Empirical Exploration for the Correlation between Class Object-Oriented Connectivity-Based Cohesion and Coupling

Jehad Al Dallal

***Abstract*—**Attributes and methods are the basic contents of an object-oriented class. The connectivity among these class members and the relationship between the class and other classes play an important role in determining the quality of an object-oriented system. Class cohesion evaluates the degree of relatedness of class attributes and methods, whereas class coupling refers to the degree to which a class is related to other classes. Researchers have proposed several class cohesion and class coupling measures. However, the correlation between class coupling and class cohesion measures has not been thoroughly studied. In this paper, using classes of three open-source Java systems, we empirically investigate the correlation between several measures of connectivity-based class cohesion and coupling. Four connectivity-based cohesion measures and eight coupling measures are considered in the empirical study. The empirical study results show that class connectivity-based cohesion and coupling internal quality attributes are inversely correlated. The strength of the correlation depends highly on the cohesion and coupling measurement approaches.

***Keywords*—**Object-oriented class, software quality, class cohesion measure, class coupling measure.

I. Introduction

An important goal of software engineering is developing the techniques and the tools needed to develop high-quality applications that are more stable and maintainable. Developers and managers use several measures to quantify and enhance the quality of an application during the development process. These measures estimate the quality of different software attributes, such as cohesion, coupling, and complexity.

The cohesion of a module refers to the relatedness of the module components. A module that has high cohesion performs one basic function and cannot be split into separate modules easily. Highly cohesive modules are more understandable, modifiable, and maintainable [1]. The coupling of a module refers to the degree to which a module is related to other modules [2].

Since the last decade, object-oriented programming languages, such as C++ and Java, have become widely used in both the software industry and research fields. In an object-oriented paradigm, classes are the basic modules. The members of a class are its attributes and methods. Therefore, class cohesion refers to the relatedness of the class members, and class coupling refers to the degree to which a class is related to other classes.

Researchers have introduced several measures to indicate class cohesion during high- or low-level design phases. These measures follow different cohesion measuring approaches and use different formulas. For example, some of the measures are based on counting the number of pairs of methods that share common attributes (e.g., [3]). Some others are more precise and they are based on measuring the similarity between each pair of methods in terms of the ratio of the shared common attributes (e.g., [4]). Other measures consider the connectivity pattern of a graph that represents the cohesive relations between methods and attributes in a class. In this case, the cohesion is measured as the connectivity degree of the graph. In this paper, we consider four connectivity-based class cohesion measures: CBMC [5], ICBMC [6], OL_n [7], and PCCC [16]. The measures are widely theoretically (e.g., [8]-[11]) and empirically (e.g., [12]-[26]) studied.

Researchers have proposed several measures to indicate class coupling. In this paper, we considered eight coupling measures: CBO [3], CBO_U [25], CBO_IUB [25], RFC [3], MPC [26], DAC1 [26], DAC2 [26], and OCMEC [27]. The selected measures consider different coupling aspects such as coupling due to method invocations, method parameters, and attribute types.

Understanding the relation between internal quality attributes helps in explaining some practical issues and results regarding the relation between internal and external quality attributes. In addition, the understanding of this relation helps software engineers and practitioners to determine the quality factors that have to be considered and focused on during the quality assessment process. In this paper, we report a correlation study between the two key internal quality attributes of cohesion and coupling. The study involves three Java open-source systems and 12 class cohesion and coupling measures. The results of the study show that the selected cohesion measures are negatively correlated to the selected coupling measures, which gives an indication that class cohesion and coupling are inversely correlated.

This paper is organized as follows. Section II provides an overview of the class cohesion and class coupling measures proposed in literature. Section III describes the considered systems and the data collection process. Section IV reports the correlation study analysis and discusses its results. Finally, Section V concludes the paper and discusses future work.

The author would like to acknowledge the support of this work by Kuwait University Research Grant QI02/13.

Jehad Al Dallal is with Department of Information Sciences, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait (e-mail: j.aldallal@ku.edu.kw).

II. Related Work

Software quality attributes are classified into internal and external. Internal quality attributes, such as cohesion and coupling, are those that can be measured using only the knowledge of the software artifacts such as the source code, whereas the external quality attributes, such as testability and maintainability, are those that require the knowledge of other factors, such as the environment, to be measured. Researchers have proposed several class cohesion and coupling measures in the literature. Class cohesion measures can be applicable based on high-level design (HLD) or low-level design (LLD) information. HLD class cohesion measures rely on information related to class and method interfaces. The more numerous LLD class cohesion measures require an analysis of the algorithms used in the class methods (or the code itself if available) or access to highly precise method postconditions.

