http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (15): 3199-3203, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.3199.3203

A Novel Service Preprocessing Method Based on RGPS Ontology

[1,2]Rong LI, [1]Keqing HE, [1]Jianxiao LIU
[1]State Key Lab of Software Engineering, Wuhan University, 430072, Wuhan, China

Abstract: With the rapid growth of Web services, services must be stored in repositories. To enhance the discovery efficiency of Web services, it is necessary to preprocess the services in the repository to achieve an effective organization structure. A RGPS-based service preprocessing method is proposed in this study. Services are firstly clustered based on the domain ontology and the RGPS model. The particle swarm optimization algorithm is then used in the service composition according to the frequent service request. This method leads to a rapid matching between services and the RGPS requirement model and acceleration in the service discovery.

Key words: Web service, preprocessing, service clustering, service composition, RGPS

INTRODUCTION

Web services (Yue *et al*., 2004) are a new type of software paradigm providing an interoperable software entity which is independent of programming languages and platforms. Service discovery and recommendation usually requires a predefined service repository (Huhns and Singh, 2005), in which service providers are enable to register their services in a unified way to facilitate the user query.

To manage the service repositories efficiently, the services stored in the repository need to be clustered in certain rules to prune the searching range during the service selection process. The laboratory in which the authors work has proposed an ontology-based meta-modeling method named O-RGPS (Ontology-Role/Goal/Process/Service)to describe requirements and services (Wang *et al*., 2008). The RGPS framework intends to help users to describe and organize their demands and provide a unified description language-independent method for the service registry and composition.

O-RGPS FRAMEWORK REALIZATION

Figure 1 illustrates the on-demand service selection framework under the domain-model of the O-RGPS. In this framework, service requesters are expected to input the service requirements and semi automatically obtain the services in the repository registered by service providers.

This is achieved by means of the domain ontology and domain model of the RGPS established by domain engineers. The framework can be divided into three parts: elicitation and analysis of requirements, registration and organization of services and selection of services.

In the process of requirement elicitation and analysis, service requesters input the requirements by wiki-based requirement acquisition tools. The tool SKLSEWiki has been developed and validated (Ning, 2012). Based on the most users' selection and domain ontology established by the domain engineers, this tool is able to recommend requirements probably neglected by service requesters.

For the section of service registry and organization, service providers register their services in the MFI-based registration tool according to the international standard ISO 19763. The tool S2r2 is on hand (Zeng *et al*., 2011). Referring to the domain ontology, the services after registration are clustered in various fields according to the RGPS domain model. This work is done in the tool of service preprocessing based on the domain model and the clustered services become components stored in the service repository organized by RGPS. In this way, the service selection and recommendation in the service repository can be processed more efficiently according to the users' RGPS requirement model.

According to service requesters' RGPS requirement model, the service selection tool can select appropriate services in the RGPS service repository. If no service meets the requirements, services available for combination are chosen by the service selection tool and composed by the service composition tool which in turn gives feedback to the service requester. The corresponding theory and tools can refer to reference (Liu *et al*., 2012).

METHOD of SERVICE PREPROCESSING

It is necessary to store the large amount of services obtained in a service repository. In order to accelerate the

Corresponding Author: Rong LI, State Key Lab of Software Engineering, Wuhan University, 430072, Wuhan, China

