Weighted Cohesin Support (WChS): A Metric to predict the fault proneness of Object Oriented application

AMJAN.SHAIK
Research Scholar,
Department of CSE,
JNTUH, Hyderabad,
Andhra Pradesh, India.
amjansrs@gmail.com

Dr.C.R.K.REDDY
Professor and HOD of CSE
CBIT, Gandipet,
Hyderabad,
Andhra Pradesh, India
crkreddy@gmail.com

Dr.A.DAMODARAM
Professor of CSE &
Director, Academic Audit Cell,
JNTUH,Kukatpally, Hyderabad,
Andhra Pradesh, India.
damodarama@rediffmail.com

Abstract: This paper presents a novel metric called weighted cohesion support as a metric in short WChS metric that is an extension of our earlier weighted coupling support metric in short WCS metric for predicting fault proneness. WChS metric measures a new measurement unit called weighted cohesion support for each class of the object to rank them based on their fault proneness, which is based on the well-known HITS algorithm that measures hub and authority weights in link analysis, One advantage of the proposed metric is that it can be computed in a simpler (and in many cases, programming language independent) way as compared to some of the structural model metrics. We empirically studied WChS for predicting fault-proneness of classes in a large open source systems and fault prediction sensitivity compared with CBO. As the result, we found that the proposed WCS can predict fault proneness nearly with 99% sensitivity, which is around 40% sensitive when compared to LCOM, LCOM2, LCOM3 and LCOM4.

I. INTRODUCTION:

The work reported in this paper arose as part of an idea whose goal is to develop a statistical analysis approach to predict the degree of fault proneness in OO systems. A well designed object, in which the functionality has been appropriately distributed to its various dependent objects, is more likely to be fault free and will be easier to adapt. Appropriate distribution of function underlies two key concepts of object-oriented design: coupling and cohesion. Coupling is the extent to which the various objects interact. If they are highly interdependent then changes to one are likely to have significant effects on the behavior of others. Hence loose coupling between relying objects is a desirable characteristic of an object. Cohesion is the extent to which the functions performed by a subsystem are related. If a dependent object is responsible for a number of unrelated functions then the functionality has been poorly distributed to dependent objects. Hence high cohesion is a characteristic of a well designed object. Many metrics have been proposed to measure the coupling and cohesion to predict the fault-prone and maintainability of software. However, few studies had been done using coupling and cohesion to measure the ability of objects reuse because of their limitations and the difficulties to evaluate the ability of objects reuse.

Coupling and cohesion measures capture the degree of interaction and relationships among source code elements, such as classes, methods, and attributes in object-oriented (OO) software systems. One of the main goals behind OO analysis and design is to implement a software system where classes have high cohesion and low coupling between them. These class properties facilitate comprehension activities, testing efforts, reuse, and maintenance tasks.

A vast majority of coupling and cohesion metrics abound in the literature relies on structural information, which captures relations, such as method calls or attribute usages. These metrics have been proved useful in different tasks, such as, assessment of design quality [1, 2], impact analysis [3, 4, 5], prediction of software quality [6], and faults [7, 8, 9], identification of design patterns [10] etc. However, these structural metrics lack the ability to identify the impact weight of the each class with CBO and highly fault prone coupled classes. To fill this gap we proposed a metric called weighted coupling support WCS [11]. The process of WCS measurement is motivated by HITS link analysis algorithm [12]. With the motivation gained from our earlier proposed metric WCS [11], here in this paper we propose a cohesion measurement based statistical analysis approach to predict the degree of fault proneness in OO systems.

The rest of the paper organized, which exploring the related work in section II , Proposed WCS metric measuring process in section III and section IV explores the process with an example, section V revealed the results analysis, the section VI concludes the proposal that followed by references in section VII.

