Semantic Web Services Discovery Using Relational Database

Thu Thu Tun, Nay Zar Chi Htoo
University of Computer Studies, Yangon, Myanmar
tttun.ucsy@gmail.com, htoo.nayzarchi@gmail.com

Abstract

The Web is moving toward a collection of interoperating Web Services. Achieving this interoperability requires dynamic discovery of Web Services. This paper will develop a service repository that extends the UDDI registries and combines the information of service providers, capability descriptions of services and referred ontologies based on relationship model. That enables users to reference ontology data directly from SQL. This paper will also discuss how semantic-based web services' matching is implemented in Oracle RDBMS.

1. Introduction

Web Services are modular and self-describing applications that can be mixed and matched with other Web Services to create business processes and value chains. A critical step in the process of developing applications based on the service oriented architecture is the service discovery. For the Web Service discovery the most relevant proposal is UDDI, a standard for an online registry, to enable publishing and dynamic discovery of Web Services [1]. A Web Service provider registers its advertisements using keywords for categorization. A Web Services user retrieves advertisements from the registry based on keyword search. Current Web Services discovery technology based on UDDI does not make any use of semantic information and therefore fails to address the problem of matching between capabilities of services and allowing service location on the bases of what functionalities are sought, failing therefore to address the problem of locating Web Services. Describing the semantic of Web Services provides the ability for automatic Web Service discovery, invocation, composition and interoperation, and Web Service execution monitoring. Recently there is an important initiative in this respect, namely, OWL-S [2]. It defines an upper ontology that models the capability of a Web Service. The capability is behavior-oriented and represented by the service operation, input, output, precondition, and effect. Using the ontology concept to enhance service discovery has now become a hot research topic. However, with the large volume of information generated during service registration and the complexity of the semantic interrelationships among information, users cannot be expected to maintain the repository without some machine assistance. A challenge is providing rapid, precise access to repository information despite the large volume. Furthermore, complex semantic web applications require easy-to-use tools for data and rule storage along with query mechanisms. Although relational databases maybe excellent tools for storing and querying large data volume, lack the ability to perform OWL ontology.

In this work, we extend the UDDI register with capability description of Web Service which is defined in OWL-S profile. We design the relational schema for these descriptions and create tables on top of Oracle RDBMS to implement ontology-based semantic matching. The users can directly refer to ontology data from SQL. It is convenient to manage and replicate the UDDI register via this method. Furthermore, it takes the united data structure to express semantic descriptions related to Web Services and ontology data, which improves the efficiency of discovery.

The organization of the rest of this paper is as follows: In section 2, we motivate the reader by describing the limitations of the current Web Services discovery. Then, in the section 3 we design the storing model for Web Services Descriptions based on RDBMS schema. In the following section, the approaches of Web Services query are presented. In the section 5, we analyze the related work. Finally, in the last section we conclude the paper.

2. Problem Description

UDDI is an industry effort to provide a register for Web Services offered by businesses. It allows businesses to publish their services in a directory and enable other business to locate partners and to form business relationships based on the Web Services they provide. The UDDI specification provides structural templates for representing information about business entities, the nature of their services, and mechanisms to access them. Services can be searched by specifying business name, service name, service category and Tmodels. Unfortunately, the current search functions in UDDI are limited in their

support for making automatic service selection decisions. First, In UDDI, a Web Service can describe its functionality using a classification schemes. But it is not capable of inferring semantic relationships between entities during search. For example, a Car Rental Service might advertise itself under 'Car Rental Services' in car renting service category but a request that is looking for the services under 'Travels Reservation Services' category would not find any matches although 'Car Rental Services' is a sub category of 'Travels Reservation Services'. Second, UDDI use XML to describe its data model so it does not deal with the semantic matching for capabilities of services.

Therefore, two identical XML descriptions may have very different meaning, and vice versa. For example, UDDI can search for services that offer Car Rental Services with input parameter Price. However, it cannot search for a service that can create a reservation by taking input parameter such as Cost, although the concept Price is the same as the Cost.

