An Efficient Two-Phase Service Discovery Mechanism

Shuiguang Deng, Zhaohui Wu, Jian Wu and Ying Li
College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China
{dengsg, wzh, wujian2000, cnliying}@zju.edu.cn

ABSTRACT

This paper brings forward a two-phase semantic service discovery mechanism which supports both the operation matchmaking and operation-composition matchmaking. A serial of experiments on a service management framework show that the mechanism gains better performance on both discovery recall rate and precision than a traditional matchmaker.

Categories and Subject Descriptors

D.2.12 [**Software Engineering**]: Interoperability – *data mapping, distributed objects, interface definition languages*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *data sharing, Web-based services*

General Terms: Algorithms, Design, Experimentation, Measurement, Performance

Keywords: Web Service, Service Discovery, Service Matchmaking, Composition, Bipartite Graph Matching

1. INTRODUCTION

Service discovery has been a well-recognized challenge in the application of SOA (Service-oriented Architecture). At present, there is a good body of work on service discovery [1-3]. Among the work, the effort of semantic web service from the semantic web community has been regarded as the most promising way to retrieve services in an accurate and automatic way. However, they can achieve even better performance with the following two factors taken into consideration: (1) Not all inputs are compulsory for each output for an operation of a service; (2) The operations within a service are not isolated from each other; but some of them may be concatenated to provide value-added functions. In this paper, we propose a two-phase semantic service discovery mechanism that considers the above two factors. Given an advertised service and a request, it checks whether there is a single operation matching the request at the first phase, where the semantic matchmaking is carried out between each operation and the request, and the interface dependency of an operation is also considered. If no single operation matches the request, it performs operation-composition matchmaking to check whether there is such a composition of operations matching the request. A serial of evaluations suggest that considering these two factors in TSSD offers better performance

2. TWO-PHASE SERVICE DISCOVERY

2.1 First Phase: Operation Matchmaking

We import the bipartite graph matching to improve the efficiency of matchmaking between an operation and a request. As a matchmaking example shown in Fig. 1, the dashed lines within the operation denote the dependency between outputs and inputs; and the real lines annotated with decimal fraction denote the similarity between the two connected objects.

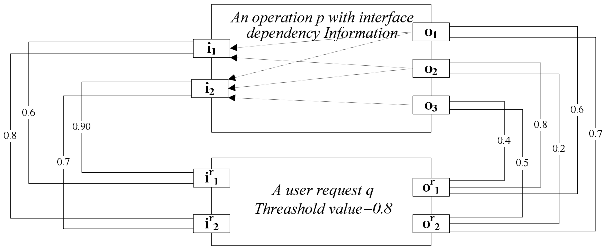

Figure 1. Operation Matchmaking Example

We carry out the matchmaking between their inputs and outputs, respectively, to compute the similarity degree between *p* and *q*. For the matchmaking between inputs or outputs, we model it as a weighted bipartite graph; and then we find an optimal matching. Based on the matchmaking, the similarity degree between an operation and a request is computed by the following formula.

$$SimPR(p,r)=\frac{\sum_{o^r\in O^r}\left\{SimCC(o^r,f(o^r))\times\sum_{i\in f_p(f(o^r))}SimCC(i,g(i))\Big/\left|f_p(f(o^r))\right|\right\}}{\left|O^r\right|}$$ where

SimCC is the semantic similarity function between two concepts [4]; *f* and *g* is the injection from the optimal matchmaking in the bipartite graph of inputs and outputs, respectively. According the formula, we get the similarity degree between *p* and *q*. If the value is larger than the threshold value specified in the request, it denotes that the operation satisfies the request. However, if none of the operations alone within a service can satisfy the request, it doesn't mean that the service should be ignored. In fact, a composition of operations within a service can bring value-added functions. Thus, in this case, the discovery mechanisms go on to its second phase.

2.2 Second Phase: Composition Matchmaking

Operations can be connected by feeding one's output to the other's input in an orchestration way similar to an assembly line in a factory. However, to compose operations together on-line for each new incoming request is time-consuming, especially when the number of operations within a service is large. In order to avoid the time consumption in the service discovery, we can transfer the composition process from the service discovery to the service registration. That means when a service is registered, we can find out all the possible operation-sequences in it using a background program running on the service registry. After all the operation-sequences are constructed, they can be used for the service discovery based on composition matchmaking. As the concatenation of operations will bring inaccuracy to functions, we import the concept *error-distance* $\zeta(os)$ to evaluate the inaccuracy.

Copyright is held by the author/owner(s).
WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

$$\zeta(os) = \lambda(1 - \frac{1}{\ell(os)}) + (1-\lambda)\hbar(os) \quad (0 \le \lambda \le 1)$$

where $\ell(os)$ is the *physical error-distance* evaluated by the number of operations in the sequence; and $\hbar(os)$ is the *semantic error-distance* evaluated by the average semantic degree between concepts.

