Service Composition Based on Semantic Vocabulary

YuHui Ning, ShuXia Yu, YuYue Du and Wei Liu
College of Information Science and Engineering, Shandong University of Science and Technology,
579 Qianwangang Road, Huangdao, China
{ningyuhui, syuqd, yydu001,yanchunchun9896}@163.com

Keywords: Web Service, Service Composition, Semantic Vocabulary, Concept Replacement, Service Clustering.

Abstract: With the development of Web services, semantics is introduced into Web services, and the information recognition ability can be enhanced with the semantic description of Web services. Thus, the technology of service composition and binding can be improved. At present, the main method of service composition, discovery, binding and replacement is similarity computing based on an ontology tree. However, constructing the ontology tree and deploying the concepts in it are difficult, and the operability of service composition with similarity computing cannot meet user requirements. To solve the above problem, a new method of service composition is introduced based on semantic vocabulary. The constructing methods of the semantic vocabulary, concept replacement, service clustering, and service composition are given. Finally, the validity and correctness of proposed methods are illustrated by simulations and the comparative analysis of simulation results.

1 INTRODUCTION

With the development of information technology, Web services based on XML (Extensible Markup Language) have developed rapidly (Sun and Jiang, 2008). Web services can be used distributed on a cross-platform system. At present, the service providers keep releasing many services into Internet. Although there are many choices for users, the efficiency and precision of service discovery are decreased. In order to solve this problem, the concepts of service groups (Liu *et al*. 2007), service pools (Sheng *et al*. 2009), service clusters (Deng and Du, 2013, Gao *et al*. 2006) and service communities (Quan *et al*. 2009, Liu *et al*. 2009) have been introduced. Their main idea is that the services with the similar functions should be clustered before binding a particular service to a composite service (Hu and Du, 2013). Clustering services can minimize the cardinal number for service discovery. Thus, the service discovery can be sped up.

WSDL (Web services Definition Language) has some limitations (Deng and Du, 2013). The data of services described with WSDL cannot be recognized completely by the computers. Thus, the ability of service clustering is limited.

With the research of Web services, many scholars introduce the semantics into Web services (Xiong *et al*. 2010). The information recognition ability can be enhanced with the semantic description of Web services. Thus, the service composition and binding would be improved. The current main method for service composition, discovery, binding and replacement is similarity computing based on an ontology tree (Wu *et al*. 2005). However, constructing an ontology tree and deploying the concepts in it remain a difficult task. The operability of service composition based on similarity computing (HAN *et al*. 2009, and Xie, 2011) is insufficient.

To solve the above problems, a method for composing Web services based on semantic vocabulary is introduced in this paper. The constructing methods of semantic vocabulary, concept replacement, service clustering, and service composition are given. At last, the validity and advantage of the proposed methods are illustrated by simulations.

The rest of this article is organized as follows. The overall design is introduced in Section 2. A semantic vocabulary is constructed in Section 3. The method of service clustering is given in Section 4. The service composition is introduced in Section 5. Simulations and comparative analysis are given in Section 6. Concluding remarks are made in Section 7.

Ning Y., Yu S., Du Y. and Liu W.
Service Composition Based on Semantic Vocabulary.
DOI: 10.5220/0005426402230228
In *Proceedings of the Fourth International Symposium on Business Modeling and Software Design (BMSD 2014)*, pages 223-228
ISBN: 978-989-758-032-1

2 OVERALL DESIGN

The overall design of service composition is given in this section. The advantage of the methods proposed in this work is introduced.

The procedure to realize service composition is shown as follows. Firstly, the semantic vocabulary is constructed via expert judgment (Wu, 2007 and Wang, 2008). The service concept class set and instance set are defined in semantic vocabulary. For example, a service concept class can include {Book Ticket, Charge, Grade …}, The instance of the concept "Book Ticket" can include {Boat Ticket, Air Ticket …}. Second, the descriptive concepts in Web services are replaced with the class concept based on the semantic vocabulary. The services are clustered based on the types of services. Finally, service composition is oriented to user demands.

