Evolutional Analysis of Licenses in FOSS

Yuki Manabe
Osaka University
Osaka, Japan
y-manabe@ist.osaka-u.ac.jp

Yasuhiro Hayase
Toyo University
Saitama, Japan
hayase@toyo.jp

Katuro Inoue
Osaka University
Osaka, Japan
inoue@ist.osaka-u.ac.jp

ABSTRACT

FOSS (Free and Open Source System) is repeatedly modified and reused by other FOSS or proprietary software systems. They are released to others under specific licenses whose terms and conditions are usually written on the source-code files as program comments. There are a few researches which automatically analyze the licenses in a FOSS release, but there is no statistical study on the evolution of licenses along the evolution of FOSS. In this paper, we analyze licenses through FreeBSD, OpenBSD, Eclipse, and ArgoUML evolution, using our license analysis tool Ninka, and discuss characteristics on the evolution of the license used in those systems.

Categories and Subject Descriptors

D.2.7 [Software Engineering]: Distribution, Maintenance, and Enhancement—Version Control; D.2.9 [Software Engineering]: Managements—Copyrights

General Terms

Measurement, Experimentation

Keywords

Software License, Repository Mining

1. INTRODUCTION

Licensing is one of the ways to protect intellectual property of FOSS (Free and Open Source Software). Open Source Initiative[1] approved 66 licenses, which are commonly used by major FOSS.

Developers have to read the licenses carefully, which are generally written on each source-code file as program comment, and to understand and follow the written requirements and constraints. One of serious risks in using FOSS is to integrate thousands of FOSS files where the developer can hardly check the license of each file[8].

There are several tools which automatically analyze and report the licenses of source-code files. There is a license detection tool named Ninka, which can analyze the files more efficiently and accurately than other tools[6].

Using those tools, there are a few papers (or studies) on the analysis of licenses in FOSS[2, 4, 6, 7]. However, there is no study of license evolution for specific evolving FOSS.

A large change of license in software affects reusability of software. Some people know about a large change of license in software. However, there is no research of quantitative analysis of a large change of license in software. So, with software evolution analysis, it is useful for improving reusability of FOSS to examine what often such license change, how the change occurs and what information we can predict the change with.

As a first step to a large license change research, this paper reports the analysis result of large-scale FOSS with respect to the evolution of those FOSS. We analyzed each release of FreeBSD, OpenBSD, Eclipse and ArgoUML. Here we mainly focus on the ratio of licenses used in each release of those systems.

The major findings are as follows.

- The number of different licenses used in the operating systems such as FreeBSD and OpenBSD are larger than those used in specific applications such as Eclipse and ArgoUML.
- Licenses can sometimes drastically change in a release version.
- License ratios in the kernel files of those operating systems are similar to those of the overall operating systems.
- FreeBSD and OpenBSD kernels contain files with GPLv2 license, which are located separately from others.

In Section 2, we describe the background knowledge used in this paper. Section 3 sets the research questions for the analysis of license evolution. In Section 4, we show our analysis of results for four major FOSS, and discuss their threats to validity in Section 5. Section 6 shows the related works and Section 7 concludes our discussion with a few remarks.

2. SOFTWARE LICENSE AND NINKA

Software license (or simply license) means a set of directions to software users, which are set up by the software

[1] http://www.opensource.org/

IWPSE-EVOL '10 September 20-21, 2010 Antwerp, Belgium

author. We list the names and their abbreviation names of licenses used in this paper in Table 1. Many of these licenses have several versions. In that case we use the suffix v<number> to identify it. If it is followed by +, that means the user can choose this version or any newer: "or later". For example, GPLv2+ means "GPL version 2 or later". This paper does not argue about the legal issues of those directions. A source-code file contains comments and program code. Comments are used for the explanation of program code, license direction, or other purposes. The license direction contained in a file would generally consist of several natural-language sentences.

