HCTL Open International Journal of Technology Innovations and Research (IJTIR)
http://ijtir.hctl.org
Volume 16, July 2015
e-ISSN: 2321-1814, ISBN (Print): 978-1-943730-43-8

Web Services: Trends & Evolution

Meenu[1]

myself_meenu@yahoo.co.in

Abstract

Web services offer a new paradigm for distributed computing and system collaboration. Recently, Web services have generated great interests in both vendors and researchers. Web services, based on existing Internet protocols and open standards, can provide a flexible solution to the problem of application integration. With the help of WSDL, SOAP, and UDDI, Web services are becoming popular in Web applications. The objective of this paper is to survey major Web service standards in order to assists adapting organization in identifying the interrelationships among these standards and relevancy of these standards to a particular WS implementation approach.

Keywords

Web services, SOAP, WSDL, UDDI, Web Services deployment.

Introduction

Web Services

Web Services refers to the technologies that are used for making connections. Whatever we connect together using Web Services is Service. A service is the endpoint of a connection. The combination of services in an organization constitutes a service-oriented architecture. Web service is created to integrate heterogeneous applications [1]. A web service resides on the web server and is available for use over the internet. After creation, a web service is uploaded on a web host. The web host will make available the Web service, to anyone over the internet.

Service-Oriented Architecture (SOA)

A service-oriented architecture is essentially a collection of services. These services communicate with each other [2].

A. Services

Whatever we connect together using Web Services is Service. A service is the endpoint of a connection. The figure 1 illustrates a basic service-oriented architecture. It shows a service consumer sending a service request message to a service provider. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined such that is understandable to both the service consumer and service provider. A service provider can also be a service consumer.

[1]Assistant Professor, Department of Computer Science & Engineering, M.M.M. University of Technology, Gorakhpur (Uttar Pradesh), India.

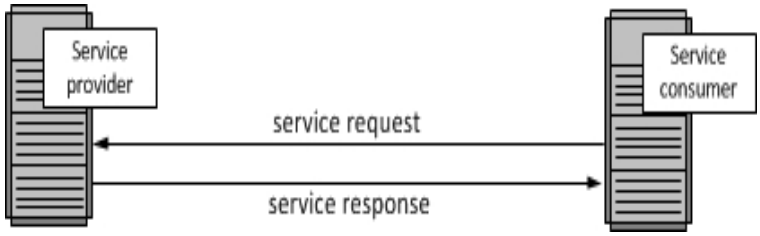

Fig.1. Service-oriented architecture

B. Prior Service-Oriented Architecture Specifications

Service-oriented architectures are not a new terms .Before the arrival of Web services, other technologies and architectures existed that met the functionality of today's web service. The first service-oriented architecture in the past was DCOM or Object Request Brokers (ORBs) based on the CORBA specification.

C. COM

Microsoft COM (Component Object Model) technology in the Microsoft Windows-family of Operating Systems enables software components to communicate [20]. COM is used by developers to create re-usable software components, link components together to build applications, and take advantage of Windows services.

D. DCOM

DCOM is the acronym for the Distributed Component Object Model, an extension of the Component Object Model (COM). DCOM was introduced in 1996 and is designed for use across multiple network transports, including Internet protocols such as HTTP. It works mainly with Microsoft Windows. DCOM, its two main disadvantages were it works properly only with binaries written with Microsoft languages and platforms. Many firewalls blocked such traffic, further impeding the use of DCOM and CORBA beyond the enterprise which resulted in poor performance.

E. CORBA

CORBA is the acronym for Common Object Request Broker Architecture. It was developed under the auspices of the Object Management Group (OMG). It is middleware. A CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network. Any organization today works with multiple vendors, suppliers, contractors and other entities. Each of these entities would have developed their own software systems based on Microsoft or on other technologies. Each of these software systems would have been developed over period of time with hundreds of thousands of dollars investments. It will be almost impossible for any of them to change their systems for compatibility.

