IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 2, February 2015.
www.ijiset.com
ISSN 2348 – 7968

A Survey on Formulation of Faulty Class Detection System for Object Oriented Software

Biraj Upadhyaya[1],Santanu Kumar Misra[2]

[1] Department,of Computer Science and Engineering,Sikkim Manipal Institute of Technology, Sikkim ManipalUniversity Majitar, East Sikkim, PIN 737136, Sikkim, India

[2]Department,of Computer Science and Engineering,Sikkim Manipal Institute of Technology, Sikkim ManipalUniversity Majitar, East Sikkim, PIN 737136, Sikkim, India

Abstract

Classes are the basic building blocks of object oriented design. A class can be considered as a blueprint which is realized with the help of objects. An object represents any real world entity such as customer, car etc. Under the CK (ChidamberKemerer) metric suite, several metrics are proposed which are used to extract objective information about properties of object oriented software. Moreover the extracted information can be used to evaluate software quality, software development process etc. Some CK metrics are extracted from a given java text file and their values are tested against the threshold values defined in the literature. It can be considered as a standard against which the software reliability can be evaluated and necessary corrective actions can be implemented.

Keywords:*Metric, Reliability, Threshold, Regression.*

1. Introduction

In the year 1994, Shyam R. Chidamber and Chris F. Kemerer in their paper [a] defined six metrics which were aimed at measuring some unique aspects of object oriented approach for improving the development of software. IEEE defines the term 'Metric' as "a quantitative measure of the degree to which a system, component, or process possesses a given attribute". It is the measurement of an attribute or computed value of several attributes.

The metrics included in the CK metric suites are Weighted Methods per Class (WMC), Number of Children (NOC), Depth of Inheritance Tree (DIT), Response for Class (Class), Coupling between Objects (CBO) and Lack of Cohesion in Methods (LCOM). Further the above metric are classified into Intra-class metrics and Inter-class metrics. Intra-class metrics are used for measuring characteristics concerned with one class. Examples of Intra-class metrics are WMC, NOC and DIT. Inter-class metrics compute the characteristics prevalent among two or more classes. Examples of Inter-class metrics are RFC, CBO and LCOM. Although various other metric suites have been proposed by various researchers for measuring features of an object oriented design, the CK metric suite remains the most widely used metric suite.

2. Literature Survey

Software metrics is classified into four main categories. A classification hierarchy is presented in figure below:

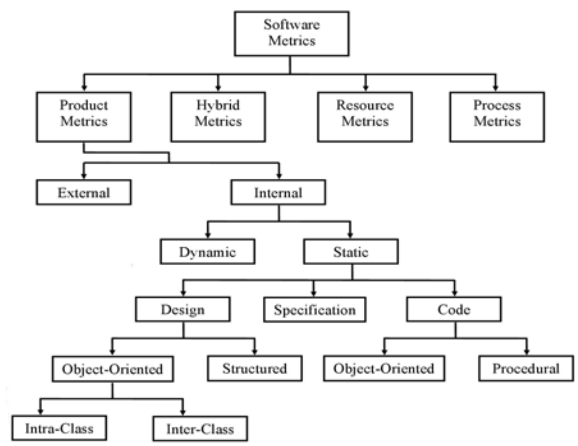

Fig 1. Software metrics hierarchy

2.1 Description of Metrics

Process metrics are used to measure attributes related to the software development life cycle processes. The most significant process attributes are time, cost and effort that could be estimated by Boehm's constructive cost model (COCOMO). *Product metrics* describe characteristics of the software development life cycle processes outputs such as requirement specifications documents, design diagrams, source code listings and the executable software. *Resource metrics* describe the characteristics of available resources. For example, number of developers and their skills, and hardware reliability and performance. *Hybrid metrics* are combination of product and process metrics. Examples

ISSN 2348 – 7968

are: N cost per function point, and time to deliver per LOC.

