

Kindred Pointcut Complexity Metric: A Cognitive Approach

Dr. G. Arockia Sahaya Sheela¹ and Dr.K.R.Martin^{2*}

¹ Department of Computer Science, Holy Cross College (Autonomous), (Affiliated to Bharathidasan University), Trichy – 620002 Tamil Nadu, India.

^{2*} Department of Computer Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University), Trichy – 620002 Tamil Nadu, India.

ABSTRACT

As the paradigm grows in popularity, aspect oriented systems in directive to evaluate their quality become importance. As a result, many Aspect-oriented metrics for evaluating different features of these systems have been presented. This paper provides a novel cognitive complexity metric specifically cognitive weighted point cut between objects for measuring point cut in Aspect- Oriented systems. In computing the Cognitive Weighted Kindred Point cut Complexity (CWKPC) metric, many sorts of point cuts that can exist among aspects: system call, method implementation, field get, field set, exception handler, function Object(), native code call, function Object(), native code execution, and static initialization are taken into account.

KEYWORDS: Software Metrics, method call, method execution, constructor call, constructor execution, field get, field set, exception handler and static initialization, Cognitive Weighed Kindred Point cut Complexity (CWKPC).

1. OVERVIEW

Software industrial is a difficult and time-consuming task. Software metrics are one way to forecast within a system, point to complex areas that can be discussed prior to the deployment of software.. Software Metric is described as “The continuous software of measurement based strategies to the software program envelopment procedure and its software products to deliver significant and timely control information, collectively with using the ones strategies to improve that procedure and its product”[1]. Software metrics are used to assess the quality of a software programme. The term "metrics" refers to a set of precise measurements collected on a certain object or procedure. The primary targets of the software program

metrics are to evaluate and to expect the first-rate of software program. AOP is a programming paradigm that involves breaking down a programme into coherent functional sections that are distributed throughout several regions with the goal of increasing modularity. Many other programming paradigms include support for abstractions (such as classes, methods, and so on) to aggregate and encapsulate concerns in to exact entities. But concerns (including “Logging”) are examples of crosscutting concerns, due to the fact each logged a part of the device is affected through the approach used for logging. The All AOP implementations pay close attention to having proper crosscutting expressions to capture all concerns in one place. Aspect is an aspect-oriented programming (AOP) extension for the Java programming language developed at PARC [2]. It is to be had in Overshadowing Establishment open-supply projects, each backup myself and included into Obscuration. In Perspective has no intellectual weighted Kinded Pointcut metric to degree the various type of Kinded Pointcut proposed through various analysts [15]. Along these lines, there's a requirement for intellectual weighted Kinded Point Cut measurement implied for the Angle stage Kinded Pointcut estimation. Subsequently our key spot is to unequivocal an intellectual weighted Kinded Pointcut metric to degree the Kinded Pointcut on the various levels.

2. KINDED POINTCUT

The kinded Pointcut designators match stranded on the scope of a join point.[17]. These area unit the call, implementation, get, set, pre-initialization, data formatting as initialization, static initialization, handler, and advice execution designators.[1][2]. In Table 1, the typical kinded pointcut with their pointcut syntax are tabulated.

2.1 Method pointcut

2.1.1 The method-execution

The method-execution be part of (join) point incorporates the implementation of all the code at intervals the build of the tactic [7].

2.1.2 The method-call

The method-call be part of (join) point transpires at the places wherever a technique is presence invoked.

2.2 Constructor pointcut

Constructor be part of (join) points square measure associated with technique be part of points, excluding they characterize the execution and request of object construction.

2.2.1 The constructor-execution

The constructor-execution be part of (join) point embraces the implementation of the code at intervals the body of a constructor for associate in object.

