Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.* 

# Impact of Mediated relations as Confounding Factor on Cohesion and Coupling Metrics: For Measuring Fault Proneness in Oo Software Quality Assessment

Amjan.Shaik<sup>1</sup>, Amjan.Shaik<sup>2</sup> and Dr.C.R.K.Reddy<sup>3</sup>

<sup>1</sup> JNTUH, Hyderabad, Andhra Pradesh, India.

Received: 6 December 2011 Accepted: 1 January 2012 Published: 15 January 2012

#### 8 Abstract

<sup>9</sup> Mediated class relations and method calls as a confounding factor on coupling and cohesion

<sup>10</sup> metrics to assess the fault proneness of object oriented software is evaluated and proposed new

<sup>11</sup> cohesion and coupling metrics labeled as mediated cohesion (MCH) and mediated coupling

<sup>12</sup> (MCO) proposed. These measures differ from the majority of established metrics in two

<sup>13</sup> respects: they reflect the degree to which entities are coupled or resemble each other, and they

take account of mediated relations in couplings or similarities. An empirical comparison of the
 new measures with eight established metrics is described. The new measures are shown to be

- <sup>15</sup> new measures with eight established metrics is described. The new measures are show <sup>16</sup> consistently superior at measure the fault proneness.
- 17

Δ

5

6

#### 18 Index terms—

#### <sup>19</sup> 1 Introduction

bject Oriented (OO) design and code, for instance, [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]. These metrics offer ways to 20 evaluate the excellence of software and their use in former phases of software development can help organizations 21 in evaluating large software development quickly, at a low cost [3]. But how do we know which metrics are 22 functional in capturing important quality attributes such as Degree of Fault prone, effort, efficiency or amount of 23 24 maintenance adaptations. Experiential studies of real systems can provide relevant answers. There have been few 25 empirical studies evaluating the effect of objectoriented metrics on software quality and constructing models that utilize them in predicting quality attributes in the system, such as [16,17,18,19,5,20,21,22,23,8,12,24]. More data 26 based by empirical studies, which are capable of being verified by observation or experiment are needed. The 27 evidence gathered through these empirical studies is today considered to be the most powerful support possible 28 for testing a given hypothesis. 29

A well designed component, in which the functionality has been appropriately distributed to its various 30 subcomponents, is more likely to be fault free and will be easier to adapt. Appropriate distribution of function 31 underlies two key concepts of object-oriented design: coupling and cohesion. Coupling is the extent to which 32 the various subcomponents interact. If they are highly interdependent then changes to one are likely to have 33 significant effects on the behavior of others. Hence loose coupling between its subcomponents is a desirable 34 35 characteristic of a component. Cohesion is the extent to which the functions performed by a subsystem are 36 related. If a subcomponent is responsible for a number of unrelated functions then the functionality has been 37 poorly distributed to subcomponents. Hence high cohesion is a characteristic of a well designed subcomponent. 38 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 assess the 39 quality of components. 40

In this context we therefore analyzed the mediated relations of the classes and method calls as a confounding factor for coupling and cohesion metrics and proposing two new metrics called Mediated coupling and Mediated cohesion to measure the fault proneness to assess the quality of the software. The rest of the paper organized as, in section II the traditional cohesion and coupling metrics revealed, which followed by section III that explores transitivity as a confounding factor.

#### 46 **2** II.

## 47 3 The Coupling and Cohesion in OO Programming a) Measur 48 ing Coupling

The term coupling is usually used in a derogatory manner in design review meetings. Even so, it's not possible to design aefficient OO application without coupling. flexible, less scalable application software. However, coupling can be used so that it enables objects to talk to each other while also preserving the scalability and flexibility. Though this seems like a difficult task, OO metrics can help you to measure the right level of coupling.

52 Though this seems like a difficult task, OO metrics can help you to measure the right level of coupling.

Coupling between Objects (CBO): CBO is defined as the number of non-inherited classes associated with the target class. It is counted as the number of types that are used in attributes, parameters, return types, throws clauses, etc. Primitive types and system types (e.g. Java.lang.\*) is not counted.

Data Abstraction Coupling (DAC): DAC is defined as the total number of referring types in attribute declarations. Primitive types, system types, and types inherited from the super classes are not counted.

