Indian Journal of Science and Technology, *Vol 9(21), DOI: 10.17485/ijst/2016/v9i21/89912, June 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Coupling Measures and its Impact on Object-Oriented Software Quality

V. S. Bidve* and P. Sarasu

Veltech Dr. RR and Dr. SR Technical University, Avadi, Chennai - 600062, Tamil Nadu, India;
vijay.bidve@gmail.com, sarasujivat@gmail.com

Abstract

Metrics measurement is an important means to access the quality of software systems. Software metrics measurement helps in quick and accurate analysis of software quality. Coupling metrics is a kind of software metrics having a significant impact on several qualities attributes of the object-oriented software. In literature, coupling metrics is used to access the software quality attributes like reliability, efficiency, integrity, maintainability, flexibility, portability, reusability and interoperability, etc. of object-oriented software. However, the threshold values for software coupling measures are still not fully defined. This paper investigates the values of seven coupling metrics and explores the impact of coupling metrics values on different quality attributes of the object-oriented software. The Java source code is analyzed to find coupling between different modules of java code. The coupling values of twenty five Java projects are measured to find behavior of coupling values. Using the resulting coupling values of the twenty five projects, percentage threshold values of seven coupling measures are defined. Also the behavior of coupling metrics is plotted using graph. It is found from the results that the percentage threshold value of parameter coupling is in between 24 to 34. Similarly, the threshold percentage of inheritance coupling is between 3 to 13, global coupling is 0 to 7, data abstraction coupling is 9 to 19, import coupling is 18 to 28, export coupling is 19 to 29 and external coupling is 0 to 5. It means parameter coupling, import and export coupling has higher range of values. Inheritance and data abstraction coupling has a medium range of values and global and external coupling have a least range of values. It is also found that the coupling measures have an impact on various quality attributes of the object-oriented software. If the developer could control the values of few important metrics like parameter, inheritance and data abstraction coupling, it will result in improving the overall quality of object-oriented software. **Applications**: This work is useful for software professionals to maintain level of coupling while developing software. The data set used in this work is totally new and unique.

Keywords: Coupling, Impact, Metrics, Object-Oriented, Quality, Software

1. Introduction

Object-oriented terminology plays an important role in software design and development[1]. One important factor that should be considered during software design is coupling, which measures the dependency level between two modules. Several studies in the literature show that coupling directly impacts on quality attributes of object-oriented software[2–5].

In this paper, an attempt is made to measure coupling amongst modules of Java software and to find its impact on object-oriented software quality attributes. This work have considered seven coupling metrics from the literature and used the repository Souceforge.net as a resource for open source Java projects. Twenty-five projects belonging to five different categories are downloaded for this study. A tool named JCMT is used to measures coupling amongst the modules of Java software. The coupling of twenty-five projects, the average coupling of five categories and overall average coupling of all twenty-five projects is calculated. Using these coupling values the percentage of the threshold range of coupling values is predicted. Finally, the impact of each coupling metrics on various software quality attributes is mentioned.

**Author for correspondence*

The rest of the paper is organized as follows. Section 2 gives methodology used to conduct this study. Section 3 presents results and discussion. Section 4 describes the impact of coupling on object-oriented software quality attributes. Section 5 concludes the paper followed by the appendix in Section 6.

2. Methodology

This section describes our dataset, coupling metrics considered and data collection procedure used to find coupling values of object-oriented software.

2.1 Data Set

The repository used for this work is Sourceforge.net, which has ten different categories of projects. A selection criterion for this repository is applied using as given as next. First, our study only considers the projects developed in Java language. The input required for our research project is a folder containing .Java files. Hence, the projects which are developed in Java and possess folders containing .Java files are only chosen. The Java projects containing .exe or .jar files are excluded, as it was not our requirement. Secondly, out of ten categories of the projects, the five categories including Banking, Communication, E-commerce, Games and HRM are selected. Finally, five projects from each of these five categories are selected. Table 2 gives the five project categories and the names of projects belonging to each category.

2.2 Coupling Metrics Considered

There are various object-oriented coupling metrics described in the literature so far. In this study, the seven metrics as shown in Table 1 have considered. The chosen metrics considers all aspects of coupling and have totally different types of interactions and mechanisms to form the coupling. The coupling metrics considered is a part of our previous study.

