Analysis of software Metrics Tools (A Survey Approach)

Sushil Kumar Bagi , Mr. Sumit Sharma, Dr. Sanjeev Bansal

***Abstract*—Measure the software and the software development process are very important for organization.This paper presents result of study in which we describe about the suit of object oriented design metrics based on java language and also i will explain which approach is better and easy when we are going to build the tool in java. As we know when we are calculate software metrics by different different tools the result will differ due to metrics tools use different methodology. For this purpose i will compare two methodology of sofwtare metrics tools with practical approach.**

Index Terms—LOC(line of code), package, coupling, metrics,Measure, object-oriented software

1 INTRODUCTION

Java is very powerfull language for today scenario. Today more than billion of software has been made with the help of java. As we are aware that programming language Java,which is simple, distributed,object-oriented, interpreted, secure, robust, portable,multi threaded ,high-performance and dynamic, adds better performance and best support for latest wave of open, quick web services. It has many features like it provide various packages and by this packages we can achieve any calculation . Now today there are various tools available in the marcket for calculating java metrics for java software . Software metrics are quantative measure of software products and process. Software metrics are basically two types Product metrics and process metrics. In this paper focus on product metrics. Only. Product metrics contains the size of the product, the logical structure complexity, the data structure complexity, the function of the product, and combinations of them. Especially, the complexity metric for a program code is the most well-known metric, and is usually used in the implementation phase and test phase, since it is a good indicator of whether the product is well-designed, understandable, and easy to modify. For example, Software Science by Halstead and Cyclomatic Number by MaCabe have been used in many software development organization. In java there is not only the concept of simple class. If we want to mesure our quality then it can only possible when we measure all metrics which affect the java programs either directly or indirectly. Good quality can be measure when we use the language specific metrics .Because we know that different language has different features interms of class and object .so our approach to make tool for measuring the metrics based on particular language . Now when we analyze the different tool for calculation then we found there are various methodology for which we can design our tool. so in this paper we will discuss on software metrics tools which is based on java and also support which techniques is better for when we want to developed our tools.

2 METRICS FOR OBJECT ORIENTED SOFTWARE ENGINEERING

In object-oriented design, the class is the basic term of concern, not the procedure or statement. Hence, the metrics used to measure such software should be class-oriented. In order to analyze the software which are coded by object oriented language java. We can easly use conventional metrics to measure some parts of characteristics of object-oriented programs(e.g. the number of classes, the number of methods, the number of variables and the complexity of methods).But of course these metrics can not use to measure the main characteristics for those software which has written in java.

3 TRADITIONAL PRODUCT METRICS

Most tools available today that gather and analyze traditional software metrics. Table 1 shows some commonly used traditional metrics along with a brief description of the metrics. These metrics can certainly be applicd to objectoriented software development (in fact, nearly any style of software development).however, there is evidence that such measures can be misleading when applied to object-oriented designs [6]

Metric	Definition
NCSS	Number of non commented source statements.
Defects	Number of faults in specification, design, or implementation
Number of procuders	Number of statement groups which can be compiled indendently of program.
Total Operand Total Operator	Self explanatory used for calculating Halstead metrics (NI. N2).

Table **1.** Traditional Metrics

OO METRICS SUPPORTED BY JAVA

Software based on object oriented languages like Java are collection of classes and objects which interact with one another to achieving the desired functionality. Interaction between objects generates a relationship between objects. Object oriented metrics proposed by Chidamber & Kemerer and others , provide methods to measure the quality of object design, their relationship, dependency and other OO principles.

Weighted Methods Complexity (WMC)
Response For Class (RFC)
Lack Of Cohesive Methods (LCOM)
Coupling Between Objects (CBO)
Depth of Inheritance Tree (DIT)
Number of Children (NOC)

Most of the software metrics tools are based on above metrics.

4 ANALYSIS OF AVAILABLE SOFTWARE METRICS TOOL FOR JAVA SOFTWARE

There are various metrics tool available for java software.When we search Tools on internet we found 46 different tools for analysis of metrics.For example (i)Analyst4j is based on the Eclipse platform and available as a stand-alone Rich Client Application or as an Eclipse IDE plug-in. It features search, metrics, analyzing quality, and report generation for Java programs. Analyst4j tool are most popular to findout the quality related metrics. This tool is based on Chidamber & Kemerer metrics. It generate vaious reports like

- Number of packages of the project.
- Number of classes of project .
- Number of lines of code of the project.
- Average inner classes of a class in the project.
- Average percentage of non private fields of a class in the project.
- Average response for a class in the project.
- Average coupling between objects of a class in the project.
- Average number of files of a package in the project.
- Average Halstead volume of the project.
- Average number of children of a class in the project.
- Average number of unused variables of method in the project.
- Average abstractness of a package in the project.
- Average essential complexity of a method in the project.
- Percentage of comments of the project.
- Number of files of the project.
- Number of interfaces of the project.
- Average cyclomatic complexity of the project.
- Average weighted method of a class in the project.
- Average weighted method complexity of a class in the project.
- Average lack of cohesion of methods of a class in the project.
- Average inheritence depth of a class in the project.
- Average Halstead effort of a file in the project.
- Average maintainability index of a file in the project.
- Average number of unused parameters of a method in the project.
- Maximum depth of inheritence of the project.

