

ABSTRACT

•Service Oriented Architecture (SOA)[2] is an architectural style which allows interaction of diverse applications regardless of their platform, implementation languages and locations by utilizing generic and reliable service that can be used as application building block, SOA includes methodologies and strategies to follow in order to develop sophisticated applications and information systems. It is different from the traditional architectures [2] because it has its own unique architectural characteristics and regulations. It needs to be analyzed and clarified so as to apply the information that should be included in the architectural model of SOA correctly to service based application development. SOA is an architectural style which utilizes methods and technologies that provides for enterprises to dynamically connect and communicate software applications between different business partners and platforms by offering generic and reliable services that can be used as application building blocks. This paper aim to describe the architectural framework of SOA and its implementation in industries (i.e. mostly in small scale industries etc.) in enterprise application development.

KEYWORDS

- SOA [Service Oriented Architecture]
- Implemenattion
- Application
- Architecture
- Middleware
- Enterprise

INTRODUCTION

From the period of earliest computing units development to the present times, which we call as Information Age, software architectures [2] evolve rapidly to achieve building of sophisticated application structures capable of not only fulfilling basic functionalities expected from each computing systems, but also effecting humans life by providing corporate agility, operational efficiency and innovative improvements that result in utilization of universally shared application functionalities and services. SOA provides this vision to cope with technical complexities faced with EADD (enterprise application development and integration), as well as aligning business needs and providing coarse grained business functionalities. SOA is an architectural style and a combination of methodologies that aims to achieve interoperability of remotely or locally located homogeneous and heterogeneous applications by utilizing reusable service logic. Service orientation shows variation in adopting technology for implementation, rather than focusing on the technology itself, as SOA considers the description of the problem domain before concentrating on the usage of a specific execution environment.

Although SOA does not a direct implication of a certain technology and has been applied to software development before the invention of Web services, the capable architectures that realize the SOA vision in a more applicable way are built with Web service technologies. Driven by these competent technologies, the enterprise is practicing open standards for communication over network, messaging and description of service interfaces. SOA with Web services[1] are at the present gaining momentum, as with Web services there is fundamental improvement in SOA based application development.

It provides a strong architectural discipline and focus area centered on service creation, modeling and development, formation of process information, and service oriented integration approaches and strategies. Services are the building blocks of SOA and new applications can be constructed through consuming these services and orchestrating them within a business process. Services are reusable units for articulating common business and technology functionalities.

Implementation Requirement:

To implement a successful SOA in enterprise requires consideration of various concepts and implementation strategies, which formulate the essential characteristics of service oriented enterprise. A complete SOA implementation reflects on not only the deployment of services, but also the possibility of using them to integrate diverse application logics, and building of composite applications.

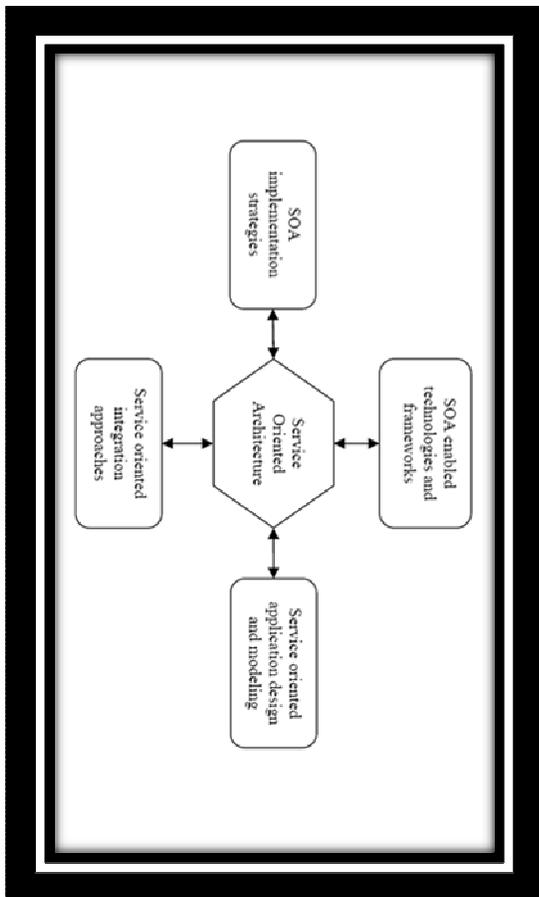


Figure 1. Service Oriented Architecture in Enterprise

Upon a successful implementation of SOA, the enterprise gain benefit by reducing development time, utilizing flexible and responsive application structure, and allowing dynamic connectivity of application logics between business partners.

Motivational approach:

SOA provide an evolutionary approach to software development, however, it introduce many distinct concepts and methodologies that needs to be defined and explained in order to understand the SOA offerings in an accurate way and build a competent architecture that satisfy the SOA vision. The main issue is to analyze and assess the differences of SOA from past architectural styles, investigate the improvement that SOA has brought to computing environment, and apply this knowledge to service based application development so as to have a satisfactorily SOA.

Since SOA is a concept independent from any certain technology and focus on the definition of architectural aspects of the application including service design, processes and modeling, it is likely to observe different variety of SOA implementation in enterprise. Especially, nowadays SOA implementation with Web Services[1] is

diverge from the initial SOA implementations done with CORBA, COM/DCOM and RMI technologies. SOA offers an extensive operability area including process flow and service oriented integration, which finally reaches to the development of a unified and single application logic, which may include various services and applications within the enterprise and aims to solve a specific business problem domain and serve widely focused united functionalities.