Software license detection tool Ninka identifies each English sentence in the leading comment of a source-code file, and performs the lower-level pattern matching against the meta-license statement collection of well-used licenses. From this matching result of every license-related sentence in the comments, then, Ninka performs the higher-level pattern matching against the sentence-arrangement pattern collection of the well-used licenses, and it determines the license. This two-phased matching approach is flexible to identify minor variation of license description, and it also allows extending to new licenses. Currently, Ninka has 427 meta-license statements and 126 sentence-arrangement patterns in its license-knowledge database. Ninka can identify 110 different licenses with 93% accuracy, and it can handle more than 600 files per minute[6].

License ratio means the ratio of the files with specific licenses against the total number of overall files in the target.

Abbrev.	Name
Apache	Apache Public License
BSD4	Original BSD, also known as BSD with 4 clauses
BSD3	BSD4 minus advertisement clause
BSD2	BSD3 minus endorsement clause
CPL	Common Public License
CDDLic	Common Development and Distribution License
EPL	Eclipse Public License
GPL	General Public License
LesserGPL	Lesser General Public License (successor of the Library GPL, also known as LGPL)
LibraryGPL	Library General Public License (also known as LGPL)
MIT/X11	Original license of X11 released by the MIT
MITold	License similar to the MIT/X11, but with different wording

Table 1: Names of common open source licenses and their abbreviations used in this article.

3. RESEARCH QUESTIONS

We set up the following three research questions to examine the license ratios in detail.

RQ1: How the license ratios of operating systems are different from the one of non-operating systems?

RQ2: What are the evolutional patterns of license ratios of the operating systems?

RQ3: Are the license rations of the kernels of operating systems different from the other parts of those systems?

4. EMPIRICAL RESEARCHES

We have conducted an experiment for these research questions. For the experiment, we have used four major FOSS, FreeBSD, OpenBSD, Eclipse and ArgoUML. Table 2 shows their characteristics. Note that we have used the entire module under the Eclipse platform project, as Eclipse. Also we have used the base system of FreeBSD and OpenBSD, as FreeBSD and OpenBSD, respectively. The reasons why we use these system are that they are used in[2], they has many release version and we can examine the difference between an application consisted of simple application and a number of application.

4.1 RQ1

At first, we have found each release version of those systems, by reading the logs stored in the version control systems and the release information in the web site of those systems.

Secondly, we have downloaded the source tree in each release version. Here, we used files written in C (*.c), C++(*.cpp, *.c++, *.cxx, *.cc), Java(*.java), Python(*.py), Perl(*.pl, *,pm), Emacs lisp(*.el) and Sawfish scripting language(*.jl).

Finally, we have identified the licenses of each release of those systems, by using Ninka. Ninka reports the identified license for each file, say, BSD4 or GPLv2+. Ninka also says NONE (the file includes no license-related sentences) and UNKNOWN (the file includes the license-related sentences but it failed to match them to the sentence-arrangement patterns).

We have repeated second and third process for each release version until we have obtained all needed source-code files of each release.

Figures $1 \sim 4$ show the analysis result of FreeBSD, OpenBSD, Eclipse, and ArgoUML respectively, over the evolution of each release. In each graph, X-axis means release version, y-axis means the number of files. Each layer corresponds to each license. We have chosen and presented 5 most popular licenses in those Figures.

Interesting findings here are as follows.

- In the case of FreeBSD (Figure 1), BSD4 license decreases over the evolution. On the other hand, BSD2 and BSD 3 increase along the evolution. This is due to the policy change of the Berkeley distributor such that no more acknowledgements were needed within advertising materials and they started to change licenses in the new files. So, this change relaxes the condition of the license.

- The same tendency can be seen in the case of OpenBSD (Figure 2). In this case, there are many NONE files. This is because they use COPYING file of GPLv2+ to indicate licenses, instead of adding license statement to each file.

