



SOA- the backbone of Modern Business

MANISH SINGH

Allahabad Institute of Engineering & Technology
Gohania, Allahabad
manish.singh076@gmail.com

ABSTRACT

The present study examines about the key features of Service Oriented Architecture (SOA) and its utilization in modern business integration like various processes, employees, information, managers etc. Integration is the biggest challenge which is fulfilled by implementing of SOA in business. It provides more strength to business working as a backbone for various services and connecting the cross-platform application together.

KEYWORD

**Service Oriented
Architecture**

Application

Implementation

**Information
Technology,**

Web Services

Cross-Platform

Dreface

Modern Businesses need to integration among processes, people, employee, managers, executives and information within the organization and across organizational boundaries to subsidiaries or trading partners[1]. If a lack of integration among Information Technology (IT) assets—systems, applications and data—makes it difficult for IT to respond quickly and effectively to changing business needs. Due to this inflexibility increases costs, decreases customer responsiveness, hinders compliance, and decreases worker productivity. In short, a lack of integration is the biggest challenge that organizations face in their efforts to remain competitive and grow. Therefore it is required that introduce a new web technology in business by which the integration retain in industries and fill the gap in communication, maintain consistency in data base. The present paper explore the concept to SOA (Service Oriented Architecture) and their benefits which should be utilizes by modern business.

Why Service Oriented Architecture?

Service Oriented Architecture (SOA) is a design approach which helps to organizing existing Information Technology assets such that the heterogeneous (mixed) array of distributed, complex systems and applications can be transformed into a network of integrated, simplified and highly flexible resource(s). A well-executed SOA project aligns IT resource(s) more directly with business goals, helping organizations to build stronger connections with customers and suppliers, providing more accurate and more readily available business intelligence with which to make better decisions, and helping businesses streamline business processes and information sharing for improved employee productivity. The net result is an increase in organization agility.

SOA distinguishes three different roles of services: service provider, service consumer and service registry. It postulates a general protocol for interaction: A service provider registers at the service registry by submitting information about how to interact with its service. The service registry manages such information about all registered service providers and allows a service consumer to

find an adequate service provider. Then, the service of the provider and the service of the consumer may bind and start interaction.

A service has two kinds of interfaces: *required* and *provided* interfaces. Required interfaces specify which services are used by the service. In contrast, provided interfaces specify which services are offered by the service. So in terms of the service roles in SOA a service plays the consumer's role at the required interfaces and at the provided interfaces it plays the provider's role.

Apart from these technical paradigms services in SOA are also based on an *economical paradigm*: a service is comparable with a business unit. So it should create *value* for its environment. Therefore the two kinds of interfaces can be seen as the *buy side* and the *sell side* of the service. On the buy side a service behaves as a service consumer or *client* and buys other services. On the sell side a service behaves as the service provider and offers its service to other services. Services are operating as actors on a market place. This means, they offer their services to any consumer who needs it and they buy services from providers with the best value proposition. So both parties publish their needs and offerings at a repository, respectively.

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.

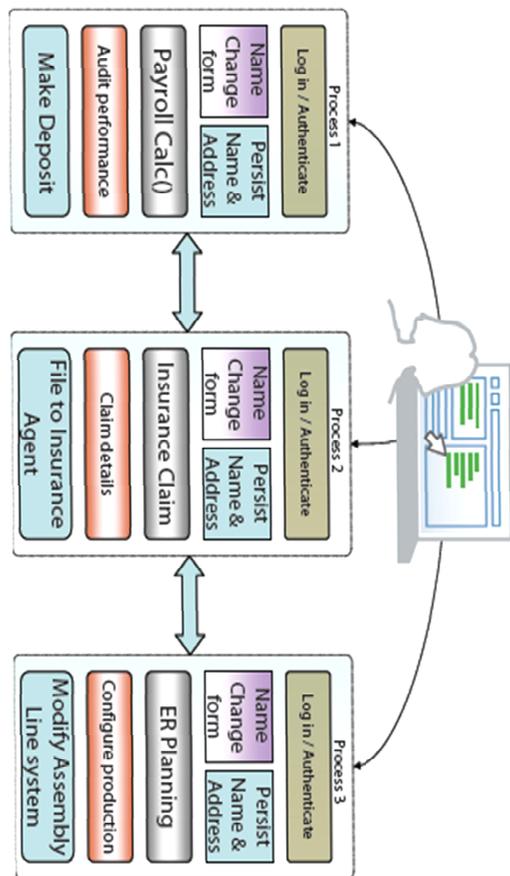


Figure 2.1 – three business processes within one company duplicating functionality

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.