3. Storing Model for Web Service Descriptions

The information that makes up a UDDI registry consists of instances of four core data structure types, the BusinessEntity, the BusinessService, the BindingTemplate and the TModel[1]. Figure 1 shows UDDI core data structures. Each BusinessEntity entity contains descriptive information about a business or organization and, through its contained information like name of the business, points of contact such as the physical address, telephone number, email and fax number, the URL of the company web site, and so on.

A BusinessEntity is associated with one or more Business Services which represents a logical service and contains descriptive information in business terms. Technical information about the BusinessService is found in the contained BindingTemplate entities. Each BindingTemplate describes an instance of a Web Service offered at a particular network address, typically given in the form of a URL. In addition to the description of businesses, services and binding templates, UDDI provides an important data structure called TModel that provides a reference system based on abstraction of technical details about the service.

UDDI is becoming the de facto standards repository of Web Services. OWL-S supplies Web Service providers with a core set of markup language constructs for describing the properties and capabilities of their Web Services in unambiguous, computer-interpretable form [3]. An OWL-S description always consists of three parts as following: Service Profile tells "what the service does"; Service Model tells "how the service works"; Service Grounding specifies "how to use the service". Among these, the Service Profile is typically required, since it provides the information needed for an agent to discover a service that meets its requirements. The Functional Description of the service describes the capabilities of the service in terms of inputs, outputs, preconditions and effects. An input is what is required by a service in order to produce a desired output. The output is a confirmation that the order has been received and successfully processed. The preconditions represent conditions in the World that should be true for the successful execution of the service. The execution of the service may result in changing the state of world. These conditions are described as the effects of the service. The service profile has purposes of advertising, discovery and matchmaking. It enables creating a richly expressive "yellow pages" of services by taxonomically encoding the kinds of information a service-requestor (whether human or software agent) must have to determine if the service meets its needs. Service profiles can be published in service registries, such as UDDI.

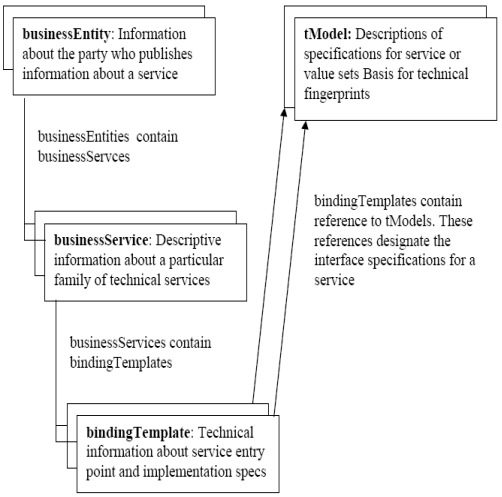

Fig.1. UDDI Core Data Structures

This paper takes united relationship schema for modeling the UDDI data structure and it's referred semantic data. An RDBMS schema has been created for storing ontologies specified in OWL. A major advantage of using a relational database is that it provides a scalable solution. Car Rental Service is used to illustrate the service discovery based on relational schema. The tables in this schema are as follows:

BusinessEntity
(BusinessKey, BusinessName, Phone, Fax, Email, PhysicalAddress, WebURL, OntologyId)
BusinessService
(ServiceKey, BusinessKey, ServiceName,

TextDescription, ServiceCategory, QualityRating, OntologyId)
Parameters
(PID, PName, Type, OntologyId, ServiceKey)
Ontologies
(OntologyID, OntologyName, Owner,…)
Terms
(TermID,OntologyID,Term,Type)
Properties
(PropertyID, OntologyID, DomainClassID, RangeClassID, Characteristics, PropertyValue)
Relationships
(OntologyID, TermID1, PropertyID, TermID2)
Restrictions
(OntologyID, NameClassID, PropertyID, MinCardinality, MaxCardinality, SomeValuesFrom, AllValuesFrom)

