



Working of Web Services using SOA

Mrs. Aarti M. Karande*
V.J.T.I., Matunga,
Mumbai, India
aartimkarande@gmail.com

Prof. Vaibhav N. Chunekar
V.J.T.I., Matunga,
Mumbai, India
chunekar@rediff.com

Dr. B.B. Meshram
V.J.T.I., Matunga,
Mumbai, India
bbmeshram@vjti.ac.in

Abstract: Service Oriented Architecture is used to achieve loose coupling among diverse interacting software applications. SOA is used for reduction in development time and cost. Web services standards used for SOA are distributed software components that provide information to applications rather than to humans, through an application-oriented interface. SOA with web services standards provide greater interoperability. It also provides protection from lock-in to proprietary vendor software. Web services combine the advantages of the component-oriented methods and web techniques. Maintaining Web service quality requires more effort to manage overall Web service framework than each of Web service. Web Services Manager is a security administrator's environment designed to secure access to Web services and monitor activities performed on protected Web services. Web services provide platform independence for the service oriented communication. This way data integration can be done providing the service as a request and service as provider. The purpose of Web service quality management system is to provide the integrated framework, called One-stop service processing the quality information.

Keywords: Service Oriented Architecture, Web service

I. INTRODUCTION

SOA is an architectural style whose goal is to achieve loose coupling among assorted interacting software applications. It also enables organizations to take advantage of existing investments in applications and systems. Using a SOA an organization can be more focused on resources and budget on innovation and on delivering new business services as re. SOA reuses services to automate a business process [1]. Services include Web Service Description Language (WSDL) for service interface definition and XML Schema Documents for message structure definition [3]. Instance of service is Web Services involves operation like Find, Bind, and Execute. Web Service is independent, modulated application which is described, published, located and called through the Internet. The architecture of SOA based on Web Service consists of three roles: service provider, service requester and service register [5].

Data within a SOA generally falls into categories:

- **Service state** - This data pertains to the current state of the business process/service, i.e. where is current process instance at this point in time, which processes are active vs. closed vs. aborted, and so on.
- **Service Result** - This data is delivered by the business process/data service back to the presentation tier. Typically, this data is persistent and stored in backend databases and data warehouses.

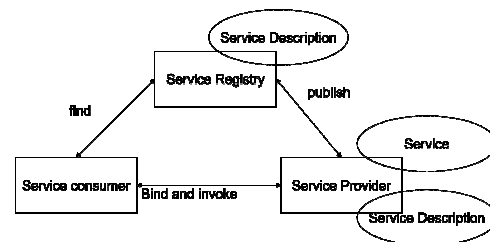


Figure 1. SOA basic architecture specifying component.

A. Component Role in SOA

- **Service Provider:** The service provider creates a web service and publishes its interface and access information to the service registry. Service provider must decide which services to expose, how to make trade-offs between security of various services. With finding easy availability of services, it also decides how to price the services. Provider also check how/whether to exploit services for other value along with category the service should be listed in for a given broker service along with trading partner agreements required to use the service.
- **Service consumer:** The service consumer or web service client locates entries in the broker registry using various find operations and then binds to the service provider in order to invoke one of its web services.
- **Service Registry:** This is the centralize dictionary of the entire service list. It registers what services are available with service provider, and lists all the potential service recipients. Before providing the

service to consumer from provider, registry validates available services in the service registry.

To build and deploy a distributed SOA [9]

- **Service enablement:** Each discrete application needs to be exposed as a service.
- **Service orchestration:** Distributed services need to be configured and orchestrated in a unified and clearly defined distributed process.
- **Deployment:** Emphasis should be shifted from test to the production environment, addressing security, reliability, and scalability concerns.
- **Management:** Services must be audited, maintained and reconfigured. The latter requirements require that corresponding changes in processes must be made without rewriting the services or underlying application.

B. Advantages of using SOA

- **Reduction in development time and cost:** SOA services are build using existing services to form composite applications saving time and cost.
- **Lower maintenance cost:** Reusable services reduce the number and internal complexity of enterprise services.
- **High-quality services:** Increased service reuse creates high-quality services through multiple testing cycles from different service consumers.
- **Lower integration costs:** Standardized services know how to work together, enabling disparate applications to quickly and easily connect.
- **Reduce risk:** Fewer, reusable services provide greater control over corporate and IT governance policies, and reduce the overall compliance risk to an enterprise.

