Licensing Implications of the Use of Open Source Software in Research Projects

Iryna Lishchuk
Institut für Rechtsinformatik
Leibniz Universität Hannover
Hannover, Germany
e-mail: lishchuk@iri.uni-hannover.de

***Abstract*—As more and more areas of science make use of the open source software (OSS), legal research in the field seeks to reconcile various open source licenses (which may be used in a single research project) and explores solutions to allow exploitation of project components in a license compliant way. Innovative software solutions contribute to the field of computer science from the technical side, while exploration of the legal implications of open source licensing enriches the topic from the legal perspective. In this paper, we consider what uses of what OSS may have licensing implications and suggest some solutions on how software developments may be used and distributed in a license compliant way.**

Keywords-open source software; free software; open source licensing; copyleft.

I. INTRODUCTION

Some key areas of computing, such as Linux/GNU, Google/Android, rely on open source software. Many research projects use the potential of OSS and contribute to the open source movement as well. One example is the EU FP7 CHIC project in the health informatics (full title "Computational Horizons In Cancer (CHIC): Developing Meta- and Hyper-Multiscale Models and Repositories for In Silico Oncology" [1]). CHIC is engaged in "*the development of clinical trial driven tools, services and infrastructures that will support the creation of multiscale cancer hypermodels (integrative models)"* [1]. In the course of this, it makes use of OSS and explores the possibility of open sourcing the project outcomes itself. For example, the hypermodelling framework VPH-HF relies on an open source domain-independent workflow management system Taverna [2], while an open source finite element solver, FEBio, is used in biomechanical and diffusion modeling [3].

This is part of a wider trend, in which OSS is becoming increasingly popular in all areas of scientific research. However, while the use of OSS may benefit the conduct of the project and promote its outcomes, it may later also have the effect of limiting the project exploitation options.

In this paper, we look into the licensing implications associated with the use of OSS and open sourcing the project outcomes. Also, we seek to suggest solutions on how licensing implications (and incompatibility risks) may best be managed. The rest of this paper is organized as follows. Section II describes the notion of free and open source software (FOSS) and elaborates on the license requirements for software distribution. Section III addresses peculiarities of the set of General Public Licenses (GPL) and points up some specific aspects stemming from the use of GPL software. In Section IV, the article concludes by way of a case study showing how the use of OSS may impact on future licensing of a project component.

II. FREE AND OPEN SOURCE SOFTWARE

Open source software is not simply a popular term, but it has its own definition and criteria, which we describe below.

A. Open Source Software

According to the Open Source Initiative (OSI), "*Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria…"* [4]. These requirements normally determine how the program may be distributed either in its source code (a script in a human readable form, usually written in one or another programming language, such as C^{++}, Java, Python, etc.) or as a compiled executable, i.e., object code ("*a binary code, simply a concatenation of "0"'s and "1"'s*." [5]).

The basic requirements of open source are as follows:

1. Free Redistribution. The license may not restrict distributing a program as part of an aggregate software distribution and/or may not require license fees.

2. Source Code. The license must allow distribution of the program both in source code and in compiled form. By distribution in object code, the source code should also be accessible at a charge not higher than the cost of copying (download from Internet at no charge).

3. Derived Works. The license must allow modifications and creation of derivative works and distribution of such works under the same license terms.

4. Integrity of The Author's Source Code. The license may require derivative works to be identified from original, such as by a version number or by name.

5. No Discrimination Against Persons or Groups.

6. No Discrimination Against Fields of Endeavor.

7. Distribution of License. The license terms apply to all users without the need of concluding a separate license agreement with every user.

8. License Must Not Be Specific to a Product. The license may not be dependent on any software distribution.

 ISBN: 978-1-61208-478-7

9. License Must Not Restrict Other Software. The license must not place restrictions on software distributed with the program (e.g., on the same medium).

10. License Must Be Technology-Neutral. The license may not be pre-defined for a specific technology [4].

There are currently more than 70 open source licenses, which can be categorized according to the license terms.

