A Survey on Approach for Service Selection
Mr. Yatin R Belani¹ Mr. Siddharth S. Patel²
¹,²Department of Computer Engineering
¹Shree Pandit Nathulalji Vyas Technical Campus, India ²U.V.Patel College Of Engineering Ganpat University, Kherava, India

Abstract— Now a days there are lots of approaches for service selection but there are lots of methods are available mainly there are two approaches SOAP and Restful web service selection by which we can select the service. Here the survey is available by which the service selection process is executing in a different ways. SOA architecture is there for selection of the services. Now a days Restful architecture is used for the service selection. After that there is also ontology is available for selection of services. Use of LOD is for easier for selection in web services.

Key words: restful web service, service selection, different approaches for service selection, LOD(linked open data)

I. INTRODUCTION
First of all there is a service broker is there who works as intermediate between client and server. After that the OWL is introduce for selection of services. OWL is used to find the services which matches with the user requirements.

After that proxy server are introduced between client and server. And this the evolution is there for selecting the web services which will as explain below.

First of all we will see the SOA architecture.

A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed.

Fig. 1:
RESTful Web services are a lightweight alternative to the heavy, SOAP-based standards. In RESTful web services, the emphasis is on simple point-to-point communication over HTTP. REST is an architectural style where distributed systems are built on a shared model and have agreement between nouns (resource names as URIs), verbs (HTTP methods used), and content types (usually XML or SOAP Example JSON).

RESTful web services are based on the way how our web works. Our very own world wide web (www) – the largest distributed application – is based on an architectural style called REST – Representational State Transfer. REST is neither a standard nor a protocol. It is just an architectural style like say for example client-server architecture (client-server is neither a standard nor a protocol). Web services following this architectural style are said to be RESTful Web services.

Fig. 2: RESTFULL Web Service (Architectural Approach)
REST Web service follows four basic design principles: 

- Use HTTP methods explicitly.
- Be stateless.
- Expose directory structure-like URIs.
- Transfer XML, JavaScript Object Notation (JSON), or both

Here is the comparison of SOAP vs Restful Web services

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abbreviation</td>
<td>Simple Object Access Protocol</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>2</td>
<td>Developer View</td>
<td>Object oriented</td>
<td>Resource oriented</td>
</tr>
<tr>
<td>3</td>
<td>Standard base</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Security</td>
<td>SSL, WS Security</td>
<td>SSL</td>
</tr>
<tr>
<td>5</td>
<td>Transaction</td>
<td>WS Automatic Transaction</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>Reliability</td>
<td>WS reliable messaging</td>
<td>Application specific</td>
</tr>
<tr>
<td>7</td>
<td>Performance</td>
<td>Good</td>
<td>Better</td>
</tr>
<tr>
<td>8</td>
<td>Caching</td>
<td>No</td>
<td>Yes (GET method can cache)</td>
</tr>
<tr>
<td>9</td>
<td>Message Communication</td>
<td>XML</td>
<td>HTML,XML,JSON</td>
</tr>
<tr>
<td>10</td>
<td>Message Size</td>
<td>Heavy &amp; SOAP with WS markup</td>
<td>Lightweight &amp; no XML markup</td>
</tr>
<tr>
<td>11</td>
<td>Message Encoding</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>Service Description</td>
<td>WSDL</td>
<td>No formal description</td>
</tr>
<tr>
<td>13</td>
<td>Orientation</td>
<td>Wrap business logic</td>
<td>Access resource</td>
</tr>
</tbody>
</table>
A. Web Service

Varieties of definitions of web services are given by researchers and web service consortia. According to World Wide Web Consortium: “A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.” Web Services are self-independent application that exhibits modular and distributed concepts [2]. Web services can be used by any application irrespective of platform in which it is developed. Web service description is provided in WSDL document, it can be accessed from internet using SOAP protocol. The primary aim of Web services is to demystify and normalize application interoperability within and across establishments, leading to growth in operational efficiencies and intimate partner relations. In industry, many applications are built by calling different web services available on the internet which results in overwhelming acceptance of web services in recent years and the trend will continue for many years to come.

B. Web Service Selection

Web service selection refers to the process by which a service implementation is chosen from numerous services discovered in response to requester’s functional requirement. Service discovery is a prerequisite for service selection process; however, service selection is a core issue that must be addressed in order to retrieve appropriate service for a requester. Functional and Non-Functional properties especially QoS are the two main classes of requirements that are considered in selecting optimal service for a requester. Much work has been done in the domain of web service discovery, which mainly focusses on functional properties of web services. However, in view of large number of services with comparable functionalities, web service discovery alone is inadequate for selecting optimal service that would satisfy users’ expectations, hence; efficient methodologies and procedures are required for appropriate web service selection, which is the main concern in the domain of service oriented computing.

Complex information may be built aggregating simpler information units, but unlike the current Web paradigm, which conceives complex information as a whole, the information units are individually addressable and linkable. Principles of linked data are:[10]

- Use URI as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using standards (RDF, SPARQL).
- Include links to other URIs, so that they can discover more things”.

In the first principle the term “things” is used, but “resources” would be more appropriate. Indeed, Uniform Resource Identifiers (URIs) are the recommended identifiers. In the second principle, a constraint is put on the URI scheme, since HTTP URIs can be dereferenced (i.e. used as reference to retrieve the resource representation) by exploiting existing Web technologies. The third principle states that the information the URI refers to has to be meaningful and useful, suggesting for this purpose the use of RDF and SPARQL standard technologies. Finally, the fourth principle specifies that the resources have to be connected to other resources to support browsing and discovery.

Linked Data use URIs as names for things; use HTTP URIs so that people can look up those names; when someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL) and include links to other URIs, so that they can discover more things.

A common misapplication of both approaches is to assume semantics (or abuse implied semantics) encoded in a URI, when both REST and Linked Data explicitly expect clients to regard URIs as opaque strings when used for identification. In this way both follow the principle of separating identification from the semantics of interaction, description, and structure.

Linked Data services, in implementing the HTTP range, semantics to the content negotiation to distinguish between URIs that is non-information resources (identifiers for conceptual or real-world objects) and URIs that are information resources (documents) that describe the non-information resources.

Linked Data over-loads URI usage so that it is also a mechanism for retrieving triples describing that resource (in a document, i.e. an information resource)

The registry needs to pre-filter services only based on categorization of the Web services and pass the semantic requirements to all Web services that are registered within the required category. Finally, the registry selects or orchestrates Web services based on Web services’ semantic responses about whether they are qualified to the requirements.

With the potential of LOD service-oriented architecture can use the dataset directly to develop semantic services rather than to add semantic value later. In fact, LOD has been proposed as an approach for publishing and describing services, namely linked services and Linked Open Service.

<table>
<thead>
<tr>
<th>14</th>
<th>Simplicity</th>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Transport Protocol Support</td>
<td>HTTP, SMTP</td>
<td>HTTP</td>
</tr>
</tbody>
</table>

Fig.3: architecture using web agent

Linked Data is a new Web concept that promotes a paradigm shift in how information is modeled and accessed.
II. CONCLUSION

Today the number of users are increasing on the internet. So after studying this paper we can say that by using the Restful architectural approach we can provide better and efficient as well as appropriate web service selection to the client as well as we can also say that LOD is very easy to store and retrieve so we can get the better response time.

REFERENCES

[1] Vikas Sitapara, ”RESTful Web Services Discovery and Composition”
[2] Mohammed Husain Bohara, ”RESTful Web Service Integration”
[5] Dr. Iavarasan Egambaram, G.Vadivelou, S.Prasanth Sivashubramanian “Qos based web service selection”