The Marking Properties Discovery Model of Web service

Xiaoyin Zhang [a], Xiang Yu [b]

School of Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a]zhangxxyy1112@163.com, [b]xiangyu@cqupt.edu.cn

Keywords: OWL-S, Web service, domain ontology, property marking algorithm.

Abstract. In view of the traditional Web services based on keyword and syntax discovery in the existing problems of low efficiency on coverage and precision, on the basis property marked on the concept of input conditions and service descriptions, this paper designs a set of web services discovery model based on OWL-S and domain ontology, integrates a variety of efficient matching algorithms to determine the list of Web services, which the service requester requirements .This discovery model can truly reflect service developers and service requestors real intension to guarantee the matching efficiency. The experiment on this model shows a huge improvement on coverage and precision efficiency compared with the traditional model.

Introduction

With the rapid increase in the number of Web services in the internet, how to find a suitable service requester requirement of Web services has become a key technology in Web service development.

Recent years, many scholars researched on how to accurately search out the target Web services. Document [1] used the semantic indexing mechanism to optimize service library. Document [2] determined the classification of Web services by comparing service name, input, output and other parameters. Document [3] established a service operation tree to store information about Web services. Document [4] the algorithm ensured that users can search within a certain range of services. Document [5] proposed a discovery method based on semantic Web service by using vector to calculate the semantic distance. These methods, to some extent, improved the efficiency of Web service discovery, but they ignored the semantic interpretation of the concept of service parameters, so they can't determine the true meaning of the concept.

This paper combining with ontology concepts and related technologies and QoS performance indicators, imposes a Web service discovery model based on the making properties, this model is mainly through the OWL-S[6] describing service information that service developer provides and condition information that service requester needs as input, output, precondition and result, and then on the basis of domain ontology library and related knowledge on the conditions of service requests and service developers to provide services input, output, premise and result marked property concepts and finally to get a list of services required by the Web service requester via service matching and QoS matching.

Service matching

String matching. Because ontology's development is a complex process, which requires the participation of experts in the related field [7], so in order to make concept matching more precise, this article adopts the method of document [8].

Concept category matching. Now concept category is divided into two parts: Assume C_i, $C_i^{'}$ are respectively the itch concept instance of the two matching concepts which are waiting to be matched, so the similarity between them can be calculated according to the following two ways:

Non-numeric concept: If C_i, $C_i^{'}$ are non-numeric concepts, firstly map the concept in ontology repository to the classification tree, make the concept as the node, the relationship between the concepts as distance, the distance between the nodes is inversely proportional to the relations between concepts. Based on the above assumptions, we can define the non-numeric concept matching as:

$$\mathrm{Sim}_{CF}(C_i, C_i^{'}) = \begin{cases} 0 & C_i \nRightarrow C_i^{'} \\ 1/1 + 2 \times \mathrm{dis}(C_i, C_i^{'}) & C_i \Rightarrow C_i^{'} \\ 1 & C_i = C_i^{'} \end{cases}, \alpha$$

Among them, $\mathrm{dis}(C_i, C_i^{'})$ represent the shortest distance of C_i， $C_i^{'}$ in classification tree, $C_i \nRightarrow C_i^{'}$ represent C_i has no path to$C_i^{'}$, $C_i \Rightarrow C_i^{'}$ represent C_i has path to$C_i^{'}$, $C_i = C_i^{'}$ represent C_i is equal to$C_i^{'}$.

Numeric concept: If C_i和$C_i^{'}$ are two numeric concepts, the similarity between them are [9]:

$$\mathrm{Sim}_{CF}(C_i, C_i^{'}) = 1 - \frac{|C_i^{'} - C_i|}{C_i^{'}}$$

If C_i和$C_i^{'}$ are two interval concepts, the similarity between them are [10]:

$\mathrm{Sim}_{CF}(C_i, C_i^{'}) = 1 - \sqrt{(a_i - b_i)^2 + (a_i^{'} - b_i^{'})^2}$,among them , $C_i = [a_i, a_i^{'}]$represent interval value after standardization of the itch interval property that service developer provides , $C_i^{'} = [b_i, b_i^{'}]$ represent interval value after standardization of the itch interval property that service requestor provides . The normalization methods refer to interval data normalization method in document [7].

