relations. To say that the fact or event is a cause (or condition) of the change does not mean that it is the sole cause (or condition). It is merely one of the group of factors that is necessary to produce the change.[144]

Another renowned contract law scholar defines condition as:

> [A]n act or event, other than a lapse of time, that, unless excused, must occur before a duty to perform a contractual promise arises (condition precedent), or that discharges a duty of performance that has already arisen (condition subsequent). This definition covers both conditions precedent and conditions subsequent and suggests the basis for distinction. It also retains the idea that the duty of performance is affected.[145]

It is thus clear that a condition is “an event that must occur before performance of a contractual duty becomes due” and “[i]n general, a party whose duty is conditioned on such an event is not required to perform unless the event has occurred.”[146] Moreover, this explanation is also consistent with the Restatement (Second) of Contracts, which states that a condition is “an event, not certain to occur, which must occur, unless its non-occurrence is excused, before performance under a contract is due.”[147]

On the other hand, a “covenant” is defined as a “promise,” and its legal consequences are very different from those of a condition:

[144] CATHERINE M.A. MC CAULIFF, CORBIN ON CONTRACTS: CONDITIONS, 10 (Joseph M. Perillo, ed., LEXIS Law Publishing 1999).

[145] JOSEPH M. PERILLO, CALAMARI & PERILLO ON CONTRACTS, 361 (Thomson Reuters, 6th ed. 2009).

[146] *See* E. ALLEN FARNSWORTH, FARNSWORTH ON CONTRACTS, 413 (Aspen Publishers, 3rd ed. 2004).

[147] *See* RESTATEMENT (SECOND) OF CONTRACTS, § 224 (1981).

> While failure to perform a promise, unless excused, is a breach, failure to comply with an express condition is not a breach . . . . One cannot be liable for breach of contract unless one breaches a promise . . . . Often, however, it is difficult to interpret whether particular language creates a promise or a condition. It is a matter of the intention of the parties, and all the rules of contract interpretation apply . . . . In a borderline case, the courts prefer the interpretation that particular language creates a promise rather a condition.[148]

Contract scholars have also offered guidance on how to distinguish between a condition and a covenant. These scholars offer an approach whereby one first asks whether the expression was intended to be an assurance by Party A to Party B that Party A would render some performance in the future that Party B could rely on. If this is the case, then the expression is a promise that the specified performance will occur. In the alternative, you must ask whether the expression at issue was intended to one party's duty conditional and dependent on some performance by the other party (or on some other fact or event). If this is the case, then specified performance is a condition of duty, but no one has promised that the performance will occur.[149] These contract scholars suggest that:

> It is not difficult to draw the logical distinction between a promise that a specified performance will be rendered, and a provision that makes a specified performance a condition of the legal duty of a party who promises to render another performance. The first creates a legal duty in the promisor; the second limits and postpones a promisor's duty. Often the contracting parties do not make this logical distinction and therefore so word their agreements as to make interpretation difficult. When such is the case, the court is free to give the contract the "construction" that appears to be the most reasonable and just.[150]

[148] PERILLO, *supra* note 145, at 365-66.

[149] MC CAULIFF, *supra* note 144, at 27.

[150] *Id.* at 27.

Based on these definitions, a condition is an action or event that changes the legal relationship and the duties between the parties to a contract because it directly affects the duty of performance under the contract. A condition involves some action that either must occur before the other party's performance under the contract is due (condition precedent) or else discharges the duty of performance that has already arisen (condition subsequent).[151] On the other hand, a covenant is a promise from one party under the contract to the other, which if broken, results in a breach of the contract. The main difference between a condition and a covenant is that a condition is an act or event upon which the other party's duty to perform under the contract is dependent, while a covenant is merely a promise from one party to the other that the specified performance will take place.[152]

For example, under the "Source Code License" section of the MPL, the initial developer and the contributor grant the downstream user certain rights, but the language of the license makes it clear that these grants are not necessarily dependent on the downstream user adhering to any requirements listed in the license.[153] Specifically, under the "Distribution Obligations" section of the license, the language indicates that downstream users must include a copy of this license with every copy of the source code they distribute and that any modifications that the downstream users contribute must be available in source code format.[154]

The language of this license, however, does not make the initial

[151] *See* PERILLO, *supra* note 145, at 361. For example, under the GPL v.2, when a downstream user redistributes his source code under the terms of this same license, he is discharging his duty of performance imposed by the condition subsequent (the condition being that he redistribute the code by adhering to the requirements written into the license).

[152] *See generally* HOWARD O. HUNTER, MODERN LAW OF CONTRACTS § 10.1 (1993) (explaining that "[i]f the failure of a condition predates performance, the contract never comes into existence . . . . The failure of a condition precedent is not a breach of contract because the contract, and the concomitant obligations, do not come into existence until satisfaction of the condition precedent. If the condition fails to occur after performance has begun, the contract ceases to exist.").

[153] Mozilla.org, *Mozilla Public License Version 1.1*, http://www.mozilla.org/MPL/MPL-1.1.html (last visited Apr. 29, 2011).

[154] *Id.*

developer's grant or the contributor's grant dependent on the downstream user adhering to these provisions. The use of the word "must" indicates that there is an obligation on the part of the downstream user to adhere to these restrictions. The provisions contained in these grants are not contingent upon the downstream user's performance of his obligations under the "Distribution Obligations" section. This stands in stark contrast to the GPL v.2, where the drafters made it explicit that further copying, modification, or distribution of the code is dependent on the downstream user adhering to the stated requirements.[155]

The common law defines covenants and conditions and draws a distinction in much the same way as do contract law scholars. For example, New York courts found the distinction between a condition and a covenant is based on the intent of the parties and the language used, and aided by reference to all the circumstances surrounding the parties at the time of the execution of the document.[156] New York courts have also indicated that certain words such as "upon condition" and "provided always" are commonly used to create a condition.[157]

Other states have also recognized that the language used in a contract may determine whether the contract contains a covenant or a condition. Courts have stated that while the use of the word "provided" ordinarily denotes a condition, the intention as discerned from the whole document may override such technical meaning, creating a covenant instead.[158] Thus, while "provided that" normally denotes a condition, if there is other contradictory language in the contract, then courts will look at all the language contained in the contract as a whole to determine the meaning of the words.[159]

In California, courts have recognized that a condition is a qualification and is created by mutual agreement of the parties and is binding upon both, while a covenant is an agreement of the

[155] GNU Operating System, *GNU General Public License Version 2*, http://www.gnu.org/licenses/gpl-2.0.html (last visited Apr. 29, 1011).

