

# ***Open Source, Free Software and Contractual Issues***

Prof. Dr. José J. González de Alaiza Cardona (LLM, Harvard)

# Table of contents.

<b>Table of contents.....</b>	<b>1</b>
<b>I.– Introduction.....</b>	<b>3</b>
<b>II.– General Part.....</b>	<b>5</b>
1. Open source numbers. ....	5
2. Technical concepts. ....	8
3. Historic overview. ....	11
4. Free software foundations and perspectives.....	17
I. Introduction. ....	17
II. Definition. ....	18
III. Moral foundations. ....	19
IV. Legal approach. ....	22
V. Software development. ....	24
VI. Economic models. ....	28
5. Free software v. Open Source Software. ....	30
I. Introduction. ....	30
II. Terminology.....	31
III. Definitions. ....	33
IV. Philosophy.....	36
V. Representative licenses: GNU GPL v. BSD license.....	37
VI. Conclusions. ....	38
<b>III.– Contractual Issues.....</b>	<b>40</b>
1. Introduction. ....	40
2. Contract v. License.....	42
I. Relevance of the discussion. ....	42
II. Conceptual approach.....	43
III. The dual solution. ....	46
3. Consideration.....	50
I. Introduction. ....	50
II. Definition. ....	50
III. Peppercorn theory.....	51
IV. Consideration in copyleft software licenses.....	51
4. Clickwrap and shrinkwrap agreements. ....	52
I. Introduction. ....	52
II. Shrinkwrap and clickwrap licenses in the Courts. ....	54
III. Enforceability of copyleft licenses as shrinkwrap or clickwrap contracts. ....	59
IV. Unenforceability of copyleft licenses as shrinkwrap or clickwrap contracts.....	62
5. Privity. ....	63
I. Introduction. ....	63
II. Privity concerns in copyleft software licenses.....	64
III. Relativization of privity concerns in copyleft software licenses. ....	65
6. Warranty disclaimer. ....	67
I. Introduction. ....	67
II. Merchants.....	68
III. Non-merchants. ....	70

7. Copyright v. Contract.....	71
I. Introduction.....	71
II. Preemption.....	72
III. Deviations of the copyleft licenses from the Copyright language.....	74
<b>IV.– Conclusion.....</b>	<b>75</b>
<b>Bibliography.....</b>	<b>79</b>
<b>Other materials.....</b>	<b>84</b>

## **I.– Introduction.**

“Free software” is an increasingly used form to license computer programs, which on the one hand gives users the rights to use, modify and redistribute the program; and, on the other, forces any person redistributing an original or modified version of the program to license it with the same rights. Such a forced obligation is introduced through the so called “copyleft clause” and, basically, uses Copyright in a creative way to achieve freedom instead of control.

This paper discusses the “free software” foundations and contractual issues. The discussion is structured in two main parts and Conclusion. In Part II, the emergence of “free software” and its implications in different fields will be exposed. I will seek to explain how the “copyleft clause” affects the ways in which software is developed and distributed. Moreover, I will explain the common points and differences between “free software” and “open source software”.

In Part III, the contractual issues raised by the peculiarity of the “copyleft clause” will be addressed. I will argue that the license agreement which contains the “copyleft clause” is not a mere copyright non-contractual license, but a contract. This fact triggers a number of contract related questions, which I will seek to resolve from the U.S. perspective. In particular, I will address the concerns about lack of consideration; validity of clickwrap and shrinkwrap licenses; possible consequences of lack of privity between licensor and licensee; the enforceability of the warranty disclaimer included in most copyleft licenses; and the relation between Copyright and contractual provisions.

Finally, the paper will end summarizing the main conclusions drawn in the two main parts mentioned above.

## II.– General Part.

### 1. Open source numbers.

Nowadays the terms “free software” and “open source”<sup>1</sup> are as familiar to every computer user as “mouse”, “keyboard” or “laptop”. Even more important than that, the use of open source software is widely spread in different fields.<sup>2</sup> In particular, its success is remarkable in the Internet. Just a few examples, almost 70% of the web servers<sup>3</sup> are run by *Apache*,<sup>4</sup> a well known open source software; a vast majority<sup>5</sup> of the domain name servers (*DNS*)<sup>6</sup> use *BIND*,<sup>7</sup> another open source software; the browser *Firefox* counts over

---

<sup>1</sup> “Free software and Open Source describe the same category of software, more or less, but say different things about the software, and about values”, Richard Stallman, *The GNU Operating System and the Free Software Movement in OPEN SOURCES – VOICES FROM THE OPEN SOURCE REVOLUTION*, 53, 70 (Chris DiBona et al. eds., 1999). Also printed in RICHARD STALLMAN, *FREE SOFTWARE, FREE SOCIETY: SELECTED ESSAYS OF RICHARD M. STALLMAN* 15 (Joshua Gay ed., 2002). In section I.5, I will discuss the differences between the two concepts.

<sup>2</sup> Main stream products “get applications, trained users, and momentum that reduces future risk”, David A. Wheeler, *Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!* (revised as November 14, 2005), available at [http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html).

<sup>3</sup> “A Web server is a program that, using the client/server model and the World Wide Web's Hypertext Transfer Protocol (HTTP), serves the files that form Web pages to Web users (whose computers contain HTTP clients that forward their requests). Every computer on the Internet that contains a Web site must have a Web server program”, <http://www.whatis.com>. In other words, a web server gets the user the web page that she has requested.

<sup>4</sup> A recent Netcraft's survey (March 2006) found that of all the web sites they could find, counting by name –web server hostnames rather than physical computers–, *Apache* had 68.70% of the market, *Microsoft* had 20.51%, *Sun* had 2.43%, and *Zeus* had 0.74%, [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html).

<sup>5</sup> Representing 95%, according to Bill Manning's survey (2q2000), <http://www.isi.edu/~bmanning/in-addr-versions.html>.

<sup>6</sup> “Domain Name System (or Service or Server), an Internet service that translates domain names [e.g., [www.harvard.edu](http://www.harvard.edu)] into IP addresses. Because domain names are alphabetic, they're easier to remember. The Internet however, is really based on IP addresses. Every time you use a domain name, therefore, a DNS service must translate the name into the corresponding IP address.” <http://www.webopedia.com>.

<sup>7</sup> “BIND (Berkeley Internet Name Domain) is an implementation of the Domain Name System (DNS) protocols and provides an openly redistributable reference implementation of the major components of the Domain Name System, including: a Domain Name System server (named); a Domain Name System resolver library; and tools for verifying the proper operation of the DNS server”, <http://www.isc.org/index.pl?sw/bind/>.

150 millions downloads;<sup>8</sup> and the web client *Thunderbird* received over one million downloads in the ten days following its release, on December 7, 2004.<sup>9</sup>

Beyond the Internet, open source software is often used to control electronic devices, from antilock brakes to watches to consumer electronics (i.e., mobile phones, PDAs and TV set-top boxes) to medical equipment, etc.<sup>10</sup>

A third area where the use of open source is increasing, but only has a moderate market share, is software for desktop computers.<sup>11</sup> This is true both for operating systems and applications.<sup>12</sup> In relation to the former, *GNU/Linux* claims around 29 millions users,<sup>13</sup> far

---

<sup>8</sup> <http://www.spreadfirefox.com/node/22360>. *Firefox* was released on November 9, 2004 and 99 days later had been downloaded 25 million times. On March 3, 2006, the numbers of downloads was already 150 millions. However, *Firefox* has still a modest market share (7%) in relation with his main competitor, *Internet Explorer* (90%, adding versions 6.x and 5.x), March 2006, <http://www.thecounter.com/stats/>. The use of *Netscape*, the other open source browser, is under 1% and receding.

<sup>9</sup> See <http://www.spreadfirefox.com/node/8864> and [http://en.wikipedia.org/wiki/Mozilla\\_Thunderbird](http://en.wikipedia.org/wiki/Mozilla_Thunderbird). Unlike numbers for web servers, a central site cannot get statistics about e-mail clients unless it receives mail from the client. That makes it much difficult to obtain this information. However, due to the fact that non-open source web clients as Microsoft Outlook and Microsoft Outlook Express are installed simultaneously with the Windows operating system, one may assume that Thunderbird is clearly behind his opponents.

<sup>10</sup> LINUS TORVALDS, JUST FOR FUN – THE STORY OF AN ACCIDENTAL REVOLUTIONARY 161 (2001); Jason B. Wacha, *Taking the Case: is the GPL Enforceable*, 21 Santa Clara Computer and High Technology Law Journal, 451, 452 (2005). More examples under, Paul Gustafson & William Koff (directors), CSC's Leading Edge Forum, Open Source: Open for Business, 55 and ff. (2004), available at [http://www.csc.com/features/2004/uploads/LEF\\_OPENSOURCE.pdf](http://www.csc.com/features/2004/uploads/LEF_OPENSOURCE.pdf).

<sup>11</sup> In a survey conducted over 1,452 companies and public institutions in Germany, Sweden and UK from June 2001 to June 2002, it was observed that "the use of Open Source software on client or desktop computers is not very widespread. Only about 20% of those establishments that use OSS have some form of OSS installed on their desktops." Thorsten Wichmann, *Part I: Use of Open Source Software in Firms and Public Institutions in FLOSS Final Report, Free/Libre Open Source Software: Survey and Study 41* (2002), available at <http://www.infonomics.nl/FLOSS/report/>.

<sup>12</sup> "An operating system [i.e., *GNU/Linux*, *Windows*...] is the program that, after being initially loaded into the computer by a boot program, manages all the other programs in a computer. The other programs are called applications or application programs [e.g. *OpenOffice* and *Microsoft Office*]. The application programs make use of the operating system by making requests for services through a defined application program interface." <http://www.whatis.com>.

<sup>13</sup> That was an approximate guess of the Linux Counter Project in March 2005, <http://counter.li.org/>. Richard Stallman, the main developer of the GNU/Linux operating system, stated in a speech given at New York University on May 29, 2001: "In general, in business most users are not using GNU/Linux. Most home users are not using our system yet. When they are, we should automatically get 10 times as many

behind the numbers of the non-open source operating systems provided by *Microsoft*, the *Windows* family (i.e., *Windows XP*, *Windows Me*, *Windows 2000* and so on).<sup>14</sup> Among the applications the most popular are probably *OpenOffice*<sup>15</sup> and *MySQL*,<sup>16</sup> which occupy again a very small market share in comparison with *Microsoft Office*.<sup>17</sup>

The development of these numbers will depend mainly on the support that major companies, with the power to influence the technological market, will offer to open source software. On the one hand, it is well known the fight of *Microsoft* against the open source movement.<sup>18</sup> On the other hand, a fair number of large companies, such as *IBM*, *Intel*, *Hewlett-Packard*, *Oracle*, *SAP*, *Sun Microsystems*, *Dell*, *Motorola*, *Sony* and so on, are supporting this movement.<sup>19</sup> Another decisive factor in the success of open source will be the attitude of the public institutions. So far, at least the governments of

---

volunteers and 10 times as many customers for the free software businesses that there will be. And so that will take us that order of magnitude.” RICHARD STALLMAN, *supra* note 1, at 179.

<sup>14</sup> Nevertheless, it may be surprising that there is a high percentage –70%– of large companies that use GNU/Linux as operating system, Ted Shadler, *Your Open Source Strategy*, 3 (2003), available at [http://www.redhat.com/whitepapers/forrester/Forrester\\_OSS\\_Sep.pdf](http://www.redhat.com/whitepapers/forrester/Forrester_OSS_Sep.pdf). Moreover, GNU/Linux count on his site very notable customers, as the NASA, LINUS TORVALDS, *supra* note 10, at 161; or the U.S. Army, Paul Gustafson & William Koff, *supra* note 10, at 59.

<sup>15</sup> *OpenOffice* is an application suite —a set of applications that are designed to work together, which includes a word processor, spreadsheet, presentation graphics and drawing program and provides access to popular databases. *OpenOffice* runs on Linux, Windows, Mac and other operating systems.

<sup>16</sup> *MySQL* is a database management system.

<sup>17</sup> “First released in 2000, OpenOffice has a 14 percent share of the large enterprise office systems market, according to Forrester Research, suggesting it is becoming a true alternative to Microsoft Office, which holds a 94 percent share of the overall office market, has been around for over a decade, and claims 300 million users worldwide. Microsoft, which in general has scoffed the open source approach, nonetheless recognizes the threat,” Paul Gustafson & William Koff, *supra* note 10, at 60. On the other hand, *MySQL* was declared to be used by 6% of U.S. large companies, Ted Shadler, *Your Open Source Strategy*, 3 (2003), available at [http://www.redhat.com/whitepapers/forrester/Forrester\\_OSS\\_Sep.pdf](http://www.redhat.com/whitepapers/forrester/Forrester_OSS_Sep.pdf). However, *MySQL* states to be the third top deployed database, after *SQL Server* and *Oracle*, see <http://www.mysql.com/why-mysql/marketshare/>.

<sup>18</sup> A leaked confidential document, known as Halloween I (author: Vinod Valloppillil; date: August 1998), explained the thread of open source for Microsoft and the ways to compete with, STEVEN WEBER, THE SUCCESS OF OPEN SOURCE, 126, 127 (2004). This memorandum has become over the time in a series, which has been annotated and published by Eric. S. Raymond, available at <http://www.catb.org/~esr/halloween/>.

<sup>19</sup> Paul Gustafson & William Koff, *supra* note 10, at 3. See as well, Jason B. Wacha, *supra* note 10, at 452.

Argentina, Australia, Brazil, China, Denmark, France, Germany, India, Korea, Japan, Peru, the United States and even the United Nations have adopted initiatives or are already using open source software.<sup>20</sup>

## *2. Technical concepts.*

Understanding of the legal issues related to open source software requires some basic knowledge about computer programming, the subject matter that we are dealing with. First of all, it must be distinguished between hardware and software. “In information technology, hardware is the physical aspect of computers, telecommunications, and other devices. The term arose as a way to distinguish the "box" and the electronic circuitry and components of a computer from the program you put in it to make it do things.”<sup>21</sup> And “[s]oftware is a general term for the various kinds of programs used to operate computers and related devices. (...) Software can be thought of as the variable part of a computer and hardware the invariable part. Software is often divided into application software<sup>22</sup> (programs that do work users are directly interested in) and system software (which includes operating systems<sup>23</sup> and any program that supports application software).”<sup>24</sup> This paper will focus on the legal aspects of exploiting software as open source. But what does “open source” mean?

---

<sup>20</sup> Paul Gustafson & William Koff, *supra* note 10, at 2. See as well, Jordi Mas I Hernández, *Software libre en el sector público*, 9 and 10 (2003), available at <http://www.uoc.edu/dt/20327/20327.pdf>.

<sup>21</sup> <http://www.whatis.com>.

<sup>22</sup> *Supra* note 12.

<sup>23</sup> *Ib.*

<sup>24</sup> <http://www.whatis.com>.

“The source code consists of the programming statements that are created by a programmer with a text editor<sup>25</sup> or a visual programming tool and then saved in a file. For example, a programmer using the C language<sup>26</sup> types in a desired sequence of C language statements using a text editor and then saves them as a named file. This file is said to contain the source code.”<sup>27</sup> The source code can be read and modified by other programmers. However, a computer does not understand these instructions, but only binary inputs –1s and 0s.

To convert the source code into something readable for the computer, the programmer uses a compiler, which is a “special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses.”<sup>28</sup> The resulting output, the compiled file, is known as machine code or object code.<sup>29</sup> In opposition to the source code, the object code file contains a sequence of instructions that the processor can understand or execute but that are almost impossible for a human to read because they consist entirely of numbers. In short, “[s]ource code is what programmers write; object code is what computers run”.<sup>30</sup>

---

<sup>25</sup> “A text editor is a computer program that lets a user enter, change, store, and usually print text”, <http://www.whatis.com>. “The distinction between editors and word processors is not clear-cut, but in general, word processors provide many more formatting features. Nowadays, the term editor usually refers to source code editors that include many special features for writing and editing source code.” <http://www.webopedia.com>.

<sup>26</sup> A programming language is a “vocabulary and set of grammatical rules for instructing a computer to perform specific tasks. The term programming language usually refers to high-level languages, such as BASIC, C, C++, COBOL, FORTRAN, Ada, and Pascal. Each language has a unique set of keywords (words that it understands) and a special syntax for organizing program instructions. High-level programming languages, while simple compared to human languages, are more complex than the languages the computer actually understands, called machine languages.” <http://www.webopedia.com>

<sup>27</sup> <http://www.webopedia.com>.

<sup>28</sup> <http://www.whatis.com>.

<sup>29</sup> RICHARD STALLMAN, *supra* note 1, at 3.

<sup>30</sup> Jonathan Zittrain, *Normative Principles for Evaluating Free and Proprietary Software*, University of Chicago Law Review, Vol. 71, No. 1, 265, 271 (2004).

In some legal articles, the relationship between source code and object code has been analogized to that between recipe and dish. The source code is like a recipe, which one cannot eat, but it allows one to cook the dish. On the other hand, if one has only the dish, you may enjoy it, but not improve it, because of the lack of knowledge about its ingredients and quantities.<sup>31</sup>

In the vast majority of purchases and any other acquisitions of software, only the object code is delivered but not the source code. That may mean little for the typical user, who is only interested in running the program and would not even know what to do with the source code. However, it is easy to realize that if the source code were released, a small number of users who are computer experts could customize or improve the program.<sup>32</sup>

Finally, it may be asked if it is possible to extract the source code from the object code. The answer is in the affirmative, and this can eventually be done through decompilation.<sup>33</sup> However, this process implicates certain technical and legal issues. Firstly, the decompilation of object code into source code is not a straightforward technological method; on the contrary, it may fail because of various reasons.<sup>34</sup>

---

<sup>31</sup> *Id.* at 265, 271; Jordi Mas I Hernández, *supra* note 20, at 1, 2.

<sup>32</sup> Christian H. Nadan, *Open Source Licensing: Virus or Virtue?*, 10 Texas Intellectual Property Law Journal, 349, 351 (2002).

<sup>33</sup> “To decompile is to convert executable (ready-to-run) program code (sometimes called object code) into some form of higher-level programming language so that it can be read by a human. Decompilation is a type of reverse engineering that does the opposite of what a compiler does.” <http://www.whatis.com>.

