BPM and SOA

Mike Rosen
Chief Scientist
Wilton Consulting Group

Mike.rosen@WiltonConsultingGroup.com

Data, Information, and SOA

I had an interesting conversation recently about enterprise architecture frameworks and data. At its core, an EA framework is a model and classification of how we can think about the different aspects of enterprise architecture. One common EA categorization divides the enterprise into the domains of business, data, application and technology (sometimes referred to as BDAT). An important value of frameworks is that they give everyone who uses them the same model for thinking about and classifying architecture. But already I have a nit to pick with this classification. When I talk about the EA domains, I use business, *information,* application, and technology (notice: information, not data).

This leads to the obvious question of the distinction between data and information. To shed some light on this, we can look at a classical model of knowledge, often called the DIKW Model (Data, Information, Knowledge, Wisdom) shown in Figure 1. This model describes a hierarchy of information usage from data to wisdom. While there are many different versions of the model, some definitions I like are:

- Data - discrete, objective facts or observations, which are unorganized and unprocessed (and therefore have no meaning or value because of lack of context and interpretation).
- Information – (data in context) organized or structured data, which has been processed in such a way that the information has relevance for a specific purpose or context (and is therefore meaningful, valuable, useful and relevant).
- Knowledge – (know-how / know-when) a fluid mix of framed experience, values, contextual information, expert insight and grounded intuition that provides an environment and framework for evaluating and incorporating new experiences and information.
- Wisdom – (know-why) the ability to increase effectiveness through knowledge. Wisdom adds value, which requires the mental function that we call judgment.

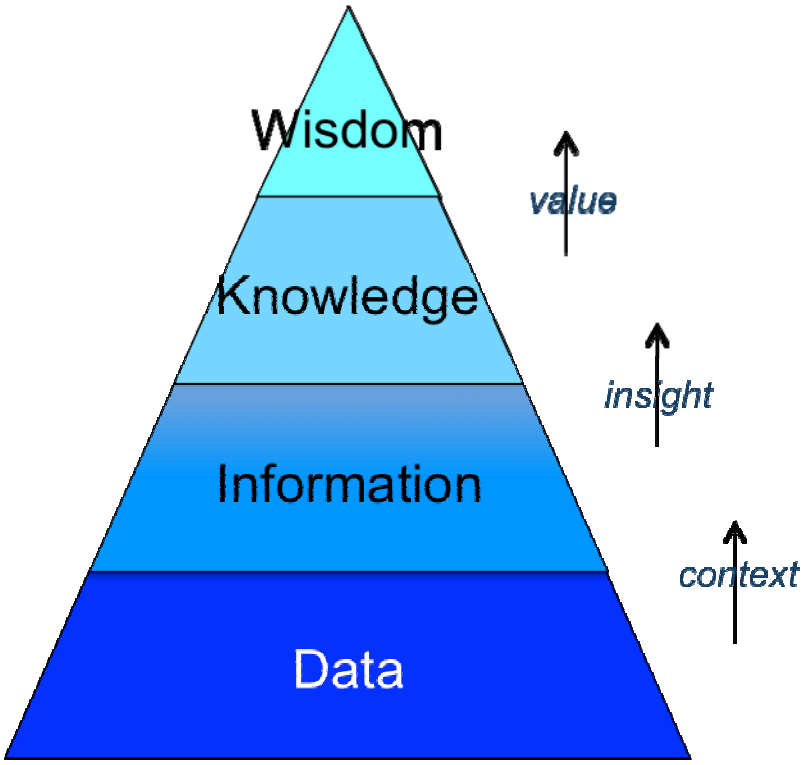

Figure 1 – DIKW Model

In other words, data is a collection of unorganized / uninterrupted facts. When we put those facts in the context of a schema or classification, then we have useful information. When we put that information into the context of experience (interpretation), then we have knowledge. When we apply that knowledge to add value, then we have wisdom.

So what's the big deal? One of the big challenges of Enterprise architecture is to understand the big picture view of the enterprise, and to put things (processes, applications, data, and systems for example) into that larger perspective. Not understanding that applications are part of a larger enterprise has lead to many of the problems that enterprises face today with complexity, redundancy, and inconsistency, and which incidentally, SOA is expected to help address.

So, from an enterprise perspective, it is 'data in context' that we are particularly concerned with. And what is data in context? Information. Hence BIAT, not BDAT. Now this perspective also happens to be particularly important when working with SOA. To see why, let's review the relationship between business processes and services. BPM decomposes business processes into a set of tasks. The tasks of a business process are implemented by operations of a business service. For the process to work as a whole, information needs to be shared between the different services that are invoked during the process's execution. That information is passed in and out of service interfaces. When I talk about service design, I often talk about "the information that needs to be shared between services for them to work together in a business process".

Information, not data. While each service implementation may have it's own data definition and storage, those details should be encapsulated behind the service's interface, which should be defined based on shared semantic information (in the context of the process or better yet, of the enterprise). Figure 2 shows the different types of data and information in SOA.

