

Web Services Discovery Based on the Fixed-point Semantics

Quan Lixin

department of information management
hunan university of finance and economics,
changsha 410205, china

Abstract-Based on the relevant theories of fixed-point, our approach uses data mining and statistical studies the semantics of behavioral constraint and probability statistics of Web Services. Aiming at the insufficient of traditional qualitative description of the invoking, this paper presents an approach for quantitative description of the invoking, we defines the fixed-point function of Web services allocation and its calculating way. At last, the experiment proves the feasibility and effectiveness of the way we adopted.

Keywords-Semantic Web, Least fixed point, Web Services discovery, Strongly-call rule

I. INTRODUCTION

Supposing a tourist plans a DIY tour which includes a flight from Beijing to zhangJiaJie, he needs to find a service of buying a ticket at first. The service should meet the following demands: R1, tourist could buy a flight ticket of Beijing to Zhang-jia-jie from it. R2, in the service of buying flight ticket, tourist could make choice among several airline companies. R3, The WSDL provided by the supplier should include the description about how to buy tickets. For example, when tourist inputs "destination (zhangjiajie), date (2011-5-1)", he will get the output information "accept ticket".R4, to meet the tourist's demands, the service of buying a ticket might provides some accessional services like: giving the ranking list of the airline companies by the satisfaction, providing the most popular hotels and travel agencies at destination for booking or appointment, and so on.

In all the services above, for R1, tourist can only enter some keywords like 'ticket from Beijing to Zhangjiajie' in the search engine, then to find out the desirable services among the thousands searched results. For R2, tourist can realize in the frame of UDDI. for R3, he can solve this problem by using the existing technique of services discovery which are based on the service function interface described by WSDL. However, for R4, which are exactly service information that most tourists could get them only by much strenuous searches. This is a process to find and match time and again. All these demands are difficult to realize directly for the existing technique of Web services discovery. How to carry out these services distributed on Web by the information of semantic is one of the major problems that the semantic Web must solve[1]. The Web service discovery based on semantic is the basis to solve the problem above. Nowadays it is being one of the hot topics in the community of related scholars [2,3,4].

Most the existing methods of semantic Web service discovery adopt a matching level to filter services. They are very difficult to find out a content result and don't think about the quality of the service yet. Papers [5,6] give some different methods of describing the semantic information of Web service. These representative method include based on share context, based on rule and by increasing semantic note to extend the WSDL description information etc. it has some certain subjectivity and arbitrariness for they usually describe the qualitative relation between services based on the specialist system and the predefined rules. Paper[7] use Bayesian networks as the tool of semantic description among services, based on historical information of service call to build a Bayesian networks for Web services, by extending the probability of OWL can build the Bayesian network from the main body. Such type of method is based on Markov assumptions, building Bayesian networks of describing the Web service semantic by detecting the independence of condition among services to do probabilistic reasoning. In this paper I will present a service discovery method which is based on behavioral constraints and semantic of probability statistics.

II. FIXED POINT SEMANTICS OF WEB SERVICES

In the Web application environment, Web services are always in a state of calling or being called, you can see the calling relation among services as a tree or network structure. According to the feature of super-function, the calling relation of Web services is a recursive relation which is the premise of this paper to using the fixed point theory [8] of mathematics. This section I will define the quantitative expression of the calling relation of Web services and the fixed point function, by solving the fixed point function, I can get the definite and integrity data about the calling relation among services.

A. Expressions of Web service and their calling relation

Suppose a method described in WSDL as the "atomic Web services", namely no longer split Web service. In the rest of this paper I will use "service" on behalf of "atomic Web services.

Definition 1: Let $S = (URI, In, Out, SL)$ describes the service. ζ is a particular service of S that corresponds to a specific function or operation, where, URI is the unique resource identifier, In and Out are the input and output data types respectively, SL is the SOAP address.

In the service call, related service is worked with some particular structure. Those include three basic types, say,

sequential, parallel and branch, shown in Figure 1 [8] where a, b, c, and d is the unique identifier of the service.