[156] Rockwell v. Utz, 139 N.Y.S. 529, 530 (1913).

[157] Tallmann v. Coffin, 4 N.Y. 134, 134 (1850).

[158] Newton v. Vill. of Glen Ellyn, 27 N.E.2d 821, 823 (Ill. 1940); Williams v. Johnson, 143 S.W.2d 738, 739 (Ky. 1940).

[159] *Newton*, 27 N.E.2d at 823; *Williams*, 143 S.W.2d at 739.

covenantor (promisor) only.[160] These courts have further stated that "the entire context, and not a single clause, will determine the intention of the parties."[161] Other jurisdictions have expressly recognized that "a covenant differs from a condition, both in the language that constitutes it, and in the consequences that follow from a breach."[162] While the breach of a covenant merely gives the covenantee (promisor) the right to sue, the breach of a condition may terminate the contract all together.[163]

Courts have also consistently stated that the intention of the parties generally determines whether a covenant or condition exists, and determining the intention of the parties is a matter of legal interpretation by the courts.[164] In cases where the intent of the parties is ambiguous, courts generally favor an interpretation that finds a covenant rather than a condition. This is likely because finding a condition could effectively undermine the existence of the contract.[165]

Using these definitions, the next section will interpret the language of various open-source licenses.

B. Applying the Definitions of "Condition" and "Covenant" to the Language of Open-Source Licenses

1. Applying the *Jacobsen* Reasoning to the GPL v.2

[160] Moe v. Gier, 2 P.2d 852, 855 (Cal. Dist. Ct. App. 1931).

[161] *Id.*

[162] *See* Bartell v. Senger, 155 A. 174, 176 (Md. 1931).

[163] *See id.*

[164] *See* Hawley v. Kafitz, 83 P. 248, 249-50 (Cal. 1905).

[165] *See generally* Oregon & Cal. R.R. v. United States, 238 U.S. 393, 420 (1915); L.A. Univ. v. Swarth, 107 F.798, 802-03 (9th Cir. 1901); Mathis v. Mathis, 83 N.E. 2d 270, 273 (Ill. 1949); Fed. Land Bank of Louisville v. Luckenbill, 13 N.E. 2d 531, 533 (Ind. 1938); Bd. of Councilmen of Frankfort v. Capital Hotel Co., 224 S.W. 197, 199 (Ky. 1920); Dixon v. Milling, 59 So. 804, 804 (Miss. 1912); In re Gaffers' Estate, 5 N.Y.S. 2d 671, 678 (N.Y. App. Div. 1938); *see also* Priddy v. Sch. Dist. No. 78, Cotton Cnty, 219 P. 141, 141-42 (Okla. 1923); Commmonwealth ex rel. Bard v. Del. Div. Canal Co., 1 A.2d 672, 675 (Pa. 1938); Hughes v. Donlon, 261 S.W. 960, 965 (Tenn. 1924); Weiss v. Claborn, 219 S.W. 884, 887 (Tex. Civ. App. 1920); Shreve v. Norfolk & W. Ry. Co., 64 S.E. 972, 973 (Va. 1909); Engel v. S. Penn. Oil Co., 146 S.E. 385, 388 (W.Va. 1928).

The Preamble to GPL v.2 not only contains an explanation of the philosophy behind the license, it explicitly states that the license includes restrictions that prevent users from denying each other the rights granted under the license.[166] It further states that there are certain responsibilities for users if the users want to distribute or modify copies of the software.[167] The Preamble also states that "the precise terms and conditions for copying, distribution, and modification follow."[168] Consequently, the force of the language contained in the Preamble and the use of the word "condition" make it clear that the drafters of this license intend the provisions to be conditions that—if not adhered to—would change the legal relationship between the parties to the license. Here, the upstream user would not need to make his source code available unless the downstream user follows the requirements of the license. The downstream user would not be able to obtain the source code in the first place unless the upstream user made the code available subject to the requirements of the license.

Moreover, the last sentence of the Preamble suggests that the drafters of GPL v.2 view the following language as conditions of the license.[169] Thus, as the Federal Circuit found with the Preamble of the Artistic License in *Jacobsen*,[170] the Preamble of GPL v.2 also indicates that the license is intended to create conditions under which the software can be copied.

As listed throughout the "Terms and Conditions For Copying, Distribution And Modification" section of the license, the provisions contained in this section are contingent upon each other and their application to the user is also contingent upon the user adhering to terms. This is evidenced by Section 1 of the Terms and Conditions, which explicitly states that users may copy, distribute, or modify copies of the source code as they receive it or that they may distribute a derivative work based on the source code, "provided that" the users conspicuously and appropriately publish on each copy a copyright

[166] GNU Operating System, *GNU General Public License Version 2*, http://www.gnu.org/licenses/gpl-2.0.html (last visited Apr. 29, 1011).

[167] *Id.*

[168] *Id.*

[169] *Id.*

[170] *Jacobsen,* 535 F.3d at 1381.

notice and disclaimer of warranty and that the users follow the other requirements listed under subsection 2.[171] In other words, a user's ability to copy and distribute copies of the source code verbatim is contingent upon conspicuous and appropriate publication of the copyright notice and a disclaimer of warranty. If users do not adhere to this provision, they do not have the right to copy and distribute copies of the source code.

