# Systematic Review of Object Oriented Metric Tools

Kayarvizhy N, PhD
Associate Professor
BMS College of Engineering
Bangalore

## ABSTRACT
Tools that extract metrics from object oriented code are widely used as part of static code analysis which acts as a feedback mechanism for the managers, developers and other stake holders to improve the software quality. The software industry and academic research have confirmed the necessity of such tools and the impact they have on ensuring quality software. There is a transition of tools from measuring traditional software metrics to object oriented metrics as the focus has shifted to object oriented design and development. This paper presents a systematic review of both commercial and open source object oriented metric tools, highlighting the features supported and extensibility. The results are useful to arrive at the most suitable tool depending on the requirements of the stake holder. The results also identify a potential for an object oriented tool that can address the need for a tool that can work effectively across many object oriented languages and also be flexible for extending it to different languages and metrics.

## Keywords
Object-oriented, metrics, tools, systematic review.

## 1. INTRODUCTION
It is a challenging task to develop high quality software within the bounds of cost and time. To achieve this, astute use of limited resources coupled with continuous quality management is required. Metrics measure the internal structure of the software and the process of software development. This helps in continuous monitoring of quality of software and the development process. Many metrics have been proposed to measure various aspects of software. Quality prediction models are constructed using metrics collected from historical project data [1]. These models can then be used for identifying potential risk factors during the development of future projects and releases. Since developers and stake holders depend on these models to make important decisions, the metrics collected should be accurate [2]. If decisions are based on inaccurate metric values, they would not be effective in mitigating the risks.

When the size of the software is small, collection of metrics can be a manual process. However commercial systems are large and complex. To apply the metrics effectively for such systems, the computation of metric values need to be automated with an appropriate tool. Automated tools overcome the problems in manual collection like user errors and helps provide accurate values. When the same tool is used across projects and releases, it improves the measurement process by providing a standard procedure to collect metrics. This guarantees uniformity in the way the metrics are computed and allows them to be compared between projects and releases.

When complex systems are implemented by object oriented design techniques, the traditional software metrics measuring object oriented code are not effective in capturing the OO quality attributes. So there is a need for object oriented metrics. To measure the object oriented metrics, the traditional tools are extended. Specific OO tools are also implemented when extending is not feasible. Several open source and commercial tools have been implemented to support automated metric computation for OO languages [3]. This poses a selection dilemma. The developers, stake holders and researchers require guidance in understanding the tools and a framework to choose the appropriate tool for their usage. In this study a systematic literature review of OO tools has been proposed.

A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question [4]. Compared to traditional reviews, a systematic review is rigorous in its approach, has a predefined search strategy, ensures completeness of the search and follows well defined methodology. It also requires considerably more effort compared to a traditional research. In a systematic review, the review protocol, search strategy, selection criteria, inclusion and exclusion criteria are well documented to ensure a transparent and unbiased approach. This also enables a replication of the study if deemed necessary.

## 2. LITERATURE REVIEW
Several OO metric tools have been proposed and developed. The tools differ in a variety of attributes and features. The developers, research community and other stake holders are interested in measuring quality attributes. Hence the have to select the appropriate tool that will help them measure the corresponding metric values. This task is not trivial and involves a lengthy process of filtering depending on various attributes. This study aims to help identify the major OO metric computation tools and compares them based on a set of quality attributes.

Literature studies have been undertaken to analyze the object oriented metric tools.

- Lincke et al [5] has performed an analysis of OO tools - Analyst4j, CCCC, ckjm, Dependency Finder, Eclipse plug-in 1.3.6, Eclipse plug-in 3.4, OOMeter, Semmle, Understand for Java and VizzAnalyzer. The tools that support CK metrics were shortlisted. The conclusion from the study was that metric tools outputs different values for the same metric. This is due to the difference in interpretation of the metric.
- Rutar et al [6] has compared five tools that do static analysis of Java source code. In his findings, Rutar concludes that the tools output data which are not trivial to understand and hence their usability is difficult.
- Lamas compared OO Metric tools - FindBugs and PMD [7] and concluded that the metrics supported by the tools complement each other.

- Bakar et al [8] compared ckjm, JStyle, RSM, JHawk for a number of metrics. They concluded that metric values obtained by the tools are different. The values obtained through manual calculation and metric tools were also different.
- Novak and Rakic [2] also concluded that the decision based on the metric values could be significantly different depending on the tool. They analysed Visual Studio, Borland, SourceMonitor, ReflectorAddin and NDepend to arrive at this conclusion
- Alikacem and Sahraoui [9] have introduce a meta-model to map the language constructs written in different programming languages. They also propose a metrics description language – PatOIS to allow representing metrics.
- Tomas et. al. [10] compares open source tools that support Java language for the metrics supported by them without providing an empirical validation.
- Martin & Bernhard [11] compare five tools on Windows platform for the input, output and automation attributes. They concluded that three out of five tools are not automated.

