SOA fundamentals in a nutshell

Prepare to become an IBM Certified SOA Associate

Skill Level: Introductory

Mohamed I. Mabrouk (mohamedi@eg.ibm.com)
Software Engineer
IBM

05 Sep 2008

Thinking about getting certified in Service-Oriented Architecture (SOA)? Want to catch the wave of interest in SOA? Take this tutorial to prepare for the IBM® SOA fundamentals test leading to your certification as an IBM Certified SOA Associate. Even if you're not planning for certification right now, this tutorial is a good place to start learning about what SOA is and what it can do for your organization.

Section 1. Before you start

About this tutorial

The tutorial structure is based on the objectives of IBM exam 669: SOA fundamentals, the only required exam to be an IBM Certified SOA Associate. This tutorial, though not to be used as the sole resource, is a great place to start if you're interested in getting certified or just learning more about SOA.

Objectives

This tutorial is an additional resource in your quest to become an IBM Certified SOA Associate. Following the objectives of the IBM SOA Fundamentals exam, this tutorial is composed of five main sections, each covering a major topic through a set of subsidiary questions and answers. You'll learn about:

- The value of SOA.
- The main driver for SOA's rise to prominence.
- Basic SOA concepts.
- The realization of SOA.
- SOA management.
- Preparations to adopt and implement an SOA, and what you can expect.

Prerequisites

The tutorial discusses SOA from a vendor-, implementation-, and technology-independent point of view, so you don't need any specific technical knowledge to follow along. A basic background in the concept of web services and SOA is helpful although not required. It's a good idea to review the objectives of exam 669 before you get started.

Section 2. Introduction to SOA

If you're still learning about SOA, you might want to read this introduction for some basic information before jumping into the tutorial.

SOA is an architecture approach for defining, linking, and integrating reusable business services that have clear boundaries and are self-contained with their own functionalities. Within this type of architecture, you can orchestrate the business services in business processes. Adopting the concept of services—a higher-level abstraction that's independent of application or infrastructure IT platform and of context or other services—SOA takes IT to another level, one that's more suited for interoperability and heterogeneous environments.

Because an SOA is built on standards acknowledged and supported by the major IT providers, such as web services, you can quickly build and interconnect its services. You can interconnect between enterprises regardless of their supported infrastructure, which opens doors to delegation, sharing, reuse, and maximizing the benefits of your existing assets.

With an SOA established, you bring your internal IT infrastructure to a higher, more visible, and manageable level. With reusable services and high-level processes, change is easier than ever and is more like disassembling and reassembling parts (services) into new, business-aligned processes. This not only promotes efficiency and reuse, it provides a strong ability to change and align IT with business.

Section 3. The value of SOA

So why is everyone so excited SOA? What does it provide, and how can it help? Should it be used in all cases? Let's answer these questions one at a time.

What's the best fit for SOA?

You might be wondering in which business functions and situations SOA fits best and which best shows its potential? There are some situations and business functions that should conjure SOA immediately, because SOA can boost competitiveness and productivity, and clearly display its benefits. Such situations mainly include:

- **Centralized business functions used by multiple entities:** SOA helps to identify such functions and package them into reusable, self-contained services that aren't affected by process changes around them.
- **Integration with partners:** SOA promotes using standards, which is critical in any integration because standards create a common baseline for all parties to work on. Also, the agility provided by SOA enhances the integration experience with the flexibility to plug in, change, or update services almost seamlessly to your clients with SOA's decoupling capabilities.
- **The existence of old technologies that are still working:** Some organizations aren't willing to give up their tried-and-true technologies. Security concerns make some customers, especially in sensitive industries such as banking, suspicious of new software systems and their unknown vulnerabilities. In these cases, SOA can help by wrapping legacy technologies in standardized ways, enabling their exposure in a standards-based environment suited for integration and reuse.

What factors contribute to SOA's most popular capability: business agility enablement?

Because change is inevitable, the only guarantee of the continuity of a business is its ability to anticipate and adapt to changes, also known as *business agility*. Crucial to the future of any business, SOA makes business agility possible with the following factors.

Loose coupling

- Enables real-time business capabilities because it removes the hard connections that impede the ability to change
- Changes the way IT costs are distributed, with less expenses in implementation and more investments in reuse
- Increases the feasibility of real-time remote access to original sources of information, thus reducing the delay and dependencies
- Integration projects are driven by business needs, with the visibility of capabilities provided (that is, business is the main driver)
- Lets companies extract more data measuring business performance in real time by exposing and sharing information
- Decreases time to market because connections to customers and partners can be made faster
- Makes it easier for partners to do business with your company
- Promotes and publicizes your services, making it easier for customers to find you and your services
- Makes it easier to find new partners and services by helping you search for the most suitable service for your need

Reuse

- Makes processes more consistent because they depend on the same reused components
- Promotes increased quality through competition between the services providers
- Gives consumers a wide choice of suppliers
- Covers essentially all classes of IT assets: hardware, software, data, and process assets
- Decreases the impact of change because it's done in a central location and reflects on all concerned parties
- Lets you focus on business processes rather than technical

implementation

- Helps decrease the cost of integration because the component has already been integrated
- Lets you make system changes without constraining business change
- Promotes flexibility, which gives you more space to innovate
- Lets you publish once but consume many times

Extensibility

- Makes SOA solutions available to all sizes of organizations
- Changes software-deployment activities from a big-bang model into a more dynamic, less-time-consuming model, which is more appropriate to the business
- Makes it easier to add or change partners
- Accelerates mergers and acquisitions
- Facilitates exposed services, which represent potential new revenue sources

So what will a company lose if it doesn't adopt SOA?

Given that SOA is a plausible solution for a company, the cost of not implementing it can result in three major setbacks:

- Inability to move to higher-value markets that provide more business growth and exposure. Because a company is bound to its existing tailored systems, it becomes stuck in its original place in the market and struggles to address the higher-value markets. However, with SOA, an organization can change business tactics and enable new ones, giving it an edge.
- Inability to address more technologically advanced competition.
- Competition from lower-cost sources.

