FIFTY YEARS OF OPEN SOURCE MOVEMENT: AN ANALYSIS THROUGH THE PRISM OF COPYRIGHT LAW

V.K. Unni[*]

I. INTRODUCTION

The evolution of the software industry is a case study in itself. This evolution has multiple phases and one such important phase, termed open source, deals with the manner in which software technology is held, developed, and distributed.[1] Over the years, open source software has played a leading role in promoting the Internet infrastructure and thus programs utilizing open source software, such as Linux, Apache, and BIND, are very often used as tools to run various Internet and business applications.[2]

The origins of open source software can be traced back to 1964–65 when Bell Labs joined hands with the Massachusetts Institute of Technology (MIT) and General Electric (GE) to work on the development of MULTICS for creating a dynamic, modular computer system capable of supporting hundreds of users.[3] During the early stages of development, open source software had a very slow beginning primarily because of some preconceived notions surrounding it. Firstly, open source software was perceived as a product of academics and hobbyist programmers.[4] Secondly, it was thought to be technically inferior to proprietary software.[5] However, with the passage of time, the technical issues got relegated to the backside and issues pertaining to copyright, licensing, warranty, etc., began to be debated across the globe.[6]

[*] Professor—Public Policy and Management, Indian Institute of Management Calcutta; Thomas Edison Innovation Fellow (2016–17), Center for the Protection of Intellectual Property, George Mason University School of Law.

[1] Shawn W. Potter, *Opening Up to Open Source***,** 6 RICH. J.L. & TECH. 24, ¶ 1 (2000). *Also available at* http://jolt.richmond.edu/v6i5/article3.html (last visited Feb 24, 2016).

[2] *Id.* at ¶ 2.

[3] *See* HISTORY OF UNIX (NOV. 18, 2016, 2:05 PM), https://www.cs.duke.edu/csl/docs/unix_course/intro-2.html). ("MULTICS" is the abbreviation of MULTiplexed Information and Computing Service). *See also* Victoria Nemiah, *License And Registration, Please: Using Copyright "Conditions" to Protect Free/Open Source Software*, 3 N.Y.U. J. OF INTELL. PROP. & ENT. LAW 358, 362 (2014).

[4] Potter, *supra* note 1, at ¶ 2.

[5] *Id.*

[6] *Id.*

II. GENESIS OF COMPUTER PROGRAMMING

To get a holistic view of the legal issues arising out of open source software and its development, a brief introduction to software programming will be beneficial. In fact, the word "source" in open source refers to source code, which is the language that programmers use to communicate to computers and direct the machines to execute desired functions.[7] Scholars often refer to source code as the "holy grail" of the open source movement.[8]

Interestingly, programmers never communicate directly with computers, as the latter cannot understand any word in the source code.[9] Computers can only understand another type of language, typically referred to as object code or executable code, which is a binary language of consecutive ones and zeros used to control on/off switches in the computer hardware.[10] Programmers utilize numerous programs like C, BASIC, and JAVA to write the source code, which is very much readable and understood by humans. Source code for a computer program is written in a high-level language, such as C or FORTRAN. But if the computer has to perform the programmer's directions, then the source code needs to be converted into the computer-readable object code.[11] This process, called compilation, is usually carried out by standardized compiler software.[12]

A. Importance of Source Code

The differentiation between the source code and object code is a very important one in the context of open source.[13] This distinction also becomes the foundation stone for differentiation between traditional proprietary software and open source software. The most important difference between open source and proprietary software is the manner of development and distribution.[14]

[7] Mathias Strasser, *A New Paradigm in Intellectual Property Law? The Case Against Open Sources*, 2001 STAN. TECH. L. REV. 4. ¶ 8, (2001).

[8] Greg R. Vetter, *The Collaborative Integrity of Open-Source Software*, 2004 UTAH L. REV. 563, 578 (2004).

[9] *See Source Code*, THE TECH TERMS COMPUTER DICTIONARY (2007), *available at* http://techterms.com/definition/sourcecode (Accessed Feb. 5, 2016).

[10] Potter, *supra* note 1, at ¶ 7. *See also Executable Code*, PC MAGAZINE ENCYCLOPEDIA (2016), *available at* http://www.pcmag.com/encyclopedia/term/42842/executable-code.

[11] Christian H. Nadan, *Open Source Licensing: Virus or Virtue?* 10 TEX. INTELL. PROP. L.J. 349, 350 (2002).

[12] *Id.*

[13] *See* Potter, *supra* note 1, at ¶ 7.

[14] Matthew D. Stein, *Rethinking UCITA: Lessons From the Open Source Movement*, 58 ME. L. REV. 157, 160 (2006).

Commercial software developers have utilized the proprietary model from the early days and this involved keeping the source code secret while software was distributed in the object code form only.[15] When a program is distributed in object code format, the user cannot do anything other than run the program on a computer.[16] Most commercial software license agreements prevented the user/licensee from having access to the source code. The reason for this stemmed from the fact that distribution of software in source code form could result in reverse engineering of the first program by any other skilled engineer leading to misappropriation of its innovative features. This model of software distribution has been fiercely criticized by proponents of open source software who often compare the proprietary distribution model to selling a car with the hood welded shut.[17]

Even though a substantial segment of ordinary users are not concerned about the source code, there still exists a significant section of programmers, professionals and hobbyists who would be interested in knowing about the ideas and concepts behind the program by gaining access to the source code. Thus open source software owes its origins to a group of programmers who share their source code with each other and distribute their software along with the source code. The hallmark of the open source approach is built upon the principles of source code access and the ability to change and improve the source code.[18]

Thus in the field of software two different licensing models operate.[19] One is a closed model where users are not given access to the source code and most of the intellectual property is protected as a secret wherein the program is released only in its object code version.[20] Under the second model, called open model or open source model, users are given access to the source code and all of the software copyrights are made available to the market by providing a very liberal license.[21]

[15] Dennis M. Kennedy, *A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture*, 20 ST. LOUIS U. PUB. L. REV. 345, 346 (2001).

[16] In the object code version, the user cannot make any alterations or integrate it with some other software, or study it to see how it functions, and for doing all the aforesaid things access to the source code is required.

[17] Stein, *supra* note 14, at 160.

[18] Kennedy, *supra* note 15, at 346.

[19] Software is generally licensed, not sold, and the reasons for this are explained in the later portions of this article.

[20] Daniel B. Ravicher, *Facilitating Collaborative Software Development: The Enforceability of Mass-Market Public Software Licenses***,** 5 VA. J.L. & TECH. 11, ¶ 29 (2000). *Also available at* http://www.vjolt.net/vol5/issue3/v5i3a11-Ravicher, (last visited Feb. 4, 2016).

[21] *Id*. at ¶ 30.

III. PROTECTION OF SOFTWARE UNDER COPYRIGHT LAW

The United States of America ("U.S.") and the United Kingdom were the first countries to afford protection to computer software within the ambit of intellectual property legislation.[22] Even though copyright law was being used to protect authors of literary, artistic, musical and dramatic works in the early days of computer programming, software was not protected under copyright law.[23] Later on as technology progressed, new and different modes of expression, including software, received copyright protection.[24] Copyright law provides software developing companies numerous benefits and virtually no drawbacks, and its greatest advantage is that it protects them against verbatim copying, an act which makes them most vulnerable.[25]

However the most difficult issue in copyright protection is to pinpoint the exact nature and scope of monopoly granted by such protection. Copyright protects both the source code and the object code of computer programs.[26] In the U.S., the Copyright Act has provided for restrictions about the scope of protection for all works by stipulating that "copyright protection for an original work of authorship does not extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."[27]

One of the landmark cases, which dealt with the issue of an idea-expression dichotomy in the context of software, arose in the year 1986.[28] That case, which was decided by the Court of Appeals for the Third Circuit, involved a copyright infringement action concerning a software program written to manage the operation of a dental laboratory.[29] In that case the appellant, Jaslow Dental Laboratory, Inc., was manufacturing dental prosthetics and devices. Plaintiff-appellee Whelan Associates, Inc. was also a company in the business of developing and marketing custom computer programs.[30]

[22] Nancy Kemp DuCharme & Robert F. Kemp, *Copyright Protection for Computer Software in Great Britain and the United States: A Comparative Analysis*, 3 SANTA CLARA COMPUTER & HIGH TECH. L. J. 257, 260–61 (1987).

