Improving the quality of software cohesion metrics through dynamic analysis

Ravneet kaur, Amritsar
Department of Computer Science,
Lovely Professional university, Phagwara, India

***Abstract* - Class cohesion is an important object oriented software quality attribute. Assessing the class cohesion and improving the class quality accordingly during the object oriented design phase allows for cheaper management of the later phases. Most of the cohesion metrics planned in the journalism describes static cohesion measurement at design level. Static metrics is a measure that is applied at the design level. Dynamic metrics is some measures that are performed at object level. It provides more accurate cohesion value as compared to the static metric. Dynamic cohesion performed measurement and provide good approach into behavioral aspects of the system. Dynamic cohesion measurement provides the capacity of cohesion metric at object level and using various object oriented characteristics such that encapsulation, polymorphism, parameters for the period of measure. Dynamic cohesion metrics introduce the correct meaning of the measurement and then define measures, validation and verification. The dynamic cohesion metrics are validated by using byte code, java applications to find the affected class and performs changes into it. The execution based approach is used to measure module cohesion of legacy software. The cohesion metric is based on definition-use pairs in the dynamic slices of outputs. This approach significantly improves the accuracy of cohesion measurement. The dynamic technique is applied on the static metrics and measures module cohesion that will give more appropriate values of cohesion.**

***Keywords* - Cohesion Metrics, Dynamic measures, static measures, AMC, AAC, MAAC**

I. INTRODUCTION

Software engineering is the application of progress, advancement, maintenance and implementation of software. Software engineering is split into ten various parts as follows[1]:
1. Requirements engineering: The investigation, description, verification of needs for software components.
2. *Software design:* It is defined as the procedure of construction, mechanism, interfaces, and some characteristics of a software system or component.
3. Software construction: This is complete building of meaningful software by combining coding, validation, testing.
4. Software testing: The object level certification is a program of set of test cases is satisfied to select unlimited domain execution and contrasting to the expected performance.
5. Software configuration management: In configuration management is defined as systematically controlling the configuration variations and maintaining the integrity in software life cycle.[2].
6. Software engineering management: The requisitions of organization are- arrangement, combination, and reported-to certify that the software construction and implementation of the software is systematic.
7. Software engineering process: It is defined as the major key parts of the projects such as implementation, maintenance, estimation, development and quality.
8. Software engineering tools and methods: This is one tool that is related to the software engineering tool. It provides the awareness about the life cycle. To make the software engineering activities in the systematically way[3].
9. Software quality management: It is the ratio of inherited properties that fulfills the customer needs.

Definition of Metrics

It is very important views of organization according to consultant Peter Drucker. If you cannot measure, if the developer and manager are not able to measure precisely, it means they could not evaluate the performance of the software projects. The success rate have been best achievement of the companies, yet the measures that provide benefit of measurement the success or project failures are very dissimilar and these type of metrics scarcely has a good cohesion. Utilizing congruous quality metrics is critical to easily manage the projects, as on the other hand it can be hard for a project manager to find out that project is improving according to the adaptation. Measurements are involved current situation of the projects and evaluate its strength. Measures are evaluated project conditions and divided into requisites, hazards, errors, testing and documents. When metric is not defined at a task level rather it is a combination of various metrics that can represents the behavior of the project. [4].

Cohesion*:* cohesion metric measures how many the methods and class functions are inter-related to each other. A cohesive class performing not more than single function, it means using one function. A non-cohesive class performed more than two non-related functions. A non-related function class might required be constructed again multiple similar classes.
Basic idea of related class: A related cohesive class performed only a single function. If non-related function class performed un-related functions, it would be split into them [5].

1. High cohesion is preferred it assist encapsulation. Drawback, a related function class includes highly coupling in between the modules method.
2. Low cohesion indicates poor design.

In case of dynamic cohesion metrics, basically performing a dimension. Dynamic or object level cohesion metrics to measure the run-time level and provide the same class objects and all the objects lies at same class . The design based cohesion measured at class level..

II. REVIEW OF LITERATURE

In this paper [1] focused on the previous study and discussion about dynamic metrics provides the dynamic behavior. Again defined a component of a unit cohesion metrics are SFC (strong functional cohesion), WFC (weak functional cohesion). The expectation of the dynamic metrics are shows the run time performance on the dynamic slicing. They are used dynamic slices of outputs to measured unit of cohesion. According to author define SFC dynamic metric is module cohesion obtained from common definition-use pairs of each type common to the various dynamic slices of the output variables and WFC metric cohesion obtained from definition-use pairs of each type find in dynamic slices of multiple object values.