Is SOA always a better solution?

SOA provides benefits in almost all cases of business organizations. However, in very special cases, it might prove to be a liability more than a drive towards better business. These cases include:

- **A homogeneous IT environment:** If an organization depends on a set of coherent products—belonging to a same vendor, for example—, has a limited scope of work, and has no need to add or change any of these products, an SOA might be a liability more than a useful strategy.
- **When true real-time performance is critical:** To provide loose coupling between different consumers and producers, an SOA depends on interoperable protocols, which are slow by nature. It can also induce mediation logic and asynchronous protocols, which aren't suitable for real-time performance.
- **When things don't change:** If the customer sees no change happening to the business logic, presentation, data flow, process, or any other aspect of the application, converting old systems to SOA might not return sufficient value to make the effort worthwhile.
- **When tight coupling is not an inconvenience:** Loose coupling is of best use when it's used with a component that's not under your control and, this, you can't control its change. On the other hand, when the component is yours and under your control, loose coupling can be a burden, especially if the component isn't really reusable.

Section 4. SOA concepts

Now let's take a look at some SOA concepts to better understand what SOA is.

Definition of a service in SOA

There are a lot of different definitions of services, but I think these do the best job of explaining what a service really is.

From *Web Services and Service-Oriented Architecture: The Savvy Manager's guide* (see Resources for a link):

> "A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services."

From SearchSystemChannel.com (see Resources for a link):

> "...a service is defined as a unit of work to be performed on behalf of some computing entity, such as a human user or another program."

The concept of loose coupling in SOA

To understand the concept of loose coupling in SOA, you should first examine the concept of loose coupling in general. The following items demonstrate what loose coupling is and why it's valuable:

- An entity is coupled if changes to the entity by one party in the interaction require corresponding changes by the other parties (for example, business data models).
- An entity is declared if its behavior is specified in the interface to the service, and service requesters and providers can only interact if they have matching declared behavior. Declared aspects include security, transactional behavior, and quality of service (such as response time and delivery).
- An entity is transformed if it's declared by both service requesters and service providers, but the infrastructure provides some transformation capability to enable interactions between service requesters and providers that declare mismatched behavior.
- An entity is negotiated if both requester and provider declare a spectrum of behaviors they are able to support, and the intermediary infrastructure is capable of negotiating an agreed-upon behavior between them for each interaction.
- An entity is decoupled if changes to the aspect by one party in the interaction don't require corresponding changes by the other parties.

Loose coupling manifests itself in the SOA paradigm as follows:

- It helps to have an abstraction layer between the service producers and service consumers.
- Loose coupling promotes flexibility in changing the service implementation without impacting the service consumers.
- In the SOA approach, functionality is organized as a set of modular, reusable shared services. These services have well-defined interfaces that encapsulate the key rules for accessing the services. They're also built without making any assumptions of who will use or consume these services. Thus, they are loosely coupled to the consumer of these services.

How does XML contribute in an SOA?

Based on open standards and promoting platform-independent business integration, SOA needs a common platform to base its infrastructure on. This infrastructure needs to be supported by all involved parties to form a common base of understanding. XML is at the core of this infrastructure for the following reasons:

- XML is the foundation for virtually all web services standards, such as XML schema, SOAP, Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI). These standards leverage the core concept of XML-based representations, a worldwide supported format that carries out information interchange between service providers and requesters in an SOA.
- Using XML resolves the challenge of working with different data formats in different applications across multiple platforms.
- XML has the benefit of ease of representation, being text-based, flexible, and extensible by nature.

Examples of standards built on XML that SOA leverages include:

- **SOAP:** This simple XML-based protocol lets applications exchange information over transportation protocols like HTTP. Using XML in SOAP guarantees that the SOAP protocol is:
  - Platform independent.
  - Internet usable.
  - Humanly readable, structured, and text based.

  With the benefits above, SOAP is the recommended and most widely used communication protocol for web services. Knowing that web services are the cornerstone for SOA, it's therefore also the basic communication protocol for SOA solutions.

- **WSDL:** WSDL is a document written in XML to describe a web service. It specifies the location of the service and the operations (or methods) the service exposes to let individuals access those services. A WSDL file describes four main things:
  - Services available by the web service interface, such as listing names of methods and attribute messages
  - Data types of messages
  - Binding information for the transport protocol, such as HTTP and JMS
  - Service address to be used when calling it
- **Electronic Business using eXtensible Markup Language (ebXML):**

ebXML is a standard way to define the business transactions that can be performed between different businesses. ebXML defines standard methods for business messages exchange, establishing trading communications and registering business processes between companies

Service registries

A service registry is a directory of services available in an SOA system. It contains the physical location of services, versions and validity periods of services, service documentation, and policies. A service registry is one of the main building blocks of an SOA architecture. Its role is described below:

- The service registry realizes the SOA promise of loose coupling. By holding the service endpoint locations, it removes the high coupling resulting from hard-wiring the consumer to the provider. It also eases the potential difficulties in replacing one service implementation with another if needed.
- A service registry is highly scalable; it evolves seamlessly should the system it serves grow.
- It enables systems analysts to survey an enterprise's business services portfolio. They can then determine which services are available to automate processes to address pressing business needs and which aren't, letting you know what needs to be implemented and added to the portfolio, providing a catalog of the available services.
- A service registry can step into the role of governing services by enforcing compliance for subscribing services. This helps ensure the integrity of service governance and policies. You'll learn more about governance and its importance in SOA later in this tutorial.
- Visibility of the available services and their interfaces allows speedier development, greater application reuse, improved governance, and better business planning and management. The lack of a service registry leads to redundancy and inefficiency.
- Service registries help reduce time wasted in locating service information.
- Without a registry to track services and their relationships, an SOA environment not only lacks coherence and control, it invites chaos.

What's a business process?

Business process is a term you hear used frequently in this environment. Here are two definitions of a business process:

From "Business processes and workflow in the web services world" (developerWorks, Jan 2003):

> "A business process can be defined as a set of interrelated tasks linked to an activity that spans functional boundaries. Business processes have starting points and ending points, and they are repeatable."