- BusinessEntity A given instance of the businessEntity structure is uniquely identified by its businessKey. Simple textual information about the businessEntity is given by its name, short business description and contacts. OntologyId is the reference to the ontology that BusinessEntity dependents.
- BusinessServices A given businessService entity is uniquely identified by its serviceKey. The businessKey attribute uniquely identifies the businessEntity which is the provider of the businessService. Simple textual information about the businessService, potentially in multiple languages, is given by its name and short service description. bindingTemplates is a list of technical descriptions for the Web Services provided. The categoryBag contains a list of business categories. OntologyId is the reference to the ontology that BusinessServices dependents.
- Parameters. This table stores the descriptions of capability parameters in the OWL-S profile. A given parameters entity is uniquely identified by its PID attribute. The PName attribute represents the name of capability parameters. The Type attribute identifies the type of capability parameters, such as input, output, precondition and effect. OntologyId is the reference to the ontology related to capability matching.
- Ontologies: Contains basic information about various ontologies.
- Terms: Represent classes, individuals, and properties in the ontologies. A term is a lexical representation of a concept within an ontology. TermID value is generated to be unique across all ontologies.
- Properties: Contains information about the properties. Domain and range of a property are represented with TermID values of the corresponding classes. Characteristics indicate which of the following properties are true for the
- property: symmetry, transitivity, functional, inverse functional. PropertyValue contains the value associated with PropertyID.
- Restrictions: Contains information about property restrictions. Restrictions on a property results in definition of a new class. This new class is not necessarily named (i.e., 'anonymous' class) in OWL. However, internally we create a new class for ease of representation.
- Relationships: Contains information about the relationship between two terms.

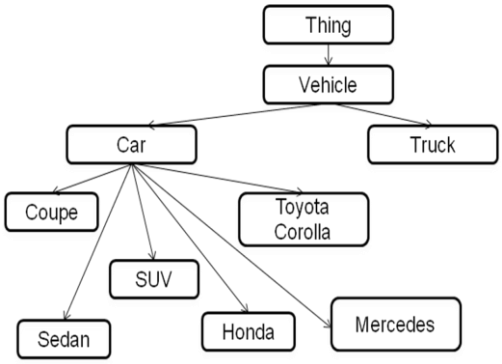

Fig.2. Sample Vehicle Ontology

For example consider a vehicle ontology, given an ontology fragment as show in Figure 2, in which the arrow represents the subclass of relationship between classes.

4. Proposed System

The Proposed system extends the UDDI register with capability description of Web Service which is defined in OWL-S profile. The relational schema is designed for these descriptions and creates tables in Oracle RDBMS to implement ontology-based semantic matching. The users can directly refer to ontology data from SQL. It is convenient to manage and replicate the UDDI register via this method. It takes the united data structure to express semantic descriptions related to Web Services and ontology data, which improves the efficiency of discovery.

According to table as shown in Section 3, we can know about semantic relationship between Sedan class and Car class as well as SUV class and Car class i.e. Sedan and SUV classes are the subclasses of Car class. "Vehicle" ontology is used to implement Car Rental Service. For example, requester is looking to rent a car. The requester asks for a Web Service's output using the class 'Car'. There are two advertised services: Car Rental Service and Travel Reservation Service, whose outputs are

Sedan and SUV respective. These services would like to take advantage of the vehicle ontology to provide better match for the requester queries. The content of tables for Vehicle ontology database and Car Rental Service database used are as shown in below:

Ontologies

OntologyID	Name	Owner
1	Vehicle	DHU

Relationships

Ontology ID	Term ID1	Property ID	Term ID2
1	Sedan	SubClass	Car

Terms

TermID	OntologyID	Term	Type
1	1	Thing	Class
2	1	Vehicle	Class
3	1	Car	Class
4	1	Truck	Class
5	1	Sedan	Class
..	..	..	..