[23] *Id.* at 261.

[24] *See* 17 U.S.C. § 117 (1998). (Originally enacted in 1976, the United States Congress amended the Computer Software Copyright Act in 1980, defining "computer program" in the federal copyright statute. Since then, courts have generally considered computer programs "literary works" and therefore protected by copyright).

[25] Strasser, *supra* note 7, at ¶ 35.

[26] Apple Comput., Inc. v. Franklin Comput. Corp., 714 F.2d 1240, 1249 (3d Cir. 1986).

[27] 17 U.S.C. 102(b) (1990).

[28] Whelan Assocs. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3d Cir. 1986).

[29] *Id.* at 1225.

[30] *Id.*

Before the district court, the plaintiff and defendants claimed "exclusive proprietary rights to a computer software system developed as a complete computer business management and control program for the operation of dental laboratories."[31] Defendants rejected this claim and contended that their program was developed independently and thus there was no copyright infringement.[32] The District Court held against the defendant by holding that its program, although written in a different computer language than the plaintiff's, was substantially similar to plaintiff's program.[33] This substantial similarity, with defendant's acknowledged access to the plaintiff's program, led the district court to issue judgment in favor of the plaintiff. The defendant appealed to the Third Circuit Court of Appeals.[34]

The Court of Appeals noted that the main question it confronted in *Whelan* was "whether mere similarity in the overall structure of programs [could] be the basis for a copyright infringement, or, put differently, whether a program's copyright protection covers the structure of the program or only the program's literal elements, *i.e.*, its source and object codes."[35] It further held that the idea of the impugned program was the "efficient management of a dental laboratory," which had significantly different requirements from those of other businesses.[36] That idea "could be accomplished in a number of different ways with a number of different structures," the structure of the program, therefore, was part of the program's expression rather than its idea.[37] The *Whelan* court affirmed the decision of the District Court, and held that the structure, sequence, and organization of the parties' programs were substantially similar.[38] Thus the court held that the program's copyright protection also covered the structure of the program apart from the program's literal elements like source code.[39] This decision brought the structure, sequence, and organization of a computer program within the ambit of copyright protection.[40]

[31] Whelan Assocs. v. Jaslow Dental Lab., 609 F. Supp. 1307, 1316 (E.D. Pa. 1985).

[32] *Id.*

[33] *Id.* at 1321.

[34] *Whelan*, 797 F.2d at 1232.

[35] *Id.*

[36] *Id.* at 1236 n.28.

[37] *Id.* at 1236.

[38] *Id.* at 1243.

[39] *Id.*

[40] *Id.* at 1244.

A. Abstraction, Filtration, Comparison Test

After six years, in *Computer Associates*, *Int'l v. Altai, Inc.*,[41] the Second Circuit Court of Appeals was once again involved in deciding a similar case.[42] In this case, the plaintiff created a computer program containing a sub-program that enabled the primary program to run on different operating systems.[43] The defendant hired a former employee of the plaintiff, who misappropriated copies of the plaintiff's source, which he used to create a program for Altai.[44] After learning of the employee's copying, Altai rewrote its program, leaving out the portions that had been copied from Computer Associates' subprogram.[45] Plaintiff sued for copyright infringement, arguing both versions of defendant's program violated its copyrights.[46]

The lower court found the copied program violated Computer Associates' copyright and awarded damages, but found that the rewritten program did not violate the plaintiff's copyright, and the plaintiff appealed.[47] Appellant-plaintiff contended that despite Altai's rewriting of its program's code,[48] the resulting program was substantially similar to the structure of its program.[49] The Court of Appeals opined that the approach the court followed in *Whelan*,[50] which separated idea from expression in computer programs, relied too heavily on metaphysical distinctions and did not place enough emphasis on practical considerations.[51] Instead of following the decision in *Whelan,* the court in *Computer Associates* enunciated an *Abstraction Filtration Comparison* test.[52] To ascertain substantial similarity under this test, a court would first break down the allegedly infringing program into its constituent structural parts and "[t]hen examine each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, so that it could then sift out all non-protectable material."[53] After filtering out these elements, the remaining "kernel" would be compared with the structure of an allegedly infringing

[41] 982 F.2d 693 (2d Cir. 1992).
[42] *See generally id.*
[43] *Id.* at 698.
[44] *Id.*
[45] *Id.*
[46] *Id.* at 698–99.
[47] *Id.* at 697.
[48] The defendant's program was named OSCAR.
[49] Appellant's program was names ADAPTER.
[50] *Whelan*, 797 F.2d at 1233.
[51] *Comput. Assocs*., 982 F.2d at 708.
[52] *Id*. (The importance of this test has been reiterated in many subsequent cases including the most recent *Oracle* case; see 750 F.3d 1339 (Fed. Cir. 2014), discussed in Part IX.D., *infra*.)
[53] *Id.* at 706.

program.[54] "The result of this comparison [would] determine whether the protectable elements of the programs at issue were substantially similar so as to warrant a finding of infringement."[55] Finally, the court held that there was no copyright infringement in *Computer Associates*.[56]

The abovementioned two cases have contributed a great deal toward the evolution of a licensing model in software where the user is given access to the program's source code and, as a result, the user need not be unduly worried about the prospect of litigation involving modification or distribution of the source code.

IV. SOFTWARE LICENSING

Even though copyright can protect software, software developers are not fully dependent on intellectual property law's capability to adequately protect their interests.[57] Because of the particular nature of software, widespread copying is possible, and copyright law many not be able to provide 100% protection. Software can be subjected to multiple unauthorized copying without causing degradation in program's functional features. This feature, combined with the connectivity amongst software users offered by the Internet, enables rampant copying of software programs without any constraints of space, time or geographies.

Thus software distributors have extended the protection offered by intellectual property laws using contract law, in the form of licensing agreements.[58] These software licenses help developers and distributors to substantially increase the basic protection granted by intellectual property law. For a software licensor, a licensing agreement is better than a sale, because the former allows the licensor to maintain its precious intellectual property rights in the program and grant only those rights specified in the license.[59]

Secondly, licenses confer considerable flexibility to software developers for customizing licenses, like granting an exclusive license to a single licensee or nonexclusive licenses to several or many licensees.[60] Furthermore under the *first sale* doctrine, once the copy of a copyrighted work has been sold, the copyright owner has no authority to control subsequent transfers of that particular tangible copy. These licenses can be

[54] *Id.*

[55] *Id.*

[56] *Id.* at 708–11.

[57] Apart from copyright law, patent and trade secret law can be used to protect the rights in software.

[58] Stein, *supra* note 14, at 170.

[59] One licensor could allow a licensee to make copies of a program, but can limit any distribution of those copies.

[60] Ira V. Heffan, *Copyleft: Licensing Collaborative Works in the Digital Age*, 49 STAN. L. REV. 1487, 1496 (1997).

used to circumvent the first sale doctrine, as the doctrine can cover only sale of software and not licensing of it.[61] To attain this objective of circumventing the first sale doctrine software developers initially relied on a license called "shrink-wrap" licenses to impose contractual obligations upon licensees.[62] The name shrink-wrap license emerged from the manner in which the software is packaged, where the terms of the agreement are visible through the shrink-wrap covering on the box, or the package, which contains the stipulation that the use of the software is subject to the terms and conditions provided inside the package.[63] The validity of these licenses was upheld subsequently in the famous case of *ProCD, Inc. v. Zeidenberg*.[64]

A. First Sale Doctrine

The first sale doctrine can have serious implications for software.[65] Since the nature of software is such that it can be copied easily without degrading the program, the first sale doctrine could potentially permit and facilitate widespread copyright infringement.[66] Thus, by structuring the transaction in the form of a license, the creators of software can circumvent the first sale doctrine.[67] Courts have held that a software license is not a sale and therefore the first sale doctrine is not applicable to licensing arrangements.[68] Thus through licensing agreements, software creators and distributors get the right to put more stringent restrictions and limitations on the use of the intellectual property involved in the software than would be possible if the transaction were classified as a sale.[69] The sale of a product places substantial restrictions on the degree of control that intellectual property rights holders can retain over the product. Thus, if the rights holder devises a model wherein it retains ownership by merely licensing the product to end-users, such drastic limitations do not arise.[70]

[61] In the USA, the first sale doctrine is based upon the Supreme Court case of *Bobbs-Merrill Co.* v. *Straus*, 210 U.S. 339 (1908), where the Court held that once a valid first sale of a copy of a copyrighted work happened, the copyright owner could no longer have any control over any subsequent distribution of that copy. The doctrine was codified as 17 U.S.C. § 109 (2008), in the 1976 Copyright Act.

