TELKOMNIKA, Vol. 11, No. 8, August 2013, pp. 4505~4511
e-ISSN: 2087-278X

An Optimal Semantic Network-Based Approach for Web Service Composition with QoS

Naji Hasan AH*, Gao Shu, AL-Gabri Malek, Jiang Zi-Long
School of Computer Science, Wuhan University of Technology, Wuhan, 430063 China
Corresponding author, e-mail: hasanye1985@gmail.com[1], gshu418@163.com[2],
malekye2004@gmail.com[3], wuhanjzl@163.com[4]

Abstract

As the spread of web services, the composition of web services becomes a hot topic on both the academia and IT domains. The composition of services depends on the semantic descriptions, which describe the functionality of the services, provided by the owners of services and on the description of the consumer's requests. The composition also depends on non-functional descriptions (QoS). In this paper, we present an approach of web services composition based on both semantic description and QoS. The proposed approach builds a network of web services by matchmaking the semantic concepts in OWL-S, using the outputs-inputs similarity between services. Then, many composed assemblies can be created by combining web services in the semantic network according to consumer's needs. Those composed assemblies will be ranked by the QoS that provided by the consumer and the suitable one will be sent back to the consumer. This approach takes the advantages from Pellete DL Reasoner, depth-first and forward chaining algorithms. A prototype and a study case are presented to illustrate the efficiency of our approach.

Keywords: *semantic network, web service composition, QoS, similarity, forward chaining algorithm*

1. Introduction

Service-oriented computing (SOC) uses web services as the basic constructs to develop interoperable distributed applications. Web services are autonomous, platform-independent and loosely coupled entities which can be described, published, discovered, and composed [1]. Web Service is the key concept in Service Oriented Architecture (SOA), which is a model for utilizing distributed capabilities that may be under the control of different ownership domains [1]. In SOA, service consumers seek to satisfy their specific needs by using the capabilities offered as a service by a service provider. One approach to realize SOA is Web services [2].

Web service is a software system which designed to support interoperable machine-to-machine (application-to-application) interactions over the internet. It is based on eXtensible Markup language (XML) [4] that constitutes the main technology of Web services. However, Web services are, usually syntactically described with standards like (UDDI, SOAP, and WSDL). Universal Description, Discovery and Integration (UDDI) [3] is a virtual registry that stores and exposes information about Web Services. Simple Object Access Protocol (SOAP) is a protocol to exchange structured information in a distributed and decentralized environment. It utilizes XML to demean extensible framework of messages which provides a constructed message that can be exchanged through a variety of underlying protocols. The protocol SOAP is independent from any particular programming model and from any specific semantics of an implementation [5].

Web Service Description Language (WSDL) [3] provides a model and an XML format for describing Web services. It separates the description of the abstract functionality, offered by a service, from the concrete details of a service description such as how to get the service and where to get it [4-7]. WSDL files describe only the syntactic interface of Web services. Hence, the pure WSDL is not enough to be used for automatic Web services composition: Semantic Annotation is required in order to make information accessible to autonomous agents [8]. QoS are also another type of formats use to describe the non-function of web services.

Received January 25, 2013; Revised May 15, 2013; Accepted May 25, 2013

The main purpose of this paper is to present a new approach of web service composition based on both Semantic annotations using ontologies and QoS. In our proposed approach, a semantic network will be built according to the matchmaking among inputs and outputs of Web services, which supposed to be described semantically via ontologies.

This paper is organized as follows. Section 2 is devoted to the work related to Web service composition. In Section 3, we present our proposed approach, its architecture its algorithms and strategies. Section 4 presents a study case as an example to prove the efficiency of our approach. Finally, Section 5 summarizes our work, findings and future works.

2. Background And Related Works

2.1. Definitions and Concepts

(1) Semantic Network: is a web service network which is built according to the similarity measure among the inputs and outputs of web services using the semantic annotation.

(2) Semantic Annotation [7]: an annotation assigns to an entity, which is in the text, a link to its semantic description. A semantic annotation refers to an ontology.

(3) Similarity Measure [8]: The reasoner defines four levels of similarities between two concepts A and B, and Similarity Measure can be one of the following results:

A. Equivalence (Match): in this measure the concepts A and B are equivalent, similar concepts.

B. Subsumption (Subsumes): it means that the concept A is more general than the concept B;

C. Opposite Subsumption (Plug-in):it means A concept is subsumed to the concept B,i,e., the concept B is more general than A.

