Metrics for Component Based Measurement Tools

P. Edith Linda, V. Manju Bashini, S. Gomathi

Abstract— The main aim of this paper is to integrate the different object oriented metric tools and make them available as a single add-on. The first part of this paper analyzed five different tools and they are migrated into one to make use of those tools in efficient manner.

Index Terms— Metrics, component, de-compiler, SLOC, LOC, NOS, NOP, Cyclomatic complexity

1 Introduction

THE emerging Object oriented programming is most widely used by the programmers to develop the software. The object oriented programs are used in the IT world which is used for secured transactions and mainly convenient. Even though they are widely used, the complex and confusing structure may arise if the program is not properly measured and not properly structured. To overcome this problem many proposed varieties of metric tools to measure the criteria of Object oriented programming are used. Again the problem arises that some metric tools measure the particular criteria of the object oriented programs. In order to overcome the individuality of those tools, they should be migrated.

Component based measurement tool is the concept of migrating many tools available widely. The main aim is to search some important tools and making them as a single add-on to provide the user friendly environment.
Each individual tool will measure some parameter to measure the java program. But a single tool will not measure all the parameters.

Thus there exist a need of collection of tools to measure the java programs which will satisfy all the parameters to be measured.

2 Metrics

Software metrics are used to measure the software quality to check whether it satisfies the requirements. Metrics are defined as "Quantifiable measures that could be used to measure characteristics of a software system or the software development process." Software metrics are essential to plan, predict, monitor, control, evaluate, products and processes. The main goal of the software metrics is to reduce costs, Improve quality, Control/Monitor schedule, small testing effort, many reusable fragments, to better understand the quality of the product and the program.

3 Metrics Categorization

Metrics can be categorized into three Kinds, measures and size.

Two *kinds* of software metrics they are Product metrics: quantify characteristics of the product being developed to measure the size, reliability and Process metrics: quantify characteristics of the process being used to develop the software to measure the efficiency of fault detection.

Types of *Measures* are Direct Measures (internal attributes): to measure the Cost, effort, LOC, speed, memory and Indirect Measures (external attributes): to measure the Functionality, quality, complexity, efficiency, reliability, maintainability.

Size-oriented metrics are *LOC* - Lines Of Code, *KLOC* - 1000 Lines Of Code, L*OC* – Statement Lines of Code (ignore whitespace).

4 Component

Components are the collections of many pre programmed tools which is used as the add-on page which is to make use of those tools. There are various tools available to measure the Java source code. Those tools are developed

- *P. Edith Linda, Assistant Professor, School of IT & Science, Dr. G. R. Damodaran college of science, Coimbatore-641014* p.lindavinod@gmail.com
- V. Manju Bashini,, *Assistant Professor, School of IT and Science, Dr. G. R. Damodaran college of science, Coimbatore – 641014* manjubashini@gmail.com
- *S. Gomathi, Research scholor, Dr. G. R. Damodaran college of science, Coimbatore- 641014.* Gomathisrinivasn88@gmail.com

and modeled based on the developer's point of view. Some tools will measure some parameters to be measured in java program. Initially the tools which are used to measure the java object oriented programs are searched and analyzed individually to make it as a component. The component based tools will be executed separately but available in a same location and some tools will provide a chart for the results when the program is executed.

5 EVOLUTION OF COMPONENTS

In 1990's, software systems for e-business transactions were built of components across multiple platforms, programming language and network protocols. No tools were available for detecting "bugs" or to measure the efficiencies in the programs (object oriented programs). Initially the errors are printed using the output statement and the number of lines [1] , number of packages and most important concepts of object oriented programs with the critical programs

Those programs which are used to measure the OOPS programs seem to be critical and they are implemented with the large number of coding. In 1997, Intermetrics, a software engineering company , proposed to develop a debugging tool for Component- based software systems. It only identifies the errors and yields the error free program.

But finding the error won't give the remedy. Thus the problem to find the efficiency of the program is solved with the help of the component based tools which efficiently provides the charts for the results and measures the program.

The main advantage of component based tool is that the_user can select their own tool to measure their program according to their needs. In this, the user can also see the links from where the tools are available, that is the user can also download or know more about the tool from the direct link.

The collection of more tools which will provide more flexibility and choice to the user or the programmer and he can know more about tools. Those tools will display an error if the program is not correct.

Some tools will also provide the charts and warning message if the program is not structured and programmed in a proper format based on the measurement criteria.

