Exploiting Web Service Semantics: Taxonomies vs. Ontologies

Asuman Dogac, Gokce Laleci, Yildiray Kabak, Ibrahim Cingil

Software Research and Development Center
Middle East Technical University (METU)
06531 Ankara Turkiye
email: asuman@srdc.metu.edu.tr

Abstract

Comprehensive semantic descriptions of Web services are essential to exploit them in their full potential, that is, discovering them dynamically, and enabling automated service negotiation, composition and monitoring. The semantic mechanisms currently available in service registries which are based on taxonomies fail to provide the means to achieve this. Although the terms "taxonomy" and "ontology" are sometimes used interchangably there is a critical difference. A taxonomy indicates only class/subclass relationship whereas an ontology describes a domain completely. The essential mechanisms that ontology languages provide include their formal specification (which allows them to be queried) and their ability to define properties of classes. Through properties very accurate descriptions of services can be defined and services can be related to other services or resources.

In this paper, we discuss the advantages of describing service semantics through ontology languages and describe how to relate the semantics defined with the services advertised in service registries like UDDI and ebXML.

1 Introduction

When looking towards the future of web-services, it is predicted that the breakthrough will come when the software agents start using web-services rather than the users who need to browse to discover the services. Currently well accepted standards like Web Services Description Language [WSDL] and Simple Object Access Protocol [SOAP] make it possible only to "dynamically access" to Web services in an application. That is, when the service to be used is known, its WSDL description can be accessed by a program which uses the information in the WSDL description like the interface, binding and operations to dynamically access the service. However to dynamically *discover* services, say through software agents require detailed semantic information about the services to be available.

Currently, a number of taxonomies are being used to discover services in service registries like [UDDI] or [ebXML]. The most widely used taxonomies are North American Industrial Classification Scheme [NAICS] for associating services with "industry" semantics; Universal Standard Products and Services Classification [UNSPSC] for classifying product/services and [ISO 3166] for locale.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

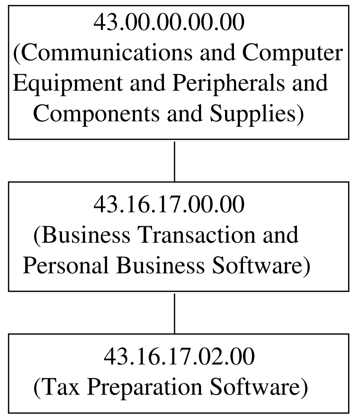

Figure 1: A part of an UNSPSC Taxonomy

A taxonomy is a hierarchy and a unique code is usually assigned to each node of the hierarchy. This code also encodes its path. For example, as shown in Figure 1, UNSPSC code for *Communications and Computer Equipment and Peripherals and Components and Supplies* is "43.00.00.00.00". One of the classes under this class is *Business transaction and personal business software* whose code is "43.16.17.00.00". Going one more level deep,"43.16.17.02.00" is the code for *Tax preparation software*.

By relating a service with the codes in such taxonomies, it is possible to give the service a certain amount of semantics. For example, when a service uses the code "43.16.17.02.00" to describe its semantics, we understand that the service is about "tax preparation software". Therefore, a user looking for a service related with a "tax preparation software" can search the service registries with the corresponding UNSPSC code to obtain all services that have declared themselves to be related with this UNSPSC code. Note however that these services could be anything; they may be "selling" such software; they may be providing it as a service. Furthermore the service may have any number of properties like certain service qualities, say, minimum and maximum service delivery times; may require certain payment obligations, say, advance credit card payment. Additionally a service may be available at a discount price when it is used in aggregation with the other services that the company provides. There can be several such properties of a service and hence the user has to go through the services found to manually pick the service that satisfies her requirements. Notice for example that, in UDDI registries, this information can only be made available informally (i.e., with no formal semantics) through "OverviewDocs" or "OverviewURLs" of the tModels associated with the services. In short, taxonomies do not help in this respect.

It follows that to exploit the Web services to their full potential we need more powerful tools, that is, ontologies to describe their semantics. In fact, currently, describing the semantic of Web in general [Berners-Lee], and semantic of Web services in particular are very active research areas. There are a number of efforts for describing the semantics of Web services such as [McIlraith a, McIlraith b, Denker]. Among these [DAML-S] defines an upper ontology, that is, a generic "Service" class. In order to make use of DAML-S upper ontology, the lower levels of the ontology need to be defined. To provide interoperability, application domains must share such specifications. In fact, an ontology describes *consensual knowledge*, that is, it describes meaning which has been accepted by a group not by a single individual. Standard bodies need to define domain specific ontologies.