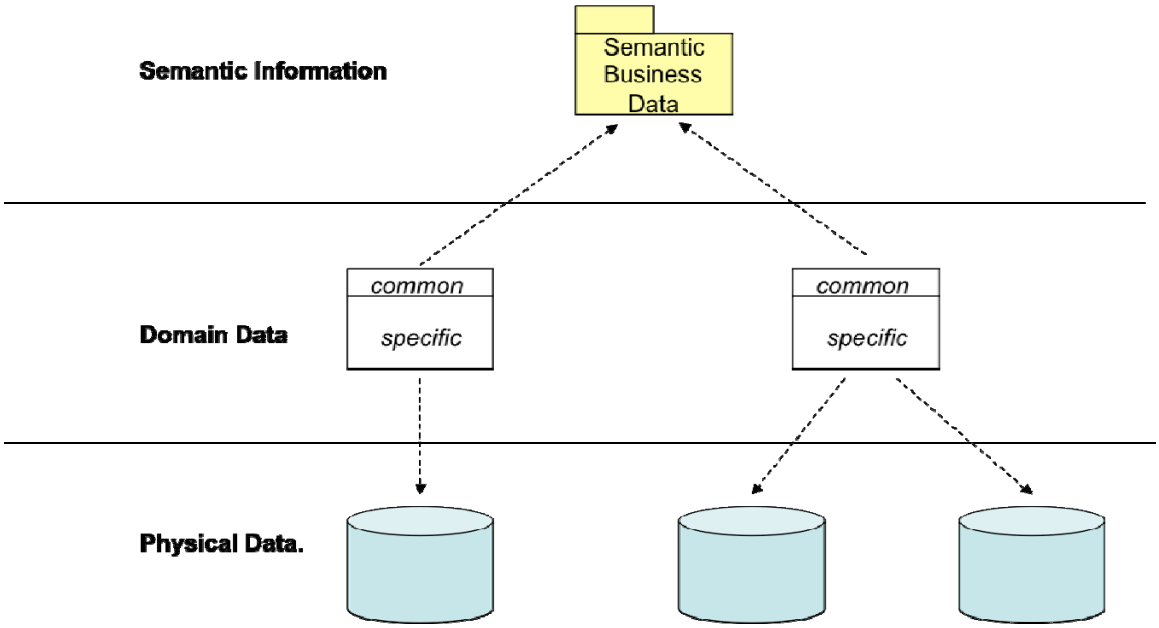

Figure 2 - Data Layers for SOA

We can describe the different types of information in terms of three layers (from bottom to top): physical data (sources), domain data (service implementation), and semantic information (shared). Physical data is the persistent enterprise data, usually preexisting in enterprise data stores. Domain data is the classes that encapsulate information needed to implement services., or views of the data. This often uses some form of object/relational mapping. Semantic information is the information exchanged between service consumers and providers. It is often a non-normalized view on domain data or data sources. This separation of concerns isolates the service consumers, service providers, and persistent sources to provide more reusable, maintainable, agile solutions.

- Physical data —This is the data that is actually stored on disk. The details of how it is stored are described in a database schema. The schema is optimized for the performance characteristics and requirements of the particular data store.
- Domain data—This is the data that is used in the service implementation. It is described in a standard data model and describes all of the information that is used in the implementation of a service. It represents the private knowledge of the data. A subset of the data is the service's view of the common information. Service data is a view of the physical data and may come from one or more physical data stores.
- Semantic data — This is the data that describes the common understanding of business entities and information that must be shared between services. It is described in the shared information model and is closely aligned to the business model. It is used to describe information that is exchanged through service interfaces. Semantic data is a normalized view of the common data from all the different services.

One of the key steps in the SOA design process is the creation of the enterprise semantics (semantic information model)— a definition of the standard business entities for the enterprise; for example, insurance policy, claim, and so on. A common semantic definition ensures that:

- Each term throughout the enterprise has a clear and concise definition.
- All enterprise terms are used consistently (mean the same thing and use the same definitions) throughout the enterprise.
- Each term is used in at least one process/activity definition.

- Only terms defined in the enterprise semantic information model are used by process/activity definitions.

Unfortunately, far too many SOA projects ignore the semantic information and instead design service interfaces based on the service implementation view of data. As a result, services do not work together, or processes are required to implement data transformation steps in order to coerce interoperability. It is much better to have the transformation occur once, in the service interface, rather than many times in many different services (potentially in many different ways). In some cases, the difference in data meaning is not immediately obvious, resulting in inconsistent and incorrect results.

The semantic information model is influenced by both the enterprise's business and the information architecture. The business architecture identifies the processes required to support the business goals and objectives. The semantic information model defines the information, concepts, and meanings that must be common throughout those processes to effectively pass information between the process steps.

The semantic information is not the same as the domain data. It does not define all of the details of the information needed within each step of a process. Rather, it defines the information that must be common and shared between then. Each individual process's step (implemented by a business service) provides any transformation required between the semantic information model and its own internal domain model.

Although the semantic information model seems similar to a standardized enterprise data model, the two are different and should not be confused with each other. The semantic information model defines the messages exchanged by services. The messages implement inter-service communication. Thus, they are transient and do not reside in a data store (at least not explicitly). In contrast, the enterprise data model defines the data structure and the relationships between data in the database.

Because in practice implementation of SOA often involves service enabling existing enterprise applications, changing the underlying data model is an extremely expensive proposition that often requires the complete rewriting of applications. In other words, it's probably not going to happen. Instead, a system that provides interoperability without changing existing models is much more practical and likely to succeed. SOA, and semantic information models enable this.

Such an implementation offers enhanced interoperability between services. At the interface level, all of them work with the same objects. In effect, this eliminates the need for message transformations between services. Because service interfaces are created according to the enterprise semantic information model, it is guaranteed that every service can understand and correctly interpret messages, regardless of who the service consumer is.

What information do you pass to you service interfaces?

BPTrends Linkedin Discussion Group
We recently created a BPTrends Discussion Group on Linkedin to allow our members, readers and friends to freely exchange ideas on a wide variety of BPM related topics. We encourage you to initiate a new discussion on this publication or on other BPM related topics of interest to you, or to contribute to existing discussions. Go to Linkedin and join the **BPTrends Discussion Group.**