There are many compelling reasons for to use Web Services. The first reason for using Web Services is that it allows us to communicate among all different entities without affecting their existence. The second reason why we use Web Service is that, it uses text based protocol (XML) that all applications can understand and firewalls do not block Text information. So, the popular way to represent data on the Internet is XML .The third reason for using Web services is that they do not rely on special protocols. Web services communicate using WWW (World Wide Web). They rely on standard Internet protocols HTTP and SOAP which are present in every system.

e-ISSN: 2321-1814, ISBN (Print): 978-1-943730-43-8

Web Services Specifications

Three specifications for Web Services are: SOAP, REST, and JSON [5].

SOAP (Communication)

Simple Object Access Protocol (SOAP) is an XML protocol. SOAP, which was initially created by Microsoft and later developed in collaboration with Developmentor, IBM, Lotus, and UserLand[4],[10]. SOAP, was originally part of the specification that included the Web Services Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI)[6],[7],[11]. It is used now without WSDL and UDDI. The interaction is illustrated in the figure 2.

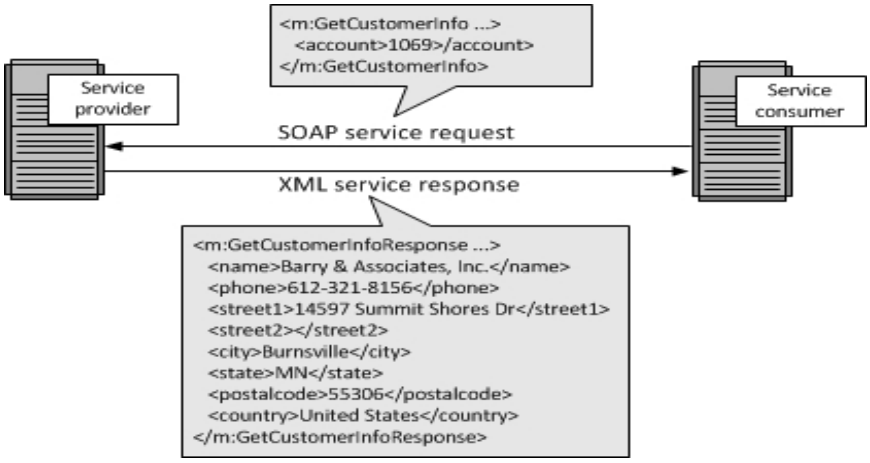

Fig 2: SOAP Interaction

SOAP provides the envelope for sending Web Services messages over the Internet/Internet. It is part of the set of standards specified by the W3C. SOAP is an alternative to Representational State Transfer (REST) and JavaScript Object Notation (JSON).

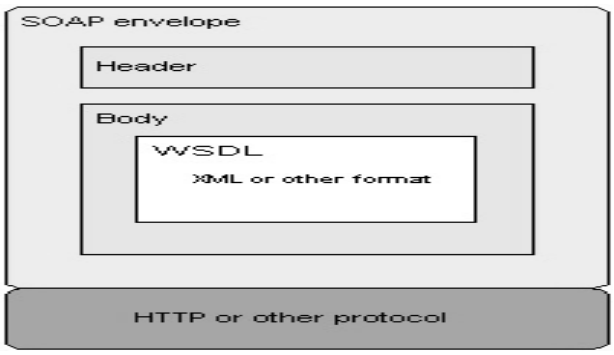

Fig 3: SOAP Envelope

The SOAP envelope contains two parts as illustrated in figure 3:

i. An optional header provides information on authentication, encoding of data, or how a recipient of a SOAP message should process the message.
ii. The body contains the message. These messages can be defined using the WSDL specification.

SOAP commonly uses HTTP, but other protocols such as Simple Mail Transfer Protocol (SMTP) may by used. SOAP at one time stood for Simple Object Access Protocol. Starting with SOAP Version 1.2, the letters in the acronym have no particular meaning.

