Assessing Weight of the Coupling between Objects towards Defect Forecasting in Object Oriented Programming

By A. Saidulu

Abstract- This study presents a new approach for predicting fault proneness in object classes. A new metric called weighted coupling between objects or wcbo, measures a new unit called weighted coupling between objects. The approach then ranks each class of the object based on its fault proneness using the HITS algorithm that measures hub and authority weights in link analysis. The proposed metrics advantage lies in the fact that it can be computed without using any programming language, in a simpler way, comparatively. The study undertook with huge open source systems to forecast fault proneness, shows the projected wcbo fault proneness sensitivity to be 98.7% and compared with CBO which was 42% sensitive.

Keywords: software engineering, design metrics, defect forecasting, coupling between objects, cohesion.

GJCST-C Classification : I.2.5 D.3

Strictly as per the compliance and regulations of:
Assessing Weight of the Coupling between Objects towards Defect Forecasting in Object Oriented Programming

A. Saidulu

Abstract- This study presents a new approach for predicting fault proneness in object classes. A new metric called weighted coupling between objects or wcbo, measures a new unit called weighted coupling between objects. The approach then ranks each class of the object based on its fault proneness using the HITS algorithm that measures hub and authority weights in link analysis. The proposed metrics advantage lies in the fact that it can be computed without using any programming language, in a simpler way, comparatively. The study undertook with huge open source systems to forecast fault proneness, shows the projected wcbo fault proneness sensitivity to be 98.7% and compared with CBO which was 42% sensitive.

Keywords: software engineering, design metrics, defect forecasting, coupling between objects, cohesion.

I. INTRODUCTION

Software fault proneness and maintainability has been outlined imprecisely owing to the complexity and confines in assessing objects failure and reusability. Object-oriented designs two key concepts are coupling and cohesion, however evaluation of reusability of objects applying coupling and cohesion has been limited. Several metrics proposed tried to quantify coupling and cohesion to analyze software fault proneness and maintainability. The paper presents a statistical model with abstraction to contain the degree of fault proneness in OO systems using HITS link analysis algorithm [1].

Object oriented analysis and design primary objective is, software arrangement classes should allow maximum cohesion and low coupling. Coupling is the extent to which the assorted objects interact. Cohesion is the extent to which the functions executed by a subsystem are related. The class design gives an insight of activities, testing efforts, continuance tasks and reuse. Structural metrics coupling and accord are present in majority in structural information. They assess relations like attributes usages or adjustment calls and compute the degree of alternation and relationships in allocation of preceding code essentials such as classes, methods and attributes in acquisitive (OO) software systems. An object is accepted as error free and simpler to adjust if it is in form, calculated and its functionality has been appropriately disseminated to its several dependent objects.

However if the objects are exceptionally inter-reliant then the radical ones are expected to have cogent affects on the performance of others. Proper distribution of function thus forms the basis of the two OO design concepts i.e. coupling and cohesion.

If a reliant object is altered in levels of various functions, then the distribution of functionality is carried to reliant objects and the metrics, cohesion and coupling used to evaluate modified tasks like design quality [2], [3], collision analysis [5], [6], [8], naming the design patterns [11], forecasting software quality [7] and errors [8], [9], [10], etc are accepted advantageous. So for a calculated object in form, ascending cohesion is proper and loose coupling among dependent objects is a helpful characteristic of an object. However to address the issue of objects that that are neither calculated nor dependent, this paper proposes a metric called weighted coupling between objects or wcbo.

The remaining paper is organized as, related work in section II, proposed wcbo metric measuring process in section III, process explored with example in section IV, results and analysis presented in section V, concluding the proposal in section VI and references in section VII.

II. RELATED WORK

The structural metrics analyzing the adeptness of the appulse weight of the anniversary chic with CBO gives inaccurate accountability decumbent coupled classes. In the best of our knowledge and from articles cited recently in conferences and journals, it is evident that the CBO and other CBO related metrics are not sensible to be considered as metrics to predict fault proneness.

The issue of predicting fault-prone classes or simply bug forecast in software has been discussed in seminars and journal publications, generated a number of research papers in the previous 10 years and is an on the go area of research. Unique techniques were designed such as PROMISE [12] and MSR [13] with their specific data sets for prediction of classes fault-prone in software. This paper defines new intangible
metrics for class cohesion and coupling which is distinctive and is an enhancement over earlier cutting edge work. Current research demonstrates that software metrics can be used as good signs for the fault proneness of classes in OO systems [3], [8], [9], [10], [14], [15], [16], [17], [18]. Specific models available use machine learning [9] and logistic regression analysis [3], [9], [14], [15], [16], [17], [18] to develop metric-based models for predicting class faults. So a metric called weighted coupling between objects is presented. The weighted metric coupling assesses using an innovative statistical evaluating method governed by conditions of HITS algorithm [1].

