IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 4, Ver. V. (Jul.-Aug. 2017), PP 70-73
www.iosrjournals.org

Implementation of Reusability Metrics in Object Oriented Applications

*Dr T.N.Sharma[1], Mannam Chandrasekhar[2],
[1]*Professor, Global Institute of Technology, Jaipur*
[2]*Ph.D Scholar, Department of Computer Science and Engineering, MJRP University Jaipur*
*Corresponding Author: *Dr T.N.Sharma*

Abstract: *Reusability is one of the major advantages of using object oriented paradigm in software development. The goals of software metrics are to identify and control essential parameters that affect the parameters related to software development. A lots of metrics are defined which are related to reusability of object oriented applications. These metrics are related to various aspects like inheritance, number of members of class etc. Calculation of these metrices are equally important. This paper has taken example of a few projects for calculation of some of the metrices which we use for reusability of object oriented applications*

Key Words: *Reusability, metrices, Object-Oriented applications, Coupling, Cohesion, Inheritance.*

Date of Submission: 24-07-2017 Date of acceptance: 14-08-2017

I. Introduction

There are various properties in object oriented paradigm which are useful for reusability. Encapsulation, inheritance, coupling and cohesion are some of these properties. The reusability is a relative context and dependent of various issues. How much one application can be reused is also important. Any object oriented application can be defined as a collection of objects. In a object oriented application reusability can be defined as the percentage of objects which are being reused. The Amount of reuse metrics are used to assess and monitor a reuse improvement effort by tracking percentages of reuse of life cycle objects over time.

The complete study is based on three java projects which are later considered as Project1, Project 2 and project 3.

All the metrics which are important for object oriented applications are given in table below:

S.No	Metric	Object-Oriented
		Attribute
1	Number of Attributes per Class (NOA)	Class
2	Number of Methods per Class (NOM)	Class
3	Weighted Methods per Class (WMC)	Class
4	Coupling between Objects (CBO)	Coupling
5	Number of Children (NOC)	Inheritance
6	Depth of Inheritance (DIT)	Inheritance
7	Method Inheritance Factor (MIF)	Inheritance
8	Attribute Inheritance Factor (AIF)	Inheritance
9	Number of Methods Overridden by a	Polymorphism
	subclass (NMO)	
10	Polymorphism Factor (PF)	Polymorphism
11	Reuse ratio	Reuse
12	Specialization ratio	Reuse

All above metrics can be sub-divided into 7 different categories: size, coupling, cohesion, inheritance, information hiding, polymorphism and reuse metrics.

Size Metrics

These metrices are related to size of classes in the project. There are various options by which size of the class can be categorized. Some of the size metrices are given below:

Number of Attributes per Class (NOA): It defines total number of attributes in a class. In project1 the number of attributes in calculated as 4, in project2 it is 3 and in project3 it is 6.

Number Of Methods per Class (NOM): It counts number of methods a class. In our examples projects we have counted get/set methods also. In project1 NOM in calculated as 11, in project2 it is 8 and in project3 it is 15.

Weighted Methods per Class (WMC): The WMC is a count of sum of complexities of all methods in a class. Let $C_1, \ldots\ldots\ldots. C_n$ be the complexity of the methods

$$\text{WMC} = \sum^{N} Ci$$

If complexity of all the methods is assumed same which is a unity value.

Coupling metrics

Coupling is important for reusability of any of the object oriented application.

Coupling Between Objects (CBO):CBO count of the number of other classes to which a method of class is coupled. Two classes are known as coupled when methods declared in one class use methods or instance variables defined by the other class. The average value of metric CBO for project1 is 2 and for project2 is 0.4 and project3 is 0.8.

Inheritance Metrics

Following are the main methods of inheritance metrics which are useful for reusability of object oriented applications.

Depth of Inheritance Tree (DIT) The DIT is calculated as the maximum length from the node to the root of the tree. The average value of metric DIT for project1 is 1.5 and for project2 is 2 and project3 is 1.

Number of Children (NOC): The NOC is the number of immediate subclasses of a class in a hierarchy. The average value of metric NOC for project1 is 2 and for project2 is 3 and project3 is 1.

Method Inheritance Factor (MIF) It is defined as system level metrics and is defined as follows:

$$\text{MIF} = \frac{\sum_{i=1}^{TC} Mi(Ci)}{\sum_{i=1}^{TC} Ma(Ci)}$$

where, $Ma(Ci) = Mi(Ci) + Md(Ci)$

TC= total number of classes

$M_d(C_i)$= the number of methods declared in a class

$M_i(C_i)$= the number of methods inherited in a class

The average value of metric MIF for project1 is .5 and for project2 is .4 and project3 is .7.

