JarAnalyzer: Java Jar File Anaylzer

Wahee Ahmed
School of Computer Science
Carleton University
E-mail:wahmed2@connect.carleton.ca

Karim Tantawy
School of Computer Science
Carleton University
E-mail: karim.tantawy@gmail.com

ABSTRACT

The main barrier for open source development and adoption is violation of intellectual-property. This paper introduces an open source project JarAnalyzer, to solve the code-clone detection problem for Java open source projects. JarAnalyzer will help people to detect similarities between two JAR files by comparing their fingerprint. Fingerprint of the JAR file can be configured and scaled, and can contain different properties of JAR file. The fingerprint generation and comparison algorithms are also discussed in this paper.

I. INTRODUCTION

The widely adopted solution to proprietary software is using the open source software. According to the survey conducted by Gartner Inc. indicates that 85% of companies which were surveys are using open source software in their enterprises, while 15% were still planning to use in next 12 months. Along with this survey Gartner analyzed that about 69% of surveyed companies which are using open source software in their enterprises don't have formal methodology of evaluating and cataloguing them. Thus leading to huge potential liabilities for intellectual-property violations [1].

A .Problem

The total time it takes to develop the software and risk of producing bug can be reduced by using existing code. When IP management is not done in proper way, software developers may reuse and copy the code of open source software without having knowledge of license. Proprietary software has the risk of opening the entire software code if it inadvently includes one piece of code that is under GPL lincese [2].

IP infringement is another problem in open source software. Without compatible license or permission the IP fringers uses open source code in their product. Once the software get distributed in close source binary format it becomes really hard for open source provider to verify it against their own code.

Our project JarAnalyzer is java based application detecting the similarities of the two Jar files. To compare the two JAR files, a fingerprint will be generated to store the JAR files information. The key issue is getting the quick and accurate comparison using the algorithm which generates the fingerprint content. If the fingerprint is large or even if its identical to original JAR file we will get accurate result but fingerprint generation and comparison will take long time. Alternative to this would be generating a very light weight fingerprint it would save lot of running time but the result won't be accurate and reliable.

B. Motivation

It requires lot of time and effort to compare the original code in order to detect the software source code infringement. The majority of software products aren't in human readable format. The motivation of this paper is to develop an open source java based application to detect similarities of the two JAR files and thus contribute to open source community.

C. Goals

To detect the code clone in JAR files is our main goal. The JAR files are basically used to distribute the Java applications and libraries. Among all the programming languages Java is the mostly common used language to develop open source software [3] [4].The implementation of our software is also done in Java. The program will facilitate with percentage similarity of the two JAR files in an acceptable time frame. Not only this but the comparison result will also include the similar java classes.

D. Objectives

In order to achieve our goals we will implement the following objective(s)

- The implementation of algorithm as an open source Eclipse plug-in project. It will enable Eclipse users to generate the JAR file fingerprint, compare JAR file similarity

D. Outline

The paper is organized in this manner: 2nd section describes background information of this project. The 3rd section will address the design approach and the decision made by us during the implementation. The validation of our goal and results of our testing are mentioned in 4th section. In our 5th section we will talk about future work that can be done in order to achieve accuracy.

II. BACKGROUND

The researches have been carried out to do code code-clone detection. These researches classify the comparison algorithm into two categories: structural based comparison and content based comparison.

At the present JarAnalyzer is performing the comparison bases on the content. The files from each directory are compared with fingerprint of know JAR file.

III. APPROACH

A. Design

1) *Generating Fingerprint:* JarContent computes a fingerprint of a JAR file as a collection each class file in the JAR.

2)Copy Detection: JarContent detects whether a JAR file A contains code copied from JAR file A by computing and comparing their fingerprints.

To compare two fingerprints, JARContent iterates all the class fingerprints in the fingerprint of JAR A and compares each of them with all class fingerprints in the fingerprint of JAR B.

3) Summary: The fingerprint generation and code-copy detection algorithms are summarized as follows:

Algorithm 1: Generate fingerprint
Input: A JAR file.
Output: JAR file fingerprint
1.
2.
3.
4.
5.
6.

Algorithm 2: Fngerprint comparison
Input: Fingerprint for JAR file A.
Output: Result in % percentage
1.
2.
3.
4.
5.
6.

B. Decision Made

At the beginning we were just comparing the content of JAR files (class files) on basis of names. But now in order to enhance our percentage of similarity we will be comparing other JAR file attributes as well like sizes , path etc.

IV. RESULTS

This section describes the current results of the experiments that we carried out. So far our algorithm is able to detect 45% of similarity among the JAR files.

V. CONCLUSION

REFERENCES

[1] Gartner Inc. http://www.gartner.com/it/page.jsp?id=801412. Accessed

[2] GNU GPL. http://www.gnu.org/copyleft/gpl.html. Accessed Febuary 18,

[3] Programming Language Popularity. http://langpop.com/. Accessed

[4] TIOBE.http://www.tiobe.com/index.php/content/paperinfo/tpci/index.