

Survey Paper on Coupling Measurement Tools for Object-Oriented Software

Sampada P. Kale¹ Prof. V. S. Bidve²

¹Student ²Assistant Professor

^{1,2}Department of Information Technology

^{1,2}Smt. Kashibai Navale College of Engg., Pune

Abstract— In software engineering, coupling is the manner and degree of interdependence between software modules. It is a measure of how closely connected two routines or modules are, the strength of the relationships between modules. Low coupling is a sign of a well-structured computer system and a good design. A large numbers of metrics have been projected for measuring properties of object-oriented package like size, inheritance, cohesion and coupling. As object oriented analysis and design techniques become widely used, the demand on assessing the quality of object-oriented designs substantially increases. Recently, there has been much research effort to develop and empirically validate metrics for OO design quality. The coupling measurement metrics are more useful when they are correctly measured. This paper contains the survey of types of coupling measurement tools that are available till the date.

Key words: Measurement, Metrics, Object Oriented Programming, Object Oriented Metrics, Coupling

I. INTRODUCTION

Coupling is one of the most important properties that affect the quality of the design and implementation of a software system. In software engineering, the coupling is the manner or degree of interdependence between software module, a measure of how closely connected different routines or modules are and the strength of the relationships between system modules. Coupling is contrasted with cohesion. Low coupling is correlated with high cohesion, and high cohesion is associated with low coupling. Low coupling is a sign of a well-structured computer system and a better design. Software metrics helps us to make meaningful estimates for software products and guide us in taking technical and managerial decisions. However, traditional static metrics has been found to be insufficient for new object-oriented software due to the presence of object-oriented characteristics such as inheritance, dynamic binding, and unused code. This fact motivates us to concentrate on dynamic metrics instead of conventional static metrics. Evaluating run time structure at design level is essential. The system's run-time structure must be focused more by the designer than the language. The relationship between objects determines how good or wrong the run-time structure is. Therefore, it must be designed with great care. The coupling measurement metrics are more useful when they are correctly measured. If the metric measurement is error prone it can affect the complexity of the system. The extra effort in developing executable models should be justified for many complex real-time applications. This paper contains survey of the tools for measurement of Object-Oriented software.

A. Motivation

- To study the various types of Coupling Measurement Metrics for object-oriented software.
- Most of the study in this field is made theoretically, and all the research should be organized in such a way that it should have much impact on industry practices.
- Developing an effective coupling measurement tool for object-oriented software by reviewing various types of coupling.

II. RELATED WORK

A. *Work by Jarallah Al ghamdi, Mohamaud Ellis, Moataz Ahmad [1]*

The measurement of inheritance coupling. It support over twenty three metrics related to inheritance coupling measurement. In this the data is parsed and then it is abstracted to the language dependent format. Other Object-Oriented parser can be implemented without making changes to the system. This tool is made up of three components. Parsing Engine, central metric repository, and query engine. Parsing engine is used to extract the information needed for measuring the inheritance metrics. Parser is Object-Oriented language dependent. Central metric repository is responsible for all the calculations needed for calculating the metric values. Query engine is used for applying SQL queries to the central metric repository to get the results. This tool was implemented by using Javadoc in JDK 1.2, Microsoft Visual Basic 6.0 and Microsoft Access 97. Representation of tool is separated from calculation procedure.

B. *Work by Erik Arisholm, Lionel C. Briand, Audun Føyen [2]*

In this paper the tool named JDissect is described. It is used for calculating dynamic metric measures. Collection and analysis of metrics are done in two phases. In first phase data is collected from Java program and it is stored. Library of data collection routines is loaded into Java Virtual machine for collecting and storing the data. In second phase the data is analysed. This tool uses the interface of JVM for getting the message traces and other dynamic analysis information.

C. *Work by Aine Mitchell and James F. Power [3]*

In this paper the dynamic metrics are evaluated by using Java Platform Debug Architecture (JPDA). It consists of two interfaces, the Java Virtual Machine Debug Interface (JVMDI), the Java Debug Interface (JDI) and a protocol the Java Debug Wire Protocol (JDWP). It also has two software components which tie them together the back-end and the front-end [3]. JVMDI is used to control the state as well as

execution of Java program running into the Java Virtual Machine. It is lowest level of JDB. JDWP is used to define the format of information; it works as a parser in other tools. Dynamic metric measurement is done so that it provide supplement to the existing static metric.

D. Aine Mitchell and James F. Power[4]

In this tool Apache Byte Code Engineering library is used for metric data collection. It can be used to create, manipulate and analyze the Java source code. Using the instrument or program BCEL adds probes to the Java program to flag events like object creation, function call etc. In instrumentation process the file having information of field access and method is created. Information of each method and field is given a unique index in this file. When the application runs the probes inserted earlier hit in another file made from instrumentation process. By using these files metric program calculates run time program.

E. Work by A. Kavitha, Dr. A. Shanmugham[5]

This paper illustrates a dynamic coupling measurement tool which first introspect the source code and add trace events to it. This source code is compiled and executed. While it is running the trace events are logged. Log provides the actual number tool presented in this paper is specially designed for of function called while running the program. Actual runtime used source code is filtered from the code and this source code is given to any standard coupling measurement technique for measuring dynamic coupling. The trace events are added to each method of all classes.

