

A Comparative Analysis of Web Service Metrics Under Changes



Rachna Kohar
Jawaharlal Nehru University
New Delhi
India
rachnajnu20@gmail.com

ABSTRACT: *Different metrics have been proposed for measuring the properties of a service. In this paper, a comparative study of the behavior of the metrics, when changes are made to a service, is presented. The metrics that are addressed use the elements of a WSDL document for their computation. These are complexity, cohesion and coupling metrics. The comparative analysis is performed by creating a common set of changes and studying the behavior of the metrics before and after changes. The results show that different metrics for the same property do not necessarily vary the same way.*

Keywords: Web Service, Changes, Metrics, Intra Metric, Inter Metric

Received: 18 June 2015, Revised 21 July 2015, Accepted 29 July 2015

© 2015 DLINE. All Rights Reserved

1. Introduction

Service-oriented architecture (SOA) is a framework which guides the service-oriented development of software. A service is the smallest entity in SOA, which is loosely coupled and independent in nature. SOA has three major building blocks: service provider, service consumer and a central repository. Service provider describes a service via an interface document specified using web service description language (WSDL). It comprises of the service operations, messages, types, address etc. A service is published by the service provider in a central repository. Service consumer searches for a service in the repository and if found, invokes the service using simple object access protocol (SOAP) [1, 2, 3, 4].

In today's competitive world, web services constantly undergo changes. These could be triggered to having to adapt to the changing demands in the market. Also, to enhance the functionalities of an existing service, the service provider can incorporate some changes. In [5], the taxonomy of changes that can occur at service level is given. These changes are categorized as functional and non-functional changes. The functional changes are further sub-divided into two categories: structural changes and behavioral changes. Both the changes deal with the WSDL document. We consider the functional changes in our study.

Consider, now, the metrics. A metric represents standard quantitative measurement for the assessment of quality, progress or performance of a product or a process [6, 7]. In the context of services, the metrics are proposed to quantify the properties of a service. Of these the ones which deal with the functional aspects of a service are complexity [8, 9], cohesion [10, 15,16] and coupling among services [17, 18, 19, 20]. These metrics are defined using the elements of a service such as operations, messages etc. Thus, when functional changes are made to a WSDL document, these metrics may undergo a change. Our focus, therefore, is to study the change in these metrics when functional changes take place.

Our aim is to analyze the behavior of the metrics when changes take place in a service. The following questions motivated us for performing the analysis.

1. How do the different metrics defined for the same property behave when a service changes? In other words, do all complexity measures behave in the same way when changes take place?
2. Are there any appropriate metrics which helps to provide a measure of the complete change in a service?

The approach adopted in this paper is as follows.

A common example is considered for computing the metrics for complexity, cohesion and coupling. Each metric is computed. The services in the example are changed. The metrics are then re-computed. A comparative analysis is then performed.

We do not validate the efficiency of metrics for what they are proposed and do not claim that they are weak or strong enough to reflect their concerned properties. Instead, we focus on the analysis of the behavior of metrics when services undergo a change from the perspective of comparing them.

This paper is organized as follows. In section 2, we list the changes that are used to evaluate the metrics. In section 3, we explain an example scenario and list the changes that are to be effected. In section 4, we assess the metrics proposed by different authors after applying the appropriate changes. Finally, we conclude the paper in section 5.

2. Changes in a Service

A service is expressed using WSDL. The following elements are used in a WSDL document to describe a service [4]:

type– data type definition.

message– Input/output data for a service.

operation– a description of the action to be performed by a service.

port Type–a set of operations supported by one or more endpoints of a service.

binding– protocol detail and specification of data format for a particular port type.

service – a set of endpoints which is a combination of a binding and a URI where a service is located.

The elements which are common to all the metrics considered here are type, message, operation and port- Type. Hence, we assess the metrics for changes in these elements. As mentioned in [5, 10, 11, 12], there are five change operations which are addition, deletion, modification, merge and split. It is empirically analyzed that additions and deletions are the most frequent change operation in a service [13, 14]. Therefore, we consider both addition and deletion change operations. Taking into account all the aforementioned factors, the changes that are addressed in this paper are listed in Table 1.

