Global Journal of researches in engineering
Numerical Methods
Volume 11 Issue 7 Version 1.0 December 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 2249-4596 & Print ISSN: 0975-5861

Cohesion Metric and Its Relation with Coupling: A Class Level Variable Assessment Approach

By Dr.M.V.Vijaya Saradhi, Dr.B.V.Ramana Murthy

Aurora's Engineering College (AEC), Bhongiri, Nalgonda (Dist), Andhra Pradesh, India

Abstract - Cohesion Metrics is an important technical development which helps for the better Assessment Of class cohesion that is the measurement of relatedness among members In a class. The higher this relatedness is the best the performance will be. Hence this is an important feature of Object oriented systems. Present paper presents an advanced metrics. The pervious metrics have got few limitations. Whereas this advanced metrics considers few more characteristics of class Cohesion. This is based on common object parameters. Moreover this metric is statistically advanced and measures cohesion by observing the relationship between cohesion Metric and Coupling. This is cohesion measurement tool for Java and it is tested on several systems. These systems which are used for the experiment are deferent in size and domain. This test proved that this extended metrics captures other pairs of relatedness among members in a class and also the correlation between cohesion Metric and coupling.

Keywords : Metrics, Cohesion, LCOM, CBO, MIC, DCD, DCDE, DCI, TCC, LCC,MIC,DCIE, Coupling, Class level variables, Spearman correlation,MPC,CLC,OLC.

GJRE Classification : FOR Code: 050299

COHESION METRICAND ITS RELATION WITH COUPLINGA CLASS LEVEL VARIABLE ASSESSMENT APPROACH

Strictly as per the compliance and regulations of :

Cohesion Metric and Its Relation with Coupling: A Class Level Variable Assessment Approach

Dr.M.V.Vijaya Saradhi[α], Dr.B.V.Ramana Murthy[Ω]

Abstract - Cohesion Metrics is an important technical development which helps for the better Assessment Of class cohesion that is the measurement of relatedness among members In a class. The higher this relatedness is the best the performance will be. Hence this is an important feature of Object oriented systems. Present paper presents an advanced metrics. The pervious metrics have got few limitations. Whereas this advanced metrics considers few more characteristics of class Cohesion. This is based on common object parameters. Moreover this metric is statistically advanced and measures cohesion by observing the relationship between cohesion Metric and Coupling. This is cohesion measurement tool for Java and it is tested on several systems. These systems which are used for the experiment are deferent in size and domain. This test proved that this extended metrics captures other pairs of relatedness among members in a class and also the correlation between cohesion Metric and coupling.

Keywords : Metrics, Cohesion, LCOM, CBO, MIC, DCD, DCDE, DCI, TCC, LCC,MIC,DCIE, Coupling, Class level variables, Spearman correlation,MPC,CLC,OLC.

I. Introduction

In Software Engineering there is a raising importance to Metrics and cohesion which are interlinked [28, 27]. Metrics are very much useful in assessing several software characteristics such as complexity, cohesion, coupling and size. Cohesion helps for the better performance and metrics to decide the level of cohesion. So both are equally important in the field of software development. Hence research is going on to improve the cohesion as well as metrics which is a tool to measure it. Cohesion can be defined as the degree of relatedness among elements in a component. It was first introduced and utilized by Yourdon and Constantine in the context of traditional applications. They used it as a tool to estimate level of functional relationships of the elements in a module [30]. These modules are structured for different uses. Class cohesion is a very important feature of object oriented software. This feature of software is of great help to software developers and managers to improve the software quality during the development process.

Author [α] : Professor& HoD in the Dept. of IT at Aurora's Engineering College (AEC), Bhongiri, Nalgonda (Dist), Andhra Pradesh, India. (Telephone : +91-9849275094, E-mail : meduri_vsd@yahoo.co.in),
Author [Ω] : Professor in the Dept.of CSE at Jyothismathi College of Eng. & Tech, Shamirpet, R.R.Dist, Andhra Pradesh, India. (Telephone : +91-9247331980, E-mail : drbvrm@gmail.com)

There are three different types of cohesions. Functional, sequential and coincidental etc [19] is few among them. This feature of software is important because of its wide range uses. If the component has good range of cohesiveness among its members, it can be reused and maintained well [25,7,9,13]. With the help of cohesion one can evaluate the structure quality of class. A class which has good cohesiveness can not be broken easily. It can be used to find out the poorly structured classes. In addition while designing a class cohesion is of great assistance to bring out the best [24]. As a reference we can observe Grady Booch views. He describes that high functional cohesion can be achieved it the elements of a component has good cohesiveness among them which provides some well bounded behavior [8]. This can be brought with single logical function. If all the parts of a class contribute to this single logical function high degree of cohesiveness can be achieved. In contrast if the members are desperate and non related then the coherence will be very low. Since cohesion has such a large scale importance, much number of metrics was proposed to estimate it in object oriented systems. Most of these metrics have been experimented and widely discussed in literature [19,12, 16, 18, 11, 5]. From the above paragraph one can understand that class cohesion is a very important feature of object oriented systems. Many successful metrics have been elaborated and categorized in [10]. These metrics estimate the cohesiveness according to them relatedness among the elements of the class. They count two features; first one is the number of instance variable used by methods and second is the number of method pairs that share instance variables. Though many metrics were proposed in literature they were not total successful in finding out the cohesiveness of classes [22, 13, and 3]. There are some basic reasons for this failure. Some of them are that they do not undertake few features of classes that are the size of cohesive components and relatedness among elements as said in [13].

