Volume 01, Issue 01 (Jan-Feb 2021), PP: 21-23
www.ijsreat.com

Expending Aspect Oriented Methodology to Extent of Dynamic Complexity

H. Ramamohan[1], S. Raminaidu[2]

[1]*Department of CSE, Yeshwant Mahavidyalaya, NANDE-, M.S., India.*

[2]*Department of CSE, Poona College, Pune.India.*

Abstract: *Multifaceted design expects a basic part to find nature of programming. There are lots of static or collect time estimations open recorded as a hard copy to find multifaceted design of programming, but not a lot of estimations are available to find runtime unpredictability of programming. In this paper, advantage of existing multifaceted design still up in the air by using CodeMR contraption which is an Eclipse module. Further, we introduced one more estimation named DWMC, dynamic weighted method complexity to check multifaceted nature of a class at run time. One more device is purposed named DynaDwmc completed in AspectJusing point of view organized programming to check the value of purposed estimation. Starting then and into the foreseeable future, assessment of static WMC and runtime DWMC metric characteristics is done. An exploratory survey is done on 15 java classes and it is contemplated that purposed metric DWMC expects a colossal part to find multifaceted design, therefore nature of softwaresystem.*

Watchwords: *Dynamic, multifaceted nature, quality, perspective*

I. Introduction

Estimations at some irregular time during the endeavor execution show the idea of the thing being developed. Metric assessment for near endeavors execution gives the limit levels of cycles being used all through some unclear time span. Metric are not difficult to portrayed, easy to understand,robust to use and prepared to improve.Chidamber and Kemerer (CK) have introduced six estimations in his suite[3]. Weighted technique per class(WMC) is one of the estimation in his suite. It is portrayed as the weighted complete multifaceted nature of all the technique described with in a class. If all of the procedures in class have unpredictability as fortitude, WMC will be comparable to number of methodologies. i.e.n.

$$WMC = \sum n$$

cI

Where c1,c2… .cnbe the multifaceted design of the procedures m1… .mn portrayed inside the class.In this paper,we separate the value of WMC metric by using CodeMR[9] tool.CodeMR is basically an Eclipse plugin[10] to find the idea of programming to the extent that 3 head quality measures for instance complexity, association and coupling for java,C++ and scalalanguages.It moreover gives Graph and Dependency viewpoints on not entirely settled by the gadget. For dynamic code analysis[4] we purposed one more contraption did in AspectJ[1], executed in java using perspective arranged programming.There are a couple of advantages of point of view arranged approach[8] over various ways of managing follow the events at runtime.

1. The course of building a following or profiling frame work using point of view arranged approach is decently more straight forward than any other approach.
2. This approach doesn't deliver a gigantic total ofdata.
3. This approach produces results which are significant among all program executionenvironment.
4. This approach is reasonably in expensive and significantly more rational innature.

II. Proposed Metric and Tool

DWMC (Dynamic Weighted procedure multifaceted nature): It is portrayed as the times a method is executed at runtime with in a class. If everything the procedure for a class is executed at runtime, static strategy unpredictability and runtime weighted method complexity are same for that class.

$$DW=\sum n \quad \mu I$$

Where μiis the times procedure iis executed at runtime and n is the hard and fast number of systems inside a class.

DynaDwmc is a gadget to figure the purposed metric.Tool is executed in AspectJ. AspectJ is transcendently an execution of Aspect Oriented programming in java.Figure 1. shows the coding of perspective in AspectJ which is to be run with any class to find the value of DWMC metric.

