ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Web Services: An Overview

Sonia Jasrotia

M Tech (CSE), Guru Nanak Dev University Regional Campus, Gurdaspur, India

Abstract: Service-Oriented Computing (SOC) has emerged as an vital computing model and changed the way of designing, delivered and consuming of software. In SOC, services are used as fundamental elements to support fast, low-cost development of distributed applications in heterogeneous environments. Web services technology is the most capable choice to implement service oriented architecture and its strategic objectives, it can provide a flexible solution to the problem. With the help of WSDL, SOAP, and UDDI, Web services are becoming popular in Web applications In this paper, a survey on web services has been done.

Keywords: Web services, atomic, composite web services, web service model, development life cycle.

I. INTRODUCTION

Web services technology is the most capable selection to implement service oriented architecture and its strategic objectives. A Web service is fundamentally a semantically well defined abstraction of a set of computational or physical activities which involving a number of resources, which are helpful for accomplish a business necessity or a customer requirement. A Web service is interacted through Internet based protocols and described, advertised and revealed using standard based languages and. Nowadays, two types of Web services are most popular and widely used is SOAP based Web services and RESTful Web services. With the technology of Web services, enterprises are able to represent their internal business processes as services and make them available via the Internet. Nowadays, many big companies have offered Web services to provide simple access to some of their resources, companies such as Google, Amazon, Twitter, and Facebook. A Web service is an interface that describes a collection of operations that are network accessible through standardized XML messaging. A Web service is described using a formal, standard XML notion, which are known as its service description. Service description covers all the details which are necessary to interact with the service, including message, location and transport protocols. The interface hide the implementation details of the service,that interface allowing it to be used independently of the software or hardware platform on which it is implemented and also independently of the programming language in which it is written.

Atomic vs. composite Web services

There are two types of web services, atomic and composite. We differentiate between atomic and composite Web services. An atomic service which is also called elementary service, is an access point to an application that does not depend on another Web service to fulfill user requests. Each and every atomic service provides a programmatic interface based on SOAP and WSDL which provides the services to user. For inheritance applications such as those written in CORBA, suitable adapters can be developed so that they can be invoked as Web services. A composite service is a combination of other composite and atomic web services, its just like an umbrella structure that brings together other composite and atomic services. They are collaborate to implement a set of operations. The services bring together by a composite service are referred to as its component services. An example of a composite service would be a travel preparation service, integrating services for booking flights, booking hotels, searching for attractions, etc. Whether atomic or composite, a Web service is identified by an identifier (e.g., URL), a set of attributes, and a set of operations. The attributes of a service provide information, which is useful for the potential consumers.

II. THE WEB SERVICE MODEL

The Web Services architecture is based upon the interactions between three roles: service provider, service registry and service requestor. The interactions involves between these roles are the publish, find and bind operations. Together, these operations and roles act upon the Web Services artifacts: the Web service software module and its description. In a usual scenario, a service provider hosts a network accessible software module or we can say an implementation of a Web service. The service provider in this model defines a service description for the Web service and publishes it to a service registry or service requestor. The service requestor in this model used a find operation to retrieve the service description locally or from the service registry and uses the service description to bind with the service provider and invoke or interact with the implementation.

Fig 1: Web Services roles, operations and artifacts

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

2.1 Roles in a Web Services Architecture

- *Service provider.* Service provider provide the services and define the service description. From a business point of view, this is the owner of the service. From an architectural point of view, this is the platform that hosts access to the service.

- *Service requestor.* The service requestor in this model used a find operation to retrieve the service description locally or from the service registry. From a business point of view, this is the business that require many functions to be satisfied.

  From an architectural point of view, this is the application that is looking for and initiating or invoking an interaction with a service.

  The role of the service requestor can be played by a browser driven by a person or a program without a user interface, for example another Web service.

- *Discover agencies.* This is a searchable engine of service descriptions where service providers publish their service descriptions. Service requestors find services and attain binding information for services during execution for dynamic binding or during development for static binding.