Few other serious drawbacks which above stated metrics are that they are based only on few categories like instance variable and number of method pairs as stated in [23]. This often leads to wrong estimation of the cohesiveness among members in a component. So the previous metrics face a serious problem when the systems work in functional relationship. In this category, cohesiveness can not be decided by the above said connections but has to be done with the help of the relationships that may exist among methods. If the same old metrics is followed many features of class cohesion will not represented. Hence we believe that class cohesion will not be exact if it does not go beyond above cited categories. Research on source code on several systems tells us that several methods functionally attached even without sharing any instance variable and these can't be divided into different classes. As such the focus is to be extended by taking into account different ways of estimating class cohesion and should not be restricted to any two. First development of the metrics is that connections among class methods will be considered [5, 6]. These systems help to find out much number of pairs of related methods which are not found by previous cohesion metrics. This criterion proved successful when it was tested on several Java systems. In these experiments they gave correct statistical information. In these last years many such developments were brought in. of them [17], concentrates on tradition of maintainability [Zho 03] and [1] on the intimation of mistakes, and [23] on examine the relationship between cohesion and coupling in the one hand and the relationship between cohesion and changeability in the other hand. The area has a raising importance in these last years.

Till now cohesion metrics were limited to object oriented systems but we used this information for other systems also [5, 6]. The previous metrics were based only on instant variables and number of methods pairs. But this information is very primitive to depend on. With this one cannot give good cohesion results. As such research was undertaken to find out other categories which are more authentic and will be helpful to give exact cohesive results. This paper gives an extension of two methods that is DCD and DCI which was proposed in [5, 6]. A new criterion of common object parameters was introduced to calculate cohesion levels. This criterion tells us that two methods of a given class can very well share same object passage in a parameter without being correlated. It does not need to share a method or an instance variable to get connected. So depending on instance variable take us wrong, where as this object passage can present authentic results. Further more it was discovered that in the object context objects themselves collaborate to accomplish a given task. As such certain design principles [24] like design patterns among others and classes play an important role in successful completion of a given job. This collaboration can be located at two levels. One at the group of objects belonging to different classes, second at the collaboration between groups of methods with in a unique given class. This last kind of collaboration can be observed among other things also. These are used in the form of instance variable or passes as arguments at the method level, public in particular. In this kind of collaboration cohesion helps to assign responsibilities to classes [24] in a cohesive manner. Form the above conclusions and according to the experiments done since 2003, this new category to estimate cohesion levels is more dependable. This is proved after conducting many experiments on systems. These clearly show that the extended cohesion metrics based on the addition of the proposed category captured more pairs connected methods than the old metrics DCD and DCI did. These experiments that were done on several systems gave correct, authentic, statistical results.

Software Engineering developers state that there is a correlation between cohesion and coupling. They state that if the cohesion is high, coupling will be low and vice versa [24, 28, 27]. But this notion was not proved by any empirical work. Many experiments were done to bring out the realities and they could only present the necessity for a refined cohesion metrics. They failed because of the limitations of the previous metrics [23]. This paper presents such an extended cohesion metrics. This technique is tested to find out the truth in the relationship between extended cohesion and coupling. Here it was decided to use both the old and new cohesion metrics. The experiment shows that there is a significant correlation between our cohesion metrics and the considered coupling metrics [CBO – Coupling between objects] of Chidamber et al [14, 15]. But the correlation degrees between them varied much when they are presented by the considered cohesion metrics. The empirical experiments as well as the obtained results are discussed in section [7]. Our final goal is to validate the new cohesion metrics as a good indicator for changeability, testability and for many more things. This will be dealt seriously in the future research.

The following paper is arranged in the following way: Section 2 presents an overview of major class cohesion metrics. Section 3 gives the idea of coupling between objects and few important coupling metrics. Section 4 gives some related work which focuses the relatedness of object-oriented metrics and few quality characteristics. Section 5 redefines class cohesion that was proposed depending on the new criterion that we introduced in this paper. Section 6 gives the first step of the experiment that was done (statistic test). Section 7 gives the empirical investigation that we have done to find out the relationship between cohesion and coupling. Finally, conclusions and future work ideas are presented in section 8.

II. Class Cohesion Metrics

Classes are that primary units of object oriented software. In these classes we find cohesion. It is an important feature in software design. The higher the cohesion better the performance will be. A class will have best cohesiveness as stated in [13] if large number of its instance variables are used by a method (LCOM5 [19], Coh [10]), or a larger number of methods pairs share instance variables (LCOM1 [14], LCOM2 [15], LCOM3 [25], LCOM4 [21], Co [21], TCC and LCC [7], DC [4]). Other than the above two, sometimes these metrics also observe the relatedness between the methods to assess cohesion. To get such good cohesion software developers struggle hard at the design phase of classes. If the modeling of these classes is not at its best then cohesion will automatically go down. Hence after designing the classes one would like to assess class cohesion. To assess them many metrics have been proposed in literature. Different authors have defined class cohesion by proposing their cohesion metrics. These cohesion metrics have been presented in detail and are categorized in [10]. One such cohesion metric is LCOM. It is lack of cohesion in methods. It is a metric which is defined by Chidambar and Kemeter [14,15,16]. This metric stands as a role model for many other proposed cohesion metrics. In addition to other proposed metrics, few others tried to redefine LCOM itself. All this cohesion metrics have come into literature to find out the class cohesion in objected oriented systems.

a) Coupling Between Classes

Cohesion metrics measure cohesion between members. Whereas coupling measures the strength of a connection between two modules. Stevens & al [29] explain coupling as one which assesses the strength of the association which is formed by the relatedness between two modules. Coupling between classes helps to assess in which proportion an entity uses other entities. There are both positive and negative effects of the coupling. To speak of the positive low coupling between components helps to minimize interdependencies and gives a chance for evolution [24, 28, 27]. If the modules are structured with low coupling then the complexity of a system can also be minimized. On the negative side because of high coupling module becomes complex. Because of this complexity module will be tough to understand, tough to detect and correct errors, and to change that module. After analyzing these effects one can understand that low coupling is preferable. So it is always encouraged by software engineers. After much research it was discover that with the help of these coupling metrics the maintainability of the oo systems can be easily imagine. Moreover there are few empirical insufficiencies due which there is lot of importance to coupling metrics for the prevision in maintainability [17]. Among coupling metrics, we cite CBO (Coupling between Objects) of Chidamber and Kemerer [15], MPC (Message-Passing Coupling) and DAC (Data Abstraction Coupling) of Li and Henry [25, 26] or OLC (Object Level Coupling) and CLC (Class Level Coupling) of Hitz and Montazeri [21]. Brian & al. counted 23 coupling metrics [9]. For this research work CBO was used which is proposed by [15]. Largely known as a good coupling metric between classes. In our future work, we plan on extending our study to integrate other coupling metrics.

