Licensing schemes in the production and distribution of Open Source software.

An empirical investigation

Andrea Bonaccorsi*, Cristina Rossi

Sant'Anna School of Advanced Studies

Institute for Informatics and Telematics (IIT)

Pisa, Italy

1. Introduction

A well-known definition accounts for Open Source software: a program is Open Source if its source code is publicly available. Although this characterization encompasses Open Source software, in the following we will clarify that it has a wider scope and fails to define the boundaries of the phenomenon.

The best categorisation makes reference to the Open Source Definition (OSD) established by a group of pragmatic Open Source evangelists (Lee, 1999). In the late 1990s they were wondering how to push forward the diffusion of the new paradigm. They aimed in particular at fostering its exploitation by business firms.

In order to achieve their goals, in 1998 they established a foundation, the Open Source Initiative (OSI). The OSI elaborated the Open Source Definition that marshals the guidelines to define exactly whether a program is Open Source or not.

The preamble of the Open Source Definition - Version 1.0[1] explicitly states that *Open source doesn't just mean access to the source code*. Further conditions need to be met. A software is considered Open Source if and only if distribution terms [i.e. the license] comply with a given

---

* Corresponding author. Address: P.zza Martiri della Libertà 33, 56127, Pisa; Tel.: +39 050 883323; Fax: +39 050 883344. *E-mail address:* bonaccorsi@sssup.it

set of well-defined criteria. *To say that a code is Open Source is to say that it is subject to a member of a particular category of licenses* (McGowan, 2000).

This is an objective criterion. The Open Source Definition is a well-formed document that allows to determine without any doubt whether a license complies or not with its provisions. It does not mandate for a closed set of licenses. Everyone is permitted to draw his own distribution scheme as long as it complies with the criteria of the OSD. The licenses that comply with the OSD criteria are called OSI approved. At present there are over 40 Open Source license schemes and a painstaking taxonomy would be required in order to account for the great variety of their provisions.

Many studies dealing with Open Source licenses have focused on the description and classification of their features (Fink, 2002; Kennedy, 2001; Oksanen and Valimaki, 2002; Pearson, 2000), while little has been done in order to understand why the agents choose a particular software distribution scheme (Lerner and Tirole, 2002a). This analysis cannot ignore the differences in motivations between individuals and firms. In particular, the distribution terms adopted by firms whose business models are based on the supply of Open Source services and products are still unexplored.

In fact, the incentives that lead the firms to be involved in the Open Source movement are very different from the motivations of individual developers. Firms aim at profiting from their participation. In turn individual developers are acting out of a wide range of extrinsic and intrinsic motivations that include reputation gain, talent signalling and learning new programming skills (Lerner and Tirole, 2002a, Ghosh, 2003). As a consequence the determinants of license choice are different in the two groups.

---

[1] The Open Source Initiative updated the OSD several times, however it is a matter of minor revisions that do not alter the essence of the document. Version 1.0 is no longer at the Open Source Initiative (OSI) Web site. http://www.opensource.org/osd.html.

The preferences of individual developers for the GNU General Public License (GPL)[2] have been explained on a theoretical basis as a safeguard against private appropriation of collective efforts and a mechanism for lowering barriers to entry and maximising the number of participants (Kasper, 2001). Several analyses focus on the strategies followed by large software houses that release the source code of their programs to the community (Cusumano and Yoffie, 1998; Hecker, 2000; Madanmohan and Nilendu, 2001). O'Mahony (2003) collected data on six large Open Source projects and studied the strategies that contributors to *community managed software projects* adopt *to protect their work.* She concludes that the license choice plays a leading role within this perspective.

Valimaki and Oksanen (2002) evaluate frequently used Open Source licenses and licensing models as convenient from the perspective of a company developing mass-market software products, in order to ensue compatibility. Mustonen (2002) proposes a model where a monopoly firm that sells a copyrighted program has an incentive to support the development of free copyleft substitute programs.

As many others in the Open Source field, the issue at hand suffers from lack of empirical data.

Although in the literature there are empirical studies that explore the relationships between license choice and project characteristics (Lerner and Tirole, 2002a), at present we are not aware of surveys that collect data on licensors, that is on firms producing and distributing software on an Open Source basis.

This study addresses his shortcoming. We examine the license choice of the firms that supply Open Source products and services and relate it to their structural characteristics, business models and attitudes towards the movement and its community.

---

[2] Richard Stallman, the founding father of the Free Software Movement, elaborated the GNU General Public Licence in 1984 in order *to guarantee everybody's freedom to share and change* software making sure that *the software is free for all its users* (GNU General Public License Preamble).
The Free Software movement can be regarded as the precursor of the Open Source movement. Although they differ in some respects, they rest on the same philosophical bases, so that the terms *Open Source* and *Free Software* are used as synonymous (Stallman, 1984b, 1984c). The origins and aims of the Free

The paper is organised as follows.

Section II surveys the literature on Open Source licenses. We go into details of the provisions of the Open Source Definition and analyse how they account for the survival and diffusion of this software production mode.

Given that the freedom of modifying the software plays a leading role within the Open Source movement, we go deeper into the rules that regulate *derivative works*. The distinction between *copyleft* and *non-copyleft* licenses is addressed.

Section III reports about a test of several theoretical hypotheses advanced by the literature on Open Source licenses. Our empirical investigation deals with licenses used by firms that do business with Open Source software and makes use of the data collected by a survey on 146 Italian firms in carried out in 2003. The sample and the variables of the survey are described. Section IV summarizes the main conclusions of the paper.

2. *Open Source licences and the viability of the Open Source production mode. A survey of the literature*

An analysis of the Open Source licenses must not leave aside their goals, first of all *preventing that anybody is locked out of the process* (Rothfuss, 2002). According to McGowan, (2000) Open Source licenses *create a social space devoted to this kind of [software] development*. They are a governing mechanism that enforces the non-written norms of the Open Source community (Camp and Syme, 2000), provides incentives for programmers (Kasper 2001; Lerner and Tirole, 2002b) and distinguishes the Open Source community from the world of the proprietary software. The social identification with the Open Source community (Saers, 2002) plays a leading role in determining participation in the movement (Hertel et al., 2003).

Software movement are beyond the scope of this paper. For an overview of this issue see Williams, 2002. An essay on the relationships between the two movements was written by Moglen (2000).

Contrary to what most people assume, the Open Source software production mode *rests on property rights*[3]. As it has been observed, (McGowan, 2001), intellectual property rights are *an unnoticed force behind* Open Source software (Gomulkiewicz, 1999). The new paradigm, in fact, *depends upon copyright protection for its continued existence* (Bobko, 2001) and *works within the existing copyright doctrine to make code publicly available* (Heffan, 1997). The Open Source phenomenon was born out of a legal innovation. Instead of establishing intellectual property rights in order to exclude others from the use of protected software, the founding father of the movement exploited the copyright law to *encourage free use* (Horne, 2001; Maher 2000). In order to understand how this is made effective we have to refer to the United States law. The Open Source phenomenon, in fact, was born within this legal framework.

United State law provides the developers with three means of protecting themselves against unauthorized copying and/or use of their programs: *trade secret, patents and copyright* (Oz, 1998). Each of them provides for different aspects of the software product.

Copyright protects *the expression of the ideas*. Copyright law was developed long before the advent of computer and programs for protecting literary and artistic works[4]. In 1980, the extension of the copyright protection to computer programs was made explicit by amendments to the Copyright Act[5].

The peculiar use of the copyright made by the Open Source community relies on its very structure. Copyright is a *bundle of five rights* that *are separable according to the choice of the owner* (Lee, 1999, Rosenberg, 2002). The copyright holder is entitled to reproduce, modify, distribute, publicly perform and publicly display[6] the original copyrighted material. The Open Source licenses unbundle these rights and establish which of them are granted to the licensees (Monti, 2002).

