



Journal of
**Software
Engineering**

ISSN 1819-4311



Academic
Journals Inc.

www.academicjournals.com

Three-dimensional Granularity Division Method for Service Clusters

^{1,2}Hu Qiang, ¹Du Yu Yue and ³Li Peng

¹College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China

²School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China

³Department of IT Management, Affiliated Hospital of Medical College, Qingdao University, Qingdao 26655, China

Corresponding Author: Du Yu Yue, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China

ABSTRACT

To quickly locate an appropriate web service from the large count of services on the Internet, a cluster method is adopted to manage services. The services with similar functions are defined as a service cluster and then the architecture oriented service cluster and service binding algorithm are presented. However, the granularity of service clusters impacts greatly on the efficiency of service discovery and aggregation. A three-dimensional granularity division method for service clusters is proposed in this study. It includes structure granularity, quantity granularity and quality granularity. And each kind of the granularity are divided into three levels, the division rules for each level are also presented. Simulation experiments are provided to illustrate the rationality of the proposed granularity division method.

Key words: Web service, service cluster, granularity, service discovery

INTRODUCTION

The number of Web services is dramatically growing with the development and maturity of service computing technology. Many service providers have published various services in Internet. More and more Web applications were developed and deployed by the paradigm of Web services. An appropriate service meeting the user request can be found and bound based on the current methods of invoking services. However, there are some significant shortcomings in the methods, such as poor response self-adaptability. The response self-adaptability refers to automatically retrieve a substituted Web service for the current one once the latter is unavailable or the user request is slightly changed.

A group of Web services with similar functions and interfaces respond for a service request and it can quickly find and bind a substitute Web service from the service group when needed. The group of Web services with similar functions and interfaces (input and output parameters) are defined as a service cluster in this study. Different from the current service request and response, the service request and response based on service clusters is named as the service cluster schema. Some similar concepts, such as service community (Sheng *et al.*, 2009), service container (Benatallah *et al.*, 2003), service group (Maguire *et al.*, 2006) and service pool (Liu *et al.*, 2009),

are put forward to describe a group of Web services. Group method has been adopted in these concepts, however, less flexibility can be provided by them because only Web services with the same interfaces can be classified as a group.

Since web services in a service cluster may have small differences in the interfaces and functions, they can satisfy the diversity and difference of a service request to some extent. Thus, the service cluster can provide good flexibility and self-adaptability. However, the interfaces, quantity and quality of the component Web services in a service clusters determines its granularity and the granularity of a service cluster is closely related with the discovery efficiency of Web service in the service cluster schema (Ardagna *et al.*, 2011; She *et al.*, 2009). Aiming to provide a reasonable method for the granularity division, the granularity of service clusters is investigated in this study from the aspects of interfaces, quantity and quality of their component Web services.

ARCHITECTURE ORIENTED SERVICE CLUSTER

Service architecture based on service cluster: A new architecture oriented on service clusters is proposed in Fig. 1. The services in the physical resource layer are grouped as service clusters in the virtual resource layer. The virtual resource layer is constructed based on the physical resource layer and services for the business model layer.

According to service requests from the users, a service cluster or a group of service clusters will be responded in the virtual resource layer and an optimal Web service or a service composition (service flow) will finally be bounded and returned for the user. An algorithm is proposed in next section to bind an optimal Web service for the service request.

Definition of service cluster: Web services with similar interfaces and functions are grouped as a service cluster. Definitions of Web services and the service cluster are presented.

Definition 1: A Web service is a 6-tuple:

