

ULRR

An information retrieval based approach for measuring service conceptual cohesion

Item Type	Meetings and Proceedings
Authors	Kazemi, Ali;Rostampour, Ali;Zamir, Amin;Jamshidi, Pooyan;Haghighi, Hassan;Shams, Fereideon
Citation	11th Interntional Conference on Quality Software (QS/C 2011);07/2011
Publisher	IEEE Computer Society
Download date	2026-04-16 08:45:14
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/1729

An Information Retrieval Based Approach for Measuring Service Conceptual Cohesion

Ali Kazemi¹, Ali Rostampour¹, Amin Zamiri¹, Pooyan Jamshidi², Hassan Haghighi¹, Fereidoon Shams¹

¹Automated Software Engineering Research Group

ECE Faculty, Shahid Beheshti University GC, Tehran, Iran

{Ali.Kazemi, A.Rostampour, A.Zamiri}@mail.sbu.ac.ir, {h_haghighi, F_Shams}@sbu.ac.ir

²Lero - The Irish Software Engineering Research Centre,

School of Computing, Dublin City University, Dublin, Ireland

pooyan.jamshidi@computing.dcu.ie

Abstract—High cohesion as a desirable principle in software design has an incredible impact on software reuse, maintenance and support. In service-oriented architecture (SOA), the focus of services on single business functionality is defined as conceptual cohesion. Current metrics for measuring service cohesion reflect mostly the structural aspect of cohesion and therefore cannot be utilized to measure conceptual cohesion of services. Latent Semantic Indexing (LSI), on the other hand, is an information retrieval technique and is widely used to measure the degree of similarity between a set of text based documents. In this paper, a metric namely SCD is proposed that measure the conceptual cohesion of services based on LSI technique. This metric consider both service functionality and operation sequence to measure the conceptual cohesion. An evaluation of the metric based on a set of cohesion principles and comparison with the previously proposed metrics are also provided.

Keywords: Software metric, service conceptual cohesion, service-oriented design principle, Latent Semantic Indexing

I. INTRODUCTION

Service-Oriented (SO) brings specific constraints and requirements to the design of software systems [1]. In contrast to earlier paradigms which treat an application as a collection of inter connected procedures or objects, SO software applications are developed in terms of reusable, stateless services which aim to demonstrate autonomy from other services in the system [2]. To be more specific, the procedural paradigm has only one main level of abstraction, a procedure, whereas the OO paradigm has two levels of abstraction, methods which are aggregated into classes. In contrast, SO introduces an additional level of abstraction and encapsulation, a service in which operations are composed into elements such as OO classes, business process scripts, and procedural packages that implement the functionality of the service as exposed via operations in the service interface [1].

Although SO and its associated computing paradigm, service-oriented computing (SOC), is becoming an increasingly popular paradigm for the implementation of enterprise software [3], SO software applications are often designed in an ad hoc manner [1] [4], with little consideration given to the underlying design structures, thereby potentially resulting in decreased maintainability of the produced software.

Service-orientation urges the services to be reused regardless of the fact that there might not be any direct requirement for service reuse. The chance of a service to accommodate future requirements with minimum development efforts increases using design standards which makes every service potentially reusable [2]. Cohesion is a principle that should be considered during all stages of service-oriented design and development. High cohesion increases clarity and comprehension of the design and thus simplifies software maintenance [5]. On the other hand, by putting related operations in one service, reusability in different contexts is improved, since it enables the service to focus on one single business functionality [6].

Cohesion, due to its inherently conceptual nature, is considered as one of the most complex structural software attributes to quantify [1]. This quality attribute could be measured based on the conceptual (semantic) relationships of operations within a service. Current cohesion metrics have only considered the structural aspects so far [1, 6, 8], and have ignored the conceptual aspects. Each service encapsulates specific business functionality, so service conceptual cohesion could be defined as the level of concentration of a service on a single functionality or semantically related business functionalities.

In this paper, we propose a method for measuring the degree of conceptual cohesion in a service. This method evaluates the operations of the service in both functionality and sequential aspects. To measure the conceptual cohesion we use LSI technique. LSI was first used for information retrieval purposes [8]. One of the applications of this technique is calculation of text cohesion [8]. LSI provides a completely automatic approach that compares information units in order to measure conceptual relationship. Conceptual relatedness degree of units measurement is based on a powerful mathematical method called Singular Value Decomposition (SVD) [8].

We utilize business processes in order to measure service conceptual cohesion, which is aligned with the nature of services as a business concept. The proposed method uses business processes as input and then computes the relationships between business processes and their related business entities using a matrix structure. By applying the SVD and then domain reduction algorithm over this matrix, the introduced metric could be calculated.

The rest of this paper is organized as follows. In the next two sections a summary of most related work together with definitions of adopted basic concepts are provided. Section IV is dedicated to the applicability of LSI in cohesion metric. Detailed description of the proposed metric is provided in Section V, followed by an evaluation in section VI. Summary of the work and future research directions are discussed in the final section of this paper.

II. RELATED WORK

In this section, a brief overview of the metrics proposed for measuring cohesion in object-oriented and service-oriented paradigms is presented.

Many metrics have been proposed for cohesion measurement in object-oriented design. These metrics could be applied using either high-level or low-level design. In high-level design, the metrics should use information like classes and method interface since in this level only such information is available. Metrics in low-level design make use of fine-grained information like algorithm analysis, source code and identifier properties. Thus in low-level design all method-method, method-attribute and attribute-attribute interactions could be used. On the other hand, defining metrics that are applicable in high-level design have a greater advantage in contrast to metrics that are only applicable in low-level design because they help to detect more characteristics related to cohesion at early stages of development. Therefore, class cohesion improvement during high-level design reduces development cost and increases software quality.

