



## SERVICE ORIENTED ARCHITECTURE – A NEW PARADIGM

<sup>1</sup>Asha H.V., <sup>2</sup>Dr Shantharam Nayak

Dept of CS&E, Nitte Meenakshi Institute of Technology, Bangalore,

Dept of IS&E, R V College of Engineering, Bangalore

Email: ashahbandi@gmail.com, shantaram\_nayak@yahoo.com

**Abstract— One of the most important and critical task in the development of any software system is its architecture. As an important column of software development process, software architecture has various roadmaps and methods. To control the complexity of construction of system development and its evolution, architecture based approaches have been promoted. The concepts of Service Oriented Architecture (SOA) have exhibited themselves as one of the evolutionary step for software development. This paper outlines a brief review on the concepts, characteristics and the structure of SOA. First, architecture definitions are presented. Further, SOA's conceptual view is discussed with a brief introduction to Web Services. Finally it is concluded by giving an overview on the characteristics of SOA making it as one of the useful choices for software development**

**Index Terms— Software Architecture, Service-Oriented Architecture, Web Services.**

### 1. Introduction

During the history of evolution of computing, Software has become increasingly complex. [1] has proposed "structured programming" and idea of "conceptual integrity" by Fred Brooks [2] to address the complexity. The design phase of any software development life cycle has often

been divided into high-level design and detailed design. Most concepts in the process of building, noted that the architecture will be useful to describe the software that gave birth to the term "software architecture". From then, it has emerged as a design solution for complex problems where it plays a key role in bridging the gap between requirements and implementation. According to [3] software architecture is defined as: 'the structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time'. With a rapid progress in software development, SOA has evolved from Component-Based Architecture, Interface Based Design and Distributed Systems. SOA, a new paradigm aims at building loosely-coupled systems that are extendible, flexible and fit well with the existing legacy systems. SOA offers a number of promises like re-use, cost-efficiency, agility, adaptability and flexibility by making use of distributed capabilities to bring a technical solution to an enterprise business problem.

### 2. Service Oriented Architecture

Service Oriented Architecture (SOA) architectural paradigm enables building of infrastructures for diverse application integration and interaction across different locations, domains of technology and platforms. Unlike

traditional architectures, SOA focuses on integrating capabilities that are distributed heterogeneously and implemented using a mixture of technologies by providing a set of methodologies and strategies among different technology stacks.

SOA is not a framework or product that can be bought and used. Rather it is thought of as a paradigm or an architectural style. It is entirely a thinking approach that helps in designing interoperable software architectures according to enterprise business needs. SOA aims at combining architectural style and methodologies to achieve interoperability among heterogeneous and homogenous systems by creating reusable services. It concentrates on the problem domain more rather than the underlying technology used to implement it. The conceptual model of SOA can be given as in the figure 1. [4].

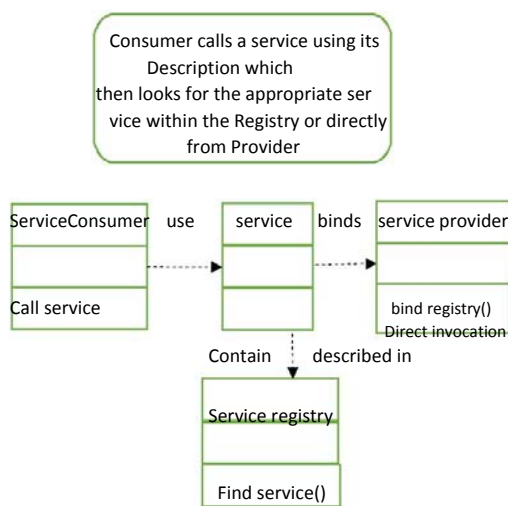


FIG-1 SOA Conceptual Model

There are different implementation models and techniques for SOA but the most common is through standardized Web Services which make interoperability easy to implement among heterogeneous systems.

## 2.1 Working of Services in SOA

The core concept of SOA is based on Services. Services can be considered as the business tasks performed by an external component in accordance with pre-defined contract or specification. On requirement of any specific functionality service, elements are identified and that particular service is executed. This is attained with the help of three entities: requester, provider, and broker.

**Service Requester:** Service requester requires some operational work provided by some other service. It will search through the list of service descriptions provided by the service broker.

**Service Provider:** Service provider has access to other services. It accepts and executes requests from the consumer. It contains all the required details about service description and its implementation. It can be a mainframe system, a component or any other type of system which executes the consumer's requests. It creates services and publish them to the service broker.

**Service Broker:** Service broker comprises all the statistics about services which are registered with it. It will convey requests to the corresponding service requester and service provider.