C. Steps to deploy SOA [5]

- **Service enablement:** each discrete application needs to be exposed as a service.
- **Service orchestration:** Distributed services need to be configured and orchestrated in a unified and clearly defined distributed process.
- **Deployment:** Emphasis should be shifted from test to the production environment, addressing security, reliability, and scalability concerns.
- **Management:** Services must be audited, maintained and reconfigured. The latter requirements require that corresponding changes in processes must be made without rewriting the services or underlying application.

II. Web Services

Web services using XML provide information to applications through an application oriented interface, so that it can be parsed and processed easily rather than being formatted for display [3]. It publishes details of their functions and interfaces, keeping their implementation details private; thus a client and a Service interact regardless of the platforms on which they run or the programming languages in which they are written. Thus Web services applicable to a distributed heterogeneous environment. A Web service is a software system designed to support

interoperable machine-to-machine interaction over a network. [2]

1. The key specifications used by Web services

- eXtensible Markup Language - for formatting, exchanging structured data used in web services.
- SOAP (Simple Object Access Protocol)—an XML based protocol specifying envelope information, contents and processing information for a message.
- WSDL (Web Services Description Language)—an XML-based language used to describe the attributes, interfaces and other properties of a Web service. A WSDL document can be read by a potential client to learn about the service.

2. Agents and Services: A Web service must be implemented by a concrete agent. The agent is the concrete piece of software or hardware that performs the operation of sending and receiving messages. Here the service is the resource characterized by the abstract set of functionality provided by web service.

3. Requesters and Providers: A Requester entity's Web service will use a requester agent to exchange messages with the provider entity's provider agent. This is for the exchange of the messages.

4. Service Description: A Web service description (WSD) is a machine processable specification of the Web service's interface, written in WSDL. It defines the message formats, data types, transport protocols, and transport serialization formats that should be used between the requester agent and the provider agent with the specification of one or more network locations at which a provider agent can be invoked, with information about the message exchange pattern that is expected.

5. Semantics: The semantics of a Web service is the shared expected result of the service, particularly in response to messages that are sent to it. This is the "contract" between the requester entity and the provider entity regarding the purpose and consequences of the interaction for communication or message exchange. The semantics represents a contract governing the meaning and purpose of interaction.

A. Overview of Engaging a Web Service

- 1)The requester and provider entities become known to each other with their structure and working style.
- 2)The requester and provider entities agree upon standard protocol for communication, with the required service description and semantics.
- 3)The service description and semantics are realized by the requester and provider to their respective agents.
- 4)The requester and provider agents exchange messages as per the service requirement, performing some task on behalf of the requester and provider entities.

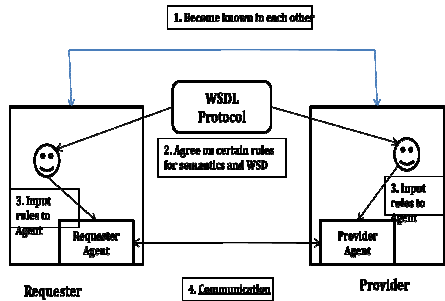


Figure II. working of Web Services

B. For developing software system based on SOA

- Find the dynamic relation between function requirements and acquired services and relation between services and bottom technology of the used services [6].
- The basic services must meet the requirement of definition definitude and function singularity in SOA. Each single deployment services in SOA is independent of else system.
- The whole system function is implemented by the services process organization.

III SOA and Web Service

SOA is a distributed systems architecture characterized by web service properties as [2] :

- 1) **Logical view:** The service is an abstracted, logical view of actual programs databases, business processes, etc. It is defined in terms of what it *does*.
- 2) **Message orientation:** The Web Service defined as the messages exchanged between provider agents and requester agents, but not defines as the properties of the agents themselves. By avoiding any knowledge of the internal structure of an agent, one can incorporate any software component or application that can be "wrapped" in message handling code that allows it to adhere to the formal service definition.
- 3) **Description orientation:** A service is described by machine-processable metadata which supports the public nature of the SOA. Details that are exposed to the public and important for the use of the service should be included in the description. The semantics of a service should be documented by its description.
- 4) **Granularity:** Services tend to use a small number of operations with relatively large and complex messages.
- 5) **Network orientation:** Services tend to be oriented toward use over a network, even though this is not an absolute requirement.
- 6) **Platform neutral:** Messages are sent in a platform neutral, standardized format delivered through the interfaces. XML is the most obvious format that meets this constraint.

Implementers commonly build SOA using web services standards that have gained broad industry acceptance. These standards provide greater interoperability and some protection from lock-in to proprietary vendor software. SOA can be implemented using any service-based technology.