<sup>34</sup> “Decompilation is not always successful for a number of reasons. It is not possible to decompile all programs, and data and code are difficult to separate, because both are represented similarly in most current computer systems. The meaningful names that programmers give variables and functions (to make them more easily identifiable) are not usually stored in an executable file, so they are not usually recovered in

Moreover, a successful decompilation does not provide us with the original source code, but only with a source code which is functionally equivalent to the original<sup>35</sup> and usually much more difficult to maintain. Secondly, decompilation involves reproductions of the decompiled computer program or parts of it and often circumvention of technological protection measures. Both will most certainly violate intellectual property laws. Although most countries provide exceptions for reverse engineering and, therefore, for decompilation, such exceptions are only applicable under certain circumstances – basically, this action has to be undertaken to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs.<sup>36</sup>

### *3. Historic overview.*

In the 1960s the distinction between hardware and software was not as clear as nowadays. The first computers were designed to perform one or a few specific tasks. And early programs were designed by machine manufacturers to be used in conjunction with a specific computer model or individual machine.<sup>37</sup> The users of computers were mainly companies and governmental institutions, which purchased the hardware and received the software as part of the deal. The business model of selling the software did not exist as such.<sup>38</sup> In this scenario, the producers of software usually delivered the program with its

---

decompiling. (...) Programs can be designed to be resistant to decompilation through protective means such as obfuscation.” <http://www.whatis.com>.

<sup>35</sup> Jonathan Zittrain, *supra* note 30, at 271.

<sup>36</sup> See § 1201(f) Copyright Act and art. 6 Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs.

<sup>37</sup> Final Report of the National Commission on New Technological Uses of Copyrighted Works 11 (1978), available at <http://digital-law-online.info/CONTU/PDF/>.

<sup>38</sup> JESÚS GONZÁLEZ BARAHONA ET AL., INTRODUCCIÓN AL SOFTWARE LIBRE 32 (2003).

source code, because they did not have any interest in hiding it. Moreover, users and computer scientists used to share the source codes of the programs without limitations.<sup>39</sup>

For these reasons, it has been claimed that the free software preceded the distribution of software as a protected work.<sup>40</sup> That may have been the situation *de facto*, but it cannot be stated that users had a right to reproduce the programs or to access to the source code without the authorization of the author or producer. It is true that the kind of legal protection deserved by computer programs was still unclear at the time,<sup>41</sup> but even at a very preliminary stage the consensus was that computer programs were somehow protectable.<sup>42</sup>

The distribution of computer programs independent from a particular machine began to grow up gradually.<sup>43</sup> And taking into account that the production cost of a program was,

---

<sup>39</sup> *Id.*; Richard Stallman, *supra* note 1, at 53.

<sup>40</sup> Richard Stallman, *supra* note 1, at 53; Miquel A. Zarza, *¿Qué es el Software Libre? Regulación y Autoorganización in* INTERNET Y PLURALISMO JURÍDICO: FORMAS EMERGENTES DE REGULACIÓN 297 (Pompeu Casanovas ed., Comares 2003).

<sup>41</sup> Before the adoption of a legal response, the literature was divided between the protection through the copyright laws or through the patents system. *Vid.* Pamela Samuelson et al., *Manifesto concerning the legal protection of Computer Programs*, 94 *Columbia Law Review* 2308, 2310 (1994).

<sup>42</sup> Although the 1976 Copyright Act did not address the issue of the copyrightability of computer programs, in 1964, the Register of Copyrights announced that computer programs would be accepted for registration, provided that (1) they contained sufficient original authorship, (2) they had been published, and (3) copies submitted for registration were in human-readable form” and the National Commission on New Technological Uses of Copyrighted Works stated that “it was clearly the intent of Congress to include computer programs within the scope of copyrightable subject matter in the Act of 1976”, *Final Report of the National Commission on New Technological Uses of Copyrighted Works* 15 and 16, respectively (1978), available at <http://digital-law-online.info/CONTU/PDF/>.

<sup>43</sup> The National Commission on New Technological Uses of Copyrighted Works described this process in a very detailed form: “As the number of computers has increased dramatically, so has the number of programs with which they may be used. While the first computers were designed and programmed to perform one or a few specific tasks, an ever increasing proportion of all computers are general-purpose machines which perform diverse tasks, depending in part upon the programs with which they are used. Early programs were designed by machine manufacturers to be used in conjunction with one model or even one individual computer. Today, many programs are designed to operate on any number of machines from one or more manufacturers. In addition, and perhaps even more importantly, there is a growing proportion

and it is still today, far beyond that of making copies of the program, a certain legal standard had to be achieved.

In 1980 the U.S. amended the Copyright Act extending its protection to computer programs.<sup>44</sup> This decision probably influenced Japan and the European Union<sup>45</sup> to take steps in the same direction. Finally, during the 1990s WTO<sup>46</sup> and WIPO,<sup>47</sup> the international organizations regulating copyright issues, agreed on the protection of computer programs under Copyright law.

At the same time, software developers stop delivering computer programs in source code. In their view, the exclusive distribution of the object code has two advantages: firstly, the product is more appealing for the standard and non-sophisticated software users; and

---

of programs created by persons who do not make machines. These people may be users or they may be – and increasingly are – programmers or small firms who market their wares for use by individual machine owners who are not in a position to write their own programs”, Final Report of the National Commission on New Technological Uses of Copyrighted Works 10, 11 (1978), available at <http://digital-law-online.info/CONTU/PDF/>.

<sup>44</sup> Pub. L. No. 96-517, 94 Stat. 3015, 3028. Although computer programs are not expressly quoted in the works of authorship list of § 102(a) U.S. Copyright Act, the introduction of a computer software definition in § 101 (“A “computer program” is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result”) and an exception to make copies or adaptations to use the computer program or for archival purposes in § 117 were enough for the Courts to consider that “the copyrightability of computer programs is firmly established after the 1980 amendment to the Copyright Act”, *Williams Electronics, Inc. v. Artic Intern., Inc.*, 685 F.2d 870, 875 (1982); in the same sense, *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1248 (1983).

<sup>45</sup> Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs (Official Journal L 122, 17/05/1991 p. 42).

<sup>46</sup> “Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention (1971)”, art. 10(1), Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPs) (1994).

<sup>47</sup> “Computer programs are protected as literary works within the meaning of Article 2 of the Berne Convention. Such protection applies to computer programs, whatever may be the mode or form of their expression”, art. 4 WIPO Copyright Treaty (adopted in Geneva on December 20, 1996).

secondly, it protects the source code from disclosure and, therefore, from a possible modification.<sup>48</sup>

As a reaction against the distribution of computer programs in object code, Richard Stallman founded in 1985 the Free Software Foundation,<sup>49</sup> which coordinated since then the efforts of the free software movement.<sup>50</sup> This operating system developer, as a researcher at the MIT Artificial Intelligence Lab, experienced with great dislike not only the new ways of distributing software, but also the transformation in the ways of producing it. Now software developers had to sign a nondisclosure agreement to get access to the software, even as object code.<sup>51</sup>

Stallman advocates that no one should have to pay for software<sup>52</sup> and that the access to the source code should be granted.<sup>53</sup> If a different system governs the production and distribution of software, a few people would dominate computing.<sup>54</sup> Stallman compares his free software system with a non-free or proprietary software system and considers the

---

<sup>48</sup> Christian H. Nandan, *supra* note 32, at 351.

<sup>49</sup> <http://www.fsf.org>.

<sup>50</sup> “Richard Stallman is the God of Free Software. (...) Basically, he pioneered the notion of free source-code availability as something intentional, not just an accident, the way it happened with original Unix open development.” LINUS TORVALDS, *supra* note 10, at 58. Eric S. Raymond, *A Brief History of Hackerdom in OPEN SOURCES – VOICES FROM THE OPEN SOURCE REVOLUTION*, 19, 24 (Chris DiBona et al. eds., 1999).

<sup>51</sup> Richard Stallman, *supra* note 1, at 54.

<sup>52</sup> “[Y]ou should be free to redistribute copies, either with or without modifications, either gratis or charging a fee for distribution, to anyone anywhere. Being free to do these things means (among other things) that you do not have to ask or pay for permission.” In other words, “you may have paid money to get copies of free software, or you may have obtained copies at no charge. But regardless of how you got your copies, you always have the freedom to copy and change the software...”, RICHARD STALLMAN, *supra* note 1, at 15, available at <http://www.gnu.org/philosophy/free-sw.html>.

<sup>53</sup> “In order for freedoms 1 and 3 (the freedom to make changes and the freedom to publish improved versions) to be meaningful, one must have access to the source code of the program. Therefore, accessibility of source code is a necessary condition for free software.” RICHARD STALLMAN, *supra* note 1, at 41, available at <http://www.gnu.org/philosophy/free-sw.html>.

<sup>54</sup> Chris DiBona et al., *Introduction in OPEN SOURCES – VOICES FROM THE OPEN SOURCE REVOLUTION*, 1, 2 (Chris DiBona et al. eds., 1999).

later antisocial.<sup>55</sup> In opposition to free software, the proprietary software is characterized by the fact that “[i]ts use, redistribution or modification is prohibited, or requires you to ask for permission, or is restricted so much that you effectively can't do it freely.”<sup>56</sup>

Taking into account that the proprietary software was flooding the market, Stallman began to create his own software, which would be controlled under the free movement principles.<sup>57</sup> The first step in order to build a consistent alternative to the proprietary software was to have an operating system,<sup>58</sup> without which you cannot even run a computer.<sup>59</sup> This ongoing operating system was called GNU.<sup>60</sup> Doing that from scratch is a huge task and therefore Stallman asked at an early stage of the project for help and money.<sup>61</sup>

An important strategic decision in the development of GNU was to make the system compatible with Unix.<sup>62</sup> That had three main advantages: firstly, Unix design was already

---

<sup>55</sup> Richard Stallman, *supra* note 1, at 54.

<sup>56</sup> <http://www.gnu.org/philosophy/categories.html>.

<sup>57</sup> “So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free.” Richard Stallman, *Initial Announcement of the GNU Project*, available at <http://www.gnu.org/gnu/initial-announcement.html>.

<sup>58</sup> See footnote n. 12.

<sup>59</sup> Richard Stallman, *supra* note 1, at 55.

<sup>60</sup> The name GNU was chosen following a hacker tradition, as a recursive acronym for “GNU’s Not Unix.” Richard Stallman, *supra* note 1, at 56.

<sup>61</sup> In September 1983, Stallman made the Initial Announcement of the GNU Project. And longer version called the GNU Manifesto was published in September 1985. Richard Steuer, *Overview of the GNU System*, available at <http://www.gnu.org/gnu/gnu-history.html>. Moreover, in 1985 was created the Free Software Foundation, a tax-exempt charity for free software development. Richard Stallman, *supra* note 1, at 60.

<sup>62</sup> “Pronounced *yoo-niks*, a popular *multi-user, multitasking* operating system developed at Bell Labs in the early 1970s. Created by just a handful of programmers, UNIX was designed to be a small, flexible system used exclusively by programmers.

UNIX was one of the first operating systems to be written in a high-level programming language, namely C. This meant that it could be installed on virtually any computer for which a C compiler existed. This

proven; secondly, it was portable;<sup>63</sup> and thirdly, Unix users could switch easily to GNU.<sup>64</sup> The Unix operating system and its source code were delivered by its copyright owner (AT & T) to many academic and research institutions around the world to allow its study, improvement and enlargement. Unix was not free software because it was not licensed to everybody and its redistribution was not free.<sup>65</sup> However, the free software group found in Unix the perfect model to develop its own operating system, replacing component by component.<sup>66</sup> This job took over 5 years, from 1984 to 1990. By then, only one piece of the new operating system that was being built was lacking: the kernel.<sup>67</sup>

The kernel is the “central module of an operating system. It is the part of the operating system that loads first, and it remains in main memory [or RAM. It provides] all the essential services required by other parts of the operating system and applications. Typically, the kernel is responsible for memory management, process and task management, and disk management.”<sup>68</sup> The GNU community was taking longer than

---

natural portability combined with its low price made it a popular choice among universities. (It was inexpensive because antitrust regulations prohibited Bell Labs from marketing it as a full-scale product.) Bell Labs distributed the operating system in its source language form, so anyone who obtained a copy could modify and customize it for his own purposes. By the end of the 1970s, dozens of different versions of UNIX were running at various sites.

After its breakup in 1982, AT&T began to market UNIX in earnest. It also began the long and difficult process of defining a standard version of UNIX.

Due to its portability, flexibility, and power, UNIX has become a leading operating system for workstations. Historically, it has been less popular in the personal computer market.”

<sup>63</sup> “When used to describe software, portable means that the software has the ability to run on a variety of computers. Portable and machine independent mean the same thing -that the software does not depend on a particular type of hardware.” <http://www.webopedia.com>.

<sup>64</sup> Richard Stallman, *supra* note 1, at 56; Richard Steuer, *supra* note 61.

<sup>65</sup> JESÚS GONZÁLEZ BARAHONA ET AL., *supra* note 38, at 35.

<sup>66</sup> Richard Stallman, *supra* note 1, at 62. The recursive acronym GNU (see foot note n. 58) pays tribute to the importance of Unix in the GNU Project.

<sup>67</sup> Richard Steuer, *supra* note 61.

<sup>68</sup> <http://www.webopedia.com>. It is still fair to make the point that “[a]n operating system does not mean just a kernel, barely enough to run other programs. In the 1970s, every operating system worthy of the name included command processors, assemblers, compilers, interpreters, debuggers, text editors, mailers, and much more.” Richard Stallman, *supra* note 1, at 56.

expected in the creation of a kernel, but in 1991 a computer science student, Linus Torvalds, released a more modest (i.e., non-portable, crashed often...) operating system, which did have an own created kernel. Torvalds made this operating system free and in 1992 the combination of Linux with the almost-complete GNU system resulted in a complete operating system:<sup>69</sup> the GNU/Linux system (or simply Linux).<sup>70</sup>

#### *4. Free software foundations and perspectives.*

##### **I. Introduction.**

The free software philosophy and structure were triggered by practical problems. As an example, Stallman explains his dissatisfaction with the fact that the lack of access to the source code of the MIT AI Lab printer prevented him from adding any convenient feature, as the one that notifies the users when the job has been actually printed.<sup>71</sup> We

---

<sup>69</sup> Richard Steuer, *supra* note 61; JESÚS GONZÁLEZ BARAHONA ET AL., *supra* note 38, at 45, 46.

<sup>70</sup> There is some controversy in using the term GNU/Linux or only Linux. Stallman states that “[i]f you call our operating system “Linux,” that conveys a mistaken idea of the system’s origin, history, and purpose. If you call it “GNU/Linux,” that conveys (though not in detail) an accurate idea.” RICHARD STALLMAN, *supra* note 1, at 51, available at <http://www.gnu.org/gnu/why-gnu-linux.html>. Moreover, “[o]ne CD-ROM vendor found that in their “Linux distribution”, [GNU software](#) was the largest single contingent, around 28% of the total source code, and this included some of the essential major components without which there could be no system. Linux itself was about 3%. So if you were going to pick a name for the system based on who wrote the programs in the system, the most appropriate single choice would be “GNU.” Richard Stallman, *Linux and the GNU Project*, available at <http://www.gnu.org/gnu/linux-and-gnu.html>. See as well, Jonathan Zittrain, *supra* note 30, at 268. However, Linus Torvalds suggests that the use of the term Linux is just fine, LINUS TORVALDS, *supra* note 10, at 163.

<sup>71</sup> “The MIT Artificial Intelligence Lab (AI Lab) received a graphics printer as a gift from *Xerox* around 1977. It was run by free software to which we added many convenient features. For example, the software would notify a user immediately on completion of a print job. Whenever the printer had trouble, such as a paper jam or running out of paper, the software would immediately notify all users who had print jobs queued. These features facilitated smooth operation.

Later *Xerox* gave the AI Lab a newer, faster printer, one of the first laser printers. It was driven by proprietary software that ran in a separate dedicated computer, so we couldn’t add any of our favorite features. We could arrange to send a notification when a print job was sent to the dedicated computer, but not when the job was actually printed (and the delay was usually considerable). There was no way to find out when the job was actually printed; you could only guess. And no one was informed when there was a paper jam, so the printer often went for an hour without being fixed.

must point out that Stallman is a computer scientist and that his concern may be shared exclusively by other computer scientists or high knowledgeable users –a tiny part of the computer users. However, if computer scientists have the right to access the source code and, consequently, to fix the bugs and to improve the programs, new ways of developing software and distributing it in the society will emerge, as it will be shown below.

Free software is a complex phenomenon with deep foundations and broad implications in different fields. In an extremely brief summary, we can say that the foundations of free software have a moral character. However, the pillars of the building are in legal documents that make the realization of free software principles possible. Finally, this legal approach leads to a certain forms of software development and economic strategies.<sup>72</sup>

## **II. Definition.**

A program is free software if users have four kinds of freedom:

“♦ Freedom 0: The freedom to run the program, for any purpose.

♦ Freedom 1: The freedom to study how the program works, and adapt it to your needs.

(Access to the source code is a precondition for this.)

♦ Freedom 2: The freedom to redistribute copies so you can help your neighbor.

---

The system programmers at the AI Lab were capable of fixing such problems, probably as capable as the original authors of the program. *Xerox* was uninterested in fixing them, and chose to prevent us, so we were forced to accept the problems. They were never fixed.” RICHARD STALLMAN, *supra* note 1, at 125, available at <http://www.gnu.org/philosophy/shouldbefree.html>.

<sup>72</sup> “[O]pen source can be seen as a revolutionary new way of doing and distributing software...” MIKKO VÄLIMÄKI, THE RISE OF OPEN SOURCE LICENSING – A CHALLENGE TO THE USE OF INTELLECTUAL PROPERTY IN THE SOFTWARE INDUSTRY 7 (2005), available at <http://www.gnu.org/philosophy/why-free.html>.

◆ Freedom 3: The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. (Access to the source code is a precondition for this.)”<sup>73</sup>

### **III. Moral foundations.**

Stallman’s point of departure is that “society needs to encourage the spirit of voluntary cooperation in its citizens. When software owners tell us that helping our neighbors in a natural way is "piracy", they pollute our society's civic spirit.”<sup>74</sup>

This moral philosophy supports two main free software principles. First, free software must allow access to its source code. Second, free software must allow making copies and redistribution of them. These two principles sit in opposition to proprietary software, that is, software whose “use, redistribution or modification [that implies access to the source code] is prohibited, or requires you to ask for permission, or is restricted so much that you effectively can't do it freely.”<sup>75</sup>

The first principle is based upon the idea that society “needs information that is truly available to its citizens---for example, programs that people can read, fix, adapt, and improve, not just operate. But what software owners typically deliver is a black box that we can't study or change.”<sup>76</sup> In other words, “[t]he ease of modification of software is one of its great advantages over older technology. But most commercially available software

---

<sup>73</sup> RICHARD STALLMAN, *supra* note 1, at 41.