Attribute Inheritance Factor (AIF) AIF can be defined as follows:

$$\text{AIF} = \frac{\sum_{i=1}^{TC} A_d(C)}{\sum_{i=1}^{TC} A_a(C_i)}$$

where, $A_a(C_i) = A_i(C_i) + A_d(C_i)$

TC= total number of classes

$A_d(C_i)$ = number of attribute declared in a class

$A_i(C_i)$ = number of attribute inherited in a class

The average value of metric MIF for project1 is .8 and for project2 is 1 and project3 is .9.

Polymorphism Metrics

Polymorphism defines as implementation of a given operation to be dependent on the object that “contains” the operation.

Polymorphism Factor (PF) The PF, metric is proposed as a measure of polymorphism. It measures the degree of method overriding in the class inheritance tree.

$$PF = \frac{\sum_{i=1}^{TC} M_o(C_j)}{\sum_{i=1}^{TC} [M_n(C_i) \cdot DC(C_i)]}$$

$M_n(C_i)$ = Number of New Methods

$M_o(C_i)$ = Number of Overriding Methods

$DC(C_i)$ = Descendants Count
The average value of metric PF for project1 is .2 and for project2 is .2 and project3 is .3.

Number of Methods Overridden by a subclass (NMO) The average value of metric NMO for project1 is 2 and for project2 is 1 and project3 is 1.

Reuse Metrics
An object-oriented development environment supports design and code reuse, the most straightforward type of reuse being the use of a library class (of code), which perfectly suits the requirements.

Reuse Ratio (U) The reuse ratio, U, is given by

$$U = \frac{\text{Number of superclasses}}{\text{Total Number of classes}}$$

The average value of metric U for project1 is .4 and for project2 is .4 and project3 is .66.

Specialization Ratio (S)

Specialization ratio, S, is given as

$$S = \frac{\text{Number of subclasses}}{\text{Number of superclasses}}$$

The average value of metric S for all three projects is almost 1.

Summarization of all the values

S.No	Metric	Project 1	Project 2	Project 3
1	Number of Attributes per Class (NOA)	4	3	6
2	Number of Methods per Class (NOM)	11	8	15
3	Weighted Methods per Class (WMC)	1	1	1
4	Coupling between Objects (CBO)	2	0.4	0.8
5	Number of Children (NOC)	2	3	1
6	Depth of Inheritance (DIT)	1.5	2	1
7	Method Inheritance Factor (MIF)	0.5	0.4	0.7
8	Attribute Inheritance Factor (AIF)	0.8	1	0.9
9	Number of Methods Overridden by a subclass (NMO)	2	1	1
10	Polymorphism Factor (PF)	0.2	0.2	0.3
11	Reuse ratio	0.4	0.4	.66
12	Specialization ratio	1	1	1

References

[1]. K.K.Aggarwal, Yogesh Singh(2006), Empirical Study of Object-Oriented Metrics, Journal Of Object Technology, Vol. 5, No. 8

[2]. Brij Mohan Goel(2012), Analysis of Reusability of Object-Oriented System using CK Metrics, International Journal of Computer Applications (0975 – 8887) Volume 60– No.10,

[3]. Avinash Dhole(2013), An Approach For Calculation Of Reusability Metrics Of Object Oriented Program, International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 6,

[4]. Aditya Pratap Singh(2014), Estimation of Component Reusability through Reusability Metrics, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:8, No:11

[5]. Neha Goyal et.al.(2014) , Reusability Calculation of Object Oriented Software Model by Analyzing CK Metric, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 7

[6]. Sammy Olive Nyasente(2014), A Metrics-based Framework for Measuring the Reusability of Object-Oriented Software Components, Journal of Information Engineering and Applications, Vol.4, No.4

[7]. Mahfuzul Huda1(2015), Quantifying Reusability of Object Oriented Design: A Testability Perspective, Journal of Software Engineering and Applications

[8]. Aditi Dubey et.al (2015), Reusability Types and Reuse Metrics: A Survey, International Journal of Computer Applications (0975 – 8887) Volume 131

[9]. Mannam Chandrasekhar et. al(2017), Calculation of Reusability Matrices for Object Oriented applications, IJMER, Vol. 7, Iss. 6.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Dr T.N.Sharma. "Implementation of Reusability Metrics in Object Oriented Applications." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 4, 2017, pp. 70–73.