Similarly, under Section 3, users may copy and distribute the program or a work based on the source code under Section 2 in object code or executable form under Sections 1 and 2 "provided that" they also do one of the following things listed Section 3.[172]

In addition to the "provided that" language contained throughout GPL v.2,[173] there is strong language contained in Section 4, which explicitly states that a user may not copy, modify, sublicense, or distribute the program except as provided under the license and that any attempt to do otherwise will automatically terminate the user's rights under the license.[174] This "automatic termination" provision is particularly important because it makes it unequivocally clear that failing to abide by the conditions results in a termination of the license. Since termination of the license would expose a user to potential copyright liability if he uses any code licensed under GPL v.2, this provision is extremely important.

Furthermore, as stated in Section 5, "nothing else" grants the downstream user the permission to modify or distribute the program or its derivative work. The upstream user granting permission to modify or distribute the program is dependent upon the downstream user abiding by the terms and conditions. Thus, this provision is a condition subsequent; the downstream user discharges his duty of performance by adhering to the terms of the license.

Another example of the conditions contained in GPL v.2 appears in Section 6 of the license. Section 6 states that the recipient of the program or any derivative work based on the program automatically receives a license from the original licensor to copy, distribute, or

[171] GNU Operating System, *GNU General Public License Version 2*, http://www.gnu.org/licenses/gpl-2.0.html (last visited Apr. 29, 1011).

[172] *Id.*

[173] *Id.*

[174] *Id.*

modify the program “subject to these terms and conditions, which indicates that receiving the program or the derivative work is dependent on the listed terms and conditions.[175] Consequently, based on the Preamble of the license, the use of the words “provided that,” the inclusion of the automatic termination provision, and the language contained in sections 5 and 6 of the license, suggest that the provisions contained in GPL v.2 are conditions and not covenants. Violating these provisions causes the user to exceed the scope of the license and exposes the user to copyright liability, as occurred in *Jacobsen*.

2. Applying the *Jacobsen* Reasoning to the GPL v.3

GPL v.3 contains several provisions that are similar in language and effect to the conditions contained in GPL v.2. For instance, GPL v.3 contains a Preamble that includes both the philosophy behind the license and an expression of the drafters’ intention that license users have certain responsibilities if they choose to distribute copies of software pursuant to the terms of the license or if they modify copies of software.[176] GPL v.3 explains the steps that developers that use this license have taken to protect users’ rights and also states the protections that the license has implemented to protect developers.[177]

It further states at the end of the Preamble that “the precise terms and conditions for copying, distribution, and modification follow,” and like GPL v.2, it also labels the following section with the caption “Terms And Conditions,” reflecting the drafters’ intent that the following provisions are to be interpreted as conditions for use of the license.[178]

Importantly, the language used throughout the GPL v.3 is similar to the previous version of the license in that it too contains what the *Jacobsen* court would consider to be “traditional contract language” denoting a condition.[179] Specifically, GPL v.3 uses the phrases

[175] *Id.*

[176] GNU Operating System, *GNU General Public License Version 3*, http://www.gnu.org/licenses/gpl-3.0.html (last visited Apr. 29, 1011).

[177] *Id.*

[178] *Id.*

[179] Jacobsen v. Katzer, 535 F.3d 1373, 1381 (Fed. Cir. 2008).

"provided that the stated conditions are met," "provided that," "solely under the conditions stated below," and "provided that you also meet all of these conditions" (with the four conditions listed below) to refer to what the drafters of the license clearly intended to be treated as conditions of the license.[180] While GPL v.3 retains many of the same conditions contained in GPL v.2, the new version of the license also contains additional provisions that may or may not be considered conditions under *Jacobsen*. Consequently the focus of this section of this Article will be to examine these new provisions and determine whether they would constitute conditions under *Jacobsen*.

The first new provision contained in GPL v.3 is an Anti-DRM provision contained in Section 3, which states that no covered work shall be considered an effective technological measure under any applicable law that fulfills Article 11 of the WIPO Copyright Treaty (which in the U.S. would be the DMCA). Under the DMCA, no one is permitted to circumvent a technological measure that effectively controls access to a protected work.[181] The DMCA defines what it means to circumvent such a technological measure and states what constitutes an effective technological measure.[182] Section 3 of GPL v.3 indicates that any software licensed under it that is used to implement Digital Rights Management (DRM) shall not be deemed an effective technological measure.[183] Thus, according to the license's drafters, if a party implements a DRM system using software licensed under GPL v.3, that system will not be protected by the DMCA (since it will not be an effective technological measure).[184]

This provision, however, is unlikely to have the effect the drafters desired. First, the intent of the drafters seems to run contrary to federal law, in this case the DMCA. The legal definition of effective technological measures as stated in the DMCA has nothing to do with

[180] GNU Operating System, *GNU General Public License Version 3*, http://www.gnu.org/licenses/gpl-3.0.html (last visited Apr. 29, 1011).

[181] *See* 17 U.S.C. § 1201(a)(1)(A) (2006).

[182] *See* 17 U.S.C. § 1201(a)(13)(A) to (B) (2006).

[183] GNU Operating System, *GNU General Public License Version 3*, http://www.gnu.org/licenses/gpl-3.0.html (last visited Apr. 29, 1011).

[184] Brett Smith, *A Quick Guide to GPL v3*, GNU OPERATING SYSTEM, http://www.gnu.org/licenses/quick-guide-gplv3.pdf.

GPL v.3 licensed software. Therefore, a court is unlikely to hold that a DRM system is not effective simply because it happens to contain GPL v.3 licensed software. Essentially, courts are unlikely to interpret this provision in the manner that the drafters intended since the drafters' interpretation would contravene the DMCA. What then will be the effect of this provision?

