A Behavior-Aware Matchmaking Model for Semantic Web Services Discovery

Jun-cheng Yang
School of Software
Shanghai Jiao Tong University, SJTU
Shanghai, P.R.China
deliay2006@gmail.com

Hao-peng Chen
School of Software
Shanghai Jiao Tong University, SJTU
Shanghai, P.R.China
chen-hp@sjtu.edu.cn

***Abstract*- As semantic Web services (SWS) rapidly becoming a vigorous technology research area, several large research initiatives have been produced. Meanwhile, a great deal of innovative work on semantic Web services discovery (SWSD) has been done, however, it is critical to meet the requirement of real-world applications. Hence, researchers are still seeking for better correctness and higher efficiency of semantic matchmaking approaches. The traditional work mismatched between requests and advertisements, who have the same concept type of inputs/outputs and different service behaviors. In this paper, we propose a behavior-aware matchmaking model for SWSD, based on the behavior-aware SWS description model, which can be easily presented by extended OWL-S profile. To improve the precision of discovery, we focus on how to describe the relationship between inputs and outputs of a service, and compute the similarity of the relationship as well.**

Keywords-semantic Web services discovery (SWSD); behavior-aware; matchmaking

I. Introduction

As its name implies, semantic Web services (SWS) stands at the intersection of two important trends in the World Wide Web's development [1]. The first trend is the wildly development of Web service technologies, among which SOA is certainly becoming the most outstanding one. In the shorter term, SOA appropriately caters for the driving objective behind Web services, which is to realize reliable, vendor-neutral software interoperability across platforms, net works, and organizations. At the same time, we should pay attention to the maturation of widely recognized Web service standards such as UDDI, WSDL, and BPEL.

The second trend —the Semantic Web— readily absorbed researchers by bringing knowledge-representation languages and ontologies, moreover, providing infrastructure for powerful approaches to describe, discovery and invoke activities on the Web.

A 2001 article by Sheila A.McIlraith et al, perhaps the first indicate the importance and potential of bringing Semantic Web technologies to Web services. Since then, SWS emerged as a distinct research field, and a large number of initiatives began not long thereafter, including OWL-S [4], WSMO [17], SWSF [18], and WSDL-S.

Semantic Web services discovery (SWSD), as defined by the Semantic Web Services Initiative Architecture (SWSA) committee, is the process of a service requestor identifies candidate services to achieve its objectives [2]. It includes three types of stakeholders: service providers (advertisers), service requestors, and matchmakers. More precisely, matchmakers accept descriptions of candidate services of providers and match them with requirements of requestors. In essence, the accuracy and efficiency of matchmaking algorithm mainly decides if the SWSD approach is powerful enough.

To the best of our knowledge, current SWSD approaches, with different semantic matchmaking models, are not sufficient to semantically identify services with the same inputs/outputs types but with different behaviors, because they are not able to capture the semantic relationship between inputs and outputs of a service. For instance, as highlighted in [3], traditional SWSD approaches cannot distinguish between two services, who have the same input (has type of geographic region) and output (has type of wine), one reporting wines produced in a region while the other reporting wines sold in a region.

To overcome this kind of weakness, this paper proposes a novel behavior-aware semantic matchmaking model, which takes former semantic service description model as foundation, and extended it with detailed description of relationship between inputs and outputs for identifying different behaviors of services. Making full use of the novel behavioral enhanced service description model, we can facilitate matchmaking systems by adding behavioral description features to the existing service description model.

The remainder of the paper is structured as follows. Section II briefly summaries the related works; Section III gives a motivating example and our behavioral-aware SWS description model; Section IV describes our matchmaking algorithm in detail; Section V shows a case study to process our algorithm over practical examples; and conclusion in Section VI.

II. Related Work

Currently, algorithms for Web services discovery in real-world registries like UDDI are based on a search by key words or syntactic specifications only. Aware of the conspicuous deficiency of key-words search, researchers are attempting to find solutions based on semantic description of services. OWL-S [4], based on the Semantic Web ontology language OWL, formerly DAML-S, defines an ontology for enforcing Web services with semantic, and is aiming at implement automation of Web service discovery, composition, and

This paper is supported by the National High-Tech Research Development Program of China (863 program) under Grant No. 2007AA01Z139)

execution by providing proper semantic descriptions for services.