$$Ws = (W_{id}, W_{des}, I, O, Q, L)$$

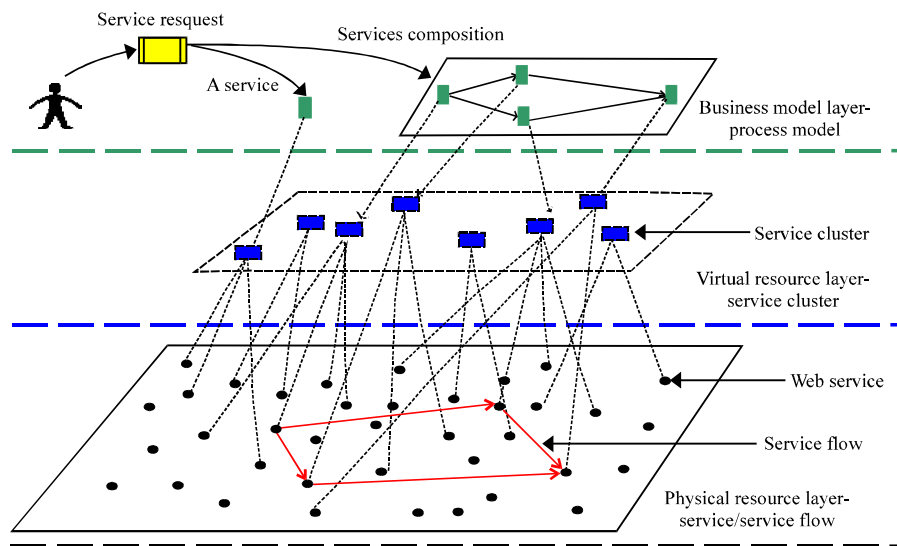


Fig. 1: A three-level service architecture based on service clusters

Where:

- W_{id} = An identification of a Web service
- W_{des} = A function description of the Web service
- I and O = The sets of input and output parameters
- Q = A set of quality parameters
- L = The URI of a Web Service

In Definition 1, W_{des} is defined as 2-tuple (Ob, Ac), Ob is the operation object of a Web service, Ac is the operation. As an example, the W_{des} of the Web service which can be used to book flight tickets may be defined as (Flight, Book).

The element in Q is defined as:

$$q = \{N, C, V, U\}$$

where, N is the name of quality parameter, C is a comparison operator, V is a value of the quality parameter, U is the unit of the quality parameter. For example, if a Web service has a quality parameter $q = (\text{cost} = 0.02, \$)$, it means that the user should pay for 0.02\$ for each invocation of this Web service.

Definition 2: A service cluster is a 7-tuple:

$$S_{cluster} = (SC_{id}, SC_{des}, I, O, S_w, Q_c, \delta)$$

Where:

- SC_{id} and SC_{des} are the identifier and function description of a service cluster, respectively
- $I = I_c \cup I_p$, $O = O_c \cup O_p$, where I_c and O_c represent the common input parameters and output parameters of a service cluster, respectively and I_p and O_p represent the private input parameters and output parameters of some Web services in the service cluster, respectively
- S_w is the component Web services in the service cluster, i.e.:

$$S_w = \{ws_1, ws_2, \dots, ws_n\}$$

where, ws_i represents a Web service and $1 \leq i \leq n$

- δ is a correlation function, which makes a mapping between the private output parameters the service cluster and its component Web services
- Q_c is a set of quality parameters

Four Web services to query flight tickets are listed in Table 1. To simplify description, the URIs of Web services are omitted. And only three items of quality parameters are provided, which are the response time, denoted as Rt, the cost of service, denoted as Co and the response rate of service, denoted as Rr. These services are grouped as a service cluster S_1 . Table 2 is the specification of S_1 . The common parameters and private parameters are shown in the column I_c , O_c and I_p , O_p . Notice that all used abbreviations in the headers of Table 1 and 2 can refer to the definitions of Web service and service cluster, respectively.

Table 1: Definitions of Web services to query flight tickets

W_{id}^1	W_{des}^2	I^3	O^4	Q^5
ws_1	Flight, query	Date, dep, des	Airline, deptime, arritime, price, remainder,	(Rt, <,5,ms), (Co, = ,0.0,\$), (Rr, >,96,%)
ws_2	Airticket, query	Date, dep, des, class	Airline, deptime, arritime, price, remainder, planemode	(Rt, <,2,ms), (Co, = ,0.01,\$), (Rr, >,98,%)
ws_3	Flight, query	Date, dep, des, pastype	Airline, deptime, arritime, price, remainder, planemode	(Rt, <,7,ms), (Co, = ,0.0,\$), (Rr, >,92,%)
ws_4^6	Flight, book	Date, dep, des, pastype	Airline, deptime, arritime, price, remainder, company	(Rt, <,1,ms), (Co, = ,0.02,\$), (Rr, >,99,%)

¹Identifier of a Web service, ²Function description, ³Input parameters, ⁴Output parameters, ⁵Quality parameters, ⁶ Ws_i represents the identifier of the concrete Web service, where $1 \leq i \leq 4$