Cohesion metrics in object-oriented are based on sharing class attributes. Some metrics in this paradigm measure the lack of cohesion. For instance, Kemerer and Chidamber in [9] proposed a metric named LCOM1 that counts the number of method pairs without any shared variable and then they provided another version named LCOM2 which calculates the difference between the number of method pairs that do and do not share instance variables [10].

Li and Henry proposed a graph based approach in which each method is considered as a graph node. In this approach there is an edge between two nodes if their corresponding methods have at least one shared variable [11]. Their metric, named LCOM3, defined as the number of connected components of the graph. In LCOM4 which extended LCOM3 metric, there is an edge between two nodes if one of their corresponding methods invokes the other [12]. In addition, Henderson-Sellers [13] proposes a lack-of-cohesion in methods metric, LCOM5, that considers the number of methods referencing each attribute.

In [14] Bieman and Kang proposed two cohesion metrics namely TCC and LCC. TCC considers two methods to be related if they share the use of at least one attribute. However, LCC considers two methods to be connected if they share the use of at least one attribute directly or transitively.

In [15] class cohesion measurement (WTCoh) is calculated by the similarities of class methods. Two methods are similar when the variable sets that they can access are overlapped. This study categorizes the cohesion of class methods to either direct or indirect. The transitivity property has been used to measure the indirect cohesion of classes in this study.

The semantic information shared between the source code elements is the basis of conceptual cohesion measurement (C3 metric) in [16] where a set of metrics are proposed. Comments and identifiers as semantic information in IR methods are used to calculate conceptual cohesion. The research assumes that each method owns a set of explanatory comments. Moreover, the use of meaningful identifiers to implement classes is another pre-assumption in this research.

Most metrics in object-oriented paradigm measure the cohesion only in structural point of view and does not consider the conceptual aspect of it. C3 Metric could not also be used to measure the conceptual cohesion of services since:

1. The proposed metrics are calculated using source code information which are not available for services in the design level.
2. Service is a notion at the business level; hence, the required information can only be obtained using business level artifacts, such as business processes.

Regardless of the service implementation, cohesion of the services could be defined based on their functionalities. This means that conceptual cohesion should be calculated based on the scope of the functionalities. The required semantic information could be found in enterprise business processes. Therefore, the conceptual cohesion could be calculated without considering implementation details. More discussion is provided in [1, 6]. Although numerous metrics has been proposed to measure class cohesion in service-oriented design, but the number of similar researches in service-oriented architecture is scarce. These studies are briefly reviewed in this section.

In [1, 6] eight semantic categories of service-oriented cohesion are defined: Coincidental, Logical, Temporal, Communicational, External, Implementation, Sequential and Conceptual. Four of these categories (Communicational, External, Implementation and Sequential) are presented as quantifiable cohesion categories. Moreover, some measurement metrics are proposed the summary of which are provided in Table I. On the other hand, four other categories Coincidental, Logical, Temporal and Conceptual are defined as purely semantic. The proposed research indicates that latter four categories require service semantic while the former four categories could be measured without having any semantic data.

In [5], the service cohesion is measured based on the data flow complexity between service operations. The data flow cohesion is calculated based on the complexity of information entities sent through data flows. In the

mentioned research, a service containing operations with complex data flow is considered as coherent. The proposed metrics calculate only the structural cohesion of services. Moreover, the complexity of information entities is calculated based on experience.

A brief representation of the cohesion metrics has been shown in Table I.

TABLE I. SUMMARY OF COHESION METRICS IN THE LITERATURE

	Name	Definition
Object-Oriented	LCOM1	Number of non-similar method pairs in a class of pairs.
	LCOM2	the difference between the number of method pairs that do and do not share instance variables.
	LCOM3	Number of connected components in the graph.
	LCOM4	The LCOM3 metric is extended in [12] by adding an edge between a pair of methods if one of them other.
	LCOM5	The number of methods referencing each attribute.
	TCC, LCC	Ratio of number of similar method pairs to total number of method pairs in the class.
	WTCoh	Number of used shared data entities by methods and also taking the transitive cohesion into account.
	C3	Use the semantic information shared between the source code elements (comments and identifiers)
Service-Oriented	DM IAUM [7]	Number of system services divided by the total number of used messages
	SIDC [6]	This metric is proposed to measure communicational cohesion and takes the return type parameters into account.
	SIUC [6]	This metric indicates that a service is deemed to be Externally cohesive when all of its service operations are invoked by all the clients of this service.
	SIIC [6]	In this metric a service is deemed to be Implementation cohesive when all of its service operations are implemented by the same implementation elements.
	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.
	V _{COHES}	The data flow complexity across activities in service to reflect the functional relevance among them.

Generally one could claim that metrics in previous researches have considered this quality attribute from the structural point of view. These measures could indicate that how one class (service) is created and how the instances are working together to address their design goal. Therefore, they cannot show that how a class (service) is coherent in conceptual point of view. In other words, what is the level of service concentration on a single business functionality can not derived based on the previous proposed metrics. It is clear that measuring service cohesion in early stages of service based software development cycle reduces cost greatly. In addition, whenever a service implements one domain concept it could be easily reused and maintained. In this paper, we propose a metric that measures the conceptual cohesion using the semantics hidden in business processes. We have utilized an information retrieval approach to uncover the required semantics.

III. SETTING THE SCENE

In this section, we first present basic concepts which will be referenced throughout the rest of the paper and then give a brief description of the LSI technique.

A. Basic Concepts

Definition 1 (Business Entities): 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. The data model describes the data dependencies between the dominant entity and the dominated entities as the former logically containing the latter [19].

Definition 4 (Conceptual Cohesion): One could say that between all operations of each service a semantic relationship based on some domain-level concept could be identified. This means that either single business functionality or some other semantically meaningful concept is the focus of the operations of a service [6].

