Semantically Enriched Web Services for the Travel Industry *

A. Dogac, Y. Kabak, G. Laleci, S. Sinir, A. Yildiz, S. Kirbas, Y. Gurcan

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

ABSTRACT

Today, the travel information services are dominantly provided by Global Distribution Systems (GDS). The Global Distribution Systems provide access to real time availability and price information for flights, hotels and car rental companies. However GDSs have legacy architectures with private networks, specialized hardware, limited speed and search capabilities. Furthermore, being legacy systems, it is very difficult to interoperate them with other systems and data sources. For these reasons, Web service technology is an ideal fit for travel information systems.

However to be able to exploit Web services to their full potential, it is necessary to introduce semantics. Without describing the semantics of Web services we are looking for, it is difficult to find them in an automated way and if we cannot describe the service we have, the probability that people will find it in an automated way is low. Furthermore, to make the semantics machine processable and interoperable, we need to describe domain knowledge through standard ontology languages.

In this paper, we describe how to deploy semantically enriched travel Web services and how to exploit semantics through Web service registries. We also address the need to use the semantics in discovering both Web services and Web service registries through peer-to-peer technology.

1. INTRODUCTION

Currently, travel information services are dominantly provided by Global Distribution Systems (GDS). A GDS gives its subscribers pricing and availability information for multiple travel products. Travel agents, corporate travel departments, and even Internet travel services, subscribe to one or more GDSs to check, for example, flight availability and prices for their customers. The GDSs get their revenue from the booking fees that these organizations pay. In addition to supplying information, a GDS typically provides hardware, software and technical support to its customers, including printers that allow agencies to print airline tickets. The leading GDSs today are Sabre [21], Galileo [8], Amadeus [1] and Worldspan [25]. All the airlines, many hotel chains and car rental companies list their inventory with major GDSs.

Yet GDSs are legacy systems and suffer from the following problems:

- Mostly they rely on their own private networks although there are recent efforts to make these systems available through the Internet.
- GDSs are mainly for human use. They have difficult to use cryptic languages (with the exception of specialized APIs some GDSs are developing for the eCommerce sites). A request to the system usually involves more than one interaction with the person on the terminal in contrast to the current trend which is the automation of services over the Internet. Furthermore, GDSs have limited speed and search capabilities.
- Being legacy systems, it is difficult to interoperate them with other systems and data sources both inside the company as well as with external resources. For example, a travel agency may wish to integrate its Customer Relationship Management (CRM) system with the GDS to better serve its customers.

There are several advantages to be gained from using Web service technology in travel information systems:

- Interoperability among very many heterogeneous systems such as flight reservation and hotel booking systems of individual companies will be facilitated. Furthermore, it will become possible to integrate the back end systems of the travel companies.
- Among the millions of travel agents, only about 10% to 20% of all travel agents are in connection with the GDS companies. Most travel agencies and travel organization companies choose to manage travel services by themselves, due to reasons which may be technical or economical, e.g. due to high commission rates and yearly subscription fees issued by the GDS company. Web service technology will enable such companies both to obtain the travel products they need from the Internet as well as to offer their products over the Internet.
- Also travel agencies generally choose to communicate with travel service providers directly bypassing GDSs,

*This work is supported by the European Commission through IST-1-002104-STP Satine project and in part by the Scientific and Technical Research Council of Turkey, Project No: EEEAG 104E013

for better pricing than GDS listings and to obtain specialized services. Nevertheless, extra effort is required for traditional communication and necessity for establishing partnerships, whereas Web service technology provides a solution for all companies, generally small-to-medium level enterprises in the travel industry, for easily publishing and selling their tourism services as well as providing large selection options from a rich service pool.

- GDS companies do not support every type of travel product in the industry, but only the major ones; airline ticketing, hotel reservation and car rental. However, there are other types of specialized services, especially different types of tour and transportation based travel products. These services generally do not have a common, uniform structure as found in airline, hotel and car-rental services hence it is difficult to develop a common interface for such kinds of services to be provided by GDSs. On the other hand, Web service technology is an ideal fit for publishing such services by facilitating their discovery through the Web service registries as well.

- GDSs provide hotel booking only for major hotel chains. There are millions of hotels that cannot sell their rooms through GDSs. These hotels will benefit from the Web service technology.

