

Where's the SOA Beef?

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

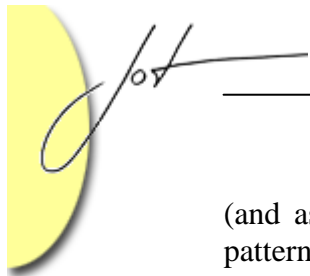
1 SERVICE ORIENTED ARCHITECTURE RECAP

“SOA is hot now because it reflects a concerted effort to have a full end-to-end architecture that works within and between enterprises.” – Bob Sutor, Blog on SOA and web services http://www-106.ibm.com/developerworks/blogs/dw_blog.jspa?blog=384

Service Oriented Architecture (SOA) seems to be on everyone “lips” and “minds.” There is a lot of ongoing work in the area, and the field continues to mature at a very fast pace. Most of the industry is focused on showing the rapid adoption of SOA within enterprises as is evidenced by the following sampling:

- Technology analysts are reporting on the state-of-the-practice, highlighting emerging practices, principles, adoption maturity and product support. For example, Zapthink provides many levels of reports covering SOA practice ranging from patterns ensuring adoption of SOA principles to antipatterns showing common practitioner mistakes (<http://www.zapthink.com/cluster.html?id=practice>.)
- Comprehensive roadmaps for enterprises to adopt SOA are being developed, for example, the one defined by CBDI at <http://roadmap.cbdiforum.com/>. This roadmap provides an organization a step-by-step guide to transition into SOA and web services. An important aspect of the roadmap is the definition of a maturity model (for the underlying web services technologies) that allows the organization to gauge the effectiveness of their adoption of SOA and the underlying technologies.
- Many books have been written on SOA and the practices of adopting it within a solution or enterprise. One such book list, http://www.amazon.com/exec/obidos/tg/listmania/list-browse/-/1MZETAUOOCQS0/ref=cm_lm_lists/002-5289932-2690458, includes a wide range of books on the topic including underlying components and technologies (e.g. Enterprise Service Bus), patterns and approaches for realizing SOA within enterprises, and tackling Qualities of Service (QoS.)

So, “where’s the beef” in SOA practices? In order to have a comprehensive practice established, we need to have methods to guide the practitioner, reference architectures



(and associated implementations) to provide a blueprint for the practitioner to follow, patterns and frameworks that provide detailed guidelines to ensure known (and best) practices are followed, and products/tools that provide the capabilities that are needed.

In the following section, we walk through the methods, reference architectures, patterns and supporting products/tools from IBM to showcase the state-of-the-practice.

2 STATE OF THE PRACTICE

We start by articulating the overall approach in adopting a SOA approach within an enterprise. We showcase the approach from the perspective of the lifecycle of an on demand service as shown in Figure 1.

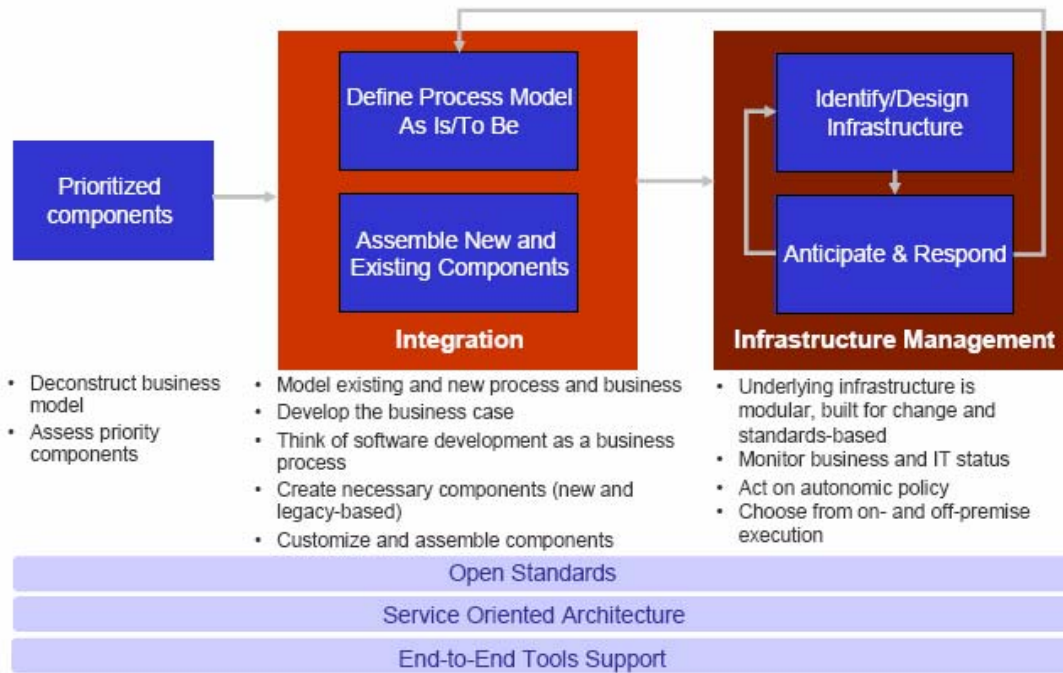
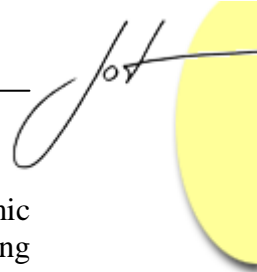


