ISSN (Online) 2249 - 054 X

International Journal of Computing and Corporate Research

Specialized and Refereed Journal for
Research Scholars, Academicians, Engineers and Scientists

INTERNATIONAL MANUSCRIPT ID : ISSN2249054X-V2I3M4-052012

VOLUME 2 ISSUE 3 MAY 2012

INTERRELATIONS AMONG MAJOR WEB TECHNOLOGIES HTTP, WEB SERVERS AND WEB SERVICES

Khushali M. Tirghoda,
IITE, Ahmedabad, Gujarat

Abstract— The affiliation between HTTP, Web Server and Web Services has been a heterogeneous set of functionalities and exchanges of information. Each elemental plays a momentous role in the thousands of functions users can access and utilize on the Internet. HTTP acquiesce users to interact with Web Servers and access information via the Internet. Web servers serve data and files to users who request them. Web Services reserves cross-system, cross-language communication among various kinds of machines and enable inter-business transaction. Although each technology works on its own and performs many useful functions, it is the amalgamation of these technologies that has created the dynamic functionalities of the Web that are available today. This paper will explore the interdependence between HTTP, Web Servers and Web Services technologies and to expose the existing function on to network, connecting Different Applications i.e. Interoperability, Standardized Protocol & Low Cost of communication.

Index Terms— SOAP, UDDI, WSDL, WSIL, Web Service, Web Server, XML

1 INTRODUCTION

In this paper, I have described the inter-relationships between HTTP, Web Servers and Web Services technologies which have expedited the functionalities and convenience of the Web with Web Technologies Architecture in Detail to understand its working.

2 HTTP

HTTP is the standard protocol currently used to access the Internet. According to the World Wide Web Consortium, HTTP “is an application-level protocol for distributed, collaborative, hypermedia information systems.” It is a very lucid protocol that accommodates raw data to be transferred across the Internet. From this simple data transfer protocol, users of the Internet can easily perform functions and give commands to the Web Servers through a Graphic User Interface as a Web page viewed through a browser and not worry about the specific details of how the command is going to be transferred or interpreted by the computers involved. HTTP allows such exchange of information between the user’s computer and the Web Server to take place increasingly and efficiently.

HTTP is a stateless protocol as each command is executed independently, without any knowledge of the commands that came before it which is the main reason that it is difficult to implement Web sites that react intelligently to user input.

2.1 HTTP Request

When a Web browser loads a Web page, it opens a connection to the Web server and sends an HTTP request to the Web server. A typical HTTP request looks like this:

```
GET/index.htmlHTTP/1.1
Host:www.google.com
Accept-Encoding:gzip
User-Agent: Firefox/3.6
```

With this request, the Web browsers asks for the object "/index.html" on host "www.http-compression.com". The browser identifies itself as "Firefox/1.0", the HTTP protocol version used here is 1.1 and claims that it can understand HTTP responses in gzip format.

2.2 HTTP Response

After parsing and processing the client's request, the Web server may send the HTTP response in compressed format. Then a typical HTTP response looks like this:

```
HTTP/1.1 200 OK
Server:Apache
Content-Type:text/html
Content-Encoding:gzip
Content-Length:26395
```

With this response, the Web server tells the browser with status code 200 that he could fulfil the request. In the next line, the Web server identifies itself as Apache. The line "Content-Type" says that it's an HTML document. The response header "Content-Encoding" informs the browser that the following data is compressed with gzip. Finally, the length of the compressed data is stated.

3 WEB SERVERS

A Web Server is the heart system of Web Site. The Web Server hosts both the elemental of a Web page such as the actual Web page HTML files, CSS files and templates and all other constitutive technologies that make a Web site function the way it does. Although all Web servers function similarly, the set up and the way a server could be set can deviate exceptionally.