Several class cohesion measures considered the connectivity patterns for the graph that represent the relationship between the methods and attributes in a class. In this paper, we consider the three measures defined in Table I. Related work in the area of software cohesion can be found in [12]-[26].

Researchers have considered several coupling measures to evaluate class coupling (e.g., [3], [27]-[29]). In this paper, we use eight coupling measures as follows. CBO_IUB [27] counts the number of classes, excluding the inherited classes that use the attributes or methods of a class of interest. CBO_U [27] counts the number of classes, excluding the inherited classes that are used by the methods of the class of interest. Coupling between object classes (CBO) [3] can be calculated as CBO_U+CBO_IUB. Response for a class (RFC) [3] is measured by counting the number of methods in the class of interest and the number of distinct methods of the other classes directly invoked by the methods of the class of interest. Message passing coupling (MPC) [28] is measured by counting the number of method invocations in the class of interest. Data abstraction coupling (DAC1) [28] counts the number of attributes in a class of interest whose types are of other classes, whereas DAC2 [28] counts the number of distinct classes used as types of the attributes of the class of interest. Finally, OCMEC [29] counts the number of distinct classes used as types of the parameters of the methods in the class of interest.

III. Selected Systems

We chose three Java open-source software systems from three different domains: Art of Illusion v.2.4.1 [30], JabRef v.1.8 [31], and FreeMind v.0.8.0 [32]. Art of Illusion consists of 430 concrete classes (not abstract classes or interfaces) and about 72 thousand lines of code (KLOC), and is a 3D modeling, rendering, and animation studio system. JabRef consists of 306 concrete classes and about 41 KLOC and is a graphical application for managing bibliographical databases. FreeMind consists of 363 concrete classes and about 64 KLOC and is a hierarchical editing system. We chose these three open-source systems randomly from http://sourceforge.net. We developed our own Java tool to automate the cohesion and coupling measurement process for Java classes using the five selected cohesion measures and the eight selected coupling measures. The tool analyzed the Java source code, extracted the information required to build the models that represent the cohesive and coupling interactions, and calculated the cohesion and coupling values using the 13 measures. Tables II and III show descriptive statistics for each of the selected cohesion and coupling measures including the minimum, 25% quartile, mean, median, 75% quartile, maximum value, and standard deviation.

TABLE I
Definitions of the Three Connectivity-Based Class Cohesion Measures

Measure	Definition
Cohesion Based on Member Connectivity (CBMC) [5]	CBMC(G)= F_c(G)×F_s(G), where F_c(G)=|M(G)|/|N(G)|, M(G)=the number of glue methods in graph G, N(G)=the number of non-special methods in graph G, $F_s(G) = [\sum_{i=1}^{n} CBMC(G^i)]/n$, n=the number of child nodes of G, and glue methods is the minimum set of methods for which their removal causes the class-representative graph to become disjointed.
Improved Cohesion Based on Member Connectivity (ICBMC) [6]	ICBMC(G)= F_c(G)×F_s(G), where F_c(G)=|M(G)|/|N(G)|, M(G)=the number of edges in the cut set of G, N(G)=the number of non-special methods represented in graph G multiplied by the number of attributes, $F_s(G) = [\sum_{i=1}^{2} ICBMC(G^i)]/2$, and cut set is the minimum set of edges such that their removal causes the graph to become disjointed.
OL_n [7]	OL_n= The average strength of the attributes, wherein the strength of an attribute is the average strength of the methods that reference that attribute. The strength of a method is initially set to 1 and is computed, in each iteration, as the average strength of the attributes that it references, where *n* is the number of iterations that are used to compute OL.
Path Connectivity Class Cohesion (PCCC) [16]	$PCCC(C) = \begin{cases} 0 & \text{if } l = 0 \text{ and } k > 1, \\ 1 & \text{if } l > 0 \text{ and } k = 0, \\ \frac{NSP(G_c)}{NSP(FG_c)} & \text{otherwise.} \end{cases}$ where NSP is the number of simple paths in graph G_c, FG_c is the corresponding fully connected graph, and a simple path is a path in which each node occurs once at most.