The likely result of this provision will be that if someone builds a DRM system that contains GPL v.3 licensed code and the DRM system that contains the code is an effective technological measure, then it will not fall under the terms of the license and could potentially result in a violation of the DMCA as well as copyright infringement. Because the license states that "no covered work shall be part of an effective technological measure," one can argue that anyone who implements an effective technological measure using software licensed under GPL v.3 no longer enjoys its protection. This then begs the question: is this provision a condition or a covenant under *Jacobsen*?

Unlike the other provisions of GPL v.3 previously discussed, this section clearly does not contain traditional contract language denoting conditions, such as the words "provided that," nor does this provision contain any covenants, since there are no promises mentioned anywhere in the provision. While this provision standing alone is neither a condition nor a covenant, in the context of the entire license, a court is likely to interpret it as a condition under *Jacobsen*. This is because the termination provision of the license (contained in Section 8) makes it clear that no one may propagate or modify a covered work except as provided under the language of the license, and any attempt to do otherwise will result in an automatic termination of the license.[185] This termination provision makes it evident that a downstream user's ability to propagate and modify code covered under this license is contingent upon following the terms and condition of the license.

In addition, Section 9 of the license also states that while a user need not accept this license in order to receive or run a copy of the program, propagation or modification of any covered work is only

[185] GNU Operating System, *GNU General Public License Version 3*, http://www.gnu.org/licenses/gpl-3.0.html (last visited Apr. 29, 1011).

possible under the license.[186] The provision further states that if a user does not accept this license, such actions will result in copyright infringement, so that by modifying or propagating a covered work, a user is implicitly accepting the terms of the license.

Another new provision of GPLv.3 is the patent grant provision contained in Section 11, which provides a non-exclusive, worldwide, royalty-free patent license from each contributor to each downstream user to make, use, sell, offer for sale, import, otherwise run, modify, and propagate the contents of the contributor's version.[187] Before exploring the features of the patent grant provision, it is important to understand that this provision applies only to the "contributor" version of the covered work, not any new features added by downstream users.[188]

Under this patent grant, if a user conveys a covered work with actual knowledge that but for this patent license, conveying the covered work or a recipient's use of the covered work would infringe on a patent, and the user does not make the source of the work available to others to copy, the user must take one of three remedial measures listed or else lose the protection of the license.[189] This provision further states that if a user redistributes a covered work knowingly relying on the patent license, then he must also shield downstream users against patent infringement claims or else deprive himself the benefit of the patent license.[190] Section 10 also references the patent grant provision and states that asserting a patent infringement claim terminates the license, as indicated in Section 8.[191]

[186] *Id.*

[187] *Id.*

[188] *See id.* (defining a "contributor" as "a copyright holder who authorizes use under [the GPL v.3] license of the Program or a work on which the Program is based. The work thus licensed is called the contributor's 'contributor version.'").

[189] GNU Operating System, *GNU General Public License Version 3*, http://www.gnu.org/licenses/gpl-3.0.html (last visited Apr. 29, 1011) (stating that the user must either (1) make the Corresponding Source available or (2) arrange to deprive himself of the of the benefit of the patent license for the particular work, or (3) arrange (in a way that is consistent with the terms of the license) to extend the patent license to downstream recipients).

[190] *Id.*

[191] *Id.*

These provisions collectively serve as conditions under *Jacobsen* even though they do not include any of the traditional contract language of conditions. The termination provision of Section 10 functions as a condition. The patent grant provision of Section 11 and the "no litigation" provision of Section 10 also function as conditions because when considered in the context of the termination provision in Section 8, these provisions effectively terminate the rights of a user who does not adhere to the license terms.[192] The user's ability to avail himself to the benefits of the license is therefore dependent on the user following the terms of these provisions.

The language contained in Section 9 also supports finding conditions because the language explicitly states that failure to accept this license results in copyright infringement liability if a user engages in conduct that is only protected by the license.[193] Therefore, it is evident that acceptance of all the terms of the license (including the patent grant) is a condition of the user being able receive the benefit of the license.

Finally, GPL v.3 also contains a new and explicit provision addressing violations of the license, as found in Section 8. This provision not only states that failing to adhere to the terms of this license will automatically terminate the license, but it also allows for provisional and permanent reinstatement of the license if the user ceases the violation.[194] This feature is also likely a condition since it makes it clear that a user's ability to benefit from the license is dependent on his compliance with the license terms. When the user fails to comply, he loses the license protection; when he is once again in compliance, he is able to acquire the protection of the license. The quid-pro-quo nature of the termination provision makes it clear that this particular provision not only makes the other provisions of the license function as conditions, but is in and of itself a condition of the license as well. Consequently, the provisions of GPL v.3, while not necessarily phrased in the traditional language of conditions, effectively function as such.

[192] *Id.*

[193] GNU Operating System, *GNU General Public License Version 3*, http://www.gnu.org/licenses/gpl-3.0.html (last visited Apr. 29, 1011)

[194] *Id.*

3. Applying the *Jacobsen* Reasoning to the Apache License v.2

Unlike the GPL v.2 and the GPL v.3, the Apache License does not contain a Preamble that establishes the intentions of the drafters of the license as unequivocally as the Preamble to the GPL v.2 and GPL v.3. However, like the language of GPL v.2, the language of the Apache License does contain what the *Jacobsen* court considers traditional contract language that denotes the existence of a condition.

The Apache License defines the "Terms And Conditions For Use, Reproduction, And Distribution," indicating that the drafters intended for the provisions to be conditions of the license.[195] The drafters explicitly labeled these provisions "terms and conditions,"[196] and most of the provisions contained in the Apache License are conditions, as illustrated below.

First, the definitions section of the license defines "license" as referring to the terms and conditions for use, reproduction, and distribution as defined by Sections 1–9 of the license.[197] Second, several other provisions use the type of language that the *Jacobsen* court would consider to be conditions. For example, Section 2's "Grant of Copyright License" makes it clear that each contributor is granting the user "a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare [d]erivative [w]orks of, publicly display, publicly perform, sublicense and distribute the [covered] [w]ork and [d]erivative [w]orks in [s]ource or [o]bject form."[198] However, this grant is "subject to the terms and conditions" of the license.[199] Similar language appears in Section 3's "Grant of Patent License."[200]

Moreover, Section 4's "Redistribution" provision makes it clear that a user may reproduce and distribute copies of the covered work or derivative works in any medium and with or without modification

[195] The Apache Software Foundation, *Apache License, Version 2.0*, http://www.apache.org/licenses/LICENSE-2.0.html (last visited Apr. 29, 2011).

