Volume 5, Issue 4, April 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Review of Static and Dynamic Analysis of Software

Silky Bindra, Er. Navdeep Chopra

Computer science & Haryana Engineering College Computer Science &
Haryana Engineering College, Haryana, India

Abstract—During the last decade, the software product measurement field has known many improvements and becomes an emerging field of the software engineering. Depends on the approaches and concept such as object-oriented and structure programming concept and so on. Many kinds of measures are proposed in the literature. The static metric has been found to be limited for modern object-oriented software due to insufficient contribution of object-oriented features such as polymorphism data hiding and inheritance concept. This concept motivates to focus on the dynamic metrics as dynamic metrics have many advantages over the static metrics. Different types of the dynamic metrics were proposed for the measurement of coupling at the run-time status. New types of the dynamic software metrics were presented for the measurement of design and complexity of the object-oriented software.

Keywords— Class level metrics, Complexity, SDLC, Hybrid Model, Testability

I. INTRODUCTION

A. Software Engineering

Software engineering is an engineering discipline that is concerned and associated with all the aspects of the software productions. Products of the software were made up of developed programs and associated documentation is required in all these products. The necessary and essential product attributes of software engineering are maintainability, efficiency, functionality, reusability and usability. The software system process contains many activities which involved in developing software products. Important activities of the software engineering are software specifications, validation or verification processes. Methods that are used organized in a ways of producing the software. Technologies of object oriented are built upon a sound engineering foundation and their related elements which are collectively called the object model. That model encompasses and also contains many useful software construction processes like abstraction, encapsulation, data hiding process, and inheritance process. Therefore, object model is very helpful for the understanding of many problems, communicating with the application experts and modelling complex enterprises into a software design. This technology is helpful in promoting the software reusability, functionality, maintainability, concepts and Performance criteria .The popularity of the internet which is coupled with advances in the local area network and very high speed network technologies have introduced many types of new distributed applications. Few examples of this which includes the software in the area of the system supported collaborative work, airlines work and the hotel reservation systems etc. The Object-oriented concepts and techniques were often used to cope with the complexity of developing these software systems which have come to be known as distributed object-oriented software systems.

B. Software Metrics

Software metrics are the units of measurement which are used to characterize the software engineering products (requirement, design, source code, testing etc.), software engineering cycle processes (analysis requirement, design, coding, testing and maintenance etc.) and software engineering professionals (the efficiency and ability of an individual tester, and the productivity and creativity of an individual designer is increased efficiently).If all these used properly then the software engineering metrics allows us to quantitatively define the degree of success or the unsuccessful , for the product, or for the person, make meaningful and useful and decisions, and after that make them quantified and meaningful properly and thus ,the incorporating metrics into development plans is a very simple step towards creating better systems. LOC (lines of code) and Cyclomatic Complexity are most popular and time honored software metrics. These measures defined for the procedural programs and later incorporated for the object-oriented concepts and processes which are present. The metric LOC (Line of Code) calculates the size of the module and the logical complexity of module was measures by cyclomatic complexity. This concept is proposed and then used for the procedural paradigm have been found incorrect and inadequate for the object-oriented software products and the reason is because of the distinguishing features of the object-oriented concept like classes, function ,concept of inheritance process ,data hiding and polymorphism and Static metrics mainly have their focus on the static properties they acquire of the software and a huge number of static metrics have been proposed in the literature for the measurement and calculation of the cohesion and the coupling and some other attributes of the object-oriented software using design or source code of software that are also static and the concepts of Static metrics are capable to quantify various aspects of the complexity of the design and the source code of the software process, but the ability or capability to predict with the accurately for the dynamic behaviour of an application is not prove today. Earlier, the static process alone may not sufficient in evaluating the dynamic behaviour of the particular application at runtime entity because the behaviour of the static metrics will be affected by the execution of environment as well as the complexity of the source code process. Dynamic coupling metrics process were used to measure the real and actual coupling which is held between the pair of the objects or classes at status runtime in a behaviour of system. The dynamic coupling process were measured at the object level and can be aggregated to the classes or at the system level packet and the dynamic coupling metrics process parameters can be defined at various different level or stages of the software development life cycle(SDLC) process like design-time or coding-time or testing time.

