

Creating an Effective SOA Service Taxonomy

What exactly is a service?

BY MARK RICHARDS



It's hard to think about Service Oriented Architecture without thinking of services; after all, services are the main focus of SOA (it's even in the name). If Service Oriented Architecture is an approach where the business and technical architecture is oriented around services, then what exactly is a service? Unfortunately, the answer to this question varies greatly depending on whom you talk to and how you're using SOA in your organization. This variation tends to create quite a bit of confusion when trying to design and implement a SOA-based solution.

There are several excellent service-oriented methodologies available today, most of which describe processes for identifying, defining, specifying, implementing, and governing services. While these methodologies provide the direction and tools necessary to help realize SOA in your organization, they don't address the fundamental question about what a service actually is.

A service is hard to define because there are in fact many different types of services in Service Oriented Architecture. Understanding what types of services exist, how those service types are defined and related, and how they are communicated to the stakeholders in your organization are key to any SOA-based initiative. In this article I will describe a method for building a SOA Service Taxonomy that will help you effectively classify services for the SOA-based initiatives in your organization.

Overview

Taxonomy is a way of classifying things using a hierarchical classification structure. We use a hierarchical classification system to classify animals into phyla, classes, orders, families, genera, and finally species. Using this classification scheme we can group animals with similar characteristics and features, from the very general (phylum) to the very specific (species). We can apply these same concepts to the way we classify and define service types in a SOA. However, unlike the binomial nomenclature originally laid down by Carl Linnaeus, there exists no foundational nomenclature for developing a hierarchical classification of services in a SOA. Fortunately, creating a classification scheme and categorizing SOA services is infinitely simpler than the task Mr. Linnaeus had several hundred years earlier. However, we still seem to get it wrong.

Service Taxonomy is a way of classifying various types of services used in SOA. The purpose of a hierarchical service classification scheme is to provide clear, concise, and non-overlapping definitions for the various types of service you might use and encounter during a SOA initiative. An effective service classification will help facilitate communication between the various groups and individuals involved in a SOA initiative, from business users to application developers. It does this by providing a common and accepted language, allowing more effective communication between the various stakeholders in your organization.

Since we don't have a standard means of classifying the types of services in a SOA, we unfortunately have to create a new classification hierarchy every time we embark on a new

CORPORATE

President and CEO

Fuat Kircaali fuat@sys-con.com

Senior VP, Editorial & Events

Jeremy Geelan jeremy@sys-con.com

ADVERTISING

Senior VP, Sales & Marketing

Carmen Gonzalez carmen@sys-con.com

Advertising Sales Director

Megan Mussa megan@sys-con.com

Advertising & Events Manager

Corinna Melcon corinna@sys-con.com

Events Associates

Krisandra Russo krisandra@sys-con.com

Susan Wechtler susan@sys-con.com

CUSTOMER RELATIONS

Circulation Service Coordinators

Edna Earle Russell edna@sys-con.com

SYS-CON.COM

Consultant Information Systems

Robert Diamond robert@sys-con.com

Web Designers

Stephen Kilmurray stephen@sys-con.com

Richard Walter richard@sys-con.com

ACCOUNTING

Financial Analyst

Joan LaRose joan@sys-con.com

Accounts Payable

Betty White betty@sys-con.com

SUBSCRIPTIONS

SUBSCRIBE@SYS-CON.COM

1-201-802-3012 or 1-888-303-5282

For subscriptions and requests for bulk orders,

please send your letters to Subscription Department

Cover Price: \$6.99/issue

Domestic: \$99/yr (12 issues)

(U.S. Banks or Money Orders)

For list rental information:

Kevin Collopy: 845 731-2684, kevin.collopy@edithroman.com;

Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

SYS-CON Publications, Inc., reserves the right to revise, republish and authorize its readers to use the articles submitted for publication.

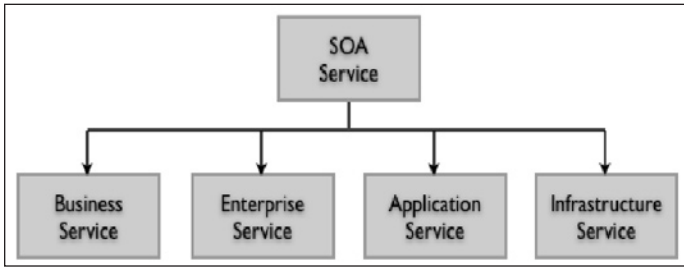


Figure1

Service Type: Business Service	
Abstract service used to represent a business process or function independent of the underlying technology or platform	
✓	Abstract Definition
✓	Defined by Business Users
✓	Course-grained
✓	Enterprise Scope
✓	Contains a name, input specification, and output specification

Figure2

Service Type: Business Service	
Abstract service used to represent a business process or function independent of the underlying technology or platform	
✓	Abstract Definition
✓	Defined by Business Users
✓	Course-grained
✓	Enterprise Scope
✓	Contains a name, input specification, and output specification