Table 2: Definition of a service cluster to query flight tickets

CId^7	$Cdes^8$	Ic^9	Ip^{10}	Oc^{11}	Op^{12}	δ^{13}	Qc^{14}	Sw^{15}
$S1^{16}$	Flight, query	Date, dep, des	Pasttype, class	Airline, price, deptime, arritime, remainder	Planemode, company	<planemode>-<pasttype>,<class>,<company>-(<pasttype>)	(Rt, <,[1,7],ms,N), (Co = [0,0.02],\$,M), (Rr, >,[92,99],%.M)	ws_1 , ws_2 , ws_3 , ws_4

⁷The identifier of a service cluster, ⁸The function description of a service cluster, ⁹The common input parameters, ¹⁰The private input parameters, ¹¹The common output parameters, ¹²The private output parameters, ¹³The mapping relation between the private output and input parameters, ¹⁴The quality interval parameters, ¹⁵The set of included Web services in a service cluster, ¹⁶The identifier of the giving service cluster

Since, all Web services can accept the common input parameters and provide the common output parameters, the function δ is used to only describe the mapping relations between private input parameters and output parameters. As shown in the Table 1, δ of S_1 is constructed as:

$$\{(<pasttype>,<class>)-<planemode>;(<pasttype>)-<company>\}$$

The value of quality parameters in a service cluster should reflect the quality characteristics of all the component Web services. Interval value method is introduced to describe the value of quality parameters in a service cluster. The quality parameter of Q_c in a service cluster is defined as:

$$q = \{N,C,[V_{min}, V_{max}],U, T\}$$

where, V_{min} and V_{max} represent the upper and lower bound value of q_i , respectively and T is the type of the parameter with only two value M and N representing the quality parameter is mandatory satisfied or not, respectively. Q_c of S_1 is defined as:

$$\{(Rt, <,[1,7],ms,N), (Co = [0,0.02],$,M), (Rr, >,[92,99],%.M)\}$$

It means that Web services in S_1 are with the following quality properties: The maximal response time is less than 7 ms, the minimal response time is less than 1 ms, the

cost of service is from 0\$-0.02\$ and the maximal response rate is more than 99%, the minimal response rate is more than 92%.

Service binding algorithm: In the literature (Du and Hu, 2013), an algorithm to bind an appropriate Web service in a service cluster was presented. To bind the resultant service more quickly and accurately, the improved algorithm is proposed in this study. p_a and p_b are used to represent two different groups of parameters and the functions $\text{Num}(p)$ and $\text{Type}(m)$ are used to represent the number of parameters in p and the parameter type of m , respectively. $\text{Val}(m)$ is defined to represent the range of parameter value of m . Some definitions (Du and Hu, 2013) used in the process of service binding are reviewed.

Definition 3: The p_a and p_b , are defined as the included parameter if $\text{Num}(p_a) \leq \text{Num}(p_b)$ and $\forall m_i \in p_a, \exists n_j \in p_b$, such that $\text{Type}(m_i) \equiv \text{Type}(n_j)$, it is denoted as $p_a \ll p_b$.

Definition 4: The p_a and p_b , are defined as the isomorphic parameter if $\text{Num}(p_a) = \text{Num}(p_b)$ and $\forall m_i \in p_a, \exists n_j \in p_b$, such that $\text{Type}(m_i) \equiv \text{Type}(n_j)$ and $\forall n_i \in p_b, \exists m_j \in p_a$, such that $\text{Type}(n_i) \equiv \text{Type}(m_j)$, it is denoted as $p_a \leftrightarrow p_b$.

Definition 5: The p_a and p_b , are defined as the satiable parameter if $p_a \ll p_b$ and $\forall m_i \in p_a, \exists n_j \in p_b$, such that $\text{Val}(m_i) \subseteq \text{Val}(n_j)$, it is denoted as $p_a \sim p_b$.

Definition 6: Let CCP be the closest common parent of ontology concepts O_1 and O_2 in domain ontology tree. $\text{Level}(\text{CCP})$, $\text{level}(O_1)$ and $\text{level}(O_2)$ denote the number of level depth of CCP, O_1 and O_2 in the domain ontology tree. Let $\text{level}(O_{\text{top}}) = 1$, O_{top} is the top concept node in the tree. The semantic distance between O_1 and O_2 is defined as follows (Zhong *et al.*, 2002):

$$\text{Dist}(O_1, O_2) = \begin{cases} (2^{-\text{level}(\text{CCP}) - \text{level}(O_1)} + 2^{-\text{level}(\text{CCP}) - \text{level}(O_2)}), & O_1 \neq O_2 \\ 0, & O_1 = O_2 \end{cases}$$

The semantic similarity of two concepts O_1 and O_2 in domain ontology tree is defined as $\text{SeSim}(O_1, O_2) = 1 - \text{dist}(O_1, O_2)$.