IT Service Management

IT Service Management provides visibility into applications as well as the supporting infrastructure. Service Management services provide reporting and monitoring services.

It supports IT service management through both features core to its BIG-IP platform as well as stand-alone products designed specifically to provide core service management features as well as additional functionality.

Enterprise Manager™ is a stand-alone device designed to aid in the configuration and management of large BIG-IP platform installations. Enterprise Manager provides configuration management services, and can act as an aggregation point for reporting and monitoring services across all BIG-IP platform instances. Additionally, F5’s BIG-IP platform contains a number of features that support reporting and monitoring services, as well as the means by which both can be easily tailored to provide better visibility into both the organizational SOA as well as its supporting Service-Oriented Infrastructure.

Infrastructure Services

The core purpose of any SOI should be to support the availability, scalability, and optimization of SOA-based services and applications. These functions are provided through product features such as:

- Load balancing
- Compressing
- Caching
- TCP connection management
- Advanced health monitoring of services

Effectiveness of SOA

While a well planned and executed SOA undertaking can help organizations realize greater responsiveness in a changing marketplace, not all service oriented efforts have been successful. SOA projects have limited success when they are driven from the bottom up by developers; building SOA for

the sake of SOA without reference to the business context is a project without organizing principles and guidance; the result is a chaotic implementation that has no business relevance. On the other hand, taking a **top-down mega-approach** to SOA requires such enormous time investments that by the time the project is complete, the solution no longer maps to business needs.

Web Services and SOA

It is the ability to readily change and optimize business processes is the success key to organizational competitiveness and growth. Organizational agility can be compromised when supporting Information Technology assets cannot respond flexibly to changing business needs. Unlocking IT resources from their application silos and making their functionality broadly available across the organization promotes business process optimization and organizational agility.

Service Oriented Architecture (SOA) is a design approach to organizing existing IT assets such that the heterogeneous array of distributed, complex systems and applications can be transformed into a network of integrated, simplified, and highly flexible resources. A well-executed SOA project aligns IT resources more directly with business goals, helping organizations to build stronger connections with customers and suppliers, providing more accurate and more readily available business intelligence with which to make better decisions, and helping businesses streamline business processes and information sharing for improved employee productivity. The result is the unlocking of IT resources for the increased agility that organizations seek[1].

Service Oriented Architecture provides the design framework to integrate soloed applications so that their functionality can be accessed as services on a network. Most commonly implemented through standards-based, technology-neutral Web Services [3], SOA breaks down monolithic applications into a suite of services, implementing functionality in a modular fashion.

From the business perspective, a service performs a specific task; as such, it can map onto a business process as simple as inputting or outputting a field of data such as 'customer ID'

alternatively services can be aggregated into a composite application that provides a higher-order service as complex as 'fill customer order,' a process that end to end spans multiple business applications.

Web services as illustrate by **Figure 1**, are the most common way to implement SOA. Web services are applications that use standard transports, encodings, and protocols to exchange information. It enable computer systems on any platform to communicate, and are used in a range of application integration scenarios, both within the organization and among trading partners.

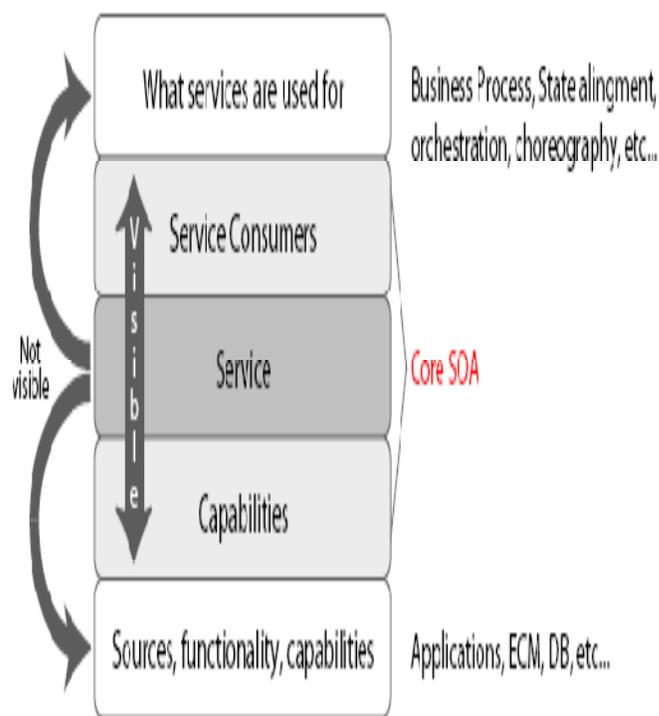


Figure 1: Web services within overall architecture in SOA [3]