- In Eclipse (Figure 3), majority of known licenses has changed from CPLv0.5 to CPLv1, and then to EPLv1.0. These changes appear very sharply in Figure 3. On the other hand, above mentioned changes of FreeBSD and OpenBSD would be rather very vague and older licenses stay longer. This is due to the difference of development management. FreeBSD and OpenBSD have been developed with many contributed source-code files, and Eclipse has been developed with more

	FreeBSD	OpenBSD	Eclipse	ArgoUML
Release Version	2.2-8.0	2.0-4.7	2.0-3.5.2	0.8.1-0.31.1
Release Date	1994/11-2009/11	1996/10-2010/5	2002/6-2009/9	2000/10-2010/6
Type	OS kernels, drivers and compilers	OS kernels, drivers and compilers	SDE platform	UML Design Tools
# releases	45	28	25	79
# Files(oldest-latest)	6245-14181	4412-21266	11419-35880	686-2208
VCS	CVS	CVS	CVS	Subversion

Table 2: Characteristics of analyzed application in this study

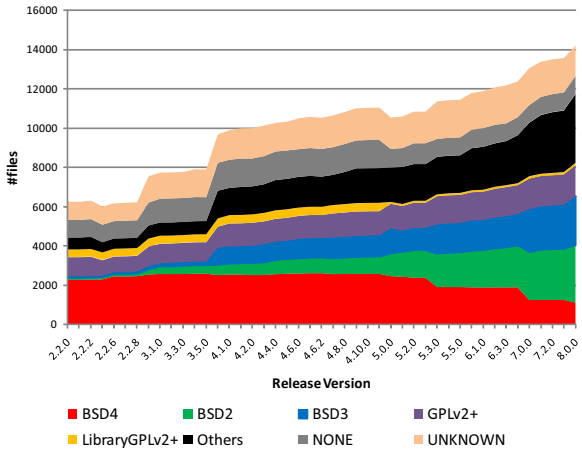

Figure 1: the license ratio of FreeBSD

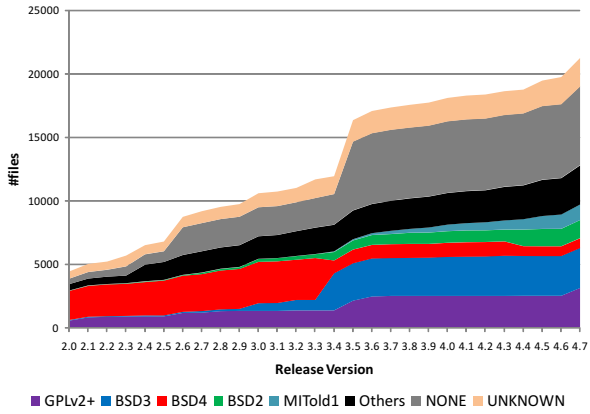

Figure 2: the license ratio of OpenBSD

strict control in accordance with Eclipse Development Process[3]. The difference between CPLv1.0 and EPLv1.0 is removing most of conditions on patent. So, this change relaxes the condition of the license.

- In ArgoUML (Figure 4), there are many UNKNOWN files through the evolution. Further analysis showed that those are mostly BSD-like license, but not the same as commonly used ones such as BSD4, BSD3, and BSD2. By adding new meta-sentences and higher-level matching pattern to the database, Ninka can now identify them. However, we have not sufficient time to re-run this experimentation. So, we show original data. BSD license has not copy-left. However, EPL has copy-left which regard source file under EPL modified or added as derivative software. So, this change which showed in this case tightens the condition of the license.

Answer for RQ1: Licenses in the operating systems (FreeBSD, OpenBSD) are rather diverse and loosely controlled, compared to non-OS systems (Eclipse, ArgoUML). A few licenses cover almost all files in those systems, and sometimes those licenses are drastically changed to others by the strong management to the overall system. And, the change relaxes or tightens the condition of the license.

4.2 RQ2

Based on the result for RQ1, We examined the changes of the number of files under each license.

We calculated the difference of the number of files between a version and its next version. In this analysis, we did not count files categorized as NONE, UNKNOWN and Other.