- A variety of business opportunities for the companies in the travel industry will be enabled by using the communication medium as ubiquitous as Internet. The travel agencies and service providers will collaborate with each other on a new level. The companies will no longer need to make pre-agreements with respective suppliers; the service alternatives will be found on the fly through Web service discovery. It will be a possibility to construct package services which are comprised of services provided locally by the initiator company and services discovered within the network. Negotiation on the service that will be purchased as well as customization of service properties on the fly are other possible types of enhancements in the e-business for the travel domain.

A few early adopters in the travel domain have started to develop Web services. Sabre [22] and Datalex [20] are among the first companies to develop Web services. Sabre Web Services provide all the functionality needed to sell travel (air, car rental, hotel, passenger name record). It is especially convenient for agencies with no Sabre system format expertise. Galileo also provides a Web service based solution and claims to have cut down the development time by 80% [9]. Another travel company which benefited from the Web service technology is the Continental Airlines in USA. To gain a competitive edge, Continental decided to provide real-time data of flight status in multiple contexts, such as interactive devices and customer service agent consoles. By wrapping the data it already has in its legacy flight operations management system as Web services, the information became accessible from devices such as cellular phones or PDAs.

However to be able to exploit Web services to their full potential, it is necessary to introduce semantics to Web services. Without semantics, it is difficult for Web Services to succeed because if we cannot describe the Web service we are looking for, we cannot find it in an automated way and if we cannot describe the service we have, the probability that people will find it in an automated way is low.

Furthermore, to make semantics machine-processable and interoperable, we need to define semantics through standard ontology languages. Web Ontology Language (OWL) [19] by W3C is a good candidate. In fact, OWL-S [3] consortium has specified an upper ontology to define the semantics of Web services. OWL-S specification, although very useful, is generic and applicable to all services. There is a need for domain specific (such as travel or healthcare domain) ontologies to be able to associate semantics with Web services.

For travel domain, there are some promising efforts for developing travel domain ontologies such as the Harmonise project [11]. The Harmonise project allows participating organisations to keep their proprietary data format and use ontology mediation while exchanging information. For this purpose, they have defined the *Interoperability Minimum Harmonization Ontology* (IMHO) and an interchange format for tourism industry [10]. The Mapping FRAmework (MAFRA) [12] tool is used for ontology mediation which supports semantic mapping definitions and the reconciliation engine.

Another important initiative for describing travel domain knowledge is the XML Schema specifications defined by the Open Travel Alliance (OTA) [18]. OTA includes majority of the key players in the industry covering airlines, hotels, car rental, rail, and tour companies and Global Distribution Systems. It has produced XML schemas of the message specifications to be exchanged between the trading partners. These messages include availability checking, booking, rental, reservation, reservation canceling and modifying, query services for service details and quality, insurance quote request for all of the hotel, airline, vehicle sectors as well as the commission exchange services, and the statistical information services. We believe that OTA specifications can prove very useful for defining travel Web service ontologies. If such ontologies become available for the travel domain, the interoperability of all sorts of Web services can be better addressed at the semantic level through ontology mapping.

In this paper, we describe how to deploy semantically enriched Web services for the travel industry and how to exploit service semantics for service interoperation and service discovery.

Another issue addressed in this paper is the following: semantics is also necessary for the discovery of Web service registries. Today, the main mechanism for service discovery is service registries. However Web service registries themselves are also in need of discovery. Providing a mechanism to facilitate the automated discovery of Web service registries is also needed and we propose a peer-to-peer mechanism for the discovery of Web service registries.

The paper is organized as follows: Section 2 briefly summarizes the OTA specifications which reveal considerable domain expertise in the travel domain and hence can be exploited for defining travel ontologies. Section 3 describes the need for Web service functionality and message ontologies for describing the semantics of Web services and how OTA can be taken advantage of for this purpose. Section 4 discusses how to relate service ontologies with Web services registries. Semantic-based discovery of service registries in peer-to-peer networks is given in Section 5. Finally, Section

6 concludes the paper.