The concept matching based on the making properties

Assuming that Cr is the concept of service requestor parse, Cs is the concept in service description, the similarity between concepts is:

$$\mathrm{Sim}(Cr, Cs) = \alpha S_N(Cr, Cs) + (1 - \alpha) \times (\beta \times \sum_{i=0}^{n} S_{O_i}(Cr, Cs) + (1 - \beta) \sum_{j=0}^{m} S_{D_j}(Cr, Cs))$$

Among them , S_N is the similarity of concept name , S_{O_i} represent the similarity of the itch object property , S_{D_j} represent the similarity of the j-the data property, n is the number of the object property that participate to the matching , m is the number of the data property that participate to the matching , α and β are adjustable weight values. The similarity calculation formula of concept name or concept property is that:

$$S(Cr, Cs) = \gamma \mathrm{Sim}_{SM}(Cr, Cs) + (1 - \gamma)\mathrm{Sim}_{CF}(Cr, Cs)$$

Among them, Sim_{SM} is string concept similarity matching, Sim_{CF} is concept similarity matching based on the classification

Service overall similarity

The service request that the service requester input is defined as: S_r=<I_s,O_s， P_s E_s>, I_s is the input, O_s is the output, PS is the premise, E_s is the result, corresponding to it, the service that the service developer is defined as: S_p=<$I_s^{'}$,$O_s^{'}$ $P_s^{'}$,$E_s^{'}$>, the similarity between them is:

$$\mathrm{Sim}(S_r, S_p) = w_1 \mathrm{Sim}(I_s, I_s^{'}) + w_2 \mathrm{Sim}(O_s, O_s^{'}) + w_3 \mathrm{Sim}(P_s, P_s^{'}) + w_4 \mathrm{Sim}(E_s, E_s^{'})$$

Among them, Sim (I_s, $I_s^{'}$), Sim (O_s, $O_s^{'}$), Sim (P_s, $P_s^{'}$), Sim (E_s, $E_s^{'}$) are respectively represent the input condition similarity, the output condition similarity, the premise similarity and the effect similarity. w_1， w_2， w_3， w_4 Are adjustable weight values. And $w_1 + w_2 + w_3 + w_4 = 1$, 0<w_1， w_2， w_3， $w_4 < 1$.

Ontology and the classification tree

Because the classification of ontology is wide, this paper only involves the classification tree. As figure 1, classification tree is described by the concept, each node in the tree represent a concept and the relationship distance between the concepts represent the path. Domain ontology, According to the concept of domain ontology, we can describe a concept in some field as a

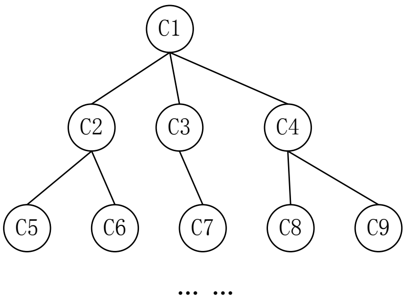


Fig. 1 The concept classification tree

Qos selection

QoS selection is about the matching of QoS, is to match the service quality of our web service which matched by service before. Define the service quality requirement of the service requester as: reaction time (rWS. RT), reliability (rWS. Re), speed (rWS. tS). Define the service quality requirement of the service developer as: reaction time (pWS. RT), reliability (pWS. Re), speed (pWS. tS). QoS matching formula is:

QosMatch <rWS, pWS> =

$$\begin{cases} 1 & (rWS.rT \geq pWS.rT) \cap (rWS.Re \leq pWS.Re) \cap (rWS.tS \geq pWS.tS) \\ 0 & \end{cases}$$

Discovery system model

Combining with domain ontology and the relevant knowledge bases of the WoT system, we implement a model Web service matching system based on making properties, as shown in figure 2.