D. Difference (Fail): The concepts A and B are different totally.

(4) NFSL [9]: Non-functional Specification Language which describe the Qos of a web service.

OWL-S: an ontology language used to describe web services semantically.

(5) Reasoner [8, 9]: A mapping engine is a reasoner like the reasoner in Artificial Intelligence; it matches service advertisements with requests. The reasoner provides a semantic algorithm to match inputs and outputs of Web services during them at the matchmaking process.

2.2. Related Works

Web Service Composition (WSC) [10-11] is required when a client's request can not be fulfilled by a single pre-existing web services. In such a case, integrating existing web services into one composite service may satisfy the request. Due to the importance of web service composition, there have been various approaches to address the web service composition.

In [12], Hassina Nacer Talantikite et al, introduce a new approach of discovery and composition based on semantic annotation. An ontology reasoner has been used in similarity measure between services, and the composition results are compared according to similarity time and memory space.

Colucci et al. in [13] introduce a formalization of matching, and propose properties that should hold in a semantic-based skill matching approach. They also define an algorithm to rank matches between skills profile descriptions and present an ontology-based system which embeds a modified NeoClassic reasoner implementing the ranking algorithms.

The QoS-aware WSC problem is proven to be NP-hard [14], so when the problem space is large, it is usually time consuming to derive a QoS-optimal service composite solution. In [15] a Global method is proposed to solve the QoS-aware WSC problem, which models the QoS-aware WSC problem as a MIP problem and decreases the time cost to some extent.

Most of the composition works above are mainly to compose services using the semantic or QoS description and some works using both descriptions. In our work, we present an optimal web services comoposition based on semantic and QoS descriptions, which deals with the inputs and outputs of services and forms a network, which can give many options to obtain a suitable service composition, as we can see in the following sections.

3. An Optimal Semantic Network-based Approach for Web Service composition with QOS

In this section, we introduce our proposed approach. We begin with the architecture of our prototype, and then we present the matchmaking and discovery of web service along with the idea of building a semantic network. Finally the combining services according to consumer's requests and the ranking strategy using QoS are presented.

3.1. The Architecture of the Prototype

Our prototype consists of five components, namely, Agent, DB, Semantic Network, Composer and Evaluator. The architecture of our prototype is illustrated in Figure 1, The Following is a brief description of our prototype's components:

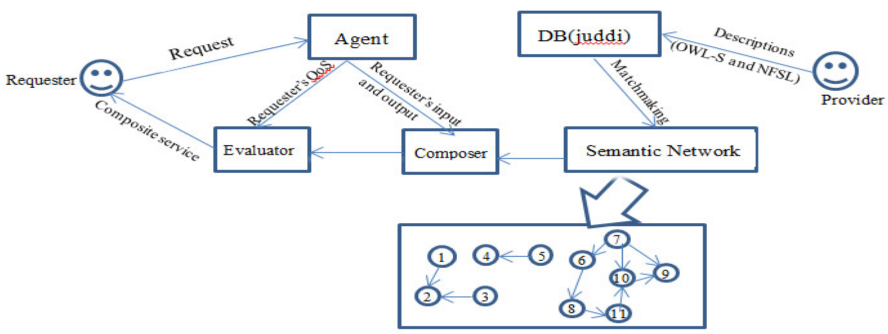

Figure 1. The Architecture of the Prototype

(1) Agent: consumer (or requester) provides requests to the Agent. The request includes inputs and output along with QoS parameters. Later, the Agent provides the Composer and Evaluator with those descriptions.

(2) Juddi or Database: In Database providers store theirs' service descriptions using OWL-S for semantic descriptions and NFSL for QoS (non-functional descriptions).

(3) Semantic network is the network which generated by matching inputs and outputs of services stored in Database using similarity measure (OWL DL reasoner), that includes a matching algorithm.

(4) Composer is completely responsible for composing Web services. The composer fulfills the responsibility of Web service composition using the generated semantic network and the requester's input and output. The composer takes the advantages from both Depth-First and Forward Chaining algorithms, and then generates composite services.

(5) Evaluator ranks composite services according to requester's QoS and presents the suitable composite service to the requester.

