



MATCHING MODEL FOR SEMANTIC WEB SERVICES DISCOVERY

¹ ALIREZA ZOHALI, ² DR.KAMRAN ZAMANIFAR

¹ Dept. of Computer Engineering, Sama Technical & Vocational Training School, Khorasgan Branch, Isfahan, Iran

² Assoc. Prof, Dept. of Computer Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran

E-mail: Alireza.Zohali@Gmail.com, Zamanifar @eng.ui.ac.ir

ABSTRACT

The Semantics-based Web Service Matching Model is proposed in this paper to improve the performance of Web Service discovery. Semantics-based Web Service Matching is a two-phase method including semantic similarity matching and qualitative filtering. Function-based semantic similarity matching used of matching algorithm in order to finding the most proper services. And also, used for selecting the best service among results of qualitative filtering.

Keywords: *Web Service, Service Discovery, Function-based Semantic similarity matching, matching algorithm, qualitative filtering*

1. INTRODUCTION

Development of Service-Oriented Architecture (SOA) to provide software systems and Web Services brilliant growth, make use of a proper and practical procedure to discover and finding the services, very essential.

It's clear that finding out the interested services, without using mechanize methods is very difficult and time consumer. This issue is similar to search in web pages without using browsers. Service consumer can be a user, another service or a program. Thus using of automated mechanisms to finding out services is very important.

Services are the autonomous elements and often platform-independent that are described, registered and discovered and due to their integrity can provide a program.

The most sensible ability of service-oriented systems is the possibility of software elements assembly in order to building a grid of services with loose-coupling that dynamically make processes and handy programs. Basically, SOA is a designing style for sharing, reuse and services interchange in a distributed environment.

Web serviced through use of integrity features and reusability, the best solution for development of wide software scale is for enterprises. But by fast growth of the number of accessible services, the people will face to problems such As find out

proper service and above selection to make their programs [1].

In respect to complexity of finding proper service in a query, using of different techniques and procedures has been proposed in recent years. Among this, most basic methods stressed on functional, semantic and qualitative matching that remove the need of user by submitting functions. Semantic matching means to find out services with regard to functions meaning and concepts and service capabilities in used domain and context. Qualitative matching means to finding services that client and server can agree on offer procedure with respect to qualitative scales. Since Universal Description Discovery and Integration (Services specifications registry platform) that briefly calls it "UDDI", does not use of any semantic information in services registry and their searching abilities emphasis only on key words, can limit capability and efficiency of queries.

Therefore, semantic matching web service is proposed to increase efficiency of query. Moreover, incorporation of results similarity matching and qualitative filtering that used in proposed model of this study can modify the accuracy of proper web services [5]. The related work is first introduced in Section 2. Basic method and idea will describe in section3. Section4 represents the proposed model and Section5, shows the future works and conclusions.

2. RELATED WORK

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. W3C use the definition as follows: A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Recent years several papers have been represented for solving the problem. Matchmaker proposes a solution based on DAML-S, a DAML language for describing web services. And the discovery query in DAML-S is compared with the results that have features compatible with the request in semantic [2]. LARKS describe services as functional parameter input, output. The matching engine of the matchmaker agent contains five different filters: context, profile, similarity, signature and constraint filter [3].

3. MAIN IDEA AND METHODS

In order to implement the efficient service matching, a service matching model is includes four steps:

1) All services are defined according to the ontology of Web Service to generate semantic description, and when they are published, they are insured to have the same description schema.

2) The semantic information of the Web Service Description is extended according to the specification of UDDI, and the extended semantic information is stored in UDDI as tModel.

3) The set of Web Services is selected according to three matching steps, which are as follows:

- a) Input parameter matching
- b) Output parameter matching
- c) Well-defined matching by user

4) The best Web Service is selected according to qualitative filtering method on the set obtained in step3 [4].

In the following will introduce the methods used in these 4 steps in details.

3.1. Semantic Descriptions and Publication of Web Services

In order to give a standard form for the providers and requestors to describe and locate Web services, two types of Ontology are used in the model.

SWSD (Semantic Web Service Description Ontology) contains the semantic description of primary information, functional information, and other non-functional information of Web services

QoWSO(Quality of Web Service Description Ontology) describes the qualitative description of Web Services.

