An Exploratory Study of the Evolution of Software Licensing

Massimiliano Di Penta
University of Sannio, Italy
dipenta@unisannio.it

Daniel M. German
University of Victoria, Canada
dmg@uvic.ca

Yann-Gaël Guéhéneuc, Giuliano Antoniol
École Polytechnique de Montréal, Canada
yann-gael.gueheneuc@polymtl.ca,
antoniol@ieee.org

ABSTRACT

Free and open source software systems (FOSS) are distributed and made available to users under different software licenses, mentioned in FOSS code by means of licensing statements. Various factors, such as changes in the legal landscape, commercial code licensed as FOSS, or code reused from other FOSS systems, lead to evolution of licensing, which may affect the way a system or part thereof can be subsequently used. Therefore, it is crucial to monitor licensing evolution. However, manually tracking the licensing evolution of thousands of files is a daunting task.

After presenting several cases of the effects of licensing evolution, we propose an approach to automatically track changes occurring in the licensing terms of a system. Then, we report an empirical study of the licensing evolution of six different FOSS systems. Results show that licensing underwent frequent and substantial changes.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Copyrights*

General Terms

Legal Aspects

Keywords

Software licenses, evolution, mining software repositories, open source systems, empirical study.

1. INTRODUCTION

> *OpenBSD founder and project leader Theo de Raadt removed a security software package called IP-Filter [written by Darren Reed] after its author changed its license.*
> *—Stephen Shankland, CNET News, 2001/05/30.*

ICSE '10, May 2-8 2010, Cape Town, South Africa

As software systems evolve, so do licenses. Although software licenses have gained prominence in the media, thanks for example to the work of the Free Software Foundation; licensing evolution has received little attention from researchers despite its many potential harmful consequences on software reuse. An interesting example of such phenomena is the licensing evolution of IPFilter [20], which prevented its redistribution as part of the OpenBSD kernel. A side-effect of this evolution was the creation of PF by OpenBSD developers as an alternative to IPFilter[1].

Licensing, copyright, and intellectual property determine what can and cannot be reused, and potentially impacts the architecture of a system. A typical scenario in which licensing evolution monitoring becomes vital is as follows. A company creates a product—*e.g.*, a hand-held multimedia player—that incorporates some FOSS components/applications, *e.g.*, a Unix kernel, the player, plus several codecs. The critical situation arises when there is the need for updating a component (*e.g.*, a codec) but its license changed (*e.g.*, from BSD to GPL) and its new license prevents its distribution, thus requiring to completely re-think the way this codec is connected to the player [10]. Consequently, developers must carefully analyze the overall licensing compatibility of all the included components. This compatibility analysis is usually done manually, or semi-automatically, by verifying that all bundled source files and binaries have been released under compatible licenses [19]. Unfortunately, as illustrated by the IPFilter case, this analysis is not a one-time activity: each modification to any bundled component may involve licensing statements and thus impact its use/integration.

The harmful consequences of licensing evolution stem from the nature of open source development: "this method of development can be worrisome from an intellectual property standpoint because it creates multiple opportunities for contributors to introduce infringing code and makes it almost impossible to audit the entire code base" [1]. Such consequences will become more prominent because (1) the United States Court of Appeals for the Federal Circuit ruled that redistributing software systems in violation of the terms of a free software license constitutes a copyright infringement [17] and (2) the sizes of modern software systems prevent manual analysis, *e.g.*, Mozilla grew from 4,845 files in release M3 (March 1999) to 12,436 in release 1.7.13 (September 2004).

Therefore, there is a need for methods and tools to (semi-) automatically audit a system for its license and for changes in its licensing statements across releases.

[1] `http://www.openbsd.org/faq/pf/`

The license (or licenses) under which a file is made available is usually contained inside blocks of comments at its beginning. We refer to such comments as the *licensing statement* of a file. We distinguish between *licenses* and *licensing statements* because a licensing statement may either contain the license itself (*e.g.*, BSD license) or the name of the license and a reference to where it can be found (*e.g.*, the Eclipse Public License). In this paper, we are interested in changes to the licensing of source code, *i.e.*, the analysis of changes occurring within licensing statements, while license evolution concerns the evolution of a license *per se*, *e.g.*, the evolution of the GPL from v2 to v3. A change to the licensing statement might be a change to the name of the license (when the statement refers to its name) or a change to the license itself (when the license is in the statement).

This paper first motivates the problem of analyzing the evolution of licensing statements by providing several examples of changes in licensing and their consequences. Then, it suggests that the only means to avoid negative consequences is to monitor licensing evolution automatically. It proposes an approach to automatically track the licensing evolution of systems, identifying changes in licenses and copyright years. Finally, it reports an empirical study analyzing the licensing evolution of six widely-adopted FOSS systems: ArgoUML, Eclipse-JDT, the FreeBSD and OpenBSD kernels, the Mozilla Suite, and Samba. The study shows that licensing evolution is a frequent and relevant phenomenon in many systems and that, while FOSS developers are concerned with licensing issues, they manage, evolve, and update licensing statements in different ways. For example, some systems, *e.g.*, Mozilla and Eclipse-JDT, have moved from more restrictive to less restrictive licenses while others have moved in the opposite direction. Copyright years are updated following different patterns in different systems.

To the best of our knowledge, no previous work investigated the licensing evolution of FOSS systems. Thus, the contributions of this paper are as follows:

- We show that *keeping track of licensing changes is important* by reporting several cases where changes in open-source licenses had major consequences in software usage and integration;
- We propose *a method to track licensing changes*. The method is based on textual analysis of licensing statements extracted from source code file and on the usage of the FoSSology licensing classification tool [11].
- We report an empirical study showing *the extent and frequency of licensing changes* by analyzing the licensing evolution of six open source systems: ArgoUML, Eclipse-JDT, the FreeBSD and OpenBSD kernels, the Mozilla Suite, and Samba.

The paper is organized as follows. Section 2 shows several cases in which licensing changes impacted software usage and integration. Section 3 describes the licensing analysis process. Section 4 describes our study and the process followed to mine data from the six chosen FOSS systems. Section 5 presents the empirical study results. Section 6 discusses the results along with threats to validity. After a discussion of related work in Section 7, Section 8 concludes and outlines directions for future work.

2. LICENSING EVOLUTION

A dual relation exists between license evolution and changes in licensing statements. Licensing statements are changed to allow developers using the license that better fits their needs and–or users' needs. Such changes occur by modifying the licensing statements of the system files to refer to a different license, to a new version of the original license, or to update the license when it is included in the licensing statements. Therefore, licensing changes depend on the availability of fitting licenses. We first illustrate license evolution and then licensing evolution using real-world cases.

