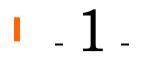# SOA (Service Oriented Architecture)

By: Amit Bhatt

Abstract

The ability to readily change and optimize business processes is the key to organizational competitiveness and growt Organizational agility can be compromised when supporting IT assets can not flexibly respond 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 that promotes better alignment of IT with business needs, enabling employees, customers, and trading partners to more quickly respond and adapt to changing business pressures. This paper explores the business rationale for SOA, as well as the approach to SOA.

Contents

Introduction

Businesses need to connect processes, people, and information both within the organization and across organizational boundaries to subsidiaries or trading partners. A lack of integration among IT assets—systems, applications and data—makes it difficult for IT to respond quickly and effectively to changing business needs. 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.

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 net result is an increase in organization agility.

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.

A "middle out" approach can be followed though. In this approach, SOA efforts are driven by strategic vision and business needs, and are met through incremental, iterative SOA projects that are designed to deliver on business goals one business need at a time

This paper describes the business context out of which grew the need for a service oriented approach, and the "middle out" approach to achieving successful SOA implementations. Finally the paper introduces key SOA tools and technologies offered as an example.

The Business Context

Organizations are increasingly reliant on their IT infrastructure to help drive success. But in an increasingly competitive marketplace, capitalizing on new business opportunities means moving fast. All too often, however, IT cannot move fast enough or with enough flexibility to compete effectively. Too much organizational inefficiency get in the way: a preponderance of manual, error-prone processes; inefficient information sharing across the organization; an inability to track business processes from end to end; too much time tracking required information for adherence to government regulations; inefficiencies in customer service. To the extent to which these areas are a challenge, worker productivity is negatively impacted, and growth and competitiveness are compromised.

At the core of these challenges is information. Not a lack of it—in fact, information is the organization's fastest growing asset—but rather a lack of readily accessible, relevant information presented to stakeholders in a coherent, unified manner. The roots of the problem lie in the fact that while legacy and line of business applications typically support rich functionality—whether in finance, marketing, customer relationships or other areas—they cannot readily share information with one another, and therefore cannot effectively provide insight into business processes that span cross-functional areas. Some form of human intervention is necessary to achieve this information transfer across disparate and often incompatible systems, whether within or across organizational boundaries.

In the past, this intervention was one of two kinds. One is to manually re-key one system's data into the other incompatible system. The other is to write code for an interface specifically designed to enable information sharing between two incompatible applications. Both solutions are costly and inefficient— the first because it is an ongoing process of duplicate, error-prone effort, and the second because such a tightly coupled means of obtaining interoperability is not only difficult to create but a nightmare to maintain: a change in either application can break the interface between them, making communication impossible.

What is needed is both a standards-based means of integrating diverse systems and applications over a heterogeneous set of platforms and communication protocols, as well as a loosely-coupled means of achieving integration so that the underlying infrastructure promotes rather than impedes further change as business needs evolve.

Born SOA

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

What exactly is a service? A service is an independent piece of functionality that can be discovered on the network, and that describes both what it can do and how it can be interacted with. 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.

The service oriented approach allows the creation of services and composite applications that exist independent of the underlying technologies. Rather than requiring that all data and logic reside on a single computer, the service model facilitates access and consumption of IT resources over the network. Since services are designed to be standalone, autonomous, and loosely coupled, they can be readily combined and recombined into composite applications according to the changing needs of the organization. Composite (also known as dynamic) applications are what enable a businesses to improve and automate manual tasks, to realize a consistent view of customers and trading partners, and to orchestrate business processes that comply with internal mandates and external regulations. The net result is that organizations adopting service orientation can create and reuse services and applications as business needs evolve, and are thereby able to gain the agility necessary for superior marketplace performance.

Web Services

Using a SOA-based design approach does not require Web services; however, as previously mentioned, Web services is the most common way to implement SOA. Web services are applications that use standard transports, encodings, and protocols to exchange information. Web services 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.

Web services are based on a core set of communication standards, including XML for representing data, the Simple Object Access Protocol (SOAP) for data exchange, and the Web Services Description Language (WSDL) to describe the capabilities of a Web service. Additional specifications, collectively referred to as the WS-* architecture, define functionality for Web services discovery, even ting, attachments, security, reliable messaging, transactions, and management. Web services model initiated a wave of innovation that has fundamentally changed the application architecture landscape

Here Goes SaaS

Another concept closely related to SOA is the notion of Software as a Service (or SaaS). Simply put, SaaS can be defined as "software deployed as a hosted service and accessed over the Internet."

