Automatic Checking of License Compliance

Hongyu Zhang, Bei Shi
School of Software
Tsinghua University
Beijing 100084, China
hongyu@tsinghua.edu.cn, shib07@mails.tsinghua.edu.cn

Lu Zhang
Key Laboratory of High Confidence Software Technologies
Peking University
Beijing 100871, China
zhanglu@sei.pku.edu.cn

Abstract— **Open source software facilitates software reuse as developers can learn from code in existing open source projects. However, license compliance is an important legal issue that should be taken into consideration during open source based software reuse. Ignorance or carelessness could result in huge financial losses. In this paper, we present LChecker, a tool we developed for automatic checking of license compliance. LChecker utilizes Google Code Search service to check whether a local file exists in an OSS project and whether the licenses are compatible. The initial experimental results show that our tool is effective in detecting license compliance problems[1].**

Keywords- license compliance; open source software; code search

I. INTRODUCTION

The open source software (OSS) movement has gained momentum in recent years. A large number of OSS projects are freely available for public downloads. OSS improves software reusability and reduces cost as software developers can learn from source code of existing OSS projects.

However, there are legal issues surrounding OSS-based software reuse. OSS projects often include a license, which specifies permissions and restrictions for using the software. Examples of these licenses include GPL, LGPL, Apache, etc. Some licenses, such as GPL, are copyleft license, meaning that derived works can only be distributed under the same license terms [1]. This restriction implies that if a commercial project uses code from a GPL-licensed OSS, it must open its source code for public and be distributed under GPL license. Otherwise, the organization that developed the commercial product could face legal consequences. Furthermore, licenses may be incompatible with each other - one license (such as GPL) may conflict with other licenses (such as BSD).

Due to the lack of educations on legal aspects of software engineering, or simply due to carelessness, some developers could easily make a license compliance mistake, which may cause a huge potential financial loss to the organization. Therefore, license compliance is becoming an important issue for many software organizations.

To help software organizations check license compliance, we develop a tool called LChecker, which can automatically scan program files in a software project, and check whether the files already exist in the Internet and whether the licenses are compatible. We evaluate LChecker using real projects and the results are promising. Many previously-unknown license compliance problems have been discovered successfully. We also show how LChecker can be used to detect changes in licenses during software evolution.

II. THE DESIGN OF LCHECKER

In this section, we present LChecker, a tool for automatic checking of license compliance. We first introduce the overall structure, and then describe some major technical points in details.

A. Overall Structure

Figure 1 shows an overall structure of the tool. For an input project, LChecker reads all its program files and extract license information from the source code (we call it local license. If the local license statements are not present in the code, it is marked as unknown). LChecker also tokenizes the program and uses the resulting token list to perform queries via the Google Code Search service[2]. If Google finds that the files already exist in the Internet as OSS programs, it will return the OSS programs together with their license information. LChecker will then parse the returned results and extract the OSS license information. Finally, LChecker checks the compatibility of the local and OSS licenses, and notify the users. Figure 2 also shows the pseudocode of the license checking method.

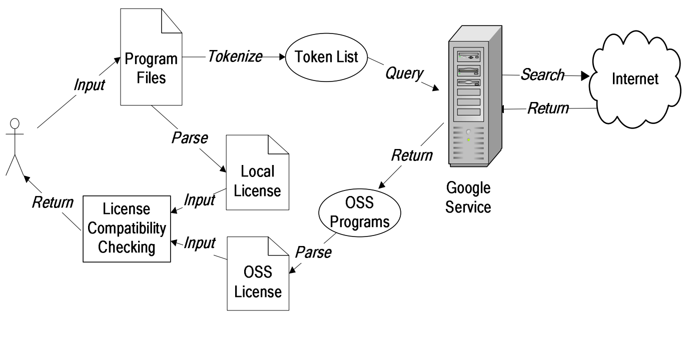

Figure 1. The overall structure of LChecker

[1] This research is supported by the NSF China grant 60703060, the State 863 Project 2007AA010301, and the MOE Key Lab of High Confidence Software at Peking University. The authors thank student Y.F. Liang for helping with license collection.