<sup>74</sup> *Id.* at 48, available at <http://www.gnu.org/philosophy/why-free.html>.

<sup>75</sup> Richard Stallman, *Categories of Free and Non-Free Software* (2005), available at <http://www.gnu.org/philosophy/free-sw.html>.

<sup>76</sup> RICHARD STALLMAN, *supra* note 1, at 47 – 48.

isn't available for modification, even after you buy it. (...) Software development used to be an evolutionary process, where a person would take an existing program and rewrite parts of it for one new feature, and then another person would rewrite parts to add another feature; in some cases, this continued over a period of twenty years. Meanwhile, parts of the program would be 'cannibalized' to form the beginnings of other programs. The existence of owners prevents this kind of evolution, making it necessary to start from scratch when developing a program."<sup>77</sup> In relation to the lack of access to the source code, Stallman makes a very relevant point from the copyright point of view: "[i]n any intellectual field, one can reach greater heights by standing on the shoulders of others. But that is no longer generally allowed in the software field--you can only stand on the shoulders of the other people *in your own company*."<sup>78</sup>

The freedom of free making and distribution of copies derives from the intangible nature of intellectual works. In opposition to material objects, like cars, chairs, or sandwiches, programs are possible to reproduce at almost no cost. "It is easy to show that the total contribution of a program to society is reduced by assigning an owner to it. Each potential user of the program, faced with the need to pay to use it, may choose to pay, or may forego use of the program. When a user chooses to pay, this is a zero-sum transfer of wealth between two parties. But each time someone chooses to forego use of the program, this harms that person without benefitting anyone. The sum of negative numbers and zeros must be negative."<sup>79</sup>

---

<sup>77</sup> *Id.* at 124, 126.

<sup>78</sup> *Id.* at 126.

<sup>79</sup> RICHARD STALLMAN, *supra* note 1, at 122. In comparison, thinks Stallman, the cost of production of material objects is significant and therefore it is fair to add a share of the development cost, which does not

One common misunderstanding about the free software is that it is forbidden to make money with its distribution; this is not correct. This misunderstanding can be traced back to the GNU Manifesto, which Stallman started with the following sentence: “GNU (...) is the name for the complete Unix-compatible software system which I am writing so that I can *give it away free* to everyone who can use it. [italics added]” However, eight years later Stallman made this clarification: “[t]he wording here was careless. The intention was that nobody would have to pay for \*permission\* to use the GNU system. But the words don't make this clear, and people often interpret them as saying that copies of GNU should always be distributed at little or no charge. That was never the intent; later on, the manifesto mentions the possibility of companies providing the service of distribution for a profit. Subsequently I have learned to distinguish carefully between ‘free’ in the sense of freedom and ‘free’ in the sense of price. Free software is software that users have the freedom to distribute and change. Some users may obtain copies at no charge, while others pay to obtain copies--and if the funds help support improving the software, so much the better. The important thing is that everyone who has a copy has the freedom to cooperate with others in using it.”<sup>80</sup> The ambiguity of the term “free” has caused a search for alternatives, like “liberated,” “freedom,” “open,” and “non-proprietary”. However,

---

make a qualitative difference. More moderated free software supporters are as well against the strict enforcement of Copyright laws in case of illegal reproductions, LINUS TORVALDS, *supra* note 10, at 97.

<sup>80</sup> Richard Stallman, *Footnotes to the GNU Manifesto* (1993), available at <http://www.gnu.org/gnu/manifesto.html>. Another term that causes some confusion is “freeware”, that is, “Copyrighted software given away for free by the author. Although it is available for free, the author retains the copyright, which means that you cannot do anything with it that is not expressly allowed by the author. Usually, the author allows people to use the software, but not sell it.” Freeware is not free software, as its source code may not be delivered and it may not permit its redistribution.

Stallman insists that these other words have either the wrong meaning or some other disadvantage and maintains the original term.<sup>81</sup>

#### **IV. Legal approach.**

The legal perspective of free software cannot be underestimated. In the end, the whole free software philosophy is only possible through its legal construction.<sup>82</sup> As it has often been said, the characteristic freedoms of free software seem to clash with Copyright. It may be argued that Copyright foresees some exceptions in favour of the user of a legal copy to run the program (freedom 0) and to adapt it to the user's needs (freedom 1). However, we will not delve into this discussion, as it is clear that the redistribution of copies (freedom 2) and the release of modified versions of the program (freedom 3) are only possible under the author's authorization.

An apparently straightforward solution would be to disclaim the copyright on the program which we want to be free. In other words, we would take the necessary steps to place the program in the public domain. However, this would be unsatisfactory in order to achieve the goals of the free software movement. Firstly, a program in the public domain can be released as object code and that impedes the access to its source code. Secondly, if no copyrights are attached to the program, even being released as source code, everyone could redistribute it as proprietary –that is, turn its source code into object code and release it only as such. If the redistribution as proprietary software takes place

---

<sup>81</sup> Richard Stallman, *supra* note 1, at 57.

<sup>82</sup> Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 Berkeley Technology Law Journal 443, 447 (2005); Raymond Nimmer, *Legal Issues in Open Source and Free Software Distribution*, 861 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 7, 11 (2006).

after any copyrightable modification is made to the public domain program, the author of the modification will be protected under Copyright laws. It is easy to realize that these actions endanger free software movement goals.<sup>83</sup>

Therefore, the public domain strategy was discarded and instead a copyright license is issued, called the GNU General Public License (GNU GPL),<sup>84</sup> which is based on two main clauses: first, the license provides users with the freedom to use, modify and redistribute the software (free software clauses); second, the license forces any person redistributing the original or modified free software to do it under the same license, that is, under the free software clause (copyleft clause).<sup>85</sup> The license was written to guarantee that not only the original free software, but every single modification or derivative work will allow access to the source code and redistribution.<sup>86</sup>

The originality and complexity from the GNU GPL lies in its copyleft clause, which basically states “that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it.”<sup>87</sup> This is what is called “copyleft”.<sup>88</sup> The concept “copyleft” reflects in an amusing way that the GNU GPL is

---

<sup>83</sup> Christian H. Nandan, *supra* note 32, at 358, 359.

<sup>84</sup> <http://www.gnu.org/copyleft/gpl.html>.

<sup>85</sup> “We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.” Preamble of the GNU GPL.

<sup>86</sup> Jonathan Zittrain, *supra* note 30, at 268, 269.

<sup>87</sup> RICHARD STALLMAN, *supra* note 1, at 89, available at <http://www.gnu.org/licenses/licenses.html>.

<sup>88</sup> Stallman explains the origin of the term: “In 1984 or 1985, Don Hopkins (a very imaginative fellow) mailed me a letter. On the envelope he had written several amusing sayings, including this one: “Copyleft— all rights reversed.” I used the word “copyleft” to name the distribution concept I was developing at the time.” Richard Stallman, *supra* note 1, at 59.

achieving through the rights granted to the copyright holder the very opposite effect to the one intended by Copyright: freedom instead of control.<sup>89</sup>

The notions of free software and copyleft are not synonyms. As we have seen before, software is free if it complies with the four mentioned freedoms. However, that does not necessarily mean that it also guarantees the access to the source codes of modified versions or its redistribution. For example, if a program is placed in the public domain as object code it will be free software, but not copylefted software. Moreover, there are some licenses that allow conversion of free software into proprietary software, for example, the BSD (Berkeley Software Distribution).<sup>90</sup> The works released under this kind of license are usually called “open” rather than “free”, or if “free” are qualified as “but not copyleft.”<sup>91</sup> We will expound this difference below.

## **V. Software development.**

Naturally, the lack of control under copyleft licenses influences the way in which software is produced. The traditional model of developing software, that is, a company employing computer scientists to sell the software that they produce, needs a strict legal control over their products and it is therefore incompatible with the GNU GPL or any

---

<sup>89</sup> The literature has described this maneuver in the most diverse forms: “flips [copyright law] over”, Richard Stallman, *supra* note 1, at 59; “turn[s] normal copyright on its head”, John C. Yates & Paul H. Arne, *Open Source Software Licenses: Perspectives of the End User and the Software Developer*, 823 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 97, 104 (2005); or, even, as “a for of legal jujitsu”, Jonathan Zittrain, *supra* note 30, at 269.

<sup>90</sup> See <http://www.opensource.org/licenses/bsd-license.php>. The BSD license allows both the distribution as source code or as object code, and the single demand is to give credit to the original authors. JESÚS GONZÁLEZ BARAHONA ET AL., *supra* note 38, at 76 (2003).

<sup>91</sup> Jonathan Zittrain, *supra* note 30, at 269.

other copyleft license. If any third party could freely redistribute the software produced expensively by a private company, the latter would simply run its business at a loss.

The software development under the free software movement premises is based on the work of computer scientists or hackers,<sup>92</sup> who voluntarily and without expecting an economic benefit in return help to build a computer program. Licensing copyrights is forbidden under the GNU GPL. Therefore, the authors of the program have no direct economic advantage against any persons. The incentives to cooperate in a free software project are satisfaction of programming,<sup>93</sup> prestige<sup>94</sup> and economic gains obtained from sources different to the copyright licenses, for example, funding.<sup>95</sup> Other profit strategies are also possible under the free software principles,<sup>96</sup> as we will see in the following section.

---

<sup>92</sup> "Hacker is a term used by some to mean "a clever programmer" and by others, especially journalists or their editors, to mean "someone who tries to break into computer systems."

1) Eric Raymond, compiler of *The New Hacker's Dictionary*, defines a hacker as a clever programmer. A "good hack" is a clever solution to a programming problem and "hacking" is the act of doing it. Raymond lists five possible characteristics that qualify one as a hacker, which we paraphrase here:

- ◆ A person who enjoys learning details of a programming language or system
- ◆ A person who enjoys actually doing the programming rather than just theorizing about it
- ◆ A person capable of appreciating someone else's hacking
- ◆ A person who picks up programming quickly
- ◆ A person who is an expert at a particular programming language or system, as in "Unix hacker"

Raymond deprecates the use of this term for someone who attempts to crack someone else's system or otherwise uses programming or expert knowledge to act maliciously. He prefers the term cracker for this meaning.

2) Journalists or their editors almost universally use hacker to mean someone who attempts to break into computer systems. Typically, this kind of hacker would be a proficient programmer or engineer with sufficient technical knowledge to understand the weak points in a security system." <http://www.whatis.com>.

<sup>93</sup> ERIC S. RAYMOND, *THE CATHEDRAL & THE BAZAAR – MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY* 60, 61 (revised 1<sup>st</sup> ed. 2001); RICHARD STALLMAN, *supra* note 1, at 127, 128, available at <http://www.gnu.org/philosophy/shouldbefree.html>.

<sup>94</sup> ERIC S. RAYMOND, *supra* note 1, at 84.

<sup>95</sup> RICHARD STALLMAN, *supra* note 1, at 128, 129, available at <http://www.gnu.org/philosophy/shouldbefree.html>.

<sup>96</sup> See Frank Hecker, *Setting Up Shop: The Business of Open-Source Software* (2000), available at <http://www.hecker.org/writings/setting-up-shop>; Sandeep Krishnamurthy, *Cave or Community? An Empirical Examination of 100 Mature Open Source Projects*, First Monday (2002), available at [http://www.firstmonday.org/issues/issue7\\_6/krishnamurthy/](http://www.firstmonday.org/issues/issue7_6/krishnamurthy/).

In a very famous publication, Raymond compares the proprietary and free models of software development with a cathedral and a bazaar. The cathedral represents the model where a single person or a small group crafts the program and works in isolation until the release of the final product. In opposition, in the bazaar anyone can contribute to the program with different approaches, debugging solutions and improvements.<sup>97</sup> Under Raymond's view, an open source project begins with the creation of a program. This program should be released to the community as soon as it constitutes a "plausible promise", even if it does not work particularly well or is still incomplete.<sup>98</sup> Once the program is being developed by the community, it is important to release often the new versions in order to get more corrections.<sup>99</sup> The free software movement strongly believes that the most effective way to produce software is working in a community, because "[g]iven a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone. Or, less formally, "Given enough eyeballs, all bugs are shallow."<sup>100</sup>

Raymond's metaphor has found some criticisms in the literature. Firstly, it has been argued that most open source software "are developed by individuals, rather than communities."<sup>101</sup> That may be true, but it would not change Raymond's point in relation to the big programs, like the development of an operating system. Secondly, many raise

---

<sup>97</sup> ERIC S. RAYMOND, *supra* note 93, at 21, 22, based on the earlier paper of the same author, *The Cathedral and the Bazaar*, February 10<sup>th</sup>, 1998 available at [http://www.firstmonday.org/issues/issue3\\_3/raymond/](http://www.firstmonday.org/issues/issue3_3/raymond/).

<sup>98</sup> *Id.* at 47.

<sup>99</sup> *Id.* at 28 – 31.

<sup>100</sup> *Id.* at 30. Arguing the same, Richard Stallman, *supra* note 1, at 55. LINUS TORVALDS, *supra* note 10, at 226, 227.

<sup>101</sup> Sandeep Krishnamurthy, *supra* note 96..

the point that Raymond underestimates the role of the productivity-multiplying effect of conventional management. But Raymond's contention is that the community itself or the project leader of an open source project are more effective in all the tasks attributed to the conventional management team.<sup>102</sup> Thirdly, it has been observed that Linux, which is the open source project used by Raymond to fundament his theory, "looks more like a highly centralized (Cathedral) development model."<sup>103</sup> In fact, the most important open source projects have a leader or coordinator –some compare their role to a dictator. For example, in Linux is Linus Torvals, with the support of a small group of lieutenants, who takes the final decisions about the patches that will be implemented in the next versions of the program. In *Apache* the committee directors are periodically elected in a democratic manner by the members of the Apache Foundation.<sup>104</sup> This criticism may affect the open source main version of a certain project; however it is still true that nobody will prevent an independent programmer from using the source code to improve it and to make available his own version.

To sum up, open source projects are developed under a variety of models and in certain cases their characteristics are not so far away from proprietary models. However, when an open source program is released, at the very last as a final and complete product (also known as version 1.0), and taking for granted that it has some appeal for other computer scientists, the global review process by the community will occur. This will never happen under the proprietary approach.

---

<sup>102</sup> See Raymond's response in ERIC S. RAYMOND, *supra* note 38, at 55 – 60.

<sup>103</sup> Nikolai Bezroukov, *A Second Look at the Cathedral and the Bazaar*, First Monday (1999), available at [http://www.firstmonday.org/issues/issue4\\_12/bezroukov/](http://www.firstmonday.org/issues/issue4_12/bezroukov/).

<sup>104</sup> JESÚS GONZÁLEZ BARAHONA ET AL., *supra* note 38, at 203, 204.

## VI. Economic models.

For many years, the overwhelming opinion among jurists has been that granting copyright protection over the works is the only way to guarantee production and quality. Along this line of thinking, the National Commission on New Technological Uses of Copyrighted Works considered that “some form of protection is necessary to encourage the creation and broad distribution of computer programs in a competitive market.” And at the same time it dismissed the possibility that “the creator is indifferent to cost and donates the work to the public”, with the only exception of the academic and government sponsored research.<sup>105</sup> This idea has been promoted by the proprietary industry. Bill Gates asked the following rhetorical question in a letter dated in 1976 to open source programmers: “One thing you do is prevent good software from being written. Who can afford to do professional work for nothing?”<sup>106</sup>

However, as we have already seen, the free software movement has proved that it is possible to renounce all economic advantages from copyright and still produce software of a reasonable quality.<sup>107</sup> The question now is if after renouncing the royalties coming from the copyright licenses, there is any economic model supporting the production, distribution and maintenance from free software.

---

<sup>105</sup> Final Report of the National Commission on New Technological Uses of Copyrighted Works 11 (1978), available at <http://digital-law-online.info/CONTU/PDF/>.

<sup>106</sup> Widely mentioned in Internet and also in, LINUS TORVALDS, *supra* note 10, at 227.

<sup>107</sup> Richard Stallman, *supra* note 1, at 54.

As we have seen, before the release of an open source computer program, some funding can come from donations. However, once the software is in the computers of the final users, the revenues will be made by the support sellers or redistributors. This model includes “media distribution, branding, training, consulting, custom development, and post-sales support.”<sup>108</sup> One famous example of companies based on open source software redistribution is Red Hat, which not only distributes the software but also provides maintenance services to his customers.<sup>109</sup> The support seller economic model can even indirectly explain the interest of developers in working on open source projects for free, as they will get experience that will allow them or their companies to guarantee a high quality support to final users of the open source program that they developed.<sup>110</sup>

There are other economic models: 1. loss leader - the open source product is used to promote the sales of related proprietary software; 2. widget frosting - the open source software supports the hardware, which is the actual source of revenues; 3. accesoring - distribution of books and other physical items associated with a certain open source software; 4. service enabler - company creates and distributes open source software to support access to revenue generating on line services; 5. sell it, free it - a company

---

<sup>108</sup> Frank Hecker, *supra* note 96..

<sup>109</sup> “Red Hat is a leading software company in the business of assembling [open source](#) components for the [Linux](#) operating system and related programs into a distribution package that can easily be ordered. Red Hat provides over 400 different software packages (...). The advantages to buying the distribution from Red Hat rather than assembling it at no cost yourself from various sources is that you get it as a single assembled package. Red Hat also offers service that isn't provided as quickly by the individual component developers, including members of the Free Software Foundation. Like all free software, Red Hat's packages allow the buyer to modify and even resell modified versions of code as long as they do not restrict anyone else from further modification.

Red Hat was one of the first companies to realize that "free" software could be sold as a product. After examining the successful marketing campaign of Evian water, Red Hat concluded that to achieve success, the company had to create more Linux users and brand Red Hat as the Linux name that customers preferred. Today, the "Red Hat Plan" is discussed as a model in business schools.” <http://www.whatis.com>.

