Application of Web Services Discovery Mechanism Based on Semantics in Coal Mine System Integration

Deng Wei

Shaanxi Normal University, China.

Abstract

In order to increase production efficiency and ensure safe production, various monitoring systems and information management systems based on communication technologies and computer technologies have been increasingly applied in the coal mining industry. It has become a development trend to highly integrate a variety of independent operating systems and provide a unified control and control platform. In order to meet this requirement, various manufacturers publish system access interface externally in the form of Web services. How to find services to meet the requirements from a large number of Web services has become one of the hot issues in the field of system integration. In this paper, a three-level matching algorithm for Web services based on semantics has been designed and realized, and the ontology base in mine field has been established and tested. The results show that the algorithm effectively improves the precision and recallrate of the services discovery, has certain theoretical and practical significance for the development of semantic Web services and its application in the field of coal mine system integration.

Keywords

System integration; Web services discovery; Ontology; Semantic web; Semantic matching; OWL-S.

1. Background

In recent years, with the development of the emerging disciplines such as computer software engineering, communication engineering, the degree of modernization of the coal mine production continues to improve, various monitoring systems, information management systems have been widely used in the field of coal mine, which is of great significance of improving the production efficiency and ensuring safety production. But the subsequent problem is that there is the coexistence of a variety of heterogeneous systems, each isolated system consists of "information island". In order to achieve data integration and sharing, the construction of a unified management platform has a positive role in improving the office efficiency of management staff. In order to facilitate system integration, various manufacturers have published a system access interface externally in the form of Web services. The Web services uses the simple and flexible protocol, and the remote service access semantic definition and data representation use the popular XML format, message transmission supports HTTP protocol binding widely recognized, which is a good solution to the difference between the system heterogeneity between services and customers and development language used for services. At present, Web services have become a research hot-spot in the field of computer. As the number of Web services has been increasing, how to accurately and quickly find the service that meets the requirements has become one of the hot topics in the field of system integration.

It is found that the application of the retrieval method of traditional web search engine to web services will result in a low accuracy of the query result. This is because the method is matched based on the similarity of the text keywords and the computer cannot capture the service request and semantic information of serving advertisement contained in the text. After Berners Lee put forward the concept

of semantic Web[1] , he introduced it into the research field of Web discovery. Service information is represented in the form of computer understanding and processing, which is of great significance to meet the growing needs of Web service location [2].

2. Related Technologies

The Web services discovery mechanism based on semantics must first model specific domains, and establish ontology knowledge base, which is the basis for publishing services advertisements and finding services. At present, the methods of constructing domain ontology include IDEF-5 method, skeleton method, enterprise modeling method, METHODOLOGY method and seven-step method. The seven-step method is suitable for constructing small-scale, small-scale ontology[3]. Figure 1 is the ontology base established by the seven-step method in the field of coal mine production. When the service is published and searched, the OWL-S language[4] specifications are uniformly described. OWL-S defines a service through three levels of ServiceProfile, ServiceModel , and ServiceGrounding .

Web service discovery based on semantics mainly uses ServiceProfile. It usually contains three types of information [5]: service and its provider description, service function description and service non functional description. The functional attributes describe two aspects of the service: information conversion (described by the input and output) and state transitions generated by the service execution (described by the precondition and the effect), abbreviated as IOPE. Non-functional attributes include serviceCategory and qualityRating.