In general, matchmaking roughly could be divided into three categories: syntactic matchmaking, semantic matchmaking, and the combination of two. The most influencing semantic matchmaking we are aware of is the Paolucci et al. algorithm [5], which has been cited extensively in subsequent proposals ([6 - 11]). In [5], Paolucci et al. proposed an ontology-based solution, which matching Inputs/Outputs of Services by compare them according to the hierarchical concept subsumption relationships defined in an ontology tree. There are four semantic similarity grades: Exact, Subsumes, PlugIn, and Fail.

Some researchers argue that the Paolucci algorithm suffers from some shortcomings and propose some improved version of it. In [6], Peng and Shi replace the match grades with fine values denoted by real number, to further rank advertisements which have been matched in the same grade. In [7], Bellur and Kulkarni propose a more exhaustive algorithm, borrowing ideas from finding complete matching of bipartite graph, trying to solve the problem of multi-inputs/outputs pairing. In [8], Bener et al. announce how to perform semantic matching of input and output, as well as precondition and effect. They also provide ordered ranking based solution for peculiar needs of PE-matchmaking on OWL-S documents written in SWRL (Semantic Web Rule Language). However, they don't provide predicate similarity matching, while only consider if conditions have the same predicate.

In [9], Klusch and Fries declare the advantages of combine logic-based reasoning with content-based information retrieval, and recommend a hybrid matching algorithm with seven similarity grades. In [10], Liu and Shen also state that the service matching and discovery is analogous to information-retrieval and component retrieval. The solution supports resolution among potentially useful services with seemingly irrelevant semantic similarity. In [11], Liu et al. achieve a fusion with five grades of matching, in precise, a collaboration of syntactic and semantic matching, as well as considering Qos and other dependency features.

Majority of the matchmaking work above ([5 - 11]) mainly analyze IOPE (Inputs, Outputs, Preconditions, and Effects) features based on the subsumption reasoning on taxonomies of concepts. However, behavior incompatible may exist between services, even when they have exact match in IOPEs. It is strongly recommended in [12] that a better behavioral description is vital for SWSs interaction.

III. BEHAVIOR-AWARE SWS DESCRIPTION MODEL

In the perspective of behavior, services could be grouped into two kinds: data providing services (DP services), and effect providing services (EP services). Because of that DP services only return data but EP services have effects may change the state of the world. As shown in Table1, we list a motivating example of services and a request, represented by their inputs, outputs and behavior description. According to their behavior, we see that R1, S1, S3 and S4 are DP services while S2 is an EP service. This paper discusses the description model for both these kinds of services, and more complex ones can be easily denoted by decomposing them into single kind.

TABLE I. MOTIVATING EXAMPLE OF SERVICES AND REQUEST

ID	Input	Output	Behavior Description
R1	owl: region	owl: wine	Reports wines produced in a region
S1	owl: region	owl: wine	Reports wines sold in a region
S2	owl: creditCard	owl: notification	Charges a credit card
S3	owl: patient owl: physician	owl: drug	Reports drugs that the physician prescribed for the patient
S4	owl: region	owl: wine	Reports wines operated in a region

We defined the behavior of a service as the semantic relationship between its input and output sets. When the relationship is direct, like S3 in Table1, we use horn clauses like "Physician {prescribe} Drug". Otherwise, when the relationship is indirect, like S1 in Table1, the actions are taken by some operators (the service providers or some agents with sufficient permission). Service returns the Output because of the operator take some actions on the Input. With little influence on the accuracy of behavioral description, we could omit the operator part, and represent the relationship by the action taken by the operator, as "Region {produce} Wine" in S1, and "Notification {charge} CreditCard" in S2.

Similar with properties in ontology, predicates like {produce} and {charge} in horn clauses, represent the characteristics of behaviors. In this sense, we give the formal definition of the behavior-aware SWS description model as (1).

$$S = \{ I^C, O^C, \Phi(I^C, O^C, P^P), Ct \} \tag{1}$$

Where $I^C = \{I_1, \dots, I_n\}$ represent a set of inputs with types of concepts; $O^C = \{O_1, \dots, O_m\}$ represents a set of outputs with types of concepts; $\Phi(I^C, O^C, P^P)$ is the semantic relationship holding between I^C and O^C variables, and is represented in the form of OWL triples, like horn clauses talked above; $P^P = \{P_1, \dots, P_m\}$ is a set of ontology properties represent predicates relating I^C and O^C; Ct is the constraints set imposed on I^C and O^C separately.