A. Web Services and SOA Infrastructures

The purpose of a SOA infrastructure is to allow consumers to invoke services exposed by providers. Web services rely on XML-based industry standards for [8]:

- A data format that allows uniform communication between web services consumers and web services providers (XML specification).
- A framework that describes XML vocabularies used in business transactions (XML Schema).
- An envelope used to send structured requests to, and receive structured responses from the web service provider (SOAP).
- WSDL defines what a web service does.
- UDDI is a framework to publish and look up web services on the Internet (Universal Description, Discovery, and integration)

C. Web Services and Hierarchy Framework

Web services combine the advantages of the component-oriented methods and web techniques, and can describe their own services. It can also publish, locate and transfer modularized application on web[6]. Web Services provided function may be simple, with complicated business logic.

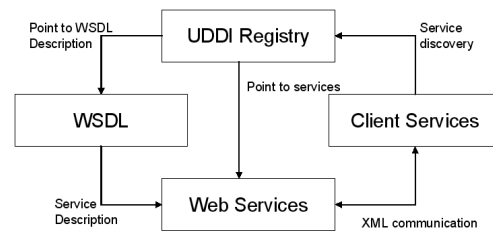


Figure 3. key techniques for web service

D. Operation mode of web services

- Publish the service: Services provider publishes services WSDL description information in the UDDI register centre.
- Compose, or orchestrate, the services into business flows: Then through querying UDDI registering centre, services requester gains the services WSDL documents to provide the web services interoperability information.
- service requester sends SOAP request to the services provider, and then services provider returns SOAP response messages to services requester.

D. Web services properties

- Discoverable: using the concept of serving to other users so new services has to be discovered and accessed by consumers (human users or other Web services)[11]
- Communicable: This is often asynchronous messaging as opposed to synchronous messaging.
- Conversational: A conversation involves sending and receiving documents in a context. This involves complex interactions between Web services and entails multiple steps of communication that are related to each other.
- Secure and Manageable: Security, manageability, availability, and fault tolerance are critical for a commercial web-service.

E. Why Web Service?

- Exposing the function on to network: service can be remotely invoked using HTTP that is, it can be activated using HTTP requests. So, Web Services allows exposing the functionality of existing code over the network.
- Connecting Different Applications
- Standardized Protocol: All the four layers (Service Transport, XML Messaging, Service Description and Service Discovery layers) use the well defined protocol in the Web Services protocol stack.
- Low Cost of communication
- Loosely Coupled Applications: Web Services are self-describing software modules which encapsulates discrete functionality.
- Web Services Sharing
- Web Services are Self Describing: reduces the software development time.
- Automatic Discovery: Web Services automatic discovery mechanism helps the business to easy find the Service Providers. This also helps customer to find their services easily increasing revenue by exposing their own Web Services available to others.
- Business Opportunity: Web Services has opened the door to new business opportunities by making it easy to connect with partners.

F. Uses of Web Services

Web services are a set of tools can be used in a number of ways most commonly like:

- **Remote Procedure Calls:** RPC Web services present a distributed function call interface, which is familiar to many developers. Typically, the basic unit is the WSDL operation.
- **Service-oriented architecture:** Under SOA Web services are used to implement an architecture in which the basic unit of communication is a message, rather than an operation. This is often referred to as "message-oriented" services.
- **Representational state transfer:** REST Web Services uses WSDL to describe SOAP messaging over HTTP, which defines the operations. REST describes operations, can be implemented as an abstraction purely on top of SOAP or can be created without using SOAP at all.
- In Reusable application-components uses Web Services offers most frequently used services like currency conversion, weather reports, language translation etc.

IV. WEB SERVICE QUALITY

Quality could be classified as interoperability, manageability, security, business processing capability, business value, measurable quality, which have to be considered as applying Web service into real world [8]. Web service quality model technical committee [9] established the Web service quality model and the XML schema for representing the specification. Maintaining Web service quality requires more effort to manage overall Web service framework than each of Web service. The Web service registry has a core position in Web service framework,

implemented according to UDDI specification [10]. The registry takes charge of mediating, registering, and searching Web service information using WSDL.