C. Dynamic Metrics Have An Extre Edge Over Static Metrics

In this section, we will discuss benefits of having dynamic metrics in comparison to their static metrics. Static measures are simpler in nature to collect because there is no need to execute the software. To generate the dynamic metric process, the code and the simulation models of software system were necessary needed and all these models are available very late in software development life cycle (SDLC) process. The Static metric coupling process used efficiently and mostly due to the reason that these can be obtain easily, especially at the early stages of software development process stages. Initially, the benefits of dynamic metrics process were collected by executing the program outweigh the complexity and estimating cost of them all. Static metrics processes have satisfied the need of analysing the quantity attributes such as the size and complexity of the software artefacts. But these metrics are less in use and precise than dynamic metrics process in the calculation and measurement of the quality attributes of software such as usability, reliability, testability, functionality, maintainability and so on as the static metrics were analyse only by the means of static inspection of the system and the Quality and quantity of software system is more dependent on the runtime behaviour than the potential characteristics that are implied by the static analysis of the software In the dynamic metrics all the requirement were analysed and computed on the basis of data which is collected during the actual execution of the software thus directly show or reflect the quality attributes such as performance ,reliability, testability, usability ,error rates etc. of software in its operational mode of behaviour. Thus all the static metrics deals with all types of structural aspects of the software system and on another side the present runtime metrics also deal with the behavioural aspects of software. Let suppose an example, according to result of the controlled experiment conducted by Briand and static metrics coupling process analyses and measures may be insufficient to explain discrepancies in changeability for the object-oriented basic concepts. Moreover, the static metrics constrained somehow in their ability to deal with the inheritance, free from error, polymorphism and more dynamic binding issues since the run-time types of field access and method invocation are not too much known. On other side, the dynamic metrics are capable to deal with all such types of issues.

D. Coupling Measurement

Coupling measurement has traditionally been performed by using the static code metrics analysis due to the reason that mostof existing work was done on the non-object oriented code and because dynamic code analysis is very expensive and there is difficult to be perform. For modern system, this type of focus on the static analysis concept can be problemized due to the reason that the dynamic binding exists before the advent of the object-oriented process however its usage has been increased inthe past few year. We refers to this type of coupling process as the dynamic coupling process. An evaluation of the proposed dynamic coupling measures is getting reported in which there is a study that show the relationship of these measures with the changed proneness of the classes concept .The results suggests that some dynamic coupling that are measures were significant indicators of change of proneness and that they are complement existing coupling measures which are based on the static analysis ceriteria.The presently coupling metrics process measures can be broadly divided into the following two groups:

- Metrics process of coupling for measures of Procedural programming: It measures the coupling process of the software components which are implemented in the procedural programming languages and their examples includes metrics that are proposed by Briand ;Cartwright and; Chaumun and Kemerer This class of metrics process is heavily connected by the classification of the various coupling levels
- Metrics process of coupling for measures of object oriented : These measure the coupling process of the software components which are implemented in the object-oriented programming languages and their examples includes metrics that is proposed by the Chidamber Deligiannis and Freund and Wilson.

E. Package Level Metrics

Packages are defined as re-usable components for the modern object-oriented systems. Package was commonly consisting of classes, interfaces and the sub-packages concept. To promote reuse criteria in the object-oriented systems and to make the tasks like deployment and maintenance quite easy, in that type of case packages in object-oriented systems should follow the basic principles of design i.e. maximum cohesion and the less coupling. The process of coupling between packages is defined as the degree of the interdependence between them. Each and every package is a stand-alone unit.

II. RELATED WORK

A. Role of software metrics in software development projects.

Software metrics were increasingly playing a central role in the planning process. Existing literature on the software metrics process is its main focus on the centralized systems, while work in the area of the distributed systems, mainly in the service-oriented systems is very scared. The Distributed systems that are having the service based components were even more heterogeneous networking and execution of environment. In past, the coupling measures take into account only the "static". They do not even have account for the "dynamic" coupling due to the polymorphism and may significantly under estimate the complexity of the software. This is expected to be result in the poor predictive of models and accuracy of the quality models in distributed Object Oriented systems that utilize the concept of static coupling. To find a solution to all these kinds of issues, A hybrid model is proposed in the Distributed Object Oriented Software system for the measurement of coupling dynamically. In this proposed method, there are few steps such as Instrumentation, the process of Post processing and the Coupling measurement concept. Firstly the instrumentation process is being done. In this method, the instrumented JVM has been modified to trace the method calls. In between this step, three trace files were generated called .prf, .clp, .svp. In the second step, all the information which is collected in all file were merged together. At the end of this step, the merged detailed trace of each and every JVM contains pointers to the merged trace files of the other JVM file such that the path of the every remote call from the client to the server can be uniquely identified .At last, all the coupling metrics has been measured dynamically. All these implementations results show that the proposed system will be effectively measures the coupling metrics dynamically.

