Measuring Service Cohesion Using Latent Semantic Indexing

Ali Kazemi, Ali Rostampour, Fereidoon Shams, Pooyan Jamshidi, Ali Nasirzadeh Azizkandi
Automated Software Engineering Research Group
Electrical and Computer Engineering Faculty, Shahid Beheshti University GC,
Tehran, Iran
E-mail :{Ali.Kazemi,A.Rostampour}@mail.sbu.ac.ir
{F_Shams, P_Jamshidi}@sbu.ac.ir,Ali_Nasirzadeh@tabrizu.ac.ir

***Abstract*—As low coupling, high cohesion is a service-oriented design and development principle that should be kept in mind during all stages. High cohesion increases the clarity and ease of comprehension of the design that simplifies maintenance and achieves service granularity at a fairly reasonable level. However, unlike coupling that only measures the degree of structural and behavioral dependency to the other services, cohesion metrics need to evaluate the degree of semantic relationships between operations within a service in order to measure functional relatedness. Latent semantic indexing (LSI) is one of the techniques in the field of information retrieval which is widely used to measure the degree of semantic relatedness between a document and a given query and also used to measure the cohesion of a text. In this paper, we propose an approach to automatically measure the strength of conceptual cohesion of a service based on LSI technique. Finally, it has been evaluated theoretically based on a set of cohesion principles.**

Keywords-Servivce cohesion; Latent semantic indexing; Software metric.

I. INTRODUCTION

Service-oriented architecture (SOA) is a promising solution to build enterprise application programs which supports processes and functions as a set of well-defined services [1], [2]. Simply, a service is defined as a set of related operations. Thus, it is a logic encapsulated by individuals which supposed to be reusable [3]. Considering design standards, which causes services to be potentially reusable, the chance of a service to be able to accommodate future requirements with the least development effort increases [4]. Therefore, reusability is an important quality attribute which must be measured to satisfy an important need in SOA, the need for independent services to deliver a reusable functionality [3], [4]. One of the design attributes which has a great impact on reusability of a particular service, is cohesion, so that higher cohesion significantly increases service reusability [5], [6]. Also in [7], there is an elaborate discussion about the impact of cohesion on maintainability. The higher the cohesion of a service, the easier the test and analysis and higher cohesion will improve system stability and changeability [8] and, consequently maintainability of a system will be improved.

Because of inherently conceptual nature, cohesion is one of the most complicated and difficult structural attribute of a class, component, or a service from quantifying point of view [8].This quality attribute can be measured based on conceptual relatedness degree of operations which are exposed in service interface. According to a cohesion category proposed in [6], [7], conceptual cohesion is considered as the strongest type of cohesion. However, this type of service cohesion cannot be easily measured using the previous traditional metrics due to additional level of abstraction and highlighted characteristics of service interfaces in comparison with procedural and object-oriented paradigms [6].

Using the concept of latent semantic indexing (LSI), we will evaluate the conceptual relatedness degree of operations existing in a service. For the first time, LSI was used in information retrieval techniques [9], [10]. One of the applications of this method is measuring cohesion of a text [11]. LSI provides completely automatic approach that compares information units in order to measure conceptual relatedness. Measuring conceptual relatedness degree of units relies on a powerful mathematical method called Singular Value Decomposition (SVD) [10], [11]. Therefore, the objective of this research work is to propose a LSI-based approach for measuring the degree of conceptual cohesion in a service.

In order to adopt the LSI technique to measure service cohesion, we get the required information of interactions between business processes and business entities that are mostly used during service identification [6]. The SVD method is applied on a well-defined structure comprising this information. The output of this algorithm is used to quantitatively measure conceptual cohesion degree of the identified services. Utilization of semantics existing in enterprise processes is completely proportional to this inherently conceptual nature, and therefore we will have a more precise measurement of conceptual cohesion of services.

The rest of this paper is organized as follows. Section II introduces the most related works. In Section III, basic concepts of the utilized terminologies are defined. The LSI concepts and the way of adopting them are introduced in Section IV. The proposed metric and complementary example and issues are discussed in Sections V, VI and VII, respectively. The theoretical principles of the metric are

 ISBN: 978-1-61208-124-3

evaluated in Section VIII. Finally, the conclusion, which leads to further research, is explained in Section IX.