III. Related Work

During the research of the last three decades many software metrics like coupling, cohesion, and complexity were proposed to calculate certain aspects like maintainability, testability, changeability and many more things. But in finding out the quality these were not totally successful. Few reasons are that they are to some extent based on the little understanding of the empirical hypothesis. In addition to it all these proposed metrics can be used to determine any aspect of quality. There is no particular division that this metrics is for the assessment of this quality. As such there is no information about which metric is most suitable to assess a quality. This is the second major problem. Through the research it was also found that of all the metrics only six are efficient and can present sufficient information to depend on. Whereas all the remaining metrics can't give any extra information and they just correspond to subsets of the retained metrics. These drawbacks were discussed by many engineers. Dag &al opines in [17] about the prediction of maintainability. He argues that because of the empirical insufficiencies that these metrics have got the assessment will not be very much dependable. Of all the research papers Dagpinar & al [17] paper presents in new development. In this paper they opines that inheritance cohesion and indirect exportation coupling are not the right factors by which maintainability can be measured well. They advise taking to consideration metrics of size and direct coupling importation which can give good results. A lot of research was undertaken to find out the correlation among the proposed coupling metrics and how much they are prone to fault results. One such research was done by Aggar & al [1]. In prediction model of [2] it was clearly proved that these metrics are very much prone to fault reasons. Zhou & al [31] also undertake similar work to find out the relationship among design metrics (CBO ,WMC, RFC, LCOM etc) and fault proneness when taking fault severity into account. This was understood after conducting a thorough research on many number of oo coupling metrics [1]. This study focuses to find out the best methods according to the given data. After all the research about the relationship between coupling and cohesion leads us to confusion that their may exist a connection between cohesion and for example maintainability, testability as well as fault proneness. Any how a lot more is necessary to find out the direct relationship that may exist between cohesion and above said attributes. This final aspect will be the subject of further research and is out of discussion of the present paper.

IV. Class Cohesion Assesments : a New Measure

Class cohesion at the beginning of this paper is defined as the relative number of related members in a class. This definition is redefined twice in this paper so as to get best methodology which can give authentic assessments. As a first step, two more strategies were added to the relative number. First one is the extension of the methods invocation criteria and indirect utilization of the characteristics explained by Bienan & al in [17]. This idea was extended to the methods invocation criterion as well. After defining the new methodology was tested on several systems [5,6]. This shows a lot of improvement in assessment than the first noted procedure. From the results one can observe that new criteria and the extension of the original criteria are capable of finding out more pairs of connected methods which were not found by the old methods. To come to this conclusion the procedure was tested on several systems. These experiments gave a chance to observe the code of some program. With this code observation and from the obtained results it was found that methods of a class may be functionally, related in other ways. In addition some facts about attributes were also found. From the experiment it was known that attributes, on which first development is dependent on, are not unique to any method. These attributes in reality are reference attribute. Such a one which is not unique but shared is used to assess class cohesion. Many systems were analyzed to come to the above conclusion and observations say that more than 20% of the attributes were reference attributes. This is very much possible oo systems because classes collaborate in accordance with the respective responsibilities so as to finish a given task. Reference attributes are utilized to confirm the needed visibility between objects [24]. Because of these drawbacks we tend to improve this second definition also. We tried to bringing in criteria which will not be primitive, shared but will be more authentic. For this a new criteria that is common objects parameter is introduced. In the following page we explain this and the metrics which work with this new methodology. Our first and second procedures to assess cohesion were already talked about and also tested in the previous papers [5, 6]. This newly introduced criterion is the one which will be prominently discussed in this paper. Both these ways have got many similarities. This new way to assess class cohesion is very much dependent on different connections that are present between its methods. All the three proposed criterions: Attributes Usage Criterion, Methods Invocation Criterion, and Common Objects Parameters will be utilized to find out the functional cohesion in a class. Class cohesion can be said as the connectedness of public methods of a class, with the help of functionalities utilized by its clients. The others methods of the class are included indirectly through the public methods.

a) Attributes Usage Criterion (UC)

Let us take a class C. Let A = {A1, A2,…, Aa} be the group of its characteristics and SPM ={M_1, M_2, …, M_n} be the group of its public methods. Let UCMi be the group of all the characteristics used directly or indirectly by the public method Mi. A characteristic is used directly by a method M_i, if the characteristic shown in the body of the method Mi. The characteristic is indirectly used by the method M_i, if it is used directly by another method of the class that is implored directly or indirectly by Mi. There are n sets UC_{M1}, UC_{M2}, …, UC_{Mn}. Two public methods M_i and M_j are directly related by the UC relation if $UC_{Mi} \cap UC_{Mj} \# \Phi$. It shows that there is at least one characteristic shared (directly or indirectly) by the two methods.

b) Methods Invocation Criterion (MIC)