2. OPEN TRAVEL ALLIANCE SPECIFICATIONS

OTA [18] exposes considerable amount of domain knowledge which can be used in describing the semantics of travel Web services. In this section, we briefly describe the functionality provided by OTA. In Section 3, we describe how this knowledge can be exploited in defining Web service semantics. It should be noted that our aim is not to develop ontologies for the travel domain but to demonstrate that a higher level interoperability can be achieved among existing applications through ontology mapping.

OTA XML message schema specifications can be investigated in the following groups:

- *Generic messages*: Although many messages are specific to a particular travel subdomain such as *Air*, some messages are generally applicable and may be used more broadly. Such messages include cancel (OTA_CancelRQ.xsd), and delete (OTA_DeleteRQ.xsd) messages.

- *Air message specifications* address the structure and elements of requests and responses for airline flight related information and contain extensive set of messages like air availability request (OTA_AirAvailRQ.xsd), air availability response, (OTA_AirAvailRS.xsd), and low airfare search request (OTA_AirLowFareSearchRQ.xsd).

- *Car message specifications* include all sorts of messages related with vehicles such as vehicle availability request (OTA_VehAvailRateRQ.xsd) and vehicle location details (OTA_VehLocDetailRQ.xsd).

- *Hotel message specifications* provide ability to search and obtain all related information about hotel products such as search for a hotel (OTA_HotelSearchRQ.xsd), and to get a list of rooms (OTA_HotelRoomListRQ.xsd).

- *Golf Tee Times* provides three separate request/response pairs of messages to request data from another system to find a golf course, to inquire availability and to book a tee time. For example, the message schema "OTA_GolfCourseAvailRQ.xsd" is used for requesting golf course availability.

- *Insurance message specifications* contain XML schemas for insurance related messages such as insurance plan search (OTA_InsurancePlanSearchRQ.xsd), and insurance quote request (OTA_InsuranceQuoteRQ.xsd).

- *Package Tours/Holiday Bookings*: A package holiday usually consists of a single "pre-defined" offering with or without a choice of a number of bookings. A booking can contain any number of itinerary elements, such as transport, accommodation, car rental, extra products or services, special services, extras, etc. Some example schemas from this group include package availability request (OTA_PkgAvailRQ.xsd) and package booking request (OTA_PkgBookRQ.xsd).

- *Travel Itinerary messages* are widely used to integrate, manage and service travel content and include the industry segments "Air", "Car", "Hotels", "Rail" and "Tour&Cruise". Travel itinerary read request (OTA_TravelItineraryReadRQ.xsd) and response are used to get this information.

- *Rail information messages* contain three request/response pairs to check rail availability (OTA_RailAvailRQ.xsd), rail booking (OTA_RailBookRQ.xsd) and (OTA_RailRetrieveRQ.xsd) to retrieve a previously booked itinerary.

- *Loyalty message specifications*: Many companies in the travel industry offer loyalty programs. In the past, companies managed their own loyalty programs but now there are specialized companies to manage them. This standard message set allows the travel industry to communicate with loyalty industry. Some example messages include creating loyalty account (OTA_LoyaltyAccountCreateRQ.xsd), and creating loyalty certificates (OTA_Loyalty CertificateCreateNotifRQ.xsd).

- *Profile*: The Profile messages define the detailed business content of a customer profile from a travel industry perspective. This specification provides a set of common messages for transmitting customer profile data that customers provide to travel services, and for the exchange of profile information between travel services within the industry. An example message schema is "OTA_ProfileCreateRQ.xsd"

These message schemas use several common types defined by OTA. Certain common types contain more basic common types. For example "OTA_HotelCommonTypes.xsd" contain "OTA_CommonPrefs.xsd", "OTA_CommonTypes.-xsd", and "OTA_VehicleCommonTypes.xsd".

