CHAPTER 6

SEMANTIC WEB SERVICE DISCOVERY MODEL

6.1 INTRODUCTION

A Semantic Service Discovery Model (SSDM) is used to discover the web services that are clustered semantically. The discovery also supports the backup services which are used in the case of composite web services. The discovery of backup services is essential in case any web service forming the composite web application fails leading to the failure of the entire application. The potential backup services can be an additional feature in the model that lists some of the web services that performs some or all of the operations performed by the discovered web service.

6.2 SYSTEMDESIGN

The system design is based on the problem description and the objectives mentioned above. A test bed has been designed to address the objectives of this research and Figure 5.1 shows the semantic service discovery model of the research. The following are the components used in the experimental test bed.
Service Provider: The service provider is the entity that has the web service and the service repository is maintained by the service provider. The service provider (SP1, SP2, SPn in the fig 6.1) is termed as the actual owner of the web service and he registers the service to the UDDI which can be used by the service requestors.

Figure 6.1 Semantic Service Discovery Model
**WSDL Parser:** The WSDL parser is used to parse the WSDL file for finding the service name, descriptions, operations and end point of the web service. The data obtained from the WSDL parser are tabulated in the UDDI Business Registry (UBR) which is then looked up during the discovery process. The WSDL parser implements the Algorithm 1 presented below which can be viewed as XML parsing.

---

**Algorithm 1** WSDL Parser

**In:** Service Descriptions $W_{ws}$

**Out:** Service Name $N_{ws}$, Operations $O_{ws}$ and Location $L_{ws}$

\[ O_{ws} = \text{null} \]

for each service $ws$ in $W_{ws}$

    get Service Name $N_{ws}$

    for each port $p$ in $ws$

        get Service Location $L_{ws}$

        get PortName, PortType

        for each Operation $op$ in PortType

            $O_{ws} = O_{ws} \cup OperationName$

    end

end

---
Business Interface: The business interface is used by the service providers for registering them as a service provider and to register the available service to the service registry. The business interface receives the service descriptions in the form of a WSDL document and the functionalities are obtained from the service descriptions.

UDDI Business Registry: The UDDI Business Registry (UBR) is responsible for creating the registry of web services. The service provider publishes the web service using the WSDL file along with the functional descriptions which are parsed by the WSDL Parser component and the service is registered with the UBR with its own description. Each entry in the UBR has its own service description and specifications which the service requestors search for finding the appropriate web services.

Clustering Module: The clustering module clusters the web services using concept lattices approach. A concept lattice behaves as a cluster of web services and the search process can be easily done in the lattice.

Web Service Interface: The web service interface is used by the service requestor or the client to search the services available in the service registry. The web service interface is the front end and the client communicates to the service registry using this component.

Query Analyser: The query analyser is an important component responsible for analysing the search query which is provided by the client and it classifies the
query into functional requirement and non-functional requirements which in turn are
given as inputs for the respective components.

*User Preference:* The user preference component is used to set the user
preferences to the concept lattice. The user preference component obtains the non-
functional requirements as the input from the *query analyser* component.

*Search Query:* The search query component obtains the functional
requirement from the *query analyser* and searches the sub lattice formed as a result
of applying the *user preferences* on the concept lattice and finally returns the web
service descriptions to the client.

6.3 USER PREFERENTIAL MODEL

A 2-tier *User Preferential Model* (UPM) is also proposed in order to improve
the user desirable search results. The model is designed to reduce the semantic gap
between the service request and the discovered service. Tier I deals with the
Qualification of Parameters while Tier II deals with Quantification of the Qualified
Parameters.

**Definition 2:** A User Preferential Model (UPM) can be described as follows,

\[
UPM: P_{\text{qualify}} \rightarrow P_{\text{quantify}}
\]  
(6.1)
where

- \( P_{qualify} \) represents the qualification of parameters
- \( P_{quantify} \) represents the quantification of the qualified parameters

The if-then relation between the two tiers indicates that the parameters must be qualified for quantification.

**Definition 3**: The quantification of parameters can be described as a tuple set containing

\[
P_{quantify} = \{SR_{id}, PR_{id}\} \quad (6.2)
\]

where

- \( SR_{id} \) is the unique identifier for the service requestor
- \( PR_{id} \) is the set of preferences for the particular service requestor and it can be defined as

\[
PR_{id} = \{P_1, P_2, \ldots, P_n\} \quad (6.3)
\]

where, \( P_i \) refers to any of the quality of the service parameter.

This work considers only five QoS parameter (n=5) values which are shown below:
• **Response Time**, which is the time duration between service users sending a request and receiving the corresponding response.

• **Availability**, which is the ratio of the number of successful invocations to total invocations.

• **Throughput**, which is the total number of invocations for a given period of time.

• **Reliability**, which is the ratio of the number of error messages to total messages.

• **Latency Time**, which is the time taken for the server to process a given request.

The user preferences are obtained from the user and a sub-lattice is formed based on the preferences from the clustered lattice. This will reduce the semantic gap and the time for discovery. The sub-lattice formed from the clustered lattice contains the services and operations that can be relevant to the user as the preferences will be set by the user. The semantic gap between the user request and the discovered web services can be reduced to a large extent using this approach.

### 6.4 SEMANTIC DISCOVERY MODEL

The clustering is based on lattices, as they are easy to implement and provide the hierarchical relation among the services. The semantic service discovery model can be viewed as a sub-lattice search on a graph. The SSDM can be realised if the
clustering technique is based on lattices. The lattices can be represented using graphs and it will be easy to search on graphs rather than a vast search space. The output of lattice clustering approach gives us a lattice that is a cluster of web services based on its operation and ontology. The lattice can also be used to find the potential backup services for the discovered web service. Thus a lattice based clustering will enable semantic service discovery. The Algorithm 2 depicts the depth first search algorithm used in this research.

**Algorithm 2** Sub-LatticeSearch(L\_SUB, n)

**In:** Sub-Lattice L\_SUB, Query q, Parameters p

**Out:** Service Name N\_ws

1. Label n as explored
2. For all links l in AdjacentLinks of L\_SUB do
   1. If link l is not explored then
      1. Get the N\_ws of node n with q and p
      2. If node of N\_ws is not explored then
         1. If SemSim evaluates to TRUE then
            1. Call Sub-LatticeSearch(L\_SUB, node of N\_ws)
            2. EndIf
         Else
         1. Label node of N\_ws as a back link
         2. EndIf
      3. EndIf
   2. Else
      1. Label node of N\_ws as a back link
      2. EndIf
   3. EndIf
3. EndFor

114
The classical depth first search algorithm can be used to search the sub-lattice as lattices can be represented using graph data structure. The links are explored from the sub-lattice recursively and the proposed semantic similarity measure is evaluated for calculating the relevance, which discovers the desirable web services as per the user request. Thus the proposed framework can enable the discovery of user desired results.

6.5 SUMMARY

To summarize the chapter, a semantic service discovery model with a novel semantic similarity measure has been proposed. A two-tier user preferential model has also been used to discover user desirable services from the quality point of view. The user preferences are obtained from the user and a sub-lattice is formed based on the preferences from the clustered lattice. This will reduce the semantic gap and the time for discovery. The sub-lattice formed from the clustered lattice contains the relevant services and operations since the preferences are set by the user. The semantic gap between the user request and the discovered web services can be reduced as a result of this approach.