II. RELATED WORK

In this section, we briefly present some of the previous works on measuring cohesion in service-oriented, object-oriented and procedural paradigms. The concept of cohesion in OO and procedural paradigms has been widely discussed and examined. For example, in [12], six semantic categories of procedural cohesion namely Coincidental, Logical, Temporal, Communicational, Sequential, and Functional have been proposed. The concept of cohesion later was extended by Eder et al. [13] to cover conceptual and technical features introduced in OO paradigm. Eder et al. [13] proposed five cohesion categories (from the weakest to strongest): Separable, Multifaced, Non-delegated, Concealed and Model. Moreover, in [6], eight semantic categories of service-oriented cohesion are proposed. These categories are: Coincidental, Logical, Temporal, Communicational, External, Implementation, Sequential, and Conceptual. In [6], four categories namely Communicational, External, Implementation, and Sequential are represented as quantifiable cohesion categories. On the other hand, four categories namely Coincidental, Logical, Temporal, and Conceptual are identified in this paper as purely semantic cohesion categories. They believe that second four categories are semantic based whereas first ones are measurable without considering semantic issues. The proposed quantifiable cohesion categories have indirect impact on conceptual cohesion. A brief representation of the cohesion metrics has been shown in Table I.

TABLE I. SUMMARY OF COHESION METRICS IN THE LITERATURE

Name	Definition
LCOM [14]	Non-similar method pairs are counted in a class of pairs.
LCOM3 [15]	The number of connected components in Graph is counted: Nodes are methods and edges are connections between similar methods.
RLCOM [16]	Number of non-similar method pairs, to The total number of method pairs ration in the class.
TCC [17]	Ratio of number of similar method pairs to total number of method pairs in the class.
WTCoh [18]	Number of used shared data entities by methods and also taking the transitive cohesion into account.
SIDC [7]	Number of shared parameters of the service operations divided by the total number of parameters
DM IAUM [19]	Number of system services divided by the total number of used messages
SIDC [6]	This metric is introduced to measure communication cohesion and it considers parameters and common return types.
SIUC [6]	This metric describes that a service is externally cohesion when all of its operation are invoked by all clients of this service.
SIIC [6]	This metric describes that a service has implementation cohesion when its all operations are implemented by the same implementation.
TICS[6]	A service is deemed to be Sequentially cohesive when all of its service operations have sequential dependencies, where a post condition/output of a given operation satisfies a precondition/input of the next operation.

It is worth to mention that in [20] and in [8], in addition to the number of shared parameters, other shared attributes such as number of service consumers, operations sequence and some more shared attributes are considered. To the best of our knowledge, there is no metric which measures the degree of relationship between operations of a service from conceptual point of view. Most proposed cohesion metrics in previous studies expect the services to have common inputs and outputs and does not consider the inter-relation of their parameters. To measure the conceptual cohesion, we require additional semantics. Therefore, we should look for methods that can measure the strength of conceptual relationship between two operations of a service by means of assets which are available in design level (processes from which services are identified) and then propose a metric for measuring conceptual cohesion of a service.

III. BASIC CONCEPTS

In this section, we present definitions of several key notions that will be utilized in this paper.

Definition 1 (**Business Entity**): A business entity (BE) is a dominant information entity with an associated data model and an associated behavior model in the context of a process scope [23].

Definition 2 (**Elementary Business Process**): An elementary business process (EBP) can be defined as EBP = {n, (BE_j, sr)}, where n is the name of elementary business process, BE_j is the *j*th business entity which semantically relate to corresponding EBP. sr={"C", "R", "U", "D"} is the type of semantic relationship between EBP and BE_j[22].

Definition 3 (**CRUD matrix**): A CRUD matrix can be defined as M= {(EBP_i, BE_j) i=1...#row, j=1...#column}, where EBP_i is the i^{th} EBP and BE_j is the j^{th} BE. #row is the number of EBP and #column is the number of BE in the model [22].

Definition 4 **(Conceptual Cohesion) :** There is a meaningful semantic relationship between all operations of a service in terms of some identifiable domain-level concept. [6].

IV. APPLICABILITY OF LSI IN COHESION MEASUREMENT

LSI is a vector model-based technique which is applied in many information retrieval applications. In the vector model, each document is simply represented by a $A_{n\times m}$ term-document matrix, where n is the number of terms and m is the number of documents in the collection. Each cell,$a_{i,j}$, is the frequency of term t_i in the document d_j. LSI technique includes the following main steps:

1. A matrix is formed; each row of this matrix is corresponded to a term which occurs in the document. Each element (m,n) in the matrix is corresponded to number of times that term m occurs in document.

