Review and Evaluation of Cohesion and Coupling Metrics at Package and Subsystem Level

Shouki A. Ebad^{1*}, Moataz A. Ahmed²

¹ Faculty of Computing and IT, Northern Border University, Saudi Arabia.
 ² Info. & Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia.

* Corresponding author. Tel.: +966 14 6615454; email: shouki.abbad@nbu.edu.sa Manuscript submitted January 4, 2016; accepted March 21, 2016. doi:10.17706/jsw.11.6.598-605

Abstract: Cohesion and coupling metrics at package and subsystem level play a crucial role in guiding software packaging (partitioning) and analyzing the maintainability and reusability of software. There has been a number of attempts to propose frameworks to assess the cohesion and coupling metrics at class level. A little work has been done at a higher level. In this paper, we survey the existing cohesion and coupling metrics at package and subsystem level and present an attribute-based framework to assess these metrics. The framework is meant to guide researchers interested in proposing new metrics at package level. The paper discusses a number of metrics against the framework.

Key words: Cohesion, coupling, package, software metrics.

1. Introduction

In object-oriented (OO) development, the classes of system are packaged in the architectural design phase [1]. The packaging process is classified as a combinatorial optimization problem [1]-[3]. Software metric at package level could be used as the fitness function of this problem [1]-[3]. Defining a fitness function that maximizes the cohesion of the individual packages and minimizing the coupling among all of the packages is a widespread method implemented by software architects [1]-[3]. Cohesion and coupling metrics are important in guiding software packaging [1], [3]. However, it is a little focus on this area compared with that at class level [3]-[5]. Based on eight attributes identified as a result of an intensive survey, this paper evaluates the package cohesion and coupling metrics against the attribute-based framework. Such a framework contributes to an increased understanding of the state-of the-art as it is a mechanism for comparing metrics and their potential use as well as integrating existing metrics which examine the same concepts in different ways. The framework can be used to facilitate more decision making regarding the definition of new metric and the selection of existing metrics for a specific goal [5]-[7]. We discuss eleven representative metrics against the attribute-based framework. This paper contributes to this direction.

2. Cohesion and Coupling Frameworks

The previous frameworks would be surveyed here. Eder *et al.* [8] presented a framework aimed at providing comprehensive criteria for cohesion (method level and class level) and coupling (interaction, components, and inheritance coupling) in OO systems. Hitz and Montazeri [9] introduced a framework for a comprehensive metric for coupling in OO systems on both object level (dynamic interactions) and class

level (static interactions). About cohesion, they presented a graph theoretic improved version of the LCOM metric. Briand *et al.* [6] provided a framework for the comparison, evaluation, and definition of cohesion measures in OO systems. Briand *et al.* [7] presented another framework for coupling measures; the coupling framework is complementary to the cohesion one. The coupling Briand's framework was used by Arisholm [10] to describe how coupling can be defined based on dynamic analysis of OO systems. Ebad and Ahmed [5] described an evaluation framework to cohesion metrics at package level. The work reported in this paper extends that framework. Particularly, the package/subsystem coupling metrics are included. According to [11], coupling as a software design concept is not less important than cohesion if not more.

3. An Evaluation Framework Acquisition

We studied cohesion and coupling metrics at high level available in the literature. Based on our observations on metrics along with some wish items, we were able to identify a set of attributes to analyze different metrics. The attributes, in Table 1, are general enough to be applicable to a wide set of artifacts.