### III. Weighted Coupling between Objects (WCBO) Metric

**a) Hypothesis**

Coupling between objects being high or excessive [19] is unfavorable to modular design and inhibits object reusability [19] [20]. In order to improve modularity and promote encapsulation, inter-object class couples should be kept to a minimum. The more independent a class is, the easier it is to reuse it in another application [19] [21] [22]. As the number of couples increase the sensitivity to changes in other parts of the design automatically increases resulting in maintenance problems [23]. The fault proneness of a class is more if it has high coupling and also if it has more import coupling compared to export coupling [24]. Thus the requirement of rigorous testing arises.

Previous research states [24] the metric CBO compared to other metrics has high sensitivity in predicting fault proneness which however is the possibility predicting the fault proneness rather than predicting the fault proneness. CBO also ranks the objects by their fault proneness that is not sensible. Hence the metric weighted coupling between objects is proposed to predict the fault proneness with high sensitivity. The description of the WCBO measurement process is as follows.

**b) Assumptions**

Let’s consider a set of classes \(c_1, c_2, c_3, \ldots, c_n\)

Two classes ‘\(c_i\)’ and ‘\(c_j\)’, where ‘\(c_i\)’ is coupled with ‘\(c_j\)’ if and only if, any of the methods of ‘\(c_i\)’ invoke any of the methods that belongs to ‘\(c_j\)’ [19].

A graph known as directed graph is created. Vertices represent classes’ i.e. \(c_1, c_2, c_3, \ldots, c_n\).

Edges are flanked by classes and an edge is feasible if at least one method of source class invokes the method of the target class.

A set of classes where each class is shown by an individual path and joined by edges is viewed as a single connected transaction \(ct\).

The vertices belonging to a set of classes in a connected transaction is taken as connected set \(cs\).

The set of all the connected sets is indicated as ‘\(SCS\)’.

**c) Process**

To determine ‘\(wcs\)’ of every class, first a bipartite graph is created between all feasible connected sets related to connected sets \(SCS\) and set of all classes.

![Figure 1: Bi-partite graph between connected sets and classes](image)

A set of connected sets \(SCS\) is shown as a bipartite graph with zero loss of information. Let \(SCS = \{cs_1, cs_2, cs_3, \ldots, cs_m\}\) be a record of connected sets and \(C = \{c_1, c_2, c_3, \ldots, c_n\}\) be the related set of classes. Evidently then is equivalent to the bipartite graph

\[ G = (SCS, C, E) \]

where

\[ E = \{(cs, c) : c \in cs, cs \in SCS, c \in C\} \]

In Fig 1, the bipartite graph shows a class \(c\) that has coupling support proportional to its degree of fault proneness. Connection between connected sets and classes shown in the graph is similar to the association of hubs and authorities in the HITS model [14]. The relation between connected sets and classes is evaluated from their weights; however obtaining the weights from a set of connected sets is a challenge. Naturally a connected set having high coupling weights is supposed to contain several classes with high coupling support. Similarly a class having high coupling support is contained by many sets with high coupling weights. Having different coupling weights for different connection sets is decisive to show their different levels of importance and for working with link-based models for evaluation of connected sets with high coupling weights. The HITS algorithm can be applied to this bipartite graph considering the sets as “pure” hubs and
the classes as ‘pure’ authorities and explore the process as below.

Matrix A represents connected sets and class connections as binary matrix (table 1) i.e. shows the connection between a class and each connected set.

Table 1: Matrix A

<table>
<thead>
<tr>
<th>Cs1</th>
<th>Cs2</th>
<th>Cs3</th>
<th>Cs4</th>
<th>Cs5</th>
<th>Cs6</th>
<th>Cs7</th>
<th>Cs8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Value 1 represents that this class c is existing in related connected set cs.

If c ∈ cs then matrix value is 1
Else if c ∉ cs then matrix value is 0

Consider the matrix that represents each hub initial value as 1 (see fig 2).

Figure 2: Each hub weight is considered as 1 by default and shown as a matrix as below

Then the wcbo of class c can be measured as follows

\[ \text{wcs}(c) = \sum_{i=1}^{m} \{ u(c_{si}) : c \in cs_{i} \} / \sum_{i=1}^{m} u(c_{si}) \]

IV. Ranking Objects and Find Fault Prone Coupling using WCBO: An Example

Let's consider the bi-partite graph in figure 1, the Table 1 is the matrix A generated from that bi-partite graph and the Table 2 is transpose matrix A’ of matrix A.

