Available online at www.sciencedirect.com
SciVerse ScienceDirect
Physics Procedia 33 (2012) 236 – 243
Physics Procedia

2012 International Conference on Medical Physics and Biomedical Engineering

Web Service Matchmaking Based on Linguistic Variables

Dexin Zhao[1,2], Wenjie Li[1,2], Degan Zhang[1,2]

1. Tianjin Key Lab of Intelligent Computing & Novel Software Technology, Tianjin University of Technology，300191, Tianjin, China
2. Key Lab of Computer Vision and System (Tianjin University of Technology), Ministry of Education, 300191, Tianjin, China
qiqiharxin@163.com, lwj13579@sina.com, gandegande@126.com

Abstract

Matchmaking is considered as one of the crucial factors to ensure pervasive discovery of web services. Current matchmaking methods are inadequate to semantic information with machine understandable, therefore intelligent service discovery can not be carried on. In this paper, we use fuzzy linguistic variables to represent the vague or imprecise data at abstract level. The match method performs two levels, first level is capability match by inputs and outputs interfaces based on description logic reasoning, and the second level is the fuzzy match with linguistic variables, thus the more reasonable results will be presented to the users for selection.

Keywords—service matchmaking, linguistic variables, fuzzy logic, Web service

Introduction

Web service [1] has its advantages in solving resource sharing, realizing enterprise application integration, it is a software program identified by an URL, which can be accessed via the internet through its interface described by a series of operations using standard XML message. Web service may be registered with the UDDI [2] registry, which can subsequently be browsed and queried by users, services and applications. UDDI adopts services matchmaking mechanism based on framework, that is, the advertisement items and searching items are all in framework scheme. Only when the value of service items and the value of corresponding checking items are consistent completely, the matchmaking is successful.

At present, widely used service matchmaking technology, represented by UDDI, basically adopts syntactic levels (based on keywords, properties, interfaces, frameworks) matchmaking. With the enrichment of service and the demands of pervasive computing, development of semantic service

 doi:10.1016/j.phpro.2012.05.057

matchmaking is a trend, and web service based on semantics is a hot issue. Semantic Web [3] and Web service create intelligent automatic service and commercial processing facilities by utilizing the stored information to realize the intelligent service discovery of network resource, which can be understood by machines [4]. For instance, OWL-S/UDDI matchmaking engine in [5] is combined with OWL-S semantic matchmaking function.

However, classic semantics based on binary logic could not realize the representation and ratiocination of uncertain information. Currently, some scholars have done some valuable researches [6-8], but the theories are not mature. This article proceeds research of service matchmaking based on soft semantics [9-13], and addresses the fuzzy semantic matchmaking problem by using fuzzy logic and linguistic variable theory to be as the fundament of formal representation. We expand the traditional accurate keyword matchmaking results to approximate matchmaking, using linguistic value to measure the matchmaking degree and also respond to users. It is capable of dealing with terms for describing vague or imprecise information, which may relate to web services. This expression would be more efficient and effective than crisp terms in the search query.

A match algorithm is put forward in this paper: a step-by-step approach that performs two levels of matchmaking. The first level is capability match by inputs and outputs interfaces, and the second level is the fuzzy match with linguistic variables. This method is more suitable to the real situation and will provide users more choices to find the suitable services.

Definition of capability matchmaking

Most service discovery protocols are built on centralized architecture, which is the "service requester - service providers - service registry" model. In this model, the service provider releases advertising to the service registry, when the service requester asks for the service, service registry will compare the requirement with service request description, and then match the service request description to the advertising description. So the key issue of service discovery is the match between the request and the advertising description. Service requirement as input, service matchmaking will return all potential services in registry that meet the input. We give the following definition of matchmaking.

Definition 1 (service matchmaking). Service matchmaking is $matches(Q) = \{A \in \alpha \mid compatible(A, Q)\}$, where A is service advertising, Q is service request, α is the set of service advertising in service registry.

Definition 2 (service compatible). Compatible of two services means the service request is satisfiable, represented as $satisfiable(D_1, D_2) \Leftrightarrow \neg(D1 \sqcap D2 \sqsubseteq \perp)$, which represents the intersection of the two description sets is nonempty.