In this paper [2] cohesion metrics measures only pattern Interactions but does not shows the variation any write interaction from read interaction. Thus, does not reflect properties of the class. This research measures the improving the cohesion measurements considering read and write interaction in addition to dynamic environment. In this paper, defines the interaction in between the multiple variables. LCOM1 and LCOM2 is the dynamic metric based on the object oriented programming approach.LCOM1 counts the digit of non-related two methods, so LCOM1 is computed by subtracted the number of similar type pairs of methods from the complete amount of distinct method pairs.LCOM2, amount of related two methods are subtracted from the amount of non-related method pairs. TCC (tightly class cohesion) considered the two different methods are inter-relate if the share some common occurrence variable in using. The previous version of TCC (tightly class cohesion) is considered the degree of the total addition of coherency heaviness of each group of methods to the relative amount of method pair off. As an expectations work, this work can be extending for the previous cohesion measurement.

In paper [3] Cohesion class is an important object-oriented quality defined quality attribute values. It shows how many participant of one class are similar to that group of class. Assess the class cohesion and getting improves the class feature considered the object-oriented during the static level. The metrics are considering the method-method, attribute-attribute, and attribute-method interactions. Attribute- attribute and attribute-method direct interactions allowed for cheapest management of the previous phases. According to this paper defined classes of attribute and method in various classes such as one method-to another method, one attribute-to another attribute values and one attribute connect to the method. The introduced metrics can be improving in several instructions, like considering not a direct interactions and method invocation interactions. A cohesion method-method interaction is representing in the DAT by two inter-related rows share binary values 1 in a column. Similarly, a cohesion class attribute-attribute interaction is representing in the DAT by two. The metric used the distance in between the pairs of methods and pairs of attributes as based to calculate their degree of similarity. According to author several predefined design-based class cohesion metrics are overviewed and discussed.

In this paper [4] software metrics are used to check and evaluate various aspects of the complexity of a software product. Coupling and Cohesion are considered to be the most important attributes. The increasing need for software quality measurements has led to extensive research into software metrics and the development of software metric tools. Many Software Metrics have been proposed for object oriented paradigms to measure various attributes like complexity, cohesion, software quality, and productivity. As object oriented analysis and design appears to be at the forefront of software engineering technologies, many different object-oriented coupling and cohesion metrics have been developed. To maintain high quality software, developers' choice is always low coupled and highly cohesive design. The aim of the paper is to reinforce the existing coupling and cohesion metrics specifically used in object oriented environment, to analyze their significance in software development, expose their limitations and some suggestions for further investigations.

III. COHESION METRICS

Cohesion can be explained as the intra-modular functional relatedness of a software module. It can categorize cohesion is into seven levels whereas ranging from low cohesion to high cohesion.

Static Cohesion

There are a lot of alternative measures which are being proposed for measuring cohesion.
1. Chidamber and Kemere suggested that the Lack of Cohesion in Methods (LCOM1) metric measures the lack of cohesion of a component. If a component C1 has n operations O1,... On then {Ij} is the set of instance variables used by operation Oj. Let |P| be the number of null intersections between instance variables sets. Let |Q| be the number of non-empty intersections between instance variables sets. Then: LCOO = |P| – |Q|, if |P| > |Q|, LCOO = 0 otherwise. LCOO actions the amount of method/advice pairs that do not an admission the same instance variable. It is an evaluate of lack of cohesion. This metric expand the CK metric LCOM. It is advices that methods of aspects in the same way that CK regard methods of classes [5].
There are many other methods too that classify set of cohesion measures of object-based systems which can be adapted in object oriented systems. For an adaption a class is viewed as a group of data declarations and methods. A data declaration is a local,

public type declaration, public attributes or the class itself. There can be data declaration interactions between classes, attributes, types of different classes and methods. These are as follows [6]:
1. Ratio of Cohesive Interactions (RCI)
2. Neutral Ratio of Cohesive Interaction (NRCI)
3. Pessimistic Ratio of Cohesive Interactions (PRCI)
4. Optimistic Ratio of Cohesive Interactions (ORCI)

Dynamic Cohesion Metrics

It shows the measurement for dynamic cohesion metrics using program execution approach based upon dynamic slices. They use dynamic slices of outputs to measure module cohesion. Module cohesion metrics based on static slicing approach has some issues in cohesion measurement [8]. This approach limelight the limitations of static cohesion and introduce a new scheme for dynamic cohesion. It is of two types:
1. Strong Functional Cohesion
2. Weak Functional Cohesion

SFC is a module cohesion which is obtained from common diffuse pairs of each type common for all the output of slices [9].
WFC is a module cohesion which is obtained from diffuses pairs of each type common for one and two output of slices [10].