The advantages of the methods proposed in this paper are described below. Firstly, from the procedure to the realization of service composition, the semantic vocabulary can be obtained from the expert judgment. The service concept class and concept instance can be obtained from semantic vocabulary. The descriptive concepts of services can be unified before service composition. The information recognition ability can be enhanced with the semantic description of services. Thus, the service composition and binding can be improved. Second, the types of services can be obtained from the expert judgment before service clustering. The overall design is shown in Figure 1.

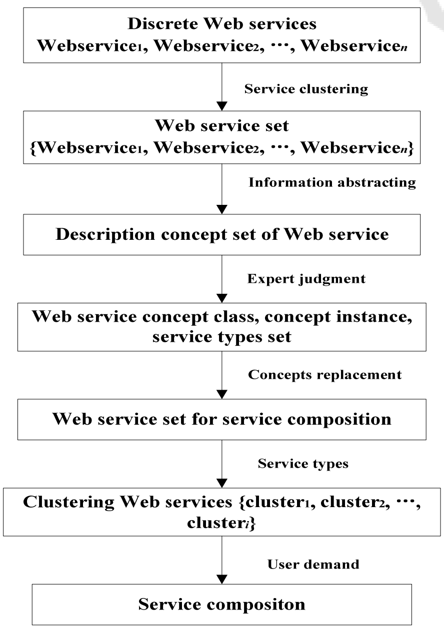

Figure 1: The overall design of service composition.

3 CONSTRUCTION OF SEMANTIC VOCABULARY

The service and service semantic vocabulary are defined in this section. The constructing algorithm for semantic vocabulary is given next.

Definition 1. *Service*=(*No*, *Cons*) is a Web service, where

(1) *No* denotes the unique label of the service; and

(2) *Cons* denotes the descriptive concepts of the service.

Definition 2. *Ctable*=(*Class*, *Instance*, *Relation*, *Category*) is the semantic vocabulary, where

(1) *Class* denotes the service concept class;

(2) *Instance* denotes the service concept instance;

(3) *Relation* denotes the relations between concept classes and instances; and

(4) *Category* denotes the types of services.

Algorithm 1: Construction of semantic vocabulary.

Input: Service set $Tp=\{Wservice_1, Wservice_2, \ldots, Wservice_n\}$.

Output: Semantic vocabulary *Ctable*=(*Class*, *Instance*, *Relation*, *Category*).

Step 1: Create a new semantic vocabulary *Ctable*, and *Ctable*.*Class*=*Ctable*.*Instance*=*Ctable*.*Relation* =∅. Create a concept variable $K=\varnothing$. Create a one-dimension array $A[e]$, and $e=+\infty$.

Step 2: Traverse a service set $Tp=\{Wservice_1, Wservice_2, \ldots, Wservice_n\}$.

2.1: Suppose that the current item of *Tp* is $Wservice_t$, traverse $Wservice_t.Cons$.

2.1.1: Suppose that the current item of *Cons* is Con_j, put Con_j to *K*.

Step 3: Traverse *K*.

3.1: Suppose that the current item of *K* is K_o, a result can be obtained via the expert judgment. If K_o belongs to the concept class, copy K_o to *Ctable*.*Class*.

Step 4: Copy *K* to *Ctable*.*Instance*.

Step 5: Traverse *Instance*.

5.1: Suppose the current item is $Instance_p$, traverse *Ctable*.*Class*.

5.1.1: Suppose that the current item is $Class_q$, a result can be obtained via the expert judgment. If $Class_q$ is the concept class of $Instance_p$, put the relation=<$Class_q$, $Instance_p$> to *Ctable*.*Relation*.

Step 6: Traverse *Ctable*.*Class*.

6.1: Suppose that the current item is $Class_q$, and there are already *w* types of services. A result can be obtained via the expert judgment. If $Class_q$

belongs to the rth type of a service, and $r<w$, then, put $Class_q$ to $A[r]$. Else, put $Class_q$ to $A[w+1]$.

Step 7: If the number of types of services is z, then put $A[1]$, $A[2]$, …, $A[z]$ to *Ctable*.*Category*.

Step 8: Output semantic vocabulary *Ctable*.