Representation State Transfer (REST)

Representation State Transfer (REST) appeals to developers because it has a simpler style that makes it easier to use than SOAP [8],[9]. It is also less verbose so less volume is sent while communicating. The interaction is illustrated in the figure 4.

Fig 4: REST Interaction

Representational State Transfer (REST) is a style of architecture based on a set of principles that describe how networked resources are defined and addressed. These principles were first described in 2000 by Roy Fielding. REST is an alternative to SOAP and JSON.REST is a style of software architecture as opposed to a set of standards. So it is sometimes referred to as RESTful or REST-style applications or architectures. REST is a popular choice for implementing. It is one of the options for Amazon Web Services.

JavaScript Object Notation (JSON)

Both SOAP and REST use XML for interchange, JavaScript Object Notation (JSON) uses a subset of JavaScript. This is illustrated in the figure 5. JavaScript Object Notation (JSON) uses name/value pairs which is similar to the tags used by XML. JSON is an alternative to SOAP and REST.

There is no best option for Web Services. If we use multiple service providers, it is possible that we will be using all three Web Services specifications: SOAP, REST, and JSON.

Web Services Description Language (WSDL) (Description)

The Web Services Description Language (WSDL) forms the basis for the original Web Services specification. WSDL was developed by IBM and Microsoft. WSDL is a format for describing a Web Services interface[3]. It is a XML language that describes a Web service and how they should be bound to specific network addresses. The figure 6 illustrates the use of WSDL.

The steps involved in providing and consuming a service are:

i. A service provider describes its service using WSDL. This definition is published to a repository of services.
ii. A service consumer issues one or more queries to the repository to find a service and decides how to communicate with that service.
iii. WSDL provided by the service provider is sent to the service consumer. This tells the service consumer about the requests and responses for the service provider.
iv. The service consumer uses the WSDL to send a request to the service provider.
v. The service provider provides the response to the service consumer.

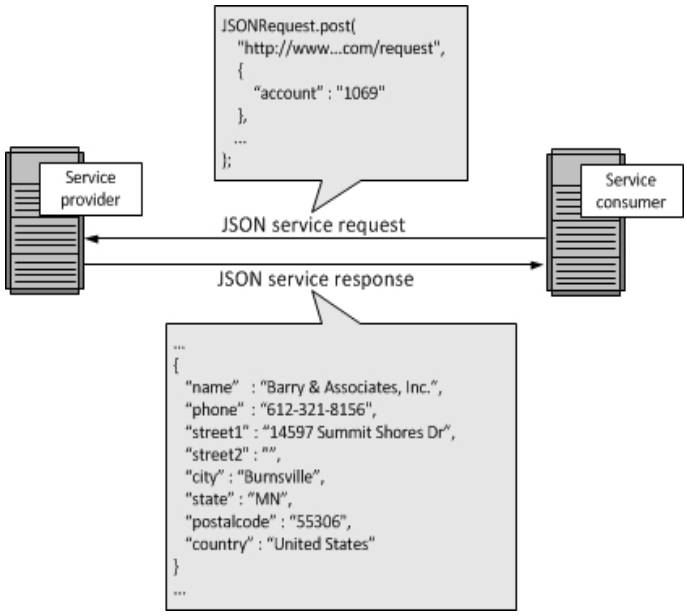

Fig.5: JSON Interaction

WSDL has three parts:

i. Definitions
ii. Operations
iii. Service bindings

Definitions are generally expressed in XML and include both data type definitions and message definitions that use the data type definitions. XML, however, is not necessary required for definitions. The OMG Interface Definition Language (IDL), for example, could be used instead of XML.

Operations describe actions for the messages supported by a Web service. There are four types of operations:

i. One-way: Messages sent without a reply required
ii. Request/response: The sender sends a message and the received sends a reply.
iii. Solicit response: A request for a response
iv. Notification: Messages sent to multiple receivers. Operations are grouped into port types. Port types define a set of operations supported by the Web service.