Size Metrics

The software size physically measures the length of software system's code and design. Size metrics has following attributes:
1. Vocabulary Size: It counts the number of components. It helps to calculate number of classes and its aspects. This metric deal with the system vocabulary size . Each component name is counted as part of the system vocabulary. The component instances are not counted.
2. Line of Code: It counts the number of code per line. It is used to measure the size of the code. It was the traditional method for cost estimation.
3. Number of Attributes: This helps to count the number of attributes of each class and aspects. It plays important role to calculate internal vocabulary.
4. Weighted operation per components: This metric measures the complexity of a component in terms of its operations. Consider a component C1 with operations O1 ,..., On. Let c1 ,..., cn be the complexity of the operations. Then: WOC = c1 + ... + cn. This metric originally does not specify the operation complexity measure, which should be tailored to the specific contexts. The operation complexity measure is obtained by counting the number of parameters of the operation, assuming that an operation with more parameters than another is likely to be more complex. This metric extends the CK's WMC metric. It is an advice and methods of aspects in the same way that CK treats methods of classes [11].

IV. PROPOSED METHODOLOGY

The software complexity can be estimated with the help of cohesion and coupling values. The cohesion is degree to which the different function is performed. The importance of the cohesion value can be judged at the software time is maintained. When some software is developed and installed on the user end at that time the cohesion value will be calculated by the developer and it must be high. For the software maintained purpose you need change the software modules due which the cohesion value will be reduced. In the previous time various static techniques had been applied to calculate cohesion value. The static technique will not able to exactly calculate the cohesion value which will reduce the efficiency of cohesion. The definition use pair technique is the slicing technique which is applied to calculate cohesion value, but the static slicing will either calculate over or under cohesion metric. In this work, we apply dynamic technique to calculate dynamic cohesion metric exactly when some changes will be applied in the software. In this research, we are using some dynamic technique that is applied on the some static metrics. Static techniques are Def-use pair defines dynamic slicing method to calculate the measurement of the metrics. Ontology object metric is also a dynamic technique that is based on patterns. Static metrics are RCI (Ratio of cohesion interaction), DCC (Dynamic class cohesion), CAMC (Cohesion among Method of class), NHD (Normalized Hamming distance), SNHD (Scaled Normalized Hamming distance). In this work, we will provide some calculations which will calculate the cohesion value of the software modules dynamically using ontology object metric technique. The ontology of object is techniques which are based on the patterns, with the help of ontology technique similarity between various software's are derived and on the basis of derived patterns you can estimate the cohesion value.

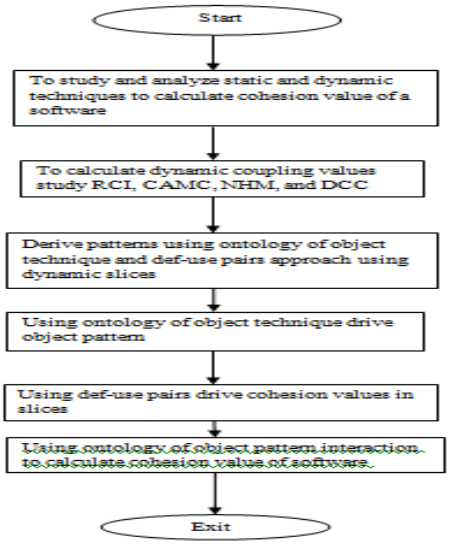


Fig1.1 *Flowchart of Methodology*

V. EXPERIMENTAL RESULTS

The whole scenario has been implemented on MATLAB.

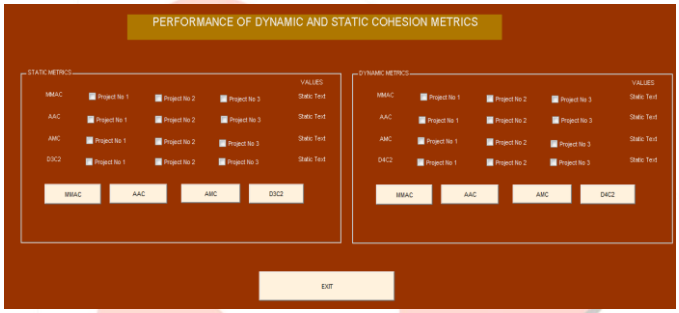


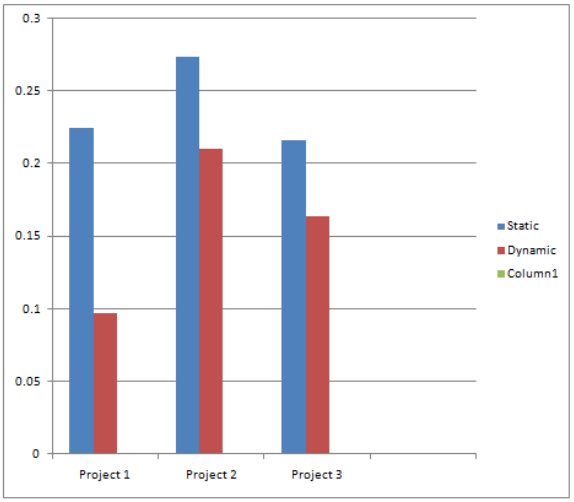
Fig.1.2 *Ideal state*

Figure 1.2 D3C2 metric is applying definition use pair dynamic approach on the D3C2 static metric to calculate the cohesion value is dynamically for the project2.