2.1 License Evolution

License evolution is driven by many factors. On the one hand, copyright owners want licenses to adapt to the new legal landscape and include their specific requirements. For example, the Netscape Public License, the IBM Public License, and the Apple Public Licenses were created to satisfy their organizations' requirements. On the other hand, users want licenses to adapt to their needs, often by becoming less restrictive, *e.g.*, the original BSD license (also known as 4-clauses BSD) evolved towards its less restrictive 3-clauses and 2-clauses variants. Some licenses evolve towards more restrictive ones, such as the changes made to the General Public License (GPL) version 2 to avoid hardware locks and digital rights management, which led to the GPL v3. In other cases, licenses evolve due to external pressures. For example, the evolution of the license of Mozilla from Netscape Public License (NPL) to Mozilla Public License (MPL) v1.1 was triggered by the opposition of the open-source community to some of the terms in the NPL. This evolution reflected an interest of the Mozilla Foundation, the copyright owner of Mozilla, to address its users' concerns.

Table 1 shows some of the most commonly used FOSS licenses, the rationales for their evolution, and where applicable, the other licenses on which they are based. With a large number of available licenses, among which 65 are certified by the Open Source Initiative, it is not surprising that licensing statements evolve.

2.2 When Licensing Evolution affects Software Usage: Five Cases

We now report cases in which licensing evolution was triggered by a specific requirement and influenced the way in which the system is used. Table 2 summarizes these cases.

Case 1: OpenBSD IPFilter Replacement.

In 2001, the author of IPFilter, a firewall package used by OpenBSD, added an extra sentence to the licensing statement of each files of IPFilter and, hence, to the IPFilter license. According to the author, this sentence was a clarification to the terms of the license but developers of OpenBSD considered this a new condition, incompatible with the license of OpenBSD. OpenBSD developers decided to replace IPFilter with a new OpenBSD-based implementation [12].

Changing the license of FOSS system might result in users no longer being able to reuse the software.

Case 2: Java.

Until November 2006, one of the major problems to include Java in Linux distributions was its license. The license of Java JDK v1.2 included the following sentence: *"Except as specifically authorized in any Supplemental License Terms, you may not make copies of Software, other than a single copy of Software for archival purposes."* This requirement disallowed the inclusion of Java in Linux distributions. Con-

Table 1: Examples of how some open source licenses have evolved over time.

Names of licenses	Years	Derived from	Types of changes with respect to its predecessors
General Public (GPL) v1	1989	Emacs Public	Major rewrite. Generalizes the Emacs Public License
GPL v2	1991	GPL v1	Replaces some philosophical language with legal one. Adds two clauses: the "Liberty or Death clause", and allows geographical exclusion
GPL v3	2007	GPL v2	Major rewrite. Adds hardware restrictions, addresses software patents
Library GPL (LGPL) v2	1991		
Lesser GPL (LGPL) v2.1	1999	LGPLv2	Replaces language with a more legal one. Major modifications to preamble. Allows geographical exclusion under certain circumstances
Lesser GPL (LGPL) v3	2007	LGPLv3	Major rewrite. Improves legal language. Licensed software under it can only be changed to GPLv3
Mozilla Public (MPL) v1.0	1998		
Netscape Public (NPL) v1.0	1998	MPL v1.0	Identical to the MPLv1.0 except for the addition of an Amendments section
MPL v1.1	1999	MPL v1.0	Replaces language with a more legal one, adds patent clauses. It adds the option of using dual licensing
NPL v1.1	1999	MPL v1.1 and NPL v.1.1	Identical to MPLv1.1 except for the addition of an Amendments section (moves it from end of the license to the beginning of it)
IBM Public (IBMPL) v1.0	1999		
Common Public (CPL) v0.5	2000	IBMPL v1.0	Clarifies definitions; makes the license reusable by replacing IBM's name with *Contributor*, and declaring IBM as the steward of the license
CPL v1.0	2001	CPL v0.5	Identical to the CPL v0.5
Eclipse Public (EPL) v1.0	2002	CPL v1.0	Removes a clause regarding patent litigation against developer, and changes the steward of the license from IBM's to the Eclipse Public Foundation
BSD 4 clauses	1983		
BSD 3 clauses	1999	BSD 4	Removes Advertisement clause
BSD 2 clauses	2008	BSD 3	Removes Endorsement clause, making it similar to the MIT/X11

Table 2: Effects of licensing changes.

Systems	Licenses	Changes	Effects
IPFilter	IPFilter-specific license	Added a "clarification" sentence	IPFilter was removed from the OpenBSD distribution
Java	Java-specific License	GPL v2 with CLASSPATH exception	It allows modifying and updating Java
Mono	GPL v2	MIT/X11	It allows using Mono with systems under any license
QT	FreeQT	Q Public License, then GPL v2, and finally both LGPLv2.1 and GPLv3	Project Harmony, a replacement of QT, was abandoned
MySQL	LGPL v2.1	GPL v2	It prevents PHP systems to connect to MySQL

sequently, for many years, end-users had to manually download and install Java. Sun Microsystems worked with the Free Software Foundation (FSF) and released Java 5.0 under the GPL v2 with an addendum known as the CLASSPATH exception [9]. This change in the licensing of Java had two major implications: first, Java could from then on be modified and updated under the GPL v2, without interference by Sun; and, second, Java programs could be released under any license as long as they satisfy the conditions stated in the CLASSPATH exception.

Changing the license of a system can promote and ease the distribution and reuse of a software system.

Case 3: Mono.

Mono[2] is a framework produced by Novell to support the .Net API (and thus Microsoft software systems) under operating systems different than Microsoft Windows. Originally, the project was distributed under the GPL v2. According to Mono developers, this license created a potential problem when running .Net systems because they could be considered derivative works of Mono and, hence, required to be also released under the GPL v2. Consequently, Mono developers changed its license to MIT/X11, a simple FOSS license that allows its use along systems distributed under any commercial or FOSS license [18]. According to Mono's project leader, this change was also required by HP as a condition for its participation as a contributor to the project [21]. Thus, .Net systems can use Mono regardless of their respective licenses.

[2]http://mono-project.com

A change to a more permissive license (and in particular, allowing commercial derivative works) may increase the size of the community of contributors to a FOSS system.

Case 4: QT.

