

A New Algorithm for Semantic Web Service Matching

Bo Jiang

School of Computer and Information Engineering, Zhejiang Gongshang University,
Hangzhou, Zhejiang, China
Email: nancybjiang@mail.zjgsu.edu.cn

Zhiyuan Luo

School of Computer and Information Engineering, Zhejiang Gongshang University,
Hangzhou, Zhejiang, China
Email: miluo003@163.com

Abstract—Considering the existing problems in Web service such as lack of semantic information and interoperability, atomic services can not cooperate with each other and be combined into composite services. In this paper, based on extended UDDI, we present a semantic annotation framework for Web service description and composition. The novel service matching algorithm not only takes into account the height factor of ontology tree and local density factor for the impact of semantic distance, but also the degree of semantic overlap. Experimental results show that the proposed service matching algorithm improved the efficiency of Web service discovery, and thus Web service can be composed easily to form more complicated inter-related services.

Index Terms—ontology, interoperability, semantic annotation, Web service matching

I. INTRODUCTION

Nowadays, academic as well as industrial communities focuses part of their research and development activities on technologies like data exchange, Web service discovery and composition, security, performance evaluation and so on.

The fundamental architecture of Web service implements SOA (Service Oriented Architecture)[1] which enables applications to be integrated on different network platform. SOA guides the creation of collaborative services that are loosely coupled and independent of their implementation technologies and enables a variety of services to interact with each other. Web service is the most suitable technical solution to implement SOA [2]. Based on Web service, data and information on the Web can be interacted with each other and be integrated effectively, which solves the problems brought out by heterogeneous information systems. A set of technical specifications supports Web service, such as Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web service Description Language (WSDL), Universal Description, and Discovery and Integration (UDDI).

With the increasing number of Web service on the network, the way to find users' needs from the mass of service becomes more and more critical. Matching effect of Web service is directly related to the quality of the services that users invoked. Therefore it affects the implementation of Web service composition and the effect. As it is known, the recall and precision rates of UDDI and WSDL-based service matching are relatively low [3]. Combining ontology with Web service, semantics enables automatic service matching.

The purpose of this work is to present, a semantic annotation framework for describing Web service and extending UDDI to support semantic Web service. Furthermore, an improved algorithm based on semantic similarity computation improves the efficiency of service discovery, which sets up a solid foundation for service composition.

The rest of this paper is organized as follows. Section 2 is mainly related to the work concerning semantic annotation. Section 3 is devoted to the related work concerning semantic annotation and extended UDDI. Section 4 focuses on improved service matching algorithm. Section 5 presents experimental validation of the proposed algorithm and contrast to other algorithms. Section 6 concludes the paper and suggests some future works.

II. RELATIVE WORK

A. Definition of Terms

There are plenty of relative terms and concepts about semantic Web service; definitions [4] of related concepts are as follows:

- **Ontology:** A philosophical term which means: "the knowledge of what is to be in oneself". Ontology used in data processing indicates a structured set of knowledge in a domain. Ontology is an explicit share specification of the various conceptualizations in a particular domain.

- **Semantic annotation:** An annotation is a link to its semantic description assigned to an entity in the text. A semantic annotation is referent to a relative ontology.
- **Reasoner:** A mapping engine is a reasoner to match service advertisements with requests. The reasoner provides a semantic algorithm to match inputs and outputs of Web service during the matchmaking process.
- **Matching:** The matching operates two concepts according to some similarity features of Web service. There are some functional properties of services such as inputs, output, precondition and effects (IOPE). The matching of relevant concepts for Web service was introduced by Paolucci [5].
- **Similarity measure:** The reasoner defines four levels of similarity between two concepts A and B.

Equivalence: A and B are seen as equivalent and they represent the same concept.

Subsumption: Concept A is more general than concept B.

Opposite subsumption (plugin): Concept A is subsumed by the concept B, which means that concept B is more general than concept A.

Difference (fail): Concept A and concept B are different.

B. Semantic Description Languages of Web Service

Numerous semantic Web service frameworks are proposed and promoted for standardization by W3C. The most prominent ones are OWL-S (Ontology Web Language for Services), WSMO (Web service Modeling Language), WSDL-S (Web service Description Language Semantic) and SAWSDL (Semantic Annotation for Web service Description Language), which evolved from the WSDL-S specification. All these works were studied according to the following criterion:

- **Resource:** Semantic description (XML schema, WSDL, UDDI, and etc).
- **Property:** It means what described semantically in the document, such as inputs and outputs.
- **Language:** It describes the representation language of the semantic model (WSDL, OWL).
- **Annotation:** It specifies if the annotations are independent or are saved in other documents.
- **Model:** It specifies if the model of semantic domain is internal or external.
- **Matching:** It indicates the type of the matching algorithm and its properties.