In this paper, we discuss the advantages of defining service semantics through ontology languages. We further note that, once the semantic is defined, it is also necessary to relate the defined semantics with the services advertised in the service registry. Therefore we describe how this can be achieved in UDDI and ebXML registries.

The paper is organized as follows: Section 2 briefly introduces DAML-S upper ontology and describes the advantages of ontology languages over taxonomies. Section 3 describes mechanisms provided by UDDI and ebXML registries for associating semantics with the services advertised.

2 Service Semantics through Ontology Languages

Web services, like their real life counterparts, may have many properties. The aim of this section is to demonstrate that all the necessary properties of services can easily be defined through an ontology language. While developing domain specific ontologies, it is a good idea to ground them in upper ontologies since in this way they are more consistent and it becomes easier to integrate them within distributed heterogeneous systems.

DAML-S provides such an upper ontology, that is, it defines a top level "Service" class with some generic properties common to most of the services. The "Service" class has the following three properties:

- *presents:* The range of this property is *ServiceProfile* class. That is, the class Service *presents* a *ServiceProfile* to specify what the service provides for its users as well as what the service requires from its users.

- *describedBy:* The range of this property is *ServiceModel* class. That is, the class Service is *describedBy* a *ServiceModel* to specify how it works.

- *supports:* The range of this property is *ServiceGrounding*. That is, the class Service *supports* a *ServiceGrounding* to specify how it is used.

DAML-S is based on [DAML+OIL] and DAML+OIL allows very sophisticated ontologies to be defined and queried through, for example, DAML APIs. The queries on ontologies usually access the properties of classes or traverse the ontology. Hence it is possible to standardize queries to facilitate their use in an automated way.

In the following we provide examples for some of the properties of DAML-S Service Profile:

- *serviceParameters:* In DAML-S, service parameters denote an expandable list of RDF properties that may accompany a profile description. The range of each property is unconstrained, i.e. no range restrictions are placed on the service parameters as shown in the following:

```
<daml:Property rdf:ID="serviceParameter">
 <daml:domain rdf:resource="&service;#ServiceProfile"/>
 <daml:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Thing"/>
</daml:Property>
```

- *degreeOfQuality:* This property of Service Profiles provide qualifications about the service.

```
<daml:Property rdf:ID="degreeOfQuality">
 <daml:domain rdf:resource="&service;#ServiceProfile"/>
 <daml:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Thing"/>
</daml:Property>
```

2.1 An Example Service Ontology

In this section we define an example ontology grounded in DAML-S for tax services for the sole purpose of describing the power of ontology languages. A detailed description of several properties of services including the methods of charging and payment, the channels by which the service is requested and provided, constraints on temporal and spatial availaility, service quality, security, trust and rights attached to a service, is given in [O'Sullivan]. Through the example ontology we show how some of these properties can be defined.