Definition 2: in the course of a service combination, the service P_s calls service C_s at time t_b , which call ends at time t_e , the calling relation between P_s and C_s is denoted as $P = (id, P_s, C_s, t_b, t_e)$. Let T represent the time domain of the timestamp, where: id indicates a type of the service combination; P_s is the calling service to other services; C_s represents the service being called; t_b is the start time that P_s is calling to C_s ; t_e is the terminal time of this calling, and $t_b, t_e \in T$.

B. The least fixed-point function of Web services

Typically, the agent of service provider can record the direct calls of services in a database or data warehouse, but can not record the indirect calls among services. As a result, there are always some very useful indirect calls for users being not recorded. The recorded direct calls can not completely show the relationship among all services. Obviously, the synthesis of all calling data about certain particular service is a limited data set. The fixed points[8] of limited data set can calculate from the monotony and iterative of the immovable structures.

Using the basic idea of fixed point, now I give the least fixed point function of Web service.

Definition 3: Let $\zeta = (id, P_s, C_s, t_b, t_e)$ on behalf of the association between two services. The instances in ζ are composed of direct associations and indirect ones derived from direct of instances in P . To obtain an indirect association, A function from (ζ, P) to ζ is defined as follows:

$$P = f(P, P) = P_d \cup \pi_{1,2,8,4,10} (P_{1=1 \wedge 3=2 \wedge 5 \leq 4} \times P) \quad (1)$$

Where, π and \times , means "projection" and "connection" operation respectively. That is similar with definitions in the relational algebra. Initially, p is empty. Clearly, solving (1) is an iterative process, giving P_d as the initial input of f , we can get the indirect call between services. Equation (1) can be simplified as:

$$p = f(p) \quad (2)$$

Know from the fixed point theory, the function f defined by author is a fixed point function. f has the iterative feature that the latter calculations are based on the previous, the number of instance in the next results less than in the previous, namely f is monotonous. Next, I will prove that the function defined above is the least fixed point function, i.e. by finite iteration steps of f , the final result P is uniquely determined.

Testify: Suppose that all instances of P are denoted as $D(P)$, $D(P)$ satisfies the formula (2), and f is monotone. Know from the definition of fixed point, $D(P)$ is the fixed point of f . Now let's prove that $D(P)$ is the least fixed point of f .

Initially, $P = \Phi$, let f_i denotes the result after i iteration, from the knowledge about the definition of the composite function, $f_{i+1}(\Phi) = f(f_i(\Phi))$, with $f_0(\Phi) \subseteq f_1(\Phi) = P_d$. And according to the monotony of f , it is easily get that, if $f_i(\Phi) \subseteq f_{i+1}(\Phi)$, then $D(P) \subseteq f(D(P))$. If $f_i(\Phi) = f_{i+1}(\Phi)$,

then $D(P) = f_i(\Phi)$, that is, from Φ can be obtained $D(P)$ after i step of iteration.

Therefore, $D(P)$ is the fixed point of f and a subset of all fixed point of f . i.e. $D(P)$ is the least fixed point of f .

C. The calculation of the fixed point function

Knowing from the fixed point function defined by the previous section, for a given P_d , f after a finite step iterative calculation can get an uniquely identified set P which is described the direct and indirect calls between services. From the calculation of P , we know, there must have: $f_i(P_d) \subseteq f_{i+1}(P_d)$, that means, there exist an incremental Δ makes $f_{i+1}(P_d) = f_i(P_d) \cup \Delta$, i.e. the $i+1$ th calculated results contains the i th results, so you can just need to calculate the increment Δ . See Algorithm 1.

