

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Intoduction to SOA

Ojasvi Sharma

School of Computer Science and Engineering
Bahra University
Himachal Pradesh, India
ojasvi.sharma93@gmail.com

Abstract - Achieving flexibility, balance, adaptability, and coordination under one umbrella in business and easing the integration of systems are main goals for software development. Business agility is the ability to change the business process quickly in response to the change in the business environment, such as adding a new service to the organization portfolio [1]. One way to achieve this in a fast, yet cost effective way is to adopt the Service Oriented Architecture (SOA) approach. SOA is an architecture framework that views the system as set of services. The service is a software component that implements a reasonable amount of work. Each service is loosely coupled to increase its scalability and reusability. So, whenever a business need emerges and requires adding new functionality to the system, the development team in the organization can develop a service that fulfill that need, and rapidly integrates it in the current system without the need to change the overall architecture of the system.

Keywords—Service Oriented Architecture; SOA; Services; Orchestration; Choreography.

1. INTRODUCTION

A Service Oriented Architecture (SOA) is a design approach for building business applications as a set of loosely coupled black box components orchestrated to deliver a well-defined level of service by linking together business processes [2]. SOA is a way of organizing applications and processes in terms of services. It is a set of principles and methodologies for designing and developing software in the form of interoperable services. These services have well-defined business functionalities that are built as software components (discrete pieces of code and/or data structures) which can be reused for different purposes. SOA is used to develop Enterprise applications by using a collection of services which communicates each other. A Service can handle a business process. Also service can access other services. It can also access traditional programs and respond to different type of requesters. Service is relatively independent of other software. This is generally known as loose coupling.

2. CHARACTERSTICS

SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language (WSDL) is the standard used to describe the services. SOA services communicate with messages formally defined via XML Schema (also called XSD). Communication among consumers and providers or services typically happens in heterogeneous environments, with little or no knowledge about the provider. Messages between services can be viewed as key business documents processed in an enterprise. SOA services are maintained in the enterprise by a registry that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration (UDDI) is the standard used for service registry. Each SOA service has a quality of service (QoS) associated with it. Some of the key QoS elements are security requirements, such as authentication and authorization,

reliable messaging, and policies regarding who can invoke services.

3. WHY SOA

The reality is that infrastructure is heterogeneous across operating systems, applications, system software, and application infrastructure [3]. Some existing applications are used to run current business processes, so starting from scratch to build new infrastructure isn't an option. Enterprises should quickly respond to business changes with agility; leverage existing investments in applications and application infrastructure to address newer business requirements; support new channels of interactions with customers, partners, and suppliers; and feature an architecture that supports organic business. SOA with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments. Some of more differences between SOA and 3-Tier Architecture are given in table 1. An overview is provided such as SOA is completely language independent whereas prior 3-Tier Architecture was completely language dependent.

TABLE 1: 3-Tier Architecture vs. SOA

3-Tier Architecture	SOA
Homogenous	Heterogeneous
Language Dependent	Language Independent
Centralized Application Tiers	Massively Distributed Services
Code Centric Applications	Flexible Composite Applications
Request/Reply Driven	Request/Reply, Pub/Sub, Events
HTML Pages	AJAX Rich Internet Applications

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

4. FEATURES

SOA starts with a simple idea – the concept of service. This makes it possible to introduce other ideas, such as service bus, service composition, and service virtualization, each of which can be applied to the architecture of an enterprise to deliver benefits [4].

a. Service

A service is a self-contained unit of software that performs a specific task. It has three components: an interface, a contract, and implementation. The interface defines how a service provider will perform requests from a service consumer, the contract defines how the service provider and the service consumer should interact, and the implementation is the actual service code itself [5].

b. Service Re-Use

In SOA, we can use already written modules/services than writing new ones. Thus, this results in less development & testing cost, also in lower maintenance cost.

c. Messaging

The services in SOA can either invoke each other directly or they invoke by exchanging messages, even if they are being executed on the same processor.

d. Message Monitoring

It is a valuable source of business intelligence. Sometimes it is also known as business activity monitoring.

e. Message Control

Message control provides application of management & security policy.

f. Message Transformation

Transformation means change from one form to other. Message transformation provides data translation - The conversion of data from one format to another through automated field mapping.

g. Message Security

Security to a message can be provided by encrypting a message or through addition of cryptographic integrity check fields [6].

h. Complex Event Processing

CEP provides generalization of software structure, since it removes functionality that is not business related.

i. Service Composition

It is the ability to develop new function combinations rapidly. In other words, putting several different services together to make a complex one is service composition.

There are two ways of composition:

a. Orchestration

b. Choreography

j. Service Discovery

When a program uses a software service, the identity of that service can be explicitly given in the program code. The benefit of service discovery is the ability to optimize performance, functionality, and cost – by selecting component services.

k. Asset wrapping

An important feature of SOA is the recognition that these assets perform services, and the development of software façades that provide access to these assets and have interfaces that are in the same form as the interfaces to other software services of the enterprise. This is called asset wrapping. In IT the assets could be hardware or software.

l. Virtualization

Virtualization can be used to enable programs that were written to use one asset to be executed with a different asset. It delivers two benefits:

a. Improved reliability

b. Ability to scale operations to meet different demand levels

m. Model Driven Implementation

Model-driven implementation refers to the automatic realization of a system or application from an abstract model. Where the model starts at a high level of architectural abstraction, it is usually referred to as Model-Driven Architecture (MDA). It provides the ability to develop new functions rapidly.

5. BENEFITS

a. Provide better business services which in the long run by reducing IT & operational cost as well as bring process efficiency.

b. SOA allows applications of an enterprise to connect with their respective partners from which you can access various types of services and have control of vital processes.

c. From a developer perspective reusing written code helps them to develop solutions more efficiently rather than trying to “re-invent the wheel”.

d. Being on a SOA stack means that your infrastructure and architecture are split up into various services. This makes it pretty obvious to write software that tends to be loosely coupled.

e. SOA gives you the flexibility to write various components of your architecture in whatever language and platform you choose to. So this means, you could write the client facing side in a more dynamic and productive language.

f. Having all your components isolated into various services makes it easy to test and debug all of them individually. Many organizations have separate teams to develop, test and maintain these components.

g. Scalability goes on a par with Flexibility. Having separate components makes it much more simpler and stress-free to scale up your architecture. You could easily scale up a particular component and test it in isolation, without affecting other components at all. This makes it easy to add in servers without facing any downtime.

h. Reusable, Since various components are built out separately, it becomes much easier to reuse them later.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

6. CONCLUSION

It is architecture. We cannot compare it to JAVA or .net. Since it is not an implementation. SOA is a way of doing things. It does not really say about how to do it. Hence if in coming time we wish to reduce the burden on development team of any organization, then SOA must be used. It is because of the features it provide.

REFERENCE

- [1] SECC_Tutorials_A Quick Introduction to SOA, Mahmoud Mohamed AbdAllah
- [2] Service Oriented Architecture For Dummies, 2nd IBM Limited Edition.
- [3] <http://www.javaworld.com>
- [4] <http://www.opengroup.org/subjectareas/soa>
- [5] <https://www.mulesoft.com/resources/esb/services-in-soa>
- [6] http://www.opengroup.org/soa/source-book/soa/soa_features.htm#soafabmsgsec