The perfect match between two service descriptions is very few, referring to paper [5], service advertising and user requirement are regard as ontology concepts, we define the service matches the following basic types in accordance with the difference in matchmaking accuracy degree.

Definition 3 (types of capability match). Service request Cr and service advertising Cs, if Cr and Cs are two same concepts, or a direct sub-concept relation, it is called *Exact* match $Cr \equiv Cs$; if Cs subsumes Cr, it is called *PlugIn* match $Cr \sqsubseteq Cs$; if Cr subsume Cs, it is called *Subsume* match $Cs \sqsubseteq Cr$, if the intersection of Cr and Cs is compatible, it is called *Intersection* match; Other cases are the failure match called *Disjoint*.

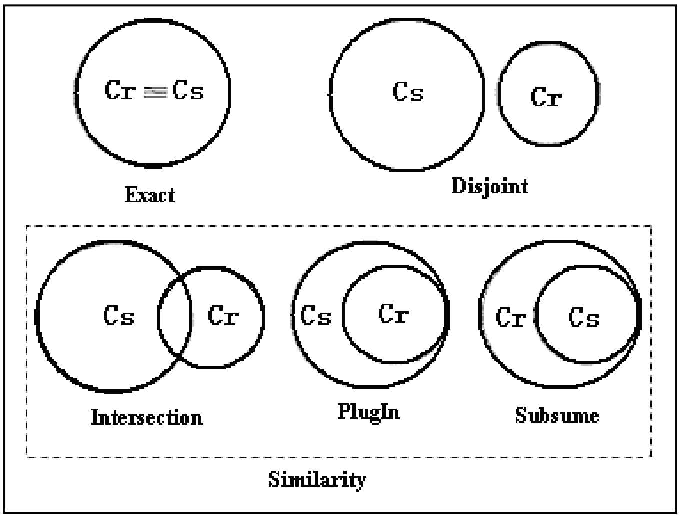

Figure 1. Types of capability matchmaking

From the above definition, *Exact* match is the most accurate and rigorous match, it is the special case of *PlugIn* and *Subsume*. While, *PlugIn*, *Subsume* and *Intersection* are the varying degree alternative scheme when the *Exact* match can not be satisfied for the user. *PlugIn* match is merely inferior to *Exact* match, since the service advertisement has contained the service request, moreover possibly has some other services. While the *Subsume* match is opposite with *PlugIn* match, the service request contains the service advertising, actually, it is a non-direct–inheritance relation. The *Intersection* match refers to the compatibility between the service advertising and the service request.

According to the match degree, the above five types in descending order is *Exact* > *PlugIn* > *Subsume* > *Intersection* > *Disjoint*. The match process is based on logic reasoning of ontology concepts, we propose that *PlugIn*, *Subsume* and *Intersection* the three match types can be unified into one group, called *Similarity* match, with together the *Exact* match and *Disjoint* match, there are three groups of service match types, as shown in Figure 1. **Capability machmaking algorithm**

In this algorithm, the match degree rank is "*Disjoint* < *Similarity* < *Exact*" order. Sub-class or super-class relations of ontology can be used for logic reasoning, and meeting a service request and service advertising will be one of the three types of results: *Exact*, *Approximate* and *Disjoint*. The algorithm consists of three parts, first is the main loop, which is the user requests matching with all service advertisement in the service registry center; second, customer's purchases match with each service supplier on input and output aspects. Finally, it is the three categories (*Disjoint*, *Similarity*, *Exact*) of service matching.

```
doMatch(Request){
    for all services in Repository do{
        globalDegreeMatch = Exact
        degreeMatch = matchDegree(outR, outS);
        if(degreeMatch = disjoint) return fail
        if(degreeMatch<globalDegreeMatch)
            globalDegreeMatch = degreeMatch
        degreeMatch = matchDegree(inR, inS);
        if(degreeMatch = disjoint) return fail
        if(degreeMatch<globalDegreeMatch)
            globalDegreeMatch = degreeMatch
     storeMatchList(service, globalDegreeMatch) }
    }
    matchDegree(Cr, Cs){
       if concept-equivalent(Cr, Cs) return Exact
       if concept-subsumes(¬Cr, Cs) return Disjoint
     Return Similarity}
```