B. Overview of Latent Semantic Indexing

In the traditional approach to information retrieval (IR), a vector space is defined for a collection of documents such that each dimension of the space is a term occurring in the collection, and each document is specified as a vector with a coordinate for each term occurring in the given document. The value of each coordinate is a weight assigned to the corresponding term, or a measure of the importance of the given term in characterizing the given document and distinguishing it from the other documents in the given collection [8]. 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 $A_{n \times m}$ 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 . The following steps describe the LSI technique in more details:

1. A matrix is constructed in which each row corresponds to a term that occurs in the document and each column corresponds to a document. Each element (m, n) in the matrix corresponds to the weight of the term m in document n . Figure 1 demonstrates a term-document matrix.

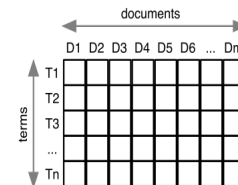


Figure 1. The Term-Document Matrix

2. A weight is assigned to each term in the document. Different weighting methods are proposed in [8] in details. The simplest weighting model could be achieved by calculating the number of term occurrence in document. In this paper we propose a new weighting model since previous models could not be applied directly in our approach.

3. The term document matrix A is decomposed to three matrixes S , T and D using the SVD method. Matrix S , T and D contain the information on terms, unique values and documents respectively. The original matrix A could be using the formula $A = TSD^T$ where D^T is the transpose of matrix D .
4. The matrices S , T and D are reduced to k domains. The value for k is 2 in this approach, as suggested in [8]. With $k=2$, the reduced matrix $BBR = T_2S_2(T_2S_2)^T$ is achieved. The BBR matrix demonstrates the relationship between business entities and is used in our proposed metric.

The LSI technique uses the term co-occurrence to obtain relationship between terms. Two terms a and b are co-occurring if there is at least one document which contains both of them. We say that, in this case, a and b have a first order co-occurrence. Similarly, a second order co-occurrence path between two terms a and b is made with a term c such that a co-occurs with c and b co-occurs with c too. The number of unique terms c is the number of second order co-occurrence paths between a and b . Higher order co-occurrences are defined in a similar form. SVD algorithm considers all information about co-occurrence of terms.

IV. APPLICABILITY OF LSI IN MEASURING COHESION

The contribution of this paper is to propose a method to measure the conceptual cohesion of services. To achieve this objective, we map the mentioned concepts in section II to SOA concepts. Similar to LSI, we define Business Entity - Elementary Business Process (BE-EBP) matrix, where m is the number of business entities and n is the number of elementary business processes. Figure 2 shows a process which has access to eight business entities, Customer, Vehicle, Work Order, Offering, Skills, Shipping Durations, Appointment and parts [5]. This business process is placed in one of BE-EBP matrix columns. Business entities are also placed in rows of the mentioned matrix. This matrix should be completed for all enterprise business processes. Having completed the BE-EBP matrix, we should give values to its elements using weighting models. We will use a specific weighting model in this paper. This model must be able to reflect semantic relationships that existed in business processes. In the following two sub-sections, key aspects of conceptual cohesion have been examined. In section *A*, the functionality aspect and in section *B* the operation sequence aspect is discussed.

A. The Functionality Aspect of Cohesion

In the LSI technique, term co-occurrence information is used; two terms that co-occur several times in documents are more related. According to [19, 22], we call two business entities are related if both are accessed or processed by at least one business process. The relationship between business entities can be specified based on the type of the actions performed by processes on these entities. For example, the action of creating a business entity has a higher affinity to the business process in comparison to a reading action. Generally, we can consider priorities $C > U > D > R$ for actions which are performed on business entities, where C ,

U , D and R refer to Create, Read, Update and Delete respectively [22]. Using this idea, we can propose a weighting model which explains the exact relationship between business entities. Keeping in mind the mentioned priorities, we can consider the degree of relationship between business entities using some simple rules like the following samples:

- Two business entities which are processed by a business process have the highest relationship degree if this business process performs Create action on both. In other words, the affinity of these two business entities to the business process is very high, and consequently the relationship between them is very strong. We consider this group as the strongest type of relationship.
- Based on the degree of relationship, two business entities are placed in the second group if the business process performs action C on one of them and action U on the other one. Since totally 4 actions C , R , U , and D are performed on every BE, 7 relationship degrees can be obtained between two BEs. These seven degrees are shown in Table II.

TABLE II. THE PRIORITIES OF ACTIONS

Action	Create	Update	Delete	Read
<i>Create</i>	1			
<i>Update</i>	2	3		
<i>Delete</i>	3	4	5	
<i>Read</i>	4	5	6	7

Numbers inside Table II are representative of the categories of different degrees of relationships. For example, the value 1 shows the strongest type of relationship whereas the value 7 shows the weakest type of relationship because the action of reading two business entities by a process, does not provide tight relationship between those two business entities. The target weighting model must completely reflects these priorities.

In the LSI technique, we can easily create a term-document matrix by counting the number of occurrences of a term in a document. In this technique, the most important thing is occurrence of terms in a document, and there is not any difference between occurrences of two different terms whereas in our proposed method the type of the action which is performed on business entities has a high level of significance (this is the reason that we utilize a new weighting model in our method). Hence, we fill the elements of BE-EBP matrix based on the amount of affinity between business entities and business processes. Regarding the priorities of actions, we give weights of 4, 3, 2 and 1 to C , U , D and R , respectively. This way of weighting perfectly reflects the categorization mentioned in Table II.

B. The Operation Sequence Aspect of Cohesion

Service operations are sequentially related if either the output or post-condition from one operation serves as the input or pre-condition for the next operation [6]. Sequential dependency between the operations of a service causes a sequential control flow in service to satisfy specific business functionality. Choreography of business process is performed according to its control flow. When the operations

of a service are sequentially dependent, it encapsulates a domain concept without the need of calling another service, thus focusing on a single functionality. This is the conceptual cohesion in definition.

