Executive Summary

The Four Degrees of Service Orientation

by Max Dolgicer, Senior Consultant, Cutter Consortium; and Gerhard Bayer

Cutter Consortium Enterprise Architecture Executive Summary Vol. 10, No. 7

Service-oriented architecture (SOA) has the potential to play a significant role in aligning business and IT and to deliver key benefits to the business. However, those benefits do not come automatically. The architecture needs to be defined following sound SOA guidelines, and the services must be designed according to service-oriented analysis and design (SOAD) principles, with loose coupling being one of the most important principles. Finally, the services must be implemented based on a variety of technology choices.

There appears to be a fundamental divide between (application) architectures that are based on component technologies on the one hand, and SOAs that are based on "pure" Web services technologies on the other. In practice, however, both services and components are required, depending on a number of criteria.

The accompanying *Executive Report* takes a practical view on this topic, recognizing that there are many shades of gray between an object (and component) paradigm and "pure" Web services (a "pure" Web service is defined as utilizing XML, SOAP/HTTP, WSDL, and potentially UDDI). An SOA can be designed and implemented based on a variety of combinations of technologies (e.g., Java Message service [JMS], components, HTTP, XML), which leads to architectures that are characterized by different degrees of service orientation.

CRITERIA FOR DEFINING DEGREES OF SERVICE ORIENTATION

The goal of the report is to define an SOA evaluation matrix that provides a categorization scheme for different degrees of service orientation and that helps to understand how different technologies can be utilized to build a loosely coupled SOA. Though the categorization scheme is focused on one of the key characteristics of service orientation — loose coupling — additional characteristics, such as self-containment, composability, reusability, and statelessness, must be considered when selecting the most appropriate technologies to implement SOA.

There are a number of factors that determine the degree to which services are coupled. However, for the most part, these factors revolve around the service interface (i.e., the data representation) and the interaction metaphor (i.e., the communication protocol):

- **The service interface.** A service should expose well-defined interfaces and follow the notion of encapsulation, that is, the service implementation should be separate from the interface and therefore hidden from the client's view. Service interfaces should be described using a formal mechanism. While these concepts have always been an intrinsic part of object- and component-based technologies, the problem was that even a small change in the service interface breaks the client that

is using the service. An architecture based on Web services standards, however, prescribes less rigid and thus less brittle interfaces.

- **The interaction metaphor.** Another important aspect of services is the interaction metaphor, that is, the communication protocol used for establishing connectivity between the service requester and the service provider. The object- and component-oriented technologies that are part of .NET and J2EE use communication protocols that are platform-specific and cannot interoperate without special bridging solutions. Web services, on the other hand, interact over SOAP/HTTP and do not create any middleware dependencies for the service consumer and provider.

SOA EVALUATION MATRIX

There are a number of technology choices for implementing an SOA that address the service interface (i.e., the data representation) and the interaction metaphor (i.e., the communication protocol) in different ways:

- **Data representation —** XML, XML Schema, WSDL, binary data
- **Communication protocol —** SOAP, HTTP, JMS, Windows Communication Foundation (WCF)

Given this breakdown into distinct technology choices, the report evaluates different combinations in terms of their usefulness and practicality for particular requirements scenarios and application architectures by organizing the technology choices in the form of a matrix. The matrix contains two groups of columns: one for the different approaches to data representation, the other for the different choices of interaction metaphors. Each row of this SOA evaluation matrix represents the degree of service orientation, which is formed by combining a number of elements from the two column groups. This matrix visualizes that there isn't a black-and-white choice between objects (or components) and Web services. Rather, there are several degrees of service orientation, which are rated by the level of support for the service characteristics. The four degrees (or levels) in our definition include "pure" Web services, REST-based Web services, asynchronous services, and lightly coupled components.

COMPOSABILITY OF SERVICES INTO BUSINESS PROCESSES

Service composability means that services can be combined (i.e., orchestrated) into business processes. Orchestration logic deals with the "fabric" of a business process, that is, the logic that ties together a number of functions into a coordinated flow of activities; in other words, the logic that makes up a business process. In an SOA, this logic is contained in an orchestration service that coordinates what application services and infrastructure services have to be invoked in what sequence and under what conditions.

The report discusses implementing the orchestration service layer with Business Process Execution Language (BPEL) and how BPEL relates to WSDL, which then allows us to understand how services that are not "pure" (i.e., SOAP/WSDL-based) Web services can be included in a business process that is implemented in BPEL.

CONCLUSIONS

The *Executive Report* expands the typical discussion of objects versus services, realizing that there is a continuum that consists of four degrees of service orientation. We do not claim that "four" is the magic number for breaking this continuum down into specific levels, nor would we insist that the evaluation criteria that we have provided for each level are the only ones that should be considered. Our approach should be taken as an example from which an IT organization can derive a more or less different model that fits its particular environment. Once this matrix is in place, a key followup activity that needs to be performed is to provide guidance for selecting the appropriate level of service orientation for a given scenario. The report discusses a number of general pros and cons and gives some examples to illustrate appropriate usage scenarios for each level. The next step should be to expand these criteria for each level with the different factors that are relevant to a particular project and to the particular IT organization.

About the Service

ENTERPRISE ARCHITECTURE ADVISORY SERVICE

SENIOR CONSULTANTS

Michael Rosen, Practice Director; Scott W. Ambler; Douglas Barry; Max Dolgicer, Don Estes; Pierfranco Ferronato; Clive Finkelstein; Michael Guttman; David Hay; Tushar K. Hazra; J. Bradford Kain; Bartosz Kiepuszewski; Sebastian Konkol; María Luisa Kun; André LeClerc; Jean Pierre LeJacq; Arun K. Majumdar; Thomas R. Marzolf; Jason Matthews; James Odell; Ken Orr; Wojciech Ozimek; Oliver Sims; Borys Stokalski; John Tibbets; Sandy Tyndale-Biscoe; William Ulrich; Jeroen van Tyn; Jim Watson; Tom Welsh; Bryan Wood

FOR MORE INFORMATION

For more information on Cutter Consortium's Enterprise Architecture Advisory Service or its other services, contact: Tel: +1 781 648 8700 or, within North America, +1 800 964 5118; Fax: +1 781 648 1950 or, within North America, +1 800 888 1816; E-mail: sales@cutter.com.

CUTTER CONSORTIUM