B. Free Software

One category is free software, which also has its own criteria. As defined by the Free Software Foundation (FSF), a program is free software, if the user (referred to as "you") has the four essential freedoms:

1. "The freedom to run the program as you wish, for any purpose (freedom 0).

2. The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.

3. The freedom to redistribute copies so you can help your neighbor (freedom 2).

4. The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this." [6].

The GPL, in its different versions, is a true carrier of these freedoms and GPL software (when distributed in a GPL compliant way) is normally free. The licenses, which qualify as free software licenses are defined by the FSF [7].

C. Free Software and Copyleft

The mission of free software (providing the users with these essential freedoms) is achieved in a way that not only the original author, who licenses his program under a free license, but also the subsequent developers, who make modifications to such free program, release their modified versions in the same "free" way.

Maintaining and passing these freedoms for subsequent software distributions is usually achieved by the so called copyleft principle. "*Copyleft is a general method for making a program (or other work) free, and requiring all modified and extended versions of the program to be free as well."* [8]. A copyleft license usually requires that modified versions be distributed under the same terms. This distinguishes copyleft from non-copyleft licenses: copyleft licenses pass identical license terms on to derivative works, while non-copyleft licenses govern the original code only.

However, a free license does not necessarily involve copyleft and a copyleft license is not always free. On the other hand, a license that "*requires modified versions to be nonfree does not qualify as a free license"* [6].

D. Licensing Implications on Software Distribution

From the whole spectrum of FOSS licenses, mostly the free licenses with copyleft produce licensing implications on software exploitation. Some other free licenses without copyleft are, in contrast, rather flexible, provide for a wider variety of exploitation options, subject to rather simple terms: acknowledgement of the original developer and replication of a license notice and disclaimer of warranties.

Such more relaxed non-copyleft licenses usually allow the code to be run, modified, distributed as standalone and/or as part of another software, either in source form and/or as a binary executable, provided the license terms for the original code are met. Among the popular non-copyleft licenses are: the Apache License [9], the MIT License [10], the BSD 3-Clause License [11], to name a few. "*Code, created under these licenses, or derived from such code, may "go "closed" and developments can be made under that proprietary license, which are lost to the open source community.*" [12].

As a condition for distributing the MIT or BSD licensed code (or its modified versions), these licenses require that the use of the original code should be acknowledged. For this, the developers of the original program and the program license with disclaimer should be replicated (maintained) throughout the whole re-distribution chain. For instance, the MIT license requires that "*copyright notice and this permission notice shall be included in all copies or substantial portions of the Software*" [10]. Failure to do so may, at one hand, compromise the ability of the developer to enforce his own copyright in parts of the code, which he wrote himself, and, on the other hand, put him at risk of being found liable for copyright infringement, because distribution of the program in breach of the license terms may be a ground for claiming copyright violation [12]. Once these requirements of notice preservation are met, a developer may exploit the software as he deems fit.

E. Copyleft Licenses

In contrast, the free licenses with copyleft by promoting the four essential freedoms to the users may at the same time take away the developer´s freedom to decide on licensing of his own software, pre-determining a license choice for him. While supporters of free software speak about copyleft as protecting the rights, some developers, affected by the copyleft against their will, tend to refer "*to the risk of "viral" license terms that reach out to infect their own, separately developed software and of improper market leverage and misuse of copyright to control the works of other people*." [13].

The GPL Version 2 (GPL v2) [14] and Version 3 (GPL v3) [15] are examples of free licenses with copyleft. GPL copyleft looks as follows. GPL v2, in Section 1, allows "*to copy and distribute verbatim copies of the Program's source code… in any medium"* under the terms of GPL, requiring replication of the copyright and license notice with disclaimer and supply of the license text. In Section 2, the GPL license allows modifying the program, "*thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above*", i.e., under GPL itself. In doing so, it implies that a developer may distribute his own developments, only if he licenses under GPL. In some cases, it may put a developer up to a dilemma: either to license under GPL or not to license at all.