Figure 2. Matchmaking algorithm

In the service match process, users' descriptions of services are not precise usually and sometimes it is impossible to be precise, because some conceptions can not be specified quantitatively, or it is not necessary to give accurate description. Under these cases, qualitative coarse description is enough. So the match method should have certain relaxation ability, which means, the algorithm can not only return the match type as result, but also return the corresponding match degree to provide the meaningful reference information, so that the user can select the most appropriate service. We will introduce a concept of keyword weight in the user service requests. Considering the representation habits of human natural language, we use linguistic weight instead of numerical weight, because in real problems, it is not so meaningful to differentiate the weight 0.8 and weight 0.84. The linguistic weight can be represented with words "importance", "frequency" and other abstract degree meaning in different cases, as described in next section.

Matchmaking with linguistic variables

We can regard "weight" and "matchmaking degree" as linguistic variables. Linguistic variables are the special variables which values could be defined as phases and words in natural languages or artificial languages. Its definition is given below [14].

Definition 4 (Linguistic Variable). Linguistic variables are represented with 5-tuples $(L, H(L), U_L, G_L, M_L)$. L is the name of the variable; U_L is the domain of L ; $H(L)$ is the set of linguistic variables, each linguistic variable is fuzzy set defined with U_L ; G_L is grammar rule used in generating the name of linguistic variable. M_L is semantic rule used to generate the membership degree function of fuzzy linguistic sets.

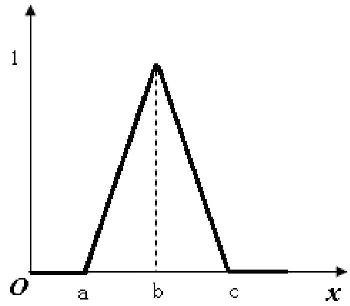

Figure 3. Triangular function

Here we define the fuzzy linguistic variable "weight" and "matchmaking degree", the domain is $U_L = [0,1]$, the linguistic values "weight" and "matchmaking degree" are unified into H(L)={none, extremely-low, very-low, low, medium, high, very-high, extremely-high, total}, which can be simplified as L={L0, L1, L2, L3, L4, L5, L6, L7, L8}. The grammar rule connects the modifier "very" "extremely" with fuzzy linguistic variable "low" "high". The membership degree of representation of the fuzzy set is specified by grammar rule M_L. The elements in linguistic value set may be defined with the triangular membership function in [0,1], as shown in figure 3.

To rational number $a < b < c$, triangular function $tri(a,b,c)$ is defined as following:

$$tri(a,b,c) = \begin{cases} 0, & x \le a \\ (x-a)/(b-a), & x \in [a,b] \\ (c-x)/(c-b), & x \in [b,c] \\ 0, & x \ge c \end{cases}$$

This paper uses 9 linguistic value tags to construct set for computing, adopts triangle fuzzy numbers as the semantic explanation of linguistic variable "weight" and "matchmaking degree", divides the linguistic value cardinality in [0,1] averagely, and get the semantic explanation in table 1.

In addition to capability description, service is also described with keywords, the service provider and demander can give each keyword its linguistic value weight. Such as, a service request of stock code checking can use "stock" as one keyword, and give the keyword a linguistic value weight. The value may be "*veryHigh*", which means the stock information's importance degree is *very high*. Formally, service possesses a linguistic value based keyword expression (k_i, λ_i), k_i is the *ith* keyword, λ_i represents threshold, which means service provider or demander request the importance degree of k_i in the content expressing.