(ii) CCCC is an open source command-line tool. It analyzes C++ and Java _les and generates reports on various metrics, including Lines Of Code and metrics proposed by Chidamber & Kemerer and Henry & Kafura.

(iii) Chidamber & Kemerer Java Metrics is an open source command-line tool. It calculates the C&K object-oriented metrics by processing the byte-code of compiled Java files.

(iv) Dependency Finder is open source. It is a suite of tools for analyzing compiled Java code. So if we analysis these all metrics then we found that these all metrics are static metrics . so to identify the static metrics of our project we should know about all of the static parameter.But when we talk about the accuracy about metrics then it is necessary to know the how our tool collects the information for further processing.

5 COMPARE ANALYST4J AND CHIDAMBER & KEMERER JAVA METRICS

I have found the comparitive analysis for above tool and saw the result has very big differences for same project.[8]

Tool	Data	CBO	DIT	NOC	NOM	RFC	WMC
Analyst4j	MAX Value	32	3	1	25000	155	42
C&K Java Metrics	MAX Value	25	6	1	37000	118	

Table 2

Now when we see above reports then the question is why this result has contained big differences.But one main point is this differences not due to any bug in tools. Coupling metrics (CBO,RFC) calculate the coupling between classes. Decisive factors are the entities and relations in the scope and their types, e.g., class, method, construct or, call, access, etc. Analyst4j calculates for CBO and RFC . These values can be

explained by API classes being part of the scope. These are all imported classes, excluding classes from java.lang (String and Object). Constructors count as methods, and all relations count (including method and constructor invocations).[8]

6: TWO WAYS TO DEVELOP THE SOFTWARE METRICS TOOL FOR JAVA SOFTWARE:

As we found that there are lots of tools available in the market. When we go through then we found that tools can work on two approach . one is based on processing the java file and second one is based on processing the byte code of compiled java file.So we can say that When we are going to developed the Tools for java software then there are two approach for that

(i) Program parsing of java file

(ii) Program parsing of .class file

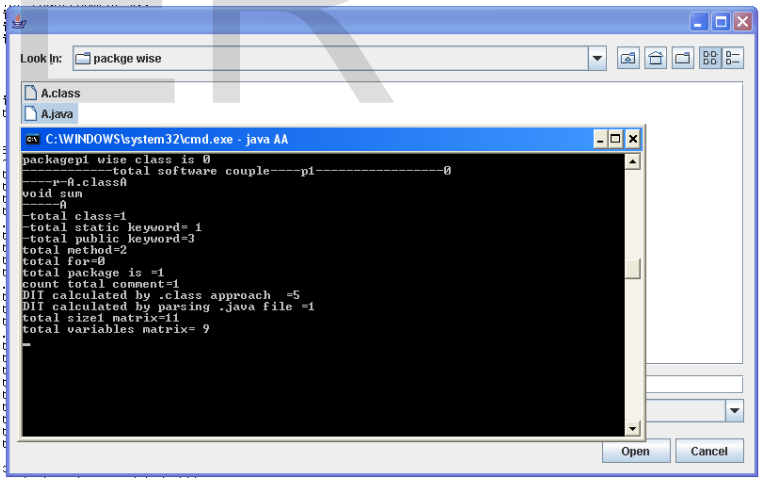
(i) Program Parsing of java File:

When we parse the java file to identify the static parameter for obtaining metrics then we analize that tool become more complex. Because we go and read line by line of our code through program and takes extra effort to develop the software. And this way it is very hard to design tool which calculate more accurate. Accuracy is very importent for our tools because this what quality related work. If our tool not identify accurate parameter then we can not judge quality.In this paper i will show you why this method is not giving accurate metrics.

