International Journal of Computational Intelligence Systems and Applications. Volume 5, Issue 1, (2017) 51-57
http://www.cayley-nielson.com/ijcisa/

Service Matching Approach Base on Similarity of Distance of Concept Semantic

Zhihai Zhan, Desheng Li
College of Information & Networking Engineering, Anhui Science and Technology University, 233100, Fengyang, 233100, China
E-mail: *zhanzhh@ahstu.edu.cn, ldsyy2006@163.com*

Abstract:
With the emergence of a large number of Web services in ubiquitous environment, how to correctly locate the appropriate Web service has become the main problem. UDDI based on key words and classification of service discovery mechanism cannot meet the current needs. Through the definition of ontology concept semantic distance to compute the similarity between ontology concepts, proposed one kind based on the similarity of Web service matching algorithm. The experimental results show that the algorithm has high matching rate and better matching speed, to be used in pervasive computing environment.

Keywords: Semantic Web Service, Matching Algorithm, Pervasive Environments**.**

1. Introduction

With the emergence of a large number of Web services, we found that the Web service to match users' demand has become the key issue of the Web service system among a number of services in the pervasive environments. The existing Web service description document WSDL [1] lack of the description of the Web service function. Service registration system UDDI [2] accurately matches the key word to discover the service through the service registration information such as service name, classification and company name etc. Such syntax-level service can't achieve the satisfaction results both in the precision and the recall sides. Therefore, in order to meet the requirements of pervasive computing environment and provide the robustness and application of the service matching algorithm looking for service accurately through the service semantic matching becomes the focus of attention.

This paper has put forward an improved service matching strategy based on the traditional algorithm, which is to use similarity computation under the pervasive environment during the ratiocination process. This strategy has combined the advantages of both methods under the certain level and has the accurate matching ability and the better operation efficiency under the worst situation.

2. Algorithm idea

The so-called Web services, is based on services provided by the requester to find the best match of the demand for services. Unlike traditional keyword-based matching service method, the semantic service discovery processes the requirements description and matching judgment in the semantic layer. Therefore it requires owl-s to describe the function information of the service and save the information in the IOPE description of the OWL-S profile with input and output parameters, then match the input and output parameters of the user request service and advertisements service by the corresponding matching

ISSN 2331-1924 / Copyright © 2017 CAYLEY NIELSON PRESS, INC.
May, 2017

algorithm and decide to finish the same function when the both services have the same input and output parameters. The basic idea of the semantic Web service discovery matching algorithm is: to semantically describe the user request service and advertisements service respectively and then match the two semantic models and eventually convert to match every concept in the same ontology. During the process of matching, the calculation of the similarity of the two concepts is the key to decide whether the matching is successful or not.

3. The services matching process and algorithms

Figure 1 shows the stars level in accordance with the service class, the stepwise method of filtering of the input and output parameters and the QoS parameters required for the matching of the service process. Registered to the service directory can use the services provided in the system n, the first system according to the type of service request these n service matching give n - p which satisfy the service request type of service; If the service request is not input output parameters and QoS parameters requirements, according to the type of service that matches the similarity to return matching services; otherwise, these

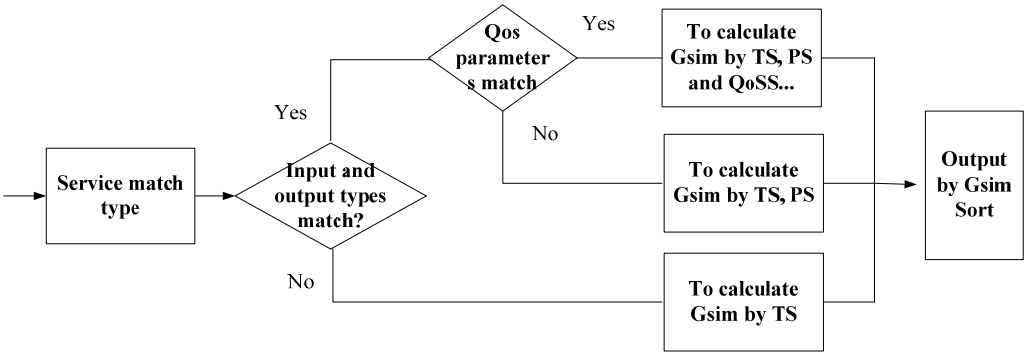

Fig.1. Service matching process

N-P service is selected on the basis of the meet the required output outputR to satisfy input inputR service (assuming obtained meet the requirements n-p-q service), If the service request does not have the requirements of QoS parameters, according to the service matches the type of similarity and the similarity of the output parameter returns the matching service; otherwise, these n-p-q filtered, according to the required QoS parameter conditions obtained n-p-q-r service; Finally, these n-p-q-r service to calculate the similarity of each service in accordance with the similarity Lowest ordering.