---

[3] The Open Source Initiative registered the trademark OSI. Software distributed under a license approved by the Open Source Initiative is *OSI certified* (Raymond, 2000).
[4] Copyright Act 1909.
[5] The 1980 Amendments to the Copyright Act extended this law to apply to computer programs. The new act even provided a definition of *computer program* as *a set of statements or instructions to be used directly or indirectly in a computer to bring about a certain result.*

On a practical level Open Source licenses serve to weaken intellectual property rights (Lee, 1999). It is then of great interest to understand why a system grounds itself on intellectual property rights in order to weaken them.

According to McJohn (2000), agents that take part in the Open Source movement think of intellectual property rights as *a necessary evil* (Lee, 2003) that allows the new software production mode to survive by *keeping the code open.* Because of the copyright law the author of a computer program has the full power to legally deny its improper uses and forbid to turn his code into proprietary. Placing the code into the public domain does not leave this chance.

Several scholars (Gomulkiewicz, 1998; Kennedy, 2001; Lee, 1999; Lee, 2003; McGowan, 2001) examine the differences between Open Source and public domain software.

An intellectual work is into the public domain if *it is not copyrighted* (Anez, 1999; Boyle, 2001; Bassi, 2000; 1984a). Public domain undoubtedly plays a leading role *for the progress of science and technology* (Bollier, 2002) but it does not meet the need for legal protection of the Open Source movement. When the source code of a program is into the public domain, everyone is entitled to make whatever use of it. As a consequence everyone is permitted to hijack the code and turn it into proprietary.

This undermines the very essence of the movement: i.e. the free circulation of ideas (Sloan, 2002), and shatters the incentives of developers to take part in it. Individuals succeed in gaining a reputation as valuable programmers and signalling their talent to software houses (Lerner and Tirole, 2002b) only if the outcome of their efforts, the code, is under everybody's eyes. Moreover if the Open Source model called for public domain code, the fear of code hijacking would hang over the firms and irremediably keep them off from this new production mode. At any time, in fact, competitors might take their code away, make no change, release it under a proprietary license scheme and profit from their R&D activity without any effort.

Scholars define Open Source software as a *privately produced public good* (Bessen, 2001; Kollack, 1998; Johnson, 2001; Von Hippel and von Krogh, 2003). On one hand it is the

---

[6] In case of software the first three rights matter.

outcome of the private programming activity of hundreds of developers. On the other hand it meets the definition of public good. It is *nonexclusive* and *joint in supply*. Everybody is allowed to download the code from the Internet and use it. The consumption does not reduce its availability to other agents (Samuelson, 1954; Snidal, 1979). This lends to Open Source software the characteristics of common pool resources (McKean, 2000) and several authors refer to it as a *common* (Lessing, 1999; Khalak, 2000; O'Mahony, 2003).

The *commons* are prone to the *tragedy* described by Hardin (1968). Individuals do not have incentive to *show temperance* in using the resource that is then fated to be over- exploited until it gets unavailable to the community. However some scholars (Ostrom, 1990, 1999; Ostrom et al., 1999) have shown that this is not inevitable insofar as groups can succeed in setting up governing mechanisms (Sneath, 1998; Schlager, 1994) that allow them to manage the commons in a *sustainable way* (O'Mahony, 2003).

However Open Source software differs in some respects from the common resource analysed by Hardin (Lessing, 1999). First of all it is *subtractable*. The Open Source community does not own any more the code that is turned into proprietary.

Given that software is an intangible asset, the problem is not that the agents overuse the source code[7] but that they subtract it from the common pool. As a consequence there is no need to regulate the access to the common but rather the tragedy is avoided by mean of mechanisms that forbid subtraction.

The application of the copyright framework to the Open Source software is then a way to protect this peculiar kind of common (McJohn 2000). It prevents any subtraction of code from the common pool of the Open Source projects hosted on the Internet. Shortly, the governing mechanism that lays at the basis of the Open Source production mode is the following:

---

[7] Open Source developers aim that as many people as possible access their code. On one hand this makes effective the *Linux law: given enough eyeballs all bugs are shallow* (Raymond, 2001). On the other hand, developers have better chances to enhance their reputation among their peers if a large community inspects the code.

developers *own the copyright on the code*[8] *and then license it according to a peculiar mass market-licensing model*[9]. In this way copyright holders relax their control on the software but retain *some privileges* (Wheeler, 2003) so that they can claim a copyright infringement if someone hijacks their code. Even though Open Source licenses have not yet been tested in courts (Lerner and Tirole, 2002a), several scholars (Bobko, 2000; Lee, 1999; Kennedy, 2001; Järvinen, 2002) argue that they are *enforceable* under copyright law[10].

Rosenberg (1998) observes that today Open Source licenses are *abundant and surrounded by a lot of confusion*[11]. The Open Source Definition is a good guideline for addressing their characteristics. In fact, all its provisions must apply for a license to be Open Source (Fink, 2002; Gomulkiewicz, 1999; Lee, 1999). Fink (2002) underlines that there is no corresponding set of rules for proprietary licences.

The ten statements of Open Source Definition account for the way in which the bundle of rights provided by the copyright law serves the new paradigm.

The first provision deals with software redistribution: *the license shall not restrict any party from selling or giving away the software*. As a rule, the standard proprietary licenses strictly forbid redistribution. In turn, Open Source licenses allow everybody to redistribute computer programs by gifting or selling it. However, in the latter case, *the license shall not require a royalty or other fees for such sale*. Fees may cover the cost of media or duplication (Gomulkiewicz, 1999) and the provision of additional services or products (for e.g. of

---

[8] Within some Open Source projects developers set up a foundation, a legal person without a view to profit, to which they transfer the copyright on their software (O'Mahony, 2003). Let's think, for instance, of the Apache Software Foundation that rules the Apache project (http://www.apache.org). The Free Software Foundation (Stallman, 1984a) encourages developers to transfer to it the copyright of their programs.
This strategy aims at making easier to take legal steps in case of copyright infringement. In fact, many Open Source programs are written by a large number of developers. As a legal person, the foundation can proceed by itself and this is clearly less time and resource wasting than contacting and bringing together all the copyright holders.
[9] For an overview of the rationales under the licensing in the software field see Neukom and Gomulkiewicz, 1993; Lee (1999) summarizes the advantages of licensing software instead of selling it.
[10] Lee (1999) makes reference to *the enforceability of the Open Source licenses without law*. The non written norms of the Open Source community, in fact, lead developers to keep the code open (Lash, 1999).

proprietary programs) together with the Open Source software. The explicit reference made by this statement to *software selling* witnesses that Open Source Initiative aimed at fostering the exploitation of the new paradigm by commercial firms.

Then the Open Source Definition provides for the availability of the source code and states that the *program must include the source code* (Fink, 2002) together with the binary form. When this does not happen the license agreement must indicate how to obtain the source code free of charge. This second statement summarizes the very essence of the Open Source production mode. The accessibility of the source code allows programmers to detect, report and eventually fix bugs. Moreover it enables programmers modify the software and create *derivative works*. This improves the quality of Open Source programs. The rationale under this second provision[12] says that *the access to un-obfuscated source code [is required] because you can't evolve programs without modifying them.*

Derivative works increase the mass of Open Source and Open Source-based software in circulation. Every Open Source license allows to prepare and redistribute them (statement 3 of the OSD).

A taxonomy of the Open Source licenses based on their provisions for derivative works distinguishes between *copyleft* and *non-copyleft* license schemes. This depends on whether they *impose [or not] a burden of reciprocity upon the licensee* (Fink, 2002; McGowan, 2000; Stallman 1984d).

Copyleft licenses provide that *once a program is licensed by a developer the subsequent programs based on the original must also be licensed similarly* (Mustonen, 2003).