[62] Heffan, *supra* note 60, at 1498.

[63] *Id*.

[64] 86 F.3d 1447, 1450 (7th Cir. 1996).

[65] *See* John A. Rothchild, *The Incredible Shrinking First Sale Rule: Are Software Resale Limits Lawful?* 57 RUTGERS L. REV. 1, 14-15 (2004) (explaining the ease in which software may be redistributed).

[66] *Id*.

[67] *Id*. at 22–23.

[68] Adobe Sys. Inc. v. One Stop Micro, Inc., 84 F. Supp. 2d 1086, 1092 (N.D. Cal. 2000).

[69] Heffan, *supra* note 60, at 1498.

[70] Ravicher, *supra* note 20, at ¶ 34.

V. EVOLUTION OF OPEN SOURCE

The history of open source and that of software programming happened almost at the same time.[71] The history of the open source approach begins with an MIT computer programmer named Richard Stallman who was disillusioned by the restrictions that software licenses placed on the use and modification of computer programs.[72] According to Stallman, these restrictions prevented the effective and efficient development of innovative software technology, and he also felt that these licenses limited the ways in which the software was being developed.[73] Stallman's vision and philosophy was that software should be free as in speech, and he firmly believed that proprietary, commercial development of software was the main cause for numerous problems dealing with security, loss of innovation, incompatibilities, and so forth, because there are very few skilled, independent programmers who could analyze and correct source code.[74]

After giving serious thought to these issues, Stallman, with the help of law professors, published the General Public License, commonly known as the *GPL*.[75] The GPL is generally referred to as a policy statement *cum* license, because in it Stallman spells out the fundamental philosophy of free software.[76]

> When we speak of free software, we are referring to freedom, not price. Our general public licenses are designed to make sure you have the freedom to distribute copies of free software, that you receive source code or can get it if you want it, that you can change the software or pieces of it in new free programs; and that you know that you can do these things.[77]

Thus, Stallman specified the three essential components of free software: the right to distribute software, the right to access the source code, and the right to modify the source code. Thus GPL is widely considered the most stringent open source license because of its *copyleft*

[71] *See* OPEN SOURCE INITIATIVE (2012), *available at* http://opensource.org/history (last visited Jan. 4, 2016). Open Source or Free Software is also called by the name Free and Open Source Software (FOSS), *available at* http://freeopensourcesoftware.org/index.php?title=Main_Page (last visited Jan. 4, 2016).

[72] Kennedy, *supra* note 15, at 349.

[73] *Id*.

[74] *Id*. at 350.

[75] *See generally* Richard Stallman, *Why Open Source Misses the Point of Free Software* (Feb. 12, 2016, 11:48 AM), http://www.gnu.org/philosophy/open-source-misses-the-point.en.html.

[76] *Id*.

[77] *See* PREAMBLE OF GNU GENERAL PUBLIC LICENSE, http://www.gnu.org/copyleft/gpl.html (last visited Jan 4, 2016) [hereinafter GNU].

provision.[78] The copyleft provision ensures that any follow-on distribution of an open source project or derivative work is only done per the terms and conditions of the GPL open source license. Thereafter, Stallman founded the Free Software Foundation (FSF) in 1985 to encourage the development and distribution of open source software under the GPL. However the most significant development in the evolution of the GPL is Linux, created by Linus Torvalds, a Finnish computer science student, who developed an operating system that was more stable than Microsoft Windows NT, at a much lower cost.[79]

A. Berkeley Software Distribution License

Almost at the same time when Stallman was trying to establish the FSF and promote the GPL, programmers at the University of California at Berkeley were working on improving the Unix operating system.[80] Their efforts closely resembled Stallman's free software projects in the sense that programmers were given free and open access to source code, and they could make derivative works in order to fix bugs, improve the software, and fine-tune the program.[81] The project also became popular, which eventually forced them to draft a license called the *Berkeley Software Distribution* (BSD) License.[82] The BSD license allowed licensees to work with source code and create derivative works.[83] Interestingly, the BSD license was similar in many ways to Stallman's GPL, but there was one major difference: in the BSD license, there was no requirement that the derivative works must be distributed under the same licensing terms.[84] Thus, the GPL notion of copyleft is not embodied in the BSD License.

B. Mozilla Public License

In 1998, Netscape Corporation released its browser program, *Netscape Navigator*, which was also the most popular Internet browser at the time, as an open source project.[85] This decision by Netscape eventually led to two key developments in the history of open source. Firstly, discussions about

[78] Stein, *supra* note 14, at 182.

[79] *See* LINUX (Nov. 18, 2016, 2:12 PM), http://en.wikipedia.org/wiki/Open-source_movement.

[80] Andrew Leonard, *BSD Unix: Power to the People, From the Code*, SALON (February 12, 2016, 11:54 AM), http://www.salon.com/tech/fsp/2000/05/16/chapter_2_part_one/print.html; http://en.wikipedia.org/wiki/Open-source_movement.

[81] Kennedy, *supra* note 15, at 351.

[82] *Id.*

[83] *Id.*

[84] *See generally* DONALD K. ROSENBERG, OPEN SOURCE: THE UNAUTHORIZED WHITE PAPERS (M&T Books 2000), *available at* http://www.stromian.com/Book/FrontMatter.html (last visited Feb. 12, 2016).

[85] Kennedy, *supra* note 15, at 353.

the licensing issues involved a number of strong proponents of open source and paved the way for the publication of a definition on open source.[86] Secondly, as a commercial developer, Netscape knew the major issues raised under the existing open source licenses particularly for commercial projects where the underlying code contained software licensed from a large number of developers.[87]

Had Netscape wanted to release its browser under an open source license like the GPL, every underlying licensor of software used in the browser would have had to agree to release their incorporated code under the open source license ultimately chosen by Netscape. Because of this herculean task, and the difficulties involved in GPL, Netscape decided to draft its own open source license. In order to make the license acceptable to the open source community, Netscape developed the *Mozilla Public License* (MPL) and this license now serves as an important model of an open source license useful in situations involving commercial software or commercial development.[88]

Netscape coordinated with Bruce Perens, who was already involved in the open source Linux community and was in the process of framing some standards for open source.[89] Perens ultimately gave the *Open Source Definition*.[90] The open source definition is a comprehensive statement of open source principles, and a way of tracking the licenses that complied with the open source definition.[91] The *Open Source Initiative* (OSI) was also established as an entity responsible for maintaining the open source definition, promoting open source, and helping pave the way for continuing involvement of commercial entities in open source projects. The OSI is thus a not-for-profit body steering many initiatives in the open source community. It deals with the 10 conditions needed for the software program to be considered as open source.[92]

Firstly, the software license should not limit the licensee from selling or giving away the software as a component part of a larger software program.[93] Secondly, dissemination of the program must include distribution of the source code, which must be easily accessible.[94] The license must also allow the software to be modified and must allow

[86] *Id.*

[87] *Id.* at 354.

[88] *Mozilla Public License Version 1.1*, MOZILLA FOUNDATION, https://www.mozilla.org/en-US/MPL/1.1/ (last visited February 12, 2016).

[89] ROD DIXON, OPEN SOURCE SOFTWARE LAW 7 (2004).

[90] *Id.* at 8.

[91] *Id. See also Open Source Initiative*, OPENSOURCE.ORG, http://opensource.org/osd (last visited Jan. 27, 2016).

[92] KEN KRECHMER, OPEN SYSTEMS IN DIGITAL CONVERGENCE 117-18 (Sangin Park et al. eds., 2007) (explaining the ten conditions).

