

CHAPTER 7

DEDUCING AND MAPPING THE CLASS PATH OF JAR FILE

7.1 INTRODUCTION

Now-a-days, to reduce the effort of human and to improve the efficiency of the process, everything in this world is being computerized. The users may face some problems while executing such computerized process otherwise called as program. A Program is software that contains a set of instructions to perform a task. In order to make the process of developing program, an easier one, some developers have developed software and implemented them as a package.

To access these packages easier, some of them are released as an open source software. An open source software is a software that is downloaded at free of cost. In this open source software, there exist many packages, classes and methods. While using this open source software, to develop an application, these packages are well-connected with that application. It is necessary to generate a class file to start executing the application. But the user doesn't know the classes or its properties to compile them.

To overcome this situation, in this thesis, a method to start compiling the classes in the package automatically is proposed. If the path is not set correctly, it is done systematically through the methodology proposed.

7.2 PROPOSED AUTOMATED TECHNIQUE

7.2.1 Methodology

The objective of the proposed work is to propose a methodology to compile the package automatically and to set up the proper class path for implementing the methods in JAR files.

The summary of the proposed work is as follows: First, the user selects the software to be developed in the program. Upon selecting the software, the methodology proposed compiles the software to identify the errors. From the deduced errors, the error rate is to be found out.

After calculating the error rate, the proposed method goes through the errors and identifies the reporting statement about the error. The error may occur due to the pre-defined packages or due to the JAR files. If the pre-defined packages contain errors, it may be calculated as error rate. If the JAR file connected with the program contains error, it may be due to wrong mapping of the class file of JAR or due to the incorrect set up of the path. If this is the case, then the methodology of this research proceeds as follows:

The user might not be able to identify the class files in the JAR, since it is already a compiled program. The user may not know about the properties and execution of the JAR files. In this work, the automatic search engine has to search for the class files in the JAR and has to set up the class path properly. This automatic process helps the user to run the software without any delay. The user may be free from knowing about the JAR files and the class path set up. As shown in Figure 7.1, the automated process of mapping and deducing the class path is carried out, where the output of the adaptive fault tolerance level is taken as an input for the automated process, where A and B are deduced from the Fault tolerance of OSS.