As shown in Figure 2, the top level class of this ontology is "TaxServices" which inherits from DAML-S "Service" class. In this way, several properties of the "TaxServices" class are conveniently defined by inheriting from the properties of the DAML-S "Service" class. For example, "paymentMethod" is defined as a subproperty

```
<!DOCTYPE uridef[
 <!ENTITY rdf     "http://www.w3.org/1999/02/22-rdf-syntax-ns">
 <!ENTITY rdfs    "http://www.w3.org/2000/01/rdf-schema">
 <!ENTITY xsd     "http://www.w3.org/2000/10/XMLSchema">
 <!ENTITY daml    "http://www.daml.org/2001/03/daml+oil">
 <!ENTITY profile "http://www.daml.org/services/daml-s/2001/05/Profile.daml">
 <!ENTITY service "http://www.daml.org/services/daml-s/2001/05/Service.daml">
 <!ENTITY tp      "http://www.srdc.metu.edu.tr/2002/10/TaxPayment.daml">
 <!ENTITY unspsc  "http://www.eccma.org/unspsc/browse/43.html">

<rdf:RDF
     xmlns:rdf =     "&rdf;#"
     xmlns:rdfs =    "&rdfs;#"
     xmlns:xsd =     "&xsd;#"
     xmlns:daml =    "&daml;#"
     xmlns:profile = "&profile;#"
     xmlns:service = "&service;#"
     xmlns:unspsc =  "&unspsc;#"
     xmlns:tp =      "&tp;#"

<daml:Ontology rdf:about=" ">
  <daml:imports rdf:resource="http://www.daml.org/2001/03/daml+oil"/>
  <daml:imports rdf:resource="&service;">  </daml:Ontology>

<daml:Class rdf:ID="TaxServices">
  <rdfs:subClassOf rdf:resource="&service;"/> </daml:Class>

<rdf:Property rdf:ID="paymentMethod">
   <rdfs:subPropertyOf rdf:resource=&profile;serviceParameter/>
   <rdfs:domain rdf:resource="&service;#ServiceProfile"/>
   <rdfs:range rdf:resource="&daml;#Thing"/> </rdf:Property>

<rdf:Property rdf:ID="serviceGuarantee">
   <rdfs:subPropertyOf rdf:resource=&profile;degreeOfQuality/>
   <rdfs:domain rdf:resource="&service;#ServiceProfile"/>
   <rdfs:range rdf:resource="&daml;#Thing"/> </rdf:Property>

<rdf:Property rdf:ID="requiredService">
  <rdfs:domain rdf:resource="&damls;#Service/>
  <rdfs:range rdf:resource="&damls;#Service/> </rdf:Property>

<rdf:Property rdf:ID="discountAmount">
  <rdfs:domain rdf:resource="#PromotionRequirements"/>
  <rdfs:range rdf:resource="&xsd;#nonNegativeInteger"/> </rdf:Property>

<daml:Class rdf:ID="TaxPreparationService">
  <rdfs:subClassOf rdf:resource="#TaxServices"/>
  <rdfs:label> Tax Preparation Service </rdfs:label>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#promotion"/>
      <daml:toClass rdf:resource="#PromotionRequirements"/>
    </daml:Restriction> </rdfs:subClassOf> </daml:Class>

<rdf:Property rdf:ID="promotion">
   <rdfs:subPropertyOf rdf:resource=&profile;serviceParameter/>
   <rdfs:domain rdf:resorce="#TaxPreparationService"/>
   <rdfs:range rdf:resource="#PromotionRequirements"/> </rdf:Property>

<daml:Class rdf:ID="PromotionRequirements">
   <rdfs:subClassOf rdf:resource="#TaxServices"/>
   <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#requiredService"/>
       <daml:toClass rdf:resource="&damls;#Service"/>
    </daml:Restriction> </rdfs:subClassOf> </daml:Class>

<daml:Class rdf:ID="LegalConsulting">
  <rdfs:subClassOf rdf:resource="#TaxServices"/>
  <rdfs:label> Legal Tax Consultancy Service </rdfs:label>
</rdfs:subClassOf> </daml:Class>

<rdf:Property rdf:ID="serviceCodeUNSPSC">
  <rdfs:label> Defining a property to denote UNSPSC codes of services</rdfs:label>
  <rdfs:domain rdf:resource="&damls;#Service/>
  <rdfs:range rdf:resource="&unspsc;#UNSPSCCodes"/> </rdf:Property>
</rdf:RDF>
```

Figure 2: An Example Ontology Description for Tax Payment Domain

of DAML-S "ServiceProfile serviceParameter" property and "serviceGuarantee" as a subproperty of DAML-S "ServiceProfile degreeOfQuality" property. Notice that, in this generic ontology, ranges of these properties are defined to be the most general class in a DAML+OIL ontology, that is, *daml+oil#Thing* class. This class is specialized when defining ontology instances as shown in Figure 3. This general ontology also states that a

```
<!DOCTYPE uridef[
 <!ENTITY rdf     "http://www.w3.org/1999/02/22-rdf-syntax-ns">
 <!ENTITY tp      "http://www.srdc.metu.edu.tr/2002/10/TaxPayment.daml">

<rdf:RDF
     xmlns:rdf =     "&rdf;#"
     xmlns:tp =      "&tp;#"

<tp:TaxPreparationService rdf:ID="TaxHeavenTaxPreparationService">
  <tp:serviceCodeUNSPSC>43.16.17.02.00 </tp:serviceCodeUNSPSC>
  <tp:paymentMethod> CreditCard </tp:paymentMethod>
  <tp:serviceQuarantee>24</tp:serviceQuarantee>
  <tp:promotion rdf:resource="#MyPromotionRequirements"/>
</tp:TaxPreparationService>

<tp:PromotionRequirements rdf:ID="MyPromotionRequirements">
  <tp:requiredService rdf:resource="#LegalConfort"/>
  <tp:discountAmount>10 </tp:discountAmount>
</tp:PromotionRequirements>

<tp:LegalConsulting rdf:ID="LegalConfort">
</tp:LegalConsulting>
</rdf:RDF>
```

Figure 3: Service Ontology of the company “TaxHeaven”

“promotion” property can be associated with “TaxPrepatationService” which requires another service to be used and specifies the discount amount when the two services are used in aggregation. Finally, the ontology states that the UNSPSC codes may be associated with services.