2. Local and global weighting of terms is applied to each element of the term-document.

3. SVD is used by LSI and decomposes the matrix into three other matrices: T, a term in the dimension; S, a

 ISBN: 978-1-61208-124-3

diagonal matrix of singular values, and D a document matrix in the dimension. The number of dimensions is considered as t= min (m,n) where m and n are the number of the terms and the number of documents in the main term-document matrix respectively. The matrix can be provided by $A = TSD^T$ where D^T is transposed of matrix D.

4. In the LSI system, the T, S and D matrices are truncated to k dimensions. Dimensional reduction reduces "noise" in the term–term matrix resulting in a richer term relationship structure that reveals latent semantics and is a crucial step in this research work.

Now we explain each one of the above steps in more details.

In the first step, the term-document matrix A is formed.

In the second step, a weight is assigned to each term in the document. There are different weighting models which are explained in [20][9]. The simplest weighting model can be obtained simply by counting number of frequency of a term in the document. In order to put the weights in the interval [0,1], the weight of each term is divided in document by tf_{max}, where tf_{max} is the maximum of term in the document.

In the third step, term-document matrix, A is taken and then is decomposed into three matrices T, S, and D using SVD. Matrices T, S and D keep the information related to terms, singular values and documents respectively.

In the fourth step, T, S, and D matrices are decreased to K domains. After dimensional reduction, the term-term matrix can be approximated using the formula: $TTS = T_K S_K (T_K S_K)^T$.

In this work, we suppose that the value which exists in location (i,j) of TTS matrix show the similarity between terms i and j in the collection. The value of K is optional, in this paper according to [20], K=2. Our main goal in this paper is to present cohesion metric which is able to measure the strength of conceptual similarity between operations of a service. In the following lines we explain the way of mapping above concepts to the ones which exist in SOA.

Similar to LSI, we define the BE-EBP matrix $A_{n\times m}$ of enterprise processes and business entities. Each (i,j) element in the matrix A shows the weight of *i*th business entity in the *j*th business process which is defined as the number of times that *j*th business process accesses the *i*th business entity. Each process is considered as a document. For example, the claim business process in [23] is able to access three business entities Loss Event, Claim, and Payment. The related row to this process in matrix is shown in Table II.

TABLE II. THE ASSOCIATED BE-EBP MATRIX

Business Entity ↕ / Process ↔	P1				
Claim	10				
Loss Event	3				
Payment	5				

As shown in Table II, process P1 have accessed the Claim, Loss Event , and Payment business entities 10 ,3, and 5 times respectively. The above matrix is completed for all enterprise business processes in a way that the number of its columns is equal to enterprise business processes and the number of its rows is equal to enterprise business entities. Then, three matrices are obtained using SVD. Considering K=2, the reduced matrix $TTS = T_2 S_2 (T_2 S_2)^T$ is formed.

TTS matrix shows the relationship between business entities. The values of the elements in TTS matrix are not normalized, and they can even be negative. Since negative values have no meaning, we substitute it by zero which means no degree of cohesion between service operations. Also to normalize values, we multiply matrix by $1/_{max}$. Where *max* is the greatest value in the TTS matrix. Therefore, using the LSI concepts, we could show the existing semantic in business process in the form of TTS matrix. Finally, we use this matrix to obtain the relationship between operations of a service.

V. THE PROPOSED METRIC

The metric will be introduced in this section can be used for measuring the cohesion of a service in design time, based on the exposed operations in its interface. Note that the proposed metric is defined on an absolute scale, where a value is assigned to it in a range between 0 to1. Value 1 shows the strongest cohesion and 0 shows lack of cohesion. Values between 0 and 1 are considered as different degrees of cohesion.

As we mentioned in Section IV, to measure a service cohesion using the proposed metric, first the BE-EBP matrix should be formed. This matrix can be formed based on those enterprise processes which services are intended to be obtained from their decomposition as defined in definition 3.

The measuring procedure has the following form. Firstly a matrix $A_{n\times m}$ is formed where *m* is the number of enterprise business entities and *n* is the number of enterprise processes. Then the number of times that each business entity i accessed by business process j, is considered as element (i,j) of matrix A. In order to obtain conceptual relatedness between business entities, we apply SVD on matrix A. Its outputs are three matrices which are shown as $A = TSD^T$.

