Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Service–Oriented Architecture: Concepts and Challenging Issues

Shikha Arya
Lecturer
Department of Computer Science
S.R.M.S.WCET, Bareilly, India

Manisha Yadav
Research Scholar
Software Engineering
S.R.M.S.CET, Bareilly, India

Abstract*— This paper presents the concept of Service –Oriented Architecture in the field of Information Technology. Service-Oriented Architecture (SOA) is an IT architectural style that supports the transformation of your business into a set of linked services, or repeatable business tasks that can be accessed when needed over a network. This may be a local network, it may be the Internet, or it may be geographically and technologically diverse, combining services in New York, London, and Hong Kong as though they were all installed on your local desktop. These services can coalesce to accomplish a specific business task, enabling your business to quickly adapt to changing conditions and requirements When SOA implementation is guided by strategic business goals, you ensure the positive transformation of your business and can realize the chief benefits on an SOA, as follows:

- ***Alignment of IT with the business***
- ***Maximal reuse of IT assets***

Keywords— SOA, distributed computing, web services, BMM, Service metadata.

I. INTRODUCTION

Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains and implemented using various technology stacks. In general, entities (people and organizations) create capabilities to solve or support a solution for the problems they face in the course of their business. It is natural to think of one person's needs being met by capabilities offered by someone else; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner. One perceived value of SOA is that it provides a powerful framework for matching needs and capabilities and for combining capabilities to address those needs by leveraging other capabilities. One capability may be repurposed across a multitude of needs.

SOA is a "view" of architecture that focuses in on services as the action boundaries between the needs and capabilities in a manner conducive to service discovery and repurposing.

Service Oriented Architecture (SOA) is an increasingly popular approach to delivering business functionality through software that is structured to provide and/or consume services. In very simple terms, a service within SOA is much like a procedure call. In concrete terms, a service, like a procedure, is implemented as a chunk of code that returns a result or causes some side effect. For example, the code might calculate and return a credit rating for an individual, reserve an airplane seat, send an email message, or start a piece of equipment.

The key aspects of a Service Oriented Architecture

SOA services have three key technical aspects:

- Applications can invoke SOA services locally or remotely via an invocation protocol that isn't tied to any language, operating system, or Web application platform
- SOA services can be invoked asynchronously as well as synchronously
- The service implementation doesn't maintain any state between invocations

The first two characteristics provide greater flexibility and interoperability than language- or platform-dependent local and remote procedure calls. SOA's pivotal contribution is the ability to make procedure calls across a broad range of environments. For example, a LANSA application running under IBM i can invoke a service implemented in C# running in a .NET environment.

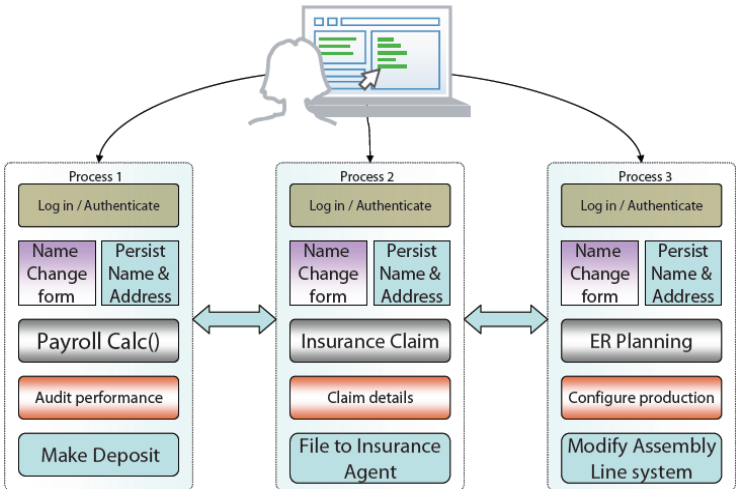
The third characteristic, so-called "stateless" services, produces looser coupling between an application using a service and the implementation of the service, which allows services to be invoked in a simple manner across the Internet .