QT is a library of GUI widgets, originally developed by Trolltech, bought by Nokia in 2008. QT was first released under a non-open source but free license, called the FreeQT License, and a commercial license. QT became the basis for KDE, the desktop suite for Unix systems. Many objected to the use of a non-open source library as the basis of a major open-source system, including Richard Stallman. To address these issues, QT v2.0 was released under a new license, called the Q Public License. The Q Public License was approved by the Open Source Initiative but deemed incompatible with the GPL by the FSF [7]. According to the FSF, because many application in KDE are licensed under the GPL, this incompatibility makes their use of QT a violation of their own license (the KDE project disputed this view). Consequently, the GNOME project was started as a QT-free alternative to KDE, while the Harmony project was started to implement a replacement of QT to be licensed under the GPL. Trolltech changed the license of QT v3 to the GPL v2. The Harmony Project was no longer necessary and abandoned [15]. When Nokia acquired Trolltech, it changed the license of QT v4.6 to a dual LGPL v2.1 and GPL v3.

Changing the license of FOSS system towards a more permissive might cause the abandonment of a competing system.

Case 5: MySQL.

In 2004, MySQL AB changed the license of its client li-

braries from LGPL v2.1 to GPL v2. This change was intended to prevent industrial companies from using the libraries within proprietary products without paying for a commercial license. Unfortunately, it had also unintended consequences: PHP systems were no longer able to connect to MySQL because the PHP license is incompatible with the GPL v2. MySQL addressed this problem by adding the MySQL FOSS License Exception to the GPL v2 [9].

Changing the license of a FOSS system might have unintended or undesirable consequences to its legitimate users.

Lessons from these five cases.

We illustrated using five cases, summarized in Table 2, that changes in licensing can have various consequences (expected and unexpected). Thus, developers and their organizations should be aware of licensing changes and their potential effects. FOSS development encourages the contributions of many developers, who can willfully or inadvertently change licensing statements. Therefore, an approach to analyze licensing changes in source code is needed.

3. LICENSING ANALYSIS METHOD

The analysis of licensing changes takes as input source code file revisions extracted from version control systems (such CVS or SVN) and the corresponding change logs. It consists of four steps: the first step extracts licensing statements from files, the last three collect data to analyze how the statements changed. Without loss of generality, we limit our analysis to `.java` files for Java systems; `.h` and `.c` for C systems; and, `.h`, `.c`, and `.cpp` files for C++ systems. We also analyze changes occurring to copyright owners' names, as presented in another paper [3].

Step 1: Extracting licensing statements.

We extract the licensing statement of a file as its first two blocks of comments, where a block is a sequence of consecutive comments with no source code in between. We created our own comment extractor based on a comment-removal tool adapted to export comments instead of removing them[3]. We consider the first two blocks of comments because licensing statements are very often interleaved with `#include` directives, preprocessor macros, or package declarations.

Step 2: Identifying changes in licensing statements.

Licensing statements are usually English text, thus we cannot compare the licensing statements in two file revisions using a line differencing tool, such as `diff`. Therefore, we compare licensing statements of subsequent file revisions by indexing them using Information Retrieval Vector Space Models and by comparing their models using the cosine similarity [6]. In this step, we are interested in *any* change, therefore we use all alpha-numeric words, neither pruning stop words nor performing stemming.

[3]Our comment extractor can be downloaded from `http://turingmachine.org/~dmg/comments-1.0.tar.bz2`

Step 3: Classifying licenses.

As in our previous work [8], we detect the license(s) of each file (a licensing statement can contain multiple licenses) using the license identification tool in FoSSology 1.0.0 [11], which detects licenses using the Binary Symbolic Alignment Matrix (bSAM) pattern matching algorithm. For each file, we first classify the license(s) in its first revision. Then, we perform the classification every time the cosine between the licensing statements of two subsequent file revisions is less than 0.99. We choose 0.99 for two reasons: first, FoSSology is very slow (it might take more than a minute to analyze a source code file even on a fast computer); second, a manual inspection of the classifications shows that, for higher cosine values, changes did not affect the legal implications of the licensing statements.

Step 4: Identifying changes in copyright years.

We extract copyright years from licenses by mining numeric sequences of two or four digits, matching years between 1990 and 2009. Our heuristics can detect single years and year ranges, *e.g.*, 1998–2001, which we convert into a series of years: 1998, 1999, 2000, and 2001. We prune years automatically inserted by CVS and SVN (Id tags) or followed by time, which should not occur in copyrights.

4. EMPIRICAL STUDY

The *goal* of this study is to analyze licensing evolution, with the *purpose* of investigating how developers change licensing statements in source code files. The *quality focus* is related to the kind of changes occurring in licensing statements. The *perspective* is of researchers who want to gain insights on when and how licensing statements are changed to understand the relevance and impact of licensing changes. It is also of practitioners who want to realize the extent and importance of licensing evolution.

The *context* consists of the CVS or SVN repositories of six FOSS systems: ArgoUML, Eclipse-JDT, the FreeBSD and the OpenBSD kernels, Mozilla, and Samba. The systems have different sizes, are developed with different programming languages (C, C++, and Java), and belong to different domains: ArgoUML is a Java-based UML modeler; Eclipse-JDT an extensible development environment in and for Java; FreeBSD and OpenBSD are kernels of two open Unix operating systems in C/C++; Mozilla is a suite comprising a Web browser, an email client, and other Internet utilities in C/C++; Samba is a file and printer service inter-operating between Unix and Windows operating systems in C. Table 3 reports characteristics of the six systems, while Table 4 shows the distribution of the licenses in their first and last releases. Only Samba licensing statements never changed.

We choose systems different from those illustrated in Section 2.2 for three main reasons. First, this choice gives us is the possibility of analyzing long revisions histories, which are not available for systems such as MySQL or Java. Second, while the cases reported in Section 2.2 motivated our study by showing that licensing changes impact software usage, this study investigate the extent and relevancy of licensing changes, *i.e.*, whether they often occurs during a software lifetime. Third, this choice allows us to analyze large systems, such as the OpenBSD and FreeBSD kernels, Eclipse-JDT, and Mozilla, in which it is possible to observe a wide variety of licensing changes.

Table 3: Main characteristics of the six systems.

Characteristics	ArgoUML	Eclipse-JDT	FreeBSD	OpenBSD	Mozilla	Samba
Languages	Java	Java	C	C	C/C++	C
Release ranges	0.10–0.20	1.0–3.0	2.0–7.1	2.0–4.4	M3–1.7.13	1.9–3.0
#of source files ranges	777–1,421	578–3,274	895–6,729	3,359-6,483	4,845-12,436	299–860
KLOC ranges	129–280	79–697	325–3,292	994-2,242	1,827–4,104	156–332
CVS/SVN start dates	2000-09-14	2001-05-02	1993-06-12	1995-10-18	1998-03-28	1996-05-04
CVS/SVN end dates	2005-12-30	2006-11-07	2009-02-16	2009-02-07	2008-01-11	2004-04-03
Analyzed file revisions	32,582	128,611	195,077	110,430	468,747	29,018
# of committers	40	51	383	212	681	35

Table 4: Distributions of licenses in the first and last releases of each of the analyzed systems. Column f shows the number of files with such licenses and % the corresponding percentages. We only show licenses covering at least 5% of files.