SaaS as a concept is often associated with the application service providers (ASPs) of the 1990s, which provided "shrink-wrap" applications to business users over the Internet. These early attempts at Internet-delivered software had more in common with traditional on-premise applications than with modern SaaS applications in some ways, such as licensing and architecture. Because these applications were originally built as single-tenant applications, their ability to share data and processes with other applications was limited, and they tended to offer few economic benefits over their locally installed counterparts.

Today, SaaS applications are expected to take advantage of the benefits of centralization through a single-instance, multi-tenant architecture, and to provide a feature-rich experience competitive with comparable on-premise applications. A typical SaaS application is offered either directly by the vendor

or by an intermediary party called an aggregator, which bundles SaaS offerings from different vendors and offers them as part of a unified application platform.

In contrast to the one-time licensing model commonly used for on-premise software, SaaS application access is frequently sold using a subscription model, with customers paying an ongoing fee to use the application. Fee structures vary from application to application; some providers charge a flat rate for unlimited access to some or all of the application's features, while others charge varying rates that are based on usage.

SaaS also embraces a foundation of service orientation. For the purpose of this whitepaper, we will refer broadly to SOA as encompassing both on-premise and hosted services "in the cloud". We consider SaaS to be a strategic component of any customer's SOA strategy.

Business Process Management (BPM)

The topic of Business Process Management (BPM) is also frequently related to SOA. BPM is a management discipline that combines a process-centric and cross-functional approach to improving how organizations achieve their business goals. A BPM solution provides the tools that help make these processes explicit, as well as the functionality to help business managers' control and change both manual and automated workflows.

Business process management has its origins in total quality management and business process reengineering. While it adds to these a technological framework, it is more than just the combination of these disciplines. BPM is an IT enabled management discipline that promotes organizational agility and supports the efforts of people to drive process change and rapid innovation. As such, BPM supports the alignment of IT and business activities both within the organization and with business partners and suppliers.

Business processes may be structured or unstructured, depending on the extent to which the underlying steps are fixed and therefore automated or changeable and generally executed by people or people interacting with systems. People are a critical part of nearly every business processes – they drive the solutions and insight that advances a business, so the goal should be to empower them to create new innovation and be more productive (and not to "re-engineer" people out of the process).

While BPM can be pursued separately from SOA initiatives, the ability to quickly and flexibly define new business processes is made much easier if you also have exposed your system IT resources in a service oriented manner. For the purpose of this whitepaper, we will refer to the composition and monitoring of business processes as a key part of developing a SOA strategy.

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:

- **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.

- **Greater 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.

- **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.

SOA also helps organizations document their business model, and to use the documented business model to capture changing business dynamics and to optimize accordingly.

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.

- **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.

- **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.

- **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.

Overcoming SOA Challenges

Embarking on an SOA project has its challenges, both at the organizational and technical level, and these challenges will be exacerbated if expectations are that SOA is a cure-all for all ailments.For SOA initiatives to be successful, it is critical to:

- **Define clear business goals**. The first step in any SOA undertaking is to clearly identify critical business problems or challenges. The more precisely these can be defined the easier it will be to determine the scope and direction of each SOA project. And by setting clear vision and direction from the top, it will be easier to obtain buy in on projects that are cross-functional in nature.

- **Clearly define the scope of the SOA project**. The goal of any given SOA project should not be to massively renovate the entire IT infrastructure. Such mega undertakings fail to realize their lofty goals because by the time the solution is built business conditions have changed enough that the problems the organization face are no longer the ones that the infrastructure has been built to solve. The real goal of each SOA endeavor is to take on specific business challenges and create solutions in iterative, incremental steps.

- **Avoid SOA for the sake of SOA**. SOA should not be technology driven—it should be driven by the business needs of organization. Taking the opposite approach—building SOA for the sake of SOA and creating services without clear business meaning, without appropriate granularity, or with too many interconnections will result in a convoluted, unmanageable, and costly implementation.