SOA can be useful for integrating applications running entirely on a single system, especially in workflow processes or when integrating legacy and newer applications. SOA is also a great tool for distributed applications that communicate across the Internet.

Figure 1.1 Stages on the road to implementing SOA

II. REQUIREMENTS FOR SOA

Figure 2.1 shows an example of an information system scenario that could benefit from a migration to SOA. Within one organization, three separate business processes use the same functionality, each encapsulating it within an application. In this scenario, the login function, the ability to change the user name, and the ability to persist it are common tasks implemented redundantly in all three processes. This is a suboptimal situation because the company has paid to implement the same basic functionality three times.



Moreover, such scenarios are highly inefficient and introduce maintenance complexity within IT infrastructures. For example, consider an implementation in which the state of a user is not synchronized across all three processes. In this environment users might have to remember multiple login username/password tokens and manage changes to their profiles in three separate areas. Additionally, if a manager wanted to deny a user access to all three processes, it is likely that three different procedures would be required (one for each of the applications). Corporate IT workers managing such a system would be effectively tripling their work –and spending more for software and hardware systems. In a more efficient scenario, common tasks would be shared across all three processes. This can be implemented by decoupling the functionality from each process or application and building a standalone authentication and user management application that can be accessed as a service. In such a scenario, the service itself can be repurposed across multiple processes and applications and the company owning it. only has to maintain the functionality in one central place. This would be a simple example of Service Oriented Architecture in practice. The resultant IT infrastructure would resemble Figure 2.2.

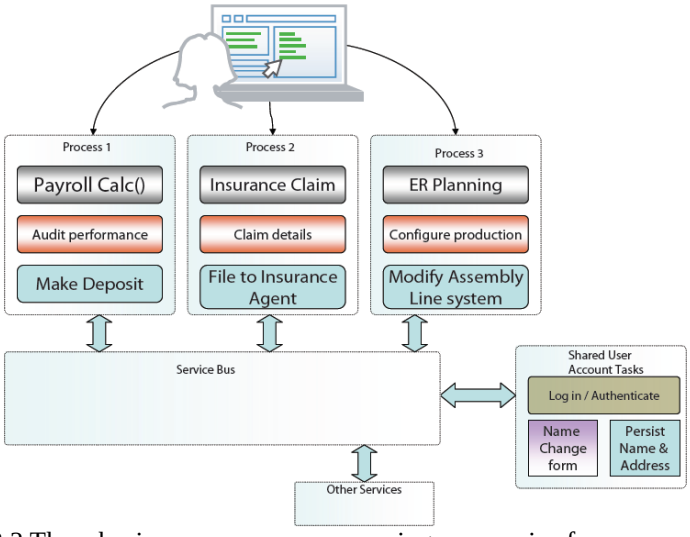


Figure 2.2 Three business processes repurposing one service for common tasks.

III. SOA MODELS

A. Basic SOA

SOA constitutes a concept to provide services to clients through published interfaces and to coordinate interaction through the exchange of messages. Generally, the basic SOA describes the relationship between three kinds of participants: the service providers, the registry, and the service requestors. The service represents a logical separation of declaration and implementation, its implementation is hidden from the client and can be subject to changes which may not influence the client so long as the service interface stays unchanged.

An important mechanism in a SOA is the Dynamic Discovery of services: The interaction model of the basic SOA consists of three key players, the service providers, the service requestors, and the intermediating directory service. First, the service providers register with the directory service, then clients can query the directory service for providers and browse the exposed service capabilities. Typically a directory service supports:

• A look-up service for clients
• Scalability of the service model: services can be added incrementally
• Dynamic composition of the services: the client can decide at runtime which services to use.

Figure.3.1 Service providers publish their interface in the service registry.