Another definition is: *A business process can be seen as a set of activities performed by a business entity in response to an event. This set of activities is harmonized, described and integrated within the business process.*

Issuing an identification card for a person is an example of a business process. You present your certificate of birth, your educational and professional papers, and a photograph to initiate the process. Then an internal file is created, a security investigation is conducted on you, and finally, after all the processing is done, you get an ID card.

In the SOA paradigm, the business process controls the flow of services. The business process drives the flow of events, calls and coordinates services, and creates a context for them to intercommunicate. Business processes represent the business abstraction; decoupled from the implementation of services, a process cares about the flow of business. This separation of concerns not only allows more focus on process creation, it makes it easier to edit processes according to need without having to edit the underlying service implementations.

Elements of a business process

It might be better to define a business process in terms of its composing elements; this provides some technical insight into a business process.

- **Input:** The information needed by the activities of the process to produce a result. In the example of the ID card, the inputs would be your credentials, birth certificate, and photograph.
- **Output:** All the data and information generated by the process. The output represents business goals and measurements needed for the business. In the ID card example, this would be an internal file for you and a physical ID card as well as measurements on how the process proceeded.
- **Events:** Notifications of some occurrence of importance. An indication, for example. They can occur before, during, and after the execution of a process. In the ID example, this might be the input of a new document that wasn't present at first and that needs to be included.
- **Subprocess:** Smaller process, or process steps, inside a process. A

subprocess is used when it's not possible to represent the scope of work with only a set of activities. It has the same elements as the process. In the ID example, this might be the subprocess of investigating your criminal record and getting the results.

- **Activity:** The lowest level of work in a process. In the ID example, this can be the creation of a new internal file for you, the person getting the ID card.
- **Performance metrics:** Attributes that represent the effectiveness of a process to determine if it meets the required performance. These metrics help determine the performance and compare it to the required figures. They also point out potential areas of improvement in the process, ultimately, and ideally, realizing the cycle of improvement that the SOA promises. In the ID example, measurements would calculate which part of the process consumed most of the time or had the highest processing hit. This helps later on in improving the process.

How does SOA address transaction control?

Because a process spans multiple activities, business transactions occurring within an SOA environment can be very complex. This is due to the nature of the services in long-running processes within the SOA context, which are often asynchronous, stateless, distributed, and opaque.

Web services are a perfect representation of services in an SOA environment. Being self-contained as needed by SOA, they are limited when it comes to the need of a cross-service transaction. As long as a service is at the root of a transaction and the scope of the transaction is limited to activities that are performed by the service's underlying solution logic, there's no need for cross-service transaction functionality, and the transaction can be managed by whichever proprietary technology (component-based, legacy, or otherwise) it encapsulates. But as the number of services in an environment grows, the need to span transactions across those services increases.

Some web services specifications were developed to address the problem of transactions. These include:

- **WS-Coordination:** Enables registered processes to participate in an activity to create a shared context that's responsible for holding the stateful data and information propagated between them as well as the transaction state. The framework enables existing transaction processing, workflow, and other systems for coordination to hide their proprietary protocols and to operate in a heterogeneous environment. This protocol provides the infrastructure for other protocols, such as

WS-AtomicTransaction or WS-BusinessActivity, which make use of its framework.

- **WS-AtomicTransaction:** Is used with short-lived distributed activities. It provides three types of protocols that can be used with the WS-Coordination framework for two phase commit ACID-type transactions (transactions supporting atomicity, consistency, isolation, and durability) to choose from:
    - Completion
    - Volatile two-phase commit
    - Durable two-phase commit
- **WS-BusinessActivity:** This protocol is used with long-running transactions with compensation processes. As with the WS-AtomicTransaction protocol, it uses the WS-Coordination framework to provide two protocols for business activity coordination:
    - BusinessAgreementWithParticipantCompletion
    - BusinessAgreementWithCoordinatorCompletion

What's the role of standards in SOA?

In general, SOA projects are highly reliant upon standards, and leverage them because of these critical benefits:

- Standards ensure interoperability across system and partners.
- Using standards speeds up development and delivery through processes and tools.
- Standards enable better management and visibility of IT assets.
- Standards ensure quality of service (QoS).
- Standards help with flexibility by reducing dependencies on a specific implementation.

Next, explore a few examples of standards leveraged by SOA and see how they help realize its promises.

WS-Security

The WS-Security protocol is based on adding SOAP extensions to the message header to store security metadata that's intended to provide protection through message integrity, confidentiality, and authentication. Those extensions provide a general-purpose mechanism to associate security tokens to the message rather than a fixed security mechanism. The generic platform supports different security mechanisms. The protocol is designed to be extensible.

BPEL4WS

Business Process Execution Language for Web Services (BPEL4WS) is defined in the OASIS online community for BPEL:

> "This protocol defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. It also defines how multiple service interactions with partners are coordinated to achieve a business goal, as well as the state and the logic necessary for this coordination."

As they are clearly needed, BPEL4WS introduces methods to deal with business exceptions and faults, as well as ways to compensate other committed processes that may need to be reversed in case of errors. Because BPEL needs to be supported universally, it's based on the universally acknowledged WSDL protocol, which, itself, is layered on XML.

WS-I

As declared on the WS-I Web site (see Resources for a link):

> "The Web Services Interoperability Organization (WS-I) is an open industry organization chartered to establish Best Practices for web services interoperability, for selected groups of web services standards, across platforms, operating systems and programming languages."

This group is concerned with the development of web services standards among different implementations, platforms, and their actual interoperability. Its main goal is to guide and advise organizations on how to ensure interoperability while interconnecting systems using web services.

WS-I has four main deliverables:

- Profiles that are versioned specifications describing implementation guidelines and best practices for web services that are interoperable and work together as a set
- Use cases and usage scenarios to demonstrate the guidelines in the profiles
- Sample applications

- Testing tools for profile conformance

Section 5. Basic SOA architecture

Now let's take a look at some more complex, technical subjects, such as the role of the enterprise service bus (ESB), business processes, their choreography, and the role of web services.