III. SEMANTIC ANNOTATION FRAMEWORK AND EXTENDED UDDI

There are two ways to use ontologies for Web service description. The first way is adding semantic information

directly in the existing Web service standards, and using domain ontology to annotate directly on the WSDL file directly. The second way is to define ontology of Web service description, such as WSMO or OWL-S [6]. Web service description augments the concept annotation of domain-specific ontology. [7] builds a domain ontology to describe service functional information and provide consistent description. In addition, a matching algorithm was given for the requester and provider, but the paper didn't show how to realize the ontology mapping. [8] used domain ontology to annotate WSDL. [9] was based on the latest SAWSDL specifications of W3C. Automatically matching algorithm took into account services function and interface parameters. However, related concepts could not be expressed effectively based on the domain ontology.

A. Semantic Annotation Framework

In this paper, we proposed a semi-automatic semantic annotation framework based on the existing semantic annotation. On one hand, the framework inherits Paolucci's approach [3]. The WSDL document is automatically converted to DAML-S according to the mapping relationship. On the other hand, the framework introduces the OWL ontology so as to add semantic information to services. WSDL document does not include the yellow pages and white pages information of Web service. Therefore, the framework utilizes editing tools to add involved description information that further improves the semantic description of Web service. In this paper, the semantic annotation framework is shown in Fig. 1.

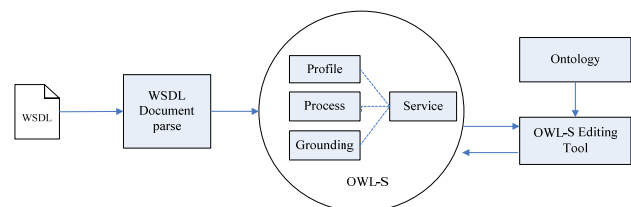


Figure 1. Semantic annotation framework.

- (1) Firstly, we use WSDL Document parser tool to parse WSDL Document. In this paper, we parse WSDL document with WSDL2OWL-S tool. But the parse results only translate WSDL information; this method can't provide semantic support of WSDL document XSD vocabulary. There are four OWL-S file generated after WSDL document has been parsed.
- (2) The XML vocabulary in WSDL document with related concepts in the domain ontology corresponding up through the human-computer interaction, it is also semantic annotation of WSDL document in an XML vocabulary. This step is based on ontology; we set up corresponding ontology before annotation of WSDL document.
- (3) The WSDL document itself doesn't contain description of yellow page and white page Web service information, TextDescription and other

information also need to be added to the Profile file after WSDL has been parsed.

After the above three steps, we also need Profile file, Process file, Grounding file and Service file combined into a complete OWL-S service description file.

B. Extended UDDI

It is known to all that pure WSDL doesn't have semantic description capabilities, which leads to low precision for Web service matching results. OWL-S provides machine-understandable semantic information for Web service, and can effectively improve the performance of the WSDL description. Therefore, it is necessary to extend UDDI so as to support OWL-S specification.

The semantic extended UDDI achieves two main functions. (1) Conversion mechanism from OWL-S to UDDI is established by referencing tModel to store semantic information of Web service; and (2) Clients (requesters) can query Web service referencing tModel type from the service library by invoking the service interface of UDDI API and obtaining the URL of the service description. Web service matching performance can be greatly improved by using extended UDDI.

In order to make the UDDI registry to support OWL-S specification and store semantic information of Web service, it is necessary to establish mapping from the OWL-S to the UDDI, information in the OWL-S Profile which is embedded to UDDI. In this Paper, we refer to [11] which proposed DAML-S Profile to UDDI mapping mechanism, it implements OWL-S to UDDI mapping by extending the tModel type. The main idea of this method is described as follows: if the UDDI data elements corresponded to elements of OWL-S Profile, then it can be mapped directly; if no corresponding data elements in UDDI, and then it need create a new tModel type, so both of them produce a mapping relation.

