Journal of Software Engineering

ISSN 1819-4311

Academic Journals Inc.

www.academicjournals.com

Journal of Software Engineering 8 (2): 100-107, 2014
ISSN 1819-4311 / DOI: 10.3923/jse.2014.100.107

An Efficient Algorithm of Service Discovery Based on Service Clusters

JunJing Gai and YuYue Du
Shandong University of Science and Technology, 266590, Qingdao, China

Corresponding Author: YuYue Du, Shandong University of Science and Technology, 266590, Qingdao, China

ABSTRACT

A service discovery algorithm based on service clusters is proposed in this study. By using ontology and Petri nets, the formal definition of services is given and service clusters are formally represented as service cluster net units. The usefulness of the algorithm is demonstrated by an experiment.

Key words: Petri net, service discovery, service cluster, algorithm

INTRODUCTION

With the development of Service Oriented Architecture (SOA), the number of services is becoming larger and larger. Due to the growth of Web services, service discovery faces a huge challenge.

The semantic Web Service can describethe function of services. The service matching based on semantic description can understand the semantic information of services and service requests well and improve theaccuracy of semantics.

Mass services in a libraryare clustered by using the clustering technology. The functions of the services in a cluster are similar. However, the functions of the services in several different clusters are highly dissimilar. The service clustering can improve the accuracyof the serviceclassification and reduce the size of service searching. Therefore it can improve the efficiency of service discovery.

Petrinets are atool for modeling and analysisof distributed systems and have the strict mathematical definition and the ability of graphic expression. It can effectively describe the information processing systems with concurrency and uncertainty. Service clusters have the nature of parameter uncertainty. Petrinets will be used to describe and analyze service clusters in the study.

With the wideapplication ofsemantic Web, the service discovery methods is researched extensively based on semanticdescription. In Pan and Zhang (2009), a service discovery method is proposed and it improves the performance of services by combining the syntax and semantics. In Wu *et al.* (2005), a service discovery method is proposed based on ontology and semantic similarity. The semantic similarity of services is discussed by constructing service ontology. They can improve the accuracy of the service discoveryfrom theperspective of semantics. But the efficiency of service discovery is not increased. And the service compositions satisfied a service request can not be found by previous methods efficiently.

The efficiency of the traditional service discovery is enhanced in this study. Services are described based on semantics and they are clustered by semantic similarity (Rajagopal and Selvi, 2006). Through clustering services, the searching scale can be reduced greatly. Service clusters are described formally and service cluster net units are introduced. By using service cluster net units

proposed in this study, the services can be quickly found. And service compositions can be built efficiently. The efficiency and the accuracy of service discovery are greatly improved.

In this study, R is a matrix and R^p represents the p'th row of R. $R^p(q)$ is the q'th value in the p'th row of R. When p = 1, the q'th value in the first row of R can be expressed by R (q).

BASIC CONCEPTS

Ontology and Petri nets are briefly reviewed in this section. In this study, N is a natural number set, i.e., $N = \{0, 1, 2, \cdots\}$; $N_k = \{0, 1, 2, \cdots, k\}$; $N^+ = N \setminus \{0\}$; and $N_k^+ = N_k \setminus \{0\}$.

Ontology can be represented as an oriented tree, an ontology tree. The nodes of the ontology tree represent concepts.Connections between the nodes correspond to the semantic concepts.

Definition 1 (Semantic similarity): Concepts X and Y are similar in the semantics if and only if X and Y are synonymous, or the similarity of X and Y is higher than a certain value.

Definition 2 (Ontology concept set): An ontology concept set is a set of some ontology concepts, represented by $C = \{c_1, c_2, \ldots, c_n\}$, where c_i is an ontology concept, $n \in N^+$, $i \in N_n^+$.

Petri nets can not only describe the structure of a system but also simulate the operation of the system (Du *et al*., 2008). Petri nets (Murata, 1989) are defined as follow.

Definition 3 (Petri net): N = (S, T; F) is a net where:

- $S \cup T \neq \Phi$
- $S \cap T \neq \Phi$
- $F \subseteq (S \times T) \cup (T \times S)$
- $Dom(F) \cup cod(F) = S \cup T$, where $dom(F) = \{x \in S \cup T : \exists y \in S \cup T: (x, y) \in F\}$ and $cod(F) = \{ x \in S \cup T : \exists y \in S \cup T: (y, x) \in F\}$

Petri nets are used to describe services and service clusters in this study.