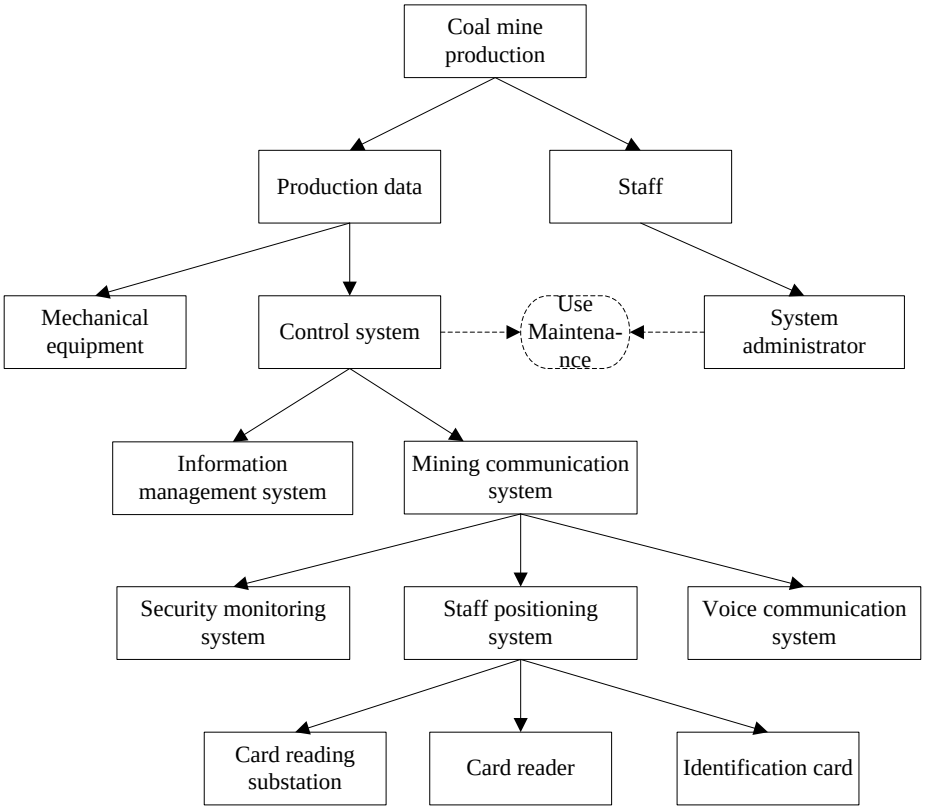

Fig 1:The ontology in the field of coal mine production

At present, the main researches devoted to the semantic Web services discovery in the academic community are the elastic matching algorithm based on Web services capabilities proposed by Massimo Paolucci of Carnegie Mellon University of the United States [6], OWL-S Matcher[7] of Berlin Technical University, and the METEOR-S project of the University of Georgia of the USA[8].

The elastic matching algorithm has become the core idea of later service discovery research. It is a method based on logical reasoning. It is matched with the input and output of service requests and service advertisements as parameters, and the matching degree is divided into four types: Exact, Plugin, and Subsume, and Fail. The algorithm mainly uses the inheritance relationship between

concepts in the domain ontology, and matches according to the relationship between concepts expressed in the domain ontology.The matching between service requests and service advertisements depends on the matching between all of their output parameters and input parameters. The matching between each output parameter or input parameter in turn depends on the degree of matching between their corresponding concepts. Assuming that OutR represents the concept definition of the output parameters of the service request in the corresponding domain ontology, OutA represents the concept definition of the output parameters provided by the service advertisement in the corresponding domain ontology, then the meaning of the four matching degrees is as follows.

Exact: which means that OutR and OutA are equivalence classes or OutR is a direct subcategory of OutA. The mining communication system (request) in Figure 1 can be seen as an exact matching for the control system (advertisement).

Plugin: which means that OutA contains OutR, OutR is an indirect subcategory of OutA, and it is one level lower than the exact matching. In Figure 1, the mining communication system (request) can be seen as an plugin matching for card reading substation (advertisement).

Subsume: which means OutR contains OutA. Containment relationship is not always correct because only specific subcategory is provided, so it is one level lower than the plugin matching. The card reading station (request) in Figure 1 can be seen as an Subsume for the staff positioning system (advertisement).

Fail: which means the OutA and OutR do not have the above three relationships. In Figure 1, the mining communication system (request) and the system administrator (advertisement) can be seen as a match failure.

The shortcomings of the elastic matching algorithm are reflected in the following aspects:

The results of matching between services are defined too roughly. The elastic matching algorithm can only divide the service matching results into four levels, but still cannot distinguish the relative matching degree of two Web services that belong to the same level.

2)The semantic relationships used are too thin. Ontology not only has inheritance relations between concepts, but also has a large number of binary relationships between concepts. There is a relationship of use and being used between system administrators and mining communication systems in Figure 1. Ontology concepts form a complex network through these numerous binary relationships to express richer semantic meanings. Binary relationships are weaker than inheritance relationships, but ignoring them may result in the loss of relevant and reasonable services, and thereby, resulting in the reduction in the recall rate. Therefore, the service matching algorithm should introduce the binary relationship semantics between ontology concepts into the calculation of the matching process.