6 COMPONENT INTEGRATION

All the tools won't meet the user required criteria. Some may be useful for generating report and some may be used just for measuring the program. Some tool may measure the oldest method of measuring the program with some limited parameters which are not in current use and which may be efficient to measure. All the listed tools will measure the common parameter like SLOC (Source Line Of Coding) and LOC (Lines Of Coding). The need depends on the programmer or the user who uses the component metrics. Integrating the various tools have many advantages and also some disadvantages. Those are listed.

Advantages on integrating components

1. Flexible
2. User friendly
3. Compatibility
4. Comparability

Disadvantages on integrating components

1. Confusion may arise about which tool to be used
2. Some may not meet the user requirement
3. Some tools are only trial versions
4. All the tools which are listed may have some common criteria which is not useful.

7 RELATED WORKS

The paper by Amandeep Kaur et. al. [7] to formulate a framework of metrics representing the attributes of object oriented system. And the suggestion is that the impact of OO design on software quality characteristics made an experimental validation and a rework in defect programs. Encapsulation, inheritance, polymorphism, reusability, Data hiding and message-passing are the major attribute of an Object Oriented system and are the indicators of the object oriented programs. The well known quantifiable approach to express any attribute is a metrics. Empirical Data is collected from three different projects based on object oriented paradigms to calculate the metrics.

Lakshmi Narasimhan and Bayu Hendradjaya[8] defined a two suites which cover the static and dynamic aspects of the component assembly. The triangular metric which is used to classify the type and nature of the application, is formed with the complexity and criticality metrics. During the runtime of the complete application the dyamic metrics are collected. This paper is mainly deals with the collection of some metrics and evaluated the results.

Shyam R. Chidamber and Chris F. Kemerer [11] stated that the metrics is must and needed for a new organization which is adopting a new technology for whom the proper practice is needed to evaluate the program size and other paradigms. To practice and to evaluate the programmer need to implement the a new metric suite for Object Oriented programs for OO design. The empirical validation of a theoretical metrics of OO design and the development is the main idea of the paper.

Dr. Linda H. Rosenberg [12] stated the development of object oriented programs is very much needed in today's life cycle. A different approach of design and implementation, and also a different approach to software metrics is required by Object oriented development. Object oriented technology uses the objects as its fundamental blocks and not algorithms. The object oriented programs needs a special metrics to evaluate. Cyclomatic complexity is standard for traditional functional and procedural programs. There are many proposed object oriented metrics for evaluating the object oriented program.

Dr K. Anbumani, K.P. Srinivasan [13] stated use of software metrics in evaluating the quality of software products and the software development process. There are currently three sets of design metrics for object oriented software which are suggested by different groups of authors, Chidamber and kemerer metrics.

Lakshmi Narasimhan et. al.[14] of Department of Computer Science from East Carolina University, Greenville, NC, USA suggested that during the execution phase, there exist lack of metrics which will reduce the maintenance costs and defined metrics whose values are collected. Metrics are useful for increase the re-usability and assessing the maintenance cost of individual components and that of the application in which the component is integrated.

Radu Marinescu [15] given that the object oriented paradigm has influenced the software engineers for ease of program writing and implementing. The impact of misunderstanding the object oriented designs made the design complex and more tedious to program and poor OO design. Object oriented metrics have the benefits with the high degree of code reuse, higher maintainability and flexibility, etc., Now they can improve the quality of the design using the object oriented metrics.

Lakshmi Narasimhan and Bayu Hendradjaya[16] defined a two suites which cover the static and dynamic aspects of the component assembly. The triangular metric which is used to classify the type and nature of the application, is formed with the complexity and criticality metrics. During the runtime of the complete application the dyamic metrics are collected. This paper is mainly deals with the collection of some metrics and evaluated the results.

Magnus Andersson Patrik Vestergren[17] targeted to evaluate whether the software metrics can be used to determine the object-oriented design quality of a software system. There are several metrics to evaluate the object oriented design and programs. To validate each metric an experimental study was conducted. The final conclusion is software metrics can improve the system design quality such as complexity of methods/classes, package structure design and the level of abstraction in a system. Therefore metrics can be assured that the rules are followed and the program is properly written with the rules.

According to Saida Benlarbi et. al.[18] the basic premise behind the object oriented metrics which is an earlier predictors of classes that contains error which may be found during the testing. In this paper they conclude that the empirical validation methods provide misleading conclusion to the validity of object oriented metrics. The result of the empirical validation is studied with the C++ which is an object oriented program.

The paper by Tieng Wei Koh et. al.[19] review the 12 objected- oriented software metrics proposed in 90's by Chidamber, Kemerer and Li. The authors stated that the software cost estimation is mainly based on the size or predicting the volume of various kinds of software deliverable items. Size and complexity is very critical and it influences the integral software development effort. Software cost estimation is criticsl because it is mainly based on the software size estimation.

