COMPARATIVE STUDY OF STATIC METRICS OF PROCEDURAL AND OBJECT ORIENTED PROGRAMMING LANGUAGES

Manik Sharma
Assistant Professor
Sewa Devi S.D. College Tarn Taran

Ankur Bhardwaj
Assistant Professor
Sewa Devi S.D. College Tarn Taran

Lakhbir Singh
Centre Head
SGRD, PTU Centre
Amritsar

Navpreet Singh
IT Head
MINT, Ajmer
Rajhasthan

Chandni Sharma
Department of Computer Science & Engineering
Adesh Institute of Engineering & Technologuy

ABSTRACT

The goal of this study is to compute and analyze an assortment of static software metrics for different programming methods or techniques. Software engineering is branch of Computer Science that deals with an effective development and analysis of software product. Software engineering provides the concept of metrics with the help of which the complete investigation of code can be done in static or in dynamic way. The static metrics helps in measuring the effectiveness of code without executing the actual program. In this study the focus is given on analyzing the different programming methods like function, constructor, overloading and virtual functions.

General Terms

Software Metric, function, overloading, constructor.

Keywords

Software, Static Metric, programming etc.

1. INTRODUCTION

One of the major aspects of software engineering is to develop and manage software for commercial use. Software metric is a field of software engineering that is associated with diverse measurements of computer software and its developments. Software metrics [1] [2] [3] is one of the important tools for analyzing the software product in an effective way. In other words software metrics are measures that enable software developers and software analyst to gain insight into the efficiency of the software process and projects that are conducted using the process as framework. Software metrics measures different aspects of software complexity and therefore play an important role in analyzing and improving software quality [3]. With the help of software metric one is able to understand the software product in an effective way. We apply some software logic or mathematical technique to software process or product to supply or improve engineering and management information. One can create relationship between various measures by using the concept of software metrics. Normally metrics are classified [4] as metrics analytical model, metrics for software design, metrics for coding, software testing metrics, software quality and software reliability metrics.

Beside this classification Somerville [5] describes metrics into two types called static and dynamic metrics. Here static metrics help in analyzing the code before its execution whereas dynamic metric help in analyzing the code during execution. In this study the accentuate is given on static metrics to understand the performance and productivity of procedural and object oriented programming languages.

Software metric [6] plays a major role in civilizing the quality of software, planning its budget, its cost estimation etc.

2. PROBLEM DEFINTION

The objective of this study is to compute and analyze the static metrics for different approach of programming like functional approach, recursive approach, constructor approach, overloaded approach and virtual approach etc. The study starts with introduction and fundamentals of metrics. In this study main focus is given on coding metrics for their analysis in procedural and object oriented programming languages. The various objectives of this study are:

- To gain basic knowledge about metrics and their types.
- Understanding static metrics
- Measuring the attributes of static metrics for different programming methods.
- Comparing the various static metrics for different programming methods.

3. ANALYSIS

Software metrics plays a great role in coordinating [9] and managing the software project. With the help of software metrics one is able to compute and analyze various attributes of a software project. The favourable area of application of software metrics is the estimation of cost and size. There are different types of metrics like size metrics, quality metrics, satiability metrics, object oriented metrics etc. The credit of introducing the concept of software metrics goes to Wolver ton who performs a research on production ratio of the programmer by using the concept of LOC i.e. line of code. Software metrics explore the attributes of software to measure the different characteristics of software.. Metrics helps in to measure the various attributes like cost of software development, its complexity, number of operands, operators and statement, hiding factor, coupling factor etc. Predictive metric are normally associated with software product. According to Somerville the metric can be classified into two categories i.e. control metric and predictive metric. Predictive software metric [7] plays a major role in determining both static as well as dynamic characteristics of the software.