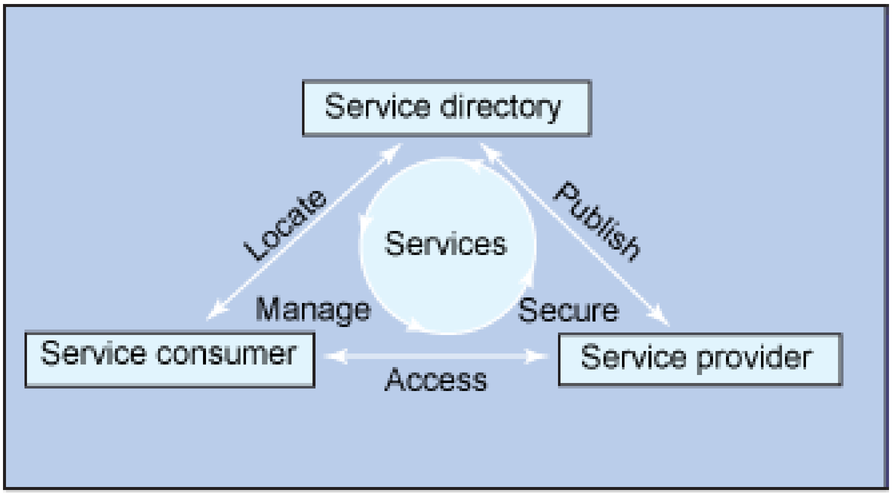
What constitutes a basic SOA architecture?

A basic SOA architecture is composed of a service provider, service, and an optional service directory. Application-to-application messaging is used in the information exchange.

The similarity between this model and that of straight web services is very visible, with WSDL being the invocation contract stored in a service directory where it can be queried and fetched via UDDI. Web services are actually a realization of SOA at its most basic level.

In this model, the basic scenario is as follows: First the service provider creates a service and decides to expose it and publish it. Publishing is done by posting the service information on the service directory. On the other side, a service requester, in need of a certain service, searches the service directory for one that meets the necessary criteria. Upon finding one and using the information available on the service directory, the service requester is able to directly contact the service provider in the correct way to fulfill the business need.

Figure 1. Basic SOA architecture



Here are some definitions of terms used in this section:

- **Service provider:** Provider of services whose invocation contract and location are published
- **Service consumer:** Consumer of services matching his or her business need found in a service directory
- **Service directory:** Directory for publishing and listing available services for consumers

What's the role played by an ESB in an SOA?

An ESB plays an important role in an SOA. At the base of its roles, it represents the backbone and infrastructure capable of connecting service providers and service consumers.

Below are the detailed roles of the ESB:

- Provides an integration infrastructure consistent with the principles of SOA:
  - Enforces the use of explicit implementation-independent interfaces to define services with loose coupling
  - Uses communication protocols that stress location transparency and interoperability
  - Promotes the definition of services that encapsulate reusable business functionalities
- Provides the means to manage the service infrastructure

- Operates in the distributed, heterogeneous environment because it:
    - Supports synchronous and asynchronous communication
    - Uses standard interfaces and standard protocols
- Centralizes control and distributes processing
- Supports mediation to formulate the request/response as needed between different parties without the need of change in any
- Applies security and QoS to the SOA project

What's the role of web services in SOA?

Although web services came before SOA, they represent the answer and realization of the SOA question seeking the need for interoperability between systems and platforms. This helped get SOA up and running quickly because it already had a supporting technology to satisfy its needs. It's clear now that web services represent the cornerstone of the SOA and its recommended technology for interoperability.

Web services are the cornerstone of SOA because they:

- Enforce standards and, thus, promote compatibility and portability.
- Are cross-platform and cross-language.
- Are widely supported, making SOA relatively easy to adopt.
- Are message-oriented.
- Provide faster tooling support, which speeds the implementation of SOA.

What is choreography? How does it fit in the SOA big picture?

Business service choreography is concerned with the development and execution of business flow logic, independent from underlying services and business logic. This means that the process choreography cares about the sequence of events and how the events are related, but it doesn't care about the events themselves. This separation of concerns between process and services provides flexibility to easily change the processes without changing the core services. This follows the loose-coupling aim of SOA.

To describe business processes, an emerging standard, BPEL4WS, was created. BPEL4WS is layered over web services standards. The compatibility of such standards enables processes to call underlying services and partner services in an open standards-based infrastructure.

A process that's defined in the BPEL4WS is composed of:

- The activities, which are the individual business steps within the process. The activities can be basic or formed of other activities (structured).
- Partner links, which specify external entities that interact with the process, or vice versa, using WSDL interfaces.
- Variables that store messages passed between activities, thus, representing state.
- Correlation sets, used to correlate multiple service requests or response messages with the same business process instance.
- Fault handlers to deal with exceptional situations that can occur when a business process runs.
- Event handlers, which receive and process messages in parallel to the normal execution process.
- Compensation handlers, which specify the compensation logic to undo an activity or more when an exception happens.

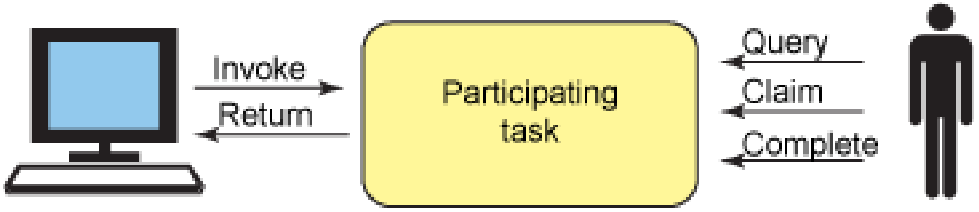
Human tasks

Business choreography also provides support for *human tasks*, which are components involving human intervention either with a service or with another person. An example is managerial approval on travel requests or handling a customer request by an individual.