To specify the description model, we give a more complicated example. As shown in Fig.1 (part (b)), S3 has inputs: patient, physician and output: drug, all denoted by OWL concepts. Lines represent the relationship between inputs and outputs, indicating that the behavior is to report drugs that the physician prescribed for the patient. The literal notation of S3 in our description model is given in Fig.1 (part (a)). Note that this kind of semantic description can be annotated easily in extended OWL-S Profile (or other semantic Web service description languages), thus it fits well with semantic Web service standard stack.

The above model is adapted from RDF parameterized view and SPARQL, which allows catching the semantic relationships between inputs and outputs of a service, by supported with ontology concepts and properties. Further detailed discussion of SPARQL is beyond the scope of this paper, and the interested researchers are referred to [13] for further reading.

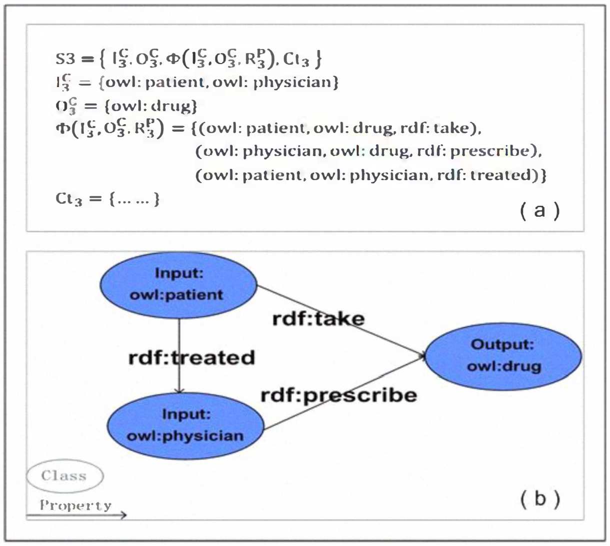

Figure 1. (a): Literal notation of S3; (b):Graphical representation of S3.

IV. Behavior-Aware SWS Matchmaking Model

Based on the previous service description model, in this section, we propose our behavior-aware SWS matchmaking model in detail.

A. Concept Similarity

The algorithm for matching any parts included in a service functional description model, computes semantic similarity of two entities (concepts, predicates, or conditions). For the sake of behavior-aware, in this paper, we focus on concept similarity and predicate similarity. Furthermore, condition similarity could be gained by merging the concept similarity and property similarity.

With the development of SWS technologies, a host of semantic distance computing algorithms appear. Semantic distance is inversely proportional to the corresponding semantic similarity between two concepts. Generally, we classify three kinds of semantic distance calculation model [8]: hierarchical distance model, information theoretic model, and the combination of two.

In hierarchical distance models, like [5 - 8] and [11], matchmakers compare two concepts based on their distance in a taxonomy tree that contains the subsumption relationships between all the concepts of the ontology, and return the grade of similarity or a number between 0 and 1 denoting the grade of similarity.

In information theoretic models, like [9] and [10], researchers take advantages of more mature information retrieval technologies, and combine useful words similarity metrics with concept similarity computation. Specially, semantic distance calculates the information content of concepts and attaches the value to the semantic subsumption relationship between two concepts. Most of these approaches were applied WordNet [14]. WordNet is a lexical database for general English developed at Princeton University. In WordNet, nouns, verbs, adjectives and adverbs are organized into synonym sets (synsets), which are interlinked by conceptual-semantic and lexical relations. For more details on WordNet, see [14].

In Paolucci algorithm [5], the degree of match is decided by the minimal distance between concepts in a taxonomy tree as show in Fig.2 (part (a)). The degree assignment is described below, where outR represents one output of the request and outA represents one output of the advertisement:

- **Exact** If outR and outA are equivalent, namely outR=outA, we call the match degree Exact.
- **PlugIn** If outA is a set that includes outR, namely outA subsumes outR, and we call the match degree PlugIn.
- **Subsume** If outR subsumes outA, we call the match Subsume.
- **Fail** Failure occurs when no subsumption relation between outR and outA is identified.