Marco Scotto et. al.[20] metrics tools is to be updated for the standardization of software measures but it is not still standardize. Intermediate abstraction layer is the possible solution suggested here to decouple the extraction process from the use of the information. This paper presents the web metrics which is an automated tool for software metrics collection. This open source project is computed with the CK metric suite.

8 EXISTING SYSTEM

The tools which are analyzed in this paper are LOCC, JHAWK, CODE COUNTER. These tools are used to measure the object oriented paradigms in the program. The main advantages of these tools are, it will generate the charts and will produce the reports about the programs. The programmer should re-model or re-structure the program according to the report or the warning chart generated by those tools[9,10]. Another software called cavaj Decompiler which is used to convert the class files into java files for user convenience is also integrated into this component.

8.1 LOCC

This is the tool to measure the Object oriented program to yield the LOC, SLOC, NOP, NOC, NOM, NOI in that particular program which is given as input. The main drawback of this metric tool is that it won't generate the charts for the report or result generated by the tool [4]. LOCC metrics, which provides a grammar-based architecture and interface to the production and use of different size metrics. Developers can use the size metrics distributed with LOCC or can design their own metrics, which can be easily incorporated into LOCC. LOCC pays specific attention to the problem of supporting incremental development, where a work product is not created all at once but rather through a sequence of small changes applied to previously developed programs.

8.2 JHAWK

This tool is used to measure the various parameters of CYCLOMATIC COMPLEXITY, COMMENT LINES, NOP in the given input program. The main advantage is that it will generate a chart for the program and also it will show the warning chart. It will mainly measures the looping and iterations which are present in the input program[6].

JHawk is a Stand alone, Eclipse plugin and command line versions. JHawk Metric interchange format that allows you to record a snapshot of a code base and keep it for future use either in JHawk itself or in our new DataViewer product. JHawk Data Viewer, a standalone product that allows graphical and textual comparison of metrics over time.

The disadvantage is that the Halstead metrics are not used now for measuring the program. Some may need to measure those metrics.

8.3 CODE COUNTER

This metrics is mainly used to measure the NOS, COMMENT LINES, TOTAL of the two metrics. The main advantage of this tool is, it will measure and generate the report for many languages like ASP, HTML, C++, C etc. The main advantage is it will measure many languages and will generate the three forms of reports according to the user needs. The main drawback is it won't cover the important parameters to be measured for the object oriented programs [4].

The main features of code counter are Easy to use with just 3 steps, Quickly counts several types of source code - including Java, JSP, C or C++, VB, PHP, HTML, Delphi or Pascal, ASM, XML, COBOL etc., Smart Comment feature - knows which comment types are used by each language and counts accordingly, Save and load count profiles - no need to enter the same information over and over for the same source files to count, Supports unicode files and contents. Can also count unix type or windows type of text files (CRLF/CR only)

8.4 LOC METRICS

This tool is mainly used to measure the total lines of code (LOC), blank lines of code (BLOC), comment lines of code (CLOC), lines with both code and comments (C&SLOC), logical source lines of code (SLOC-L), McCabe VG complexity (MVG), and number of comment words (CWORDS). Physical executable source lines of code (SLOC-P) is calculated as the total lines of source code minus blank lines and comment lines. Counts are calculated on a per file basis and accumulated for the entire project. Loc Metrics also generates a comment word histogram[4].

8.5 SLOC METRICS

SLOC Metrics measures the size of your source code based on the Physical Source Lines of Code metric recommended by the Software Engineering Institute at Carnegie Mellon University (CMU/SEI-92-TR-019). Specifically, the source lines that are included in the count are the lines that contain executable statements, declarations, and/or compiler directives. Comments, and blank lines are excluded from the count. When a line or statement contains more than one type, it is classified as the type with the highest precedence[5]. The order of precedence for the types is: executable, declaration, compiler directive, comment and lastly, white space.

Main features of SLOC are Fast and efficient processing of source files, Process any size of file, Process any number of files. Flexible support for C/C++, C#, Java, HTML, Perl, Visual Basic and more. Ability to add definitions for other file types, Formatted and hyperlinked results in HTML file format. Project folders and files are sorted by their Source Lines of Code so that you can easily identify code intensive modules and files, Shows differences in counts of source lines of codes between two measurements by SLOCMetrics as DIFF.html in the output folder , Formatted results in CSV (Comma Delimited) file format, Prepares results as a well-formed XML document , Multiple SLOC Configurations can be saved as files and reused, can be run as a Console application using *sloccmd.exe* present in the SLOCMetrics 3.0 installation directory. Configuration file path can be passed as a parameter to sloccmd.exe, An easy to use, Windows-based graphical user interface.