F. Work by J. Offutt, A. Abdurazik, S. R. Schach [6]

J. Offutt, A. Abdurazik, S. R. Schach [6] defined a tool named JCAT for the analysis of structure and different components of java source code packages and find out the coupling among classes. The type of coupling used in this research are Parameter coupling, External/File coupling, Inheritance Coupling, Global Coupling. This tool was developed in Java which collaborates with number of software applications like Java parser which is generated by ANTLER (Another Tool for Language Recognition), Access and Excel. It accept Java source code file path name as an input then by using Java parser it generates abstract syntax tree files for every source code file in that package. JCAT find information of classes, parameters, methods and their definition by extracting data from AST files. After proceeding all ASTs JCAT calculate coupling with the help of SQL queries and the results are saved into Excel spreadsheet and ASCII Text File.

G. Work by Husein and Oxley [7]

Husein and Oxley [7] introduced the coupling and cohesion metrics tool called CCMETRICS for calculating coupling and cohesion metric and implemented it for the object-oriented software systems. The formalized metrics have been applied in CCMETRICS. In this tool, the tool receives Object-Oriented software source code. This input is then parsed according to the grammar provided by the developer. ANTLER is used to generate the grammar. According to the rules defined the parsed input is reused by extracting keywords. The various coupling measures used in this paper are Method invocation Coupling, Global data coupling, Data abstraction coupling, Inheritance coupling [7]. Coupling

metrics, defined here, indicate how dependent a class is on others. The higher the metric value, the more dependent the class is. This means that the class significantly uses, and thus relies on, other classes to complete its tasks.

H. Work by N. Kayarvizhy1 and S. Kanmani [8]

This tool converts any particular object oriented language to the XML format that is language independent. This XML formats are later used for calculating the metric values. The proposed tool in this metric contains the parser, XML converter, generic XML container, information extractor and metric collector. Parser is use for breaking the source code into the constructs like classes, methods, attributes, etc. Generic Xml converter is used to provide XML file. Generic XML container is used for making the metric calculations easy; it is common for all the Object-Oriented languages supported by the tool. Actual calculations of coupling measurement metrics are done in the metric calculator part. It can be extended easily to add new measurement metric. By using this tool the researcher can add new metrics to the existing tool.

I. Work by S. K. R. Yeresime Suresh, Jayadeep Pati[9]

S. K. R. Yeresime Suresh, Jayadeep Pati[9] not only illustrates a huge benefit in the usage of software metrics for Software Quality Assessment but also illustrate about evaluating the system effectively through these metrics[9]. In this paper, the metrics like Parameter coupling, External/File coupling, Inheritance Coupling, Global Coupling are evaluated. Real -time application i.e. ATM is evaluated for the coupling measurement by using the above parameters[9]. Software metrics as mentioned in this paper help us to assess the complexity, fault proneness or the system reliability of software. The results obtained in this paper provide a basis for quality assessment to software professionals to fetch the required amount of information about those metric suites that can predict faults while developing the metric-quality software products, estimate a number of test cases required to get maximum code coverage and also to know the complexity of the system using the Traditional and Object-Oriented approach [9]. In this paper the metrics values are extracted with the help of eclipse.

J. Work by Rani Geetika, Paramvir Singh[10]

According to the author it is necessary to collect accurate dynamic metric data to evaluate it. Dynamic profiling is used to collect data from the particular java program. To evaluate the dynamic coupling data selected java application is given as an input to the metric evaluation system. The system uses Javassist library for byte code instrumentation. Depending on the user need the byte code data is sent to the static or dynamic evaluator. The log file is created at runtime that contains the event traces like method call, object creation, etc. The dynamic metric evaluator calculates dynamic metric values by using the run time traces. Comparison between static CBO and dynamic CBO shows that dynamic CBO capture more information than that of static CBO and coupling before runtime is not same as the coupling at runtime.

III. DISCUSSION AND COMPARISON OF EXISTING WORK

Now a day, software engineering provides the best solutions for the software problems in object-oriented programming. Perfect measurement is an necessity for all engineering disciplines and software engineering is not an exception. To improve software products and process,

measures are essential. Software analysis plays a significant role in finding the software quality, performance, maintenance and other characteristics of software products. The concept of measurement requires appropriate measurement tools to measure, to collect, to verify and validate related metric data.