3.2. Matchmaking and Building a Semantic Network

In our approach, Services' providers are supposed to describe their services semantically and map their inputs and output as subclasses of classes (concepts) in a common ontology called Currency. The process of matchmaking and building a Semantic network, which builds before any submitted requests, and goes as follows:

(1) Obtain all services' inputs.

(2) Obtain all services' outputs.

(3) Measure the similarity between the concepts (output) of a specific service A and the concept (input) of other services using "Pellet DL" Reasoner which includes the following matchmaking strategy:

(a) If the output of a service A is "equivalent" to the input of service B, then create an edge between the two services and label the edge by similarity value "1".

(b) If the output of a service A is "subsumed" by the input of service B, then create an edge between the two services and label the edge by similarity value "2".

If the output of a service A is "independent", no relation with, the input of service B, then the similarity value is"0", no edge creates between the two services.

3.3. Composition Algorithm

The process of building a composite service from Semantic network goes as follows:
(1) Get requester's input (as Source) and output (as Destination).
(2) In semantic network, we look for all services whose inputs are similar with the Source.
(3) Create a composition for each found service.
(4) For each new source which is the inputs of services:

Search again in the semantic network all services whose outputs are similar with the new source whose output are not already treated.

(5) If all the inputs are treated or no more sources, then the loop will terminate, else go back to step (4).
(6) At the end, composer generates a list of composite services, each composite service starts from requester's input (source) and ends with requester's output (destination). Our study case will describe composition algorithm clearly.

3.4. Evaluation

Our Evaluator uses the following strategy to evaluate and rank composite services.
(1) Cost: can be calculated by getting the sum of the value of costs of services that form a composite service.
(2) Some QoS such as availability, can be calculated by getting the multiply of the availability value of services that consist a composite service. For example, a composite service which includes three web services and their availability values as follows:

(a) WS1 (availability) =0.94, WS2 (availability) =0.80, and WS3 (availability) =0.79. Then, the availability value of this composite service is 0.94*0.8*0.79=0.594

(b) Some QoS such as security can be calculated by choosing the lowest value. For example, a composite service which includes three web services and their security values as follows: WS1 (security) = high, WS2 (security) = medium and WS3 (security) = medium. The security value of this composite service is medium.

(3) Each composite service will be compared to requester's QoS and gotten a ranking as follows:

(a) The cost: the cost of composite service should be equal or smaller than the cost that is provided by the consumer.

(b) Other QoS: such as Availability, security and response time of composite service should be equal or larger than it the cost that is provided by the consumer.

Finally, the suitable composite service will be sent back to the requester as the optimal composite service.

4. Study Case

In this section, we illustrate and prove the efficiency of our approach using a study case. The following case study illustrates the techniques and strategies discussed for using Semantic Network and QoS on service composition. This study uses currency exchange services as its domain. The requester is Chinese tourist who wants to visit a country (Yemen), he needs to exchange Chinese currency (RMB) to Yemeni currency (Yemeni Riyal, YR in short) and he adds some QoS as his requirement.

Table 1. Web Services with their Descriptions.

No	Service Name	Input	Output	Cost	Security	Availability	Response time
1	RMB2YR	RMB	YR	20	HIGH	0.90	2
2	RMB2EURO	RMB	EURO	10	HIGH	0.77	1
3	RMB2USD	RMB	USD	5	MEDIUM	0.85	2
4	EURO2YR	EURO	YR	5	HIGH	0.76	1
5	EURO2SA	EURO	SA	2	MIN	0.89	3
6	SA2YR	SA	YR	1	MIN	0.78	0.5
7	USD2YR	USD	YR	1	HIGH	0.97	1
8	USD2SA	USD	SA	1	HIGH	0.69	1.5

There are lots of currency exchange services, and each service should be provided with its Semantic descriptions (OWL-S) and non-functional descriptions (QoS). Our approach tries to help in finding the suitable service (or composite service) and offer it to the requester. Table 1 shows some web services with their descriptions.
And Figure 2 shows the description of QoS in an NFSL file.

```
<?xml version="1.0" encoding="utf-8" ?>
<NFSL>
  <WSName>RMB2YR</WSName>
  <CostByUSD> 2 </CostByUSD>
  <Security> Low </Security>
  <AvailabilityByPercentage> 0.89 </AvailabilityByPercentage>
  <TimeBySeconds> 3 </TimeBySeconds>
</NFSL>
```

Figure 2. An Example of QoS in NFSL

In an NFSL file, WSName shows the name of web service, CostByUSD shows the cost of using this service, Security shows the level of security of the service, AvailabilityByPercentage shows the availability of the service by percentage, and TimeBySeconds describes the required time for executing the service.