The service oriented approach allows the creation of services and composite applications that exist independent of the underlying technologies.

At the beginning with version 1.0 of the .NET Framework, Microsoft investments in tools together with the intrinsic support for Web services in the Windows platform have helped make Service Orientation mainstream. Shortly thereafter, Microsoft worked with IBM to develop the Web services Interoperability Organization (WS-I) which promotes interoperability across platforms, operating systems

and programming languages. Now grown to nearly 150 member companies, WS-I has created Web services that address critical areas such as interoperability, security, and the reliability of messaging.

Microsoft's real world approach has helped organizations of all sizes to optimize their business processes and to realize greater business agility and faster time to value through use of Microsoft's SOA design principles, best practices, tools and technologies.

The concept of **SOA** can be viewed from several perspectives, which help in understanding the underlying architectural requirements of the architecture.

There are three abstract capability layers exposed within a **SOA**.

1. **Expose**: focuses on how existing IT investments are exposed as a set of broad, standards-based services, enabling these investments to be available to a broader set of consumers. A Service Implementation Architecture describes how services are developed, deployed and managed.

2. **Compose** : focuses on combining services into applications or cross-functional business processes. A Service Integration Architecture describes a set of capabilities for composing services and other components into larger constructs such as business processes.

3. **Consume** : focuses on delivering new applications that enable increased productivity and enhanced insight into business performance. A Service Oriented Application Architecture describes how "composed services" are made available for consumption through business processes, new services or new end-user applications.

Focus on the Business Drivers

Before a developer writes a single line of code, it is critical to identify both specific business drivers of the SOA endeavor and the dependencies between the business and the underlying

technologies. Two approaches are commonly pursued for implementing SOA:

- Top-down and ,
- Bottom-up.

Both approaches have possible pitfalls that can prevent success. Most of the organizations that have attempted to roll out SOA infrastructure through a top-down approach have discovered that when the infrastructure is finally delivered it is out of sync with the needs of the business. Likewise, a bottom-up approach can fail as well, because it can lead to a chaotic implementation of services created without regard to organizational goals.

The "middle-out" approach is a successful hybrid of the two other approaches. Business drivers and strategic vision are first employed to set clear direction and priorities. Based on these, the organization takes multiple iterative steps to build out slices of end-to-end capabilities, with each iteration delivering a new, dynamic application back to the business that is used to create business return. Company like Microsoft has long advocated this "real-world" approach to leveraging service-oriented architectures: The approach is focused on rapid time-to-value, and it delivers business results through iterative, incremental steps that facilitate close alignment of IT resources with changing business conditions.

Benefits of SOA

SOA benefits accrue for the organization at two different levels, that of the business user and that of the IT organization. From the business point of view, SOA enables development of a new generation of dynamic applications that address a number of top-level business concerns that are central to grow and competitiveness. SOA solutions promote:

1. **Enhanced Business Decision Making** - By aggregating access to business services and information into a set of dynamic, composite business applications, decision makers' gain more accurate and more comprehensive information. People, processes and systems spanning multiple departments can more readily be mapped into a single unified view, enabling organizations to better understand the cost tradeoffs that they are making in daily

business operations. And by providing better information faster, organizations can react more quickly to problems as they arise.