As we discussed earlier, $TTS = T_2 S_2 (T_2 S_2)^T$ matrix shows the conceptual relatedness between business entities which are used to obtain the strength of service cohesion. For this purpose, we use a graph based approach.

Suppose that service S has a set of operations $O = \{O_1, O_2, \ldots, O_m\}$. Each operation O_j of the service S accesses a set of business entities which is shown as $BE_j = \{BE_{j,1}, BE_{j,2}, \ldots, BE_{j,n}\}$. For each pair of operations O_i and O_j in the service S we form a complete graph G=(V,E) so that $V = BE_i \cup BE_j$.

Now, in set E, we assign a value for each edge that represents the degree of relationship between business entities, which is considered as nodes in graph G. The degree of relationship between two business entities can be measured from TTS matrix. The degree of conceptual

 ISBN: 978-1-61208-124-3

relatedness between two operations i and j is calculated through formula:

$$OCV(i,j) = \begin{cases} \dfrac{\sum_{\forall p \in V} \sum_{\forall q \in V, p>q} TTS_{p,q}}{|V| \times (|V|-1)/2} & |V| > 1 \\ 1 & |V| = 1 \end{cases} \quad (1)$$

where:

- p and q are two business entities in V.
- $TTS_{p,q}$ is the degree of relationship between two business entities, BE_p and BE_q.
- |V| is the cardinality of set V.
- The denominator is the number of edges in the complete graph G.

The strength of cohesion is defined as the degree of relationship between service's operations.

$$SCV(S) = \begin{cases} \dfrac{\sum_{\forall i \in o} \sum_{\forall j \in o, i>j} (OCV(i,j))}{m \times (m-1)/2} & |m| > 1 \\ 1 & |m| = 1 \end{cases} \quad (2)$$

where:

- m is the number of operations in service S.

VI. EXAMPLE

In this section, we show how the proposed metric works using an example. To do that, we must have the enterprise processes and services which are identified using those processes. Using a real-world business process the effectiveness of the proposed metric is studied and evaluated. The sales department is studied in this scenario [22].

Using CRUD matrix is one of the ways to identify a service [22]. Table III illustrates the CRUD matrix associated to our scenario. Identified services are shown in the form of clusters with different colors (Table III).

TABLE III. THE CRUD MATRIX FOR SALES DEPARTMENT SCENARIO

EBP \ BE	customer	Credit	Account receivable note	Order	Discounts	Invoice	Shipping schedule	Draft	Inventory	Warehouse voucher
Add Customer	C	C								
Add an Account receivable note	R	U	C			R				
Check Credit	R	R		R						
Receive order	R			C						
Calculate discounts				R	R					
Check inventory				R					R	
Calculate price				R	R					
Add discounts					C					
Issue invoice	R	R		R		C				
Schedule shipping						R	C			
Issue draft						R	R	C		
Add an Item									C	
Add a warehouse voucher	R					R			U	C

The BE-EBP matrix is shown in Table IV. Since EBPs existing in the CRUD matrix access each business entities just 0 or 1 time, elements of this matrix are just 0 and 1. For example, Add Customer accesses only customer and credit BEs, therefore there are just two ones in Add Customer column.

TABLE IV. THE BE-EBP MATRIX

BE \ EBP	Add Customer	Add an Account note	Check Credit	Receive order	Calculate discounts	Check inventory	Calculate price	Add discounts	Issue invoice	Schedule shipping	Issue draft	Add an Item	Add a warehouse voucher
Customer	1	1	1	1	0	0	0	0	1	0	0	0	1
Credit	1	1	1	0	0	0	0	0	1	0	0	0	0
Account receivable note	0	1	0	0	0	0	0	0	0	0	0	0	0
Order	0	0	1	1	1	1	1	0	1	0	0	0	0
Discounts	0	0	0	0	1	0	1	1	0	0	0	0	0
Invoice	0	1	0	0	0	0	0	0	1	1	1	0	1
Shipping schedule	0	0	0	0	0	0	0	0	0	1	1	0	0
Draft	0	0	0	0	0	0	0	0	0	0	1	0	0
Inventory	0	0	0	0	0	1	0	0	0	0	0	1	1
Warehouse voucher	0	0	0	0	0	0	0	0	0	0	0	0	1

After the matrix of BE-EBP is obtained, we apply SVD algorithm on it. To do that, MATLAB version 7.6.0.324 has been used. To obtain business process entity matrix we use this equation:

$$TTS = T_2 S_2 (T_2 S_2)^T \quad (3)$$

The resulted matrix has been shown in Table V. Also this matrix has been normalized and its negative values have been substitute with 0s.