Figure 5 shows the changes of the number of FreeBSD files and Figure 6 presents that of OpenBSD. In these graphs, X-axis represents release version and Y-axis represents the difference of the number of files between a version and its previous version. Some of the findings are as follows.

- In the case of FreeBSD, the reduction of BSD4 happened periodically, and BSD2 and BSD3 have increased in response to such reductions. A detailed analysis of Release Version 5.2.1 to 5.3 shows that total 531 BSD4 files have been changed to 423 BSD3 files, 13 BSD2 files, and 95 obsolete files.
- Also in the case of FreeBSD, GPLV2+ and LibraryGPLv2+ were deleted and BSD2 $\sim$ 4 are added.
- In the case of OpenBSD, the reduction of BSD4 is compensated with GPLv2+ and MITold1. A detailed analysis shows that from Release Version 3.3 to 3.4, total 2255 files have been changed to 1957 BSD3 files, 271 BSD2 files, and 27 obsolete files (KerberosV files).

Answer to RQ2: Detail analysis of the evolution of licenses in these two operating systems shows that there are periodical large shifts of licenses, along with system evolution.

4.3 RQ3

In section 4.1, we have shown existence of various licenses in the operating systems. We do not know whether those

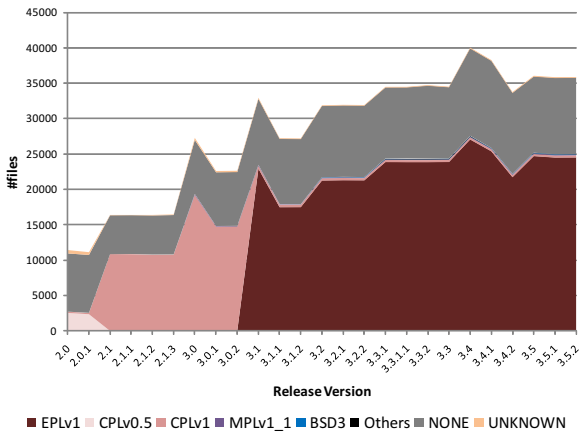

Figure 3: the license ratio of Eclipse

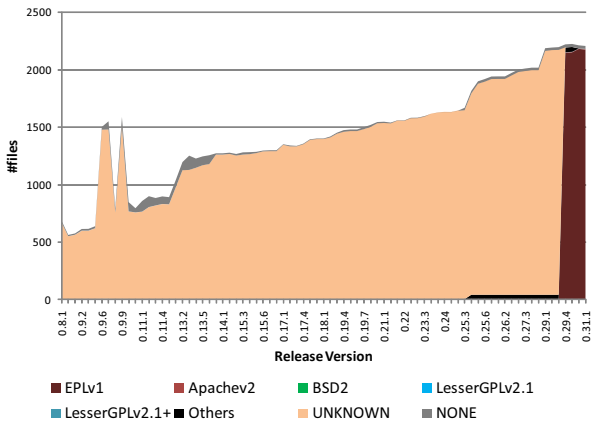

Figure 4: the license ratio of ArgoUML

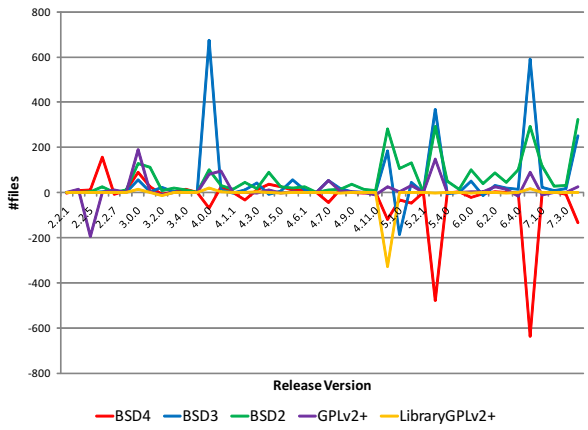

Figure 5: the changes of the license ratio of FreeBSD

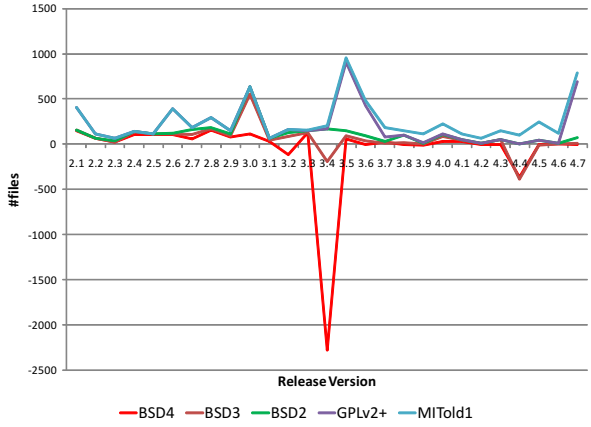

Figure 6: the changes of the license ratio of OpenBSD

different licenses appear in the kernel part of those operating systems. The kernels have been developed generally in a well- controlled environment and it is supposed that there exist fewer licenses with drastic change pattern. We have applied the approach used in the RQ1 to the source files of both OpenBSD kernel and FreeBSD kernel. Those are about 10% $\sim$ 25% files of the total operating systems.

