WEB SERVICE UTILIZATION IN SERVICE ORIENTED ARCHITECTURE

[1]Asha H.V., [2]Manohar R., [3]Dr Shantharam Nayak
[1]Dept of CS&E, NMIT, B'lore, [2]Dept of IS&E, Sir MVIT, B'lore, [3]Dept of IS&E, RVCE, B'lore
Email: [1]ashahebbandi@gmail.com, [2]mailmanu.r@gmail.com,[3]shantaram_nayak@yahoo.com

Abstract

In the recent years, because of the technology trend in Information Technology (IT), Service Oriented Architecture (SOA) has emerged as a new software engineering principle which provides a reference model for distributed dynamic systems to cope with the complexity of software development. IT industry has progressively adopted service oriented technology because of its foreseen benefits like loose coupling and dynamic interoperability. Core concept of SOA is services. Services are business tasks performed by an external component according to a predefined specification.

Lots of services are available over the internet. But, it may not be possible for individual standalone service to meet all the service requirements of many users. With the rapid increase of number of services there is a need for methodologies to locate desired services that provide access and data mining capabilities. With this requirement, SOA tries to construct a distributed, dynamic, flexible, and re-configurable service system over Internet that can meet information and service requirements of many different users. SOA can be implemented using different types of technologies. But, Web Services are the most common method for implementing SOA as they provide well defined standards that make ideal for interoperability.

1. Introduction

IT industry has revolutionized to a great extend with the development of new technologies. They are moving from tightly coupled architecture to loosely coupled architecture which provides competence in terms of development cost, time and efficiency when compared to the other traditional architectural types. Industry trends and needs are converging to drive elemental IT changes around the concepts and implementation of service orientation [4]. The key technologies are: Extensible Markup Language (XML), a common independent data format across enterprise, Web Services, an XML-based technology for messaging, service description, discovery and extended features. SOA is built upon a tradition of technology and business needs. It focuses on reusable code and modular design, objects, components, and enterprise application integration [14].

2. Service Oriented Architecture

SOA is the result of the natural evolution from Object-oriented models to Component based models and to Service Orientation. Service orientation retains the benefits of object oriented systems and component-based development (self- description, encapsulation, dynamic discovery and loading). The shift is in the paradigm which has stirred from remotely invoking methods on objects, to passing messages between services. SOA is a way of designing, developing and managing systems [12]. It is a system architecture where a collection of loosely coupled components communicate with each other using standard interfaces and message-exchanging protocols [5]. SOA approach to software resources are

unlike when compared with traditional architecture. It represents a model in which functionality is decomposed into distinct independent units called services which can be distributed over a network and can be combined together and reused to create business applications with well-defined invokable interfaces. In traditional systems, functionality is built into the systems so that, when a particular functionality is used in a different application, the code is often rewritten. As a result, there might exist a slightly different version of the same piece of code. When there is a business change, all of this code needs to be updated. But in SOA, each functionality is put in a component called service and that service will be made available across a number of applications. The dictionary definition of a service is given as "a unit of work performed by one for another" [15]. Primarily, a service can be a simple and independent business function or a business transaction which may be implemented as composite function. Further, they are software modules that are accessed by name via an interface, typically in a request-reply mode. SOA can be implemented using different technologies (Figure 1). But, the most preferred is Web Services.

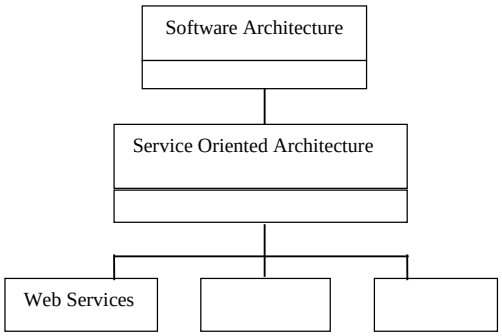


Figure1: Web services are one set of technologies for implementing SOA

The services are autonomous and platform independent with the ability to interact with one another regardless of being developed in inconsistent platforms and programming languages [4]. They can reside on different computers and use each other's services to achieve the desired goals and end results. A new service can be composed at runtime based on locally or remotely available services. Remote services can be searched and discovered through service brokers that publish services for public accesses. Figure 2 depicts the basic structure of SOA.

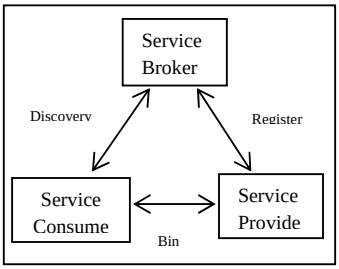


Figure 2: SOA Basic Structure

Each time a specific functionality is required, service elements are identified, terms and conditions are negotiated, and the service is executed to provide a solution. Thus, SOA is an architectural style for building software applications that use services available in a network such as the Web. SOA is a principle concept underneath Web Service implementation. It promotes loose coupling between software components so that they can be reused.