(ii) Program parsing through .class File:

This is another approach to find out the static parameter for java project. As we know that java provides various packages. So we need to know about Java.lang.reflect,java.lang ,java.net packages. Reflection API is a use for to find-out its environment as well as to inspect the class itself. Reflection API came in Java 1.1. By use of this package we can obtain Implemented interface Obtain class name easly by this API. Obtain method name which is used in API. Finding out the super class name of the class. Obtain all methods used in application etc. So when we use this packages then we can get all static parameter from .class file. It means if we use .class file then it can only use by java interpeter at run time and we can get all static measured inquantity.Comparision between these two techniques with Program :In this paper we analysis the result of these two above approach I have developed one java based tool for obtaining software metrics by using both approach. Here i am not going to give all details about how to make this tool. My Tool is finding out the software metrics using two above approach. So i want to calculate DIT (depth inheritance tree) metrics using my tool and for this i am going to write simple program and then calculate DIT.

```
Package p1;
import java.awt.*;
public class A extends Frame
{
public void sum(){System.out.println("hello!");
}
public static void main(String args[])
{
try
{
int z;
int y;
int r,w;
int e=0;
boolean e1=true;
/*
*/
int u=0;int t; int i;
}
catch(Exception e)
{
System.out.println(e);
}}}
```



Diag 1(out put of tool)

So now the result are :

-total class=1
-total static keyword= 1
-total public keyword=3
total method=2
total for keyword=0
total package is =1

```
count total comment=1
DIT calculated by .class approach  =5
DIT calculated by parsing .java file =1
total size1 matrix=9
total variables matrix= 9
```

Now we can easly see the difference between this two approach. When we use .class approach only few code will require to generate the static parameter in comparision to second approach. As if we decompile Frame class then we found public class java.awt.Frame extends java.awt.Window implements java.awt.MenuContainer and again we decompile Then we found Object class so the heirarchy of inheritence also depends on internal java file. Thats why when we developed the tool then we must ensure what approach we should adopt. And now when we write the code for to identify DIT then we require very less effort.

```
/* code for calculating DIT  by .class file proceessing.
public void  searchDIT1(String pathclass1)
{try  {
File fsearchclass=new File(pathclass1);
File[] childrenclass=fsearchclass.listFiles();
for(int i=0;i<childrenclass.length;i++)
{
String
newclasspath=pathclass1+"/"+childrenclass[i].getName();
File classfile=new File(newclasspath);
String findname=classfile.getName();
if(findname.contains(".class"))
   {
try
{
String classname=classfile.getName();
String classnameonly=classname.replace(".class","");
System.out.println("-----"+classnameonly);
String g1="dit";
Class R=Class.forName(classnameonly);
DITCOUNTMIN=0;
while(true)
            {
if(g1.equals("java.lang.Object"))
break;
R =R.getSuperclass();
g1=R.getName();
DITCOUNTMIN++;
} // end of while
if(DITCOUNTMIN>DITCOUNTMAX)
{
DITCOUNTMAX=DITCOUNTMIN;
}// end of if

}// end of try
catch(Exception e)
   {
   System.out.println(e);
   }// end of catch

}// end of if
}// end of for
   }// end of try

   catch(Exception e)
      {
      System.out.println(e);
      }
```

Explanation:

Now look above code this is very simple way to obtain DIT because i have used java.lang.reflect package for this. First we obtain object of that class using class.forname

Like Class R=Class.forName(classnameonly);

Once the instance has been created we can easly analyze static parameter of that class.And second approach (.java file processing) is very complex because in this approach first go to find out all classes by parsing then find the superclass of that. Again go and relate with all find classes. Now when we talk about the second approach then we need to know the java.lang.String class in details because String class provides all methods by which we can easly parse the .java file .I am not going to write the code for second approach because it is very long .Now i am going to conclude all .

7 CONCLUSION:

Software Metrics Tools not only provide us with an insight into the quality of our source code - some of them can give us clues about other information of our code. No doubt by using metrics we can measure performance about the software quality. As we know that every language has own semantics so we can create matrix for different different language and by language specific metrix can gives better quality measurements software.Here we analize that when we go through to develop the tool for java software using java langauge then we must first use java package for finding the static parameter of our project.Because java also provide deprecated API so it may be case if some person use this deprecated API then result will be differ.so i want to include all classes and method of API must be in the scope then only we find out better quality of sofware.At last we want to conclude that the second approach is not only give you

accurate parameter but also it will reduce the complexity of our tool.

REFERENCES:

[1].Analysis of Metrics for Object-Oriented Program Complexity 1994

[2] S. Yamada and M. Kimura, "Software Reliability Assessment Tool Based on Object-Oriented Analysis and Its Application," Annals of Software Engineering, Vol. 8, Numbers 1-4, pp. 223-238, 1999.

[3]. Towards a semantics metrics suite for object-oriented design. IEEE 2000.

[4] Analysis of quality of object oriented systems using object oriented metrics IEEE 2011

[5] Package Coupling Measurement in Object-Oriented Software , November 24, 2008

[6] A tool for automatically gathering –oriented metrics. IEEE

[7] A Reliability Model for Object-Oriented Software. 2010 19th IEEE Asian Test Symposium

[8] Comparing Software Metrics Tools. ACM 2008

[9] CodeSWAT.COM