S.No	Changes
1	Addition of an operation
2	Deletion of an operation
3	Addition of input or output message
4	Deletion of input or output message
5	Addition of schema type
6	Deletion of schema type
7	Addition of a portType
8	Deletion of a portType

Table 1. List of functional changes addressed

3. An Example

For the sake of uniform comparison, an example scenario of a Car Parts process given in [10] is used to evaluate the metrics proposed by different authors. The example comprises of two services: Customer and Order services. Customer service orders the part using “partorderrequest” operation and Order service receives the order request via “partorder” operation. We also consider these two services for analysis of metrics. The author has used Customer and Order service to calculate the values of the proposed metrics. The example scenario [10] has been reproduced in Figure.1.

Customer service	Order service
<pre> element name="partorderrequestelement" complex type xsd:element name="cust_name" type="xsd:string" xsd:element name="part_name" type="xsd:string" xsd:element name="count" type="xsd:int" message name="partorderrequest" part name="submit" element="partorderrequestelement" portType name="customer" operation name="partorderrequest" output="partorderrequestmessage" </pre>	<pre> element name="partorderrequestelement" complex type xsd:element name="cust_name" type="xsd:string" xsd:element name="part_name" type="xsd:string" xsd:element name="count" type="xsd:int" message name="partordermessage" part name="receive" element="partorderrequestelement" portType name="order" operation name="partorder" input="partordermessage" </pre>

Figure 1. WSDL fragment of Customer and Order service

We now discuss the changes to both the services in detail. The additions are discussed in section 3.1 and the deletions in section 3.2.

3.1 Additions

The operations that are added in the services are listed in Table 2.

In addition, we add the corresponding messages and types to support the added operations in the services.

The modified services are shown in Appendix A.

#	Operations in Customer service	Operations in Order service
1	-	checkavailability
2	confirmorderreceive	confirmorder
3	-	calculatediscount
4	-	calculateamount
5	receiveorder	sendorder
6	checkorder	-
7	makepayment	acceptpayment
8	sendfeedback	receivefeedback

Table 2. List of added operations

3.2. Deletions

To study the impact of deletions, a few operations have been deleted from Customer and Order services of Table 2. The deleted operations in the services are listed in Table 3.

We have presented the modified WSDL fragments in Appendix B after performing deletions from Appendix A.

#	Operations in Customer service	Operations in Order service
1	-	calculatediscount
2	sendfeedback	receivefeedback
3	stopservicereceive	stopservice

Table 3. List of deleted operations

4. Analysis of Metrics for Changes

In this section, we analyze the metrics for three properties which are complexity, cohesion and coupling. For all the metrics, a uniform method of evaluation is adopted. The analysis is performed stepwise. First, the metric is explained so that the computations in the next two steps are easily understood. The metric value is computed for the example scenario given in Figure.1. In the results, this value is shown under the column “Before”. Next, the metrics are recomputed after additions (services considered from Appendix A). In the results, this value is shown under the column “After A”. Subsequently, the modified services from Appendix B are considered for deletions. The metrics are recomputed. In the results, this value is shown under the column “After D”. We then calculate the percentage of change in the metric values after changes. This percentage of change is shown with the help of bar charts. Using these bar charts, we compare the metrics. It must be noted that the set of changes across all the metrics are constant. Complexity, cohesion and coupling are analyzed in that order.

4.1 Complexity

The complexity metrics are those that are given in [8,9].

4.1.1 Metrics by Thi et al.[8]

Description: In this paper, three metrics are proposed to measure the complexity of a WSDL file. We now explain each metric in turn.

a) Complexity based types (CBT) define complexities of all data types used in a service. $CBT = \sum c_i / n$; c_i is depth of tree correlating type t_i and n is number of types.

b) Complexity based message metric (CBM) define complexities of all messages used in a service. Complexity of message $(COM_k) = \sum c_i$; c_i is depth of tree correlating with i^{th} part of k^{th} message. $CBM = \sum COM_k / m$ where m is the number of messages.

c) Complexity based operations (CBO) assess complexity of service in terms of the complexities of input and output operations. $CBO = [\sum COM_i + \sum COM_o] / m$; where i is the input message and o is output message. Table 4 shows the re-computed metric values.