The metric which we are concerned about are the static object oriented design metrics. Apart from CK metric suite, various other metric suites for object oriented design have been proposed. Examples include MOOD, QMOOD, SATC [2] etc. Structured /Procedural systems are based on operations /functions and data does not depend upon operations in such systems. But data is closely related to operations in real-world problems as it defines the state of a real world object whereas operations define the behavior of that object. Object-Oriented systems use this concept in solving the given problem. Booch in 2004 defined Object Oriented Design as "*Object-oriented design is a method of designencompassing the process of object-oriented decomposing and a notation for depicting both logical and physical as well as static and dynamic models of the system under design*." Classes and objects are the basic units of object-oriented design. The variables and methods in the design define the state and behavior of the objects. Objects interact and communicate with each other by sending messages to each other. A class is an object-oriented entity in which state and behavior are bundled together for all the objects that have common behaviors. Improving design quality, providing high reusability features, shifting towards a faster development process and providing a module-based architecture are some of the advantages of object oriented design.

The following table presents the metrics defined under the CK metric suite:

Table 1: CK metric suite

Metric	Description
WMC	Counts the number of methods implemented in a class
NOC	Counts the number of classes that inherit from a particular class
DIT	Length of the longest path fromthe class to the root in the inheritance hierarchy.
CBO	The number of other classes with which a class is coupled to.
RFC	Number of methods that can be invoked in response to a message
LCOM	counts the sets of methods that are not related through the sharing of some of the class's instance variables

3. Object Oriented Metrics and Reliability

Numerous studies have empirically validated the association between OO metrics and quality of software.Various OO metrics based prediction models and estimation models that focus on validating the effectiveness of OO metrics for either predicting or estimating fault-prone classes or reliability of the system.

Shiak A, et.al. [3] made study on object oriented software reliability models and proposed a new model stating the number of initial parameters serves as an important parameter of reliability model. Suresh Y et. al. [4] discussed how NASA projects in conjunction with SATC (Software Assurance Technology Centre) are applying software metrics to improve the quality and reliability of software products. Reliability is a by-product of quality and can be measured. Johny Antony P[d] measured product attributes of object-oriented system using object oriented metrics based on their effects on product attributes. From the study of the above mentioned literature, threshold for the reliability (RT) is calculated using the relationship established between Reliability and CK metrics. Thus the following relationships can be established between reliability and CK metrics:

Reliability α 1/WMC
Reliability α 1/DIT
Reliability α 1/CBO
Reliability α 1/RFC
Reliability α 1/LCOM

Thus a project whose **R-Value** (Reliability value) lies within the thresholds will have less number of defects and high reliability[6].

3.1Mathematical Observations

If RT (Min) < R-Value < RT (Max),

Then P = Defect (Min) & Reliability (Max)

(Where RT = Threshold of Reliability, P = Project).

Proposed Threshold for CK Metrics

	WMC	DIT	RFC	CBO	LCOM	NOC
Threshold	6-36	1-6	6-36	3-9	1-3	1-3

Assigning Weighted values to the metric:

Rule -1

If Value of Metric lies between the lower limit and (mean of lower limit and upper limit) of the threshold, then the Weightage given to Metric is 1.

ISSN 2348 – 7968

Mathematically: If (Lower Value of Threshold ≤ Value of Metric ≤ Mean of Threshold), then Weightage (Metric) = 1

Rule – 2
If Value of Metric lies between the (mean of lower limit and upper limit) and upper limit of the threshold, then Weightage given to Metric is 2

Mathematically: If (Mean of Threshold ≤ Value of Metric ≤ Upper Limit of Threshold), then Weightage (Metric) = 2

Rule-3
If Value of Metric lies outside the Threshold, then the Weightage given to Metric is 7.

Rule 4
In the case of NOC, (log (upper threshold)) ^2 is considered for RT (Max) and (log (lower threshold)) ^2 is considered for RT (Min). If any of the CK metric value is outside the thresholds, then this metric is neglected.

3.2 Calculating threshold of Reliability Using Rule 1 to Rule 4
Considering only the metrics WMC, DIT and NOC, the formula gets derived as follows:

RT (Max) = K * [(1/WMC + 1/DIT) + (log U-Lt (NOC)) ^ 2]

RT (Min) = K * [(1/WMC + 1/DIT) + (log U-Lt (NOC)) ^ 2]

We can assume the value of K = 1 during calculations.

4. Fault Prediction Techniques

The following fault prediction techniques are described below:

4.1 Statistical methods:

Regression analysis methods can be used for fault prediction. They can classified as into linear regression and logistic regression.

