JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm published by ETH Zurich, Chair of Software Engineering ©**JOT, 2006**

Vol. 5, No. 8, November-December 2006

SOA 2006: State Of The Art

Mahesh H. Dodani, IBM Software, U.S.A.

1 SOA GOES MAINSTREAM?

“Service-oriented architecture is losing its aura of high-risk early adoption to become a mainstream IT model with a growing body of lessons learned.” – http://www.eweek.com/article2/0,1759,2025155,00.asp?kc=EWRSS03129TX1K0000612

As we have progressed through 2006, SOA has dug its heels into all aspects of software engineering and become entrenched as the main enabler of the holy grail of aligning IT to business, along with the ability to facilitate business innovation through a flexible and agile IT. By all accounts, SOA has been adopted as the enterprise architecture approach (e.g. see http://www.forrester.com/Research/Document/Excerpt/0,7211,39289,00.html):

- Many of the largest companies in key industries including banking, insurance, automotive, retail, telecommunications, and healthcare have significant investments in SOA, and are able to innovate in their industries because of the flexibility and agility that is provided by their SOA-based IT implementation.
- Several studies (e.g. http://www-935.ibm.com/services/us/index.wss/ibvstudy/bcs/a1025716?cntxt=a1000074) of the many companies that have invested in SOA have confirmed the business value and ROI that can be derived from SOA. A key finding of these studies is that companies do not describe what they were doing as "SOA projects"; instead, they were undertaking business-focused projects and were using the principles of SOA to support their need for being able to achieve strategic innovation capabilities. Several studies find that cost reduction is the initial justification for incorporating SOA practices into projects. However, as these projects progress, companies quickly realize that the primary value and benefit that can be derived is the greater flexibility which in turn can support greater innovation resulting in greater revenue.
- From a technology perspective, SOA has also matured over this last year. As an example, an IBM technology-focused study of hundreds of companies’ SOA implementations found that the top five results were improved customer satisfaction (e.g. 77% increase in dealer satisfaction for whole-goods ordering processes), business operational savings (e.g. 67% reduction of fees compared with paper-based system), increased competitiveness (e.g. online ordering will

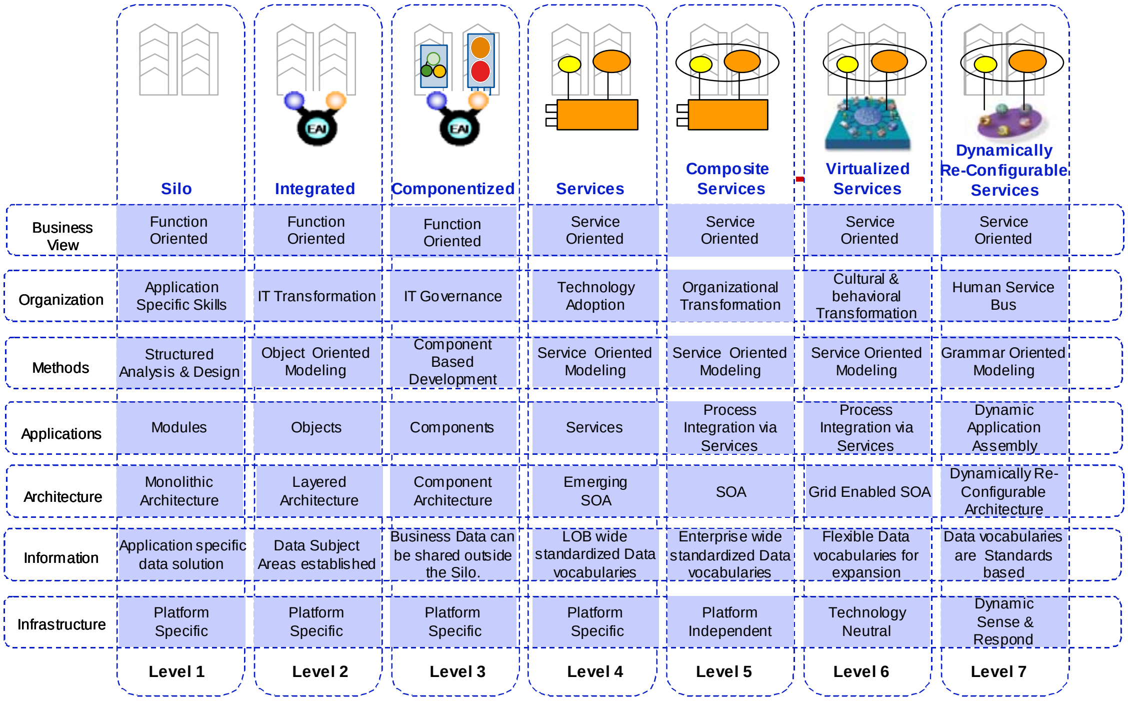
Cite this column as follows: Mahesh Dodani: “SOA 2006: State of the Art”, in *Journal of Object Technology*, vol. 5. no. 8, November - December 2006, pp. 41-48 http://www.jot.fm/issues/issue_2006_11/column5

