A Metric for Measuring Degree of Service Cohesion in Service Oriented Designs

Mohammad Daghaghzadeh[1], Ahmad Baraani Dastjerdi[2] and Hossein Daghaghzadeh[3]

[1] Department of Computer Engineering, Sheikhbahaee University of Isfahan, Isfahan, Iran

[2] Department of Computer Engineering, University of Isfahan, Isfahan, Iran

[3] Department of Computer, Falavarjan Branch, Islamic Azad University, Falavarjan, Isfahan, Iran

Abstract

Service Orientation has gained momentum as a fundamental design paradigm for developing distributed enterprise applications. However, comprehensive and quantitative metrics for estimating the appropriateness of the service designs are still lacking. One of the quality attributes as to a SOA is cohesion, which is a determining factor for many other desirable features of the software including reusability, agility and etc. The previous studies on the measuring the degree of cohesion failed to comprehensively consider the relationship among entities to evaluate service cohesion. Therefore, this paper puts forward a new metric for measuring the cohesiveness of service and whole design based on design level information, especially the information embedded in entity model. This metric takes into account both Entity-Entity and Entity-Activity relations. Furthermore, the metric conforms to mathematical properties that cohesion metrics must have. The metric is empirically evaluated in a case study.

Keywords: *Service orientation, Cohesion, Software Metrics, Service Identification*

1. Introduction

Service-oriented architecture (SOA) is an information technology (IT) architectural approach that supports the creation of business processes from functional units defined as services [1] [2]. It is actually an evolved architectural concept from object-oriented and component-based developments. SOA promises to provide more agility for organizations adhering to SOA design principles during the entire software development life cycle. In [3] authors consolidated nine design principles which are contract orientation, abstraction, autonomy, coupling, statelessness, cohesion, discoverability, reusability and composability.

However, SOA is still hype because there is no clear method to identify and evaluate the building blocks of a SOA, namely services, against SOA design principles. Designers still have to identify services based on their experience and intuition. Consequently, the designed services barely conform to the principles so that SOA still remains as an unreached promise land for the time being [4].

Cohesion is a prominent attribute of software services and is used as a major quality factor in service design. Cohesion is the degree of functional relevance of activities which are performed by a service to realize a business process [1]. It shows how much an individual service is instrumental in performing one single task [5]. High cohesion brings about ease of understanding of the design model and makes the system more agile. Furthermore, it is conducive to reaching a service granularity at an almost adequate level [2]. In [6] authors recognize seven types of cohesion which are coincidental, logical, communicational, external, sequential, implementation and conceptual cohesions. A service has communicational cohesion if its interface operations operate on the same data. As stated in [5] a service with communicational cohesion is analyzable, changeable, stable and testable. Here in this paper, we examine communicational cohesion because of its significant effect on the quality of software.

There are several works in the literature which have tried to propose some metrics for evaluating and measuring the compliance of the service design against some of the design principles. Most of these studies, which are examined in related work section, focus their attention on common input and output parameters of service operations in order to estimate the service cohesion. However, as some of them indicate, the cohesion between entities should be taken into account while measuring cohesion between service operations. By regarding interrelation among entities, reusability rises [7]. Therefore, in this paper, we aim to put forward a metric for cohesion in order to quantitatively measure the degree of cohesion within each service and whole service design. The introduced metric, considers the interrelation among entities in entity model.

The rest of the paper is arranged as follows. First, previous studies are discussed in section 2. Then, we present some definitions on the main concepts in section 3. Afterward, in section 4, the concept of entity cohesion is investigated. Then, we elaborate our metric in section 5. Next, a case study is conducted to show applicability of the introduced metric in section 6. Later, we evaluate our metric against indispensible mathematical properties that characterize the quality attributes in section 7. In section 8, we empirically validate our metric by investigating the correlation of our metric with subjects' ratings and two previously presented metrics. In the end, we give the conclusion and the future work.

