2011 International Conference on Asia Agriculture and Animal
IPCBEE vol.13(2011) © (2011)IACSIT Press, Singapore

A non-deterministic ontology-based method for geo web service discovery

Rezvani Nabi A. [1] and Hakimpour Farshad

1 Department of Computer Engineering and Information Technology, Amirkabir University of Technology, nrezvani@aut.ac.ir

[2] Department of Surveying and Geomatics Engineering, University of Tehran, farshad@hakimpour.com

Abstract. Automated web service discovery is one of the major challenges among recent research works. Bringing uncertainty to the process of web service discovery is also an important concern since deterministic discovery methods are sometimes too restrictive. In this paper we propose a method for web service query processing which gives discovery responses in a non-deterministic manner. We also propose a framework for handling uncertain responses which modifies the query in a way that discovering fully matching web services becomes possible. Result of applying the proposed method on a typical example shows more efficient discovery procedure.

Keywords: Web service discovery, geo web services, ontology, SDI

1. Introduction

Automated web service discovery is one of the major challenges among recent research works, due to increasing number of web services available on any service domain and especially the Internet [5, 6, 7, 8]. Besides, web service discovery can be the prerequisite step through the automated web service composition which in turn has attracted remarkable attention during recent years of research on web services and semantic web [11, 12, 13, 14]. Composing isolated web services and creating more complex services, enables more complex processing tasks as a result.

In geo-web services area, which is the specific domain that this research we will focus on, catalogue services are the UDDI repositories which service consumers query for their intended web services based on keyword string matching search mechanisms [16]. To overcome the weaknesses of this simple search scheme, variety of methods have been proposed. In some works, NLP techniques have been used to improve search results so that they would be semantically more relevant to the submitted query [11]. A few research activities have been focused on the interdependence of web service discovery and composition [14]. They do not address these two problems separately and try to solve them as a whole. In some other works, ontologies have been used to enhance query results more effectively [2, 3, 7, 8].

The uncertainty problem is one of challenges that very much impacts the process of web service discovery [4]. There are research works that have inspected uncertainty in web services and their discovery process. What these works have mostly addressed, is the uncertainty in the inputs of service discovery and not the output [9, 10]. They mostly address the ontology description and representation formalisms. In this research our aim is to investigate uncertainty problem from the query result generation point of view.

We propose an ontology based method for web service discovery which has been augmented by a non-deterministic query result generation mechanism. In our approach, web service search requests are responded in a non-deterministic fashion and any result does not imply a fully matching web service. Returned web services either fully or partially satisfy desired criteria which were intended in the query. This gives the service discovery an advantage to retrieve services which are approximately but not totally acceptable. If the requester did not receive total matching results, it may reach its intended result with some modification to the submitted query. Few research works have also focused on uncertain query result generation, but their intention is to give a quantitative score to each discovered web service, and do not try to increase the chance of finding matched services for discovery purpose [4].

The reminder of this paper is organized as follows: In section two a review of the concepts discussed in this paper will be presented. In section three we offer our conventions for the specification of the problem and the solution. Also a typical example is given which clarifies former theories and discussions. Section four elaborates our proposed method. In section five a conclusion of topics will be offered.

2. Problem definition

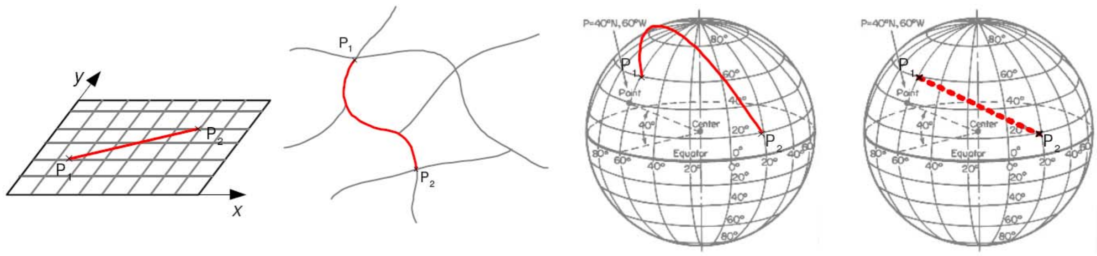
To clarify the concepts presented in this paper, a typical example of services in geospatial domain is discussed. The example is similar to the one from [5] with a minor modification and supposes a querying party which aims at composing a complex service for computing distance between airports. It has already discovered a service (**GetAirportLocation**) that provides the locations of airports and is trying to find services for computing the distance between these two locations.