The types of human tasks are:

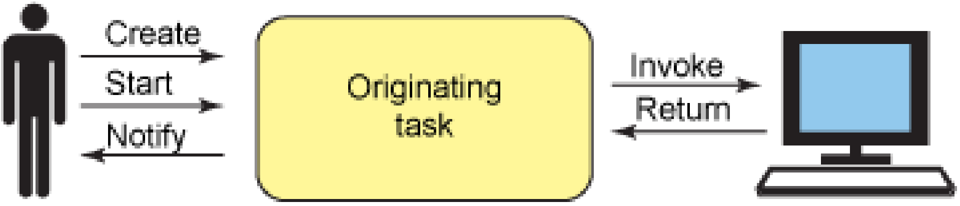
- **Participating tasks:** These are initiated by the system (process), which requires a human response to continue execution. The system initiates the task and an individual from the candidate executers claims the task and executes it. Then she provides the output back to the system, notifying it of its completion. An example for this is a travel reimbursement process awaiting manager approval.

  Figure 2. Participating task members and interactions

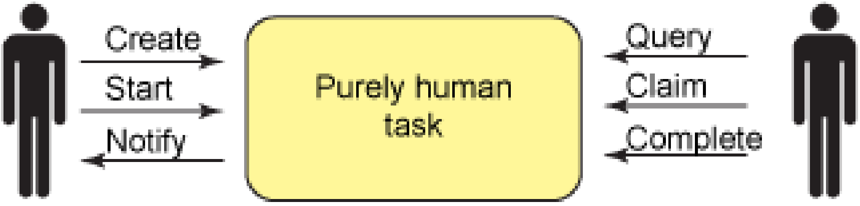
- **Originating tasks**: As their name signifies, these are initiated by a person through a user interface. They target a system; a person creates an originating task and starts it; and a request is sent to the system to run the services that are needed. As soon as the system finishes executing, a notification is sent to the initiator. An example of such a task is the initiation of a travel reimbursement process by an employee.

Figure 3. Originating task members and interactions



- **Purely human tasks**: These are, like originating human tasks, created and started by a person. And, like participating human tasks, they target another person who then claims and completes the task. Purely human tasks don't interact with business processes or other web services. They're not automated tasks, yet they pass through the same cycle of assignment and notifications.

Figure 4. Purely human task members and interactions



It's logical that human tasks can take much more time than automated tasks, which raises another question: Can processes afford the interruption and wait time caused by human tasks? The answer is yes. And to get into more detail, let's tackle the subject of business process types.

Business process types

Business processes can be either long-running or micro-flow:

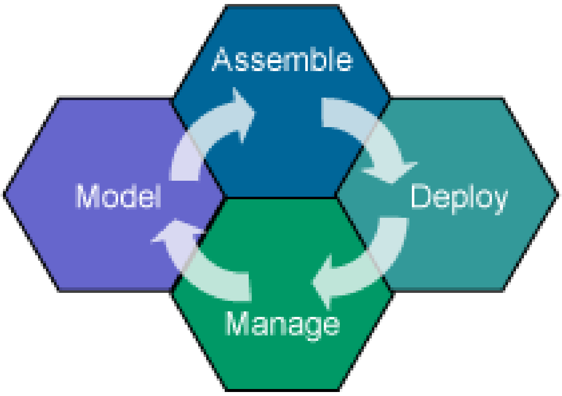
- **Long running processes** are interruptible and can also run in several transactions. They can wait for external stimuli, like those resulting from the use of human tasks. A rule of thumb is that if a process contains a human task, then the process must be long running. Long-running processes can also contain both synchronous and asynchronous services. Long-running processes store each intermediate process state to be forward-recoverable.
- **Micro-flows** run in a single thread without interruption. They are also called noninterruptible business processes. Micro-flows run in only one transaction, are short in duration, and consist of synchronous services only.

The SOA life cycle and its different stages

SOA is characterized by a dynamic life cycle. Inherent in it is the possibility of continuous improvement of the processes, which, associated with the loose coupling enforced with the SOA, allows processes to improve as easily as disassembling and reassembling the building blocks (services in this case) without rework. This improves time to market and alignment of business and IT.

A famous diagram of the SOA life cycle includes the four interconnected hexagonal figures representing the four stages of SOA. As visible in the diagram in Figure 5, the four stages form a closed loop, representing the continuous cycle of monitoring and improvement.

Figure 5. The four stages of the SOA life cycle



Let's break these down:

Model stage

The model phase includes business analysis and requirements gathering, which are then followed by modeling and optimizing the business process. The model helps lay a common understanding of the process, its objectives, and outcomes. It also makes sure that the design meets the business requirement and provides a baseline to measure the performance later on.

Assemble stage

During this phase, existing assets—such as enterprise resource planning (ERP), financial systems, IBM CICS® applications, and so on)—that are needed in the modeled processes are wrapped as services, while nonexisting needed functionalities are implemented and tested. After all services are available, they can be orchestrated to implement the business process.

Deploy stage

During the deployment phase, the runtime environment can be configured to meet the required quality-of-service levels and security requirements. The environment can be scaled and optimized to be capable of reliably running the mission-critical processes, and at the same time providing flexibility to make updates dynamically in case of change.

Manage stage

During this phase, several aspects are managed and monitored, such as the services assets, services availability and response times, and version control over services. An important role in this phase is monitoring the key performance indicators (KPIs) of the processes. This helps to prevent or isolate and diagnose emerging problems in real time as well as provide feedback on the business process performance and bottlenecks to help improve it. This feedback is sent to the model phase, the first step, helping improve the process.

Section 6. SOA management

As covered in the first section, an SOA needs a robust active management framework or else it gets out of hand. SOA management is realized through the governance concept, which controls the different aspects of SOA. Security is another aspect that has to be enforced in an SOA-enabled environment because of its open nature. Details of SOA management are discussed in this section.

SOA governance

Without a controlling entity, an SOA is not only challenging to manage, but it invites chaos because of its open and distributed nature. Because of this, it needs a management and controlling entity: governance.