**Service Consumer:** Service consumer munches the functionality delivered by a service. It requests different functions within a service for different functionality. The consumer can be an application, another service or a software module. The consumer can either directly call the service if its site is known or can look-up the service location in the registry.

**Service registry:** Service registry is a directory consisting addresses of all the available services. Its serves the purpose of storing and publishing the service contracts of different service providers. On request the same will be provided to the interested service consumers.

**Service Contract:** A service contract is the portrayal of the way a service consumer would interact with a service provider. It holds information about the exchange message formats, quality of service and, the pre and post conditions which need to be fulfilled before the service can be executed.

In SOA model, system development is centered on loosely-integrated suite of services that will be used within multiple business domains [5]. Commonly, SOA makes use of web service software components, designed to support interoperable machine-to-machine interaction over a network. Primarily, web services use SOAP (Simple Object Access Protocol), an XML-based protocol, to communicate over the HTTP protocol. Besides, they use WSDL (Web Service Description Language) to describe their internal functionalities and UDDI (Universal Description, Discovery, and Integration), a global registry and repository, to register and store their WSDLs [6]. Web services are governed by provider-requester model where the provider owns the necessary equipment to host web services, and the requester connects to these web services and starts calling visible functions through method invocation method. Several styles and types of web services exist, which includes SOAP, REST, .NET Remoting, RMI, RPC, and others. Figure 2 illustrates the provider-requester model of a SOAP-based web service [7].

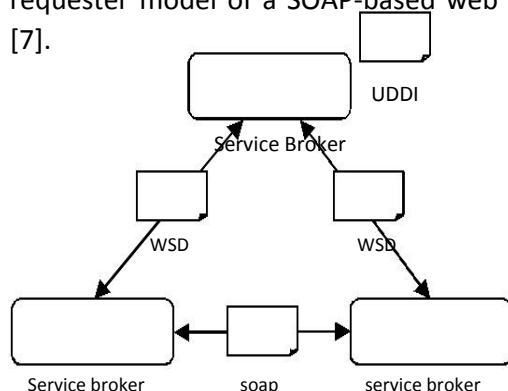


Figure 2: Provider-Requester Model

## 2.2 Web Services

Web Service defined according to [8] is given as:

“Web services are typically application programming interfaces (API) or Web API's that are accessible through HTTP and executed in a remote system hosting the requested services”

Another definition proposed by [9] is:

“A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine process-able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.”

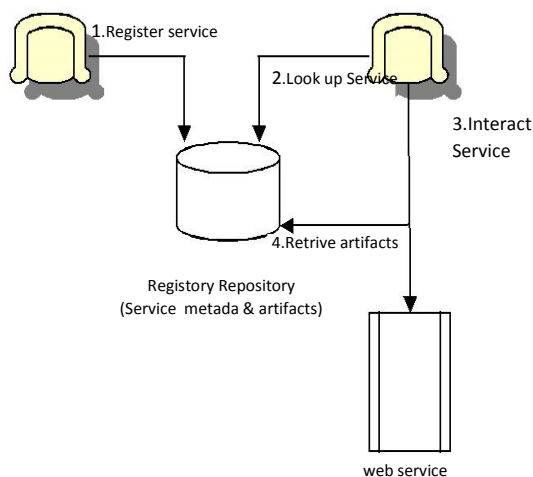
Web services is also viewed as the collection of standards, of which the most important ones are SOAP, XML, WSDL, and UDDI. Different platforms and programming languages can exchange messages (like function calls and replies) through a language called XML. The prescribed format of these messages is called the “Simple Object Access Protocol” (SOAP). A web service has a description of its interface, so that clients can see how to consume a web service. This interface is written in an XML-based language called “Web Service Description Language” (WSDL). There is a way for centrally registering the web service enabling others to find it, through “Universal Description, Discovery and Integration” (UDDI).

## 2.3 Working of Registry

The implementation artifacts of SOA is registered within a repository to maximize reuse and provide management of enterprise assets [10]. Registries main purpose is to register services and equip them with the proficiency of being dynamically discovered. The service gets registered within the registry by providing its address and Meta information. A registry allows diverse applications to communicate reliably, without any interoperability problem. It can be considered as a catalog that have all the

information about the available services. This catalog is regularly updated as the information about services gets modified. A SOA registry allows business services to get dynamically discovered to be used by the consumer.

Different approaches can be used to implement SOA registry. The most common approach is the UDDI specification, an XML based registry developed for the purpose of making systems interoperable for e-commerce and interaction among web services. Also, UDDI facilitates other enterprise applications to discover and use up-to-date and appropriate information in a more competent method [10, 11].