2.3 Data Collection Procedure

Coupling is a property of every object-oriented programming language. This study considers only Java projects to calculate coupling. A specific tool named **J**ava **C**oupling **M**easurement **T**ool (JCMT) to measure coupling of amongst the modules of Java project is developed. The folder containing Java source code is given as an input to the JCMT tool which then calculates values of all types of coupling measures described in Table 1.

Table 1. Coupling metrics

Sr. No.	Coupling Metrics	Type of interactions	Coupling Mechanisms	Source
1	Parameter coupling	Method-Method,	The method of one class invokes method/passes parameter/passes message to methods of another class or to make a call to the constructor of another class.	[6]
2	Inheritance coupling	Class-Class	One class is a superclass of another class (Inheritance).	[6, 7, 8, 9]
3	Global coupling	Method-Attribute/ Class-Attribute (Friend)	The method of one class can directly access parts of the internal structure, of another class method (friend). Also to access common, shared, non-local variables of another class.	[6]
4	Data Abstraction Coupling	Class-Method/Class-Attribute	One class is used in the implementation of methods of another class. One class is the domain of the instance variable, the local variable of another class.	[9]
5	Import Coupling	ALL	All type of coupling due to any import mechanism.	[10]
6	Export Coupling	ALL	All type of coupling due to an export mechanism.	[10]
7	External coupling	Sharing of global devices.	Sharing an external device like the printer, HDD, external file by the two classes.	[10]

Table 2. Coupling values obtained from JCMT

Sr. No.	Project Category	Project Name	Number of classes in Project	Parameter Coupling	Inheritance Coupling	Global Coupling	Data Abstraction Coupling	Import Coupling	Export Coupling	External Coupling
1	Banking	barter-0_2_0	18	8	16	0	5	8	8	0
2		cyclos_3.7.3	2443	17403	4617	125	9602	12109	12153	30
3		DTAUS-Java	5	10	0	0	3	10	10	0
4		Online Banking	36	35	61	0	91	35	35	0
5		Psbh	8	3	6	0	1	3	3	0
	Average Coupling of five projects(x)			3491.8	940	25	1940.4	2433	2441.8	6
	Scale Down values between 0 to 50 (x/100): A			34.918	9.4	0.25	19.404	24.33	24.418	0.06
6	Communication	chipchat-1.0-Beta2	13	45	5	11	14	52	52	0
7		clnews-1.3.1	260	1349	321	100	485	1103	1116	12
8		fict-1.0	16	12	22	14	14	26	26	0
9		GpsTracker-v5.0.0	30	30	55	0	18	30	96	0
10		jbittorrentapi-v1.1	43	176	29	18	40	144	156	7
	Average Coupling of five projects(x)			322.4	86.4	28.6	114.2	271	289.2	3.8
	Scale Down values between 0 to 50 (x/10): B			32.24	8.64	2.86	11.42	27.1	28.92	0.38
11	E-commerce	cyclos_3.7.3	2443	17403	4617	125	9602	12109	12153	30
12		facecart_src1.0.3	440	1265	717	157	1242	994	999	8
13		jspShop	99	144	350	4	196	137	190	0
14		pushComm	509	11311	975	1377	2064	12253	17564	36
15		QuickStart-sources	3654	43471	5568	3882	13255	29055	32838	97
	Average Coupling of five projects(x)			14718.8	2445.4	1109	5271.8	10909.6	12748.8	34.2
	Scale Down values between 0 to 50 (x/300): C			49.06	8.15	3.69	17.57	36.36	42.49	0.11
16	Games	connectsquares-2	4	1	5	8	8	9	9	0
17		Java2DPhysicsEngine	6	55	2	4	7	59	59	0
18		java game maker	35	50	41	81	49	120	120	0
19		javaml-0.1.7	235	976	300	29	463	671	677	4
20		Passenger-0.3	199	525	276	190	293	648	648	4
	Average Coupling of five projects(x)			321.4	124.8	62.4	164	301.4	302.6	1.6
	Scale Down values between 0 to 50 (x/10): D			32.14	12.48	6.24	16.4	30.14	30.26	0.16
21	HRM	baraza	95	1186	83	12	332	1004	1004	7
22		contacts and event management	16	14	12	44	14	58	58	0
23		cornelius-2.0-alpha2	495	1873	585	10	972	1453	1453	0
24		hcmone	445	2719	442	0	2068	1573	1573	0
25		OpenHR	141	189	126	0	122	160	173	3
	Average Coupling of five projects(x)			1196.2	249.6	13.2	701.6	849.6	852.2	2
	Scale Down values between 0 to 50 (x/30): E			39.87	8.32	0.44	23.38	28.32	28.40	0.06
	Overall average of A, B, C, D, E			36.72	9.91	3.28	17.43	29.39	30.79	0.15