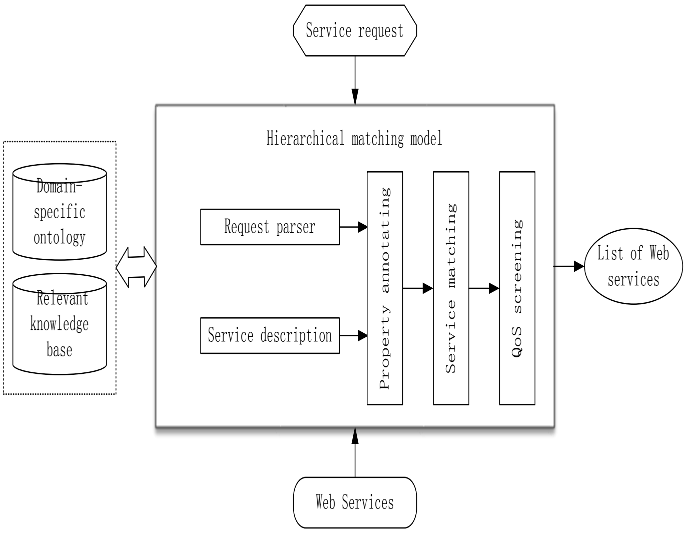


Fig. 2 The discovery model for Web service

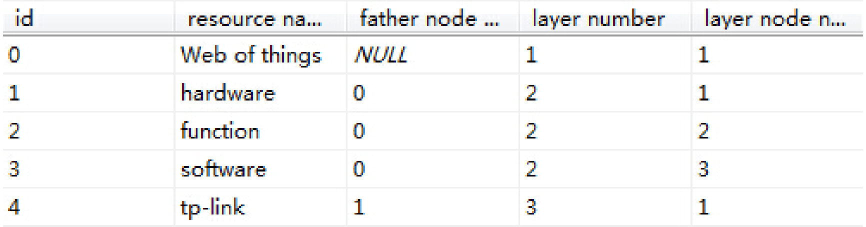
The request condition of the Web services requester input and service provided by the Web service developer, combine with domain ontology, through the OWL - S described as the input, the output, the premise and the result, on the basis of the relevant knowledge bases, to make the making properties of the input, the output, the premise and the result . The content of the label is according to the meaning of the concept in different scenarios, user's hobbies, habits and preferences, experience of service developers. Then through service matching, to match the request conditions that have been labeled the property and service concept, the matched web service need to satisfy the request of the requester. Finally through the QoS to get Web services list which need to meet the requirements of the service requester, then return to the requester.

Simulation experiment

To test the efficiency of the proposed model of Web service discovery based on making properties, service in the library public a lot of service described by OWL - S, the service number is 100-120.In the platform of Visual Studio 2010, through c # and SQL Server 2008 development tools to develop the application, to test the discovery model.

As shown in table 1, firstly map the ontology concept to the classification tree, then entry the concept classification tree in the database.

Table. 1 Classification tree in the database



id	resource na...	father node ...	layer number	layer node n...
0	Web of things	*NULL*	1	1
1	hardware	0	2	1
2	function	0	2	2
3	software	0	2	3
4	tp-link	1	3	1

This experiment define the weight values in the algorithm as follows $\beta = 0.5$, w_1, w_2, w_3, $w_4 = 0.25$Figure 3 says, along with the change ofα, model efficiency index's change trend. We can find when α is about 0.7, system can achieve optimal efficiency.