Let us take a class C. Let SPM = {M1, M2, …, Mn} be the group of its public methods and PRM = {I1, I2, …, Ik} be the group of its other (private and protected) methods. Let SPMMi be the group of all the public methods of the class C, which are implored directly or indirectly by the public method Mi. A public method Mj is called directly by a public method Mi, if Mj is seen in the body of Mi. A public method Mj is indirectly called by a public method Mi, if it is called directly by another method of the class C that is implored directly or indirectly by Mi. There are n sets SPM_{M1}, SPM_{M2}, …, SPM_{Mn}. Let PRM_{Mi} be the group of all the other methods (private and protected) of the class C, which are implored directly or indirectly by the public method Mi. There are n sets PRM_{M1}, PRM_{M2}, …, PRM_{Mn}. Let $MIC_{Mi} = PRM_{Mi} \cup SPM_{Mi}$ be the group of all the methods of the class C, which are implored by the public method Mi. There are n sets MIC_{M1}, MIC_{M2}, …, MIC_{Mn}. Two public methods Mi and Mj are directly connected by the MIC relation if $MIC_{Mi} \cap MIC_{Mj} \# \Phi$. We also take it into account that Mi and Mj are directly related if Mj $\in MIC_{Mi}$ or Mi $\in MIC_{Mj}$.

c) Class level variables (CO)

Let us consider a class C. Let SPM = {M1, M2, …, Mn} be the group of its public methods. Let UCOMi be the group of all the parameters (of object type) of the method Mi. There are n sets UCOM1, UCOM2, …, UCOMn. Two public methods Mi and Mj are directly related by the UCO relation if UCOMi $\cap$ UCOMj # Φ. From the above we understand that there is at least one parameter of object type that is utilized by the two methods.

d) Cohesion based on the direct relation

Two public methods Mi and Mj may be directly interlinked in different ways: they share at least one instance variable in common (UC relation), or get connected at least with another method of the same class (MIC relation), or share at least one object passed as argument (CO relation). In this context, the two methods may be directly interlinked by one or more criteria. It shows that the two methods are directly interlinked if: $UC_{Mi} \cap UC_{Mj} \# \Phi$ or $MIC_{Mi} \cap IM_{Mj} \# \Phi$ or $UCO_{Mi} \cap UCO_{Mj} \# \Phi$.Let us consider a class C with SPM = {M_1, M_2, …, M_n in character are directly connected. Let ED be the number of edges in the graph GD. The level of cohesion in the class C is dependent on the direct connection between its public methods is explained as: DC} the group of its public methods. The highest number of public methods pairs, is n * (n – 1) / 2. Let us take an undirected graph GD, in which vertices are the public methods of the class C, and there is an edge between two vertices if the methods which are equal DC_{DE} = |ED| / [n * (n – 1) / 2] [0,1]. DC_{DE} (as an extension of DC_D [5, 6]) presents the percentage of public methods pairs, which are directly (as defined below) connected. The Lack of Cohesion in the Class (LCC_{DE}) is than given by : LCC_{DE} = 1- DC_{DE} € [0,1].

e) Cohesion based on the indirect relation

Two public methods Mi and Mj can be indirectly connected if they are directly or indirectly related to a method Mk. The indirect relation, brought in by Bieman and Kang in [7], is the transitive closure of the direct relation. We use this idea in our method to mark the indirect related methods. Let us take now an undirected graph GI, where the vertices are the public methods of the class C, and there is an edge between two vertices if the methods are directly or indirectly connected (transitive closure of the graph GD). Let EI be the number of edges in the graph GI. The degree of cohesion in the class C in this case (direct and indirect relations) is said as: DCIE = |EI| / [n * (n – 1) / 2] € [0,1]. DCIE (as an extension of DCI [5, 6]) presents the percentage of public methods pairs, which are directly or indirectly related. The Lack of Cohesion in the Class (LCCIE) is than given by: LCCIE = 1- DCIE € [0, 1].

V. Experimental Study

Several systems were downloaded from the web to experiment on the new criterion. The goal was to achieve significant and general results. To collect the significant data was the main goal of these experiments. Hence many number of Java classes from different systems are taken. Through this experiment it was explored if the proposed criterion is statistically significant before more investigation. We extended the cohesion measurement tool (in Java) for Java programs, that we developed for [6], to automate the computation of our metrics (DCD, DCDE, DCI and DCIE). Many classes in the chosen systems have only one method or do not have any methods. These classes were taken as special classes and have not used for our measurements. All abstract classes are also not used. Overloaded methods within the same class were taken as one method. In addition to it, all special methods (constructors, destructors) were not used. We gathered the values for all the selected metrics from the test systems. For each metric, we calculated some descriptive statistics (minimum, maximum, mean, median, and standard deviation).

VI. Selected Systems

The experiment concerned more than 800 classes. The followed methodology and the obtained results are presented in the following sections. The selected systems are:

- **System1** : JIU0.10 (Java Imaging Utilities) is a library in Java for the change, the edition, the analysis and the backup of pixels of image files (http://sourceforge.net/projects/jiu). This system consists of 180 classes.
- **System2** : JIU0.11 (Java Imaging Utilities) is an improvement of the first system (http://sourceforge.net/projects/jiu) and consists of 191 classes.
- **System3** : FujabaUML is a software development tool which helps for the easy betterment of UML and the progress with Java by adding plug-ins (http://www.fujaba.de). This system consists of 186 classes.
- **System4** : Wbemservices is a Java open source implementation of Web Based Enterprise Management (WBEM) for commercial and non commercial applications. It is compiled of API, of servers, client applications and tools http://wbemservices.sourceforge.net/). It contains 463 classes.

Systems	Des. Stat	DC_D	DC_{DE}	DC_I	DC_{IE}
Jiu1	Mean	0.16027	0.17384	0.1922	0.2178
	Sdt.dev	0.13686	0.1378	0.1638	0.2178
Jiu2	Mean	0.2497	0.2635	0.3102	0.3350
	Sdt.dev	0.16466	0.1714	0.2292	0.2246
Fujaba	Mean	0.01597	0.05244	0.0207	0.0656
	Sdt.dev	0.01479	0.05861	0.0201	0.0739
WBEM	Mean	0.08138	0.2286	0.1013	0.2747
	Sdt.dev	0.14164	0.2051	0.1678	0.2332

Table 1: Average values of cohesion.