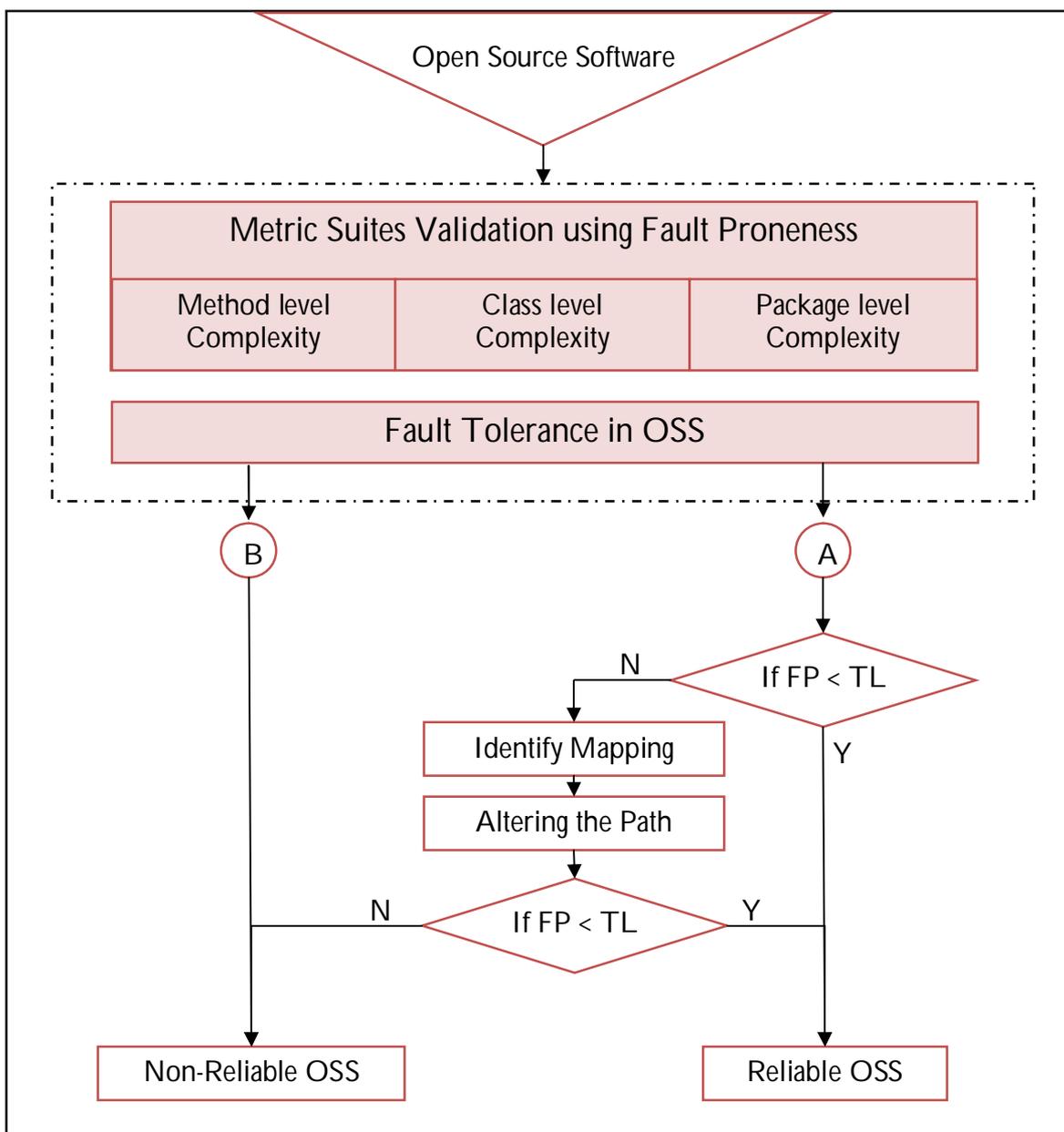


Figure 7.1 Methodology for Mapping and Deducing Class Path

The process of locating the JAR files is carried out as follows:

- Identify the Errors.
- If the error is due to incorrect mapping of the JAR file, then the JAR file is analyzed to identify the class path.

- Detect the class path and map it with the program.
- Execute it after setting up of class path. Thus the JAR file is implemented correctly.

7.2.2 Algorithm For Deducing The JAR File

The proposed Automation Search Engine (ASE) is an algorithm to make the process of compiling and mapping the JAR files automatically with the program. This algorithm compiles the program as shown in section 7.2.2 The algorithm is given in Figure 7.2:

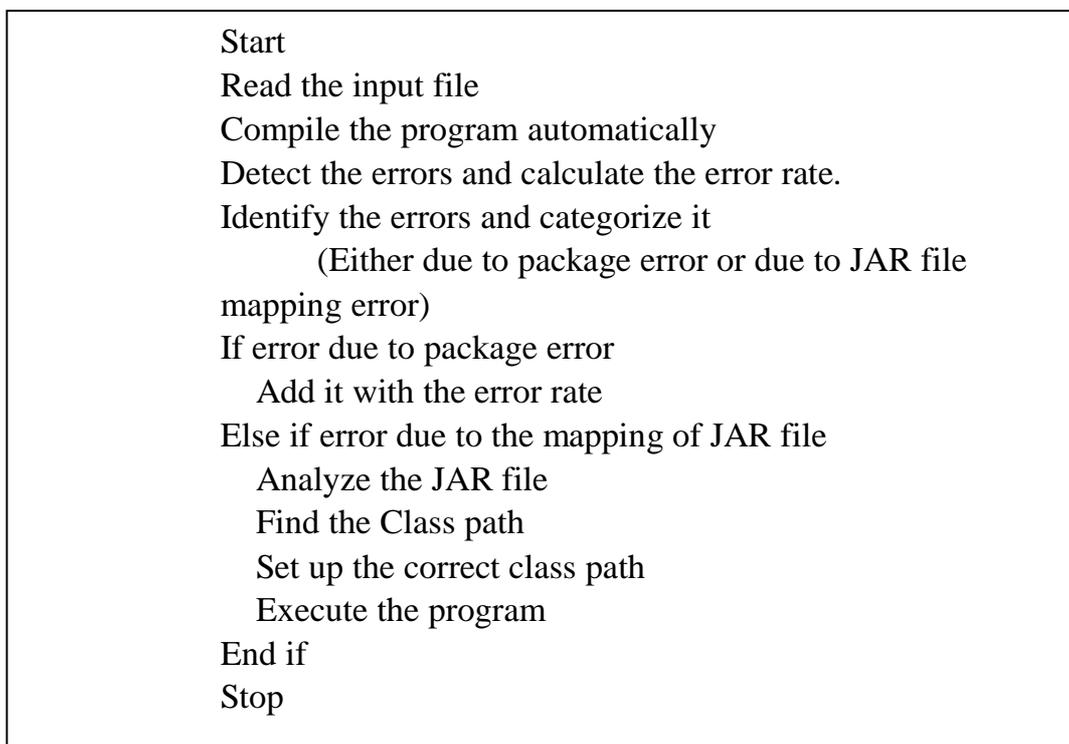


Figure 7.2 Algorithm for automated technique

7.2.3 Experimental Results

The proposed work has to undergo an experimental test for its performance. The experimental setup and the results are discussed in this section.

The experimental setup is undertaken by taking 10 programmers with windows. The programmers are instructed to develop a program by using the open source software. The program must inherit the properties and classes of the packages in the software. The program must inherit the JAR files in its development. Upon developing the program, the program is compiled by the automation search engine proposed and the compilation errors are deduced. From the compilation results, the error rate is deduced and then the errors are categorized. From the classification, if it is found that the error has occurred due to the class path of the JAR file, the algorithm is used to detect the correct class path. The execution is made automatically upon setting up of the class path. Thus the proposed method is implemented with proper experimental setup and the mapping is done automatically. (Appendix 3)

To explain the efficiency of the research work, some open source software are taken for analysis. The execution rate of the open source software is given in Table 7.1. The performance of the open source software before the implementation of the proposed algorithm is shown in Table 7.1 and Figure 7.3.

Table 7.1 Execution rate (Before implementation)

Software	Execution rate of the software before implanting the proposed method (%)
ActiveMQ	50
Apache	78
Beahshell	95
BlueJ	25
IBM (WebSphere)	24
Eclipse	68
EJB Bench mark	87
Hibernate	90
Jasper	91

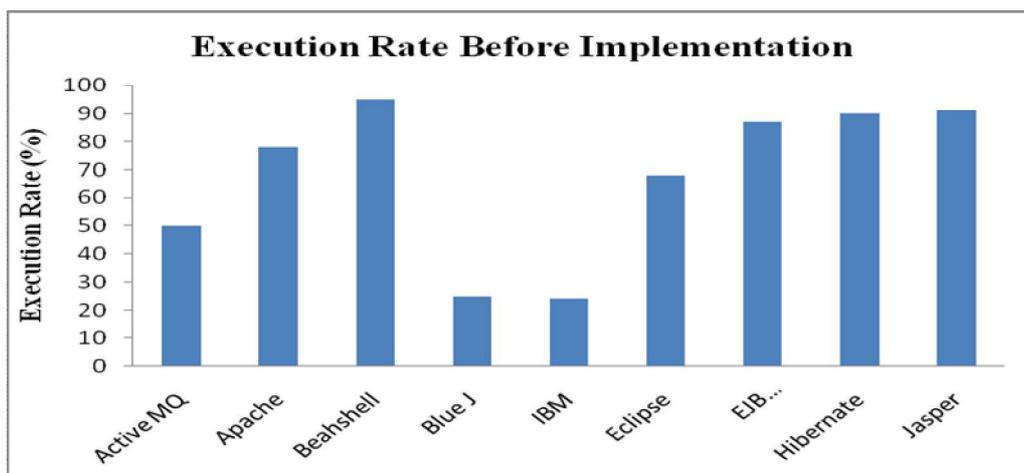


Figure 7.3 Execution Rate (Before implementation)

The performance of the softwares after the implementation of the proposed algorithm is shown in Figure 7.4 and the results in ratio are given in Table 7.2.

