CHAPTER 7

WEB SERVICES REPRESENTATION

This chapter deals with the web services representation system by annotating it with semantic descriptions. This chapter discusses about the annotation done and the domain specific ontology defined. This chapter also discusses about the discovery of web services.

7.1 INTRODUCTION

Semantic Annotation is being widely used which is about tagging the entities in the text and links with their semantic descriptions. Discovery of Web services which are not semantic, provides the results for client's requests, which may be inaccurate or less accurate. This is because it has only syntax descriptions in WSDL. Web Services should be integrated with semantic web by annotating the various semantic descriptions of web services in WSDL document. Semantically annotating WSDL file and relating the operations in WSDL to semantic concepts in OWL which describes the ontology system makes discovery of web services easier.

The semantic description of web services helps to provide the accurate and most relevant results to client's requests. Using SAWSDL instead of OWL and RDF-S to integrate the semantic web and web services, enhances the semantic annotation, service discovery, matching, composition, etc and helps in the research for automation of those processes.
Nikola Milanovic and Miroslaw Malek (2004) describe about web services discovery, composition and the need for non-functional requirements also.

7.2 SEMANTIC DISCOVERY OF WEB SERVICES

Discovery of web services can be divided into two sub processes namely functional matching and I/O parameter matching. Functional matching ensures that the functional requirements of client will be satisfied by discovered web services and I/O parameter matching ensures the same for I/O requirements. Using the SAWSDL, we can annotate the functional and I/O descriptions of web services in its WSDL documents. Semantic annotations of components are also discussed by Anne et al (2010). Such annotations will ease the discovery of web services which matches the functional and I/O requirements of client’s request. The function semantics is a kind of ontology concept that is suited with the function in service request and description, so degree of function matching can be calculated according to matching degree of two ontology concepts. Function matching includes category (C) and description (D) matching and its matching algorithm is defined. Web services should provide all the output parameters required by client and it should be able to operate with the set of input parameters provided by client. Such web services meant to satisfy input and output constraints of client’s request and those should be shortlisted for further processing. Weise et al (2008), Vadivelou (2010) discuss about service discovery that aids in service composition, agent based methods to compose web services respectively. Also, Carlos et al (2005), Jong Hyun Lim and Kyong-Ho Lee (2010) talks about how reusability of web service is possible and workflow techniques respectively.

Web service developers need to know about ontology languages (OWL/RDF) to map semantic concepts to their web services as dealt by Ge et al (2008). In addition to that, the discovery of Web Services using only the
functional and I/O matching satisfy only the functional requirements of clients and does not fulfill the client’s QoS expectations. Hence it is better to rank the discovered web services based on their various QoS values.

7.3 WEB SERVICES REPRESENTATION SYSTEM

WSDL documents of web services will be semantically annotated using SAWSDL and will be registered in UDDI. In this proposed system, UDDI registry contains the SAWSDL documents of registered web services which maps the web service concepts to the ontology concepts. The user interface gets the request from client and forwards it to the proposed system. The system discovers the web services which satisfies the functional and I/O requirements of client.

Then the QoS values of the discovered web services will be evaluated based on user preferences. Finally, the discovered web services will be ranked using the evaluated QoS values and suggested to the clients. Figure 7.1 illustrates the architecture of the web services representation system.

![Figure 7.1 Web Services Representation System](image-url)
7.3.1 Annotation of Web Services Using SAWSDL

Web services developed by service providers are syntactically described by WSDL document. For automated and efficient web service discovery, composition and matching, web services should be annotated with its semantic descriptions. Ahmed K et al (2003) talks about composing of web services for which the discovery of exact services is important and annotation is required. Using SAWSDL for the above mentioned purposes have some advantages such as that, service providers can annotate the Web services using the simple SAWSDL attributes and through external semantic model, service providers can choose the existing ontology model to annotate. SAWSDL, the latest after OWL-S and WSMO, refers the semantic ontology models in WSDL document by means of annotating those references. Figure. 7.2 shows the flow diagram of semantic annotation of web services using SAWSDL.

![Figure 7.2 Semantic Annotation of Web Services using SAWSDL](image-url)
Figure 7.3 shows the ontology graph for currency. Figure 7.4 shows the ontology of currency(currency.owl). It describes the ontology concepts of currency such as Indian_Rupee, United_States_Dollars, etc. The basics of domain ontology and how it improves the semantic search is stated by Amal Zouaq and Roger Nkambou (2008). The URI of those OWL concepts are used as value for modelReference attribute in WSDL document of currency converter. Figure 7.5 shows the WSDL document of currency converter(currency.wsdl) annotated using SAWSDL. For the operation named rupeetodollar, the input “r” is annotated with OWL concept “Indian_Rupee” and the output “d” is annotated with United_States_Dollars.