Definition 7 (quality similarity): There are n items of quality parameters in the quality parameter set of services S_i and S_j , representing by as $S_i.Q$ and $S_j.Q$, if $\forall q_m \in S_i.Q, \exists q_m' \in S_j.Q$, such that $S_i.Q.q_m \sim S_j.Q.q_m'$, then, let $T_{q_m} = 1$ else $T_{q_m} = 0$. The quality similarity between $S_i.Q$ and $S_j.Q$ is defined as:

$$\text{Sim}_Q(S_i.Q, S_j.Q) = \sum_{m=1}^n T_{q_m} \cdot \omega_m$$

where, ω_m represents the weight of q_m in:

$$S_i.Q, \sum_{m=1}^n \omega_m = 1$$

A service request is formally represented as 5-tuple $Sr = (R_{Id}, R_{des}, I, O, Q)$. There two main operations in the binding algorithm including matching interface and computing similarities of the function description, interfaces and the quality parameters. The algorithm is described as follows:

Algorithm 1: Bind an appropriate Web service from service clusters

Input: A service request Sr and a set of service clusters SC

Output: An appropriate Web service with the optimal Qos :

```

 $SC_R = \emptyset, SF = False$ 
for each  $S \in SC$ 
{compute  $SeSim(S.C_{des}, Sr.R_{des})$ 
if  $SeSim(S.C_{des}, Sr.R_{des}) > v, SC_R = SC_R \cup \{S\};$ 
While (!Flag)
 $\{S' = \{S \in SC_R \mid \max(SeSim(S.C_{des}, Sr.R_{des}))\};$ 
if  $(S.I \ll r.I \wedge Sr.O \ll S.O)$  and  $(S.Qc \sim Sr.Q)$ 
 $\{O' = Sr.O - S.Oc$ 
if  $(O' = \emptyset) SF = True$ 
else for  $\forall o_i, o_j \in O'$ 
if  $(\delta(o_i) = \delta(o_j) \in Sr.I) SF = True\}$ 
}
}
for each  $ws \in S', S_w$ 
if  $(ws.I \ll Sr.I \wedge Sr.O \ll ws.O) S_w' = S_w' \cup \{ws\}$ 
for each  $ws \in S_w', s = \{ws \mid \max(Sim_Q(ws))\}$ 
Output(s)

```

GRANULARITY OF SERVICE CLUSTERS

By dividing the granularity of service clusters reasonably, the accuracy and efficiency of service discovery can be improved greatly. The granularity of service clusters is analyzed from structure, quantity and quality in this section. Let SC be a service cluster and S be the interface of SC , $S.Qc$ represents the quality parameters of SC .

Structure granularity: Structure granularity describes the granularity of service clusters from the interfaces of services clustered in the service clusters. From the structure of interfaces of the component Web services, a service cluster can be divided as E level (excellent level), O level (ordinary level) and P level (poor level):

- **G level:** $\forall S_i \in SC, S.I \Leftarrow S_i.I \wedge S.O \Leftarrow S_i.O$
- **O level:** $\exists S_i \in SC, S.I \Leftarrow S_i.I \wedge S.O \Leftarrow S_i.O \wedge \exists S_j \in SC, S_j.I \ll S.I \wedge S_j.O \ll S.O$
- **P level:** $\forall S_i \in SC, S_i.I \ll S.I \wedge S_i.O \ll S.O$

For a service cluster in a good level, the private input and output parameters set, i.e., I_p and O_p , are empty. All the parameters are common parameters of its component Web services. Some Web services are with the isomorphic parameter while other are with private input and output parameters in the ordinary level. As a poor level service cluster, all of its component Web services have private input parameters or output parameters.