II. Related Work

Existing research showed that software metrics can be used as good indicators for the fault proneness of classes in OO systems [13, 14, 15, 16, 7, 8, 17, 9, 18]. More specifically, some of the existing approaches also utilized machine learning [8] and logistic regression analyses [13, 14, 16, 16, 8, 17, 18] to build metric-based models for fault prediction. Our paper is different from the previous work as it defines new conceptual metrics for class cohesion and coupling, which appear to be an improvement over the state-of-the-art. Finally, this work explores a set of machine learning techniques and regression analyses to test a number of models based on the combinations of structural and conceptual metrics along with the detailed investigation into principal factors impacting the performance of the conceptual metrics. Finally, prediction of fault-prone classes or simply bug prediction is an active area of research, which produced a number of research publications in the last decade. Besides conference and journal publications on the topic, specialized conferences were organized such as PROMISE [19] and MSR [20] with their specialized data sets for predicting fault-prone classes in software. In the best of our knowledge and articles cited recently in conferences and journals, it is evident that the CBO and other CBO related metrics are not sensual to consider as metrics to predict fault proneness. Hence here we proposed a metric called weighted coupling support that measured by a novel statistical measurement approach, which is defined under the motivation of HITS algorithm [12].

III. WEIGHTED COHESIN SUPPORT (WChS) METRIC

A. Hypothesis

High CBO is undesirable but in contrast height cohesion value of a class is desirable [21]. Excessive coupling between object classes is detrimental to modular design and prevents reuse, but cohesion indicates modularity and ability of the class reuse [21]. The more independent and cohesive a class is, the easier it is to reuse it in another application [21] [22] [23]. In order to improve modularity and promote encapsulation, inter-object class couples should be kept to a minimum and cohesiveness must be kept maximum. A high coupling and low cohesion have been found to indicate fault-proneness; Rigorous testing is thus needed [24].

In the facts of analysis [25] the metric CBO is with high sensitivity in predicting fault proneness that compared to other metrics, but the results indicating that CBO concludes the possibility of the fault proneness but predicting the fault proneness and ranking the objects by their fault proneness is not sensual. Hence in our earlier research paper [11], the metric weighted coupling support proposed to predict the fault proneness with high sensitivity. With the motivation gained from our earlier proposed metric WCS [11], here we propose a new cohesion metric called Weighted Cohesion support metric in short WChS to predict the fault proneness. The description of the WChS measuring strategy follows

The approach of measuring WChS metric proposed in this paper is motivated by our earlier proposed metric WCS [11]. Here in the case of WChS we consider the same bipartite graph to represent the cohesion weights.

B. Assumptions:

Let set of functions $f1, f2, f3, \ldots\ldots, fn$

Let two methods 'fi' and 'fj', 'fi'connected with 'fj'if and only if the method f_i invokes method f_j either direct or indirect.

Build a directed graph with methods as vertices and edges between methods. An edge between the two methods is possible if the method act as source vertex invokes the method acts as target vertex.

Each path of the graph that representing set of methods as vertices and connected with edges will be considered as one connected transaction ct .

The set of methods that are vertices in a connected transaction will be considered as connected set cs .

Hereafter the set of all connected function sets will be referred as '$SCFS$'.

C. Process

In the process of detecting the '$WChS$' of each class, initially we build a bi-parted graph between all possible connected functions set belongs to set of connected function sets SCS and the set of all functions.

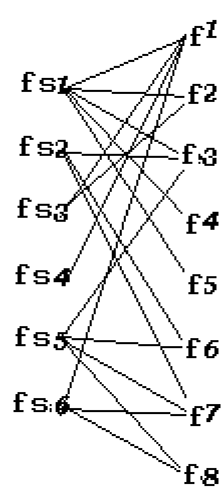

Fig 1: bipartite graph between connected function sets and functions

If a method f_1 one or more times connected to method f_2 of function set fs_1 and $(\{f_1, f_2\} \in c$, then the weight of the connection between f_1 and fs_1 will accumulate by 1(Here c is a class).