Linear regression analysis: The linear regression is one of the most commonly used statistical technique which consists of two types:

a. Univariate linear regression
b. Multivariate linear regression

In univariate linear regression, the equation is given as:

$$Y=\beta_0+\beta_1 X$$

Where β_0 and β_1 are constants, Y is the dependent variable (accuracy rate) and X is the independent variable (CK metrics).

In multivariate linear regression, the equation is given as:

$$Y=\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3+\ldots.+\beta_p X_p$$

Where X_i is the independent variable, β_0 is the constant and Y is the dependent variable.

Logistic Regression Analysis: It can be classified into two types:

a. Univariate Logistic Regression
b. Multivariate Logistic Regression

In univariate logistic regression, the equation is given as:

$$\pi(x)=\frac{\beta_0+\beta_1 X_1}{1+e^{\beta_0+\beta_1 X_1}}$$

Where xis the independent variables and β_0 and β_1 are constant and coefficient values respectively and π represents the probability of a fault found in the class during validation
Phase.

In multivariate logistic regression, the equation is given as:

$$\pi(x)=\frac{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$$

Where X_i is the independent variable, π represents the probability of fault found during validation phase in the class and p represents the number of independent variables.

4.2 Machine learning methods:

There are various machine learning methods which can be used to predict the fault proneness of a class. The accuracy can be measured in terms of sensitivity, specificity, precision, area under curve (AUC) etc. To predict the fault proneness of a class, we can use machine learning algorithms such as random forest, adaboost, bagging, multilayer perceptron, support vector machine, genetic programming etc.

5. Conclusions

Object Oriented metrics provide a standard for measuring the fault proneness and reliability of an object oriented design. The CK metric suite is widely used metric suite by the research community. Reliability of any object oriented design can be predicted by defining suitable threshold value for the metrics

REFERENCES

[1] Singh and Paramvir, "Design and validation of dynamic metrics for object-oriented software systems" in *http://hdl.handle.net/10603/10463*, Chapter 2, pp. 12-47, 7 Aug 2013

[2] RuchikaMalhotra and Ankita Jain, "Fault Prediction Using Statistical and Machine Learning Methods for Improving Software Quality" in *Journal of Information Processing Systems,*Vol. 8, No.2, June 2012

[3]AmjanShaik, Dr.C.R.K.Reddy and Dr.A.Damodaran, "Statistical Analysis for Object Oriented Design Software Security Metrics" in *International Journal of Engineering Science and Technology*, Vol. 2, No. 5, pp 1136-1142, 2010

[4] Yeresime Suresh, Lov Kumar and Santanu Ku Rath, "Statistical and Machine Learning Methods for Software Fault Prediction Using CK Metric Suite: A Comparative Analysis" in *IEEE Transactions in Software Engineering,* Vol 30, No. 12, pp. 564-570, September 2004

[5] Shyam R. Chidamber, "A Metric for Object Oriented Design" in *IEEE Transactions in Software Engineering,*Vol. 20, No. 6, June 1994

[6]Johny Antony P "Predicting Reliability of Software Using Thresholds of CK Metrics" in *International Journal of Engineering Research & Technology,* Vol. 2 No. 6, June 2013

[7] Indu Sharma, Ms. ParveenBano, "A Combined Approach of Software Metrics and Software Fault Analysis to Estimate Software Reliability" in *IOSR Journal of Computer Engineering*, Vol. 11, No. 6, pp. 01-14, May - Jun. 2013

[8] JieXu, Danny Ho and Luiz Fernando Capretz "An Empirical Validation of Object-Oriented Design Metrics for Fault Prediction" in *Journal of Computer Science,* Vol. 4, No. 7, 2008

BirajUpadhyayais currently pursuing M.Tech in Computer Science and Engineering from the Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Majitar, Sikkim, India. He completed his B.E in Computer Engineering from Institute of Engineering, Tribhuvan University, Nepal. His areas of interest includes software engineering, data structures, algorithm design.

Santanu Kumar Misrais currently working as an Associate Professor in the Department of Computer Science and Engineering, Sikkim Manipal University, Majitar, Sikkim, India. He has more than 8 years of teaching experience. His areas of interest includes software engineering, operating systems, algorithm design.