3) Just match the parameter concepts of the input and output of the services, the category matching of the services and the non-functional matching of the services are not taken into account. The algorithm does not support quality-constrained service selection and does not fully satisfy users' search requirements.

3. Three-level Matching Algorithm

In view of the deficiencies of the existing algorithms, a three-level matching algorithm is proposed in this paper, which is specifically classified in the matching process as follows:

The first level is the matching comparison of service categories; the second level is the matching comparison of service functions based on input and output. It is the most important part of the entire matching process and is the core part of the matching algorithm; the third level is the matching of service quality, mainly It is the final performance screening of the discovered services for non-functional attributes. Finally, the comprehensive similarity of service matching is obtained by the comprehensive weighting of the similarities calculated at the previous stages. The three-level matching algorithm proposed in this paper is discussed in detail below.

3.1 Matching of service categories

Matching based on service categories requires a widely recognized directory of service categories. The matching of service categories is to find out which service advertisements have the same or similar service types as the service requests.All service categories are stored in a directory tree structure. Figure 2 shows an abstract service category tree. In this directory tree, different nodes represent different categories, and directed edges represent the generalization relationship between categories.

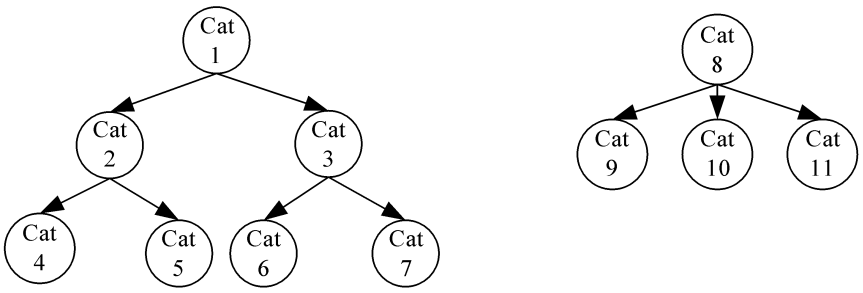

Fig 2 Abstract service category tree

The degree of similarity between two categories can be compared by type similarity. This article refers to Tversky's attribute-based similarity model [9] to define category similarity.

Assuming Cat to be a node on the category tree, then all the ancestors of Cat and Cat itself constitute a non-empty set. This non-empty set is called the upper category of Cat, denoted as $UpCategory_{Cat}$.

The similarity $Sim_{Cat}(Cat_i, Cat_j)$ between any category Cat_i and category Cat_j is between [0,1], and its value can be obtained by Formula 1.

$$Sim_{Cat}(Cat_i, Cat_j) = \frac{|UpCategory_{Cat_i} \cap UpCategory_{Cat_j}|}{|UpCategory_{Cat_i} \cup UpCategory_{Cat_j}|} \quad (1)$$

Wherein, $|UpCategory_{Cat_i} \cap UpCategory_{Cat_j}|$ indicates the number of elements in the upper category intersection of Cat_i and Cat_j .

$|UpCategory_{Cat_i} \cup UpCategory_{Cat_j}|$ indicates the number of elements in the upper category union of Cat_i and Cat_j .

When Cat_i and Cat_j belong to the same category, namely, $Cat_i = Cat_j$, $Sim_{Cat}(Cat_i, Cat_j) = 1$.

When there is no public upper category between Cat_i and Cat_j ,namely, $UpCategory_{Cat_i} \cap UpCategory_{Cat_j} = \varnothing$, $Sim_{Cat}(Cat_i, Cat_j) = 0$.

When the value of category similarity is greater than 1, the categories between two services are more similar. In practical applications, a threshold needs to be set. When the category similarity is less than the threshold, the service advertisement and the service request are not regarded to be in the same category, and the result of the matching failure is returned.

3.2 Matching of service function

Function matching is the core of the Web service matching algorithm. The input and output of the services represent the externally published interfaces of the service and they are particularly important in function matching.

