



BPM and SOA

Mike Rosen

CTO

AZORA Technologies, Inc.

mike.rosen@azoratech.com

SOA, EA, and Tools

In my last column “SOA, EA, and Tools,” I looked at the relationship of SOA to Enterprise Architecture. In this column, I’m going to continue that theme to explore the relationship of SOA tools with EA and Process Modeling Tools.

Layered SOA Architecture

To start, we need to review and update the layered architecture that I described in my column “BPM and SOA: Where does one end and the other begin?” The original architecture described a four layered architecture that used SOA to support BPM. Figure One shows an updated version of the architecture that is expanded to incorporate enterprise business architecture concerns. This is done by adding an Outcome layer that ties performance of the business processes to Key Performance Indicators (KPI) at the enterprise level and support BAM and BI tools and initiatives.

The original four layers remain essentially the same. SOA provide the application platform that bridges the business processes and the operational resources (systems), as shown in the figure. The business services layer provides interfaces that directly support executing process tasks. But it defines those interfaces within an enterprise context to support consistency and reuse. A key attribute of the interface design is the use of common information across service interfaces. At the resource level, the integration layer exposes existing capabilities as integration services. But, it doesn’t do this by directly mapping existing applications as services. Rather, it provides new service interfaces based on enterprise semantic and functional requirements and maps them to the existing systems. Finally, these top and bottom layers are brought together through service composition. There will undoubtedly be some trickle down into the lower layers based on supporting business outcomes at the top. At each layer, we may be required to collect and report some information that was not part of the original layered architecture design.

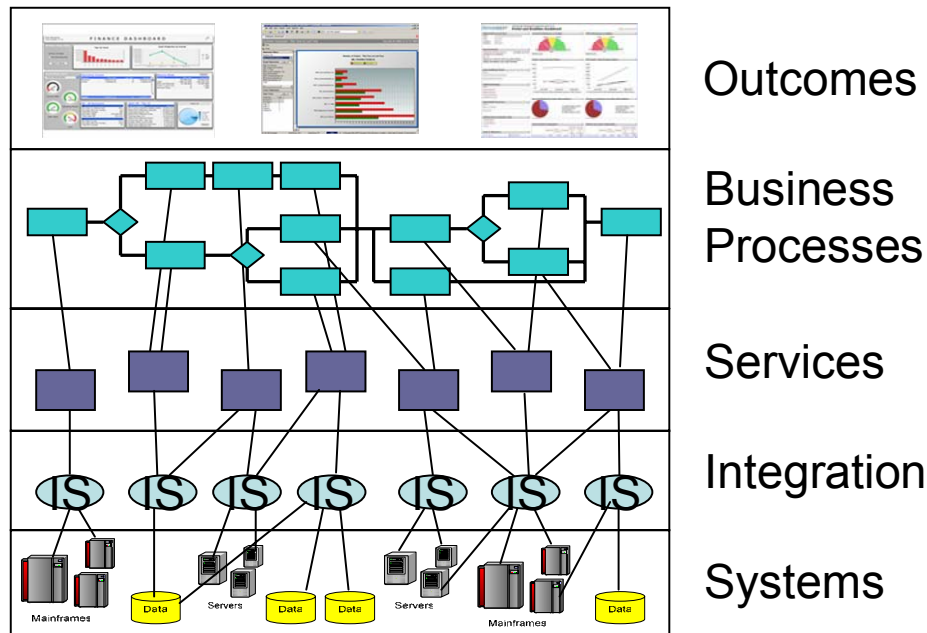


Figure 1. Layered Enterprise Architecture

Enterprise Repository

There is another piece of background information that we need to cover before pulling everything together – the enterprise repository. Many of the tools described in the BPTrends 2006 Enterprise Architecture, Process Modeling, and Simulation Tools Report support – or are based on – a repository. The repository is the central storage of the enterprise information and models. Although all products are different (some having more, less or different organization), Figure 2 shows a representative repository layout. One benefit of the repository is that if all the different models are stored in the same place, the relationship between the different assets can be maintained, discovered, or managed. This is especially true if all the models are based on a common, underlying metamodel, such as MOF.

I've divided the repository into four sections based on subject matter. Starting on the left is the description of existing systems, applications, products, platforms, etc. This is what we might call the IT inventory, which describes the IT assets of the enterprise. We also might think of this as describing the enterprise technology architecture. In the middle is the set of models (or other artifacts) that describe the business. We might think of this as the business architecture, which describes strategy, goals, outcomes, KPIs, organizational structure, etc. On the right are the artifacts that describe specific IT solutions. This would include BPM models, information and data models, software design models, SOA Service descriptions and models, etc. These are models of specific IT applications and artifacts that conform to the enterprise information and application architectures.

The application architecture, in particular, is not so much the design of a specific system as it is a pattern for building a set of similar systems. We can also think of this as a “reference,” architecture,” which leads us to the final segment of the repository. On the bottom is where we describe the architectures to which the rest of the contents of the repository conform. This is

made up of a set of reference architectures, and the formal metamodels that define all of the repository models and assets (and enable the management of relationships between assets).