A sequence flow is used to show the order of activities which are executed in a business process. Every flow has only one source and one target. Source and target could be the following flow objects: Events (Start, Intermediate, and End), Activities (Task), and Gateways.

In standard business process modelling languages, sequence flow has been shown in different ways and has various types. For example, in BPMN different flows such as normal sequence flow, conditional sequence flow, and default sequence flow are existed. Therefore, using the information which exists in business process diagrams, the degree of sequential relation between activities could be obtained. In this way, if two business activities are related sequentially should be placed in the same service.

In order to measure this aspect of cohesion, we first categorize the sequence flow between the activities of a business process and then give each category a weight. To do so, the OMG documentation has been used [23].

1. If there is normal sequence flow between two activities, sequence flow of them is at highest degree of relation. The logic behind this rule is that the normal sequence flow represents the sequence of flow elements in a business process. This means that the control of the flow is passed from on activity to another. Therefore the functionality of business process is realized whenever these two activities are performed sequentially. Thus by putting these two activities in one service, service cohesion in sequential view is improved.
2. If there is a parallel gateway between two activities, their sequential flow is in the second category of relation degree. The logic behind this rule is that the Parallel Gateway is used to synchronize multiple concurrent branches (merging behavior). Therefore if these activities are in one service, service cohesion in sequential view is improved since we are certain that these activities should be performed sequentially in order for their functionality to be realized.
3. If there is an inclusive gateway between two activities, their sequential flow would be in the third category. This is because the inclusive gateway selects a subset of alternative flows. Since a flow path might not be selected in some conditions, the weight of this category is less than the one of parallel gateway. On the other hand, because all the conditions are evaluated, their weight is bigger than exclusive gateway.
4. If there is an exclusive gateway between two activities the weight of their sequential flow is the least among other categories. This is because the exclusive gateway selects only one flow between the alternative flows and does not consider any priority among them. This means that in specific conditions, there is a sequential flow between these two activities. Therefore putting these two activities in a service brings about the least degree of cohesion in sequential point of view.

Considering above rules, four categories of sequential flow could be defined, as mentioned in Table III.

TABLE III. FLOW ELEMENT WEIGHTS

Flow elements	category	weight
<i>Sequence Flow Normal</i>	1	4
<i>Parallel Gateway</i>	2	3
<i>Inclusive Gateway, Complex Gateway</i>	3	2
<i>Exclusive gateway, Event-Based Gateway</i>	4	1

As mentioned earlier, the LSI technique uses term co-occurrence concept to obtain relationship between terms. The information about the sequence of operations should be contained in the BE-EBP matrix. In order to do this, whenever the actions of two BE are sequentially related, their weight in the matrix is increased by the weights. Therefore, the business entities that are processed sequentially for many times together, gain a higher relationship degree. This method of weighting is completely aligned with the co-occurrence concept in IR methods. In fact we have maintained the sequential relationship of actions by giving more weight to them.

C. The BE-EBP Matrix Weighting

To explain the weight calculation of BE-EBP matrix, consider the business process shown in Figure 2. This process has access to eight BEs, Customer, Vehicle, Work Order, Offering, Skills, Shipping Durations, Appointment and parts. The column related to this process in BE-EBP matrix has been shown in Figure 3. For example, customer is created onetime. Therefore, its weight equals to $1 \times 1 = 1$. As another example the Work Order BE is created onetime and updated one time and its weight is calculated as: $1 \times 4 + 1 \times 3 = 7$. The calculated weight values for other entities are depicted in the left matrix in Figure 3.

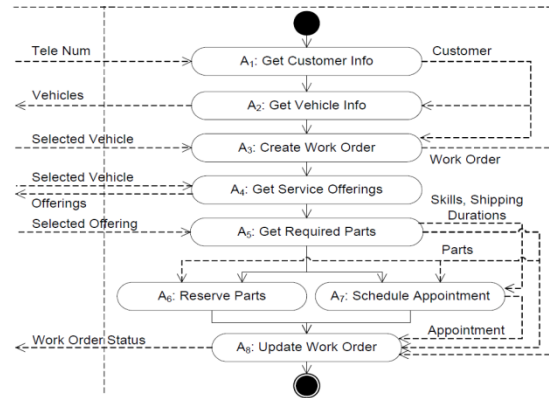


Figure 2. Business process of automotivework order scenario [5]

The sequential relationship between two business entities could also be identified. For example, the customer and vehicle as two BEs are sequentially related since in the business process the operation 1 first reads the customer entity and then the operation 2 reads the vehicle entity. Therefore, their weight is increased by 1 unit to reflect their relationship (In this paper, for the sake of simplicity, we have considered the weight of 1 for normal sequence flow). The

right matrix in Figure 3 shows the calculated values for all BEs considering both of the mentioned aspects.

BE	EBP		BE	EBP	
	1	2		1	2
Customer	1		2		
Vehicle	1		3		
Work Order	7		10		
Offering	1		3		
Shipping	1		1		
Appointment	3		5		
Skills	1		1		
Parts	4		6		

Figure 3. The BE-EBP Matrix

The above matrix can be completed for all of enterprise business processes so that the number of its columns equals to the number of enterprise business processes, and its rows equals to the number of enterprise business entities. Now, performing SVD, three matrices T, S and D can be obtained, where $A = TSD^T$ (A is the BE-EBP matrix). Then, considering $k=2$, the reduced matrix T is formed. The BBR matrix $BBR = T_2S_2D_2$ indicates relationships between business entities. Values of the elements in this matrix are not normalized, and they can even be negative values. To solve this problem, negative values were replaced with zero values because negative values are not meaningful when measuring service cohesion, and zero means lack of cohesion. In addition, to normalize values, we multiply this matrix by $1/Max$ where max is the maximum value in BBR matrix. Thus, using the LSI concept, it is possible to show semantics existing in a business process in the form of BBR matrix. This matrix has been used to obtain the conceptual cohesion between services.