2. Related Work

Cohesion is a software feature that has been attracting the attention of many researchers working on different kinds of software development systems including procedural, object-oriented, component-oriented and service-oriented systems. The metrics for previous systems do not work for service oriented systems without adaptation because of the unique characteristics of service orientation [5]. Therefore, many authors have proposed cohesion metrics for the SOA context by adapting previous metrics. In the context of service oriented systems, there are several works that either investigate cohesion along with other SOA quality attributes or merely focus on proposing a new cohesion metric. In [7], authors consider the number of activities in a service along with shared data flows across such activities. Although, they take into account the relative complexity of each entity, they do not pay attention to the relationships among entities. In [6] [9] [10], the proposed metrics merely take into account number of shared messages among operations. In fact, the number of shared parameters of the service operations is divided by the total number of parameters. One of these metrics is SIDC that we use in this paper to evaluate our metric. In [7], a new cohesion metric called CCM for communicational cohesion is proposed. Authors consider the relationships among entities by relating entities that are accessed by the same operation. However, as they confess at the end of the paper, strengths of relationships are not examined. In [11], two quantitative metrics for cohesion and coupling is put forward. Even though entity-entity relationships are considered, this is done in an incorrect way. The fundamental flaw in this work is the assumption that the higher relationship cardinality between two entities strengthens the cohesion between them. Moreover, different kinds of relationships among entities are not taken into account. Finally, the metric has not empirically evaluated.

3. Basic Concepts

3.1. Service Portfolio Model

Service portfolio model is a model that represents architectural elements and their relationships in establishing a service portfolio from business process decomposition [8].

In this paper, we leverage UML 2.0 profile for software services to model service portfolio which is presented in Figure 1.

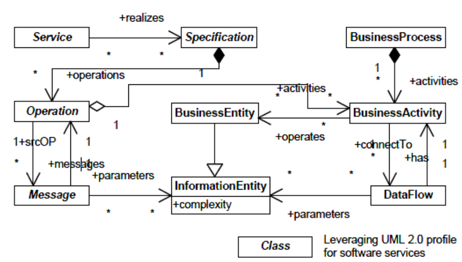

Fig. 1 Service portfolio model [8]

3.2. Entity Model

Entity model is a model that describes entities and their relationships. Entities indicate main stable domain abstractions of an enterprise. The entities are modeled in terms of organizations' proprietary rules and business policies. This brings about entities that are interrelated with each other [12]. Entity model can be described using entity relationship diagrams. Here, we assume that the reader is acquainted with notations used to draw an ER diagram.

- **Strong and weak entities**

Entities which are dependent on other entities for their existence are called weak entities and the relied entities are called strong.

- **Degree of relationship**

The number of entities which participate in a relationship is called degree of relationship. The general form for degree n is the n-ary relationship. A binary relationship is of degree two and a ternary relationship is of degree three.

- **Relationship cardinality ratio**

Determines the cardinal number of entities to which another entity can be connected through a relationship. The primary types of cardinality ratio for relationships are: one-to-one, one-to-many, and many-to-many.

- **Relationship participation**

Relationship participation states whether whole or part of an entity occurrence is associated to another entity. There are three types of relationship participation: mandatory-mandatory, mandatory-optional and optional-optional. In addition, there are some other types of relationship added to basic ER model which are aggregation, generalization and exclusive binary.