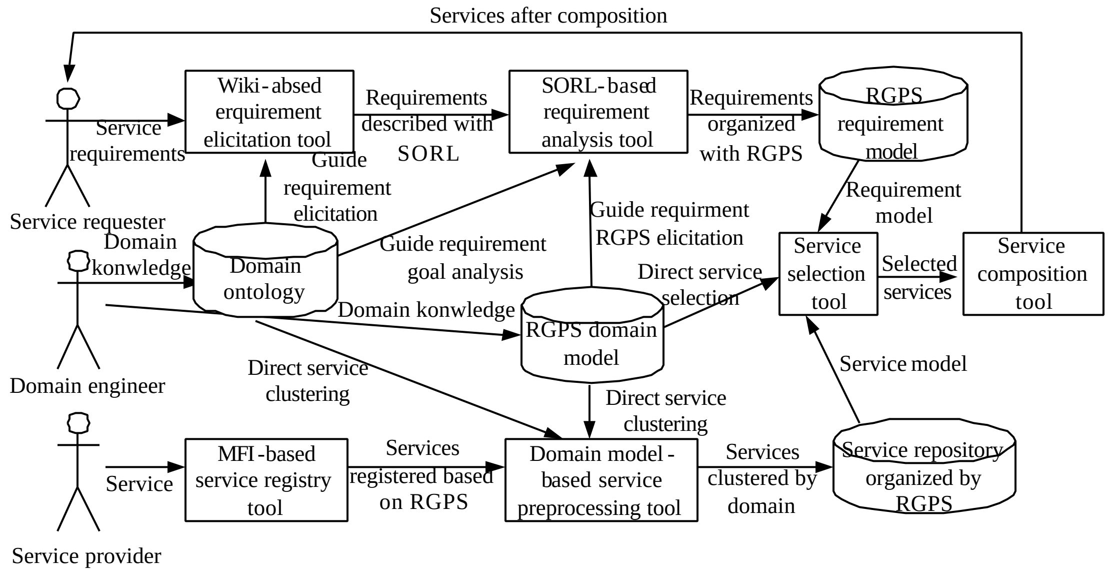

Fig. 1: An on-demand service selection framework based on the RGPS

service discovery, the services in the repository can be preprocessed to be organized in a certain pattern.

RGPS-based service clustering: The first step for preprocessing services is to cluster similar services to reduce the service searching range and enhance the efficiency. The service clustering method in this study is based on the domain ontology and the RGPS domain model. The clustering is applicable for both atomic and composite services and the information of SName, R, G, P, S of services are of great importance.

Firstly, the matching similarity between the service name SName and the domain concept c is calculated. In the domain ontology, the concepts c_i ... c_n relevant to c can be found by distinguishing relation between concepts (e.g., EquivalentClass, SubClassof, Partof, Instanceof, etc). The similarity between SName and c_i ... c_n can then be calculated and multiplied by the corresponding weights w_i ... w_n, respectively. Among the results the maximum value is defined as the similarity between SName and the domain c. If the similarity exceeds a predefined threshold, the service is confirmed within this domain.

Definition 1: (Concept similarity) The concept similarity represents the similarity between the input concept and the existing concepts in the domain which is given by Eq. 1 as:

$$\text{sim (sn, c)} = \max(\text{simt (sn, c)}, w_i{*}\text{simt (sn, } c_i), w_j{*}\text{simt (sn, } c_j), \ldots, w_n{*}\text{simt (sn, } c_n)) \quad (1)$$

where, c, c_i … c_n?C, w_i, …, w_n is the user-defined weights.

Definition 2: (Domain ontology satisfaction) Given a certain threhold θ, when sim (sn, c)>θ, the concept sn falls into the domain ontology do, i.e:

$$\text{sim(sn, c)}>\theta>\text{SatDO(sn, do)}$$

A tool WordNet is used to calculate the function sim. The semantic similarity of words m_1 and m_2 can be expressed as:

$$\text{simt (m1, m2)} = \frac{1}{2}\left(\frac{\text{dis}(c_1, c_f)}{\text{dis}(c_1, c_r)} + \frac{\text{dis}(c_2, c_f)}{\text{dis}(c_2, c_r)}\right) \quad (2)$$

where, c_1 and c_2 are the exact meanings of m_1 and m_2 in the domain, respectively; c_f is the parent concept node of m_1 and m_2; c_r is the root node of the classification tree where the two words locate; dis (c_1, c_f) represents the path length between c_1 and c_f in the semantic tree of WordNet.