- **Manage the process**. Service and applications map to processes and desired business outcomes across functional areas of the organization. Because they represent shared processes, it is critical that ownership be assigned so that they can be inventoried and managed in order to ensure that the needs of the organization continue to be met.

Realizing a SOA Project: An Example

Having discussed SOA from a relatively high level, it's time to drill a little deeper in on how an SOA project actually unfolds in the organization. We'll use a fictitious case study example to illustrate what a typical project approach could look like.

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. Neglecting the business context can result in a project in which SOA infrastructure is pursued for its own sake, or where investments are made that do not line up well with the needs and priorities of the business.

Two approaches are commonly pursued for implementing SOA: top-down and bottom-up. Both approaches have possible pitfalls that can prevent success. Many 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 following example illustrates the process of beginning with defining clear business drivers.

Example: Business drivers for SOA in the enterprise

BP Low Electronics is an electronics maker in Low, New Zealand with a partially-owned manufacturing subsidiary in Tripura, India. The company is experiencing a number of competitive pressures, and is looking to gain advantage by managing their value chain more efficiently. There is general consensus within the company that in order to do so, they need to identify and capitalize on opportunities to consolidate and automate business activities within the value chain.

Order fulfillment at BP Low Electronics spans sales, accounting, inventory management, and product transport. Product shipping, because it involves significant coordination with third parties—including freight consolidators, warehousing, and trucking—has been especially problematic. And exception handling—getting the right information to the right person to rectify problems as they arise—has been inefficient because of the many manual steps involved. The net result is that the company has incurred additional overhead costs and in some cases lost business.

The head of operations assigns a project manager to create an order fulfillment process inventory, with the goal of understanding people's roles, their tasks, and the technologies that support the process from end to end. As the process inventory list grows, it becomes increasingly apparent that the numerous manual steps around customer quoting, inventory tracking, and scheduling logistics are adding time, errors and inefficiencies to the process; moreover the lack of integration among legacy and line of business applications—the supporting customer databases, inventory, and accounting

systems—does not facilitate the level of business intelligence that is needed to respond quickly to changes in the order fulfillment process.

Company executives want to proceed with integration and automation to help remove inefficiencies in the order fulfillment shipping process, and IT staff has suggested that the SOA approach to integration can help BP Low Electronics create a flexible, connected technology infrastructure. The business vision and drivers outlined by BP Low Electronics executives are used to identify and scope out a set of well-defined and targeted SOA capabilities that are needed to support their desired "to-be" process.

An Incremental Delivery Approach

Once the business drivers are defined, the process of implementing the technology can begin. Based upon the clearly defined and prioritized vision, each implementation project is an iterative one of creating ("exposing") new services, aggregating ("composing") these services into larger processes, and making the outputs available for use ("consuming") by the business user.

Expose

The expose phase of the SOA approach focuses on which services to create from the underlying applications and data. Service creation can be fine grained (a single service that maps on to a single business process, such as 'insert part number'), or coarse grained (multiple services come together to perform a related set of business functions, such as 'process purchase order').

The expose phase is also concerned with how the services are implemented. The functionality of underlying IT resources can be made available natively if they already speak Web services, or can be made available as Web services though use of an adapter.

Compose

Once services are created, they can be combined into more complex services, applications or cross-functional business processes. Because services exist independently of one another as well as from the underlying IT infrastructure, they can be combined and reused with maximum flexibility. And as business processes evolve, business rules and practices can be adjusted without constraint from the limitations of the underlying applications.

Consume

By creating composite applications that consume these services and processes, you deliver to the business new dynamic applications that enable increased productivity and enhanced insight into business performance. Users can consume the composed service through a number of avenues, including web portals, rich clients, Office business applications, and mobile devices.

The following example illustrates a typical implementation project for the same fictitious enterprise we discussed earlier.

Example: Implementing SOA in the Enterprise

In the order fulfillment process with BP Low Electronics starts with the opening of a purchase order, then an inventory check to make sure that items are in stock, and finally, in a collaborative process, transport is arranged to pickup and deliver the order to the customer. Each of these parts of the process consists of multiple, non-automated steps.

BP Low Electronics decides to shift from paper-based purchase orders to an electronic form, automating the process of information transfer between sales and the shipping department. Shipping does an inventory check: if the items are not in stock, the PO is put on hold and the items backordered. If the items are in stock, the shipping clerk proceeds with ordering transport for the shipment and with notifying the loading dock clerk of that the items will need to be obtained from the warehouse and made ready for shipment. BP Low Electronics decides that these two portions of the order fulfillment phase are good candidates for being automated and implemented using Web services.

