

SOA, BPM, EA, and Service Oriented Enterprise Architecture

Adrian Grigoriu

Service Oriented Architecture (SOA) appears to be at the top of its hype curve. Definitely it gets a lot of attention. It is promoted as a cure for the evils of Today's [Enterprise](#) since it reduces costs through service reuse and, by promoting modularity at service level, it enables the flexibility and agility required to respond to the greater than ever pace of change.

But what is SOA? Is it a [technology](#), an architecture, a program, or a product? Is it a business or an IT development? What is the rapport to BPM? And what is the relationship to Enterprise Architecture (EA)?

After all, both SOA and EA are about the Enterprise and its architecture, with the EA supposed to remedy similar malfunctions of the Enterprise. It is just that SOA appears to attract an even more IT oriented audience than EA. The question is, should we implement SOA and EA, or SOA, or EA, if any?

What is SOA

For most business people, SOA looks like yet another over hyped information technology. SOA, which may have its roots in a long history of distributed components architecture, is usually associated to Web Services technologies and is typically promoted by IT.

Like the OO, SOA is about providing encapsulated data and behavior, accessible solely through published interfaces. But, as opposed to OO, which addresses the SW development community, SOA aims at services business can understand and orchestrate to implement end-to-end Enterprise processes.

In a W3C definition, SOA is "a set of components which can be invoked and whose interface definitions can be published and discovered."

From an IT point of view, a SOA service is a component providing a service which exposes an interface hiding the internal implementation technology, published in a registry, and dynamically discovered. [Web](#) Services have been implementing for a while now the SOA paradigm over Web protocols. As a result, SOA is often associated to Web Services and its technologies (SOAP, XML, UDDI, WSDL...).

But SOA is more than IT although its origins are in IT.

From a business angle, SOA is a style of business architecture design and, ultimately, a way of structuring your business. It enables a Business Oriented Architecture (BOA, a new acronym) by allowing the business define Enterprise workflows around reusable business services. The granularity of the service is targeted at the business cognizant, rather than at the software application developer.

The concept of service is not new; one can find it in the Yellow Pages. An SOA service may not necessarily be implemented by software or provided by IT technology. From a business point of view, one may think of an SOA service as an internal postal service with a pigeon hole interface provided by a team of people, no matter what technology. In fact, services can be performed by human beings and/or other non-IT technologies. A service, as in every day life, is an activity executed by people and/or technology returning value to its consumer for a price.

A business service would require definition and enforcement of SLAs to control the quality of the service delivered and the service consumption within defined usage limits. Security is another

important aspect of SOA as it regulates access and enforces privacy and integrity of the information exchange in a distributed environment.

SOA Benefits

More often than not, agility and technology reuse are the major benefits associated to SOA. The reality is that SOA, frequently approached outside an Enterprise Architecture context, is developed incrementally, without the benefit of the big picture the Enterprise Architecture delivers. As a consequence, the promised agility is not achieved or is achieved late, towards the end of the SOAization of your Enterprise when technology reuse may require costly redesign.

It is worth mentioning though that Business process reuse is the advantage rather than IT reuse, since SOA identifies similar business activities and groups them in a service. SOA reduces process replication and, only afterwards, for the same type of process, the application duplication.

Nonetheless, there are a few other major SOA benefits, which should be more easily achieved, understood, and accepted. Invoking these advantages would make SOA a joyous sell rather than the often reported stressful experience.

1. Business service accountability that improves business governance

Applications and suites, usually a bundle of many functions, provide many services; in practice, a large group of business and IT people will share the responsibilities for the data and behavior of applications. But who can hold accountable such a group of individuals with many other responsibilities? In such an environment, neither accountability nor authority can be assumed for specific functions in an application. On the other hand, for an SOA business service, there is a specific function or role assigned the responsibility: it does not matter if it is an IT or business issue, there is one single point of contact for the service and an SLA to deliver against.

2. IT technology virtualization conveyed by SOA business services that hide the implementation technology and offer a clear, contractual-like interaction between business and IT.

IT becomes a service provider, offering business services at a QoS secured by an SLA, well comprehended, quantified, and eventually remunerated by the business through a payback mechanism. This is a major achievement since your applications and technology are hidden behind IT services with contract interfaces supplied to the business. From a business perspective, this is what really counts. No longer the division between business and IT, but well cut interfaces, SLAs, and contracts. No more blame culture. The separation of concerns pacifies the parties.

3. Untangling the applications providing a clean architecture, reducing the side effects of change

There is no more random access, through the back door, to parts of applications or databases, which makes any change a burden and any modification of an application a major risk because of unforeseeable effects.

4. Extended lifetime for your legacy applications, reducing the immediate pressure to replace them

This is achieved through the encapsulation of legacy in a host of services. Although there are other increasing costs related to legacy technology, there is no more pressure to replace it; you can do it at your own convenience when a viable alternative exists. This is an extension of the technology virtualization.

SOA and BPM

The relationship between SOA and BPM has been the subject of debate. No wonder, since there are different professional groups, magazines or activities to address them.

BPM was in vogue in the 90s as BPR, Business Process Reengineering. As a concept it is the practice of discovering your processes, improving and automating them to reduce human error,

and reduce costs. There are notations and languages to describe business processes. There are models that help in evaluating the maturity of processes and frameworks, such as CMM and Six Sigma, for process improvement.

Business processes are also a part of the Business layer of the Enterprise Architecture (EA). The Enterprise processes would be described by current and target process architecture parts of As-Is and To-Be EA. Processes are still abstract in that they still have to be performed by humans and/or machines.