Figure 1: Lifecycle of an on demand service

As shown in the figure, the first step for the enterprise is to develop their business model to identify the core and differentiating business services that support particular business goals the enterprise cares about. These independent business services typically implement a meaningful business process or task that is provided to users or other businesses the enterprise interacts with. The business service itself is assembled from a set of integration services and the resulting assembly is deployed and managed in the on demand infrastructure. The integration services must provide all the capabilities to ensure that the business process or task is fully realized in terms of new and existing legacy-based components, and can be effectively used by the user and business communities. The deployment and management of the on demand service require support for service-level automation and orchestration services to ensure that the service can be configured,



secured, optimized and managed in an automated manner (e.g. using autonomic computing capabilities.) To facilitate flexibility and simplification of the underlying infrastructure, the use of resources (hardware, storage, etc.) is handled through virtualization services. These services are capable of “sensing and responding” to workloads to assign resources on demand.

You can follow the “day in the life” of an on demand business service in order to better understand the capabilities, services, and their interactions at <http://www-128.ibm.com/developerworks/library/i-odoe1/>. This article shows the life cycle of an on demand application from modeling of the underlying business functions and performance indicators to construction and assembly of a business service component implementing the model objectives, to the deployment of that component as a business service into the on demand infrastructure.

Figure 2 depicts a reference SOA to support the development and deployment of on demand services. Business services provide users and external businesses access to the exposed business processes and functions. Application services are used to implement the exposed business processes and functions, which are in turn deployed and managed using the infrastructure services. These services communicate, interact and integrate with each other through the enterprise service bus.

