Effect of coupling on software faults: An empirical study

Conference Paper · March 2017

DOI: 10.1109/C-CODE.2017.7918930

Sajid Anwer
Griffith University

Mohammad Alshayeb
King Fahd University of Petroleum and Minerals

Muhammad Shakeel Anjum
National University of Computer and Emerging Sciences

Effect of Coupling on Software Faults: An Empirical Study

Sajid Anwer, Ahmad Adbellatif,
Mohammad Alshayeb
Information & Computer Science Department
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
{sajidanwer, aabdellatif, alshayeb}@kfupm.edu.sa

Muhammad Shakeel Anjum
Computer Science Department
National University of Computer & Emerging Sciences
Chiniot-Faisalabad Campus, Pakistan
shakeel.anjum@nu.edu.pk

***Abstract*—Software product's quality is one of the important aspects that affect the user, the developer, and the product. Measuring quality in the early phases of the project life cycle is a major goal of project planning. Accordingly, several research studies have been proposed to measure the software product quality attributes. In this paper, we empirically study the impact of afferent coupling (Ca), efferent coupling (Ce) and coupling between object (CBO) metrics on fault prediction using bivariate correlation. We built a prediction model using these metrics to predict faults by using multivariate logistic linear regression. A case study of an open source object oriented systems is used to evaluate the correlation between coupling metrics and faults. The results indicate that the efferent coupling (Ce) is a better indicator for fault prediction than afferent coupling (Ca) and CBO (coupling between object)**

Keywords— Software Metrics; Fault Prediction; Afferent Coupling; Efferent Coupling

I. INTRODUCTION

Delivering high-quality software products is one of the major goals of any development organization; also it becomes a fundamental demand of the users. Faults prediction in the early phases of software development life cycle improve the software quality, increase the software life, and reduce the cost and the time to market. It also increases the customer satisfaction. Many research studies have been proposed to predict faults based on different metrics using artificial intelligence techniques and regression models.

Studying the impact of different metrics on fault prediction in OO systems is becoming very important. In literature, many studies [1-7] used the metrics suites (C&K), QMOOD and MOOD with decision trees, neural networks, and regression models to evaluate the relationship between different metrics and faults in OO systems. Their results indicate that some of the metrics have a significant effect on fault predictions. They reported that CBO metric is a better indicator for fault prediction. But CBO depicts a very generic view of coupling, literature could not differentiate whether one class is affecting the whole system (one class is coupled with all other classes) or all classes are coupled with each other.

In this paper, we study the impact of afferent coupling, efferent coupling and CBO on fault prediction using bivariate correlation. We assemble a case study of 7 object oriented systems to find the correlation between Ca, Ce, CBO, and faults. We also built a prediction model using multivariate logistic linear regression. The results of this paper tend to helps project managers and designers to predict faults as early as possible.

The rest of this paper is organized as follows; section 2 discusses the theoretical concepts. Section 3 discusses the different approaches for fault prediction presented in the literature. Section 4 describes the data that was used in this study and the proposed methodology. Section 5 analyzes the results and section 6 presents threats to validity for this study. Finally, section 7 discusses the conclusion and the future work.

II. THEORETICAL BASIS

This section describes the different entities that we used in this study. In our study, we used CBO (Coupling between objects), afferent coupling (Ca), and efferent coupling (Ce) as independent variables to predict faults (dependent variable). Below, we list the definitions of the used metrics.

A. CBO (coupling between objects)

"The coupling between object classes (CBO) metric represents the number of classes coupled with a given class. This coupling can occur through method calls, field accesses, inheritance, arguments, return types, and exceptions."[8]

B. Ca(Afferent couplings)

"A class's afferent coupling is a measure of how many other classes use the specific class. Coupling has the same definition in the context of Ca as that used for calculating CBO."[8]

C. Ce(Efferent couplings)

"A class's efferent coupling is a measure of how many other classes is used by the specific class. Coupling has the same definition in the context of Ce as that used for calculating CBO."[8]

978-1-5090-4448-1/17/$31.00 ©2017 IEEE

III. LITERATURE REVIEW

This section describes the existing studies related to fault prediction, and at the end what are shortcomings in this studies and how this study overcome those shortcomings.Singh et al. [6] used C&K metrics suite and SLOC (Software lines of code) metric to predict the fault-proneness in large size OO systems. They developed models using linear regression and data mining techniques on these metrics to predict faults and severity level. They categorized faults in three different levels i.e. severe, medium and low according to their effects. They used NASA data set as an input to those metrics. They reported that CBO, SLOC, WMC, and RFC have better results for predicting faults across severity levels. They also reported that NOC is not a good indicator for prediction high and low severity faults.