Algorithm 1 presents a method for constructing semantic vocabulary. Its workflow is shown in Figure 2.

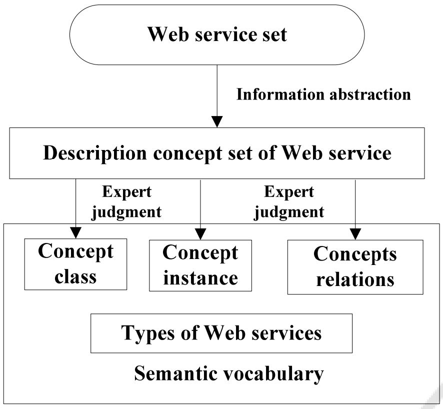

Figure 2: Workflows of constructing vocabulary.

4 SERVICE CLUSTERING

The method for service clustering is given in this section. The algorithm of service clustering is given as follows.

Algorithm 2: Service clustering

Input: Service set $Tp=\{Wservice_1, Wservice_2, \ldots, Wservice_n\}$, semantic vocabulary $Ctable=(Class, Instance, Relation, Category)$.

Output: Service cluster set *Clusters*.

Step 1: From *Ctable*, obtain the number of types of services i. Create a one-dimension array $A[i]=\{0\}$. Create a service cluster set $Clusters=\varnothing$.

Step 2: Traverse service set Tp.

2.1: Supposing that the current item is $Wservice_j$, traverse $Wservice_j.Cons$.

2.1.1: Supposing that the current item is Con_k, traverse *Ctable*.*Relation*.

2.1.1.1: Suppose that the current item is $Relation_q$=<*class*, *instance*>. If *instance* is Con_k, then replace Con_k with *class* in $Relation_q$.

Step 3: Traverse *Ctable*.*Category*.

3.1: Supposing that the current item is $Category_r$, traverse the concepts in $Category_r$.

3.1.1: Supposing that the current item is Con_e, traverse Tp.

3.1.1.1: Supposing that the current item is $Wservice_j$, traverse $Wservice_j.Cons$.

3.1.1.1.1:Suppose that the current item is Con_w. If $Con_w==Con_e$, then put $Wservice_j$ to $A[r]$.

Step 4: Put $A[1]$, $A[2]$, …, $A[i]$ to *clusters*, and output *clusters*.

From Algorithm 2, the descriptive concepts in services are unified based on semantic vocabulary. Services are clustered according to the types of services. The workflow of service clustering is shown in Figure 3.

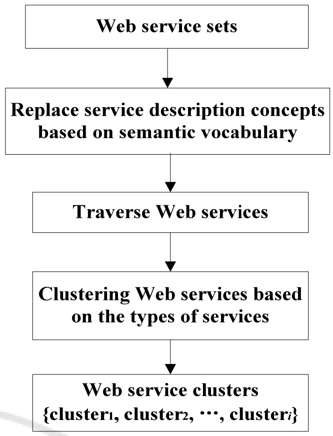

Figure 3: Workflows of service clustering.

5 SERVICE COMPOSITION

The method for service composition is introduced in this section. The user requirements are defined and the workflow of service composition is described. Four types of service composition are given. The first one is the simple composition. It means that the user requirements can be satisfied by a single service. The second is sequence composition. It means that the requirements can be satisfied by some service executed consecutively. The third is parallel composition. It means that the requirements can be satisfied by some services executed in parallel. The last is mixed composition. It is the type mixed by sequence and parallel composition.

From the types of composition, the results of the service composition binding to a user are a set of services in an order. The user requirements are introduced as follows.

Definition 3. $Ur=(No, Cons, Relation)$ is a user requirement, where

(1) *No* denotes the unique label of a user requirement.

(2) *Cons* denotes the descriptive concepts of a user requirement. The structure of *Cons* is $Cons=(v, w, \ldots, k)$, where v, w, and k are descriptive concepts, and denote the atomic demand.

(3) *Relation* denotes the relations between descriptive concepts in a user requirement. The structure of *Relation* is $Relation=(x\&i\&\ldots\&p)\|(q\|e\|\ldots\|n)$, where "||" denotes logic "or", and "&" denotes logic "and".