Attribute	Description
Metric Objective	This attribute is expected to determine the external quality attributes such as reusability; or internal
	quality attributes such as size and complexity.
Reliability	The metric is not reliable in case of ambiguity. An ambiguity exist if the metric gives the same value
	for packages that are, intuitively, of different cohesion or coupling.
Domain of measure	It refers to the level of granularity; there are three considered levels: method level (i.e., fine grain),
	class level (i.e., medium grain), and package or subsystem level (i.e., coarse grain) [6][7].
Normalization &	It determines if the cohesion value is between 0 (i.e., least cohesion) and 1 (i.e., perfect cohesion); and
non-negativity	the coupling value is non-negative. Normalization allows us to make a meaningful comparison of the
0	cohesion metrics of package which have different sizes [6]. It is worth noting here that the non-
	negativity attribute of software metrics (among other properties) is proposed in the literature and
	has been widely adopted as a formal property to evaluate software metrics [12]-[15].
Applicability phase	Five general development phases: requirements definition, system design, implementation, testing,
	and operation and maintenance. The attribute specifies the phase that the metric will be applicable.
Applicability to	Because early availability of metrics is very useful, it is expected to use UML diagrams in measuring
UML Diagrams	cohesion and coupling metrics at package level would be used [16].
Weighting	Does the metric treat all connections equally? A metric might consist of two types of connections. For
	a reason experts might suggest the first type should have double the weight of the second
Validation	This attribute determines whether the metric is theoretically or empirically validated
vanuation	This attribute determines whether the metric is theoretically of empirically valuated.

Γable 1: Description of	our Proposed Attributes
-------------------------	-------------------------

4. Cohesion and Coupling Metrics

In chronological order, we present a definition of the surveyed metrics based on our attributes. The list of metrics is not exhaustive, we gave attention to those works we considered significant as regards the subject under discussion. What it takes into account is that the number of metrics to measure cohesion and coupling at the higher levels is a few compared to those have defined up to class level [4], [5], [17], [18].

4.1. Marchesi 1998 [19]

Three coupling metrics were defined: (1) PK_1 , number of dependencies whose clients are classes of a given package P_k and whose servers are outside Pk, (2) PK_2 , number of dependencies on server classes belonging to P_k and is related to the degree of reuse of these classes, and (3) PK_3 , the average of PK_1 ; it is an estimate of overall coupling among packages. A "good" system should have packages with low values of PK_1 and PK_2 .

4.2. Doval et al. 1999 [20]

The intra-connectivity metric of a package was measured as the number of intra-edge dependencies

divided by the maximum number of possible dependencies between the components (i.e., classes). The inter-connectivity metric between two packages was measured as the ratio of inter-dependencies between the two packages and the maximum possible number of inter-edge dependencies between them.

4.3. Vernazza et. al. 2000 [21]

An extension of two CK metrics was presented: (1) External CBO: number of external classes coupled to a package, (2) Component Cohesion (CC), number of internal classes to which a class is coupled normalized with the number of the possible coupling relationship among the classes.

4.4. Martin 2003 [22]

A package cohesion was defined as the average of internal relationships per class in a package as follows:

$$\frac{R+1}{N} \tag{1}$$

where *R* is the number of class relationships i.e., internal to the package, *N* is the number of classes in the package, and the extra 1 prevents the metric value to be zero when N=1.

4.5. Bauer and Trifu 2004 [23]

They defined cohesion and coupling subsystem (package) metric as follows:

$$\operatorname{avgCohesion}(D) = \frac{\sum_{\substack{Si \in D \\ |Si| > 1}} \frac{\operatorname{noInternalEdges}(Si)}{|Si|^2 - |Si|}}{|D|^*}$$
(2)

$$\operatorname{avgCoupling}(D) = \frac{\sum_{\substack{Si,Sj \in D}} \frac{\operatorname{noExternalEdges}(Si,Sj)}{|Si| \times |Sj|}}{\frac{|D|^2 - |D|}{2}}$$
(3)

where *D* is a decomposition, *Si* is the *i*th subsystem in *D*, |Si| is the number of classes in subsystem Si, noInternalEdges (*Si*) is the number of edges between the classes of *Si*, noExternalEdges (*Si*, *Sj*) is the number of edges between classes from *Si* and *Sj*. and $|D|^*$ is the number of not single-class subsystems in *D*.

4.6. Khan's MS-Thesis 2004 [12]

Three metrics are proposed: Inter-Package Coupling (IPC): the total coupling among the packages of the system, Internal Package Coupling (INPC): the total coupling between the classes of the same package, and External Package Coupling (EPC): the total coupling that a package has with all other packages.