UDDI provides the Web Service specification for sharing of information and applications. In order to semantically development of UDDI, will store the semantic and qualitative information of Web service into the UDDI .In the matching model, we try to improve the quality in order to enhance the capability of UDDI.

1) Semantic Web Services Descriptive Ontology

The four types of information described in SWSD for service discovery are [4]:

- Primary Information and Provider Information
- Functional Description
- Quality Description
- Other Attributes Description

The structure of Semantic Web Service Description Ontology SWSD is shown in Fig .1.

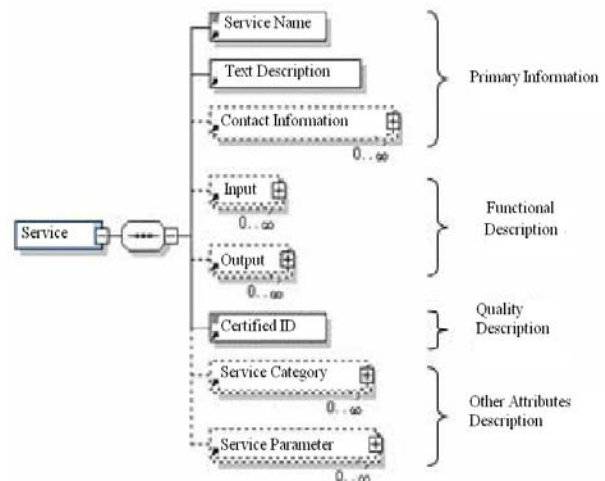


Fig. 1 Semantic Web Service Description Ontology (SWSD)

2) Quality of Web Service Description Ontology

The model applies a Quality of Web Service Description Ontology (QoWSO) to construct the consistent description of Web service quality (historical statistical information and the up-to-minute information of Web Service quality).

QoWSO is described by the class QoWSProfile, which is described by five quality parameters [4]:

- Stability
- Response Time
- Reliability
- AccessedTimes
- Grade

3) Mapping from SWSD to UDDI

Like the mapping from DAML-S Profile to UDDI, which is used in the registration of WSDL in UDDI Registry, the mapping from SWSD to UDDI includes:

The mapping from Provider Information in SWSD to Contacts of Business Entity in UDDI, the mapping from Primary Information as service Name and text Description in SWSD to name and description in Business Service, the mapping from SWSD service semantic description to tModel, the mapping from service quality description to tModel. The detailed mapping is shown in Fig .2.

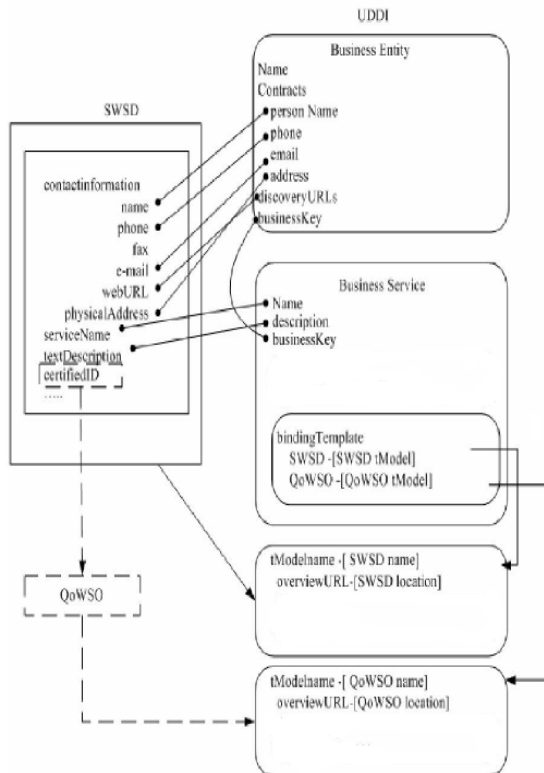


Fig. 2 Mapping from SWSD to UDDI [6]

3.2. Function-based semantic similarity matching of Web Service

Function-based semantic similarity matching of Web Service includes the semantic similarity matching of input and output, which is called IO matching for short. Assume that the input set and output set of service request are R_I and R_O respectively, and the input set and output set of advertised service are A_I and A_O respectively, therefore if $R_O \subseteq A_O$ which means that advertised service can provide all the outputs required by the service request, then output matching is successful; and if $A_I \subseteq R_I$ which means that service request can provide all the inputs needed by the advertised service, then input matching is successful.