Licenses		f	%
ArgoUML			
First	'Free with copyright clause'-style, 'UC Regents free with copyright clause'-style	735	94.6
	Others	42	5.4
Last	'Free with copyright clause'-style, 'UC Regents free with copyright clause'-style	1401	98.6
	Others	20	1.4
Eclipse-JDT			
First	None	579	100.0
Last	Eclipse Public License v1.0	4063	99.1
	Others	36	0.9
FreeBSD			
First	'BSD UCRegents'-style (4-cl. BSD)	522	58.3
	None	122	13.6
	'BSD UCRegents'-sty.,'CWI'-sty. (4-cl.BSD)	54	6.0
	Others	197	22.0
Last	'Cryptix'-style (2-cl. BSD)	1374	20.4
	'INRIA-OSL'-style (3-cl. BSD)	997	14.8
	BSD (unknown BSD)	813	12.1
	Others	3545	52.7
OpenBSD			
First	'BSD UCRegents'-style	2054	61.1
	None	486	14.5
	'Carnegie Mellon University 1991'-style	226	6.7
	Others	593	17.7
Last	'INRIA-OSL'-style (3-cl. BSD)	1495	23.1
	BSD (unknown BSD)	1145	17.7
	None	940	14.5
	Others	2903	44.8
Mozilla			
First	NPL	4430	91.4
	None	245	5.1
	Others	170	3.5
Last	'MPL v1.1'-style, Dual MPL GPL	6591	53.0
	'Dual MPL GPL'-style, 'MPL v1.1'-style	1881	15.1
	'Dual MPL GPL'-style, MPL	1826	14.7
	Others	2138	17.2
Samba			
First	GPL v2	247	82.6
	None	35	11.7
	LGPL, LGPL v2+	15	5.0
	Others	2	0.7
Last	GPL v2	606	70.5
	None	210	24.4
	Others	44	5.1

4.1 Research Questions

RQ1: *How frequently do the licensing statements of source files change?* This research question is preliminary to the following questions. It aims at providing overall, quantitative data on the frequencies with which licensing statements are modified by developers across file revisions. Also, it investigates whether such a frequency significantly differs among systems.

RQ2: *To what extent are files changing their licenses?* This research question investigates whether licensing evolution corresponds to moving to a completely different license, *e.g.*, from BSD towards GPL, or to adding a new license to a file already licensed, *e.g.*, from BSD towards a disjunctive BSD and GPL license.

RQ3: *How are copyright years changed in licensing statements?* Specifically, we investigate whether, when a file is changed in a given year, its copyright statement contains such a year. We also investigate if files reporting a particular year in their copyright undergo significantly more changes during that year than files that do not report it.

5. RESULTS

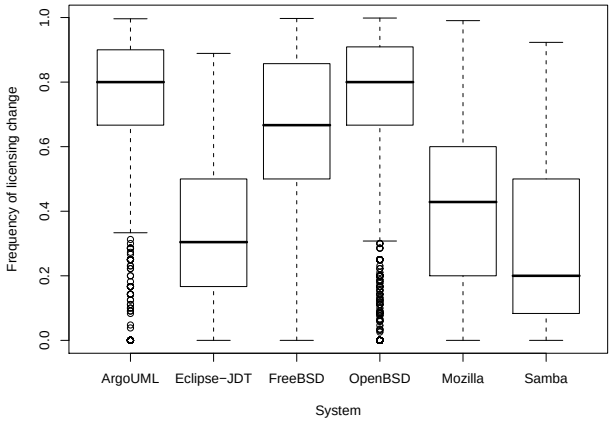
This section reports the results of our empirical study to answer the previous research questions. Data for verification and replication are available on-line[4].

5.1 RQ1: How frequently do the licensing statements of source files change?

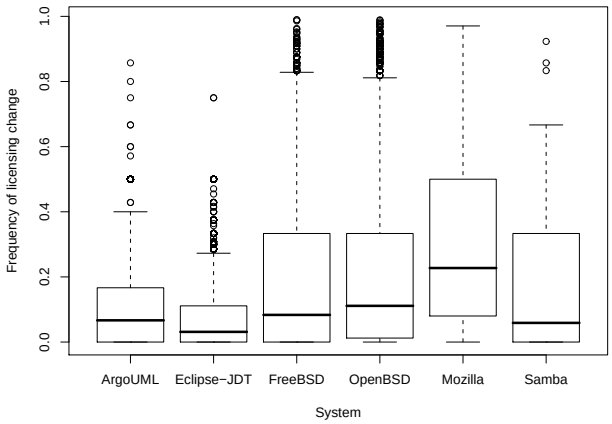
Figure 1 shows the box-plots of change frequencies for the licensing statements of the files belonging to the six systems, counted as the fraction of commits that involved changes in their licensing statements. Figure 1(a) shows the box-plots of change frequencies related to any change occurring in the licensing statements while Figure 1(b) only considers substantial changes, changes for which the similarity with respect to the previous file revision is below 0.99. The box-plots for changes below 0.99—the threshold used to trigger FoSSology classifications in RQ2—is very similar to Figure 1(a), thus it is not shown.

To statistically compare whether the change frequencies significantly differ among systems, we test the null-hypothesis H_0*: the average change frequencies among systems does not significantly differ*. Results related to all changes, in Figure 1(a), indicate that the change-proneness significantly differs among systems (p-value < 0.001 using Kruskal-Wallis test); OpenBSD, ArgoUML, and FreeBSD, having a higher licensing change-proneness than the other systems. In particular, OpenBSD and ArgoUML licensing statements change

[4] `http://www.rcost.unisannio.it/mdipenta/lic-rawdata.tgz`



(a) All changes



(b) Cosine <0.99

Figure 1: Frequency of licensing changes.

more than all the other systems (p-value < 0.001 using Mann-Whitney test), while there is no significant difference among them. When considering substantial changes, see Figure 1(b), results are different: there is still a significant difference among systems (p-value < 0.001) but Mozilla has the higher licensing change-proneness.

We conclude that licensing statements *do* change and that, therefore, it is interesting to study their changes in more details by answering the following research questions.