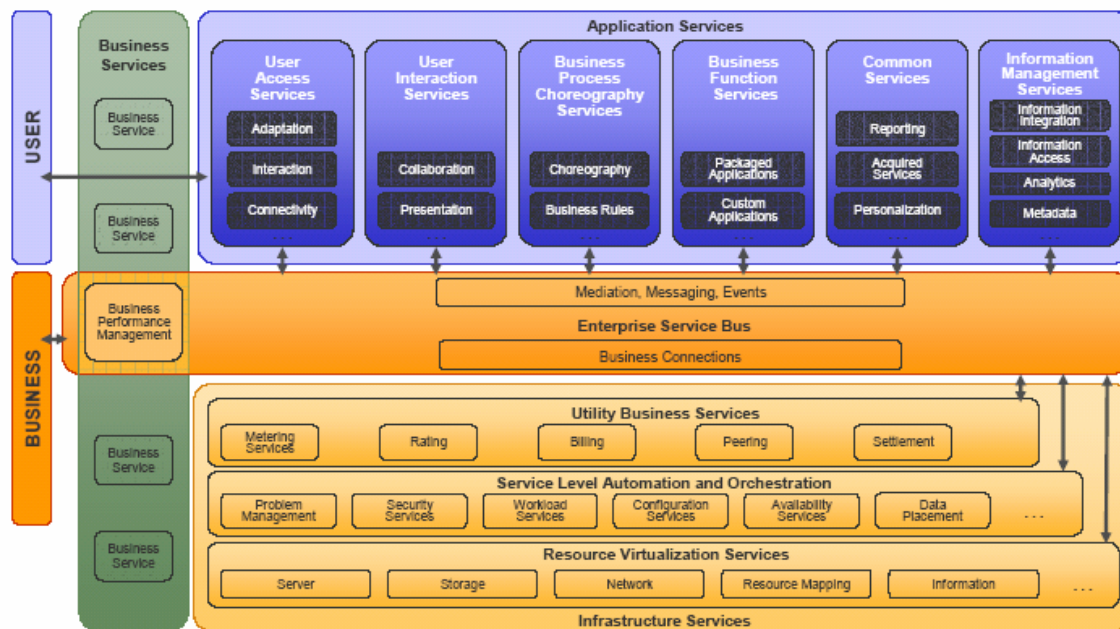


Figure 2: on demand operating environment Reference Architecture

Exposed business services are implemented using application services, which in turn comprise the following services. User access services provide support for handling various device types, interaction modes, and connection mechanisms that users can use to access the business services. The user interaction services complement these access services to facilitate a collaborative environment for the users to participate in the business process. Business function services provide “atomic” functions (e.g. packaged

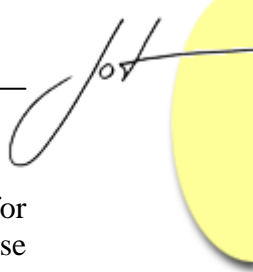
applications) that are needed to implement the business service. Common services provide “helper” functions to augment the implementation. Information management services support all the information needs of the business service, including access to integrated information on disparate data sources, analysis of the data to provide business intelligence, and manipulation of the service metadata. The choreography of all the needed services to implement the business services is supported by the business process choreography services.

The implemented business service is deployed and managed using infrastructure services. Deployed services are managed using the service level automation and orchestration services. These services ensure quality of service policies for the business service, and also support specialized management services dealing with availability, configuration, workloads, problems and end-to-end security. Autonomic managers monitor service execution based on policy declarations, analyze the service behavior, and based on the analysis may plan an action to resolve a problem and execute the plan. This autonomic approach is called the Monitor-Analyze-Plan-Execute (MAPE) feedback loop, and is an integral component for attaining infrastructure simplicity. The actual resources (server, storage, network, etc.) are virtualized and made available to the business services through resource managers. These resource managers can map available resources to the requirements posed by business services and their QoS declarations, and dynamically change these based on utilization. To support services or their components being hosted, outtasked, or outsourced, utility business services provides functions for billing, metering, rating, peering and settlement.

All interactions between services flow through the Enterprise Service Bus (ESB). The ESB supports both event-based as well as message-based interactions. The ESB facilitate these interactions by providing services to find services, handling interface mismatches, transforming inputs and outputs, and connections to legacy applications and business functions.

Note the patterns used to build infrastructure and application services – adapters enable integration of existing infrastructure components into autonomic management and resource virtualization, process choreography is used for scripting MAPE execution plans, and the ESB provides the infrastructure for exchange of events between managed elements and autonomic managers.

SOAs require standards for the definition of services and their capabilities and interactions. The growing acceptance of XML as a standard representation of structured information and of Web services standards (often called WS-* standards) have greatly facilitated the adoption of this architectural approach. To allow the ESB to provide all the needed services, it is important that it has a normalized representation of services, which is accomplished by making all application components look like WS-*-defined services. Note that the ESB facilitates SOA principles to apply to the virtualization of both business functions and physical infrastructure. The on demand operating environment spans the construction of applications as well as their deployment and management. Clients (users or businesses) only see a collection of business services and are interested in their quality of service, but the on demand operating environment shields them from



the details of application assembly and service delivery. To explore the architecture for the On Demand operating environment and its guiding principles in more detail, please visit <http://www-128.ibm.com/developerworks/library/i-odoe1/>.

To guide the development of robust, industrial-strength SOA based implementations we require supporting patterns and frameworks. In previous columns I have introduced IBM Patterns for e-business (<http://www-106.ibm.com/developerworks/patterns/>) which define a set of proven, reusable architectures that can drive the design, development, implementation and extension of complex applications. These patterns have been culled from the documentation and analysis of successful IBM projects. They match business challenges with Business and Integration patterns, use proven Application and Runtime patterns, populate the Runtime patterns with pre-tested Runtime Product Mappings, and establish best practice guidelines for application design, development and management. Figure 3 shows the extension of the IBM Patterns for e-business to address SOA. In particular, the SOA pattern can be viewed as a composite encompassing access to the exposed services using self service and collaboration patterns and interaction with other external businesses and partners using the extended enterprise patterns. The self service business pattern allows users to interact with business services, while the collaboration pattern enables business partners and users to be part of the business process workflow. These business patterns are combined on the front-end using access integration patterns and on the back-end using application integration patterns.