The studies focus on a limited number of OO metric tools and compare them for a subset of metric values like CK. The criteria used for selecting the list of tools is arbitrary and not defined. The reviews have not adhered to the thoroughness and rigor suggested in the systematic review process [4]. This has motivated to perform a review on the state of art of the OO metric tools.

The aim of the proposed comparative study is to provide insight about the object oriented metric tools and the features supported by the tools. In this study the focus is on a broader set of object oriented metric tools with a well-defined search criteria. The study also follows the rigorous approach of systematic review process. The purpose of this study is to serve as a guide for future research and provide a platform to select the appropriate tool required.

The following review questions are formulated to aid the systematic review process

- How many tools support more than one OO language? This question is particularly relevant if the user is interested in using the tool across OO languages
- Which tools are extensible for new user defined OO metrics?
- Given a set of quality attributes to measure, which tool would be an ideal choice? The review questions are targeted at aiding the population of research community, developers, testers, managers and other stake holders of object oriented software systems.

# 3. OO METRIC TOOLS REVIEW

## 3.1 Search Strategy

The search strategy was to target the following digital libraries. This collection was arrived at based on the relevant object oriented metrics and tools publications listed and cited in them.

- IEEExplore
- Science Direct
- ACM Digital library
- Google scholar
- Citeseer
- SpringerLink

Both journals and conferences have been include in the search. The search terms consisted of the following combinations of terms

- Object oriented Metric Tools
- Commercial OO Tools
- Open Source OO Tools
- Measuring OO Metrics
- OO Metrics and tools for measurement

With this search strategy, the following OO tools listed in Table 1 are obtained.

**Table 1. List of OO Metric Tools**

| Tool Name | Proposed by | Reference |
|---|---|---|
| SDMetrics | Commercial Tool | [12] |
| OOMeter | Alghamdi, Jarallah S., Raimi A. Rufai, and Sohel M. Khan, 2005 | [13] |
| JBOOMT | Xie, Tao, et al, 2000 | [14] |
| ES2 | Stojanovic, Marta, and Khaled El-Emam, 2001 | [15] |
| RSM | Commercial Tool | [16] |
| JHawk | Commercial Tool | [17] |
| Quality Metrics | Mythli, Swathi, 2010 | [18] |
| QMOOD++ | Bansiya, Jagdish, and Carl Davi, 1997 | [19] |
| Ckjm | Spinellis, 2005 | [20] |
| SAAT | Muskens, Johan, Michel Chaudron, and Rob Westgeest, 2002 | [21] |
| SARA | Sheik, K., et al, 2008 | [22] |
| MetricAnalyzer | Jyothi, Veerapaneni Esther, Kaitepalli Srikanth, and K. Nageswara Rao, 2012 | [23] |
| JMetric | Commercial Tool | [24] |
| JMT | Commercial Tool | [25] |
| JDepend | Clark, Mike, 2005 | [26] |
| JavaNCSS | Lee, Clemens, 2005 | [27] |
| Analyst4j | CodeSwat, 2007 | [28] |
| Dependency Finder | Jean Tessier, 2008 | [29] |
| Eclipse Metrics Plugin 1.3.6 | Frank Sauer | [30] |
| Eclipse Metrics Plugin 3.4 | Lance Walton, | [31] |
| Semmle | Commercial Tool | [32] |
| Understand | Commercial Tool | [33] |

## 3.2 Selection Criteria

The following selection criteria have been applied to arrive at the OO tools for the study. The primary criteria was that the tool be available to be run at least in binary form. The intention was to conduct the fresh and unbiased opinion of the tool and hence previous comments alone were not sufficient. The tool must support at least one OO metric. If the tool restricts itself to only non-OO metrics like Lines of Code (LOC), such tools were not considered for the final study. Commercial tools, which offered a limited time trial version were also considered. After applying the search and selection criteria, list of OO tools got reduced to those listed in Table 2.