5.2 RQ2: To what extent are files changing their licenses?

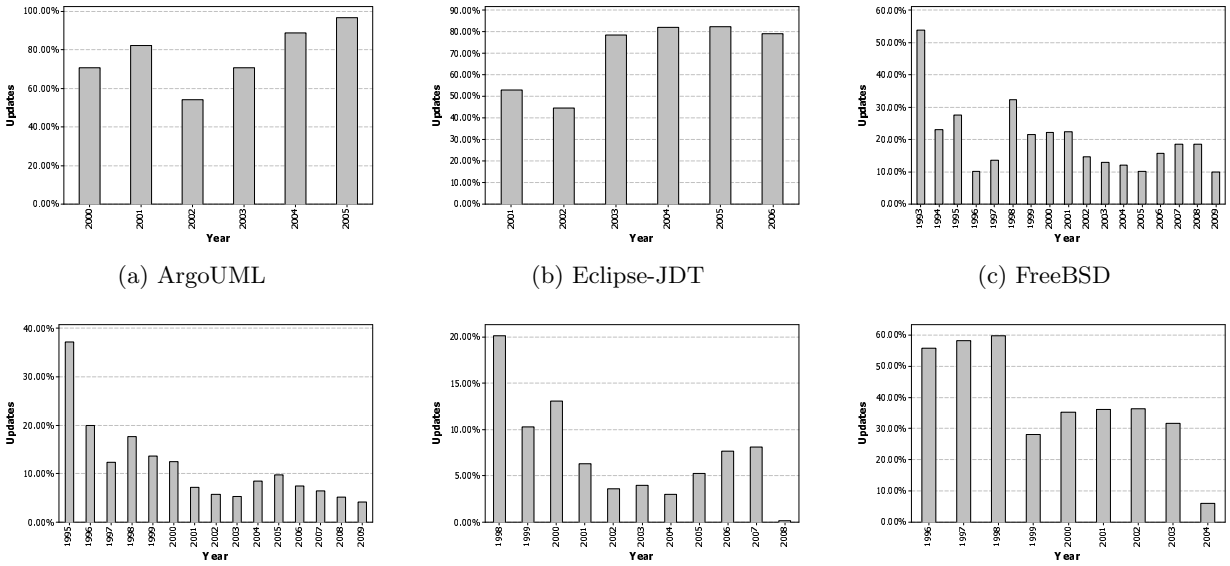
Table 5 shows the counts of the most frequent license changes occurring in the six systems. It shows that files that had their licenses changed between two revisions and excludes files that have had the same license(s) since their first revision. Each system has different changed patterns.

ArgoUML: Table 4 shows that the licenses of ArgoUML files are essentially the same in its first and last version. The results in Table 5 support this observation. There are only few files that changed their licenses from *None* to the *'UC Regents free with copyright clause'-style*, which is a permissive license that imposes very few constraints.

Eclipse-JDT: Table 4 shows that no file in its first version had a license. As time progressed, licenses have changed

Table 5: Changes of license types.

Licenses Transitions	#
ArgoUML	
None → 'Free with copyright clause'-style+'UC Regents free with copyright clause'-style	127
'LGPL GNU C Library'-style → 'Free with copyright clause'-style+'UC Regents free with copyright clause'-style	6
'Free with copyright clause'-style+'UC Regents free with copyright clause'-style → 'Free with copyright clause'-style	1
'Free with copyright clause'-style+'UC Regents free with copyright clause'-style → None	1
Eclipse-JDT	
Common Public License v1.0 → Eclipse Public License v1.0	2394
Common Public License v0.5 → Common Public License v1.0	808
None → Common Public License v1.0	692
None → Common Public License v0.5	588
Unknown → None	161
None → 'Common Public License v1.0'-style	76
None → Common Public License v1.0	55
Common Public License v0.5 → Common Public License v1.0	51
None → Eclipse Public License v1.0	30
Common Public License+Eclipse Public License → Eclipse Public License v1.0	20
Others	34
FreeBSD	
BSD UCRegents (4-cl BSD)→ 'BSD UCRegents'-style (4-cl BSD)	491
'BSD UCRegents'-style (4-cl BSD) → 'INRIA-OSL'-style (3-cl BSD)	300
GPL v2 → 'GPL v2'-style	114
'INRIA-OSL'-style (3-cl BSD) → 'Cryptix'-style (2-cl BSD)	68
None → 'Cryptix'-style (2-cl BSD)	68
'FreeBSD'-style (2-cl BSD)→ 'Cryptix'-style (2-cl BSD)	48
Unknown → CCDL	46
'CWI'-style+BSD UCRegents (4-cl BSD)→ 'BSD UCRegents'-style (4-cl BSD)+'CWI'-style	43
'INRIA-OSL'-style (3-cl BSD)→ 'FreeBSD'-style (2-cl BSD)	41
None → 'INRIA-OSL'-style (3-cl BSD)	35
'Cryptix'-style (2-cl BSD)→ BSD (Unknown BSD)	34
None → 'BSD UCRegents'-style (4-cl BSD)	33
None → 'FreeBSD'-style (2-cl BSD)	31
Others	695
OpenBSD	
'BSD UCRegents'-style (4-cl BSD)→ 'INRIA-OSL'-style (3-cl BSD)	964
BSD UCRegents (4-cl BSD)→ 'BSD UCRegents'-style (4-cl BSD)	414
'BSD UCRegents'-style (4-cl BSD)→ 'FreeBSD'-style (2-cl BSD)	262
BSD UCRegents (4-cl BSD)→ 'INRIA-OSL'-style (3-cl BSD)	210
'BSD UCRegents'-style (4-cl BSD)→ None	98
'BSD UCRegents'-style (4-cl BSD)→ BSD (Unknown BSD)	85
None → 'BSD UCRegents'-style (4-cl BSD)	83
BSD (Unknown BSD)→ 'FreeBSD'-style (2-cl BSD)	78
None → 'INRIA-OSL'-style (3-cl BSD)	43
'CWI'-style + BSD UCRegents (4-cl BSD)→ 'CWI'-style + 'INRIA-OSL'-style (3-cl BSD)	40
'BSD UCRegents'-style (4-cl BSD)→ 'BSD UCRegents'-style (4-cl BSD)+BSD (Unknown BSD)	37
Others	809
Mozilla	
NPL → 'NPL v1.1'-style+GPL v2+LGPL v2.1	2914
NPL → 'Dual MPL GPL'-style+MPL	1274
'Dual MPL GPL'-style+MPL → NPL	1194
GPL v2+MPL → 'Dual MPL GPL'-style+MPL	942
MPL → 'MPL v1.1'-style+Dual MPL GPL	908
NPL → 'MPL v1.1'-style+Dual MPL GPL	543
NPL → GPL v2+MPL	375
MPL → 'Dual MPL GPL'-style+'MPL v1.1'-style	361
NPL → GPL v2+LGPL v2.1+NPL	149
GPL+NPL → 'MPL v1.1'-style+Dual MPL GPL	148
GPL+NPL → 'Dual MPL GPL'-style+MPL	144
Others	1736
Samba	
None → GPL v2	15
GPL v2 → LGPL v2	2
GPL v2 → None	1
None → 'LGPL v2.0'-style	1
None → LGPL v2	1
Public Domain+GPL → None	1



(a) ArgoUML (b) Eclipse-JDT (c) FreeBSD

(d) OpenBSD (e) Mozilla (f) Samba

Figure 2: Percentage of files modified during a year for which the copyright year was updated.

from *None* to CPL v0.5, CPL v1.0, and, finally, EPL v1.0. The growing number of files updated from one license to the next reflects the growth in number of the files. This change pattern is consistent with Table 4: in the last analyzed release, almost all files were licensed under the EPL v1.0.