1) Matching Algorithm

The main rationality behind our algorithm is that an advertisement satisfies a request when the advertisement provides all the functions (Inputs/Outputs) of the request's service. Also, the user can specify weights for all parameters, so the users can tell which parameters in their requests are most important and must be provided, and which parameters are not necessarily exactly matched.

The value of weights is limited as a decimal in $[0, 1]$ i.e. $0 < \omega < 1$. The default value for every weight is 1, which is an important factor for matching.

2) Main Control Loop

Requests are matched against all the advertisements stored in the registry in the main control loop of the algorithm as shown in Fig.3.

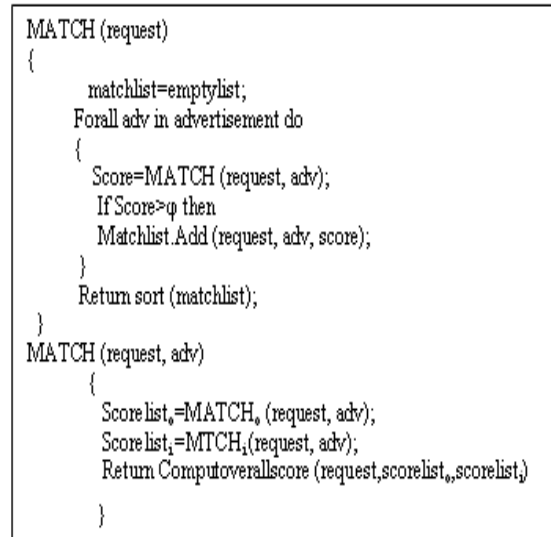


Fig.3. Main control loop



When the score (match degree figured out) of a match between the request and an advertisement is higher than ω which is a user specified threshold in the request, we add the advertisement and score to the result list. Because the services may have multi input or multi output, we define $ScoreList_o$ and $ScoreList_i$, as two lists to store the match degree of every output and input respectively.

Then we use the degrees acquired above and weights of every parameter given in the request to compute the overall score.

3) I/O Matching

The I/O matching is based on the degree assignment logic as follow [2].

Exact:

If request R and advertisement A are equivalent concepts, we label the match as Exact. Of course, it is the best level since the request and advertisement are exactly the same.

Sub-concept:

If request R is sub-concept of advertisement A (inductive advertisement) which is the second matching level.

Super-concept:

If request R is super-concept of advertisement A (the advertisement can provide only some specific cases of what the request desires) which is the third matching level.

Fail:

There are no relation between the advertisement A and request R. This is the lowest level which represents an unacceptable result.

Since the requests may have multi-output, we return a list containing the degrees of every output. And the list is used to compute the overall score of the matching with other parameters and their weights later.

The matching between inputs follows the above algorithm Fig.4 As outputs matching algorithm. The difference is just in that inputs of request are matched against the inputs of the advertisement.

```

MATCHo(request, adv)
{
  degreelist=emptylist;
  Forall outR in request do;
  {
    find outA in adv such that
    matchDegree=maxDegreeMatch(outR,outA);
    degreelist.add(matchDegree);
  }
  Return degreelist;
}
DegreeMatch(R,A)
{
  If R equivalent A then return Exact;
  If R sub-concept A then Return sub-concept;
  If R super-concept A then Return super-concept;
  Otherwise return Fail;
}
    
```

Fig.4 Algorithm for outputs matching

4) Computing the Overall Score

After the I/O matching, we should let the user know an overall score of the selected advertisement in the result list. And we accomplish the work with the function Fig.3:

Compute Overall Score (request, $ScoreList_o$, $ScoreList_i$)

We get the weight for every parameter from the request. Then we define the variable used to compute the overall score as follow

$ScoreList_o$: list of matching degrees of outputs, which is the return value of the outputs matching

$ScoreList_i$: list of matching degrees of inputs, which is the return value of the inputs matching.