Service	CBT			CBM			CBO		
	Before	After A	After D	Before	After A	After D	Before	After A	After D
Customer	2	.88	1.14	3	2.5	2.62	3	2.52	.62
Order	2	.77	1.11	3	2.36	2.5	3	2.36	2.5

Table 4. Metric values [8]

4.1.2 Metrics by Baskil et al.[9]

Description: The authors have proposed four metrics which reflect the complexity of a web service. Data weight (DW) is the sum of complexities of each input and output message. $DW = \sum C(M_i)$ where $C(M_i)$ is the complexity of i^{th} message having <part> definitions which in turn have complexity. $C(M_i) = \sum w_{pi}$, where w_{pi} is the weight value of i^{th} <part> definition of the message. Data message ratio (DMR) reflects the complexity in terms of similar structured messages. $DMR = \sum [C(M), arg] / n$;

$C(M)$ reflects distinct message complexity value and arg is total no of parts in message and n is the total number of messages. Message entropy (ME) reflects number of occurrences of similar structured messages. $ME = -\sum(nom_i/n_m) * \log_2(nom_i/n_m)$ where nom_i is number of occurrences of similar structured messages and n_m is total number of messages. Message repetition scale (MRS) is measure for the variety in message structures in a WSDL document. $MRS = \sum (nom_i)^2/n_m$ where nom_i is number of occurrences of similar structured messages and n_m is total number of messages. The changed metric values after additions and deletions are shown in Table 5 and Table 6.

Service	DW			DMR		
	Before	After A	After D	Before	After A	After D
Customer	4	35	33	1	.5	.625
Order	4	31	27	1	.29	.40

Table 5. Metric values [9]

Service	ME			MRS		
	Before	After A	After D	Before	After A	After D
Customer	0	.59	.645	1	3.2	2
Order	0	.44	.51	1	6.43	3.6

Table 6. Metric values [9]

4.1.3 Comparative Analysis of Complexity Metrics

A comparative analysis of metric values for Customer and Order services for complexity property is shown in Figure.2. The metrics CBT, CBM, CBO and DMR show negative percentage of change for both additions and deletions. As specified by the authors, higher value of the metric reflects more complexity. However, values of all these metrics decreases. This behavior is an

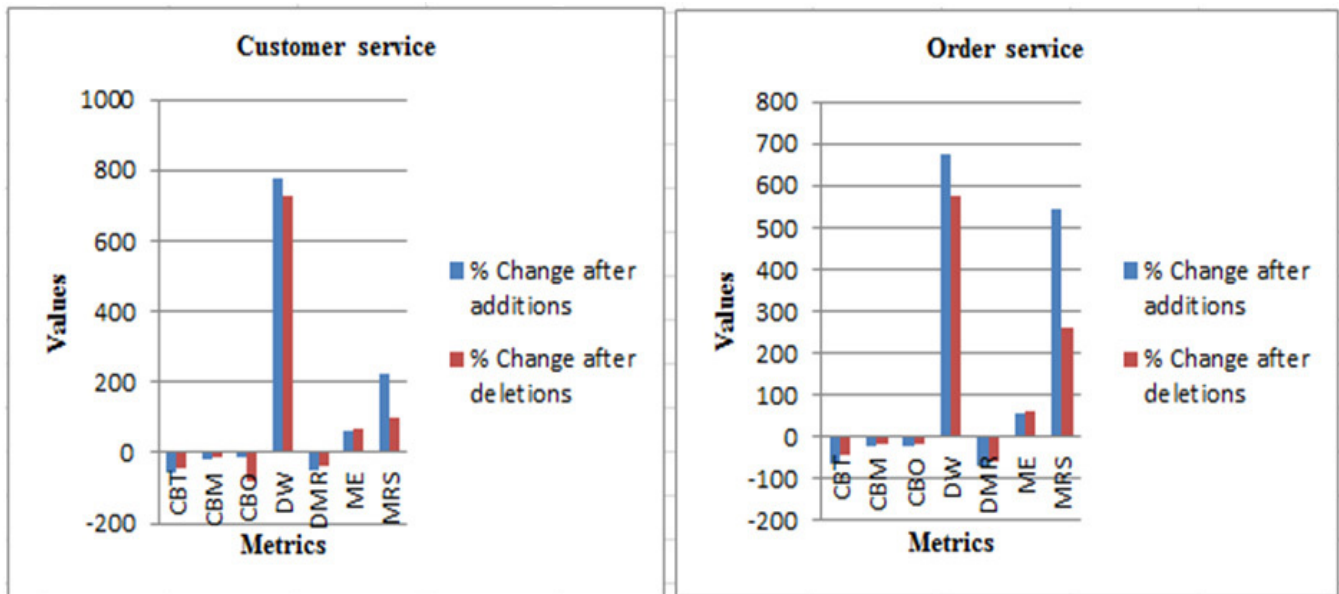


Figure 2. Complexity metric values

unexpected one. It may be recalled that the same set of additions and deletions are used for computing the metrics. The three metrics, DW, ME, MRS vary very differently for the same changes. Thus, it is unclear as to which metric is showing the correct change in complexity after service changes.