Pai and Beta [5] calculated conditional probability densities of Bayes Network by using linear regression, logic regression, and Poisson regression. After that, they used these networks to calculate the fault proneness of NASA Datasets. They used object-oriented metrics suite (C&K) and one structural metrics (lines of code) to measure different aspects of selected systems. They set precision, specificity, sensitivity, false negative parameters and false positive parameters as their evaluation parameters. Based on the results and evaluation parameters they reported that RFC, CBO, LOC and WMC metrics are very useful in the prediction of fault-proneness of the modules. Poisson regression results showed that LCOM is significant for predicting fault proneness, while NOC and DIT are not significant in predicting fault proneness.

Olague et al. [4] used C&K, QMOOD and MOOD metrics suites to predict the faults in object-oriented systems. They applied these metrics suites on open source project Rhino (six versions) developed based on agile software development process. They analyzed their results by using multivariate binary logistic regression (MBLR) and univariate binary logistic regression (UBLR); they also used spearman correlation to examine the effect of individual metrics on fault prediction. After applying these techniques, they reported that QMOOD and C&K have better results than MOOD in fault prediction. UBLR analysis showed that class interface size metric (CIS) which represent the count of the public methods in a class and number of methods in a class (NOM) metrics from MOOD metrics suite have better results for fault prediction. Moreover, their analysis showed that CBO, RFC and WMC metrics from C&K suite are useful in fault prediction. Their analysis also showed that C&K metrics suite is more useful for fault prediction.

Aggarwal et al. [1] used 36 metrics (include C&K and Briand metrics) to predict the relationship between object-oriented metrics (independent variable) and software faults (dependent variable) in OO systems. They collected data from one java application that contains 136 classes. They reported that these 36 metrics cover only six dimensions, five of which are related to coupling and one is related to inheritance. They used principle component analysis (PCA) to find these dimensions, Univariate, and multivariate logistic regression to test the hypothesis against these dimensions and find out the relationship between dependent and independent variables. They also considered import coupling, which is called efferent coupling. They reported that coupling (CBO, CBO1) is a good indicator for fault proneness, cohesion (LCOM1) having a positive coefficient factor, which means that the probability of fault proneness increases as the cohesion of a class decreases. Inheritance metrics like DIT and NOC are not good indicators for fault proneness. The proposed model was able to predict faulty classes with more than 80% accuracy.

Catal and Diri [2] built a model based on Artificial Immune Recognition System (AIRS) algorithm that is used to predict the faults using product level and class level metrics. They used C&K metrics suite and other method-level metrics such as Halstead and McCabe metrics to predict the fault-proneness in OO systems. They used NASA data sets as an input to construct fault prediction model. They identified best and most significant combination of OO metrics for fault prediction. They reported that all the C&K and other metrics have better results for fault prediction in AIRS models except DIT (depth of inheritance) that has not good results when performance indicator threshold level is 0.5. Their reported results are similar to the results reported in [7], which also identified significant metrics for faults prediction by using univariate logistic regression (ULR). Therefore, we can observe that CBO is more significant for fault prediction.

Yuming and Leung [7] used C&K metrics suite and SLOC to predict low and high severity faults. They applied these metrics on KCI NASA datasets. To predict fault-proneness in OO systems, they used logistic regression random factors, Naïve Bayes and nearest neighbor with generalization techniques. They used correctness, completeness and precision parameters for metrics as evaluation parameters. After applying these metrics and evaluation parameters, they reported that CBO has better results for predicting high severity faults than NOC and DIT. They also reported that other metrics form C&K suite is also useful for fault prediction.