The researchers put forward all sorts of algorithms to the Web service matching based on the ontology library. The classic algorithm is in the documents [3]. This algorithm uses the description logic and ontology language OWL to realize the matching of Web service. The main matching subject is IOPEs parameter in service profile. The matching results between two services is defined to 4 levels, Exact, Plug-in, Subsume, Fail[4,5].

1) Exact: When the output description of service exactly equals to the output description of requirement or the requirement output is the direct subclass of the service output. The matching level is Exact.

2) Plug-in: when the output description of service includes the output description of requirement, the matching level is Plug-in.

3) Subsume: when the output description of requirement includes the output description of service, the matching level is Subsume.

4) The rest circumstance: the matching level is Fail.

Its core algorithm is expressed as follows:

```
Match(outR,outA)
{
    If outA = outR return exact;
    If outR subclassOf outA return exact;
    If outA subsume outR return plug-in;
    If outR subsume outA return subsume;
}
```

outR indicates the required output; outA indicates the provided output; subclassOf indicates the direct subclass relationship; subsume indicates the inclusion relationship.

4. Description of functional information semantic matching

Definition 1: Web service semantic description model W=<I,O>. I=<I_1,I_2,… I_n> is a concept vector indicates the semantic description of N input parameters in W of Web service. I_1,I_2, …， I_n indicates the responsible semantic description of each W input parameter in the ontology library. O=<O_1,O_2,…,O_m> is a concept vector indicates the semantic description of M input parameters in W of Web service. O_1,O_2,…O_m indicates the responsible semantic description of each W output parameter in the ontology library. Thus the Web service matching is converted to the ideal service semantic description model. W=<I, O> is matching the service semantic description model W=<I', O> in the service library and furthermore is converted to the match of concept vectors like I and I' or O and O 'in the same ontology.

The following is to define the similarity calculation function of two concepts;

Definition 2: the similarity calculation function of any two concepts in the same area of ontology library: Sim:$C_1 \times C_2$. C_1, C_2 indicates any of two concepts in the ontology library. Sim(C_1,C_2) is a positive real number to measure the similarity level of C_1,C_2 and Sim(C_1,C_2)∈[0,1]. The bigger Sim(C_1,C_2) is, the higher the similarity level of two concepts is. The following sections will explain the definition of similarity function in detail. Based on this definition, we can define the similarity of any two concept vector V and V in the same ontology library.

Definition 3: the calculation formula of the similarity of any two concepts vectors in the same ontology library.

$$s(v,v') = \sum_{i=1}^{n} \max(sim(v_i, v_j'), 1 \le j \le n, \tag{1}$$

witch: V=(V_1,V_2, …,V_n), V'=(V_1',V_2', …, V_n'), V_1,V_2, …, V_n, V_1',V_2', …, V_n' are the concept of ontology library.

Thus the semantic Web service matching is abstracted to match the Web service semantic description model and then convert the matching of the concept vectors into the matching of the concept vector elements by definition 3.

5. QoS parameters match

In Service QoS parameters matching(mainly consider the services available, service delay and cost, the present algorithm for the other QoS parameters may be appropriately extended), For the QoS parameters of the service Advservice is AdvQoS, the service requestor Reqservice the QoS parameters for ReqQoS, both to meet the QoS parameters matching condition when both satisfy the following formula.

$$\begin{aligned} &QoS\,\mathrm{Pr}\,ameterMatch(Adv_{service,}\,\mathrm{Re}\,q_{service}) = \forall pqn \\ &\in \mathrm{Re}\,q_{service}, \mathrm{Re}\,q_{QoS} \exists pqn' \in Adv_{service} Adv_{QoS} \\ &NumericExpressionMatch(pqn', pqn) \end{aligned} \tag{2}$$

Which NumericExpressionMatch (pqn', pqn) primarily use =、>、<、≤、 ≥ to compare two expressions value. When the value of pqn is a subset of pqn' value, the expression is true. Such NumericExpressionMatch (Latency = 3.5, Latency <5), because the Latency = 3.5 within the scope of Latency <5, so it true.

QoS parameters similarity calculation.For the similarity of service QoS parameters, by comparing these two values as the ratio of the degree of similarity as the parameter WQoS, the plurality of parameter pairs obtained by a process of taking the average. The similarity in the range [0, 1].