A more positive aspect of GPL is that at times it may be rather flexible. In particular, not all modes of using a GPL program create a modified version and not all models of software distribution are necessarily affected by GPL.

III. GPL AND GPL COPYLEFT

Among the decisive factors whether software is affected by GPL copyleft are: the mode, in which software uses a GPL program, the version and wording of the applicable GPL license, and the method of how software will be distributed.

A. Mode of Use

The mode of use essentially determines whether a development qualifies as "*a work based on a GPL program*" or not. If because of using a GPL program, software qualifies as a *"work based on the Program",* then according to the terms of GPL it shall go under GPL [14]. Otherwise, if a program is not a modified version of GPL, then there is no binding reason for it to go under GPL.

In this regard, not all uses of a GPL program will automatically produce a derivative work. For example, developing a software using the Linux operating system, or creating a piece of software designed to run on Java or Linux (licensed under GPL v2 [16]) does not affect licensing of this software (unless it is intended to be included into the Linux distribution as a Linux kernel module). Also, calculating algorithms by means of a GPL licensed R (a free software environment for statistical computing and graphics [17]) in the course of developing a software model does not affect licensing of a model, because the model is not running against the GPL code.

Another distinctive feature of GPL is that, in contrast to the majority of other open source licenses, which do not regard linking as creating a modified version (e.g., Mozilla Public License [18], Apache License [9]), the GPL license considers linking, both static and dynamic, as making a derivative work. Following the FSF interpretation criteria, "*Linking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination*" [19]. This position may be tested against the technical and legal background involved [20].

The controversy Android v Linux [21] illustrates how Google avoided licensing of Android under GPL because the mode how it used Linux was beyond the scope of applicability of Linux GPL license. This case concerned the Android operating system, which relies on the GPL licensed Linux kernel and which was ultimately licensed under the Apache License. Android is an operating system, primarily used by mobile phones. It was developed by Google and consists of Linux kernel, some non-free libraries, a Java platform and some applications. Despite the fact that Android uses Linux kernel, licensed under GPL v2 [16], Android itself was licensed under Apache 2.0 License. "*To combine Linux with code under the Apache 2.0 license would be copyright infringement, since GPL version 2 and Apache 2.0 are incompatible"* [21]. However, the fact that the Linux kernel remains a separate program within Android, with its source code under GPL v2, and the Android programs communicate with the kernel via system calls clarified the licensing issue. Software communicating with Linux via system calls is expressly removed from the scope of derivative works, affected by GPL copyleft. A note, added to the GPL license terms of Linux by Linus Torvalds, makes this explicit:

*"NOTE! This copyright does *not* cover user programs that use kernel services by normal system calls - this is merely considered normal use of the kernel, and does *not* fall under the heading of "derived work". Also note that the GPL below is copyrighted by the Free Software Foundation, but the instance of code that it refers to (the linux kernel) is copyrighted by me and others who actually wrote it."* [16].

Examples of normal system calls are: *fork(), exec(), wait(), open(), socket(),* etc. [21]. Such system calls operate within the kernel space and interact with the user programs in the user space [22]. Taking into consideration these technical details, "*Google has complied with the requirements of the GNU General Public License for Linux, but the Apache license on the rest of Android does not require source release."* [21]. In fact, the source code for Android was ultimately released, however, in the view of the FSF, even the use of Linux kernel and release of the source code do not make Android free software. As explained by Richard Stallman [21], the aspects that Android comes up with some non-free libraries, proprietary Google applications, proprietary firmware and drivers, prevents the users from installing and running their own modified software, accepting versions approved by some company, and – what is most interesting – that the Android code is insufficient to run the device undermine the philosophy of free software [21].

B. GPL Weak Copyleft and Linking Exceptions

Another factor that matters whether a development is subject to GPL copyleft is the GPL license used.