Figure3

Service Type: Enterprise Service	
Concrete implementation service used to implement a Business Service using a one-to-one or many-to-one relationship	
✓	Concrete Definition
✓	Defined by Enterprise IT Architects
✓	Course-grained
✓	Enterprise Scope
✓	Custom or vendor-supplied implementation

Figure4

Service Type: Infrastructure Service	
Concrete implementation service used to support the non-business related functions and aspects of the enterprise, subsystems, and platform	
✓	Concrete Definition
✓	Defined by Application or System Developers
✓	Medium- to course-grained
✓	Enterprise Scope
✓	Supports the system or enterprise infrastructure

Figure5

SOA initiative. Service classifications differ greatly among various companies and SOA initiatives; some get it right, but most seem to get it wrong. How to recognize a good classification and poor classification is part of what this article is about; the other part is how to build off of a standard set of service classifications to create a more effective classification scheme for your SOA initiative.

An effective service taxonomy is one in which the service type definitions are clear, concise, and most importantly non-overlap-

ping. When creating a service taxonomy you should try to simplify essential complexity (complexity that is inherent in the problem domain itself) while at the same time trying to avoid accidental complexity (unnecessary complexity we introduce ourselves). One way to accomplish this is to start with four basic SOA service types, and only extend these service types if necessary.

The Basic SOA Service Types

When developing any SOA service taxonomy, a good place to start is with the four basic service types Business Service, Enterprise Service, Application Service, and Infrastructure Service. This is the simplest possible hierarchy, and in most cases will probably satisfy the needs of your particular domain or initiative. The following diagram illustrates the basic service classification structure:

The following sections describe the attributes and characteristics of each of these four basic SOA service types. While you will most likely find these types sufficient for your particular needs, you can certainly extend these further if needed. I present some guidelines at the end of this article on when this makes sense and how to avoid the common pitfalls associated with extending this basic hierarchy.

Business Services

Services contained in this service type are considered core services in SOA. They can be derived from use cases, user stories, user scenarios, or through the service identification and specification steps found in many SOA-based methodologies. They are course-grained, usually identified and defined by business users, and represent a business process or function. When choreographed they represent the manifestation of a high-level use case or user scenario. Business Services are abstract definitions containing a service name, input specification, and output specification independent of the underlying technology. In other words, the input and output specifications to the service represent data and information collected and consumed by the service.

For example, to produce an auto quote an insurance company collects specific information from the customer, stores that information, and then presents the auto quote to the customer. The information the insurance company collects for creating the auto quote would be represented in the service input specification, and the information it provides back to the customer would be represented in the service output specification. The name of this business service might be CreateQuote. It's important to realize that the input and output data specification for this business service is completely independent of the underlying technology, language, or platform used to implement the business service. Technically speaking, while Business Services are typically implemented through standards such as WSDL (Web Services Definition Language), they can be implemented in any sort of CDL (Contract Definition Language), be it WSDL, XML, or some other interface language.

The name of a Business Service is typically constructed in a verb-noun format, with the verb being one of the typical CRUD verbs (Create, Read (or get), Update, and Delete) and the noun representing one of the major business entities found in a typical Business Entity Model. Examples of typical Business Services include CreateQuote, ExecuteTrade, GetCustomer, GetPolicy, and PlaceOrder.

A service classification template containing the primary characteristics for the service type of Business Service might look as follows:

Enterprise Services

Services contained in this service type are also considered core SOA services. Enterprise Services are concrete services that implement Business Services. The relationship between an Enterprise Service and a Business Service is either a one-to-one or many-to-one relationship (many Enterprise Services implement a single Business Service). Since Enterprise Services have scope across application domains, they are typically identified and defined by an Enterprise IT Architect or a shared services team. They are concrete services, meaning they are implemented through some sort of underlying technology or vendor product. Like Business Services, these services are typically course-grained and represent actions against major data entities.

Since the relationship between an Enterprise Service and a Business Service is commonly a many-to-one relationship, Enterprise Services usually require some sort of service orchestration, which can be implemented through an aggregate service or through middleware technology (e.g., Enterprise Service Bus or workflow engine). For example, several Enterprise Services would need to be orchestrated to implement a CreateQuote Business Service, including createCustomer, checkMotorVehicleReport, calculateQuote, saveQuote, and so on.

A common misconception is that Enterprise Services must be shared across the enterprise. Although Enterprise Services are generally shared, it is certainly not a requirement. However, Enterprise Services should be course-grained and have the ability to be shared across the enterprise if needed.

Enterprise Services have scope within the context of a Business Service, and therefore implement some sort of business logic. For example, an auditing service, although required for regulatory compliance, would not be considered an Enterprise Service. Rather, that type of service would be classified under Infrastructure Services (described below) because it does not directly serve a specific business function.

The potential impedance mismatch between the data (and format) specified in the Business Service and the data (and format) expected by the Enterprise Service implementation is usually addressed through message transformation and message enhancement, which are usually implemented through XSLT transformations located in a middleware component or Enterprise Service Bus.