From the perspective of a user, an operation-sequence can be regarded as a pipe-line that accepts inputs from the beginning operation, generates outputs at the end and hides its inner details. Thus, an operation-sequence can be transformed into an operation extended with a sequence-distance attribute. In this operation, the input set is the input set of its first operation and the output set is the output set of its last operation. Thus, for an operation-sequence and a request, we can follow the method used in operation matchmaking to compute the similarity between them except that we consider *error-distance*. Thus, we get the following formula to compute the similarity.

$$SimSR(os,r) \frac{\sum_{o^r \in O^r} \left\{ SimCC(o^r, f(o^r)) \times \sum_{i \in f_p(f(o^r))} SimCC(i, g(i)) \Big/ \left| f_p(f(o^r)) \right| \right\}}{\left| O^r \right|} \times (1 - \zeta(os))$$

3. EXPERIMENTS AND EVALUATIONS

We evaluate the performance of the service discovery mechanism by using three well-recognized metrics, namely service recall rate, precision rate, and scalability, in our service composition platform DartFlow [5] where we have implemented the propose discovery mechanism through the combination of two algorithms named Operation Matchmaker (OM) and Composition Matchmaker (CM). Moreover, we also provide a simple version of Operation Matchmaker named Simple Operation Matchmaker (SOM) which doesn't consider the interface dependency. In order to prepare the test set for the discovery experiments, we developed a tool based on the IBM XML Generator that enables one to generate random XML files based on schemas. With this tool, we generate 5 groups with 100 services for each as Table 2 shows.

Table 1. Test Set Preparation

Group No.	Service Number	Proportion of Partially-Dependent Outputs (PPDO)
G-1	100	0%
G-2	100	20%
G-3	100	60%
G-4	100	80%
G-5	100	100%

Fig. 2 shows the influence of the Partially-Dependent Outputs on the Recall Rate and Precision. It indicates that taking interface dependency information into consideration can bring a better recall rate and precision; especially that it can improve the recall rate to a great extent.

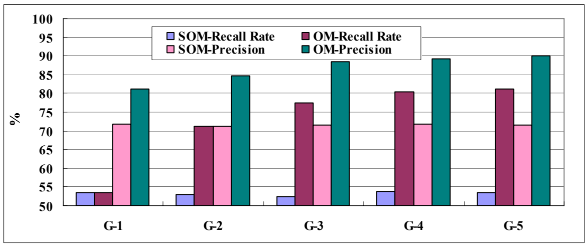

Figure 2. Influence of Partially-Dependent Outputs

Fig. 3 shows the influence of the Operation Composition on the Recall Rate and Precision. It indicates that, for each group, both the recall rate and the precision from the combination of OM and CM are better than that from anyone from SOM, OM or CM.

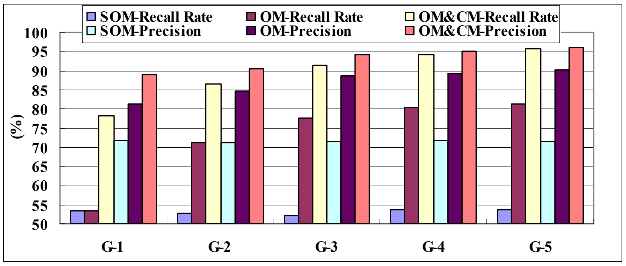

Figure 3. Influence of Operation Composition

The above experiments and evaluations illustrate that considering both interface dependency information and operation composition can improve the efficiency for service discovery.

4. CONCLUSION

This paper proposes a two-phase semantic-based service discovery mechanism to discover services in an accurate, efficient and automatic way. Compared to other approaches, the new method has two salient characteristics: (a) it takes into account the interface dependencies implied within an operation while performing matchmaking; b) it supports two-level matchmaking, namely operation matchmaking and operation-composition matchmaking. A serial of experiments demonstrate that the proposed mechanism has both a good recall rate and precision.

5. ACKNOWLEDGMENTS

This paper is supported by the National Key Technology R&D Program under Grant No.2006BAH02A01; the National Natural Science Foundation of China under Grant Nos.60603025, 60503018; the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z171; China Postdoctoral Science Foundation No. 20070421183.

6. REFERENCES

[1] Paolucci M, Kawamura T, Payne T.R, Sycara K. Semantic Matching of Web Services Capabilities. International Semantic Web Conference (ISWC'02), p36-47, 2002.

[2] Klein M., Bernstein A. Towards High-Precision Service Retrieval. IEEE Internet Computing, 8(1), p30-36, 2004.

[3] Ulrich K, Birgitta K, Mirco S., Michael K. DIANE: an Integrated Approach to Automated Service Discovery, Matchmaking and Composition, International Conference on World Wide Web (WWW'07), p1033-1042, 2007.

[4] Li, Y., Bandar Z.A., Mclean D. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. IEEE Transactions on Knowledge and Data Engineering, 15(4), p871-882, 2003.

[5] Deng S., Li Y., Xia H., Wu J., Wu Z. Exploring the Flexible Workflow Technology to Automate Service Composition (ASWC'06), p444-458, 2006.