Some GPL licenses have so-called weak copyleft. Examples are the GNU Library or "Lesser" General Public License, Version 2.1 (LGPL-2.1) [23] and Version 3.0 (LGPL-3.0) [24]. In these cases, a program, which merely links to a LGPL program or library (without modifying it), does not have to be licensed under LGPL. As LGPL-2.1 explains, "*A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this Licens*e." [23]. LGPL allows combining external programs with a LGPL library and distributing combined works under the terms at the choice of the developer, provided: (a) the library stays under

LGPL; and (b) license of the combined work allows "*modification of the work for the customer's own use and reverse engineering for debugging such modifications*" [23].

Some practical consequences of how a switch from LGPL to GPL in one software product may affect exploitation and usability of another software product are demonstrated by the controversy: MySQL v PHP [20].

PHP is a popular general-purpose scripting language that is especially suited to web development [25]. PHP was developed by the Zend company and licensed under the PHP license, which is not compatible with GPL [26]. PHP is widely used and distributed with MySQL in web applications, such as in the LAMP system (standing for: Linux, Apache, MySQL and PHP), which is used for building dynamic web sites and web applications [27]. MySQL is the world's most popular open source database, originally developed by MySQL AB, then acquired by Sun Microsystems in 2008, and finally by Oracle in 2010 [28]. In 2004, MySQL AB decided to switch the MySQL libraries from LGPL to GPL v2. That is when the controversy arose. The PHP developers responded with disabling an extension in PHP 5 to MySQL. If PHP was thus not able to operate with MySQL, the result would be negative for the open source community [20], which widely relied on PHP for building web applications with MySQL. To resolve the conflict, MySQL AB came up with a FOSS license exception. The FOSS license exception (initially called the FLOSS License Exception) allowed developers of FOSS applications to include MySQL Client Libraries (also referred to as "MySQL Drivers" or "MySQL Connectors") within their FOSS applications and distribute such applications together with GPL licensed MySQL Drivers under the terms of a FOSS license, even if such other FOSS license were incompatible with the GPL [29].

A similar exception may be found in relation to the programming language Java. Java is licensed under GPL v2 with Classpath Exception [30]. It is a classic GPL linking exception based on permission of the copyright holder. It consists of the following statement attached to the Java GPL license text: "*As a special exception, the copyright holders of this library give you permission to link this library with independent modules to produce an executable, regardless of the license terms of these independent modules, and to copy and distribute the resulting executable under terms of your choice, provided that you also meet, for each linked independent module, the terms and conditions of the license of that module. An independent module is a module which is not derived from or based on this library."* [30]. Originally, this allowed free software implementations of the standard class library for the Java programming language [20].

Adding special permissions or exceptions to the standard terms of GPL is explicitly permitted by GPL v3. This makes GPL v3 more flexible and license compatible in comparison to GPL v2. "*Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions."* [15]. The linking exception to GPL v3, as recommended by the FSF, appears as follows: "*If you modify this Program, or any covered work, by linking or combining it with [name of library] (or a modified version of that library), containing parts covered by the terms of [name of library's license], the licensors of this Program grant you additional permission to convey the resulting work"* [31].

In this respect, it must be noted that adding additional permissions or exceptions to GPL license terms is an exclusive prerogative of the copyright holder. Thus, if a developer builds his program on top of a third party GPL code, he may not add such a linking exception to the GPL license of the whole code, unless he obtained consent to this from all the other copyright holders [31].

A software developer may be motivated to add such linking exceptions to solve GPL-incompatibility issues, which may arise if a GPL program is supposed to run against GPL incompatible programs or libraries, or to allow use of GPL software in software developments, which are not necessarily licensed in a GPL compatible way.

C. Mode of Distribution

Thirdly, the mode of distribution, namely: whether a component is distributed packaged with a GPL dependency or without it, may matter for the application of GPL.

According to the first criterion of OSS, which says that a license must permit distribution of a program either as standalone or as part of "*an aggregate software distribution containing programs from several different sources*" [4], the GPL license allows distributing GPL software "*as a component of an aggregate software"*. As interpreted by the FSF, "*mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License"* [32]. Such an "aggregate" may be composed of a number of separate programs, placed and distributed together on the same medium, e.g., USB. [32].