The first and most widespread of the Open Source licenses, the GNU General Public License (GNU GPL), defines a *derivative work* with reference to a computer program[13] (Moglen, 1999)

---

11 At present the Web site of the Open Source Initiative lists 43 OSI approved Open Source licenses (http://www.opensource.org/licenses).
12 http://www.opensource.org/docs/definition.php.
13 A general definition of *derivative work* is provided by the Copyright Act (USC Title 17, Chapter 1, Sec. 101): *a derivative work is a work based upon one or more pre-existing works, such as a translation, musical arrangement, dramatization, fictionalisation, motion picture version, sound recording, art*

as *any work that in whole or in part contains or is derived by the program*[14]. At section 2(b) it states explicitly that *any derivative work that you distribute or publish… [must] be licensed as a whole … under the terms of this License*[15]. The GPL preamble accounts for the rationale under this statement. The provision under section 2(b) forbids to release under a proprietary license the works that are derived from a GPLed program. This enhances the diffusion of GPLed software (Free Software programs) and guards the incentives of developers because it assures them that their code will remain open forever (Lerner and Tirole, 2002a; Kasper, 2001). Moreover the restriction is placed in order to guard the rights of the users*,* so that everybody is forbidden *to deny or to ask* them *to surrender* these rights.

However we must take into account the other side of the coin. In 1999 The Economist used the term *viral* to describe how the GNU GPL rules derivative works. The so called *viral provision* of the GPL (Lerner and Tirole, 2002a; Fernhout and Kurtz, 2001) frightened many potential users, in particular firms. It led them to think that *the mere interaction of their own proprietary software with* GPLed software *would have required them to publish their source code* (Rosen, 2001a). This prevented firms not only from developing Open Source programs but also from running them on their systems.

Nevertheless the term *viral* is not correct from a legal point of view and the firms' fear springs from a massive misunderstanding of the terms of copyleft licenses.

Legal scholars define copyleft licenses as *persistent* and speak about *inheritance* (Rosen, 2001b) property instead of viral provision. Reciprocity obligations do not arise from the use of the software. Programs *do not catch the GPL simply by touching [GPLed] software*.

The inheritance property of the GPL applies only to derivative works. A program is classified as a derivative work only if it modifies *the original work in some way*. The GPL license agreement

---

reproduction, abridgment, condensation, or any other form in which a work may be recast, transformed, or adapted.
[14] GNU General Public License, Version 2, June 1991, Term and Conditions, Section 2(b). http://www.gnu.org/copyleft/gpl.html, accessed on June 8th 2001.

[15] http://www.gnu.org/copyleft/gpl.html, accessed on 9th June 2003.

explicitly states that *if identifiable sections of the [derivative work] are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then the [GPL license], and its terms, do not apply to those sections when [they are] distribute as separate works.* As a consequence the simple distribution of the GPLed software together with other programs on the same medium does not affect the license schemes of these programs[16].

The Open Source Initiative addressed the confusion surrounding the inheritance provision of the GPL. First of all there is not an OSD statement that calls for reciprocity. Developers are allowed to *distribute derivative works under the same terms as the license of the original software* but this is not an obligation. Every Open Source license freely establishes how to distribute derivative works. Moreover OSD states that that *license must not contaminate other software.* The rationale under this *non-contamination clause* (Gomulkiewicz, 1999) is that distributors must not be forced to distribute only Open Source software on a given medium. According to Wichmann (2002) and Hecker (2000), software distribution is one of most important business opportunities that arise from the new paradigm of software production. Non-contamination clause aims at fostering it[17].

---

[16] It is worth to notice that GPL does not forbid multiple licensing: *the owner of the copyright in a software program always has the option to use multiple licenses* (Rosen, 2001b). Developers are allowed to release a program under the GPL and at the same time distribute it under a proprietary license scheme. The modifications made to the software released under the proprietary scheme do not have to comply with the GPL provisions. Moreover Stallman *provided a workaround for the viral problem by issuing some software under the Library General Public License (LGPL)* (Fink, 2002). This license scheme was elaborated in order to promote the usage of GNU C libraries and of Free Software libraries in general.
A computer program can *call* a library released under the LGPL and maintain its own license scheme. That is, to say that inheritance provision does not apply in this case,. However this holds only in case of *dynamic linking* that is if the program calls the library during its execution (Rosen, 2001a; Rosenberg, 2002). *The code has not to be compiled together or statically linked* (Rosenberg, 2002). Static linking, in fact, requires a modification of the program. Programmers involved in the Free Software movement have recently taken to call LGPL the *Lesser GPL*. Stallman, in fact, has started to discourage its use (Stallman, 1999).

[17] At the same time, the Open Source definition safeguards the users by stating that the rights attached to the program must not depend on the program's being part of a particular software distribution. They are allowed to extract the program from that distribution without loosing the rights granted by the Open Source license scheme.

Besides the GNU General Public License and the GNU Library General Public License (LGPL)[18], the group of copyleft licenses includes the Mozilla Public License and the IBM Public License. They were elaborated respectively by Netscape and IBM in order to release the source code of some programs to the Open Source community. Aiming at receiving feedbacks and contributions from individual developers, these software companies chose copyleft licenses in order to enhance cooperation. The fear of code hijacking, in fact, would drastically reduce the contributions from the community and choosing a copyleft license was the only way for the firms to credibly commit themselves not to hijack the code (Kuster et al., 2002; Osterloh et al., 2002).

Empirical investigations show that the Berkeley Software Distribution (BSD)[19] elaborated by the University of California at Berkley, is the most diffused of the non copyleft licenses[20] (Lerner and Tirole, 2002a; Robles et al., 2001). BSD[21] is very popular because it allows to do anything with the code (Lee, 1999). It is very short and places no restriction on redistribution and use in source and binary forms, with or without modifications. It allows to redistribute derivative works under any license scheme. Moreover it is possible to prepare derivative works by mixing BSD and proprietary code (Fink, 2002).

In accordance with the goals of the Open Source Initiative, this license reduces the gap between the Open Source community and the proprietary software world. As a consequence it turns out to be very suitable for commercial developers.

---

[18] For a description of the LGPL see note 21 and http://www.gnu.org/copyleft/lesser.html.

[19] http://www.freebsd.org/copyright/license.html, accessed on June 11th 2003.

[20] Other well-know non-copyleft licenses are the Apache Software License, ruling the distribution of the Web Server Apache, the Artistic License that was developed for the Perl language, the MIT license and the Python license, that is specific to the Python language. SUN Microsystems elaborated a non-copyleft license scheme, the SUN Industry Standard Source License (SISSL) that is applied to the Open Office Project.

[21] The BSD license was originally utilized as an Open Source licensing scheme for a variant of BSD UNIX called Free BSD and developed at the University of California at Berkley (Lee, 1999). FreeBSD has a distinguished story and at this time is probably a stronger and more scalable operating system than Linux (Rosenberg, 2002). For instance it powers large sites such as Yahoo. For a detailed description of BSD license scheme see Laird and Soraiz (1998).

Although this license meets firms' needs, it paves the way for code hijacking and increases the risk of project forking[22]. However BSD contains the copyright notice of the author[23] and states that *redistributions must reproduce* it. This grants the integrity of the author's source code as required by the fourth statement of the Open Source Definition[24]. In this way *[everybody] knows who is the responsible for the source code* (Fink, 2002). Developers win a reputation for their own programming activity and the mistakes made by others must not ruin it.

Open Source licenses safeguard programmers also in other respects. Licenses must not discriminate against any individual, group (Gomulkiewicz, 1999) or field of endeavour[25]. Everybody is allowed to use and develop Open Source software whatever his purposes[26]. Moreover all the Open Source licenses contain a disclaimer of warranty[27] that *shifts the risk to the licensee* (Gomulkiewicz, 1999) and states that the software is provided *as is* (Valimaki, 2002). This provision is very important for the survival of the Open Source model. Many programmers write software in their spare time without being paid[28]. The legal liability would imply to take *the risk of multi-million dollar class action law suit* (Gomulkiewicz, 1999) and would detract them from participating in the movement (Heffan, 1997).

A business opportunity arises from this risk-shifting strategy: firms working with Open Source software can charge for offering warranty protection. The section 1 of the GPL explicitly considers this option.

---

[22] For a detailed description of forking and its consequences see Rosenberg (1998, 2002).

[23] Copyright © <YEAR>, <copyright owner>. All rights reserved.