IV. ALGORITHM OF SERVICE MATCHING AND MATCHING FRAMEWORK

A. Algorithm of Classic Matching

Based on domain ontology of semantic Web service matching, researchers have proposed various calculation approaches. In [3], Matching results were divided into four categories according to the degree of service matching. The categories are 'exact', 'plugin', 'subsume' and 'fail'. It is shown in Fig. 2.

```

DegreeOfMatch (outB, outA):
if outB =outA then return exact
if outB subclassOf outA then return exact
if outA subsumes outB then return plugin
if outB subsumes outA then return subsumes
otherwise fail

```

Figure 2. Degree of service matching.

In the above approach, outB corresponds to one output of the request and outA corresponds to one output of the advertisement. If outB=outA then outB and outA are equivalent, which we label as *exact*. The second clause is that outB subclassOf outA, and then the result is still *exact*. If outA subsumes outB then outA is a set that includes outB, the result is *plugin*. If outB subsumes outA, then the provider does not completely fulfill the request, and the result is *subsumes*. Failure occurs when there is no subsumption relation between advertisement and request which is identified.

B. Algorithm of Semantic Matching

By introducing Web service ontology, Web service matching is converted into the calculation of concept similarity in the domain ontology library. This paper presents an improved ontology-based semantic similarity algorithm technology, which also improves the semantic matching algorithm. The basic idea of the algorithm is building domain ontology as reference. Based on domain-specific ontology, we refer the depth factor and the local density factor when we calculate semantic similarity based on the semantic distance, and we also introduce degree of semantic overlap to calculate the semantic similarity of any two concepts.

In this paper, we mainly consider the input parameters and output parameters of Web service in the semantic Web service matching. So we define an abstract Web service semantic description model as follows:

Definition 1: We define Web service semantic description model as $W=\langle I, O \rangle$. $I=\langle I_1, I_2, \dots, I_n \rangle$ is a concept vector, which represents semantic description of n input parameters in W . $I_1, I_2, \dots, I_n$ represent their corresponding semantic concepts in domain ontology basis of n input parameters. $O=\langle O_1, O_2, \dots, O_m \rangle$ is a concept vector, representing semantic description of m output parameters in W . $O_1, O_2, \dots, O_m$ represent their corresponding semantic concepts in the domain ontology basis of m output parameters.

So we can change Web service matching into the matching between requestor's service semantic description model $W=\langle I, O \rangle$ and Web service semantic description model in $W'=\langle I', O' \rangle$. Furthermore it can convert into the match between concept vector I and vector I' , vector O and vector O' in the unified domain ontology.

In the network environment, providers and requestors of the Web service often have different understandings and needs of the same type of service, which reflected in the Web service is that requested services and Services provided may be inconsistent at the input and output of the dimensions and order. It is the reason why semantic descriptions of Web service are independent with each other. So the concept vectors I, I', O and O' may have different situations in aspects of the dimension and the elements order. We must define the match of any two concept vectors in the same domain ontology before defining the concept of Web service matching formally.

Definition 2: We define the calculation function of similarity of any two concepts in the same domain ontology as $\text{Sim}(C_1, C_2)$. C_1, C_2 are any two concepts in

the domain ontology, and the value of $\text{Sim}(C_1, C_2)$ is between 0 and 1(including 0 and 1). The bigger the value is, the more similar two concepts are.

Based on definition 2, we can define the same similarity of any two concepts vectors A, B in the same domain ontology.

Definition 3: Assume that $A = (A_1 A_2 \dots A_n)$, $B = (B_1 B_2 \dots B_m)$ are any two concept vectors in the same domain ontology, and the correlation matrix is as follows.

$$S_{AB} = \begin{pmatrix} \text{Sim}(A_1, B_1) & \text{Sim}(A_1, B_2) & \dots & \text{Sim}(A_1, B_m) \\ \text{Sim}(A_2, B_1) & \text{Sim}(A_2, B_2) & \dots & \text{Sim}(A_2, B_m) \\ \vdots & \vdots & & \vdots \\ \text{Sim}(A_n, B_1) & \text{Sim}(A_n, B_2) & \dots & \text{Sim}(A_n, B_m) \end{pmatrix}$$

So that we can get the formula of any two concept vectors in the same domain ontology:

$$S(A, B) = \frac{1}{n} \sum_{i=1}^n (\max(\text{Sim}(A_i, B_j))), \quad j \in [1, m] \quad (1)$$

Semantic Web service matching can be abstracted as matching of semantic Web service description model, and matching of the semantic description model is based on the vector concept. Vector concept matching translates into matching of vector concept elements by definition 3. Matching degree of them can be calculated with similarity calculation.