4.2 Cohesion

The metrics which are defined for cohesion property are given in [10,15, 16].

4.2.1. Metrics by Gebart et al.[15]

Description: As mentioned by the author, the proposed metrics are comparable with the cohesion property of OO systems. This makes the following metrics eligible for our consideration.

a) DANF (Division of agnostic and non-agnostic functionality) metric is a measure of how much a service provides agnostic or non-agnostic functionalities. Agnostic functionality is a generalized functionality i.e.it can be used in other contexts. The desired value of the metric is either 0 or 1.

$$|AF(O(RI(SI(s))))|$$

$$O(RI(SI(s)))|$$

where s is service; SI is service interface; RI is realized interface; AF is Agnostic functionality; O is total number of operations.

b) CEU (Common Entity Usage) metric is a measure which reflects whether all operations work on the same entity. Using this, operations meant for different entities can be included in separate services.

$$CEU(s) = \frac{|OUE(O(RI(SI(s))), CMP\{O(RI(SI(s))), MOUE(O(RI(SI(s))), UE(O(RI(SI(s))))\})|}{|O(RI(SI(s)))|}$$

where Operations Using Entities $OUE(O, BE)$ are operations out of O that only use entities from BE ; Composition $CMP(O, E1, E2)$ is biggest set of entities managed by operations O out of $E2$ that depend on $E1$; Used Entities $UE(O)$ are entities used within operations O as input; Mostly Often Used Entities $MOUE(O)$ are entities that are mostly used within one operation out of operations O . All calculated metrics are shown in Table 7.

Service	DANF			CEU		
	Before	After A	After D	Before	After A	After D
Customer	0	.37	.17	1	1	1
Order	0	.60	.43	1	1	1

Table 7. Metric values [15]

4.2.2.. Metrics by Athanasopouloset al. [16]

Description: The proposed metrics are discussed below.

a) Lack of sequential cohesion ($LoC_s(s_i)$) metric is the complement of the average sequential similarity between the pairs of operations that belong to $CS(s_i)$.

$LoC_s(s_i) = 1 - [\sum_{(opi,opj) \in CSsi} OpSseq(opi, opj)] / [|si.O|*(|si.O| -1)] / 2$ where $si.O$ is the service operation, opi and opj are the operations, $OpSseq(opi, opj)$ are the operations that are sequentially related to each other like input message of a service is the output message of other service.

a) Lack of communicational cohesion ($LoCC(s_i)$) metric is the complement of the average communicational similarity between the pairs of operations that belong to $CS(s_i)$.

$LoCc(s_i) = 1 - [\sum \forall_{(opi, opj) \in CSs_i} OpS_{comm(opi, opj)}] / [|s_i \cdot O| * (|s_i \cdot O| - 1)]/2$ where $OpS_{comm(opi, opj)}$ are the operations that have some commonality in each other's input/output messages .

The recomputed values of metrics using the data are shown in Table 8.

Service	$LoCS(s_i)$			$LoCC(s_i)$		
	Before	After A	After D	Before	After A	After D
Customer	1	.986	.975	1	.992	.974
Order	1	.986	.974	1	.984	.945

Table 8. Metric values [16]

4.2.3. Metrics by Wang et al. [10]

Description: Cohesion metric is a measure for average degree of dependency of all service elements. $Cohesion(x) = [Ix + \sum D(i)] / m$, where m is the total number of elements of a service $x = (e_1, e_2, \dots, e_m)$. I_x is the intradependency and $D(i)$ is the interdependency value of the service. Here, the metric is referred to as COH in the comparative study, even though this is not the acronym used by the authors. We recalculate all the metric values and the results after additions and deletions shown in Table 9.

