

When SOA Runs Into MDM, You Are On The Right Track!

Author: Sarah Taghavi

April 2014

INTRODUCTION

With the proliferation of the ‘out of the box’ solutions to define and implement Service Oriented Architectures (SOAs) in internal IT environment or when tapping into external business and information services, it is very important to know the main rationales behind the SOA’s existence .

SOA must always be viewed as an enterprise level concept, although it may be defined to serve selected business domains.

It is a robust integration methodology within the broader (Enterprise Information Architecture (EIA). Having a solid global understanding of SOA requirements at the conceptual level should be a prerequisite for IT and business management undertaking application integration projects.

SOA is considered a crucial shift in designing and launching software applications, today. However, for many organizations, the real advantages of bringing SOA into their IT environments are being unintentionally circumvented.

IS THIS SOA AUTHENTIC ?

Services themselves are not a new concept. SOA as a viable service implementation option has persisted for many years. However, SOA in its mainstream adoption has come about by a gradual progression of linking and exposing business services in a distributed application processing style. It has emerged to be an alternative to point-to-point integration and it is. And it is much more than that.

For various reasons (which is outside of the subject for this paper), including rigid organizational lines and higher cost, service orientation implementations used to be mostly home grown practices, within the boundaries of selected business domains.

Today, many software development professionals have become very proficient in services and XML. Web services have become a solid development area of IT. However, in too many IT shops SOA is still about the technology rather than an architecture.

SOA is not fully realized yet – not as a properly implemented layered architecture.

SOA's strengths are much more about independence and abstraction of reliable components or layers than interfacing with or wrapping existing application code for the service consumers. The global models and governance procedures which promote agility and reduce future development efforts are noticeably inadequate.

How is it possible to have an SOA, without proper architecture? Simply put , you cannot! Or you can, but that is not the authentic SOA. The authentic SOA is not just about mapping and coding interfaces. It is primarily about accurately and efficiently representing

the business itself in its IT infrastructure. And business functions do not reliably integrate, when the business's semantics are lost in translation.

Enterprise unaware data architectures are a persistent problem and the least understood culprits in their contribution to the overall complexity and the high operating cost of IT systems.

SOA embeds data architecture and governance in its own layers, rather than in some modeling tool or corporate document form-separate and foreign to developers.

It is an architecture, a sound discipline and an integrated operational system, as a whole. It is essentially about solving the enterprise integration complexities that have plagued the IT industry, since its inception. However, there is an alarming rate of failure in realizing these crucial benefits of SOA and therefore gaining any significant ROI.

Even so, there has been some good news. The significance of architecting the business and Enterprise Architecture (EA) is now being recognized, by more organizations. This is also partially due to the introduction and adoption of standards in global data exchange and NIEM (National Information Exchange Model) into federal and state agencies.

If EA efforts are consistently more prevalent, then SOA may not become yet another technology oriented 'out of the box' non-solution.

Currently, the very problems it sets to resolve (i.e. vertical solutions mindsets and application independence) are again seeping back into

the IT solutions and are impeding many organizations from reaping the main rewards of SOA.

SOA AND MDM

A main task for SOA is to enable disparate applications to exchange data and perform business services, across various functional lines. This means there is no way out of analyzing, organizing and documenting the corporate data structures! To do this, a solid MDM needs to be in place prior to defining SOA's data rules and mapping data elements.

Enterprise wide data architecture expertise is a must. It is still troublingly lacking at the conceptual level, in it's true design sense.

The blueprint of the enterprise data and business is really the basis for SOA to perform the application integration and enforce business rules. Entrenched data quality issues persist and expand, corrupting not only the application outputs, but also distorting the organizational business rules and knowledge sources.

When EIA efforts get underway, the majority of resources shift from business processes to data integrity issues. The best way to proceed further, is to put in place proper MDM and formal data governance procedures. A canonical data model developed separately from EIA and MDM will lead to even harder to decipher business rules.

In a typical SOA implementation , the underlying applications don't change (reflecting different or partial rules), SOA derives and models the global rules and so does a 'to be' MDM system (on the data). There must be a pause in SOA efforts to study and connect the all the entities of MDM and SOA. This has turned into a copy or a tool oriented (drag and drop) option. A manual review is needed! The specific problem areas of the distribution of business logic without

logically central definitions are not readily identifiable. This occurs when SOA doesn't closely follow the enterprise models. I will have more on this specific issue, in my next article.

To quickly get applications interfaced and adding on external services is still thinking local and not global. The most common reason behind the failure to realize the full potential of SOA, is the perception that enterprise architecture is unnecessary or is still an ivory tower type of practice. EA is the blueprint of IT implementation and should be closely tied into the operational environment.

EA and SOA are fundamental to integration and have many commonalities, among which are the enterprise data model and governance procedures.

The move to SOA is very promising but it needs to be methodical and consistently validated against a global business-focused abstraction layer.

SOA efforts can easily turn out to be a costly experiment with services, adding on more complexity to processing business services. The typical scenario is an incremental canonical model built or a CIM put in place to solve the organization's data integration issues, outside of EIA and without MDM efforts. In such cases, SOA really becomes about implementing the ESB middleware, outside of governance procedures.

ARCHITECTURE AS A SERVICE

In actual implementation, SOA has been most successful with plugging into new external services (such as cloud based or packaged apps). This could open the doors to the eventual outsourcing of design services or the replacement of legacy systems or internal problematic application environments. It is perceivable that even enterprise-level modeling gets stronger and more prominent, as an outsourced service.

When architecting is commonly recognized as a requirement rather than a luxury in IT, when various vendors strike the right balance and start guiding IT efforts via highly experienced enterprise architecture teams, architecture as a service is a real possibility. They are the entity with the time and the incentive to do so. Their own marketability and competitiveness is in moving towards a better more sound integration planning. This could be a great solution to the ever-persisting issues of enterprise business processes and data integration and achieving true application independence.

It will certainly give rise to several new issues- mainly of security and confidentiality, at all levels of IT environment. Additionally, the competitive and innovative nature of most organizations may mean keeping their business architecture internal.

Prior to bringing an SOA into their IT landscape, organizations must ask very specific questions about the design and implementation of it's components at all levels of abstraction and the problems they will be solving.

SOA is an architecture built upon the crucial need to decouple processes and implement an application agnostic operational environment. This means establishing a component based and layered approach in the design, definition and execution of the business processes.