V. THE PROPOSED SERVICE COHESION METRIC

The proposed metric in this section can be used to measure cohesion of a service in service-oriented design based on the semantic relationships of operations exposed in its interface. It should be mentioned that the proposed metric has been defined based on an absolute scale where the metric takes values in the range of 0 to 1. Value 1 indicates the strongest possible cohesion while value 0 indicates lack of cohesion.

The main application of the proposed metric is in the service identification phase, the first step in the modeling phase of SOA lifecycle [11]. Service identification is considered to be one of the most critical phases in service-oriented projects success, since the business requirements are covered in this phase [5]. The purpose of this phase is to produce a set of candid services together with their operations. Business process decomposition is a service identification approach that aims to partition business activities into services [5]. Thus a set of business processes are available from which one could identify the services. As it has been stated in section III, to measure cohesion in a service using the proposed metric, first the BE-EBP matrix must be formed. We can create this matrix based on business processes which are to be decomposed to obtain services. Figure 4 shows the position of the proposed metric in the process of measuring method.

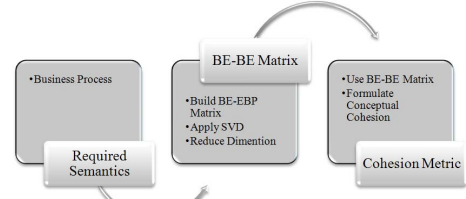


Figure 4. The Proposed Method for Measuring Conceptual Cohesion of Services

As it has been shown in Figure 4, initially the required semantics are obtained from business processes. These semantics include information about co-occurrence of actions and their types. Then, BE-EBP matrix $A_{m \times n}$ is created, where m is the number of enterprise business entities and n is the number of enterprise business processes. Then, by adopting the introduced weighting model, each element of A takes a weight. This weight describes the importance degree of i th business entity in j th business process. We apply SVD on A to obtain conceptual relationship between business entities. The outputs of this phase are three matrices consolidated as $A = TSD^T$. The matrix $BBR = T_2S_2(T_2S_2)^T$ describes the conceptual relationship between business entities which must be used to measure the degree of service cohesion.

To form the BE-EBP matrix $A_{m \times n}$, the weighting model mentioned earlier should be followed in order to fill the elements of the matrix. We consider both functional and sequential relationship between service operations. For the functional aspect one should first obtain the number of times that business process j access business entity i. The type of the action which is applied on the business entity is also important to weight the elements of matrix A. The sequential relationship of business entities is also taken into account, as discussed in previous chapter. Finally, by applying the SVD algorithm the output is consisted of three different matrices T, S and D where $A = TSD^T$.

To formulize our metric we use a graph-based approach. Consider service S with a set of operations $O = \{O_1, O_2, \dots, O_m\}$. Each operation O_j of service S accesses a set of business entities that are shown as $BE_j = \{BE_{j,1}, BE_{j,2}, \dots, BE_{j,n}\}$. For each pair of operations O_i and O_j in service S a complete graph $G = (V, E)$ is formed such that $V = BE_i \cup BE_j$. A value is then assigned to each edge in set E that shows the degree of relationship between business entities that are represented as graph G nodes. This degree of relationship could be obtained from the BBR matrix. The conceptual relationship between two operations i and j is calculated as follows:

$$ORD(i, j) = \frac{\sum_{p \in V} \sum_{q \in V} (BBR_{p,q})}{|V| \times (|V|-1) / 2} \quad (1)$$

Where:

- p and q are the identifying number for their corresponding business entities.
- $BBR_{p,q}$ is the degree of relationship between business entities BE_p and BE_q .

- $|V|$ is the cardinality of set V .
- The denominator of the fraction in this formula is the number of edges of complete graph G .

The degree of cohesion of a service is consequentially defined (and visually illustrated in Figure 5) as the degree of relationship between its operations as follows:

$$SCD(S) = \begin{cases} \frac{\sum_{i \in O} \sum_{j \in O (ORD(i,j))}{i > j}}{m \times (m-1) / 2} & m > 1 \\ 1 & m = 1 \end{cases} \quad (2)$$

where, m is the number of operations in service S .

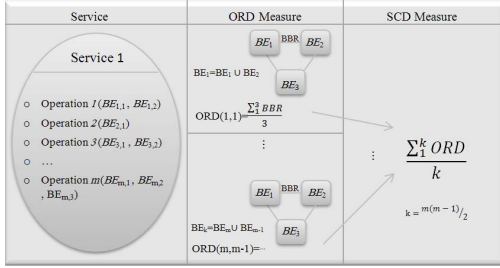


Figure 5. Metric Calculation Conceptual Diagram

VI. EVALUATION OF THE METRIC

In this section, an evaluation of the metric SCD is provided. Firstly, service cohesion calculation using the proposed metric is described in a case study. Using the cohesion principles, it is shown that our metric is a valid measure of cohesion in the measurement theory point of view.

A. Case Study

Service cohesion calculation method is described in this section by an example. The effectiveness of the approach is evaluated using a real-world scenario in a purchase order business process that their scenario has been described in [22].

CRUD matrix could be used to identify services [24]. Figure 6 depicts a CRUD matrix in which the identified services are colored distinctively.

EBP \ BE	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

Figure 6. The CRUD Matrix for Sales Department Scenario [16]

The operation input and output messages, can be extracted from the BEs that the EBP deals with, through one of the following semantic relationships [20].

- Creation (C) of a BE by an EBP results in an output message by the service through the corresponding operation.
- Reading (R) a BE by an EBP means an input message to the operation of the service.
- Updating (U) a BE by an EBP includes an input message to the service, and the updated information as an output message.
- Deletion (D) of a BE by an EBP requires an input message to the service.