TABLE II
Descriptive Statistics for the Cohesion Measures

Measure	Min	Max	25%	Med	75%	Mean	Std. Dev.
CBMC	0	1	0	0	1	0.286	0.439
ICBMC	0	1	0	0	1	0.274	0.438
OL	0	1	0	0	1	0.286	0.439
PCCC	0	1	0	0.019	1	0.375	0.463

TABLE III
Descriptive Statistics for the Coupling Measures

Measure	Min	Max	25%	Med	75%	Mean	Std. Dev.
MPC	0	1739	9.000	22	60	53.651	102.631
RFC	0	413	7.000	12	31	23.489	29.264
DAC	0	131	1.000	2	4	3.676	6.855
DAC2	0	22	1.000	2	3	2.440	2.849
OCMEC	0	22	2.000	3	5	3.601	3.050
CBO_U	0	58	1.000	2	5	3.980	4.797
CBO_IUB	0	287	0.000	1	2	4.446	16.195
CBO	0	298	2.000	4	8	8.426	17.658

IV. Correlation Study Analysis and Results

We calculated the nonparametric Spearman correlation coefficient [33] between the considered cohesion and coupling measures. Table IV shows the resulting correlations among the considered measures accounting for all three systems. All results were found statistically significant (p-value < 0.0001).

The results reported in Table IV lead to the following observations:

1. Coupling is always negatively correlated to cohesion. This observation is indicated by the negative signs of the results. Therefore, the results indicate that coupling and cohesion are negatively correlated. That is, increasing the cohesion level causes coupling to decrease and vice versa. This observation is expected because a class of high cohesion level is more independent and, consequently, less coupled on other classes and vice versa.
2. The correlations between PCCC and most of the considered coupling measures are higher than those of other considered cohesion measures. This result is expected because it matches the results found in [16], which show that PCCC quantify cohesion more precisely than the other connectivity based cohesion measures. This result gives an indication that the more precise is the cohesion measure, the more are the values of correlations it has with coupling measures. In addition, the results give an evidence that the cohesion measuring approach has a great impact on the correlation results.
3. The coupling measures that consider the number of methods (MPC) and number of method invocations (RFC) were found to feature the highest correlations to connectivity-based cohesion measures. On the other side, the measures that account for the access of class members among different classes (i.e., CBO and its two extensions, CBO_IUB and CBO_U) exhibited the weakest correlations to connectivity-based cohesion measures.

V. Conclusions and Future Work

This paper investigates the correlation between class cohesion, measured using four existing connectivity-based cohesion measures and class coupling, measured using eight exiting coupling measures. The selected measures consider different cohesion and coupling measuring approaches and factors. Classes of three Java open source systems were involved in the study. Generally, the results confirm the expectation that cohesion and coupling are inversely correlated. In addition, the results indicate that the correlation between coupling and cohesion measures depend highly on the approach followed and source code artifacts used in cohesion and coupling measurement.

In the future, we plan to extend the study to involve more cohesion and coupling measures and more studied systems. In addition, we intend to empirically investigate the correlation among other internal quality attributes such as inheritance and complexity and between cohesion and coupling and the other quality attributes.

TABLE IV
Correlation Analysis Results

Measure	CBMC	ICBMC	OL	PCCC
MPC	-0.486	-0.485	-0.486	-0.572
RFC	-0.474	-0.474	-0.474	-0.552
DAC1	-0.401	-0.401	-0.401	-0.372
DAC2	-0.407	-0.407	-0.407	-0.370
OCMEC	-0.453	-0.452	-0.453	-0.445
CBO_U	-0.329	-0.329	-0.329	-0.353
CBO_IUB	-0.141	-0.141	-0.141	-0.199
CBO	-0.309	-0.309	-0.309	-0.368