4.7. Seng et al. 2005 [24]

The cohesion is defined by the number of classes inside the subsystem known by some class which belong to the same subsystem and divide this by the square of the number of classes in the subsystem. The coupling is calculated by the number of dependency edges between classes inside package and classes belonging to other packages and divide this by the overall number of dependency edges in the system.

4.8. Hussain's MS-Thesis 2005 [4]

He proposed the package interaction cohesion metric as the number of methods in a particular class that have an interaction with the methods of other classes within the same package.

4.9. Abdeen et al. 2009 [25]

Based on the Common Closure Principle by Martin [22], the authors defined the package cohesion quality (CohesionQ) as proportional to the number of internal dependencies within the package |P_{Int.D}| as follows:

$$CohesionQ(P) = 1 - \frac{|P_{Int,D}|}{|P_D|}$$
(4)

For normalization, they used $|P_D|$ which is the number of dependencies of the package with $|P_D|$ >0.

4.10. Ali's MS-Thesis 2010 [26]

Ali aggregated the cohesion metric (which was LCOM) and coupling metric (which was CBO) from class level to package level. After he adopted the LCOM version of Seller [27], Ali redefined it at package level by calculating the average of LCOMs. A similar aggregation was done for CBO metric.

4.11. Gupta and Chhabra 2009, 2012 [17], [18]

The cohesion is the ratio of the number of relations between ordered/unique pairs of package elements and maximum number of relations between ordered and unique pairs of elements. The coupling between two packages is the total number of directed connections between ordered/unique pairs of their elements.

5. Discussion

Based on previous evaluation criteria for coupling and cohesion metrics, we have created our own evaluation criteria, we have surveyed in the literature for relevant coupling and cohesion metrics at package and subsystem level and we have evaluated the quality of those based on our evaluation criteria. This is shown in Table 2 A discussion about reliability of the metrics is given in detail in the next section.

Study	Objective	Reliability	Domain	Normalized	Phase	UML	Weighting
Marchesi	To early estimate of development efforts using package coupling	Ambiguity in PK ₃	Coarse	Yes	Analysis & Dosign	Class diagram with no	Equally
Doval et al.	Trade-off between cohesion & coupling to find	Ambiguity in cohesion	Coarse	Yes	Design	No	Equally
Vernazza <i>et. al</i> .	To measure cohesion and coupling of package	Ambiguity	Coarse	Yes	Coding	No	Equally
Martin	To manage the package structure via quantifying the package cohesion	Ambiguity	Coarse	Yes	Design	No	Equally
Bauer & Trifu	To find an optimal design using high cohesion and low external coupling	Ambiguity	Coarse	Yes	Coding	No	Not Equally
Khan's Thesis	To detect software flaws and to early assess software architecture	Ambiguity	Coarse	No	Design	No	Equally
Seng et al.	Compromise between cohesion & coupling to find the decompositions guality	Ambiguity	Coarse	Yes	Coding	No	Not Equally
Hussain's Thesis	To propose a new cohesion metric at package level	Ambiguity	Coarse	Yes	Design	No	Equally
Abdeen et al.	To quantify the package quality within a modularization	Ambiguity	Coarse	Yes	Design	No	Not Equally
Ali's Thesis	To find change prediction and implementation effort estimation for packages	Ambiguity	Coarse	Yes	Coding	No	Equally
Gupta & Chhabra	To predict reusability and improve package structure	Ambiguity	Coarse	Yes	Coding	No	Equally