For service 1 (the blue cluster), operations input and output messages shown as Figure 7:

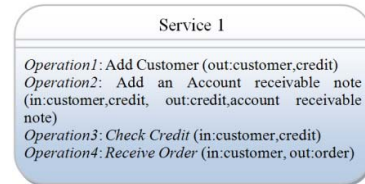


Figure 7. Service Interface for Service 1

At this time, the BE-EBP matrix could be formed using the method described in section III. The rows of this matrix represent business entities and the columns are elementary business processes. This matrix is shown in Figure 8 where its elements are filled using the weighting model in section III. It is worth pointing out that the sequence flow between the actions of each EBP is a normal flow. In other words the actions of CRUD, each of which are equivalent to an activity in business processes, are run sequentially (have a normal sequential flow). In this paper, for the sake of simplicity, we have considered the weight of 1 for normal sequence flow.

EBP \ 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	5	2	2	2	0	0	0	0	2	0	0	0	2
Credit	5	5	3	0	0	0	0	0	3	0	0	0	0
Account receivable note	0	6	0	0	0	0	0	0	0	0	0	0	0
Order	0	0	2	5	2	2	2	0	3	0	0	0	0
Discounts	0	0	0	0	2	0	2	4	0	0	0	0	0
Invoice	0	2	0	0	0	0	0	0	5	2	2	0	3
Shipping schedule	0	0	0	0	0	0	0	0	0	5	3	0	0
Draft	0	0	0	0	0	0	0	0	0	0	5	0	0
Inventory	0	0	0	0	0	2	0	0	0	0	0	4	5
Warehouse voucher	0	0	0	0	0	0	0	0	0	0	0	0	5

Figure 8. The BE-EBP Matrix

Once the BE-EBP matrix is formed, the SVD algorithm could be applied. In order to do this we implement the measuring process in Matlab version 7.6.0.324 software. The BE-BE matrix is then constructed using the formula $BBR = T_2 S_2 (T_2 S_2)^T$. The resulting matrix is shown in Figure 9. This

matrix is normalized and its negative values are replaced by zero.

EBP \ BE	Customer	Credit	Account receivable	Order	Discounts	Invoice	Shipping schedule	Draft	Inventory	Warehouse voucher
Customer	1.00									
Credit	0.71	1.00								
Account receivable note	0.32	0.47	1.00							
Order	0.31	0.36	0.16	1.00						
Discounts	0.02	0.02	0.00	0.01	1.00					
Invoice	0.42	0.40	0.15	0.26	0.01	1.00				
Shipping schedule	0.05	0.00	0.00	0.05	0.00	0.13	1.00			
Draft	0.03	0.00	0.00	0.03	0.00	0.09	0.04	1.00		
Inventory	0.12	0.00	0.00	0.12	0.01	0.38	0.17	0.11	1.00	
Warehouse voucher	0.10	0.00	0.00	0.10	0.00	0.28	0.12	0.08	0.35	1.00

Figure 9. The BE-BE Matrix after decomposition and normalization

Figure 6 shows a CRUD matrix with four identified services. The calculation of the first service cohesion (with blue color) is performed as follows. This service has four operations: Add Customer, Add an Account receivable note, Check Credit and Receive Order.

$$O = \{O_1, O_2, O_3, O_4\}$$

$$BE_1 = \{Customer, Credit\}$$

$$BE_2 = \{Customer, Credit, Account\ receivable\ note\}$$

$$BE_3 = \{Customer, Credit, Order\}$$

$$BE_4 = \{Customer, Order\}$$

In order to obtain the conceptual relationship between service operations a graph should be constructed. This graph for operations O_1 and O_2 is depicted in Figure 10.

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

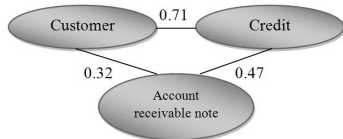


Figure 10. Business Entities Graph for Service O_1 and O_2

$$ORD(1,2) = \frac{0.71 + 0.32 + 0.47}{3} = 0.50$$

Calculation result for other operations is shown in Table IV:

TABLE IV. ORD VALUE FOR SERVICE 1

O_i, O_j	O_1, O_2	O_1, O_3	O_1, O_4	O_2, O_3	O_2, O_4	O_3, O_4
Metric						
ORD	0.50	0.46	0.46	0.38	0.38	0.46

Finally, the conceptual cohesion of the service is calculated as follow:

$$SCD(S_1) = \frac{0.50 + 0.46 + 0.46 + 0.38 + 0.38 + 0.46}{6} = 0.44$$

Table V contains the calculated conceptual cohesion values of the other services:

TABLE V. SCV VALUES FOR THE IDENTIFIED SERVICES

Service	Metric	Proposed Cohesion Metric Value (SCD)
S_1		0.4400
S_2		1.0000
S_3		0.1300
S_4		0.1800

B. Analytical Validation

There are different methods for metric theoretical validation. Some of them have subjective nature whereas others have the basis of axiomatic or calculation theory. Among them, the property-based software engineering measurement framework [25] is widely used; hence, we make use of it to validate the introduced SCD metric.

Property1: **Non-negativity and Normalization** is satisfied since the metric can only take values between 0 and 1.

Property 2: **Null Value** is satisfied since the metric SCD is 0 when the number of relationships between the business entities that are used by the operations of a service is 0.

Property 3: **Monotonicity** is satisfied because adding another related BE to a pair of EBP does not decrease the overall cohesion. In other words the overall cohesion is not decreased by adding a related BE to a set of BEs that are accessed by a specific pair of operations.

Property 4: **Cohesive Modules** is satisfied by joining together two unrelated service interface the resulting cohesion would not be bigger than the maximum of original relations. In other words, the degree of cohesion between the operations of two unrelated services is not greater than the cohesion of each of them, because their operations access unrelated business entities.

Therefore, the proposed metric satisfies all of the cohesion properties and therefore it can be a valid measure of cohesion from the measurement theory point of view.