FreeBSD: as reported in [8], the files of FreeBSD and OpenBSD use a very large number of licenses (albeit most of them are variants of the BSD license, including the 2- and 3-clauses variants). This use is reflected in Table 4 and in the license changes in Table 5. We observe that the licenses are moving towards the less restrictive 2- and 3-clauses BSD licenses. FreeBSD itself is available under a 2-clauses BSD license.

OpenBSD: OpenBSD is a fork of FreeBSD; it is not surprising that its license changes are similar to that of FreeBSD. The major difference is that OpenBSD is available under a 3-clauses BSD license while FreeBSD under 2-clauses BSD. This difference is reflected in Tables 4 and 5, where OpenBSD has more transitions towards 3-clauses BSD licenses while FreeBSD has more towards 2-clauses BSD. There are also many files that changed from having a license to *None*.

Mozilla: Mozilla has seen a natural progression from NPL towards the current disjunctive GPL v2 and MPL v1.0[5]. It has moved from the original NPL v1.0 to the v1.1 (equivalent to the MPL v1.1) plus the GPL. We were surprised to also observe a change in the opposite direction: on Jan 19, 2001, 1,111 files were changed from GPL/MPL to NPL. This change was reverted few hours later. The corresponding defect (#98089) states that there was a bug in the script responsible for changing the license.

Samba: Table 4 shows that Samba has had almost no changes in its licenses; it has always been released under the GPL v2[6]. Compared to other projects, Samba had significantly fewer files that changed license. These changes are likely files that were originally inserted without a license and later fixed.

[5]Mozilla is currently licensed under a disjunctive license consisting of the GPL v2, the MPL v1.0, and the LGPL v2.1; FoSSology is not able to detect the LGPL in this case.

Table 6: Relationship between file changes and copyright year updates.

Systems	Changes to files with outdated copyright year			Changes to files with outdated copyright year			p-values	Eff. sizes
	Mean	Median	σ	Mean	Median	σ		
ArgoUML	3.3	2.0	3.7	4.9	3.0	6.0	< 0.01	0.3
Eclipse-JDT	3.8	2.0	6.1	6.0	3.0	8.4	< 0.01	0.3
FreeBSD	4.3	2.0	6.5	5.4	3.0	7.5	< 0.01	0.1
OpenBSD	2.6	1.0	4.9	4.7	2.0	8.3	< 0.01	0.5
Mozilla	4.7	2.0	10.2	4.7	2.0	10.2	< 0.01	0.0
Samba	5.8	2.0	12.4	12.6	7.0	17.1	< 0.01	0.4

5.3 RQ3: How are copyright years changed in licensing statements?

Figure 2 reports the percentages of file revisions, for each systems, where the files that underwent at least one change in a particular year also had a copyright year added or modified in their licensing statements.

ArgoUML and **Eclipse-JDT**—the two analyzed Java systems—started with a (relatively) low percentages of files with years updated, about 70% for ArgoUML and above 50% for Eclipse-JDT. Then, the percentages increased towards 80% for Eclipse-JDT and close to 100% for ArgoUML. **FreeBSD** and **OpenBSD**, overall, exhibit a lower upda-

[6]Samba has recently changed to GPL v3 but we retrieved its history before this change.

ting of copyright years than the two Java systems, with a relatively higher number of update in the first year, above 50% for FreeBSD, below 40% for OpenBSD.

Mozilla is similar to FreeBSD and OpenBSD but with a lower percentages of updates, starting from 20% in the first year and then decreasing towards 10%.

Samba has percentages of changes of copyright years higher in the first three years, above 50%, then lower in the subsequent years, below 40%.

We also assess whether files containing a year in the copyright underwent a higher number of changes in that year than other files. We test the null hypothesis: *H_0: the number of changes, during one year, for files reporting such a year in their copyright does not significantly differ from the number of changes occurring to other files.*

Table 6 shows descriptive statistics for the two groups of files, p-values resulting from the Mann-Whitney, two-tailed test used for the comparison, and the Cohen d effect size [2], indicating the magnitude of the difference. The effect size is defined, for independent samples, as the difference between the means, M_1 and M_2, divided by the pooled standard deviation σ of both groups $\sigma = \sqrt{(\sigma_1^2 + \sigma_2^2)/2}$, *i.e.*, $d = (M_1 - M_2)/\sigma$. It is considered small for d≥0.2, medium for d≥0.5 and large for d≥0.8. Results indicate that the difference is always statistically significant (p-value < 0.01), *i.e.*, files exhibiting a change in their copyright year underwent significantly higher number of changes during that year. The effect size is always small or even negligible, but for OpenBSD where it is medium. This analysis was repeated by counting lines added/removed instead of the commits and consistent results were found.

6. DISCUSSION

Changes to licensing statements account for a large proportion of the changes occurring to source code files. Our results show that, for three of the six systems (ArgoUML, FreeBSD, and OpenBSD), licensing statements changed in about 60% of the file revisions (median value), while for the other systems (Eclipse-JDT, Mozilla, and Samba), the median frequency is between 20% and 40%. However, small changes are most frequent and more substantial changes only occur on a small percentage of file revisions—below 20%. Yet, in some cases, we found that even relatively small changes meant a transition towards a different license (*e.g.*, a change from a CPL 1.0 to EPL 1.0 requires a change in only two words: *Common* to *Eclipse* and *cpl* to *epl*).