[93] *Id.* at 117.

[94] *Id.*

derivative works to be distributed under the same license terms.[95] Restrictions on such distribution shall only be allowed if patch files are instead distributed with the source code in order that the program may be modified at build time.[96] The license should not discriminate against any person or group of persons, and should not restrict anyone from making use of it in a commercial field.[97] Additional licenses are not required for redistribution, and the license must not be product-specific or restrict other software that is distributed along with the licensed software.[98] Finally, the license must be technology neutral, which essentially means that click-wrap license is forbidden.[99] Interestingly, the GPL, the BSD license and the Mozilla Public License are all in conformity with the Open Source Definition.[100] With the release of the Open Source Definition, the acceptance of the open source style of development and licensing has subsequently increased a great deal.

C. Public Domain Software and Open Source Software

People tend to confuse open source software with public domain software and thus it is necessary to distinguish the two terms.[101] There can be some software that is released freely into the public domain and in that the author will not hold a copyright.[102] In the case of open source software this does not happen. Instead, a license governs open source software and the software's copyright owners continue to own the copyright and assert their rights to it.[103] In other words, open source can be compared to a software license, wherein the license gives the users more rights than they can expect under commercial software licenses.[104] In the case of a person who releases his or her software into the public domain, there is a surrender of the copyright.[105] In such a case, others can then use the author's work as they want, including modifying it, removing the author's name etc.[106] In

[95] *Id.*

[96] *Id.*

[97] *Id.*

[98] *Id.* at 118.

[99] *Id.* Click-wrap licenses typically require an end user to give his or her assent to the license by clicking an "ok" button on a dialog box or pop-up window.

[100] Robert W. Gomulkiewicz, *How Copyleft Uses License Rights to Succeed in the Open Source Software Revolution and the Implications for Article 2B*, 36 HOUS. L. REV. 179, 186–87 (1999).

[101] MORENO MUFFATTO, OPEN SOURCE: A MULTIDISCIPLINARY APPROACH 35 (2006) (explaining the differences through graphical representations).

[102] *Id.* at 35–42.

[103] *See generally* ANDREW M. ST. LAURENT, UNDERSTANDING OPEN SOURCE AND FREE SOFTWARE LICENSING (1st ed. 2004).

[104] Kennedy, *supra* note 15, at 358.

[105] MUFFATTO, *supra* note 101, at 35–36.

[106] *Id.* at 45.

some extreme cases they can even remove a particular version from the public domain by asserting their copyright ownership.[107]

In the case of open source software, however, a combination of copyright law and licensing serves to permit the free distribution of the software by copyrighting the software and thereafter allowing a liberal license under one of the various open source models. Open source software also needs to be distinguished from shareware or freeware software because in the latter case, the developer offers a standard license, with some special discounts in price or, in some cases, the user is charged no money.[108] However, unlike in the case of open source software, a shareware user will never get any access to the source code or the right to make derivative works.[109]

VI. OPEN SOURCE SOFTWARE AND ITS IMPACT UPON PUBLIC DOMAIN

In the case of a proprietary software license, its main thrust is to protect the copyright interests of the owner by restricting the use of the software. But in the case of an open source license, there are very few restrictions on the use of software. All open source licenses require source code to be made available, and all protect the right of users of the software to make derivative works.[110] All the open source licenses disclaim warranties, and many attempt to limit liabilities. Because the open source licenses have few restrictions, and they provide access to the source code, they have a positive impact upon the public domain.

A. Impact of General Public License (GPL)

The GPL contains strong language regarding the freedom of software, as well the stipulation regarding *copyleft*, which specifies that all derivative works must also remain free.[111] Thus GPL has its share of conflicts with open source and commercial developers, specifically those who would like to make their derivative work private or subject to a standard commercial license.[112] In essence, the GPL gives permission to the licensees to copy

[107] Kennedy, *supra* note 15, at 358.

[108] *See Shareware*, PC MAGAZINE ENCYCLOPEDIA (2016), *available at http://www.pcmag.com/encyclopedia/term/51251/shareware* (defining the term "shareware").

[109] ALEXANDROS PINO, ASSISTIVE TECHNOLOGIES AND COMPUTER ACCESS FOR MOTOR DISABILITIES 112-114 (Georgios Kouroupetroglou ed., 2014).

[110] STEFANO COMINO, FABIO M. MANENTI, & ALESSANDRO ROSSI, HANDBOOK OF RESEARCH ON OPEN SOURCE SOFTWARE: TECHNOLOGICAL, ECONOMIC, AND SOCIAL PERSPECTIVES 415 (Kirk St. Amant & Brian Still eds., 2007).

[111] *See* GNU, *supra* note 77.

[112] *Id.* at § 6.

and distribute verbatim copies of source code, which they receive upon satisfying the certain conditions.[113] One condition is that the licensees must publish an appropriate copyright notice and disclaimer of warranty on every copy; it should also keep all the notices that refer to the license and to the absence of warranties intact, and must give all other recipients of the program a copy of the license with the program.[114]

GPL also permits the licensee to modify a copy of the program, or any portion of it, as well as copy and distribute the licensee's modifications under the terms of the GPL, provided that each modified file carries a prominent notice stating that the licensee modified the file indicating the date of the change.[115] Each work that contains the program, or any part thereof, or is derived from it, must be licensed as a whole at no charge to third parties under the terms of the GPL.[116] Furthermore, the GPL also requires the object code to be distributed along with source code, or for the source code available in one of several permitted ways.[117] Users who try to copy, modify, sub-license, or distribute a program, except as expressly provided under the GPL, will have their licenses automatically terminated pursuant to the terms of the GPL.[118] The GPL is modeled along the lines of a shrink-wrap[119] or click wrap[120] type of license and thus no signature is required and the simple act of modifying or distributing the program indicates acceptance of the license and all terms and conditions of the GPL.[121] In many respects, the provisions of the GPL reflect standard license agreements.

[113] *See generally id.* (explaining that "[e]veryone is permitted to copy and distribute verbatim copies of this license document . . .").

[114] *Id.* at § 15.

[115] *Id.* at § 1.

[116] *Id.* at § 2.

[117] *Id.*

[118] *Id.* at § 8.

[119] Daniel B. Ravicher, *Facilitating Collaborative Software Development: The Enforceability of Mass-Market Public Software Licenses*, 5 VA. J.L. & TECH. 11, 40 (2000).

> If a consumer purchases a copy of the program, the license is typically included in print form either on or within the packaging. The license gives the consumer the opportunity to review the license and return the software for a full refund if she does not wish to abide by its terms. Notice of the license and the requirement that the consumer consent to its terms typically appears on the exterior of the software packaging. This method of mass-market licensing is referred to as "shrink-wrap" licensing.

Id.

[120] *Id.* at 41.

> If the consumer downloads a program via an Internet web site the screen usually displays the license agreement, upon prompting, the user must point and click on a button labeled "I accept" or "Agree" in order to proceed with use of the program. This method is referred to as "click-wrap" licensing.

Id.

[121] GNU, *supra* note 77, at § 9.

The most important feature of the GPL is that licensees cannot make modifications of GPL programs private or proprietary.[122] Licensees must also distribute their modifications under the same terms contained in the GPL, including the copyleft provisions.[123] Furthermore, the GPL does not allow software licensed under its terms to be mixed with non-GPL software without also making the non-GPL software subject to the terms of the GPL.[124] It should be noted that the copyleft provision is restricted to derivative works distributed to the public.[125] The GPL only covers those "works licensed under the GPL and all derivative works; thus a user who does not make a derivative work does not have to comply with copyleft."[126]

B. Impact of BSD Licenses

Compared to GPL, BSD licenses have very few restrictions.[127] Under the BSD licenses, distribution of source code is permitted.[128] However the distributing source code for derivative works is not mandated.[129] This means that programs made under a BSD license could be combined with proprietary software.[130] Thus the BSD licenses are perceived to be freer than the GPL because they allow developers to release derivative works under whatever license they want. Such licenses for derivative works are not required to contain the same terms and conditions contained in the BSD license, which is only applicable to the original code.