Table 1: matrix A as follows that represents the connection weights between a function and each connected function set fs.

	f1	f2	f3	f4	f5	f6	f7	f8
fs1	3	5	2	1	4	0	0	0
fs2	0	0	3	0	0	2	3	0
fs3	2	4	0	0	0	0	0	0
fs4	5	0	0	0	0	0	0	0
fs5	0	0	2	0	0	3	4	1
fs6	1	0	0	0	0	0	5	2

Table 2: Transpose matrix A' of matrix A as fallow that represents the connection between a class and each connected set cs.

	fs1	**fs2**	**fs3**	**fs4**	**fs5**	**fs6**
f1	**3**	**0**	**2**	**5**	**0**	**1**
f2	**5**	**0**	**4**	**0**	**0**	**0**
f3	**2**	**3**	**0**	**0**	**2**	**0**
f4	**1**	**0**	**0**	**0**	**0**	**0**
f5	**4**	**0**	**0**	**0**	**0**	**0**
f6	**0**	**2**	**0**	**0**	**3**	**0**
f7	**0**	**3**	**0**	**0**	**4**	**5**
f8	**0**	**0**	**0**	**0**	**1**	**2**

Let consider a set of connected function sets $SCFS$ as a database and depict it as a bipartite graph without loss of information. Let $SCFS = \{fs_1, fs_2, fs_3, ..., fs_m\}$ be a list of connected sets and $F = \{f_1, f_2, f_3, ..., f_n\}$ be the corresponding set of methods. Then, clearly $SCFS$ is equivalent to the bipartite graph $G = (SCFS, F, E)$ where $E = \{(fs, f) : f \in fs, fs \in SCFS, f \in f\}$.

The graph representation (see fig 1) of the set of connected function sets $SCFS$ is inspiring. It gives us the idea of applying link-based ranking models for the evaluation of connected sets. In this bipartite graph, the cohesion support of a class c is proportional to degree of all its methods weight. However, it is crucial to have different cohesion weights for different connected function sets in order to reflect their different importance. The evaluation of influence connected sets ics should be derived from these weights. Here comes the question of how to acquire weights in a set of connected function sets. Intuitively, a connected function set with high cohesion weights should contain many of the methods those belongs to the same class with high cohesion support; at the same time, a class with high cohesion support should be contained by less or zero other connected function sets with high cohesion weights. The reinforcing relationship of connected function sets and functions is just like the relationship between hubs and authorities in the HITS model [13].

Regarding the connected function sets as "pure" hubs and the methods as "pure" authorities, we can apply HITS to this bipartite graph. The following explored the process:

Let matrix representation of connected function sets and method connections as a binary matrix 'A'(see table 1). The value represents that a method connected how many methods of the same class

If a method f_1 one or more times connected to method f_2 of function set fs_1 and $\{f_1, f_2\} \in c$, then the weight of the connection between f_1 and fs_1 will accumulate by 1(Here c is a class).

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

Fig 2: Initially consider the each hub weight as 1 by default as fallow and represent them as matrix u.

1
1
1
1
1
1
1
1
1
1

Transpose the matrix A as A'(see table 2)

Find Authority weights by multiplying A' with u as $v = A' \times u$ (Matrix multiplication between A' and u gives a matrix v that representing the authority weights)

Now find the original hub weights through matrix multiplication between $A and v$.

$$u = A \times v$$

Then the WChS of method f can be measured as follows

$$WChS(f) = \frac{\sum_{i=1}^{m} \{u(fs_i) : (f \to fs_i) \neq 0\}}{\sum_{i=1}^{m} u(fs_i)}$$

Then the WChS of cohesion between methods f_i and f_j where $(f_j, f_k) \in c$ can be measured as follows

$$WChS(f_j \to f_k) = \frac{\sum_{i=1}^{m} \{u(fs_i) : (f_j, f_k) \subset fs_i \wedge (f_j \to f_k) \in cs_i\}}{\sum_{i=1}^{m} u(cs_i)}$$

D. RANKING OBJECTS AND FIND FAULT PRONE COHESION USING WChS: AN EXAMPLE

Let's consider the bipartite 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 the 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

$$v = \begin{bmatrix} 11 \\ 9 \\ 7 \\ 1 \\ 4 \\ 5 \\ 12 \\ 3 \end{bmatrix}$$

Then measure the original hub values as $u = A \times v$ and resultant hub values are

$$u = \begin{bmatrix} 109 \\ 91 \\ 58 \\ 55 \\ 80 \\ 77 \end{bmatrix}$$

Table 5: WChS values of the methods in example BiPartite graph

Class	WChS value
f1	0.6362
f2	0.3553
f3	0.5957
f4	0.2319
f5	0.2319
f6	0.3638
f7	0.5276
f8	0.3340