A. Web Services Quality Management System

The purpose of Web service quality management system (WSQMS) is to provide the integrated framework, called One-stop service processing the quality information. The Web service quality information life cycle is of phase: test, evaluation, registration, search, and abolition. For providing quality data it is necessary to integrate Web service registry and WSQMS, enabling Web service users to easily get the Web service quality information gathered by the WSQMS. When a Web service is registered and a registry notifies it to WSQMS, the WSQMS automatically tested the Web service and stored quality data at its local storage. The WSQMS, after then, continue to collect the quality data of the Web service by periodic monitoring. Web service quality factors which have to be considered as applying Web service into real world. [12]

- interoperability,
- manageability,
- Security
- business processing capability
- business value
- measurable quality

B. Working of Web Services Quality Management

For providing quality data during all the lifecycle of Web service, it is necessary integration of a Web service registry and WSQMS, enabling Web service users to easily get the Web service quality information gathered by the WSQMS.

- When a Web service is registered and a registry notifies it to WSQMS, the WSQMS automatically tested the Web service and stored quality data at its local storage.
- The WSQMS, after then, continue to collect the quality data of the Web service by periodic monitoring.
- After installing an agent, WSQMS requests the operation information (OS (Operating System), platform, etc) of target Web service system to the agent.

V. DATA QUALITY IN SOA

- **Uniqueness:** avoid redundancy for improve data quality in data integration.
- **Completeness:** when the information that is missing is critical to specific business services and processes, completeness becomes an issue.
- **Accuracy:** Incorrect spellings of per-son names and addresses can significantly impact operational and analytical applications.
- **Consistency:** Data inconsistency can become a serious problem when trying to integrate applications or implementing business processes that use different data sources.
- **Availability:** Data availability depends on reliability of data sources.

- **Conformity:** Maintaining conformance to specific formats is important in data representation, reporting and searching.
- **Integrity:** The inability to link related records together may introduce redundancy across systems.

VI. CONCLUSION

- Service-Oriented Architecture is an architectural style whose goal is to achieve loose coupling among diverse interacting software applications, enabling reusing services to automate a business process. A standard interface and message structure define services. These standards include the Web Service Description Language file for service interface definition and XML Schema Documents for message structure definition.
- System constructed on SOA using web services standards provide greater interoperability and some protection from lock-in to proprietary vendor software. The UDDI registry takes charge of mediating, registering, and searching Web service information, *WSDL* form in most cases.
- Web services combine the advantages of the component-oriented methods and web techniques, and can describe their own services.
- Web Services automatic discovery mechanism helps the business to easy find the Service Providers. This also helps customer to find their services easily increasing revenue by exposing their own Web Services available to others.
- SOA with Web Service based standards provide greater interoperability and some protection from lock-in to proprietary vendor software.
- The purpose of Web service quality management system (WSQMS) is to provide the integrated framework, called One-stop service processing the quality information.
- Web services provides platform independence for the service oriented communication

VII. REFERENCES

- [1] R.Radhakrishnan, B.Sriraman, Aligning Architectural Approaches towards an SOA-Based Enterprise Architecture, in Proc. Working IEEE/IFIP Conference on Software Architecture 2007 (WICSA), Mumbai, India, Jan. 2007.
- [2] Web Services Architecture W3C Working Group Note 11 February 2004.
- [3] An Overview of SOA computing by latha srinivasan and jem Treadwell.
- [4] Study on EAI Based on Web Services and SOA, Wu Deng, Xinhua Yang, Huimin Zhao, Dan Lei, Hua International Symposium on Electronic Commerce and Security
- [5] Service-Oriented Architecture Concepts, Technology, and Design,Thomas Erl
- [6] Study on EAI Based on Web Services and SOA Wu Deng, Xinhua Yang, Huimin Zhao, Dan Lei, Hua Li, International Symposium on Electronic Commerce and Security 2008
- [7] Securing web services & service oriented architectures with oracle web service manager11g.
- [8] Design of SOA based web service systems using QFD for Satisfaction of quality of service requirements Xiaoqing liu ,Lianzhang zhu, 2009 ieee international conference on web services.
- [9] Min and Eunju Kim, A Study for Web service quality Model and Test guideline, NCA IVRER 04052, 2004.12
- [10] UDDI TC, Universal Description, Discovery and Integrationv 3.0.2 (UDDI), <http://www.oasisopen.org/committees/uddispec/doc/spec/v3/uddiv3.0.220041019.htm>
- [11] Enterprise Architecture and Web Services Dinarle Ortega, Elluz Uzcátegui, María M. Guevara 2009 Fourth International Conference on Internet and Web Applications and Services
- [12] QoS Management for SOA by Synchronizing Quality Context in UDDI Youngkon Lee International Journal of Grid and Distributed
- [13] 13.Service-Oriented Architecture Concepts, Technology, and DesignThomas Erl