TABLE 1 kinded pointcut with their pointcut syntax

Kinded pointcut	Pointcut syntax
Call	
(1) Method call (2) Constructor call	call(Method Signature) call(Constructor Signature)
Execution	
(3) Method execution (4) Constructor execution	execution(Method Signature) execution(Constructor Signature)
(5) Field get	get(Field Signature)
(6) Field set	set(field Signature)
(7) Handler	handler(Type Signature)
(8) Static initialization	static initialization(Type Signature)
(9) Pre-initialization	Pre initialization(Constructor Signature)
(10) Initialization	initialization(Constructor Signature)
(11) Advice execution	advice execution()

2.2.2 Constructor-call

Constructor-call is be a part of (join) points characterize the points that raise the creating of object.

2.3 Field access pointcut

2.3.1 Field get

These be a part of (join) points match to the deliver access to occurrence or category member of a class.

2.3.2 Field set

These are a part of (join) point's match to the write access to an incident or category member of a category.

2.3.3 Exception-handler be a part of (join) points

Envision that you basically wish to answer to any dealt with exemptions of sure sorts. Might be you might want to log the special case prior to rethrowing. Angle offers special case overseer be a piece of (join) focuses, that address the controller block (the catch block) of partner degree exemption sort to shape such a crosscutting execution feasible [16].

2.4 Static data formatting (initialization) pointcut

Class-initialization be a part of points characterize the loading of a category, count the data formatting (initialization) of the static portion.

2.5 Initialization pointcut

Instantiation of a piece of (join) focuses select the initialisation of article, from the arrival of a parent class' constructor till the remainder of the essential alluded to as constructor. Such be a piece of (join, dislike a constructor execution be a piece of (join) point, happens exclusively inside the underlying alluded to as constructor for each sort inside the order. Dislike class-instantiation that happens once a class loader track a class, object initialization happens whenever object is shaped.

2.6 pre-initialization pointcut

The item pre-introduction be a piece of (join) point is every so often utilized. It includes the entry from the essential alluded to as constructor to the instate of its parent constructor. It much incorporates calls made though framing contentions to the super () call inside the constructor.

2.7 Advice execution be a part of (join) points

Not to be outshone by the quality Java builds, Perspective offers one among its own be a piece of (join) focuses that incorporates the execution of any proposal inside the framework. you'll have the option to instruct such be a section concerning (join) focuses for capacities like ID the actual proposal or watching executions of suggestion for unit-testing of perspectives. It's typically intriguing to try not to educate be a section with respect to (join) point while a be a piece of (join) point is mortal recommended, to keep away from a sort of recursive circumstance [2].

3. EXPERIMENT

In this segment, associate experimentation is conducted to allocate purpose weightage to the many quite pointcut deliberated in section 3(except pre initialization, initialisation and advice execution). A comprehension take a look at have been attended for a crowd of scholars to get out the time engaged to acknowledge quality of feature oriented program with relevance dissimilar kinds of Kinded pointcut.[8]. the gathering of scholars selected had adequate acquaintance in exploring the aspect bound programs, as they'd undergone courses in AspectJ language [3]. Thirty students who recorded sixty fifth and on top of within the semester Examination were appointed to require part within the comprehension check. The time taken by students to understand the programs was documented when the completion of every program. The time taken for comprehension of of these programs was illustrious and therefore the time unit to progress was intended [4]. Normal time was pondered for each program from the singular time drew in by understudies that show in Figure1.

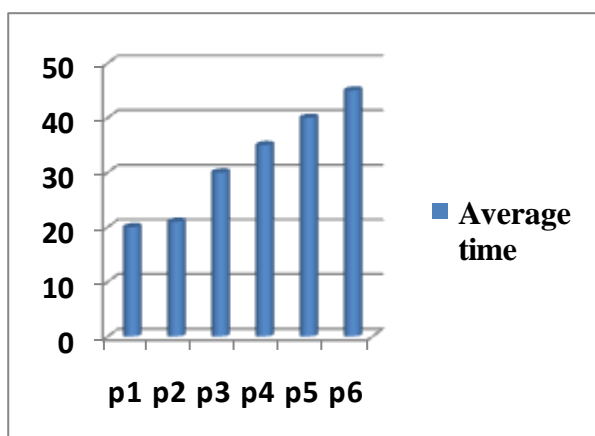


FIGURE 1 Standard conception Time for sample programs

TABLE 2 characterized mean conception time

Programs	Average Comprehension Time	Category
1	19.52	MEX CEX
2	20.37	MCL CCL
3	38.79	FST
4	40.29	FGT
5	55.18	EXH
6	69.7	SINT