Expose
BP Low Electronics decides to implement a flexible, reusable service oriented layer that exposes business logic from their existing applications (and business partners), in order to reduce the inefficiencies associated with manual service requests and status notifications.   Some examples include:
- ShippingService - which is Northern Electronics' web service used to send and receive details of the shipment pickup.
- Pickup Service - which is Northern Electronics' internal process of notifying the loading dock clerk of the need for product pickup and confirmation that pickup occurred.
- Transport Service - which is the transport provider's web service used to order transport and confirm that the shipment was picked up. This piece of the web service solution must be implemented by the transport company.

Each of these business services consist of underlying fine-grain services or API calls that are together combined to create the larger service layer.

Compose
Once their applications are enabled as services, BP Low Electronics maps their end-to-end business process into an orchestrated process model that sequences, controls, and monitors their end-to-end flow of purchase orders. A key strategic principle for them is to not hard-code their business logic and rules into implementation code, but to externalize the logic into a model that can be more easily viewed and changed.

This also permits a greater level of visibility and traceability, including the ability to aggregate real-time process metrics and events into a decision dashboard or scorecard. You can monitor the process health end to end – not just whether individual applications are complete, but the full set of activities that contribute to the business process. If steps in the process fail to be met, such as a failure for transportation to arrive during the expected pickup time, business users can be immediately notified and take appropriate action.

Consume

Because the end-users are non-technical information workers, BP Low Electronics decides to build a composite application based upon Office (often called an "Office Business Application", or OBA). This OBA enables the end-users to easily access the information and process they need, within the context of the familiar tools they are already used to using. This enables greater end-user productivity and avoids the need for extensive end-user training.

Summary

SOA solutions help organizations of all sizes integrate and access existing IT resources, assemble them into larger business processes, and make the outputs available to users in order to run their organization more effectively. This "real world" approach to SOA, beginning with a focused understanding of business problems, enables organizations to align their IT assets with changing business needs and to deliver on business goals one need at a time.

How can your organization get started with SOA?

1. Make sure that you have sound business drivers. When an organization struggles to justify their SOA projects, it is almost always because they are trying to "do SOA" rather than address a business need.

2. Top-down approaches do not work in the real world. Bottom-up approaches are not manageable either. In contrast, organizations that are successful with SOA often adopt a middle-out approach. These organizations all have something in common—they start with clear business challenges and focus on creating business value.

3. Try to avoid subscribing to the "build it and they will come" approach. Some organizations spend 18 to 30 months building a services infrastructure. When they finally reach the service consumption or user-experience layer, they find that the business needs have changed, rendering the investments a waste of time and money. It is often more practical to partition your usage scenarios into small sets and build out the entire scenario top to bottom, from the data through to the application consuming the services. Partitioning functionality in this manner can help you track changing business needs much more effectively.

4. Demonstrate value in rapid iterations. Time-to-value is a critical, healthy metric. The "trust-me" approach is not a healthy model for successfully leveraging SOA.

5. Last, but not least, organizations that have successfully adopted a SOA solution often use a "snowball" approach. How do you build a big snowball? You start with a small snowball. This is probably the most important take-away with respect to leveraging SOA to drive business value.

USA	India		
Corporate Office	**Development Center**	**Regional Offices**	
7 Lincoln Highway, Suite 205, Edison, **NJ** 08820 Phone 732.548.9268 Fax 732.548.8913	Plot No. 38, Electronics city Sector 18, **Gurgaon**-122 015, India Phone: 91-124- 2397660-62, 5017660 Fax: 91-124- 2397655, 5019955 http://www.binarysemantics.com	20, 2nd Floor, Arihanth Complex 1st Cross, CKC Garden (Off: Mission Road) **Bangalore** - 560 027 Phone: 91-80- 22240222 / 22241222 Fax: 91-80- 22277867	Basera - Plot No.48, 3rd Floor, Santhawadi Lane Opp. Jain Upasarai, Jaiprakash Road, Andheri (W) **Mumbai** - 400 058 Phone: 91-22- 26705762 /26286748/62 TeleFax: 91-22-26705763