automate approximately 30,000 transactions this year), IT cost improvement (e.g. decreased average integration time by up to 84% from six months to two-four weeks), and reduction in business cycle time (e.g. queries that once took up to 36 hours shrunk to 10 minutes or less.) These show tangible business value from using SOA.

Another measure of a maturing approach is to look at the availability of supporting assets (models, reference architectures, etc.) along with the experiences and practices of applying these assets in implementations. I want to quickly summarize what we have available currently as SOA assets in major categories that cover all aspects of an enterprises' SOA journey – maturity models, reference architectures, capabilities, and standards.

A key component of an SOA journey is to establish the business goals that need capabilities that can be facilitated by SOA, and then establish a roadmap for the journey. It is important to think of SOA adoption at two levels. At the strategic level it is important to have a vision or a master plan that can be used by the enterprise as a guideline in to be applied as various interests work together to meet specific business goals. At the tactical level are the individual project plans that are the piece parts of the roadmap. SOA will never be achieved in one big-bang project. Instead it will be implemented incrementally and the business will demand return at each incremental project step. Consequently, the roadmap must have individual project plans to meet the most immediate goals of the business and yet created in a way that is consistent with and helps the enterprise move toward the goals articulated in the strategic vision. To build a roadmap you need to have a well defined maturity model that describes capabilities that an enterprise would need to achieve a required maturity across a defined domain. The Services Integration Maturity Model (SIMM) in Figure 1 shows the evolution of our understanding of SOA maturity within an organization. Underlying each cell in the maturity model are well defined characteristics and capabilities needed to support these characteristics. SIMM can be used to understand the current or as-is state of the enterprise as well as the needed or to-be state. A catalogue of projects and capabilities are defined to move from one cell to the next in a given domain (row.) A roadmap prioritizes these projects and creates a step-by-step process and timeline for getting an enterprise from their as-is to their to-be states. Such a roadmap provides the overall blueprint for how an organization will move towards SOA.

The next key component is a reference architecture, which shows how the key principles of SOA are realized to achieve the necessary alignment of IT with business and providing the flexibility within IT to respond with agility to business changes. There are many SOA Reference Architectures – the one used within IBM is shown in Figure 2. The Services layer is the specification of the services that are used by the service consumers and implemented by the service providers.



	Silo	Integrated	Componentized	Services	Composite Services	Virtualized Services	Dynamically Re-Configurable Services
Business View	Function Oriented	Function Oriented	Function Oriented	Service Oriented	Service Oriented	Service Oriented	Service Oriented
Organization	Application Specific Skills	IT Transformation	IT Governance	Technology Adoption	Organizational Transformation	Cultural & behavioral Transformation	Human Service Bus
Methods	Structured Analysis & Design	Object Oriented Modeling	Component Based Development	Service Oriented Modeling	Service Oriented Modeling	Service Oriented Modeling	Grammar Oriented Modeling
Applications	Modules	Objects	Components	Services	Process Integration via Services	Process Integration via Services	Dynamic Application Assembly
Architecture	Monolithic Architecture	Layered Architecture	Component Architecture	Emerging SOA	SOA	Grid Enabled SOA	Dynamically Re-Configurable Architecture
Information	Application specific data solution	Data Subject Areas established	Business Data can be shared outside the Silo.	LOB wide standardized Data vocabularies	Enterprise wide standardized Data vocabularies	Flexible Data vocabularies for expansion	Data vocabularies are Standards based
Infrastructure	Platform Specific	Platform Specific	Platform Specific	Platform Specific	Platform Independent	Technology Neutral	Dynamic Sense & Respond
	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7