Algorithm 1: Algorithm for the fixed point function

input: P_d ; output: P

steps:

$$\Delta = \pi_{1,2,8,4,10} (P_{d_{1=1 \wedge 3=2 \wedge 5 \leq 4}} \times P_d)$$

$P \leftarrow P_d, \Delta \leftarrow \Phi // \Delta$ incremental
while $\Delta \neq \Phi$ do

$$\Delta \leftarrow \pi_{1,2,8,4,10} (P_{d_{1=1 \wedge 3=2 \wedge 5 \leq 4}} \times \Delta)$$

//incremental part

$p \leftarrow p \cup \Delta$
end while
return P

III. WEB SERVICE DISCOVERY BASED ON SEMANTIC

By the method of algorithm 1, we can obtain a complete data of service call. These data are the basis of excavating calling relation among services. In this section, the author uses the related thought of data excavating [9], to dig out a determined and effective calling model which is called the service call rules. Call rules could provide basis for making decision to find related services in function and content-related.

A. Description of call rules

For further instructions of call rules, assuming that there are 5000 call records in the database or data warehouse, including at least the call information about service a, b, e, f and so on. I fixed a threshold of 5% in advance. When the probability of a call is greater than this threshold it is thought as an often call, called a frequent call. Assuming that the total of call initiated by a is 1000, there are 260 call of $a \rightarrow b$, the probability of its occurrence was 0.52, it is a frequent call. But the calls $a \rightarrow e, a \rightarrow f$ were 410 and 330, then for the calls of a, such a call $a \rightarrow b$ is not so "frequent". So, when we want to find out the call rules, we need not only to consider if a call is a frequent call but also it is relative frequent to its initiate call.

Here I will give the formal description of the mining problem of call rules. Suppose $U = \{\zeta_1, \zeta_2, \dots, \zeta_m\}$ is a finite set with m basic services, namely, $\zeta_i \in U$. Where ζ_i has a

unique identity URI. And $H = \{h_1, h_2, \dots, h_n\}$ is a database or data warehouse recorded the history of calls, where each record h_i represents the i -th call in H , s a call. Each call is associated with a unique identifier HID .

If $\zeta_i \in h_i.P_s \vee h_i.C_s$, say h_i contains ζ_i . The formal description of the call rules is expressed as the implication of $\zeta_i \Rightarrow \zeta_j$, which $\zeta_i \in U$, $\zeta_j \in U$, and $\zeta_i \neq \zeta_j$. When we want to describe the probability of some call in historical data, a quantity frequency degree is used.

Definition 4: frequency degree of call $\zeta_i \rightarrow \zeta_j$ in H Fre:

$$Fre = Call(\zeta_i \rightarrow \zeta_j) \quad (3)$$

Where, H is the historical data of call, $Call(\zeta_i \rightarrow \zeta_j)$ is the call number in line with $\zeta_i \rightarrow \zeta_j$ in H , n is the total number of records in H .

Definition 5: the correlation of call $\zeta_i \rightarrow \zeta_j$ in H Rel:

$$Rel = Fre(\zeta_i \rightarrow \zeta_j) / Fre(\zeta_i) \quad (4)$$

Clearly, by definition 4,5, we can solve the problem should be considered in discovering call rules. The threshold of frequency degree and correlation degree is called the minimum frequency degree $minfre$ and the minimum correlation degree $minrel$ respectively. Usually, these two values determined by the customer.

Definition 6: These call rules fulfill $Fre \geq minfre$ and $Rel \geq minrel$ are called strongly-call rules. They help customers find those call service closely contact with the customer needs and frequent used.

B. Call rule discovery

Know from the analysis above, an effective call rules can only be generated in those calls which frequency degree is not less than the minimum frequency degree and the correlation degree is not less than the minimum correlation degree. An effective way to the call rules is embodied by the algorithm 2.

Algorithm 2: stress call rule generate algorithm

input: historical data of call H , $minifre$, $minrel$

output: stress call rule $p \Rightarrow c$

$M_1 \leftarrow$ select $p,c,count(p,c)$ as ct from h group by $p,c//$
 count each type call and placed the number in a temporary table M

```

for each row in  $M$  do
//determined whether is a strongly-call rule
if  $(ct / sum(ct) \geq minifre/minrel)$ 
{output the valid call rule  $p \Rightarrow c$  }
end for
    
```

By statistics, excavating the stress call rules, come to the purpose of discovery of Web services.