3. Results and Discussion

The couplings value of an object-oriented software module represents the facts that how the module is dependent on other modules. If the coupling is more, the module is difficult to understand, change or correct itself. A larger number of couples mean the module is more sensitive to changes in other parts of the software[11,12]. Software developers usually try to keep coupling as low as possible. This study calculates coupling of five projects of each five categories mentioned in Section 2A. Table 2 provides the detailed listing of categories of the projects, names of the projects belonging to each category, the total number of classes in each project and total coupling value of all types of coupling.

Table 2 gives coupling values of every project obtained from JCMT tool. The names of each project belonging to all five categories in association with the total number of classes present in the project are mentioned. Total coupling of each type is mentioned by the rows of the table. We have taken project category wise average of each type of coupling to observe the behavior of coupling values with the help of graph. We have scaled down all the category wise coupling averages to fit in the range 0 to 50, which is given in Table 2. The purpose of scaling down the averages is, to plot the graph of all category projects within the same range. Figure 1 shows the graph of average coupling values of all five category projects. Each project category is represented by a different color in the graph.

From the graph shown in figure 1, it is observed that all categories of the projects have similar coupling behavior i.e. parameter, import and export coupling have greater values and global and the external coupling has least values. Where, inheritance and data abstraction coupling have the medium range of values. The overall average of all coupling types by considering scaled down values A to E (shown in Table 2) for each project category is calculated. A graph of overall average coupling values is shown in Figure 2.

From Figure 1 and Figure 2 it is observed that the behavior of coupling metrics values is same in all categories of projects (Figure 1) and it is matching with the overall coupling behavior (Figure 2). The parameter coupling, import coupling and export coupling have greater values and the global coupling, external coupling are with the least values and the inheritance coupling and data abstraction coupling have a medium range of values. In order to maintain complexity level of the project, the software developers have to concentrate on parameter coupling, import coupling and export coupling as they have greater values. The developers have to keep these coupling values as low as possible.

The overall average coupling values (of all 25 projects) for all seven types of couplings is considered and drawn a pie chart; the percentage of each coupling type is as shown in Figure 3. If it is agreed that projects belonging to

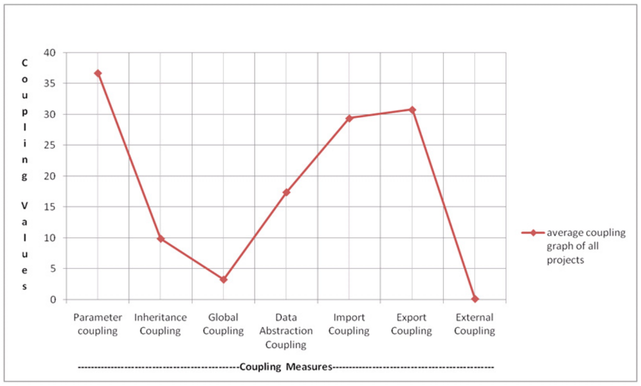

Figure 2. Average coupling graph of all category projects.

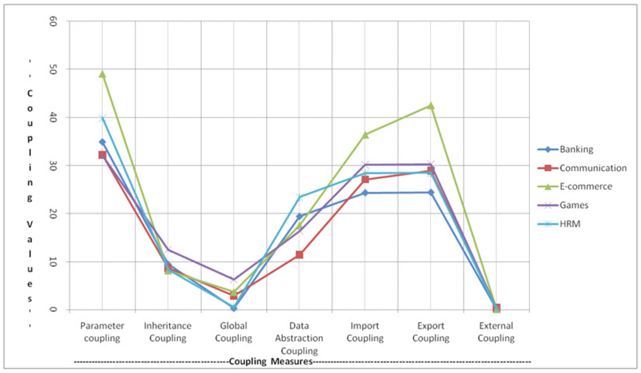

Figure 1. Average coupling graph of all category projects.