Figure 1: Service Integration Maturity Model

This services layer directly supports the principle of "programming to an implementation not to an interface." As shown in Figure 2, atomic services can be exposed directly to consumers, or exposed via a business process; and implemented via service components or directly (via a service façade) through the operational systems. The consumer of the service does NOT know how the services are implemented, which facilitates great flexibility in changing implementations without impacting the consumers. The services layer also shows the ability to compose services to achieve increasing levels of behavior. The business process layer shows the ability to separate collaboration/control behavior and logic from the underlying business behavior and logic that is provided by the services themselves. This approach to separating behavior logic from business logic (especially when defined "declaratively") supports changing of behavior without the need to re-code the underlying logic. The reference architecture directly handles separation of concerns by providing separate "back planes" that support integration (the ESB), qualities of services, data architecture and governance.

The next key component are the capabilities to support the SOA principles, the Reference Architecture, and the underlying services lifecyle. Figure 3 shows the key functionalities that are required to implement comprehensive, enterprise wide SOA solutions and to support the SOA lifecycle (model, assemble, deploy and manage.)

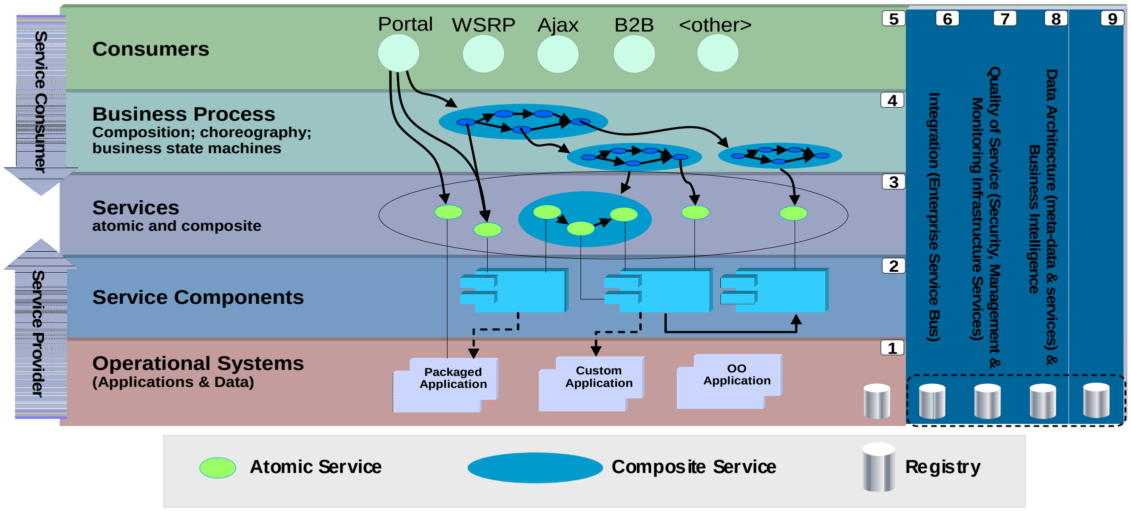


Figure 2: SOA Reference Architecture

The Development Services are used to implement custom artifacts that leverage the infrastructure capabilities, and Business Services are used to monitor and manage the runtime implementations at both the IT and business process levels. At the core of the SOA Reference Architecture is the Enterprise Service Bus which delivers all of the inter-connectivity capabilities required to leverage the services implemented across the entire architecture. Transport services, event services, and mediation services are all provided through the ESB. The SOA also contains a set of services that are oriented toward the integration of people, processes, and information through Interaction Services which provide the capabilities required to deliver IT functions and data to end users, meeting the end-user's specific usage preferences. Process Services which provide the control services required to manage the flow and interactions of multiple services in ways that implement business processes. and Information Services which provide the capabilities required to federate, replicate, and transform data sources that may be implemented in a variety of ways. Many of the services in an SOA are provided through existing applications via the Access Services; others are provided in newly implemented components via Business Application Services; and others are provided through external connections to third party systems via the Partner Services. Underlying all these capabilities of the SOA is a set of Infrastructure Services which are used to optimize throughput, availability and performance. IT Service Management Services include capabilities that facilitate the management and security of the deployed services, composite applications, and hardware/storage/network resources. These services also provide the capabilities to monitor the deployed environment to collect both technical and business KPI metrics and present these via appropriate “dashboards” for inspection by appropriate personnel and to help them take the necessary actions to optimize the managed environment or the business service/process. Note that all of these capabilities are provided by several well established products and tools.