Let consider that

$$(f1, f5) \in c1$$
$$(f2, f8) \in c2$$
$$(f3, f6) \in c3$$
$$(f4, f7) \in c4$$

Here in the above equations '$\{c1, c2, c3, c4\}$' are classes. In table 4 we listed the WChS value of the each class. If these classes ranked by their lowest to highest WChS value, the 'c_2' is highly fault prone. The cohesion 'c_1' to other classes listed in table 6

Table 6: WChS of classes '$\{c1, c2, c3, c4\}$

$WChS(c1)$	0.8681
$WChS(c2)$	0.6893
$WChS(c3)$	0.9595
$WChS(c4)$	0.7595

We can find the WChS of all sample classes in table 6. Based on results explored in table 6, it is clear evidence that though c_2 is ranked high as fault prone due to its low WChS values.The class $c3$ ranked low as fault prone due to its heigh WChS value.

In the case of class '$c1$', the fault proneness due to lack of cohesion between methods other than '$\{f1, f5\}$' those belongs to same class '$c1$' limited to . And the degree of fault proneness $dfp(c)$ for a class c can be measured as follows

$$dfp(c_i) = 1 - \frac{\sum_{j=1}^{m} \{WChS(f_j) : (f_j \in c_i)\}}{|c_i|}$$

Here in the above equation $|c_i|$ represents the total number of methods in the class 'c_i'.

E. Results Analysis:

We conducted experiments on applications build by following SDLC standards. We make sure the heterogeneity in the number of classes and the number of methods in each class of the applications considered for experiments. We measured the Fault proneness prediction accuracy of the $WChS$ as follows:

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

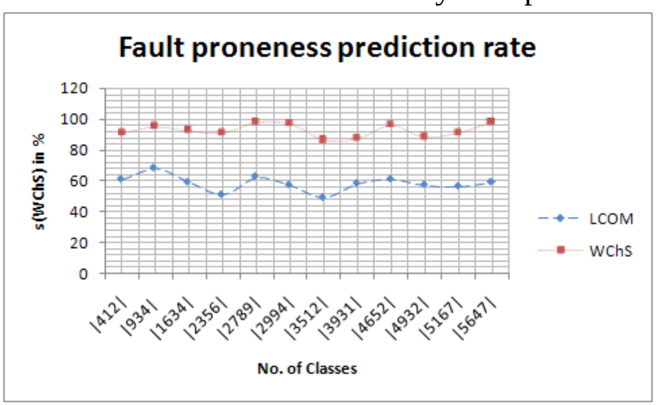

Fig 3: percentage of Fault proneness prediction accuracy

Here in fig 3 we can observe the performance of the WChS in predicting the degree of fault proneness, which stands with an approximate value 97% and miles ahead when compared to LCOM and other cohesion related metrics.

F. Conclusion

In our earlier research paper[11], we proposed a novel coupling metric called Weighted Coupling support (WCS). With the motivation gained from that here in this paper we proposed a metric called weighted cohesive support which is a part of our intention to find a novel methodology to predict the degree of fault proneness in object orient system. Our research considered hypothesis quoted in research papers of software engineering [21] [22] [23] [16] [24] [25] that the metric LCOM value is proportional to fault proneness. But the accuracy of fault proneness prediction is still theoretically hyped. And the past research is not having any proper evidence about the detecting role of a class with high Cohesion value. Hence we proposed a novel metric that measures the weighted cohesion support of each class. The WChS is inversely proportional to fault proneness. The experiments evident that WChS of class alone not significant to conclude the fault proneness, and WChS of all methods belongs to that class helps to predict the degree of fault proneness with average 94% accuracy, which is miles ahead when compared to LCOM and other cohesion related metrics. In our future work we can follow the similar statistical strategy to find the weighted support for other OO metrics considered for predicting fault proneness.