` First static metric [2] (LOC/KLOC) was used to measure the productivity of a program. The most commonly used complexity metric before 1990 was cyclomatic [8] complexity that was measured by McCabe. He uses the flow graph and some mathematical equations to compute software complexity. This metric was used in code development risk analysis [1], change risk analysis in maintenance and in test planning. In 1976 McCabe [8] defined the cyclomatic complexity number metric. The metric measures the number of independent paths through a software module. Although cyclomatic complexity is widely used, critique on it exists claimed that it's based on poor theoretical foundations and an inadequate model of software development. The cyclometic complexity has been selected to be a part of the benchmarks.

Halstead has brought the revolution in the field of metric by collaborating information science and psychology. By using the concept of Halstead metrics an analyst is able to compute the size, complexity, volume, length, difficulty of a project. The basic attributes of Halstead metrics are n1,n2, N1 & N2. In this study we will try to analyze the various Halstead metrics for different programming approach as discussed above. A same segment of code will be implemented in different approached as given above and analyzed. Functional approach is one of the favorable approaches of structured programming that implement the concept of reusability in an effective way. Constructor approach is associated with object oriented programming In this case sample code is implemented using a constructor one of the important type of a member functional that is executed whenever an instance of a class is executed. The following table will give the analysis of various metrics of programs developed using different approaches as given below:

Characteristics	n1	n2	N1	N2	V(G)	V'(G)	D	E	V	LOC
Functional Approach	7	9	10	23	2	4.25	6	430.20	71.70	18
Constructor	7	9	10	25	2	8.50	5.25	459.74	87.57	27
Overloading	8	10	14	33	2	8.75	6.56	640.97	97.67	32
Virtual	7	12	10	30	2	8.50	4.67	419.34	89.86	33
Functional Approach	7	9	10	23	2	4.25	6	430.20	71.70	18

n1,n2, N1, N2 Halstead [13] has proposed different metrics for measuring the size of a program he uses different variables n1,n2, N1,N2 the number of distinct operators, number of distinct operands, number of operators and number of operands respectively.

V (G) Cyclometic Complexity: it is one of the important measures of programming code. Normally the introduction of conditional and looping statement adds some complexity in the program. The concept of cyclometic complexity is given by McCabe. Mathematically it is calculated as V(G)=e-n+p

V' G): Extended Cyclometic Complexity

D: Halstead program difficulty

E: Halstead Program Effort metrics helps in determining the programming effort required to develop project.

V Halstead Program Volume [16] is one the important metrics that instruct the analyst to consider the programming language while calculating the length of the program. In technical terms it is minimum number of bits that are used to encode the program.

LOC: It is one the basic static metric that is used to measure the size of code segment. It helps in measuring the cost of project in an effective way.

The following chart shows the relationship between different programming approaches with Halstead basic metrics. From the following figure1(a) it is clear overloading approach has higher number of n1,n2,N1 and N2.