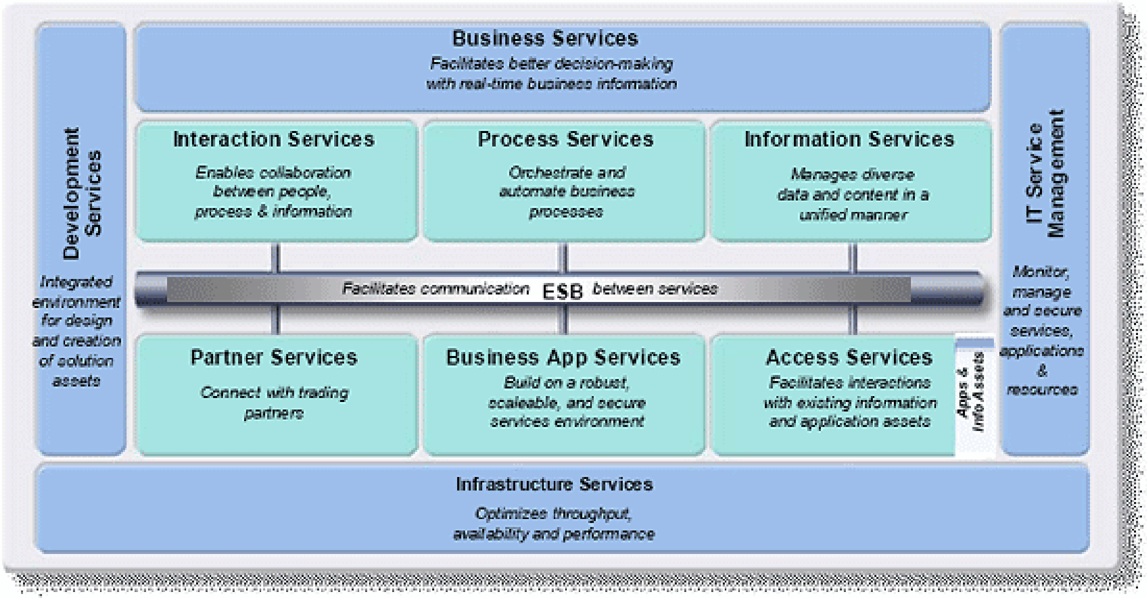

Business Services
Facilitates better decision-making with real-time business information
Development Services
Integrated environment for design and creation of solution assets
Interaction Services
Enables collaboration between people, process & information
Process Services
Orchestrate and automate business processes
Information Services
Manages diverse data and content in a unified manner
IT Service Management
Monitor, manage and secure services, applications & resources
Facilitates communication ESB between services
Partner Services
Connect with trading partners
Business App Services
Build on a robust, scaleable, and secure services environment
Access Services
Facilitates interactions with existing information and application assets
Apps & Info Assets
Infrastructure Services
Optimizes throughput, availability and performance

Figure 3: SOA Capabilities

The final component are the underlying standards that provide the foundation of the flexibility provided by SOA, as well as to facilitate the interoperation between services both within the enterprise along with those from the wider ecosystem. Figure 4 summarizes these standards. Typically, the discussion around SOA standards focus on the infrastructure standards – those that are needed across the entire IT stack as well as to support the SOA capabilities described above. These infrastructure standards are based on widely adopted XML and Web Services (WS) specifications, and this wide adoption has been a differentiator (compared to other approaches) in the success of SOA. In order to support the alignment of IT to business, these infrastructure standards are extended into the business domain through semantic standards. These semantic standards are becoming the cornerstone of the acceptance of SOA as they define industry semantics and associated common business services that can be provided by any service provider, which in turn allows these services to be combined and used to implement the business services exposed by an enterprise. These semantic standards continue to be defined for all industries, and will be a key driver of the wide applicability of SOA. Important in the discussion of standards are also those that are used in the SOA programming model. This SOA programming model allows solutions to be created through the composition of services. These composite applications are developed by integrating people (user interactions), process (composition) and information (information) with reusable business services (business components.) A common invocation model ensures that the services within the composite applications can collaborate and inter-connect. The Service Component Architecture (SCA) provides the standard for supporting this common invocation model and the definition of services in the programming model. A related standard, Service Data Objects (SDO), provides the standard for data used by services as part of the composite application