TABLE V. THE BE-BE MATRIX AFTER DECOMPOSITION AND NORMALIZATION

BE \ BE	Customer	Credit	Account receivable	Order	Discounts	Invoice	Shipping schedule	Draft	Inventory	Warehouse voucher
Customer	0	0.99	0.27	0.94	0.15	1.00	0.19	0.10	0.33	0.21
Credit	0.99	0	0.19	0.68	0.11	0.72	0.14	0.07	0.23	0.15
Account receivable	0.27	0.19	0	0.23	0	0.28	0.08	0.04	0.06	0.06
Order	0.94	0.68	0.06	0	0.57	0.14	0	0	0.22	0.03
Discounts	0.15	0.11	0	0.57	0	0	0	0	0.03	0
Invoice	1.00	0.72	0.28	0.14	0	0	0.36	0.19	0.23	0.24
Shipping schedule	0.19	0.14	0.08	0	0	0.36	0	0.08	0.04	0.08
Draft	0.10	0.07	0.04	0	0	0.19	0.08	0	0.02	0.04
Inventory	0.33	0.23	0.06	0.22	0.03	0.23	0.04	0.02	0	0.05
Warehouse voucher	0.21	0.15	0.06	0.03	0	0.24	0.08	0.04	0.05	0

Next we show how to calculate the cohesion of a service using the proposed metric. Table III show a CRUD matrix with four identified services. First we show how to calculate the metrics for the first service which is shown by blue color.

The service has three operations which are specified by following names: Add Customer, Add an Account receivable note, Check Credit.

We have:

$O = \{O_1, O_2, O_3\}$

$BE_1 = \{Customer, Credit\}$

$BE_2 = \{Customer, Credit, Accountreceivablenote\}$

$BE_3 = \{Customer, Credit\}$

In order to obtain conceptual relatedness between operations of a service we use a graph. For operations O_1 and O_2, graph G= (V,E) has the form of Figure1. In this graph the set V has the following form.

$$V = BE_1 \cup BE_2 = \{Customer, Credit, Accountreceviablenote\}$$

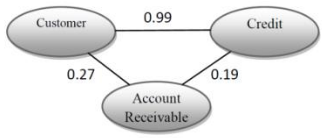

Figure 1. Business Entities Graph for Service 1

$$OCV(1,2) = \frac{0.9967 + 0.1954 + 0.2712}{3} = 0.4877$$

The results of the alternatives for the specified service S1 is shown in Table VI.

TABLE VI. OCV VALUE FOR SERVICE 1

Metric \ O_i,O_j	O_1,O_2	O_1,O_3	O_2,O_3
OCV	0.4877	0.9967	0.4877

Finally, the strength of conceptual cohesion of service is obtained.

$$SCV(S_1) = \frac{0.4877 + 0.9967 + 0.4877}{3} = 0.6573$$

Table VII shows the conceptual cohesion of four identified services on CRUD matrix of Table III.

TABLE VII. SCV VALUES FOR IDENTIFIED SERVICES

Service	The value of cohesion (SCV)
S_1	0.6573
S_2	0.6256
S_3	0.2658
S_4	0.0524

VII. DISCUSSION

The results clearly demonstrate that our proposed metric for cohesion appropriately measure conceptual cohesion of a service. Now, we analyze the values which provided by the proposed metric and the previous metrics such as SIDC [6] and TCC [17] and CCM [22]. As explained earlier, the operations of a service must be related in terms of some domain-level concepts. In other words, they must be focused on single business functionality. The analysis shows that semantics in business process are utilized properly in the proposed metric, so it evaluates service cohesion completely in conceptual point of view.

We consider two identified services in CRUD matrix (Table III), highlighted with red and blue colors, as the material for analysis. Each of these services has three operations; their operations and the resulted cohesion value, obtained by three mentioned metrics have been shown in Table VIII. Values shown in Table VIII state the relationship between two operations of each service. Consider group B1 of the first service and group R3 of the second service. For B1 and R3 groups, the SIDC and TCC give the same result value whereas these groups have different cohesion in conceptual point of view.