2.2 Operations in a Web Service Architecture

To take advantage of Web Services, For an application, three behaviors must take place: publication of service descriptions, lookup or finding of service descriptions, and binding or invoking of services based on the service description.

These behaviors can take place individually or iteratively. In detail, these operations are:

- *Publish.* To be easily access able, a service description needs to be published so that the service requestor can find it easily. Publishing of the service can be different depending upon the requirements of the application.

- *Find***.** In the find operation, the service requestor retrieves a service description directly or queries the service registry for the type of service.

  The find operation can be concerned in two dissimilar lifecycle phases for the service requestor: at runtime to retrieve the service's binding and location description for invocation and at design time to retrieve the service's interface description for program development.

- *Bind.* Eventually, a service needs to be invoked. In the bind operation the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact and invoke the service.

2.2 Artifacts of a Web Service

- *Service.* Where a Web service is an interface described by a service description, its implementation is the service. A service is a software module deployed on network accessible platforms provided by the service provider. It exists to be invoked by or to interact with a service requestor. It can also function as a requestor, using other Web Services in its implementation.

- *Service Description*. The service description contains the details of the interface and implementation of the service. This includes its data types, operations, binding information and network location. It could also include categorization and other metadata to facilitate discovery and utilization by service requestors. The service description might be published to a service requestor or to a service registry.

III. WEB SERVICES DEVELOPMENT LIFECYCLE

The Web Services development lifecycle includes the design, deployment, and runtime requirements for each of the roles: service registry, service provider and service requestor. Each role has specific requirements or need for each element of the development lifecycle.
The development lifecycle can have four phases:

1. Build

The build phase of the lifecycle includes development and testing of the Web service implementation, the definition of the service interface description and the definition of the service implementation description. Web service implementations can be provided by creating new Web Services, composing new Web Services from other Web Services and applications. And transforming existing *applications into Web Services.*

2. Deploy

The deploy phase includes the publication of the service interface and service implementation definition to a service requestor or service registry and deployment of the executables for the Web service into an execution environment, its just like a Web application server.

3. Run

During the run phase, the Web service is available for invocation. At this point, the Web service is fully deployed, operational and network accessible from the service provider. Now the service requestor can perform the find and bind operations.

4. Manage

The manage phase covers ongoing management and administration of the Web service application.

The manage phase of the lifecycle provides security, availability, performance, quality of service and business processes must all be addressed.

DOI 10.17148/IJARCCE.2015.4441

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

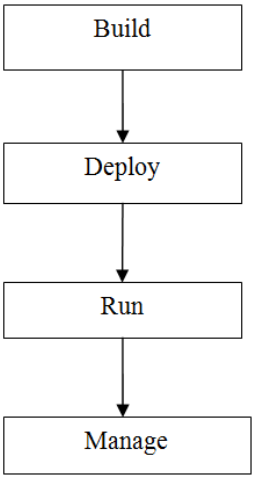

Fig 2: Development lifecycle of web services.

IV. WEB SERVICES BENEFITS

Web services provide numerous technological and business profit, which include:

- Application and data integration
- Versatility
- Code re-use
- Cost savings

1. The inherent interoperability feature that comes with using language independent technologies, vendor, platform, and the ever present HTTP as a transport mean that any application can communicate with any other application using Web services. The user only require the WSDL to successfully exchange the data with the service and it does not care about that in what format its underlying data is stored and no needs to know that how the implementation is done by others. These benefits allow organizations to integrate different applications and data formats with no difficulty.

2. Web services are also flexible by design. Via a Web based client interface Web services can be accessed by humans or they can be accessed by any other Web services and other applications. A client can also combine data from multiple Web services even if the systems themselves are unsuited. For the reason that the systems exchange information via Web services, a change to the sales record, for example, it will not change the service itself.