4. Entity Cohesion

There are several works in the literature which investigate the problem of entity clustering and Cohesion [13][14][15][16]. Generally, they carry out the grouping based on some prioritized grouping operations. According to [14] there are four kinds of relationship groupings which are: Dominance grouping, abstraction grouping, constraint grouping and relationship grouping**.** These operations should be done consecutively. In [14], authors define a precedence order of grouping operations based on the concept of cohesion borrowed from the software engineering field. They postulate that the level of cohesion between a strong and weak entity, and a dominance and dominated entity, are the highest level. The second level is the cohesion between a supertype and its subtype. Even though, they have not considered aggregation relationships between two entities, based on [13] it can be inferred that aggregation relationships bring about the same level of cohesion. They consider the cohesion between two entities that participate in a constraint relationship at the next level. At the fourth level, unary relationships have the highest cohesion, binary one-to-one next, then binary one-to-many, and finally binary many-to-many. Ternary and higher degrees of relationships are at the fifth level. At the end, there is no cohesion between entities that do not participate in any kind of relationship. Although, they posit the precedence order, they do not propose any figures as to the degree of cohesion. In [15], authors propose some weight numbers corresponding to each level. They use the concept of distance to do so. The distance between a weak entity and a regular entity is considered one. Furthermore, the distance between a supertype entity and each of its subtype entities is equal to 10. This distance is 100 for participating exclusive entities. For a binary relationship, the distance is deemed 1000. In the end, for ternary (or more) relationships, the distance is 10000.

Based on the mentioned studies, we propose the following distances for each kind of relationship (Table 1).

Table 1: Distances for relationship types

Relationship Type	Distance
Strong- Weak	1
Generalization	10
Aggregation	10
Exclusive Binary	100

For binary relationships which are not categorized under the above relationships we have the following distances (Table 2).

Table 2: Distances for binary relationships

Participation / Cardinality ratios	M-M	M-O	O-O
1-1	1	10	100
1-N	5	50	500
M-N	10	100	1000

In addition, for n-ary relationships (n>2), we multiply each distance in Table 2 by 10^{n-2} .
According to proposed distances, we define the distance between two distinctive entities as the number of edges in the shortest path between them.

$$Distance\ (E_1,E_2) = \begin{cases} \min_p \sum_{i=0}^{e_p} d_i & p > 0 \\ \infty & \mid p = 0 \end{cases} \qquad (1)$$

where e_p is the number of edges in a path between E_1 and E_2 and p is the number of existing paths between E_1 and E_2. In addition, d_i is relationship distance which comes either from Table 1 or Table 2. Moreover, for the purpose of this paper, we deem that the distance between identical entities is one. We can measure the degree of cohesion between E_1 and E_2 by Formula 2,

$$\text{Cohesion}\ (E_1,E_2) = \frac{1}{Distance\ \ (E_1,E_2)} \qquad (2)$$

For every entity model, the table in which the distance among all entities is specified can be constructed. In this context, we call such a table Distance Table.

5. The proposed service cohesion metric

In this section, we put forward a new metric for service and whole service design cohesion. Our metric takes into account relationships among entities as well as relationships among activities and entities. Moreover, in this metric, the number of activities in a service affects the degree of cohesion. The reason being is that adding operations to a service lessens its cohesion since the aggregated service does not focus on the semantic of one single task any longer [5].

To calculate cohesion between two activities in a service, we firstly build a complete bipartite graph with business entities on which each activity operates. Each edge with endpoints A and B has weight w coming from row A and column B in the distance table. Then, we take a greedy approach in order to match between the entities that each activity deals with. The summary of the approach can be seen in the pseudo code presented in Figure 2.

```
Void MatchingAlgorithm (BE1, BE2,DT, R)
{
Input: BE1, BE2 are the sets of business entities on which each activity operates and DT is the distance table for the corresponding entity model.
Output: R is a set that keeps the selected edges.
Build a complete bipartite graph G from BE1 and BE2;
Give each edge of graph G weight w that comes from corresponding row and column in DT;
R=∅;
S=∅;
While (|S| < |BE1|+|BE2|)
{
Select an edge e from G that has the minimum weight;
If no edge with e's endpoints is present in R and it does not make a circle with other edges in R then
 {
Add the new edge e to set R;
Add e's endpoints to set S;
}
}
}
```

Fig. 2 The algorithm for matching between the entities on which two activities work

