Composition of Semantic Web Services Based on ER-Model

Ashok Kumar P. S, Dr. G. Mahadevan, Sohan Kumar Guptha and Gopal Krishna

Abstract—Ontology makes traditional web service to be semantic web services, where semantic annotation can be employed to discover and composite web services semantically. OWL-S can be used to automate a variety of service-related activities involving service discovery, interoperation, and composition. The ER modeling is used to produce a conceptual schema or semantic data model of a system. In this paper, we use ER model, for database management, to construct domain ontology and semantic web services effectively.

Keywords—Ontology, OWL-S, SWS, Web service composition.

I. INTRODUCTION

WEBSERVICES are API’s, designed to support interoperability in machine-to-machine interactions over the Internet. Web services can be registered, discovered and invoked by a third-party user, and be composed into a composite service with complex functionality [1]. Web Services provide a basis for interoperability between service providers and consumers, based on the reliable exchange of messages. Semantic web services are based on SWS, and OWL-S (Web Ontology Language for Services) which provide the semantic instructions to web services [3]. Semantic web services are based on knowledge representation. Composition of Web services can be conducted in either static or dynamic ways. In static composition, user pre-defines a process model which includes activities and data dependencies among the activities. In this paper, we propose to use the ER (entity-relationship) model to construct domain ontology, with which the SWS are annotated, reasoned and composed [5].

II. BACKGROUND

The semantic descriptions information will defined as a “well-defined”, often referred to as metadata

A. OWL-S:

“Ontology is a explicit specification of a shared conceptualization.” The primary role of ontologies is to enhance establishing a shared vocabulary. OWL-S is specified as a conceptual framework for describing semantic web services. OWL-S is also a language that enriches Web Services descriptions with semantic information from OWL ontologies [2][3]. OWL-S has three components –

1. ServiceProfile
2. ServiceModel
3. ServiceGrounding

Where, ServiceProfile - describe what are the different services are provided; ServiceModel - describe how the service is implemented; ServiceGrounding describe how to access the service. Web services standards (WSDL, UDDI) enable automatic web service description, discovery, and binding, and OWL-S makes web services knowledgeable [7].

III. SEMANTIC WEB SERVICES BASED ON ER-MODEL

Web Services have become the standard interface for providing access to systems and software over the web for their successful deployment in stable and secure environments [6].

A. Transformation from ER to OWL

The ER model is an abstract and conceptual representation of data and their relationships. It Provide all semantic information about ER model [8]. The basic concepts of OWL include: Class, subClass, Property and subProperty. Employee is a Class, employeOf is a Property, Ashok is an Individual of Class Employee. There are two types of properties: ObjectProperty (the property is a Class) and DataTypeProperty (the property is a DataType).

Classes are major players in OWL, which represent entities in certain domain. The entities in ER model can be readily mapped to the Classes in OWL. The Property in OWL is, corresponding to the property of entity in ER model. A mapping definition from ER model to OWL with an example is shown in Figure 1. The ForeignKey denotes the relationship between entities [19]. The relationships in the ER model can only be represented in the form of 1:1, 1: n, n: m,

Ashok Kumar P.S. is Reserch Scholar, Anna University, Coimbatore, ashokps2000@yahoo.com
Dr. G. Mahadevan is Prof & Head, AMCE, Bengaluru, mahadevan_g@yahoo.com
Sohan Kumar Guptha, Asst. Prof, Dept. of CSE, YDIT, Bengaluru, ksohan1@yahoo.co.in
Gopal Krishna, Asst. Prof, Dept. of CSE, AIT, Chikmagaluru, nithingopal@rediffmail.com
The purpose for transforming the ER model to ontology is to construct SWS. OWL-S is one of major ontology description languages for SWS. AtomicProcess is the process which can be directly invoked, and cannot be decomposed further. SimpleProcess is also composed by several AtomicProcesses, and it is normally abstracted as a single-step action. SimpleProcess can be regarded as a static composition, in which the control structures are fixed. CompositeProcess is composed of a set of AtomicProcesses followed by control structures, [6].

IV. SEMANTIC WEBSERVICES COMPOSITION BASED ON ER

SWS composition requires domain experts, i.e. we need to understand domain knowledge and underlying data structure, this makes SWS composition a challenging task [7]. We present the scenario of dynamic service composition according to ER models in the next subsections.

A. ER Transformation in Tree View

When the ER model is in a tree view, the relationship path between two entities is unique. When composing the service that queries the related classes of a certain class and no such SimpleProcess available for the query, Figure 2, represents an example for the class Management System of an ER Transformation, where The upper half in a figure is the ER model, and the lower half is a transformed OWL model [4][6].

V. CASE STUDY

In this section, we demonstrate how our approach can be applied in a concrete application by a case study. This case study is described as follows – A Car Dealer manage their in a database system, which includes data for the Car provided by different Car Vendors. To satisfy branded Car requests of various customers, the Dealer tries to provide customers more flexibility by introducing the SWS to query of Car from various Vendors. In this case study, SWS for Car query is annotated and composed based on the ER model.

The Entities in ER model are translated into Classes in OWL, the attributes of entity are translated into DataTypeProperties in OWL, and the Relationships between entities are translated into ObjectProperties in OWL has [2].

```
<owl:Class rdf:ID="Car">  
  <owl:DatatypeProperty rdf:ID="hasColor">  
    <rdfs:domain rdf:resource="#Car"/>  
    <rdfs:range rdf:resource="string"/>  
  </owl:DatatypeProperty>  
  <owl:ObjectProperty rdf:ID="hasMaker">  
    <rdf:type rdf:resource="Functional Property"/>  
    <rdfs:domain rdf:resource="Car"/>  
    <rdfs:range rdf:resource="Car"/>  
  </owl:ObjectProperty>  
  ..........  
</owl:Class>
```

The subClass can be created by the value of DataTypeProperties of entity. For example, the Class Car has subClasses like, WhiteCar, BlackCar, RedCar, etc. The OWL description of the WhiteCar subClass is shown below,

```
<owl:Class rdf:ID="WhiteCar">  
  <owl:intersectionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="Car"/>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="hasColor"/>  
      <owl:hasValue rdf:datatype="string">White</owl:hasValue>  
    </owl:Restriction>  
  </owl:intersectionOf>  
  ..........  
</owl:Class>
```

The final OWL model of the Car ontology by above the translation steps shown in Figure 3.
The original user requirement is to request the \textbf{RedDieselSwiftCar} from \textbf{Bengaluru}. This requirement can be further decomposed into 2 sub-requirement queries,

i. \textbf{RedDieselSwiftCar} produced in Bengaluru,

ii. \textbf{RedSwiftCar} and \textbf{DieselSwiftCar}.

\section*{VI. CONCLUSION}

In this paper, we described OWL-S, it is a representational framework that provides for more complete specifications of the capabilities and behavior of Web Services, it support greater automation of service-related activities like SWS discovery, reasoning and composition. The SWS construction and composition method based on the ER model, which transforms existing knowledge into ontology into OWL for the service annotation. We propose a method that can construct semantic ontology for service annotation, but it requires lots of pre-effort to understand the business relationships. Our method relies on the ER model, which is sufficient to address most of web services requests and is especially beneficial for small and medium enterprises, but for large enterprises, transform the user requirements into SQL queries is a challenge task.

\section*{REFERENCES}


[2] David Martin1, Massimo Paolucci2, Matthias Wagner, Bringing Semantic Annotations to Web Services: OWL-S from the SAWSDL Perspective


