

# SERVICE ORIENTED ARCHITECTURE (SOA) - MAKING THE IT INFRASTRUCTURE SPEAKS BUSINESS

*Kamal Wickramanayake (IT/Software Architect & Trainer) – kamal@swview.org  
Software View (<http://www.swview.org/>)*

**Abstract:** Service Oriented Architecture is the present way large enterprise IT systems are implemented. This paper on SOA demonstrates the business and technical capabilities of SOA. It describes how organizations should drive towards SOA and the context within which SOA should be fitted into the organizations. The paper also discusses about the issues faced by organizations specially within our parts of the world. Providing a detailed technical breakdown of SOA has not been an objective.

## 1. What is Service Oriented Architecture (SOA)?

Service Oriented Architecture (or SOA for short) is the present paradigm of implementing enterprise scale software systems. In the evolution of software engineering paradigms, SOA supersedes the procedural programming paradigm and what we had next – the object oriented paradigm. However, not like them that are close to the way software systems are coded, focus of SOA is at a higher level. SOA deals with business processes.

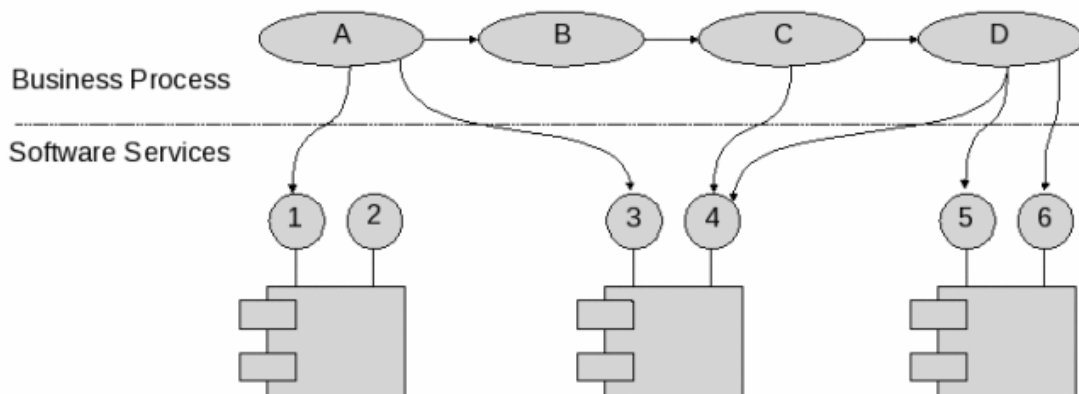
SOA describes what are known as “services” that are the automated counterparts of activities of a business process, and interactions (also known as orchestration) between many such services forming the IT applications infrastructure of an enterprise. “Services” can be implemented in whatever the programming language, run on whatever the software or hardware platform, developed using a procedural or object oriented programming language (And hence SOA is not a replacement for object oriented paradigm either).

While different people express what SOA is in different terms, here are two explanations easy to digest:

“Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.” - OASIS Reference Model

“The architecture style defining an SOA describes a set of patterns and guidelines for creating loosely coupled, standards-based business-aligned services that, because of the separation of concerns between description, implementation, and binding, provide a new level of flexibility in responsiveness to business threats and opportunities.” - IBM

And the graphical illustration below explains the same in a simplified manner where only one business process is represented:



SOA is not a technology that can just be bought and installed. It falls under the category of Enterprise Architecture. Another way to look at it is as a style. A style cannot be bought and installed, but systems built according to the style.

Maturity of SOA has grown during the latter part of the past decade from the period of hype to production ready capabilities. Hence, many organizations are presently not interested in the implementation of massive scale monolithic IT systems. For example, expensive and change resistant ERP systems are not the choice of many organizations at present times, except if they are SOA enabled.

## **2. Businesses Capabilities of SOA**

SOA development is business-driven. Business analysts can identify and define business processes and services without focusing on technical matters.

SOA provides business agility - the ability of IT systems to respond fast (even within milliseconds, seconds, minutes, hours, days,...) in terms of configurations and reconfigurations for changing business demands. Applications in an SOA environment are built by establishing interactions between services. To facilitate the same, there are service orchestration engines. Such engines understand standard compliant process description languages such as BPEL which stands for Business Process Execution Language. Hence, by changing the BPEL description, the business process can be modified. In the same way, new applications can be defined by composing already existing services together. IT infrastructure becomes live and welcomes changes instead of being static and change resistant.