<sup>110</sup> JESÚS GONZÁLEZ BARAHONA ET AL., *supra* note 38, at 118, 119.

releases software products first as proprietary and then convert them to open source products when the benefits of developing them in an open source environment outweigh the direct software license revenue they produce –typically when a new proprietary version is released; 6. brand licensing - even if a company releases his software products as open source, it still retains the rights to its product trademarks (e.g. *Linux* and *Mozilla*) and logos; 7. software franchising - the company would authorize other developers to use its brand names and trademarks in creating associated organizations doing open source support and custom software development.<sup>111</sup>

## *5. Free software v. Open Source Software.*

### **I. Introduction.**

The terms “free software” and “open source” are often taken for one and the same thing. Although it may be true that the differences are not big, they are at least worth an explanation. The “open source” concept was born in February 1998, as a reaction to Netscape’s announcement to give away the source of its browser. A small group of computer scientists (Open Source Initiative) coined the term “open source” with the intention of making open development processes more appealing for the corporate world. This concept was supported by important actors in the free movement, like Linus Torvalds, but not for Richard Stallman or his Free Software Foundation, and that caused split.<sup>112</sup> Paradoxically, the expression “open source” has become more popular –92 matches in the titles of law review articles of the Westlaw database against only 10 for

---

<sup>111</sup> Frank Hecker, *supra* note 96.

<sup>112</sup> Open Source Initiative, *History of the OSI* (last update 2006), <http://www.opensource.org/docs/history.php>.

“free software”–,<sup>113</sup> but the free software licensees, in particular, the GNU GPL, are more used by far.<sup>114</sup>

The differences between “free software” and “open source” can be observed in four areas: the terminology, the definition, the philosophy and their respective characteristic licenses.

## II. Terminology.

The members of the Open Source Initiative claim that “open source” is a much clearer term than the ambiguous “free software”. The latter, they maintain, has many different meanings for different groups of people, from distribution for free to distribution under a copyleft license.<sup>115</sup> The counterargument from the Free Software Foundation is that “the

---

<sup>113</sup> “In mid-2004, the President of OSI did a statistical Web-content analysis on the usage frequencies of the phrases “open source” and “free software. (...) A summary of the conclusions:

- ◆ Among software developers and in the technology trade press, use of the term “open source” dominates use of the term “free software” by 95%-5% or more.
- ◆ On the general Web, the ratio is 80%-20% or more.
- ◆ The gratis/libre ambiguity in the term “free software” produces about an 80% false-positive rate in Web searches.

◆ Use of the term “free software” is in long-term decline, and older or obsolete pages form a larger part of its share than for “open source.” Open Source Initiative, *Why “Free” Software is too ambiguous*, available at <http://www.opensource.org/advocacy/free-notfree.php>.

<sup>114</sup> Two of the most popular Open Source Software websites, Freshmeat.net and Sourceforge.net, confirm this fact. The former provides the following data: out of 43,000 projects, up to 66.79% use the GNU GPL, the LGPL is used in up to 6.29% and BSD licenses (original + revised) are used in up to 5.58%. See <http://freshmeat.net/stats> (last visited April 29<sup>th</sup>, 2006). And Sourceforge.net has 118,615 registered. From them 51,533 are using the GNU GPL (43.44%); 9,055 the LGPL (7.63%); and 5,716 (4.82%). See <http://sourceforge.net/softwaremap/> (last visited April 29<sup>th</sup>, 2006).

<sup>115</sup> “Some software is called “free” because it costs no money to download or use – but source code is not available. The license that covers Microsoft Internet Explorer is a good example.

Some software is called “free” because it (and the source code for it) has been placed in the “public domain”, free from copyright restrictions.

A lot of software is called “free” even though the source code for it is covered by copyright and a license agreement. The license usually includes a disclaimer of reliability, and may contain additional restrictions.

The restrictions on non-public-domain “free” software range from mild to severe. Some licenses may prohibit (or require a fee for) commercial use or redistribution. Some licenses may prohibit distributing modified versions. Some licenses may contain “copyleft” restrictions requiring that the source code must

obvious meaning for the expression "open source software" is "You can look at the source code." This is a much weaker criterion than free software; it includes free software, but also includes semi-free programs<sup>116</sup> such as Xv, and even some proprietary programs.”<sup>117</sup> To summarize, the expression “open source software” underlines the fact that one has access to the source code, but not the possibility to modify and/or redistribute the program, whereas “free software” emphasizes the freedoms to copy, redistribute and modify the program. However, “free” could mean and it is often misunderstood as “at no charge” and, moreover, the reference to the access to the source code is only indirect.

In coherence with their respective main concepts, the Free Software Foundation uses the term non-free software, which includes semi-free and proprietary software. In opposition, the Open Source Initiative refers with the term “closed” to software the access to the source code of which is restricted, and this term is expressly excluded from the Free Software Foundation terminology.<sup>118</sup> Both organizations consider the software proprietary when either its redistribution or the access to the source code is disallowed. Generally, the terms “non-free software” or “closed software” imply proprietariness. However, there are some exceptions. For example, under the terminology of the Free Software Foundation, if a piece of software allows its use, copy, distribution and

---

always be made available, and that derived products must be released under the exact same license. Some licenses may discriminate against individuals or groups.” Open Source Initiative, *supra* note 113..

<sup>116</sup> “Semi-free software is software that is not free, but comes with permission for individuals to use, copy, distribute, and modify (including distribution of modified versions) for non-profit purposes. PGP is an example of a semi-free program.” <http://www.gnu.org/philosophy/categories.html>.

<sup>117</sup> RICHARD STALLMAN, *supra* note 1, at 56, available at <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

<sup>118</sup> “To prevent people from thinking we are part of them, we take pains to avoid using the word “open” to describe free software, or its contrary, “closed”, in talking about non-free software.” Richard Stallman, *Why “Free Software” is better than “Open Source”*, (last update 2005), available at <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

modification exclusively for non-profit purposes, it would be semi-free software, that is, non-free software, but not proprietary; but under the definition of the Open Source Initiative, if a piece of software allows access to the source code and eventually its modification for private purposes, but not its redistribution, it is not closed source software, but still proprietary.

In order to encompass both free software and open source software, the combination of both expressions has been proposed: “free and open source software” (FOSS). Another possibility is to refer to them as the opposite of proprietary software, that is, as “non-proprietary”.<sup>119</sup> Finally, a working group within the European Union adopted the term “Libre Software”, where “libre” means in Spanish and French “free” as opposed to “freedom” and hence does not cause any ambiguity.<sup>120</sup> Alternatively the combination of the two last phrases has been used: Free/Libre Open Source Software.<sup>121</sup>

### **III. Definitions.**

In order to eliminate or at least reduce these ambiguities, both groups have published definitions of the terms that they defend.<sup>122</sup> We have already seen that the free software movement characterizes “free software” as the one which grants the user the freedoms to

---

<sup>119</sup> Andrés Guadamuz González, *Viral Contracts or Unenforceable Documents? Contractual Validity of Copyleft Licences*, European Intellectual Property Review 331, 332, 333 (2004).

<sup>120</sup> Working Group on Libre Software, *Free Software/Open Source: Information Society Opportunities for Europe?* 5 (2000), available at <http://eu.conecta.it/paper.pdf>.

<sup>121</sup> *FLOSS Final Report*, *supra* note 11, at 41.

<sup>122</sup> The Open Source Initiative has even created a certification for the licenses complying with the conditions of the Open Source Definition: “We think the Open Source Definition captures what the great majority of the software community originally meant, and still mean, by the term “Open Source”. However, the term has become widely used and its meaning has lost some precision. The OSI Certified mark is OSI’s way of certifying that the license under which the software is distributed conforms to the OSD; the generic term “Open Source” cannot provide that assurance, but we still encourage use of the term “Open Source” to mean conformance to the OSD.” [http://www.opensource.org/docs/certification\\_mark.php](http://www.opensource.org/docs/certification_mark.php).

run, copy, distribute, study, change, improve the software and, therefore, access to the source code. The Open Source Initiative guidelines are exposed in the Open Source Definition,<sup>123</sup> which contains ten clauses: (1) free redistribution; (2) the program must include source code, and must allow distribution in source code as well as compiled form; (3) the license must allow modifications and derived works; (4) the license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of “patch files”<sup>124</sup> with the source code for the purpose of modifying the program at build time –in this way, “unofficial” changes can be made available but readily distinguished from the base source; (5) the license must not discriminate against any person or group of persons; (6) the license must not restrict anyone from making use of the program in a specific field of endeavor; (7) the rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties –this clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement; (8) the rights attached to the program must not depend on the program's being part of a particular software distribution; (9) the license must not place restrictions on other software that is distributed along with the licensed software; and (10) no provision of the license may be predicated on any individual technology or style of interface.

---

<sup>123</sup> <http://www.opensource.org/docs/definition.php>.

<sup>124</sup> “A patch (sometimes called a “fix”) is a quick-repair job for a piece of programming. During a software product's beta test distribution or try-out period and later after the product is formally released, problems (called bug) will almost invariably be found. A patch is the immediate solution that is provided to users... A patch is usually developed and distributed as a replacement for or an insertion in compiled code (that is, in a *binary file* or object module).” <http://www.whatis.com>.

The four freedoms of the free software movement are almost equivalent with the ten clauses of the Open Source Definition.<sup>125</sup> Both definitions guarantee the possibility to run,<sup>126</sup> copy, distribute,<sup>127</sup> study, change, improve the software and, therefore, access to the source code.<sup>128</sup> In some minor aspects, the Open Source Definition is less advantageous for the recipients of the program or more friendly to commercial interests.<sup>129</sup> For example, the Open Source Definition allows the authors to restrict the distribution of source code in modified form in very specific circumstances;<sup>130</sup> the Open Source Definition does not allow restrictions on other software that is distributed along with the licensed software, while the free software definition is silent about this point. To sum up, the relevant differences between the position of the Free Software Foundation and that of the Open Source Initiative are not contained in the definitions that they use, but in the copyleft clause, which we will analyse later.

As a matter of fact, many of the licenses complying with the Open Source Definition can be qualified as free software as well, and the reverse is also true. Most of the 56 licenses approved by the Open Source Initiative<sup>131</sup> are recognized as free software by the free software movement<sup>132</sup> (i.e., Academic Free License, Apache License 2.0, Apache Software License, New BSD license, Eclipse Public License, X11, also called MIT license, PHP license, Python license, zlib/libpng license and so on) –though not

---

<sup>125</sup> The literature has pointed out that “[t]he four freedoms are consistent with (albeit broader than) the criteria employed by the Open Source Initiative.” Christian H. Nandan, *supra* note 32, at 357.

<sup>126</sup> See freedom 0 and implicit in the whole Open Source Definition (i.e., clause 6, but also, 1, 2...).

<sup>127</sup> See freedom 2 and clause 1 of the Open Source Definition.

<sup>128</sup> See freedoms 1 and 3 and clause 2 of the Open Source Definition.

<sup>129</sup> Raymond Nimmer, *supra* note 82, at 16, 17.

<sup>130</sup> See clause 4 of the Open Source Definition.

<sup>131</sup> <http://www.opensource.org/licenses/>.

<sup>132</sup> <http://www.gnu.org/licenses/license-list.html>.

necessarily compatible with the GNU GPL, because many of them fail to include a copyleft clause. The licenses that have been approved by the Open Source Initiative as Open Source licenses but do not satisfy the requisites of the free software definition are rare, though not completely non-existent (i.e., Reciprocal Public License).<sup>133</sup>

#### **IV. Philosophy.**

The free software movement and the Open Source Initiative describe their own philosophic background in a fair distinguishable manner. On the one hand, the free software movement justifies the freedoms to distribute and to access to the source code as a matter of social fairness. The proprietary control over these acts, they say, causes different levels of harm, in particular, fewer people using the program, none of the users being able to adapt or fix the program; and other developers not being able to learn from the program, or base new work on it.<sup>134</sup> For these reasons, the free software movement embarks on a crusade against the proprietary software, which they qualify as “the enemy”.<sup>135</sup> Because of the above, the free software movement has been depicted by the literature as idealistic<sup>136</sup> or, even, radical.<sup>137</sup>

---

<sup>133</sup> “The Reciprocal Public License is a non-free license because of three problems. 1. It puts limits on prices charged for an initial copy. 2. It requires notification of the original developer for publication of a modified version. 3. It requires publication of any modified version that an organization uses, even privately.” David Turner, *Various Licenses and Comments about Them* (last updated 2006), <http://www.gnu.org/licenses/license-list.html>.

<sup>134</sup> RICHARD STALLMAN, *supra* note 1, at 122 – 126, available at <http://www.gnu.org/philosophy/shouldbefree.html>.

<sup>135</sup> RICHARD STALLMAN, *supra* note 1, at 55, available at <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

<sup>136</sup> César Iglesias Rebollo, *Otra forma de distribuir software: las licencias de software libre o de código abierto*, 2 Revista General de Legislación y Jurisprudencia 219, 223 (2004).

<sup>137</sup> Andrés Guadamuz González, *supra* note 119, at 332.

On the other hand, the Open Source Initiative agrees in the superiority of an open development process, but they consider that this is compatible with the commercial software companies, that is, with the proprietary software. They consciously distinguish themselves from the “confrontational attitude that has been associated with "free software" in the past and sell the idea strictly on (...) pragmatic, business-case grounds”.<sup>138</sup> Consequently, the Open Source Initiative tolerates the conversion of open source software into closed or proprietary software and dislikes the use of the word “free”, which can be easily misunderstood as “free from charge”. The members of the Open Source Initiative have been described as pragmatic.<sup>139</sup>

#### **V. Representative licenses: GNU GPL v. BSD license.**

We have already seen that the opposite philosophic foundations of the free software movement and the Open Source Initiative did not lead to very diverse definitions of free software and open source software. However, their different foundations are starkly reflected in the most representative licenses for both movements: the GNU General Public License (GNU GPL) and the Berkeley Software Distribution license (BSD),<sup>140</sup> respectively. In particular, the GNU GPL includes a copyleft clause whereas the BSD license does not. The copyleft clause imposes the obligation over modified or, in general, redistributed software to be licensed under the same license that allows its modification and redistribution. In other words, whoever wants to redistribute the software must pass

---

<sup>138</sup> Open Source Initiative, *History of the OSI* (last update 2006), <http://www.opensource.org/docs/history.php>.

<sup>139</sup> César Iglesias Rebollo, *supra* note 136, at 223.

<sup>140</sup> The BSD license has an original and a modified version (<http://www.xfree86.org/3.3.6/COPYRIGHT2.html>, under the sections UCB/LBL and General, respectively), with the difference that the modified version has removed the obligation to acknowledge the development done by the University of California, Berkeley and its contributors.

along the freedom to further copy and change it. This might be seen as a restrictive condition for the commercial software companies or as the enlargement of the software users' rights. In opposition, the BSD license does not have any copyleft clause, which means that it is possible to transform open source software into proprietary software.

Also, the Free Software Foundation describes the BSD license as free software, but not copyleft.<sup>141</sup> On the other hand, the Open Source Initiative considers the GNU GPL as open source software.

## **VI. Conclusions.**

The most relevant differences that we have examined show a classification of three kinds of licenses or software strategies.

- ◆ First, proprietary license, which generally means that the access to the source code is restricted and its modification and redistribution is not allowed without authorization.
- ◆ Second, free software license or copyleft software, the access to the source code of which and the modification and redistribution of which are allowed under the condition that the modified and/or redistributed program will recognize these same rights.
- ◆ Third, open source license or non-copyleft open source software, which unconditionally allows the access to the source code, its modification and redistribution –and therefore to convert open source software into proprietary software.

The first two categories are completely irreconcilable, even antagonistic. The Free Software movement and the Open Source movement have different philosophies and

---

<sup>141</sup> David Turner, *supra* note 133..

goals, but still enough common characteristics to allow collaboration and sympathy between both movements.<sup>142</sup>

---

<sup>142</sup> RICHARD STALLMAN, *supra* note 1, at 55, available at <http://www.gnu.org/philosophy/free-software-for-freedom.html>.

### III.– Contractual Issues.

#### *1. Introduction.*

The non-proprietary licenses, both copyleft and non-copyleft open source software licenses, introduced a completely revolutionary way to deal with copyrights.<sup>143</sup> However, the most problematic issue in the free software and open source software arena is the copyleft clause.<sup>144</sup>

The non-copyleft open source software licenses do not generally impose any obligations on the licensors, besides those related to recognition of authorship. Therefore, the likelihood of a legal dispute for this kind of license is rather small. Meanwhile, the copyleft licenses impose on the downstream licensees the obligation to release their programs under the same copyleft clause. Due to this expansion of the effects of the copyleft clause over all the subsequent modifications and redistributions of the work the

---

<sup>143</sup> Richard Stallman and the copyleft advocates had been criticized for using copyright to achieve a system which is apparently contrary to copyright itself, Andrés Guadamuz González, *supra* note 119, at 334. The answer that Stallman gives to this critic is that the use of the copyleft clause “doesn't mean that we are in favour of copyright law as a general matter. We're not totally against copyright law, in a simple or blanket sense either, but we're not defending the global copyright system that has mostly been imposed on the world merely because we use it because it's there. And that has to be very clear. We are not endorsing the Berne plus WTO system of copyright law as it stands as a good thing, but it exists and whatever harm it may do in other areas, we're trying to do some good with it when we can.” Transcript of Opening session of first international GPLv3 conference, Boston, January 16<sup>th</sup>, 2006, available at <http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html>. Even under a more conservative view, a fair point that may be made is that copyright is not as much about control of the exclusive rights as about promoting the progress of science and useful arts. And from this latter perspective copyright and copyleft may be not that distant.

<sup>144</sup> Raymond Nimmer, *supra* note 82, at 53; Andrés Guadamuz González, *supra* note 119, at 334.

copyleft licenses have been described as “viral contracts”,<sup>145</sup> that is, as “an attempt to make commitments run with a digital object”.<sup>146</sup>

It may be useful for our discussion to quote the GNU GPL copyleft clause: “Section 2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions: (...) b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.”<sup>147</sup>

The first debate that the copyleft clause has raised is about the legal nature of the document that introduces it. A significant number of scholars have questioned that the GNU GPL should be treated as a license and suggest that we are rather dealing with a contract.

---

<sup>145</sup> Some authors prefer to use the term “reciprocal”, e.g. Brian W. Carver, *supra* note 82, at 447; Raymond Nimmer, *supra* note 82, at 15.