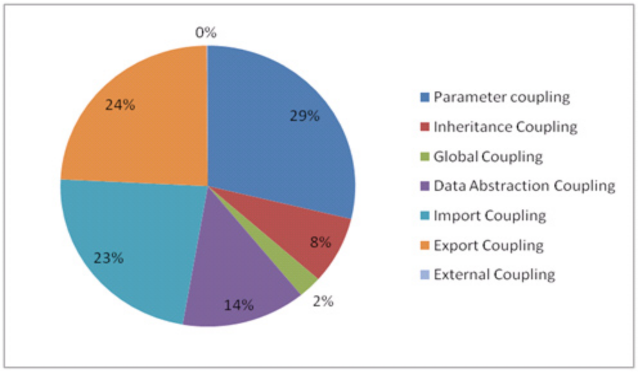

Figure 3. Percentage of each coupling type.

Sourceforge.net follows all standard practices of software engineering and they have designed the projects by considering standard coupling limits for each type of coupling. Then, the threshold values of coupling percentage for each coupling type can be predicted.

Table 3 gives the percentage threshold coupling values for seven coupling types considered in this study. The standard deviation of five percent is considered in the threshold coupling values. The negative threshold percentage value is considered as zero. These threshold values can be a benchmark for software developers.

4. Impact Analysis

In Section 3 calculates seven coupling metric's values for twenty-five projects. It is observed that the impact of these seven coupling metric's values on object-oriented software quality attributes listed in Appendix. The impact of each coupling metric on various quality attributes is given one by one in the following subsection.

4.1 Metric 1: Parameter Coupling

Parameter coupling gives a count that method(s) of one class accesses method(s) of other class(es) due to any means. Our results show that parameter coupling has highest values than other coupling metrics. The higher value of parameter coupling means the methods of one class is accessing multiple methods of other classes. Hence, the greater value of parameter coupling leads to the complex design of object-oriented software. The complex software design always leads to high maintainability/testability and less understandability, flexibility and portability[13]. It means the parameter coupling has the impact on various quality attributes such as complexity, understandability, flexibility and portability.

4.2 Metric 2: Inheritance Coupling

Inheritance coupling is a type of relationship among classes that enables programmers to use previously defines entities including methods and variables. Inheritance decreases class complexity by reducing the number of members of an individual class, but this makes maintenance and design difficult[13]. Inheritance increases efficiency and reusability as it uses existing entities. At the same time, inheritance leads to difficulties in understanding and testing of software. It means inheritance coupling has an impact on multiple quality attributes like complexity, efficiency, reusability, maintainability/testability and understandability.

4.3 Metric 3: Global Coupling

Global coupling occurs when members of one class access non-local variable of another class. More use of variables of other classes leads to increases in complexity of the software. The sharing of variables may increase the efficiency and interoperability but at the same time, it leads to high maintainability. Also, the software understandability and flexibility will be affected. Hence, the global coupling has an impact on complexity, maintainability, understandability, flexibility, efficiency and interoperability.

4.4 Metric 4: Data Abstraction Coupling

Data abstraction is an important type of coupling in which one class uses other classes as a domain or data type. Our research shows that data abstraction coupling metric has a medium range of values. This coupling is very useful to increases reusability and efficiency of the software. But, the use of other classes as a data type may affect the understandability, flexibility, portability. It means

Table 3. Threshold coupling values

Sr. No.	Coupling Type	Actual Coupling % achieved from results of 25 projects	Threshold Coupling % range with 5% standard deviation
1	Parameter coupling	29	24 to 34
2	Inheritance coupling	8	3 to 13
3	Global coupling	2	0 to 7
4	Data Abstraction Coupling	14	9 to 19
5	Import Coupling	23	18 to 28
6	Export Coupling	24	19 to 29
7	Externalcoupling	0	0 to 5

data abstraction coupling has the impact on reusability, efficiency, understandability, flexibility and portability.

4.5 Metric 5: External Coupling

External coupling occurs due to sharing of external devices like printer, HDD by more than one class. Our research shows that external coupling occurs very rarely. External coupling increases interoperability between the modules, but it may affect the portability. It means external coupling has an impact on interoperability and portability of the software.

4.6 Metric 6: Import Coupling

Import coupling does not have a particular type of coupling interaction, but it considers a count of all import coupling interactions due to above five coupling types. Any import of method, attribute and domain by class will be considered under the import coupling. Our research shows that import coupling has higher coupling values. As import coupling considers all type of coupling interactions; it makes an impact on all software quality attributes listed in the appendix.

4.7 Metric 7: Export Coupling

Similar to import coupling, export coupling also considers all type of coupling interactions. It considers a count of all export coupling interaction due to above first five coupling types. Any export of method, attribute and domain of class will be considered under export coupling. As export coupling considers all type of coupling interactions; it makes the impact on all software quality attributes listed in the appendix.