Table 1 Relationship of linguistic values and fuzzy numbers

Symbol	**Match Degree**	**Fuzzy Numbers**
L_0	*none*	(0.0, 0.0, 0.0)
L_1	*extremelyLow*	(0.0, 0.0769, 0.1538)
L_2	*veryLow*	(0.1154, 0.2115, 0.3076)

L_3	*low*	(0.2596, 0.3605, 0.4614)
L_4	*medium*	(0.4110, 0.5131, 0.6152)
L_5	*high*	(0.5642, 0.6666, 0.7690)
L_6	*veryHigh*	(0.7178, 0.8203, 0.9228)
L_7	*extremelyHigh*	(0.8716, 0.9358, 1.0)
L_8	*total*	(1.0, 1.0, 1.0)

In the service register center, there is some data as following: service set $S=\{s_1,s_2,\cdots,s_m\}$, the keyword set $K=\{k_1,k_2,\cdots,k_n\}$, and fuzzy relation R. R is a fuzzy set on $K\times S$, it is a binary relation, called the relation between S and K, which can be represented by a $n\times m$ fuzzy matrix $A=[a_{ij}]$. Here, n represents the number of different keywords; m represents the number of services. That means, each word corresponds each row of the fuzzy matrix A, and each service corresponds each column of the matrix A. To fuzzy relation R, its membership function is represented as: $\mu_R(k_i,s_j)$: $K\times S\rightarrow[0,1]$, $(k_i,s_j)\in K\times S$, which $i=1,\cdots,n$; $j=1,\cdots,m$. In the fuzzy matrix, the element $a_{ij}=\mu_R(k_i,s_j)$, represents the strength of the relation between the keyword k_i and the service s_j, which can be viewed as the importance degree of keyword k_i with the service file s_j.

A simple example is as following: if $S=\{s_1,\cdots,s_6\}$, $K=\{k_1,\cdots,k_7\}$, the fuzzy relation can be represented as the fuzzy matrix:

$$A=\begin{bmatrix} & Y_{s1} & Y_{s2} & Y_{s3} & Y_{s4} & Y_{s5} & Y_{s6} \\ X_{k1} & 0.8 & 0 & 0.7 & 0.5 & 1 & 0 \\ X_{k2} & 0.5 & 1 & 0.6 & 0.4 & 0 & 0 \\ X_{k3} & 0 & 0.6 & 0.5 & 0.6 & 0.3 & 0.7 \\ X_{k4} & 0.7 & 0 & 0 & 1 & 0 & 1 \\ X_{k5} & 0.6 & 0.7 & 1 & 0 & 0.5 & 0 \\ X_{k6} & 1 & 1 & 0 & 0.7 & 1 & 0.3 \\ X_{k7} & 0 & 0 & 0.9 & 0 & 0 & 0.9 \end{bmatrix}$$

We can get the content of the service $s_1: Y_{s1}=0.8/k_1+0.5/k_2+0.7/k_4+0.6/k_5+1/k_6$, the intension of keyword k_1 is: $X_{k1}=0.8/s_1+0.7/s_3+0.5/s_4+1/s_5$, noting the represent method is a custom in fuzzy sets, "/" is used to represents the membership degree and the corresponding elements. And, in fuzzy matrix, $a_{ij}=0$ corresponds to the "*none*" in fuzzy language, and $a_{ij}=1$ corresponds to the fuzzy linguistic value "*total*".

Table 2 Linguistic values and their numbers

Symbol	**Linguistic Weight**	**Numeric Weight**
L_0	*none*	0.000
L_1	*extremelyLow*	0.077
L_2	*veryLow*	0.212
L_3	*low*	0.361
L_4	*medium*	0.513

L_5	*high*	0.667
L_6	*veryHigh*	0.820
L_7	*extremelyHigh*	0.936
L_8	*total*	1.000

As the users of the service request is based on the linguistic value weight, in the semantic correlation calculation, it is necessary to change the linguistic value into the numeric value weight, the relationship between these conversions as shown in table 2. The numeric weights are the average of triangular fuzzy numbers corresponding with the linguistic value. For example, if the user requests the key word is "tourism", given the weight of a "*veryHigh*", then in the calculation of semantic correlation, the weight is converted to numeric weight of 0.82. From the above two levels of matchmaking, the more reasonable results with fuzzy linguistic value will be presented to the users for selection.