Gyimothy et al. [3] used logistic regression decision trees, neural networks, and linear regression to validate object-oriented metrics for fault prediction. They applied these metrics on open source system Mozilla. They used only class level metrics and used completeness to evaluate performance. Their study analysis showed that CBO has better results for fault prediction, while DIT and NOC are not recommended for fault prediction. They reported that four methods, which are mentioned above, have similar results. They also suggested using multivariate models instead of lines of code.

In summary, Researchers used object-oriented metrics with different techniques for faults prediction. Most of the studies showed that coupling between objects is a good indicator for fault prediction. However, CBO is the very generic view of coupling, thus, we cannot differentiate whether one class is affecting the whole system (one class is coupled with all other classes) or all classes are coupled with each other. This paper reported results based of different perspectives of coupling.

IV. Research Methodology and Data Description

In this section, we present the data description that is used in this experiment and the research method to investigate the effect of selected coupling metrics (CBO, Ca, and Ce) on fault prediction. In this work, we investigate the effect of CBO, Ca, and Ce with respect to faults at the class level. The motivation behind this study is to know which type of coupling leads to a better fault prediction.

A. Data Description

This section describes the data that is used in this study. For this study, we collected data from PROMISE repository [9]. To perform this experiment, the selected projects are implemented in java, as it is a widely used language and it also fulfills most of the requirements for metrics measurement. We collected data from seven systems from different domains. The seven systems that we used for this study and related information shown in TABLE 1:

TABLE 1 Descriptive Statistics

Sr.#	Project	Version	No. of Classes
1	Jedit	4.3	492
2	Camel	1.2	608
3	Ant	1.3	125
4	Arc	1	234
5	Velocity	1.4	196
6	Xerces	1.4.4	588
7	Poi	1.5	237

The first system is a text editor for more than 200 hundred languages (JEdit 4.3). The second system is an application that facilitates routing with in the application, bean binding, and unit testing (camel 1.2) for java applications. The third system is related to object-oriented programming paradigm (ant 1.3). The fourth system is a search engine based on the OAI-PMH protocol (arc 1.0). The fifth system is a Java-based template engine that allows anyone to reference objects defined in Java code (Velocity 1.4). The sixth system is a library used for XML documents parsing, validating and manipulating (Xerces 1.4). And finally, the seventh system is the Apache POI Project's mission is to create and maintain Java APIs for manipulating various file formats based upon the Office Open XML standards (OOXML) and Microsoft's OLE 2 Compound Document format (OLE2) (poi 1.5).

B. Methodology

This section presents the methodology adopted to perform this study, to perform the statistical analysis we used SPSS tool version 21. SPSS statistics is an integrated family of products that addresses the entire analytical process from planning to data collection to analysis, reporting, and deployment [10]. We performed statistical analysis on the following hypothesis, which is our main focus in the study. First, we will test the null hypothesis if it is rejected, then the alternative hypothesis will be tested. Otherwise, the null hypothesis will be accepted and the alternative hypothesis will be rejected.

- *Null Hypothesis:* There is no difference between CBO, Afferent coupling and Efferent coupling in fault prediction.
- *Alternative Hypothesis 1:* CBO has more impact on fault prediction than afferent and efferent coupling.
- *Alternative Hypothesis 2:* Afferent coupling has more impact on fault prediction than CBO and efferent coupling.
- *Alternative Hypothesis 3:* Efferent coupling has more impact on fault prediction than afferent coupling and CBO.

Hypothesis acceptance or rejection process consists of two steps, in the first step we have to check the confidence factor (p-value) if it is less than 5% of all independent variables in at least one system then the null hypothesis will be rejected, as it predicts that there is a correlation between independent and dependent variables. On the other hand, if the p-value is greater than 5% against in any independent variable then the alternative hypothesis will be rejected against that independent variable and will perform the correlation for remaining independent variables. In the second step, we will measure the correlation value between independent and dependent variables involved in our study, for example between Ca and faults. We will perform this measurement process for all of the three independent variables on selected projects and report the results. For this study, we will use Spearman correlation.

We used these projects from different areas like programming, business, search engine and editor so that minimize the threats to external validity.