[24] The Apache Software License exploits intellectual property rights to serve the Open Source cause. This license allows the redistribution of derivative works under any distribution terms but forbids to use *the names "Apache Server" and "Apache Group" to promote any derivative works without the prior written permission of the Apache Group* (Lee, 1999). The Apache Software Foundation, ruling the Apache software, in fact, registered the trademark Apache because it was well aware of worldwide reputation of Apache as a reliable web server (O' Mahony, 2003).

[25] Fifth and sixth provisions of the Open Source definition

[26] These statements are quite controversial. For instance *an Open Source license cannot prevent terrorists from using the software* (Fink, 2002).

[27] For instance the Apache license agreement (http://www.apache.org/docs/license, accessed on 11th June, 2002) states that *this software is provided by the Apache Group "as is" and any expressed or implied warranties…are disclaimed*.

[28] Surveys made on Open Source developers (Hars and Ou, 2002; Hertel et al., 2003) have highlighted that there are programmers that are paid for their Open Source programming activity. Moreover many of them develop Open Source software during the regular working hours.

Finally the Open Source Definition accounts for license distribution. Open Source licenses are *shrink-wrap*. They are *provided without the opportunity to negotiate terms and are not signed* (Kennedy, 2001). The *rights attached to the software [simply] apply to everyone to whom the software is redistributed* (Gomulkiewicz, 1999) without the need *for execution of an additional license by those parties* (Pearson, 2000). This makes easier the circulation of Open Source software and safeguards once again the incentives of Open Source programmers. The right to create derivative works follows the software throughout the chain of distribution so as warranty disclaimer. Moreover, as stated by the rationale under the distribution provision,[29] distributing the license in such a way forbids *closing up the software by indirect means such as requiring a non-disclosure agreement*.

3. Open Source licenses and Open Source firms

The Open Source phenomenon is deeply affecting the software industry. On one hand several large software companies are releasing the source code of their programs to the Open Source community in order to receive contributions and feedbacks from individual developers. On the other hand new firms are entering the market. They supply Open Source based products and services and aim at profiting from the new production paradigm.

Which licenses do the new entrants use? At present we are not aware of empirical analyses that answer this question. The economic literature focuses mainly on the business models of these firms (Wichemann, 2002; Scott, 2002) and neglects to study their structural characteristics and their attitudes towards the Open Source software and its community. Moreover the research on Open Source phenomenon suffers from a lack of empirical data that is particularly marked in case of Open Source firms.

---

29 It is worth to notice that version 1.6 of the Open Source Definition *edited section 10*. This section states that *no provision of the license may be predicated on any individual technology or style of interface*. That is an Open Source license cannot *require certain functions, such as a popup dialog box for establishing an agreement* (Wedemeyer, 2003).

In order to explore these issues, between September 2002 and March 2003 we conducted a survey on Italian firms that do business with Open Source software[30]. We submitted a structured questionnaire[31] that gathered information on a large set of variables dealing with structural characteristics, business models, attitudes towards the Open Source software and participation in its community. In particular, we asked firms to indicate *the Open Source licenses with which they work*, *for the distribution of their software as well as the production process*. We made reference to the distinction between copyleft and non-copyleft distribution schemes (Agrain, 2003). Using these data, this paper aims at testing several theoretical hypotheses advanced by the literature on Open Source licenses. In order to make the discussion more lively, for each issue we present the hypothesis and our findings in sequence.

We classify the respondents as

- *copyleft firms (CF)* that work only with copyleft licenses.

- *mixed-license firms (MLF)* that make use of both classes of licenses

- *non - copyleft firms (NCF)* that rely exclusively on non copyleft distribution schemes

The first hypothesis that we aim at testing deals with firms' license choices.

H1: Like individual developers, firms prefer copyleft licenses (Perens, 2002)

Perens (2002), one of the founding fathers of the Open Source Initiative, highlights that *individual developers prefer* the GNU General Public License (GPL). This is a well grounded opinion. On August 2000, Robles et al. (2001) took data from BerliOS SourceWell[32], a large database that contains almost 2,279 Open Source applications[33]. They found that over 85% of the programs adopted the GNU General Public Licence (GPL). At

---

[30] 146 firms compose the sample. The firms were selected according to a snowball procedure. We approached an initial short list of firms and asked their collaboration in referring to other firms active in Open Source. We stopped when no new referral was originated. In this way we succeeded in contacting 275 firms that represent a reasonable cross-section of the Italian firms operating in the supply-side of the Open Source market. A first call took place from October to December 2002 and was carried out by e-mail using a questionnaire on line. A second call was carried out through phone interviews from January to March 2003.The database was built on April 2003.
[31] For further information about the methodology of the survey see Bonaccorsi and Rossi (2003a).
[32] http://sourcewell.berlios.de.

present 78.7% of the applications hosted at the BerliOS Open Source repository[34] are released under the GPL[35].

Also the largest Open Source repository Sourceforge[36] witnesses the leading role of the GPL. Lerner and Tirole (2002a) detected that *fully 72% of the nearly 40,000 Open Source projects* hosted at Sourceforge on May 2002 are GPLed. Currently 21,290 projects out of 28,959 are GPLed.

The distribution of the firms across the three groups corroborates H1 (table 1). The bias towards copyleft licenses shapes the choices of Open Source firms.

Variable	Acronym	%	Cumulative %
Mixed – license firms	MLF	60.1	60.1
Copyleft firms	CF	21.0	81.1
Non - copyleft firms	NCF	18.9	100

Table 1: Distribution of the firms according to the Open Source licenses that they use.

Most respondents use both copyleft and non-copyleft licenses while 21% of them use only the former. In all 112 (81.1%) firms work with copylefted software. Because of the legal implications of adopting copyleft, it is clear that the group of the firms using mixed models add to the diffusion of copyleft, even though other schemes are used. There are only 27 non copyleft firms. It is of interest to study the differences in business models and attitudes towards the Open Source software and its community that arise among the three groups.

Exploring the relationship between license policy and business models is very challenging. In fact the use of copyleft licenses by a firm might be due to a business model based on GPLed software instead of being a deliberate choice. That is, we want to test:

---

[33] The database witnesses the astonishing growth of the Open Source software phenomenon. In August 2000, when Robles et al. (2001) collected their data, SourceWell hosted 1,136 applications.
[34] http://sourcewell.berlios.de/stats.php?option=licenses, accessed on June 28th 2003.
[35] The Lesser GPL (5.6%) ranks second while the BSD license ranks third (2.8%).
[36] http://sourceforge.net. Accessed on 5th June 2003.

H2: The Open Source business models chosen by firms shape their license choices (Hawkins, 2003).

At a first glance this seems a sound hypothesis. 123 respondents out of 146 (84.2%) supply Linux based products[37] or provide services for them. Because of the inheritance property of the GPL that rules the Linux distribution, these products are released under the GPL too. In turn, firms must apply the GPL to all the software derived from them, even to the slight modifications that they make in order to meet their customers' needs. Kruskal - Wallis test shows that firms working with copyleft licenses (MLF and CF) supply on average more Linux based products than non copyleft firms (table 2).