The client now can use the service registry to find out about (published) interfaces and how to bind to the selected service.

B. Extended SOA

The higher layer of the SOA pyramid provides support for service composition and management, and service orchestration, transaction, and security.

In the composition layer several atomic services can be consolidated into one composite service. Depending on their requirements clients apply atomic or composite services as applications and/or solutions. Service aggregators may utilize such composite services as components in further service compositions thus becoming service providers by publishing the service description they create. A composer of several services must encompass functionalities such as:

• Coordination: establish and manage the control of data flow among the services.
• Monitoring: subscribe to events generated by component services.
• Conformance: ensure integrity of composite service by controlling conformance of component services.
• Quality of Service (QoS): bundle QoS of component services to derive the composite QoS, e.g. performance, security, authentication, privacy, scalability, availability, etc.

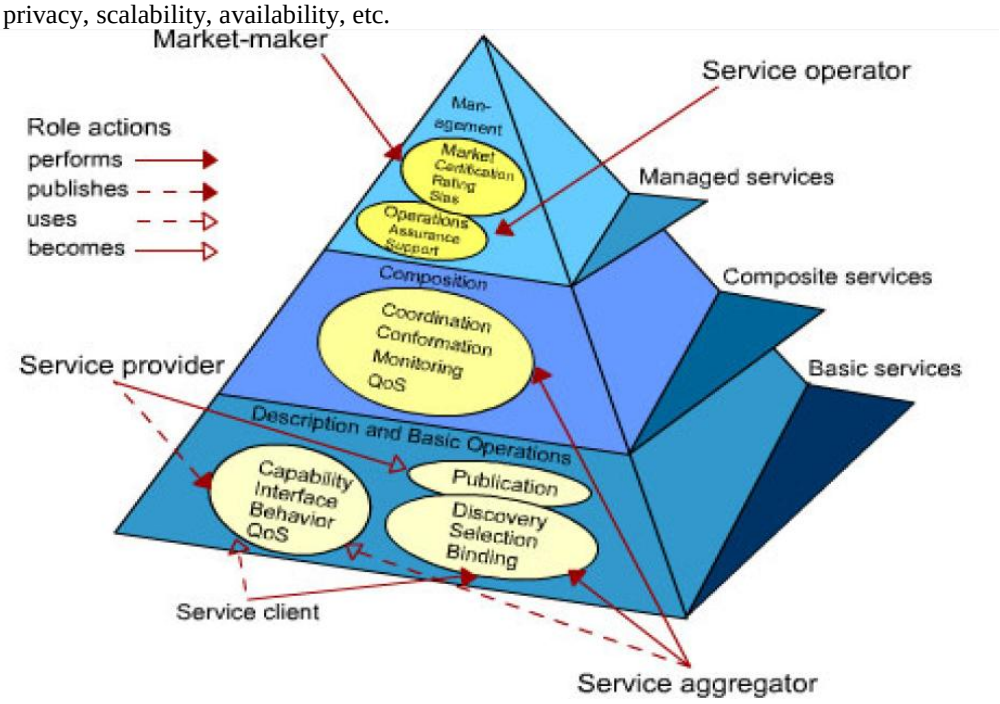


Figure 3.2 Extended Service oriented Architecture

IV. WEB SERVICES

Web service is a realization of SOA. It is important to note that the SOA is an architectural model that is independent of any technology platform and Web Services the most popular SOA implementation. As the name implies, web services offers services over the web. This is not surprising as the choice of the Internet it already connects many different systems from all over the world. In this section, we will give a brief description to web services and introduce various web service terminologies.

The application of Web services allows the constitution of an SOA. In general, a Web service is a specific kind of service which can be identified unambiguously by an URI and which uses Internet standards such as HTTP for transport. The World Wide Web Consortium (W3C) provides a more specific and accurate definition:

"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." [W3C Web Services Architecture Group] According to this definition Web services are built on top of well-known and platform-independent protocols fulfilling the key requirements of an SOA: the dynamic discovery and invocation of a service is provided by UDDI (Universal Description, Discovery and Integration), WSDL(Web Service Definition Language), and SOAP(Simple Object Access Protocol). The usage of XML supports the required platform-independence, and HTTP offers internet-wide interoperability.