Definition of governance

SOA governance is a framework for decision and role identification to encourage IT actions that are synchronized with the enterprise strategy and prevent those that aren't. This framework is managed by a group or committee responsible for creating policies to enforce governance and role identification, empowerment, and accountability of individuals who are given the capability of decision making and policy enforcement. In brief, the committee needs to address three main questions:

- What decisions need to be made to ensure effective management of IT assets?

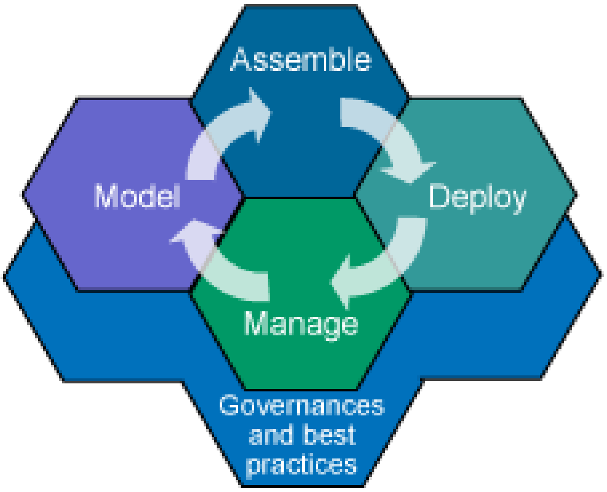
- Who should be responsible for making these decisions?
- How can such decisions be enforced and monitored?

As part of the governance realization, service level agreements (SLAs) are identified and monitored for verification. Performance metrics are also collected to represent the effectiveness of the governance.

What role does governance play in an SOA environment?

The role of governance in SOA is crucial; it needs to be enabled on all phases of the SOA life cycle, as shown in Figure 6.

Figure 6. Governance location with respect to the SOA life cycle stages



The need of SOA governance is obvious because:

- Governance involves applying the principles of an enterprise strategy to direct and control IT.
- Governance aims to encourage behaviors consistent with the organization's mission, strategy, and values toward achieving the enterprise's business goals, adding value while balancing risk and return.
- Governance assures keeping services at a defined level in terms of integrity, performance, reliability, and currency.
- Governance makes sure that business application needs are being correctly assessed and prioritized to drive creation and consumption of services, thus ensuring the best usage in alignment with business goals.
- Governance ensures that IT investments are being used in a profitable

manner.

- Governance ensures that an enterprise-wide SOA-enabling architecture is the main guide for design of any service.
- Governance, as a controlling entity, leverages the best practice of IT principles.
- To protect the business assets, governance also enforces security of enterprise data and privacy of information shared across boundaries.
- Governance, managing the IT of the enterprise, enforces integrity and reliability of data and processes to leverage reuse and maximize profit.
- Governance ensures a certain level of performance and quality of service on all components in the consumer-provider chain of services.
- Standards are at the base of SOA, so governance helps to enforce high levels of interoperability, which leverages the enterprise with all the benefits of open standards.
- Governance uses metrics to audit and monitor the progress of the development of the IT infrastructure and its conformance with established policies.

Quality of service compliance in SOA governance.

In a framework with SOA governance, QoS policies are defined and enforced on the organization. This is essential in an open environment where integration and services exchange is not limited to the internal functions of an enterprise, but to other enterprises of different sizes, different scopes, and different IT sizes to maintain and guarantee a steady level of the overall processes. For example, if you consider response time to be a QoS, if QoS is not enforced on services to respond in a given time, the slowest service can create a bottleneck and waste the QoS provided by other faster services. The same goes for security: One noncompliant service may jeopardize the whole system. In some systems, the infrastructure is made to detect QoS levels and reject noncomplying services.

Why are security systems in SOA environments complex and distributed?

Such complex security systems are needed because:

- Distributed systems require distributed security.
- There's a need to manage user registries and access control across

multiple applications, platforms, business partners, and entities, which can't be managed at a single point.

- You have to consistently enforce security policies across the environment.
- The security system needs to be able to evolve as the enterprise and its applications evolve.

In the SOA life cycle, what's the impact of change in services?

With the decoupling principle applied, changes in services in the SOA environment are handled simply, because service consumers are decoupled from service contributors by the ESB, which sits in the middle and can mediate the messages. Changes on the provider side can be consumed by the ESB so that the consumer remains the same and stays seamless to the change.

On the other hand, I have to point out the importance of unmanaged change in an SOA environment. With the principle of reuse, each service may be an enterprise-level service, not just a local one within its department or unit. Any unmanaged change in such a service can lead to unpredictable enterprise-wide failures and halting processes. This shows the importance of governance in ensuring that a policy is managing the change. This policy should measure the impact, allow the change, and ensure a system of notification for the parties impacted (ESB or direct consumers). Changes in distributed systems require stern rules to manage them.

What's the role of the ESB in SOA governance?

The ESB plays an important role in enforcing governance. Security and QoS policies can be applied to the ESB to control their levels and allow only conforming requests. In general, an ESB plays the role of a unifying platform on which required policies are mandated. The nature of an ESB as a central place where all communication occurs makes it a perfect place to activate such rules. And rest assured that everyone either complies or is isolated.

Section 7. Prepare to implement an SOA

The process of introducing SOA in an organization requires special skills, including:

- The ability to measure the readiness of the organization to such adoption.

- Identifying boundaries and entry points.
- Enlightening people with the benefits that SOA can bring to the business and IT.
- Measuring the challenges and drivers to SOA induction on both the business side and the technical side.

What benefits does SOA provide to business and IT strategy?

SOA's benefits to the business include:

- Increasing the responsiveness of the business to market changes and improving agility in the organization.
- Bypassing organizational boundaries and synergizing with the existing assets.
- Helping reduce development time.
- Exposing inefficiencies in business processes.
- Ensuring the alignment of IT resources to business strategy and goals.
- Decreasing the cost of compliance and security with standards enforcement.
- Making it easier for partners and customers to find you and making it easier for you to find them.
- Granting more consistent processes.
- Providing a different choice of suppliers because of the standards enforcement.
- Enabling asset reuse.
- Reducing the cost of integration.
- Easing upgrades and mergers.