Moreover most respondents provide services such as installation, support, and maintenance (Wichmann, 2002; Bonaccorsi and Rossi, 2002b) and about 45% of them supply Open Source products that are not Linux based (Figure 1).
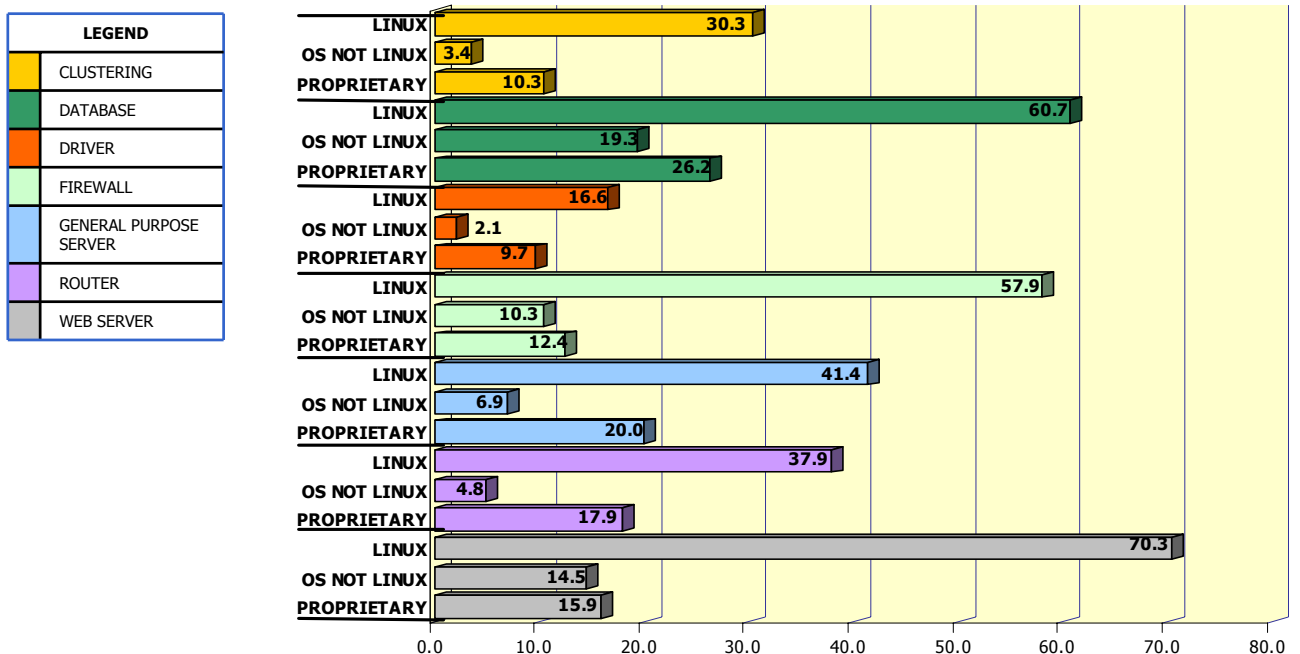

LEGEND	
	CLUSTERING
	DATABASE
	DRIVER
	FIREWALL
	GENERAL PURPOSE SERVER
	ROUTER
	WEB SERVER

CLUSTERING — LINUX 30.3, OS NOT LINUX 3.4, PROPRIETARY 10.3
DATABASE — LINUX 60.7, OS NOT LINUX 19.3, PROPRIETARY 26.2
DRIVER — LINUX 16.6, OS NOT LINUX 2.1, PROPRIETARY 9.7
FIREWALL — LINUX 57.9, OS NOT LINUX 10.3, PROPRIETARY 12.4
GENERAL PURPOSE SERVER — LINUX 41.4, OS NOT LINUX 6.9, PROPRIETARY 20.0
ROUTER — LINUX 37.9, OS NOT LINUX 4.8, PROPRIETARY 17.9
WEB SERVER — LINUX 70.3, OS NOT LINUX 14.5, PROPRIETARY 15.9

Figure 1: Products supplied by the firms.

---

[37] For instance Linux based Web server are offered by 69.9% of the firms.

17

Given that the GPL is the most widespread of the Open Source licenses, firms are very likely to come into contact with it in carrying out their activity[38].

Variables	Acronym	Group	N	Mean	Std. Dev.	Kruskal- Wallis test (p value)
Supply of Linux based products	LBP	Copyleft firms	28	3.2	2.2	0.065
		Mixed- license firms	83	3.5	1.9	
		Non - copyleft firms	26	2.4	2.2	
		Total	137	3.2	2.0	
Supply of proprietary products	PP	Copyleft firms	28	0.5	1.1	0.071
		Mixed- license firms	83	1.2	1.7	
		Non - copyleft firms	26	1.4	2.1	
		Total	137	1.1	1.7	

Table 2: Supply of Linux based and proprietary products in the three groups.

Despite the above cited results further investigations are needed. The respondents had the chance to fill the questionnaire on line or giving us a phone interview. During these interviews several of them claimed to work with the GPL because it guards the very values of the Open Source movement. It allows to keep the code open and forbids competitors to turn it into proprietary. This finding contrasts with the literature that underlines firms' fear of the GPL. Firms in our sample seem to be well aware of the pros and cons of this distribution scheme.

It is worth to notice that firms that use copyleft licenses supply on average less proprietary products (table 2). On one side this is due to their business models that are more Open Source oriented. On the other side it may indicate that firms weight carefully the inheritance property of the GPL when they take their production decisions.

Our survey gives much scope to motivations that lay at the basis of firms' involvement in the Open Source movement. We measure the incentives that lead firms to engage in Open Source activities using a Likert scale ranging from 1 (not at all important) to 5

(very important). Eleven motivations are taken into account. We follow Feller and Fitzgerald (2002) and distinguish among economic, social and technological incentives (Bonaccorsi and Rossi, 2003c).

Among social motivations we asked firms about the importance that they attach to the gift culture of the Open Source community (M5), to its values (M6) and to its fight for software freedom (M7). The literature has underlined that the belief that software should be free and that derived works will remain communal property is invoked by individual developers in favour of the copyleft schemes (Lerner and Tirole, 2002a). We therefore derive the following hypothesis:

H3: Firms that engage in Open Source activities because of social motivations choose copyleft licenses.

Table 3 summarize our findings on social motivations of the firms in the three groups, and the results of the Kruskall Wallis tests are reported[39].

In accordance with the theoretical literature, firms that make use of copyleft licenses, assign higher score to social motivations. They conform to the non written norms of the Open Source community and claim that the software should not be a proprietary good. This corroborate our hypothesis that Open Source firms choose copyleft licenses not only because of the inheritance property of the GNU GPL but also because of the same ideological reasons claimed by individual developers.

Moreover mixed license and copyleft firms attach an higher importance to the statement *we work with Open Source because in this field we can find easily good IT specialists.* This is a further confirmation that firms conform to the non written norms of the Open Source community because they aim at winning the trust of individual developers. As

---

[38] Chi Square tests show no difference in license use between the firms that provide installation, support and maintenance and the other ones.

[39] For sake of convenience we use the same acronyms of the companion paper.

Lerner (2002) underlines firms that do business with Open Source software *can spot the talented programmers for hiring purposes*.

Motivation	Acronym	Area	GROUPS												
			Copyleft firms			Mixed - license firms			Non copyleft firms			Total		
			N	Mean	Std. Dev.	N	Mean	Std. Dev.	N	Mean	Std. Dev.	N	Mean	Std. Dev.
Because we conform to the values of the Free Software movement***	M6	S	29	4.0	1.1	81	3.9	1.3	22	2.9	1.4	132	3.7	1.3
Because we think that software should not be a proprietary good**	M7	S	27	3.1	1.5	78	3.2	1.5	22	2.3	1.2	127	3.0	1.5
For the reliability and quality of the Open Source software*	M10	T	29	4.2	0.9	82	3.9	1.2	22	3.5	1.1	133	3.9	1.2

Table 3: Social motivations of the firms in the three groups, Kruskal – Wallis tests.

We distinguish between firms that assign high (4 and 5) and low (1 and 2) scores to M6 and M7 and run the Chi Square test for the three groups.

Table 4 summarize the results.

Motivation	Acronym	Score	Groups			Chi Square test (p value)
			Copyleft firms	Mixed- license firms	Non - copyleft firms	
Because we conform to the values of the Free Software movement	M6	Low score	13.6	17.9	50.0	0.008
		High score	86.4	82.1	50.0	
Because we think that software shoula not to be a proprietary good	M7	Low score	50.0	43.1	72.2	0.091
		High score	50.0	56.9	27.8	

Table 4: Social motivations of the firms in the three groups: high and low values, Chi Square tests.

The large majority of copyleft (86.4%) and mixed license (82.1%) firms assign a high score to M6 while only half of the non-copyleft firms do the same. M7 displays a different pattern. On one side there are less non copyleft firms (27.8%) attaching much important to the fight for software freedom. On the other side firms that choose a high score represent a tiny majority among the ones that work with copyleft licenses. These findings witness once again that at a firm level social incentives display the feature of extrinsic motivations.