Service	COH		
	Before	After A	After D
Customer	1.86	1.49	1.51
Order	.86	1.36	1.37

Table 9. Metric values [10]

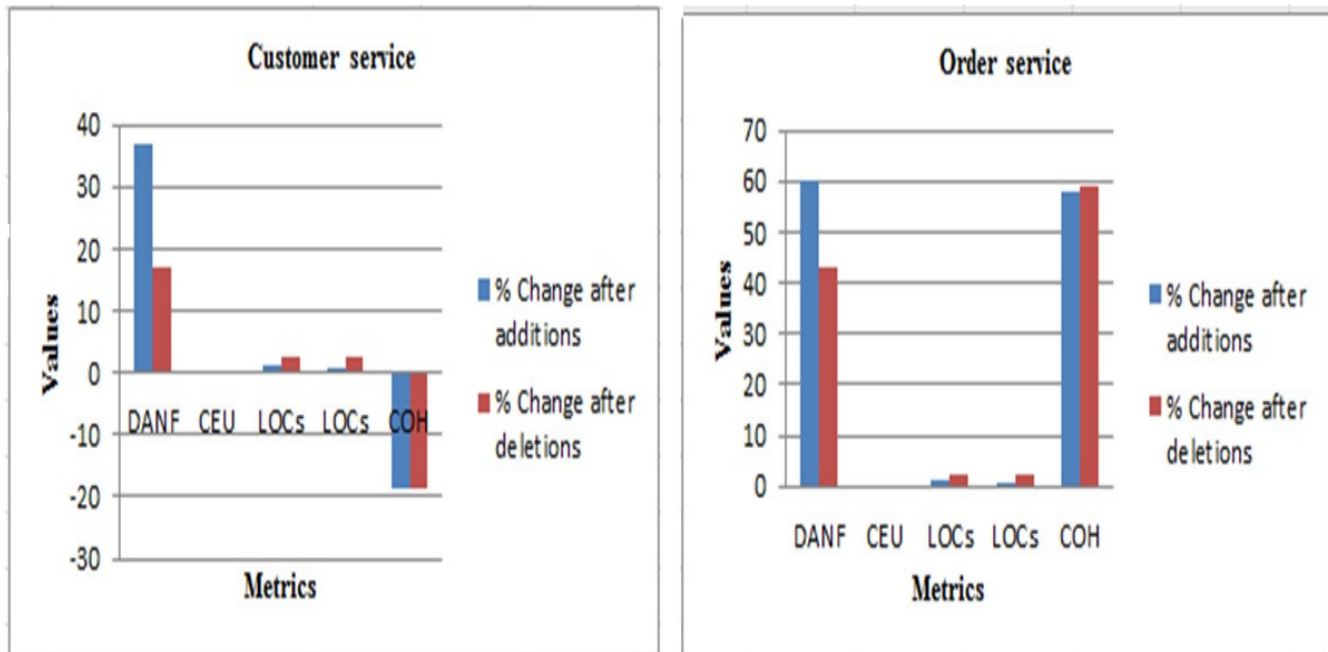


Figure 3. Cohesion metric values

4.2.4 Comparative Analysis of Cohesion Metrics

Figure. 3 shows the comparative metric values for Customer and Order services. It can be seen from the charts that cohesion levels show an increase with additions in all except COH. Deletions also show an increase in cohesion except in COH for Customer service. Further, for the same additions and deletions to the services, in DANF cohesion level change is better for additions whereas these levels are better for deletions in the rest of the services (except, of course, COH for customer service). Therefore, despite of the fact that all the metrics represent the same property, their behavior remains unclear.

4.3 Coupling

The coupling metrics deal with the relationships among services. The relationships between the operations of Customer and Order services (from Appendix A) are shown in Table 10. The relationships are depicted as arrows. The source operation for the arrow provides the input to the operation which is at the end of the arrow. For example, the arrow from operation partsorderrequest to operation partsorder shows that the output from partsorderrequest is used as input for partsorder operation.

#	Operations in Customer service	Operations in Order service
1	partorderrequest	partorder
2	-	checkavailability
3	confirmorderreceive	confirmorder
4	-	calculatediscount
5	-	calculateamount
6	receiveorder	sendorder
7	checkorder	-
8	makepayment	acceptpayment
9	sendfeedback	receivefeedback
10	partschangerequest	partschange
11	stopservicereceive	stopservice

Table 10. Relationships between operations of services

Coupling metrics considered here are those which are proposed by authors in [17-20]. We discuss and analyze each of them in this section.