S_{oi} : ($0 < i < |scoreList_o|$): match degrees of outputs in $scoreList_o$;

S_{ii} : ($0 < i < |scoreList_i|$): match degrees of inputs in $scoreList_i$;

W_{oi} ($0 < i < |scoreList_o|$): the weights of the outputs;

W_{ii} ($0 < i < |scoreList_i|$): the weights of the inputs;

$Score_i$: weighted average degree of inputs matching;

$Score_o$: weighted average degree of outputs matching;

overallScore : the overall score of the match;

Then we compute the match degree from (1), (2) and compute matching overall Score From (3)

$$Score_i = \frac{\sum_{i=1}^{|Scorelist_i|} W_{ii} * S_{ii}}{\sum_{i=1}^{|Scorelist_i|} W_{ii}} \quad (1)$$

$$Score_o = \frac{\sum_{i=1}^{|Scorelist_o|} W_{oi} * S_{oi}}{\sum_{i=1}^{|Scorelist_o|} W_{oi}} \quad (2)$$

$$OverallScore = \frac{Score_i + Score_o}{2} \quad (3)$$

5) Sorting Rules

Arrange the result in a reasonable order. If the scores in advertisement are the same, we would sort as follow: Over all Score > Score₀ > Score₁

3.3. Semantic filtering on QoWS

Use Semantic filtering on QoWS to find the best service in the case of existing a lot of functional similar Web Services. The base of Semantic filtering on QoWS is to compare the measured value of service quality parameters in the QoWS between different services. The service quality parameters include:

- Stability
- Response Time
- Reliability
- AccessedTimes
- Grade

The service quality parameters are classified as two kinds: One has the higher measuring value such as Reliability, AccessedTimes and Grade The other has the smaller measure such as Response Time and Stability .Finally can obtain the best service by compare the measured value and quality parameters.

4. ARCHITECTURE OF THE MODEL

The architecture of the service matching model is shown in Fig.5.

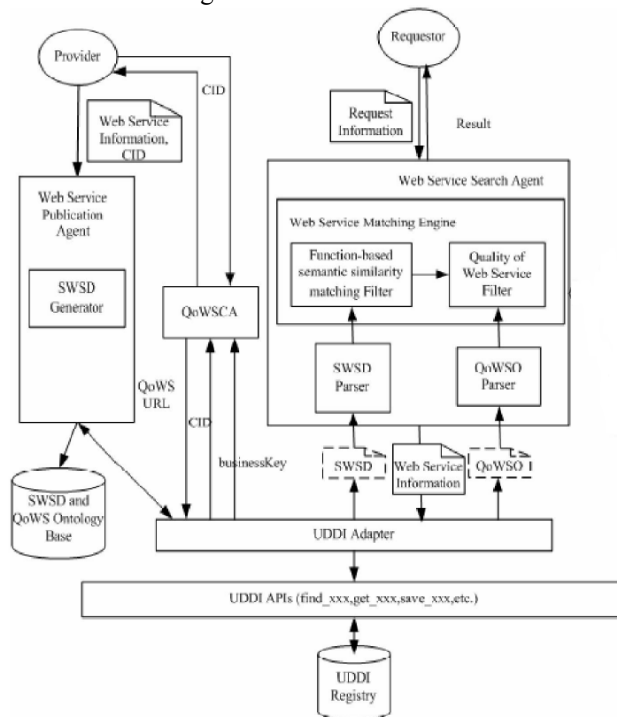


Fig. 5 The Architecture of Web Service Matching Model

The four function modules are Web Service Publication Agent, Web Service Search Agent, UDDI Adapter, and QoWS Certificate Authority. The two libraries are SWSD and QoWS Ontology Base, and UDDI Registry.

4.1. Web Service Publication Agent

The function of Web Service Publication Agent is to process the interaction between Service Provider and UDDI Adapter, and implement service publication. And the main part of Web Service Publication Agent, which is named SWSD Generator, is designed to generate the SWSD description of Web Service according to Web Service information and Specification of Web Service Description SWSD.