[196] *Id.*

[197] *Id.*

[198] *Id.*

[199] *Id.*

[200] *Id.*

and in source or object code form "provided that" the user "meet the following conditions," and then lists the four conditions that users must follow.[201] Section 4 further states that users may add their own copyright statements or their own additional or different license terms and conditions for use, reproduction, or distribution of the user's modifications or for any such derivative works as a whole "provided that" the user's use, reproduction, and distribution of the work "otherwise complies with the conditions stated in this license."[202] Finally, Section 5 provides that unless the user explicitly states otherwise, any contribution that is submitted for inclusion in the covered work by the user to the licensor shall be under the terms and conditions of the license and without any additional terms or conditions.[203]

Based on these examples, it is clear that the provisions of the Apache License are in fact conditions, though they are not as unequivocal as those contained in GPL v.3. While the Apache license may be more permissive than GPL v.2 and does not contain the same explicit automatic termination provisions found in GPL v.2, its provisions are as legally binding as conditions as those contained in GPL v.2.

There are some important distinct features of the Apache License, however, that affect the condition versus covenant analysis. First, despite the "Terms And Conditions" label contained in the caption of the license, not everything that follows the caption is in fact a condition. As is the case with all open-source licenses, some language of the Apache License is neither a condition nor a covenant. That is, certain provisions merely serve as explanations of the drafters' intent and views.

For example, Section 8 of the license contains a limitation of liability provision that essentially states that no contributor under the license shall be liable to the downstream user for damages arising as a result of the license or out of the inability to use the work.[204] The way this provision is phrased, it is clear that it is neither a condition nor a

[201] *Id.*

[202] The Apache Software Foundation, *Apache License*, *Version 2.0*, http://www.apache.org/licenses/LICENSE-2.0.html (last visited Apr. 29, 2011).

[203] *Id.*

[204] *Id.*

covenant since it neither makes one party's duty under the license dependent on another party's performance or contains a promise of any kind from one party to another. Rather, this provision merely states that a user cannot hold a contributor liable for damages. What is left unclear by the language of this provision is what, if any, consequences flow from a user trying to hold a contributor liable under the license. The provision is unclear as to whether the result of such action would be a dismissal, a termination of the license, or some other result.

Similarly, the drafters worded Section 3's "Grant of Patent License" in such a way that it raises questions as to its implications. As noted above, it contains the "subject to the terms and conditions" contract language indicative of conditions.[205] It also contains an automatic termination provision that states that if a user institutes a patent litigation action against any entity alleging that the work or contribution incorporated within it constitutes patent infringement, then any patent license granted to the user under this provision shall terminate as of the date of filing of such an action.[206] This provision begs the question why the other provisions of the license do not mention automatic termination, as contained in the GPL?

If in fact a user is subject to all the "terms and conditions" contained in the license, then a user's failure to abide by any of these terms and conditions would likely result in a termination of the license. The right to terminate a license, when tied to a license grant, would make the "terms and conditions" of the grant "conditions" under contract law. Therefore, stating in a license that a breach of a particular provision of the license terminates the license would in fact lead to the conclusion that the violated term was a condition of the license. However, this is not expressly stated in the Apache License. To the contrary, the only provision of the Apache License that even mentions an automatic termination is the patent grant provision.[207]

Thus, the drafting of this provision is confusing because it is unclear what purpose this automatic termination provision serves. Is the provision meant to indicate that this is the only provision whose

[205] The Apache Software Foundation, *Apache License*, *Version 2.0*, http://www.apache.org/licenses/LICENSE-2.0.html (last visited Apr. 29, 2011).

[206] *Id.*

[207] *Id.*

violation will result in a termination of the license, or is it merely redundant because violating any term or condition would also result in termination of the license? Is the provision simply intended to specify a time element to indicate that the point at which the license terminates is the date on which the patent litigation action is filed?

Since the language used throughout the license is indicative of the conditions contained in the license, why did the drafters of this license fail to mention the automatic termination provision with respect to these other conditions, as is the case with the GPL? While it is likely that under *Jacobsen* the provisions of the Apache License will be held to be conditions, there are a few aspects of this license that may not fit as easily into the condition category as do the provisions of GPL v.2.

4. Applying the *Jacobsen* Reasoning to the BSD License

The BSD License is significantly shorter than any other open-source license examined in this Article. As a result, applying the reasoning of *Jacobsen* is more straightforward. The BSD License contains one main license grant that states that redistribution and use in source and binary forms with or without modification are permitted "provided that" the user meets three conditions: (1) when a user redistributes source code, the redistributed version must contain the copyright notice, the list of conditions, and the attached disclaimer; (2) when a user redistributes code in binary form, he must include a reproduction of the copyright notice, the list of conditions, and the attached disclaimer in the materials provided with the distribution; and (3) the user may not use the name of the organization and the name of the contributor for endorsement purposes without specific prior written permission.[208] Thus, based on the use of the words "provided that" and based on a literal interpretation of *Jacobsen*, the BSD License grant provision contains a condition.

Unlike the other licenses examined in this work, the BSD lacks the additional supporting elements that are present in the GPL or the Apache License. Specifically, each of those licenses contains either a

[208] Open Source Initiative, *The BSD License*, http://www.opensource.org/licenses/bsd-license.php (last visited Apr. 29, 2011).

Preamble or other language throughout the license that indicates the specific intent of the drafters to provide the license grant to only those individuals or entities that abide by the conditions contained in the license. Those licenses also contain at least one termination clause that expressly states that a violation of that provision will result in a termination of the user's rights under the license. These additional provisions in the GPL and the Apache License strengthen the conclusion that these licenses contain conditions rather than covenants.