Method Invocation Coupling (MIC): MIC is defined as the relative number of classes that receive messages from a particular class. nMIC is the total number of classes that receive a message from the target class.

Demeter's Law: Ian Holland first proposed the Law of Demeter. The class form of Demeter's Law has two versions: a strict version and a minimized version. The strict form of the law states that every supplier class of a method must be a preferred supplier. The minimization form is more permissive than the first version and requires only minimizing the number of acquaintance classes of each method.

Definition 1 (Client): Method M is a client of method f attached to class C, if in M message f is sent to an object of class C, or to C. If f is specialized in one or more subclasses, then M is only a client of f attached to the highest class in the hierarchy.

<sup>67</sup> Method M is a client of some method attached to C . Definition 2 (Supplier): If M is a client of class C then C

is a supplier to M. In other words, a supplier class to a method is a class whose methods is called in the method.

 $\,$  69  $\,$  In Listing 1, the Product class is a supplier class to the client class Order.

#### 70 4 Definition 3 (associate Class

 $_{\rm 71}~$  ): A class 1 C is an acquaintance class of method M attached to class 2 C , if 1

72 C is a supplier to M and 1 C is not one of the following:

The same as 2 C ; A class used in the declaration of an argument of M A class used in the declaration of an instance variable of 2 C  $\,$ 

Definition 4 (Preferred-supplier class): Class B is called a preferred-supplier to method M (attached to the

 $_{76}$   $\,$  class C ) if B is a supplier to M and one of the following conditions holds: B is used in the declaration of an

 $^{77}$   $\,$  instance variable of C B is used in the declaration of an argument of M , including C and its super classes. B is

 $_{\rm 78}$   $\,$  a preferred acquaintance class of M .

### <sup>79</sup> 5 b) Measuring Cohesion

In OO methodology, classes contain certain data and exhibit certain behaviors. This concept may seem fairly obvious, but in practice, creating welldefined and cohesive classes can be tricky. Cohesive means that a certain class performs a set of closely related actions. A lack of cohesion, on the other hand, means that a class is performing several unrelated tasks. Though lack of cohesion may never have an impact on the overall functionality of a particular classor of the application itself-the application software will eventually become unmanageable as more and more behaviors become scattered and end up in the wrong places.

Thus, one of the main goals of OO design is to come up with classes that are highly cohesive. Luckily, there's a metric to help to verify that the designed class is cohesive.

### **6** The LCOM Metric: Lack of Cohesion in Methods

The Lack of Cohesion in Methods metric is available in the following three formats: LCOM1: Take each pair of methods in the class and determine the set of fields they each access. If they have disjointed sets of field accesses,

91 the count P increases by one. If they share at least one field access, Q increases by one. After considering each

#### 92 pair of methods:

(

#### 93 7

- 94 ) ( ) If the number of methods or variables in a class is zero (0), LCOM2 is undefined as displayed as zero (0).
- 95 LCOM3: This is another improvement on LCOM1 and LCOM2 and is proposed by Henderson-Sellers. It is
- defined as follows: RESULT P Q ? P Q : 0 => ? A low( ) ( ) ( ) LCOM3 m sum mA / a / m 1
- where m, a, mA, sum mA are as defined in LCOM2.

 $_{98}$  8 =??

99 The following points should be noted about LCOM3:

The LCOM3 value varies between 0 and 2. LCOM3>1 indicates the shortage of cohesion and is considered a kind of alarm.

<sup>102</sup> If there is only one method in a class, LCOM 3 is undefined and also if there are no attributes in a class <sup>103</sup> LCOM3 is also undefined and displayed as zero (0).

Each of these different measures of LCOM has a unique way to calculate the value of LCOM.

An extreme lack of cohesion such as LCOM3>1 indicates that the particular class should be split into two or more classes.

107 If all the member attributes of a class are only accessed outside of the class and never accessed within the 108 class, LCOM3 will show a high-value.

A slightly higher value of LCOM means that you can improve the design by either splitting the classes or re-arranging certain methods within a set of classes.