SOA is a style of business process design for the target architecture with processes implemented as an orchestration of loosely coupled SOA services. SOA is an evolution of BPM aiming to hide and encapsulate complexity in business services.

In SOA, the business workflows will consist of orchestrated SOA services that encapsulate both the process and the technology implementing it.

From a technology viewpoint, BPM is offered at the EA business layer, as Business Process and rules design, execution and monitoring engines. Now these are offered as part of an overall SOA proposition, since they provide an orchestration service.

On the other hand, SOA is an integration technology based on products as Enterprise Service Buses (ESB), Service Registries, and management tools. But the definition of SOA services is still in the realm of business; services should be specified by business people, since they are not an IT concern or in the IT domain of expertise. This being said, ERPs, embedding and integrating various Enterprise processes, are the products of IT companies.

SOA versus EA

SOA alone is misleading if not taken in the context or scope of the development. After all, it can be applied to any architecture (an application architecture, for instance) and not necessarily to an Enterprise Architecture (EA). A note of caution: the Enterprise Wide IT Architecture (EWITA) is often called Enterprise Architecture.

SOA harbors under its umbrella developments that are in the scope of EA. SOA does cover architecture, but it does not specifically address business process automation, IT alignment to strategy, even if it helps; it does not document the As-Is state like EA does; and it does not provide guidance for the development program as EA frameworks do.

SOA requires a large Enterprise process re-engineering and re-design effort, with significant consequences, at process, applications, infrastructure, and people Enterprise Architecture layers. Services will be reused, access will be enforced by SLA contracts, and a new SOA services governance will be in use affecting the existing organization and applications suites.

SOA development may hide the recognition and complexity of an EA program, even though it is not an EA "inhibitor" per se. SOA, given its scope and ambition, should be a joint business and IT effort, a key part of a full EA development, and not considered in isolation as a light IT Enterprise Integration effort.

SOA + EA = SOEA

A "Service Oriented Enterprise Architecture," SOEA, defined as an EA with an SO style of target architecture, would better describe the positioning of SOA with regard to EA. EA may be implemented without SOA while a stand-alone SOA development, mainly driven by a Service Orientation Architectural requirement, will tend to ignore business objectives and strategy.

The EA sets in place a process to achieve technology and organization alignment to business processes, strategy, and objectives.

SOA, as a style of business architecture, is adding value to the EA by enabling modularity at the business service level, and, as such, agility, reuse, Quality of Service, facilitating payback mechanisms and service contracts. This means a more decoupled business where services are provided and consumed based on contracts. And this offers enhanced manageability. More, there

are benefits from enabling services provided over the Web using Web Services technologies and from making possible service outsourcing on an on-demand basis, such as Software as a Service (SaaS).

SOEA, the development of an EA with an SOA flavor, must have support from top management and involve business since it requires process re-engineering, technology alignment, and firm re-organization, in other words, SOEA transforms the whole Enterprise, more than EA or SOA taken separately.

As both SOA and EA are usually initiated by IT, the lack of business stakeholders' engagement and firm's management support or funding may foil the success of SOEA.

The SOA should be initially implemented as an additional EA service layer – on top of the EA applications layer – which would exist during the Enterprise Transformation stages. Some time into the future, the Applications will be, hopefully, implemented as Business Services, and the SOA services layer will cease to exist. This requires the applications suppliers to adopt SOA, which would be an advantage for everybody except, may be, for them.

Once implemented, SOEA (EA + SOA), becomes a powerful competitive asset since it is the blueprint of a service-based Enterprise, with best of breed components easily outsourced, no matter what technology or geography.

Key Messages for the Road

- SOA is primarily a business development, a way to structure your Enterprise, a style of target business architecture, and only then an integration and orchestration technology.
- SOA must have support from top management and involve business since it requires process re-engineering, technology alignment, and firm re-organization.
- Since both SOA and EA are usually initiated by IT, the lack of business stakeholders' engagement and a firm's management support or funding may foil the success of SOEA.
- Pick only the technology you need for integration and orchestration; vendors have piled all their products in the SOA basket – BPM and rules engines, user interaction (Portals), application servers, B2B gateways, messaging middleware, repositories/registries, data management, business intelligence products, development environment and management equipment. After all SOA is about best of breed, vendor independent services.
- SOA does not succeed outside an Enterprise Architecture development since, in itself, does not cover the development process, the current situation discovery (process, apps, infra...), information architecture, the alignment to strategy...
- SOA will not succeed without an Information Architecture and service in place because, after all, SOA services will have to use the same vocabulary and documents definition.
- It is still at the top of Gartner's hype cycle so be aware that it still has to reach its plateau of productivity and its market maturity

SOA Benefits

- provides good business governance with clear accountabilities for business service delivery
- SOA transforms IT in a business service provider
- it virtualizes IT applications technology behind service interfaces and subsequently reduces the division between business and IT
- untangles the applications, removing the random, back door connections
- extends the life time of your recent legacy through encapsulation

- when fully deployed, at the Enterprise level, provides reuse and agility
- enables outsourcing, such as SaaS (Software as a Service)

Author

Adrian Grigoriu, a consultant in Enterprise Strategy and Architecture, is the author of *An Enterprise Architecture Development Framework* available from Amazon and www.trafford.com/06-0421,. He has published articles in various IT magazines, and worked in senior roles in such companies as Accenture, LogicaCMG, Vodafone, Nokia and Lucent Bell Labs. His blog can be seen at <http://blogs.ittoolbox.com/eai/grigoriu>.