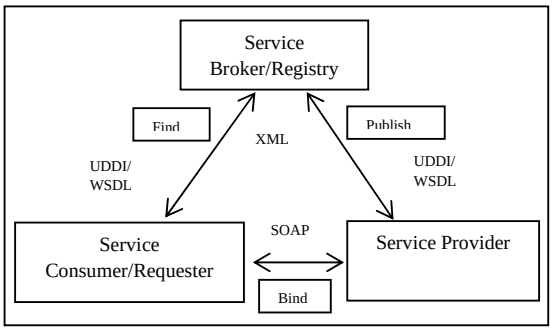
SOA has three basic components: service provider, service registry, service requester which performs three basic operations: publish, find, and bind. A SOA relates the roles of the three components with the three operations to maintain automated discovery and the use of services.

- **Service Provider:** executes business logic**;** publishes services to a registry and makes it available on the Internet for the requests of the consumers.
- **Service Requester/Consumer:** use one or more services provided by service provider; performs service discovery operations on the service registry in order to find the needed service.
- **Service Registry/Repositories:** services are registered**;** helps service providers and service requesters to find each other by acting as a registry of the services [11].

The providers register services in the registry and the service consumer's access services from there.

SOA relies on message based communication which involves SOAP messages containing XML document payloads which are structured to be self-sufficient with a wide range of meta-information, processing instructions, and policy rules. XML and Web Services introduce sophisticated data representation and offer a solution to the proprietary platform boundary

problem. It can be depicted as shown in the figure 3.



Quite a lot of benefits can be listed for SOA. As it is been reflected in the literature, it is mainly based on reusability and standardized components by providing users with just the functionality they need.

>> One of the prominent benefits of service orientation is loose coupling [3]. The principle of loose coupling is that of using a resource only through its published service and not by directly addressing the implementation behind it. By this, changes to the implementation by the service provider will not affect the service consumer. By maintaining a consistent interface, the service consumer could choose an alternative instance of the same service type without modifying their requesting application, apart from the address of the new instance. The service consumer and provider do not have to have the same technologies for the implementation, interface, or integration when Web Services are used [9].

>> Another major intention of the design of SOA is, data used for a given processing activity are not stored locally. Rather, they are decentralized close to the source of production which helps in avoiding inconsistency in local copies and repositories of data. Hence, the quality of the output is possibly increased in cases where data from various different sources are used. Furthermore, redundancy in the algorithms used for specific processing tasks are also avoided.

The other benefits of SOA can be listed as below:

i) SOA can be developed from existing system. The services can be created using existing technologies with component-based approach. Thus, it can generate new routes with flexibility. [13].

ii) SOA is embedded in object-orientation. However, it adds a layer of abstraction. Basically, service-orientation is not a departure from object-orientation, but rather an evolution [1].

iii) SOA is not just architecture of services as seen from a technology perspective. Instead it can be considered as policy, a practice, and a framework by which it is ensured that right services are provided and consumed [2].

iv) The development of Web Services using service oriented technology will provide heterogeneous network addressable service component for location transparency [2]. SOA offers new services to customers without worrying about the underlying IT infrastructure.

v) SOA provides cost effectiveness by integrating historically separate systems with reduction in cycle times and costs.

vi) SOA aims at reducing risk by improving visibility of business operation.

4. Web Services

SOA is implemented at different network environments. Web Service concept is based on SOA paradigm using which a complete application can be constructed from various services which provide different functionalities.

Web Services are self-contained, self-describing, new breed of Web applications that can be published, located, and dynamically invoked across the Web. It provides access to a set of operations/functions which can be accessed through one or more standardized interfaces using standard web technology that adheres to web service standards [8]. The main aspect of the Web Services is that they allow program-to-program communications. With the help of several Web Service specifications a complete cycle of describing, publishing, and finding services can be made possible [13].

Web Service architecture is the logical evolution of object-oriented analysis and design that reflects a new service-oriented architectural approach, based on the concept of building applications by discovering and orchestrating network-available services. It is also the logical evolution of the architecture, design, implementation, and deployment of e-business solutions. Fundamental to Web Service is the conception that, everything is a

service, publishing an API for use by other services on the network and encapsulating implementation details [8].

In the literature properties of Web Services is given as: First, Web Services are for application-to- application communication [16]. Second, Web Services are accessed over Internet. And finally, Web Services are XML based and not for proprietary solutions [10].

4.1 The Standards for Web Services

Web services use XML (eXtensible Markup Language) software systems designed to support interoperable machine- to-machine interaction over a network. This interoperability is gained through a set of XML based open standards, such as WSDL, SOAP, and UDDI that provides a common approach for defining, locating, publishing, and using web services for sophisticated communications between various nodes in a network. The Web service protocol stack (Figure 4) is a collection of these protocols that are used to make Web services interact with each other.