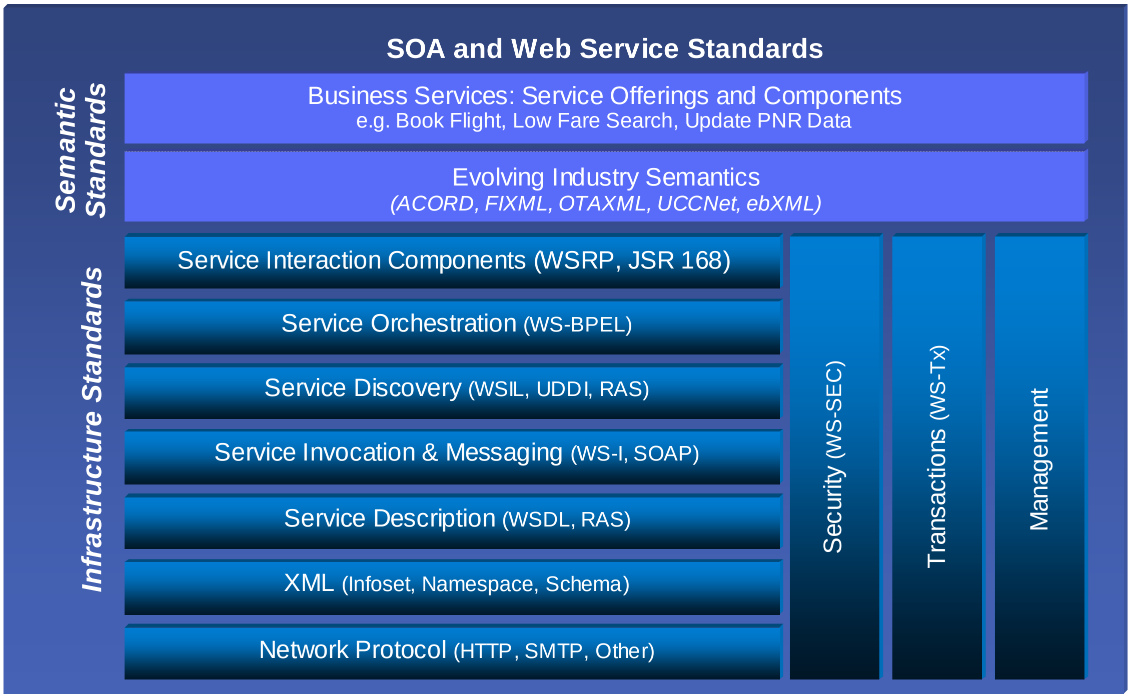

SOA and Web Service Standards
Semantic Standards
Business Services: Service Offerings and Components
e.g. Book Flight, Low Fare Search, Update PNR Data
Evolving Industry Semantics
(ACORD, FIXML, OTAXML, UCCNet, ebXML)
Infrastructure Standards
Service Interaction Components (WSRP, JSR 168)
Service Orchestration (WS-BPEL)
Service Discovery (WSIL, UDDI, RAS)
Service Invocation & Messaging (WS-I, SOAP)
Service Description (WSDL, RAS)
XML (Infoset, Namespace, Schema)
Network Protocol (HTTP, SMTP, Other)
Security (WS-SEC)
Transactions (WS-Tx)
Management

Figure 4: SOA Standards

2 STEPS TO SOA SUCCESS

One way to gauge the maturity of an approach is to look at the codification of best (and worst) practices – as these allow the continual maturing and optimizing of the approach, and the consistent application of the principles across varied implementations. For SOA there are many indications that we understand the emerging best (and worst) practices:

- Many people have published their steps to SOA success based on experiences with customers (e.g. http://www.infoworld.com/article/05/11/07/45FEsoaintro_1.html, http://www.computerworld.com/developmenttopics/development/webservices/story/0,10801,109387,00.html?source=x606 and http://news.zdnet.com/2100-3513_22-6108621.html.) Though the number of steps may differ, the essence of these experiences have common elements to them – how to ensure business and IT coordination and collaboration in driving flexible and agile business models and implementations.
- There have been collections of patterns published to codify the best practices that focus on some particular aspect of SOA. As specific examples, note the

"redbooks" that have been published on specific components of SOA (e.g. Enterprise Service Bus patterns http://www.redbooks.ibm.com/abstracts/sg246773.html?Open), on incorporating specific platforms or existing components into SOA designs (e.g. http://www.redbooks.ibm.com/abstracts/redp4190.html?Open and http://www.redbooks.ibm.com/redpieces/abstracts/sg245466.html?Open), and specific best practices that are applicable on particular SOA scenarios that enterprises experience (e.g. http://www.redbooks.ibm.com/abstracts/sg247228.html?Open and http://www.redbooks.ibm.com/abstracts/sg247234.html?Open.)

- Of course, there are also worst practices and antipatterns that have been discovered and documented for SOA, including http://www.infoq.com/vendorcontent/show.action?vcr=47, http://www.soaranch.com/russ-miles-on-soa/2006/9/7/beating-soa-worst-practices.html and http://www-128.ibm.com/developerworks/webservices/library/ws-antipatterns/. In fact, I pointed to several emerging "patterns of antipatterns" in a column I wrote earlier this year http://www.jot.fm/issues/issue_2006_07/column4 that apply to SOA as well.

The SOA success factors can be summarized as follows:

- SOA is a team sport, and the entire team is needed to implement flexible business processes. This teaming is needed in several contexts: between the business and IT sides of the enterprise and through the teams involved in the SOA lifecycle – modeling the business services/processes, assembling the services to realize these business services/processes, integrating the people, process and information needed to deploy these assembled business services/processes on an infrastructure, and managing the business processes.
- Use the SOA Foundation (as described by the reference architecture and capabilities above) for implementation. The SOA Foundation helps you establish an enterprise architecture and infrastructure based upon SOA principles.
- Leverage existing best practices and avoid common pitfalls as described above. One key consideration is to avoid the "Big Bang" approach. It is important in the SOA journey to have a roadmap that defines the projects that are needed and go through this roadmap in an incremental and iterative fashion. This approach allows measurements of how effective the implementation of the SOA principles leading to flexibility and agility are, and to make appropriate adjustments through each increment.
- Finally, governance is a must for SOA success. Any organization adopting SOA and wanting to be successful have to deal with big challenges including changes in behavior, ensuring rules and policies are followed; making right decisions; finding, using and sharing services; defining high-value business services; ensuring service characteristics are appropriately defined and managed; and facilitating communication and collaboration. Governance establishes decision-

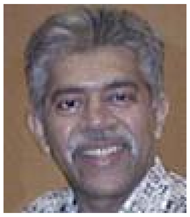
making rights along with the associated policies and mechanism to control and measure how these decisions are carried out. SOA governance focuses on the decisions across the entire service lifecycle to enable organizations to realize the business benefits of SOA and mitigate the risks inherent in SOA adoption. Specifically, SOA Governance defines the principles, processes, and roles required to manage, use and update the SOA.

3 THE ROAD AHEAD

So, what do we expect to see in 2007 around SOA? Well, we will definitely see more in each of the areas we described in this article – better support for SOA through maturity models, reference architectures, capabilities, and standards; better best practices being used to drive SOA success, and a better understanding of common pitfalls and how to avoid them.

Other areas to watch include the merging of SOA and event driven architecture (http://www.zapthink.com/report.html?id=ZAPFLASH-06092004), the integration of Web 2.0 principles (http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1) into SOA, and a better understanding of how to get an organic "nature of order" SOA design approach (http://www.cmcrossroads.com/bradapp/docs/NoNoO.html.)

About the author



Mahesh Dodani is a software architect at IBM. His primary interests are in enabling communities of practitioners to design and build complex business solutions. He can be reached at dodani@us.ibm.com.