VII. Results

We assessed cohesion values for the 4 chosen systems. Table 1 gives the mean values of the metric for the chosen system. The results that we have got for DCDE et DCIE show clearly that they find more pairs of connected methods than DCD et DCI did.

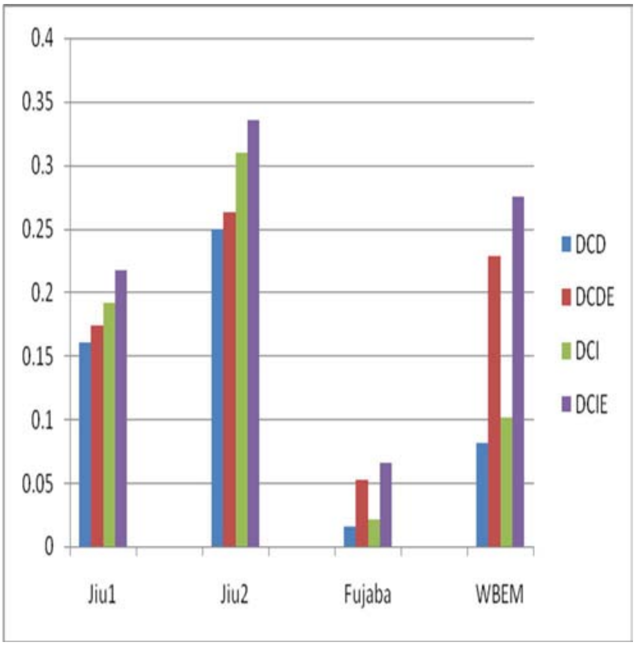

Fig 1: Representation and comparison of the average values of selected systems

From the above figures we can come to a conclusion that DCDE and DCIE are able to find out many other details of attributes of classes which are not found by other metrics. Through this research we would like to prove the importance of new criteria. Hence we are not going to evaluate the cohesion values of the selected systems. From the above given statistics in table 1, it can be said that these systems don't have cohesiveness.

VIII. Validation of the New Criterion

The goal of this chapter is to check the effects of DCD and DCDE on one side and the effects of DCI and DCIE on the other. The goal of this comparison is to find out if there is any difference brought by the introduced new criteria. Through this we would like to prove that DCDE and DCIE are much preferable to DCD and DCI. Because DCDE and DCIE help to find out more pairs of related methods. To prove our above said assumption we have under taken one statistical test: the PAIRED t-TEST [20]:

Let μ1 be the mean value of DCDE (or DCIE) and μ2 be the mean value of DCD (or DCI).

Below we give two statistical hypotheses :

- H0 : μ1= μ2 The metrics are equivalent.
- H1 : μ1> μ2 DCDE (or DCIE) is more significant than DCD (or DCI).

Let Diff be the value of (μ1- μ2). The above test is equivalent to:

- H0 : Diff = 0.
- H1 : Diff >0.

The test statistic is: Z =d/ [Sd / sqrt(N)]

With d : the mean value of sample Diff
Sd : the standard deviation of sample Diff and
N : the number of classes in sample Diff.

Tables 2 and 3 present respectively the comparison between DCD and DCDE on one side and DCI and DCIE on the other.

Systems	Des. Stat	DCD	DCDE	Diff	Z	Zα
Jiu1	Mean	0.16027	0.17384	0.01356	1.799	1.645
	Sdt .dev	0.13686	0.1378	0.01685		
Jiu2	Mean	0.2497	0.2635	0.0228	2.4635	1.645
	Sdt .dev	0.16466	0.1714	0.0207		
Fujaba	Mean	0.01597	0.05244	0.03646	2.6547	1.645
	Sdt .dev	0.01479	0.05861	0.05663		
WBEM	Mean	0.08138	0.2286	0.1472	4.7917	1.645
	Sdt .dev	0.14164	0.2051	0.1869		

Table 2 : Comparison between DCD and DCDE

The methodology consists on comparing Z, for each system, to a value Zα (the value of α is 0.05). If the value of Z is higher than Zα, we do not agree with hypothesis H0 : Diff = 0 and accept H1 : Diff > 0. In this case, the statistical test is significant and we can conclude that metric DCDE (or DCIE) is preferable than metric DCD (or DCI). This means that the added criterion is significant and allows capturing an additional aspect of classes' properties. We have taken data on the metrics from the selected systems and estimated Diff and Z for these systems. These observations are given in tables 2 and 3.

Systems	Des. Stat	DCI	DCIE	Diff	Z	Zα
Jiu1	Mean	0.1922	0.2178	0.0255	1.620	1.645
	Sdt .dev	0.1638	0.2178	0.0352		
Jiu2	Mean	0.3102	0.3350	0.02485	1.5498	1.645
	Sdt .dev	0.2292	0.2246	0.0358		
Fujaba	Mean	0.0207	0.0656	0.0448	2.6494	1.645
	Sdt .dev	0.0201	0.0739	0.0697		
WBEM	Mean	0.1013	0.2747	0.1734	5.7969	1.645
	Sdt .dev	0.1678	0.2332	0.1819		

Table 3 : Comparison between DCI and DCIE

They clearly show that, for the many tested systems Zα lower than Z. The systems for which Zα is higher than Z are the systems for which N is low. Through out the world, the results show that DCDE (or

DCIE) is preferable than DCD (or DCI). This statistical validation shows the applicability of the new cohesion criterion for finding new pairs of connected methods. The results that we have got prove that the extended cohesion metrics, based on the newly introduced criteria, find more pairs of connected methods than metrics DCD and DCI.