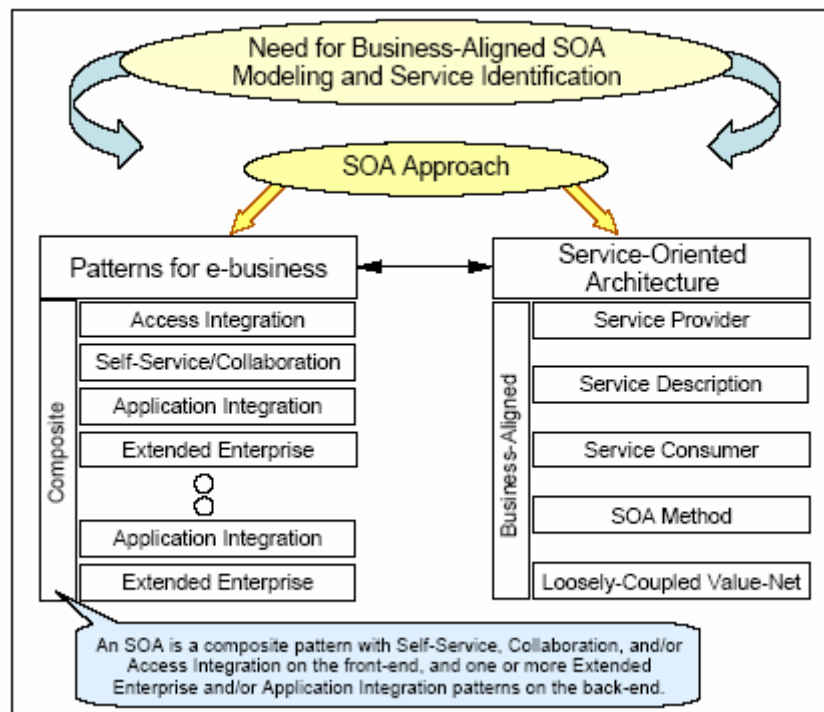


Figure 3: SOA and Patterns for e-business

To explore these SOA patterns and related best practices, access the available redbooks that address these topics at <http://www.redbooks.ibm.com/cgi-bin/searchsite.cgi?query=Service-Oriented+Architecture>.

Finally, we need to address the capabilities needed to support a successful implementation and deployment of an SOA. There are many ways to define the needed capabilities and the support of these in products and tools. For example, <http://www-106.ibm.com/developerworks/library/i-odoe2/> defines the capabilities required to support integration and infrastructure management that are needed to build an on demand service. Figure 4 shows the capabilities that are needed for integration and the products and tools that support these capabilities. For more detailed coverage of these capabilities and products, visit <http://www-306.ibm.com/software/info/openenvironment/soa/>.

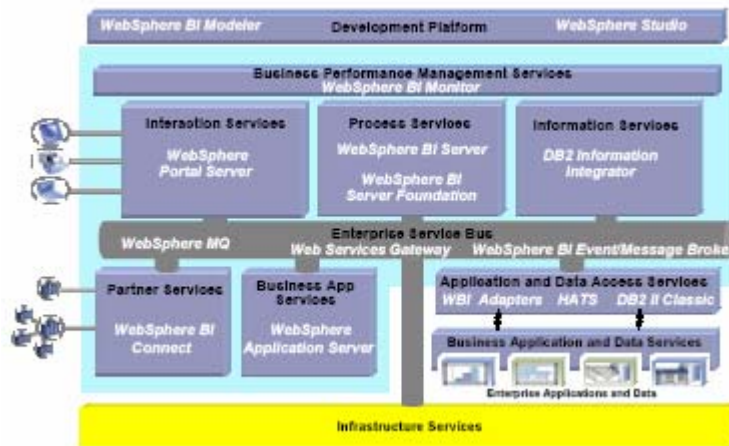


Figure 4: SOA Capabilities and Product/Tool Support

This column has provided a comprehensive view of the growing maturity of the state-of-the-practice for SOA, and presented all aspects of the needed support including methods, reference architectures, patterns, and products/tools. Underlying this maturity are the many projects that have successfully implemented SOA. These projects continue to prove that SOA is “beefy” and will become the de-facto guiding principle in the development of complex applications.

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 on demand business solutions. He can be reached at dodani@us.ibm.com.