6. The algorithm of concept similarity

Service matching is eventually converted to the matching between the corresponding concept vectors of service input and output parameters in the domain ontology. And the matching level between the concepts is measured by calculating their similarity. In order to calculate the similarity of two concepts, we must calculate the semantic distance of the two concepts. The longer the semantic distance between the two concepts is, the lower their similarity is; the shorter the semantic distance is, the higher the similarity is. [7] The traditional algorithm is to calculate the similarity of two concepts by the same application without classification. This paper considers the various relationships of input and output parameters in the concepts tree of the domain ontology and uses the different methods to calculate the similarity. The reason is that the measurement of service matching to the output parameter is more important than it to the input parameter. And we only consider the top-bottom inclusion relationship between concepts in the concept tree. To the input parameters, we should consider not only the top-bottom inclusion relationship but also other binary relation existed because the Web service input parameters are provided by the users. Thus the users have more control on these parameters and also reflect some objective relationship existed in the reality.

Definition 4: the relationship between the concepts distance and concepts similarity:

$$\mathrm{Sim}(C_1, C_2) = 1/\mathrm{Dis}(C_1, C_2) \tag{3}$$

C_1, C_2 are any two concepts in the domain ontology, $\mathrm{Dis}(C_1,C_2)$ indicates the distance between two concepts. During the calculation of the corresponding concepts similarity of Web service output parameters, this paper defined as the shortest distance contacting the two concepts points formed by the top-bottom relationship in the ontology. The improved Web service output concepts distance algorithm is:

1) input the O corresponding symbol class1 and the O' corresponding symbol class2, depth=0，try_list=Φ, abandoned_list=Φ;

2) add class1to the head of try_list;

3) if try_list is not empty process as follows：take away the first element u from the try_list:

a) if Synset(u) includes class2，depth＝depth，return to algorithm;

b) utilize the reasoning machine to get the concept collection H(u) which has the up-down relationship with u from the ontology;

c) depth＝depth＋1;

d) if H(u) includes class2，return to depth;

e) to every element v in H(u)；if abandoned_list and try_list includes v, then continue；add v to the end of the try_list;

f) add u to the abandoned_list.

Synset(u) indicates to use the reasoning machine to catch all the synonymous concepts of u; The try_list is a list which the elements indicate the non-traversal concepts in the ontology； the abandoned_listis also a list which the elements indicate the ergodic concepts in the ontology. The concept tree will be traversal from requester output services by using the width priority method in this algorithm, end at traversing all the nodes [8].

The improved Web service input concepts distance algorithm is:

1) input the I corresponding symbol class1 and the I' corresponding symbol class2, try_list=Φ，abandoned_list=Φ;

2) initialize the weight of class1 to 0，and add to the try_list;

3) if try_list is not empty, operate the following process：take away the lightest weight of element u from the try_list:

4) if the Synset(u) includes class2 return to the weight of u, end of algorithm;

5) utilize the reasoning machine to get the concept collection H(u) which has the up-down relationship with u from the ontology;

6) if H(u) includes class2，return to (weight of u+1)，end of algorithm;

7) to every element v in the H(u)∩try_list, if（the weight of u＋1）< the weight of v, amend the weight of v to (the weight of u＋1); to every element w in the H(u)-try_list-abandoned_list, set the weight of w to (the weight of u+1), and add it to the try_list;

8) utilize the reasoning machine to get the range collection B(u) which has the duality relationship with u from the ontology;

9) if B(u) includes class2, return to (the weight of u+1), end of algorithm;

10) to every element v in the B(u)∩try_list, if（the weight of u＋1）< the weight of v, amend the weight of v to (the weight of u＋1); to every element w in the B(u)-try_list-abandoned_list, set the weight of w to (the weight of u+1), and add it to the try_list;

11) add u to the abandoned_list.

Calculate the distance between each corresponding concepts of the output parameters of both requirement service and advertisements service by the above algorithm. And utilizes the definition 4 to calculate the similarity of each concept and eventually utilize the definition 3 to calculate the similarity of the two concept vectors [9], which is the similarity of the output parameter, to receive the matching result.

7. Algorithm performance analysis and evaluate

In order to test improvement effect of algorithm, we select a domain ontology pizza. owl [10] (you can find in http://www.coode.org/ontologies/pizza/2005/05/16/pizza.owl) to evaluate matching degree.