<sup>146</sup> Margaret Jane Radin, *Humans, Computers, and Binding Commitment*, Indiana Law Journal 1125, 1132 (2000). The term “viral” must be used carefully, as the literature has attributed to it a second meaning. In particular, referring to a copyleft license as “viral” may intend to evoke the belief that using free software code in a proprietary program can lead to the obligation to release the entire work as open source software, Stephen Mutkoski, *Open Source Software Issues in Acquisitions and Other Inbound Transactions*, 861 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 337, 342, 347 (2006).

<sup>147</sup> In the GPLv3 Draft the copyleft provision is moved to Subsection 5b and three clarifications are introduced: it states that the GPL applies to the whole work; it removes the words “at no charge,” which was often misinterpreted by commentators; and the last sentence of subsection 5b explicitly recognizes the validity of disjunctive dual-licensing. Free Software Foundation, *GPLv3 First Discussion Draft Rationale* 12 (2006), available at <http://gplv3.fsf.org/rationale.pdf>.

## 2. *Contract v. License.*

### **I. Relevance of the discussion.**

The distinction between contract and license is not merely academic. Most of the scholars entering this debate are considering at the same time the legal consequences that the qualification under one of the two categories may have for the GNU GPL.

First, the Copyright is a highly harmonized body of law. The Berne Convention for the Protection of Literary and Artistic Works (1886), the Rome Convention for the Protection of Performers, Producers of Phonograms and Broadcasting Organizations (1961), the WIPO Copyright Treaty (1996), the WIPO Performances and Phonograms Treaty (1996)<sup>148</sup> and the WTO Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPs) (1994)<sup>149</sup> have pursued the harmonization goal at an international level quite successfully. Other supranational agreements, such as the eight European Directives passed in the Copyright field,<sup>150</sup> are also remarkable achievements in this same enterprise. On the other hand, the law of contracts varies in a notable manner between different countries and even within the same country, as happens across the U.S.<sup>151</sup>

Second, if one considers the GNU GPL as a contract, a considerable array of problems will arise. For example, is the delivery of the software under the GNU GPL a contract

---

<sup>148</sup> The number of contracting parties of these four WIPO administered Treaties is: 160, 83, 58 and 57, respectively. See <http://www.wipo.int/treaties/en/>.

<sup>149</sup> The WTO counted 149 member States on December 1995. See list at [http://www.wto.org/english/thewto\\_e/whatis\\_e/tif\\_e/org6\\_e.htm](http://www.wto.org/english/thewto_e/whatis_e/tif_e/org6_e.htm).

<sup>150</sup> In particular, related to Protection of Computer Programs (1991), Rental and Lending Right (1992), Satellite and Cable (1993), Term of Protection (1993), Protection of Databases (1993), Copyright in the Information Society (2001), Release Right (2001) and Enforcement (2004). See all at [http://europa.eu.int/comm/internal\\_market/copyright/index\\_en.htm](http://europa.eu.int/comm/internal_market/copyright/index_en.htm).

<sup>151</sup> Jason B. Wacha, *supra* note 10, at 455, 456.

without consideration and, therefore, void? How and when do the offer and acceptance take place? Are warranty waivers enforceable when the recipient is a consumer?<sup>152</sup> This question seems irrelevant if we would deal with the GNU GPL as a Copyright license.

Third, the remedies applicable will depend on whether it is a contract or a license. If it is a contract, it seems that a person who refuses to comply with the terms of the GNU GPL could be forced to release the source code of his derivative work. Meanwhile, if it is a license, the only remedies available are to hinder or stop the use of copyleft licensed code under a non-copyleft license and, eventually, a claim for damages. However, the release of the code of a derivative work which is using copyleft code under a proprietary license cannot be legally pursued.<sup>153</sup>

Fourth, the enforceability of a license can be only claimed by the author or the copyright holder.<sup>154</sup> On the other hand, contracts can be enforced by the contracting parties, but not by third persons. That may interfere with the possibility of the original author to sue downstream parties, who are not in privity with him.

## **II. Conceptual approach.**

An appealing and simple solution to determine the right category for the GNU GPL is to look up the definitions of contract and license.

---

<sup>152</sup> Raymond Nimmer, *supra* note 82, at 25.

<sup>153</sup> Eben Moglen, *Enforcing the GNU GPL* (2001), available at <http://www.gnu.org/philosophy/enforcing-gpl.html>; Pamela Jones, *The GPL Is a License, not a Contract* (2003), available at <http://lwn.net/Articles/61292>.

<sup>154</sup> LAWRENCE ROSEN, *OPEN SOURCE LICENSING – SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW* 55 (2005).

Eben Moglen, who is chairman of the Software Freedom Law Center and General Counsel for the Free Software Foundation, opines that “[t]he word 'license' has, and has had for hundreds of years, a specific technical meaning in the law of property. A license is a unilateral permission to use someone else's property. The traditional example given in the first-year law school Property course is an invitation to come to dinner at my house. If, when you cross my threshold, I sue you for trespass, you plead my 'license,' that is, my unilateral permission to enter on and use my property.”<sup>155</sup> In short, a license is a permission to do something.<sup>156</sup>

The Free Software Foundation advocates that the GNU GPL is a license. In the GPLv3 Draft, this is even expressly stated in the title of Section 9: Not a Contract. The resulting advantages for the Free Software Foundation position to consider the GNU GPL as a license are fairly clear. First, it would avoid the extreme uncertainty caused by the diverse contract regulations. Second, it would avoid the multiple inconveniences caused by the law of contracts (e.g. some of the parties using the GNU GPL are not familiar with the requirements to make a valid offer). Third, it would avoid the privity requirement. And the only point where a qualification of a contract could be more interesting, which is specific performance of the terms of the agreement, is not comprised within the Free Software Foundations goals.<sup>157</sup>

---

<sup>155</sup> Quoted by Pamela Jones, *supra* note 153. A similar definition states that “[a] license is a unilateral abrogation of rights. The licensor has, by law, the ability to enforce certain rights against the licensee, and the license functions as a promise not to enforce those rights.” Jason B. Wacha, *supra* note 10, at 456.

<sup>156</sup> LAWRENCE ROSEN, *supra* note 154, at 51.

<sup>157</sup> The Free Software Foundation gives to the infringer the following choice: “you can stop using the stolen code and write your own, or you can decide you'd rather release under the GPL.” Pamela Jones, *supra* note 153. “In approximately a decade of enforcing the GPL, I have never insisted on payment of damages to the

On the other hand “[a] contract is a promise or a set of promises for the breach of which the law gives a remedy, or the performance of which the law in some way recognizes as a duty.” (Restatement, Second, Contracts § 1.) From this definition two important characteristics have been derived. In the first place, the presence of a promise, which is “a manifestation of intention to act or refrain from acting in a specified way, so made as to justify a promise in understanding that a commitment has been made.” (Restatement, Second, Contracts § 2.) Secondly, the necessity of an exchange, as courts do not enforce contracts unless the promisee has given the promisor something in return –the doctrine of consideration.<sup>158</sup>

The dichotomy between permission and exchange gets a bit more complicated when we look at the Uniform Computer Information Transactions Act (UCITA) § 102(a)(40), which defines a "license" as “a contract that authorizes access to, or use, distribution, performance, modification, or reproduction of, information or informational rights, but expressly limits the access or uses authorized or expressly grants fewer than all rights in the information, whether or not the transferee has title to a licensed copy.”

This definition seems to be formulated for the proprietary software license strategy. The proprietary software companies have avoided selling their products and instead they usually transfer the software under a license agreement. The main advantage of this

---

Foundation for violation of the license, and I have rarely required public admission of wrongdoing.” Eben Moglen, *supra* note 153.

<sup>158</sup> E. ALLAN FARNSWORTH, CONTRACTS 3 (4<sup>th</sup> ed., Aspen Publishers 2004).

strategy is that a license is not a sale<sup>159</sup> and, therefore, the “first sale doctrine” is not applicable.<sup>160</sup> Nevertheless, the Courts seem to take for granted that a software license is a sale of goods subject to UCC or, at least, the equivalent to it.<sup>161</sup> Moreover, the Courts agree that licenses are enforceable unless their terms are objectionable on grounds applicable to contracts in general (for example, if they violate a rule of positive law, or if they are unconscionable).<sup>162</sup>

From the preceding statements and rules some authors have drawn the conclusion that two different definitions of license are possible: a non-contractual license, when it authorizes acts restricted by copyright; and a contractual license, when it is supported by consideration.<sup>163</sup>

### **III. The dual solution.**

The literature has tried to categorize the GNU GPL as a whole. However, the recipients of free software can be divided in two groups: mere users and redistributors.

---

<sup>159</sup> *ISC-Bunker Ramo Corp. v Altech, Inc.*, 765 F.Supp. 1310, 1331.

<sup>160</sup> “Notwithstanding the provisions of section 106(3) [distribution right], the owner of a particular copy or phonorecord lawfully made under this title, or any person authorized by such owner, is entitled, without the authority of the copyright owner, to sell or otherwise dispose of the possession of that copy or phonorecord.” § 109(a) Copyright Act. The House Report 94-1476 underlines that § 109(a) Copyright Act “restates and confirms the principle that, where the copyright owner has transferred ownership of a particular copy or phonorecord of a work, the person to whom the copy or phonorecord is transferred is entitled to dispose of it by sale, rental, or any other means.”

<sup>161</sup> “So is the purchase of software a transaction in goods? Despite Article 2's requirement of a *sale*, courts in Massachusetts have assumed, without deciding, that Article 2 governs software *licenses*. [decisions omitted] (...) Admittedly, the UCC technically does not govern software licenses, [but] the UCC best fulfills the parties' reasonable expectations.” *I.LAN Systems, Inc. v. Netscout Service Level Corp.*, 183 F.Supp.2d 328, 331, 332 (D. Mass 2002). Phillip Johnson, *All Wrapped up? – A Review of the Enforceability of “Shrink-wrap” and “Click-wrap” Licenses in the United Kingdom and the United States*, 2 European Intellectual Property Review 98, 100 (2003).

<sup>162</sup> *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1449.

<sup>163</sup> Diane Rowland & Andrew Campbell, *Supply of Software: Copyright and Contract Issues*, 10 International Journal of Law and Information Technology 23, 25, 26 (2002).

Mere users may get free software at no charge. In other words, they do not have any obligation in exchange for reproducing the software. Moglen explains that the GNU GPL “license does not require anyone to accept it in order to acquire, install, use, inspect, or even experimentally modify GPL'd software. (...). The free software movement thinks all those activities are rights, which all users ought to have; we don't even *want* to cover those activities by license. Almost everyone who uses GPL'd software from day to day needs no license, and accepts none.”<sup>164</sup>

Beyond the philosophic foundations of the free software movement, it is a fact that Copyright law recognizes the author's exclusive right to authorize the reproduction of the work. That means that a person who is downloading a computer program onto his hard drive or copies it from a CD is making a reproduction and, therefore, needs the authorization of the author. Although the GNU GPL does not contain the authorization to make copies for the user's own use, a right to make private copies without restrictions can be derived from the recognition of the right to make copies and distribute them, provided that they are released under the terms of the same license. That is, if the redistribution of the copies demands observance of the copyleft clause, the non-redistributed copies can be made without any requisite.<sup>165</sup> Moreover, the license does recognise explicitly a right to run the program (Section 0). If that argument is not

---

<sup>164</sup> Eben Moglen, *supra* note 153.

<sup>165</sup> This license to use the work for private purposes has been included in GPLv3 Draft: “Propagation of covered works is permitted without limitation provided it does not enable parties other than you to make or receive copies” (Section 2 Paragraph 3). The term “propagation” is an invention of the GPLv3 drafter and includes approximately all the exclusive rights of the copyright holder: “To “propagate” a work means doing anything with it that requires permission under applicable copyright law, other than executing it on a computer or making private modifications. This includes copying, distribution (with or without modification), sublicensing, and in some countries other activities as well” (Section 0 paragraph 2).

persuasive enough, a license may be implied by the conduct of the licensor<sup>166</sup> (for example, uploading a computer program in a server publicly available). In both cases, we are facing a mere permission, that is, a license in the strict sense.

There are also redistributors, to whom the GNU GPL is directed. As we have seen, the Free Software Foundation advocates that the GNU GPL is a license to them. This position is reflected in Section 5: “You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.” Although the Section uses up to three times the word “license”, and with capital “L”, there are two facts suggesting that it is not a license, but a contract. First, the Section speaks about “acceptance”, which is the second step of the “meeting of minds” characteristic for a contract –the first step is the offer, in this case, the GNU GPL itself. Second, the Section imposes obligations on the licensee by referring to the rest of the license, in particular to the copyleft clause (Section 2(b)).<sup>167</sup> The licensee may copy and distribute the work, with or without modification, but in return he has to release the copies under the same terms. That is not a mere permission, but an exchange.<sup>168</sup> Therefore, the relation between licensor and licensee is a contract or a contractual license.

---

<sup>166</sup> LAWRENCE ROSEN, *supra* note 154, at 52.

<sup>167</sup> Jason B. Wacha, *supra* note 10, at 482.

<sup>168</sup> Peter Brown, *Legal Issues in the Open Source Community*, 780 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 309, 317, 318 (2004). In the

It appears that this question has not been yet addressed by any U.S. court. Nevertheless, in two settled cases where the enforceability of the GNU GPL was at issue, breach of the GPL as a contract was alleged.<sup>169</sup> In German Courts, the GNU GPL has been considered as a contract.<sup>170</sup>

Taking into account what has been said in this section, it can be stated that the distribution of free software is done under two different ways. In the first place, if the person getting the software is a mere user who does not redistribute the software, he obtains his rights through a license. In the second place, if the person who gets the software redistributes it, with or without modification, he will be bound by certain obligations by virtue of the copyleft clause, and these duties do not follow directly from the Copyright Act, but from a contract or, to use the more common terminology in the field, from a contractual license.

The understanding of the GNU GPL as a contract in the cases mentioned above raises various contractual issues, which the Free Software Foundation wanted to avoid. These

---

German literature, Thomas Hoeren, *Open Source und das Schenkungsrecht – eine durchdachte Liason?*, RECHT UND RISIKO, FESTSCHRIFT FÜR HELMUT KOLLHOSSER ZUM 70 GEBURTSTAG, 229, 237 (Reinhard Bork ed., 2004).

<sup>169</sup> See Countercl., at ¶¶ 110 to 118, *Progressive Software Corp. v. MySQL AB*, 195 F. Supp. 2d 328 (D. Mass. 2002); First Am. Compl. ¶ 50, *MontaVista Software, Inc. v. Lineo, Inc.*, No. 2:02 CV-0309J (D. Utah filed July 23, 2002). See Jason B. Wacha, *supra* note 10, at 456.

<sup>170</sup> “[T]he Panel has no doubt whatsoever that the general business conditions have been effectively incorporated into a possible contractual relationship between the defendant and the plaintiff pursuant to German Civil Code Section 305 Para. 2.” *LG München I*, Urt. V. 19.5.2004 – 21 O 6123/04, available at [http://www.jbb.de/urteil\\_lg\\_muenchen\\_gpl.pdf](http://www.jbb.de/urteil_lg_muenchen_gpl.pdf) (translation into English available at [http://www.jbb.de/judgment\\_dc\\_munich\\_gpl.pdf](http://www.jbb.de/judgment_dc_munich_gpl.pdf)). See *LG München I: Wirksamkeit einer GPL-Lizenz*, *Computer und Recht*, 774 to 780 (2004). In the U.S. literature, see Peter Brown, *SCO v. IBM: Other Legal Issues in the Open Source Community*, 808 *Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series* 103, 113, 114. (2004).

uncertainties affect the contract formation, the possibility of passing obligations and rights to third parties through the license, the content of the license (in particular, the warranty disclaimer) and the complex relation between Copyright and Contract law. The next two sections will be devoted to contract formation: consideration and shrinkwrap-clickwrap licenses.

### *3. Consideration.*

#### **I. Introduction.**

In the common law system, to make a promise enforceable, the promisee has to do something that is either a detriment to the promisee or a benefit to the promisor.<sup>171</sup> That is called consideration. Without consideration, a contract cannot be formed.

#### **II. Definition.**

The Restatements of Contracts have defined consideration in terms of bargain. Restatement, Second, of Contracts, § 71 [Requirement of Exchange; Types of Exchange] states:

- (1) To constitute consideration, a performance or a return promise must be bargained for.
- (2) A performance or return promise is bargained for if it is sought by the promisor in exchange for his promise and is given by the promisee in exchange for that promise.
- (3) The performance may consist of

---

<sup>171</sup> “It is only when the party making the promise gains something, or he to whom it is made loses something, that the law gives the promise validity.” *Mills v. Wyman*, 3 Pick. 207 (Mass. 1825).

- a. an act other than a promise, or
- b. a forbearance, or
- c. the creation, modification, or destruction of a legal relation.

(4) The performance or return promise may be given to the promisor or to some other person. It may be given by the promisee or by some other person.

### **III. Peppercorn theory.**

The consideration requirement is relaxed by the fact that “[v]irtually anything that anyone would bargain for in exchange for a promise can be consideration for that promise.”<sup>172</sup>

The Courts often use the hyperbole that even a peppercorn can constitute consideration.<sup>173</sup>

### **IV. Consideration in copyleft software licenses.**

Under the GNU GPL or any other copyleft software license the two parties are the copyright holder or licensor and the redistributor or licensee. On the one hand, licensor’s consideration is clearly the authorization to modify and distribute the software. On the other hand, licensee’s consideration is more obscure. It has been argued by some legal scholars that the licensor is not getting anything in return, which means that no contract is formed.<sup>174</sup> However, it seems more reasonable to understand that licensee’s consideration

---

<sup>172</sup> E. ALLAN FARNSWORTH, *supra* note 158, at 48.

<sup>173</sup> “A cent or a pepper corn, in legal estimation, would constitute a valuable consideration.” *Whitney v. Stearns*, 16 Me. 394 (1839).

<sup>174</sup> LAWRENCE ROSEN, *supra* note 154, at 63 – 65. This author considers that the GNU GPL would be enforceable under the doctrine of promissory estoppel (see Restatement, Second, Contracts, § 90).

is his promise to abide by the copyleft clause.<sup>175</sup> Therefore, the agreement is completely enforceable.

#### *4. Clickwrap and shrinkwrap agreements.*

##### **I. Introduction.**

Proprietary software is not sold but rather only licensed.<sup>176</sup> Under this strategy, the software industry aspires to design its own rights and duties, even beyond the rights granted by Copyright law.<sup>177</sup> The release of the software is done under standard form contracts or “mass market licenses,”<sup>178</sup> in particular, shrinkwrap or clickwrap licenses.