Table 2. The Existing Metrics Against the Framework Attributes

5.1. Metric Reliability

Table 2 raises doubts as whether the metrics are reliable. A metric would not be considered reliable if it gives the same value for packages that are, intuitively, of different cohesion/coupling qualities; or vice-versa. [28]. Likewise, a metric would not be considered reliable if it ranks a class as better than another where intuitions suggest the other way around. As an example, two packages P_1 and P_2 , with six classes and six interactions for each; the levels of interactions are different. As in Fig. 1 (a), we observe that P_1 interactions are equally distributed on the classes of the package so that each class interacts with two classes. Intuitively, P_1 is expected to be of different cohesion than P_2 since most interactions in P_2 are done via one class (called C'). The cohesion among the rest of classes in P_2 is weaker than that of C'. Fig. 1 (b) describes coupling case. There are two subsystems: Subsystem1 and Subsystem2 with three packages and three interactions for each; the levels of interactions are different. We observe Subsystem1 has three interactions distributed equally on the packages of the subsystem so that each package interacts with two packages. Intuitively, Subsystem1 is expected to be of different coupling than Subsystem2 because most interactions in Subsystem2 (i.e., 2 out of 3) are done via P_2 and P_3 . It means the coupling of P_1 and P_2 in Subsystem2 is weaker than that between P_2 and P_3 in the same subsystem. This is shown in Table III & IV.



(b)

Fig. 1. (a) Two packages (b) two subsystems. A circle is a class and a directed edge is an interaction.

	P1	P_2	
Study	Fig.1	Fig.1	Comments
	(a)	(a)	
Doval et	1/6	1/6	It is 3/36; the max number of
al			connections is n ²
Vernazza	1/5	1/5	It is $6/(6\times 5)$; the number of
et al			possible coupling relationships
			among the classes is n(n-1)
Martin	7/6	7/6	It is (6+1)/6 where R= 6 & N= 6
Panor &	2/5	2/5	It is 6/15 where (19/2
Datter &	3/5	3/3	$ S_{i} ^{2} = 15$ $ D ^{2} = 2$ and chaster
IIIu			-2 /5
Khan's	6	6	The INPC was considered a
MS	0	0	coupling metric but it applies
110			the idea of package cohesion
Seng et al	1/6	1/6	For P1.
	-/-	-/-	1/36+1/36+1/36+1/36+1/36+
			1/36. For P ₂ ,
			5/36+1/36+0/36+0/36+0/36+
			0/36
Hussain's	1	1	It is 6/6; if every class has 1
MS			method
Abdeen	6/7	6/7	Assume one interaction to an
et al			external class
Ali's MS	1	1	If the LCOM value of each class is
			1, the avg. of LCOM is 1
Gupta &	5/6	5/6	It is 6/(6 *5)
Chhabra			

Table 4.	Reliability	7 Test	Examp	les fo	r Coup	ling

	S1	S2	-
Study	Fig.1	Fig.1	Comments
	(b)	(b)	
Marches	1	1	In case of PK3, it is
i			(1+1+1)/3 for S1; and
			(0+1+2)/3 for S2
Vernazz	2	2	For P2 package in both
a et al			subsystems
Bauer &	N/A	1/4	For S1, it is N/A because of
Trifu			a single-class package. For
			S2, it is
			(0/4+1/4+2/4)/3=0.75
Khan's	3	3	In case of IPC metric; it is a
MS			counting number
Ali's MS	1	1	If the CBO value of each
			class in a package is 1, the
			avg. of CBO in that package
			is 1. The avg. of CBO is
			classified, in this study, as
			coupling metric but it gives
			an idea of the package
			cohesion
Gupta &	3	3	If you take coupling
Chhabra			between P1 and P2 in S1 and
			S2, they will be 0 and 1. In
			fact, there is indirectly
			coupling between both
			packages in S1

This reliability issue is expected for us because there is not a consensus yet on the concepts used in measuring of the metrics in spite of the efforts in this field during the last decade. Some researchers did not focus on the connections but the classes [21], others excluded some connection types like external and undirected connections [17], [18], [20], [22], [25], others again focused on the methods in the class [4].