In Table 2, the typical conception periods, for programs square measure enumerated. These programs square measure established on aspect orienting programming. The time unit is additionally thought of for every kind of the programs and is mean time of SINT is advanced than MEX, CEX, MCL, CCL, FST, FGT, EXH.

4. Cognitive Weighted Kindred Pointcut (CWKPC)

A metric suite projected by Chindamber and Kemerer (C&K) is one among the best notable set-ups of Item Situated measurements [13] [6]. The leaving CBO metric proposed by C.K in OO Framework rehearses the amount of numeral of articles the current Perspective Kindred Pointcut. Each Kindred Pointcut is dispense a weight 1. This measurement doesn't think the different sorts of Kindred Pointcut. The proposed metric called Intellectual Weighted Kindred Pointcut (CWKPC) [15], which thinks the intellectual intricacy of the divergent sorts of Focuses cuts like call, execution, get, set, static introduction, overseer. CWKPC can be conscious by the Situation as follows [11],

$$CWKPC = ((MEX * WFMEX) + (CEX * WFC EX))$$

$$\begin{aligned}
 &+ (\mathbf{MCL} * \mathbf{WFMCL}) \\
 &+ (\mathbf{CCL} * \mathbf{WFCCL}) \\
 &+ (\mathbf{FST} * \mathbf{WFFST}) \\
 &+ (\mathbf{FGT} * \mathbf{WFFGT}) \\
 &+ (\mathbf{EXH} * \mathbf{WFEXH}) \\
 &+ (\mathbf{SINT} * \mathbf{WFSINT}) \\
 &\text{----> (1)}
 \end{aligned}$$

Where,

MEX is the sum of Method Execution Point cut.

CEX is the sum of Call Execution Pointcut.

MCL is the sum of Method Call Pointcut.

CCL is the sum of Constructor Call Pointcut

FST is the sum of Field Set Pointcut

FGT is the sum of Field Get Pointcut

EXH is the sum of Exception Handler Pointcut

SINT is the sum of Static Initialization Pointcut

The weight factor of every reasonably Kindred Pointcut is standardized exploitation the method discourse within the earlier section and also the values area unit given as follows,

TABLE 3 Kindred Pointcut with their Weight Value

Kindred Pointcut	Weight value
WFMEX WFCEX	1
WFMCL WFCCL	1
WFFST	2
WFFGT	2
WFEXH	3
WFSINT	4

If there are many Aspects specifically CWKPC is the calculation of all CWKPC for discrete classes. The next segment clarifies how CWKPC is considered by earnings of a circumstance learning.

5. METHODOLOGY

The projected complexness metric such as by combining weight one is assessed with the illustration program.

KPC:

$$\text{KPC} = (\text{MEX} + \text{CEX} + \text{MCL} + \text{CCL} + \text{FST} + \text{FGT} + \text{EXH} + \text{SINT})$$

$$\text{KPC} = 1 + 0 + 0 + 1 + 1 + 1 + 1 + 1 = 6$$

CWKPC:

$$\begin{aligned} \text{CWKPC} = & ((\text{MEX} * \text{WFMEX}) \\ & + (\text{CEX} * \text{WFCEX}) \\ & + (\text{MCL} * \text{WFMCL}) \\ & + (\text{CCL} * \text{WFCCL}) \\ & + (\text{FST} * \text{WFFST}) \\ & + (\text{FGT} * \text{WFFGT}) \\ & + (\text{EXH} * \text{WFEXH}) \\ & + (\text{SINT} * \text{WFSINT})) \end{aligned}$$

$$\text{CWKPC} = ((1 * 1) + (0) + (0) + (1 * 1) + (1 * 2) + (1 * 2) + (1 * 3) + (1 * 4))$$

$$\text{CWKPC} = 1 + 1 + 2 + 2 + 3 + 4 = 13$$

Kinded Pointcut Complexity Metric value for the illustration program. In Table 4, the illustration program with their weight value.