In the information query field, it is always the case that calculating the similarity between the two concepts is related to calculating the semantic distance of two concepts. So the larger semantic distance between two concepts, the lower the similarity; the smaller the semantic distance between concepts, the higher their similarity.

Semantic distance is the shortest length of all relation chain between two concepts in the same domain ontology library. $\text{Dis}(C_1, C_2)$ denotes the shortest distance between concept C_1 and concept C_2 . Semantic distance is the most important factor to determine the semantic similarity.

Definition 4: the semantic similarity distance formula is as follows:

$$\text{Sim}_{\text{Dis}(C_1, C_2)} = \frac{1}{1 + \text{Dis}(C_1, C_2)} \quad (2)$$

$$\text{Dis}(C_1, C_2) = \sum_{i=1}^n \text{Weight}_i \quad \text{Concept } C_1 \text{ and concept } C_2$$

are arbitrary two concepts of the domain ontology library, Weight_i is the weight of the shortest path from C_1 to C_2 .

Definition 5: Formula for calculating the node weights is as follows:

$$\text{Weight}(C) = \text{Weight}(\text{depth}(C)) * \text{Weight}(\text{density}) \quad (3)$$

$\text{Weight}(C)$ signifies the concept C corresponding to the weight of the height factor. $\text{Weight}(\text{density})$ is local closed weight factor of node C . $\text{Weight}(\text{depth}(C)) = 1/(2 * \text{Dep}(C))$. $\text{Dep}(C)$ is the height of node C .

Definition 6: The similarity formula for the degree of semantic overlap is as follows:

$$\text{Sim}_{\text{Coin}(C_1, C_2)} = \frac{P(C_1) \cap P(C_2)}{\text{Max}(P(C_1), P(C_2))} \quad (4)$$

$P(C_1)$ represents the amount of parent nodes in the concept C_1 . It is the amount of nodes from the concept C_1 dating back to the root node. $P(C_1) \cap P(C_2)$ represents the amount of parent nodes whose concepts is shared by C_1 and C_2 . $\text{Max}(P(C_1), P(C_2))$ represents the bigger amount of parent nodes between the concept C_1 and concept C_2 .

Based on the above analysis and definitions, we can draw an integrated semantic similarity calculation.

Definition 7: Comprehensive formula for calculating the semantic similarity is as follows:

$$\text{Sim}(C_1, C_2) = \alpha * \text{Sim}_{\text{Dis}(C_1, C_2)} + (1 - \alpha) * \text{Sim}_{\text{Coin}(C_1, C_2)} \quad (5)$$

α can be adjusted according to different applications and requirements as a regulatory factor.

Definition 8: The formula for calculating the similarity of input and output is as follows:

$$\text{Sw}_{w'} = \gamma * S(O, O') + (1 - \gamma) * S(I, I') \quad (6)$$

$\langle I, O \rangle$ and $\langle I', O' \rangle$ are the input and output of Web service W and W' . $0.5 < \gamma < 1$. Output is more important than input for users and users have a greater control over in the output. Therefore, we increase the weight of output when calculating the similarity.

C. Web Service Matching Framework

In this section, a Web service matching framework has been proposed, as shown in Fig .3.

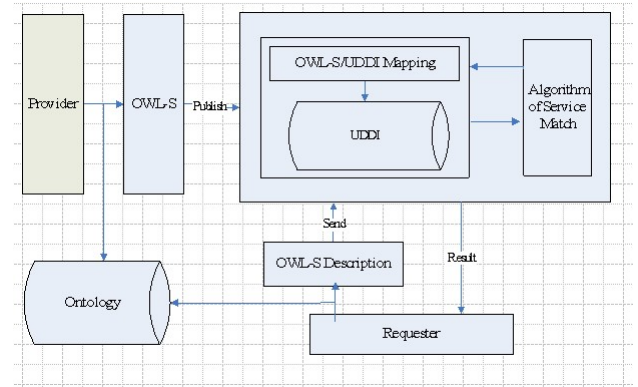


Figure 3. Web service matching framework.

In this framework, an agency provides server that deployment extended UDDI, provider/requester could publish or search base on semantic Web service by it. We describe the process of release and find service as follows: provider describes Web service with OWL-S according to domain ontology library, and then publish to server by published API; requester describes service request with OWL-S according to domain ontology, and send to server by query API; server deals with request, service request and registered services match by our matching algorithm in the UDDI, finally, server returns appropriate Web service to requester.