By relating what we discussed in Section 2 about license evolution with results shown in Table 5, we can observe that some of the analyzed systems have changed their licenses in ways similar to which these licenses have evolved. The IBM Public License evolved into the CPL v0.5, the CPL v1.0, and, finally, the EPL v1.0. Our results show how the licensing of Eclipse-JDT followed that evolution. The differences between the original IBM Public License and the EPL are minor and primarily divided in two areas: first, IBM has relinquished control of these licenses to the Eclipse Foundation; and, second, clarifications are made regarding patents and the ways in which the Eclipse Foundation accepts source code contributions [4].

Similarly, Mozilla has seen its licenses changing from the original NPL to a combination of MPL and the GPL. The NPL guaranteed to Netscape the right to distribute the source code under any condition and license, regardless of who contributed to the source code. This license allowed to release Netscape 6 as a proprietary system. The MPL removed this asymmetry between Netscape and other contributors, allowing to re-distribute the source code if it remained under the MPL. Mozilla's most recent changes highlight another important fact: the Mozilla Foundation is aware of the constraints imposed by incompatible software licenses. In an attempt to deal with this problem, they have licensed Mozilla source code under any of three licenses: the MPL v1.1, the GPL v2+, and the LGPL v2.1+. The user can select the license that best fits her purpose.

In both Eclipse-JDT and Mozilla, licensing changes rigorously followed the evolution of the licenses. This is not surprising: the CPL and the EPL were created for the Eclipse Foundations's projects; the NPL and MPL for Mozilla Foundation's projects. In both cases, the actual changes to the license are minor and answer concerns of the communities of each of the Foundations.

FreeBSD and OpenBSD are more eclectic in their license changes. This is probably the result of a very heterogeneous community of contributors, and the result of code that is frequently imported from external sources (as we reported in [8]). Nonetheless, we can see a clear pattern moving from the old BSD-4 clauses license to the more permissive BSD-3 and BSD-2 licenses.

Some other systems—such as ArgoUML and Samba—have kept the same licenses over the entire analyzed time span. In the few cases in which the licensing statement of files changed in these systems, licensing statements were added to the files, *i.e.*, a change from *None* to the usual license. The implications of the changes in licenses for each of the projects is different. In ArgoUML, the change is from *None* to a simple license. We presume that at some point its authors realized the importance of including a license.

The numbers of times that systems have changed their licenses varies. We observed that Eclipse-JDT has used four different licenses and Mozilla two. These numbers show the willingness of the Eclipse and Mozilla Foundations to adapt the licensing statement of their systems to their users' needs.

Changes in copyright years followed different patterns in different systems: in ArgoUML and Eclipse-JDT, years were almost always updated, with increasing updating percentages, reaching values above 80%. We found lower and decreasing percentages in the other systems: the percentages are higher during the first year when the files are created with licensing statements. Also, we found that files for which the copyright years were updated underwent higher numbers of changes: when developers perform substantial changes to a file, they also update copyright years. Copyright regulations (and in particular, the copyright regulations of the USA) require the copyright notice to include the year of the publication of new code. Although this requirement is not mandatory to grant copyright protection, failing to properly update the copyright year when substantial changes occur would allow an infringer to claim "innocent infringement".

In some systems, Eclipse-JDT for example, it is possible to notice commits explicitly targeted to copyright years updates. For example, the commit done on 2003-03-12, 12:22:01 by *dmegert* says "Updated copyrights", or the commit done on 2004-01-13, 15:48:41 by *jlanneluc* says "Updated copyrights to 2004".

6.1 Threats to Validity

This section discusses the main threats to the validity of our study. *Construct validity* threats concern the relation between the treatment and the outcome. They can be due to our measurements, *i.e.*, the way we extracted licenses, classify them, and identify their changes. We extracted licenses using an already-existing approach [8, 10]. We have considered the first two blocks of consecutive comments in a file as its licensing statement to reduce the risk of missing a license. It can be therefore possible that we included in our study comments not belonging to a license. However, such a measurement error would only affect RQ1. It would not affect the results of the other questions, because the classification of the license is unlikely to be affected by more text than the license itself. As reported in [11], the license classification performed by FoSSology has some imprecision, particularly in complex licensing statements, *e.g.*, it did not detect the LGPL in the disjunctive license of Mozilla. Nevertheless, from the inspections we made, such an imprecision is mostly limited to discern among variants of licenses embedded in the licensing statement, *e.g.*, among BSD licenses. We have added new licenses [8] and submitted defect reports to FoSSology to improve its classification. Our experience with FoSSology is positive but needs to be empirically evaluated to properly assess its accuracy. We could not find cases where a license dramatically changed due to small textual changes: a manual validation shows that that any change in the license types—even the transition from CPL to EPL—yields a textual similarity between the previous and the new licensing statement lower than the threshold of 0.99.

Threats to *internal validity* do not affect this study, being an exploratory study [22]. For the same reason, threats to *conclusion validity* are also not important, although we used statistical tests where appropriate and made sure that the conditions for their applicability held.

Threats to *external validity* are related to the generalizability of our findings. Our study concerns a reasonably large variety of six systems, developed in different programming languages, belonging to different domains, and experiencing different kinds of evolutions, *e.g.*, systems developed from scratch (ArgoUML and Samba) and others that originated in industry (Eclipse-JDT and Mozilla). Yet, it is necessary to replicate this study on other systems, in particular industrial systems, to confirm its generalizability and to study the use of FOSS code in industrial systems.

Regarding *reliability validity*, *i.e.*, the possibility of replicating this study, we have detailed the data extraction process and the source code and changes for the six systems are available from their CVS/SVN repositories. Furthermore, we made available the extracted data and tools used in this study.

7. RELATED WORK

To the best of our knowledge, only few recent works specifically dealt with licensing evolution. This section discusses related work concerning the analysis of licenses and the evolution of source code comments, because licensing statements are a particular kind of comments.

Licenses impose constraints and thus can be defined as logical formulae constraining what can and cannot be done with a system. Software licensing patterns have been recently studied by German *et al.* [10] using such a formalization of licenses. They introduced several legal patterns, along with examples of occurrences of these patterns. In a previous work [8], we presented a study of the influence of software licenses on code migration between the FreeBSD, Linux, and OpenBSD kernels. Our findings support the hypothesis of a preferential code flow induced by permissive licenses from FreeBSD and OpenBSD towards Linux.