Consequently, since the BSD License lacks these additional terms found in the GPL and the Apache License, there is only a small chance that another court that declines to follow the reasoning used by the Federal Circuit in *Jacobsen* (or which interprets contact law language differently than the Federal Circuit) will hold that the language of the BSD License is not a condition.

5. Applying the *Jacobsen* Reasoning to the Mozilla Public License

Similar to the Apache License, the MPL does not contain a Preamble. Rather, it contains a list of definitions of terms followed by subsections, starting with Section 2's "Source Code License," which includes an "Initial Developer Grant" and a "Contributor Grant."[209] The "Initial Developer Grant" expressly states that the user is being granted a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims, and then lists what the license grant actually allows the user to do (i.e. use, reproduce, modify, display, perform, sublicense and distribute the original code or portions of it with or without modifications and/or as part of a larger work).[210]

Under the "Contributor Grant," the MPL similarly states that subject to third party intellectual property claims, each contributor is granting the user a world-wide, royalty free, non-exclusive license and then lists what the user is permitted to do under the license grant (use, reproduce, modify, display, perform, sublicense and distribute the modifications created by the contributor either with or without

[209] Mozilla.org, *Mozilla Public License Version 1.1* http://www.mozilla.org/MPL/MPL-1.1.html (last visited Jan. 12, 2011).

[210] *Id.*

modifications as a covered work and/or as part of a larger work, etc.).[211]

The drafters worded these provisions in such a way that the license grants are not dependent on any actions by the user. Unlike the GPL v.2, GPL v.3, Apache, and BSD licenses discussed above, these grant provisions are independent of any actions of the user and are not in any way conditioned upon any action or inaction by the user.[212] Rather, under these grant provisions the user is merely obtaining permission outright to engage in the activities that the grants specifically outline.[213]

Thus, while the user may be implicitly promising to adhere to the license terms by using code that is made available through this license, the user's ability to use the code is not contingent upon following the license terms. The user may face repercussions for not adhering to the license terms, but the user's ability to obtain the license is separate and independent from these obligations. Certain features of the drafting of the MPL illustrate this point.

First, the language used throughout the license indicates that the user "must" do certain things (i.e. the user must include a copy of the license with every copy of source code he distributes), "may" do certain other things (i.e. the user may include additional documents offering the additional rights described in Section 3 of the license), and "may not" do certain things (i.e. the user may not offer or impose any terms on any source code version that alters or restricts the applicable version of the license or the recipients' rights hereunder).[214]

Unlike the previous licenses examined in this Article, however, the MPL does not contain any terms or provisions that would qualify as a condition under *Jacobsen*. To the contrary, the use of the words "must," "may," and "may not" suggest that this license contains covenants or promises from both the upstream user and the downstream user. The upstream user's promises are contained in the grant provisions of the license while the downstream user's promises are contained throughout the license and are implicit in the user's

[211] *Id.*
[212] *Id.*
[213] *Id.*
[214] *Id.*

decision to avail himself to the code made available to him under this license. His promises are also evident in every section where the license states what the downstream user "must" do, "may" do, and "may not" do.

Under traditional contract law, such language would qualify as covenants from the upstream user to the downstream user and vice versa.[215] Furthermore, the language contained in the "Termination" provision in Section 8 of the license is the traditional contract language of covenants, as defined by contract law.[216] For example, Section 8.1 states that if the user fails to comply with the terms of the license and fails "to cure such breach within 30 days of becoming aware of the breach," the license and the rights granted under it will terminate automatically.[217] This reference to "breach" and "curing the breach" is evidence of the existence of a covenant since, as contract scholars have explained, one breaches a covenant and not a condition.[218] Accordingly, the language of the MPL contains covenants not conditions, since the grant of rights under the license to downstream users is not conditioned upon what the downstream user does or does not do.

The net effect of the existence of these covenants, however, leads to the same result as if these terms were in fact conditions, because of the existence of the termination provision in Section 8 of the license. As mentioned in the discussion of the GPL, the existence of a termination provision in an open-source license is significant. While the drafters of an open-source license may phrase the license

[215] *See* MC CAULIFF, *supra* note 144, at 27 (explaining that, "The first step, therefore, in interpreting an expression in contract, with respect to condition as opposed to promise, is to ask oneself the question: Was this expression intended to be an assurance by one party to the other that some performance by the first would be rendered in the future and that the other could rely upon it? If the answer is yes, we found the expression to be a promise that the specified performance will take place.").

[216] *See* PERILLO, *supra* note 145, at 365-66 (explaining that, "The distinction between an express condition and a promise is critical. While failure to perform a promise, unless excused, is a breach, failure to comply with an express condition is not a breach . . . . One cannot be liable for breach of contract unless one breaches a *promise*.").

[217] Mozilla.org, *Mozilla Public License Version 1.1* http://www.mozilla.org/MPL/MPL-1.1.html (last visited Jan. 12, 2011).

[218] *See* PERILLO, *supra* note 145, at 365-66.

language in a manner that does not appear to be very restrictive, they can easily undo this permissiveness with the inclusion of a termination clause. This is the case with the MPL. The MPL language generally makes the license more permissive than the GPL. However, the MPL (like the GPL) contains a termination clause that actually undermines its permissiveness. While the covenants contained in the MPL indicate a certain degree of permissiveness in the license, Section 8.1's termination provision completely undermines this permissiveness.

In Section 8.1, the MPL explicitly states that the license and all the rights granted under it will automatically terminate if a user fails to comply with the terms contained in the license and if the user fails to cure such non-compliance within thirty days of becoming aware of it.[219] Unlike the rest of the license, this provision seems to function much like a condition because it effectively tells users that their rights under the license are actually dependent on following the terms of the license, and that failure to do so will in fact terminate the license and possibly expose the user to copyright liability if they use any code derived from the license. As with the conditions contained in the GPL, therefore, the termination of the MPL is conditioned upon something the downstream user does, despite what the other provisions of the license states.