V. EXPERIMENT

A. Constructieon of Domain Ontology and Datasets

Experimental IDE: Eclipse 3.52, the third-party tools used: Protégé and Protégé-OWL.

We refer the relevant domain ontology, and then utilize Protégé to build a relatively perfect tourist domain ontology, as shown in Fig 4. A great many experimental data are required to support ours semantic Web service matching algorithm in order to compare to other algorithms.



Figure 4. Tourism domain ontology.

It is very difficult to do experiments to test semantic Web service which are described with WSDL. On one hand, semantic Web service applications are still in the primary stage and semantic annotation Web service are rare. On the other hand, it will cost too much time and effort to convert Web service into effective semantic Web service. In this paper, we explore semantic Web service matching and use the program which can automatically create random pair concepts to directly generate a large number of Web service description languages as the experimental test datasets.

We program to generate 950 Web service semantic descriptions and annotate 50 services manually based on our semantic framework. There are 1000 released semantic Web service description, and we manually describe 10 service requests as well. We match published semantic Web service description with Web service

matching algorithm for each service request, and matching results are sorted according to similarity values.

B. Experimental Results and Analysis

In this experiment, there are three kinds of semantic similarity calculation methods based on semantic distance. Because our algorithm introduces the degree of semantic overlap, we did two sets of experiments to prove the effect of the semantic overlap degree.

(1) Semantic similarity algorithm based on distance, $Weight_i = 1$, we mark it as algorithm I. (2) GCSM semantic distance calculation algorithm [10] is marked as algorithm II in Fig. 5. (3) The proposed improved algorithm we proposed is marked as algorithm III. In this experiment, $\gamma = 0.7$, the output parameter of semantic similarity weight is 0.7, the input parameter similarity weight is 0.3.

We use three semantic matching algorithms respectively to calculate the semantic similarity between the services in set and the query target service. The results are shown in Fig. 5.

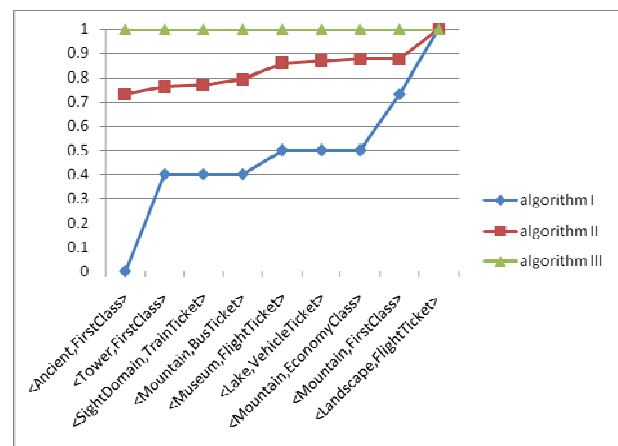


Figure 5. Three algorithms comparison.

Another set of experiments is to consider the degree of semantic overlap, $\alpha = 0.7$, and semantic overlap degree weight is 0.3. The results are shown in Fig .6.

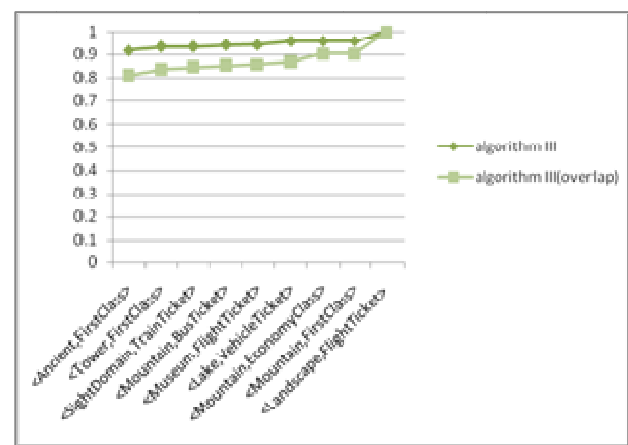


Figure 6. Algorithms comparison.