To select the projec2 and calculate the MMAC, AAC, AMC and D3C2 values.

To provide the accurate result as compare to the static metric.

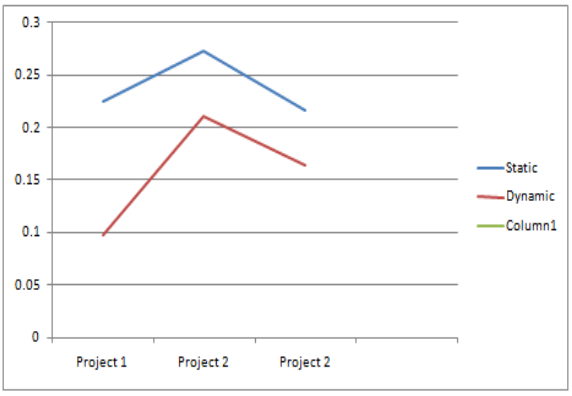
Project2 static is providing less accurate values as compared to the dynamic analysis project2.



Static and dynamic comparison of D_3C_2

Table 1: static and dynamic of cohesion for project

	Project1	Project2	Project3
Static cohesion values of D_3C_2	0.225	0.27361	0.21667
Dynamic cohesion value of D_4C_2	0.0972	0.21013	0.16358



VI. CONCLUSION

The problem to improve the quality of cohesion metrics led to emergence of dynamic analysis. The proposed dynamic cohesion measures are more accurate it because take into consideration run-time behavior of classes. The proposed dynamic technique uses def-use pairs have been applied on the static metric D_3C_2 to analyze it dynamically. Using dynamic techniques, we can obtain more accurate measurement of functional cohesion compared with new techniques. The static cohesion metric is using some dynamic approaches to obtain accurate results and better considerations of run time behavior then the existing metrics. Definition use pair is dynamic technique using dynamic slicing criteria on the some static metrics to take more accuracy. The proposed dynamic cohesion measures are better indicators of external software quality attributes such as change defects than the existing cohesion metrics. To use any other dynamic approaches to perform dynamic analysis and to take better and more accurate cohesion values in future. The introduced metric can be improved in several directions such as dynamic analysis. In Future, use another static metric performs dynamic analysis to carry out the cohesion values are more accurate.

VII. REFERENCES

[1]Varun Gupta, Jitender kumar Chhabra, dynamic cohesion measures for object-oriented software-Journal of system architecture,2011.
[2]Payal Khurana & Puneet Jai Kaur- Dynamic metric at design Level- International Journal of Information Technology and Knowledge Management July-December 2009, Volume 2, No. 2, pp. 449-454.
[3]Jehad Al Dallal- A Design-Based Cohesion Metric for Object- Oriented Classes- World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering, 2007.
[4]Jitender Kumar Chhabra and Varun Gupta-A Survey of Dynamic Software Metrics-Journal of computer science and technology, 2010
[5]Kuljit Kaur and Hardeep Singh-An Investigation of Design Level Class Cohesion Metrics-Department of Computer Science and Engineering, Guru Nanak Dev University, India January 2012.
[5]N.Sasirekha, A.Edwin Robertand Dr.M.Hemalatha-Program slicing techniques and its applications -International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.3, July 2011.
[6]N. Gupta, P. Rao, Program execution based module cohesion measurement, in: Proc. 16th International Conference on Automated on Software Engineering (ASE '01), San Diego, USA, 2001.
[7]A. Mitchell, J.F. Power, Run-Time Cohesion Metrics for the Analysis of Java Programs, Technical Report Series No. NUIM-CS-TR-2003-08, National University of Ireland, Maynooth Co. Kildare, Ireland, 2003.
[8]Esubalew Alemneh-Current States of Aspect Oriented Programming Metrics-**Volume** 3 Issue 1, January 2014.
[9]Varun Gupta, Jitender kumar Chhabra, dynamic cohesion measures for object-oriented software-Journal of system architecture. 2011
[10] http:/ www.infoq.com/articles/projectmetrics
[11] http:/ www.ito.darpa.mi/research areas/EDCS.html