BusinessServices

ServiceKey	ServiceName
463D55B8-8A7A-4AA1-8578-060C0EB0AFC6	CarRental Service
9E148792-7976-4F2A-B9B6-BD29DF596071	Travel Reservation Service

ID	P Name	Type	Onto_ logyID	ServiceKey
1	Sedan	Output	1	463D55B8-8A7A-4AA1-8578-060C0EB0AFC6
2	SUV	Output	1	9E148792-7976-4F2A-B9B6-BD29DF596071

5. Design and Implementation

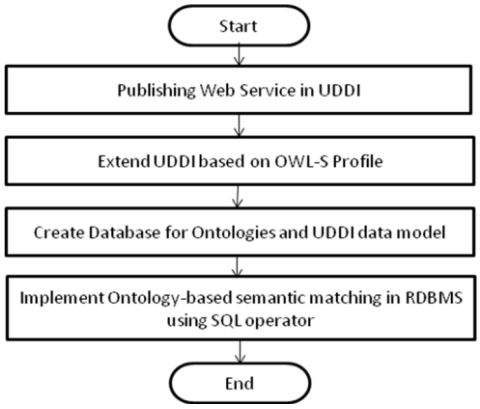

Fig.3. Overview of the Proposed System

In this system, as shown in Figure 3 we publish web service in UDDI. We use UDDI data structure in our implementation as relational tables. Oracle database is used for unified data structure, UDDI and Ontology. Figure 4 shows database design.

Database Design includes 8 tables. They are BusinessEntity, BusinessServices, Parameters, Ontologies, Terms, Relationships, Properties and Restrictions. In this system, semantic web services are published from service provider and convert into OWL-S profile and then this system store service information in our oracle database. From service requester, 'output' keyword is entered and system retrieve service key, service name, output parameters and webURL for these services.

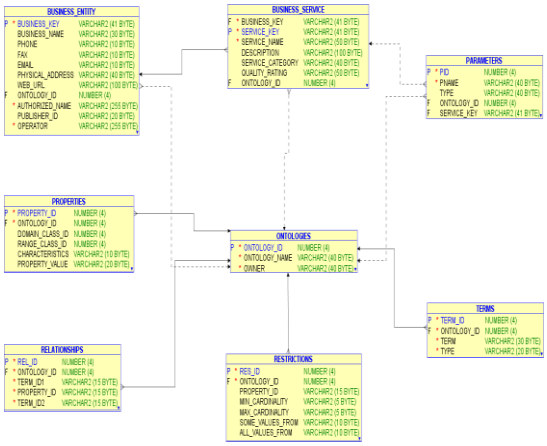

Fig.4. Database Design

To support ontology based semantic matching in RDBMS several new operators are defined. A set of SQL operators, namely ONT_RELATED, ONT_EXPAND, ONT_DISTANCE, and ONT_PATH are introduced to perform ontology-based semantic matching. ONT_RELATED operator is the most important one of them. This operator models the basic semantic matching operation. It determines if the two terms are related with respect to the specified RelType argument within ontology. If they are related it returns 1, otherwise it returns 0.

ONT_RELATED
(Term1,RelType,Term2,OntologyName)
RETURNS INTEGER;

Basic processing for both ONT_RELATED involves computing transitive closure, namely, traversal of a tree structure by following relationship links given a starting node. Oracle supports transitive closure queries with CONNECT BY clause as follows:

SELECT … FROM …START WITH <condition> CONNECT BY <condition>;

The starting node is selected based on the condition given in START WITH clause, and then nodes are traversed based on the condition given in CONNECT BY clause. The parent node is referred to by the PRIOR operator. For simplicity and compactness, we use Term1 and Term2 directly instead of TermId in Relationships table. Basically, the third argument is translated into START WITH clause and the second argument into CONNECT BY clause.

With the help of the operator, an original query:

SELECT BS.ServiceKey, BS.BusinessName
FROM BusinessServices BS, Parameters P
WHERE BS.ServiceKey = P.ServiceKey AND P.Type = 'Output' AND
ONT_RELATED(P.PName,'SubClassof','Car','Vehicle_ontology')=1;

Transformed Query:

SELECT BS.ServiceKey, BS.ServiceName
FROM BusinessServices BS, Parameters P
WHERE BS.ServiceKey = P.ServiceKey AND P.Type = 'Output' AND
P.PName In(SELECT Term1 FROM Relationships
START WITH Term2 = 'Car' AND PropertyID = 'Subclassof'
CONNECT BY
PRIOR Term1 = Term2 AND PropertyID = 'Subclassof');

This meaning that above query wants to convey is: Looking for the name and ServiceKey of Web Service whose output parameter is car or the subclass of the car. The result for this query is as follows, because the service 'Car Rental Service' has output parameter Sedan that is the subclass of Car. And the service 'Travel Reservation Service' has output parameter SUV that is the subclass of Car.