4.2. Web Service Search Agent

The function of Web Service Search Agent is to process the interaction between Service Requestor and UDDI Adapter, and implement service query. It includes the following three parts:

- SWSD Parser
- QoWSO Parser
- Web Service Matching Engine

1) SWSD Parser

Extracts the Web Service attributes from SWSD description file, generate the corresponding object instance, and provide the necessary data to Function-based semantic similarity matching filter.

2) QoWSO Parser

It extracts the QoWS information from QoWSO description file, generate the corresponding object instance, and provide the necessary data to QoWS Semantic filter.

3) Web Service Matching Engine

It matches the Service Requests and the advertised services registered in UDDI Registry to obtain the list of proper candidate services. It consists of Function-based semantic similarity matching filter and Quality of Web Service Semantic filter.

Function-based semantic similarity matching filter: It first gets the attributes of required service from SWSD Parser, and then executes semantic similarity matching between the attributes and SWSD function descriptions which are extracted from UDDI Registry by UDDI Adapter, and generates a list of functionally similar Web Services.

Quality of Web Service Semantic filter: It first uses QoWS descriptions of Web Services in the list of functionally similar Web Services, which are



extracted from UDDI Registry by UDDI Adapter, gets the attributes of QoWS from QoWS Parser, then execute the QoWS semantic filtering, and finally return the result to the Requestor.

4.3. UDDI Adapter

The function of UDDI Adapter is to process the interaction between other parts and UDDI Registry, which includes:

1) Find out the business services referred by the tModels in the list of function-based semantic similarity matching and obtain the list of matched Web Services.

2) Create the business service information structure of UDDI and store the information in UDDI Registry, by using Web Service information provided by Web Service Publication Agent and URL of QoWS description obtained from QoWSCA, and applying the mapping method from SWSD to UDDI. And transfer the Web Service ID to QoWSCA.

3) Invoke the interface of UDDIfind_tModel() to search all tModels belonging to swsdSpec in UDDI Registry for function-based semantic similarity matching.

4.4. QoWSCA

QoWSCA provides Quality Certification to the Web Services published in UDDI Registry

5. CONCLUSION

How to give up the proper service description in UDDI and how to match customer service request with service description, become the major challenges in Web Service discovery. Development of Web Service matching model proposed in this paper uses semantic description and quality description of Web Service based on ontology as the basis of service matching, and its matching algorithm is based on this logic that when a notice can satisfy a request that ever functions (Input/Output) provides request service. Moreover a user can specify some weights for all parameters and can say that which parameter of his/her request is premier and should provide and which one is unnecessary.

The service matching method composed of two phases. In the first phase, the proper Web Services that satisfy the functionality matching of the desired service is found, and in the second phase, the best one is selected from Web Services set obtained in the first phase using semantic filtering. Using these descriptions and matching phases, the performance of Web Service matching can be improved effectively.

REFERENCES:

- [1] R. A. Tsalgatidou, T. Pilioura,, "An Overview of Standards and Related Technology in Web Services" [J].*Distributed and Parallel Databases*, 2002(12),pp.135-162
- [2] M.Paolucci,T.Kawamura,T.R.Payne,"Semantic Matching of Web Services Capabilities",*International Semantic Web Services Conference*, 2002.
- [3] K. Sycara,S.Widoff,K.Klusch,and J.Lu, "LARKS:Dynamic Matchmaking among Heterogeneous Software Agents in Cyberspace," *Autonomous Agents and Multi-Agent Systems*, 2002, pp. 173-203.
- [4] J. Fan, B. Ren, L. Xiong, "An Approach to Web ServiceDiscovery Based on the Semantics", *Proceedings of 2005 International Conference on Fuzzy Systems and Knowledge Discovery*, 2005, pp. 1103 - 1106.
- [5] UDDI Universal Description, Discovery and Integration (UDDI2.0).*UDDITechnical White Paper* <http://www.uddi.org/specification.htm>.
- [6] F. Leymann and S. Pottinger , "Rethinking the Coordination Models of WS-Coordination and WS-CF", *Proceeding of the Third European Conference on Web Services(ECOWS'05)*, 2005