Thus it becomes clear that the BSD licenses do not have copyleft terms, which has been the hallmark of GPL.[131] Undoubtedly, the absence of copyleft terms has made BSD licenses more appealing to commercial developers. The BSD licenses permit licensees to do anything they want with the source code, provided they follow the specific requirements of the license.[132] Scholars attribute this to the funding the licensors received from the U.S. government.[133]

[122] *See id.* at Preamble.

[123] *See generally* GNU, *supra* note 77.

[124] Vetter, *supra* note 8, at 600.

[125] FABRICIO BERTINI PASQUOT POLIDO & MONICA STEFFEN GUISE ROSINA, FREE AND OPEN SOURCE SOFTWARE (FOSS) AND OTHER ALTERNATIVE LICENSE MODELS: A COMPARATIVE ANALYSIS 99 (Axel Metzger ed. 2016).

[126] Theresa Gue, *Triggering Infection: Distribution and Derivative Works under the GNU General Public License*, 2012 U. ILL. J.L. TECH. & POL'Y 95, 99–100 (2012).

[127] FADI P. DEEK & JAMES A.M. MCHUGH, OPEN SOURCE TECHNOLOGY AND POLICY 246–47 (2008).

[128] *See id.*

[129] *Id.*

[130] "Some code under the BSD Licenses can now be found in common commercial software such as Windows NT and the Macintosh operating system." Kennedy, *supra* note 15, at 363.

[131] *See* DIXON, *supra* note 89, at 43.

[132] *Id.*

[133] Kennedy, *supra* note 15, at 364.

C. Impact of the Mozilla Public License

As mentioned earlier, in 1998, Netscape Corporation decided to release its browser program, Netscape Navigator, as an open source project.[134] However, there were many difficulties "because the browser's source code included code licensed under a variety of commercial licenses and permissions . . . ."[135] Thus re-licensing of the underlying code would have been required for the release of the code under the GPL in particular. This being a very difficult task, Netscape held consultations with the "leading figures in the open source movement in order to develop a license and also received support from the open source community."[136] Thus Netscape ultimately produced two licenses.[137] The first one was the Netscape Public License (NPL), which dealt specifically with the issues implicated by the underlying third party licensed code used in the code for the Netscape browser, as well as other concerns related to the conversion of existing code.[138]

The second license, called the Mozilla Public License (MPL), was more important.[139] The MPL is an ideal model for open source licensing which can be used by commercial software entities.[140] Notably, Netscape consulted various experts, including lawyers, before preparing the MPL license.[141] When compared with GPL or BSD licenses, the MPL looks more like a professional commercial software license prepared by experts.[142] The significance of MPL is paramount because it can act as a reliable model for future releases of commercial software like Navigator into open source.[143]

In preparing the MPL, a lot of work was done to define the term *Covered Code* so as to differentiate between the software and code that covered by the license and those that were not.[144] In fact, the MPL attempts to combine the best features of the BSD and GPL licenses. The MPL permits commercial licensing of derivative works. However changes to the source code of the covered program must be made freely available to anyone. Importantly, the MPL does not contain the copyleft provisions of

[134] *Id.* at 353.

[135] *Id.* at 364.

[136] *Id.*

[137] *Id.* at 365.

[138] *Id. See also Netscape Public License: Version 1.1, available at* http://directory.fsf.org/wiki?title=License:NPLv1.1 (last visited Jan 29, 2016).

[139] *See id.*

[140] *See id. See also* Details of MPL *available at* http://www.mozilla.org/MPL/1.1/ (last visited Jan. 29, 2016).

[141] *Id.*

[142] *Id.*

[143] *Id.*

[144] *Mozilla Public License Version 1.1, supra* note 88, at § 1.3.

the GPL. Thus the MPL more closely resembles the BSD license than the GPL.[145]

VII. THE SUCCESS OF CODE SHARING

The name of Linux is synonymous with open-source software.[146] In fact, Linux is an operating system, and the history of operating systems, especially with respect to personal computers, concerns the evolution of two operating systems.[147] One of them, Microsoft Windows, grew to dominate the desktop computer market.[148] Linux, the other operating system, began as a code written by a single individual that drew the attention of many participants and developed using the open-source approach.[149] It all started in 1991, when Linus Torvalds made an operating system kernel as part of a project done with the intention of exploring a particular design for an operating-system kernel.[150] The kernel that ultimately became Linux began as a project whose aim was to "comment and improvise upon certain design aspects" of the Unix family of operating systems.[151] It is worth mentioning that traditions that would later become fixtures of the open source movement, like sharing source code, collaborating on the design and development of software components, and managing and organizing the collaborative effort, evolved in the Unix programming community.[152]

In fact, the technological innovation in the kernel behind Linux was not primarily responsible for its growth and popularity.[153] Linux's popularity can instead be attributed to the open source approach employed by Torvalds, who shared the source code with others to get their feedback on his design.[154] Torvalds subsequently used the GPL to develop his kernel using the open source model, which resulted in more sharing and exchange

[145] Kennedy, *supra* note 15, at 365.

[146] SRDJAN RUSOVAN, MARK LAWFORD, & DAVID LORGE PARNAS, PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE 108 (Joseph Feller et al. eds., 2005).

[147] L. GOPALAKRISHNAN, G. PADMANABHAN, & SUDHAT SHUKLA, YOUR HOME PC: MAKING THE MOST OF YOUR PERSONAL COMPUTER 173–75 (2003).

[148] *See* United States v. Microsoft Corp., 253 F.3d 34, 54-55 (D.C. Cir. 2001) (observing that Microsoft holds more than eighty percent of market, and that its Windows operating system "supports many more applications than any other operating system").

[149] *See* DRUMMOND REED & JERRY KINDALL, THE INTERNET ENCYCLOPEDIA G-O 493 (Hossein Bidgoli et al. eds., 2d ed. 2004).

[150] *Id.*

[151] *Id.*

[152] *Id. See also* Marshall Kirk McKusick, *Twenty Years of Berkeley Unix From AT&T-Owned to Freely Redistributable*, *in* OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION 31 (Chris DiBona, et al. eds., 1999), http://oreilly.com/catalog/ opensources/book/kirkmck.html.

[153] *Id.*

[154] *Id.*

of ideas among many participants, which eventually helped him develop the kernel into a full-fledged operating system.[155]

VIII. OPEN SOURCE AS A NEW BUSINESS MODEL

Open source software has the capability to privately provide public goods on a massive scale.[156] "[U]nder the open-source approach, excluding others' use has been mostly disclaimed."[157] In reality "[t]he conditional permissions underlying the open source approach encourage others' use."[158] These "conditions impose source code availability and prohibit royalties."[159] Although Linux and Linux-based companies have achieved commercial success in developed economies like the United States, Linux has the potential to benefit the developing world as well.[160] Thus, many of the developing countries that need cheap and efficient technology to make technological progress are now using Linux.[161]

The Linux development effort's major achievement was designing a way to prevent the program from becoming proprietary.[162] The kernel is at the heart of all Linux systems and is developed by individuals, companies, and organizations working on their own time and with their own money.[163] Linux's use of the GPL licensing system ensures that all computer code licensed under it remains accessible for everyone so that they too may be able to make modifications.[164] In case anybody changes the kernel, then those changes must be made known to the public.[165] As a result of this disclosure requirement, no innovation can be kept secret and this enables developers to build upon previous developer's work.[166]

Thus, GPL will permit Linux development to be self-propagating, which means that any advancement has to be shared with the public, and others are able to further advance the development.[167] Although Linux is

[155] *Id. See also* Vetter, *supra* note 8, at 607.

[156] Vetter, *supra* note 8, at 607.

[157] *Id.*

[158] *Id.*

[159] *Id.*

[160] Stein, *supra* note 14, at 185.

[161] *Id.*

[162] MARK TETER, PARADIGM SHIFT: SEVEN KEYS OF HIGHLY SUCCESSFUL LINUX AND OPEN SOURCE ADOPTIONS 110 (2006).

[163] *See id.* (explaining that a wide variety of people and entities are involved in the development process).

[164] P. RAGHAVAN, AMOL LAD, & SRIRAM NEELAKANDAN, EMBEDDED LINUX SYSTEM DESIGN AND DEVELOPMENT 388 (2006).