Table 7.2 Execution rate (After implementation)

Software	Execution rate (%)
ActiveMQ	80
Apache	82
Beahshell	84
BlueJ	75
IBM (Websphere)	75
Eclipse	80
EJB Bench mark	90
Hibernate	85
Jasper	85



Figure 7.4 Execution rate (After implementation)

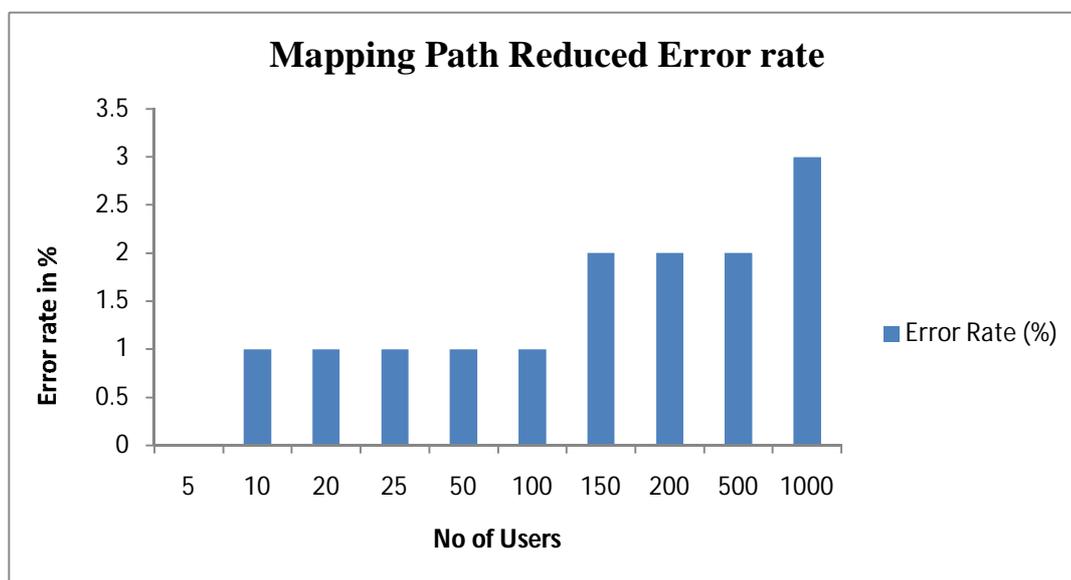
The results from Figures 7.3 and 7.4 it is evident that the execution rate has increased.

Reduced Error Rate after Automation

The error rates with respect to the number of users were analyzed. The rates are tabulated in Table 7.3. After the implementation of the automation process the error rates are identified.

Data Table**Table 7.3 Error rate after Automated**

No. of Users	Error Rate (%)
5	0
10	1
20	1
25	1
50	1
100	1
150	2
200	2
500	2
1000	3

**Figure 7.5 Error rate after automated (Bar chart)**

The analysis of the results as shown in Figure 7.5 and in Table 7.3 show that the error rate has considerably decreased after the automated mapping of the class path to the JAR file.

7.3 CONCLUDING REMARKS

The objective of this research is to develop an automation search engine to deduce the errors occurring in the JAR file mapping and the mapping up of correct path automatically by detecting the correct path of the JAR file.

This work proposes an efficient methodology to achieve the objective of the research work. This is useful in the development of the program in the use of JAR files. If the class paths in the JAR files are mapped precisely then the developed open source software will be successful.