B. Coupling metrics for the analysis of object-oriented system.

Coupling is one of the most vibrant quality concept that has been mostly worked upon to get a measure of external quality attributes like maintainability, reusability, functionality etc for the object-oriented systems. A large number of the static coupling metrics has been devised over the years to measures coupling metrics process. But metrics were present which are helpful for studying the actual coupling behaviour of the software at the runtime status. There is a strong need to study the utilization of static and dynamic metrics when used in combination. This paper of research on the concept of dynamic coupling and dynamic metrics inheritance, object-oriented concepts will proposed the new set of class level dynamic metrics to measure the coupling metrics process at the runtime status and is characterize the ability of such metrics to calculate the external quality attributes of a software design and to compare the coupling assessments that is achieved from the static and dynamic analysis of metrics. After all that a study is conducted which contains applying the proposed metrics to assess the quality of the java-based object-oriented applications.

C. Dynamic software metrics concept

Conventional static metrics process have been found to be an inadequate and in complete for the modern object-oriented software due to the presence of the object-oriented concepts like polymorphism process, data hiding, inheritance process and unused code. This fact motivates researchers to focus on the dynamic metrics process in the place of traditional static metric process. Moreover, dynamic metric process were more precise and useful and capable than static metrics as they are able to captures all the dynamic behaviour of the software system during the measurements. All types of these dynamic metrics were usually obtained from the execution traces of the code or from the executable models which is available for use. this paper having the use of dynamic metrics over static metrics process were discussed and then a study of existing dynamic metrics is carried out from all this. These both metrics were characterized into the different categories such as dynamic process, and the metric process named dynamic cohesion.

D. Tracer application used for automatic concurrent testing

Designing and after then developing a good quality software product requires a efficient measures to accurately judging all the internal software quality attributes like coupling, cohesion, size, attributes and complexity, throughout the course of a software development life cycle(SDLC) process. The Software metrics has been mostly and successfully used to measure such internal quality attributes for the object-oriented software systems. Coupling defined as the most basic and useful qualitative measures for measuring the performance of the software at the design phase or implementation phase of the development cycle and normally stated as the degree of interdependency among some modules. Coupling justifies all the external complexity of the class, i.e. how these classes are dependent on other classes. Low level of the coupling is desired among the classes of an object-oriented application. Thus, the theory behind this measure contributes to the external quality concept of a software application, like its maintenance, reusability, functionality reliability, testability and the tolerance from fault etc. The metrics which are available for coupling measurement belong to two major categories which are static and the dynamic process. Static process metrics which are applied to the design/source code can only measures the expected coupling behavior of the object-oriented software and not the actual behavior. The reason behind is that the behavior of a software application is not only influenced by the complexity factor but also by the operational or runtime environment of the source code. On the other hand Dynamic metrics can capture all the actual and run time coupling behavior as they were having from the data collected during the runtime entity.

E. Static and Dynamic process Analysis of the Object Oriented Systems

A software product which requires a efficient measures to correctly monitor the internal behavior. Software metrics process has successfully and mostly used to measures such internal quality attributes for object-oriented software systems. The term Coupling is one of the most basic qualitative measures term for measuring the performance of software at design or at the implementation phase of life cycle of software development. And it is defined as the degree of the interdependent among the all types of modules. Metrics which are present for the coupling measurement is divided into two major categories which are static metrics and the well-known dynamic metrics. The concept of the Static metrics are defined as those which are applied to design/source code can only measures the expected coupling behaviour of the object-oriented software process and not the actual behavior of the software process. Dynamic metrics concept is helpful in capture all the actual coupling behavior as they are evaluated from data which is collected during the runtime entity.

F. Analysis and study of Lack of Cohesion Metrics for the Predicting Testability of Classes

Relationship between the lack of cohesion and the testability of the classes in the object-oriented systems is studied and analyzed. Here we introduce the concept of testability from the perspective of unit testing term. An analysis is done using data which is collected from the two Java software systems for which test cases of JUnit were exist. For captures all these testability of the classes, we have to use different types of the metrics which is used to measure the characteristics of cases of JUnit. By using and analysing he Receiver Operating Characteristic (ROC) analysis, the performance of the system and the result of predicted model was then calculated and evaluated. A result which provides evidence that there exist a relationship between the lack of cohesion and testability is evaluated.