Quantity granularity: Quantity granularity is divided by the number of services in a service cluster. The number of services affects greatly on the accuracy and efficiency in binding a service. If there are too many Web services in a service cluster, the service cluster provides more opportunities for service matching with a high binding accuracy. However, the binding efficiency is low. Otherwise, the service cluster performs with high binding efficiency and low binding accuracy.

By analyzing the accuracy and efficiency of binding a Web service in a service cluster from lots of simulation experiments, the following criteria are proposed to divide quantity granularity as S level (small level), M level (middle level) and L level (large level). Not a specific number can be given to define the size of quantity granularity since the total number of the Web services in all the service clusters is not the same in the different cases. Suppose the total number of the Web services in all the service clusters is n ($2000 < n < 20000$), the following quantity granularity is proposed:

- **S level:** The number of services in a service cluster is less than $\sqrt[3]{n}$
- **M level:** The number of services in a service cluster is between $\sqrt[3]{n}$ and $\sqrt[3]{n} + \sqrt[3]{n}/2$
- **L level:** The number of services in a service cluster is more than $\sqrt[3]{n} + \sqrt[3]{n}/2$

Quality granularity: The division of quality granularity affects the generation of service clusters. The quality parameters are key factors to determine whether a Web service can be clustered into a service cluster once their interfaces match correctly. The quality granularity of a service cluster is divided as C level (closed level), S level (semi-open level) and O level (open level):

- **C level:** $\forall q \in S.Qc, q.T = M$
- **S level:** $\exists q_i \in S.Qc, q_i.T = M \wedge \exists q_j \in S.Qc, q_j.T = N$
- **O level:** $\forall q \in S.Qc, q.T = N$

Given a Web service W with the matching interfaces with a service cluster Sc , if Sc is in the O level in view of the quality granularity, the Web service W can be clustered into Sc with any values in its quality parameters. For the service cluster in S level and C level, the Web service can be grouped into Sc unless the quality parameters of W are the satiable parameters (Definition 5) of the corresponding mandatory quality parameters in Sc .

SIMULATION EXPERIMENT

Since there are few Web services developed based on ontology on the internet, web services in the simulation experiment are designed based on the concepts of an ontology tree built by ourselves. The ontology tree is designed with the width of seven and depth of nine. All the information of the Web services is expressed by the concepts in the ontology tree, such as the function description, the input and output parameters. Thus, the semantic similarity can be easily obtained from the ontology tree.

The simulation program in the experiments is designed by Java, the operating system is Windows 7 and the hardware for the computer is as follows: CPU is the Intel core i3-2120 with 3.3 GHz; memory is 2 G. Each round of the experiments is performed for 10 tests, taking the average of the results as the final simulation results.

To verify the influence of the structure granularity and quantity granularity on the service discovery efficiency, 10000 Web services were randomly generated. Six round experiments were carried out and the number of Web service in each service clusters is 20, 50, 100, 120, 200 and 500

for each round, respectively. The experiment results are showed as Fig. 2. For the structure granularity, the service finding efficiency of service clusters in the G level is higher than that in the O level and the service finding efficiency of service clusters in the O level is higher than that in the P level. Since the total number of the services is 10000, the demarcation point of S level and M level proposed in this study is 100 while the demarcation point of M level and L level is 122 in the quantity granularity. From Fig. 2a, the service clusters in the M level has the maximum service finding efficiency.

Simulation experiment was also designed to verify the relation between the service aggregation and the quality granularity. The K-means algorithm was adopted to cluster the Web services as the service clusters and the semantic similarity for two Web services W1 and W2 is defined as:

$$p = \frac{\text{SeSim}(W1.C_{des}, W2.C_{des}) + \text{SeSim}(W1.I, W2.I) + \text{SeSim}(W1.O, W2.O)}{|F| + |I| + |O|}$$

Six round experiments were carried out and the semantic similarity for the services in a service cluster is 0.7, 0.75, 0.8, 0.85, 0.9 and 0.95 for each round, respectively. Figure 2b shows that the