IX. The Relationship Between Extended Cohesion Assessment and Coupling

To validate our metrics we went for further experimentation. Software developers believe that cohesion and coupling are correlated. Though not proved it is said that coupling will be low when cohesion is high and vice versa [24, 28, 27]. Using our new criteria of cohesion we tried to know the facts about this belief. If the facts about this can be brought out, we can prove that new metrics on the new criteria is most successful way of assessment. Through these experiments we can bring out the relationship that the metrics can directly have with high level quality attributes like testability, changeability and maintainability. But this is a very beginning stage and no conclusions can be brought. We need more research to confirm the above relationship.

An empirical study

The experiment we performed considered six systems that vary in size (number of classes) and domain. The selected systems are (more than 500 classes):

- **System 1** : Gnujsp 1.0.1, GNUJSP is a free implementation of Java Server Pages of Sun (http://klomp.org/gnujsp). This system contains 56 classes.
- **System 2** : JIU 0.12, JIU (Java Imaging Utilities) is a library in Java for loading, editing, analyzing and saving pixels in image files (http://sourceforge.net/projects/jiu). This system has 77 classes.
- **System 3** : fujabaUml.4, FujabaUML is a software development tool allowing the easy extension of UML and Java development with the use of plug-ins (http://www.fujaba.de). This system contains 60 classes.
- **System 4** : jexcelapi 2.6, JExcelApi is a Java library that grants the possibility of reading, writing and modifying Microsoft Excel Worksheets (http://sourceforge.net/projects/jexcelapi). It contains 110 classes.
- **System 5** : moneyjar 0.8, Moneyjar is a Java library for financial applications. It simplifies treasury management, currency exchange, tax calculations and invoice management (http://sourceforge.net/projects/moneyjar). It contains 20 classes.
- **System 6** : wbemservices 1.0.0, Wbemservices is an open source Java implementation of Web Based Enterprise Management (WBEM) for commercial and non commercial applications. It is a project composed of APIs, of servers, of client applications and of tools (http://wbemservices.sourceforge.net/). This system contains 180 classes.

Experimental Process: First phase

We started our experiment to find out the relationship between cohesion and coupling. Of all the selective systems six were chosen and taken for the experiments. From these data is collected about the four cohesion metrics and also about the CBO metrics. The study conducted with this data proves that there is a definite correlation between cohesion and coupling. Here after it is not a belief but fact that when cohesion is high coupling will be low and inverse is also true.

Experimental Process: Second phase

In this second step we would like to explain how we have come to the above conclusion. From the following results we can prove the hypothesis that there exists a relationship between coupling and cohesion. To prove the above four cohesion metrics: DCI, DCD, DCDE and DCIE and for coupling CBO metric are undertaken. As a first step data on the selected metrics from each of the considered systems is collected. Later to find out the relationship Spearman coefficient was used. This experiment is important because it proves the above said belief. This test is well suited since the dependence seems to be non linear as stated to the previous graphs. Studies of the data sets are done by calculating the Spearman dependence coefficients for each pair of metrics (a metric of cohesion, CBO). The Spearman statistic is based on ranks of the observations. The value of the Spearman statistic is a number between -1 and 1, -1 being a perfect negative dependence and +1 a perfect positive dependence.

Results

X. Regression Study

Main aim of this study is to find out if there is any linear connection between cohesion metrics and coupling. For this a regression study was done between coupling and under different cohesion metrics. As a first step cohesion metric connected to the retained coupling.

Later a regression evaluation between two variables was done. Below are some terms utilized in this section of the paper.

- Regression model: It is the regression model used. DCDE, DCIE, DCD, DCI are the variables which are not dependent and coupling metric CBO is not independent
- Dependant variable: A random variable to predict;
- Independent variable: A predictive variable;
- R^2(r-square): The percentage of change in the dependent variable described by the independent variables in the regression model for the given example of the population.

Global Journal of Researches in Engineering (I) Volume XI Issue VII Version I December 2011

- Population : The group of classes that are chosen into consideration at the experiment level;
- Adjusted R-square: The percentage of change in the dependent variable described by the independent variables in a regression model of the population;
- Sum of squares of regression: The variance of the dependent variable described by the regression model;
- Sum of squares of residual: The change is not described by the dependent variable;
- Mean squares of residual: Total of squared residues divised by the number of freedom degrees of the residues;

To analyze some other variant of this relatedness between the metrics of cohesion and the coupling metric, the logarithm of the coupling value was explained. To get this value a regression between this logarithm and the cohesion is undertaken. The out come of the above is given in table 5.

For this first experiment we have utilized R2 statistic through this we have tried to find out the areas that connect coupling and cohesion. To find out this, values of system JIU are used. For this DCDE and DCIE values are 0.0228 and 0.0267 respectively. These values present the variance of coupling brought in by the cohesion metrics, which can be said as 2.28% and 2.67% in percentages.

System	Cohesion Metric	R^2 vs Coupling
FujabaUml	DC_{DE}	0.0118
	DC_{IE}	0.0081
	DC_{D}	0.0081
	DC_{I}	0.0054
Gnujsp	DC_{DE}	0.2835
	DC_{IE}	0.2676
	DC_{D}	0.4657
	DC_{I}	0.4506
JIU	DC_{DE}	0.0228
	DC_{IE}	0.0267
	DC_{D}	0.0186
	DC_{I}	0.0221
Moneyjar	DC_{DE}	0.0226
	DC_{IE}	0.0237
	DC_{D}	0.032
	DC_{I}	0.0331

Table 4 : Values of R2 in the different systems.

