Implementation of Coupling Measurement Tool for Java Software
Sampada P. Kale1 V. S. Bidve2
1,2Department of Information Technology
1,2Smt. Kashibai Navale College of Engineering, Pune, India

Abstract— The way in which components are connected is the key to several quality factors of software. Coupling is the manner and degree of interdependency between software modules. Low coupling is a sign of a well-structured computer system and good design. Object-oriented languages are designed for reducing interdependency between various classes but at the same time, the object-oriented language characteristics change the way the connections are established, how they must be analyzed, and measured. There has been much research effort to develop and empirically validate metrics for OO design quality and strong theoretical background but because of lack of tools we are not able to collect the accurate values of those metrics. In this paper implementation of the Java Source Code analysis tool is described which takes use of Java parser as well as entity data structure. This tool will be useful to give accurate strength of coupling between various classes present in Java software.

Key words: Measurement, Parser, Metrics, Object-Oriented, Coupling

I. INTRODUCTION
The concept of coupling was first introduced in the area of structured development techniques and defined as “the measure of the strength of association established by a connection from one module to another” [1]. The stronger coupling values shows the more dependency between two modules, the more coupling affect the quality of the design and implementation of a software system. Coupling is a measure of how closely connected different routines or modules are and the strength of the relationships between system modules [1]. Low coupling is a sign of a well-structured computer system and a better design, and when combined with high cohesion, support the general goals of high maintainability and readability. The problem of evaluating the design quality of object-oriented systems has been of interest to many researchers. Coupling, Reusability, Reliability, Maintainability, and Complexity are defined as quality measures for object-oriented systems. The relation of coupling as a metric of design quality is related to maintenance, understand ability, and testing. Class coupling measurement importance paradigm necessary for quality improvement of any software.

The goal of this work is to define a suite of measures and build a coupling measurement tool to quantify the level of coupling between classes in object-oriented software systems. The coupling measurement tool measures coupling amongst classes present in the Object-Oriented Java software. This work present Java source code analysis tool that is useful for the software professional to control quality parameters of Object-Oriented software.

The remaining paper is organized as follows. Section II related work is given. Section III describes the implementation of the tool. Section IV describes methodology adopted to analyse the Java code. In Section V results with the discussion is given.

II. RELATED WORK
An inference has been made that more coupling increases complexity which results unreliable system[1].

The tool presented by Jarallah Al ghamdi, Mohamad Ellis, Moataz Ahmad[2] is specially designed for measurement of inheritance coupling. In this, the data is parsed and then it is abstracted to the language dependent format. Parsing engine is used to extract the information needed for measuring the inheritance metrics.

Erik Arisholm, Lionel C. Briand, Audun Fresno [3] defined the tool named JDissect. It is used for calculating dynamic metric measures. Collection and analysis of metrics are done in two phases. In first phase data is collected from Java program and it is stored. In second phase data is analysed. This tool uses the interface of JVM for getting the message traces and other dynamic analysis information.

Dynamic coupling measures were firstly introduced by Erik Arisholm, Lionel C. Briand, Audun Fresno [4]. Export Object Coupling (EOC) and Import Object Coupling (IOC) described in this paper give the Intensity of the interaction of two objects.

Aine Mitchell and James F. Power[5] evaluated the dynamic metrics Java Platform Debug Architecture (JPDA). Dynamic metric measurement is done so that it provide a supplement to the existing static metric.

Aine Mitchell and James F. Power[6] presented a tool in which tool Apache Byte Code Engineering Library is used for metric data collection. When the application runs the probes inserted earlier hit in another file made from instrumentation process. By using these files, the metric program calculates run time program.

Huan LI Bing LI [7] investigates two coupling metrics to evaluate coupling interactions between the classes of object-oriented systems. An empirical comparison of the novel measures with one of the most widely used coupling metrics is described. Specifically, an experiment about the relationships of this pair metrics is conducted. The result shows that software complexity calculated from coupling interaction could not be accurately reflected by one dimension of coupling metric for negative correlation.

A. Kavitha, Dr. A. Shannumgham[8] illustrates a dynamic coupling measurement tool which first introspect the source code and add trace events to it. The trace events are added to each method of all classes.

J. Offutt, A. Abdurazik, S. R. Schach [9] defined a tool named ICAT for the analysis of structure and different components of Java source code packages and find out the coupling among classes. This tool is designed in Java and collaborates with the various software applications like Java parser which is generated by using ANTLR.

Husein and Oxley [10] introduced the coupling and cohesion metrics tool called CCMETRICS for calculating coupling and cohesion metric and implemented it for the object-oriented software systems.
The tool developed by Kayarvizhyl and S. Kanmani [11] any particular object-oriented language to the XML format that is language independent. The proposed tool in this metric contains the parser, XML converter, generic XML container, information extractor and metric collector. S. K. R. Yeresime Suresh, Jayadeep Pati[12] not only illustrates a huge benefit in the usage of software metrics for Software Quality Assessment but also illustrate about evaluating the system effectively through these metrics.

Rani Geetika, Paramvir Singh[13], proposed a tool to measure dynamic coupling of selected Java. The system uses Javaassist library for bytecode instrumentation.

The measures calculated using above tool are not comprehensive, hence does not cover all aspects of coupling. Therefore, in this work we are considering metrics like Parameter coupling, External coupling, Inheritance Coupling, Global Coupling, Data Abstraction Coupling(DAC), Import Coupling and Export Coupling to cover all aspect of coupling. Also, this tool is lightweight as it uses only inbuilt Java Components.