[2] www.google.com/codesearch

B. Program Tokenization and Query

To tokenize a program file, LChecker firstly removes all comments, blank files, headers in the program. It then performs syntactical analysis of the program and identifies unique lexical tokens. Some tokens, such as keywords, separators, operators are "stop words" that are common to all programs, therefore LChecker removes these tokens before querying. The remaining tokens are concatenated into a single string, which is used as an input for Google Code Search API.

```
LicenseComplianceDetection ( File F )
1.  L = Φ;
2.  OSSLicense = Unknown;
3.  LocalLicense = Unknown;
4.  Extract license info from F and update LocalLicense
5.  Remove all comments and tokenize F;
6.  For every token t in F
7.      If t is a stop word
8.          continue;
9.      endIf
10.     If t is not in List L
11.       add t to List L;
12.     endIf
13. endFor
14. invoke Google code search API using L
15. If found
16.     parse query results
17.    update OSSLicense
18. endIf
19.  check_compatibility (OSSLicense, LocalLicense)
20. return
```

Figure 2. The pseudocode of the license checking method

```
/**
* Exception thrown by a scanner when encountering lexical errors.
*/
public class InvalidInputException extends Exception {
    private static final long serialVersionUID =
2909732853499731592L; // backward compatible
/** Creates a new exception with no detail message. */
    public InvalidInputException() {
            super();
    }
    public InvalidInputException(String message) {
            super(message);
    }
}
```

Figure 3. An example of program tokenization

As an example, Figure 3 shows a small code fragment, after tokenization, the token list used for query is as follows:

InvalidInputException Exception serialVersionUID 2909732853499731592L super message

Given the query string, Google Code Search found that this program is actually from the Eclipse JDT.Core package (org.eclipse.jdt.core.compiler/InvalidInputException.java) with EPL license.

C. Parse Query Results

For a given query, Google Code Search returns all OSS files that satisfy the query, as well as corresponding license information. If for some reasons Google does not specify the license (this happens sometime), LChecker automatically extracts such information from the OSS programs. Also, for a project that experienced a long period of evolution, many files stored at different places across the Internet could be returned. As licenses may be wrongly specified or evolved, we use a "Voting" mechanism to determine the actual license – if most of the returned results agree with a certain license, it would be selected as the OSS license for the file.

III. Experiments

We have evaluated our tool using real-world projects developed by some Chinese organizations. We have successfully discovered some previously-unknown license compliance problems. For example, for a project called *Peony* (for sensitivity reasons, we do not disclose the organization's name here), it does not declare any license information in its source code or release notes (local license is unknown). However, LChecker scans its source code and found that 2 of its files are from an EPL-based OSS projects, 4 files are from a BSD-based OSS project, 8 files are from three Apache-based project and 2 files are from two GPL-licensed OSS (Table I). The *Peony* project has 67 files and LChecker completes the checking within 57 seconds.

TABLE I
License checking results for the Peony project

Local License	OSS License	#Files	OSS Projects
Unknown	EPL	2	WorldWind
Unknown	BSD	4	VO Manager
Unknown	Apache	8	melete, J2G-Migration, maven
Unknown	GPL	2	harpoon, Selenium

As an example, Figure 4 shows the code fragments in suspicious file *Application.java* in the *Peony* project. Clearly it is the same as the file in the *Selenium* project, which is a GPL-licensed OSS project for testing web applications. As requested by GPL, *Peony* should be a GPL project as well.

We have also evaluated our tool using six archived open source projects. We collected these projects from 2001 to 2005. We now run LChecker to see if their licenses have evolved since then. For each project, LChecker scans every program file and to check if the file can be still found through Google Search. If found, LChecker also checks if the new license matches the old one. Table II shows the results. All the files can be still found, most of them keep the old licenses and some changed. For example, for the *Hsqldb* project, the local license is BSD. All its 16 files still keep the BSD license. For the *jpos.main* package, the license used in year 2003 is BSD. All its 311 files can be still found via Google Search. However, 1 file is changed to GPL license, and 17 files are now found AGPL-licensed.

```
// local project: Peony, license: Unknown (unspecified)
package com.cvicse.mspl.src_analyzing;
import org.eclipse.equinox.app.IApplication;
      …
import org.eclipse.ui.PlatformUI;
public class Application implements IApplication {
    public Object start(IApplicationContext context) throws
        Exception{
     Display display = PlatformUI.createDisplay();
          …
     }
}
```

```
// OSS project: Selenium, license: GPL
package com.serli.helium.ui.product.workbench;
import org.eclipse.equinox.app.IApplication;
…
import org.eclipse.ui.PlatformUI;
public class Application implements IApplication {
   public Object start(IApplicationContext context) throws
       Exception {
    Display display = PlatformUI.createDisplay();
    …
     }
}
```