Concerning table 5, for the above said JIU system, values 0.0341 and 0.0430, respectively for cohesion metrics DCDE and DCIE, present the percentages of the logarithm of the variance described by the cohesion metrics. Hence, 3.41% and 4.3% of the logarithm of variance is described respectively by cohesion metrics DCDE and DCIE. Given the obtained values in this test and observing the noted observations in previous part (the relationship seems to be non linear), we undertook a second experiment utilizing the Spearman correlation.

System	Cohesion Metric	R^2 vs logCoupling
FujabaUml	DC_{DE}	0.0118
	DC_{IE}	0.0081
	DC_{D}	0.0081
	DC_{I}	0.0054
Gnujsp	DC_{DE}	0.2835
	DC_{IE}	0.2676
	DC_{D}	0.4657
	DC_{I}	0.4506
JIU	DC_{DE}	0.0228
	DC_{IE}	0.0267
	DC_{D}	0.0186
	DC_{I}	0.0221
Moneyjar	DC_{DE}	0.0226
	DC_{IE}	0.0237
	DC_{D}	0.032
	DC_{I}	0.0331

Table 5 : R2 obtained with the log of a coupling value

Spearman Correlation study (rank statistic)

Further, we calculated the correlation degree (according to Spearman) between the cohesion metrics and coupling in the chosen systems. Table 6 gives the results that we have got.

The aim of this research was to identify a correlation (negative) between the cohesion metrics and coupling metric we have chosen. This tests main goal is to find out if the connectedness is significatively lower than 0 (in the statistical sense) for a negative dependence. A statistical research was conducted. The statistical research must then be correlated to a Student variable computed with n-2 freedom degrees, and where n is the size of the example. The P-value shows the probability of getting such a value under the null hypothesis of absence of dependence. In general, if P-value < 0.05 (error margin), we come to a conclusion that a negative dependence is significant. Hence, for the group of tested systems and from the values of table 6, only the moneyjar system has P-values > 0.05 for all combinations (cohesion metric – coupling metric). We examine values of 0.48996, 0.46740, 0.4649, and 0.442451 for, respectively, cohesion metrics DCIE, DCDE, DCI, DCD correlated to the coupling metric CBO.

For the remaining chosen systems, the P-values are all < 0.05 for the whole group of combinations (cohesion metric – coupling metric). As per table 6, all systems show a significant negative dependence between cohesion and coupling. But this is not possible with moneyjar system. The reason behind this exception is that this particular system has got less number of classes [20] than the other systems. Hence we can come to a conclusion that to observe significant negative dependency is better to select systems with high number of classes. From the above results, it can be said that there is a correlation between cohesion metric and coupling metric.

Other than this it is also observed that if the number of classes are high in a system then the dependency level between cohesion and coupling (Non linear dependency relations) can be confirmed easily. Different kind of systems were selected to prove the correlation between cohesion and coupling and is proved. But there are many other kind of systems on which it is not tested. Hence it is better to prove the same on other systems also before we give any global declaration.

System	Statistic	DCIE-CBO	DCDE-CBO	DCI-CBO	DCD-CBO
Gmujsp	S .Coeff.	0.354545	-0.35892	-0.35455	-0.35892
	Test statistic	-2.786373	-2.8258	-2.78637	-2.8258
	P-value	0.0036697	0.003299	0.00367	0.003299
Jiu	S.Coeff.	0.50857	-0.47888	-0.50337	-0.47584
	Test statistic	-5.11527	-4.72409	-5.04502	-4.68533
	P-value	1.17E-06	5.27E-06	1.53E-06	6.11E-06
fujabaUml	S.Coeff.	0.425590442	-0.43809	-0.29225	-0.31195
	Test statistic	-3.5817696	-3.71155	-2.3273	-2.5005
	P-value	0.000349459	0.000232	0.011731	0.007625
Jexcelapi	S.Coeff.	0.18723	-0.22039	-0.19318	-0.21987
	Test statistic	-1.98076	-2.34805	-2.04612	-2.3423
	P-value	0.02508	0.010346	0.021587	0.010499
Moneyjar	S.Coeff.	0.00602	-0.01955	-0.02105	-0.03459
	Test statistic	-0.02552 -	0.08295 -	0.08934	-0.14683
	P-value	0.48996	0.467402	0.4649	0.442451
WBEM	S.Coeff.	0.242732	-0.295322901	-0.26708	-0.3055
	Test statistic	-3.338282	-4.124041493	-3.69767	-4.28049
	P-value	0.0005133	2.85E-05	0.000145	1.52E-05

Table 6 : Results of the Spearman rank statistic method

XI. Conclusion

With the help of this research we introduced a new criterion and gave a better, revised definition of class cohesion [5, 6]. Common objects parameters are the new criteria which is introduced and also validated in this paper. This becomes the new way to measure class cohesion. We enhanced a cohesion measurement tool for Java programs to automate the calculation of the class cohesion metrics that we propose. Different kinds of systems were taken for the experiment to prove that the new criterion and the proposed metrics for class cohesion give the best assessment. These systems are very much different in size and domain. In this test many number of classes were analyzed. After all the experiments it was understood that the extended metrics with the help of new criterion is capable of finding out more pairs of connected methods. In addition to the above experiment one more was also conducted. It helped to validate our new metrics. This was helpful to prove the correlation between cohesion and coupling. To the second step we got a chance to observe several hundreds of classes. From this second test it was found in the selected systems that there exists a negative correlation between cohesion and coupling. More over through the results we could see that if the number of classes in a system is high then the dependency relation between cohesion and coupling can be confirmed easily.