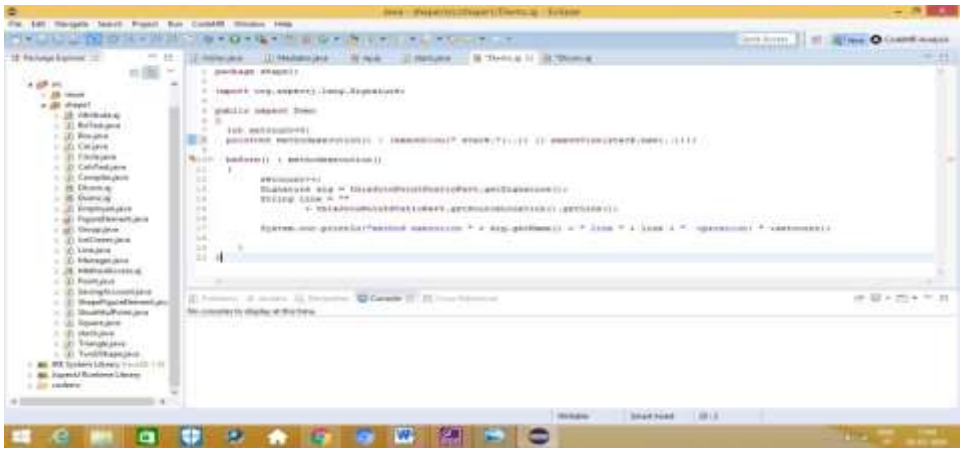

Figure 1.Working ofDynaDwmc tool in AspectJ

III. Case Study

Value of the WMC metric for this class is 5 as the number of methods in the class is 5 but CodeMR tool calculate its value 7 that means complexity of two methods is not considered as unity.If complexity of all the methods is considered as unity than it would be 5. To calculate the value of DWMC,DynaDwmc tool is used. Dwmc.aj aspect is created in AspectJ and run with stack.java class without interrupting the functioning of Stack class and the value of DWMC metric is calculated from runtime log as shown in Figure 3. Calculated value of DWMC metric is 16. So there is big difference between static and runtime weighted method complexity.

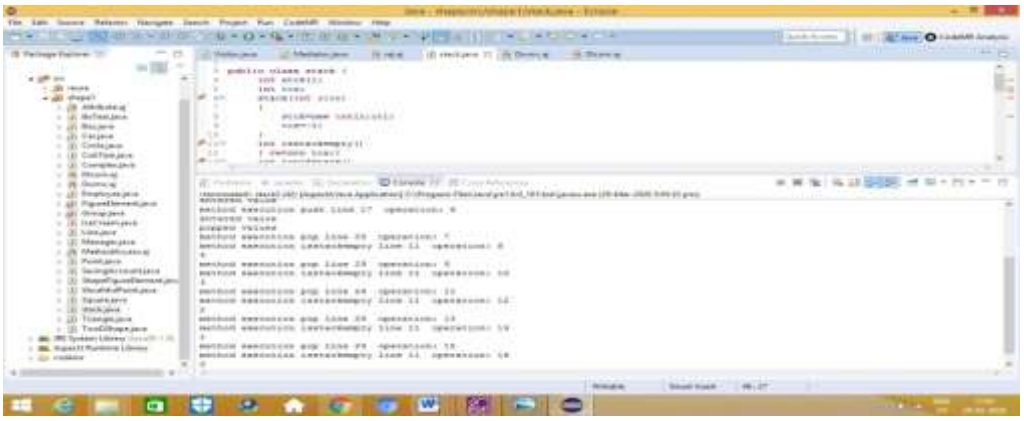

Figure 2.Event Tracing using DynaDwmc tool

IV. Experimental Study

A preliminary survey is driven for the purposed estimations using 15 java classes.Classes are taken from web[12].CodeMR gadget is used to resolve worth of WMC metric for these classes as shown in Figure.4 and DynaDwmc contraption is used to process values for purposed estimation.