4.1. The Matchmaking and Semantic Network

In matchmaking process, we focus on the “equivalent” and “independent” between the outputs and inputs of services and ignore other relations. Table 2 shows some web services and the relation between them. In this table, the rows present the inputs of services, and the columns present the outputs of services, each cell represents the relation (similarity) between the inputs and outputs of the services. For example, the output of “RMB2EURO”, in column 2, is equivalent to the input of services “EURO2YR” and “EURO2SA” in rows 4,5, then they have gotten “1”, while there is no relation between “RMB2EURO” and “USD2SA”, then they have gotten “0”.

Table 2. Web Services and their Relations after Matchmaking

	RMB2YR	RMB2EURO	RMB2USD	EURO2YR	EURO2SA	SA2YR	USD2SA
RMB2YR	0	0	0	0	0	0	0
RMB2EURO	0	0	0	1	1	0	0
RMB2USD	0	0	0	0	0	0	1
EURO2YR	0	0	0	0	0	0	0
EURO2SA	0	0	0	0	0	1	0
SA2YR	0	0	0	0	0	0	0
USD2YR	0	0	0	0	0	0	0
USD2SA	0	0	0	0	0	1	0

Figure 3 shows the web services and their relations in Semantic Network. If there is a relation between two services, the output of service A is equivalent to the input of service B, an edge will be created. Otherwise no edge between two services will be created.

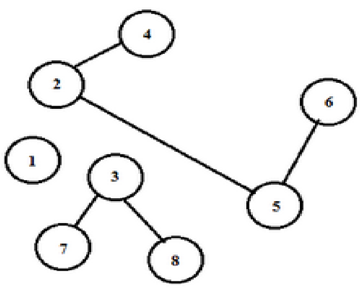

Figure 3. Web Services and their Relations in Semantic Network

4.2. Web Service Composition

The composer combines the web services in Semantic network according to requester's input and output which are submitted by the agent. It sets up the requester's input as the source point and the requester's output as the destination point. Figure 4 illustrates the result of combining Web services. As we have seen, we can fulfill requester's request using any of the following composite services:

(1) The atomic service No: 1:"RMB2YR".

(2) The composite service which goes through: No: 3" RMB2USD"→ No: 7" USD2YR".

(3) The composite service which goes through: No: 3" RMB2USD"→ No: 8" USD2SA"→ No: 6 "SA2YR".

(4) The composite service which goes through: No: 2" RMB2EURO"→ No: 4" EURO2YR".

(5) The composite service which goes through: No: 2" RMB2EURO"→ No: 4" EURO2SA" → No: 6 "SA2YR".

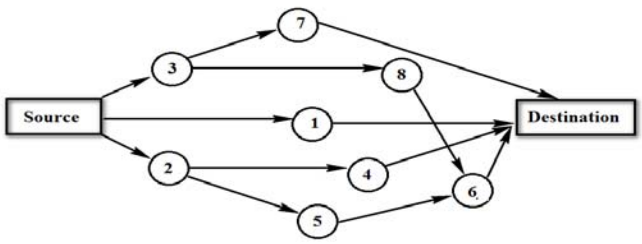

Figure 4. The Result of Combining Web services

4.3. Ranking and Evaluating

After generating the composite services, the Evaluator ranks and evaluates them according to the QoS which submitted by the Agent. The composite services, which are generated by the composer, with their QoS are listed in Figure 5.

```
1. Source(RMB) -> RMB2YR -> Destination(YR).
   cost=12.0, avai= 0.9, time= 2.0, security is High
2. Source(RMB) -> RMB2EURO -> EURO2YR -> Destination(YR).
   cost=15.0, avai= 0.722, time= 3.0, security is Medium
3. Source(RMB) -> RMB2EURO -> EURO2SA -> SR2YR -> Destination(YR).
   cost=17.0, avai= 0.65949, time= 5.5, security is Low
4. Source(RMB) -> RMB2USD -> USD2YR -> Destination(YR).
   cost=10.0, avai= 0.4365, time= 3.0, security is Medium
5. Source(RMB) -> RMB2USD -> USD2SA -> SR2YR -> Destination(YR).
   cost=11.0, avai= 0.24218999, time= 4.0, security is Low
required QoS is
 cost: 13.0, avail: 0.5, time: 3.0
The suitable composite service is :
Source(RMB) -> RMB2USD -> USD2YR -> Destination(YR).
   cost=10.0, avai= 0.4365, time= 3.0, security is Medium
```

Figure 5. The results of the study case

At the end the Evaluator sends the suitable composite service to the requester, which is the composite service:" RMB2USD->USD2YR "in our study case.