The core legal issue here is of differentiating an "aggregate" from other "modified versions" based on GPL software. "*Where's the line between two separate programs, and one program with two parts? This is a legal question, which ultimately judges will decide."* [32]. In the view of the FSF, the deciding factor is the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are exchanged). So, including the modules into one executable file or running modules *"linked together in a shared address space"* would most likely mean "*combining them into one program*". By contrast, when *"pipes, sockets and command-line arguments"* are used for communication, "*the modules normally are separate programs"* [32].

These observations bring us to the following conclusions. Distributing an independent program together with a GPL program on one medium, so that the programs

do not communicate with each other, does not spread the GPL of one program to the other programs. Equally, distributing a program, which has a GPL dependency, separately and instructing the user to download that GPL dependency for himself would release a program from being licensed by GPL. However, distributing a program packaged with a GPL dependency would require licensing the whole software package under GPL, unless exceptions apply.

D. Commercial Distribution

In contrast to the open source licenses, which allow the code to go "closed" in proprietary software "*lost to the open source community*" [12], GPL is aimed to preserve software developments open for the development community. For this reason, GPL does not allow "burying" GPL code in proprietary software products. Against this principle, licensing GPL software in proprietary way and charging royalties is not admissible.

One of the exploitation options for GPL components might be charging fees for distribution of copies, running from the network server as "Software as a Service" or providing a warranty for a fee. For instance, when a GPL program is distributed from the site, fees for distributing copies can be charged. However, *"the fee to download source may not be greater than the fee to download the binary*" [33].

Offering warranty protection and additional liabilities would be another exploitation option. In this regard, GPL allows providing warranties, but requires that provision of warranties must be evidenced in writing, i.e., by signing an agreement. A negative aspect here is that by providing warranties a developer accepts additional liability for the bugs, caused by his predecessors, and assumes "*the cost of all necessary servicing, repair and correction*" [15] for the whole program, including modules provided by other developers. The business model of servicing GPL software has proven to be quite successful, as the Ubuntu [34] and other similar projects, which distribute and provide services for Linux/GNU software, demonstrate.

IV. CONCLUSIONS

In this paper, we have considered some licensing implications, which may arise by the use of open source software. We conclude by way of a case study, showing how the use of OSS may affect licensing of a project component.

In this example, let us consider licensing of a repository for computational models. The repository links, by calling the object code, to the database architecture MySQL, licensed under GPL v2 [35], and a web application Django, licensed under BSD 3-Clause License [36].

We may identify the future (downstream) licensing options for the repository in the following way. GPL v2 considers, "*linking a GPL covered work statically or dynamically with other modules making a combined work based on the GPL covered work. Thus, GNU GPL will cover the whole combination"* [19]. In terms of GPL, a repository, which links to GPL MySQL, qualifies as a work based on a GPL program and must go under GPL. BSD 3-Clause License is a lax software license, compatible with GPL [7]. GPL permits BSD programs in GPL software. Hence, no incompatibility issues with the BSD licensed Django arise. Section 9 GPL v2, applicable to MySQL, allows a work to be licensed under GPL v2 or any later version. This means, a repository, as a work based on GPL v2 MySQL, may go under GPL v3. Hence, GPL v3 has been identified as a license for this repository. The license requirements for distribution are considered next.

A repository may be distributed in source code and/or in object code. Distribution in object code must be supported by either: (a) source code; (b) an offer to provide source code (valid for 3 years); (c) an offer to access source code free of charge; or (d) by peer-to-peer transmission – information where to obtain the source code. If the repository is provided as "Software as a service", so that the users can interact with it via a network without having a possibility to download the code, release of the source code is not required.

In distributing this repository under GPL v3, the developer must include into each source file, or (in case of distribution in an object code) attach to each copy: a copyright notice, a GPL v3 license notice with the disclaimer of warranty and include the GPL v3 license text. If the repository has interactive user interfaces, each must display a copyright and license notice, disclaimer of warranty and instructions on how to view the license.