5.2. More Observations

Besides reliability issue, more important observations of this study can be summarized as follows:

- All presented cohesion metrics are normalized and coupling metrics are non-negative.
- Around half of the metrics work during the design phase like those proposed by Marchesi, Doval *et al.*, Martin, Khan's MS, Hussain's MS, and Abdeen et al. The others work during the implementation phase like those proposed by Vernazza et. al., Bauer and Trifu, Seng *et al.*, Ali's MS, and Gupta and Chhabra.
- Designing cohesive packages means creating packages that offer coarse-grained, yet much focused behaviors. All metrics are on coarse grain level i.e., the measure domain was higher than class level.
- Most metrics suffer from weighing limitation. They treat interactions in the package equally. For example, two packages interacted with each other by one type of message passing are equivalent to two packages interacted with each other by more than one type of message passing.
- UML diagrams at the architectural design phase are not considered in all metrics except in Marchesi's study that used class diagram with excluding inheritance dependency. Recently, the authors filled this gap by proposing a search-based packaging method and metric based on UML sequence diagrams [1].
- Although no metric is perfectly accurate, imperfect measurement is better than none especially that these metrics are being widely used by different institutions and agencies.

6. Conclusion and Future Work

The main contribution of this work is an evaluation of the relevant works related to metrics for cohesion and coupling at package and subsystem level. We present an attribute-based framework to allow assessing package cohesion and coupling metrics. The paper also provides an analysis of a set of metrics in light of the framework. The results provide practitioners with an overview of prominent work in the literature and offer help with regard to making decisions as which package metric would be appropriate for their particular development efforts. This analysis is meant to serve as guide for researchers interested in developing new package metrics. In spite of active research in developing cohesion and coupling metrics at package and subsystem level, studying the reliability of such metrics remains an open issue. As we mentioned earlier, the authors recently presented a new packaging approach based on UML sequence diagrams. As a follow-up to this work, we are working on evaluation of our approach in terms of its impact on external attributes such architecture stability which can be used as indicator of maintainability.

References

- [1] Ebad, S., & Ahmed, M. (2015). Functionality-based software packaging using sequence diagrams. *Software Quality Journal*, *23(3)*, 453-481.
- [2] Clarke, J., Dolado, J., Harman, M., Jones, B., Lumkin, M., Mitchell, B., *et al.* (2003). Reformulating software engineering as a search problem. *IEEE Proc. on Software*, *150* (*3*), 161–175.
- [3] Ebad, S., & Ahmed, M. (2011). Software packaging approaches A comparison framework. *Proceedings of the 5th European Conference. on Software Architecture*, (pp. 438-446). Essen, Germany.
- [4] Hussain, S. (2005). Package cohesion metric for object-oriented systems. MS Thesis, King Fahd University of Petroleum and Minerals, Saudi Arabia.
- [5] Ebad, S., & Ahmed, M. (2011). An evaluation framework for package-level cohesion metrics.

Proceedings of the 2nd International Conference on Future Information Technology (pp. 326-330).

