Kinded Pointcut Complexity Metric: A Cognitive Approach

Dr. G. Arockia Sahaya Sheela¹ and Dr. K.R. Martin²*

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

2.2.2 Constructor-call
Constructor-call is 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 are 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
Instatement be 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-instatement 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.
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 Kinded 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 Kinded Pointcut. Each Kinded Pointcut is dispense a weight 1. This measurement doesn't think the different sorts of Kinded Pointcut. The proposed metric called Intellectual Weighted Kinded 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],

\[
\text{CWKPC} = ((\text{MEX} \times \text{WF}_{\text{MEX}}) + (\text{CEX} \times \text{WF}_{\text{CEX}}))
\]
+ (MCL* WFMCL)  
+ (CCL*WFCCL)  
+ (FST*WFFST)  
+ (FGT*WFGT)  
+ (EXH*WFEXH)  
+ (SINT*WFSINT))  

---> (1)

Where,
MEX is the sum of Method Execution Pointcut.
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 Kinded Pointcut is standardized exploitation the method discourse within the earlier section and also the values area unit given as follows,

**TABLE 3 Kinded Pointcut with their Weight Value**

<table>
<thead>
<tr>
<th>Kinded Pointcut</th>
<th>Weight value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFMEX</td>
<td>1</td>
</tr>
<tr>
<td>WFCEX</td>
<td>1</td>
</tr>
<tr>
<td>WFMCL</td>
<td>1</td>
</tr>
<tr>
<td>WFCCL</td>
<td>1</td>
</tr>
<tr>
<td>WFFST</td>
<td>2</td>
</tr>
<tr>
<td>WFGT</td>
<td>2</td>
</tr>
<tr>
<td>WFEXH</td>
<td>3</td>
</tr>
<tr>
<td>WFSINT</td>
<td>4</td>
</tr>
</tbody>
</table>

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 complexity metric such as by combining weight one is assessed with the illustration program.
KPC:
KPC = (MEX + CEX + MCL + CCL + FST + FGT + EXH + SINT)
KPC = 1 + 0 + 0 + 1 + 1 + 1 + 1 + 1 = 6

CWKPC:
CWKPC = ((MEX*WFMEX) + (CEX*WFCEX) + (MCL*WFMCL) + (CCL*WFCCL) + (FST*WFFST) + (FGT*WFFGT) + (EXH*WFEXH) + (SINT*WFSINT))
CWKPC = (((1*1) + (0) + (0) + (1*1) + (1*2) + (1*2) + (1*3) + (1*4))
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>
<thead>
<tr>
<th>Sample Program #</th>
<th>KPC</th>
<th>CWKPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>13</td>
</tr>
</tbody>
</table>

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:


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 16 years of experience in teaching published many research articles in the National/International Conferences, and 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.