Definition 3: (RGPS similarity) The RGPS similarity, defined as the similarity between the RGPS attributes and the RGPS domain model, is calculated as:

$$\text{simm (s, m)} = \frac{1}{4}(\text{wrsiml(sr,mr)+wgsiml(sg,mg)+wpsiml(sp,mp)+ wssiml(ss,ms)}) \quad (3)$$

where, s represents either an atomic or a composite service; m is the RGPS domain model; w_r, w_g, w_p, w_s are user-defined RGPS ontology weights, respectively; s_r, s_g, s_p, s_s denote the RGPS attributes of a service, respectively; m_r, m_g, m_p, m_s denote the RGPS ontologies, respectively.

Definition 4: (RGPS satisfaction) Given a certain threhold|H, when simm(s, m)> |H, the service s falls into the domain model m, i.e:

$$\text{simm}(s, m) > \eta \rightarrow \text{SatRGPS}(s, m)$$

As the RGPS attributes of services are always described in the form of several words or a phrase, the simm can be calculated by the VSM (vector space model) (Salton *et al.*, 1975) method which is a widely used document similarity measureing method in information retrieval. The important concepts in the domain ontology are extracted to form a vector $T = <t_1, t_2, \ldots, t_n>$. Vectors $S_1 = < s_{11}, s_{12}, \ldots, s_{1n} >$ and $M_1 = <m_{11}, m_{12}, \ldots, m_{1n}>$ are defined where s_{1i} and m_{1i} are the weights of t_i in S_1 and M_1, respectively. The weights can be defined by users or calculated via the occurance frequency of t_i in the domain documents. The weight of t_i in S_j can be calculated by Eq. 4 as:

$$s_{ji} = \frac{tw \times \log \frac{N}{df}}{\sqrt{s_{j1}^2 + s_{j2}^2 + \ldots + s_{jm}^2}} \quad (4)$$

where, tw is the user-defined weight of t_i in a range of 0~1; N is the referenced number of documents in the domain; df is the number of documents including t_i:

$$\sqrt{s_{j1}^2 + s_{j2}^2 + \ldots + s_{jm}^2}$$

is the cosine normalization factor and is equal to the Euclidean length of vetor s_j.

The similarity between S_1 and M_1 is given by:

$$\text{Siml}(S_1, M_1) = \sum_{t_i} s_{1i} \times m_{1i} \quad (5)$$

Algorithm 1: Domain-based service clustering algorithm

DRGPSSC(SCollection, C, DO, RGPSM)
Input: SCollection is the set of services { $s_1, s_2, \ldots, s_n$ }; C is the concepts in the domain; DO is the domain ontology; RGPSM is the domain model of RGPS
Output: SCCollection is the set of services after clustering $\{sc_1, sc_2, \ldots, sc_j\}$
Function: find the service set which accord with the domain ontology and the RGPS domain model

```
for I = 1 to n
 extract information of SName, R, G, P, S of service si
 calculate r = sim(SName, C) by Eq. 1
 if r>θ
calculate t = simm (si, RGPSM) by Eq. 3
if t>η si →SCCollection end if
 end if
end for
```

Particle swarm-based service composition: Some atomic services are too small to meet the whole requirements, so dynamic selection and composition of services are good solutions. Since some services are always combined to be provided to service requesters, the services frequently combined can be obtained either by user marking using Wiki-based requirement elicitation tool or via the domain ontology (Ning, 2012). The frequently-combined services can be pre-bound together to form a larger composite service to accelerate the service searching process.

In this study, the service composition is applied using the particle swarm optimization algorithm. The service with optimal QoS is chosen from a set of services in accordance with functionality, input and output requirements and then added into the service process. This procedure can guarantee the efficiency of the service composition and global optimum. The service composition in this study is conducted based on the hybrid particle swarm optimization algorithm proposed by Shu et al. (2011). The Qos attributes we choose include response time (Time), service price (Price) and reliability (Rel), the objective function of the algorithm is:

$$f(x) = \min (w_t \times Qos(Time) + w_c \times Qos(Cost) + w_r \times Qos'(Rel))$$

Where wt, wc and wr are weights based on user preferences and set as 1. Qos′ means the negative value of Qos. The constraints of the algorithm are selected by users.