To estimate cohesion between two activities, we must obtain average of cohesion among matched entities on which activities work. Therefore, cohesion between two activities i and j called Activity Cohesion (AC) is calculated as follows:

$$\text{AC (i, j)} = \frac{\sum_{i=1}^{|R|} \frac{1}{\text{Weight} \quad (R(i))}}{|R|} \tag{3}$$

Now, we can calculate Service Cohesion (SC) for each service k by Formula 4,

$$\text{SC}_k = \begin{cases} \sum_{i=1}^{a} \sum_{j=1, i>j}^{a} \frac{AC\ (i,j)}{\frac{a\ (a-1)}{2}} & a>1 \\ 1 & a=1, \end{cases} \tag{4}$$

where a is the number of the activities in k'th service .

In the end, Service Design Cohesion (SDC) is computed by Formula 5,

$$\text{SDC}= \frac{\sum_{k=1}^{s} \text{SC}_k}{s}, \tag{5}$$

where s is the number of identified services in service portfolio.

6. Case study

In this section, we utilize a real-world business scenario to show the application of the proposed metric and evaluate its usefulness. In this scenario, the goods request process of a mine company is studied.

Every employee in each part of the company can compose a request and fill it out with his needed goods. Then, he sends the request to his boss. Afterward, the bus examines the request to see if the goods are really necessary and the amounts of requested goods do not exceed the determined share of the part. The boss may also add some other goods to the request or edit it. Next, the boss signs the request and sends it to the CEO or his deputy. He checks whether the request conforms to the company's high level policies and regulations. After some probable negotiations and editing, he either signs and then sends the request for store's boss or rejects the request. Store's boss examines the request to make sure that the request does not disturb the balance of the store's stocks.

We analyzed the enterprise and modeled the as-is business process. Then, we obtained the to-be business processes and entity model. Entity model is shown in Figure 3.

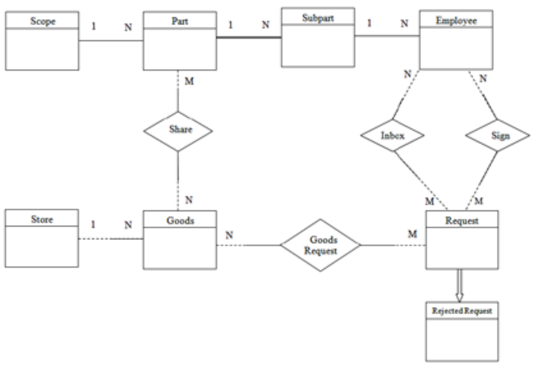

Fig 3.Entity model for good request case study

The distances among all entities are determined based on distances in Table 1 and Table 2 and can be seen in Table 3.

Table 3: The distance table for the good request case study

	Store	Goods	Scope	Part	Subpart	Employee	Request	Goods Request	Share	Inbox	Sign	RejectedRequest
Store	1	50	1055	1050	1055	1060	1050	100	100	1100	1100	1060
Goods		1	1005	1000	1005	1010	1000	50	50	1050	1050	1010
Scope			1	5	10	15	515	1055	55	65	65	525
Part				1	5	10	1010	1050	50	60	60	1020
Subpart					1	5	1005	1055	55	55	55	1015
Employee						1	1000	1050	60	50	50	1010
Request							1	50	1050	50	50	10
Goods Request								1	100	100	100	1060
Share									1	110	110	65
Inbox										1	100	65
Sign											1	65
RejectedRequest												1

We identified two services based on business process models and entity model, namely Request Processing, Goods Processing.

In the following table it is shown that each activity in Request Processing service operates on which business entities.

Table 4: Activities in Request Processing service and related entities

	Good	Scope	Part	Subpart	Employee	Request	Goods Request	Share	Inbox	Rejected Request
UpdateSign						×				
ReadRequest			×	×	×	×				
ReadSigns						×				
CreateRequest						×				
CheckInbox				×	×	×			×	
RejectRequest										×

To evaluate the cohesion between CheckInbox and ReadRequest activities we should build the graph shown in Figure 4.

