OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

Calculation of Reusability Matrices for Object Oriented applications

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

ABSTRACT: *Reusability is one of the major concerns of object oriented applications. Object oriented paradigm is having its own various advantages including reusability. There are a lots of Metrics available for quantitative measure of readabilities of any of the application which is developed using object oriented paradigm. The goals of software metrics are to identify and control essential parameters that affect the parameters related to software development. There are various types of measurements which are required in software development including size of the project, complexities involved, measurement of cohesion and coupling among modules, testability, reusability, effort and resources required etc. This paper presents a practical calculation on some of the reusability metrics which can be used for object oriented applications*

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.

II. AMOUNT OF REUSE

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.
In general, the metric is:
Amount of life cycle object reused/total size of life cycle object.
Line of code is one of the important part for getting metrics. This code reusability can be achieved by percentage of code which is being reused. A general form for metrics related to lines of code as follows:
Lines of reused code in system or module/ total lines of code in system or module

Reusability Metrics

There are so many aspects which will be dependent on various aspects which includes portability, adaptability, understandability, maintainability and reliability. One of the java project -Software Project Manager(SPM) has been taken here for consideration. Java applications are portable because of being platform independent. There might be different ways to measure reusability. Various authors define different types of parameters. Some of the parameters are presented here in this paper. The total reusability of any of the software component can be computed by
RC=Maintainability + Portability + Documentation + Generality + Understandability
All these features can be calculated with the help of various different parameters. Some of the parameters are being calculated in following sections.

Weighted Methods per Class (WMC*)*
The class which is having more number of methods that will limit the possibility of reusability. The number of methods a ideal class should have, is dependent on various aspects including the type of application, the use of class etc. The project SPM is having total 28 java classes. For simplicity, only .java files are being considered here.
Total Number of classes 28
Total Number of methods (Other than getter and setter) : 127
WMC = 28/127
WMC= .22

Height of Inheritance Tree (DIT)

It is defined as the maximum length/depth from the root to child node of the tree. In the more higher length of hierarchy. The job of prediction of behavior to each class is not so easier. The larger tree makes larger complexity of design because more components are involved. Also in more depth of the hierarchy the use of inherited methods is more. The project SPM is having only three level of hierarchy here. In-build class hierarchy is not considered here.

Number of Children (NOC)

It defines the number of immediate subclasses. The higher number of children defines the higher the chance of reuse, since inheritance is one of the way of reuse. The number of subclasses gives an idea of the potential influence a class has on the design. If a class has a large number of children, it may require more testing of the methods in that class. At each level only few subclasses 3-4 are being used in the project.

Number of Catch Blocks per Class (NCBC)

NCBC is related to exception handling mechanism used in the application. It is ratio of catch blocks and number classes. Here in project SPM, total number of catch blocks are 98 in all 28 classes.
NCBC = 98/28 = 3.5

Coupling between Object Classes (CBO):

Coupling between Object Classes (CBO) for a class is a count of the number of other classes to which the class is coupled or related. Here in total 28 classes 46 coupling is available.
CBO here is 46/28 which is 1.64

REFERENCES

[1]. ETH Zurich(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 – No.