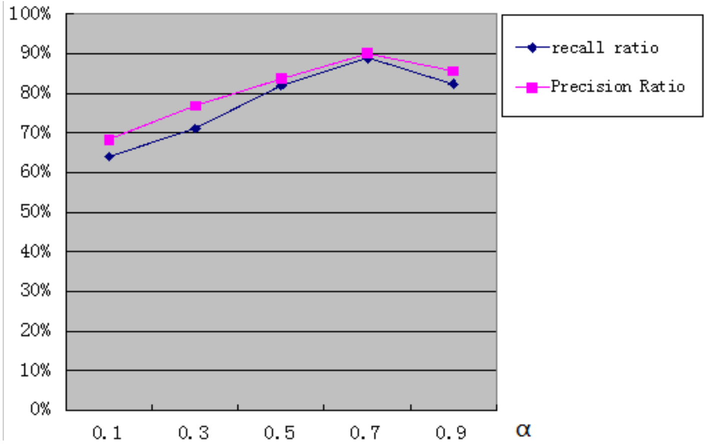


Fig. 3 The change of the model efficiency index with α

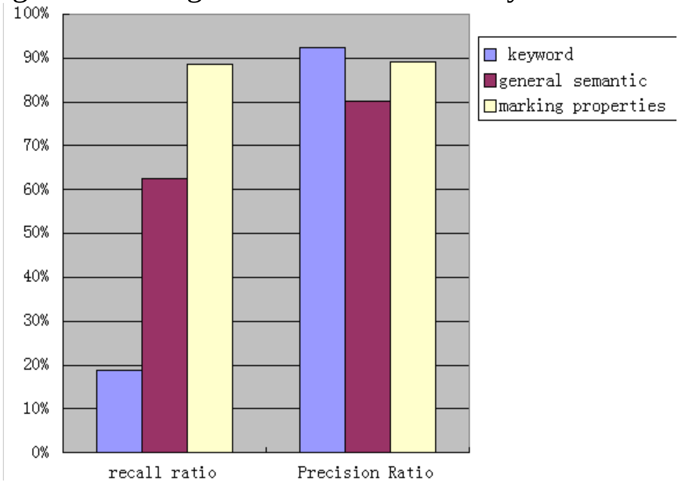


Fig. 4 Comparison of efficiency index

Figure 4 says that when α= 0.7, the efficiency index compare between this system and the traditional model. From it we can see that the present system compares the model that based on keywords and syntax has greatly improved in coverage efficiency and precision efficiency, this model also has increased significantly than the model based on semantic, which close to the model based on keywords.

Summary

This paper constructs the model of Web service discovery based on making properties, based on the algorithm which based on service matching and algorithm which based on QoS selection , the efficiency of Web service discovery has greatly improved compared to the above models. However, this label of the concept making properties by artificial way, it can’t automatically. So in future further study is needed to enable automation of making properties .At the same time, considering the weight values’ direct impact on the performance of the system, further study is also needed to determine the values in specific cases.

References

[1] Sun Ping, Jiang Changjun. Using Service Clustering to Facilitate Process-Oriented Semantic Web Service Discovery [J]. CHINESE JOURNA L OF COMPU TERS, 2008, 31(8): p.1340-1352.

[2] Wu Hang, Guo Chaozhou. The research and implementation of Web Service classification and discovery based on semantic[C]// Computer Supported Cooperative Work in Design, 2011: p.381-385.

[3] Demian A D, ANANTHANARAYANA V S. A Tree Structure for Efficient Web Service Discovery[C]// Emerging Trends in Engineering and Technology, 2009: p.826 - 831.

[4] Lu Gehao, WANG Tengfei, ZHANG Guojin, et al. Semantic Web Services Discovery Based on Domain Ontology[C]// World Automation Congress, 2012: p.1-4.

[5] Zhou Lixin. An Approach of Semantic Web Service Discovery[C]// Communications and Mobile Computing, 2010: p.537-540.

[6]Yang W S, Hwang S Y. A process-mining framework for the detection of healthcare fraud and abuse [J].Expert Systems with Applications, 2006, p.31-68

[7] He Chaobo, Chen Qimai. Study on approach for web service discovery based on ontology [J]. Computer Engineering and Design, 2010: p.1421-1423.

[8] ALEXANDER M,STEFFEN S. Measuring Similarity Between Ontologies[C] //Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, 2002: p.251-263.

[9] Hou Lijuan, Li Shuyu. A CONTEXT-BASED SEMANTICWEB SERVICE D ISCOVERY METHOD [J]. Computer Applications and Software, 2011, 28(1):p. 154-156.

[10]Li ChunMei, JIANG YunCheng. Study on Semantic Web Services Discovery with QoS Constraint [J]. Computer Science, 2007, 34(6):p.116-121.