SERVICES AND SERVICE CLUSTERS

The formal description of Web services can be given from Definition 1.

Definition 4 (Web service): A service can be represented by WS = (BI, SF, I, O, QoS) where:

- BI is the basic information of a service
- SF is the functions of a service
- I is the input set of a service and $\forall i \in I$, i = (i.concept, i.value) where:
  - i.concept is the ontology concept corresponding to the input parameter
  - i.value = {0, 1}, i.value = 0 represents the input parameter can not be given and i.value =1 represents the input parameter can be given
- O is the output set of a service and $\forall o \in O$, o = (o.concept, o.value) where:
  - o.concept is the ontology concept corresponding to the output parameter
  - o.value = {0, 1}, o.value = 0 represents the output parameter can not be obtained and o.value = 1 represents the output parameter can be obtained and
- QoS is the quality of the servic

In Definition 4, BI includes the service name, the service provider, etc. QoS includes the response time, the price, etc. Similarly, a service request can be defined as follow.

Definition 5 (Service request): $SR = \{SF_r, I_r, O_r, QoS_r\}$ is a service request where:

- Sf_r is the functions needed to be obtained
- I_r is the set of the input request. $\forall ir \in I_r$, ir = (ir.concept, ir.value)
- O_r is the output set of the request and $\forall or \in O_r$, or = (or.concept, or.value) and
- QoS_r is the set of the qualities needed to be obtained

To improve the efficiency of service discovery, service clusters are proposed. Services are clustered (Skoutas *et al*., 2010) according to their functions. The services with similar functions are in a same cluster. Petri nets can describe the uncertainty and concurrency of a service cluster. By Definition 3, the formal description of service clusters can be defined as follow.

Definition 6 (Service cluster net unit): CNU = {CF, tu, Iu, Ou,Fu} is a service net unit where:

- CF is the function of the service cluster
- tu is the transition of a service cluster net unit
- Iu is the input parameter set of the services in the service cluster and $\forall iu \in Iu$, iu = (iu.concept, iu.lable) where:
    - iu.concept is the ontology concept corresponding to the input parameter
    - iu.lable is an n×1 matrix where n is the number of the services in the service cluster and iu.lable is the value corresponding to iu.concept for a service
- Ou is t he output parameter set of the services in the service cluster and $\forall ou \in Ou$, ou = (ou.concept, ou.value) where:
    - ou.concept is the ontology concept corresponding to the output parameter
    - ou.lable is an n×1 matrix where n is the number of the services in the service cluster andou.lable is the value corresponding to ou.concept for a service; and
- $Fu \subseteq (Iu \times tu) \cup (tu \times Ou)$ is a set of directed arcs

Example 1: There exists three services: ws_1, ws_2 and ws_3, where ws_1 = {(i_1,departure, 1), (i_2, date, 1), (i_3, arrival, 1), (i_4, ID, 1), (o_1, time, 1), (o_2, price, 1), (o_3, fightno, 1)}, ws_2 = {(i_1,departure, 1), (i_2, date, 1), (i_3, arrival, 1), (o_1, time, 1), (o_2, price, 1)} and ws_3 = {(i_1, departure, 1), (i_2, date, 1), (i_3, arrival, 1), (i_4, type, 1), (o_1, time, 1), (o_2, price, 1), (o_3, flightno, 1), (o_4, flighttype, 1)}. The service cluster net unit can be shown in Fig. 1.

In Fig. 1, Iu and Ou are two matrices and they are called the input matrix and the output matrix. In an input matrix, the input parameters of services in a service cluster are numbered and each row is the values corresponding to the input parameters.

By service cluster net units, the scale of service discovery is greatly reduced and the efficiency of service searching can be improved. While by the formal description of service clusters, a novel service discovery algorithm can be proposed to further improve the efficiency.