Service Oriented Architecture:

The architecture [2] of software explores the software system infrastructure by describing its components and high level interactions between each of them. These components are abstract modules built as a “unit” with other components. The high level interactions between components are called “connectors”.



Figure 2. Abstract Definition of Software Architecture

The concept of SOA emerged in the early 1980s and become a significant architecture in detail, it is important to evaluate the existing software development concepts and related technologies to discover the revolution of SOA so as to not to develop SOA from scratch.

Component Oriented Development:

A software component is defined as a unit of composition with contractually specified interfaces and explicit context dependencies. A software component can be deployed independently and is subject to composition by third parties. It is a group of objects [3] with has a specified interface, working together to provide an application function, look like the following figure:

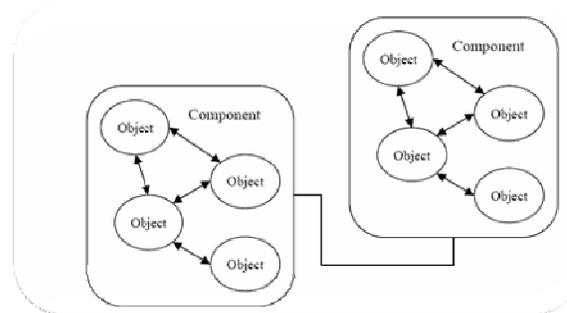


Figure 3. Component Oriented Development

A component is executed on a component execution environment provided by an application server, such as J2EE (Java 2 Enterprise Edition) container, which provides the required functions, such as transaction management and database connections pooling.

Components may be integrated to create a larger entity, which could be a new component, a component framework, or an entire system. This is called composition. The combined component acquires shared specifications from the constituent components. This is often called plug-and-play integration. Reusable components are good reflection of effective software design. Many platform vendors have already produced software infrastructures which support component oriented technology like Microsoft .NET, Sun Java EJB etc. With support of XML, Web services[1] and other standards, these technologies can interoperate for building sophisticated software applications.

Distributed Computing:

A successful SOA should overcome the difficulties faced with existing middleware technologies [5] by supporting an effective approach to application development and upcoming technologies with consideration of obtainable concepts and technologies. Typically, distributed objects[3] are supported by an Object Request Broker (ORB), which manages the communication and data exchange with potential remote objects. It provides an object[3] oriented distribution platform, location transparency and enable objects to hide their implementation details from clients.

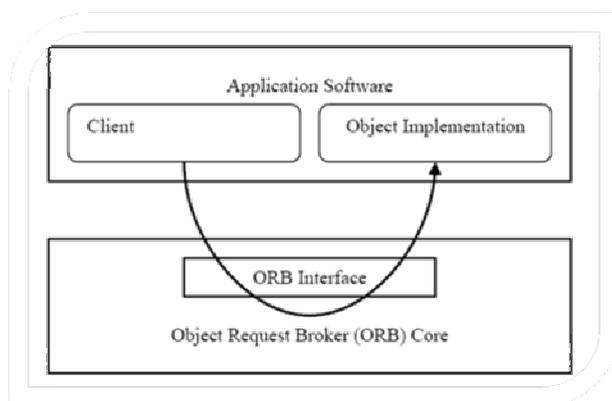


Figure 4. Object Request Broker

CORBA [7] is the most common RPC-based ORB implementation technology. It does not pay specific attention to data or program execution services, as its main aim is to provide an implementation of a proper distributed object framework.

Coupling:

A significant feature of SOA[5] is to enable a loosely-coupled[3] architectural model development in which the services are recombined and repackaged in order to build a new and compound application structure. JMS, Web

services[1], also target loosely-coupled application development.

Conclusion:

SOA works as a back-bone for component based software development providing sophisticated middle ware architecture as loosely-coupled techniques of software engineering paradigm which support the platform independency and transparency between various components to send and receive message service and connect them together. Now a days various industries introduced SOA implementation in place of their legacy system to make themselves as a big industry through various web services. The components are reused any where in the modules. Thus I say that SOA is very useful for small industries to develop their ERP system in such a way to develop in a right direction to convert all services with middle ware architecture.

Reference

1. James McGovern, Sameer Tyagi, Michael E. Stevens, Sunil Mathew *Java Web Services Architecture*, Morgan Kaufmann Publishers, 2003
2. L. Bass, P. Clements, R. Kazman *Software Architecture in Practice*, Addison-Wesley, 1997
3. E. Yourdon, L. Constantine *Structured Design*, Prentice Hall, 1975
4. G. Booch *Object Oriented Design with applications*, Benjamin-Cummings Publishing, 1990
5. Dirk Slama, Karl Banke, Dirk Krafzig *Service Oriented Architecture: Inventory of Distributed Computing Concepts*, Prentice Hall PTR, Dec. 10, 2004
6. Object Management Group, Common Object Request Broker Architecture: Core Specification, Version 3.0.3, March 2004
http://www.omg.org/technology/documents/corba_spec_catalog.htm
7. Boris Lublinsky SOA Design: Meeting in the Middle, August 20, 2004



Volume-1 Issue-1

January 2009-June 2009

Phase-III: Theme Base Paper