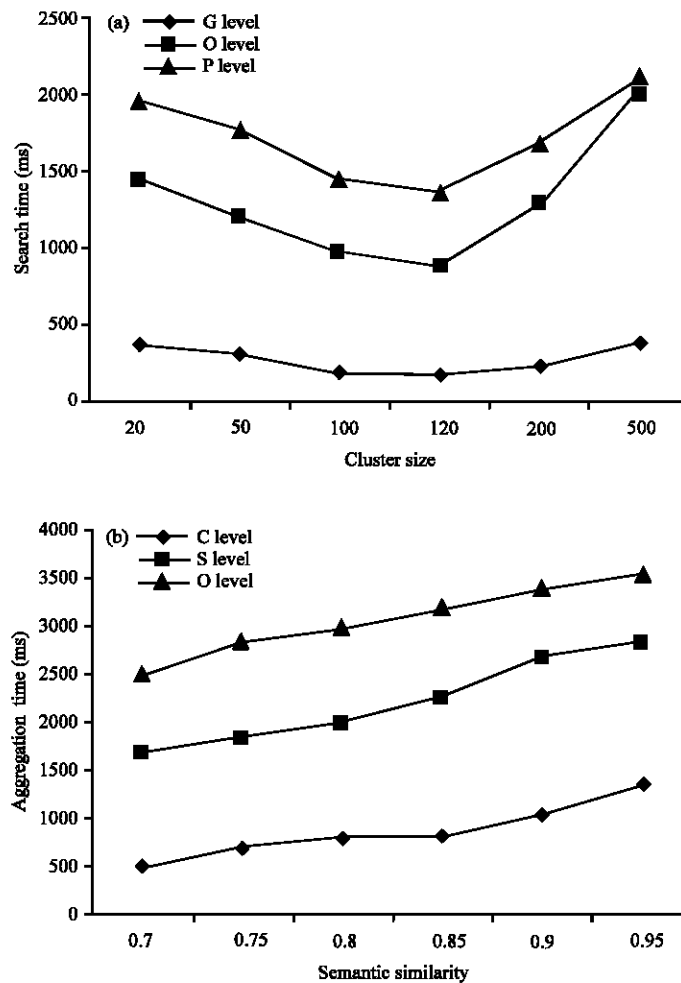


Fig. 2(a-b): (a)Service discovery time and (b) aggregation time for different granularities

aggregation time of service clusters in the C level is lower than that in the S level and the aggregation time of service clusters in the S level is lower than that in the O level.

CONCLUSION

In this study, the concept of service cluster is introduced first and then the architecture and the binding algorithm are proposed. The granularity division of service clusters is analyzed from three aspects: interface, quantity and quality. For each aspect, three levels are divided and the division rules are presented. Simulation experiments are carried out to reveal the relation of the service discovery, aggregation and their granularity divisions.

ACKNOWLEDGMENTS

This study is supported by the National Natural Science Foundation of China under grant 61170078; the doctoral program of higher education of the specialized research fund of China under grant 20113718110004; the Scientific and Technological Developing Program of Shandong Province of China under grant 2011GGX10114.

REFERENCES

- Ardagna, C.A., S. de Capitani di Vimercati, S. Paraboschi, E. Pedrini, P. Samarati, M. Verdicchio and M. Verdicchio, 2011. Expressive and deployable access control in open Web service applications. *Serv. Comput. Transac.*, 2: 96-109.
- Benatallah, B., Q.Z. Sheng and M. Dumas, 2003. The self-serv environment for Web services composition. *IEEE Internet Comput.*, 7: 40-48.
- Du, Y.Y. and Q. Hu, 2013. Soundness analysis of logic service net based on service clusters. *J. Software Eng.*, 7: 30-38.
- Liu, X., G. Huang and H. Mei, 2009. Discovering homogeneous web service community in the user-centric web environment. *IEEE Trans. Serv. Comput.*, 2: 167-181.
- Maguire, T., D. Snelling and T. Banks, 2006. Web services service group 1.2. OASIS Standard, OASIS Standard, 1 April 2006, pp: 1-41.
- She, W., I.L. Yen, B. Thuraisingham and E. Bertino, 2009. The SCIFC model for information flow control in web service composition. *Proceedings of the IEEE International Conference on Web Services*, July 6-10, 2009, Los Angeles, CA., USA., pp: 1-8.
- Sheng, Q.Z., B. Benatallah, Z. Maamar and A.H.H. Ngu, 2009. Configurable composition and adaptive provisioning of web services. *IEEE Trans. Serv. Comput.*, 2: 34-49.
- Zhong, J., H. Zhu, J. Li and Y. Yu, 2002. Conceptual graph matching for semantic search. *Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces*, Volume 2393, July 15-19, 2002, Borovets, Bulgaria, pp: 92-106.