3. SEMANTICS OF WEB SERVICES

WSDL (Web Service Description Language) specifies only the technical interface of the Web services. Web services, like their real life counterparts, may have many properties. Some of these properties such as the methods of charging and payment, the channels by which the service is requested and provided, constraints on temporal and spatial availability, service quality, security, trust and rights attached to a service can be generic. And such generic service semantics can be defined through DAML-S [3] (later OWL-S) upper ontology. Service discovery and composition through DAML-S have been addressed in the literature [15, 13]. However some other properties of the services depend on the application domain. For example, in the travel domain the properties of an "air reservation service" include the origin and destination cities. The domain specific semantics is necessary for the Web services in the following respects:

- First, in order to facilitate the discovery of the Web services, there is a need for an ontology to describe what the service does; that is, service functionality semantics in the domain. For example, in the travel domain, when a user is looking for a service to reserve a flight, he should be able to locate such a service through its meaning, independent of what the service

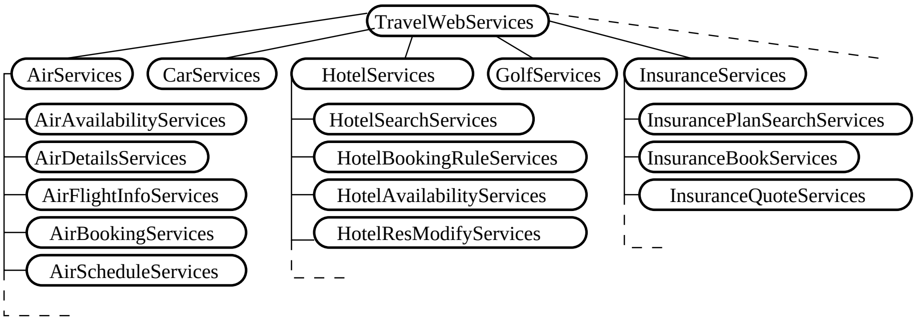

Figure 1: An OTA Compliant Service Functionality Ontology

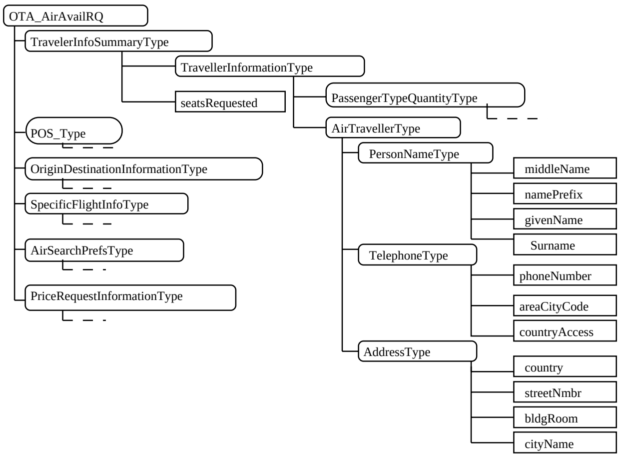

Figure 2: An Example OTA Compliant Service Message Ontology for Air Availability Request

is called and in which language, and who the business provider is. As previously noted, WSDL does not provide this semantic information.

- Service functionality semantics may suffice only when all the Web services use the same message standard like OTA in the travel domain. It is not realistic to assume all the travel Web services to be OTA compliant. Then, there is a need to transform one message format into another. Although using XML transformers like XSLT [24] may be an option for transforming one XML message into another, this approach has the following weaknesses when compared to ontology mapping: With the XSLT approach, there is a need for a hard coded mapping between two XML schemas. Ontology mapping, on the other hand, can be performed in a more generic way since it is aided by the meaning associated with the ontology languages. More specifically, ontology languages like RDFS [16], or OWL [19] express many of the important semantic information through universally accepted constructs: For example, the "subClassOf" relationship has a well defined meaning being a natural part of an ontology language. So as the "property" definition. Such definitions capture the semantic information in an application domain and prove to be useful in mapping different schemas. Although these constructs are expressed through XML eventually; in XML, they do not have this well-defined meaning.

  Hence we conclude that it is also necessary to define the semantics of messages through ontologies.

3.1 Web Service Functionality and Message Ontologies

We note that OTA specifications also reveal considerable domain knowledge and offer significant value in developing service functionality and service message ontologies. The OTA request/response pairs can be arranged into a class hierarchy, as shown in Figure 1 to define operation semantics of travel Web services. The advantage of having such a functionality ontology is twofold:

- All sorts of Web services can be classified by using the nodes of such an ontology to make their meaning clear. For example, a Web service instance "THY_Ucak_Rezervasyonu" can be classified with the "AirBookingService" node of the ontology to imply that it is an air reservation service.
- Web service instance discovery is facilitated. All the services classified through a node in the ontology can be retrieved from service registries as explained in Section 4.