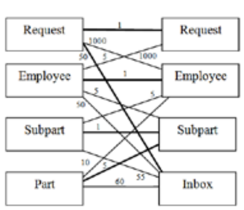

Fig. 4 The bipartite graph for CheckInbox and ReadRequest activities

According to algorithm in Figure 2 the bolded edges are chosen. Now we can calculate the cohesion between the activities as follows:

$$\mathrm{AC}\,(4,1) = \frac{\frac{1}{50}+1+1+1+\frac{1}{5}}{5} = 0.644$$

In Table 5, the cohesion among all the activities in the service is shown.

Table 5: Cohesion among all the activities in Request Processing service

	UpdateSign	ReadRequest	ReadSigns	CreateRequest	CheckInbox	RejectRequest
UpdateSign	1	0.25	1	1	0.2555	0.1
ReadRequest		1	0.25	0.25	0.644	0.0257
ReadSigns			1	1	0.2555	0.1
CreateRequest				1	0.2555	0.1
CheckInbox					1	0.029
RejectRequest						1

So, SC for Request Processing is calculated as follows:

$$\mathrm{SC}= \frac{\begin{array}{c}0.25+1+1+0.2555+0.1+0.25+0.25+0.664+\\0.0257+1+0.2555+0.1+0.2555+0.1+0.029\end{array}}{15} = 0.369$$

Cohesion of Goods Processing service is computed as we did it for Request Processing. So, cohesion of the service design is SDC= $\frac{0.369+0.644}{2} = 0.506$

7. Analytical Validation

In this section, we validate our metric through proving that it satisfies essential properties of a cohesion measure. We exploit properties based software engineering measurement framework [17] to validate our metric.

- **Non-negativity and Normalization**

AC is always between zero and one. The reason being is that all the weights of the edges, which are the distances between entities, are always between one and infinity. Consequently, $\frac{1}{\mathrm{Weight}\ \ (\mathrm{R(i)})}$ is always between one and zero so that the average is always between zero and one. As a corollary to this, SC which is the average cohesion of all pairs of activities in a service is between zero and one. Hence, SDC which is the average cohesion of all services embedded in a service design is always between zero and one.

- **Null and Maximum Values**

AC becomes zero when there is no relationship among the business entities that the activities operate on. Furthermore, AC becomes one at its maximum when all operated entities are either the same or participate in strong-weak relationships. Also, minimum of SC is zero when degrees of cohesion among all pairs of activities in a service are zero. Maximum of SC is one when either the cohesion among all pairs of activities in service is one or the service merely contains one activity. In addition, minimum of SDC is zero when the cohesion of all services in service design is zero. Maximum of SDC is one when the cohesion figures of all existing services in the design are one.

- **Monotonicity**

Our metric passes this criterion because adding a shared business entity as a cohesive interaction to the group of business entities on which activities work will not bring about decrease in the degree of cohesion between the activities. Addition of a cohesive interaction will add one edge with the weight of one to set R. Therefore, this will increase numerator and denominator of AC by 1, which does not decrease output of AC ($a \geq 0, b > 0, a \leq b \rightarrow \frac{a+1}{b+1} \geq \frac{a}{b}$).

- **Cohesive module**

Merging two services with non-related activities will result in a service with less degree of cohesion. The reason being is that this action will increase number of zeros in the numerator as it increases the value of $\frac{a\,(a-1)}{2}$ in the denominator.