A service classification template containing the primary characteristics for the service type of Enterprise Service might look as follows:

Application Services

While an Application Service is considered a basic service type, services contained in this service type are not considered core services within the context of SOA; rather, they are referred to as supporting services. These concrete services are usually fine-grained and are associated with a specific application. In other words, they have application (or silo) scope and therefore are generally not shared in the enterprise. Application Services are typically identified and defined by application developers and are specific to the application scope they are defined under.

Application Services are generally used to perform fine-grained application-specific functions such as validation, data collection, and data transfer. For example, when creating an auto quote the application developer may create services such as addDriver, addAddress, addVehicle, and so on. These services are used to accumulate

the data needed by the Business Service as defined by the Business Service input and output specification.

A service classification template containing the primary characteristics for the service type of Application Service might look as follows:

Infrastructure Services

This service type classification defines those services that are used to support the enterprise. Examples of Infrastructure Services include such aspects as logging, auditing, data access, security, and so on. These concrete services are generally shared by the enterprise and used by Enterprise Services (and sometimes Application Services).

What distinguishes Infrastructure Services from Enterprise Services is that Infrastructure Services implement non-business functionality. The gray area between Infrastructure Services and Enterprise Services appears when considering such regulatory requirements as auditing and compliance. Although some forms of auditing and compliance are designated as business requirements, these services do not specifically address a particular user or business function; rather, they support the business requirements.

Infrastructure Services are typically identified and implemented by application developers or an infrastructure support team. They generally have enterprise-level scope, which is one reason they are typically confused with Enterprise Services.

A service classification template containing the primary characteristics for the service type of Infrastructure Service might look as follows:

Service Taxonomy Guidelines

You can certainly extend the basic four service types either horizontally or vertically to suit your particular needs. However, before considering extending the basic hierarchy, you may want to consider the following service taxonomy guidelines:

- Start your service taxonomy with the four basic service types described above first. If you think you need an additional service type at the same level, make sure it is non-overlapping with respect to other service types. If you think you need an additional breakdown of service types below the basic level, then read on.
- Avoid the common pitfall of creating domain specific service types (i.e., types that are specific to divisions or functional groups within your domain). For example, if in the insurance domain, avoid creating service types such as Claims Services, Policy Services, and so on. These are not service types, but rather actual domain service names that are represented through the various types of services. To put it another way, there are many different service types that constitute a Claims Service (e.g., Business Services and Enterprise Services). Don't confuse an actual service name (e.g., Policy Service) with the service type (e.g., Business Service).
- Keep your classification hierarchy taxonomy as simple as possible and avoid accidental complexity; remember one of the goals of the service taxonomy is to facilitate communication between the various stakeholders and groups involved in the SOA initiative. A complex hierarchy of service types will create confusion and hinder the basic understanding of the service types you are using. A complex hierarchy invariably leads to increased debates and

“An effective service taxonomy developed early in your SOA initiative can significantly improve your chances for success”

lengthy meetings about how deep and wide the hierarchy should extend, and also increases the chance for overlapping service types in your classification hierarchy.

- Consider creating a simple context diagram showing the relationship between the different service types in the service taxonomy, particularly if your classification hierarchy extends beyond the basic four service types. This can greatly help in understanding where the service types fit into the big picture.
- Create an agreed-upon template for recording the definition and attributes of the service taxonomy. In general a tree-graph coupled with a context diagram and a simple document template is all that's needed to effectively document and communicate the service taxonomy. Try not to go overboard with complex tools or UML to describe the hierarchy. If you need such tools, chances are your service taxonomy is too complex and should be simplified.
- Don't wait until the SOA initiative is well underway before beginning to put together a service taxonomy; start creating a taxonomy as soon as the SOA initiative starts by using the basic SOA service types and refining it as necessary during the project

initiation phase of the initiative.

- If you end up with something like a “duck-billed platypus” when creating the service taxonomy, stop and revisit how you are defining your service types. After all, you are just classifying service types, not something as complex as the animal kingdom. Chances are you are introducing accidental complexity and making the service hierarchy more complex than it actually needs to be. Keep it simple.

Summary

An effective service taxonomy developed early in your SOA initiative can significantly improve your chances for success. Communication, clarity, and collaboration between the stakeholders in a SOA initiative are all key to the success of the overall effort. Start simple, use the four basic service types (Business Service, Enterprise Service, Application Service, and Infrastructure Service), and only extend the classification hierarchy only when necessary. Make Mr. Linnaeus proud. ■

About the Authors

Mark Richards is a director and senior solutions architect at Collaborative Consulting, LLC, where he is involved in the architecture and design of large-scale Service Oriented Architectures in J2EE and other technologies, primarily in the financial services industry. He served as the president of the Boston Java User Group in 1997 and 1998, and the president of the New England Java Users Group from 1999 to 2004. He is the author of several technical books, and speaks at conferences around the country. Prior to joining Collaborative Consulting Mark served as an executive IT architect at IBM with a focus on SOA in the financial services industry.