C. Discussion

In this section, the resulted values of SCD, that are measured and presented in Table IV, are critically discussed. We will provide a comparison of the assigned cohesion values with the output of the other proposed metrics in service-oriented architecture such as SIDC [6] and V_{COHES} [5]. A comparison of the improvements in this study to our previous metric namely SCV introduced in [26] is also discussed.

As mentioned earlier, a service should encapsulate single business functionality or in other words, the operations of a service must be related in terms of some domain-level concepts. A service encapsulating such functionality is said to be conceptually cohesive. We will show that the proposed metric uses the semantic in business processes effectively and could be used to measure conceptual cohesion of services.

Consider service 1 highlighted with blue color in Figure 6. This service has four main operations (EBP1, EBP2,

EBP3, and EBP4). The operations of this service should be related in terms of correspondence to one business functionality. The degree of cohesion between two groups of operations of this service using the mentioned metrics is provided in Table VI.

TABLE VI. COHESION VALUES OBTAINED BY MENTIONED METRICS

Group	Operations	The Proposed Metric	SIDC	V_{COHES}	SCV
G1	O_1, O_2	0.50	0.66	0.33	0.48
G2	O_1, O_3	0.46	0.66	0.33	0.87

The metrics SIDC and V_{COHES} both give an equal value for the cohesion of the operations, while the relation of these operations completely differs in conceptual point of view. The business entities, Customer and Credit, are processed by the two groups of operations G1 and G2. However, they process the following entities separately: Account Receivable Note by G1 and Order by G2 which causes different levels of cohesions. Because based on [19, 22] we consider two business entities related when they are accessed by at least one business process. More precisely, they should at least have one common activity in their behavioral model. The business entity Account Receivable Note is only related to Credit and Customer through EBP2. While the business entity Order is related to business entities Credit and Customer three times through EBPs 3, 4 and 9. In other words, they have three shared activities in their behavioral models. Therefore, any action on one of them requires an action on the other. This means that whenever an operation performed on Customer or Credit, we can expect that an operation must be done on Order entity. Thus, we can associate performing an action on one of them with performing an action on the other as an atomic activity. 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 certain that when a set of business entities are more frequently accessed by business processes, performing operations over these entities has higher reuse potential.

The calculated values for the mentioned groups are 0.5 and 0.46 respectively, using the proposed metric. These metrics justifies the above statements. The crucial question here is why the cohesion value of G2 has been reduced although Order has three shared activities with Credit and Customer in its behavioral model (In SCV metric this value was 0.87 while in the proposed metric it is 0.46). More interesting the cohesion value of G2 is even increased by 0.0.2. The answer is that in this paper the type of actions is efficiently used in cohesion measurement. Although the number of times that two business entities are accessed by business processes at the same time has a direct impact on their conceptual relation, the type of the action performed by business processes over these entities is crucial. In our previous metric [26], we defined two business entities to be related whenever there was at least one business processes that process them. But the type of operation performed by the business process is also important. For example, performing Create over two business entities denotes a

stronger relationship with a Read action over the same entities.

In G1, EBP1 performs actions Read, Update and Create on business entities Customer, Credit and Account Receivable Note respectively. The EBP1 do the action Create on Account Receivable Note and Update on Credit which denotes a much stronger relationship between them. On the other hand, in group G2, as discussed earlier, the business entity Order is three times related to customer and credit by EBPs 4, 3 and 9. The strongest relationship between these three entities is in EBP4 (receive order) where entity Order is created and entity Customer is read, and makes this relationship much weaker than the relationship between three business entities in G1 (through Add Customer an Account Receivable Note). In our previous metric where only the number of shared activities in the behavioral model where considered, the value 0.48 for G1 and 0.87 for G2 was obtained. In the new metric by considering the type of action these values are 0.5 and 0.46. Therefore, in G1 the value of cohesion is increased. In G2, the value of cohesion is decreased to 0.46 since there is a weak read relation between entities customer and credit with entity Account Receivable Note (through tasks Check Credit and Issue Invoice) and also a weak create relation (through Receive Order).

Also, we did not consider the sequential aspect in our previous methods. The effect of actions sequence on service cohesion could be further evaluated with similar analyses. For example, in group G1 the business entity Account Receivable Note and Credit are sequentially related only once through EBP2. While in Group G2 business entity Order and Credit are sequentially related twice through EBP4 and EBP9. Even without considering the type of actions (in order to have equal comparative conditions with SVC metric) the cohesion of G2 in comparison with G1 is increased. This means that the proposed metric takes the sequence of actions into account as well. Considering the performed analysis about the proposed metric and also its comparison with previous metrics and our previous metric, it is clear the new metric in this paper addresses the shortcomings of previous metrics (especially the effect of action types on business entities and also the action sequence). This metric could be used as suitable metric for conceptual service cohesion measurement.

One might assert that this analysis covers just direct relationships between two BEs such as the work reported in [26], whereas they might be related through intermediate BEs. The key to understand this point is that one of the powerful properties of SVD method is to consider co-occurrence in degrees greater than 1. In other words, this approach not only considers the amount of direct relationship, but also considers relations between BEs with multiple intermediates. Therefore, the analysis shows that the proposed metric benefits from every concept in business processes and it can be used to measure conceptual cohesion of a service.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to measure conceptual cohesion of services. The proposed metric in this study is automatically measurable and its main application is in the service identification phase where it helps to discover the right services. The semantics represent the degree of service focus on one single business functionality. To this end, we defined conceptual cohesion of service based on service functionality and operation sequence aspects. The required information to measure these aspects is acquired from enterprise's business processes. This information is then used as the input of LSI technique. Once the BE-EBP matrix is formed on the basis of the acquired information, the SVD algorithm and then domain reduction operations are performed over the matrix. The reduced matrix contains the conceptual relationship of business entities which are used in the proposed metric. Although the evaluation results shows the capability of the proposed metric in terms of conceptual cohesion measurement, a real-world case study with a complete set of enterprise's business processes would further elaborate on its effectiveness. We are considering a set of controlled experiments in order to empirically validate the output of the proposed approach in real-world environments, in a future research.