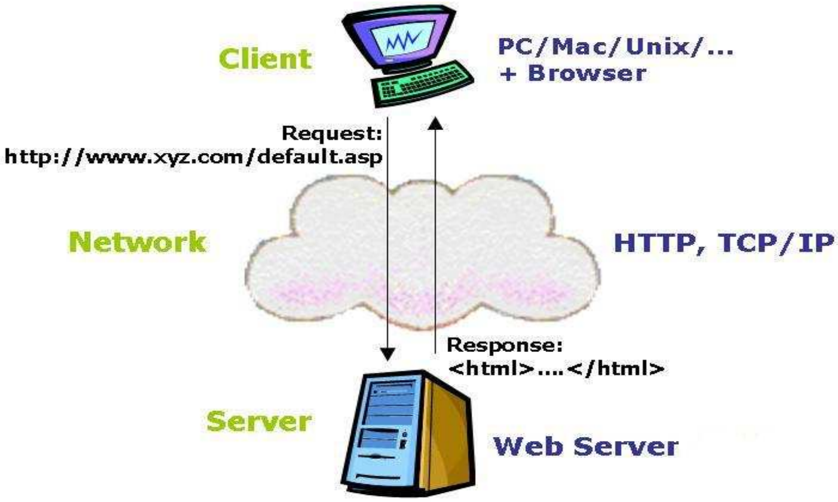

Fig 1: HTTP & Web Server Affiliatio

The Client-Server network is a highly centralized network system with one central computer as the server. This set up is easy to manage and secure. Yet, maintaining a centralized network craves tremendous amount of resources ranging from manpower to hardware. With client/server computing, end users can handle a broad range of information processing tasks. This included data entry, inquiry response, updating databases, and providing decision support.

4 WEB SERVICES

Web services are software components that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging.

Web services are designed to be accessed by other applications and vary in complexity from simple operations, such as checking a banking account balance online, to complex processes running CRM (customer relationship management) or enterprise resource planning (ERP) systems. Since they are based on open standards such as HTTP and XML-based protocols including SOAP and WSDL, Web services are hardware, programming language, and operating system independent. This means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranets or the Internet using Web services

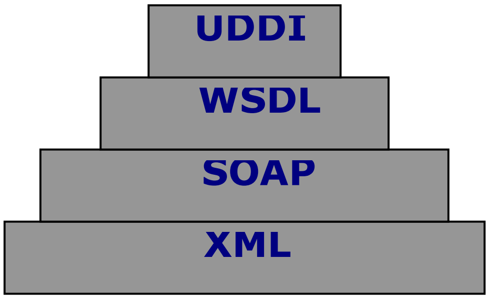

Fig 2: Layers of Web Service Architecture

Web services are powered by XML and three other core technologies: WSDL, SOAP, and UDDI. Before building a Web service, its developers create its definition in the form of a WSDL document that describes the service's location on the Web and the functionality the service provides. Information about the service may then be entered in a UDDI registry, which allows Web service consumers to search for and locate the services they need. This step is optional but is beneficial when a company wants its Web services to be discovered by internal and/or external service consumers. Based on information in the UDDI registry, the Web services client developer uses instructions in the WSDL to construct SOAP messages for exchanging data with the service over HTTP.

More about these core technologies is detailed below.

4.1 Working of Web Service

- A service broker that acts as a look up service between a service provider and a service requestor.
- A service provider that publishes its services to theservice broker.
- A service requester that asks the service broker where to find a suitable service provider and that binds itself to the provider.

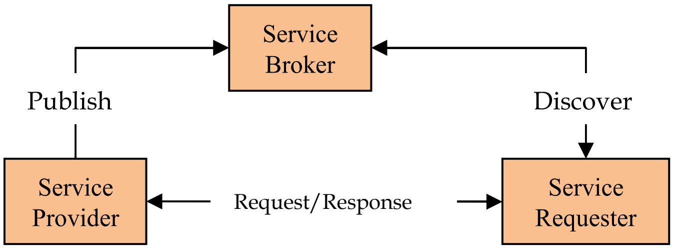

Fig 3: Interaction of Services

The standard specifications that are created for web services fall into following categories:

4.1.1 XML (Extensible Markup Language)

XML is a W3C (World Wide Web Consortium) specification that defines a meta-language for describing data. In XML applications, data is described by surrounding it with customizable, text-based tags that give information about the data itself as well as its hierarchical structure.

