Integration of Metric Tools for Software Testing

Including Eclipse Plugin Based Tools

Viplav Srivastava
Department of Computer Science and Engineering
viplav.softech@gmail.com

***Abstract*—**

Software metric is a mathematical definition mapping the entities of a software system to numeric metrics values. Furthermore, we understand a software metrics tool as a program which implements a set of software metrics definitions. There are number of software metric tools available, use different methods to assess metric based software systems and hence project different results. The results are thus tool dependent and are in question for validation. Here an attempt is made to integrate four different object oriented free metric tools. A study has been done to calculate the metrics values using the same set of standard metrics for a software projects. The results have been discussed before for showing the variations in results from different tools for same metrics. Measurements show that, for the same software system and metrics, the metrics values are tool depended. This paper will include three metrics which were not measured while integrating metrics tools. For this we will focus and study on integration of Semmle, Analyst 4j and OOMeter with other metric tools.

Keywords—

Measurement, Verification, Software product Metrics, Eclipse and Software metric tool.

I. Introduction

Accurate measurement is the priority of any software metric tool. A large body of software quality metrics has been developed, and numerous tools exist to collect metrics from program representations. This large variety of tools allows a user to select the tool best suited. Previous papers show that different metrics tools show different metrics values for same measurement and same project to overcome with this problem. We came with the integration of metric tools to get the optimized metric value. Option to select those tools whose license type is free.

II. Object oriented metrics

The metrics presented here are: class related metrics, method related metrics, inheritance metrics, metrics measuring coupling and metrics measuring general (system) software production characteristics. In this paper six metrics are considered for optimization. These metrics are: DIT (Depth of Inheritance), CBO (Coupling Between Objects), LCOM-CK (Lack of Cohesion of Methods) (as originally proposed by Chidamber & Kemerer), LCOM-HS (as proposed by Henderson-Sellers), WMC (Weighted Methods per Class), TCC (Tight Class Cohesion).

III. Software metric tool selection

With the selection of software metrics tools, we limited ourselves to test systems written in Java (source and byte code) and Eclipse based plug-in. SourceForge.NET provides a large variety of open source software projects, metric tools.

TABLE I
Requirements as a basis for the selection of Tools

S.No.	**Requirements**	**Type to suite requirement**
1.	Supporting language	java
2.	measuring metrics	object oriented metrics
3.	license type	freely available
4.	characteristics	command line tool

The selected tools are listed below:

A. CCCC (C and C++ code counter)

It is an open source command-line tool and analyzes C++ and Java files and generates reports on various metrics, including Lines of Code and metrics proposed by Chidamber & Kemerer and Henry & Kafura. It is developed by Tim Littlefair of Edith Cowan University.

B. CKJM (Chidamber & Kemerer Java Metrics)

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

C. OOMeter

It is an experimental software metrics tool developed by Alghamdi et al. It accepts Java/C# source code and UML models in XMI and calculates various metrics.

D. Semmle

It is an Eclipse plug-in. It provides an SQL like querying language for object-oriented code, which allows searching for bugs, measure code metrics.

E. Analyst 4j

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.

IV. METRIC SELECTION FOR OPTIMIZATION

Six software metrics have been selected for this study. These metrics work on different program entities, e.g., method, class, package, program, etc. The tools and metrics are shown in Table II. The hash “#" marks that a metrics can be calculated by the corresponding metric tool. It follows a brief description of the metrics finally selected:

CBO (Coupling between Object classes) is the number of classes to which a class is couple.

$$\text{CBO} = \frac{\text{Number of links}}{\text{Number of classes}}$$

Numbers of links are number of classes used (associations, use links) for all the package's classes. A class used several times by another class is only counted once.

Numbers of classes are number of classes of the package, by recursively processing sub-packages and classes. For the UML modeling project, this variable represents, therefore, the total number of classes of the UML modeling project.

DIT (Depth of Inheritance Tree) is the maximum inheritance path from the class to the root class.