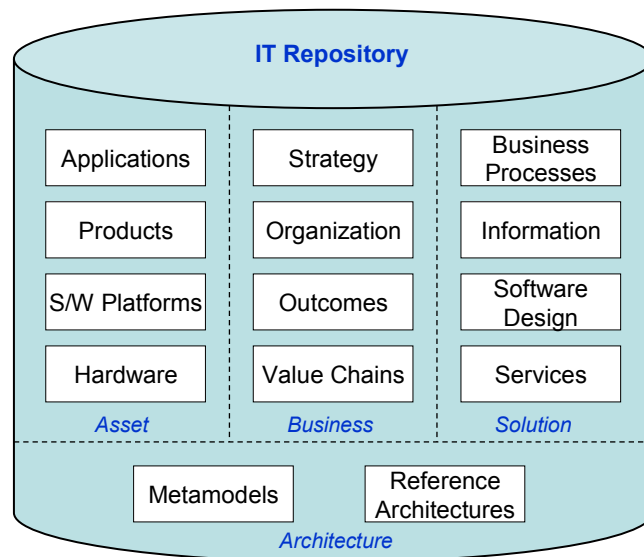


Figure 2. Sample IT Repository Organization

Tying the Tools Together

Now, let's look at the relationship of SOA, EA, and BPM tools using the layered architecture to describe the run-time environment, the tools to describe the design time environment, and the repository to represent the overall collection of runtime and design time information. Figure 3 illustrates these relationships.

We'll describe the relationships by working through the different tools (yellow boxes in the center), starting from the top. At the very top are the Enterprise Architecture tools. We use these tools to create the overall enterprise architecture – perhaps to describe the enterprise business architecture and inventory the assets, as illustrated by the arrows leading from the EA tool into the repository. More often than not, the repository is bundled with or integrated into the EA toolset.

We generally use a different set of tools to define and collect the analytical and monitoring information and to display and report on it. The drawing shows that the analytics / monitoring tools are used to define the outcomes, and used to define the runtime environment for monitoring and reporting on the outcome.

Next, we come to the BPM design tools. These are optimized for design of business processes models as illustrated by the arrow going from BPM to the repository, and they define the runtime environment for the business process layer and the BPMS.

Of course, enterprises are a combination of processes and information. We want to make sure that we share common information between processes, and use the same information for the same purpose in multiple processes. Enterprises traditionally use data modeling tools to define the information assets. Again, we show the information modeling tool providing models for the repository, and providing information definitions at both the process and services runtime layers.

Finally, we get down to the definition of the services themselves. SOA / software design tools are used to define the service interfaces and implementations. We show the service design artifacts going into the solution section of the repository, and the service design driving the runtime of the service and integration layers. Many organizations have adopted the use of a service repository or registry to support design time reuse, so we have included that in the mess as well. Ideally, there would be some integration between the EA repository and the service repository where the information and artifacts would be stored in one place and referenced from the other.

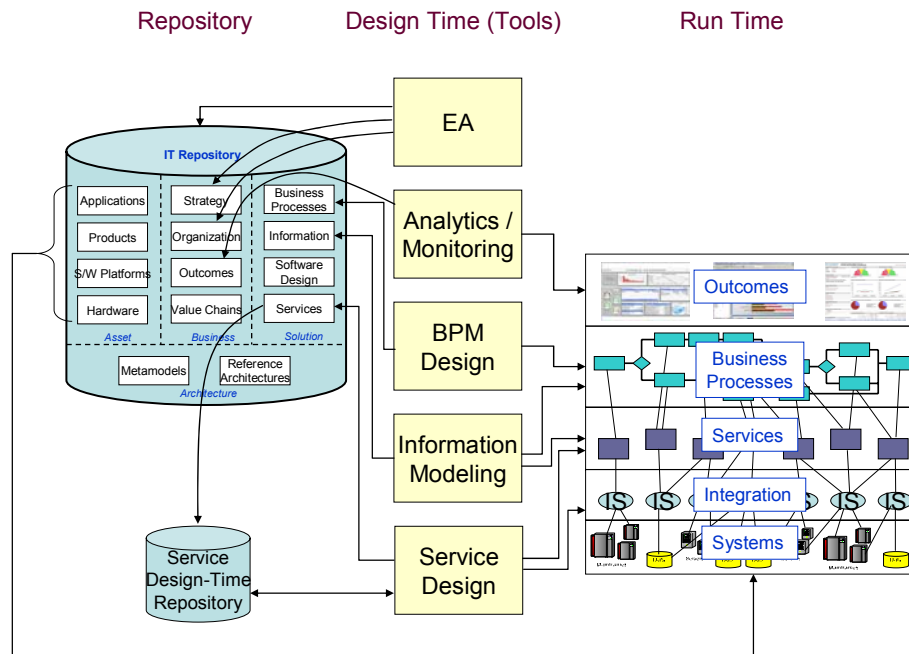


Figure 3. Relationship between EA, SOA and BPM Tools

This points out the challenge that enterprises face regarding tools. Not any single tool, or tool suite, provides all of the capabilities required to design and operate the SOA / BPM / Outcome layered architecture. And, different tools and suites will have different sets of capabilities, leading to gaps and overlaps in the overall tool space. Obviously, integration between tools is going to be extremely important. That's where support for and conformance to standards becomes critical. So, as you evaluate the design needs for your enterprise, keep this overall picture in mind, understand the role and relationship of each of the different tools, and look for conformance to standards to ease integration.