2. **Increases Employee productivity** - By providing streamlined access to systems and information and enabling business process improvement, businesses can drive greater employee productivity. Employees can focus their energies on addressing the important, value-added processes and on collaborative, semi-structured activities, rather than having to conform to the limitations and restrictions of the underlying IT systems. Moreover, since end-users can access information in the form and presentation factor (web, rich client, mobile device) that meets their needs, productivity is enhanced.
3. **Stronger connections with customers and suppliers** -The benefits of SOA extend beyond organizational boundaries. Mergers and acquisitions become more profitable, since it is easier to integrate disparate systems and applications. Integration with trading partners and streamlining of supply chain processes are readily attainable goals. Providing more responsive customer service is enabled, as are new customer initiatives, such as one-stop service portals. By making available dynamic applications and business services to external customers and suppliers, not only is richer collaboration is possible, but customer/partner satisfaction is increased. SOA unlocks critical supply and demand chain processes—such as outsourcing of specific business tasks—from the constraints of underlying IT architectures, thereby enabling better alignment of processes with organizational strategy.

Technical Benefits:

From the IT department's point of view, service orientation provides the framework through which to simplify the creation and management of integrated systems and applications, and a way to align IT assets with the business model and changing business needs.

1. **More productive, more flexible applications.** The service oriented approach enables IT to make existing IT assets—including legacy systems and applications—more productive and more profitable to the business without the need for custom-coded one-off integration solutions. Service orientation also enables the development of a new generation of composite applications that provide cross-functional capabilities to the organization irrespective of the underlying platforms and programming languages. Moreover, since services are uncoupled from their underlying IT infrastructure, there is inherently greater flexibility in solution design.
2. **Faster, more cost-effective application development.** Standards-based service design enables IT to create a repository of reusable services that can be combined into higher level services and composite applications as new business needs arise. This lowers the cost of solution development and testing, reduces redundancy, and speeds time to business value. And the use of a single development model and framework simplifies and standardizes application building, testing and maintenance.
3. **More manageable and secure applications.** Service oriented solutions provide a common infrastructure (and documentation) for developing secure, monitored, and predictable services. As business needs change, SOA makes it easier to add in new services and

capabilities that map onto critical business processes. Because services are accessed rather than the applications themselves, service orientation provides the means for protecting existing IT investments without inhibiting the deployment of new capabilities. And since a strong authentication and authorization model is used for all services—as well as because services exist independently of one another and cannot therefore impact other services—the SOA approach provides greater overall security.

Discussion:

Present study stated that if business introduced the Service Oriented Architecture (SOA) then they take various benefits not only in the application part but also in the integration of person to person upward and downward as well as linear. It provides more flexibility in solution design, cost effective in secure application development and improves the decision making. The productivity of employee no doubt increased. According to my opinion the implementation of SOA makes the smaller business to large because it exploits the business world wide. So I suggest to modern business to implement the SOA and take its cross-platform real benefits.