Degrees of match are organized in a discrete scale. Exact matches are of course perfect; PlugIn matches are the next best level, because the advertised output can probably be used by the requester. Subsume matches is the third best level, since the requirements are only partially satisfied; Fail is the lowest level and it means an unacceptable result.

For focusing on the behavior-aware issue, we reason the subsumption grades and assign a score in [0, 1] range for each grade, as in [8], which is described in (2). We extend the original concept similarity calculation method defined in Paolucci algorithm, which affected most of current matchmaking approaches. Hence, one can easily cover this part with their own concept similarity method. Given arbitrary pair of ontology concepts C_R of the request and C_A of the advertisement, the matching score between C_R and C_A, $Sim_C(C_R, C_A)$ is defined as (2).

$$Sim_C(C_R, C_A) = \begin{cases} 1, & if \;\; C_R \; and \; C_A \; are \; equivalent \\ 0.75, & if \;\; C_A \; subsumes \; C_R \\ 0.25, & if \;\; C_R \; subsumes \; C_A \\ 0, & if \;\; matching \; fails \end{cases} \quad (2)$$

B. Predicate Similarity

Similar with nouns grouped by concept class sets in ontology, we can organize verbs, more specifically predicates, into property sets in domain specified ontology. Then, we interlink property sets with their relationships as subPropertyOf. As in the running example of wine-industry domain, we built a similar taxonomy tree describing the subsumption relationships between all the properties of the ontology. A fragment of the taxonomy tree used in the running example is show in Fig.2 (part (b)).

$$Sim_P(P_R, P_A) = \begin{cases} 1, & if \;\; P_R \; and \; P_A \; are \; equivalent \\ 0.75, & if \;\; P_A \; subsumes \; P_R \\ 0.25, & if \;\; P_R \; subsumes \; P_A \\ 0, & if \;\; matching \; fails \end{cases} \quad (3)$$

Extended from concept similarity algorithm, we define the degree of matching properties as decided by the minimal

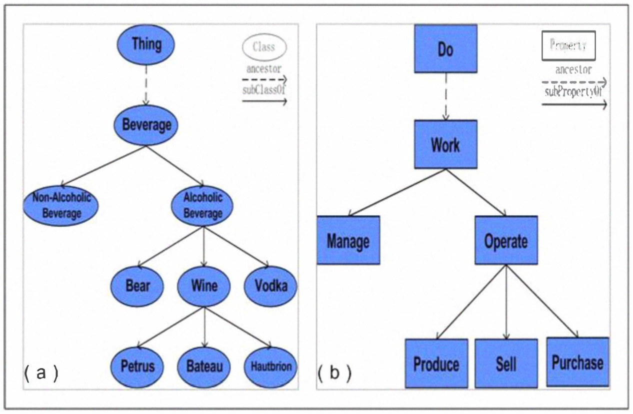

Figure 2. (a): A fragment of the wine-industry ontology used in calculating concept similarity; (b): A fragment of the wine-industry ontology used in calculating property similarity

distance between properties in a property taxonomy tree. The grade assignment is shown in (3), including Exact, PlugIn, Subsume, and Fail. Given arbitrary pair of ontology properties P_R of the request and P_A of the advertisement, the matching score between P_R and P_A, $Sim_P(P_R, P_A)$ is defined as (3).

Obviously, how to define the property taxonomy tree is essential to develop real-world applications, and is mainly determined by human's common subjective judgments. Furthermore, the majority of verbs are poly-semantic, what may leads to ambiguous understanding and multi subjective judgments. Fortunately, a great deal of work has been done in WordNet [14], and researchers can refer to WordNet for more specific support.

C. *Bipartite Matching*

When services have multi-inputs/outputs, without bipartite matching, matchmakers take the highest score pair following the order of parameters (inputs/outputs), which would cause false negative at some scenario. For instance, assuming that a request with outputs: {Entertainment, Sport}, and an advertisement with outputs: {Bowling, MovieShow}. The situation of multi-inputs is similar. First, matchmaker compare (Entertainment, Bowling) and (Entertainment, MovieShow), it turns out that they have the same score. Following the sequence of advertisement's outputs, matchmaker pair Bowling with Entertainment, and get Fail-grade of comparing (Sport, MovieShow). If we transpose the order of advertisement's outputs, the result would be non-Fail match.