6. Related Works

The reference [3] is the base of our work. It addresses the problem of supporting ontology-based semantic matching in RDBMS. Specifically, System-defined tables are provided for storing ontologies specified in OWL. And a set of SQL operators, namely ONT_RELATED, ONT_EXPAND, ONT_DISTANCE, AND ONT_PATH are introduced to perform ontology-based semantic matching. We expend their works by the combination of UDDI data model and Ontology data model based on relationship schema. It is will improve the discovery accuracy of the Web Services under the exploiting unified data structure. [4] proposes an OWL-S/UDDI Matchmaker which takes advantage of UDDI's proliferation in the Web Service technology infrastructure and OWL-S's explicit capability representation. It provides a mapping between the OWL-S profile and the UDDI data model. The matchmaking component is completely embedded in the UDDI registry. It also added a capability port to the UDDI registry, which can used to search for Web Services based on their capabilities. DLDB[5] is a knowledge base system that extends a relational database management system with additional capabilities

for DAML+OIL inference. Making use of the FaCT DL reasoner, DLDB has been successfully implemented on a common RDBMS: Microsoft Access. It can process, store and query DAML+OIL formatted semantic content. It chooses Microsoft Access as underlying database. Such a system would not be suitable for a large-scale knowledge portal. It only fits the needs of the personal or small business user who wishes to take advantage of semantic web technology. SQUID-WS[6] is a system utilizing semantic annotation of Web Services to increase the accuracy of Web Service discovery. OWL ontologies serve as the source of semantic information in SQUID-WS. Semantic enhanced UDDI (EUDDI) and source code annotations of Web Services aid in incorporating semantics into SQUID-WS. Its discovery engine processes the information present in the EUDDI to return better matches for the user-issued query. Its inference engine employs the use of restriction on ontological concepts to further refine the results of discovery. [7] extends UDDI as "UDDIe" to address some of these restrictions, and provide three main extensions: (1) support for "leasing" to enable services to register with UDDI for a limited time period, (2) support for search on other attributes of a service – achieved by extending the BusinessService class in UDDI with propertyBag, and (3) extending the find method to enable queries to UDDI to be based on numeric and logical (AND/OR) ranges. The extensions allow UDDIe to co-exist with existing UDDI implementations – and enable a query to be issued to both simultaneously. However, this model can't support the semantic discovery of Web Services.

7. Conclus

Web S
challenge to
mechanism
direct-keywo
of Web Servi
register with
Service base
relational sch
implement o
The users can
SQL. It is
RDBMS.

References

[1] UDDI.http://www.oasis-open.org/committees/uddispec/doc/tcspecs.htm#uddiv3.

[2] The OWL Services coalition http://www.daml.org/services/owl-s/.

[3] Souripriya Das, Eugene Inseok Chong, George Eadon, Jagannathan Srinivasan: Supporting Ontology-Based Semantic matching in RDBMS. VLDB 2004: 1054-1065.

[4] Ali ShaikhAli, Omer F. Rana, Rashid J. Al-Ali, David W. Walker: UDDIe: An Extended Registry for Web Service. SAINT Workshops 2003: 85-89.

[5] Zhengxiang Pan, Jeff Heflin: DLDB: Extending Relational Databases to Support Semantic Web Queries. PSSS 2003 [Souripriya Das] Supporting Ontology-based Semantic Matching in RDBMS.

[6] Jagannathan Srinivasan, Ravi Murthy, Seema Sundara, Nipun Agarwal, Samuel DeFazio: Extensible Indexing: A Framework for Integrating Domain-Specific Indexing Schemes into Oracle8i. ICDE 2000:91-100.

[7] Ali ShaikhAli, Omer F. Rana, Rashid J. Al-Ali, David W. Walker: UDDIe: An Extended Registry for Web Service. SAINT Workshops 2003: 85-89.