TABLE 4 Illustration program with their Weight Value

Sample Program #	KPC	CWKPC
1	6	13

6. In Conclusion and Recommendations

A CWKPC metric intended for calculating the Aspect level difficulty has been expressed. The difficulty of the Aspect comprises the Kinded Pointcut difficulty of the Aspect [20]. CWKPC embraces the cognitive difficulty due to disparate kinds of Kinded Pointcut. CWKPC has confirmed that, difficulty of the aspect accomplishment exaggerated that is made on the cognitive weights of the many varieties of Kinded Pointcut. The allotted cognitive weight of the many varieties of Kinded Pointcut is confirmed exploitation the understanding take a look at and initiate that the cognitive load to apprehend the SINT is overweight than MEX, CEX, MCL, CCL, FST, FGT, EXH. Novel metrics may additionally be advised and supported for considering the cognitive quality of alternative pre initialization, initialization and advice execution Kinded Pointcut [18].

Journals:

[1] G.Arockia Sahaya Sheela, (2017). Design and Analysis of Aspect Oriented Metric CWCOAR using Cognitive Approach, IEEE, ISBN: 978-1-5090-5573-9, pp. 195-197.

- [2] G.Arockia Sahaya Sheela, (2016). Statistical Analysis of Pointcut Complexity Metric using Cognitive Approach, International Journal of Control Theory and Applications (IJCTA), ISSN: 0974-5572, 9(27), 2016, pp. 29-34.
- [3] G.Arockia Sahaya Sheela and A.Aloysius (2015). Analysis of Measuring the complexity of Advice using a Cognitive Approach, International Journal of Applied Engineering Research (IJAER), Vol. 10, No.82.
- [4] G.Arockia Sahaya Sheela, and A.Aloysius (2016). 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. (Scopus Indexed)
- [5] Mandeep Kaur (2015). Improving the Design of Cohesion and Coupling Metrics for Aspect Oriented Software Development, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.5, pg. 99-106, May.
- [6] Aloysius.A. (2012). Coupling Complexity Metric: A Cognitive Approach., Published Online August 2012 in MECS (<http://www.mecs-press.org/>)
- [7] Joseph D. Gradecki, Nicholas Lesiecki., (2003). Mastering AspectJ Measure for Aspect-Oriented Programming in Java. Published by Wiley Publishing, Inc., Indianapolis, Indiana.
- [8] Alexander, R.T., (2003). The Real costs of Aspect-Oriented Programming, IEEE Software.20(6), pp:91-93.
- [9] The AspectJ Team(2001). The AspectJ Programming Guide.
- [10] Mc Cabe, T.J.(1976). A Complexity Measure, IEEE Trans. Software Engineering,SE-2(4),pp.308-320.
- [11] Kiczales,G.,Hilsdale,E.,Hugunin,,J.,Kersten,M.,Palm,J.,Griswold,.W.G, (2001). An Overview of AspectJ, L NCS vol.20-72, pp.327-355.
- [12] Velan, S. Senthil (2019). Investigating the Complexity of Computational Intelligence using the Levels of Inheritance in an AOP based Software, 2019 Advances in Science and Engineering Technology International Conferences (ASET). IEEE.
- [13] K.R.Martin (2017), Coupling on Field Access Metric: A Cognitive Approach, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), ISSN: 2277 128X, Volume 7, Issue 1, January, pp. 208-212, 2017.
- [14] Kumar, A., Kumar, R., and Grover, P.S., (2010). Unified Cohesion Measures for Aspect-Oriented Systems. Accepted for publication in international Journal of Software Engineering and Knowledge Engineering (IJSEKE), September Issue.
- [15] Kumar,A., Kumar, R.,a nd Grover, P.S., (2009a). A Fuzzy Logic Approach to Measure Complexity of Generic Aspect-Oriented Systems. Journal of Object Technology (JOT), 9(3),pp: 43-57,May/June 2010.
- [16] Kumar, A., Grover, P.S., Kumar, R., (2009b). A Quantitative Evaluation of Aspect-Oriented Software Quality Model (AOSQUAMO). ACM SIGSOFT Software Engineering Notes. 34(5), pp:1-9.