It is also necessary to define the semantics of the messages exchanged so that the party receiving the message can interpret it. When ontologies are used to describe the messages, since the messages can refer to ontology concepts, it becomes possible to map one message instance into another through ontology mapping although they may be defined through different ontologies.

An example input message ontology for the "AirAvailability" generic Web service is shown in Figure 2. This message ontology is based on OTA. It should be noted that our aim is not to propose ontologies but to show how such ontologies, once developed, can be used for semantic mapping.

Figure 3 demonstrates a part of the ontology mapping which is later used to translate the instance given in Figure 4 to the instance given in Figure 5 through the MAFRA tool. MAFRA [12] uses a meta-ontology called Semantic Bridge Ontology (SBO) that defines the relations and transformations between ontologies. Semantic Bridges in SBO encapsulate the required information to translate one source entity (concept, relation, property) to a target entity. MAFRA has two primitive semantic bridges: Concept Bridge, and Property Bridge. A Concept Bridge defines the semantic equivalence between two ontology classes. At execution step, an instance concept of the target ontology is created for each source concept when the two concepts are related via a concept bridge. In the same way a Property Bridge defines the equivalence between source and target properties.

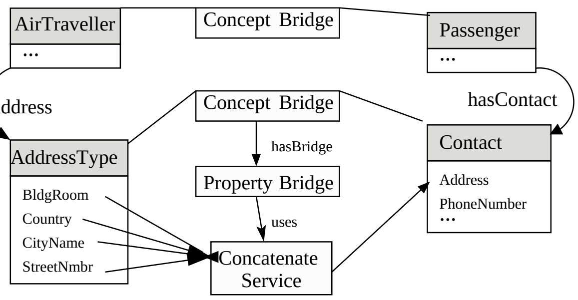

Figure 3: An Example Ontology Mapping through MAFRA

To clarify the issues involved, we provide an example: In Figure 3, a mapping fragment which constructs the “Address” attribute of the “Contact” concept using the “BldgRoom”, “CityName”, “Country” and “StreetNmbr” attributes of the “AddressType” concept is given. The “AddressType” and the “Contact” concepts are bound via a concept bridge which represents a semantic relation between them. The aim of the design is to concatenate the four attributes of “AddressType” into a single string and make this string the value of the “Address” attribute of “Contact”. This is accomplished by defining a property bridge that uses “Concatenate Service”. “Concatenate Service” takes the four attributes as input, and sets the “Address” attribute with the desired single string output.

```
<rdf:RDF xml:base="file:/C:/codes/mix/mapping/rdf/otasimple/ota1.rdf"
    xmlns:kaon="&kaon;"
    xmlns:rdf="&rdf;"
    xmlns:rdfs="&rdfs;"
    xmlns:a="&a;">

<a:AdressType rdf:ID="i-1072865188911-2105573014"
    a:BldgRoom="14/4"
    a:CityName="Ankara"
    a:County="Turkey"
    a:StreetNmbr="352"/>
<a:TelephoneType rdf:ID="i-1072865204895-1238306851"
    a:AreaCityCode="312"
    a:CountryAccessCode="90"
    a:PhoneNumber="2124029"/>
<a:PersonNameType rdf:ID="i-1072865222739-1263964887"
    a:GivenName="Mustafa"
    a:NamePrefix="Dr."
    a:Surname="Parlar"/>
<a:AirTravelerType rdf:ID="i-1072865240661-1116461751">
    <a:adress rdf:resource="#i-1072865188911-2105573014"/>
    <a:personName rdf:resource="#i-1072865222739-1263964887"/>
    <a:phone rdf:resource="#i-1072865204895-1238306851"/>
</a:AirTravelerType>
</rdf:RDF>
```