Finally we compare the values assumed by %CS for firms that assign high and low scores to M6 and M7 (table 5). As we expected firms that assign low score to these social motivations make use on average of less copylefted software.

Variables	Acronym	Score	N	Media	Std. Dev.	Man -Withney test (p value)
Because we conform to the values of the Free Software movement	M6	High score	11	50.5	33.4	0.104
		Low score	21	29.5	37.6	
		Total	32	43.3	35.7	
Because we think that software should not to be a proprietary good	M7	High score	21	56.1	31.2	0.057
		Low score	9	31.9	34.4	
		Total	30	39.2	34.8	

Table 5: Percentage of copylefted software: firms that assign high and low score to M6 and M7.

However scholars agree that firms conform to the values of the Open Source community mainly because of an extrinsic motivation (Bonaccorsi and Rossi, 2003c; Osterloh et al., 2002). They aim at lowering down development costs through contributes and feedbacks from the Open Source community. As we have stated above, firms credibly commit themselves not to turn the code written by individual developers into proprietary only using copyleft licenses. As a consequence these distribution schemes are the most suitable way to sustain cooperation.

We cite the case of Netscape and its Mozilla project as an example. When, in 1998, the code of the Mozilla Web browser was released to the Open Source community under a non copyleft license, the Netscape Public License (NPL), it was able to attract very few contributions[40]. This forced Netscape to add a copyleft license, the Mozilla Public License, to the NPL (Lee, 1999) in order to generate the feedbacks that allowed to release the first version of Mozilla in 2002[41].

In short:

H4: Firms choose copyleft licenses in order to attract as many contributions as possible from the Open Source community (Lerner and Tirole, 2002b, Bonaccorsi and Rossi, 2003a and 2003c;Hawkins, 2003).

---

[40] NPL does not appear on the list of the OSI approved licenses although Netscape claims that it meets the requirements of the Open Source Definition.

In order to test this hypothesis we make reference to three indicators of the firms' attitudes towards the contributions from the community, that is motivations to carry out Open Source activities, level of engagement in projects and social contacts with the community.

H4.A: Firms that attach much importance to contributions and feedback from the Open Source community choose copyleft licenses with higher probability.

The role played by the contributions from the community to provide incentives for firms' engagement in the Open Source movement was addressed by the following item: *we work with Open Source software because we think that contributions and feedbacks from the community are very useful to fix bugs and improve our software* (M8). Firm attach much importance to M8. Ranking the eleven incentives by their average scores, this motivation ranks second.

Kruskal Wallis test shows that there is no difference in the score assigned to this item by the three groups of firms. Then we distinguish between firms that choose a high (4 and 5) and a low score (1 and 2) for M8 and run the Chi Square test in order to detect whether there are more copyleft firms that choose a high score. The test does not corroborate H4.A. About 70% of the respondents choose a high scale for this statement independently on the licenses that they use.

It would be better to merge mixed-license and copyleft firms. Firms that belong to mixed license group, in fact, might use their copyleft distribution schemes to attract contributions to projects whose development and R&D costs are high. Nevertheless, also in this case no statistically significant difference emerge between the two groups

---

[41] Starting from this first stable version, the releasing process of Mozilla has gained momentum. Mozilla 1.4 was released on June 30th 2003.

The decision process of a firm that use copyleft licenses only for a small fraction of its software is likely to be very different from the one of a firm that copyleft most of its programs. In the former case the use of copyleft licenses is likely to spring from the inheritance property instead of being the outcome of a deliberate decision. This may have affected the results of the our statistical tests. In order to overcome this problem we asked firms for the percentage of copylefted software (%CS) on the total of software that they supply to their customers or use as an input in their production process.

The respondents are highly heterogeneous with respect to this variable (table 6) that is lower than 30% for the majority of them (51.2%).

Variable	Acronym	N	Min	Max	Mean	Std. Dev
% of copylefted software	%CS	43	0	100	44.0	36.4

Table 6: Percentage of copylefted software (%CS): descriptive statistics.

As we expected, firms that assign high values to M8 displays higher values of %CS (table 7). As a result, the original formulation of H4.A should be rejected, while the following hypothesis is supported by data: *the firms that attach much importance to the cooperation of the community make a wider use of the copyleft schemes in order to sustain this cooperation*.

Motivation	Acronym	Area	Value	N	Mean	Std. Dev	Mann-Whitney test (p value)
We work with Open Source Software because contributions and feedbacks from the community are very useful to fix bugs and improve our software	M8	Technological	Low value (1-2)	8	28.7	39.8	
			High value (4-5)	24	50.2	33.0	0.094
			Total	40	45.4	36.0	

Table 7: Percentage of copylefted software: differences in mean values between firms that attach a low and a high value to the feedbacks from the community.

H4.B: Firms that participate to Open Source projects choose copyleft licenses with higher probability.

There are two rationales under this hypothesis. On one side, as we have underlined above, most Open Source projects are released under the GPL. As a consequence, the

higher the level of Open Source activity of a firm the more its chances to get into projects ruled by this license.

On the other side Firms that participate in Open Source projects show interest in interacting with the Open Source community and in obtaining contributions from developers. Moreover a firm that coordinate Open Source projects faces in a crucial way the problem to attract as many contributions as possible. As a consequence it is likely to use the copyleft schemes not only for projects that it coordinates but also for the other software with which it works. The adoption of copyleft license, in fact, improves firm's image within the Open Source community.

A companion paper (Bonaccorsi and Rossi, 2003d) accounts for the level of Open Source activity of the firms in our sample. We gathered data on the number of projects joined and coordinated by firms throughout their Open Source activity and during the year 2002[42]. Their developing effort is measured by the percentage of Lines of Code (LOCs) contributed on average to each project and by the number of patches and modules accepted into project official versions.

These data allow to distinguish between firms that participate and do not participate in Open Source projects. Chi square test highlights no difference in their license choice. The same happens if we take into account project coordination (firms that coordinate vs. firms that do not coordinate projects) and project contribution (firms that contributed LOCs or patches and modules to the projects vs. firms that make no contribution).

Following the same methodology used to test H3.A, we take into account %CS and run Mann Whitney tests for the above cited groups of firms. Tables 8 reports our findings. On average this variable is higher for firms that join and coordinate Open Source

---

[42] 2002.

projects and for the ones that contribute patches and modules to the projects to which participate. H4.B is then supported.

Variable	Acronym	Value	N	Mean	Std. Dev	Mann -Whitney test (p value)
Project membership from the very start of the firm's Open Source activity	PM	YES	17	56.2	32.0	0.074
		NO	15	35.3	34.4	
		Total	32	46.4	34.3	
Project coordination from the very start of the firm's Open Source activity	PC	YES	9	67.2	26.6	0.055
		NO	23	41.3	36.2	
		Total	32	48.6	35.4	
Firms' contributions (patches. modules) accepted into project official versions	AC	YES	5	71.0	25.6	0.059
		NO	22	39.5	33.9	
		Total	27	45.4	34.4	

Table 8: Percentage of copylefted software: firms' activity within Open Source projects.

H4.C: Firms that have social contacts with other agents working with Open Source software choose copyleft licenses with higher probability.

Our survey gathered data on social contacts of the respondents with several classes of agents, that is other firms that do business with Open Source software, firms that work only with proprietary software, members of the Open Source community, Universities and Research Centres, practitioner and firm groups and public bodies. We asked in particular for the number of agents that firms contact in order to exchange source code and information on the Open Source production mode. The contacts with other firms that do business with Open Source software (C_OSF) and with the Open Source community (C_OSC) then play a crucial role in determining whether hypothesis H4.C holds.

In order to test H4.C we make reference to the social contacts with the first and the third of the above cited categories. Table 9 reports descriptive statistics of these two variables.

Variable	Acronym	N	Min.	Max	Mean	Std. Dev.
Number of contacts with other firms doing business with Open Source software	*C_OSF*	102	0	200	10.6	25.5
Number of contacts with the Open Source community	*C_OSC*	86	0	1,000	54.0	186.7

Table 9: Firms social contacts: descriptive statistics.