Because XML syntax consists of text-based mark-up that describes the data being tagged, it is both application-independent and human readable. This simplicity and interoperability have helped XML achieve widespread acceptance and adoption as the standard for exchanging information between heterogeneous systems in a wide variety of applications, including Web services.

XML forms the basis for all modern Web services, which use XML-based technologies to describe their interfaces and to encode their messages. WSDL, SOAP, and UDDI all use XML-based messaging that any machine can interpret.

4.1.2 WSDL (Web Services Description Language)

Also maintained by the W3C, WSDL is an XML-based format for describing Web services. Clients wishing to access a Web service can read and interpret its WSDL file to learn about the location of the service and its available operations.

In this way, the WSDL definition acts as the initial Web service interface, providing clients with all the information they need to interact with the service in a standards-based way. Through the WSDL, a Web services client learns where a service can be accessed, what operations the service performs, the communication protocols the service supports, and the correct format for sending messages to the service.

A WSDL file is an XML document that describes a Web service using six main elements:

- Port type – groups and describes the operations performedwe2w3 by the service through the defined interface.
- Port – specifies an address for a binding, i.e., defines a communication port.
- Message – describes the names and format of the messages supported by the service.
- Types – defines the data types (as defined in an XML Schema) used by the service for sending messages between the client and server.
- Binding – defines the communication protocols supported by the operations provided by the service.
- Service – specifies the address (URL) for accessing the service.

The WSDL document that describes a Web service acts as a contract between Web service client and server. By adhering to this contact the service provider and consumer are able to exchange data in a standard way, regardless of the underlying platforms and applications on which they are operating.

4.1.3 SOAP (Simple Object Access Protocol)

SOAP is an XML-based protocol from the W3C for exchanging data over HTTP. It provides a simple, standards-based method for sending XML messages between applications. Web services use SOAP to send messages between a service and its client(s). Because HTTP is supported by all Web servers and browsers, SOAP messages can be sent between applications regardless of their platform or programming language. This quality gives Web services their characteristic interoperability.

SOAP messages are XML documents that contain some or all of the following elements:

- **Envelope** – specifies that the XML document is a SOAP message;encloses the message itself.
- **Header (optional)** – contains information relevant to the message, e.g., the date the message was sent, authentication data, etc.
- **Body** – includes the message payload.
- **Fault (optional)** – carries information about a client or server error within a SOAP message.

Data is sent between the client(s) and the Web service using request and response SOAP messages, the format for which is specified in the WSDL definition. Because the client and server adhere to the WSDL contract when creating SOAP messages, the messages are guaranteed to be compatible.

4.1.4 UDDI (Universal Description Discovery and Integration)

UDDI is a standard sponsored by OASIS (Organization for the Advancement of Structured Information Standards). Often described as the yellow pages of Web services, UDDI is a specification for creating an XML-based registry that lists information about businesses and the Web services they offer.

UDDI provides businesses a uniform way of listing their services and discovering services offered by other organizations. Though implementations vary, UDDI often describes services using WSDL and communicates via SOAP messaging.

Registering a Web service in a UDDI registry is an optional step, and UDDI registries can be public or private (i.e. isolated behind a corporate firewall). To search for a Web service, a developer can query a UDDI registry to obtain the WSDL for the service he/she wishes to utilize. Developers can also design their Web services clients to receive automatic updates about any changes to a service from the UDDI registry.