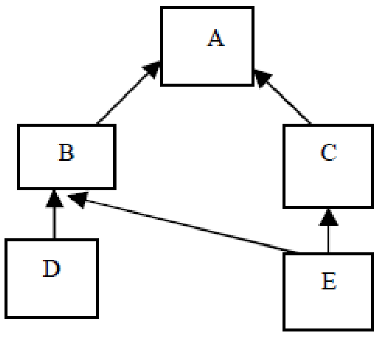

Figure1: Sample measurement of DIT

LCOM-CK (Lack of Cohesion of Methods) (as originally proposed by Chidamber & Kemerer) describes the lack of cohesion among the methods of a class.

$$LCOM(C) = \begin{cases} P-Q & \text{if } P>Q \\ 0 & \text{otherwise} \end{cases}$$

- P = #pairs of distinct methods in C that do not share variables.
- Q = #pairs of distinct methods in C that share variables.

LCOM-HS (Lack of Cohesion of Methods) (as proposed by
Henderson-Sellers) describes the lack of cohesion among the methods of a class.

- M is the number of methods in class (both static and instance methods are counted, it includes also constructors, properties getters/setters, events add/remove methods).
- F is the number of instance fields in the class.
- MF is the number of methods of the class accessing a particular instance field.
- Sum (MF) is the sum of MF over all instance fields of the class.

WMC (Weighted Methods per Class) (using Cyclomatic
Complexity as method weight) is the sum of weights for the methods of a class. It is an indicator of how much effort is required to develop and maintain a particular class. A class with a low WMC usually points to greater polymorphism. A class with a high WMC indicates that the class is complex (application specific) and therefore harder to reuse and maintain. The lower limit for WMC in RefactorIT is default 1 because a class should consist of at least one function and the upper default limit is 50.

TCC (Tight Class Cohesion) The Tight Class Cohesion metric measures the cohesion between the public methods of a class.
NDP – number of pairs of methods directly accessing the same variable.
NIP – number of pairs of methods directly or indirectly accessing the same variable.
NP – number of pairs of methods: n (n-1)/2
Tight class cohesion TCC = NDP/NP

TABLE II
Tools and metrics used in evaluation

Tools	Metrics					
Name	CBO	DIT	LCOM-CK	LCOM-HS	WMC	TCC
CCCC	#	#				
CKJM	#	#	#			
OOMeter	#	#	#			#
Semmle	#		# #	#	#	
Analyst 4j	#	#				

V. **CONCLUSION**

Today a large number of software metrics tools exist. But give different values for the same projects and hence none of them have been validated experimentally for the software metric values they measure. Most tools computed different values for the same metrics on the same projects. From the study it is observed that a new metric tool can be developed which covers metrics values which were emitted before. For more accurate values manual investigation can be done. Since metrics results are strongly dependent on the implementing tools, a validation in terms of manual investigation only supports the applicability of some metrics as implemented by a certain tool. All five different object oriented metrics measured by them have been optimized by investigating the results manually.

References

[1]. Rüdiger Lincke, Jonas Lundberg and Welf Löwe, "Comparing software metrics tools", software technology group school of mathematics and systems engineering växjö university, Sweden issue, July 20–24, 2008, Seattle, Washington, USA.

[2]. Kemerer, C. F. "An empirical validation of software cost estimation models",Comm.ACM 30,5 (May 1987), pp. 416-429.

[3]. W. li and S. henry "Maintenance metrics for the object oriented paradigm". in proc. software metrics symposium, 1993, pp.52-60.

[4]. Shubha Jain, Preeti Katiyar and Prof. Raghuraj Singh," An integration and optimization of software metric tools", international conference on advance computing and communication technologies -2013.

[5]. http://cccc.sourceforge.net/.

[6]. http://depfind.sourceforge.net/.

[7]. http://www.spinellis.gr/sw/ckjm/.

[8]. www2.informatik.hu-berlin.de/swt/intkoop/jcse/**tools**/jmt.html

[9]. B. Henderson-Sellers, L. Constantine and I. Graham. "Coupling and Cohesion (Towards a Valid Metrics Suite for Object Oriented Analysis and Design)". In proc. Object Oriented Systems, Vol. 3,1996, pp. 143-158

[10]. Jarallah S. Alghamdi, Raimi A. Rufai and Sohel M. Khan "OOMeter: A Software Quality Assurance Tool" proceeding of the Ninth European Conference on Software Maintenance and Reengineering (CSMR'05).