References

[1] L. C. Briand, C. Bunse, and J. Daly, A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs, IEEE Transactions on Software Engineering, 27(6), 2001, pp. 513-530.
[2] L. C. Briand, J. Daly, and J. Wust, A unified framework for coupling measurement in object-oriented systems, IEEE Transactions on Software Engineering, 25(1), 1999, pp. 91-121.
[3] S.R. Chidamber and C.F. Kemerer, Towards a Measures Suite for Object-Oriented Design, Object-Oriented Programming Systems, Languages and Applications (OOPSLA), Special Issue of SIGPLAN Notices, Vol. 26, No. 10, 1991, pp. 197-211.
[4] J. Al Dallal and L. Briand, A precise method-method interaction-based cohesion measure for object-oriented classes, ACM Transactions on Software Engineering and Methodology (TOSEM), 2012, Vol. 21, No. 2, pp. 8:1-8:34.
[5] H. Chae, H.S., Kwon, Y. R., and Bae, D. A cohesion measure for object-oriented classes, Software—Practice & Experience, 30(12), 2000, pp.1405-1431.
[6] B. Xu and Y. Zhou, Comments on 'A cohesion measure for object-oriented classes' by H. S. Chae, Y. R. Kwon and D. H. Bae (Softw. Pract. Exper. 2000, 30: 1405-1431), Software—Practice & Experience, Vol. 31, No. 14, 2001, pp. 1381-1388.
[7] X. Yang, Research on Class Cohesion Measures, M.S. Thesis, Department of Computer Science and Engineering, Southeast University, 2002.
[8] J. Al Dallal, Mathematical validation of object-oriented class cohesion measures, International Journal of Computers, 2010, 4(2), pp. 45-52.
[9] L. C. Briand, J. Daly, and J. Wuest, A unified framework for cohesion measurement in object-oriented systems, Empirical Software Engineering - An International Journal, Vol. 3, No. 1, 1998, pp. 65-117.
[10] J. Al Dallal, Qualitative analysis for the impact of accounting for special methods in object-oriented class cohesion measurement, Journal of Software, 2013, Vol. 8. No. 2, pp. 327-336.
[11] J. Al Dallal, Theoretical analysis for the impact of including special methods in lack-of-cohesion computation, Procedia Computer Science, Volume 1, 2012, Pages 167–171.
[12] J. Al Dallal, The impact of inheritance on the internal quality attributes of Java classes, Kuwait Journal of Science and Engineering, 2012, Vol. 39, No. 2A, pp. 131-154.
[13] J. Al Dallal, Constructing models for predicting extract subclass refactoring opportunities using object-oriented quality measures, Information and Software Technology, 2012. Vol. 54, No. 10, pp. 1125-1141.
[14] J. Al Dallal, Incorporating transitive relations in low-level design-based class cohesion measurement, Software: Practice and Experience, 2013, Vol. 43. No. 6, pp. 685-704.
[15] J. Al Dallal, The impact of accounting for special methods in the measurement of object-oriented class cohesion on refactoring and fault prediction activities, Journal of Systems and Software, 2012, Vol. 85, No. 5, pp. 1042-1057.
[16] J. Al Dallal, Fault prediction and the discriminative powers of connectivity-based object-oriented class cohesion measures, Information and Software Technology, 2012, Vol. 54, No. 4, pp. 396-416.
[17] J. Al Dallal, Transitive-based object-oriented lack-of-cohesion measure, Procedia Computer Science, Volume 3, 2011, pp. 1581-1587.
[18] J. Al Dallal, Measuring the discriminative power of object-oriented class cohesion measures, IEEE Transactions on Software Engineering, 2011, Vol. 37, No. 6, pp. 788-804.

[19] J. Al Dallal, Improving the applicability of object-oriented class cohesion measures, Information and Software Technology, 2011, Vol. 53, No. 9, pp. 914-928.
[20] J. Al Dallal and L. Briand, An object-oriented high-level design-based class cohesion measure, Information and Software Technology, 2010, 52(12), pp. 1346-1361.
[21] J. Al Dallal, Software similarity-based functional cohesion measure, IET Software, 2009, Vol. 3, No. 1, pp. 46-57.
[22] J. Al Dallal and S. Morasca, Predicting object-oriented class reuse-proneness using internal quality attributes, Empirical Software Engineering, 2014, Vol. 19, No. 4, pp. 775-821.
[23] J. Al Dallal, The effects of incorporating special methods into cohesion measurement on class instantiation reuse-proneness prediction, IET Software 2014, Vol. 8, No. 6, pp. 285-295.
[24] J. Al Dallal, Object-oriented class maintainability prediction using internal quality attributes, Information and Software Technology, 2013, Vol. 55, No. 11, pp. 2028-2048.
[25] 4. J. Al Dallal, Assessing the discriminative power of object-oriented cohesion measures in practice, AWERProcedia Information Technology and Computer Science, 2012, Vol. 1, pp. 447-452.
[26] J. Al Dallal and S. Morasca, Investigating the Impact of Fault Data Completeness over Time on Predicting Class Fault-Proneness, submitted for publication in IEEE Transactions on Software Engineering, 2015.
[27] H. Kabaili, R. Keller, and F. Lustman, Class cohesion as predictor of changeability: an empirical study, L'Objet, Hermes Science Publications, 2001, 7(4), pp. 515-534.
[28] W. Li and S.M. Henry, Object-oriented measures that predict maintainability, Journal of Systems and Software, 1993, 23(2), pp. 111-122.
[29] L. Briand , P. Devanbu, and W. Melo, An investigation into coupling measures for C++, Proceedings of the 19th International Conference on Software Engineering, Boston, Massachusetts, United States, 1997, p.412-421.
[30] Illusion, http://sourceforge.net/projects/aoi/, November 2012.
[31] JabRef, http://sourceforge.net/projects/jabref/, November 2012.
[32] FreeMind, http://freemind.sourceforge.net/, November 2012.
[33] S. Siegel and J. Castellan, Nonparametric Statistics for the Behavioral Sciences, McGraw-Hill, 2nd edition, 1988.