Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 8 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

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

ASSESSINGWEIGHTOFTHECOUPLINGBETWEENOBJECTSTOWARDSDEFECTFORECASTINGINOBJECTORIENTEDPROGRAMMING

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

Author : M. Tech in Computer science. e-mail: siddu33@gmail.com.

Global Journal of Computer Science and Technology (C) Volume XIV Issue VIII Version I Year 2014

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 $c1, c2, c3, \ldots\ldots, cn$

Two classes 'ci' and 'cj', where 'ci' is coupled with 'cj' if and only if, any of the methods of 'ci' invoke any of the methods that belongs to 'cj' [19]. A graph known as directed graph is created. Vertices represent classes' i.e. $c1, c2, c3, \ldots\ldots, cn$. 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 bi parted graph is created between all feasible connected sets related to connected sets SCS and set of all classes.

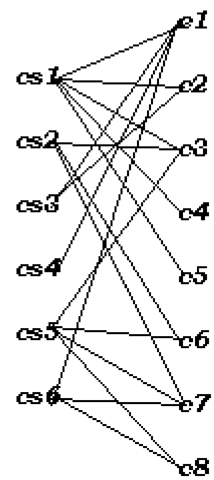

Figure 1 : Bi-partite graph between connected sets and classes

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 $C = \{c_1, c_2, c_3, \ldots\ldots, c_n\}$ 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

	C1	C2	C3	C4	C5	C6	C7	C8
Cs1	1	1	1	1	1	0	0	0
Cs2	0	0	1	0	0	1	1	0
Cs3	1	1	0	0	0	0	0	0
Cs4	1	0	0	0	0	0	0	0
Cs5	0	0	1	0	0	1	1	1
Cs6	1	0	0	0	0	0	1	1

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

If $c \in cs$ then matrix value is 1

Else if $c \notin cs$ then matrix value is 0

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

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

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

Transpose the matrix A as A'

Table 2 : Transpose matrix of matrix below shows the connection between a class and each connected set

	Cs1	Cs2	Cs3	Cs4	Cs5	Cs6
C1	1	0	1	1	0	1
C2	1	0	1	0	0	0
C3	1	1	0	0	1	0
C4	1	0	0	0	0	0
C5	1	0	0	0	0	0
C6	0	1	0	0	1	0
C7	0	1	0	0	1	1
C8	0	0	0	0	1	1

Authority weights are determined by multiplying A' with u i.e. $v = A' \times u$. Matrix representation of authority weights, matrix v is obtained by multiplying A' and u.

The original hub weights are obtained from matrix multiplication between $A \, and \, v$.

$u = A \times v$

Then the wcbo of class c can be measured as follows

$$wcs(c) = \frac{\sum_{i=1}^{m} \{u(cs_i) : c \in cs_i\}}{\sum_{i=1}^{m} u(cs_i)}$$

Then the wcbo of coupling between classes c_i and c_j can be measured as follows

$$wcs(c_j \rightarrow c_k) = \frac{\sum_{i=1}^{m} \{u(cs_i) : (c_j, c_k) \subset cs_i \wedge (c_j \rightarrow c_k) \in cs_i\}}{\sum_{i\ 1}^{m} u(cs_i)}$$

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

Global Journal of Computer Science and Technology (C) Volume XIV Issue VIII Version I Year 2014

Global Journal of Computer Science and Technology (C) Volume XIV Issue VIII Version I Year 2014

$$v = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \\ 1 \\ 2 \\ 3 \\ 2 \end{bmatrix}$$

Original hub values measured by $u = A \times v$ give 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

Class	wcbo value
C1	0.625
C2	0.354
C3	0.604
C4	0.229
C5	0.229
C6	0.375
C7	0.562
C8	0.396

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)

$c_1 \to c_2$	0
$c_1 \to c_3$	0.229
$c_1 \to c_4$	0.229
$c_1 \to c_5$	0
$c_1 \to c_6$	0
$c_1 \to c_7$	0.187
$c_1 \to c_8$	0.187
$c_2 \to c_1$	0
$c_3 \to c_1$	0
$c_4 \to c_1$	0
$c_5 \to c_1$	0
$c_6 \to c_1$	0
$c_7 \to c_1$	0
$c_8 \to c_1$	0

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;

$$dfp(c_i) = \frac{\sum_{j=1}^{m} \{wcs(c_i \to c_j) + wcs(c_j \to c_i) : i \neq j\}}{wcs(c_i)}$$

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 wcs.

$$S(\text{wcs}) = \frac{\text{Classes correctly predicted as fault prone}}{\text{Classes actually fault prone}}$$

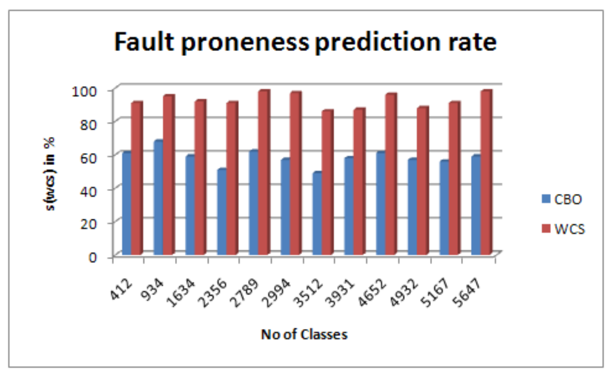

Figure 3 : percentage of Fault proneness prediction accuracy

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 statistical approach to calculate the weighted cohesion support and evolving the strategy for predicting the degree fault proneness.