We see that the result of the matchmaker depends on the order of parameters. Semantic matchmaking should ignore the syntactic ordering of parameters in Advertisements and Requests. Therefore, bipartite matching is desired. In this paper, we use bipartite matching to find the optimum parameter pairing. More specifically, first translate parameter pairing problem to bipartite matching problem, then apply a maximum complete bipartite matching algorithm to find the optimum parameter pairing.

In graph theory, a bipartite graph is an undirected graph $G = (V, E)$, in which vertices V are grouped into two sets: the left set L and the right set R. Edges E can only exist between vertices of different set, and may have weight. A matching is a sub-graph $G' = (V, E')$, where $E' \subseteq E$, and no any two edges $e_1, e_2 \subseteq E'$ share the same vertex. The matching is complete, if and only if all vertices in V are matched. The score of the matching is the sum of all edges' weights in the corresponding complete matching bipartite sub-graph.

In aspect of semantic matching, we mapping output set of request $O_R = \{O_{R_1}, \ldots, O_{R_n}\}$ as the left set L, and output set of advertisement $O_A = \{O_{A_1}, \ldots, O_{A_m}\}$ as the right set R. Assigned with two vertices' concept similarity scores as their edge's weight W_i, we can find optimum pairing through computing a maximum weight of complete matching. We use the Umesh Bellur's algorithm [7], which has appropriate mapping model and cited Hungarian algorithm [15] to calculate maximum weight complete matching. Given a request $R = \{I_R, O_R\}$ and an advertisement $A = \{I_A, O_A\}$, the final bipartite matching method is defined as (4).

$$Sim_B(R, A) =$$

$$max\left\{\sum Sim_C(O_{R_i}, O_{A_j})\right\} + max\left\{\sum Sim_C(I_{R_h}, I_{A_k})\right\} \quad (4)$$

Where O_{R_i}, O_{A_j} $\in$ {complete matching of O_R and O_A}, and I_{R_h}, I_{A_k} $\in$ {complete matching of I_R and I_A}.

D. *Matchmaking Algorithm*

In this section, we will elaborate our matchmaking algorithm in detail. As shown in Fig.3, the proposed behavior-aware SWS matchmaking algorithm begins its execution at receiving a request from customer or some outside agent.

First, the syntactic filter checks which domain the request belongs to, and then loads all service profiles within the same domain from the service repository. Then, it filters services having the same number of parameters with the request, and the numbers are denoted as N_{in} and N_{out}. Service domain specifies the category of a given service, like in real world, the category of service in the UNSPSC classification system. With explosively growth of service number, this step will obviously improve the efficiency than comparing with all services in the repository.

Afterwards, the concept similarity calculator downloads

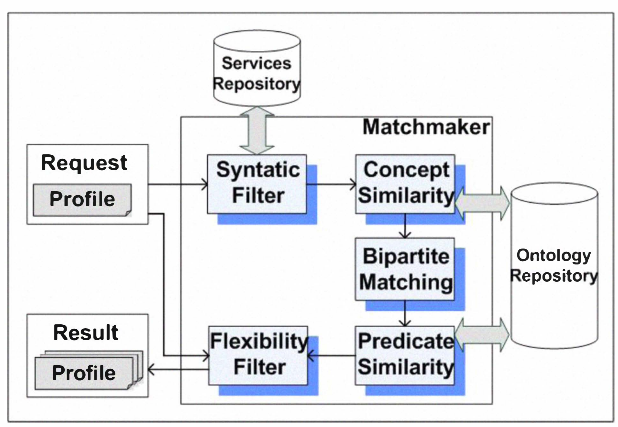

Figure 3. Behavior-aware SWS matchmaking model

the domain ontology definitions from the ontology repository. For each advertisement, according to (2), it computes concept similarity of all possible pairs of request's outputs and advertisement's outputs, then the inputs. During the bipartite matching process, matchmaker find the optimum parameter pairing, by means of calculating maximum weight complete matching of the corresponding bipartite graph. The bipartite matching score is computed by (4).