Services are reusable. This is different from the other forms of reuse that software development communities are familiar with like code reuse or library reuse. Once a service is developed and deployed, it becomes a reusable asset that can participate in different business processes.

Services are broadly usable across multiple departments, branches or even different ownership boundaries. Hence, it is possible to have a business process that uses services internal to the department, some from other departments and some others from external organizations. If done well, SOA can create immense amounts of new business opportunities.

A service orchestration engine can easily monitor the performance of atomic and composite services and the way they are used - and hence can provide business intelligence easily.

Enterprise IT infrastructure can grow little by little as the demand increases. There is no need to spend massive amounts of money on large systems from the start.

## **3. Technical Capabilities of SOA**

SOA provides high degree of independence from both the technology and related functionality. Hence, this expands the possibilities of implementing services in different ways. Having multiple options to choose from provides a platform to pick the best for the job at hand based on the other environmental factors like vendor preferences, technology preferences and so on.

If implemented properly, SOA can provide true platform independence. Hence, composing applications based on an SOA environment may become feasible even if the participating software agents (other systems, etc) were not integrable during the past due to technical incompatibilities.

If done properly, systems integration can become simple and less time consuming. In some situations, even fully automated system-to-system integrations are feasible where software will discover the services, evaluate them based on set criteria, select the best, connect and use. A simple example is an organization having an item buying process where multiple item selling services (provided by item sellers) will be evaluated dynamically and the best is used. Evaluation criteria can be the compliance to some trading standards, rules and regulations to be followed by the buyer and the seller, discounts, service responsiveness or other seller service qualities.

Service reuse can be much more easier and cost effective than source code or library reuse since there is only one instance of deployment. Savings can be both short term and long term.

Developing small software bundles (services) is much more easier than building large scale systems. Individual service implementation duration and cost is predictable due to the limited scope.

## **4. Your SOA Roadmap**

The benefits of SOA mentioned above should not be the sole reasons to determine whether your organization should walk towards SOA. Any IT project should be seen as an IT enabled business project and hence an SOA value assessment is the first to perform.

A value assessment provides insights into what should be done and what should not be done. It clarifies the objectives of migrating to SOA and builds a common organizational vision.

The above described initiation phase is to be followed by a comprehensive roadmap development. This includes identification of business, technical, application and data principles (if not available already). Also included are the identification of business processes, business entities and their relationships (Business Architecture), information sources, transformations and consumers (Data Architecture), applications required (Application Architecture) and technical views (Technical Architecture).

Then follows implementation and migration planning.

Within an environment of proper governance, the SOA project can then progress from this point onwards where the plans will be executed.

Even a small SOA project may easily consume about six months. Multi-stage large scale projects may take even 2-3 years. And some organizations may prefer lengthier migrations.

## **5. Barriers in Implementing SOA**

Misunderstanding – SOA being relatively new (at least for our part of the world) can easily be something not understood well and hence leads to confusion. Hence, some expect outcomes from SOA projects that are irrelevant. And in some other cases, projects are done and the outcome is called SOA without any relevance.

No commitment from business executives – SOA is not a purely technical matter. It spans across both business and technical domains. If the commitment from business executives is poor, SOA fails to deliver value or impossible to implement.

No commitment from technical executives – Similar to the above, the reverse is also true.

Lack of skills – Particularly in our parts of the world, lack of skilled SOA professionals is an issue. Interestingly, we have off-shore companies developing software that can be deployed in SOA environments. Still, a great vacuum is seen in the domain of enterprise architecture.

Poor or no governance at all – An almost proved mechanism to fail an SOA project is to have poor or no governance.

Improper paths taken – Different organizations select different entry points to SOA. Not all produce equal or comparable values. In some cases, portions of SOA principles are applied to solve certain IT related problems in organizations like integration issues. Many fails to build the bridge between the business and IT and hence end up with suboptimal solutions.

## **6. Conclusion**

The business and technical advantages discussed above make SOA the ideal tool to be possessed by organizations. Walking towards SOA requires discipline. An SOA implementation requires the commitment from both business and IT executives of the organization and governance is almost a fundamental need to produce and operate an SOA environment that delivers its promises.