In this paper, the rank of elastic matching algorithm is extended by introducing a binary relationship, a BinRelation matching level is added after Subsume. And at the same time, in order to improve the insufficiency of the definition of the Exact relationship in the elastic matching algorithm, it is

specified when the request parameter is a direct subcategory of the service advertisement parameters, the relationship between the two is Plugin.

In this paper, each level is quantified with a number of [0,1], and the values corresponding to the matching hierarchies Sim_g obtained by the extended elastic matching algorithm are shown in Table 1.

Table 1 Matching rank quantization table

Matching rank	Exact	Plugin	Subsume	BinRelation	Fail
Sim_g	1	0.8	0.6	0.5	0

In this paper, the similarity is calculated by first measuring the semantic distance between concepts, and then converting the semantic distance into semantic similarity.

According to Literature [10], there is a directed graph between the concepts of domain ontology. This paper also uses the shortest path distance to represent the semantic distance between two concepts. The semantic distance $Dist(C_1, C_2)$ of two concepts C_1, C_2 is defined to connect the sum of the weights of the n edges on the shortest path, as shown in Equation 2.

$$Dist(C_1, C_2) = \sum_{i=1}^{n} weight_i \tag{2}$$

Wherein, $weight_i$ is the weight weight on the i-th edge on the shortest path of connecting C_1, C_2. The shortest path here can be obtained by the Floyd-Warshall algorithm[11].

Considering the influence of the depth and breadth of the concept tree on the semantic distance of the concept, this study uses Formula 3 to define the weight of the edge.

$$weight(C_1, C_2) = \begin{cases} 1 & C_1 \text{ is the root node} \\ \dfrac{weight(parent(C_1), C_1)}{\max(|children(C_1)|, 2)} & C_1 \text{ is other non-leaf node} \end{cases} \tag{3}$$

Wherein, $weight(C_1, C_2)$ represents the weights of two adjacent concepts of the edges of C_1, C_2. Wherein, C_1 is the parent node of C_2, $parent(C_1)$ indicates the parent node of C_1; $weight(parent(C_1), C_1)$ represents the weight of the edge between Concept C_1 and its parent node, $|children(C_1)|$ indicates the number of child nodes in the concept tree C_1.

According to the Floyd-Warshall algorithm, after the semantic distance between concepts is calculated, it is needed to convert it into semantic similarity. The similarity must be guaranteed within the [0, 1] interval. This study defines the similarity function as Formula 4 here.

$$Sim_d(C_1, C_2) = 1 - \sqrt{\frac{Depth(C_2)}{Depth(C_1) + Depth(C_2)} \times Dist(C_1, C_2)} \tag{4}$$

The concept similarity $Sim_{ps}(C_1, C_2)$ of the matching level Plugin and Subsume can be obtained by Formula 5:

$$Sim_{ps}(C_1, C_2) = \frac{Sim_g(C_1, C_2) + Sim_d(C_1, C_2)}{2} \tag{5}$$

If the matching level between the two concepts is BinRelation, then assuming $Sim_b(C_1, C_2) = Sim_g(C_1, C_2) = 0.5$, this similarity is propagated to all descendants of C_1 and C_2, namely:

$$Sim_b(allSubClasses(C_1), C_2) = Sim_b(C_1, allSubClasses(C_2))$$

$$= Sim_b(allSubClasses(C)_1, allSubClasses(C_2)) = Sim_b(C_1, C_2)$$

According to the above algorithm, after the semantic similarity of the input and output parameters of the service requests is obtained, the similarity of the service function can be calculated.

The calculation method of service output matching degree is shown as Formula 6:

$$Sim_O(reS, adS) = \frac{\sum_{o \in O_{reS}} Sim_{\max}(o, O_{ads})}{|O_{res}|} \tag{6}$$

Wherein, O_{res} represents a set of service requests output parameters, $|O_{res}|$ indicates the number of the parameters contained in output sets of the service requests. $Sim_{\max}(o, O_{ads})$ represents the similarity value corresponding to the parameter with the highest degree of similarity to the requested specific output parameters in the output parameter set of the service advertisement.