Figure 4: An example message ontology (source)

```
<rdf:RDF xml:base="file:/C:/codes/mix/mapping/rdf/otasimple/ota2.rdf"
    xmlns:rdf="&rdf;"
    xmlns:a="&a;">

<a:Country rdf:ID="i-1072869321442-1144005597"
    a:Name="Turkey"
    a:PhoneAccessCode="90"/>
<a:Contact rdf:ID="i-1072869321442-1759642294"
    a:Address="352 Street 14/4  Ankara / Turkey">
    <a:hasCountry rdf:resource="#i-1072869321442-1144005597"/>
</a:Contact>
<a:Passenger rdf:ID="i-1072869321442-2072327560"
    a:PassengerName="Dr. Mustafa Parlar">
    <a:hasContact rdf:resource="#i-1072869321442-1759642294"/>
</a:Passenger>
</rdf:RDF>
```

Figure 5: An example message ontology (target)

4. RELATING ONTOLOGIES WITH WEB SERVICE REGISTRIES

In relating the semantics with the services advertised in service registries, there are two key issues: the first one is where to store the ontologies. UDDI [17] does not provide a mechanism to store an ontology internal to the registry. ebXML [7], on the other hand, through its classification hierarchy mechanism allows domain specific ontologies to be stored in the registries. Note that for UDDI registries, domain specific ontologies 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.

The second key issue is how to relate the services advertised in the registry with the semantics described through an ontology. The mechanism to relate semantics with services advertised in the UDDI registries are the tModel keys and the category bags of registry entries. tModels provide the ability to describe compliance with taxonomies, ontologies or controlled vocabularies. Therefore if tModel keys are assigned to the nodes of the ontologies and if the services put the corresponding tModel keys in their category bags, it is possible to locate services conforming to the semantic given in a particular node of this ontology. This issue is elaborated in [4].

An ebXML registry [7], on the other hand, allows to define semantics basically through two mechanisms: first, it allows properties of registry objects to be defined through “slots” and, secondly, metadata can be stored in the registry through a “ClassificationScheme”. Furthermore, “Classification” objects explicitly link the services advertised with the nodes of a “ClassificationScheme”. This information can then be used to discover the services by exploiting the ebXML query mechanisms which is detailed in [5]. How to store ontologies in ebXML registries is described in [6].

5. SEMANTIC-BASED DISCOVERY OF SERVICE REGISTRIES IN P2P NETWORKS

Currently, the main mechanism to discover the Web services is searching the service registries. Therefore the consumer should first know where the related service registries are located.

We believe that the automation of the service discovery should be extended to include the discovery of service registries themselves automatically. For this purpose, we pro-

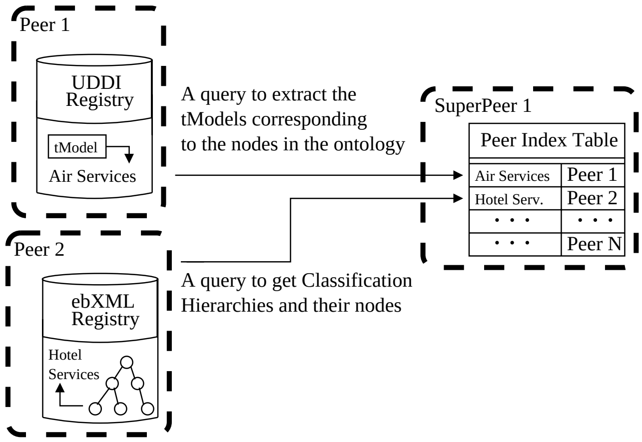

Figure 6: Exporting Web service Registry Semantics as Super Peer Indexes

pose to expose the semantic of Web service registries and connect the service registries through a peer-to-peer (P2P) network. With the semantics of service registries, we mean the semantics of Web services stored in these registries. The means to obtain this semantic information from UDDI and ebXML registries are as follows:

- As previously mentioned, in UDDI registries, the tModel keys are used to associate semantic with Web services.

  To be able to associate the semantics of the proposed travel ontologies to Web service instances, we first obtain a tModel key for each node of an ontology from a UDDI registry. Then, Web service instances are advertised by putting the tModel keys of related ontology nodes in their category bags.

  To discover the semantics associated with a service registry, that is, to obtain all the tModel keys used in a UDDI registry, the following command is used:

```
uddiProxy.find_tModel("%",null,null,null,0);
```

- In ebXML registries, the semantic is associated with Web services through the "ClassificationSchemes". In other words, by relating a Web service with a node in the classification hierarchy, we make the service an explicit member of this node and the service inherits the well-defined meaning associated with this node as well as the generic properties defined for this node. These class hierarchies are then exposed as the semantic information of Web service registries.