V. CHALLENGING ISSUES

- One obvious and common challenge faced involves managing services metadata. SOA-based environments can include many services that exchange messages to perform tasks. Depending on the design, a single application may generate millions of messages. Managing and providing information on how services interact can become complex. This becomes even more complicated when these services are delivered by different organizations within the company or even different companies (partners, suppliers, etc.). This creates huge trust issues across teams; hence SOA Governance comes into the picture.

- Another challenge involves the lack of testing in SOA space. There are no sophisticated tools that provide testability of all headless services (including message and database services along with web services) in a typical architecture. Lack of horizontal trust requires that both producers and consumers test services on a continuous basis. SOA's main goal is to deliver agility to businesses. Therefore it is important to invest in a testing framework (build it or buy it) that would provide the visibility required to find the culprit in the architecture. Business agility requires SOA services to be controlled by the business goals and directives as defined in the business Motivation Model (BMM).

- Another challenge relates to providing appropriate levels of security. Security models built into an application may no longer suffice when an application exposes its capabilities as services that can be used by other applications. That is, application-managed security is not the right model for securing services. A number of new technologies and standards have started to emerge and provide more appropriate models for security in SOA.

  Finally, the impact of changing a service that touches multiple business domains will require a higher level of change management governance. Interoperability becomes an important aspect of SOA implementations. The WS-I organization has developed basic profile (BP) and basic security profile (BSP) to enforce compatibility. WS-I has designed testing tools to help assess whether web services conform to WS-I profile guidelines. Additionally, another charter has been established to work on the Reliable Secure Profile.

  Significant vendor hype surrounds SOA, which can create exaggerated expectations. Product stacks continue to evolve as early adopters test the development and runtime products with real-world problems. SOA does not guarantee reduced IT costs, improved systems agility or shorter time to market. Successful SOA implementations may realize some or all of these benefits depending on the quality and relevance of the system architecture and design.

VI. BENEFITS OF SOA

SOA allows you to customize your business processes without modifying source code. Making the processes in your systems match your business is a matter of configuration, not customization. That means that when it's time to upgrade to the next release, you can do so much more easily than if you had customizations scattered throughout your implementation. Service-oriented architecture can ease the integration of the diverse environments found in many organizations. SOA facilitates collaboration and information sharing throughout the organization and with external partners. By exposing business processes, service-oriented architecture helps businesses to focus on the best ways to improve operations. SOA provides the ability to support a business model that crosses organization lines. SOA enhances collaboration, facilitates end-to-end business processes, and improves operational effectiveness. It has several benefits as:

- The benefit of service-oriented architecture is that it provides the ability to streamline business processes, which in turn promotes agile business process management. SOA provides a way to make business processes more visible so they can be customized and optimized to better meet increasing customer demands for reduced response time while maintaining high quality and reliability. And perhaps most importantly, keep the complexities of application-

to-application and business-to-business integration at arms length, significantly reducing costs and raising technology to a business level.

- In architectural terms, a modern architectural design should be Service Oriented, loosely coupled, driven by events, able to support both integration and assembly, aligned with valuable life cycle support processes, and able to leverage existing infrastructure and applications. When it comes to Service Oriented Architecture, it tends to offer a variety of different advantages over more traditional methods of distributing computing. These include offering business services across several platforms; providing location independence; providing authentication as well as authorization support on ever tier; a loosely coupled approach; and dynamic search and connectivity to other services. At the same time, it allows that services do not have to be located on a particular system or network.