SOA's benefits to the IT strategy include:

- Architecting systems to effectively use standards and services to gain the benefits they promise the business.
- Allowing various communication mechanisms to be used.
- Allowing flexible and reliable security systems to be incorporated to ensure security.

- Providing a service bus where the flow of messages and messages themselves can be managed providing another dimension to flexibility and adaptability of the system.
- Easing integration with modular, componentized services and a connecting services bus.
- Being built on standards and protocols that are widely supported to enable interoperability, a goal of SOA since its start.
- Promoting reuse with a services repository and mediation modules.
- Boosting connectivity using the ESB, which takes connectivity to its highest peak. The ESB is responsible for mediation of protocols, data, and formats to ensure compatibility.

What business issues and drivers can organizations expect when preparing for SOA adoption?

The business domain cares about the form and impact that this new paradigm will have on the organization, so there will likely be some business issues that need to be identified and confronted.

Business issues

Business issues can include:

- Management doubting or questioning SOA because it's a new idea that's more IT-driven than business-driven.
- Defining the strategy and level of adoption, taking into account the current situation of the organization and how ready it is to adopt SOA.
- Mapping process to services.
- Lack of knowledge about SOA and what it can provide.
- The misconception that SOA is an IT architecture method only, which can lead to neglecting the critical role of governance.
- Underestimating IT business value.

Most of these issues can be resolved, or at least highlighted, by conducting educational sessions to show the benefits and real value of SOA.

Business drivers

The main business driver is SOA's potential to:

- Drive a business' return on investment (ROI), with reduced implementation costs through adopting standards, reuse, exposing services, and integrating with partners.
- Decrease time to market by reusing assets and incorporating partner-provided services.
- Increase the visibility of IT assets and their alignment to the business goals.
- Improve flexibility both internally in communication and externally in dealing with partners.
- Provide more efficient processes by reusing IT assets and leveraging standards.
- Promote business agility and the ability to adapt easily and quickly to business and market changes.
- Reduce costs throughout the organization.

What IT issues and drivers can organizations expect when preparing for SOA adoption?

Don't forget the IT department. Some of the issues and drivers that are important to them are listed next.

IT issues

IT issues can include:

- Changing the existing tailored systems into standards-based services.
- Management, governance, and control of services.
- Security challenges of distributed systems.
- Reliability of new systems versus the existing, dependable systems.
- Optimizing and unifying the existing asset to remove redundancy.

IT drivers

IT drivers might be:

- Adopting standards. The drive for standards is also considered an issue, but despite the effort needed to adopt standards, the benefit is clear to every IT specialist.

- Ensuring high QoS.
- Reuse of existing IT assets.
- Loose coupling of services.
- Independence from a certain provider or partner.

What factors affect the adoption of SOA in an organization?

While preparing for SOA adoption, you will have to identify factors that might affect SOA adoption and measure their impact to identify the organization's readiness. The factors revolve around people and technology, for example:

- The organization's experience with SOA.
- The level of awareness of SOA and its benefits.
- The existing methodology of identifying services and reusable components.
- Readiness of the existing business to be exposed as services.
- The current ability to access heterogeneous systems.
- The reusability level of legacy systems.
- The existence of a governance model in the organizational structure.
- The availability of shareable service layer.
- The existing architecture's ability to support advanced interactions between applications.
- The infrastructure's ability to support SOA with security, connectivity, and so on.
- The existence of a methodology to measure business processes and their efficiency levels.

Identify barriers to SOA adoption

Organizations need to identify and tackle any barriers blocking the advancement towards SOA. Such barriers can include:

- Old-fashioned IT practitioners insisting on old-fashioned waterfall development cycles.

- The notion that complex systems are better, and fear of the unknown.
- Overlooking the importance of architects and considering them theorists that cost more than the solution needs. It's important to note that architects are instrumental in SOA, and their absence will surely result in undesirable results.
- Organizational resistance to adopt an SOA model. SOA requires cooperation from all groups in the organization, not just the mere implementation of the IT framework.

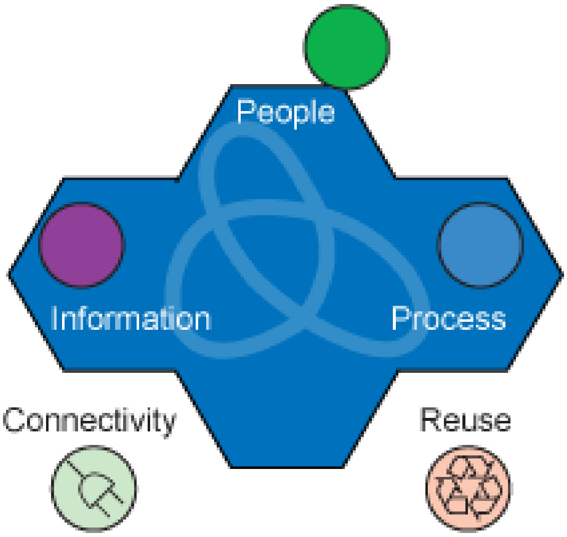
What are the entry points for SOA in an organization?

To start adopting SOA in an organization, five entry points have been identified:

- People
- Process
- Information
- Connectivity
- Reuse

The organization should choose the entry point that's most ready to adopt SOA and focus on it, while not ignoring the other entry points.

Figure 7. Entry points to SOA



Here are more details about the entry points.

People

Empowering people through SOA solutions can help boost efficiency and innovation, and provide a foundation for greater productivity and collaboration. Because people drive the interaction with the SOA services that execute business results, focusing on people is critical to the success of SOA implementations. The people entry strategy to SOA can help:

- Accelerate productivity.
- Reduce costs of access to multiple applications and information sources.
- Reduce time to deployment for new services.
- Increase access to process flexibility and orchestration.
- Enable collaboration inside and outside the enterprise.

Process

By entering SOA from a process entry point—a business-centric starting point for SOA—an organization can streamline processes across the enterprise, including improving the efficiency, flexibility, and control of key business processes. This helps align business and IT goals, and reduces the complexity of building process. Focusing on the processes entry points helps:

- Improve employee productivity.
- Increase collaboration.
- Accelerate time to market.
- Respond quickly to business challenges.
- Implement new processes in less time.
- Maximize ROI.