**Table 2. Selected list of OO Metric Tools**

| Sl. No | Tool Name |
|---|---|
| 1 | SD Metrics |
| 2 | RSM |
| 3 | JHawk |
| 4 | QMOOD++ |
| 5 | ckjm |
| 6 | JMetric |
| 7 | JMT |
| 8 | JDepend |
| 9 | Eclipse Metrics Plugin 1.3.6 |
| 10 | Eclipse Metrics Plugin 3.4 |

1. SDMetrics: The initial version of the tool was released in 2002 as Version 1.0. The current version 2.31 was released on July 2013. The tool specifically designed for use with UML, the design tool for OO development. The UML models are analysed by the tool. Design rules can be specified for completeness, consistency and correctness including design style issues. Other than an interactive GUI, it also has a variety of options for data export. It is a commercial tool.
2. Resource Standard Metrics (RSM): This is a commercial tool available for measuring OO metrics in C++, Java and C# languages. The current version of the tool is 7.75 and is extensively supported and updated regularly. It can be integrated with Visual Studio, .NET, JBuilder, Eclipse and other popular IDEs.
3. JHawk: JHawk is a commercial code quality tool for Java and has been in use for more than a decade. The tool is updated often and the latest release Verion 6.0 was released in April 2015
4. QMOOD++: The tools is freely available in both executable and source code form and supports 30+ OO Metrics. QMOOD++ is a comprehensive, multiuser, multithreaded, integrated Windows tool.
5. ckjm: ckjm is a free tool developed by Diomidis Spinellis to measure CK metric suite on Java projects. The tool works only on compiled class files and hence gives account of all internal methods also during the measurement like default constructors etc. It is a text based application.
6. JMetric: JMetric was developed as part of a research initiative by School of Information Technology, Swinburne University of Technology. It supports only the Java language. The metric values are provided in tables and charts. It also supports inner and abstract classes.
7. JMT: JMT tool also supports OO metrics only from the Java language.
8. JDepend: JDepend is tool for measuring design quality metrics from Java files. The tool supports both graphical and textual user interface.
9. Eclipse Metrics Plugin 1.3.6: This is a metrics plugin for Eclipse IDE. The plugin is also provided integrated as an EasyEclipse package. The plugin computes the metrics and displays it in the integrated view.
10. Eclipse Metrics Plugin 3.4: The eclipse plugin 3.4 developed by Lance Walton is also integrated with Eclipse and is available for all Java projects developed using the IDE

## 3.3 Selection Criteria

The following attributes have been considered for the review to compare the tools

- Number of OO Metrics supported
- Languages supported
- Source code availability
- Free/Commercial
- Input options
- Output options
- Semi-Automated or Fully Automated
- Validations.

## 3.4 Methodology

Recent and updated information on each tool was obtained by visiting the tool's homepage. The tool was downloaded from the tool's website or from sourceforge website. The tools were then run on the same dataset. The dataset was obtained from an implementation of an ATM machine project in C++, C# and Java. Some tools like ckjm for example require compiled Java class files. This was obtained by compiling the Java source files using JDK 1.8 obtained from oracle website [37].

# 4. REVIEW RESULTS

## 4.1 Supported Metrics and Languages

The metrics tools support a range of OO metrics. They also support different OO languages. The results in Table 3 show that Java is the language that is supported by majority of the tools. The tool SDMetrics is language agnostic and supports UML designs in XMI format. It can also be noted that RSM supports all the three popular OO languages. The tools also support one or more metrics from the CK Metric suite. QMOOD++ supports 30 OO metrics. The number of methods

(NOM) and number of attributes (NOA) are the widely supported metrics.

**Table 3. Supported Metrics and Languages**

| Tool Name | Language | Metrics |
|---|---|---|
| SD Metrics | UML | NOC, DIT, Ca, Ce, NOA, NOM, CLD |
| RSM | C++, Java, C# | DIT, NOA, NOM |
| JHawk | Java | LCOM, CBO, RFC, NOM, NOA |
| QMOOD++ | C++ | 30+ OO Metrics [19] |
| ckjm | Java | CK Metrics, CA, NPM |
| JMetric | Java | LCOM, NOA, NOM |
| JMT | Java | AIF, MIF, NOA, NOM, CF, CK Metrics, NOM, NOA, WAC |
| JDepend | Java | Ca, Ce, A, I |
| Eclipse Metrics Plugin 1.3.6 | Java | WMC, NOM, NOA, LCOM, Ce, Ca, NOC, SI, DIT |
| Eclipse Metrics Plugin 3.4 | Java | LCOM, WMC |

## 4.2 Tool Source Code, Automation and Availability

The OO metric tools considered are compared for the availability of their source code, if they are fully automated and available with free license. Table 4 summarizes the results of this comparison.

Three out of the ten tools considered are commercial tools for which the source code is not available. JMT in spite of being available under free license does not provide access to source code. ckjm and JDepend require the source code to be compiled and the binary (*.class) files to be provided and hence are not fully automated.