[165] *Id.* (explaining the legal complexity of this issue).

[166] *Id.*

[167] Kerry D. Goettsch, *Recent Development: SCO Group* v. *IBM: The Future Of Open-Source Software*, 2003 U. ILL. J.L. TECH. & POL'Y 581, 583 (2003).

referred to as open-source software, it is not always *free* in terms of price.[168] There are companies who "have emerged as profit-making Linux distributors either by charging a fee for service and support or by selling the hardware on which Linux can run."[169]

The Apache web server project is another important example of open source software.[170] The Apache project began with the source code of an early web server developed at the University of Illinois by the National Center for Supercomputer Applications (NSCA).[171] After getting the code, a team of webmasters used the NCSA source code, which was freely available, to develop the software of their choice.[172] Although the initial setup consisted of an informally organized group, it led to the formation of a formal organization called the Apache Software Foundation (ASF).[173] As a foundation, the ASF has the authority to implement formal structures to govern and manage the projects.[174]

The Apache open-source license is different from the GPL used for Linux in a number of ways.[175] Generally speaking it is much less restrictive and it allows anybody to do whatever he likes with the source code as long as certain attributions and notices remain in modified or unmodified versions.[176] The crucial difference is that anyone under the Apache license can make use of public domain material, even for a commercial offering, by charging royalty fees for use.[177] This is clearly not possible under the GPL. The Apache license provides this freedom and does not require that the source code be included with redistributions of the software.[178]

History helps us understand the main differences between the Apache license and the GPL. Stallman's main objective while writing the GPL was to create a freely available software along with source code that could not be brought to the private domain in certain specific ways like, for example, no royalties for use and the same GPL terms had to apply to closely

[168] *Id.*

[169] *Id.*

[170] JOHNNY RYAN, A HISTORY OF THE INTERNET AND THE DIGITAL FUTURE 114–15 (2010).

[171] BETTINA FABOS, WRONG TURN ON THE INFORMATION SUPERHIGHWAY: EDUCATION AND THE COMMERCIALIZATION OF THE INTERNET 32 (2004).

[172] Josh Lerner & Jean Tirole, *The Simple Economics of Open Source*, NBER WORKING PAPER SERIES (last visited Jun. 30, 2015), http://papers.nber.org/papers/w7600.pdf.

[173] *See* APACHE SOFTWARE FOUNDATION, http://www.apache.org/foundation/ (last visited Jan. 4, 2016).

[174] *Id.* at http://www.apache.org/foundation/how-it-works.html (last visited Jan. 4, 2016).

[175] Bruce Perens, *The Open Source Definition*, *in* OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION, *supra* note 152 at 183.

[176] *See* APACHE SOFTWARE FOUNDATION, http://www.apache.org/licenses/LICENSE-2.0 § 2 (last visited Feb. 24, 2016).

[177] GREG R. VETTER, COPYRIGHT AND RELATED RIGHTS: INTELLECTUAL PROPERTY AND INFORMATION WEALTH: ISSUES AND PRACTICES IN THE DIGITAL AGE (Peter K. Yu ed. 2007).

[178] *See* APACHE SOFTWARE FOUNDATION, *supra* note 176, at § 4.

connected software.[179] In contrast, the Apache license owed its existence to the BSD license that had been used to release the source code that contained ingredients of the Unix operating system developed at Berkeley.[180]

Apache is only one of the many important Internet technologies that are dependent upon open-source software and on many occasions it is combined with Linux for web server installations.[181] This combination of two open source models gives a very tough competition to companies like Microsoft.[182] Undoubtedly "Linux and Apache are two of the most successful examples of the open-source approach" that "have been oriented to a technical user base."[183] This also clearly demonstrates the point that teams of technical users who are similarly placed are in the best position to fully exploit the opportunity to collaborate offered by the open-source approach.[184]

IX. PROMINENT LITIGATION INVOLVING OPEN SOURCE SOFTWARE

The jurisprudence involving open source software is in its infancy.[185] Still, these cases can effectively guide the stakeholders about the various rights granted by such software. The licensor of open source software might insist that its licensee shall abide by the common principles of attribution, sharing alike, and non-discrimination. In case the licensee violates any of these principles, the licensor can sue the licensee under two categories: (i) breach of contract; and (ii) copyright violation.[186] The first category involving breach of contract is easy to proceed as license falls under the category of contracts.[187] However, to proceed under the second category, *i.e*, under the claim of copyright infringement, the licensor has to prove before the court that no license was granted in the first place to the licensee or the license which was given was a conditional one which mandated certain conditions on the licensee and due to a violation of the

[179] *See* Perens, *supra* note 175.

[180] *See* VIRGIL ANDERSON, ANALOGOUS MODELS & DIGITAL COMPUTING 71 (1st ed. 2012) (explaining the origins of the Unix operating system).

[181] *See* MIKE HENDRICKSON, ROGER MAGOULAS, & TIM O'REILLY, ECONOMIC IMPACT OF OPEN SOURCE ON SMALL BUSINESS: A CASE STUDY 6 (1st ed. 2012) (explaining the dependency on open source software).

[182] *Id.* (noting the difficulties some companies may face).

[183] Vetter, *supra* note 8, at 613.

[184] *Id.*

[185] *See* Nemiah, *supra* note 3, at 368 (comparing the state of the law to "an undisturbed minefield").

[186] *Id.*

[187] *Id.*

conditions the license effectively did not exist.[188] The following are some of the important cases that will have a bearing on open source software licensing practices.

A. *Graham v. James*[189]

Graham v. James laid down many basic principles of contract law dealing with licenses. Graham had an agreement with James wherein James had to develop a custom file retrieval program to be used with a CD-ROM compilation that Graham published. In return for developing the file retrieval program, James received a consideration of $1,000 for each new version of the CD-ROM, plus one dollar for each disc sold.[190] Subsequently, Graham parted ways with James and still he continued to use the program James wrote in subsequent versions of the CD-ROM.[191] Graham did not pay the promised royalties to James and also removed a copyright notice from the program's source code, which resulted in the lawsuit, in which James accused Graham of infringing the program's copyright.[192] The trial court held in favor of James on the copyright infringement claim and noted that, as an independent contractor, James owned the copyright in the program and permanently enjoined Graham from using it.[193] Graham appealed the order of the trial court.[194]

Graham contended that James could not make any claim for copyright infringement, and at best, James might be able to recover damages for the breach of contract resulting from the removal of the copyright notice and the failure to pay royalties.[195] James contended that Graham breached the conditions of the license agreement when he removed the copyright notice and failed to pay royalties.[196] The court rejected this contention and held that such activities were mere breaches of contractual covenants between the parties rather than a failure to satisfy the conditions of the license agreement.[197] Finally, James argued that even if the nonpayment of royalties and the removal of James's authorship credit amounted to mere breaches of covenants, these breaches terminated the license.[198] However,

[188] *Id.*

[189] 144 F.3d 229 (2d Cir. 1998).

[190] *Id.* at 233.

[191] *Id.*

[192] *Id.* at 234.

[193] *Id.*

[194] *Id.*

[195] *Id.* at 235.

[196] *Id.* at 236.

[197] *Id.* at 237. "Under New York law, there is a presumption that terms of a contract are covenants and not conditions. Contract obligations that are to be performed after partial performance are not treated as conditions, but as covenants." *Id.*

[198] *Id.* (citing Rano v. Sipa Press Inc., 987 F.2d 580, 586 (9th Cir. 1993) (relying upon the principle

the Court also rejected this contention by holding that rescission did not happen automatically upon a substantial breach, and James could not demonstrate any affirmative step taken by him, within a reasonable time, to rescind the license to Graham.[199] The *Graham* decision is important from the perspective of having written agreements regarding software and of drafting important provisions as clear-cut conditions to the license. Subsequently, courts in the United States decided similar issues relating to software licenses by applying the rationale of the *Graham* decision.[200]

B. *Jacobsen v. Katzer*[201]

The landmark judgment in *Jacobsen v. Katzer* has given the open source movement a new impetus by reversing the district court's decision that open source licensing agreements are only enforceable under contract law.[202]

Jacobsen was the manager of Java Model Railroad Interface ("JMRI"), an open source software group that essentially was a collective work of many participants that developed an application called DecoderPro.[203] Model railroad enthusiasts used DecoderPro to run and control model trains.[204] The code was available to the public on the JMRI website and distributed free of cost, but was subject to the terms and conditions of the Artistic License mentioned within the JMRI website.[205] Notably, Katzer offered a competing software product, called Decoder Commander.[206] One of Katzer's employees was alleged to have downloaded the decoder definition files from DecoderPro during development, and used portions of the code as part of the Decoder Commander software.[207] "The Decoder Commander Software files that

that a material breach of a covenant will allow the licensor to rescind the license)).