Service bindings connect port types to a port. A port is defined by associating a network address with a port type. A collection of ports defines a service.

Universal Description, Discovery, and Integration (UDDI) (Discovery)

UDDI was first developed by UDDI.org and then transferred to OASIS. It is based on a common set of industry standards, including HTTP, XML, XML Schema, and SOAP. The repository shown in the figure 6 could be a UDDI registry. The UDDI registry was intended to eventually serve as a means of "discovering" Web Services described using WSDL. An alternative to UDDI is the ebXML Registry.

UDDI provides the definition of a set of services supporting the description and discovery of

i. Businesses, organizations, and other Web Services providers,
ii. The Web Services they make available, and
iii. The technical interfaces which may be used to access those services.

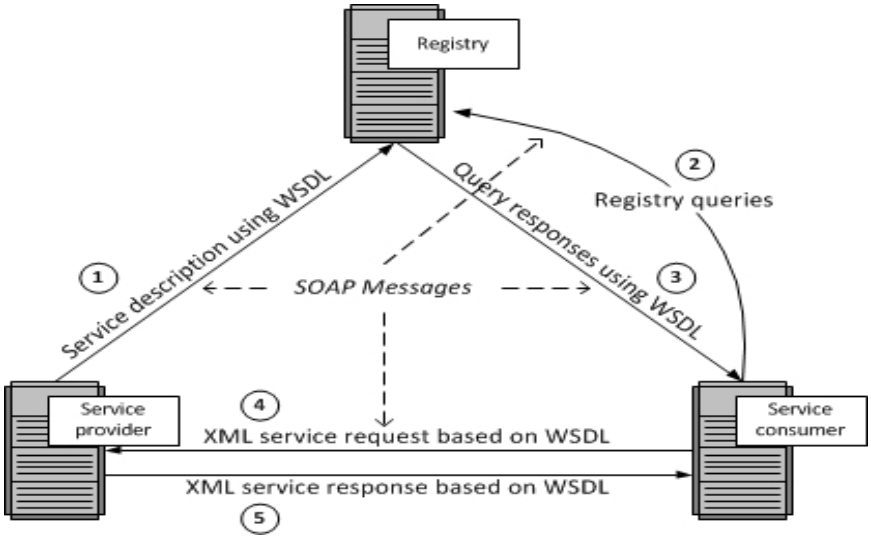

Fig.6: WSDL Interaction

The UDDI Business Registry system consists of three directories:

i. **UDDI white pages:** It consists of basic information such as a company name, address, and phone numbers, as well as other standard business identifiers.
ii. **UDDI yellow pages:** It consists of detailed business data, organized by relevant business classifications according to the newer NAICS (North American Industry Classification System) codes, as opposed to the SIC (Standard Industrial Classification) codes.
iii. **UDDI green pages:** It consists of information about a company's key business processes, such as operating platform, supported programs, purchasing methods, shipping and billing requirements, and other higher-level business protocols.

Tools to Create a Web Service

Runtime environment

A Web service runtime environment provides a set of runtime libraries that implement and support the SOAP specification. A web services platform includes development and deployment tools.

Web services Development and Deployment tools

Development tools are used to generate SOAP code for both clients and services. They are used to package and configure the runtime settings for the applications.

Examples of web service platforms are: Microsoft Visual Studio.NET and IBM WebSphere Studio Application Developer (WSAD). Web services platforms also include command line tools.

Development tools are:

i. WSDL generators
ii. Code generators
iii. UDDI browsers
iv. Command line tolls
v. Graphical tools

Web Services Deployment tools are:

i. Packaging tools
ii. Configuration tools
iii. UDDI registration tools

Testing a Web Service

A Web Service does not have an interface for user to test. But the CLR will create a small interface, which looks like a site that enables the user to test each method.