- [17] Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., Griswold, W., (2001b). Getting Started with AspectJ. *Communication of the ACM*, vol.44, no.10, October 2001, pp:59-65.
- [18] Kafura, D., Henry, S., (1981). Software Quality Metrics, Based on Interconnectivity, *Journal of Systems and Software*. Vol.2, pp: 121-131. Douence, R., Fradet, P., Südholt, M.: Trace-based aspects. In Filman, R. E., Erlad, T., Clarke, S., Aksit, M. (eds.): *Aspect-oriented Software Development*, Addison-Wesley, 2005, pp. 201–217.
- [19] Hanenberg, S.(2005). Design Dimensions of Aspect-Oriented Systems. PhD dissertation. Duis-burg-Essen University, October 2005.
- [20] Masuhara, H., Kawauchi, K.(2006). Dataflow pointcut in aspect- oriented programming. In 1st Asian Sym. on Programming Languages and Systems, 2003, LNCS vol. 2895, pp.105-121.
- [21] Stein, D., Hanenberg, S., Unland, R.(2006). Expressing Different Conceptual Models of Join Point Selections in Aspect-Oriented Design. In Proc. of AOSD’06, Bonn, Germany, March 2006, ACM Press, pp. 15-26.
- [22] K.R.Martin, Dr.E.Kirubakaran and Dr.E.George Dharma Prakash Raj (2017). Extended Metric of cognitive weighted coupling on Method Call, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, vol.2, Issue 5, September.
- [23] Aloysius, “A cognitive complexity metrics suite for Object-oriented design” Thesis, 2012
- [24] G. Arockia Sahaya Sheela and Dr. A. Aloysius (2018). Aspect Oriented Programming - Cognitive Complexity Metric Analysis Tool, *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN No. 2456-3307, vol.3, Issue 1. (UGC Approved Journal)
- [25] Amalarethnam, DI George, and PH Maitheen Shahul Hameed (2016). Validation of Object Oriented Metrics Using Open Source Software System: Fuzzy Logic Technique, ISSN: 0974-5572, *I J C T A*, 9(27), 2016, pp. 165-173
- [26] G. Arockia Sahaya Sheela and Dr. A. Aloysius (2017). Maintenance Effort Prediction Model using Aspect Oriented Cognitive Complexity Metrics, *International Journal of Advanced Research in Computer Science (IJARCS)*, ISSN No. 0976-5697, vol. 8, Issue 9, pp. 1-4, September. (UGC Approved Journal)



Dr. G. Arockia Sahaya Sheela is working as Assistant Professor in Department of Computer Science, Holy Cross College (Autonomous), Tiruchirappalli, Tamil Nadu, India. She has obtained the Master of Computer Science degree in 2005 and Master of Philosophy degree in 2007 from Bharathidasan University and Doctor of Philosophy in Computer Science degree in

2019, from Bharathidasan University, Trichy. Cloud Metrics and Software Metrics. She has She has 16 years of experience in teaching published many research articles in the Computer Science and Research Experience. National/ International Conferences, and Her research areas are Sensor Networks, Journals.



Dr. K R Martin is working as Assistant Professor in Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has

obtained the Master of Computer Applications degree in 1999, Master of Philosophy degree in 2007 from Bharathidasan University and Doctor of Philosophy in Computer Science degree in 2021 from Bharathidasan University, Trichy. He has 24 years of experience in teaching Computer Science and Research Experience. His research areas are Software Engineering, Cloud Metrics and Software Metrics. He has published many research articles in the National/International Conferences and Journals.