[199] *Id.* at 237–38.

[200] Sun Microsystems, Inc. v. Microsoft Corp., 2000 WL 33223397, at *3 (N.D. Cal. 2000). In this case the written contract Sun had with Microsoft required the latter's commercially distributed software, which it had developed with Sun's copyrighted software, be compatible with certain other software. The Court held that the compatibility provision in the said contract was a separate contractual covenant and not a restriction on the scope of the license. By citing the *Graham* case the Court noted that a copyright owner who had granted a nonexclusive license generally waived his right to sue the licensee for copyright infringement. *Id.* at *3.

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

[202] *See generally id.*

[203] *Id.* at 1376.

[204] *Id.*

[205] *Id.* at 1379 (explaining the artistic license required "that changes to the computer code be tracked so that downstream users know what part of the code is the original code created by the copyright holder and what part has been newly added or altered by another collaborator").

[206] *Id.*

[207] *Id.*

used DecoderPro definition files did not comply with the terms of the Artistic License."[208]

Jacobsen moved for a preliminary injunction, on the ground that violating "the terms of the Artistic License constituted copyright infringement and that, under Ninth Circuit law, irreparable harm could be presumed in a copyright infringement case."[209] After reviewing the Artistic License, the District Court held that the "Defendant's alleged violation of the conditions of the license might have constituted a breach of the nonexclusive license, but does not create liability for copyright infringement."[210] The District Court held that Jacobsen's cause of action only pertained to breach of contract, rather copyright infringement based on a breach of the conditions of the Artistic License.[211] The District Court denied the motion for a preliminary injunction because a breach of contract did not create a presumption of irreparable harm.[212]

Jacobsen appealed the rejection of his cause of action for copyright infringement to the Court of Appeals for the Federal Circuit.[213] Although the Federal Circuit had mostly been dealing with appeals concerning patents, it granted the request for appeal.[214] The Federal Circuit had the requisite jurisdiction because Jacobsen's complaint against Katzer in the district court included claims for declaratory judgment that Jacobsen did not infringe a patent issued to Katzer, and that certain patents held by Katzer were invalid, in addition to his claim for copyright infringement.[215]

The Federal Circuit noted that open source software projects invited computer programmers from different parts of the globe to observe software code as well as change and improve it.[216] Such collaborative enterprise can write and debug software programs considerably faster and at a much lower cost.[217] As a quid pro quo, "the copyright holder permit[ted] users to copy, modify and distribute the software code subject to conditions that serve to protect downstream users and to keep the code accessible."[218] The condition that users must copy and restate the license and attribution information allows "a copyright holder [to] ensure that recipients of the

[208] *Id*. (noting that "[s]pecifically, the Decoder Commander software did not include (1) the authors' names, (2) JMRI copyright notices, (3) references to the COPYING file, (4) an identification of SourceForge or JMRI as the original source of the definition files, and (5) a description of how the files or computer code had been changed from the original source code.").

[209] *Id.* at 1376–77.

[210] *Id.* at 1377.

[211] *Id.*

[212] *Id.*

[213] *Id.*

[214] *Id.* (finding "appellate jurisdiction under 28 U.S.C. § 1292(c)(1)," (citing 28 U.S.C. § 1338(a)).

[215] *Id.*

[216] *Id.* at 1378–79.

[217] *Id.* at 1379.

[218] *Id.*

redistributed computer code kn[e]w the identity of the owner as well as the scope of the license granted by the original owner."[219] The Artistic License at issue in this case also required collaborators to track changes to the code "so that downstream users knew what part of the computer code was the original code created by the copyright holder and what part had been newly added or altered by another collaborator."[220]

The conditions provided in the Artistic License were crucial because they enabled the copyright holder to retain the ability to benefit from the work of downstream users.[221] "By requiring that users who modify or distribute the copyrighted material retain the reference to the original source files, downstream users [were] directed to Jacobsen's website."[222] "Thus, downstream users kn[e]w about the collaborative effort to improve and expand the SourceForge project once they learn[ed] of the 'upstream' project from a 'downstream' distribution, and they [might] join in that effort."[223] A user who downloaded the JMRI copyrighted materials could modify and distribute the materials subject to restrictive terms of the Artistic License. A copyright holder could grant the right to make certain modifications, while also preventing other modifications.[224]

The Federal Circuit ultimately held that an express or an implied condition requiring a licensee to affix a proper copyright notice to all copies of the work that he publishes renders any copy published without the required notice unauthorized by the licensor and therefore an infringing act under copyright law.[225]

C. *Versata Software, Inc. v. Ameriprise Fin., Inc.*[226]

This case involved Versata's proprietary software product called Distribution Channel Management (DCM), that contained an open source XML parsing utility that was licensed under GPL Version 2 from a company called XimpleWare.[227] The problem arose when Versata licensed its DCM software to a financial services company Ameriprise, and thereafter filed a case for breach of license against Ameriprise when its subcontractor decompiled Versata's software.[228] Ameriprise then countersued Versata on the ground that Versata's software included open

[219] *Id.*
[220] *Id.*
[221] *Id.*
[222] *Id.* at 1381.
[223] *Id.*
[224] *Id.*
[225] *Id.* at 1382.
[226] No. A-14-CA-12-SS, slip op, 2014 WL 950065, (W.D. Tex. Mar. 11, 2014).
[227] *Id.* at *1.
[228] *Id.* at *1-2.

source software licensed under the GPL Version 2 and was a derivative work.[229] It further contended that the whole of Versata's DCM product came under the GPL Version 2 license, and therefore Ameriprise, or its subcontractor, could decompile and modify the software without any restriction.[230] In another interesting turn of events, when Ximpleware found out that Versata violated its open source parser, Ximpleware started legal action against Versata, Ameriprise and other Versata customers for copyright and patent infringement.[231]

Even though the parties settled this case in March 2015, it has given useful guidance on the interpretation of certain GPL 2 provisions, especially regarding the nature of distribution. The court held that distributing code to independent contractors is very much coming within the ambit of distribution envisaged by GPL as GPL did not differentiate between independent contractors and other third parties.[232] Dealing with the question on the liability of customers of infringing parties, the court held that the use of GPL Version 2 licensed code by such customers did not amount to infringement even though they got it from an infringing party.[233] According to the Court, the GPL Version 2 constituted a direct license from the intellectual property owner Ximpleware to the customers.[234]

D. *Oracle Am., Inc. v. Google Inc.*[235]

This ongoing case involves a dispute between Oracle and Google over copyright protection for Application Program Interfaces (APIs).[236] Many years back Google held discussions with Sun, who originally developed the programming language Java, about the manner in which Java APIs could be implemented in the open source Android mobile operating system.[237] The negotiations broke down because of "Google's refusal to make the implementation of its programs compatible with the Java virtual machine or interoperable with other Java programs."[238] Even though no license was granted, Google decided to implement the APIs in a manner that it thought did not infringe any of Sun's copyrights.[239] Oracle acquired Sun in 2010

[229] *Id.* at *2.

[230] *Id.* at *4.

[231] *Id. See also* Ximpleware v. Versata, 2014 U.S. Dist. LEXIS 165060 (N.D. Cal. Nov. 25, 2014).

[232] *Id.* at *32.

[233] *Id.* at *21.

[234] *Id.* at *22.

[235] 750 F.3d 1339 (Fed. Cir. 2014).