Figure 3 shows a specific instance of the generic ontology given in Figure 2, for the company “TaxHeaven”. “TaxHeaven” is providing a tax preparation service. The service has the maximum delivery time of 24 hours; and accepts only credit card payment. “TaxHeaven” also provides a seperate legal consultancy service for tax payment, called “LegalConfort” and provides 50% discount for this service to the users of their tax preparation service, namely, “TaxHeavenTaxPreparationService”.

To provide ease in readability, these ontologies are also shown graphically in Figure 4. The graphical representation followed is inspired by [Gonzalez-Castillo].

3 Associating Semantics with Service Registries

In this section we describe how to associate the semantic defined with services advertised by using the mechanisms provided by UDDI and ebXML registries.

3.1 UDDI Registries

The mechanism to relate semantics with services advertised in the UDDI registries are the tModel and the catagory bags of registry entries. tModels provide the ability to describe compliance with a specification, a concept, a shared design or a taxonomy. Services have category bags and any number of tModel keys can be put in these category bags.

In relating the semantics defined in DAML+OIL with the services advertised in UDDI registries, the first question to be answered is where to store the semantic descriptions. Generic descriptions can be stored by the standard bodies who define them and the server, where the service is defined, can host the semantic description of the service instance. This facilitates the maintenance of the service descriptions. However there are times, when it is necessary to query all the individual service descriptions. Therefore a combined schema per industry domain containing all the semantic descriptions of the services pertaining to this domain may be necessary to facilitate global querying.

The second issue is relating the ontology defined in DAML+OIL with the services advertised in the UDDI registry. For this purpose, similar to WSDL, DAML+OIL should be classified as “DAMLSpec” with “uddi-org:types” taxonomy. A seperate tModel of type “DAMLSpec” for the combined schema of each industry

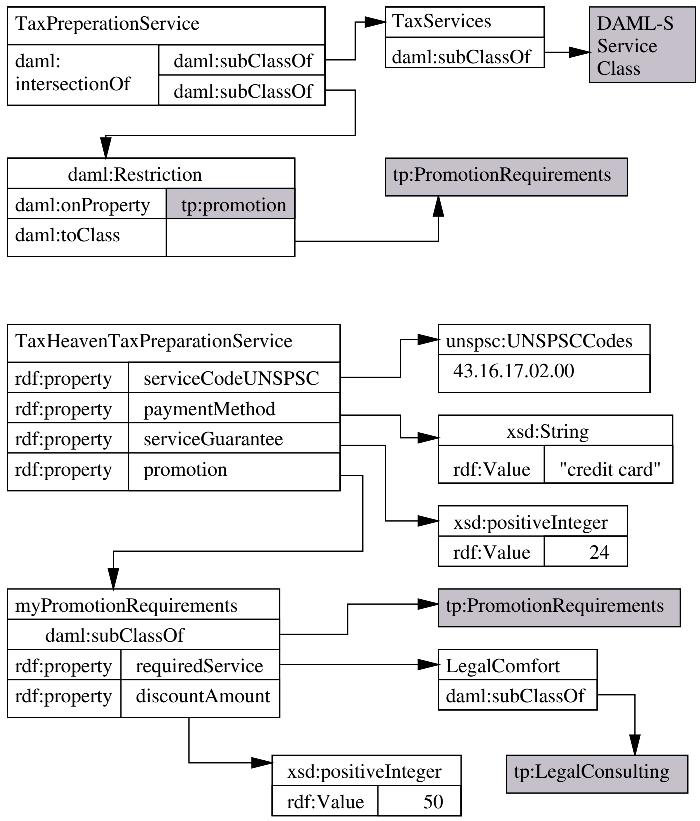

Figure 4: A pictorial description of “TaxHeaven” ontology

domain should be created. The services in an industry domain must contain the key of this tModel in their category bags. Hence, this tModel key can be used to find services in an industry domain through the UDDI registry.

There may be times where it is necessary to find all the instances of a generic class. For example, to choose a tax payment service with required functionality, it may be necessary to retrieve all semantic tax payment service descriptions to check their properties. To be able to do this, that is, in order to find instances of a generic class, it is necessary to associate a tModel key for each generic service class and store this tModel key with individual service descriptions.