Django and MySQL, as incorporated into software distribution, remain under BSD and GPL v2, respectively. Here the BSD and GPL v2 license terms for distribution must be observed. It means, all copyright and license notices in the Django and MySQL code files must be reserved. For Django, a copyright notice, the license notice and disclaimer shall be retained in the source files or reproduced, if Django is re-distributed in object code [11]. Distribution of MySQL should be accompanied by a copyright notice, license notices and disclaimer of warranty; recipients should receive a copy of the GPL v2 license. For MySQL, distributed in object code, the source code should be accessible, either directly, or through instructions on how to get it.

As this case study suggests, the use of open source software under copyleft licenses, such as GPL, may be a preferential option for keeping the project components open for development community. On the other hand, if commercial exploitation is intended, the use of open source software under Apache License or MIT or BSD would most likely suit these interests better.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement No 600841. The

author thanks for the helpful comments on this work Prof. Dr. Nikolaus Forgó and Dr. Marc Stauch, Institut für Rechtsinformatik, Leibniz Universität Hannover, Hannover, Germany. Appreciation is also credited to Luis Enriquez A. and his Master Thesis "Dynamic Linked Libraries: Paradigms of the GPL license in contemporary software", which was used in doing this research.

REFERENCES

[1] CHIC, Project, <http://chic-vph.eu/project/> [retrieved: 5 April, 2016].

[2] D. Tartarini, et al, "The VPH Hypermodelling Framework for Cancer Multiscale Models in the Clinical Practice", In G. Stamatakos and D. Dionysiou (Eds): Proc. 2014 6th Int. Adv. Res. Workshop on In Silico Oncology and Cancer Investigation – The CHIC Project Workshop (IARWISOCI), Athens, Greece, Nov.3-4, 2014 (www.6thiarwisoci.iccs.ntua.gr), pp.61-64. (open-access version), ISBN: 978-618-80348-1-5.

[3] F. Rikhtegar, E. Kolokotroni, G.Stamatakos, and P. Büchler, "A Model of Tumor Growth Coupling a Cellular Biomodel with Biomechanical Simulations", In G. Stamatakos and D. Dionysiou (Eds): Proc. 2014 6th Int. Adv. Res. Workshop on In Silico Oncology and Cancer Investigation – The CHIC Project Workshop (IARWISOCI), Athens, Greece, Nov.3-4, 2014 (www.6thiarwisoci.iccs.ntua.gr), pp.43-46. (open-access version), ISBN: 978-618-80348-1-5, pp.43.

[4] Open Source Initiative, Open Source Definition, <http://opensource.org/osd> [retrieved: 6 April, 2016].

[5] Whelan Associates Inc. v. Jaslow Dental Laboratory, Inc., et al, U.S. Court of Appeals, Third Circuit, August 4, 1986, 797 F.2d 1222, 230 USPQ 481.

[6] GNU Operating System, The Free Software Definition, <http://www.gnu.org/philosophy/free-sw.en.html> [retrieved: 6 April, 2016].

[7] GNU Operating System, Various Licenses and Comments about Them, <http://www.gnu.org/licenses/license-list.en.html> [retrieved: 6 April, 2016].

[8] GNU Operating System, What is Copyleft?, <http://www.gnu.org/licenses/copyleft.en.html> [retrieved: 5 April, 2016].

[9] OSI, Licenses by Name, Apache License, Version 2.0, <http://opensource.org/licenses/Apache-2.0> [retrieved: 6 April, 2016].

[10] OSI, Licenses by Name, The MIT License (MIT), <http://opensource.org/licenses/MIT> [retrieved: 5 April, 2016].

[11] OSI, Licenses by Name, The BSD 3-Clause License, <http://opensource.org/licenses/BSD-3-Clause> [retrieved: 6 April, 2016].