This ontology include 166 concepts in Pizza domain, that are suit for performance test. The algorithms of OWL-S/UDDI and Similarity of Ontology Concepts are accepted to test, the returned valued of services similarity is from 02 to 1. if returned valued is required from 0.2 to 0.4 or above, OWL-S/UDDI matching take practical result; if returned valued is required 0.5 above, substitutable result will be accepted in the OWL-S/UDDI matching, experimental results are showed in figure 2, figure 3.

The experimental results show that the average precision and recall of similarity measures are higher than OWL-S/UDDI algorithm. But when similarity is above 0.5, the matching degree is similar in the separatrix of similarity.

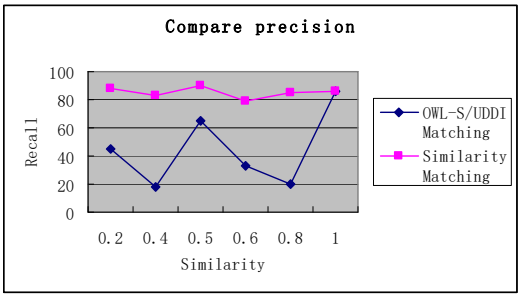

Fig.2. Comparing precision of two algorithm

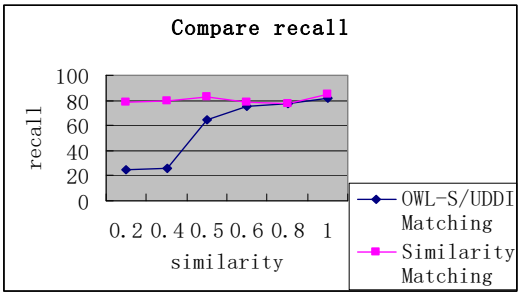

Fig.3. Comparing recall of two algorithm

8. Conclusion

The Web service matching is the final also the key step in the application of Web service. A good algorithm can help to find the required service quickly and accurately and provide the important base for the service composition and also ensure a better service quality. A service matching algorithm base on similarity of concept semantic distance is proposed in this paper, but due to the various binary relation of the input parameters, we will do the further research for its matching algorithm and it is also the problem for the next step.

Acknowledgments

The research was supported by Natural Science Foundation of Anhui Province (1708085MF161), Key Project of Supporting Program for Outstanding Young Talents in Universities of 2016 (Gxyqzd2016214), Key Project of Natural Science Research of Universities in Anhui (KJ2015A236), and was partially supported by grants of National Science Foundation for Young Scientists of China (No. 61702007),

Natural Science Foundation of Anhui province (No. 1508085MC55), Key Project from Education Department of Anhui Province (No. KJ2013A076), Project of Supporting Program for Outstanding Young Talents in Universities of 2016 (Gxyq2017045), Internet and Information Security Anhui Key Laboratory Open Project (ahnis2018002).

References

[1] MARTIN D. OWL2S:semantic markup for Web service [EB/OL]. Http://www.w3.org/Submission/OWL2S. 2005.

[2] Paolucci M., Kawamura T., Payne T.R. Sycara K.P.Semantic Matching of web services Capabilities, In Proc of 2002 Semantic Web Conf.

[3] Li Lei, Horrocks I. A software framework for matchmaking based on semantic web Technology, Proceedings of the Twelfth international World Wide Web Conferenc (WWW 2003). Budapest:ACM Press, 2003: 331-339.

[4] Mokhtar S B, Kaul A, Georgantasn, Et Al. Towards efficient matching of semantic Web service capabilities, Proceedings of the Workshop of Web Services Modeling and Testing (WS2 MATEp06). Palermo: ELSEVIER Press, 2006:137-152.

[5] Dimitrios Skoutas, Alkis Simitsis, Timos K.Sellis. A ranking mechanism for semantic web service discovery. In IEEE SCW, pages 41-48, 2007.

[6] Angela Schwering. Hybrid model for semantic similarity measurement. Lecture notes in computer science, pages 1449-1465, 2005.

[7] Elie Sanchez, editor. Fuzzy Logic and the Semantic Web. Capturing Intelligence. Elsevier, Amsterdam, 2006.

[8] Zhu Xiaojun. Semantic supervised learning literature survey, TR1530. University of Wisconsin Madison: Department of Comuter Sciences, 2006.

[9] Jeffrey Hau, William Lee, John Darlington. A Semantic Similarity Measure for Semantic Web Services. WWW 2005, May 2005:10214.

[10] Verma K, Sivashanmugam K, Sheth S, etal. METEOR-SWSDI:A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services. Journal of Information Technology and Management, 2005, 6(1): 17-39.