G. Estimation of Testability framework.

"Testability" always been an elusive and important concept and its correct evaluation or measurement founds the difficult problem. Most of case studies examined that the term testability or more precisely attributes that having their impact on this concept. The concept of the process Testability at source code level is good indicator of the effort estimate also it leads to the late arrival of the information in the development process cycle of software and after this a decision is made to change the design in the order to improve the testability after coding has started and may be very costly. On calculating the concept of testability in early in the development process cycle may greatly reduce the overall cost. This paper gives the direction to the industry personnel and researchers to continue further research, and also quantify the concept of software testability in the design step. A job is done in order to get the integrate testability within the development life cycle process of software cycle and used as benchmark to software products according to the need of testability.

H. Implications of the software defects

To produce the high quality of the object-oriented concepts, strong and hard study on the design aspects, mainly occurs during the early phases of the software development which is very necessary and important. The design metric plays an important role for helping developers to understand design phases of the software and also improves the software quality and developer productivity throughout the process also. This empirical evidence which are supports the role of object oriented concepts of design complexity metric process, helps in determine the defects .The results are then based on the industry data from software developed in two popular programming languages used in OO development, that indicates the size of the software, metrics are related to each other with all these defects. In addition, the effects of all these metrics on the defects vary across the samples from two programming languages like C++ and Java. All these types of results having a significant implications for designing high-quality software products using the OO approach.

Table I Summary of various Filtering Methods

AUTHOR/YEAR	TECHNIQUE USED	FINDING
Paramvir singh and hardeep singh/2010	Class level metrics process	The four types of dynamic coupling process metrics were designed to accesses the quality of the object-oriented software system.
S.Babu and R.M.S parvathi /2011	Hybrid model is used to measure the dynamic coupling.	Many types of coupling measurement is studied and analysed in the object-oriented software and propose and examined the hybrid model for measurement of dynamic coupling.
Linda Badri,Mourad Badri and Fadel Toure/2011	Testability of class	Empirical study on data which is collected from two open source named java software system for which test case of junit exist.
Kritika /2013	Coupling Measurement	It is found that AOP approach provides a balanced method for the dynamic analysis of the programs which is being used. The Tool which is used name is "N crunch", it is concurrent automated tool of testing is successfully studied. N crunch provides you a large amount of useful information your tested code such as code as code coverage and performance metric.
Ramanath Subramanyam and M.S. Krishnan/2003	CK Metrics	The CK Metrics were analysing that the defect counts as indicators of quality of the OO system, allowing for more accuracy in the dependent variable. This approach has the license for the complexity effect from interaction between the two CK metrics which may play an additional role in explaining all defects.

III. CONCLUSION

Four types of dynamic coupling metrics were designed to allow the quality of the object-oriented software system. The dynamic metrics gives the idea of the real behavior of a software at runtime entity in the comparison to the static metrics that provides an idea of the actual behavior of the system. It is found in research paper that the AOP approach provides a balanced method for the dynamic analysis of programs. The tool which named as N crunch is used an automated concurrent testing tool is successfully examined. The tool having the name N crunch provides you a large amount of useful information to your tested code like code coverage and performance metric and therefore it is analyse that the defect counts as the indicators of quality of the OO system concept, allowing for the more variability in the dependent variable. This approach also allows for the complexity effects between the interactions between two CK metrics that may plays an additional somewhere important role in explaining the defects.

REFERENCES

[1] S.Babu and Dr. R.M.S Parvathi,"Development of dynamic coupling measurement of distributed object oriented software based on trace events, International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.1, January 2012

[2] Paramvir Singh1 and Hardeep Singh" Class-level Dynamic Coupling Metrics for Static and Dynamic Analysis of Object-Oriented Systems", International Journal of Information and Telecommunication Technology, IJITT, Vol. 1, Issue 1, 2010

[3] Jitender kumar chhabra and varun gupta"a survey of dynamic software metrics", journel of computer science and technology, 1016-1029, sept2010.

[4] Kritika." Review Paper on Static and Dynamic Analysis of Object Oriented Systems International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 9, September 2013.

[5] Divya Sharma "Review Paper on Static and Dynamic Analysis of Object Oriented Systems", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 4, April 2014

[6] Linda Badri, Mourad Badri & Fadel Toure" An Empirical Analysis of Lack of Cohesion Metrics for Predicting Testability of Classes", International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011

[7] Mohd Nazir, Dr. Raees A. Khan, Dr. K. Mustafa" Testability Estimation Framework", International Journal of Computer Applications (0975 – 8887) Volume 2 – No.5, June 2010.

[8] Ramanath Subramanyam and M.S. Krishnan "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", IEEE transaction on software engineering, VOL. 29, NO. 4, APRIL2003.