[12] A. M. St. Laurent, "Understanding Open Source and Free Software Licensing", O´Reilly, 1 Edition, 2004.

[13] R. T. Nimmer, "Legal Issues in Open Source and Free Software Distribution", adapted from Chapter 11 in Raymond T. Nimmer, The Law of Computer Technology, 1997, 2005 Supp.

[14] GNU General Public License, Version 2 (GPL-2.0), <http://opensource.org/licenses/GPL-2.0> [retrieved: 7 April, 2016].

[15] GNU General Public License, Version 3 (GPL-3.0), <http://opensource.org/licenses/GPL-3.0> [retrieved: 6 April, 2016].

[16] The Linux Kernel Archives, <https://www.kernel.org/pub/linux/kernel/COPYING> [retrieved: 6 April, 2016].

[17] The R Project for Statistical Computing, R Licenses, <https://www.r-project.org/Licenses/> [retrieved: 6 April, 2016].

[18] Mozilla, MPL 2.0 FAQ, <https://www.mozilla.org/en-US/MPL/2.0/FAQ/> [retrieved: 6 April, 2016].

[19] GNU Operating System, Frequently Asked Questions about the GNU Licenses, <http://www.gnu.org/licenses/gpl-faq#GPLStaticVsDynamic> [retrieved: 6 April, 2016].

[20] L. Enriquez, "Dynamic Linked Libraries": Paradigms of the GPL license in contemporary software", EULISP Master Thesis, 2013.

[21] R. Stallman, "Android and Users' Freedom", first published in The Guardian, <http://www.gnu.org/philosophy/android-and-users-freedom.en.html> [retrieved: 6 April 2016].

[22] Hartman Greg Kroat, Linux kernel in a nutshell, O'Reilly, United States, 2007.

[23] Open Source Initiative, Licenses by Name, The GNU Lesser General Public License, version 2.1 (LGPL-2.1), <http://opensource.org/licenses/LGPL-2.1> [retrieved: 6 April, 2016].

[24] Open Source Initiative, Licenses by Name, The GNU Lesser General Public License, version 3.0 (LGPL-3.0), <http://opensource.org/licenses/LGPL-3.0> [retrieved: 6 April, 2016].

[25] The PHP Group, <http://php.net/> [retrieved: 06 April, 2016].

[26] OSI, Licenses by Name, The PHP License 3.0 (PHP-3.0), <https://opensource.org/licenses/PHP-3.0> [retrieved: 7 April, 2016].

[27] Building a LAMP Server, <http://www.lamphowto.com/> [retrieved: 7 April, 2016].

[28] Oracle, Products and Services, MySQL, Overview, <http://www.oracle.com/us/products/mysql/overview/index.html> [retrieved: 7 April, 2016].

[29] MySQL, FOSS License Exception, <https://www.mysql.de/about/legal/licensing/foss-exception//> [retrieved: 7 April, 2016].

[30] GNU Operating System, GNU Classpath, <http://www.gnu.org/software/classpath/license.html> [retrieved: 7 April, 2016].

[31] GNU Operating System, Frequently Asked Questions about the GNU Licenses, <http://www.gnu.org/licenses/gpl-faq#GPLIncompatibleLibs> [retrieved: 6 April, 2016].

[32] GNU Operating System, Frequently Asked Questions about the GNU Licenses, <http://www.gnu.org/licenses/gpl-faq#MereAggregation> [retrieved: 6 April, 2016].

[33] FSF, Frequently Asked Questions about the GNU Licenses, <https://www.gnu.org/licenses/gpl-faq.html#DoesTheGPLAllowDownloadFee> [retrieved: 6 April, 2016].

[34] Ubuntu, <http://www.ubuntu.com/> [retrieved: 6 April, 2016].

[35] MySQL, MySQL Workbench, <http://www.mysql.com/products/workbench/> [retrieved: 6 April, 2016].

[36] Django, Documentation, <https://docs.djangoproject.com/en/1.9/> [retrieved: 6 April, 2016].