5. Conclusion and Future Work

This study evaluated five Java projects of each five different categories. It is observed that coupling behavior of all five project categories is similar. The overall average of all category projects for each coupling type is calculated. The overall average coupling behavior of all category projects is also similar to the average coupling behavior of each project category. In every category of projects parameter, import and export coupling has a higher range of values. The inheritance and data abstraction coupling have a medium range of values; where the global and external coupling has lower coupling values. The percentage of each coupling metrics in all projects is calculated and predicted the threshold percentage range for each type of coupling metrics. As the second part of the study, the impact of each coupling metric on different quality attributes of the object-oriented software is analyzed.

The study is helpful to software professionals in order to maintain coupling level in the object-oriented software.

In future, the object-oriented software projects of other categories can be investigated to predict the percentage range of coupling metric values more precisely.

6. Appendix

1) **Complexity:** It is a term that encompasses various properties of a piece of software, all of which affect internal interactions.
2) **Efficiency:** The amount of computing resources and code required by a program to perform a function.
3) **Reusability:** Extent to which a program can be used in other applications – related to the packaging and scope of the functions that programs perform.
4) **Maintainability/Testability:** Effort required locating and fixing an error in an operational program/Effort required testing a program to ensure that it performs its intended function.
5) **Understandability:** It is an extent to which the user can comprehend the software.
6) **Flexibility:** Effort required modifying an operational program.
7) **Portability:** Effort required transferring software from one configuration to another.
8) **Interoperability:** Effort required to couple one system with another.

7. References

1. Kayarvizhy N, Kanmani S. Analysis of quality of object oriented systems using object-oriented metrics. 2011 3rd International Conference on Electronics Computer Technology (ICECT); Kanyakumari. 2011 Apr 8-10. p. 203–6.
2. de Souza LBL, de Almeida Maia M. Do software categories impact coupling metrics? Proceedings of the Working Conference on Mining Software Repositories; 2013. p. 217–20.
3. Oliveira MFS, Redin RM, Carro L, da Cunha Lamb L, Wagner FR. Software quality metrics and their impact

on embedded software. 5th International Workshop on Model-based Methodologies for Pervasive and Embedded Software (MOMPES); Budapest. 2008 Apr. p. 68–77.

4. Suresh Y, Pati J, Rath SK. Effectiveness of software metrics for object-oriented system. Procedia Technology. 2012; 6:420–7.
5. Ahmad IA, Rizvi MAK, Al-Lawati A, Al-Abri D, Awadalla M. Design of a probabilistic based software tool for evaluating controllability, observability and testability models of digital systems. Indian Journal of Science and Technology. 2014 Oct; 7(10):1525–37. Doi no: 10.17485/ijst/2014/v7i10/52137.
6. Offutt J, Abdurazik A, Schach S. Quantitatively measuring object-oriented couplings. Springer's Software Quality Journal. 2008 Dec; 16(4):489–517.
7. Eder J, Kappel G, Schrefl M. Coupling and cohesion in object-oriented systems. Technical Report. University of Klagenfurt; 1994.
8. Li H. A novel coupling metric for object – Oriented software systems. IEEE International Symposium on 2011 Knowledge Acquisition and Modeling Workshop; Wuhan. 2008 Dec 21-22. p. 609–12.
9. Husein S, Oxley A. A coupling and cohesion metrics suite for object-oriented software. International Conference on Computer Technology and Development; Kota Kinabalu. 2009 Nov 13-15. p. 421–5.
10. Huang R, Li M, Li Z. Research of improving the quality of the object-oriented system. International Journal of Information and Education Technology. 2013 Aug; 3(4):433–6.
11. Khan FF, Mallika R. Analysis of various types of bugs in the object oriented Java script language coding. Indian Journal of Science and Technology. 2015 Sep; 8(21):1–9. Doi no:10.17485/ijst/2015/v8i21/69958.
12. Kaur J, Singh S. Neural network based refactoring area identification in software system with object oriented metrics. Indian Journal of Science and Technology. 2016 Mar; 9(10):1–8. Doi no: 10.17485/ijst/2016/v9i10/85110.
13. Briand LC, Daly JW, Wust JK. A unified framework for coupling measurement in object-oriented systems. IEEE Transactions on Software Engineering. 1999 Jan-Feb; 25(1):91–121.