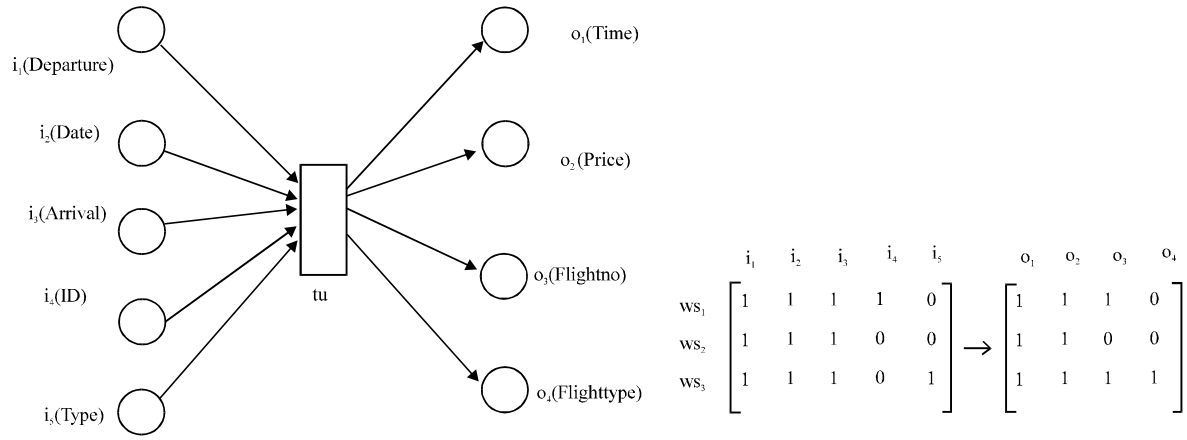

Fig. 1: A service cluster net unit

SERVICE DISCOVERY ALGORITHM

A service discovery algorithm based on service clusters is proposed in this section. Firstly, a service cluster discovery algorithm is presented to reduce the searching scale based on service clusters. Then, a service matching algorithm is given to reduce the difficulty of the service discovery and a service discovery algorithm in the service cluster is shown to further improve the efficiency of the searching services. A service composition algorithm is proposed to search the service compositions satisfying a service request. In the follow, the service satisfying a service request is called an atomic service.

In the study, the number of services is n, the number of the service clusters is w and the number of the services in the service cluster is m, $n \in N^+$, w, $m \in N_n^+$. Let the services be $Ws_i = (BI_i, SF_i, I_i, O_i, QoS_i)$ and the number of input and output parameters are n_i and n_o respectively. $CNU_k = \{CF_k, tu_k, Iu_k, Ou_k, Fu_k\}$ is a service cluster net unit, $i \in N_n^+$, $k \in N_w^+$. A service request is $s_r = \{sf_r, i_r, o_r, qos_r\}$. In the next algorithms, Iu_k, Ou_k, i_r and o_r are represented as matrices. And Iu_k is an $m \times n_i$ matrix. Ou_k is an $m \times n_o$ matrix. i_r is an $1 \times n_i$ matrix. o_r is an $1 \times n_o$ matrix. Iu_k^p is the p'th row of the matrix Iu_k. $Iu_k^p(q)$ is the q'th value in the p'th row of the matrix Iu_k.

A matching algorithm is proposed first in this subsection.

Algorithm 1: A service cluster discovery algorithm

Input: $WS_i=(BI_i, SF_i, I_i, O_i, QoS_i)$, $CNU_k=\{CF_k, tu_k, Iu_k, Ou_k, Fu_k\}$ and $s_r=\{sf_r, i_r, o_r, qos_r\}$;
Output: the service cluster net unit cnu of the service cluster that satisfy s_r;
Step 1: u=1,
Step 2: if u<=w, then
Step 2.1: if CF_k is not similar with sf_r, u=u+1;
Step 2.2: else output cnu= CNU_u;
Step 3: else cnu is not found

From Algorithm 1, the searching scale can be reduced a lot. The searching number without service clusters is n. The searching number of Algorithm 1 is w. And $w \leq n$, the efficiency of the service discovery can be improved.

The process of the matching services is very important in the process of the service discovery. Services can be found by a determination as long as the services can satisfy a certain matching conditions. A novel service matching algorithm is proposed as follow.