References Références Referencias

1. K.K.Aggarwal, Yogesh Singh, Arvinder Kaur, RuchikaMalhotra, Empirical study of object-oriented metrics, In Journal of Object Technology, vol. 5. no. 8, November-December 2006, pp. 149-173.
2. K.K.Aggarwal, Yogesh Singh, Arvinder Kaur, RuchikaMalhotra, Investigating the effect of coupling metrics on fault proneness in objectoriented systems, SQP, vol. 8 no. 4, 2006.
3. H. Aman, K. Yamasaki, H. Yamada and MT. Noda, A proposal of class cohesion metrics using sizes of cohesive parts, Knowledge-Based Software Engineering, T. Welzer et al. (Eds), pp. 102-107, IOS Press, September 2002.
4. L. Badri, M. Badri and S. Ferdenache, Towards Quality Control Metrics for Object-Oriented Systems Analysis, Proceedings of TOOLS (Technology of Object-Oriented Languages and Systems) Europe'95, Versailles, France, Prentice-Hall, March 1995.
5. L. Badri and M. Badri, A New Class Cohesion Criterion: An empirical study on several systems, Proceedings of QAOOSE'03, 2003.
6. L. Badri and M. Badri, A proposal of a new class cohesion criterion: An empirical Study, In Journal of Object Technology, 2004. [Bas 96] V.R. Basili, L.C. Briand and W. Melo, A validation of object-oriented design metrics as quality indicators, IEEE

Transactions on Software Engineering, 22 (10), pp. 751-761, October 1996.
7. J.M. Bieman and B.K. Kang, Cohesion and reuse in an object-oriented system, Proceedings of the Symposium on Software Reusability (SSR'95), Seattle, WA, pp. 259-262, April 1995.
8. G. Booch, Object-Oriented Analysis and Design With Applications, Second edition, Benjamin/Cummings, 1994.
9. L. C. Briand, J. Daly, V. Porter, and J. Wuest, The Dimensions of Coupling in Object-Oriented Design, OOPSLA'97,1997.
10. L.C. Briand, J. Daly and J. Wusr, A unified framework for cohesion measurement in object-oriented systems, Empirical Software Engineering, 3 (1), pp. 67-117, 1998.
11. L. Briand, J. Wuest, J. Daly and V. Porter, Exploring the relationships between Design Measures and software quality in object-oriented Systems, Journal of Systems and Software, No. 51, pp. 245-273, 2000.
12. H. S. Chae and Y.R. Kwon, A cohesion measure for classes in objectoriented systems, Proceedings of the fifth International Software Metrics Symposium, Bethesda, MD, pp. 158-166, November 1998.
13. H. S. Chae, Y. R. Kwon and D H. Bae, A cohesion measure for objectoriented classes, Software Practice and Experience, No. 30, pp. 1405- 1431, 2000.
14. S.R. Chidamber and C.F. Kemerer, Towards a Metrics Suite for Object-Oriented Design, Object-Oriented Programming Systems, Languages and Applications (OOPSLA), Special Issue of SIGPLAN Notices, Vol. 26, No.10, pp. 197-211, October 1991.
15. S.R. Chidamber and C.F. Kemerer, A Metrics suite for object Oriented Design, IEEE Transactions on Software Engineering, Vol. 20, No. 6, pp. 476-493, June 1994.
16. S.R. Chidamber, David P. Darcy, and C.F. Kemerer, Mangerial use of metrics for object-oriented sofytware : An exploratory analysis, IEEE Transactions on Software Engineering, Vol. 24, No. 8, pp. 629-639, August 1998.
17. Melis Dagpinar and Jens H. Jahnke, Predicting maintenability with objectoriented metrics – An empirical comparaison, Proceedings of the 10th working conference on reverse engineering (WCRE'03), IEEE computer society, 2003.
18. K. El Emam and W. Melo, The prediction of faulty class using objectoriented design metrics, National Research Council of Canada NRC/ERB 1064, 1999.
19. B. Henderson-sellers, Object-Oriented Metrics Measures of Complexity, Prentice-Hall, 1996.
20. W. W. Hines, D. C. Montgomery, D. M. Goldsman and C. M. Borror, Probability and statistics in engineering, Fourth edition, John Wiley & Sons, Inc., 2003.
21. M. Hitz and B. Montazeri, Measuring coupling and cohesion in object oriented systems, Proceedings of the Int. Symposium on Applied Corporate Computing, pp. 25-27, October 1995.
22. H. Kabaili, R.K. Keller, F. Lustman and G. Saint-Denis, Class Cohesion Revisited: An Empirical Study on Industrial Systems, Proceeding of the Workshop on Quantitative Approaches Object-Oriented Software Engineering, QAOOSE'2000, France, June 2000.
23. H. Kabaili, R.K. Keller and F. Lustman, Cohesion as Changeability Indicator in Object-Oriented Systems, Proceedings of the Fifth European Conference on Software Maintenance and Reengineering (CSMR 2001), Estoril Coast (Lisbon), Portugal, March 2001.
24. G. Larman, Applying UML and Design Patterns, An introduction to object-oriented analysis and design and the unified process, Second edition, Prentice Hall, 2003.
25. W. Li and S. Henry, Object oriented metrics that predict maintainability, Journal of Systems and Software, Vol. 23, pp. 111-122, February 1993.
26. W. Li, S. Henry, D. Kafura and R. Schulman. Measuring Object-Oriented Design. In Journal of Object-Oriented Programming, Vol. 8, No. 4, pages 48-55, July/August 1995.
27. R. S. Pressman, Software Engineering, A practitioner's approach, Fifth edition, Mc Graw Hill, 2005.
28. I. Sommervile, Software Engineering, 2004.
29. W.P. Stevens, G.J. Myers and L.L. Constantine. Structured Design. In IBM Systems Journal, Vol. 13, No. 2, pages 115-139, May 1974.
30. E. Yourdon and L. Constantine, Structured Design, Prentice Hall, Englewood Cliffs, N.J., 1979.
31. Yuming Zhou, Hareton Leung, Empirical analysis of object-oriented design metrics for predicting high and low severity faults, IEEE Transactions on software engineering, vol. 32, no. 10, October 2006.