There are 10 semantic service requests matched in test set, Fig. 7 shows average precision and average recall ratio of relationship in the algorithm I, II and III

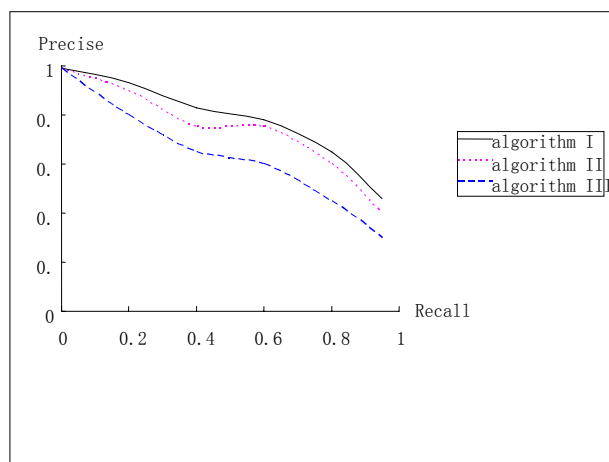


Figure 7. Relation between precise and recall.

As shown in Fig. 7, precision rate and recall rate of algorithm I was significantly lower than the algorithm II and algorithm III, while algorithm III slightly higher than algorithm II.

VI. CONCLUSION AND THE FUTURE WORK

This paper presented an available approach to crack the Web service matching problem. It presented a new annotation framework for implementing the transformation of WSDL documents to OWL-S service descriptions and extended query interfaces utilizing UDDI. The paper defined a semantic description model and proposed an improved algorithm to obtain good query results, the algorithm likes general algorithm considering the height factor of ontology tree and local density factor, but more than that, we also introduce the degree of semantic overlap. The results of experiments showed the feasibility and effectiveness of the proposed algorithm.

With the rapid development of the technology of Web service, we will confront more and more new problems and numerous challenges. Further research about other approaches to solve the issue is necessary. We promote the technological development in the process of solving problems about engineering techniques. But there are some problems to solve. For example in this algorithm, there are some Web service matching factors that we ignore, such as precondition and effects of Web service operations, along with Quality of Service (QOS). So we need a more comprehensive model of semantic Web service description and further improve the proposed algorithm in order to further enhance the semantic Web service matching capability.

ACKNOWLEDGMENT

We would like to thank all those who helped in the preparation of this paper. In particular, I am grateful to Weifeng Pan Ph. D and Tao He for their constructive suggestions.

REFERENCES

- [1] Nicolai Josuttis, SOA in Practice: The Art of Distributed System Design, O'Reilly, in Press.
- [2] Qusay H. Mahmoud. Service-Oriented Architecture (SOA) and Web service: The Road to Enterprise Application Integration (EAI). Available at <http://www.oracle.com/technetwork/articles/javase/soa-142870.html>.
- [3] Berners-Lee T. Hendler J, Lassila O. The Semantic Web [J]. Scientific American, 2001, 284(5), pp. 34-43.
- [4] Talantikite, H. N, Aissani, D, Boudjlida, and N. Semantic annotations for Web service discovery and composition. Computer Standards & Interfaces, 2009, 31(6), pp.1108-1117.
- [5] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Semantic Matching of Web service Capabilities. In Proceedings of the First International Semantic Web Conference, 2002, pp. 333-347.
- [6] Martin D, Burstein M, McDermott D, et al. Bringing Semantics to Web service with OWL-S [J]. World Wide Web, 2007, 10 (3), pp. 243-27787.
- [7] Lei Ye, Bin Zhang. Discovering Web service Based on Functional Semantics. IEEE Computer Society, 2008, pp. 348-355.
- [8] Verma K, Sivashanmugam K, Sheth A, Patil A, et al. METEOR-S WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web service. Journal of Information Technology and Management, 2005, 6(1), pp. 17-39.
- [9] SONG Ting-xin, TIAN Pei-jun, LIU Yao-he, et al. Web service's semantic annotation and auto-matching based on SAWSDL. IEEE Computer Society, 2008, pp.577-580.
- [10] Web service discovery based on Similarity calculation of the comprehensive ontology, Computer Application and Software, 2011, 8(2), pp. 87-89.
- [11] M. Paolucci, T. Kawamura, T. R. Payne and K. Sycara. Import the Semantic Web in UDDI. In Proceedings of E-Services and the Semantic Web, 2002, pp.225-236.

Bo Jiang was born in 1970. She received the PhD degree in Computer Science from Zhejiang University, China, in 2007.

She is a professor in College of Computer and Information Engineering, Zhejiang Gongshang University, P.R.China. Her current research interests include CSCW, artificial intelligence and ubiquitous computing.

Bo Jiang is a member of ACM and IEEE

Zhiyuan Luo was born in Anhui of China in 1th July 1986. He is major in artificial intelligence technology and application, and is a graduate student in the school of Computer and Information Engineering, Zhejiang Gongshang University. His current main research interests include ontology engineering and web intelligence.