Information

By entering SOA from an information entry point, an organization can improve the availability and consistency of information, while removing barriers to information sharing, thus offering information access to heterogeneous data sources inside and outside the organization's boundaries. It can also help people better understand the organization's operational, transactional, analytical, and unstructured information, and make it available in new ways through SOA. This entry point can help:

- Collect and clean date, and make data widely accessible, enabling transparency and business insight.
- Reduce the cost of migration and rationalization of data by decoupling

information from applications.

- Increase an organization's agility by providing reusable information services spanning the whole organization that can be used by applications and processes, and at the same time reduce costs associated with accessing and transforming data.

Connectivity

This IT-centric entry point to SOA is designed to simplify the IT environment with a more secure, reliable, and scalable way to connect within and beyond a business, linking people, processes, and information in the business. Empowering connectivity through SOA helps:

- Ensure seamless flow of information with different protocols inside and outside the organization.
- Execute enterprise-level business processes that span the organization and business partners efficiently.
- Build trusted relationships with partners.
- Scale the business to grow smoothly.
- Deliver a consistent user experience regardless of channel or device.

Reuse

Reuse is another IT-centric entry point to SOA. It focuses on deriving continued value from existing assets and identifying services to be outsourced instead of implemented. By entering SOA from this entry point, the organization can reuse, extend, enhance, or create new processes. This enables it to increase business flexibility and responsiveness through reduced development time and elimination of duplicate processes. Using this entry point can help:

- Reduce the amount of new code that must be created for business initiatives.
- Improve efficiency.
- Reduce risk by reusing dependable resources.
- Lower maintenance costs by eliminating redundant systems.
- Wrap services performed by legacy applications into standards-based services that can participate in the broader image while delivering the same dependable output.

Section 8. Conclusion

This tutorial examined the fundamentals of SOA and covered the following topics:

- The value of SOA, how it can benefit an organization, and when it should and shouldn't be used
- SOA concepts, including services, processes, and the role of standards and service registry
- Basic SOA architecture, including more technical concepts, such as the role of web services, ESB, and business process choreography
- SOA management, why it's important, the QoS contract, and security
- Preparing for SOA, including the SOA benefits to business and IT, possible issues and drivers in both and how to handle them, readiness of organizations and how to measure it, and the entry points for SOA

Acknowledgments

I would like to express my gratitude to all those who helped during the different stages of this project. I am most indebted to Ahmed El-Maadawy for his continuous guidance and support. Special thanks to Ahmed Abbas, Ahmed El-Maadawy, Hala Aziz, and Salma El-Sheribini, who took time out of their busy schedules to review the tutorial and provide comments. I would also like to acknowledge the support of Ahmed Abbas during the writing process and the encouragement of Ahmed Fouad in the early stages of the project.

Resources

Learn

- Take the IBM course SW717: Introduction of the Value and Governance Model of Service-Oriented Architecture.
- Take the IBM course SW719: Technologies and Standards for SOA Project Implementation.
- Check out the IBM SOA entry points:
  - IBM reuse SOA entry point
  - IBM people SOA entry point
  - IBM information SOA entry point
  - IBM connectivity SOA entry point
  - IBM process SOA entry point
- Read "SOA Governance Solution" from Sun Microsystems.
- Read "SOA Connectivity Delivering the ESB without limits" [PDF], an IBM paper about ESB and its value.
- Learn when not to use SOA in Jason Bloomberg's article on ZapThink.
- Take a WSDL tutorial.
- Read an excerpt from O'Reilly's *Web Services Essentials*.
- Get information about ebXML.
- Learn more about transaction support in SOA platforms.
- Check out the different web services transactions specifications (developerWorks, Nov 2004) and the different web services security specifications (developerWorks, Apr 2002).
- Read about business process activities as web services.
- Read "Patterns: Using Business Service Choreography In Conjunction With An Enterprise Service Bus" [PDF] by Chris Nott, one of many helpful IBM Redbooks®.
- Learn more about IBM WebSphere® Process Server for z/OS:
  - IBM WebSphere Process Server for z/OS, Business Process Choreographer [PDF]
  - WebSphere Process Server help on business process types

  - WebSphere Process Server product overview

- Read "A case for SOA governance" (developerWorks, Aug 2005).
- Visit the IBM academic initiative page about SOA skills.
- Read the book *Web Services and Service-Oriented Architectures: The Savvy Manager's Guide* by Douglas K. Barry.
- Check out the top five tech buzzwords according to SearchSystemsChannel.com.
- Learn about XML compression and its role in SOA performance.
- Read "Business processes and workflow in the web services world" (developerWorks, Jan 2003).
- Visit OASIS' online community for BPEL.
- Visit the WS-I home page.
- The SOA and web services zone on IBM developerWorks hosts hundreds of informative articles and introductory, intermediate, and advanced tutorials on how to develop web services applications.
- Play in the IBM SOA Sandbox! Increase your SOA skills through practical, hands-on experience with the IBM SOA entry points.
- The IBM SOA Web site offers an overview of SOA and how IBM can help you get there.
- Stay current with developerWorks technical events and webcasts.
- Browse for books on these and other technical topics at the Safari bookstore.
- Check out a quick web services on demand demo.

Get products and technologies

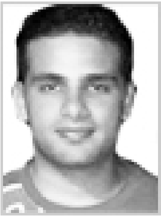
- Innovate your next development project with IBM trial software, available for download or on DVD.

Discuss

- Participate in the discussion forum for this content.
- Get involved in the developerWorks community by participating in developerWorks blogs.

About the author

Mohamed I. Mabrouk



Mohamed I. Mabrouk is a software engineer at IBM Egypt Global Delivery Center. Working at IBM Cairo Technology Development Center then moving to IBM Egypt Global Delivery Center, he gained experience in services projects in the areas of J2EE, Microsoft® .NET, business integration, and SOA, which grabbed his interest with its capability of bridging different technologies.

Trademarks