#### 111 **9 III.**

## 10 Mediated relations of classes and method calls as confound ing factor a) Confounding Factor

The term confounding refers to a situation in which an association between an independent variable and a 114 dependent variable is thought to be the result of the influence of a third variable [17]. The suggestion is that an 115 apparent association between the independent and dependent variables may be partly or completely accounted 116 117 for by a third variable. By the same token, the absence of an apparent association between independent and dependent variables may be the result of a failure to account for the effects of a third variable. The third 118 variable that distorts the true association between the independent and dependent variables is usually called a 119 confounding variable. The distortion that results from perplexing may lead to overestimation or underestimation 120 of an association, depending on the direction and magnitude of the relations that the confounding variable has 121 with the independent and dependent variables [18]. 122

To quantitatively analyze the confounding factor, a number of confounding factor analysis models using various 123 modeling techniques, such as linear, logistic, and probity regression, have been developed [16], [17], [19], [20], [21], 124 [22]. Among these models, the confounding factor analysis model based on linear regression techniques has been 125 widely used in health sciences and epidemiological research [16], [19], [20]. Compared to models based on other 126 modeling techniques, the linear-regression-based model has two main advantages: 1) A number of statistical 127 methods have been developed for this model to test for a confounding variable [16], [19] and 2) it is easy to 128 determine whether a confounding variable leads to overestimation or underestimation of the true association 129 between the independent and dependent variables [16], [20]. 130

#### <sup>131</sup> 11 b) Mediated relation as dependent variable

The objective of this study is to empirically investigate to identify the cohesion and coupling metrics under 132 consideration of mediated class relations and method calls as confounding factors and assessing the association 133 between these cohesion and coupling metrics and degree of fault-proneDegree of Fault prone is an important 134 external quality attribute and identifying faults-prone classes is very useful because: 1) It enables software 135 developers to take focused preventive actions that can reduce maintenance costs and improve quality and 2) 136 it helps software managers to allocate resources more effectively. In this study, Degree of Fault prone denotes 137 the extent of class responsibility in component failure. We need to select the depth of the transitivity in class 138 relations and method calls as the dependent variable for our study. 139

#### <sup>140</sup> 12 Mediated coupling between objects[mcbo]

We begin by regarding any object-oriented software system as a directed graph, in which the vertices are the classes comprising the system. Suppose such a system comprises a set of classes C (C |{ 1.. }) i C i m ? ? = . Let { } m(C) m(C) () | ( 1.. ) j j i j m C i n ? ? =() ( ) 1 m mI mI C C C C C j j i i = ? ? = ? a)() ( ) ( ) mI C C j i cw C C j i mI C C j ? = ? ?

, the directed edge weight also can refer as degree of direct coupling (DDC) between two classes cw is always
 between 0 and 1.

b) Finding a degree of mediated coupling (DMC)

Based on this degree of direct coupling between two classes, we can generalize the process of detecting the degree of mediated coupling mcw between any two classes j C and k C exists such that() 0 mI C C j k??,

which follows: 1 1 ( , ) | | 1 mcw C C p e j k j k cw i i = ? ? ? ? ? ? ? ? ? ? ? ? ? ( ) ( mI 0) j k C C iff ? ? In above equation e j k

? is the set of DDCs of edges, which are building path p between class j C and k C i cw is DDC of an edge i that belongs to j k e ? . p is one of the path out of set of paths P between j

#### <sup>154</sup> 13 C and k C c) Applying Confounding factor

<sup>155</sup> The confounding factor of path p is ( )p cf , that

assessed as follows: | | 1 ( ) ( , ) | | ( ) e p cf C C p j k e p ? = ?

- 157 Here in the above equation( ) p
- $_{158}$   $\,$  e is set edges that belongs to the path p .
- 159 Then the generalized degree of mediated coupling between class j C and k C( ) mcw C C j k ? can be found
- 162 as fault prone:

If 'mcw 'is the degree of mediated coupling between two objects O objects in application's fault proneness.
 V.