Initial hub values:

\[ u = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \]

The resultant matrix v generated from \( v = A' \times u \) is

Transparency Notice: The image content is already in a readable format, so there is no need to convert it to a JSON format. The above text is a natural representation of the document content as per the requirements.
Original hub values measured by \( u = A \times v \) gives results as;

In Table 4 each class is listed from their highest to lowest values of their \( wcbo \) values. The classification shows ‘\( C_1 \)’ is highly fault prone.

**Table 4**: Values of weight coupling between objects of the classes of Bi-Partite graph

<table>
<thead>
<tr>
<th>Class</th>
<th>( wcbo ) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0.625</td>
</tr>
<tr>
<td>C2</td>
<td>0.354</td>
</tr>
<tr>
<td>C3</td>
<td>0.604</td>
</tr>
<tr>
<td>C4</td>
<td>0.229</td>
</tr>
<tr>
<td>C5</td>
<td>0.229</td>
</tr>
<tr>
<td>C6</td>
<td>0.375</td>
</tr>
<tr>
<td>C7</td>
<td>0.562</td>
</tr>
<tr>
<td>C8</td>
<td>0.396</td>
</tr>
</tbody>
</table>

The coupling between ‘\( C_1 \)’ to other classes listed in Table 5.

**Table 5**: Weight coupling between objects of ‘\( C_1 \)’ to other classes (export coupling)

<table>
<thead>
<tr>
<th>Coupling</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 \rightarrow C_2 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_1 \rightarrow C_3 )</td>
<td>0.229</td>
</tr>
<tr>
<td>( C_1 \rightarrow C_4 )</td>
<td>0.229</td>
</tr>
<tr>
<td>( C_1 \rightarrow C_5 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_1 \rightarrow C_6 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_1 \rightarrow C_7 )</td>
<td>0.187</td>
</tr>
<tr>
<td>( C_1 \rightarrow C_8 )</td>
<td>0.187</td>
</tr>
<tr>
<td>( C_2 \rightarrow C_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_3 \rightarrow C_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_4 \rightarrow C_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_5 \rightarrow C_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_6 \rightarrow C_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( C_7 \rightarrow C_1 )</td>
<td>0</td>
</tr>
</tbody>
</table>

In Table 5 \( wcbo \) connection of \( C_1 \) with other classes shows, though \( C_1 \) is ranked high in fault analysis, \( C_1 \) fault proneness is confined to \( C_4, C_5, C_7, C_8 \).

The degree of fault proneness \( dfp(c) \) for a class \( c \) can be measured as follows;

\[
dfp(c_i) = \sum_{j=1}^{m} \frac{wcbo(c_i \rightarrow c_j) + wcbo(c_j \rightarrow c_i) : i \neq j}{wcbo(c_j)}
\]

In the case of class \( C_1 \), the degree of fault proneness is 1.3312.

**V. Results Analysis**

The tests performed ensuring high SDLC standards, on application classes with sets of diverse numbers, calculated the accuracy of fault proneness of \( wcbo \).

\[ S(wcbo) = \frac{\text{Classes correctly predicted as fault prone}}{\text{Classes actually fault prone}} \]

**Figure 3**: percentage of Fault proneness prediction accuracy

Sahraoui, God in & Miceli [23] in their empirical studies showed \( CBO > 14 \) is very high against the observed value in our test where “ \( dfp \) ” value higher than 4.65 is severity in fault proneness. In the above graph the weight coupling between objects effectively forecasts degree of fault proneness with 90% accuracy, very efficient compared to CBO.

**VI. Conclusion**

Our research in software engineering is on the ability to forecast the degree of fault proneness in OO systems. The study first evaluated previous research citing, \( CBO \) metric value is directly proportional to fault proneness, proved theoretical and lacked support. Thus a new metric is projected, namely weighted coupling between objects which measures the weighted coupling...
between objects of each class and related coupled classes. Further it was established in predicting the degree of fault proneness, an individual class weight coupling between objects of is insufficient, whereas the wcbo of classes with coupling is able to forecast with an accuracy of 91% and is highly efficient compared to CBO. Further research will focus on developing the accuracy of 91% and is highly efficient compared to wcbo of classes with coupling is able to forecast with an accuracy of 91% and is highly efficient compared to CBO. Further research will focus on developing the statistical approach to calculate the weighted cohesion support and evolving the strategy for predicting the degree fault proneness.

References Références Referencias