V. Results and Discussion

This section discusses the experiment results. We performed a correlation to conclude whether each individual class level metric (CBO, Ca and Ce) is significantly related to the number of faults at class level in the selected systems. In order to perform the correlation between the dependent variable (faults) and independent variables (coupling metrics CBO, Ca and Ce) we used Spearman's rank correlation due to the absence of normal distribution in the data. To check the correlation significance, we set the confidence level 95% (i.e. P-level ≤ 0.05).

TABLE 2 shows the results of spearman correlation coefficient and P-Value for all independent variables against each system; where the highest statistically significant correlations are highlighted by bold values and the confidence value that fulfill the confidence level criteria are highlighted by bold. In order to identify the correlation between coupling metrics and faults, first we have to check the confidence factor (P-value) if the confidence factor is less than the confidence level, then we consider that metric for correlation analysis. As in TABLE 2 confidence factor of Ca metric for two systems (JEdit and poi) is greater than the confidence level so we did not consider this metric for correlation process.

TABLE 2: Bivariate Correlation of All Systems

Metrics/ projects	**Ce**		**Ca**		**CBO**	
	Correlation Coefficient	*P-Value*	*Correlation Coefficient*	*P-Value*	*Correlation Coefficient*	*P-Value*
Jedit	**.127**	**0.005**	0.043	0.338	0.082	**0.05**
Camel	.137	**0.01**	0.172	**<0.001**	**0.176**	**<0.001**
Arc	**.324**	**<0.001**	-0.128	**0.05**	0.283	**<0.001**
Ant	**.423**	**<0.001**	0.122	**0.023**	0.378	**<0.001**
Velocity	.154	**0.031**	0.326	**<0.001**	**0.244**	**0.001**
Xerces	**.652**	**<0.001**	0.395	**<0.001**	0.613	**<0.001**
Poi	**.255**	**0.0324**	-0.102	0.117	0.064	**<0.001**

For other two metrics confidence factor is less than 0.05, it shows that there is a correlation between coupling metrics (CBO and Ce) and faults. So, the Null hypothesis is rejected.

From the alternative hypothesis, we will not consider the alternative hypothesis 2 (Ca) as for Ca measurement does not fulfill the confidence factor condition. From the remaining two alternative hypotheses; CBO and Ce, results show that in five out of seven systems Efferent coupling is more correlated with the number of faults than CBO and afferent Coupling while in one system CBO has better value than Ce. Therefore, we accept the alternative hypothesis 3 (Ce) and reject the first hypothesis (CBO). Another observation is that all metrics are positively correlated with faults except two systems (Ca for Arc and poi) may be this is due to the nature of the system their classes have a high reusability or due to the size of the system. Another observation is CBO has greater correlation coefficient than Ca except velocity where Ca is better than CBO, so we can conclude that CBO is better than Ca for fault prediction. Although the correlation coefficient is not significant as all values are less than 0.05 but the results show that at least Ce gives a better prediction of faults than CBO.

A. Multivariate Linear Logistic Regression

In order to construct different regression models for predicting faults, we performed multivariate linear regression analysis; we constructed models for these two scenarios: Equation (1) predict the number of faults in each system of our study while “(2)” predicting the number of faults in all systems as a whole.

The general regression equation of a multivariate linear regression model with m independent (input) variables can be described as:

$$a_1 = b_0 + b_1 x_{n1} + \ b_2 b_{n2} + \ldots + b_m b_{nm} \tag{1}$$

Here a_1 is a dependent variable (faults) and $x_{n1} \ldots\ldots x_{nm}$ are independent variables, b_0 is constant and $b_1 \ldots\ldots b_m$ are independent variable coefficients in “(1)”.

There might be a case that the prediction model contains redundant information against independent variables. One more thing that needs to be considered is that there might be a high correlation between independent variables, which could have a bad effect on the results and it does not add valuable information to our prediction model. Thus, it is logical to ignore the independent variables that have redundant information and to remove the collinearity from each model. In order to achieve this, stepwise selection process under a 95% confidence level was performed. The results show that CBO and Ce are strongly correlated with each other in all projects. Thus, we can ignore one of them.