Figure 7 shows the license ratio of kernel of FreeBSD and figure 8 shows OpenBSD.

Findings are as follows.

- Both kernels include various licenses with similar pattern as the overall operating systems.
- Also similar to the operating systems, the kernel licenses have been changed continuously and periodically. The patterns of Figure 5 and 6 are more similar to those of Figure 1 and 2, and not similar to Figure 3 and 4.

Answer to RQ3: Kernel part of the operating systems have similar evolutional pattern of their licenses to the overall operating systems.

5. THREATS TO VALIDITY

One of threats to validity is the accuracy of the result from Ninka. Daniel et.al.[6] reported that Ninka's accuracy is 93%. We think that this is sufficiently high, so that the identification errors of Ninka do not seem to affect the license rations discussed in Section 4.

Other threat to validity is selection of applications. We use FreeBSD and OpenBSD as the targets of the operating systems. Although they have the same root, 386 BSD, they started to go differently ten years ago (FreeBSD started in 1991 and OpenBSD started in 1995.) with different development policies. Therefore, we think that they are related but different operating systems with different license policies.

We use four systems. This is too small to make generalize the result of this empirical study. To regard this result as general result, we must apply similar empirical study to many other samples.

6. RELATED WORKS

Di Penta et.al.[2] proposed a method to track changes in the license terms. It showed that license is changed frequently and many times. In this research, they analyzed each revision of each file and identify many patterns of li-