Figure 4. An example of license compliance problem found in the Peony project

TABLE II
EXPERIMENTS ON SIX OPEN SOURCE PROJECTS

Project (archived date)	Old License	New License (As of July 2010)				
		GPL	LGPL	BSD	AGPL	Apache
Hsqldb (2005.5)	BSD (16 files)	0	0	16	0	0
jpos /main (2003.4)	BSD (311 files)	1	0	293	17	0
jboss /main (2004.4)	GPL (79 files)	71	0	1	0	1
jetspeed /daemon (2005.5)	Apache (16 files)	0	0	0	0	16
jbook (2001.9)	GPL (116 files)	115	1	0	0	0
jMusic (2004.3)	GPL (207 files)	185	0	0	0	22

IV. DISCUSSIONS

Because of the importance of license in open source software development, many software engineering researchers have started investigating license issues. For example, German et al. [3, 4] found the license mismatch problem in a large component-based system, where different components may use conflicting licenses. Based on a study of 124 OSS packages, they developed a model that describes the interconnection of components in these packages from a legal point of view. Di Penta et al. [6] addressed the problem of license evolutions. They proposed methods for automatically tracking license changes. Unlike related work, our LChecker is developed to automatically check license compliance via code search.

LChecker can be further improved too. Currently, it is designed for checking license compliance problems caused by ignorance or carelessness. If a developer intentionally plagiarizes the open source software without acknowledging proper licenses (e.g., by changing program structure, call sequences, variable names, etc), LChecker cannot detect the license violations. In future, we will investigate advanced clone detection techniques [2, 5, 7] to improve LChecker's detection ability. In future, we will also optimize the queries used for Google Code Search. Currently all unique tokens are used to form a query string. We will investigate if there are better and more efficient queries, which utilize more program features such as program and data structures. We will also explore if token distribution information that we discovered previously [8] could be integrated for efficient query processing.

V. CONCLUSIONS

An open source license specifies the permissions and restrictions for using an OSS project. License compliance is an important legal issue for OSS-based software reuse. Ignorance or carelessness could result in huge financial losses. In this paper, we present LChecker, a tool that can automatically detect the license compliance problems. We also describe our initial experimental results. We believe that our tool has a potential to be applied to industrial practice.

LChecker is publicly available at:
http://code.google.com/p/lchecker/

REFERENCES

[1] GNU General Public License (GNU GPL), http://en.wikipedia.org/wiki/GNU_General_Public_License, July 2010.

[2] M. Gabel, L. Jiang, and Z. Su, Scalable detection of semantic clones. In *Proc. of the 30th International Conference on Software Engineering*, Leipzig, Germany, May 2008, 321-330.

[3] D.M.German, and A. Hassan, License integration patterns: Addressing license mismatches in component-based development. *Proc. of the 31st International Conference on Software Engineering*, May 2009, 188-198.

[4] D. M. German and J. M. Gonzalez-Barahona. An empirical study of the reuse of software licensed under the GNU General Public License. In *Proc. of the International Open Source Systems Conference (OSS'09)*, Springer, 2009, 185-198.

[5] R. Komondoor and S. Horwitz, Using Slicing to Identify Duplication in Source Code, *Proc. of the 8th International Symposium on Static Analysis*, July 2001, 40-56.

[6] M. Di Penta, D. M. German, Y. Gueheneuc, and G. Antoniol, An exploratory study of the evolution of software licensing. In *Proc. of the 32nd ACM/IEEE International Conference on Software Engineering,* Cape Town, South Africa, May 2010, 145-154.

[7] C. Liu , C. Chen , J. Han , P. Yu, GPLAG: detection of software plagiarism by program dependence graph analysis, *Proc. of the 12th ACM SIGKDD International conference on Knowledge discovery and data mining,* Philadelphia, PA, USA, August 2006, 872 - 881.

[8] H. Zhang, Exploring Regularity in Source Code: Software Science and Zipf's Law, *Proc. of the 15th Working Conference on Reverse Engineering (WCRE 2008)*, Antwerp, Belgium, October 2008, 101-110.