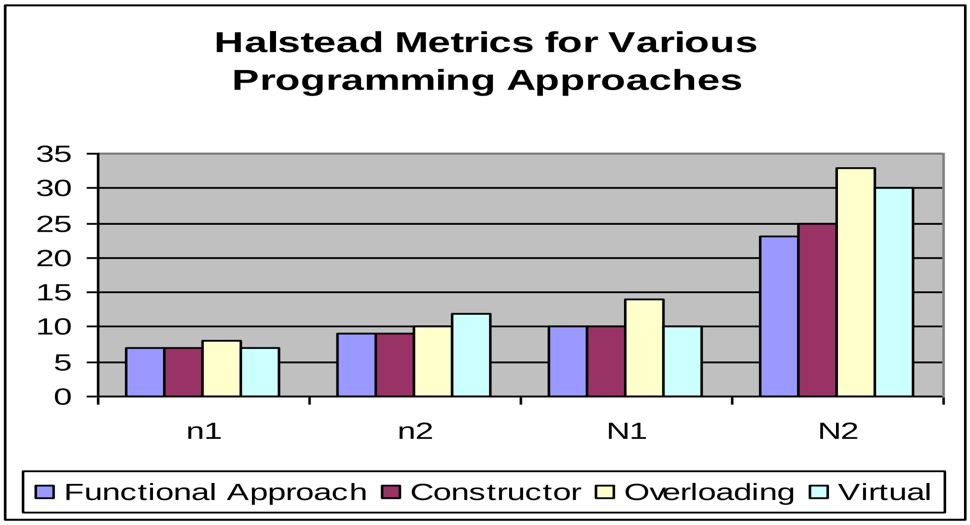

Figure 1(a): Halstead Metrics versus different programming approaches

The following figure 1(b) shows how different programming approaches react to cyclometic complexity V(G) and extended cyclometic complexity V'(G).

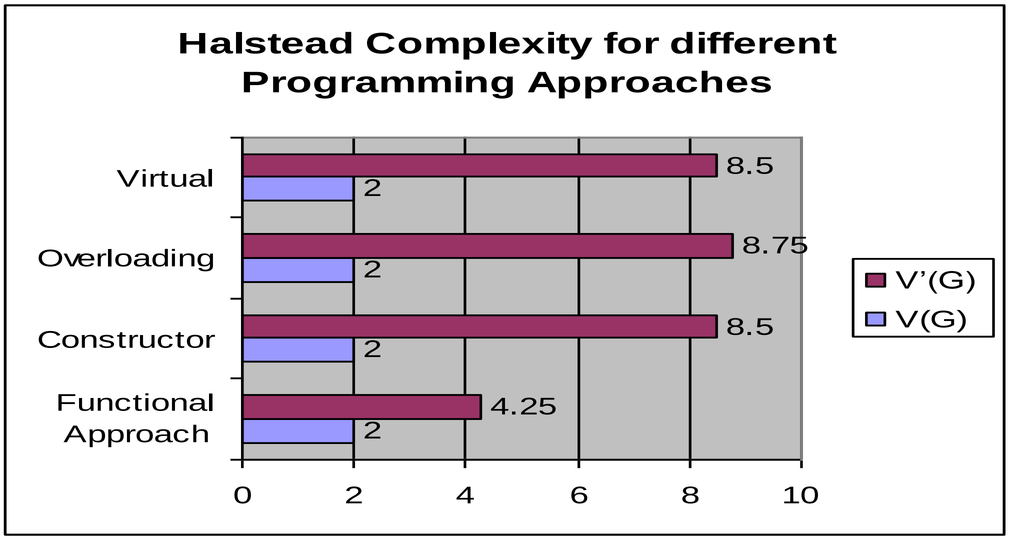

Figure 1(b): Halstead Complexity Metrics versus different programming approaches

From the above figure it is clear that overloading approach has highest value of extended cyclometic complexity whereas functional approach has least value of extended cyclometic complexity. And in regard to cyclometic complexity it is observed that all the four different programming approaches show similar behaviour. The above graph shows that the logarithmic formula showing variation between different programming methods is given as below:

$$y = 0.825\ln(x) + 6.059 \qquad \text{-Equation I}$$

The following figure 1(c) show how Halstead volume is associated with different programming methods like functional, constructor, overloading and virtual functions. From the following figure it is clear that simple functional approach has least Hallstead volume where as overloading programming method has greatest amount of Halstead volume. The variation of Halstead volume between constructor method and virtual functional approach is very small. From the following graph it is also clear that the logarithmic formula showing the variation between different programming methods for Halstead volume is given as below:

$$y = -0.18\ln(x) + 5.869 \qquad \text{-Equation II}$$

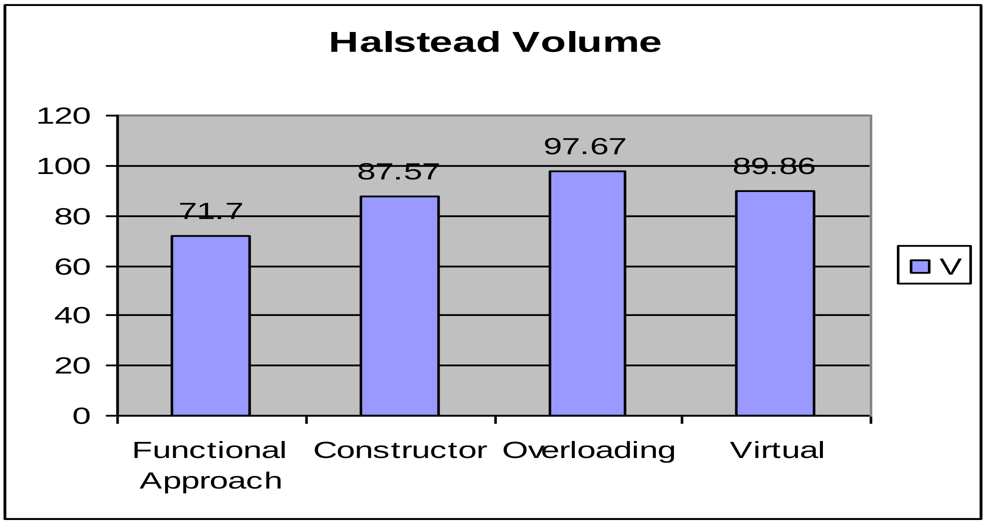

Figure 1(c): Halstead Volume versus different programming approaches

From the following figure 1(d) it is crystal clear that virtual functional has largest value of lines of code where as functional approach take minimum lines to code the same segment when compared with different programming methods like constructorm overloading and virtual.

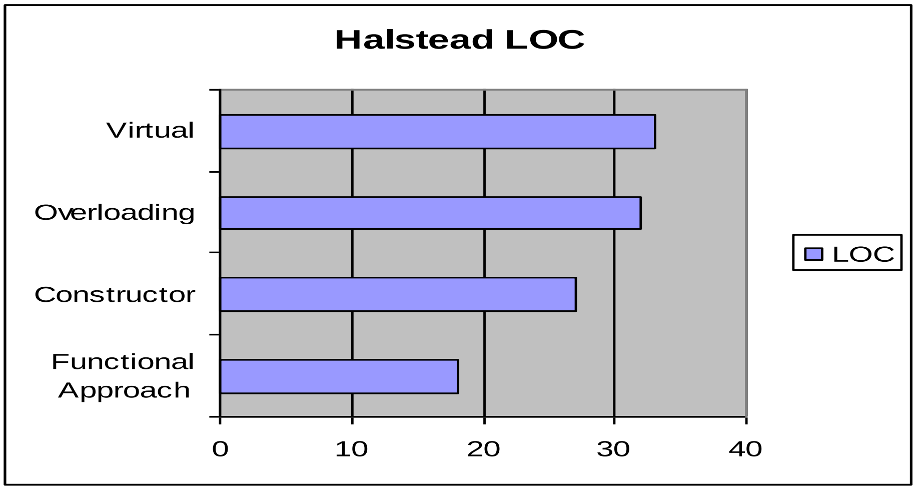

Figure 1(d): Halstead Volume versus different programming approaches

The following diagram show the combined logarithmic variation of different metrics with different programming methods discussed above.