[236] *Id.* at 1347. API, as per Wikipedia, is "a set of routines, protocols, and tools for building software applications," details at https://en.wikipedia.org/wiki/Application_programming_interface (last visited Feb. 24, 2016).

[237] *Id.* at 1350.

[238] *Id.*

[239] *Id.*

and thereafter it decided to sue Google for violating its copyrights for structure, sequence, and organization of computer APIs.[240] The District Court decided the case in favor of Google and held that "the structure, sequence, and organization of a computer" API was a "system or method of operation" under 17 U.S.C.S. § 102(b), and therefore ineligible for copyright protection.[241] Oracle appealed to the Court of Appeals for the Federal Circuit, which reversed the decision of the trial court and held that the declaring code,[242] as well as the structure, sequence, and organization of the Java API packages at issue were entitled to copyright protection.[243] Here the court applied the famous abstraction-filtration-comparison test enunciated in the landmark case of *Computer Associates v. Altai*.[244] The case has been remanded to the trial court to determine the hear Google's fair use defense.[245] Google filed a petition for writ of certiorari, but the United States Supreme Court did not accept Google's appeal.[246]

E. Litigation Outside the United States

Outside the US there have been very few court decisions in the field of open source software. A German Court decided one such case, *AVM v. Cybits*, in 2011.[247] AVM manufactures digital subscriber line (DSL) terminals, like the FRITZ!Box router, and uses the Linux kernel in its production firmware. Cybits distributes Internet filtering software called "Surf-Sitter DSL," which aims to prevent children from encountering inappropriate media on the internet.[248] AVM sued Cybits in the regional court of Berlin to restrain Cybits from modifying the original FRITZ!Box firmware and loading it back to the DSL terminal. AVM included software parts licensed under the GPL as part of its claim.[249] The court held that users of GPL software were allowed to modify and install it, even if it was shipped as a part of an embedded device's firmware.[250] The Court also

[240] *Id.* at 1347.

[241] Oracle v. Google, 872 F.Supp.2d 974, 1002 (N.D. Cal. 2012).

[242] "Declaring code is the expression that identifies the prewritten function [of a program] and is sometimes referred to as the 'declaration' or the 'header.'" *Oracle*, 750 F.3d at 1349.

[243] *Id.* at 1381.

[244] *Id.* at 1357.

[245] *Id.* at 1381.

[246] Google, Inc. v. Oracle Am., Inc., 135 S. Ct. 2887 (Mem.) (2015).

[247] FREE SOFTWARE FOUNDATION EUROPE, *Summary of the Case so Far*, https://fsfe.org/activities/ftf/avm-gpl-violation.en.html (last visited Feb. 18, 2016) (summarizing in English). *See also* https://fsfe.org/activities/ftf/lg-urteil-20111118.pdf (last visited Feb. 18, 2016) (text of the full decision in the German language).

[248] *Id.*

[249] *Id.*

[250] FREE SOFTWARE FOUNDATION EUROPE, *Verdict in the case of AVM vs. Cybits Confirmed the View of FSFE*, https://fsfe.org/news/2011/news-20111201-02.en.html (last visited Feb. 24, 2016).

rejected AVM's argument that Cybits infringed AVM's copyright by distributing Surf-Sitter DSL.[251] The Court noted that the GPL portions of the firmware could be lawfully reproduced and modified.[252]

X. OPEN SOURCE HARDWARE

Another interesting emerging trend open source field is *open source hardware*, the physical equivalent of open source software.[253] This movement "advocates the public provision of hardware design documentation."[254] Thus the purchaser of an open hardware circuit board gets access to the circuit board's full design documentation, along with the source code of all accompanying software. It is well accepted that "open principles of design have been central to the creation of significant pieces of software" like Linux and Android operating systems.

Numerous economic and legal obstacles face those who are striving to develop a sustainable model for open source hardware.[255] From an economic point of view, open source hardware and open source software are different in important ways.[256] First, while software can be compiled from source code and distributed at almost zero cost, manufacturing a piece of hardware is more involved and costly, and requires sourcing appropriate materials and designing a complex manufacturing process.[257] Secondly, although there is no cost to copy and distribute software, reproducing and distributing open source hardware involves substantial costs.[258] Thirdly, open source hardware communities may find it very hard to compete with the economies of scale of commercial manufacturers, as the cost of testing and making and improving hardware prototypes can be very costly.[259] Similar testing of open source software is very easy to do.[260]

Open source hardware also needs to overcome some legal challenges. Open source software can depend upon copyright law to protect its rights.

[251] *Id.*

[252] *Id. See also* Heather J. Meeker, *Open Source and the Age of Enforcement*, 4 HASTINGS SCI. & TECH. L.J. 267, 283–84 (2012).

[253] Robert Viseur, *From Open Source Software to Open Source Hardware*, *in* OPEN SOURCE SYSTEMS: LONG-TERM SUSTAINABILITY 288 (Imed Hammouda et al. eds., 2012).

[254] *Id.*

[255] *See* ALICIA GIBB, BUILDING OPEN SOURCE HARDWARE: DIY MANUFACTURING FOR HACKERS AND MAKERS, at xvi (2015) (explaining the "Open Source Hardware Movement" and the challenges that come with its advancement).

[256] *Id.*

[257] Christina Raasch, *Product Development in Open Design Communities: A Process Perspective*, 8 INT'L J. INNOVATION & TECH. MGMT. 557, 568 (2011).

[258] *Id.*

[259] *Id.*

[260] Jim Turley, *Open-Source Hardware*, EMBEDDED, http://www.embedded.com/electronics-blogs/significant-bits/4023974/Open-Source-Hardware (last visited Feb. 24, 2016).

Because all original software works are automatically protected by copyright, a license is required to lawfully reproduce, modify or distribute software. This copyright license can be customized to incorporate certain conditions, like "copyleft," present in the most prominent open source software license, the GPL.[261] As discussed in detail above, the GPL provides licensees broad rights to copy, modify and distribute licensed works, subject to the condition that all works based on the licensed code be available under the terms of the GPL.[262] However, scholars note that these relatively simple legal measures are unavailable in the open source hardware context because useful physical objects are generally not protected by copyright.[263] Consequently, the concept of open source hardware is still in its infancy and is yet to come of age.

XI. CONCLUSION

It is evident that the open source movement has given society innovative and value added software projects, including Linux and Apache, and thus has augmented the public domain. Similarly, the open source movement in the field of software has also triggered similar movements like Creative Commons licenses that deal with content such as music, literary works like books, and other creative materials covered by copyright.[264] From the field of software and books, the open licensing movement has spread to sectors like education, and the emergence of Massive Open Online Courses (MOOCs) is an example of the movement being replicated in various sectors.[265] If implemented properly, MOOCs can extend the reach of educational institutions to a much wider audience and thereby improve access to education.[266]

Thus the open source licensing movement in general has undoubtedly contributed to the progress of the new information society and has laid a strong foundation for economic growth. The emergence of open license initiatives has effectively checkmated forceful corporate assertions of IP rights, which were becoming widespread in all spheres of life affecting the common people. The open source licensing movement has given voice and space to many who dreamt of a world of innovation and creativity completely disconnected from the analysis of balance sheets and financial

[261] GNU *supra* note 77.

[262] *Id.*

[263] Eli Greenbaum, *Three-Dimensional Printing and Open Source Hardware*, 2 N.Y.U. J. OF INTELL. PROP. & ENT. LAW 257, 261 (2013).

[264] *See* CREATIVE COMMONS, http://creativecommons.org/about (last visited Feb. 18, 2016).

[265] *See* Carl Straumsheim, *MOOC Provider edX Finds a Powerful Partner in Google for its New Platform, Open edX*, INSIDE HIGHER ED (Sept. 11, 2013), https://www.insidehighered.com/news/2013/09/11/edx-and-google-develop-open-source-mooc-platform.

[266] *Id.*

statements that usually takes place in corporate boardrooms. The open source movements discussed in this paper give strong examples of how such movements often encourage innovation and creativity.

The open source license movement can provide free public goods to society, and law and public policy should support it. Thus, the need of the hour is to proliferate this movement so that developing countries like India can benefit and the public domain can be enhanced. Any initiative on the legal or policy front towards extending support to the open source movement will be a step in the right direction.