In the request, if there has a behavioral relationship between an output and corresponding input, the predicate similarity calculator search for the corresponding relationship of output and input of the advertisement, where the request' output and the advertisement's output are in optimum pairing. Finally, we use (3) to calculate the predicate matching score.

At last, given a request R and an advertisement A, described by our behavioral-aware matchmaking description model as $R = \{I_R, O_R, \Phi_R(I_R, O_R, P_R), Ct_R\}$, and $A = \{I_A, O_A, \Phi_R(I_A, O_A, P_A), Ct_A\}$, We give the definition of our behavioral-aware matchmaking algorithm as (5).

$$Sim_{Behavioral}(R, A) =$$

$$\frac{1}{4}(N_{in} + N_{out}) \cdot \sum Sim_P\left(P_{R_i}, P_{A_j}\right) + Sim_B(R, A) \qquad (5)$$

Where $P_{R_i} \in P_R, P_{A_j} \in P_A$, O_{R_i} and $O_{A_j} \in$ {complete matching of O_R and O_A }, and there exists relationships as $\Phi_i(I_{R_i}, O_{R_i}, P_{R_i}) \in \Phi(I_R, O_R, P_R)$ and $\Phi_j\left(I_{A_j}, O_{A_j}, P_{A_j}\right) \in \Phi(I_A, O_A, P_A)$.

This definition aims at sorting the advertisements, which have the same bipartite matching score, by the predicate scores. While most importantly, the one has lower bipartite score should always behind the higher one, no matter what their predicate score are. The multiply by the factor $\frac{1}{4}(N_{in} + N_{out})$, guarantees the result score would be within the range of the one-upper grade score and the one- lower grade score.

The matchmaker would return the result advertisements directly, if customer didn't decide any degree of flexibility they grant to the system. Our matchmaker performs flexible matches, and allows service requesters to decide the degree of flexibility. For instance, if they concede little flexibility, they reduce the chance of finding services that match their requirements. On the other hand, by increasing flexibility of match, it would increase the likelihood of false positives.

E. Behavior-aware SWSD architecture

In the description so far, we tacitly implied that a registry architecture in which service capabilities are advertised, requested, and matched. This is the architecture adopted by registries like UDDI, which is the most likely architecture to be adopted by Web services, although other forms of registry, like a pure P2P architecture, are also possible. Therefore, we tightly coupled our matchmaker with the UDDI business registry and adding capability port for its operations, like what did in [16]. The architecture of OWL-S/UDDI matchmaker is shown in Fig.4.

Receiving a normal advertisement/request through the publish/request port, the OWL-S/UDDI registry processes it

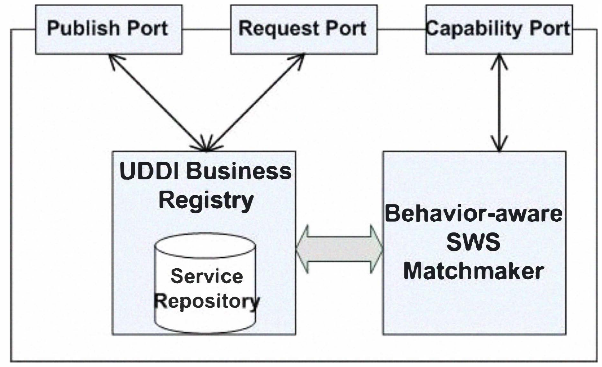

Figure 4. Architecture of OWL-S/UDDI matchmaker

like any other UDDI. The capability port handle requests, when they contain OWL-S profile information. The requests received from the capability port are processed by the behavior-aware SWS matchmaker. The Response contains a list of Service keys of the matched advertisements, sorted by the matching score. Service consumer can use this information to selecting an appropriate service and invoking it from service providers.

Practically, as the number of advertisements in the repository increases, the request response time will also increase extremely. To improve matching performance, when an advertisement is published, one can annotate all the concepts/predicates in the matchmaker with the degree of match that they have with the concepts/predicates in each published advertisement [16]. As a consequence, the time of computing concept/predicate similarity will be saved at the matching period. In addition, since the publishing of an advertisement is a one-time event, we can also process the annotation off-line, which may improve the publish performance. Note that, all the procedure above is based on the assumption that all the part nets in this architecture are trustful; and a suitable ontology has already been developed and deployed.