Consequently, the MPL, while generally viewed as being more permissive than the GPL and containing covenants and not conditions unlike the GPL, is on the same legal footing as the GPL with respect to the enforcement of the provisions of the license. Although the MPL does not contain the strong conditional language used consistently throughout the GPL, the one automatic termination clause that is contained in the MPL ultimately leads to the same outcome for users who violate its terms. That is to say, a user who violates the terms of the MPL will have his license terminated, and if he continues to use any code derived from the license, he will be exposed to potential copyright liability.

Having applied the reasoning of *Jacobsen* to a series of licenses, it is clear that while the reasoning provided by the Federal Circuit

[219] Mozilla.org, *Mozilla Public License Version 1.1* http://www.mozilla.org/MPL/MPL-1.1.html (last visited Jan. 12, 2011).

may be helpful in determining whether the language contained in an open-source license is a condition or a covenant, this analysis is more complex than merely looking for the traditional contract language denoting conditions or covenants in the language of each license.

Accordingly, the last section of this Article offers some proposals for how drafters of an open-source license can write their licenses in such a way that removes ambiguity regarding the existence of conditions and covenants.

IV. PROPOSALS

While the Federal Circuit's reasoning in *Jacobsen* is the only existing U.S. law interpreting conditions and covenants in open-source licenses, this decision is insufficient for providing a "bright line rule." Consequently, there are certain steps that drafters of open-source licenses can take to help make their intentions clear to open-source license users and courts.

First, drafters of open-source licenses should use clear, unequivocal, and unambiguous language to indicate that they intend the provisions contained in their license to be interpreted as conditions. As the *Jacobsen* court explained, the use of certain phrases in an open-source license will lead to the conclusion that the license contains conditions and not covenants based on traditional contract law principles.[220] These phrases include such words as "provided that,"[221] "subject to," "contingent upon," "dependent upon," "unless," and "conditions" to refer to the provisions contained in the license. While the appearance of these phrases in a license does not automatically lead to the conclusion that the license contains a condition, their appearance certainly strengthens the argument that the drafters intended these provisions to be interpreted as such.

Second, drafters of open-source licenses who wish to have their licenses terms interpreted as conditions should make this intention clear by including a Preamble section in the license. The Preamble section should state not only the philosophy of the drafter but should also state his intention in drafting the license. It should specify

[220] *See* Jacobsen v. Katzer, 535 F.3d 1373, 1381 (2008).
[221] *See id.*

whether the terms and provisions are only made available to users contingent upon their adherence to these terms or whether these terms are independent provisions of the license. The Preamble should also state the potential consequences if a user disregards or violates these provisions.

Third, drafters should incorporate into the license the type of termination and copyright liability language appearing in the GPL, where the drafters not only use the traditional contract language for conditions, but also fortify this use by stating unambiguously that violating the provisions of the license will result in both termination of the license and potential exposure to copyright infringement liability. Such termination clauses, especially when used in conjunction with the concept of potential copyright infringement liability, make it clear that these provisions are conditions, which if violated, terminate the users' rights under the license. The specific mention of copyright infringement liability further strengthens an interpretation in favor of conditions, since violating a covenant will not terminate the license (it only gives rise to a cause of action for breach of the license) nor will it expose the user to a potential action for copyright infringement.

Fourth, drafters of open-source licenses should include a choice of law clause of the type provided in the MPL.[222] As illustrated in *Jacobsen*, state contract law is another tool that courts use in interpreting the language of an open-source license. Depending on which state's law is applied, there is always the possibility that a court may interpret the same contract language differently in different jurisdictions. This is really more of a concern when considering interpretation of contract law by foreign jurisdictions, whose contract law may be vastly different from contract law in the United States. By listing in the license itself what state's law will be applied, the license gives notice to users that that state's contract law will dictate whether the terms contained are conditions or covenants, which in turn will dictate whether the user is exposing himself to potential copyright liability by violating the terms of the license.

Finally, since courts will interpret open-source licenses by trying

[222] Mozilla.org, *Mozilla Public License Version 1.1* http://www.mozilla.org/MPL/MPL-1.1.html (last visited Jan. 12, 2011).

to discern the intent of the parties and since ambiguous phrasing will lead to the conclusion that the terms contained are covenants and not conditions, drafters of open-source license need to decide what it is they hope to accomplish by drafting the license and how they want the license to be interpreted. Once they have decided that, they need to consult experienced licensing and contract attorneys to ensure that what they have drafted accurately reflects this intention.

If drafters write open-source licenses that leave any doubt regarding the intention, they are causing both unnecessary litigation and a potential rise in the proliferation of open-source licenses. So long as open-source licenses contain ambiguities, there will be a continued incentive to create new licenses. While drafters can phrase the language in these new licenses in a way that incorporates both the *Jacobsen* reasoning and the proposals suggested above, there is still the problem that these new licenses will not apply retroactively to code that has already been made available under a previous version or another open-source license.

V. POLICY CONSIDERATIONS

Several policy concerns arise when drafting and interpreting open-source licenses. Allowing drafters to manipulate the distinction between conditions and covenants through linguistic gymnastics (in order to unfairly extend copyright liability) may simply be the wrong outcome as a matter of public policy. Courts have held that there are limits to the notion of freedom of contract,[223] and the struggle between freedom of contract and the need to place limits on the freedom continues, as evidenced by Supreme Court jurisprudence surrounding the Commerce Clause of the United States Constitution.[224]

[223] *See generally* W. Coast Hotel v. Parrish, 300 U.S. 379 (1937) (holding that a state law establishing a minimum wage is constitutional because the United States Constitution allows for some limitations to be placed on freedom of contract where the limitation is placed to protect the health, safety, or welfare of the community).