Multivariate linear correlation is also performed to get the prediction models. The regression coefficient of each independent variable for both prediction models for all systems is shown in TABLE 3. A table cell with “- “shows that the related metric is not considered for selected independent variables in the corresponding model by using stepwise analysis (ignore parameters that do not full fill p-value condition).

Regression equation for this study is as follows,

$$\text{Number of faults} = 0.495 + 0.11\ (\text{Ce}) \tag{2}$$

TABLE 3 Multivariate correlation of all systems

Multivariate Linear Logistic regression analysis				
	Ce	**Ca**	**CBO**	**Constant**
Jedit	.004	-	-	-.001
Camel	.485	.421	-.416	.215
Arc	.022	-	-	.037
Ant	.234	.166	-.161	.002
Velocity	-	-	.014	.924
Xerces	1.444	.927	-.889	.052
Poi	.121	-	-	.923
For all system				
Combined	.11	-	-	.495

VI. THREATS TO VALIDITY

This section presents some limitations that can affect these results.

A. Construct validity

CBO, Ce, and Ca are not measuring the coupling in both directions; also, they are not considering their indirect method calling in measuring the value of the coupling.

B. *Internal validity*

The experiment was conducted on different systems and we found that Ce is the best metric among the metrics. However, the result might be different if we conducted the experiment on systems with the smaller size.

C. *External validity*

All the systems that are used in the experiment are open-source systems that developed by Java, so the results may be different when applying on other languages than Java or the systems are not open source.

VII. CONCLUSION AND FUTURE WORK

In this paper, we conduct an experiment to investigate the effect of the coupling on the software faults. CBO, Ca, and Ce coupling metrics were selected as the independent variable of this study on seven different size open-source java systems (JEdit, Camel, Arc, Ant, Velocity, Xerces, and Poi). Spearman correlation analysis was applied to investigate the relationship between these metrics and defects. The results show that the Ce has the best correlation with the defects among the selected metrics. Ce metric was the best significantly correlated with six systems, and the CBO for one system. Ce has the highest correlation in the Xerces system.

Probabilistic model instead of the discrete model and other coupling metrics with more open-source systems are considered as a future work.

REFERENCES

[1] K. K. Aggarwal, Y. Singh, A. Kaur, and R. Malhotra, "Empirical analysis for investigating the effect of object-oriented metrics on fault proneness: a replicated case study," *Software Process: Improvement and Practice,* vol. 14, pp. 39-62, 2013.

[2] C. Catal and B. Diri, "Software Fault Prediction with Object-Oriented Metrics Based Artificial Immune Recognition System," in *Product-Focused Software Process Improvement*. vol. 4589, J. Münch and P. Abrahamsson, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 300-314.

[3] T. Gyimothy, R. Ferenc, and I. Siket, "Empirical validation of object-oriented metrics on open source software for fault prediction," *IEEE Transactions on Software Engineering,* vol. 31, pp. 897-910, 2012.

[4] H. M. Olague, L. H. Etzkorn, S. Gholston, and S. Quattlebaum, "Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes," *Software Engineering, IEEE Transactions on,* vol. 33, pp. 402-419, 2007.

[5] G. J. Pai and J. Bechta Dugan, "Empirical Analysis of Software Fault Content and Fault Proneness Using Bayesian Methods," *Software Engineering, IEEE Transactions on,* vol. 33, pp. 675-686, 2007.

[6] Y. Singh, A. Kaur, and R. Malhotra, "Empirical validation of object-oriented metrics for predicting fault proneness models," *Software Quality Journal,* vol. 18, pp. 3-35, 2010/03/01 2010.

[7] Z. Yuming and H. Leung, "Empirical Analysis of Object-Oriented Design Metrics for Predicting High and Low Severity Faults," *Software Engineering, IEEE Transactions on,* vol. 32, pp. 771-789, 2006.

[8] "http://agile.csc.ncsu.edu/SEMaterials/OOMetrics.htm."

[9] T. Menzies., B. Caglayan., E. Kocaguneli., J. Krall., F. Peters., and B. Turhan., "The PROMISE Repository of empirical software engineering data http://promisedata.googlecode.com, West Virginia University, Department of Computer Science," 2012.

[10] "http://www-01.ibm.com/software/analytics/spss/products/statistics/."