**Table 4. Source Code, Automation and Availability**

| Tool Name | Code | Automated | Free |
|---|---|---|---|
| SD Metrics | No | Yes | No |
| RSM | No | Yes | No |
| JHawk | No | Yes | No |
| QMOOD++ | Yes | Yes | Yes |
| ckjm | Yes | No | Yes |
| JMetric | Yes | Yes | Yes |
| JMT | No | Yes | Yes |
| JDepend | Yes | No | Yes |
| Eclipse Metrics Plugin 1.3.6 | Yes | Yes | Yes |
| Eclipse Metrics Plugin 3.4 | Yes | Yes | Yes |

## 4.3 Input, Output and Validations

The input, output and validation details of the various tools are listed in Table 5. Most of the tools take the source code in .java and .cpp as input. ckjm and JDepend take the compiled .class files as input while SDMetrics expects XMI file as input. The output from the tools can take many forms. It can be displayed within the tool as command line text or within the GUI. Output is also available in HTML, CSV, XML formats. Majority of the tools have been validated by the research community as part of their work.

**Table 5. Input Format, Output Format and Validations**

| Tool Name | Input | Output | Validations |
|---|---|---|---|
| SD Metrics | XMI | GUI, HTML, XML | NA |
| RSM | *.cpp/cs/Java | XML, HTML, CSV, TXT | NA |
| JHawk | *.java | GUI, CSV | 25+ |
| QMOOD++ | *.cpp | GUI | None |
| ckjm | *.class | Console, XML | 25+ |
| JMetric | *.java | XML | 25+ |
| JMT | *.java | GUI | 5 |
| JDepend | *.class | GUI | 25+ |
| Eclipse Metrics Plugin 1.3.6 | *.java | Eclipse View | 25+ |
| Eclipse Metrics Plugin 3.4 | *.java | Eclipse View | 25+ |

# 5. CONCLUSION

The study compares and analyses the various object oriented metric tools. The results are tabulated under various attributes that would be of interest to developers and researchers using the tools. Only few commercial tools support the needs of current OO measurement requirements and are upgraded continuously. Open source tools, in most cases, are specific to an object oriented language, lack in extensibility and have many constraints. The results also identify that further work is needed in the field of open source OO metric tools to arrive at tools that satisfy the requirements of a flexible and extensible tools that works across many object oriented languages and object oriented metrics. The tool should also support addition of new languages and metrics. Future studies can include other criteria like performance and other quality attributes.

## 6. REFERENCES

[1] V. Yadav, R. Singh, Validating Object Oriented Design Quality using Software Metrics, Proceedings of the International Conference on Advances in Electronics, Electrical and Computer Science Engineering, vol. 2, no. 3, pp. 112-117, 2012

[2] J. Novak, G. Rakić, Comparison of software metrics tools for: net, Proc. of 13th International Multiconference Information Society-IS, Vol A. 2010.

[3] S. George, S. Avinash, J. T. Abraham, Object Oriented Design Metrics, Proceedings of the National Conference on Software Engineering, pp. 283-287, 2014

[4] SEG (Software Enginnering Group), Guidelines for Performing Systematic Literature Reviews in Software Engineering, Version 2.3, 2007

[5] R. Lincke, J. Lundberg, W. Löwe, Comparing software metrics tools, Proceedings of the 2008 international symposium on Software testing and analysis. ACM, 2008.

[6] N. Rutar, C. B. Almazan, J. S. Foster, A comparison of bug finding tools for Java, Software Reliability Engineering, ISSRE, 15th International Symposium on. IEEE, 2004.

[7] I. Lamas Codesido, Comparación de analizadores estáticos para código java, 2011.

[8] N. S. Bakar, C. V. Boughton, Validation of measurement tools to extract metrics from open source projects, Open Systems (ICOS), IEEE Conference on. IEEE, 2012.

[9] E. H. Alikacem, H. Sahraoui, Generic metric extraction framework, Proceedings of the 16th International Workshop on Software Measurement and Metrik Kongress (IWSM/MetriKon). 2006.

[10] P. Tomas, M. J. Escalona, M. Mejias, Open source tools for measuring the Internal Quality of Java software products. A survey, Computer Standards & Interfaces 36.1 (2013): 244-255.

[11] M. Auer, B. Graser, S. Biffl, A survey on the fitness of commercial software metric tools for service in heterogeneous environments: Common pitfalls, Software Metrics Symposium, Proceedings of Ninth International IEEE, 2003.