Because licenses are contained in licensing statements, the analysis of licensing evolution or of comment evolution are similar in their approaches and tools. Fluri *et al.* [5] investigated comments and code co-evolution by analyzing three FOSS systems (ArgoUML, Eclipse-JDT, and Azureus). They found that (i) new code is not much commented, (ii) most of the comments refer to class and method declarations, and (iii) comments are consistently updated with their associated code. Jiang and Hassan [14] examined the evolution of comments in PostgreSQL considering comment additions and deletions. They found that, on average, the percentage of commented functions remain constant except for some variations due to developers' commenting style. Lawrie *et al.* [16] measured the quality of identifiers by measuring the extent of their relations to words occurring in comments. They found that, in general, full identifiers ensure better comprehension than abbreviation, although there are exceptions. Ying *et al.* [23] studied the use of comments as a means of communication among Eclipse developers. They discovered that comments are not only used to help understanding the code but also to communicate tasks, activities, and to assign tasks to other team members. Hindle *et al.* [13] discovered that many of the largest commits correspond to changes to the licenses or copyright owners of files. We share with these previous works the heuristics used to identify comments and to compare them using information retrieval methods, although we specifically focus on comments related to licenses, *i.e.*, on licensing evolution and changes to copyright years. Finally, a related paper by Di Penta and German [3] studied changes occurring to copyright owners' names and found that explicit contributors and copyright owners are often added to licensing statement during larger changes, although the number of changes they performed is not higher than that of other committers.

8. CONCLUSION

As several cases occurring in the open-source world suggest, licensing changes can have an impact on the software lifetime, or even on other, competitor, software systems. This paper proposed a method to track the evolution of software licensing and investigated the relevance of licensing evolution on six FOSS systems. Most noticeably, we observed license changes, from one license to another, license additions, *e.g.*, files without license were updated with a license, and license modifications. For large systems like Eclipse-JDT, Mozilla, or the BSD kernels, the amount and frequency of licensing changes would make difficult their manual analysis, thus highlighting the usefulness of an automatic analysis method. Finally, we investigated changes occurring to copyright years and found that they are updated to protect new code when substantial changes are made to a source code file. We consequently brought evidence on the evolution of software licensing, which suggests that this field of analysis should be further studied to better understand its impact and rationale.

Future works will be devoted to study the licensing evolu-

tion in entire software distributions, with the aim of relating changes in licensing of some components/products with their removal in subsequent versions of the distribution, and with the adoption of alternative products (as in the cases of IP-Filter or QT). We will also investigate on a very large sample of FOSS systems the evolution of their licensing statements, *i.e.*, how different kinds of FOSS licenses are adapted over the time to cope with weaknesses or limitations of older licenses, and to characterize licensing evolution patterns in different categories of software systems.

9. ACKNOWLEDGEMENTS

We would like to thank Chris Nadan for valuable comments on early versions of this paper and on the impact of copyright year changes. This work has been partly funded by the NSERC Research Chairs in Software Change and Evolution and in Software Patterns and Patterns of Software. The work of D.M.German has been funded by Hewlett-Packard to support the FOSSology Project.

10. REFERENCES

[1] American Bar Association. An overview of "open source" software licenses. www.abanet.org/intelprop/opensource.html. Accessed Sept. 2009.

[2] J. Cohen. *Statistical power analysis for the behavioral sciences (2nd ed.)*. Lawrence Earlbaum Associates, Hillsdale, NJ, 1988.

[3] M. Di Penta and D. M. Germán. Who are source code contributors and how do they change? In *16th Working Conference on Reverse Engineering, WCRE 2009, 13-16 October 2009, Lille, France*, pages 11–20. IEEE Computer Society, 2009.

[4] Eclipse Foundation. Eclipse Public License (EPL) Frequently Asked Questions, 2007. Accessed Dec. 2007.

[5] B. Fluri, M. Würsch, and H. Gall. Do code and comments co-evolve? on the relation between source code and comment changes. In *14th Working Conference on Reverse Engineering (WCRE 2007)*, pages 70–79, 2007.

[6] W. B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[7] Free Software Foundation. Various Licenses and Comments about Them. www.gnu.org/philosophy/license-list.html, 2009.

[8] D. M. Germán, M. Di Penta, Y.-G. Guéhéneuc, and G. Antoniol. Code siblings: Technical and legal implications. In *Proc. of the 2009 Working Conference on Mining Software Repositories, MSR 2009*, pages 81–90, 2009.

[9] D. M. Germán and J. M. González-Barahona. An empirical study of the reuse of software licensed under the GNU General Public License. In *Proceedings of the International Open Source Systems Conference (OSS'09)*, pages 185–198. Springer, 2009.

[10] D. M. Germán and A. E. Hassan. License integration patterns: Addressing license mismatches in component-based development. In *31st International Conference on Software Engineering, ICSE 2009, May 16-24, 2009, Vancouver, Canada, Proceedings*, pages 188–198. IEEE, 2009.

[11] R. Gobeille. The FOSSology project. In *In Proc. Fith International Workshop on Mining Software Repositories*, pages 47–50, 2008.

[12] D. Hartmeier. Design and Performance of the OpenBSD Stateful Packet Filter. www.benzedrine.cx/pf-paper.html. Accessed Sept. 2009.

[13] A. Hindle, D. M. Germán, and R. Holt. What do large commits tell us? a taxonomical study of large commits. In *MSR '08: Proceedings of the 2008 international working conference on Mining software repositories*, pages 99–108, May 2008.

[14] Z. M. Jiang and A. E. Hassan. Examining the evolution of code comments in PostgreSQL. In *Proceedings of the 2006 International Workshop on Mining Software Repositories, MSR 2006*, pages 179–180, 2006.

[15] C. Knudsen. Troll Tech's QPL. *Linux J.*, (61es):10, May 1999.

[16] D. Lawrie, C. Morrell, H. Feild, and D. Binkley. What's in a name? a study of identifiers. In *14th Intl. Conference on Program Comprehension (ICPC 2006)*, pages 3–12, 2006.

[17] T. B. Lee. Court: violating copyleft = copyright infringement. http://arstechnica.com/tech-policy/news/2008/08/court-violating-copyleft-copyright-infringement.ars. Accessed Sept. 2009.

[18] Novell Inc. FAQ: Licensing. www.mono-project.com/Licensing, 2009.

[19] L. Rosen. *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall, 2004.

[20] S. Shankland. CNET News: Open-source spat spurs software change. http://news.cnet.com/2100-1001-266399.html. Accessed Sept. 2009.

[21] S. Shankland. CNET News: Ximian changes open-source license. http://news.cnet.com/Ximian-changes-open-source-license/2100-1016_3-823734.html. Accessed Sept. 2009.

[22] R. K. Yin. *Case Study Research: Design and Methods - Third Edition*. SAGE Publications, London, 2002.

[23] A. T. T. Ying, J. L. Wright, and S. Abrams. Source code that talks: an exploration of Eclipse task comments and their implication to repository mining. In *Proceedings of the 2005 International Workshop on Mining Software Repositories, MSR 2005, Saint Louis, Missouri, USA, May 17, 2005*. ACM, 2005.