### 3. Characteristics of SOA

Every architecture of a system have different principles and set of tradeoffs used by the designers. SOA too has these trail characteristics [12]:

i) Discoverable and Dynamically Bound: A service consumer discovers what service to use based on the description at runtime. The service consumer asks the registry for a service that matches the given criteria.

ii) Self-Contained and Modular: A service supports a set of interfaces and these relate to each other in the context of a module. The principles of modularity should be adhered in

designing the services to aggregate services into an application with a few well-known dependencies.

iii) Interoperability: Systems use different platforms and languages to communicate with each other and each service is invoked with help of a connector type. An interoperable connector consists of a protocol and data format that each of the potential clients of the service understands. Interoperability is achieved by supporting the protocol and data formats of the service's current and potential clients.

iv) Loose Coupling: Coupling refers to the number of dependencies between modules. There are two types of coupling: loose coupling and tight coupling. Loosely coupled modules have a few well known dependencies. Tightly coupled modules have many unknown dependencies. Every software architecture endeavors to achieve loose coupling between modules. SOA promotes loose coupling between service consumers and service providers.

v) Location Transparency: Consumers of a service do not know a service's location until they locate it in the registry. The lookup and dynamic binding to a service at runtime allows the service implementation to move from location to location without the client's knowledge. The ability to transfer service improves the availability and performance of a service. By employing a load balancer that forwards requests to multiple service instances without the service client's knowledge, we can achieve greater availability and performance.

vi) Composability: A service's composability is connected to its modular structure which enables services to be assembled into applications. Use of preexisting tested services enhances system's quality and improves its return on investment. A service may be composed in three ways: application composition, service federations, and service orchestration.

-An application: is typically an assembly of services, components, and application logic that binds these functions together for a specific purpose.

- Service federations: Collections of services managed together in a larger service domain.

- Service orchestration: execution of a single transaction that impacts one or more services in an organization. It is sometimes called a business process. It consists of multiple steps, each of which is a service invocation. If any of the service invocations fails, the entire transaction should be rolled back to the state that existed before execution of the transaction.

vii) Self-Healing: A self-healing system is one that has the ability to recover from errors without human intervention during execution.

Conclusion:

Information Technology (IT) has revolutionized the way we work, live or even think today. The objective of this paper is to give a brief overview on Service Oriented Architecture and its working principle. SOA has become a well-established concept in the field of Information Technology. At the core of SOA is the concept of services. SOA if applied correctly benefits enterprises immensely, in both IT and business integration sector. Lowered costs for integration and easy adaptability to change are the biggest benefits of SOA. Therefore, it is a concept of future and is being used for designing and architecting more and more IT solutions. But implementing SOA have the challenges of its own. Among which, testing is one of the critical and important thing. In the future work a survey on the existing testing methodology of SOA will be carried out highlighting the pros and cons and the way to overcome them.

#### REFERENCES

[1]. O. J. Dahl, E. W. Dijkstra, and C. A. Hoare, Structured Programming, Academic Press, 1972.

[2] F. P. Brooks, The Mythical Man-Month: Essays On Software Engineering, Anniversary Edition. Addison-Wesley, 1995.

[3]. David Garlan and Dewayne Perry, "Introduction to the Special Issue on Software Architecture

[4]. Bieber, Carpenter, Stevens, (2001), Jini Technology Architectural Overview, Sun Microsystems

[5]. Nicolai M. Josuttis, SOA in Practice, O'Reilly, 2007.

[6] Fielding, Roy T., Taylor, Richard N, Principled Design of the Modern Web Architecture, ACM Transactions on Internet Technology (TOIT), (2002), 2(2): 115–150.

[7]. Distributed, Cross-Platform, and Regression Testing Architecture for Service Oriented Architecture", Youssef BassilAdvances in Computer Science and its Applications (ACSA), March 2012

[8]. "WebService", [http://en.wikipedia.org/wiki/Web\\_service](http://en.wikipedia.org/wiki/Web_service).

[9]. D. Booth et al, "Web Service Architecture", <http://www.w3.org/TR/ws-arch/2004>

[10]. Jeff Davis, (2009), Open Source SOA, Manning Publications, ISBN 978-1-933988-54-2

[11]."SOA registry", Accessed: 30 June 2010 Retrieved from: [http://searchsoa.techtarget.com/sDefinition/0,,sid26\\_gci12\\_70465,00.html](http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci12_70465,00.html)

[12]. G. Bieber and J. Carpenter, "Introduction to service-oriented programming", Online whitepaper: [www.openwings.org/download/specs](http://www.openwings.org/download/specs), 2001.