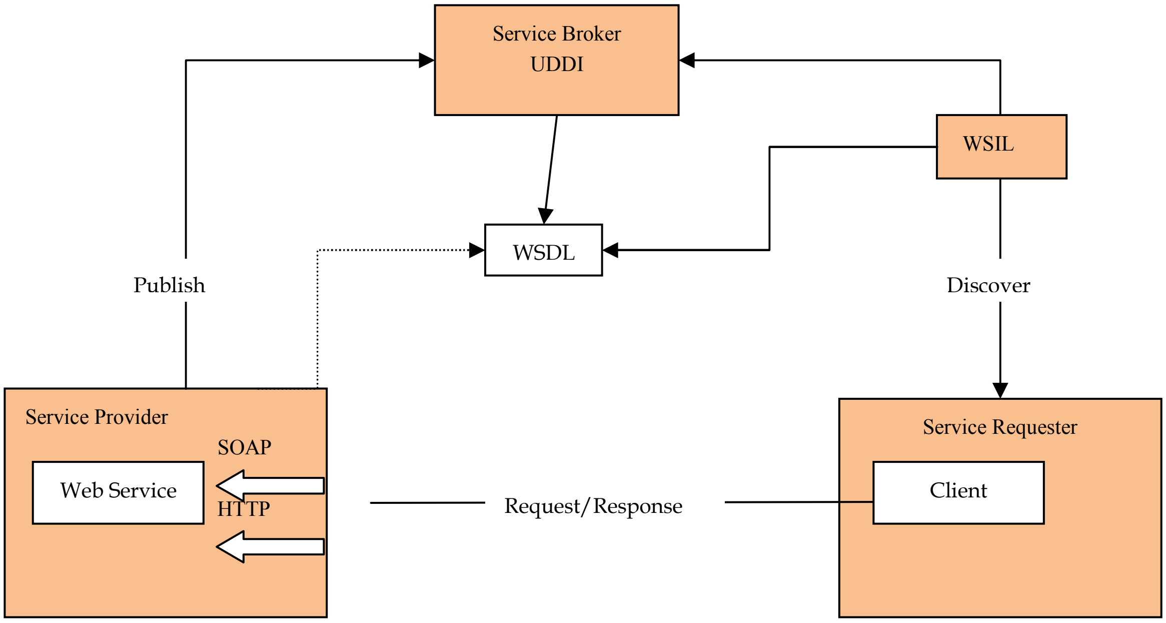

Fig 4: Interaction of HTTP & Services with SOAP, WSDL, UDDI & WSIL

Web Services offer many benefits over other types of distributed computing architetures:

- Interoperability
- Usability
- Reusability
- Deployability

5 How HTTP, Web Servers and Web Services Work Together

The interaction among HTTP, Web Servers and Web Services is simple: HTTP is a simple protocol browsers use to communicate with Web Servers. Web Servers, on the other hand, fulfill users' requests and store the information users provide. Meanwhile, Web Services allow different Web Servers to communicate and interact with one another in order to process the request and/or commands of the user.

A good example of how the interconnectivity among the three technologies works would be a user trying to buy a plane ticket online. The user would access a travel agency's Web page to query for the availability of seats, date and time of the flight and prices of the plane ticket. In this querying process, HTTP acts as the language that users end up using to communicate with the Web server that actually can access the information of flight date, time, seat availability and prices from the airlines database. According to the values users input into the Web page (i.e. GUI) and transmitted to the Web server via HTTP, the Web server performs the command of search by sending out commands of this query to each individual airline's flight schedule databases using an application to application interface, i.e. Web services.

Web services translate whichever markup language the Web server uses into the universally understood XML that gets relayed to the databases of all the airlines. When the XML is received by the airline databases, Web services then translates the XML into whatever programming language that each database is using so that the database would be able to understand the command the Web server sent out. After the query has been completed, the result would be transmitted back to the Web server through Web services again. Then the Web server would relay these search results to the user via HTTP which would present the information to the user through an HTML file that could be interpreted by a browser.

In sum, the simple function of querying for flight schedules and seats requires all three technologies, HTTP, Web server and Web services, to work together. Without any of these technologies, the query would fail or the scope of the search would be drastically limited.

REFERENCES

[1] http://www7b.software.ibm.com/wsdd/library/techarticles/0307_ryman/ryman.html#serv Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] N. Bhatti and R. Friedrich, “Web Server support for tiered services,” IEEE Network, pp. 64-71, Sept/Oct 1999.

[3] Phillip J. Windley, “Enabling Web Services,” 2002-2003

[4] http://msdn.microsoft.com/en-us/library/ms950421.aspx

[5] http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

[6] www.uddi.org/pubs/uddi_v3.htm

[7] http://www.w3.org/TR/soap/

[8] http://www.w3.org/2002/ws/desc/

[9] http://www.uddi.org

[10] http://www.xmlrpc.com/

[11] http://www.webservices.org

[12] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .

[13] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[14] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

[15] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[16] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[17] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[18] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[19] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.