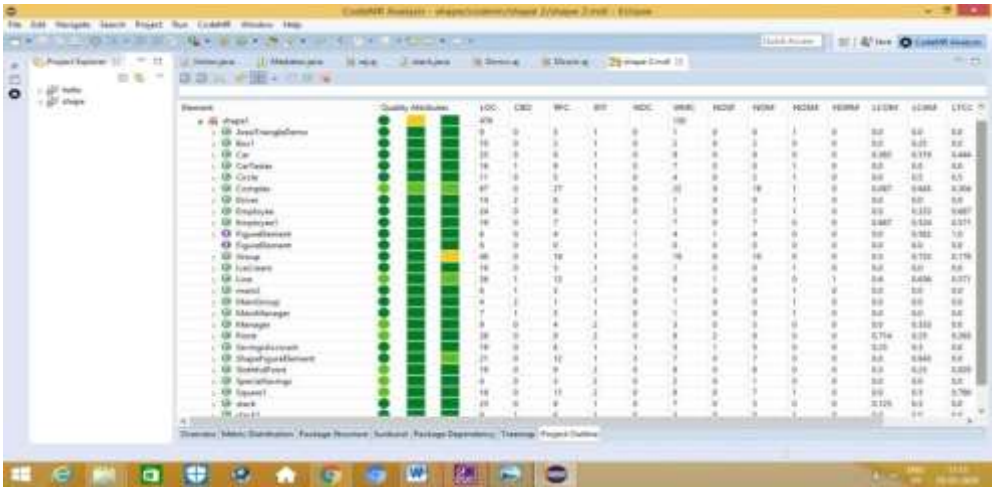

Figure. 3. WMC metric values of java classes using CodeMR tool

The potential gains of WMC and purposed estimation of 15 java classes is shown in Table 1. From Table 1.we can communicate that there is colossal qualification between organize time and runtime weighted method complexity.Behaviour of every single java class is different at collect time and run time.. A couple of classes like 1,8 and 13 shows same approach to acting at both assemble time and run time that suggests all of the procedures for these classes execute at runtime once. Classes like 2,3,6,7 and 11 shows goliath qualification among gather and run time regards that depict strategy for these classes execute essentially a couple of times at runtime and complexity of these classes increases at runtime. Classes like 10,14 and 15 have less worth at runtime than collect time that depicts techniques for these classes not execute at runtime consequently complexity reduces at runtime and we can say class isn't totally used atruntime.

Table 1.WMC and Purposed Metric DWMC values of 15 java classes

Sr. No.	Class Name	WMC	DWMC
1	Box	2	2
2	Stack	7	16
3	Car	9	24
4	Circle	4	3
5	Book	1	2
6	Complex	19	48
7	Employee	5	16
8	Ice Cream	1	1
9	Employee	7	9
10	Manager	3	2
11	Saving Account	5	10
12	Special Saving	1	2
13	Area Triangle	1	1
14	2D Shape	6	2
15	Square	8	4

V. Conclusion

CodeMR instrument is used to resolve potential gains of WMC (Weighted Method Complexity) metric from CK(Chidamber and Kemerer) metric suite at integrate time. The purposed instrument DynaDwmc is used to process potential gains of purposed metric DWMC,Dynamic Weighted Method Complexity at runtime.An preliminary Study is done on 15 java classes and it is seen that there is huge qualification between potential gains of consolidate time class multifaceted design and run time class unpredictability. Thusly, purposed estimation has its own world. Thusly, to check the idea of programming purposed estimation expects a colossal part by assessing the runtime unpredictability of programming.

References

[1] S.R. Chidamber and C. F. Kemerer, "A Metrics Suite for Object-OrientedDesign", IEEE Transactions on Software Engineering, vol. 20, no. 6, pp. 476-493,1994.

[4]S. Yacoub, H. Ammar, and T. Robinson, "Dynamic Metrics for Object-Oriented designs", Proc. 5th International Software Metrics Symposium, Boca Raton, Florida, USA, pp.50–61,1999.

[5] J. Tian, and M.V. Zelkowitz, "A Formal Program Complexity Model and its Application", Journal of Systems and Software, vol. 17, no. 3, pp.253-266,1992.

[6] D.P. Tegarden, S.D. Sheetz, D.E. Monarchi, "A Software Complexity Model of Object- Oriented Systems", Decision Support Systems: the International Journal, vol. 13, pp.241-262,1995.

[7] J.C. Munson and T.M. Khoshgoftaar, "Measuring Dynamic ProgramComplexity",IEEE Software, vol. 9, no. 6, pp. 48-55,1992.