The QoS of services after composition is calculated from the atomic services in composition and for details referring to (Xia *et al.*, 2011).

EXPERIMENT VALIDATION

Simulated experiments were carried out to examine the performance of the service preprocessing method proposed in this study. The method is implemented in Java using JDK5.0 JVM and is tested on a machine with 2.53G CPU and 2GB memory running Windows XP operating system.

In view of difficulties in finding large number of practical services available for experimental requirements, in this study, the tested data set contains simulated services via random generations. The services randomly generated include all the description information of services but no specific functionalities which makes no difference in this case.

The experiments were designed in the context of travel and its domain ontology is shown in Fig. 2. Total five hundred services were generated using RGPS domain model proposed by (Liu, 2008). Experiments show that the number of services selected reduces dramatically with the increment of the thresholds. As the limitation of our

Fig. 2: Domain ontology in the field of travel

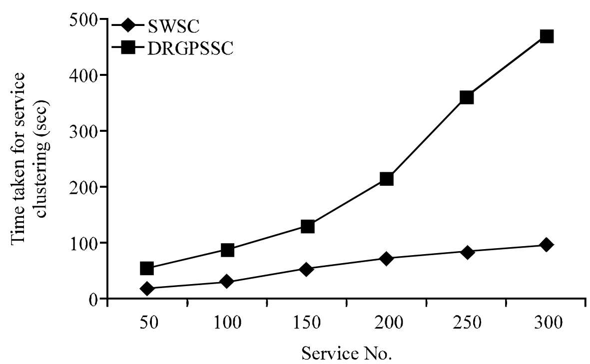

Fig. 3: Comparison of time taken in the service clustering

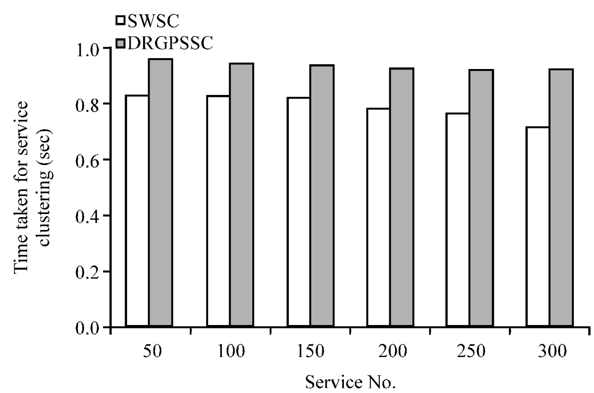

Fig. 4: Comparison of accuracy in the service clustering

services, too few recommended services may not satisfy users. So in the following experiments, we take the threshold $\theta = \eta = 0.6$.

Test 1 Comparison of service clustering methods: Two clustering methods are applied and compared in this study. One is the service clustering method (denoted by SWSC) presented by Nayak and Lee (2007) which is achieved by calculating similarity between sets of services. The other is the model proposed in this study labeled as DRGPSSC. A number of services of 50, 100, 150, 200, 250, 300 are taken and the comparison of service clustering time and accuracy from the two methods are shown in Fig. 3 and 4, respectively.

From Fig. 3, it can be seen that, for the same number of services, the clustering time of the SWSC method is less than that of the DRGPSSC method. This is because that, in the SWSC method, only the concept similarity is considered when comparing the ontology regardless of the inference relation of ontologies. In contrast, the DRGPSSC method has to take more time to make a comparison between five ontologies involving the domain and RGPS.

Figure 4 shows that the DRGPSSC method has a higher accuracy than the SWSC method in the case of the same amount of services. The enhancement in the accuracy of DRGPSSC method is due to the consideration of semantic relation which is not taken into account in the SWSC method.