3. Code re-use is another benefit of Web services' interoperability and flexibility. One service can be utilized by numerous clients, all of which utilize the operations provided to accomplish dissimilar business objectives. Instead of having to create a custom service for each single requirement, portions of a service are simply re-used as necessary.

All these benefits helps for to maximize the cost savings. Simple interoperability means for integrating data, which is necessary to create highly customized applications, which can be expensive, is removed. Existing investments in systems infrastructure and development can be utilized easily and combined to add additional values. Since Web services cost is low because it based on open standards and the associated learning curve is smaller than that of many solutions. As a final point, Web services get advantage of ever-present protocols and the Web infrastructure that already exists in every organization, therefore they need little if any additional technology investment.

V. LITERATURE REVIEW

Chunguang Li and Philip K Maini (2005), have theoretically analyzed the scaling properties of the network by using a mean-field approach and proposed an evolving network model with community structure. The analytical and numerical results indicate that the network can produce community structure.

Filippo Radicchi et al. (2003), has discussed about the communities, community detection algorithm. He studied the Divisive and agglomerative algorithms for detecting the communities in the complex network.

Gunce Keziban Orman and Vincent Labatut (2011), have discussed the properties of complex network and compare the five community detection algorithms by using a set of artificial networks.

Hongbing Wang et al. (2004), have discussed some problems of web services and also described theor solutions.They studied the web service difficulty in using the distributed technologies, web service Security Problems and Solutions, Composition Problems and Solutions.

Jinghai Rao and Xiaomeng Su, this paper gives an overview of recent progress in automatic Web services composition. They propose a five-step model for Web services composition process. The composition model consists of service presentation, translation, process generation, evaluation and execution. They concentrate on the methods of composite Web services process generation and they discussed the introduction and comparition of selected methods to support this step.

M. E. J. Newman (2006), has discussed about the community detection methods, the methods of optimal modularity, further techniques of modularity maximization and the methods for dividing the network into more than two communalities.

Michelle Girvan and M. E. J. Newman (2001), have discussed the traditional community detecting algorithms and also discussed the new ”edge betweenness” algorithm for detecting the community structure on computer-generated graphs and on real-world networks.

Nan Du, Bai Wang and Bin Wu (2008), have proposed a new method Com Tector for the community detection in complex networks. This algorithm is based on the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

overlapping nature of cliques in real world, this algorithm can be applied to many large sparse graph.

Quan Z. Sheng et al. (2014), have studied about the state of the art of Web services composition. They abstract a generic model for the life cycle of Web services composition, which is used to compare different research prototypes based a set of assessment criteria. They compared number of Web services composition standards and services composition platforms.

Steve Harenberg et al. (2014), have discussed the community detection algorithms for overlapping and disjoint community detection on large scale real world networks. The algorithms evaluated by measuring the structural properties of the communities and their performance. Their results show that two types of measures are not equivalent.

V. Gabrel , M. Manouvrier, C. Murat (2014), have discussed about the workflow based WS composition problem and present the complexity analysis of QoS (quality of service)aware workflow based WS composition.

Xizhe Zhang et al. (2013), have studied about the web services, complex structure, community structure and the community structure of structural service networks formed by public web services available on the Internet.

V. CONCLUSION AND FUTURE SCOPE

In this paper, a survey on various complex networks has been done. From the survey, it has been found that the useage of communicate discovery during classification and retrieval of cloud web services has been ignored. Moreover the effect of community discovery on the different natured users i.e. those who demands multiple things at a time has also ignored.

Therefore in near future, the enhancement of the cloud web services can be done further by using the nodes i.e. high end server and respective servers based upon the community structure to improve the results.

