IJCTA, 9(27), 2016, pp. 29-34
ISSN: 0974-5572

Statistical Analysis of Pointcut Complexity Metric Using Cognitive Approach

G. Arockia Sahaya Sheela* and A. Aloysius**

ABSTRACT

Metrics are an important technique in quantifying desirable software and software development characteristics of aspect oriented software development (AOSD). Theoretical and empirical validation of metrics and of their relation to software attributes is a cumbersome and long process. It is of paramount importance that we validate the utility of metrics we use in order to enable others to use them, too.Aspect oriented programming is an efficient approach to improve the software program at the time of software maintenance for modularizing crosscutting concerns. However, in order to take the advantages of AOP, there is a need for supporting the systematic refactoring of crosscutting concerns to aspects.This paper presents a new cognitive complexity metric namely Cognitive Weighted Pointcut per Aspect (CWPA) in Aspect Oriented System. This paper addresses the CWPA metric to measure the different type of pointcut designator and jointput signature.

Keywords: Aspect Oriented Programming (AOP), Cognitive Approach, Pointcut Designator, metric, Weighted Pointcut per Aspect (WPA), Aspect Oriented System (AOS)

1. INTRODUCTION

Software engineering is a difficult and complex task. Software metrics are one way to predict quality within a system, pointing to problem areas that can be addressed prior to software release. Metrics attempt to measure a particular aspect of a software system.Several approaches to estimate complexity of software but none of them have been accepted as a true measure of complexity of an Aspect [4]. Aspect oriented perspective is one of the most significant ways to quantify reliability of software by controlling aspect oriented constructs.

Aspect Oriented Programming (AOP) is a new technology for separating crosscutting concerns that are usually hard to do in object-oriented programming [3]. As AOP has better capability to handle crosscutting concerns than object-orientation it helps to write more modularized and more maintainable code. AspectJ is a general-purpose, aspect-oriented extension to the Java programming language [7]. Given that AspectJ is an extension to Java, every valid Java program is also a valid AspectJ program.

A pointcutis a program construct that selects join points and collects context at those points. For example, a pointcut can select a join point that is a call to a method, and it could also capture the method's context, such as the target object on which the method was called and the method's arguments. In AspectJ has CWPA metric to measure the different type of pointcut designator and joint point signature proposed by various researchers. This metric is didn't prove their metric according to the statistical approach and data are not accurate. So, there is a need for cognitive Weighted, Weighted Pointcut per Aspect (WPA) and prove given data according to the statistical approach for the Aspect level Pointcut measurement. Hence our main goal is to define a CWPA metric to measure the Complexity of pointcut [14].

* Department of Computer Science, Holy Cross College(Autonomous), Tiruchirappalli, India, *Email: gsr_sheeba@yahoo.co.in*

** Department of Computer Science, St. Joseph's College(Autonomous), Tiruchirappalli, India.

2. LITERATURE REVIEW

Several metrics have been proposed for AOP systems by researchers. ManyAOP metrics proposed by Ceccato et.al [14] and KotrappaSirbi et.al [12]. Those metrics are closely interconnected to Weighted Pointcut per Aspect (WPA) metric. But WPA metric proposed by Parthipan, Senthil Velan, and Chitra Babu [1].

WPA is calculated by adding the cognitive weight of the pointcut designator and cognitive weight of the join point signature used in an aspect. Pointcut designator describes when the advices are woven into the join points. The join point signature describes the functions that are related to the respective pointcut definitions. The drawback of the WAA metric is that they didn't prove their metric according to the statistical approach and data are not accurate. Because of empirical data collection, the data doesn't satisfy the Fenton et al[13] properties.

The motivation of proposed metric is discussed in section 3, Empirical Metric Data Collection & Evaluation Criteria 4, the experimentation of a new metric and the case study is described in section 5, a comparative study of CWPA with WPA in Section 7 and 8 presents the conclusion and future scope.

3. METRIC ANALYSIS

3.1. Existing Work

Weighted Pointcut per Aspect (WPA) is calculated by adding weight of the pointcut designator and cognitive weight of the joint point signature used in an aspect [1].