5. Conclusion

In this paper, we proposed an optimal approach for Web Service Composition based on semantic descriptions and on non-functional descriptions (QoS). In our approach, a semantic network has been established according to the similarity measure of the inputs and outputs of web services.

Lots of composite services can be built, and finally they were ranked according to consumer's QoS. A study case shows that this approach can compose services efficiently and can present the suitable composite service to the consumer according to his requirements

There will be following works in future.

1) Moving the idea of service discovery and service composition using ontology to cloud computing, specially the resources' cloud services.

2) A further development of the discovery and composition process with brokers in multi-cloud environment.

3) More study cases and tests should be carried out.

Acknowledgments

This work is supported by Wuhan International Cooperation Projects "Studying and Implementation on Semantic-based Visualization Services Publishing and Matchmaking Platform" (Project No. 200970634269).

References

[1] AB Hassine, S Matsubara, T Ishida. *A Constraint-Based Approach to Horizontal Web Service Composition*. Proc. Fifth Int'l Semantic Web Conf. (ISWC). 2006; 130-143.
[2] Alonso G, CasatiF, Kuno H, Machiraju V. Web Services- Concepts, Architectures and Applications. Berlin Verlag: *Springer*. 2004.
[3] Chia-Feng Lin, Ruey-Kai Sheu, Yue-Shan Chang, Shyan-Ming Yuan. A relaxable service selection algorithm for QoS-based web service composition. *Information and Software Technology*. 2011; 53(12):1370-1381.
[4] Hassina Nacer Talantikite, Djamil Aissani, Nacer Boudjlida. Semantic annotations for Web services discovery and composition. *Computer Standards & Interfaces Journal*. 2009; 31(6): 1108-1117.
[5] Jing Liu, Xing Guo Luo, Bai Nan Li, Xing Ming Zhang, Fan Zhang. An Intelligent Job Scheduling System for Web Service in Cloud Computing. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(6): 389-399.
[6] Lian Yong Qi, Wan Chun Dou, Xu yun Zhang, Jin Jun Chen. A QoS-aware Composition method supporting across-plat form service invocation in cloud environment. *Journal of Computer and System Sciences*. 2011; 78(5): 1316-1329.
[7] MP Papazoglou, P Traverso, S Dustdar, F Leymann. Service-Oriented Computing: a Research Roadmap. *International Journal of Cooperative Information Systems*. 2008; 17(2): 223-255.
[8] M Gudgin, M Hadley, N Mendelsohn, J Moreau. http://www.w3.org/TR/soap12-part1/. W3C Recommendation. 2012; SOAP Version 1.2 Part 1.
[9] M Papazoglou. Web Services and SOA: Principles and Technology. Canada: Pearson Education. 2012.
[10] Rajesh Karunamurthy, Ferhat Khendek, Roch, H Glitho. A novel architecture for Web service composition. *Journal of Network and Computer Applications*. 2012; 35(2): 787-802.
[11] Riyanarto Sarno, Hermawan. Developing Distributed System with Service Resource
[12] Oriented Architecture. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(2): 389-399.
[13] Schahram Dustdar, Wolfgang Schreiner. A survey on web services composition. *International Journal of Web and Grid Services*. 2005; 1(1): 1-30.
[14] Sérgio AB Cruz, Antonio MV Monteiro, Rafael Santos. Automated geospatial Web Services composition based on geodata quality requirements. *Computers & Geosciences*. 2012; 47: 60-74.
[15] Thomas H Corman, Charles E Leiserson. Introduction to Algorithms. MIT Press, US. 2009.
[16] WSDL1.1Specication.Web Services Description Language (WSDL) Version1.1.
[17] W3CNote. http://www.w3.org/TR/wsdl, 2012.