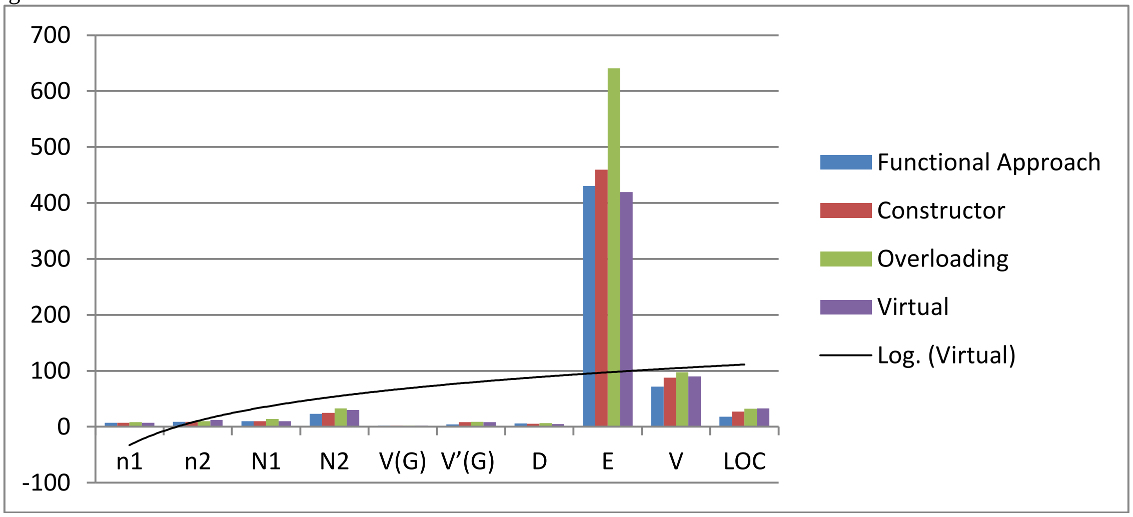

Figure 1(e): Various software metrics versus programming methods

The mathematical equation that shows the combined logarithmic variation of different software metric with different programming methods is as shown below:

$$\mathbf{y = 62.62ln(x) - 32.95} \qquad \textbf{-Equation III}$$

4. CONCLUSIONS

Every branch of science is associated with metrics and measurements. The above study shows the analysis of various static software metrics by taking different programming methods. From the study it become obvious to choose simple functional approach for coding when Halstead metrics are considered, because functional approach give better result in when the analysis of various metrics like n1,n2, N1 and N2 is performed. Further functional approach also takes least line of code as compare to other programming methods under study. In regard to complexity functional approach again gains priority.

5. ACKNOWLEDGEMENT

Authors are highly grateful to Dr. Gurvinder Singh, Associate Professor & Head, DCSE, GNDU, Amritsar and Dr. Gurdev Singh, Professor & Head, DCSE, Adesh Institute of Engineering & Technology, Faridkot for their precious guidance from time to time.

6. REFERENCES

[1] Manik Sharma, Dr. Gurdev Singh, "Analysis of Static and Dynamic Metrics for Productivity and Time Complexity" IJCA, Volume 30– No.1, September 2011.

[2] H F Li, W K Cheung "An Empirical Study of Software Metrics" Software Engineering IEEE Transactions on (1987) Volume: SE-13, Issue: 6, Pages: 697-708

[3] N E Fenton "Software Metrics" Conference Proceedings of on the future of Software engineering ICSE 00(2000) Volume: 8, Issue: 2, Publisher: ACM Press

[4] Roger S. Pressman "Software Engineering-A Practitioner's Approach" 6th Edition, McGraw Hill International Edition pp 466-472.

[5] Somerville "Software Engineering", 6th Edition, Editor: Addison Wesley.

[6] Kuljit Kaur Chahal, Hardeep Singh "Metrics to study symptoms of bad software designs" ACM SIGSOFT Software Engineering Notes (2009) Volume: 34, Issue: 1, Pages: 1

[7] Manik Sharma , Gurdev Singh "Static and Dynamic metrics- A Comparative Analysis", Emerging Trends in Computing and Information Technology 2011.

[8] Thomas J McCabe, "A Complexity Measure", IEEE Transaction on Software Engineering, Vol. SE-2 No. 4 [308-320]

[9] Satwinder Singh, K.S. Kahlon, "Static Analysis to Model & Measure OO Paradigms", SAC, ACM.