The input similarity calculation method is similar to the output similarity calculation method, such as Formula 7.

$$Sim_I(reS, adS) = \frac{\sum_{i \in I_{reS}} Sim_{\max}(i, I_{ads})}{|I_{res}|} \tag{7}$$

The similarity of service function combines the results of output similarity and input similarity, as shown in Formula 8.

$$Sim_{IO}(reS, adS) = \alpha Sim_I(reS, adS) + \beta Sim_O(reS, adS) \tag{8}$$

Wherein, α, β are the weights of input and output, in order to meet $\alpha + \beta = 1$, α, β can be assigned according to the difference in the importance of the input and output, and then the functional similarity of the service is calculated.

3.3 Matching of service quality

The key to the quality of service matching is to compare the quality of service parameters between service requests and service advertisements. For the sake of practicability and measurability, this study considers the quality of Web services in terms of cost, response time, reliability, and reputation grade. The degree of similarity is expressed as $Match_{Cost}$, $Match_{Time}$, $Match_{\text{Reliability}}$, and $Match_{Grade}$.

This study refers to the ideas of Literature [12] and divides the quality of service into cost and benefit. The so-called cost type refers to a parameter whose quality is worse as the value is larger, such as Time or Cost. The so-called benefit type refers to a parameter whose quality is better as the value is larger, such as Reliability, Grade. The direction adjustment factor θ is introduced to get Equation 9 as follows.

$$Match_{QoSn}(reS, adS) = 1 - \frac{|adS_{QoSn} - reS_{QoSn}|}{\theta \times reS_{QoSn}} \tag{9}$$

Wherein, $\theta = \frac{q_{adS_{QoSn}}}{q_{reS_{QoSn}}}$, for cost-type parameters, q is calculated as Equation 10.

$$q_{ads_{QoSn}} = \begin{cases} \frac{adS_{\max} - adS_{QoSn}}{adS_{\max} - adS_{\min}} & \text{adS}_{\max} \neq adS_{\min} \\ 1 & \text{adS}_{\max} = adS_{\min} \end{cases} \tag{10}$$

For benefit-type parameters, q is calculated as Equation 11.

$$q_{ads_{QoSn}} = \begin{cases} \frac{adS_{QoSn} - adS_{\min}}{adS_{\max} - adS_{\min}} & \text{adS}_{\max} \neq adS_{\min} \\ 1 & \text{adS}_{\max} = adS_{\min} \end{cases} \tag{11}$$

Wherein, $adS_{\max}$, $adS_{\min}$ respectively express the maximum and minimum values in all issued advertisements for the components of the specified QoS in the Formula.

This ensures that if the service quality is a cost-type (benefit-type) parameter, when the service advertisement is smaller (larger) than the service request, $\theta > 1$; when the service advertisement is

larger (smaller) than the service request, $\theta < 1$. Therefore, when the absolute values of $adS_{QoSn} - reS_{QoSn}$ are equal, Small (large) service advertisements have similar quality of service. It should be noted that after introducing the adjustment factor θ, the similarity may be less than zero. This study stipulates that the similarity of service quality in this case is equal to zero.

After the above three levels are matched in order, the synthetic similarity Sim(reS, adS) of service matching can be calculated by Equation 12.

$$\mathrm{Sim(reS,adS)}=\omega_1\mathrm{Sim}_{\mathrm{Cat}}(\mathrm{reS,adS})+\omega_2\mathrm{Sim}_{\mathrm{IO}}(\mathrm{reS,adS})+\omega_3\mathrm{Sim}_{\mathrm{QoS}}(\mathrm{reS,adS}) \quad (12)$$

Wherein, $\sum_{i=1}^{3}\omega_i = 1$, the pseudo-code of matching synthetic similarity algorithm is as follows:

ALGORITHM:SERVICE_SIMILARITY
INPUT: a service advertisement A=($Name_a, Desc_a, Cat_a, I_a, O_a, Qos_a$) and
a service request R=($Name_a, Desc_a, Cat_a, I_a, O_a, Qos_a$)
OUTPUT: return Sim(R,A)
METHOD:
1: $calSim_{Cat}(R, A)$;
2: $calSim_{IO}(R, A)$;
3: If not existing Qos_r Then
4: $\mathrm{Sim(R,A)}=\omega_1\mathrm{Sim}_{\mathrm{Cat}}(\mathrm{R,A})+\omega_2\mathrm{Sim}_{\mathrm{IO}}(\mathrm{R,A})$ and $\omega_1+\omega_2=1$;
5: Else
6: $calSim_{QoS}(R, A)$
7: $\mathrm{Sim(R,A)}=\omega_1\mathrm{Sim}_{\mathrm{Cat}}(\mathrm{R,A})+\omega_2\mathrm{Sim}_{\mathrm{IO}}(\mathrm{R,A})+\omega_3\mathrm{Sim}_{\mathrm{QoS}}(\mathrm{R,A})$
8: and $\omega_1+\omega_2+\omega_3=1$;
9: End If
10: return Sim(R,A);

4. Performance Testing

4.1 Measure index

Currently, there is no unified standard for measuring the performance of service discovery methods in the field of Web services. In order to measure the performance of service discovery in three-level matching algorithms, this study draws on the three indexes of precision rate, recall rate [13], and system response time in the field of information retrieval. The precision rate refers to the rate of the number of Web services related to the query in the query results to the total number of Web services returned by the query results. The recall rate refers to the rate of the number of Web services related to the query in the query results to the number of all service advertisements. System response time refers to the time it takes for a user to submit a request to the system to return a matching set of services. For a system, the higher the precision rate and recall rate, the shorter the response time, and the better the performance.

4.2 Test methods

This paper implements a semantic-based Web services publishing and discovery prototype system. In order to build a service test set, a number of 140-150 Web services for coal mine production are published in the prototype system, among which, the number of Web services for staff positioning system is between 60 and 70. Since the calculations of precision rate and recall rate must clearly know the number of services truly related to the query results, therefore, the services test set is not

completely released randomly. For the services related to the staff positioning system, deliberate parameter design and classified processing is performed in advance.

In this paper, the mining communication system and its sub-categories service query are searched by using three methods of discovery. The three methods are: UDDI service discovery based on keywords, service discovery based on elastic matching algorithm and hierarchical service discovery method proposed in this paper. The purpose is to reflect the advantages and disadvantages of the three-level matching algorithm through horizontal comparison. Since neither of the first two methods includes quality of service, it is not considered here for fairness of comparison.

The keyword-based method is mainly a string matching of the service name and the service description. If the name or description of the service advertisements includes the requested service name, it is considered as a matching.

The method based on the elastic matching algorithm and the hierarchical discovery method in this paper do not consider the service name and service description, but inquires about the input and output parameters and the service category. In the test, this paper takes $\alpha = \beta = 0.5$ and sets the similarity threshold of the service category and service function to be 0.65, the test results shown in Figure 3 can be obtained through experiments.