There are a number of services that perform this action with the same syntax. But these services differ in the semantic of their input, output and the type of distance they compute:

- Service 1 returns the 2D Euclidian distance between two points in a plane expressed in Cartesian coordinates (Fig. 1, (a)).
- Service 2 returns the distance between two vertices in a graph (Fig. 1, (b)). The vertices are assumed to lie in a plane.
- Service 3 returns the great circle or geodesic distance between two points on a sphere expressed in geographic coordinates (Fig. 1, (c)).
- Service 4 returns the 3D Euclidian distance between two points in R3. To allow for the third dimension, this service's input points are 3D points (Fig. 1, (d)).

There are also different discovery scenarios:

- Scenario 1: The **GetAirportLocation** service returns points in DHDN / Gauss-Krüger, zone 2 (i.e. a projected coordinate reference system) and the query is looking for a service that computes distances in the associated projection plane (Fig. 1, (a)).
- Scenario 2a: The **GetAirportLocation** service returns points in WGS84 (i.e. a geographic coordinate reference system) and the query is looking for a service that computes distances on the associated sphere (Fig. 1, (c)).
- Scenario 2b: The **GetAirportLocation** service returns points in WGS84 and the query is looking for a service that computes the 3D Euclidian distance between these points (Fig. 1, (d)).
- This set of services and discovery scenarios will be used throughout this paper to illustrate the theories and discussions.



(a) (b) (c) (d)

Fig. 1: Different kinds of distance measures in various kinds of spaces from left to right respectively: Euclidian distance in the plane, shortest path in a graph, great circle distance and Euclidian distance in R3.

3. Review of The Concepts

An ontology is an explicit formal specification of a shared conceptualization [3]. By using ontologies to enrich the description of services, their semantics become machine-interpretable, and users are enabled to pose concise and expressive queries. Furthermore, logical reasoning can be used to discover implicit relationships between search terms and service descriptions. Operations of web services can be described by ontology enrichments. In this setup, inputs and outputs of services and specifications of pre and post conditions are described using First Order Logic notions [14, 15].

We address the operation description and underlying technologies in two levels: Domain level definitions build a shared vocabulary of concepts. At the application level, more specific descriptions of a particular operation are offered.

For registering a service, the publisher party needs to choose an operation from the domain ontology, which in our example it is the distance operation between two points (Fig. 2).

$$
\begin{array}{rcl}
\mathbf{sig} & : & dist(p_1 : gm\_point, p_2 : gm\_point) : distance \\
\mathbf{pre} & : & \exists s_1 : r3subspace(s_1) \wedge in(p_1, s_1) \wedge in(p_2, s_1) \\
\mathbf{post} & : & \exists c, s_2 : r3subspace(s_2) \wedge \\
 & & \quad shortestCurve(c, p_1, p_2, s_2) \wedge \\
 & & \quad dist(p_1, p_2) = length(c)
\end{array}
$$

Fig. 2: Domain ontology specification

$$
\begin{array}{rcl}
\mathbf{sig} & : & dist(p_1 : proj\_point, p_2 : proj\_point) : distance \\
\mathbf{vars} & : & p : plane \\
\mathbf{pre} & : & in(p_1, p) \wedge in(p_2, p) \\
\mathbf{post} & : & \exists c : shortestCurve(c, p_1, p_2, p) \wedge \\
 & & \quad dist(p_1, p_2) = length(c)
\end{array}
$$

Fig. 3: A typical application ontology specification

Ontology semantics are describing the domain concepts of the operation including input, output, pre and post conditions. A more detailed explanation of these semantic notions could be seen in [14]. After specifying these items on the domain ontology, the publisher then uses the application ontology to give a more specialized ontology for a specific operation. As an example, description of the 2D-Euclidian distance is shown in Figure Fig. 3.