Paper Name	Platform used and Languages Supported	Advantages	Limitations
Work by Jarallah Al ghamdi, Mohamaud Ellis, Moataz Ahmad [1]	Platform-Javadoc in JDK 1.2, Microsoft Visual Basic 6.0 and Microsoft Access 97 Languages supported-C#, Java, Smalltalk	Supports more than one language.	Only inheritance coupling can be measured by using this tool.
Work by Erik Arisholm, Lionel C. Briand, Audun Føyen [2]	Platform: Java Virtual Machine , JVM Profiling Interface, JVM Debugging Interface	-Less overhead than the tools using instrumentation.	JDissect tool cannot be used for discovering the dead code.
Work by Aine Mitchell and James F. Power[3]	Java Platform Debug Architecture (JPDA)	Dynamic metrics are used to evaluate actual runtime properties of class and it can also be used to evaluate external quality aspects of the program.	Level of coupling between individual objects cannot be calculated by using the tool defined in this paper.
Aine Mitchell and James F. Power[4]	Platform: Sun's JVM profiler API, the Java Platform Debug Architecture (JPDA)	Proves that run time coupling metric values vary from the static metric values	-Instrumented code is tested not the original one this can lead to change in output and system state. -For instrumentation source code is necessary that is why the program containing just library function and not the source code cannot use this tool.
Work by A. Kavitha, Dr. A. Shanmugham[5]	In this tool the Actual source code used at runtime is given to any existing metric measurement tool.	Results show that coupling at runtime differs from the static coupling measures.	Introspection procedure can be errorprone.
Work by J. Offutt, A. Abdurazik, S. R. Schach [6](2008)	Platform: Java, ANTLER, Excel Language Supported: Java	Quantitative measurement of coupling metric can be calculated more precisely.	The tool defined in this paper does not differentiate among the different variations of coupling. -Not available before implementation.
Work by Husein and Oxley [7](2009)	Platform: Java, ANTLER Language Supported: Java	Software quality measurement can be done by using the defined set of metrics	Do not deal with various types of couplings which are important for considering object oriented features
Work by N. Kayarvizhy1 and S. Kanmani [8]	Platform: Java, XML, Excel. Languages Supported: Java. C#	Can be extended for other Object-Oriented languages.	Scope for adding more coupling measures.
Work by S. K. R. Yeresime Suresh, Jayadeep Pati[9]	Platform: Java Languages Supported: Java.	Metric values are evaluated for real time application which helps us to know the reliability and complexity of the software. -Metric values give clear description of stable software.	This study can be further extended by the use of cyclomatic complexity.
Work by Rani Geetika, Paramvir Singh[10]	Platform: Java Language Supported: Java	Both static CBO and dynamic CBO metrics can be calculated.	Only coupling between objects id calculated. Scope for adding more types of couplings. -This analysis should also be done by using other languages

			to check impact of language on coupling metrics.
--	--	--	--

Table1: Summery and comparison of existing work

IV. PROPOSED WORK

The proposed work is to develop a software system (tool) to measure coupling in Object Oriented software using Java to present coupling measures to the software professional as quality software parameters like complexity, reusability, reliability, maintainability. We are proposing this tool to analyze the structure and components of Java source-code packages and identify couplings among classes. This tool is a source code analysis tool that is to be developed in Java and collaborates with several software applications, including a Java Parser, Java source, Entity Data Structure.

V. CONCLUSION AND FUTURE WORK

Various Coupling Measurement metrics tools are reviewed in this report. Due to the minimization in coupling, developers can produce better quality programs. When the coupling is less, the classes can function more independently. The independent class is easier to be reused by any other application. Coupling measurement is the essential factor, and it must be accurate. Although new advanced coupling metrics has been proposed researchers should now mainly focus on applying existing metrics for very different purposes such as the identification of components in large code bases or the discovery of security issues. Also, these metrics should be useful to the software professionals to control the quality software parameters.

REFERENCES

- [1] "Tool for Measuring Inheritance Coupling in Object-Oriented Software"
- [2] Erik Arisholm, Lionel C. Briand, Audun Føyen,"Dynamic Coupling Measurement for Object-Oriented Software", IEEE transactions on software engineering, vol. 30, no. 8, august 2004, pp.491-506
- [3] Aine Mitchell and James F. Power," Runtime Coupling Metrics for the Analysis of Java Programs - preliminary results from the SPEC and Grande suites
- [4] Aine Mitchell, James F. Power "Using Object Level Run Time Metrics to Study Coupling Between Objects", 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico March 13 -17, 2005
- [5] A Kavitha, Dr. A. Shanmugham,"Dynamic coupling measurement of trace events" SAMI, 6th International Symposium on Applied Machine Intelligence and Informatics, 2008, pp.255-259
- [6] J. Offutt, A.Abdurazik, S. R."Schach, Quantitatively measuring object-oriented couplings." Software Quality Journal, Springer 16(4), 2008, pp.489-512,
- [7] S. Husein, A. Oxley, "A Coupling and Cohesion Metrics Suite for Object-Oriented Software," IEEE ICCTD '09, 2009, pp. 421-425.
- [8] N. Kayarvizhyl and S. Kanmani,"An Automated Tool for Computing Object Oriented Metrics Using XML", First international conference, ACC 2011, Kochi ,India, July 22-24,2011, pp.69-79
- [9] S. K. R. Yeresime Suresh, Jayadeep Pati, "Effectiveness of software metrics for object-oriented system,"

- Elsevier, Procedia Technology, vol. 6, 2012, pp. 420-427.
- [10]MetricsRani Geetika, Paramvir Singh,"Empirical Investigation into Static and Dynamic Coupling", ACM SIGSOFT Software Engineering Notes Volume 39 Number 1,2014