REFERENCES

[1]. 1] Chunguang Li and Philip K Maini (2005), "An evolving network model with community structure**",** journal of physics a: mathematical and general, doi:10.1088/0305-4470/38/45/002.
[2]. [2] Steve Harenberg, Gonzalo Bello, L. Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan and Nagiza Samatova (2014), "Community detection in large-scale networks: a survey and empirical evaluation", doi: 10.1002/wics.1319.
[3]. [3] Filippo Radicchi, Claudio Castellano†, Federico Cecconi, Vittorio Loreto†, and Domenico Parisi (2004). "Defining and identifying communities in networks",PNAS , vol.101, no. 9, 2658–2663, March 2, 2004. Nan Du, Bai Wang and Bin Wu (2008), "Community detection in complex networks". Journal of computer science and technology, Vol.23, No.4.
[4]. Mohd Hilmi Hasan, Jafreezal Jaafar, Mohd Fadzil Hassan (2012), "Monitoring web services' quality of service: a literature review", Springer, DOI 10.1007/s10462-012-9358-7.
[5]. M. E. J. Newman (2006), "Modularity and community structure in networks", PNAS, vol. 103, no. 23, 8577–8582.
[6]. M. E. J. Newman (2008), "Detecting community structure in networks", University of Michigan, Ann Arbor, MI 48109-1120.
[7]. M. Girvan and M. E. J. Newman (2002), "Community structure in social and biological networks", PNAS, vol. 99, no. 12, 7821–7826.
[8]. Quan Z. Sheng, Xiaoqiang Qiao, Athanasios V. Vasilako, Claudia Szabo, Scott Bourne, Xiaofei Xu (2014), "Web services composition: A decade's overview", ScienceDirect, http://dx.doi.org/10.1016/j.ins.2014.04.054.
[9]. Xizhe Zhang, Shuai Feng, Ying Yin, Bin Zhang (2013), "Community Discovery of Public Cloud Web Services based on Structural Networks" IEEE, 978-1-4673-4714-3/13.
[10]. Andrea Lancichinetti and Santo Fortunato (2009), "Community detection algorithms: A comparative analysis", DOI: 10.1103/PhysRevE.80.056117.
[11]. V. Gabrel, M. Manouvrier, C. Murat (2014), "Web services composition: Complexity and models", ScienceDirect, http://dx.doi.org/10.1016/j.dam.2014.10.020.
[12]. Jierui Xie, Stephen Kelley, Boleslaw K. Szymanski (2013), "Overlapping Community Detection in Networks: The State of the Art and Comparative Study", ACM Computing Surveys, Vol. 45, No. 4, Article 43.
[13]. E. J. Newman, "The structure and function of complex networks", Department of Physics, University of Michigan, Ann Arbor, MI 48109, U.S.A.
[14]. Santo Fortunato, "Community detection in graphs", Complex Networks and Systems Lagrange Laboratory, ISI Foundation, Viale S. Severo 65, 10133, Torino, I-ITALY.
[15]. Günce Keziban Orman and Vincent Labatut (2009), "A Comparison of Community Detection Algorithms on Artificial Networks" Springer LNAI 5808, pp. 242–256, 2009.
[16]. Natali gulbahce and sune lehmann, "The art of community detection", Center for Complex Networks Research, Northeastern University, Boston, MA 02115, USA.
[17]. Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, Ploutarchos Spyridonos (2011), "Community detection in Social Media Performance and application considerations", Springer, DOI 10.1007/s10618-011-0224-z.
[18]. "Web Services Conceptual Architecture (WSCA 1.0)" by Heather Kreger IBM Software Group, (2001).
[19]. Erin Cavanaugh, "Web services: Benefits, challenges, and a unique, visual development solution".
[20]. Esther Galbrun, Aristides Gionis, Nikolaj Tatti (2014), "Overlapping community detection in labeled graphs", Springer, DOI 10.1007/s10618-014-0373-y.
[21]. http://perso.uclouvain.be/vincent.blondel/research/louvain.html.
[22]. Hongbing Wang, Joshua Zhexue Huang, Yuzhong Qu, Junyuan Xie, Web services: Problems and Future Directions", Department of Computer Science and Technology Nanjing University,Nanjing 210093, China.
[23]. Jinghai Rao and Xiaomeng Su, "A Survey of Automated Web Service Composition Methods".