4. Proposed Method

We use a service discovery methodology based on function subtyping, like the one which is proposed in [5]. The figure depicts the outline flow of the algorithm. The algorithm has two running steps. It first uses the semantic signature of the query to filter large number of available services. The second step compares pre and post conditions of query and filtered services to find matching results. The outline of the algorithm which we call it Lutz Method is illustrated in Fig. 4.

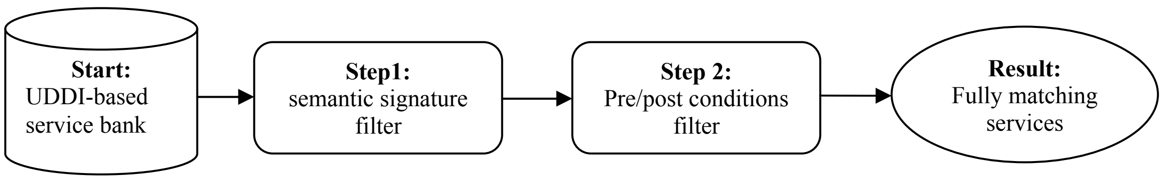


Fig. 4: Running steps of the Lutz discovery algorithm

The problem with this method of service match making is that the output is zero or a number of services based on the discovery steps. There may be cases where the service is not totally compliant with the requested query, but it is partially acceptable, based on some of matching factors considered in the algorithm. The advantage of such method is that the query returns some partially acceptable results which may be useful for the service discovery procedure, and these partially approved services may turn into totally acceptable results, if modifications are applied to the query.

We propose a method which brings some uncertainty to the way query results are generated. In our method, the querying party receives much more information about existing services that may potentially be relevant to his intended services. It is provided with a status vector for the matching factors. The vector indicates whether any of the matching factors have been satisfied or not. State vector is shown below in table Tab. 1.

Tab. 1: State Vector returned by proposed algorithm for each service

Vector cell	input	output	precondition	postcondition
Boolean value	B1	B2	B3	B4

The Boolean value in each cell of the vector shows that the corresponding factor of the service has been compliant with the query or not. The returned vector is then used in companion with a corrective ontology so that the system can modify its query to retrieve intended services. Corrective ontology uses some of the methods of the domain ontology for describing its corrections rules. The methods used in the corrective ontology are the ones that inherently perform some kind of spatial transformation. A corrective ontology in an SDI could be like the one in table Tab. 2.

Tab. 2: Conversion table for replacing items of the application ontology

	Existing item	Conversion Rule	Replaced item
Input/ Output	Geographic_point(p1)	Transform_g2p(p1,p2)	Projected_point(p2)
	Projected_point(p1)	Transform_p2g(p1,p2)	Geographic_point(p2)
	2d_point(p1)	Make_3d(p1,p2,z)	3d_point(p2)
Preconditions/ postconditions	geograhic_Curve_distance(p1,p2,c)	Convert_g_curv_to_p_eucl(c,e)	2d_Euclidian_distance(p1,p2,e)

When any cell in the state vector has the FALSE value, the querying system should look in the conversion table. The conversion table gives the possibility of replacing some elements by their counterparts and creating a patched query. The patched query involves calling more service(s) but increases the chance of finding matching service(s). The algorithm in Fig. 5 shows the running steps for the proposed match making method.

i) Based on the deterministic method, compute the state vector for each service.
ii) For each service do the following
ii-i) If the state vector contains true value for all cells, return the service as a match.
ii-ii) Otherwise, for input and output cells with value false do the following:
ii-ii-i) Look in the correction table and try to find a rule for replacing content of the item marked by the corresponding cell.
ii-iii) Do previous step for preconditions and postconditions cells.
ii-iv) If all cells with FALSE value have been replaced with values from the correction table, return the service as a match otherwise skip the service.