Algorithm 2: A service matching algorithm

Input: $CNU_k=\{CF_k, tu_k, Iu_k, Ou_k, Fu_k\}$ and $s_r=\{sf_r, i_r, o_r, qos_r\}$;
Output: A and B, where A is an $m \times n_i$ matrix and B is an $m \times n_o$ matrix.
Step 1: $p_1=1$, $p_2=1$, $q_1=1$, $q_2=1$;
Step 2: if $p_1<= m$;
Step 2.1: if $Iu_k^{p1}(q_1)=1$ and $i_r(q_1)=0$, $A^{p1}(q_1)=0$;
Step 2.2: else $A^{p1}(q_1)=1$;
Step 2.3: $q_1= q_1+1$;
Step 2.4: if $q_1> n_i$, $q_1=1$and $p_1 = p_1+1$;
Step 2.5: go to Step 2;
Step 3: if $p_2<= m$;
Step 3.1: if $Ou_k^{p2}(q_2)=0$ and $o_r(q_2)=1$, $B^{p2}(q_2)=0$;
Step 3.2: else $B^{p2}(q_2)=1$;
Step 3.3: $q_2= q_2+1$;
Step 3.4: if $q_2> n_o$, $q_2=1$and $p_2 = p_2+1$;
Step 3.5: else go to Step 4; and
Step 4: output matrix A and matrix B.

Algorithm 2 is a service matching algorithm based on service cluster net units. The time complexity of Algorithm 2 is $m\times n_i^2+m\times n_o^2$.

Matrix A is called the input matching matrix and matrix B is called the output matching matrix. $A^{p1}(q_1) = 1$ means the q_1'th parameter can be satisfied by the service ws_{p1} and $B^{p2}(q_2) = 1$ means the q_2'th parameter can be satisfied by the service ws_{p2}. Then if A^{p1}=, the service can satisfy i_r and if B^{p2} =, the service can satisfy o_r. The services can be discovered by a certain determination. A service discovery algorithm is needed.

A service discovery algorithm is represented first in this subsection.

Algorithm 3: A service discovery algorithm

Input: input matching matrix A and output matching matrix B;
Output: S : a set of services;
Step 1: p=1, $q_1=1$, $q_2=1$;
Step 2: if p<=m and $q_1<=n_i$;
Step 2.1: if $A^p(q_1)=1$, $q_1=q_1+1$;
Step 2.2: else p=p+1 and $q_1=1$;
Step 3: else if $q_1>n_i$, $q_2=1$;
Step 3.1: if p<=m and $q_2<=n_o$;
Step 3.1.1: if $B^p(q_2)=1$, $q_2=q_2+1$;
Step 3.1.2 else p=p+1, $q_1=1$ and go to Step 2;
Step 3.2 else if $q_2>n_o$, $S=S\cup\{ws^p\}$;
Step 3.1: p=p+1, $q_1=1$ and go to Step 2;
Step 4: output S.

The required services can be discovered according to Algorithm 3. In Algorithm 3, the output services in set S are that their corresponding rows of A are:

$$\underbrace{[1, 1,\ldots , 1]}_{n_i}$$

and corresponding rows of B are:

$$\underbrace{[1, 1,\ldots , 1]}_{n_o}$$

That means the service request can be satisfied by the services in the set S. The searching number of Algorithm 3 is $m\times n_i+m\times n_o$.

In a service cluster, one service may not satisfy a service request. However, a composition of several services may satisfy the service request. Service compositions conclude the parallel composition and the serialcomposition. In this study it is assumed that the granularity of service clusters is small enough and the situations of serial compositions do not exist.

From Algorithm 2, $A^{p1}(q_1) = 0$ means the input parameter q_1 is not satisfied in the service request. In other words, the service can not be fired in the service request. It is assumed that compositions with v services need to be found, $v\in N_m^+$. A service composition discovery algorithm is proposed as follow.

Algorithm 4: A service composition discovery algorithm

Input: a $v\times n_o$ matrix C, the input matching matrix A and the output matching matrix B;
Output: S: a set of services and service compositions;
Step 1: p = 0;
Step 2: if p<=m, p=p+1 and q_1=1;
Step 2.1: if q_1<=n_i and $A^p(q_1)$=1, q_1=q_1+1,
Step 2.2: else if $A^p(q_1)$=0, update all elements of A^p and B^p are 0 and go to Step 2;
Step 2.3: else go to Step 2;
Step 3: else select v rows from B_p in matrix C;
Step 4: if each column of C exists a value of 1, then S=S∪{the composition of the v services}, else select another v rows from B_p;
Step 5: if all possible combinations are selected, output S.

By Algorithm 4, the service compositions can be discovered. And when v = 1, Algorithm 4 areequivalentto Algorithm 3.

COMPARATIVE ANALYSIS

This section describes the experiments to verify the feasibility and effectiveness of the proposed algorithms. 500 servicesabout buying books online are selected from UDDI server. And the services are registered on the local server as experimental subjects. To avoid thechanciness of results, this entire experiment is repeated 100 times.