The algorithm of service composition is shown as follows.

Algorithm 3: Service composition

Input: User requirement $Ur=(No, Cons, Relation)$. Semantic vocabulary $Ctable=(Class, Instance, Relation, Category)$. Service cluster set $Acluster=\{cluster_1, cluster_2, \ldots, cluster_i\}$.

Output: The result of service composition.

Step 1: Traverse *Ur*.*Cons*.

1.1: Suppose that the current item is $Cons_j$, and traverse the item of it.

1.1.1: Suppose that the current item is Con_k, and traverse the item of *Ctable*.*Relation*.

1.1.1.1: Suppose that the current item is $Relation_q$=<*class*, *instance*>. If the *instance* is Con_k, replace Con_k with *class* in $Relation_q$.

Step 2: Traverse the atomic demands in *Ur*.*Cons*. Suppose that the current item is $Cons_i$. Traverse *Ctable*.*Category*. Create one-dimension array $a[k]=\{0\}$.

2.1: Suppose that the current item is $Category_r$. Traverse $Category_r$.

2.1.1: Suppose that the current item is Con_e. Traverse $Cons_i$.

2.1.1.1: Suppose that the current item is Con_j. If $Con_j==Con_e$, $a[r]$++.

2.2: Suppose that the maximum value of $a[k]$ is a[t]. Traverse $cluster_t$, and set $a[k]=\{0\}$.

2.2.1: Suppose that the current item is $Wservice_t$. Traverse $Wservice_t.Cons$.

2.2.1.1: Suppose that the current item is Con_w. Traverse $Cons_i$.

2.2.1.1.1: Suppose that the current item is Con_j. If $Con_j==Con_w$, $a[t]$++.

2.3: If the maximum value of $a[k]$ is a[m], then replace $Cons_i$ in *Ur*.*Relation* with $Wservice_m$.

Step 3: Output *Ur*.*Relation*.

The workflow of service composition is shown in Figure 4.

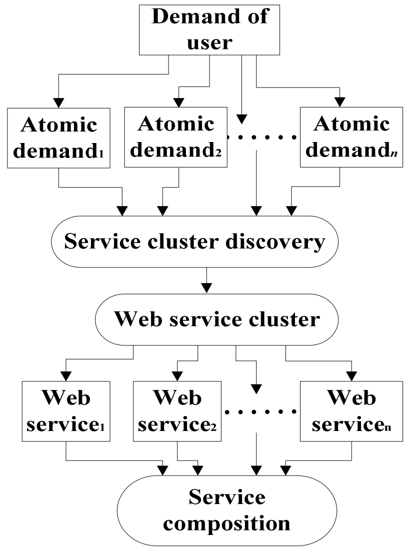

Figure 4: Workflow of service composition.

6 EXPERIMENTS

To show the advantages of the proposed methods, the experiment of service composition is given based on semantic vocabulary in this section.

(1) Construction of services

From Definition 1, 1000 services are defined in Sheet 1 of Microsoft Excel 2007. A part of services in Sheet 1 is shown in Figure 5.

No	*Cons*
1	Book Ticket, Air Ticket
2	Sells Ticket, Boat Ticket
3	Book Hotel, ABC Hotel
4	Weather Report, USA

Figure 5: A part of services in Sheet 1.

From Figure 5, each line of Sheet 1 denotes a service. The unique label of a service is defined in the first column of Sheet 1, and the descriptive concepts of the service are defined in the second column. For example, from Figure 5, $Service_1=(No, Cons)$, where $Service_1.No=1$, and $Service_1.Cons=\{Book\ Ticket, Air\ Ticket\}$.

(2) Construction of semantic vocabulary

From Definition 2 and Algorithm 1, the semantic vocabulary can be obtained, and created in Sheet 2 of Microsoft Excel 2007. A part of semantic vocabulary in Sheet 2 is shown in Figure 6.