4.3.1. Metrics by Quynh et al. [17]

Description: Coupling between services (CBS) which is the number of relationships between a service A and other services. It is calculated as $\sum A_i B_j$. Instability metric for service metric (IMS) shows interaction between the services through sent and received messages. It is defined as fan.out / (fan.in + fan.out). Degree of coupling between services metric (DC2S) gives the relationship between two services in system. It is given by the formula $[N(A, B) / \sum N(A, B_i)] * 100$. Here, $N(A, B_i)$ are the number of calls from service A to operators of Bi. Degree of coupling within a given set of services metric (DCSS) is defined in terms of a graph drawn for the service connectivity in the system and is calculated by the below formula.

$$DCSS = \frac{[Max - \sum d(u,v)]}{Max - Min}$$

where $d(u,v)$ is number of calls from node u to v for all the services in system, $Max = k*v*(v-1)$ and $Min = v*(v-1)$. Max is calculated only when nodes in graph are not connected and Min is calculated when graph is connected. K is defined only when there is no path between any of the nodes and is given the maximum value for the shortest distance between any two nodes. Table 11 and Table 12 show values of all metrics after changes.

Service	CBS			IMS		
	Before	After A	After D	Before	After A	After D
Customer to Order	1	1	1	1	.57	.60
Order to Customer	0	1	1	0	.43	.40

Table 11. Metric values [17]

Service	DC2S			DCSS		
	Before	After A	After D	Before	After A	After D
Customer to Order	100	100	100	.5	2	1.5
Order to Customer	0	100	100	0	1.5	1

Table 12. Metric values [17]

4.3.2. Metrics by Thang et al. [18]

Description: The two coupling metrics are ANPL and ANML. Average number of methods per link (ANML) = #methods / # links. In our terms, it gives the number of operations per link. Average number of parameters per link (ANPL) = #parameters / # links. In our terms, it gives the number of messages passed per link.

The results are presented in Table 13.

Service	ANML			ANPL		
	Before	After A	After D	Before	After A	After D
Customer to Order	1	4	3	1	4	3
Order to Customer	0	3	2	0	3	2

Table 13. Metric values [18]

4.3.3 Metrics by Xu et al. [19]

Description: Author has proposed four decoupling metrics for service-oriented enterprise application in the service composition. The two metrics which deals with coupling between services are given below.

1) Average Required Services Dependency Metric (ARSD) is $\sum R_i / n$, where R_i is the number of services a given service needs to complete its functionalities in a composition and n is the total number of services. It is the average count of services which uses each other's operations in a system.

2) Average Service Invocation Coupling (ASIC) is defined as $ASIC = \sum (W_{i, async} N_{i, async} + W_{i, sync} N_{i, sync}) / n$ where $N_{i, async}$ and $N_{i, sync}$ are the numbers of non-blocking asynchronous operations and synchronous operations respectively defined in service i . $W_{i, async}$ and $W_{i, sync}$ are the weights of asynchronous operations and synchronous operations respectively defined in service i . The author has mentioned that $W_{i, async} + W_{i, sync} = 1$ and $W_{i, async} > W_{i, sync}$ but weights are not assigned any values. Therefore, we assume that $W_{i, async} = .6$ and $W_{i, sync} = .4$.

Service	ARSD			A SIC		
	Before	After A	After D	Before	After A	After D
Customer to Order	.5	1	1	.3	2.4	.9
Order to Customer	0	1	1	0	.9	.6

Table 14. Metric values [19]

The newly calculated values are shown in Table 14.

4.3.4. Metrics by Zhang Q. et al. [20]

Description: The author has proposed four metrics for measuring coupling between services. Relative Coupling of Service (RCS) is a measure of degree of coupling in a service. $RCS[s] = CS[s] / NS$, where CS is the count of services that are invoked by service s and NS is the total number of services. Relative Importance of Service (RIS) is an indicator of a service importance in the system. $RIS(s) = IS[s] / NS$, where $IS[s]$ is the total number of service invokers which invoke services. Relative Network Cohesion in the System (RNCS) counts the complexity of system based on number of invocations among services. $RNCS = NCS / NS$, where NCS is the number of invocations by all services in system. Service Coupling Factor (SCF) is a measure of the overall complexity of system. $SCF = \sum CS[s]i / NS^2 - NS$.

All values before and after changes are shown in Table 15 and 16.