[224] *See generally* United States v. Lopez, 514 U.S. 549 (1995) (holding that Congress' authority to enact laws pursuant to the Commerce Clause of the United States Constitution is limited and does not include the authority to regulate the carrying of handguns in a high school); *see also* United States v. Morrison, 529 U.S. 598 (2000) (holding that sections of the Violence Against Women Act are

Scholars have offered suggestions for drawing a principled distinction between conditions and covenants. For example, one view is that only conditions that touch on the exclusive rights under copyright should qualify as license conditions.[225] The benefit of this suggestion is that it limits the power of copyright licensors.[226] Another view is that a license condition is any condition that the parties insert in the license grant. This is based on the idea that the parties choose the label themselves, rather than allowing a court to decide the label for them.[227] The benefit of this view is that it is consistent both with the Federal Circuit's view in *Jacobsen*[228] and with the notion of freedom of contract.[229] The drawback is that it may give copyright licensors too much power.[230]

As a whole, the latter view reflects the better approach for several reasons. As mentioned above, this view is consistent with both the reasoning in *Jacobsen* as well as the notion of freedom of contract. It also forces drafters of open-source licenses to be more careful in how they draft their licenses, which in turn allows for more clarity in the licenses themselves. Moreover, allowing the parties to choose whether the terms are conditions or covenants encourages parties to enter into licenses, which in turn promotes innovation.[231]

In the case of open-source licenses, this is especially true since open-source licenses allow developers to develop and improve software more quickly and allow people to obtain the benefit from the talent and expertise of programmers whose work they may not otherwise be able to access. Moreover, licensing law already has certain boundaries in place to protect against the expansion of copyright licensing abuses. For instance, canons of contract construction and contract formation (e.g., unconscionability),

unconstitutional because in enacting that law, Congress has exceeded its authority under the Commerce Clause).

[225] Gomulkiewicz, *supra* note 47, at 354.

[226] *Id.* at 355.

[227] *Id.* at 356.

[228] *See* Jacobsen v. Katzer, 535 F.3d 1373, 1378 (where the court noted that the use of a license contract allows copyright holders to control what can be done with their works).

[229] *See* Gomulkiewicz, *supra* note 47, at 356.

[230] *See id.*

[231] *Id.* at 357.

consumer protection laws, the misuse doctrine, antitrust law, and general public policy concerns all serve as boundaries on licenses.[232] Consequently, any policy concerns surrounding parties' choosing to insert conditions or covenants in their licenses are adequately addressed by other legal doctrines outside of licensing law. As suggested by the *Jacobsen* opinion, context maters.[233] In *Jacobsen*, the Federal Circuit suggests that the issue is one of contract interpretation.[234] That is, the correct approach is to look at the contract as a whole and determine what the individual phrases mean in the larger context of the contract as a whole.

Another policy consideration that arises when interpreting open-source licenses is the lack of understanding that often exists on the part of those who use software that has been distributed under the terms of an open-source license. Many individuals who use code derived from open-source code do not truly understand the terms and provisions contained in the license that enables them to use that code. Moreover, these individuals do not understand the legal consequences that flow from using open-source code. Thus, perhaps these individuals should not be held accountable for violating such provisions. However, there is a strong opposition that believes regardless of their lack of knowledge, individuals who choose to use source code made available under a specific open-source license are binding themselves to those provisions. The rationale here is that there are certain risks inherently involved when one uses open-source code or when one makes code available via an open-source license. The only thing that is evident regarding open-source license interpretation is the lack of clarity that actually exists in this area of law. Consequently, programmers, attorneys, and users of open-source code must read the terms of each license carefully before availing themselves to its benefits.

[232] *See id.* at 359.

[233] *See Jacobsen*, 535 F.3d at 1382 n.5 (citing to the example of author attribution and explains that the failure to provide attribution only triggers copyright infringement if the license so provides).

[234] *See id.* at 1381.

CONCLUSION

While many in the open-source community are pleased that the Federal Circuit has weighed in on the issue of the interpretation of open-source licenses in its decision in *Jacobsen*, it is clear that the *Jacobsen* decision is merely a guidepost, not a source of bright line rules for how to interpret conditions and covenants in open-source licenses. Consequently, courts and scholars are still left to wonder exactly how to interpret the language of an open-source license to draw this distinction. The interpretations of the open-source licenses contained in this Article are a starting point in this study.

Given the broad variation among open-source licenses, there is still room for debate as to what the provisions in these licenses mean. The ultimate answer to this question will be found in forthcoming judicial decisions following litigation of more open-source licenses. Until then, open-source scholars, attorneys, and the members of the open-source community must scrutinize and parse in great detail every phrase contained in these licenses to ascertain whether they adhere to the reasoning of *Jacobsen*.[235]

PRACTICE POINTERS

- When drafting an open-source license, the license drafter should use clear, unequivocal, and unambiguous language to indicate his intention that the provisions contained in the license should be interpreted as conditions. When possible, use the traditional contract language indicating conditions, which includes phrases such as “provided that,” “subject to,” “conditioned upon,” “dependent upon,” “unless,” and “conditions.”
- Include a preamble in the license, stating both the philosophy of the drafter and his intention in drafting the license. The preamble should specify whether the terms and conditions contained in the license are available to users contingent upon users’ adherence to these terms and conditions, or whether these terms and conditions are independent provisions of the license. The preamble should also state the potential consequences for users who disregard or

[235] *See id.* at 1380-82.

violate these provisions.

- The license drafter should include a termination clause as well as language stating that violating the provisions of the license will result in both termination of the license and potential exposure to copyright infringement liability.
- The drafter should include a choice of law clause in the license, indicating which jurisdiction's law will govern in the event of litigation.
- The license drafter should decide beforehand what he hopes to accomplish by drafting the license and how he wants the license to be interpreted. After the drafter has decided these matters (ideally before starting the drafting process), he should then consult an experienced licensing and contract attorney to ensure that what is drafted accurately reflects the drafter's intention.