III. IMPLEMENTATION DETAILS

This section describes the detailed implementation of the tool as follows.

A. System Architecture

The architecture of coupling measurement tool is given in fig. 1. It has various components like parser, calculating unit, Entity data structure, etc. The input to the tool is Java source code. Java source code file is given as input to the coupling measurement tool. The tool accepts .java file only. Parser is the part of the compiler that really understands the syntax of the language. It processes the tokens as per the syntax of the language. Token will hold information about the location of the token in the source code. Input Java source code is parsed by parser and converts it into compilation unit. Compilation unit contains token extracted from Java file. Entity data structure is used to hold the information about classes, methods, attributes of an input Java file. The various types of couplings are actually calculated in this unit by retrieving the parsed information from the entity data structure and compilation unit.

![System Architecture of Coupling Measurement](image)

**Fig. 1: System Architecture of Coupling Measurement**

<table>
<thead>
<tr>
<th>File Name</th>
<th>Parameter Coupling</th>
<th>External Coupling</th>
<th>Inheritance Coupling</th>
<th>Global Coupling</th>
<th>DAC</th>
<th>Import Coupling</th>
<th>Export Coupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Measurement</td>
<td>12</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Test</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MyClass</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>q9Multilevel</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Average Coupling</td>
<td>5.2</td>
<td>1.6</td>
<td>4</td>
<td>1</td>
<td>2.2</td>
<td>6</td>
<td>6.2</td>
</tr>
</tbody>
</table>

**Table 1: Coupling Measurement results obtained from proposed tool**

**Table 2: Comparison of Coupling Measurement Tool with the Existing Tool**

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Method and definition of JCAT tool[9]</th>
<th>Results</th>
<th>Measure and Definition implemented in this paper</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Parameter Coupling [9]: Any method call, possibly including parameters.</td>
<td>46</td>
<td>Parameter Coupling: Method or constructor of one class invokes method Passes parameter Passes message to method of another class.</td>
<td>118</td>
</tr>
<tr>
<td>2.</td>
<td>Inheritance Coupling [9]: Inheritance coupling occurs when one class is a subclass or descendant of another.</td>
<td>17</td>
<td>Inheritance Coupling: One class is a superclass or subclass of another class.</td>
<td>34</td>
</tr>
<tr>
<td>3.</td>
<td>External Coupling</td>
<td>N/A</td>
<td>External Coupling: Sharing an external device like printer, HDD by the two classes.</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>Global Coupling [9]: Refers to variables that are defined in one class and used in others.</td>
<td>05</td>
<td>Global Coupling: Method of one class can directly access parts of the internal structure, of another class method (friend). Also to access common, shared, non-local variables of another class.</td>
<td>09</td>
</tr>
<tr>
<td>5.</td>
<td>Data Abstraction Coupling</td>
<td>N/A</td>
<td>Data Abstraction Coupling: One class is used in the implementation of method of another class.</td>
<td>27</td>
</tr>
<tr>
<td>6.</td>
<td>Import Coupling</td>
<td>N/A</td>
<td>Import Coupling: All type of coupling due to import of another any object.</td>
<td>51</td>
</tr>
<tr>
<td>7.</td>
<td>Export Coupling</td>
<td>N/A</td>
<td>Export Coupling: All type of coupling due to export of a class in other classes.</td>
<td>51</td>
</tr>
</tbody>
</table>
IV. METHODOLOGY

An executable Java source code file is used as input to the tool. Following are the types of couplings that can be calculated by the tool.

A. Parameter Coupling
Method of one class invokes method or passes parameter or passes a message to the method of another class then parameter coupling count of invoking class is incremented by one. Suppose method of A class is called in class B then parameter coupling of class B is increased by one.

B. Inheritance Coupling
It is a class-class type of interaction. When one class is a superclass of another class, then there is occurrence of inheritance coupling.

C. External Coupling
When there is sharing of global devices for example if two classes are sharing the same external device like printer or HDD, then there is an existence of External Coupling.

D. Global Coupling
When a method of one class can directly access global variables, the internal structure of another class method (Friend) then it is called as Global Coupling.

E. Data Abstraction Coupling
The emphasis in Object-Oriented languages is on defining abstractions such as abstract data types that model concepts in an application domain. If there is a class method type of interaction means one class is used in the implementation of the method of another class or if there is the class-attribute type of interaction means one class is the domain of the instance variable or local variable of another class then it is called as Data Abstraction Coupling.

public Class A{
}

A Class B{
}

As shown above class A is return type of class B, so there is occurrence of data abstraction coupling.

F. Import Coupling
All type of coupling because of import of any object of another class in a current class is called Import Coupling.

G. Export Coupling
All type of coupling due to an export of a class in other classes is called as Export Coupling.

V. RESULTS AND DISCUSSION

We have calculated couplings of five different Java files. Table 1 shows coupling values of each Java file. Let us consider Measurement file. Its parameter coupling and inheritance coupling is 12 and 10 respectively which is greater than other types of coupling values. The developer should concentrate on controlling these types of coupling to maintain or improve the quality of the software. By using coupling measurement tool the coupling can be measured in various aspects. It gives the average coupling measurement graph of various classes present in a Java file as shown in fig. 2. Coupling of each type for a particular class as shown in fig.3, and particular type of coupling for each class as shown in fig. 4.

In fig 2 the coupling measurement of TulipChain Java software is done. The table shows that the coupling values measured by the new coupling measurement are comprehensive and it detects more cases of coupling than the existing tool. It is the work of developer to consider these coupling values and try to minimise the coupling in order to make the software resusable and less complex.
REFERENCES