[12] Vukelich, Sdmetrics tool - A tool for measuring object-oriented design metrics from UML models, http://www.sdmetrics.com visited in January 2010.

[13] J. S. Alghamdi, R. A. Rufai, S. M. Khan, OOMeter: A software quality assurance tool, IEEE, 2005.

[14] T. Xie, W. Yuan, H. Mei, F. Yang, JBOOMT: Jade Bird Object-Oriented Metrics Tool, Submitted to Chinese Journal of Electronics (English Version), 2000.

[15] M. Stojanovic, K. El-Emam. ES2: A Tool for Collecting Object-oriented Design Metrics for the C++ and Java Source Code, 2001.

[16] ResourceStandardMetric(http://msquaredtechnologies.com/m2rsm/index.html)

[17] JHawk(http://www.virtualmachinery.com/jhawkprod.htm)

[18] M. Thirugnanam, J. N. Swathi, Quality Metrics Tool for Object Oriented Programming, International Journal of Computer Theory and Engineering, Vol 2, no. 5, pp. 1793-8201, 2010.

[19] J. Bansiya, C. Davis, Using QMOOD++ for object-oriented metrics, Dr. Dobb's Journal , 1997.

[20] D. D. Spinellis, ckjm Chidamber and Kemerer metrics Software, Technical report, Athens University of Economics and Business, 2005.

[21] J. Muskens, M. Chaudron, R. Westgeest, Software architecture analysis tool, Proceedings of the 3d Progress Workshop on Embedded System. 2002.

[22] K. Sheik, W. Abdelmoez, K. Goseva-Popstojanova, H. Ammar, Software Architecture Risk Assessment (SARA) Tool, International Journal of Software Engineering Vol. 1, no. 2, 2008.

[23] V. E. Jyothi, S. Kaitepalli, K. N. Rao, Effective Implementation of Agile Practices-Object Oriented Metrics tool to Improve Software Quality, International Journal of Software Engineering and Applications , Vol. 3, no. 4, 2012.

[24] JMetric(http://www.it.swin.edu.au/projects/jmetric/products/jmetric/default.htm

[25] JMT(http://wwwivs.cs.unimagdeburg.de/sweng/agruppe/forschung/tools/)

[26] M. Clark, JDepend, http://www. clarkware. com/software/JDepend. html. l ast accessed in March (2005).

[27] C. Lee, JavaNCSS-a source measurement suite for Java, 2005.

[28] Analyst4j (http://www.codeswat.com)

[29] Dependency Finder (depfind.sourceforge.net)

[30] Eclipse Metrics Plugin 1.3.6 (http://www.easyeclipse.org/site/plugins/metrics.html)

[31] Eclipse Metrics Plugin 3.4 (http://eclipse-metrics.sourceforge.net/ )

[32] Semmle (https://semmle.com/ )

[33] Understand for Java (https://scitools.com/ )

[34] W. Lowe, M. Ericsson, J. Lundberg, T. Panas, N. Pettersson, Vizzanalyzer - a software comprehension framework, Third Conference on Software Engineering Research and Practise in Sweden, Lund University, Sweden. 2003.

[35] Parasoft (https://www.parasoft.com/product/static-analysis-cc/ )

[36] EssentialMetrics(http://www.powersoftware.com/download/ )

[37] JDK 1.8(http://www.oracle.com/javase/downloads/jdk8-downloads-2133151.html)

# 7. APPENDIX

The nomenclature and abbreviations listed in Table 6 are used in this study.

**Table 6. Nomenclatures and Abbreviations**

| | |
|---|---|
| A | Abstractness |
| AIF | Attribute Inheritance Factor |
| Ca | Afferent Couplings |
| CBO | Coupling Between Objects |
| Ce | Efferrent Couplings |
| CF | Coupling Factor |
| CK | Chidamber and Kemerer |
| CLD | Class to Leaf Depth |
| CSV | Comma Separated Values |
| DIT | Depth of Inheritance Tree |
| HTML | Hyper Text Markup Language |
| I | Instability |
| LCOM | Lack of Cohesion among Methods |
| LOC | Lines of Code |
| MIF | Method Inheritance Factor |
| NMI | Number of Methods Inherited |
| NOA | Number of Attributes |
| NOC | Number of Children |
| NOM | Number of Methods |
| NPM | Number of Public Methods |
| OO | Object Oriented |
| RFC | Response set For a Class |
| SI | Stability Index |
| UML | Unified Modelling Language |
| WAC | Weighted Attributes per Class |
| XML | Extensible Markup Language |
| XMI | XML Metadata Interchage |