8. Empirical Validation

In this section, we validate our metric by analyzing the data gathered from a group of 15 experts in SOA. The experts were students of master program in Sheikhbahaee University who had passed advanced software engineering course that SOA was one of its syllabuses. We asked the subjects to rate the cohesions of the services by a numerical scale between 0 and 1. The objects rated by the subjects are services embedded into two service portfolios constructed regarding the case study represented in section 6 (Case study 1) and the one presented in [18] (Case Study 2) as well as their corresponding entity models. The subjects had given guidelines as to how to do the experiment. Each subject did the experiment by himself, at home, and could use unlimited time to rate the cohesion of the given services. Our objective is to establish if any correlation exists between cohesion figures calculated by our metric and subjects' ratings. The Spearman Rank-Difference Correlation Coefficient, r_S, was employed to establish the correlation of the data collected in the experiment because the data acquired in the experiment is distribution free. The Spearman r_S is a non-parametric statistic employed to find out the relationship between two variables expressed as ranks [19].The correlation coefficient is used to determine how much a variable is able to predict the value of another variable. In our experiment the null hypothesis was as follows:

H_0: "there is no correlation between the SC metric and the subjects' rating of service cohesions".

We control the probability that the null hypothesis would be mistakenly rejected by two confidence levels: α_1=0.005 and α_2=0.05. Furthermore, the decision rules for rejecting the null hypothesis are:
For α_1: reject H_0 if r_S >= 0.689; for α_2: reject H_0 if r_S >= 0.447. The correlation coefficient for each subject is shown in Table 6.

Table 6: Spearman Rank Correlations regarding subjects' rating

Subjects	r_s	α_1	α_2
1	0.721	Reject H_0	Reject H_0
2	0.639	Reject H_0	Reject H_0
3	0.781	Reject H_0	Reject H_0
4	0.691	Reject H_0	Reject H_0
5	0.785	Reject H_0	Reject H_0
6	0.873	Reject H_0	Reject H_0
7	0.912	Reject H_0	Reject H_0
8	0.711	Accept H_0	Reject H_0
9	0.587	Accept H_0	Reject H_0
10	0.935	Reject H_0	Reject H_0
11	0.693	Accept H_0	Reject H_0
12	0.831	Reject H_0	Reject H_0
13	0.659	Accept H_0	Reject H_0
14	0.401	Accept H_0	Accept H_0

Based on the data in Table 2, we reject H_0 for 78 percent of the subjects with regard to α_1 and for 92 percent of the subjects considering α_2.
In addition, we calculated the correlation of our metric with SIDC and CCM metrics in two formerly introduced case studies. In Table 7, it is seen that our metric has high correlation with these two metrics.

Table 7: Spearman Rank Correlations regarding the case studies

	CCM	SIDC
Case Study 1	0.89	0.87
Case Study 2	0.92	0.94

9. Discussion

Despite the fact that our figures for service cohesion is close to the figures of SIDC, our metric shows its usefulness in the process of service identification. Even though the difference between figures is not too much, it has significant effect on the shape of service portfolio because an automatic service identification approach is carried out by contrasting these figures rather than by figures themselves. Accordingly, a little difference between two figures determines whether we add an activity to a specific service or not. To demonstrate our contention, we use an example. In the case study presented in section 6, if we did not consider the relationship between Request and RejectedRequest entities, an identification method would consider a separate service for activity RejectRequest despite the fact that these activities are cohesive. Hence, our metric is a rational measure for service cohesion and is effective in service identification.

10. Conclusion and future work

Comprehensive and quantitative metrics for estimating the appropriateness of service designs are still lacking. High cohesion is a determining factor for many other desirable features of software including reusability, agility and etc. Previous studies on measuring the degree of cohesion have failed to thoroughly consider relationship among entities to evaluate service cohesion. In this paper, we put forward a new metric for measuring the cohesiveness of the service and whole design. This metric takes into account both Entity-Entity and Entity-Activity relations. The metric is empirically evaluated in a case study and its correlation with experts' ratings and other metrics were investigated and examined. Furthermore, the metric conforms to mathematical properties that cohesion metrics must have. In the future, we intend to extend this metric in the way to embrace other kinds of cohesions as well. In addition, we are going to use this metric in a method for service identification. Furthermore, it is probable that the figures as to the distance between entities need to be adapted. Finally, we will conduct other case studies in different areas to show the applicability of the metric.