Service	RCS			RIS		
	Before	After A	After D	Before	After A	After D
Customer to Order	.5	.5	.5	0	.5	.5
Order to Customer	0	.5	.5	.5	.5	.5

Table 15. Metric values [20]

Service	RNCS			SCF		
	Before	After A	After D	Before	After A	After D
Customer to Order	.5	3.5	2.5	.5	1	1
Order to Customer	.5	3.5	2.5	.5	1	1

Table 16. Metric values [20]

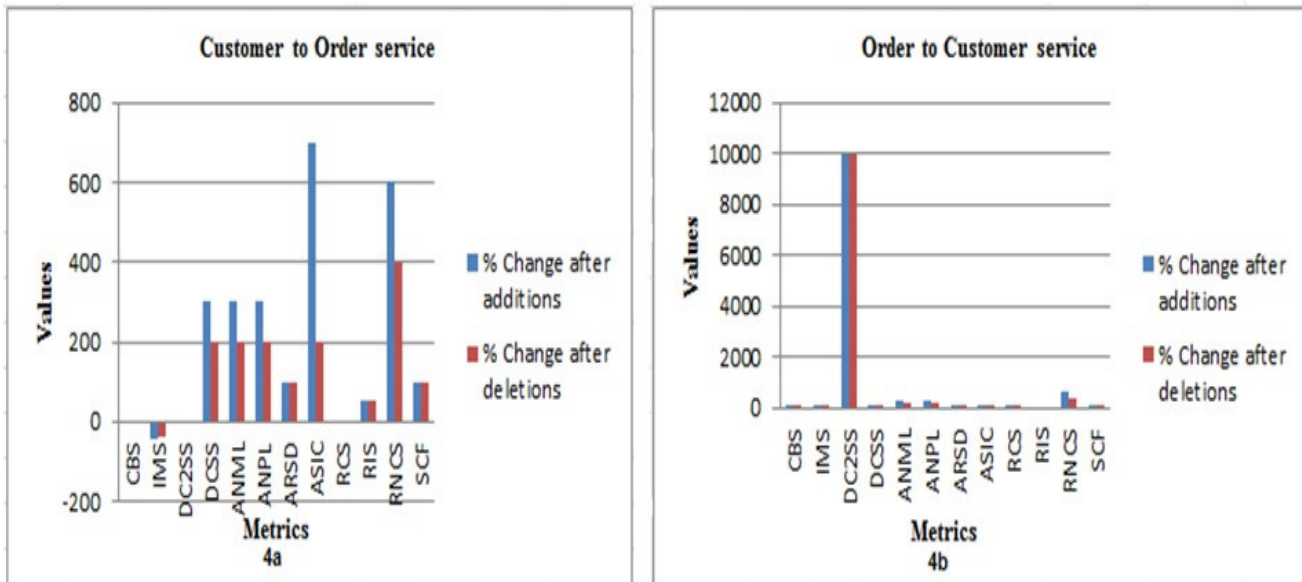


Figure 4. Coupling metric values

4.3.5. Comparative Analysis of Coupling Metrics

The comparative analysis of coupling metrics is given in Fig 4a and 4b. The metrics vary differently for the same changes. If we

consider customer to order service as given in Figure 4a, IMS metric in fact, shows that coupling has decreased whereas the rest of the metrics say coupling has increased. The question, thus, that remains unanswered is whether coupling has increased or decreased as a result of changes. If we look at Figure. 4b all the metrics show a rise in coupling. However, DC2S shows a sharp rise in its value as compared to the rest of the metrics. This could be due to the use of percentage rather than discrete values for measuring coupling.

5. Conclusion

In this paper we have presented a comparative study of the metrics proposed by different authors. An example scenario was taken for analyzing the behavior of metrics under changes to the services of this scenario. The method adopted was (a) to compute the metrics before any additions/ deletions, after additions and after deletions and (b) analyze the metrics for each property. From the comparative analysis, we found that the existing metrics, inspite of representing the same property, differed in their behavior for the same set of changes.

We believe that the study reported in the paper provides, for the first time, a clear and empirical comparative analysis of the metrics for services via a consistent set of changes. In our future work, we shall analyze the metrics for more complex scenarios comprising of many services. Also, we propose to investigate defining change metrics for services which directly measure the changes.

Acknowledgments

This research work was funded under UPE2 Project 114 from Jawaharlal Nehru University, New Delhi.