References Références Referencias

1. J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," J. ACM, vol. 46, no. 5, pp. 604-632, 1999.
2. Bansiya, J. and Davis, C. G., "A hierarchical model for object-oriented design quality assessment", IEEE TSE, vol. 28, no. 1, pp. 4-17., 2002.
3. Briand, L. C., Wüst, J., Daly, J. W., and Porter, V. D., "Exploring the relationship between design measures and software quality in object-oriented systems", Journal of System and Software, vol. 51, no. 3, pp. 245-273., 2000.
4. Briand, L., Wust, J., and Louinis, H., "Using Coupling Measurement for Impact Analysis in OO Systems", in IEEE ICSM'99, pp. 475-482., 19996.
5. Poshyvanyk, D., Marcus, A., Ferenc, R., and Gyimóthy, T., "Using Information Retrieval based Coupling Measures for Impact Analysis", Empirical Software Engineering, vol. 14, no. 1, pp. 5-32., 2009.
6. Wilkie, F. G. and Kitchen ham, B. A., "Coupling measures and change ripples in C++ application software", JSS, vol. 52, pp. 157-164., 2000.
7. Lawrie, D., Feild, H., and Binkley, D., "Leveraged Quality Assessment Using Information Retrieval Techniques", in ICPC'06, pp. 149-158., 2006.
8. El-Emam, K. and Melo, K., "The Prediction of Faulty Classes Using Object-Oriented Design Metrics", NRC/ERB-1064, vol. NRC 43609, 1999.
9. Gyimóthy, T., Ferenc, R., and Siket, I., "Empirical validation of OOmetrics on open source software for fault prediction", TSE,vol.31/10,Oct' 2005.
10. Quah, T.-S. and Thwin, M. M. T., "Application of neural networks for software quality prediction using OO metrics", in ICSM'03,pp. 116-125.
11. Antoniol, G., Fiutem, R., and Cristoforetti, L., "Using Metrics to Identify Design Patterns in Object-Oriented Software", in Proc. of 5th IEEEMETRICS'98, Bethesda, MD, pp. 23 - 34., 1998.
12. http://promisedata.org/
13. http://msr.uwaterloo.ca/
14. Arisholm, E., Briand, L. C., and Foyen, A., "Dynamic coupling measurement for OO software", IEEE TSE, vol. 30, no. 8, pp. 491-506., 2004.
15. Basili, V. R., Briand, L. C., and Melo, W. L., "A Validation of OO Design Metrics as Quality Indicators", IEEE TSE, vol. 22/10, Oct., pp. 751-761., 1996.
16. Briand, L., Melo, W., and Wust, J., "Assessing the Applicability of Fault-Proneness Models across OO Software Projects", TSE, vol.28/7,706-720., 2002.
17. Olague, H., Etzkorn, L., Gholston, S., and Quattlebaum, S., "Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes", IEEE TSE, vol. 33, no. 6, pp. 402-419., 2007.
18. Subramanyam, R. and Krishnan, M. S., "Empirical Analysis of CK Metrics for Object-Oriented Design Complexity: Implications for Software Defects", IEEE TSE, vol. 29, no. 4, pp. 297-310., 2003.
19. Shyam R. Chidamber, Chris F. Kemerer. A Metrics suite for Object Oriented design. M.I.T. Sloan School of Management E53-315. 1993. http://uweb.txstate.edu/~mg43/CS5391/Papers/Metrics/OOMetrics.pdf
20. Lionel C. Briand, John W. Daly, and Jürgen Wüst: A Unified Framework for Coupling Measurement in Object-Oriented Systems. Fraunhofer Institute for Experimental Software Engineering. Kaiserslautern, Germany. 1996. http://www.iese.fraunhofer.de/network/ISERN/pub/ technical_reports/isern-96-14.pdf
21. Victor Basili, Lionel Briand and Walcelio Melo. A Validation of Object-Oriented Design Metrics as Quality Indicators. IEEE Transactions on Software Engineering. Vol. 22, No. 10, October 1996. http://www.cs.umd.edu/users/basili/publications/journals/J60.pdf
22. Roger Whitney: Course material. CS 696: Advanced OO. Docs 6 & 8, Metrics. Spring Semester, 1997. San Diego State University. http://www.eli.sdsu.edu/courses/spring97/cs696/notes/metrics/metrics.html and http://www.eli.sdsu.edu/courses/ spring97/cs696/notes/metrics2/metrics2.html
23. Houari A. Sahraoui, Robert God in, Thierry Miceli: Can Metrics Help Bridging the Gap Between the Improvement of OO Design Quality and Its Automation? http://www.iro.umontreal.ca/~sahraouh/papers/ICSM00.pdf
24. K.K. Aggarwal, Yogesh Singh, Arvinder Kaur, Ruchika Malhotra; Investigating effect of Design Metrics on Fault Proneness in Object-Oriented Systems; JOURNAL OF OBJECT TECHNOLOGY; Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2007