---

<sup>175</sup> Jason B. Wacha, *supra* note 10, at 475; Brian W. Carver, *supra* note 82, at 458.

<sup>176</sup> “Plaintiff argues that (...) Microsoft never sells but rather only licenses its Products.” Microsoft Corporation v. Harmony Computers & Electronics, Inc., 846 F.Supp. 208, 210. “Adobe claims all Adobe software products are subject to a shrink-wrap End User License Agreement (“EULA”) that prohibits copying or commercial re-distribution.” *Adobe Systems, Inc. v. Stargate Software Inc.*, 216 F.Supp.2d 1051, 1052 (N.D. Cal 2002).

<sup>177</sup> Christian H. Nadan, *Software Licensing in the 21<sup>st</sup> Century: are Software “Licenses” really Sales, and How will the Software Industry respond?*, 32 American Intellectual Property Law Association Quarterly Journal, 555, 559, 560 (2004). “Although the enforceability of such agreements has been in doubt for many years, software vendors have continued to use them, instead of merely relying on the protection provided by copyright law, for a number of reasons: 1. To negate the copyright law “first sale doctrine,” which provides that once a copy of a copyrighted work has been sold, the copyright holder’s rights in that copy are exhausted, and the copy may be freely resold, leased, rented, lent or otherwise disposed of without the copyright holder’s consent. 2. To limited or disclaim warranties, remedies and liability as permitted by the provisions of the Uniform Commercial Code (UCC). 3. To impose other limitations on the transaction, such as use limitations on the software, prohibitions on reverse engineering, choice of law and forum, shortened statute of limitations, export control provisions, etc.” Michael D. Scott, *Protecting Software Transactions Online: the Use of “Clickwrap” Licenses*, 482 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 101, 103, 104 (1997).

<sup>178</sup> The three most remarkable characteristics of a mass market license are: “1) Their acceptance is indicated by some act other than a signature on writing; 2) they are not negotiated but are, rather a “take it or leave it” type of agreement; and 3) they are intended for use with mass market products or services by a large end user community for which negotiated licenses would be financially, administratively or otherwise infeasible.” Terry J. Hardi, *Mass Licensing – Part 1: Shrinkwraps, Clickwraps and Browsewraps*, 831 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 251, 257 (2005). See as well, Jean Braucher, *The Failed Promise of the UCITA Mass-Market Concept and its Lessons for Policing of Standard Form Contracts*, 7 Journal of Small and Emerging Business Law, 393 (2003).

The literature defines a shrinkwrap as “a license agreement that is usually contained in a box of software, which states that by opening the package, you agree to the terms of the license agreement,”<sup>179</sup> whereas clickwrap licenses “are a form of license used in an interactive manner on a computer. Typically, the user is presented with a display on a computer screen of the license and is prevented from proceeding with downloading or installation of the software until such time that he or she has indicated assent by clicking on a radio button on the computer monitor display”<sup>180</sup> –hence the name clickwrap license.

The shrinkwrap licenses can be categorized into different subgroups: a) in box license, where the license is enclosed with the product in a sealed envelope; b) box-top license, which can be read before opening the box; and c) referral license, where there is a sticker indicating that the CD-ROM should not be opened prior to reading the license. The clickwrap licenses can be presented in the following ways: a) prior to download a scroll-box appears and the users is asked to read a license and click “I agree”; b) the license is shown in a similar way, but during the installation of the program; and c) the so called browsewrap licenses, a variation of clickwrap licenses, which are ordinarily found in online transactions and whereby the user is informed of the existence of a license but not required to go through them in order to proceed<sup>181</sup>.

---

<sup>179</sup> Christian H. Nadan, *supra* note 32, at 362. See as well, Mark A. Lemley, *Shrinkwraps in Cyberspace*, *Jurimetrics Journal* 311, 312 (1995).

<sup>180</sup> Terry J. Hardi, *supra* note 178, at 256. See as well, Christian H. Nadan, *supra* note 32, at 362 (2002).

<sup>181</sup> Phillip Johnson, *supra* note 161, at 98.

## II. Shrinkwrap and clickwrap licenses in the Courts.

U.S. Courts have had much opportunity to deal with the enforceability of shrinkwrap licenses. The decisions are not completely uniform, but main guidelines can be drawn. An early case in this field is *Step-Saver Data Systems, Inc. v. Wyse Technology*<sup>182</sup>, where the plaintiff, *Step-Saver*, developed and marketed a multi user computer system with hardware from *Wyse* and software from *TSL*. Almost immediately upon installation of the system, *Step-Saver* began to receive complaints from most of its customers and at least twelve of them filed suit against him. *Step-Saver* reacted by suing his providers. The orders were placed by telephone, but on the package of each copy of the program was printed a license with a disclaimer of warranties. The issue in this case is the enforceability of such a license. The Court agreed with *Step-Saver*'s contention that the contract for each copy of the program was formed when *TSL* agreed, on the telephone, to ship the copy at the agreed price. The box-top license, as *Step-Saver* argued, was a material alteration to the parties's contract containing additional terms which did not become a part of the contract.

Some years later, the Courts changed their position. The leading case in the field of shrinkwrap licenses is *ProCD Inc. v. Zeidenberg*,<sup>183</sup> where the plaintiff, *ProCD*, compiled information from more than 3,000 telephone directories into a computer database, which could be searched according to users' criteria. The product was sold in a box declaring that the software came with restrictions stated in an enclosed license. This license was printed in a manual inside the box and appeared on the user's screen every time the

---

<sup>182</sup> 939 F.2d 91 (3<sup>rd</sup> Cir. 1991).

<sup>183</sup> 86 F.3d 1477 (7<sup>th</sup> Cir. 1996).

software ran. The license limited the use of the application program and listings to non-commercial purposes. However, one of the buyers, Matthew Zeidenberg, made the information available over the World Wide Web for a price. *ProCD* filed suit, but the district court accepted the view of the defendant that a contract includes only the terms on which the parties have agreed and not the hidden terms. As a consequence, the Judge held the license void because their terms did not appear on the outside of the packages. On appeal, the Circuit Court reversed the decision on the basis that “[n]otice on the outside, terms on the inside, and a right to return the software for a refund if the terms are unacceptable (a right that the license expressly extends), may be a means of doing business valuable to buyers and sellers alike.” The Court pointed out that contracts in which the exchange of money precedes the communication of detailed terms are common (e.g. insurance policies where the policy is received after payment of premium, airline tickets with terms received after ordering via telephone, etc.) In short, the Court endorsed the shrinkwrap license.

The analytical difference between *Step-Saver* and *ProCD* is whether “money now, terms later” forms a contract (i) at the time of the purchase order or (ii) when the purchaser receives the box of software, sees the license agreement, and does not return the software.<sup>184</sup> In the first scenario (adding terms to an existing contract), the UCC § 2-207<sup>185</sup> is applicable. Therefore if the offeree is a consumer, the new terms need to be

---

<sup>184</sup> Chief Justice Young elucidates and distinguishes both *Step-Saver* and *ProCD* and finally adapts the *ProCD* analysis in *I.Lan Systems, Inc. v. Netscout Service Level Corp.*, 183 F.Supp.2d 328 (D. Mass. 2002).

<sup>185</sup> UCC § 2-207(2) “The additional terms are to be construed as proposals for addition to the contract. Between merchants such terms become part of the contract unless: (a) the offer expressly limits acceptance to the terms of the offer; (b) they materially alter it; or (c) notification of objection to them has already been given or is given within a reasonable time after notice of them is received.”

expressly accepted; and if it is a merchant, the acceptance may be implicit under certain circumstances –but not if the new terms materially alter the agreement. In the second scenario (forming a contract), the relevant provision is UCC § 2-204,<sup>186</sup> which emphasizes that the parties can form a contract in any manner sufficient to show agreement.

Later decisions have supported *ProCD* interpretation (e.g. *Hill v. Gateway 2000, Inc.*,<sup>187</sup> *M.A. Mortenson Co., Inc. v. Timberline Software Corp.*,<sup>188</sup> *Adobe Systems, Inc. v. Stargate Software Inc.*, etc.)<sup>189</sup> with very few cases to the contrary (e.g. *Klocek v. Gateway, Inc.*)<sup>190</sup>

The clickwrap licenses present very similar questions to the ones of the shrinkwrap licenses. Therefore, the Courts have usually accepted their enforceability following the *ProCD* doctrine.<sup>191</sup> Nevertheless, the special technical features of the clickwrap licenses increase the chance that the offeree is not made aware of the existence of the license, which can lead to its unenforceability.<sup>192</sup>

---

<sup>186</sup> UCC § 2-204 “(1) A contract for sale of goods may be made in any manner sufficient to show agreement, including conduct by both parties which recognizes the existence of such a contract.

(2) An agreement sufficient to constitute a contract for sale may be found even though the moment of its making is undetermined.

(3) Even though one or more terms are left open a contract for sale does not fail for indefiniteness if the parties have intended to make a contract and there is a reasonably certain basis for giving an appropriate remedy.”

<sup>187</sup> 105 F.3d 1147 (7<sup>th</sup> Cir. 1997).

<sup>188</sup> 998 P.2d 305 (Wash. 2000).

<sup>189</sup> 216 F.Supp.2d 1051 (N.D. Cal 2002).

<sup>190</sup> “[I]f either party is not a merchant, additional terms are proposals for addition to the contract that do not become part of the contract unless the original offeror expressly agrees.” *Klocek v. Gateway, Inc.*, 104 F.Supp.2d 1332, 1341.

<sup>191</sup> *I.LAN Systems, Inc. v. Netscout Service Level Corp.*, 183 F.Supp.2d 328 (D. Mass 2002).

<sup>192</sup> In *Softman Products v. Adobe Systems*, each piece of *Adobe* software is “accompanied by an End User License Agreement (“EULA”), which sets forth the terms of the license between *Adobe* and the end user

This is particularly so in the case of browsewrap licenses. In *Specht v. Netscape Communications Corp.*,<sup>193</sup> the Court was asked to determine “whether plaintiffs-appellees (“plaintiffs”), by acting upon defendants' invitation to download free software made available on defendants' webpage, agreed to be bound by the software's license terms (which included the arbitration clause at issue), even though plaintiffs could not have learned of the existence of those terms unless, prior to executing the download, they had scrolled down the webpage to a screen located below the download button.” The Court concluded that consumers did not manifest assent, because the mentioned reference was not sufficient to make consumers aware of those terms. Therefore, consumers were not subject to the license agreement. In an apparently similar case, *Ticketmaster Corp. v. Tickets. Com. Inc.*,<sup>194</sup> the defendant ignored the prohibition of “deeplinking” included in the license agreement, which was linked to his homepage. The Court observed that “the terms and conditions are set forth so that the customer needs to scroll down the home page to find and read them. Many customers instead are likely to proceed to the event page of interest rather than reading the “small print.” It cannot be said that merely putting the terms and conditions in this fashion necessarily creates a contract with any one using the web site.” That being said, other Courts have suggested that a link is enough to create

---

for that specific *Adobe* product. The EULA is electronically recorded on the computer disk and customers are asked to agree to its terms when they attempt to install the software.” One of the terms of the license prohibits users from unbundling *Adobe's* Collections. Without installing the software, *SoftMan* unbundled it and sold the individual parts independently. The Court found “that *Adobe's* EULA cannot be valid without assent. Therefore, *SoftMan* is not bound by the EULA because it has never loaded the software, and therefore never assented to its terms of use.” 171 F. Supp. 2d 1075 (C.D. Cal 2001).

<sup>193</sup> 306 F.3d 17 (2<sup>nd</sup> Cir. 2002).

<sup>194</sup> 54 U.S.P.Q.2d 1344.

a binding agreement<sup>195</sup> and that it is not always necessary to click the “I Agree” button to be obliged by the terms of the license.<sup>196</sup>

Although the decisions are not uniform, they allow us to understand the main guidelines, though not absolute, used by the Courts to decide the enforceability of shrinkwrap and clickwrap licenses.<sup>197</sup> In any event, if the following measures are observed, the enforceability of the license is almost guaranteed. First, the offeror should give notice to the offeree at the time when the latter decides to acquire the product that the terms of a shrinkwrap or clickwrap license will govern their agreement<sup>198</sup> (e.g. through a statement in the box displayed in a shop, forcing the offeree to go through the terms of the license to download the software from a website, or probably it is enough to display in a visible place of the website a link which directs to the terms of the license.) A later notice (i.e., when the offeree receives the product) may be sufficient. However, the chance that Court will be tempted to declare unenforceable the additional terms under UCC § 2-207(2) are clearly higher. Second, the offeree should be forced to undertake positive steps to accept the license.<sup>199</sup> The implementation of this kind of procedure is extremely easy if the product is transferred online (i.e., clicking in a “I Accept” radio button). Nevertheless,

---

<sup>195</sup> *EF Cultural Travel BV v. Zefer Corporation*, 318 F.3d 58, 63 (1<sup>st</sup> Cir. 2003).

<sup>196</sup> “We recognize that contract offers on the Internet often require the offeree to click on an ‘I agree’ icon. (...) But not in all circumstances. While new commerce on the Internet has exposed courts to many new situations, it has not fundamentally changed the principles of contract. It is standard contract doctrine that when a benefit is offered subject to stated conditions, and the offeree makes a decision to take the benefit with knowledge of the terms of the offer, the taking constitutes an acceptance of the terms, which accordingly become binding on the offeree.” *Register.com, Inc. v. Verio, Inc.*, 356 F.3d 393, 403 (2<sup>nd</sup> Cir. 2004).

<sup>197</sup> “It does not seem fair to state that [t]he only thing that is certain when considering the enforceability of shrink-wrap and click-wrap software licence is that the law is *uncertain*.” Phillip Johnson, *supra* note 161, at 102.

<sup>198</sup> Terry J. Hardi, *supra* note 178, at 273.

<sup>199</sup> Christian H. Nadan, *supra* note 32, at 362.

other forms of acceptance, even remaining silent, are also possible<sup>200</sup> but the risk to be declared void is higher. Alternatively, the recipient must have the opportunity to return the product if he does not agree with the terms.

### **III. Enforceability of copyleft licenses as shrinkwrap or clickwrap contracts.**

The questions raised by the features of the GNU GPL as a shrinkwrap or clickwrap license have been considered the most serious argument against its enforceability.<sup>201</sup>

Normally, the way in which a work is licensed can be observed in practice, but it is not described anywhere. However, being the goal of the Free Software Foundation the application of the GNU GPL to downstream licensors, it provides some recommendations of how to attach a notice informing of the application of the license.<sup>202</sup> Following the Free Software Foundation guidelines, this notice<sup>203</sup> should be attached “to the start of

---

<sup>200</sup> Terry J. Hardi, *supra* note 178, at 273.

<sup>201</sup> Jason B. Wacha, *supra* note 10, at 488.

<sup>202</sup> These recommendations are to be found on the same webpage that the license itself, under the title *How to Apply These Terms to Your New Programs*, <http://www.gnu.org/licenses/gpl.html>.

<sup>203</sup> The suggested content of the notice is: “*one line to give the program's name and an idea of what it does.*  
Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program [implicitly recommends the delivery of a copy of the GNU GPL]; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.”

One should “[a]lso add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they

each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.” In the content of the notice, it is also indicated where to mail to for obtaining a copy of the GNU GPL.

These recommendations are not part of the GNU GPL itself. Therefore, programmers can release their software under the GNU GPL without observing the proposed way to attach the license to the program.<sup>204</sup> The General Counsel of *MontaVista Software, Inc.*, a company which provides a Linux-based operating system to its customers, states that “[m]ost licensors get the GPL in one of two ways: they get a piece of paper with the GPL printed on it (but not normally “wrapped” around any box or piece of software) [physical delivery] or they get, along with the software, an electronic file containing the GPL (but normally without the file being designed as a clickwrap) [digital delivery].”<sup>205</sup>

The vulnerabilities of the GNU GPL licensing method are: the recipient of the software may not receive a notice of the license before delivery; and no signature or other manifestation of assent is generally required.<sup>206</sup> Moreover, it should not be disregarded

---

could even be mouse-clicks or menu items--whatever suits your program...”  
<http://www.gnu.org/licenses/gpl.html>.

<sup>204</sup> The only reference included in the license in relation with the way in which the terms of the license should be applied affects the downstream licensees-licensors and states that they have to “keep intact all the notices that refer to this License (...) and give any other recipients of the Program a copy of this License along with the Program” (Section 1 Paragraph 1).

<sup>205</sup> Jason B. Wacha, *supra* note 10, at 489, 490. See as well, Terry J. Iardi, *Mass Licensing – Part 2: Open Source Software Licensing*, 831 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 279, 287 (2005).

<sup>206</sup> The GNU GPL considers that “by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.” (Section 5). However, it seems rather dubious that the unilateral declaration of one of the parties can overrule the shrinkwrap and clickwrap agreements’ law.

that the recipient may not get a copy, physical or digital, of the GNU GPL license. All these factors, especially when they appear jointly, increase the risk that a Court declares the license unenforceable. Given that, some authors suggest that licensees are probably not bound by the terms of the license.<sup>207</sup>

Nevertheless, there are arguments to the contrary. First, the U.S. Courts do not seem to consider the lack of delivery of a copy of the license (referral license) as a decisive factor in excluding its application.<sup>208</sup> Second, the National Conference of Commissioners on Uniform State Laws has drafted the Uniform Computer Information Transactions Act (UCITA), which has introduced provisions (§§ 209<sup>209</sup> and 210) to guarantee the

---

<sup>207</sup> Christian H. Nandan, *supra* note 32, at 362, 363.

<sup>208</sup> In opposition, European countries are more strict with this requisite, for example, in Spain Article 5.2 of the Contractual Standard Terms Act 7/1998 (April 13<sup>th</sup>, 1998) states that the standard terms will not be incorporated into the contract unless the offeror has informed the offeree of their existence and has provided him with a copy of them. The Section 2(1) German Contractual Standard Terms Act (June 29<sup>th</sup>, 2000), later integrated in the Section 305(2) Civil Code (January 12<sup>th</sup>, 2002), is more flexible in this point, as it only demands that the offeree informs at the time of the contract formation of the standard terms existence and facilitates him the possibility to get knowledge of their content in a reasonable way.

<sup>209</sup> “§ 209. Mass-Market License.