G. References:

[1] 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

[2] 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

[3] Briand, L., Wust, J., and Louinis, H., "Using Coupling Measurement for Impact Analysis in OO Systems", in IEEE ICSM'99, pp. 475-482., 19996

[4] 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

[5] Wilkie, F. G. and Kitchenham, B. A., "Coupling measures and change ripples in C++ application software", JSS, vol. 52, pp. 157-164., 2000.

[6] Lawrie, D., Feild, H., and Binkley, D., "Leveraged Quality Assessment Using Information Retrieval Techniques", in ICPC'06, pp. 149-158., 2006

[7] El-Emam, K. and Melo, K., "The Prediction of Faulty Classes Using Object-Oriented Design Metrics", NRC/ERB-1064, vol. NRC 43609, 1999.

[8] Gyimóthy, T., Ferenc, R., and Siket, I., "Empirical validation of OOmetrics on open source software for fault prediction", TSE,vol.31/10,Oct' 2005.

[9] 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.

[10] 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

[11] Amjan Shaik, Dr. C R K Reddy and Dr. A Damodaram, "Weighted Coupling Support (WCS): A Metric to predict the fault proneness of Object Oriented application" International Journal of Advanced Research in Computer Science (IJARCS); Volume 3, Number 4, July-August 2012

[12] J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," J. ACM, vol. 46, no. 5, pp. 604-632, 1999

[13] Arisholm, E., Briand, L. C., and Foyen, A., "Dynamic coupling measurement for OO software", IEEE TSE, vol. 30, no. 8, pp. 491-506., 2004

[14] 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

[15] 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

[16]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

[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] http://promisedata.org/

[20] http://msr.uwaterloo.ca/

[21]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

[22]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

[23]Roger Whitney: Course material. CS 696: Advanced OO. Docs 6 & 8, Metrics. Spring Semester, 1997. San Diego StateUniversity. 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

[24]Houari A. Sahraoui, Robert Godin, 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

[25] 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.

ABOUT THE AUTHORS:

Amjan.Shaik is a Research Scholar, Department of Computer Science and Engineering, JNTUH, Hyderabad, India. He has received M.Tech. (Computer Science and Technology) from Andhra University. He has been published and presented good number of Research and Technical papers in International Journals, National and International Conferences. His main research interests are Software Metrics, Software Engineering, Software Testing, Software Quality and Object Oriented Design.

Prof. Dr. C.R.K. Reddy is working as a Professor and HOD, Department of Computer Science and Engineering at Chaitanya Bharathi Institute of Technology (CBIT), Hyderabad, India. He has received M.Tech. (Computer Science and Engineering) from JNTUH, Hyderabad and Ph.D in Computer Science and Engineering from Hyderabad Central University (HCU). He has been published and presented wide range of Research and Technical Papers in National, International Journals and National, International

Conferences. At present 8 Research Scholars are doing Ph.D under his esteemed guidance. His main research Interests are Program Testing, Software Engineering , Software Metrics , Software Architectures, Neural Networks and Artificial Intelligence.

Prof. Dr.A.Damodaram working as a Professor of CSE and Director, Academic and Audit Cell-DAAC, JNTUH, Kukatpally, Hyderabad, India. Dr. Damodaram has successfully guided 10 Ph.D. and 2 MS Scholars apart from myriad M.Tech projects. He is currently guiding 9 scholars for Ph.D. Dr. Damodaram is on the editorial board of 2 International Journals and a number of Course materials. Dr. Damodaram has been a UGC nominee for a number of expert and advisory committees of various Indian Universities. Dr .Damodaram has been a Life Member, Vice-President, Director and President of a number of core committees spread all over the country. Dr.Damodaram has published more than 50 well researched papers in reputed National and International journals. He has also presented more than 50 Research papers at different National and International conferences. Dr.Damodaram was honoured with the award of DISTINGUISHED ACADAMICIAN by Pentagram Research Centre, India, in January 2010. His main research interests are Computer Networks, Software Engineering, Security, Ad Hoc Networks, Data Mining and Information Retrieval Systems.