  In ebXML, the following query returns all the ClassificationSchemes in the registry:

```
<AdhocQueryRequest>
 <ResponseOption returnType = "LeafClass"/>
  <FilterQuery>
    <ClassificationSchemeQuery/>
 </FilterQuery>
</AdhocQueryRequest>
```

  And the following query returns all the ClassificationNodes in a given ClassificationScheme:

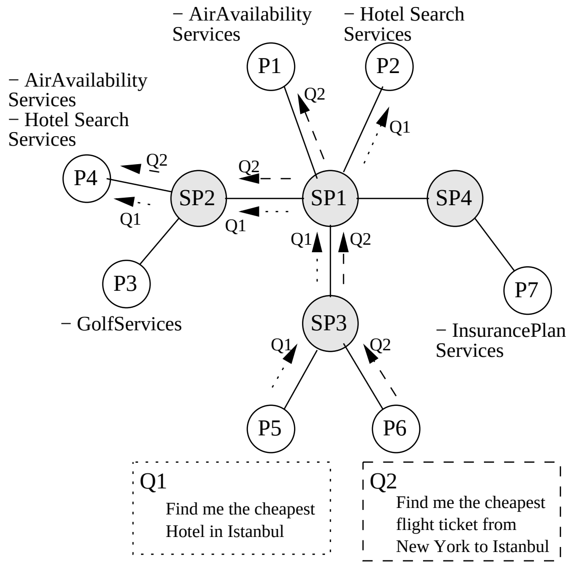

Figure 7: An Example Query Routing Based on Semantics

```
<AdhocQueryRequest>
 <ResponseOption returnType = "LeafClass"/>
  <FilterQuery>
   <ClassificationNodeQuery>
    <ClassificationSchemeQuery>
     <NameBranch>
      <LocalizedStringFilter>
       <Clause>
        <SimpleClause leftArgument = "value">
         <StringClause stringPredicate = "Equal">
           urn:ebxml:cs:myscheme
         </StringClause>
        </SimpleClause>
       </Clause>
      </LocalizedStringFilter>
     </NameBranch>
    </ClassificationSchemeQuery>
   </ClassificationNodeQuery>
 </FilterQuery>
</AdhocQueryRequest>
```

The next step is to use this semantic information for discovering the service registries through peer-to-peer networks. Discovering peers based on the semantic information has already been addressed in the literature, mainly within the scope of the Edutella project [14]. In Edutella, each peer registers the metadata of the resources it stores, i.e. the schema and attributes that are used to describe the stored content, to its super-peer [2]. These constitute the super-peer/peer (SP/P) indices. Edutella also introduces super-peer/super-peer routing indices (SP/SP) to forward queries among the super-peers. These SP/SP indices are essentially extracts and summaries from all super-peer local SP/P indices. Based on these indices, queries are first routed in the super-peer backbone, and then super-peers distribute them to the peers connected to them.

We use the same approach to facilitate the semantic discovery of service registries. That is, each peer registers the semantic information obtained from the service registries it owns as to its super peers as shown in Figure 6. The queries previously mentioned are used in extracting the semantic information from the registry. Besides service registries, individual peers can also advertise the semantics of the Web services they are hosting to their super peers. This facilitates P2P discovery of the web services that are not registered to any service registry. Once the semantics of the Web services both in service registries and individual peers are collected in the routing indices, this information is then used for semantic routing of the queries requesting a Web service given its semantics as shown in Figure 7. In this figure, P1 and P2 correspond to the UDDI and ebXML registry peer examples given in Figure 6. The figure shows the semantic category of the services it is hosting in reference to the nodes of the functionality ontology presented in Figure 1. Based on this information, the first query is routed to P4 and P2 by the super-peers, while the second query, which seeks flight availability information, is routed to P1 and P4.

6. CONCLUSIONS

The work described in this paper is being developed within the scope of the EU funded Satine project (IST-2104) [23] which is set up to develop and deploy semantically enriched services in the travel domain.