IV. EXPERIMENTAL RESULTS

In the operating system with Inter Pentium M CPU1.7G HZ, 768MB RAM, Microsoft Windows XP Professional Service Pack2, I use Java, JSDK1.4.2_11 and Access2000 programming. Suppose there are $(3 \times \text{the number of direct calls} / \text{service number} + 1)$ customers, one of them generates call of all the services, the others generates direct calls randomly. For the case, number of services 50, 100, 150, 200 corresponding to the number of direct calls were 10^3 ,

2×10^3 , 3×10^3 , 4×10^3 , 5×10^3 , 6×10^3 , 7×10^3 , 8×10^3 , 9×10^3 , 10^4 . Giving the minimum frequency degree $minfre = 0.2$, minimum correlation $minrel = 0.8$. The tested performance is given in Figure 1.

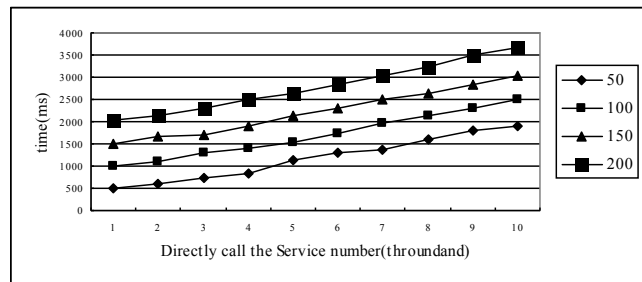


Figure 1. Web Services Discovery computing performance

As shown in figure.1, for the case given minimum frequency degree, minimum correlation, the number of services and the number of composition process of a given service determine the performance of Web service discovery. With the increasing number of composition process of services, the performance of Web service discovery based on service behavior and probability statistic semantic decreases. In a certain data range the change is linear.

V. CONCLUSIONS

Experiments show that the service discovery method given by this paper which is based on the behavior constraints and the probability and statistics semantic is feasible and effective in certain range. Its discovery function and the content-related services may provide some basis to make decision.

REFERENCES

- [1] Roy Ladner, Elizabeth Warner, Fred Petry, Uday Katikaneni, Kevin haw, Kalyan Gupta, Philip Moore. Web Services: Evolving Techniques in Net-Centric Operations. IEEE. 2004
- [2] WANG Xiao-xuan, BAO Ai-hua. Research on the Semantic Web Oriented Method for the Evolution of Composite Service. Computer Science[J], 2011, 2: pp138-142.
- [3] S. Agarwal, S. Handschuh, S. Staab. Annotation, Composition and Invocation of Semantic Web Service. Journal on WebSemantics, 2005, 2 (1): pp1-24.
- [4] D. Berardi, D. Calvanese, G. D. Giacomo, and et al. Automatic composition of e-services that export their behavior. In: Proc. of 1st Int'l. Conf. on Service Oriented Computing, Trento, Italy, 2008
- [5] ZHONG Mei, SONG Shun-lin. Multi-layer matching method on semantic Web services. Journal of Computer Applications[J]. 2007, 1. PP234-237.
- [6] ZHANG Zheng, ZUO Chun. Web service discovery method based on semantic expansion. Journal on Communications[J]. 2007, 01.
- [7] HU Bing, LIU Huai-song. Research on Automatization of Web Service Based on Ontology. Microelectronics & Computer[J]. 2006, 11.
- [8] M. D. Rougemont. Fixed-point semantics and the representation of algorithms on large data. VLDB, 1988: 264-272.
- [9] SHE Yu_mei, BAI Hong. A Method for Association Rule Mining Based on Fixed Point. Journal of The Central University For Nationalities (Natural Science Edition)[J]. 2004, 01.