Summary

We exploit fuzzy logic in order to classify and abstractly represent the underlying data of web services. The algorithm is a step-by-step approach that performs two levels of matchmaking. The first level is capability match by inputs and outputs interfaces, match results will be one of the three categories: *Exact*, *Approximate* and *Disjoint*. If it is the approximate result, then carries on the next level. The second level is the fuzzy match with linguistic variables, service advertisings and service requests are described by fuzzy key words with their values, it can further calculate the approximate match degree, and fuzzy value will return to the users.

Most web service matchmaking procedure lacks semantic information, and service description is syntax level, which may have ambiguous meanings. This paper focuses on fuzzy semantic service matchmaking method, introducing linguistic variables into service matchmaking procedure, that is, the vague concepts and human words are brought into the mathematical framework to represent and manage fuzzy concepts. The research proposed a useful way for web service matchmaking, and further works will concentrate the performance of matching algorithm.

Acknowledgment

This work is supported by National Natural Science Foundation of China (No.60773073 & 61001174), Program for New Century Excellent Talents in University of China (No. NCET-09-0895), Tianjin Natural Science Foundation (No. 10JCYBJC00500), Key project of Ministry of Education of China (No.208010), Science and Technology Development Foundation of Tianjin Higher Education Institution (No. 20080805 & 20080806).

References

[1] Alonso G., Casati F., Kno H., et al. Web service: concepts, architectures and applications[M]. Berlin: Springer-Verlag, 2004.

[2] Graham S. Building Web services with Java-making sense of XML, SOAP, WSDL and UDDI. Beijing:China Machine Press, 2003

[3] Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American, 284(5), 2001, 34-43.

[4] Katia Sycara, Massimo Paolucci, Anupriya Ankolekar, et al. Automated discovery, interaction and composition of Semantic Web services. Journal of Web Semantics, 2003, 1(1), 27~46.

[5] M. Paolucci, T. Kawmura, T. Payne, et al. Semantic matching of Web services capabilities. First Int. Semantic Web Conference, Sardinia, Italy, 2002, 333~347.

[6] Stoilos G., Simou N., Stamou G. Uncertainty and the Semantic Web. IEEE Intelligent Systems, 2006, 21(5), 84~87.

[7] Stoilos G, Stamou G, Pan JZ. Handling imprecise knowledge with fuzzy description logic. In: Parsia B, Sattler U, Toman D, eds. Proc. of the Int'1 Workshop on Description Logics. Aachen: CEUR-WS.org Publishers, 2006, 71-79.

[8] Jiang YC, Shi ZZ, Tang Y, Wang J. Fuzzy description logic for semantics representation of semantic web. Journal of Software, Vol.18, No.6, June 2007, pp.1257-1269 (in Chinese with English abstract).

[9] Dexin Zhao, Degan Zhang. Service description ontology language with vague representation. International workshop on ETT and GRS, Shanghai, China, 2008, Volume 1, 831-835.

[10] Dexin Zhao, Zhiyong Feng, Shizhan Chen, Qing Yu, Wenjie Li, Towards soft computing for service discovery. Proceedings of the Fourth International Conference on Information Technology and Applications. Macquarie Scientific Publishing, Sydney, Australia, 2007, Volume 1, 166~169.

[11] Dexin Zhao, Zhiyong Feng, Qing Yu, Guangquan Xu, A Fuzzy Extension of OWL for Vague Knowledge. Fourth International Conference on Fuzzy Systems and Knowledge Discovery. Los Vaqueros Circle, Po Box 3014, Los Alamitos, CA, 2007, Volume 4, 312~316.

[12] Dexin Zhao, Zhiyong Feng, Degan Zhang, Research on Fuzzy Semantic Service Matchmaking. International Conference on Biomedica Engineering and Informatics, Tianjin, China, 2009, Volume 4, 2230- 2234.

[13] Amit Sheth, Cartic Ramakrishnan and Christopher Thomas. Semantics for the Semantic Web: the implicit, the formal and the powerful. International Journal on Semantic Web & Information Systems, 2005, 1 (1), 1~18.

[14] L.A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. Beijing: China Science Press, 1982. 63~84.