Finally, there should be a tModel key for each service instance. This tModel key can be used in searching the UDDI registry to find a particular advertised service instance according to its semantic description. A more detailed treatment of this issue is given in [Dogac].

3.2 ebXML Registries

The basic mechanism in ebXML registries for associating semantics with the objects stored in the registry is the “classification” hierarchy, called *ClassificationScheme*. *ClassificationScheme* defines a hierarchy of *ClassificationNodes*. The nodes in this hierarchy are related with registry objects through *Classification* objects. A *Classification* instance classifies a RegistryObject instance by referencing a node defined within a particular classification scheme. As an example, assume that “TaxHeavenTaxPreparationService” stored in the ebXML registry and a *ClassificationScheme* exits for the “Tax Payment” industry like the one provided in Figure 2. Figure 5 demostrates how this service is associated with this classification schema by using the classification object.

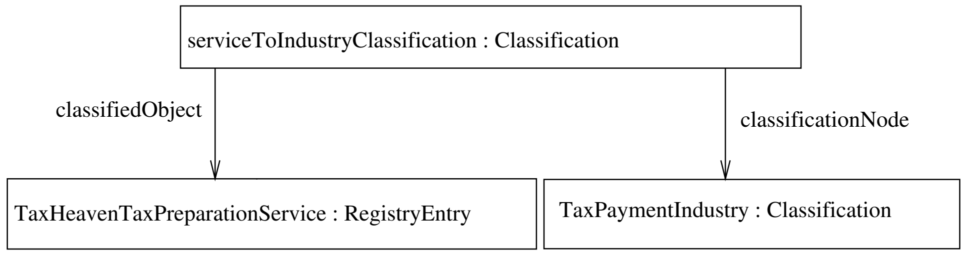

Figure 5: Associating a Service with a Classification Node in ebXML

Note that a registry object can be classified according to any number of classification schemes.

However classification structure provided by ebXML is not adequate to store DAML+OIL ontologies and need to be extended to be used for this purpose. Also ebXML registry interface needs to be extended to query DAML+OIL ontologies.

References

[Berners-Lee] Berners-Lee, T., Hendler, J., Lassila, O., “The Semantic Web”, Scientific American, May 2001.

[DAML+OIL] DAML+OIL, http://www.w3.org/2001/10/daml+oil

[DAML-S] DAML Services Coalition (A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, H. Zeng), DAML-S: Semantic Markup for Web Services, in Proceedings of the International Semantic Web Working Symposium (SWWS), July 2001.

[Denker] Denker, G., Hobbs, J. R., Narayan, S., Waldinger, R., “Accessing Information and Services on DAML-Enabled Web, Semantic Web Workshop, Hong Kong, China, 2001.

[Dogac] Dogac, A., Cingil, I., Laleci, G. B., Kabak, Y., “Improving the Functionality of UDDI Registries through Web Service Semantics”, 3rd VLDB Workshop on Technologies for E-Services (TES-02), Hong Kong, China, August 23-24, 2002.

[ebXML] ebXML, http://www.ebxml.org/

[Gonzalez-Castillo] Gonzalez-Castillo, J., Trastour, D., Bartolini, C., “Description Logics for Matchmaking of Services”, Technical Report, HP Labs, Bristol, UK.

[ISO 3166] ISO 3166, http://www.iso.ch/iso/en/prods-services/iso3166ma/index.html

[McIlraith a] McIlraith, S. A., Son, T. C., Zeng, H., “Semantic Web Services”, IEEE Intelligent Systems, March/April 2001, pp. 46-53.

[McIlraith b] McIlraith, S. A., Son, T. C., Zeng, H., “Mobilizing the Semantic Web with DAML-Enaled Web Services”, Semantic Web Workshop 2001, Hongkong, China.

[NAICS] North American Industrial Classification Scheme (NAICS) codes http://www.naics. com.

[SOAP] Simple Object Access Protocol (SOAP) http://www.w3.org/TR/SOAP/

[O’Sullivan] O’Sullivan, J., Edmond, D., Hofstede, A., ”What’s in a Service? Towards Accurate Description of Non-Functional Service Properties”, in the Journal of Distributed and Parallel Databases, Vol. 12, No. 2/3, Sept./Nov. 2002.

[UDDI] Universal Description, Discovery and Integration (UDDI), http://www.uddi.org/.

[UNSPSC] Universal Standard Products and Services Classification (UNSPSC) http://eccma.org/unspsc

[WSDL] Web Service Description Language (WSDL), http://www.w3.org/TR/wsdl