TABLE VIII. COHESION VALUES OBTAINED BY MENTIONED METRICS

Cluster	Group	EBP Index	Proposed Metric	SIDC	TCC
Blue	B1	1,2	0.48	0.66	0.66
Red	R3	10,11	0.21	0.66	0.66

In B1, Credit and Customer BEs have been accessed together four times (by 1, 2, 3, 9 EBPs of CRUD matrix), these two entities are very related conceptually, because according to [22] two BEs are related if there is at least one shared activity in their behavioral model. In behavioral model of Credit and Customer, there are four shared activities that are processed simultaneously. Therefore, any action on one of them requires an action on the other. In other words, in this organization, whenever an operation performed on Customer, we can expect that an operation must be done on Credit entity. Thus generally, we can associate performing an action on one of them with performing an action on the other as an atomic activity (Create, Update, Read and Deletion of Credit entity and vice versa). On the other hand, existing high cohesion between service operations can be considered as a reusability predictor. This capability is provided by the proposed metric clearly. It is obvious that whenever an operation on a BE is performed with another operation on another BE frequently in enterprise processes, means that performing these two operations together has higher potential reusability. Consequently, it is better to place these two operations, which are considered as an atomic activity, in the same service. Existing of Account Receivable Note in this service (Blue Cluster) results in corruption of this service. Because this BE shares one activity in its behavioral model with behavioral model of other two entities (second EBP of CRUD matrix). Therefore, the cohesion value of 0.48 has been obtained for group B1. In R3 group, two BEs, Issue and Shipping, have been accessed together just two times (by 10,11 EBPs of CRUD matrix), so we can say that these entities are less related in conceptual point of view in comparison with Credit and Customer entities. Moreover, Draft has just one shared activity with two other entities in its behavioral model (EBP11 of CRUD matrix). By conducting similar analysis, there is a lower cohesion between existing operations in R3 in comparison with B2 which our metric shows this point by obtaining cohesion value of 0.21.

VIII. EVALUATION OF THE METRIC

The proposed cohesion metric is analytically evaluated by using property-based software engineering measurement framework [24]. The metric satisfies all of the cohesion properties and therefore it can be a valid measure of cohesion from the measurement theory point of view.

Property 1: **Non-negativity and Normalization** are satisfied because SCV metric never becomes negative under any conditions, and its value will be in [0,1]. Normalization always let the direct and meaningful comparison between strength of services' cohesion.

Property 2: **Null Value** is satisfied because SCV metric gets the value if mutual relationship between all of business entities which are used by operations of a service is 0.

Property 3: **Monotonicity** is satisfied because by adding a related business entity to a pair EBP its overall cohesion is not decreased. In the other words, whenever we add a related business entity to a set of BEs which are accessed by a pair service operations, the cohesion between those two operations will not be decreased.

Property 4: **Cohesive Modules** is satisfied because by joining two unrelated service interface, the resulted cohesion will not be greater than the cohesion of original interfaces. In the other words, the strength of cohesion between operations of two unrelated services will not be greater than the strength of each service, because they access unrelated BEs.

IX. CONCLUSION AND FUTURE WORK

In this paper, using LSI technique, the strength of conceptual cohesion of a service was measured. In this technique, the business entity-business process matrix is formed using existing semantics in business processes and then by applying SVD algorithm on this matrix, the business entity-business entity matrix was resulted so that this matrix represented conceptual relationship between business entities. By adopting business entity-business entity matrix, we can measure the strength of conceptual cohesion of candidate services in the service identification phase. This quality attribute has a great impact on service reusability and maintainability but inherently conceptual nature caused it to be very difficult from quantifying point of view. Therefore measuring this important quality attribute from the conceptually point of view is very valuable. Writers' lack of access to all enterprise processes caused that the effectiveness of the cohesion measuring approach to be shown using a CRUD matrix. Although the obtained results approves the effectiveness of proposed metric well, but having all processes of a real enterprise and then using this metric in the service identification phase completely approves usefulness of this metric. Therefore, using more case studies can be considered as future work of this paper.

ACKNOWLEDGMENT

The project has been partially supported by Iran Education and Research Institute for Information and Communication Technology (formerly Iran Telecommunication Research Center (ITRC)) under contract number 4290/500 and also Shahid Beheshti University under the supervision Automated Software Engineering Research (ASER) Group, Faculty of Electrical and Computer Engineering.

REFERENCES

[1] A. Erradi, N. Kulkarni, and P. Maheshwari, "Service Design Process for Reusable Services: Financial Services Case Study," 5th International Conference on Service Oriented Computing (ICSOC'07), 2007.

