



Enterprise Integrity: The Art of Mimicry IV

Vol. 8, No. 5

Many IT organizations find the process of implementing a SOA difficult to begin, in part because it requires thinking of the business as a collection of interdependent services. Traditional functional organization may obscure services, making them difficult to recognize and leverage, let alone computerize. Focusing on services, and understanding their nature, is a good place to start.

A service is performed anytime someone or something produces an output of some sort, whether it is material (a product, a component, or a supply), informational, or managerial. Services are always consumers as well as producers. At the edges of an organization services may appear to be pure producers (e.g., supply chain partners) or pure consumers (e.g., customers). Recognizing the fallacy of this view leads to supply chain management and customer relationship management. For example, and simplistically, if the customer is understood as a service that consumes goods and services and produces revenue, the optimization of that service clearly requires an understanding of the events, processes, goals, and activities that define the service interface.

Services should be classified into business services and technical services. Business services provide a function that is entirely understandable from a business perspective, in effect encapsulating or hiding the details of its implementation. The service definition depends on the business context, goals, and operational standards, but should not depend on the technology that is used to implement them. Alternative Technologies has long maintained that IT should be a provider of business services to its business clients, encapsulating IT resources. By definition, such business services deliver value directly relating to the business' primary purpose and can be understood and used without knowledge of IT artifacts. Business users should not have to care how business services are implemented technically, so long as they faithfully address business requirements.

By contrast with business services, technical services expose the technical functionality necessary to accomplish business services in the context of available technical resources. They provide access to IT resources, and are used by developers in composing business services. A technical service should form a scalable, robust service abstraction over technical resources so that they can be managed on the basis of capacity requirements and technical innovation.

SOA has the potential for being the main conduit for both of those service types. First generation SOA efforts have focused almost entirely on technical services rather than business services. The W3C standards based approach to service implementation, known as Web Services, has done much to give SOA credibility, but architects and developers should keep in mind that they are merely technical caricatures of business process, business transactions, and the like, which must be composed into their business counterparts. Treating Web Services as the *sine qua non* of SOA or BPEL (BPEL4WS) as the *sine qua non* of orchestration are hardly defensible on the basis of business requirements. There are, in fact, good technical reasons not to implement every service as a Web Service.

Few technical standards are motivated by business requirements (read Web Services standards if you doubt), being driven heavily by technical and vendor agendas. Nonetheless, advocates repeatedly – sometimes intentionally – confuse business terms with recently invented technical terms. Business event, transaction, activity, and process (and its variants such as orchestration, choreography, and coordination,) as used in Web Services carry tremendous, constraining technological baggage not assumed in the corresponding business terms. For example, although a standard like BPEL is certainly an important step in supporting service orchestration, it is hardly a tool for executing “business processes” – at least as any business manager would understand the term. Instead, it forces the real-world business process to be modeled in well-structured computing constructs that can be guaranteed to have a deterministic result, largely ignores human-based activities (as well as their scheduling and management), and is largely oblivious to business transaction requirements. These limitations alone make it unsuitable as the sole expression of service orchestration in composing business applications that reflect a business’ operations.

Sufficient reason for separating business services from technical services is that technical services can change independent of, and at different rates than, business requirements. If you are planning or have started a SOA deployment, run a health check. Analyze the services affected by your project. If more than about forty percent are technical services and those haven’t been driven out of necessity to support identifiable business services, you are at high risk of creating a technology infrastructure that will fail to deliver sustainable business value. Your SOA probably won’t contribute to the *integrity* of your *enterprise*.