Once again Kruskall Wallis test shows no difference in C_OSF and C_OSC among the three groups of firms. The same happens also if we merge copyleft and mixed license firms and if we compare non copyleft firms with the other respondents. However copyleft firms attach a higher reliability to the information that they get from the Open Source community (table 10).

Variables	Acronym	Groups	N	Mean	Std. Dev.	Mann -Whitney test (p value)
Reliability attached to the information received from the Open Source community	*RIOSC*	*Mixed license and non copylefted firms*	62	4.0	0.8	
		Copyleft firms	12	4.4	0.7	0.107
		Total	74	4.1	0.8	

Table 10: Reliability attached to the information received from the Open Source community: the copyleft firms and the other ones.

Moreover firms that have social contacts with the Open Source community on average make use of a higher percentage of copylefted software (table 11). This clearly corroborates H4.C.

Variables	Acronym	Value	N	Media	Std. Dev..	Mann -Whitney test (p value)
Contact with the Open Source community	*COSC*	YES	16	54.7	30.3	
		NO	3	20.0	20.0	0.090
		Total	19	49.2	31.3	

Table 11 Percentage of copylefted software: firms that have and do not have contacts with the Open Source community.

4. Conclusions

We have provided evidence of the selection and use of licensing schemes by firms active in the Open Source software fields. Copyleft licenses play a dominant role, but only a small minority adopt them within a pure model, while 60% of firms combine them non- copyleft licenses, including, if needed, proprietary ones.

The intensity of use of copyleft is associated to the weight of Linux based products in the line of offer, the strength of social motivations, and various indicators of active

involvement into the Open Source community. The use of mixed licensing models imply that firms are not at all worried about the (wrongly) assumed viral property of copyleft, but rather know very well how to separate the legal implications of different licensing schemes.

At the same time, complying with copyleft is not only a direct consequence of the use of Linux-related products, but is also a way to maintain active relations with the Open Source community. Firms do not contribute intensely to the very *production* process of Open Source software (Bonaccorsi and Rossi, 2002d). However, by adopting copyleft licenses they greatly contribute to the *distribution* process enhancing the incentives of programmers by creating a larger user community. The fact that legal innovations such as the copyleft license schemes allow strict complementarity between pure social or ideological motivations and pragmatic business incentives is an interesting outcome of the Open Source movement.

5. References

Agrain P. (2002) *A framework for understanding the impact of GPL copylefting vs. non-copylefting licenses*. MIT Working Paper.

Anez J. (1999) *The true meaning behind of the Open Source licenses*. http://www.suigeneris.org/writings/1999-10-04.html, accessed on June3[rd], 2003.

Bassi N. (2000) *Open Source. Analisi di un movimento.* Apogeo, Milano, Italy.

Bessen J. (2001) *Open Source software: free provision of complex public good.* MIT Working Paper.

Bobko P. K. (2000) *Linux and general public licenses: can copyright keep Open Source software Free?.* AIPLA Quarterly Journal, 28(1), 81 - XX.

Bobko P. K. (2001) *Open Source Software and the demise of copyright.* Rutgers Computer and Technology Law Journal, 27, 51-92.

Bollier D. (2002) Why public domain matter. The Endangered Wellspring of Creativity, Commerce and Democracy. New America Foundation, Public Knowledge, Washington, DC, http://www.newamerica.net/Download_Docs/ pdfs/Pub_File_867_1.pdf, accessed on June3[rd], 2003.

Bonaccorsi A., Rossi C. (2003b) *Heterogeneity in business models in Open Source software.* Unpublished manuscript.

Bonaccorsi A., Rossi C. (2003c) *Altruistic individuals, selfish firms? The structure of motivation in Open Source software*. Unpublished manuscript.

Bonaccorsi A., Rossi C. (2003a) *Why Open Source can succeed.* Research Policy, 32(7), 1243-1258.

Bonaccorsi A., Rossi C. (2003d) *Contributing to common pool resources. Comparing firms and individuals in the production of Open Source software*. Unpublished manuscript.

Boyle J. (2003) *Foreword: the opposite of property?* Law and contemporary Problems, 66, 1-32.

Cusumano M. A., Yoffie D. B. (1998) *Competing on the Internet time: lesson from Netscape and its battle with Microsoft.* Free Press, New York, NY.

Feller J., Fitzgerald B. (2002) *Understanding Open Source Software Development.* Addison Wesley, Boston, MA.

Fernhout P. D., Kurtz F. C. (2001) *A review of licensing and collaborative development with special attention to the design of self-replicating space habitat systems*. http://www.kurtz-fernhout.com/oscomak/SSI_Fernhout2001_web.html, accessed on June 9th 2003.

Fink M. (2003) *The business and economics of Linux and the Open Source.* Person Education, Prentice Hall, NJ.

Gomulkiewicz R. W. (1998) *The license is the product: comments on the promise of article 2B for software and information licensing.* Berkeley Technology Law Journal On Line, 13(3), http://www.law.berkeley.edu/journals/btlj/articles/vol13/Gomulkiewicz/html/reader.html.

Gomulkiewicz R. W. (1999) *How copyleft uses license rights to succeed in the Open Source software revolution and the implications for article 2B*. Huston Law Review, 36, 179-194.

Ghosh (2003) *Understanding Free Software Developers: Findings from the FLOSS Study*. Paper presented at HBS - MIT Sloan Free/Open Source Software Conference: New Models of Software Development, June 19-20, 2003, Boston and Cambridge, MA, USA.

Hardin G. (1968) *The tragedy of the commons*. Science, 162, 1243–1248.

Hawkins, R. (2003) *The economics of Free Software for a competitive firm*. MIT Working papers. http://opensource.mit.edu/papers/hawkins.pdf, accessed on July 3rd 2003.

Hecker F. (2000) *Setting up shop: the business of Open-Source software*. http://www.hecker.org/writings/setting-up-shop.html, accessed on March 26th 2003.

Heffan I. V. (1997) *Copyleft: licensing collaborative works in the digital age*. Stanford Law Review, 49, 1487-1509.

Hertel G., Niedner S., Hermann S. (2003) *Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel*. Forthcoming in Research Policy Special Issue on Open Source Software Development.

Horne N. T. (2001) *Open Source software licensing: using copyright law to encourage free use*. Georgia State University Law Review, 17, 863 – xx.

Järvinen H. (2002) *Legal aspects of open Source licensing*. http://www.cs.helsinki.fi/u/campa/teaching/oss/papers/jarvinen.pdf, accessed on June 7th 2003.

Johnson (2001) *Economics of Open Source software*. MIT Working Paper.

Kasper E. (2001) Epistemic communities, situated learning and Open Source software development. http://opensource.mit.edu/papers/kasperedwards-ec.pdf, accessed on June 4th, 2003.

Kennedy D. M. (2001) *A primer on open source licensing legal issues: copyright, copyleft and copyfuture*. Saint Louis University Public Law Review, XX

Khalak A. (2000) Economic model for impact of Open Source software. MIT Working Paper. http://opensource.mit.edu/papers/osseconomics.pdf, accessed on 7th June, 2003.

Kollack (1998) *Social dilemmas: the anatomy of cooperation*. Annual Review of Sociology, 24, 183-214.

Kuster B., Osterloh M., Rota S. (2002) *Trust and commerce in Open Source – a contradiction?* MIT Working Paper.

Laird C., Soraiz K. (1998) *The story of FreeBSD*. Linux World, http://www.linuxworld.com/linuxworld/lw-1998-12/lw-12-freebsd.html, accessed on June 11th 2003.

Lash A. (1999) *Source code for the masses*. http://news.com.com/2009-1001-207659.html, accessed on June 8th 2003.

Lee J. (2003) *Open Source software licenses and innovation*. http://VirtualGoods.tu-ilmenau.de/2003/licenses_innovation_leejz.pdf, accessed on June3rd, 2003.

Lee S. H. (1999) *Open Source software licensing*. http://eon.law.harvard.edu/openlaw/gpl.pdf, accessed on June3rd, 2003.

Lerner J., Tirole J. (2002a) *The scope of Open Source licenses*. MIT Working Paper.