The formula to calculate WPA(a) is given in Equation 1.

$$\mathrm{WPA}(\mathrm{A} = \mathrm{S}_{\mathrm{i}=1}^{\mathrm{m}}\left[CW\ PD_i + CW\ JS_i\right] \quad (1)$$

3.2. Proposed work

Several metrics have been proposed for AOP systems by researchers. One of the metric proposed by Parthipan, SenthilVelan, ChitraBabu [1] is WPA.WPA is calculated by adding weight of the pointcut designator and cognitive weight of the joint point signature used in an aspect [1]. The drawback of the WPA metric is that they didn't prove their metric according to the statistical approach and data are not accurate. Because of empirical data collection, the data doesn't satisfy the Fenton et al. [13] properties. The proposed metric called CWPA, which adding cognitive weight of the pointcut designator (CWPD) and cognitive weight of the joint point signature (CWMC) [1] used in an aspect.

4. EMPIRICAL METRIC DATA COLLECTION & EVALUATION CRITERIA

This section discusses the CWPA metric, empirical data, collection statistics, analysis and its implication.

4.1. WPA Metric

For empirical analysis, WPA metric is selected for AO software. This metric used to find accurate cognitive weight value of pointcut designator using Cognitive Approach.

4.2. Calibration

In this section, an experiment is conducted to assign cognitive weight to the various types of pointcut designator. A comprehension test has been conducted for a group of students to find out the time taken to understand complexity of aspect oriented program with respect to different types of pointcut designator. The group of students selected had sufficient exposure in analysing the Aspect-oriented programs, as they had undergone courses in AspectJ language. 30 students taken from Rural, 30 students taken from Urban were selected to participate in the comprehension test.

Table 1
Categorized Mean Comprehension Time

Programs	*Average Comprehension Time(In Minutes)*				
	Call	*Execution*	*Get*	*Set*	*Handler*
P1	14.33	19.42	23.4	28.85	33.95
P2	13.8	17.47	24.08	28.82	33.98
P3	13.97	19.03	24.31	28.82	33.93
P4	13.33	18.23	23.47	28.89	33.88
P5	13.28	17.15	23.72	28.87	34.02

The time taken by students to comprehend the programs was recorded after the completion of each program. The time taken for comprehension of all these programs was noted and the mean time to comprehend was calculated. Five different programs have been administered in each case. Average time was calculated for each program from the individual time taken by students which shows in Table 1.

The average comprehension time, for programs are listed in table 1. These programs are based on Aspect Oriented Programming. The mean time is also calculated for each category of the programs and is tabulated.

4.3. Statistical Analysis

For each pointcut designator, mean was selected as a measure of central tendency and the standard deviation as a measure of dispersion.

Table 2
Illustrate statistical computation of different types of pointcut designator.

Programs	*Call (CL)*	*Execution (EX)*	*Get (GT)*	*Set (ST)*	*Handler (HD)*
1	0.2388	0.3237	0.39	0.4808	0.5658
2	0.23	0.2912	0.4013	0.4803	0.5663
3	0.2328	0.3172	0.4052	0.4803	0.5655
4	0.2222	0.3038	0.3912	0.4803	0.5647
5	0.2213	0.2858	0.3955	0.4812	0.567
Mean	0.2291	0.3043	0.3966	0.4808	0.5659
Std.Dev.	0.9684	0.8153	0.7483	0.7048	0.7097

4.4. Evaluation Criteria

Types of pointcut designator are compared on the basis of mean and standard deviation.

5. CWPA

The proposed metric called CWPA, which considers the cognitive complexity of the different types of pointcut designator such as call, execution, get, set, handler. The existing WPA proposed by Parthipan, SenthilVelan, ChitraBabu [1]. WPA is calculated by adding weight of the pointcut designator and cognitive weight of the joint point signature used in an aspect [1]. The drawback of the WPA metric is that they didn't prove their metric according to the statistical approach and data are not accurate. CWPA can be calculated using the following equation.

$$CWPA = CWPD + CWMC \quad (2)$$

Cognitive Weighted Pointcut Designator (CWPD)

$$CWPD = ((EX*WFEX) + (CL*WFCL) + (ST*WFST) + (GT*WFGT) + (HD*WFHD)) \quad (3)$$

Cognitive Weighted Method per Class (CWMC)

$$CWMC = ((BFA*WFBFA) + (AFA*WFAFA) + (ARA*WFARA)) \quad (4)$$

Where,

EX	–	Execution	WFEX	–	Weighting Factor of Execution
CL	–	Call	WFCL	–	Weighting Factor of Call
ST	–	Set	WFST	–	Weighting Factor of Set
GT	–	Get	WFGT	–	Weighting Factor of Get
HD	–	Handler	WFHD	–	Weighting Factor of Handler

The Weighting Factor of each type of Pointcut Designator is calibrated in Table 3 using the method discussed in the Empirical Metric Data Collection.