Using Web service technology will bring many advantages to the travel industry:

- The life of the existing software will be extended by exposing previously proprietary functions as Web services.
- Software development time will be reduced by wrapping already existing applications as Web services.
- Web services will provide interoperability both with internal and external applications.

We note that to exploit the Web services to their full potential, it is essential to describe both the functionality of a service and also the structure and semantics of the messages it carries. We show how the domain knowledge exposed by the Open Travel Alliance can be used to define *Service Functionality* and *Service Message* ontologies. A *Service Functionality* ontology describes the operational meaning of a Web service. A *Service Message* ontology is used in specifying the semantics of Web service messages.

We then define the semantics of a service registry to be the semantics of the Web services it contains. This semantics can easily be obtained from the registries by querying them. We show the queries obtaining the tModel keys in UDDI registries and classification hierarchies in ebXML registries describing the semantics of Web services. The semantics of service registries, in return, is used for semantic routing in peer-to-peer networks connecting the service registries.

7. REFERENCES

[1] Amadeus, http://www.amadeus.com/index.jsp

[2] Brunkhorst, I., Dhraief, H., Kemper, A., Nejdl, W., Wiesner, C., "Distributed Queries and Query Optimization in Schema-Based P2P-Systems", International Workshop On Databases, Information Systems and Peer-to-Peer Computing, Berlin, Germany, September 2003.

[3] DAML Services Coalition "DAML-S: Semantic Markup for Web Services", in Proceedings of the International Semantic Web Working Symposium (SWWS), July 2001.

[4] Dogac, A., Laleci, G. B., Kabak, Y., Cingil, I., "Exploiting Web Service Semantics: Taxonomies vs. Ontologies", IEEE Data Engineering Bulletin, Vol. 25, No. 4, December 2002.

[5] Dogac, A., Kabak, Y., Laleci, G., "A Semantic-Based Web Service Composition Facility for ebXML Registries", 9th Int. Conf. of Concurrent Enterprising, Finland, June 2003.

[6] Dogac, A., Kabak, Y., Laleci, G., "Enriching ebXML Registries with OWL Ontologies for Efficient Service Discovery", in Proc. of RIDE'04, Boston, March 2004.

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

[8] Galileo, http://www.galileo.com

[9] Galileo Web Services, http://xml.coverpages.org/GalileoGlobalWS.html

[10] Harmonise Project, IST-2000-29329, Tourism Harmonisation Network, http://www.harmonise.org/

[11] Harmonise Project, Deliverable 3.2: Semantic mapping and Reconciliation Engine subsystems, http://sourceforge.net/projects/hmafra

[12] Maedche, A., Motik, D., Silva, N., Volz, R., "MAFRA-A MApping FRAmework for Distributed Ontologies", In Proc. of the 13th European Conf. on Knowledge Engineering and Knowledge Management EKAW-2002, Madrid, Spain, 2002.

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

[14] Nejdl, W., Wolf, B., Staab, S., Tane, J., "EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network", Semantic Web Workshop, 11th Iternational World Wide Web Conference (WWW2002), Hawaii, USA, May 2002.

[15] Paolucci, M., Kawamura, T., Payne, T., Sycara, K., "Semantic Matching of Web Services Capabilities", in Proc. of Intl. Semantic Web Conference, Sardinia, Italy, June 2002.

[16] RDF Schema, http://www.w3.org/TR/rdf-schema/

[17] UDDI Registry, http://www-3.ibm.com/services/uddi/

[18] Open Travel Alliance (OTA), http://www.opentravel.org/

[19] Web Ontology Language OWL, http://www.w3.org/TR/owl-features/

[20] Peters, J., "Datalex: Web services in the Travel Industry", http://www.eyefortravel.com/papers/Web%20Services-%20in%20the%20Travel%20Industry%20(ATL-M0001.BB7).pdf

[21] Sabre, http://www.sabre.com

[22] Sabre Web Services, http://www.sabretravelnetwork.com/products_and-_services/travel_agencies/hardware_software/-sp2_007345.htm

[23] Satine Project, http://www.srdc.metu.edu.tr/webpage/projects/satine

[24] XSL Transformations (XSLT), http://www.w3.org/TR/xslt

[25] Worldspan, http://www.worldspan.com/home.asp?fPageID=2