References

- [1] Erl, T. (2015). *Service-Oriented Architecture (SOA): Concepts, Technology, and Design* (The Prentice Hall, New Jersey).
- [2] Daigneau, R. (2011). *Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services* (Addison Wesley, New Jersey).
- [3] Erl, T. (2008). *SOA Principles of Service Design* (Prentice Hall, New Jersey).
- [4] World Wide Web Consortium. <http://www.w3.org/TR/wsdl>.
- [5] Akram, S., Bouguettaya, A., Liu, X. F., Rosenberg (2010). A change management framework for service oriented enterprises. *International Journal of Next-Generation Computing*.
- [6] Fenton, N., Bieman, J. (2014). *Software metrics: a rigorous and practical approach*. CRC Press.
- [7] Curtis, B. (1980) Measurement and experimentation in software engineering. *In: Proceedings of the IEEE*, 68, 1144 – 1157,
- [8] Thi, Q. P., Quang D. T., Quyet T. H. (2009): A Complexity Measure for Web Service. *International Conference on Knowledge and Systems Engineering* 226-231.
- [9] Baski, D., Misra, S. (2011). Metrics suite for maintainability of eXtensible Markup Language web services, *The Institution of Engineering and Technology*. 320 - 341.
- [10] Wang, S, M. M. A., A. C. M. (2009). Dependency Impact Analysis Model For Web Services Evolution. *IEEE International Conference on Web Services*. 359 - 365.
- [11] Matjaz, B., Juric, A. S., Brumen, B., Rozman, I. (2009). WSDL and UDDI extensions for version support in web services, *Journal of Systems and Software*. 1326-1343.
- [12] Wang, Y., Yang, J., Zhao, W., Su, J. (2012). Change impact analysis in service-based business processes. *Service Oriented Computing and Applications*. 131-149.
- [13] Romano, D., Pinzger, M. (2011). Analyzing the Evolution of Web Services using Fine-Grained Changes. *IEEE International Conference on Web Services*. 392-399,
- [14] Fokaefs, M., Mikhael, R., Tsantalis, N., Stroulia, E., Lau, A. (2011). An empirical study on web service evolution, *IEEE International Conference on Web Services*. 49-56

- [15] Gebhart, M. (2013) Measuring Design Quality of Service-Oriented Architectures Based on Web Services. *The Eighth International Conference on Software Engineering Advances*
- [16] Athanasopoulos, D., Zarras, A.V. (2011) .Fine-grained Metrics of Cohesion Lack for Service Interfaces. IEEE International Conference on Web Services. 588 - 595
- [17] Quynh, P.T., Thang, H.Q.(2009). Dynamic Coupling Metrics for Service –Oriented Software. *International Journal of Electrical and Electronics Engineering* 3 (5) 282
- [18] Thang, H. Q., Quynh, P. T., Viet, T. Q. (2006). The Reusability and Coupling Metrics for Service Oriented Software.
- [19] Xu, T., Qian, K. He, X. (2006). Service Oriented Dynamic Decoupling Metrics.*The International Conference on Semantic Web and Web Services*
- [20] Zhang, Q., Li, X. (2009). Complexity Metrics for Service-Oriented Systems. *Second International Symposium on Knowledge Acquisition and Modeling*. 375-378
- [21] Basci, D., Misra, S. (2009). Measuring and Evaluating a Design Complexity Metric for XML Schema Documents, *Journal of Information Science and Engineering*. 1405-1425.
- [22] Gebhart, M., Abeck, S. (2011). Metrics for Evaluating Service Designs Based on SoaML. *International Journal on Advances in Software* .61-75
- [23] Chidamber, S. R., Kemerer, C. F. (1994). A Metrics suite for Object Oriented Design, *IEEE Transactions Software Engineering*. 476 - 493
- [24] Washizaki, H., Yamamoto, H., Fukazawa, Y. (2002).Software Component Metrics and It's Experimental Evaluation, *Symposium Empirical Software Engineering*. 19-20.
- [25] Lionel, S., Briand, C. Empirical Studies of Quality Models in Object Oriented Systems. Software Quality Engineering Lab, *Systems and Computer Engineering*. Carleton University.
- [26] Deitel, H. M. (2003) . Web Services: A Technical Introduction. Prentice Hall.
- [27] Mattsson, M.K. (2005). The State of Art within Evolution and Maintenance of Web Services. Technical Report, No. 2005-027, Department of Computer and Systems Sciences (DSV), *Stockholm University and Royal Institute of Technology*.
- [28] w3schools. <http://www.w3schools.com/webservices/>.
- [29] Lee, K., Lee, S. (2006). A quantitative evaluation model using the iso/iec 9126 quality model in the component based development process. *ICCSA* (4) 3983, 917–926