To normalize the mean and standard derivation value to get appropriate weight value. Average mean value of each type of pointcut is divided by corresponding mean standard derivation. Finally weight value is calculated by dividing the values by 20 because to reduce the range of values. The finalize weight values are given as follows,

Table 3
Weight Value of Each type of Joint Point

Joint Point	*Weight Value*
WFCL	0.7
WFEX	1.1
WFGT	1.6
WFST	2
WFHD	2.4

WPA

$$WPA(A = S_{i=1}^{m}\left[CW\ PD_i + CW\ JS_i\right]$$

WPA (A) = 1.5 + 0.4 = 2.1

CWPA

CWPA = CWPD + CWMC

CWPD = ((EX*WFEX) + (CL*WFCL) + (ST*WFST) + (GT*WFGT) + (HD*WFHD))

CWPD = 0.7 + 1.1 + 1.6 + 2 + 2.4 = 7.8

CWMC = ((BFA*WFBFA) + (AFA*WFAFA) + (ARA*WFARA))

CWMC = 1 + 1.33 + 1.66 = 3.99

CWPA = 7.8 + 3.99 = 12.79

Table 4
Pointcut Complexity metric value for the above program

Program#	*Existing Metric Value (WPA)*	*Proposed Metric Value (CWPA)*
1	2.1	12.79

6. COMPARATIVE STUDY

A comparative study has been made with most widely accepted the metric proposed by Parthipan, SenthilVelan, ChitraBabu [1] is WPA.The current CWPA metric is one step ahead of existing WPAmetric; because it takes cognitive weights into consideration and data collection satisfies the fenton et.al [12] properties. In order to compare the proposed metric a comprehension test was conducted to rural and urban degree students. There were sixty students who participated in the test; the students were given five different programs in AspectJ for the comprehension test. Thetest was to find out the output of the given programs. The time taken to complete the test in minutes is recorded. The average time taken by all the students is calculated. In the following Table 5, a comparison has been made with WPA, CWPA and the comprehension test result.

Table 5
Complexity Metric Values and Mean Comprehension Time

Program#	*Existing Metric Value (WPA)*	*Proposed Metric Value (CWPA)*	*Mean Comprehension Time*
1	2.1	12.79	37
2	1	6.06	23
3	1.4	9.39	31
4	1.6	9.99	28
5	2	12.8	42

CWPA is calculating adding cognitive weight of the pointcut designator (CWPD) and cognitive weight of the joint point signature (CWMC) [3] used in an aspect. This is better indicator than the existing WPA. The weight of each type of pointcutis calculated by using cognitive weights and weighting factor similar to which is suggested by Wang et al [6]. It is found that the resulting value of CWPA is larger than the WPA. Cognitive weights for calculation of the CWPA is more realistic because it considers different types of pointcut designator& advice and data are satisfies the fenton et.al [13] properties. The results are shown in the Table 5. A correlation analysis was performed between WPA Vs Comprehension Time with r = 0.900334and CWPA Vs Comprehension time with r = 0.932175. CWPA has more positively correlated than WPA. From the Table 5, it is observed that CWPA value is larger than WPA value which concludes that CWPA is a better indicator of complexity of the aspect with pointcut.

7. DATA COLLECTION PROPERTIES

Fenton et al. [13] defined some properties which were used for the data collection process and are described as follows:

a. Accuracy The higher the difference between the actual data and measured data and the lower is the accuracy and vice-versa. The difference between CWPA and WPA is lower so the accuracy is higher.

b. Replicability Means that the analysis can be done at different times by different people using the same setting. Data are taken from rural and urban PG students at different time.

c. Correctness According to the metrics definition data was collected. The value of CWPA is collected and calculated through the WPA metric.

d. Precision Data is expressed by number of decimal places. Less decimal place shows a lower accuracy. The decimal place of the data is high (i.e. 0.6089).so it shows a higher accuracy.

e. Consistency It counts the differences with the metric values when collected using different tools by different people. Accordingly we found the difference between existing metric - WPA and proposed metric – CWPA by giving different programs by different students.