V. CASE STUDY

In this section we show an example of how a request service R1 is behavioral matched with an advertised service S1, which both described in the behavioral-aware SWS description model. Meanwhile, the behavioral similarity between R1 and S4 is calculated, to explain how the behavioral-aware matchmaking algorithm works for distinguishing different behaviors.

As shown in Table1, R1 is a service for reporting wines produced in a region, and S1 is a service for reporting wines sold in a region, and S4 is a service for reporting wines operated in a region, whose literal notations are shown in Fig.5.

According to the wine-industry ontology shown in Fig.2 and the concept similarity definition as (2), the matchmaker firstly calculates the degree of matching outputs and inputs as: $Sim_C(O_{R1}, O_{S1}) = 1$ and $Sim_C(I_{R1}, I_{S1}) = 1$. Then, we calculate the bipartite matching result as $Sim_B(R1, S1) = 2$, and $N_{in} = 1, N_{out} = 1$. After that, the matchmaker calculates the predicate similarity as $Sim_P(R_{R1}, R_{S1}) = 0$. Finally, according

$$R1 = \{ I_{R1}, O_{R1}, \Phi_{R1}(I_{R1}, O_{R1}, R_{R1}), Ct_{R1} \}$$
$$I_{R1} = \{\text{owl: region}\}$$
$$O_{R1} = \{\text{owl: wine}\}$$
$$\Phi_{R1}(I_{R1}, O_{R1}, R_{R1}) = \{(\text{owl: region, owl: wine, rdf: produce})\}$$
$$Ct_{R1} = \emptyset$$

$$S1 = \{ I_{S1}, O_{S1}, \Phi_{S1}(I_{S1}, O_{S1}, R_{S1}), Ct_{S1} \}$$
$$I_{S1} = \{\text{owl: region}\}$$
$$O_{S1} = \{\text{owl: wine}\}$$
$$\Phi_{S1}(I_{S1}, O_{S1}, R_{S1}) = \{(\text{owl: region, owl: wine, rdf: sell})\}$$
$$Ct_{S1} = \emptyset$$

$$S4 = \{ I_{S4}, O_{S4}, \Phi_{S4}(I_{S4}, O_{S4}, R_{S4}), Ct_{S4} \}$$
$$I_{S4} = \{\text{owl: region}\}$$
$$O_{S4} = \{\text{owl: wine}\}$$
$$\Phi_{S4}(I_{S4}, O_{S4}, R_{S4}) = \{(\text{owl: region, owl: wine, rdf: operate})\}$$
$$Ct_{S4} = \emptyset$$

Figure 5. Literal notation of R1, S1 and S4

to (5), we got the behavioral matching score as $Sim_{Behavioral}(R1, S1) = 2$. On the other hand, we achieve the $Sim_P(R_{R1}, R_{S4}) = 0.75$, and the $Sim_{Behavioral}(R1, S4) = 2.375$.

The example shows that our matchmaking model could distinguish among services with the same input and output concept type, but different behaviors. Most importantly, we didn't change the original order of bipartite matching, which means that services with higher bipartite matching score would always in front of the lower ones.

VI. CONCLUSION

In this paper, we have proposed a novel behavioral-aware matchmaking model for semantic Web services discovery. To capture behavioral features, we built our semantic Web service description model extends from SPARQL, for describing the relationships between inputs and outputs of a service. Predicates represented by the ontology properties, imply the behavioral characteristics of services. Moreover, predicates could also have subsumption relationship like concepts in a taxonomy tree. Based on a suitable developed ontology, the predicate matching scores could be assigned inspired by the four similarity grades defined in Paolucci algorithm. Multi-outputs/inputs situation would be taken into account by means of bipartite matching.

At present, few SWSD models consider the behavioral description and matching, which are strongly recommended in [12] that better behavioral description should be enhanced for semantic Web services. Furthermore, our model is flexible, since users can grant flexibility of the results. Our model is also extendable, since researchers could cover the concept similarity part with their own concept similarity calculation algorithm. At the same time, we gave an example of wine-industry domain to explain our model practically.

Possible directions for future work consider improvements of the matchmaking model by involving new approaches in the semantic Web research area. We may involve the matching algorithm by adding context information and past action information etc. Prototype would be developed to verify and improve our model.