References

[1] A. Erradi, N. Kulkarni and P. Maheshwari, “Service Design Process for Reusable Services: Financial Services Case Study”, In 5th International Conference on Service Oriented Computing (ICSOC’07), 2007, Vol. 4749, pp. 606-617.

[2] M.P. Papazoglou and W.-J. van den Heuvel, “Service-Oriented Design and Development Methodology”, In International Journal of Web Engineering and Technology, Vol. 2, No. 4, 2006, pp.412-442.

[3] T. Kohlborn, “A Consolidated Approach for Service Analysis”, MS. thesis, Chair of Interorganisational Systems , , Westfalische-Wilhelms Universitat Munster, Münster ,Germany, 2008.

[4] C. Llorente, “Model-Driven SOA with Oslo”, The Architecture Journal, No. 21, pp. 10-15, 2009.

[5] G. Feuerlicht, “Service Aggregation Using Relational Operations on Interface Parameters”, In 4th International Conference on Service-Oriented Computing (ICSOC’06), 2006, Vol. 4652, pp. 407-424.

[6] M. Perepletchikov, C. Ryan, and K. Frampton, “Cohesion Metrics for Predicting Maintainability of Service-Oriented Software”, In Seventh International Conference on Quality Software (QSIC 2007), 2007, pp. 328-335.

[7] A. Rostampour, A. Kazemi, F. Shams, A. Zamiri and P. Jamshidi , “A Metric for Measuring the Degree of Entity-Centric Service Cohesion”, IEEE International Conference on Service-Oriented Computing and Applications (SOCA’10), 2010, pp.1-5.

[8] Q. Ma, N. Zhou, Y. Zhu and H. Wang, “Evaluating Service Identification with Design Metrics on Business Process Decomposition”, In Proceedings of IEEE SCC'2009,2009, pp.160-167.

[9] R. Sindhgatta, B. Sengupta and K. Ponnalagu, “Measuring the Quality of Service Oriented Design”, In Proceedings of the 7th International Joint Conference on Service-Oriented Computing, 2009, pp. 485-499.

[10] B. Shim, S. Choue, K. Santae, and P. Sooyong, “A Design Quality Model for Service-Oriented Architecture”, In 15th Asia-Pacific Software Engineering Conference, 2008, pp. 403-410.

[11] F. Meng, D. Zhan and X. Xu, “Business Component Identification of Enterprise Information System: A hierarchical clustering method”, IEEE International Conference on e-Business Engineering (ICEBE'05), 2005, pp.473-480.

[12] T. Erl, “Service-Oriented Architecture (SOA): Concepts, Technology, and Design”, US. : Prentice Hall, 2005.

[13] P.Jaeschke, A. Oberweis and W. Stucky, “Extending ER Model Clustering by Relationship Clustering”, In Proceedings of ER'1993, 1993, pp. 451-462.

[14] T. J. Teorey, G. Wei, D. L. Bolton and J. A. Koenig, “ER Model Clustering as an Aid for User Communication and Documentation in Database Design”, Communications of the ACM , Vol. 32, No. 8, 1989, pp. 438-450.

[15] J. Akoka and I. Comyn-Wattiau, “Entity-Relationship and Object-Oriented Model Automatic Clustering”, In Data & Knowledge Engineering Journal, Vol. 20, No. 2 , 1996, pp. 87-117.

[16] M. Tavana, P. Joglekar and M. A. Redmond, “An Automated Entity-Relationship Clustering Algorithm for Conceptual Database Design”, Information Systems, Vol. 32, No. 5, 2007, pp. 773-792.

[17] 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, pp. 89-103.

[18] A . Albani, S. Overhage and D. Birkmeier, “Towards a Systematic Method for Identifying Business Components”, In the Proceedings of the 11th International Symposium on Component-Based Software Engineering, 2008, pp. 262-277

[19] S. Siegel and J.N. john Castellan, “Nonparametric Statistics for The Behavioral Sciences”, US. : McGraw Hill, 1988.