- Some of the short term benefits of Service Oriented Architecture implementation include an enhancement of reliability; a reduction of hardware acquisition costs; an acceleration of movement towards standards based servers and application consolidation; the leveraging of existing development skills; and the providing of a data bridge that connects previously incompatible forms of technology.

- There are also numerous long-term benefits of Service Oriented Architecture implementation. These include the creation of a self healing infrastructure that effectively reduces management costs; the proven ability to build composite applications; the access to truly real time decision making applications; the resulting compilation of a unified taxonomy of data across an enterprise, which includes both partners and customers; and a whole lot more.

- From the perspective of Business Value, the advantages of Service Oriented Architecture include the ability to meet customer demands at a much faster pace than ever before; a reduction of costs previously associated with the maintenance and acquisition of key technological needs; the management of business functionality at a physically closer location to the business units; a reduction in reliance on pricey custom development; and a leverage of existing investments in the technological sector.

- One of the most important benefits of SOA is its ease of reuse. Therefore accountability and funding models must ultimately evolve within the organization. A business unit needs to be encouraged to create services that other units will use. Conversely, units must be encouraged to reuse services. This requires a few new governance components:

    1. Each business unit creating services must have an appropriate support structure in place to deliver on its service-level obligations, and to support enhancing existing services strictly for the benefit of others. This is typically quite foreign to business leaders.
    2. Each business unit consuming services accepts the apparent risk of reusing services outside their own control, with the attendant external project dependencies, etc.

VII. CONCLUSION

As enterprise systems tend to become more and more complex, system designers must ensure the scalability of the systems and how they communicate with each other. This task requires more and more effort both in terms of business functionalities and regarding the technologies chosen. Using SOA approach will reduce the difficulty to a certain level. Generally in SOA, systems expose functionalities for other systems to consume without knowing the complex logic behind. As a matter of fact, those consumers, in turn, could be service providers as well. Hence, service oriented architecture is an excellent option to scale the systems. Since more and more systems such as SAP are moving towards SOA, we believe that SOA is creating a new trend today. Being conceived and developed a long time ago, however, only until now, SOA proves its important role as the backbone of enterprise systems and an essential building block in cloud computing, which is now deemed as a promising technology in the future. SOA services are independent of each other and consequently they cannot exchange information by passing storage addresses such as monolithic programs do. However, a possible solution to this problem is to pass references to data processing services instead of the data itself. Client applications may not be able to work if a service on a remote server is temporarily not available.

REFERENCES

[1] Bell, Michael (2008). "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. p. 3. ISBN 978-0-470-14111-3

[2] Bell_, Michael (2010). "SOA Modeling Patterns for Service-Oriented Discovery and Analysis". Wiley & Sons. p.390. ISBN 978-0-470-48197-4.

[3] Microsoft Windows Communication Foundation team (2012). "Principles of Service Oriented Design" msdn.microsoft.com. Retrieved September 3, 2012.

[4] Mike P. Papazoglou, Willem-Jan van den Heuvel "Service oriented architectures: approaches, technologies and research issues", Springer Verlag 2007.

[5] Birol Berkem, GooBiz (Paris / France) From " The Business Motivation Model(BMM) To Service Oriented Architecture (SOA)", Vol. 7, No. 8, November-December 2008.

[6] Principles by Thomas Erl of SOA Systems Inc. "eight specific service Orientations principles".

[7] Erl, Thomas. About the Principles. Serviceorientation.org, 2005–06.
[8] The Microsoft website . http://msdn.microsoft.com/en-us.
[9] http://www.lansa.com/resources/
[10] Duane Nickull,Laurel Reitman, James Ward,jack Wilber "Service Oriented Architecture (SOA) and Specialized Messaging Patterns", Technical White Paper.
[11] Colan, M.: Service-Oriented Architecture expands the vision of Web services, Part 2. IBM DeveloperWorks, April 2004
[12] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: "Web Services:Concepts, Architectures and Applications". Springer, Heidelberg (2004)