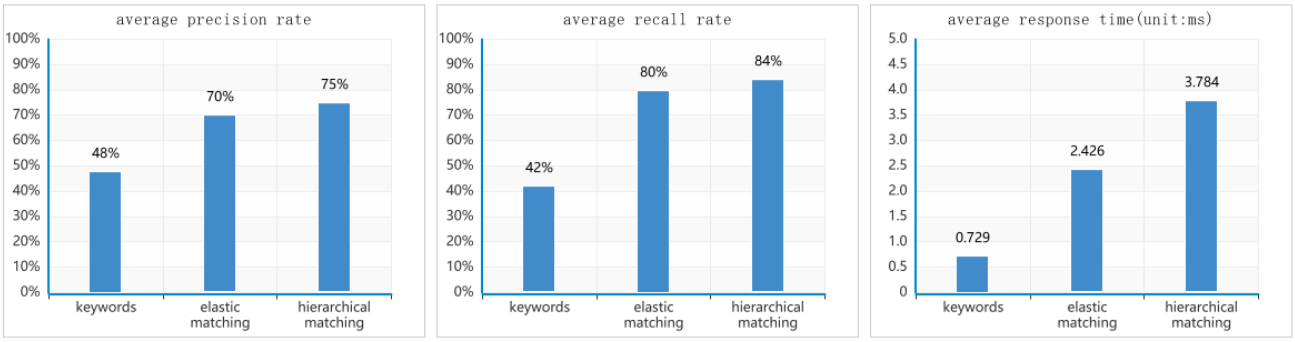

Figure 3 Performance comparison of test results

4.3 Result analysis

Figure 3 shows that the keyword-based service discovery, elastic matching service discovery, and the hierarchical service discovery method proposed in this paper have an average precision rate of 48%, 70%, and 75%, and the average recall rate of 42% and 80%. 84%, the average response time is 0.729ms, 2.426ms, 3.784ms. According to the test results, it can be found that the precision rate and recall rate of keyword-based service discovery method are relatively low, because simple string inclusion judgments cannot effectively identify homonymys and synonyms, the performances of the method based on elastic matching and the hierarchical matching method proposed in this paper are significantly improved compared with the keyword-based service discovery method.

In terms of recall rate, the service discovery method proposed in this paper takes into account the binary relationship between concepts, which increases the possibility of returning related services to a certain extent, and is therefore higher than the recall rate of the elastic matching algorithm.

In terms of precision rate, the specific algorithm of the service discovery method proposed in this paper is to perform semantic distance calculation under the guidance of matching level, and quantify the matching results of each step, and the matching similarity is expressed as the value of [0, l] , making the algorithm have a strong distinguishing ability. Therefore, the precision rate of the proposed algorithm is higher than that of the elastic matching algorithm.

3) In terms of system response time, keyword-based method only requires simple string matching, so their response time is the shortest. The hierarchical matching method of this study is preprocessed in the service publishing stage. In the query stage, only the corresponding service index database needs to be retrieved, and the response time is also within the range acceptable to the users. The elastic matching algorithm implemented in this paper is also indexed in the service index database when it

is published. Because it only performs functional matching and does not need to retrieve the semantic similarity between concepts, the response time is shorter than the hierarchical algorithm in this paper. It can be seen from the experimental results, the algorithm proposed in this study is higher than the keyword-based matching method and the elastic matching algorithm in terms of recallrate and precision rate, reflecting that the algorithm has strong service search capability and can be popularized and applied at the level of service discovery in the field of enterprise information integration.

References

[1] Berners-Lee T. Semantic Web Road Map[R/OL]. W3C, 1998-09, http://www. w3.org/ DesignIssues /Semantic.html.
[2] Berners-Lee T, Hendler J, Lassila O. The Semantic Web[J]. Scientific American,2001, 284(5):34-43.
[3] Wang SF, Method and Experience of Ontology Building[J/OL]. China XML Forum, 2007, http://bbs.xml.org.cn/dispbbs.asp?boardID=2&ID=7486.
[4] OWL-S: Semantic Markup for Web Services. [EB/OL]. W3C. [2004-11-22]. http://www.w3.org/Submission/OWL-S.
[5] Wang Xianghui. Semantic Web Service Composition Considering IOPE Matching [J]. Journal of Tianjin University, 2017, 09:984-996.
[6] Paolucci M, Kawamura T, Payne TR. Importing the Semantic Web in UDDI[C]. In Proceedings of Web Services, E-Business and Semantic Web Workshop, 2002:225-236.
[7] Tang S. Matching of Web Service Specifications using DAML-S Description[D]. Germany: Technische Universitat Berlin. 2004.
[8] Cardoso J, Sheth A. Semantic E-Workflow Composition[J]. Journal of Intelligent Information Systems,2003, 21(3):191-225.
[9] Tversky A. Features of Similarity[J]. Psychological Review,1977, 84(4):327.
[10] Budanitsky. Evaluating WordNet-based measures of semantic distance[J].Computational Linguistics,2006,32(1):13-47.
[11] Sedgewick R. Algorithms in Java(Third Edition),Part 5:Graph Algorithms. Boston[M]. USA: Addison Wesley/Pearson,2003.
[12] Zhou Jiantao. Cloud QoS Mapping Model and Its Service-Oriented Selection Algorithm [J]. Computer and Digital Engineering, 2017,02:373-381.
[13] Wu Shengli. Evaluation of Comprehensive Characteristics of Search Engine Indexes [J].Journal of Jiangsu University, 2015,02: 181-186.