REFERENCES

[1] D. Martin, J. Domingue, M.L. Brodie, and F. Leymann, "Semantic Web Services, Part1," IEEE Intelligent Systems, vol. 22, no.5, pp.12-17, September/October 2007.

[2] M. Burstein, C. Bussler, M. Zaremba, T. Finin, M.N. Huhns, et al., "A Semantic Web Services Architecture," IEEE Internet Computing, vol. 9, issue 5, pp.72-81, September 2005.

[3] M. Barhamgi, D. Benslimane, and A.M. Ouksel, "SWSMS: A Semantic Web Service Management System for Data Sharing in Collaborative Environments," the 3rd International Conference on Information and Communication Technologies: From Theory to Applications(ICTTA 2008), Damascus, Syria, April 2008.

[4] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, et al., "OWL-S: Semantic Markup for Web Services," http://www.w3.org/Submission/OWL-S/, November 2004.

[5] M. Paolucci, T. Kawamura, T.R. Payne, and K. Sycara, "Semantic Matching of Web Services Capabilities," The First International Semantic Web Conference on The Semantic Web(ISWC 2002), pp.333-347, 2002.

[6] H. Peng, Z. Shi, L. Chang, and W. Niu, "Improving Grade Match to Value Match for Semantic Web Service Discovery," The IEEE International Conference on Natural Computation(ICNC 2008), IEEE Computer Society, Jinan, China, pp.232-236, October 2008.

[7] U. Bellur, and R. Kulkarni, "Improved Matchmaking Algorithm for Semantic Web Services Based on Bipartite Graph Matching," The IEEE International Conference on Web Services(ICWS 2007), Salt Lake City, Utah, USA, pp.86-93, July 2007.

[8] A.B. Bener, V. Ozadali, and E.S. Ilhan, "Semantic Matchmaker with Precondition and Effect Matching Using SWRL," Expert Systems with Applications, vol. 36, issue 5, pp.9371-9377, July 2009.

[9] M. Klusch, B. Fries, and K. Sycara, "Automated Semantic Web Service Discovery with OWLS-MX," The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems(AAMAS 2006), Hakodate, Japan, pp.915-922, May 2006.

[10] M. Liu, W. Shen, Q. Hao, and J. Yan, "An Weighted Ontology-Based Semantic Similarity Algorithm for Web Service," Expert Systems with Applications, vol. 36, issue 10, pp.12480-12490, December 2009.

[11] M. Liu, Q. Gao, W. Shen, Q. Hao, and J. Yan, "A Semantic-Augmented Multi-level Matching Model of Web Services," Service Oriented Computing and Applications, vol. 3, issue 3, pp.205-215, September 2009.

[12] H. Zhou, S. Bhiri, K. Ning, L. Vasiliu, D. Foxvog, et al., "Better Behavioral Description for Dynamic Semantic Web Services Collaboration," The IEEE International Conference on Semantic, Knowledge, and Grid(SKG 2007), IEEE Computer Society, Xi'an, China, pp.338-341, October 2007.

[13] E. Prud'hommeaux, A. Seaborne, H.P. Laboratories, and Bristol, "SPARQL Query Language for RDF," http://www.w3.org/TR/rdf-sparql-query/, January 2008.

[14] G.A. Miller, "WordNet: A Lexical Database for English," Commun ACM, vol. 38, issue 11, pp.39-41, November 1995.

[15] H.W. Kuhn, "The Hungarian Method for the Assignment Problem," Naval Research Logistics, vol. 52, issue 1, pp.7-21, February 2005.

[16] N. Srinivasan, M. Paolucci, and K. Sycara, "Semantic Web Service Discovery in the OWL-S IDE," The 39th Annual Hawaii International Conference on System Sciences(HICSS 2006), IEEE Computer Society, Kauai, Hawaii, USA, pp.109b, January 2006.

[17] J. Bruijn, C. Bussler, J. Domingue, D. Fensel, M. Hepp, et al., "Web Service Modeling Ontology(WSMO)," http://www.w3.org/Submission/WSMO/, June 2005.

[18] S. Battle, A. Bernstein, H. Boley, B. Grosof, M. Gruninger, et al., "Semantic Web Services Framework (SWSF) Overview," http://www.w3.org/Submission/SWSF/, September 2005.