9 PROPOSED SYSTEM

Main drawback of the existing system is they are available individually. So the users may meet some difficulty in order to measure the program.In this proposed system, those tools are integrated and made as a single add-on for the users flexibility and user friendliness. All the tools are available in the single web page so that the user can access those tools efficiently. The java decompiler, an open source, also combined in this page which will convert the class files into java files.

The tools will measyre the different parameters which need to be measured. The tools itself will generate the charts and reports according to the result. The tools also shows the warning message if the program have to restructure and also the error message if the given program is contain any error.

10 COMPARISION OF EXISTING METRIC TOOLS

Metric tools	Parameters measured	Chart Generation
LOCC	loc, sloc, nop ,noc ,nom, noi	No
JHAWK5	cyclomatic complexity, comment lines, nop	Yes
CODE COUNTER	nos, comment lines, total	Yes
LOC METRICS	loc, bloc, cloc, c&sloc, sloc-l, mvg, cwords, sloc-p	Yes
SLOC METRICS	comments, blank lines, executable declarations, compiler directive, white space	yes xml, xsl, html, notepad

11 CONCLUSION

As there are wide applications of association rules in data mining, it is important to provide good performance. In view of this, we proposed a new algorithm which considers the significance of the item also but not only support of the item. The results through experiment shows that the computational cost of the link-based model is reasonable. At the expense of three or four additional database scans, we can acquire results different from those obtained by traditional counting-based models. Particularly for sparse data sets, some significant item sets that are not so frequent can be found in the link-based model. Through comparison, we found that our model and method address emphasis on high-quality transactions. In this paper, a brief discussion of a number of algorithms was presented along with a comparative study of a few significant ones based on their performance and memory usage.

REFERENCES

[1] http://www.atp.nist.gov/gems/97-06-0038gem.pdf

[2] An Analysis of Quality of Service Metrics and Frameworks in Grid Computing Environment, RussWakefield, Colorado State University, Ft. Collins, Colorado

[3] Russ Wakefield, Colorado State University, Ft.Collins,Colorado, “An Analysis of Quality of Service Metrics and Frameworks in a Grid Computing Environment”.

[4] http://www.locmetrics.com/

[5] http://microguru.com/sloc/

[6]http://www.downloadpipe.com/free-trial/JHawk-5-Personal-License-download1326333.html

[7] Amandeep Kaur, Satwinder Singh, Dr. K. S. Kahlon, “A Metric Framework for Analysis of Quality of Object Oriented Design”.

[8] V. Lakshmi Narasimhan, and Bayu Hendradjaya “Theoretical Considerations for Software Component Metrics”

[9] Boehm, Barry Dr.; Software Engineering Economics; Prentice Hall, Englewood Cliffs, NJ; 1981; 900 pages.

[10] Galorath, Daniel D. & Evans, Michael W.; Software Sizing, Estimation, and Risk Management: When Performance is Measured Performance Improves; Auerbach, Philadelphia, AP; ISBN 10-0849335930; 2006; 576 pages.

[11] Shyam R. Chidamber and Chris F. Kemerer, “A METRICS SUITE FOR OBJECT ORIENTED DESIGN”

[12] Dr. Linda H. Rosenberg “Applying and Interpreting Object Oriented Metrics”.

[13] Dr. K. Anbumani, K. P. Srinivasan, “A Set of Objected Oriented design Metrics”.

[14] *V. Lakshmi Narasimhan, P. T. Parthasarathy, and M. Das* “Evaluation of a Suite of Metrics for Component Based Software Engineering (CBSE)”.

[15] Radu Marinescu, “Using Object-Oriented Metrics for Automatic Design Flaws Detection in Large Scale Systems, *How to Use Object-Oriented Metrics in the Early Stages of a Redesign Operation”*

[16] V. Lakshmi Narasimhan, and Bayu Hendradjaya “Theoretical Considerations for Software Component Metrics”

[17] Magnus Andersson Patrik Vestergren “Object-Oriented Design Quality Metrics “

[18] Saida, Benlarbi, Khaled El Emam, Nishith Goel “Issues in Validating Object-Oriented Metrics for Early Risk Prediction”

(http://books.google.com)

[19] Tieng Wei Koh, Mohd Hasan Selamat, Abdul Azim Abdul Ghani, Rusli Abdullah "Review of Complexity Metrics for Object Oriented Software Products

[20] Marco Scotto, Alberto Sillitti, Giancarlo Succi, Tullio Vernazza "A relational approach to software metrics"