Class	*Instance*	*Relation*	*Category*
Sells Ticket	Book Ticket	<Sells Ticket, Book Ticket>	{Sells Ticket, Air Ticket}
Buy Ticket	Purchase Ticket	<Buy Ticket, Purchase Ticket>	<Book Hotel, USA>
Air Ticket	Airplane Ticket	<Air Ticket, Airplane Ticket>	<EMS, China>
Weather Forecast	Weather Report	<Weather Forecast, Weather Report>	Weather Forecast

Figure 6: A part of semantic vocabulary in Sheet 2.

From Figure 6, the service concept classes are given in the first column of Sheet 2, and the service concept instances are shown in the second column. The relations between concept classes and instances are given in the third column, and the types of services are described in the fourth column. For example, from Figure 6, the first type of service is presented by {*Sells Ticket*, *Air Ticket*}.

(3) Concept replacement

From Algorithm 2, the descriptive concepts of the services defined in Sheet 1 can be unified according to the service concept classes defined in Sheet 2. To improve the efficiency, the procedure for concept replacement can be programmed by Excel VBA (Visual Basic for Application) (Hu, 2014). It is convenient to extract and edit the data in Excel Sheets by VBA. The workflow of a VBA program for realizing service clustering is shown in Figure 7.

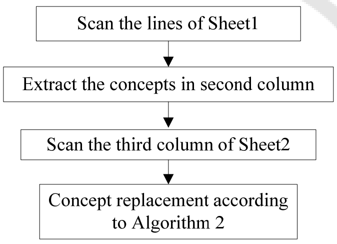

Figure 7: The workflow of a VBA program.

Sheet 1 has been changed after concept replacement. It is shown in Figure 8.

No	*Cons*
1	Sells Ticket, Air Ticket
2	Sells Ticket, Boat Ticket
3	Book Hotel, ABC Hotel
4	Weather Forecast, USA

Figure 8: Sheet 1 of Excel after concept replacement.

From Figures 7 and 8, the concept "*Book Ticket*" in $service_1$ is replaced by "*Sells Ticket*", $service_4$ is also changed, and so on.

(4) Service clustering

From Algorithm 2, fifty service clusters are obtained by using VBA programs, and created in Sheet 3. A part of service clusters in Sheet 3 are shown in Figure 9.

Service Clusters	*Web services*
$cluster_1$	$service_1$, $service_2$, $service_{58}$, se
$cluster_2$	$service_5$, $service_6$, $service_{46}$, ser
$cluster_3$	$service_3$, $service_4$, $service_{105}$, ser

Figure 9: A part of service clusters in Sheet 3.

In Figure 9, the names of service clusters are given in the first column of Sheet 3, and the services in clusters are given in the second column. For example, from Figure 9, $cluster_1$={*service*1, $service_2$, $service_{58}$,…}.

(5) Service composition oriented to user requirements

From Definition 3, fifty user demands are defined in Sheet 4. A part of user demands are shown in Figure 10.

No	*Cons*	*Relation*
1	{Book Hotel}	{Book Hotel}
2	{Ser Tichet, Air Tichet},{Book Hotel}	{Ser Tichet, Air Tichet}||{Book Hotel}
3	{Buy, Computer},{Express Delivery, Ems},{Book Hotel}	{Buy, Computer}&{Express Delivery, Ems}||{Book Hotel}

Figure 10: A part of user demands in Sheet 4.

From Algorithm 3, a service composition can be obtained, and created in Sheet 5. A part of service composition is shown in Figure 11.

Demand	Service composition	Service binding
1	Simple composition: $service_3$	{3}
2	Parallel composition: $service_1 \|\| service_3$	{1,3}
3	Mixed composition: $service_8 \& service_6 \|\| service_3$	{8,6,3}

Figure 11: A part of service composition.

(6) Comparison

From Figure 11, the user demands can be satisfied by the service composition based on the methods proposed in this paper. The result of service clustering without semantic concept replacement is shown in Figure 12.

Clusters	*Web services*
$cluster_1$	$service_2$, $service_{72}$, $service_{119}$, $service_{576}$
$cluster_2$	$service_{46}$, $service_{345}$, $service_{251}$, $service_{727}$
$cluster_3$	$service_3$, $service_{105}$, $service_{733}$, $service_{821}$

Figure 12: Clusters without semantic concept replacement.

