International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 3 Issue: 2 25-Aug-2014,ISSN_NO: 2321-3337

BASIC APPROACHES TO SEMANTIC WEB SERVICES LOAD ONTOLOGY

[1] K.RAJA, [2] D.ANUSUYA.

[1]Assitant Professor, Dept.of.Computer science, MCC college.Pattukottai.

[2]Research Scholar, Dept.of.Computer science, MCC college.Pattukottai.

ABSTRACT−*Bootstrapping ontology based on a set of predefined textual sources, such as Web services, must address the problem of multiple concepts that are largely unrelated. This paper exploits the advantage that Web services usually consist of both WSDL and free text descriptors. It has been recognized that due to the autonomy and heterogeneity of Web services and the Web itself, new approaches should be developed to describe and advertise Web services. The most notable approaches rely on the description of Web services using semantics. Several approaches have been driving the development of Semantic Web Service frameworks such as OWL-S (Ontology Web Language for Services), WSMO (Web Service Modeling Ontology) and IRS-III (Internet Reasoning Service).*

Keywords—SWS,IRS,OWL- WSMF, WSMO

I. INTRODUCTION

Ontologies are used in an increasing range of applications, notably the Semantic Web, and essentially have become the preferred modeling tool. However, the design and maintenance of ontologies is a formidable process. Ontology bootstrapping, which has recently emerged as an important technology for ontology construction, involves automatic identification of concepts relevant to a domain and relations between the concepts. A key problem with the use of standards for Web Service description and publishing is that the syntactic definitions used in these descriptions do not completely describe the capability of a service and cannot be understood by software programs. Semantic Web Services are the result of the evolution of the syntactic definition of Web Services and the semantic Web. The Web service ontology bootstrapping process proposed in this paper is based on the advantage that a Web service can be separated into two types of descriptions:

i)The Web Service Description Language (WSDL) describing “how” the service should be use.

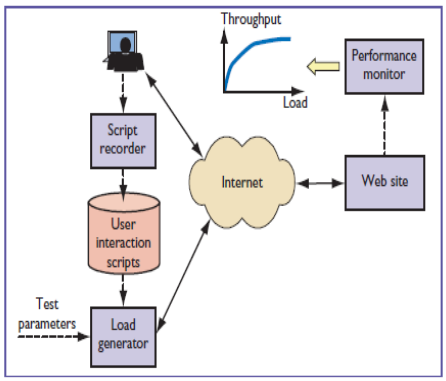
ii) A free text description of the Web service describing "what" the service does. This advantage allows bootstrapping the ontology based on WSDL and verifying the process based on the Web service free text descriptor.

II.EXISTING SYSTEM:

The Semantic Web is a set of technologies for representing, and publishing on the Web, computer-interpretable structured information. Semantic Web Services is a research field that endeavours to apply these technologies to the description and use of Web Services. Whereas interoperability is the primary motivation for Web Services, automation of information use and dynamic interoperability are the primary objectives of Semantic Web Services.

III.PROPOSED SYSTEM:

Another key QoS factor for Web and e-commerce sites is *response time*. With Web-based environments, you must measure end-to-end response time to determine how customers perceive such things as page download and keyword search times. Also, when defining end-to-end response time, you must distinguish between the download time for the base HTML page and that for other page components, such as images and ad banners. A customer's perception of a Web application's response time varies according to many different factors — some of which are outside your site's environment. These factors include your site's ISP, the customer's ISP, the customer's ISP-connection bandwidth, which networks route packets from the customer to your Web site, and the delays imposed by your Web site's thirdparty
services.



Failure to mimic real user behavior can generate totally inconsistent results. Because customers who abandon a session use fewer site resources than those who complete it, for example, planning your infrastructure capacity assuming that all started sessions will be completed can lead you to overprovision the site.

TOOL REQUIREMENTS

There are several important requirements to consider when buying a load-testing tool. Here, I discuss them in relation to the four layers of the hierarchical framework for e-business.3 The top hierarchy level is the *business model*, which describes the business type product type; the revenue-generating model; business policies; pricing; and intellectual property considerations. At this level, you need load-testing tools that help you:

1. Track revenue throughput and potential lost
2. revenue throughput,
3. Carry out load tests under the most realistic and thorough conditions possible to avoid over- and under provisioning the IT infrastructure.
4. Understand how business decisions — such as marketing campaigns or new business models— affect the IT infrastructure.

The next framework level is the *functional model*, which deals with e-business functions that implement the site's business model. Such functions might include browse, search, select, and add to shopping cart. At this level, you need tools that help you:

- Load test functions supported by many different technologies, including Flash, JavaScript, ActiveX, cookies, and SSL;
- Perform load testing for the functions you offer wireless clients.
- Account for functions that use streaming
- media.

To evaluate the relation between concepts, we analyze the overlapping context descriptors between different concepts. In this case, we use descriptors which were included in the union of the descriptors extracted by both the TF/IDF and Web context methods. Precedence is given to descriptors which appear in both concept definitions over descriptors which appear in the context descriptors. In our example, the descriptors related to both *Domain* and *Domain Address* are: *Software*, *Registration*, *Domain*, *Name*, and *Address*. However, only the *Domain* descriptor belongs to both concepts and receives the priority to serve as the relation. The result is the relation which can be identified as a subclass, where *Domain Address* is a subclass of *Domain*.

The process of analyzing the relation between concepts is performed after the concepts are identified. The relation described in the example is based on descriptors which are the intersection of the concepts. Basing the relations on a minimum number of Web services belonging to both concepts will result in a less rigid classification of relations. The process is performed iteratively for each additional service which is related to the ontology. The iterations stop once all the services are analyzed. Alternatively, an ontology administrator can decide to suspend the ontology evolution at any given time.

V. CONCLUSIONS

This paper proposes an approach for bootstrapping an ontology based on Web service descriptions. The approach analyzes Web services from multiple perspectives and integrates the results. It has been concluded that for application communication in business environments through Web Services communication,

description, and discovery features are covered, respectively, by SOAP, WSDL, and UDDI standard languages do not allow to perform some complex tasks, as integration or semantic Matching of services. High-volume Web sites are becoming more complex due to several factors, including the use of third-party services, such as CDNs and ad networks, geographical distribution and duplication, streaming media features, and wireless access.

VI. REFERENCES

[1] Berners-Lee, T., Hendler, J., Lassila, O.: *The semantic Web.* Scientific American, May 2001.

[2] McIlraith, S., Son, T.C., Zeng, H. *Semantic web services*. IEEE Intell. Syst. 16(2), 46–53 (2001).

[3] D.A. Menascé and V.A.F. Almeida, *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*, Prentice Hall, Upper Saddle River, N.J., 2000.

[4] P.J. Denning and J.P. Buzen, "The Operational Analysis of Queuing Network Models," *ACM Computing Surveys*, vol.
10, no. 3, Sept. 1978, pp. 225-261.