Figure 7.3 Ontology Graph for Currency
Figure 7.4 Ontology for Currency

```xml
<owl:ObjectProperty rdf:about="&currency;isEqualTo">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
  <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
</owl:ObjectProperty>
<owl:Class rdf:about="&currency;Euro">
  <rdfs:subClassOf rdf:resource="&currency;currency"/>
</owl:Class>
<owl:Class rdf:about="&currency;Indian_Rupees">
  <rdfs:subClassOf rdf:resource="&currency;currency"/>
</owl:Class>
<owl:Class rdf:about="&currency;United_States_Dollars">
  <rdfs:subClassOf rdf:resource="&currency;currency"/>
</owl:Class>
<owl:Class rdf:about="&currency;paise">
  <rdfs:subClassOf rdf:resource="&currency;Indian_Rupees"/>
</owl:Class>
```

Figure 7.5 Sample WSDL Annotated using SAWSDL

```xml
<xs:element name="rupeetodollar">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="r" type="xs:float" sawsdl:modelReference="http://www.semanticweb.org/ontologies/2011/10/currency.owl#Indian_Rupees"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="rupeetodollarResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="d" type="xs:double" sawsdl:modelReference="http://www.semanticweb.org/ontologies/2011/10/currency.owl#United_States_Dollars"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Figure 7.5 Sample WSDL Annotated using SAWSDL
7.3.2 Discovery of Semantic Web Services

Client's request regarding the web services will be forwarded by application interface (browser) to UDDI registry through the Simple Object Access Protocol (SOAP). Various matching models are discussed by Alireza Zohali and Kamran Zamanifar (2009). UDDI suggests the web services that satisfy the client's requirements. UDDI discovers those relevant web services by matching the function and I/O parameters with the help of registered WSDL documents. Thus the discovery of web services provides more relevant results and will be enhanced by using semantic descriptions annotated in WSDL documents using SAWSDL which maps the external domain ontology model with operation parameters in WSDL documents.

**Figure 7.6 Discovery of Web Services**

Figure 7.6 details the activities in discovery of web services. Since the function and I/O parameters will be in the ontology model, we can perform matching of those parameters of all registered web services with the
client's requests. From this, we can evaluate the degree of function matching and I/O matching for all web services and hence we can discover the web services satisfying functional and I/O requirements of client.

The details of web services are parsed from WSDL files and stored in database for further processing. The database has three tables namely services, io and qos. The table services has details such as web service id, service name, description, hits and address as shown in Figure 7.7. The table io contains details such as web service id, operation id, operation name, inputs and outputs. The table qos contains values of QoS parameters such as cost, response time, throughput, reliability and availability. When the client submits the request for web service, the system fetches the required details from database and discovers the web services that satisfy functional and I/O requirements of client.

![Figure 7.7 Screenshot of the Table “Services”](image)
For example, if the client submits the request with values such as currency (service), rupee (input(s)) and euro (output), the function matching retrieves the web services shown in Table 7.1 and the I/O matching discovers the web services shown in Table 7.2.

**Table 7.1 Results after Function Matching**

<table>
<thead>
<tr>
<th>Web service ID</th>
<th>Web service name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Currency</td>
<td>This is to convert the currency</td>
</tr>
<tr>
<td>2</td>
<td>Currencyconvert</td>
<td>This is to convert the money from one form to another</td>
</tr>
<tr>
<td>3</td>
<td>convertcurrency</td>
<td>This is to convert the money values</td>
</tr>
<tr>
<td>4</td>
<td>currencyconversion</td>
<td>Currency converter that is used to convert the money</td>
</tr>
<tr>
<td>5</td>
<td>Currencyconv</td>
<td>Money values are converted from one form to another</td>
</tr>
</tbody>
</table>

**Table 7.2 Results after I/O Matching**

<table>
<thead>
<tr>
<th>Web service ID</th>
<th>Web service name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Currency</td>
<td>This is to convert the currency</td>
</tr>
<tr>
<td>3</td>
<td>convertcurrency</td>
<td>This is to convert the money values</td>
</tr>
<tr>
<td>8</td>
<td>exchangecurrency</td>
<td>You can exchange the currency from one to another</td>
</tr>
</tbody>
</table>

As Web Services are developing into an important part in business process, there may exists many number of web services offering the same functionality and those web services discovered from the above process may differ in the various Quality of Services they provide. Discovered web
services may or may not provide the QoS expected by clients and this make
clients to be unsatisfied with the web services. Hence various parameters of
Quality-of-service to evaluate the web services discovered is considered and
by those evaluated values, the discovered web services are ranked. This helps
the service requesters to choose the web services which satisfy their
functional requirements and non-functional requirements such as Quality-of-
services.

QoS of the discovered web services for the complete satisfaction of
clients is emphasised. For this purpose, the web services are discovered based
on the various QoS parameters and is maintained in the QoS database storing
the values of various QoS parameters for all registered web services. The
various QoS parameters to be taken under consideration here are: Cost,
Response Time, Throughput, Availability and Reliability.

There is a need for QoS values in common measurement to rank the
discovered web services that satisfy functional and I/O requirements. Finally,
based on ranking, suggestion of discovered web services is done that
optimally satisfy the client’s QoS requirements.

7.4 SUMMARY

Using SAWSDL for the semantic annotation of web services,
enhances the automation of web service discovery, composition, etc. It is also
advantageous for service providers, as they can use language similar to
WSDL to describe semantics, interface, operation and other details. Also by
using external semantic models, service providers can choose their own
ontology to annotate. SAWSDL annotations are extensively used for an
efficient web services discovery by function matching and I/O parameters
matching with client's requirements. Since the number of web services
discovered by function matching and I/O matching, may be large and they all
differ in quality of services provided by them, it will be better to rank those web services based on QoS.

The next chapter deals with the service expectation of various QoS for web services. Hence, to fulfill the different client's Qos expectations the client preferences are obtained discovered, ranked and displayed to the user.