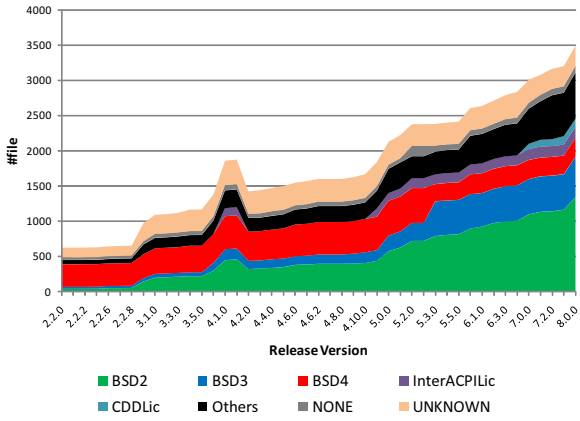

Figure 7: the license ratio of kernel of FreeBSD

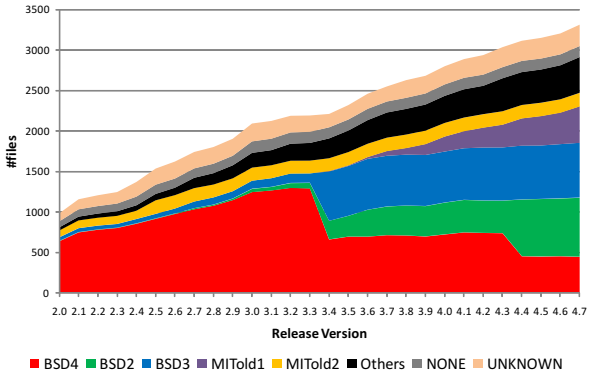

Figure 8: the license ratio of kernel of OpenBSD

cense changes between first version and last version in target releases. On the other hand, our work shows the change of license among releases of each system. The results show that there are large-scale change between two releases and two licenses. And, their work concludes that files of ArgoUML are not changed very much. However, our work shows that in newer release, a large scale change occurs. In our research, we have successfully shown that such pattern occurs through the evolution and the license ratio changes sometimes drastically or gradually based on the distinction of FOSS types. On the other hand, though their work has examined a change of year included in copy right description, our work has not examined.

Some previous works deal with license ratio. Gobeille[7] introduce FOSSology which identifies each license of source files with bSAM algorithm. He has applied it to abiword-2.6.4 as an example. Black Duck Software[2] reported daily top 20 most commonly used licenses in open source projects according to the Black Duck software knowledgebase. Our research is different from these works in the sense that we focus on large scale software systems with many release versions.

Also, some previous works deal with license mismatch. Alspaugh et.al.[1] introduced a method which describes terms of license as a tuple ⟨actor, operation, action, object⟩ and calculates propagation and conflict of duty. German et.al.[5] have formally defined software licenses and their patterns to resolve license mismatch. German et.al. [4] have also introduced a method to help to understand license mismatch occurred in software packages and to audit license mismatch problem in binary packages of Fedora 12. Our research did not deal with license mismatch. However, the result of our work may emphasize the risk of license mismatch and the importance of identifying licenses.

[2]http://www.blackducksoftware.com/oss/licenses#top20

7. CONCLUSIONS

This paper presents the evolutional pattern of licenses in the source-code files of large-scale FOSS. By this license analysis, we found that the evolutional patterns of the operating systems are different from those of non-operating systems such as Eclipse and ArgoUML. The changes of licenses in those operating systems occur periodically and continuously, and such pattern can be also seen in the kernel evolution.

As a future work, at first we apply similar empirical study to many other systems. Then, we are planning to conduct more fine-grained analysis for changes. Especially, we will examine the cases of drastic change between two neighbor versions, e.g., FreeBSD Release Version 5.2.1 and 5.3, and OpenBSD Release Version 5.3 and 5.4.

8. ACKNOWLEDGMENTS

This research was supported by Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (A) (No.21240002). This research was supported in part by "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

9. REFERENCES

[1] T. A. Alspaugh, H. U. Asuncion, and W. Scacchi. Analyzing software licenses in open architecture software systems. In Proc. FLOSS 2009, pages 54–57, Washington, DC, USA, 2009.

[2] M. Di Penta, D. M. German, Y.-G. Guéhéneuc, and G. Antoniol. An exploratory study of the evolution of software licensing. In Proc. ICSE 2010, Cape Town, South Africa, 2010.

[3] Eclipse Foundation. Eclipse Development Process. http://www.eclipse.org/projects/dev_process/development_process.php, 2010. Accessed June. 2010.

[4] D. M. German, M. Di Penta, and J. Davies. Understanding and auditing the licensing of open source software distributions. In Proc. ICPC 2010, pages 84–93, Braga, Portugal, 2010.

[5] D. M. German and A. E. Hassan. License integration patterns: Addressing license mismatches in component-based development. In Proc. ICSE 2009, pages 188–198, 2009.

[6] D. M. German, Y. Manabe, and K. Inoue. A sentence-matching method for automatic license identification of source code files. In Proc. ASE 2010, 2010. (To appear).

[7] R. Gobeille. The FOSSology project. In Proc. MSR 2008, pages 47–50, New York, NY, USA, 2008. ACM.

[8] C. Ruffin and C. Ebert. Using open source software in product development: a primer. IEEE Software, 21(1):82–86, 2004.