Lerner J., Tirole J. (2002b) *The simple economics of the Open Source*. Journal of Industrial Economics, 52, 197-234.

Lessing L. (1999) *Code and the commons*. Keynote given at a conference on Media Convergence. Fordham Law School, February 9th, New York, NY.

Maher M. (2000) *Open Source software: the success of an alternative intellectual property incentive paradigm.* Media & Entertainment Law Journal, 10 (3), 619 - XX

McGowan D. (2001) *Legal implications of Open Source software.* University of Illinois Law Review, 1, 241- XX.

McJohn S. M. (2000) *The paradoxes of the Free Software*. Gorge Mason Law Review , fall, 25-68.

McKean, M.A. (2000) Common property: what is it? What is it good for and what makes it work? In C.C. Gibson, M.A. McKean & E. Ostrom (eds.) People and Forests. MIT Press, London, UK.

Moglen E. (1999) *Anarchism triumphant: free software and the death of copyright*. First Monday, Peer Review Journal on the Internet, http://emoglen.law.columbia.edu/my_pubs/anarchism.html, accessed on June 8th 2003.

Moglen E. (2000) *Free Software matters: Free Software or Open Source?* http://emoglen.law.columbia.edu/publications/lu-07.html, accessed on June 8th, 2003.

Monti A., (2002) Open Source e diritto d'autore. http://www.andreamonti.net/info/copyfree.html, accessed on 7th June 2003.

Mustonen M. (2002) *Why do firms support the development of substitute coplyleft programs?* FPPE, University of Helsinki Working Paper, http://www.valt.helsinki.fi/fppe/GS%20fellows/ mustonen_paper4.pdf, accessed on June 4th, 2003.

Mustonen M. (2003) *Copyleft – the economics of Linux and other Open Source software*. Information Economics and Policy, 15, 99-121.

Neukom W. H., Gomulkiewicz (1993) *Licensing rights to computer software*. In Technology licensing and litigation, Practicing Law Institute Patents, Copyrights, Trademarks and Literary Property Course Handbook Series, Practicing Law Institute, New York, NY.

Nilendu P., Madanmohan T. R. (2001) *Competing on Open Source: Strategies and Practise.* http://opensource.mit.edu/papers/madanmohan.pdf, accessed on June 5th 2003

O'Mahony S. (2003) *Guarding the commons: how community managed software projects protect their work*. Research Policy, 32(7), 1179-1198.

Oksanen V., Valimaki M. (2002) *Evaluation of Open Source licensing models for a company developing mass market software.* http://www.hiit.fi/de/valimaki_oksanen_lawtech_2002.pdf, accessed on June 5th 2003.

Osterloh M., Rota s., Von Wartburg M. (2002) *Open source – new rules in software development*. http://www.unizh.ch/ifbf/orga/downloads/opensourceaom.pdf , accessed on April 25th 2003.

Ostrom E. (1990) *Governing the commons: the evolution of institutions for collective action*. Cambridge University Press, New York, NY.

Ostrom E. (1999) *Copying with tragedy of commons*. American Review of Political Science, 2, 493-535.

Ostrom E., Burger J., Field C. B., Norgaard R. B., Policansky D. (1999) *Revisiting the commons: local lessons, global challenges*. Science, 284, 278-282.

Oz E. (1998) *Acceptable protection of software intellectual property*. A survey of software developers and lawyers. Information and Management, 34, 161-173.

Pearson H. E. (2000) *Open Source- the death of proprietary systems?* Computer Law and Security Report, 16(3), 151-156.

Perens B (2002) *The future of Open Source software*. In Proceeding of the OpenGroup conference "Boundaryless Information Flow: The Role of Open Source", October 14-18 2002, Cannes, France.

Raymond E. S. (2000) *Software Release Practice HOWTO*. http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/Software-Release-Practice-HOWTO.pdf, accessed on 7th June, 2003.

Raymond E.S. (2001) *The cathedral and the bazaar. Musings on Linux and Open Source by an accidental revolutionary*. O' Really, Sebastopulus, CA.

Robles G., Scheider H., Tretkowski I., Weber N. (2001) *Who is doing it? A research on Libre Software developers*. Fachgebiet für Informatik und Gesellschaft TU-Berlin, http://widi.berlios.de/paper/study.html, accessed on May 15th 2003.

Rosen L. (2001a) *The unreasonable fear of infection*. http://www.rosenlaw.com/html/GPL.PDF, accessed on June 9th 2003.

Rosen L. (2001b) *Which Open Source license should I use for my software?* http://www.rosenlaw.com/html/GL5.pdf, accessed on June 9th 2003.

Rosenberg D. K. (1998) *Evaluation of public software licenses*. http://www.stormian.com/Public_Licenses.html, accessed on June 4th, 2003.

Rosenberg D. K. (2002) *Open Source. The unauthorized white paper*. IDG Books Worldwide, New York, NY.

Rothfuss G. J. (2002) *A framework for Open Source projects*. Master Thesis in Computer Science, Department of Information Technology, University of Zurich.

Saers N. (2002) *The Open Source community as a counter network to the software development corporate networks*. http://niklas.saers.com/articles/counternetwork.pdf, accessed on July 7th 2003.

Samuelson P. (1954) *The pure theory of public expenditure*. The Review of Economics and Statistics, 36(4), 387- 389.

Schlager (1994) *Fisher's institutional responses to common-pool resource dilemmas*. In E. Ostrom, R. Gardner, and J.M. Walker (eds.), Rules, games and common-pool resources, University of Michigan Press, Ann Arbor, MI.

Scott B. (2002) *Why Free Software's long run TCO must be lower*. http://www.linuxandmain.com/modules.php?name=News&file=article&sid=281, accessed on June 28th 2002.

Sloan B. (2002) Copyleft: the great giveaway. http://www.library.yale.edu/~llicense/ListArchives/0202/msg00000.html, accessed on 7th June, 2003.

Sneath (1998) *State policy and pasture degradation in inner Asia*. Science, 181, 1147-1148.

Snidal D. (1979) *Public goods, property rights and political organizations*. International Studies Quarterly, 23, 532-566.

Stallman (1984b) *Categories of Free and Non-Free Software*. http://www.gnu.org/philosophy/categories.html, accessed on June 3rd, 2003.

Stallman (1984c) Why Free Software is better than Open Source. http://www.fsf.org/philosophy/free-software-for-freedom.html, accessed on June 8th 2003.

Stallman (1984d) What is copyleft? http://www.gnu.org/copyleft/copyleft.html, accessed on June 8th 2003.

Stallman (1999) *Why you shouldn't use the Library GPL for your next library.* http://www.gnu.org/philosophy/why-not-lgpl.html, accessed on June 9th 2003.

Stallman R. (1984a) *The GNU Manifesto.* http://www.gnu.org/gnu/manifesto.html, accessed on June 8th, 2003.

The Economist (1999) *Hacker rule.* Economist Print Edition, February 18th 2002.

Valimaki M. (2002) *Open Source licensing for users and developers.* Turre Legal Consulting, http://www.turre.com, accessed on June 11th 2003.

Von Hippel E., Von Krogh (2003) *The private-collective innovation model in Open Source software development: issue for organization science.* Organization Science, Forthcoming.

Wedemeyer L. (2003) *The Open Source movement.* http://www.informatics-review.com/thoughts/opensource.html, accessed on June 15th 2003.

Wheeler D. A. (2003) *Open Source Software / Free Software (OSS/FS) References* http://www.dwheeler.com/oss_fs_refs.html, accessed on 7th June 2003.

Wichmann T. (2002) *Basics of Open Source software markets and business models. Free/Libre and Open Source Software: Survey and Study.* FLOSS Final Report, International Institute of Infonomics, Berlecom Research GmbH.

Williams S. (2002) *Free as in freedom: Richard Stallman's crusade for Free Software.* O'Reilly and Associates, Sebastopolous, CA.

Lerner J. (2002) *Incentives and Open Source software.* http://cip.umd.edu/lerner.htm, accessed on April 25th 2003.