Testing the web services .Net

Testing a web service is checking whether the .asmx file we create using visual studio.net is bug free and without any errors

Deploy a Web Service

After we create a web service, we deploy it using a FTP program. A Web service exists as an .asmx file. So, it cannot be used as we use a .aspx page. This is because we cannot implement a user interface for a web service. The output of a web service can be read by applications not by humans.

Commonly, we deploy three files when uploading web services.

i. .asmx file
ii. .disco file
iii. .XSD files - XML Schema Definition files

If a web service has dependencies in the form of dynamic link libraries (*.dll), these files also needs to be deployed.

i. **.asmx file:** A .asmx file is the entry file for a web service. The functionality of the web service application is listed in the .asmx file
ii. **.disco file - Discovery file:** disco files are also known as discovery files and it helps developers to find web services which are available on a particular web site

Conclusion

Web services are being increasingly adopted by organizations in order to run their business more effectively and efficiently. However, current technologies lack the support required by such organizations. The success of web services lies in their reliability, especially when economic interests are involved. Web service technologies still need time to mature and require more research to realize their ultimate potential

References

1. Alonso, G., Casati, F., Kuno, H., Machiraju, V. Web Services: Concepts, Architectures and Applications. Berlin: Springer-Verlag., 2004.
2. Issarny, V., Georgantas, N., Hachem, S., Zarras, A., Vassiliadis, P., Autili, M., Hamida, A. B. Service-oriented middleware for the Future Internet: state of the art and research directions.Journal of Internet Services and Applications, 2(1), 23-45. doi: 10.1007/s13174-011-0021-3,2011.
3. WS-CDL. Web Services Choreography Description Language (WS-CDL) Candidate Recommendation. Retrieved April 30, 2008, from http://www.w3.org/TR/ws-cdl-10/(2005, 9 November).
4. SOAP-Primer. (2007). SOAP Version 1.2 Part 0: Primer Retrieved February 2, 2011, from http://www.w3.org/TR/soap12-part0/
5. Srini Narayanan, Sheila McIlraith, Simulation, Verification and Automated Composition of Web Services, Proc. of the Eleventh International World Wide Web Conference, Honolulu, Hawaii, USA, May 7-11, 2002.
6. Eric Newcomer, “Understanding Web Services: XML, WSDL, SOAP, and UDDI,”Addison-Wesley Professional (May 23, 2002).
7. David Booth, Hugo Haas, Francis McCabe, et al., “Web Services Architecture,” http://www.w3.org/TR/ws-arch March 20, 2011.
8. Potti, Pavan Kumar, Sanjay Ahuja, Karthikeyan Umapathy, and Zornitza Prodanoff. "Comparing Performance of Web Service Interaction Styles: SOAP vs. REST." In Proceedings of the Conference on Information Systems Applied Research ISSN, vol. 2167, p. 1508, 2012.

9. Wang, Hongbing, Joshua Zhexue Huang, Yuzhong Qu, and Junyuan Xie. "Web services: problems and future directions." Web Semantics: Science, Services and Agents on the World Wide Web 1, no. 3 (2004): 309-320.
10. Xu, Tao, and Chunxiao Yi. "SOAP-Based Security Interaction of Web Service in Heterogeneous Platforms." Journal of Information Security 2 (2011).
11. Mogha, Rashim, and V. V. Preetham. Java Web services programming. John Wiley & Sons, Inc., 2002.
12. The World Wide Web Consortium: http://www.w3.org/
13. W3C Web Services Activity: http://www.w3.org/2002/ws/
14. WSDL Primer: http://www.w3.org/TR/wsdl20-primer/
15. Web Services Interoperability Organization: http://www.ws-i.org/
16. Organization for the Advancement of Structured Information Standards: http://www.oasis-open.org/
17. W3 Schools: http://www.w3schools.com/
18. http://www.vkinfotek.com
19. http://www.microsoft.com
20. http://www.service-architecture.com