Fig. 5: algorithm of the proposed discovery method

As implied by the algorithm, computational order is the number of rules in the input/output section of the table, multiplied by the number of rules in the Preconditions/postconditions section. By applying the replacements according to the conversion table, it is observable that the number of matched services for different scenarios is increased.

- In scenario 1, if the Lutz algorithm is applied, service 1 will be discovered. If we use the proposed non-deterministic algorithm, input parameters could be replaced using **Transform_p2g** rule and postcondition can be replaced using the **Convert_g_curv_to_p_eucl** rule. Having applied these conversion rules, service 3 is also discovered.
- In scenario 2b, none of the services are matched according to the Lutz algorithm, but if the input parameters are replaced by **make_3d** rule, then service 4 would be discovered as a match.

5. Conclusion and Future Work

In this paper we disputed the deterministic nature of service discovery algorithms in SDIs. We then proposed a framework for returning a non-deterministic response from the discovery system. In the next step we offered a machine interpretable method for processing these responses. Using an example, we showed that by applying the proposed method, more services can be discovered and totally the chance of discovering the intended service(s) is increased. Future research could be concentrated on designing a more robust conversion table. Besides, studies on reducing the computational cost of the proposed algorithm might be useful.

6. References

[1] H. Chen. Geospatial Semantic Web. *An invited talk at UMBC. CMSC.* 2007.

[2] A. B. Teknillinen. Semantic Web Services Survey. *Technical Report*. Helsinki University of Technology, Laboratory of Software Technology. 2008.

[3] L. Ding, P. Kolari, Z. Ding, S. Avancha. Using Ontologies in the Semantic Web. *Integrated Series in Information Systems. Springer*. 2007.

[4] F. Martín-Recuerda, D. Robertson. Discovery and Uncertainty in Semantic Web Services. *Uncertainty Reasoning for the Semantic Web I*. Springer. 2008.

[5] M. Lutz. Ontology Based Service Discovery in Spatial Data Infrastructures. *2005 Workshop on Geographic Information Retrieval (GIR'05)*. ACM. 2005.

[6] G. R. Fallahi, M. S. Mesgari, A. Rajabifard. A Methodology Based on Ontology for Geo-Service Discovery. *World Applied Sciences Journal*. IDOSI. 2008, 3 (2): 300-311.

[7] U. Keller, R. Lara, A. Polleres, I. Toma, M. Kifer. WSMO Web Service Discovery. *WSML Working Draft*, DERI. 2004.

[8] M. Klusch, B. Fries, K. Sycara. Automated Semantic Web Service Discovery with OWLS-MX. *Fifth international* joint *conference on Autonomous agents and multiagent systems (AAMAS '06)*. ACM. 2006.

[9] Z. Ding, Y. Peng, R. Pan. A Bayesian Approach to Uncertainty Modeling in OWL Ontology. *2004 International Conference on Advances in Intelligent Systems - Theory and Applications (AISTA2004)*. 2004.

[10] L. Predoiu, H. Stuckenschmidt. Probabilistic Models for the Semantic Web - A Survey. *Technical Report*. 2008.

[11] E. Toch, I. Reinhartz-Berger , D. Dori. Humans, Semantic Services and Similarity: A User Study of Semantic Web Services Matching and Composition. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*. Elsevier Science. 2008.

[12] E. Sirin, J. Hendler, B. Parsia. Semi-Automatic Composition of Web Services Using Semantic Descriptions. *Web Services: Modeling, Architecture and Infrastructure Workshop in Conjunction with ICEIS*. 2003.

[13] A. Brogi, S. Corfini, R. Popescu. Semantics-Based Composition Oriented Discovery of Web Services. *ACM Transactions on Internet Technology (TOIT)*. ACM. 2008.

[14] M. Lutz. Ontology-Based Descriptions for Semantic Discovery and Composition of Geoprocessing Services. *Geoinformatica*. Springer. 2007.

[15] R. M. Smullyan. First-Order Logic. Dover Publications. 1995.

[16] D. Nebert, A. Whiteside, P. Vretanos. OpenGIS® Catalogue Services Specification, Version 2.0.2. Open Geospatial Consortium Inc. 2007.