Reference

- i. Enabling "Real World SOA" through the Microsoft Platform, Microsoft Corporation, Published: December 2006.
- ii. UFIDA U9 ERP Software: Optimizing the SOA Implementation, White Paper, Published: 08-12-08
- iii. Duane Nickull, Service Oriented Architecture (SOA) and Specialized Messaging Patterns, Technical White Paper
- iv. Abrial, J.R., Schuman, S.A., Meyer, B.: Specification language. In: On the Construction of Programs. (1980)343{410
- v. Jones, C.: Systematic Software Development using VDM. Prentice Hall (1990)
- vi. High, R., Kinder, S., Graham, S.: IBM's SOA Foundation { An Architectural Introduction and Overview. Technical Report 1.0, IBM (2005)
- vii. Beisiegel, M., Booz, D., Edwards, M., Hurley, O., Ielceanu, S., Karmarkar, A., Malhotra, A., Marino, J., Nally, M., Newcomer, E., Patil, S., Pavlik, G., Rowley, M., Tam, K., Vorthmann, S., Waterman, L.: Service Component Architecture { Assembly Model Specification. SCA Version 0.96 draft 1, August 2006, BEA, Cape Clear, IBM, Interface21, IONA, Oracle, Princeton, Progress Software, Red Hat, Rogue Wave, SAP, Software AG, Sybase, TIBCO (2006)
- ix. McIlroy, M.D.: Mass Produced Software Components. In Naur, P., Randell, B., eds.: Proceedings of NATO Software Engineering Conference. Volume 1., Garmisch, Germany (1968) 138{150
- x. Szyperski, C.: Component Software{Beyond Object-Oriented Programming. Addison-Wesley and ACM Press (1998)
- xi. Messerschmitt, D., Szyperski, C.: Software Ecosystem{Understanding an Indispensable Technology and Industry. MIT Press (2003)
- xiii. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. 2nd edition Wesley Professional (2003)
- xiv. Chinnici, R., Moreau, J.J., Ryman, A., Weerawarana, S.: Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Candidate Recommendation 27 March 2006, W3C (2006)
- xv. White, S.A.: Business Process Modelling Notation Version 1.0. Technical report, Business Process Management Initiative (BPMI) (2004)
- xvii. Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, Version 1.1.
- xix. Specification, BEA Systems, International Business Machines Corporation, Microsoft Corporation (2003)
- xx. B.Aorger, E., St.Åark, R.: Abstract State Machines. A Method for High-Level System Design and Analysis. Springer-Verlag (2003)
- xxii. Abrial, J.R.: The B-Book: Assigning Programs to Meanings. Cambridge University Press (1996)
- xxiii. Christensen, E., Curbera, F., Meredith, G., Weeravarana, S.: Web Service Description Language (WSDL) 1.1. C Note 15 March 2001, Ariba, International Business Machines Corporation, Microsoft (2001)
- xxv. Object Management Group: UML2.0 Object Constraint Language (OCL) Specification. Specification, Object Management Group (OMG) (2003)
- xxvi. Aalst, W.: The application of Petri nets to workflow management. Journal Circuits, Systems and Computers 8(1) (1998) 21{66}
- xxvii. Reisig, W.: Petri Nets. EATCS Monographs on Theoretical Computer Science edn. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo (1985)
- xxviii. Billington, J., Christensen, S., Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., Weber, M.: The Petri Net Markup Language: Concepts, Technology, and Tools. In Aalst, W., Best, E., eds.: Applications and Theory of Petri Nets 2003, 24th International Conference, ICATPN 2003, Eindhoven, The Netherlands, June 23-27, 2003, Proceedings. Volume 2679 of Lecture Notes in Computer Science. Springer (2003) 483{505}
- xxix. Beatty, J., Blohm, H., Boutard, C., Brodsky, S., Carey, M., Dubray, J.J., Ellersick, R., Ho, M., Karmarkar, A., Kearns, D., Brettevillos, R.L., Nally, M., Preotiu-Pietro, R., Rowley, M., Smith, S., Yan, H.: Service Data Objects For Java. Specification, Version 2.01, IBM, BEA, Oracle, SAP, Siebel, Xcalia, Sybase (2005)
- xxx. White, S.A.: Business Process Modelling Notation Version 1.0. Technical report, Business Process Management Initiative (BPMI) (2004)
- xxxi. Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte,

- xxxii. S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, International Business Machines Corporation, Microsoft Corporation (2003)
- xxxiii. Böger, E., Stark, R.: Abstract State Machines. A Method for High-Level System Design and Analysis.
- xxxiv. Abrial, J.R.: The B-Book: Assigning Programs to Meanings. Cambridge University Press (1996)
- xxxv. Christensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Service Description Language (WSDL) 1.1.W3C Note 15 March 2001, Ariba, International Business Machines Corporation, Microsoft (2001)
- xxxvi. Object Management Group: UML2.0 Object Constraint Language (OCL) Specification. Specification, Object Management Group (OMG) (2003)
- xxxvii. Aalst, W.: The application of Petri nets to workflow management. *Journal of Circuits, Systems and* 8(1) (1998) 21{66}.
- xxxviii. Reisig, W.: Petri Nets. EATCS Monographs on Theoretical Computer Science Edition. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo (1985).
- xxxix. Billington, J., Christensen, S., Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., Weber, M.: The Petri Net Markup Language: Concepts, Technology, and Tools. In Aalst, W., Best, E., eds: Applications and Theory of Petri Nets 2003, 24th International Conference, ICATPN 2003, Eindhoven, The Netherlands, June 23-27, 2003, Proceedings. Volume 2679 of Lecture Notes in Computer Science., Springer (2003) 483{505}.
- xl. Beatty, J., Blohm, H., Boutard, C., Brodsky, S., Carey, M., Dubray, J.J., Ellersick, R., Ho, M., Karmarkar, A., Kearns, D., Brettevillos, R.L., Nally, M., Preotiu-Pietro, R., Rowley, M., Smith, S., Yan, H.: Service Data Objects For Java. Specification, Version 2.0.1, IBM, BEA, Oracle, SAP, Siebel, Xcalia, Sybase (2005)



<http://www.karamsociety.org>