- [6] Briand, L., Daly, J., & Wüst, J. (1997). A unified framework for cohesion measurement in object-oriented systems. *Proceedings of the* 4th *International Software Metrics Symposium*.
- [7] Briand, L., Daly, J., & Wüst, J. (1999). A unified framework for coupling measurement in object-oriented systems. *IEEE Transactions on Software Engineering*, *25(1)*, 91-121.
- [8] Eder, J., Kappel, G., & Schrefl, M. (1994). Coupling and cohesion in object-oriented systems. University of Klagenfurt.
- [9] Hitz, M., & Montazeri, B. (1995). Measuring coupling and cohesion in object-oriented systems. *Proceedings of the International Symposium on Applied Corporate Computing*, Monterrey, Mexico.
- [10] Arisholm, E., Briand, L., & Foyen, A. (2004). Dynamic coupling measurement for object-oriented software. *IEEE Transaction on Software Engineering*, *30(8)*, 491-506.
- [11] Vanderfeesten, I., Reijers, H., & Aalst, W. (2008). Evaluating workflow process designs using cohesion and coupling metrics. *Computers in Industry*, *59*(*5*), 420-437.
- [12] Weyuker, E. J. (1988). Evaluating software complexity measure. *IEEE Transaction on Software Engineering*, *14*(*9*), 1357-1365.
- [13] Zuse, H. (1991). Software Complexity: Measures and Methods. Walter de Gruyter & Co., USA.
- [14] Briand, L., Morasca, S., & Basili, V. (1996). Property-based software engineering measurement. *IEEE Transactions on Software Engineering*, *22(1)*, 68-86.
- [15] Hassan, Y. (2007). Measuring software architectural stability using retrospective analysis. MS thesis, King Fahd University of Petroleum and Minerals, Saudi Arabia.
- [16] Khan, S. (2004). Design level coupling metrics for UML models. MS Thesis, King Fahd University of Petroleum and Minerals, Saudi Arabia.
- [17] Gupta, V., & Chhabra. J. K. (2009). Package coupling measurement in object-oriented software. *Journal of Computer Science and Technology*, *24(2)*, 273-283.
- [18] Gupta, V., & Chhabra, J. K. (2012). Package level cohesion measurement in object-oriented software. *Journal of Brazilian Computer Society*, *18*, 251–266.
- [19] Marchesi, M. (1998). OOA metrics for the unified modeling language. Proceedings of the 2nd Euromicro Conference on Software Maintenance and Reengineering (pp. 67-73).
- [20] Doval, D., Mancoridis, S., & Mitchell, B. (1999). Automatic clustering of software systems using a genetic algorithm. *Proceedings of Software Technology and Engineering Practice*.
- [21] Vernazza, T., Granatella, G., Succi, G., Benedicenti, L., & Mintchev, M. (2000). Defining metrics for software components. *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*.
- [22] Martin, R. (2003). Agile Software Development: Principles, Patterns, and Practices. Prentice-Hall.
- [23] Bauer, M. & Trifu, M. (2004). Architecture-aware adaptive clustering of object-oriented systems. *Proceedings of the 8th European Conference on Software Maintenance and Reengineering.*
- [24] Seng, O., Bauer, M., Biehl, M., & Pache G. (2005). Search-based improvement of subsystem decompositions. *Proceedings of Genetic and Evolutionary Computation Conference*. Washington, DC.
- [25] Abdeen, H., Ducasse, S., Sahraouiy, H., & Alloui I. (2009). Automatic package coupling and cycle minimization. *Proceedings of the 16th Working Conference on Reverse Engineering* (pp. 103-122). France.
- [26] Ali, A. (2010). A comprehensive empirical validation of package level metrics for object-oriented systems. MS Thesis, King Fahd University of Petroleum and Minerals, Saudi Arabia.
- [27] Sellers, B. (1996). Object-oriented metrics-measures of complexity. Prentice Hall.
- [28] Ahmed, M., Abubakar, A., & AlGhamdi, J. (2011). A study on the uncertainty inherent in class cohesion measurements. *Journal of Systems Architecture Embedded Systems Design*, *57(4)*, 474-484.



Shouki A. Ebad received his M.S. in computer science from University of Jordan, Jordan, in 2000 and Ph.D. in computer science and engineering from King Fahd University of Petroleum and Minerals, Saudi Arabia, in 2012. Currently, he is working as an assistant professor at the Faculty of Computing & IT, Northern Border University (NBU) in Saudi Arabia. He also serves as an assistant dean of Technical Affairs at IT Deanship at NBU. Before that, he held several positions: Lecturer, Head of Information Systems department, and Vice-Dean. He is a Sun Certified Programmer for the Java 2 Platform. His current

research interests are search-based software engineering (SBSE), software metrics, software architecture, and requirements engineering. He published a number of articles in these areas.



Moataz A. Ahmed received his PhD in computer science from George Mason University in 1997. Dr. Ahmed is currently a faculty member with the Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Saudi Arabia. He also severs as an adjunct/guest professor in a number of universities in the US and Italy. During his career, he worked as a software architect in several software houses. His research interest includes softcomputing-based software engineering, especially, software testing, software reuse, and cost estimation; and software metrics and quality models. He

has supervised a number of theses and published a number of scientific papers in refereed journals and conferences in these areas.