(a) [Limitation on terms.] Adoption of the terms of a mass-market license under Section 208 is effective only if the party agrees to the license, such as by manifesting assent, before or during the party's initial performance or use of or access to the information. A term is not part of the license if:

(1) the term is unconscionable or is unenforceable under Section 105(a) or (b);

(2) subject to Section 301, the term conflicts with a term to which the parties to the license have expressly agreed;

(3) under Section 113, the licensee does not have an opportunity to review the term before agreeing to it; or

(4) the term is not available to the licensee after assent to the license in one or more of the following forms:

(A) an immediately available nonelectronic record that the licensee may keep;

(B) an immediately available electronic record that can be printed or stored by the licensee for archival and review purposes; or

(C) in a copy available at no additional cost on a reasonable request in a record by a licensee that was unable to print or store the license for archival and review purposes.

(b) [Right of return and reimbursement.] If a mass-market license or a copy of the license is not available in a manner permitting an opportunity to review by the licensee before the licensee becomes obligated to pay and the licensee does not agree, such as by manifesting assent, to the license after having an opportunity to review, the licensee is entitled to a return under Section 113 and, in addition, to:

(1) reimbursement of any reasonable expenses incurred in complying with the licensor's instructions for returning or destroying the computer information or, in the absence of instructions, expenses incurred for return postage or similar reasonable expense in returning the computer information; and

(2) compensation for any reasonable and foreseeable costs of restoring the licensee's information processing system to reverse changes in the system caused by the installation, if:

enforceability of shrinkwrap and clickwrap agreements.<sup>210</sup> However, only Maryland and Virginia have enacted the UCITA as law so far. Finally, the license actually affects only the persons who intend to modify and/or redistribute the software, as the license does not impose any obligations on mere users. If we are dealing with a programmer who wants to modify software released under the GNU GPL, he will necessarily open a source file and there is placed according to the recommendations of the Free Software Foundation a pointer to the full notice, which directs in turn to the license.

#### **IV. Unenforceability of copyleft licenses as shrinkwrap or clickwrap contracts.**

The enforceability of the GNU GPL will depend on the variables that have been discussed before, and there will be a considerable number of border cases. Therefore, the question arises as to what will happen if the license is declared unenforceable. Some authors argue that in such a situation, the licensees will have an implied license for modifying the free software and redistributing it as proprietary.<sup>211</sup>

In my opinion, the situation is not that disastrous for the interests of the Free Software Foundation. In the absence of a contract because the license is declared void, the open

- 
- (A) the installation occurs because information must be installed to enable review of the license; and
  - (B) the installation alters the system or information in it but does not restore the system or information after removal of the installed information because the licensee rejected the license.
  - (c) [Licensor's opportunity to review.] In a mass-market transaction, if the licensor does not have an opportunity to review a record containing proposed terms from the licensee before the licensor delivers or becomes obligated to deliver the information, and if the licensor does not agree, such as by manifesting assent, to those terms after having that opportunity, the licensor is entitled to a return.
  - (d) [Notice of refund.] In a case governed by subsection (b), notice must be given in the license or otherwise that a refund may be obtained from the person to which the payment was made or other person designated in the notice if the licensee refuses the terms."

<sup>210</sup> Paradoxically, the Free Software Foundation advocates against the UCITA. They consider that the GNU GPL is not a shrinkwrap or clickwrap license, in opposition to the ones used by proprietary software companies. Free Software Foundation, *Why We Must Fight UCITA*, available at <http://www.gnu.org/philosophy/ucita.html>.

<sup>211</sup> Christian H. Nandan, *supra* note 32, at 362 – 367.

source software will still be protected by the Copyright Act.<sup>212</sup> If the program is made available to the public, it could be argued that there is an implied license to run the software, but the implied license cannot be read more extensively. Therefore, there is no right to modify and redistribute the software. It is clearly contradictory to state that the licensee can take advantage of the rights granted in the GNU GPL, but he is not compelled by any of its obligations.

The Free Software Foundation relies on the enforceability of the license. But it may be presumed that if the license were unenforceable, they would claim the applicability of Copyright Law. The Section 5 GNU GPL states that “[y]ou are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License...” One could read: “these actions are prohibited by Copyright law if the license is declared void.”

## *5. Privity.*

### **I. Introduction.**

As an elementary principle of the common law system, contractual rights and duties can only be conferred or imposed on the parties to a contract.<sup>213</sup> In legal terms, that is described as the doctrine of “privity of contract”.<sup>214</sup> “The doctrine of privity means that a person cannot acquire rights or be subject to liabilities arising under a contract to which

---

<sup>212</sup> LAWRENCE ROSEN, *supra* note 154, at 54.

<sup>213</sup> PATRICK S. ATIYA, AN INTRODUCTION TO THE LAW OF CONTRACT 356 (5<sup>th</sup> ed. 1995).

<sup>214</sup> “Privity of contract. The relationship between the parties to a contract, allowing them to sue each other but preventing a third party from doing so”. BLACK’S LAW DICTIONARY 1237 (8<sup>th</sup> ed. 2004).

he is not a party.”<sup>215</sup> In other words, if X sells a car to Y under the condition that the car will not be sold to Z, and Y does sell the car to Z, X will not be able to file suit against Z to recover the car because X is not in privity with Z.

## **II. Privity concerns in copyleft software licenses.**

The intention of the Free Software Foundation is to leave in the hands of the author of the program the power of filing suit against any possible infringer, both licensees and downstream sublicensees. With this goal, Section 6 GNU GPL states that: “[e]ach time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions.”<sup>216</sup> (...) You are not responsible for enforcing compliance by third parties to this License.”

However, the copyleft clause implies passing obligations and rights to third parties through the license. The legal feasibility of this process has been questioned by a possible lack of privity.<sup>217</sup> That means that the copyright holder could only file suit against his licensee, but not against any downstream sublicensee.

---

<sup>215</sup> GUENTER TREITEL, *THE LAW OF CONTRACT* 538 (8<sup>th</sup> ed. 1991).

<sup>216</sup> As noted in *Harris v. Emus Records Corp.*, under U.S. law, non-exclusive copyright licenses are personal and not assignable unless expressly permitted by the license or with permission of the copyright holder: “copyright license[s] (...) are not transferable as a matter of law. Under the 1909 Act, Absent any contractual limitations, an assignee [of the whole contract] had the right to re-assign the work. A licensee, however, had no right to re-sell or sublicense the rights acquired unless he has been expressly authorized to do so.” 734 F.2d 1329, 1333 (9<sup>th</sup> Cir. 1984). Lacking an express permission in the license, the subsequent licensors will be forced to look for the direct permission of the copyright holder, which may difficult the transferability of copyleft software.

<sup>217</sup> The copyleft clause “purports to restrict subsequent transferees who receive software from a licensee (...). As this new transferee is not in privity with the original copyleft licensor, the stipulation seems unenforceable.” Robert P. Merges, *The End of Friction? Property Rights and Contract in The “Newtonian” World of On-line Commerce*, 12 Berkeley Technology Law Journal, 115, 129 (1997).

### III. Relativization of privity concerns in copyleft software licenses.

Although the lack of privity between the author of the program and the downstream sublicensees seems clear, its consequences are alleviated through different paths.

First of all, there are certain exceptions to the doctrine of privity.<sup>218</sup> Particularly relevant to our discussion is the exception for third party beneficiaries. However, to determine who has interest in the performance of a contract entered into by others so sufficient as to allow him to enforce the provisions of the contract presents some difficulties. “Perhaps the most widely used “test” is that a person who wishes to enforce a contract to which he is not a party must show that the contract was intended for his benefit in either all or part of its contemplated performance. Some courts (...) add that if there is doubt concerning such intent the doubt will be resolved against the existence of the required intent, since parties are presumed to contract for themselves [decisions omitted].”<sup>219</sup> Therefore, the exception of third party beneficiaries does not seem a safe way to solve the problem.

Secondly, the requirement of privity has been relaxed under modern laws<sup>220</sup> and generally replaced by the doctrines of implied warranty and strict liability, which allow a third-party beneficiary or other foreseeable user to sue the seller of a defective product.<sup>221</sup> However, this relaxation only goes upstream from consumer to manufacturer. While it seems that the sublicensees will have a strong case against the copyright holder, if the

---

<sup>218</sup> GUENTER TREITEL, AN OUTLINE OF THE LAW OF CONTRACT 253 to 262 (Oxford University Press 6<sup>th</sup> ed., 2004).

<sup>219</sup> *Clearwater Constructors, Inc. v. Gutierrez*, 626 S.W.2d 789, 791 (Tex. 1981).

<sup>220</sup> It has been also argued that privity considerations have all but disappeared after the introduction of the Uniform Commercial Code (UCC) and the various state statutes related to the UCC. Jason B. Wacha, *supra* note 10, at 475.

<sup>221</sup> BLACK’S LAW DICTIONARY 1238 (8<sup>th</sup> ed. 2004).

software is defective –warranty waivers apart; it is more doubtful that the copyright holder will have a comparable interest to sue the sublicensees.

Finally, even assuming that the lack of privity is a defense for the downstream sublicensees against the contractual claims of the author of the program, the result would not be that dramatic as expected. In such a situation, a third party who does not respect the terms of the license does not infringe the contract with the author –because there is no such a contract at all; but he does infringe the author’s exclusive rights of Copyright. Consequently, the author cannot demand from any user of the software to observe the terms of the license, but he is able to block anyone from using the software thanks to Copyright. Moreover, the Free Software Foundation does not force anyone to transform proprietary software into free software because he or she used copyleft code. Instead, it gives to the infringer the choice between stop using the stolen code and writing his own, or releasing the software under the GPL.<sup>222</sup> Given that policy, the outcome of simply applying Copyright is not that different to having an enforceable agreement.

---

<sup>222</sup> *Supra* note 157.

## 6. *Warranty disclaimer.*

### **I. Introduction.**

A warranty disclaimer is included in the GNU GPL<sup>223</sup> and in other open source licenses.<sup>224</sup> That is also a common practice in the proprietary software licenses. This waiver may apply in different relationships where a transfer of open source software occurs. The most common ones are: between the author of the program and a redistributor (i.e., Red Hat); between the author of the program and an end user (e.g. the programmer makes available the software in a webpage for download); and between a redistributors and an end user (e.g. Red Hat and its customers). The differences between

---

<sup>223</sup> “**NO WARRANTY**

**11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.**

**12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.”** Sections 11 and 12 GNU GPL, available at <http://www.gnu.org/licenses/gpl.html>.

<sup>224</sup> For example, in the new BSD license: “THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”, available at <http://www.opensource.org/licenses/bsd-license.php>.

these three kinds of subjects taking part in the transfer of the open source software are consequential to the validity or invalidity of the waiver, as it will be shown below.

## **II. Merchants.**

If we accept that the UCC is applicable to the sale of software,<sup>225</sup> it is necessary to direct the attention to § 2-314 [Implied Warranty: Merchantability; Usage of Trade], which states: “(1) Unless excluded or modified (Section 2-316), a warranty that the goods shall be merchantable is implied in a contract for their sale if the seller is a merchant with respect to goods of that kind...”

Therefore, the implied warranty exclusively applies when only one of the parties is a “merchant”, which is defined as “a person who deals in goods of the kind or otherwise by his occupation holds himself out as having knowledge or skill peculiar to the practices or goods involved in the transaction or to whom such knowledge or skill may be attributed by his employment of an agent or broker or other intermediary who by his occupation holds himself out as having such knowledge or skill” (§ 2-104). Under this provision, a person will be a merchant based on specialized knowledge of either the goods or the practices involved.<sup>226</sup>

Although it would be necessary a case-by-case analysis to determine who is a merchant in every particular situation, it is generally true that companies such as Red Hat, which

---

<sup>225</sup> Jason B. Wacha, *supra* note 10, at 471. However, Nimmer expresses deep doubts, Raymond Nimmer, *supra* note 82, at 88.

<sup>226</sup> E. ALLAN FARNSWORTH, *supra* note 158, at 38.

sell open source software for a fee, will be seen as merchants, while programmers who make available their works in the Internet for free will not.<sup>227</sup>

The merchants are not trapped by the implied warranty though. They may exclude it, as far as the language is conspicuous (§ 2-316(2)). In order to fulfill this requirement, the GNU GPL introduces the two sections of the warranty disclaimer with a bold and capitalized title, which reads: “**NO WARRANTY**”. The text of both sections is capitalized as well. That seems to meet the exclusion requirement. Moreover, UCC § 2-316(3)(a) foresees that “unless the circumstances indicate otherwise, all implied warranties are excluded by expressions like ‘as is’...” and this precise expression is used and emphasized in Section 11 GNU GPL. Therefore, the GNU GPL warranty disclaimer apparently meets the spirit and requisites of the UCC.<sup>228</sup> And this conclusion is coherent with the UCITA § 406 [Disclaimer of modification of warranty].

Nonetheless, two warnings are worth attention. First, if the license is void as a shrinkwrap or clickwrap license, according to the requirements which have been explained before, the warranty disclaimer will not pass the exclusion test. Second, a few states have enacted statutes prohibiting disclaimers of implied warranties.<sup>229</sup>

---

<sup>227</sup> Raymond Nimmer, *supra* note 82, at 89.

<sup>228</sup> Jason B. Wacha, *supra* note 10, at 472.

<sup>229</sup> E. ALLAN FARNSWORTH, *supra* note 158, at 309. This waiver would be also considered void in many jurisdictions around the world, like Germany. Axel Metzger & Till Jaeger, *Open Source Software und deutsches Urheberrecht*, 10 GRUR Int, 839, 846, 847 (1999).

### **III. Non-merchants.**

Common sense tells us that if a merchant can avoid the application of the implied warranty, a non-merchant should be even in a much better position. However, a more accurate analysis is still warranted.

In the relation between the author of the program and the redistributor, we have to assume that there is a contract under the copyleft software license –otherwise, the redistributors would be infringing the copyrights of the author. As we have seen before, the license meets the UCC requirements to exclude the implied warranty. Moreover, the author of the program is generally not a merchant and, therefore, the implied warranty of merchantability is not even applicable. In conclusion, the author of the program has no responsibility against the redistributors not only because the warranty disclaimer is valid, but also because of the lack of any responsibility due to his non-merchant position.

The conclusion drawn in the preceding paragraph is supported by UCITA. § 410 [No implied warranties for free software], which states “(a) [Free software defined.] In this section, "free software" means a computer program with respect to which the licensor does not intend to make a profit from the distribution of the copy of the program and does not act generally for commercial gain derived from controlling use of the program or making, modifying, or redistributing copies of the program. (b) [Implied warranties inapplicable.] The warranties under Sections 401 [Warranty and obligations concerning noninterference and noninfringement] and 403 [Implied warranty. Merchantability of computer program] do not apply to free software.”

Finally, the relationship of the copyright holder and the end user differs from the one between redistributor and end user because the former is arguably not bound by a contract, as we have seen before<sup>230</sup>. Implied warranties can be modified by contract, but not by a condition to a license<sup>231</sup>. Therefore, if there is no agreement between the parties, the warranty disclaimer does not have any effect. However, it is necessary to underline that the implied warranty will not emerge in the first place. The author of the program and the end user will be third parties to each other and, therefore, the author could only be responsible under the much more relaxed standards of tort law<sup>232</sup>.

## 7. *Copyright v. Contract.*

### **I. Introduction.**

The author of a program has three different and cumulative methods to control the exploitation of his work: copyrights, contracts and technological protection measures (e.g. copy protection devices). Undoubtedly the protection of the author's interests is crucial to promote the creation of works. However, it is a *non sequitur* that the more protection, the better. Copyright law tries to strike the right balance between authors' protection and the public interest to access information. If the authors use contracts and/or technological protection measures to go beyond Copyright borders, the balance

---

<sup>230</sup> See in this paper, III.2 Contract v. License. This departure point seems to be assumed by the drafters of the UCITA: "Many such transactions [among contributors to the development of so-called "open source" software and users] may not involve a contractual relationship and would, on that basis, fall outside of the scope of this Act." Comment a), UCITA § 410.

<sup>231</sup> Jason B. Wacha, *supra* note 10, at 482; Raymond Nimmer, *supra* note 82, at 89.

<sup>232</sup> A similar outcome has been reached in the German literature, where the open source software transfer from the programmer to the end user is considered by some authors as a donation and the liability standard is only met in cases of recklessness or malicious intent. Axel Metzger & Till Jaeger, *supra* note 229, at 847.

may break down<sup>233</sup>. The copyleft license uses both copyrights and contracts to achieve its goal and, therefore the question of whether Copyright can be overruled by the content of the copyleft license arises.

The means by which authors try to use contracts to override Copyright are multiple<sup>234</sup>. A paradigmatic example is to avoid the application of the first sale doctrine by not accepting that the software is sold, but insisting that it is only licensed.

## **II. Preemption.**

In a conflict between Copyright provisions and contract clauses, one could think of adopting one of the two following extreme solutions: always applying either Copyright law or the agreement between the parties. However, neither approach is satisfactory<sup>235</sup>. To determine which rule should prevail one needs to ask if the lack of protection under the Copyright regime is because a certain act should remain unprotected, or if the Copyright provision is simply a default rule that does not prevent protection under any other regimes, such as patents and contracts.

---

<sup>233</sup> Lucie Guibault, *Pre-emption Issues in the Digital Environment: Can Copyright Limitations be Overridden by Contractual Agreements Under European Law?* 22 (1998), available at [www.ivir.nl/Publicaties/guibault/ARTICLE2.doc](http://www.ivir.nl/Publicaties/guibault/ARTICLE2.doc).

<sup>234</sup> For example, the following software license states: “You [the user] may not transfer, sublicense, rent, lease, convey, copy, modify, translate, convert to another programming language, decompile or disassemble the Licensed Software for any purpose without VAULT’s prior written consent.” *Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255, 257 (La. 1988). This sample conflicts with the “first sale” doctrine (§ 109 Copyright Act), the right to make a back up copy (§ 117 Copyright Act) and the right to reverse engineering (§ 1201(f) Copyright Act).

<sup>235</sup> Mark A. Lemley, *Beyond Preemption: the Law and Policy of Intellectual Property Licensing*, 87 California Law Review 111, 136 (1999).