[2] M.P. Papazoglou and W.-J. van den Heuvel, "Service- Oriented Design and Development Methodology," International Journal of Web Engineering and Technology, vol. 2, no. 4, pp.412-442, 2006.

[3] M. P. Singh and M. N. Huhns, "Service-Oriented Computing: Semantics, Processes," Agents.John Wiley and Sons, WestSussex, England, 2005.

[4] T. Erl, "Service-Oriented Architecture (SOA): Concepts, Technology, and Design," Prentice Hall, 2005.

[5] J. Bansiya and C. G. Davis, "A Hierarchial Model for Object-Oriented Design Quality Assessment," IEEE Transactions on Software Engineering, 2002.

[6] M. Perepletchikov, C. Ryan, and Z. Tari, "The Impact of Service Cohesion on the Analyzability of Service-Oriented Software," IEEE TRANSACTIONS ON SERVICES COMPUTING, vol. 3, No. 2, 2010.

[7] M. Perepletchikov, C. Ryan, and K. Frampton, "Cohesion Metrics for Predicting Maintainability of Service-Oriented Software," Seventh International Conference on Quality Software (QSIC 2007), 2007.

[8] L.C. Briand, J.W. Daly, and J. Wust, "A Unified Framework forCohesion Measurement in Object-Oriented Systems," FourthInt'l Software Metrics Symp, 1977.

[9] E. Greengrass, "Information Retrieval: A Survey 30 November 2000 ," 2000.

[10] S. Dominich, The Modern Algebra of Information Retrieval, Springer, 2008.

[11] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The Measurement of Textual Coherence With Latent Semantic Analysis," Discourse Processes, 1998.

[12] W. Stevens, G. Myers, and L. Constantine, "Structured Design," IBM Systems J., vol. 13, no. 2, pp. 115-139, 1974.

[13] J. Eder, G. Kappel, and M. Schrefl, "Coupling and Cohesion in Object-Oriented Systems," ACM Conf. Information and Knowledge Management (CIKM), 1992.

[14] C.K. Chidamber and S. R. Kemerer, "A Metrics Suite for Object Oriented Design," IEEE Transactions on Software Engineering, Vol. 20, 1994.

[15] W. Li and S. Henry, "Object-Oriented metrics that predict maintainability," Journal of Systems and Software, 1993.

[16] X. Li, Z. Liu, B. Pan, and D. Xing, "A Measurement Tool for Object Oriented Software and Measurement Experiments with It," In Proc. IWSM 2000. (Lecture Notes in Computer Science 2006, Springer-Verlag, Berlin, Heidelberg, 2001), pp.44-54.

[17] J. M. Bieman and B-Y. Kang, "Cohesion and Reuse in an Object-Oriented System," In Proc. ACM Symposium on Software Reusability (SSR'95), 1995.

[18] G. Gui and P.D. Scott, "Coupling and Cohesion Measures for Evaluation of Component Reusability," MSR'06, 2006.

[19] B. Shim, S. Choue, S. Kim, and S. Park, "A Design Quality Model for Service-Oriented Architecture," 15th Asia-Pacific Software Engineering Conference, 2008.

[20] P. Ingwersen, Information retrieval, Taylor Graham Publishing, 1992.

[21] M. Qian, N. Zhou, Y. Zhu, and H. Wang, "Evaluating Service Identification with Design Metrics on Business Process Decomposition," IEEE International Conference on Services Computing, 2009.

[22] A. Rostampour, A. Kazemi, F. Shams, A. Zamiri, and P. Jamshidi, "A Metric for Measuring the Degree of Entity-Centric Service Cohesion," Service oriented Computing and Application (SOCA), IEEE, 2010, in press.

[23] S. Kumaran, R. Liu, and F. Y. Wu, "On the Duality of Information-Centric and Activity-Centric Models of Business Processes.", "Springer, CAiSE", pp. 32–47, 2008.

[24] L. C. Briand, S. Morasca, and V. R. Basili, "Property-Based Software Engineering Measurement.", "Transactions on Software Engineering", Issue. 1, Vol. 22, 1996.

[25] A. Rostampour, A. Kazemi, F. Shams, P. Jamshidi, and A. Nasirzadeh Azizkandi, "Measures of Structural Complexity and Service Autonomy," International Conference on Advanced Communication Technology, IEEE, 2011, in press.