#### <sup>165</sup> 14 Mediated cohesion (mch)

166 The proposed cohesion metric is based on transitive function calls. The Hypothesis of the proposed cohesion 167 metric can be defined as:

- 168 We build a graph based on the function calls between the functions of the same class.
- The edge between any two functions represents thetotal number of similar properties used similar functions invoked in both functions. Finding Degree of Direct Cohesion(DDCH)
- The Degree of Direct Cohesion Between two functions that represents the edge weight can be generalized as follows:

#### 173 **15 Year**

If a method A invoking a method B and method B is invoking method C, then the connection between A and C can be considerable and their cohesiveness is transitive if and only if A,B and C belongs to a same class or classes in an inheritance hierarchy .1 1 () | | () (,) (,) 1 mchw M M P j k mchw cf M M i M M i j k j k i = ?

- 177 ? ? ? + ? ? ? ? ? ? ? = ? ?
- Since the class level cohesiveness is significant to predict the fault proneness than the method level cohesiveness. The class level confounding factor of a class C measures as follows:() 1 1 | ' | | C ccf P mC = ????????
- Since the ration between number paths build in the graph and number of methods exists indicates the cohesiveness, if the majority of paths between same classes can be considered as a confounding factor. Hence
- the above equation justifies the measurement of the class level confounding factor. Then the class level mediated the above equation justifies the measurement of the class level confounding factor. Then the class level mediated cohesiveness can be measured as follows:1 1 () | '| () () 1 mchw C P mchw ccf p C i i = ? ? ? + ? ? ? ? ? =
- 184 ? ?

185 Here in the above equation

#### 186 **16** Results analysis

187 We conducted experiments on applications build under SDLC standards. We make sure the heterogeneity in of 188 classes of the applications considered for experiments. We

#### 189 17 Conclusion

These results clearly demonstrate that the proposed metrics MCBO and MCH for coupling and coherence are very good predictors for fault proneness. It is clearly identified that 1. Mediated coupling between two objects is having an impact of the number of connections and path length variation as confounding factors.

### <sup>193</sup> 18 Mediated Cohesion between the methods of a class

is having an impact of the number of paths build between any two methods of a class as confounding factors.

195 These two metrics MCBO and MCH are measuring as numeric values rather in binary quantity.

 $<sup>^{1}</sup>$ © 2012 Global Journals Inc. (US)

<sup>&</sup>lt;sup>2</sup>© 2012 Global Journals Inc. (US)Global Journal of Computer Science and Technology



Figure 1:



Figure 2:



Figure 3:

39 Volume XII Issue XIII Version I D D D D ) ( Global Journal of Computer Science and Technology

Figure 4:

- then there an edge exists between these two methods. The graph is not a directed graph, since edge weight is not changing under any direction of direct connection between the two functions. The DDCH that referred as edge weight can be measured as follows:
- Based on this degree of direct Cohesion between two methods, we can generalize the process of detecting the degree of mediated cohesion mchw between any two methods j M and k M of same class exists such that
- ? is the set of DDCHs of edges, which are building a path p between methods j M and k M i chw is DDCH of an edge i that belongs to j k e ? .
- p is one of the path out of set of paths P
- [Chidamber and Kemerer ()] 'A metrics Suite for Object-Oriented Design'. S Chidamber , C Kemerer . IEEE
  Trans. Software Engineering 1994. 20 (6) p. .
- [Tegarden et al. ()] 'A Software Complexity Model of Object-Oriented Systems'. D Tegarden , S Sheetz , D
  Monarchi . Decision Support Systems 1995. 13 (3-4) p. .
- [Briand et al. ()] 'A Unified Framework for Cohesion Measurement in Object-Oriented Systems'. L Briand , J
  Daly , J Wust . *Empirical Software Engineering* 1998. 3 p. .
- [Briand et al. ()] 'A Unified Framework for Coupling Measurement in Object-Oriented Systems'. L Briand , J
  Daly , J Wust . *IEEE Transactions on software Engineering* 1999. 25 p. .
- [Basili et al. ()] 'A Validation of Object-Oriented Design Metrics as Quality Indicators'. V Basili , L Briand , W
  Melo . *IEEE Transactions on Software Engineering* 1996. 22 (10) p. .
- [Elemam et al. ()] A Validation of Object-Oriented Metrics, K Elemam , S Benlarbi , N Goel , S Rai . ERB-1063.
  1999. National Research Council of Canada (NRC (Technical Report)
- [Cartwright and Shepperd (2000)] 'An Empirical Investigation of an Object-Oriented Software System'. M
  Cartwright , M Shepperd . *IEEE Transactions of Software Engineering*, Aug. 2000. 26 p. .
- [Harrison et al. ()] 'An Evaluation of MOOD set of Object-Oriented Software Metrics'. R Harrison , S J Counsell
  , R V Nithi . *IEEE Trans. Software Engineering* 1998. (6) p. .
- [Aggarwal et al. ()] 'Analysis of Object-Oriented Metrics'. K K Aggarwal , Yogesh Singh , Ruchikamalhotra
  Arvinderkaur . International Workshop on Software Measurement (IWSM), (Montréal, Canada) 2005.
- [Bieman and Kang ()] 'Cohesion and Reuse in an Object-Oriented System'. J Bieman , B Kang . Proc. CM Symp.
  Software Reusability (SSR'94), (CM Symp. Software Reusability (SSR'94)) 1995. p. .
- [Gyimothy et al. (2005)] 'Empirical validation of object-oriented metrics on open source software for fault prediction'. T Gyimothy, R Ferenc, I Siket. *IEEE Trans. Software Engineering* Oct. 2005. 31 p. .
- [Briand et al. ()] 'Exploring the relationships between design measures and software quality'. L Briand , J Daly
  V Porter , J Wust . Journal of Systems and Software 2000. 5 p. .
- [Chidamber et al. ()] 'Managerial use of Metrics for Object-Oriented Software: An Exploratory Analysis'. S
  Chidamber , D Darcy , C Kemerer . *IEEE Transactions on Software Engineering* 1998. 24 (8) p. .
- [Hitz and Montazeri ()] 'Measuring Coupling and Cohesion in Object-Oriented Systems'. M Hitz, B Montazeri
  *Proc. Int. Symposium on Applied Corporate Computing*, (Int. Symposium on Applied Corporate ComputingMonterrey, Mexico) 1995.
- 233 [Lee et al. ()] 'Measuring the Coupling and Cohesion of an Object-Oriented program based on Information flow'.
- Y Lee, B Liang, S Wu, F Wang. International Conference on Software Quality, (Maribor, Slovenia) 1995. [Li and Henry ()] 'Object-Oriented Metrics that Predict Maintainability'. W Li, S Henry. Journal of Systems
- [Li and Henry ()] 'Object-Oriented Metrics that Predict Maintainability'. W Li , S Henry . Journal of Systems
  and Software 1993. 23 (2) p. .
- [Henderson-Sellers ()] Object-Oriented Metrics, Measures of Complexity, B Henderson-Sellers . 1996. Prentice
  Hall.
- 239 [Lorenz and Kidd ()] Object-Oriented Software Metrics, M Lorenz, J Kidd. 1994. Prentice-Hall.
- [Ping and Xiaoxing ()] Predicting Degree of Fault prone using OO Metrics: An Industrial Case Study, Yu Ping
  Ma Xiaoxing , Lujian . 2002. Budapest, Hungary. p. .
- [Briand et al. ()] 'Replicated Case Studies for Investigating Quality Factors in Object-Oriented Designs'. L Briand
  J Wüst , H Lounis . *Empirical Software Engineering: An International Journal* 2001. 6 p. .
- [Elemam et al. ()] 'The Prediction of Faulty Classes Using Object-Oriented Design Metrics'. K Elemam , W Melo
  J Machado . Journal of Systems and Software 2001. 56 p. .
- [Chidamber and Kemerer ()] 'Towards a Metrics Suite for Object Oriented design'. S Chidamber , C Kemerer
  Proc. Conference on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA'91),
- 248 (Conference on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA'91)) 1991.
  26 p. . (Published in SIGPLAN Notices)
- [Lake and Cook ()] 'Use of factor analysis to develop OOP software complexity metrics'. A Lake , C Cook . Proc.
- 6th Annual Oregon Workshop on Software Metrics, (6th Annual Oregon Workshop on Software MetricsSilver
  Falls, Oregon) 1994.
- [Binkley and Schach ()] 'Validation of the Coupling Dependency Metric as a risk Predictor'. A Binkley , S Schach
  International Conference on Software Engineering (ICSE), 1998. p. .