ACKNOWLEDGMENT

The project has been partially supported by Iran Education and Research Institute for Information and Communication Technology (ITRC) and also Shahid Beheshti University under the supervision of Automated Software Engineering Research (ASER) Group, Faculty of Electrical and Computer Engineering. This work was also supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero - the Irish Software Engineering Research Centre (www.lero.ie)

REFERENCES

- [1] M. Pereplechikov, C. Ryan, and K. Frampton., "Cohesion Metrics for Predicting Maintainability of Service-Oriented Software" IEEE Seventh International Conference on Quality Software (QSIC), 2007.
- [2] Erl, T., *Service-Oriented Architecture Concepts, Technology, and Design*, Prentice Hall PTR, 2005, p. 792.
- [3] M. Papazoglou et al., "Service-Oriented Computing: State of the Art and Research Challenges," *Computer*, vol. 40, no. 11, 2007, pp. 38-45.
- [4] T. Erl, *SOA: Principles of Service Design*, Prentice Hall, 2007.
- [5] 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.
- [6] M. Pereplechikov, C. Ryan, and Z. Tari., "The Impact of Service Cohesion on the Analyzability of Service-Oriented Software," *IEEE TRANSACTIONS ON SERVICES COMPUTING*, Issue 2, Vol. 3, 2010.
- [7] B. Shim, S. Choue, S. Kim, and S. Park., "A Design Quality Model for Service-Oriented Architecture," *IEEE Computer Society*, 2008, pp. 403-410.
- [8] Dominich, S., "The Modern Algebra of Information Retrieval," Springer-Verlag Berlin Heidelberg, 2008.
- [9] S.R. Chidamber and C.F. Kemerer, "Towards a Metrics Suite for Object-Oriented Design, Object-Oriented Programming Systems, Languages and Applications (OOPSLA), Special Issue of SIGPLAN Notices, Vol. 26, No. 10, 1991, pp. 197-211.
- [10] S.R. Chidamber and C.F. Kemerer, "A Metrics suite for object-oriented Design," *IEEE Transactions on Software Engineering*, Vol. 20, No. 6, 1994, pp. 476-493.
- [11] W. Li and S.M. Henry, "Maintenance metrics for the object-oriented paradigm," In *Proceedings of 1st International Software Metrics Symposium*, Baltimore, MD, 1993, pp. 52-60.
- [12] M. Hitz and B. Montazeri, "Measuring coupling and cohesion in object oriented systems," *Proceedings of the International Symposium on Applied Corporate Computing*, 1995, pp. 25-27.
- [13] B. Henderson-Sellers, "Software Metrics," Prentice Hall, Hemel Hempstead, U.K., 1996.
- [14] J. M. Bieman and B. Kang, "Cohesion and reuse in an object-oriented system," *Proceedings of the 1995 Symposium on Software Reusability*, Seattle, Washington, United States, pp. 259-262, 1995.
- [15] G. Gui, P. D. Scott., "New Coupling and Cohesion Metrics for Evaluation of Software Component Reusability", *IEEE The 9th International Conference for Young Computer Scientists*, 2008.
- [16] A. Marcus, D. Poshyvanyk, and R. Ferenc., "Using the Conceptual Cohesion of Classes for Fault Prediction in Object-Oriented Systems," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, Issue. 2, Vol. 34, 2008.
- [17] A. Arsanjani, S. Ghosh, A. Allam, T. Abdollah, S. Ganapathy, and K. Holley., "SOMA: A method for developing service-oriented solutions," *IBM System Journal*, Issue. 3, Vol. 47, 2008, pp. 377-396.
- [18] S. Counsell, S. Swift, and J. Crampton., "The Interpretation and Utility of Three Cohesion Metrics for Object-Oriented Design," *ACM Trans. Software Eng. and Methodology*, Issue. 2, Vol. 15, 2006.
- [19] S. Kumaran, R. Liu, and F. Y. Wu., "On the Duality of Information-Centric and Activity-Centric Models of Business Processes," *Springer, CAISE'*, 2008, pp. 32-47.
- [20] P. Jamshidi, M. Sharifi and S. Mansour., "To Establish Enterprise Service Model from Enterprise Business Model.", "In 5th IEEE International Conference on Services Computing (SCC'08)", pp. 93-100, (2008).
- [21] G. Kowalski, "Information Retrieval Systems: Theory and Implementation," Kluwer Academic, 1999.
- [22] A. Rostampour, A. Kazemi, F. Shams, A. Zamiri, P. Jamshidi., "A Metric for Measuring the Degree of Entity-Centric Service Cohesion," *IEEE Service-oriented Computing and Application (SOCA)*, 2010.
- [23] OMG, "Business Process Model and Notation (BPMN)," Version 2.0, 2010, available at <http://www.omg.org/spec/BPMN/2.0>.
- [24] S. Khoshnevis, et al, "ASMEM: A Method for Automating Model Evolution of Service-Oriented Systems," 3rd International Workshop on a Research Agenda for Maintenance and Evolution of Service-Oriented Systems (MESOA 2009), 2009.
- [25] L. C. Briand, S. Morasca, and V. R. Basili., "Property-Based Software Engineering Measurement," *Transactions on Software Engineering*, Issue. 1, Vol. 22, 1996.
- [26] A. Kazemi, A.Rostampour, F. Shams, P.Jamshidi, and A. Nasirzadeh Azizkandi., "Measuring Service Cohesion Using Latent Semantic Indexing," *The Sixth International Conference on Internet and Web Applications and Services (ICIW')*, 2011.