Test 2 Comparison of service preprocessing methods: Zeng et al. (2011) presented a service registry system (labeled as DRLSP) where the domain ontology and the local domain ontology were used to annotate the services to facilitate the service selection. The service preprocessing method proposed in this study is

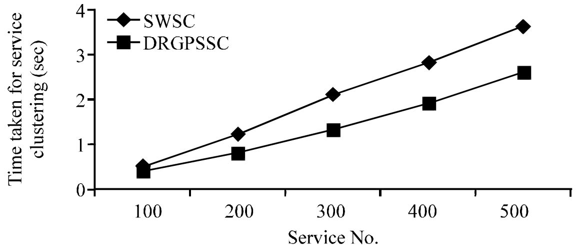

Fig. 5: Comparison of time taken in the service selection

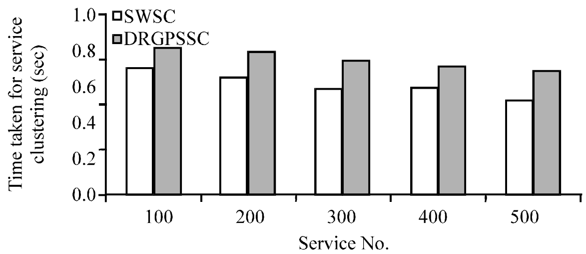

Fig. 6: Comparison of accuracy in the service selection

represented as DRGPSSP. In this test, a number of services of 100, 200, 300, 400 and 500 were taken. The average service selection time and accuracy calculated from 30 tested queries for the two methods are calculated and the results are compared as shown in Figure 5 and 6, respectively.

The results in Fig. 5 and 6 show that, for the same amount of services, the selection of services preprocessed by the DRGPSSP method is faster and more accurate. This benefits from an accurate service clustering and pre-composition of services available in the DRGPSSP method which makes the service searching easier.

CONCLUSION

In order to enhance the speed and accuracy of the service selection and composition, it is essential to preprocess the services stored in the service repository. The preprocessing of services based on the O-RGPS domain-modeling framework in this study is proved to be useful.

ACKNOWLEDGMENT

The research project is supported by the National Basic Research Program of China (973) under Grant No. 2014CB340404, the National Natural Science Foundation of China under Grant No. 61373037, 61202031, 61100017, 61100018.

REFERENCES

Huhns, M.N. and M.P. Singh, 2005. Service-oriented computing: Key concepts and principles. IEEE J. Internet Comput., 9: 75-81.

Liu, J.X., K.Q. He and D. Ning, 2012. Web service aggregation using semantic interoperability oriented method. J. Inform. Sci. Eng., 28: 437-452.

Liu, W., 2008. Study on field oriented service demand elicitation and analysis. Master's Thesis, Wuhan University, Wuhan.

Nayak, R. and B. Lee, 2007. Web service discovery with additional semantics and clustering. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, November 2-5, 2007, Fremont, CA., pp: 555-558.

Ning, D., 2012. Study on the on-demand service discovery method based on semantic web and social annotation. Master's Thesis, Wuhan University, Wuhan.

Salton, G., A. Wong and C.S. Yang, 1975. A vector space model for automatic indexing. Commun. ACM, 18: 613-620.

Shu, Z., H.H. Chen and X.S. Luo, 2011. A web service dynamic selection method based on improved hybrid particle swarm optimization algorithm. J. Central South Univ., 42: 3086-3094.

Wang J., K.Q. He, P. Gong, C. Wong, R. Peng and B. Li, 2008. RGPS: A unified requirements meta-modeling frame for net-worked software. Proceedings of the 3rd International Workshop on Applications and Advances of Problem Frames, May 14, 2008, ACM Press, pp: 29-35.

Xia, Y., P. Chen, L. Bao, M. Wang and J. Yang, 2011. A QoS-aware web service selection algorithm based on clustering. Proceedings of the IEEE International Conference on Web Services, July 4-9, 2011, IEEE Computer Society, Washington DC., pp: 428-435.

Yue, K., X.L. Wang and A.Y. Zhou, 2004. Underlying techniques for web services: A survey. J. Software, 15: 428-442.

Zeng, C., K.Q. He and B. Li *et al*., 2011. Mechanism of web service registry and repository for semantic interoperability. J. Chinese Comput. Syst., 32: 1709-1715.