An optimal solution for the sake of certainty would be that the Statute would distinguish between mandatory and default Copyright rules. However, this task seem to be infeasible for the legislator, as such distinctions are not found in U.S. law and only rarely in foreign legislation.<sup>236</sup>

As a consequence, hard cases about when Copyright law will preempt the content of the contract are left to the Courts, which have had some difficulty in establishing clear guidelines in pre-emption cases based on contractual rights<sup>237</sup>. The leading case in this field establishes that “[a]lthough Congress possesses power to preempt even the enforcement of contracts about intellectual property (...) courts usually read preemption clauses to leave private contracts unaffected.”<sup>238</sup> This tendency to favour contracts over Copyright has been supported by the general acceptance of shrinkwrap-clickwrap licenses and the approval of the UCITA<sup>239</sup>. However, on occasions the courts decided to preempt the content of the agreement, for example, in relation to a license prohibiting reverse engineering<sup>240</sup>.

In any event, if the copyright holder achieves the extension of his rights through a contractual agreement, the infringement of these rights will have only one remedy against

---

<sup>236</sup> For example, “[t]he making of a back-up copy by a person having a right to use the computer program may not be prevented by contract insofar as it is necessary for that use.” Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs; “[t]he maker of a database which is made available to the public in whatever manner may not prevent a lawful user of the database from extracting and/or re-utilizing insubstantial parts of its contents, evaluated qualitatively and/or quantitatively, for any purposes whatsoever.” Directive 96/9/EC of 11 March 1996 on the legal protection of databases.

<sup>237</sup> Maureen A. O’Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 Duke Law Journal 479, 479 (1995).

<sup>238</sup> *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1454.

<sup>239</sup> Pamela Samuelson, *Copyright and Freedom of Expression in Historical Perspective*, 10 Journal of Intellectual Property Law 319, 340, 341 (2003).

<sup>240</sup> *Vault Corp. v. Quaid Software, Ltd.*, 847 2.d 255 (5<sup>th</sup> Cir. 1988). Nonetheless, in the opposite direction ruled *Bowers v. Baystate Technologies, Inc.*, 320 F.3d 1327 (1<sup>st</sup> Cir. 2003).

the infringing party under the rules of contract law<sup>241</sup>. In other words, third parties who comply not with the agreement but Copyright law will escape responsibility.

### **III. Deviations of the copyleft licenses from the Copyright language.**

Generally, the GNU GPL uses the same language that the Copyright Act. However, there are some discrepancies. In particular, the GNU GPL gives its own definition of derivative work<sup>242</sup>, which differs from the one provided in the Copyright Act<sup>243</sup>. These deviations will probably be more important in the third version of the GNU GPL, where for example the concept of “propagation” is introduced,<sup>244</sup> which is completely alien to Copyright law.

The observed deviations seem to be minor and do not restrict access to information. Therefore, they would probably be upheld by the Courts. However, being part of a private agreement, these innovative definitions do not affect the rights of third parties, whose activities will be controlled exclusively by Copyright provisions.

---

<sup>241</sup> Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 Southern California Law Review, 1239, 1273 (1995).

<sup>242</sup> “[A]ny derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.” (Section 0).

<sup>243</sup> “A ‘derivative work’ is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications, which, as a whole, represent an original work of authorship, is a “derivative work”.” § 101 Copyright Act.

<sup>244</sup> *Supra* note 165.

## **IV.– Conclusion.**

Since the release of the GNU Manifesto in 1985, the free software presence has increased continuously. Nowadays, the free software represents a real alternative to proprietary software.

In the first part of the paper, I exposed the foundations and differences among the three main methods to license software: free, open source and proprietary.

The “free software” has two main features: first, it is free because it authorizes anyone to copy, distribute and/or modify (i.e. access the source code) the software; and second, it is copyleft because it forces any redistributor to recognize the right to copy, distribute and/or modify the software.

The concept “open source software” emerged later than “free software” and it is used to refer to the software licenses which authorize the licensees to copy, distribute and/or modify the software, but open source software licenses do not necessarily oblige redistributors to recognize these same rights to their licensees. Therefore, it is generally true that “free software” has become a subspecies of “open source software”, although there are some minor exceptions.

The “proprietary software”, in contrast with free and open source software, does not allow either modification, redistribution or, most commonly, none of those actions.

In the second part of the paper, I focused on the contractual issues in relation to free software and, in particular, the copyleft clause. However, I also referred to the peculiarities introduced by the Copyright nature of the subject matter when necessary.

The existence of a contract is characterized by the presence of agreement and exchange, which are not present in mere licenses. Therefore, I consider that free software licenses are Janus-like. On the one hand, they are a non-contractual copyright license that allows end users private utilization of the program. On the other hand, they are a contractual license that allows redistributors to copy, distribute and/or modify the software, under the condition of recognizing to any sublicensee the right to copy, distribute and/or modify the software.

As a consequence of Janus-like nature of copyleft licenses, the relationship between author of the program and end user is exclusively governed by Copyright. In opposition, the agreement between author and redistributor must be examined not only under Copyright, but also under contract law provisions.

The first issue arising due to the applicability of contract law is about the element of consideration in the agreement, which is indispensable for the existence of a valid contract. In my opinion, the consideration requirement is met in the relation between author of the program (licensor) and redistributors (licensee) because the licensor authorizes modification and distribution of the software and in exchange the licensee promises to abide by the copyleft clause. In the relation between the author of the

program and the end user the consideration is not necessary, if one accepts that it is governed by a non-contractual copyright license and therefore only Copyright rules are applicable.

The second issue, also related with the existence of a contract, refers to the enforceability requirements of shrinkwrap and clickwrap licenses. Although the analysis of the relevant law does not allow for drawing very precise rules, the following three elements have to be taken into account: making conspicuous a notice related to the applicable license before delivery of the program; delivery of the license itself; and licensee's manifestation of assent. Meeting all these requirements would guarantee the enforceability of the shrinkwrap or clickwrap license. Moreover, the Courts have shown some flexibility with a partial compliance with these requirements. In any event, if the shrinkwrap or clickwrap license is not enforceable, the author of the program will be still protected by Copyright law.

The third issue puts into question the capability of the author of the program to file suit against downstream licensees, noticing that they lack privity of contract. The proposition that the author is so incapable may have some merit, although the privity doctrine has been recently relaxed. In any event, the author would still be protected by contract law against his licensee and by Copyright law against any other sublicensee.

The fourth issue asks about the validity of the warranty disclaimer included in most copyleft licenses. The question must be answered by distinguishing between the position

of the redistributor who licenses a computer program for a fee and the author of a program who posts it in a website and does not charge anything in return. In the former case, the redistributor is a merchant under the UCC and an implied warranty crops up. However, this warranty may be disclaimed through a conspicuous notice. Therefore, if the shrinkwrap or clickwrap license is enforceable, it is most probable that the warranty disclaimer will displace the implied warranty. In opposition, the author of the program is generally not a merchant, and therefore the implied warranty does not emerge at the first place.

The fifth issue deals with the interrelation between Copyright and Contract law. The Court decisions show a clear trend to accept contract clauses which deviate from Copyright law provisions. The Courts only opt for preemption in rare cases. However, it must be underlined again that the lack of privity could impede the enforcement of the license against third parties. In this case, Copyright law would be applicable.

## Bibliography

- PATRICK S. ATIYA, *AN INTRODUCTION TO THE LAW OF CONTRACT* 265 (5<sup>th</sup> ed. 1995).
- Nikolai Bezroukov, *A Second Look at the Cathedral and the Bazaar*, First Monday (1999), available at [http://www.firstmonday.org/issues/issue4\\_12/bezroukov/](http://www.firstmonday.org/issues/issue4_12/bezroukov/)
- Jean Braucher, *The Failed Promise of the UCITA Mass-Market Concept and its Lessons for Policing of Standard Form Contracts*, 7 *Journal of Small and Emerging Business Law*, 393 (2003).
- Peter Brown, *SCO v. IBM: Other Legal Issues in the Open Source Community*, 808 *Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series* 103 (2004)
- Peter Brown, *Legal Issues in the Open Source Community*, 780 *Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series* 309 (2004)
- Brian W. Carver, *Share and Share Alike: Understanding and Enforcing Open Source and Free Software Licenses*, 20 *Berkeley Technology Law Journal* 443 (2005)
- Chris DiBona et al., *Introduction in OPEN SOURCES – VOICES FROM THE OPEN SOURCE REVOLUTION*, 1 (Chris DiBona et al. eds., 1999)
- E. ALLAN FARNSWORTH, *CONTRACTS* 3 (4<sup>th</sup> ed., Aspen Publishers 2004)
- Free Software Foundation, *Why We Must Fight UCITA*, available at <http://www.gnu.org/philosophy/ucita.html>.
- JESÚS GONZÁLEZ BARAHONA ET AL., *INTRODUCCIÓN AL SOFTWARE LIBRE* (2003), available at <http://curso-sobre.berlios.de/introsobre/libre.pdf>
- Andrés Guadamuz González, *Viral Contracts or Unenforceable Documents? Contractual Validity of Copyleft Licences*, *European Intellectual Property Review* 331 (2004).
- Lucie Guibault, *Pre-emption Issues in the Digital Environment: Can Copyright Limitations be Overridden by Contractual Agreements Under European Law?* 22 (1998), available at [www.ivir.nl/Publicaties/guibault/ARTICLE2.doc](http://www.ivir.nl/Publicaties/guibault/ARTICLE2.doc).
- Frank Hecker, *Setting Up Shop: The Business of Open-Source Software* (2000), available at <http://www.hecker.org/writings/setting-up-shop>.

- Thomas Hoeren, *Open Source und das Schenkungsrecht – eine durchdachte Liason?*, RECHT UND RISIKO, FESTSCHRIFT FÜR HELMUT KOLLHOSSER ZUM 70 GEBURTSTAG, 229, 237 (Reinhard Bork ed., 2004), available at [http://128.176.101.170/hoeren\\_veroeffentlichungen/opensourceunddasschenkungsrecht.pdf](http://128.176.101.170/hoeren_veroeffentlichungen/opensourceunddasschenkungsrecht.pdf).
- César Iglesias Rebollo, *Otra forma de distribuir software: las licencias de software libre o de código abierto*, 2 Revista General de Legislación y Jurisprudencia, 219 (2004)
- Terry J. Ilardi, *Mass Licensing – Part 2: Open Source Software Licensing*, 831 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 279 (2005).
- Terry J. Ilardi, *Mass Licensing – Part 1: Shrinkwraps, Clickwraps and Browsewraps*, 831 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 251 (2005).
- Phillip Johnson, *All Wrapped up? – A Review of the Enforceability of “Shrink-wrap” and “Click-wrap” Licenses in the United Kingdom and the United States*, 2 European Intellectual Property Review 98, 98 (2003).
- Pamela Jones, *The GPL Is a License, not a Contract* (2003), <http://lwn.net/Articles/61292>.
- Sandeep Krishnamurthy, *Cave or Community? An Empirical Examination of 100 Mature Open Source Projects*, First Monday (2002), available at [http://www.firstmonday.org/issues/issue7\\_6/krishnamurthy/](http://www.firstmonday.org/issues/issue7_6/krishnamurthy/).
- Mark A. Lemley, *Beyond Preemption: the Law and Policy of Intellectual Property Licensing*, 87 California Law Review 111, 136 (1999).
- Mark A. Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 Southern California Law Review, 1239, 1273 (1995).
- Mark A. Lemley, *Shrinkwraps in Cyberspace*, Jurimetrics Journal, 311 (1995).
- Jordi Mas I Hernández, *Software libre en el sector público* (2003), available at <http://www.uoc.edu/dt/20327/20327.pdf>
- Robert P. Merges, *The End of Friction? Property Rights and Contract in The “Newtonian” World of On-line Commerce*, 12 Berkeley Technology Law Journal, 115, 129 (1997).
- Axel Metzger & Till Jaeger, *Open Source Software und deutsches Urheberrecht*, 10 GRUR Int, 839 (1999)

- Eben Moglen, *Enforcing the GNU GPL* (2001), available at <http://www.gnu.org/philosophy/enforcing-gpl.html>.
- Stephen Mutkoski, *Open Source Software Issues in Acquisitions and Other Inbound Transactions*, 861 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 337, 342, 347 (2006)
- Christian H. Nandan, *Open Source Licensing: Virus or Virtue?*, 10 Texas Intellectual Property Law Journal, 349 (2002)
- Christian H. Nandan, *Software Licensing in the 21<sup>st</sup> Century: are Software “Licenses” really Sales, and How will the Software Industry respond?*, 32 American Intellectual Property Law Association Quarterly Journal, 555 (2004)
- Raymond Nimmer, *Legal Issues in Open Source and Free Software Distribution*, 861 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 7 (2006)
- Maureen A. O’Rourke, *Drawing the Boundary Between Copyright and Contract: Copyright Preemption of Software License Terms*, 45 Duke Law Journal 479 (1995).
- Maureen O’Sullivan, *Making Copyright Ambidextrous: An Expose of Copyleft*, Journal of Information, Law & Technology (2002), available at [http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2002\\_3/osullivan/](http://www2.warwick.ac.uk/fac/soc/law/elj/jilt/2002_3/osullivan/).
- Margaret Jane Radin, *Humans, Computers, and Binding Commitment*, Indiana Law Journal 1125 (2000)
- ERIC S. RAYMOND, *THE CATHEDRAL & THE BAZAAR – MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY* (revised 1<sup>st</sup> ed. 2001)
- Eric S. Raymond, *A Brief History of Hackerdom in OPEN SOURCES – VOICES FROM THE OPEN SOURCE REVOLUTION*, 19 (Chris DiBona et al. eds., 1999)
- LAWRENCE ROSEN, *OPEN SOURCE LICENSING – SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW* (2005)
- Diane Rowland & Andrew Campbell, *Supply of Software: Copyright and Contract Issues*, 10 International Journal of Law and Information Technology 23 (2002).
- Pamela Samuelson, *Copyright and Freedom of Expression in Historical Perspective*, 10 Journal of Intellectual Property Law 319 (2003)
- Pamela Samuelson et al., *Manifesto concerning the legal protection of Computer Programs*, 94 Columbia Law Review 2308 (1994)

- Michael D. Scott, *Protecting Software Transactions Online: the Use of “Clickwrap” Licenses*, 482 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 101 (1997)
- RICHARD STALLMAN, *FREE SOFTWARE, FREE SOCIETY: SELECTED ESSAYS OF RICHARD M. STALLMAN* (Joshua Gay ed., 2002)
- Richard Stallman, *The GNU Operating System and the Free Software Movement in OPEN SOURCES – VOICES FROM THE OPEN SOURCE REVOLUTION*, 53 (Chris DiBona et al. eds., 1999)
- Richard Stallman, *Categories of Free and Non-Free Software* (2005), available at <http://www.gnu.org/philosophy/categories.html>.
- Richard Stallman, *Why Software Should Be Free* (1992), available at <http://www.gnu.org/philosophy/shouldbefree.html>
- Richard Steuer, *Overview of the GNU System* (2006), available at <http://www.gnu.org/gnu/gnu-history.html>.
- LINUS TORVALDS, *JUST FOR FUN – THE STORY OF AN ACCIDENTAL REVOLUTIONARY* (2001)
- GUENTER TREITEL, *AN OUTLINE OF THE LAW OF CONTRACT* 246 (Oxford University Press 6<sup>th</sup> ed., 2004).
- GUENTER TREITEL, *THE LAW OF CONTRACT* (8<sup>th</sup> ed. 1991).
- David Turner, *Various Licenses and Comments about Them* (last updated 2006), <http://www.gnu.org/licenses/license-list.html>.
- MIKKO VÄLIMÄKI, *THE RISE OF OPEN SOURCE LICENSING – A CHALLENGE TO THE USE OF INTELLECTUAL PROPERTY IN THE SOFTWARE INDUSTRY* (2005), available at [http://pub.turre.com/openbook\\_valimaki.pdf](http://pub.turre.com/openbook_valimaki.pdf)
- Jason B. Wacha, *Open Source, Free Software, and the General Public License*, 20 No. 3 *Computer and Internet Lawyer*, 20 (2003)
- Jason B. Wacha, *Taking the Case: is the GPL Enforceable*, 21 *Santa Clara Computer and High Technology Law Journal*, 451 (2005)
- STEVEN WEBER, *THE SUCCESS OF OPEN SOURCE*, (2004)
- David A. Wheeler, *Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!* (revised as November 14, 2005), available at [http://www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)

– John C. Yates & Paul H. Arne, *Open Source Software Licenses: Perspectives of the End User and the Software Developer*, 823 Practising Law Institute/Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series 97 (2005)

– Miquel A. Zarza, *¿Qué es el Software Libre? Regulación y Autoorganización in INTERNET Y PLURALISMO JURÍDICO: FORMAS EMERGENTES DE REGULACIÓN* 297 (Pompeu Casanovas ed., Comares 2003)

– Jonathan Zittrain, *Normative Principles for Evaluating Free and Proprietary Software*, University of Chicago Law Review, Vol. 71, No. 1, 265 (2004) available at <http://ssrn.com/abstract=529862>

## Other materials

- Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPs) (1994)
- Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs (Official Journal L 122 , 17/05/1991 p. 42).
- Final Report of the National Commission on New Technological Uses of Copyrighted Works 11 (1978), available at <http://digital-law-online.info/CONTU/PDF/>.
- *FLOSS Final Report, Free/Libre Open Source Software: Survey and Study* 41 (2002), available at <http://www.infonomics.nl/FLOSS/report/>.
- Free Software Foundation, *GPLv3 First Discussion Draft Rationale* 12 (2006), available at <http://gplv3.fsf.org/rationale.pdf>.
- Paul Gustafson & William Koff (directors), CSC's Leading Edge Forum, Open Source: Open for Business, 60 (2004), available at [http://www.csc.com/features/2004/uploads/LEF\\_OPENSOURCE.pdf](http://www.csc.com/features/2004/uploads/LEF_OPENSOURCE.pdf).
- House Report 94-1476, available at <http://www.title17.com/contentLegMat/houseReport/chpt01/sec109.html>.
- Transcript of Opening session of first international GPLv3 conference, Boston, January 16<sup>th</sup>, 2006, available at <http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html>.
- Eric. S. Raymond, *The Halloween documents*, available at <http://www.catb.org/~esr/halloween/>
- Ted Shadler, *Your Open Source Strategy*, (2003), available at [http://www.redhat.com/whitepapers/forrester/Forrester\\_OSS\\_Sep.pdf](http://www.redhat.com/whitepapers/forrester/Forrester_OSS_Sep.pdf).
- Working Group on Libre Software, *Free Software/Open Source: Information Society Opportunities for Europe?* 5 (2000), available at <http://eu.conecta.it/paper.pdf>.