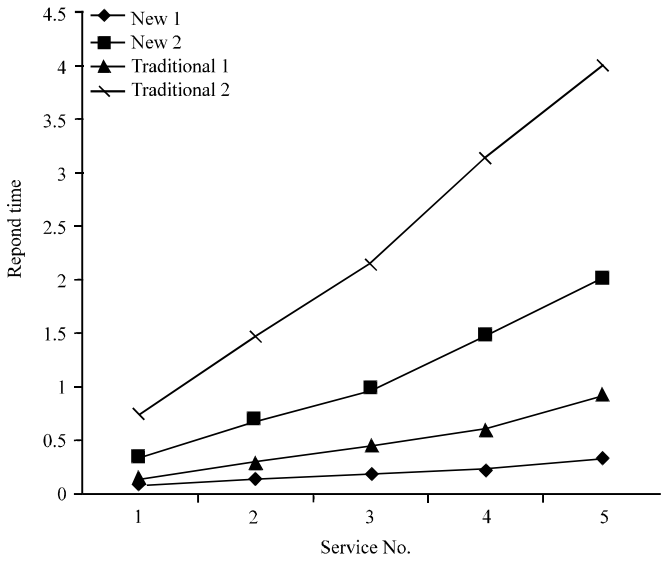

Fig. 2: Efficiency of 5 service numbers

Two experiments are done. The first experiment is atomic service discovery and the second experiment is to search service compositions of two services. The two experiments are done by a traditional algorithm (Elgazzar *et al*., 2010) and the algorithm in this study.

Figure 2 shows the results of searching services in both the traditional algorithm and in the algorithm proposed in this study. The horizontal axis is the number of services and the vertical axis is the respond time. Two lines represent the results of the first experiment and another two lines represent the results of the second experiment. The traditional algorithm is represented as "traditional" and the algorithm proposed in this study is represented as "new". In the follow analysis, the respond timeiscontrarytothe searching efficiency.

Several expected results can be found by observing Fig. 1. First, the efficiency of the algorithm proposed in this study is clearly higher than that of the traditional algorithm. Second, the efficiency is decline with the increase in the number of services. Third, the efficiency of searching service compositions is slower than that of the atomic service discovery.

CONCLUSION

A service discovery algorithm is proposed in this study. Both atomic services and service compositions can be found by the algorithm proposed in this study. And the validity of the algorithm is demonstrated by an experiment. And by the comparativeanalysis, the efficiency of service discovery can be improved a lot. The method of searching service composition can be used in the research of service composition. And service cluster net units proposed in this study can be used to analyze and study service composition and service substitution.

ACKNOWLEDGMENTS

This study is supported by the National Natural Science Foundation of China under Grants 61170078 and 61173042; the National Basic Research Program of China under Grant 2010CB328101; 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 13-1-4-116-jch; the SDUST Research Fund of China under Grant 2011KYTD102 and Graduate Innovation Foundation of Shandong University of Science and Technology under Grant No. YC130323.

REFERENCES

Du, Y.Y., C.J. Jiang and M.C. Zhou, 2008. A petri nets based correctness analysis of internet stock trading systems. IEEE Trans. Syst. Man Cybern. C Applied Rev., 38: 93-99.

Elgazzar, K., A.E. Hassan and P. Martin, 2010. Clustering WSDL documents to bootstrap the discovery of web services. Proceedings of the 2010 IEEE International Conference on Web Services, July 5-10, 2010, Miami, FL., pp: 147-154.

Murata, T., 1989. Petri nets: Properties, analysis and applications. Proc. IEEE, 77: 541-580.

Pan, S.L. and Y.X. Zhang, 2009. Ranked web service matching for service description using OWL-S. Proceedings of the International Conference on Web Information Systems and Mining, November 7-8, 2009, Shanghai, China, pp: 427-431.

Rajagopal, S. and S.T. Selvi, 2006. Semantic grid service discovery approach using clustering of service ontologies. Proceedings of IEEE TENCON Region Conference, November 14-17, 2006, Hong Kong, China, pp: 1-4.

Skoutas, D., D. Sacharidis, A. Simitsis and T. Sellis, 2010. Ranking and clustering web services using multicriteria dominance relationships. IEEE Trans. Serv. Comput., 3: 163-177.

Wu, J., Z.H. Wu and Y. Li, 2005. Web services discovery based on ontology and semantic similarity. Chin. J. Comput., 28: 595-602.