8. CONCLUSION AND FUTURE SCOPE

A CWPA metric for measuring the aspect level complexity has been formulated. The Pointcutcomplexity of the aspect includes the Advice Execution complexity and Joint Point complexity. CWPA has proven that, complexity of the class getting affected, is based on the cognitive weights of the various types of Joint Point. The metric is evaluated through a statistical analysis, case study and a comparative study, and proved to be a better indicator of the class level complexity. Newer metrics may also be proposed and validated for assessing the cognitive complexity of Aspect.

REFERENCES

[1] Parthipan, SenthilVelan and ChitraBabu , "Design Level Metrics to Measure the Complexity Across Versions of AO Software", IEEE , 2014.

[2] A. Aloysiusand G. ArockiaSahayaSheela, "Aspect Oriented Programming Metrics – A Survey", in the "International Journal of Emerging Trends in Computing and Communication Technology (IJETCCT)",Volume 1, No 3, August 2015.

[3] Chidamber S.R., Kemerer, C.F., "A metrics suit for object oriented design", IEEE, Trans. Software Engineering, vol. 20, pp. 476-498, 1994.

[4] G.ArockiaSahayaSheela, and A.Aloysius, "Design and Analysis of Aspect Oriented Metric CWCAE using Cognitive Approach", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278 – 0181, Vol. 5 Issue 04, April, 2016. (Scopus Indexed)

[5] Brian Todd Bennett, "Locating Potential Aspect Interference using Clustering Analysis", CEC Thesis, Nova Southeastern University, 2015.

[6] Eduardo Kessler Piveta, Ana Moreira, "An empirical study of aspect-oriented metrics", Elsevier, Vol. 78, pp. 117-144, 2012.

[7] Joao M.P.Cardoso, Jose G.F.Coutinho, "LARA: An aspect-oriented programming language for embedded systems", AOSD, 2012.

[8] Francis Thamburaj, A.Aloysius, "Cognitive weighted method hiding factor complexity metric", IJCSSE, Vol.4, Issue 10, pp. 272-279, Oct. 2015.

[9] G. ArockiaSahayaSheela and A. Aloysius, "Analysis of Measuring the complexity of Advice using a Cognitive Approach", "International Journal of Applied Engineering Research (IJAER)", Vol. 10, No.82, 2015.

[10] Aloysius. A., "Coupling Complexity Metric: A Cognitive Approach", Modern Education and Computer Science, vol.9, pp.29-35, 2012.

[11] J. Shao and Y. Wang, "A new measure of software complexity based on cognitive weights.",Canadian Journal of Electrical and Computer Engineering, 2003.

[12] Kotrappa Sirbi and Prakash Jayanth Kulkarni, "Metrics for Aspect Oriented Programming-An Empirical Study", International Journal of Computer Applications, pp.17-23, 2010.

[13] N.Fenton, S.P.fleeger, "Software Metrics: A Rigorous and Practical Approach". PWS Publishing Company, 1997.

[14] Ceccato, M., and Tonella, P.: „Measuring the Effects of Software Aspectization , Proc. Workshop on Aspect Reverse Engineering (WARE 2004), Delft, The Netherlands, November 2004 (Cd-rom)

[15] Joseph D.Gradecki, Nicholas Lesiecki, "Mastering AspectJ – Aspect-Oriented Programming in Java",Wiley Publishing, Inc., 2003.

[16] Bartsch, M., Harrison, R, "An Evaluation of Coupling Measures for AspectJ", LATE Workshop AOSD, 2006.

[17] AnanthiSheshasaayee, Roby Jose, "A Theoretical Framework for the Maintainability Model of Aspect Oriented Systems", Elsevier, SCSE, 2015.

[18] https://eclipse.org/aspectj/doc/next/progguide/semanticsadvice.html

[19] https://eclipse.org/aspectj/doc/released/progguide/startingaspectj.html