From Figures 11 and 12, obviously, the service clustering ability in Figure 12 is decreased. For example, the descriptive concepts "*Book Ticket*" in $service_1$ are not a concept class. The computer cannot recognize them from the service clustering. Thus, $service_1$ does not appear in $cluster_1$ of Figure 12. Moreover, the service composition ability can be decreased.

Therefore, the validity and advantages of the proposed method are illustrated.

7 CONCLUSIONS

To improve service composition, a new method of service composition is proposed based on semantic vocabulary in this paper. A semantic vocabulary is constructed and some algorithms are presented such as service concept replacement, service clustering, and service composition. Finally, the validity and advantages of the proposed methods are illustrated by simulations and comparative analysis.

Further work will be the data processing technology of the services oriented to big data. Moreover, the platform construction of service composition will be considered based on cloud computing.

ACKNOWLEDGEMENTS

This work is supported by the National Basic Research Program of China under grant 2010CB328101; the National Natural Science Foundation of China under grants 61170078 and 61173042; the Doctoral Program of Higher Education of the Specialized Research Fund of China under grant 20113718110004; Basic Research Program of Qingdao City of China under grant No. 13-1-4-116-jch; and the SDUST Research Fund of China under grant 2011KYTD102.

REFERENCES

Deng, S.Y., and Du, Y.Y., 2013. Web service composition approach based on service cluster and Qos. *Journal of Computer Applications*. 33(8), 2167-2170.

Deng, S.Y., and Du, Y.Y., 2013. Web service composition approach based on service cluster and Qos. *Journal of Computer Applications*. 33(8), 2167-2170.

Gao, Y., Na, J., Zhang, B., Yang, L., 2006. 3-layer Web services organization model for dynamic service composition. Mini-Micro Systems, 27(10), 1879-1882.

Hu, Q., Du, Y.Y., 2013. Service architecture and service discovery oriented to service clusters. *Journal of Computer Applications*. 33(8): 2163-2166.

Hu, X., 2014. Automatic transmission of delivery information based on GPRS. Journal of Neijiang Normal University, 18(9): 107-109.

HAN, S., WANG, H.Y., CUI, L.Z., 2009. A user experience oriented service discovery method with clustering technology [C]// Second Int. Symposium on Computational Intelligence and Design, China, Chang Sha: *IEEE Computer Society*, 64-67.

Liu, S.L., Liu, Y.X., Zhang, F., Tang, G.F., 2007. A dynamic Web services selection algorithm with Qos global optimal in Web services composition. *Journal of Software*, 18(3), 646-656.

Liu, X.Z., Huang, G. and Mei, H., 2009. Discovering homogeneous Web service community in the user-centric Web environment. *IEEE Transactions on Service Computing*, 2(2), 167-181.

Quan, Z.S., Boualem, B., Zakaria, M., Anne, H.H.N., 2009. Configurable composition and adaptive provisioning of Web services. *IEEE Transactions on Services Computing*, 2(1), 34-49.

Sun, P., Jiang, C.J., 2008. Using service clustering to facilitate process-oriented semantic Web service discovery, *Chinese Journal of Computers*, 31(8), 1340-1353.

Sheng, Q.Z, Benataliah, B., Maamar, Z., 2009. Configurable composition and adaptive provisioning of Web services. *IEEE Transactions on Service Computing*, 2(1), 34-49.

Wu, J., Wu, Z.H., Li, Y., Deng, S.G., 2005. Web service discovery based on ontology and similarity of words. *Chinese Journal of Computers*, 28(4), 595-602.

Wu, J.Y., 2007. Requirements and configuration management of project. *Microcomputer & its Applications*, S1(2):112-114.

Wang, X.P., Wang, F.J., Wang, W.W., 2008. The certification research of enterprise quality management system. *Management Review*, 10(8): 131-134.

Xiong, P C, Fan, Y., Zhou, M.C., 2010. A Petri net approach to analysis and composition of web services. *IEEE Transactions on Systems*, Manand Cybernetics, Part A, 40(2): 376–387

Xie, L.L., 2011. Ontology-based semantic web services cluster and discovery. *Tianjin University*, China.