Figure 4: Web Service Stack

 # Transport Protocol: It is responsible for transporting messages between network applications. HTTP (Hyper Text Transfer Protocol) is the low-level protocol used by the Internet for the transport layer.

 # Messaging Protocol: It is responsible for encoding messages in a common XML format so that they can be understood at either end of a network connection. Simple Object Access Protocol (SOAP) is the specific format for exchanging Web Services data over HTTP.

 # Description Protocol: It **is** used for describing the public interface to a specific web service. Web Service Definition Language (WSDL) is used to describe what type of message a Web Service accepts and generates.

 # Discovery Protocol: It centralizes services into a common registry. Universal Description, Discovery, and Integration (UDDI) specification can be used by the service providers to advertise the existence of their services and by requesters to search and discover already registered services [13], [7].

Service interface hides the implementation logic from the users, which allows the service to be used on different platforms than on which it was implemented. These properties allow Web Service based frameworks to be loosely coupled and component oriented. Because of the standard interfaces and messaging protocols the Web Services can easily be assembled to solve more complex problems.

The main advantage of Web Services lies in using any service remotely without the user's actual knowledge and intervention and by multiple users at the same time, eliminating the need for constant updates to locally installed software. In addition, it minimizes the network traffic, since data do not need to be transferred to the client in every step of the operation [6].

4.2 Benefits of Web Services

Benefits provided by Web services in implementing SOA are:

a) Web Services provide great level of interconnection for different clients for different services by using different devices, which is the most important aspect of SOA. They use distribution and integration of application logic by providing remote and independent services to any application, which is the most significant requirement of SOA [2].

b) They provide loosely coupled architecture with homogeneous infrastructure by providing protocols and messages, which is the base for SOA.

c) They enable programmable web which over comes the restriction on interactive web.

Conclusion:

SOA is one of the most popular architectural paradigms of today. It is an architecture that provides seamless Enterprise Information Integration between loosely coupled distributed applications or services over the network. It is shown that SOA provides integrated, efficient, reusable and platform independent environment for current business requirement. The Web Service does not equal to SOA but it fulfills the requirements of SOA by providing flexible, interconnected and efficient services on the web architecture using various technologies. Web

Service provides the interoperability and loose coupling, which are the major requirements of current needs.

References:

[1]. D.Sprott and L.Wilkes," Understanding Service Oriented Architecture", Architecture Journal", CBDI Forum, 2004, pp. [3- 4]

[2]. C. Irek: IBM Global "Services, Realizing a Service Oriented Architecture with .NET", 2005

[3]. S. Güner, "Architectural Approaches, Concepts and Methodologies of Service Oriented Architecture", M.S. thesis, Technical University Hamburg Harburg, Germany, 2005.

[4]. Newcomer.E, Lomow G., "Understanding SOA with Web Services", Addision Wesley, (2005).

[5]. W.T Tsai, Q. Huang, B. Xiao, and Y. Chen, "Verification Framework for Dynamic Collaborative Services in Service Oriented Architecture", Department of Computer Science and Engineering, Arizona State University, Tempe, AZ.

[6]. Kotzinos, D. and Chrysoulakis, N., "Design of GIS Web Services for Environmental Monitoring: Using Satellite Imaging to Calculate Vegetation Indices", *Geographical Information Systems and Remote Sensing: Environmental Applications*, Volos, Greece, 2003.

[7]. Josuttis, N.M., *"SOA in Practice"*, O'Reilly, CA, USA, pp. 211-219, 2007.

[8]. Doyle, A. and Reed, C., "Introduction to OGC Web Services", An OGC® White Paper, Wayland, MA, USA, 2008

[9]. Linthicum, D., "SOA in the Real World", MSDN Technical Article, USA, 2008.

[10]. Akinci, H., "Geospatial Web Services for e-Municipality", XX ISPRS Congress, Istanbul, TURKEY, 2004.

[11]. Sayar, A., "GIS Service Oriented Architecture", Community Grids Laboratory, USA, 2008.
http://complexity.ucs.indiana.edu/~asayar/gisgrids/docs/gis- soa.pdf , 2008.

12]. Lewis, G.,Smith, D., and Tilley. S.," Migration of Legacy Components to SOA", Environments, Carnegie Mellon University / Software Engineering Institute, 2008.

[13]. Aydin, G., "Service Oriented Architecture for Geographic Information Systems", Supporting Real Time Data Grids, Ph.D. Thesis, Indiana University, IN, USA, pp.12-47, 2007.

[14]. Toby Sucharov, "Mainframe Make-overs", Information Professional,
Institute Engineering and Technology,
Volume 4, Number 6 Dec/Jan 2007/08.

[15] Merriam-Webster Dictionary. Online at http://www.merriamthe testing of SOA-based systems requires special skills such webster.com/dictionary/.

[16]. K. Sahina, M. U. Gumusay, "Service Oriented Architecture (SOA) Based Web Services for Geographic Information Systems", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008