A service-oriented architecture is a style of multitier computing that helps organizations share logic and data among multiple applications and usage modes.

Analysis

The proliferation of computing styles, such as C/S, Internet, batch, mobile and interenterprise EDI, has increased the need to share data and logic among multiple application systems. However, conventional application topologies make it difficult to share data and almost impossible to share logic. In response, developers are increasingly considering a "service oriented" architecture, a way of structuring programs using multitier design techniques and middleware (see Figure 1).
A service-oriented architecture leverages the principle that many aspects of processing logic are inherently tied to the data rather than associated with a particular application or mode of processing (see Note 1). For example, the business rules, integrity checks and sequence of steps to enter an order or update a bank account may be common to all users (online or batch) of that data. The code associated with a specific function is organized in Tier A as a modular "service" or software "server" program that can be invoked by one or more "requesters," or software "client" programs. A service treats all transactions the same regardless of which client application invokes the service, although
online requesters may be given priority over batch requesters in the order in which the work is executed.

The difference between batch and online users is only on the requesting side (i.e., in tiers B and C). All online tasks (right side of Figure 1), including C/S, Internet and dumb-terminal OLTP, must be executed immediately because a person is waiting. This differs fundamentally from batch tasks (left side of Figure 1), where the processing is deferred and no immediate feedback is expected. Online tasks can be conversational, returning up-to-the-minute information about inventory levels or account balances to a user who is deciding in real time how to proceed. Online applications can also return input errors and other feedback from integrity checks so that the user can re-enter the input immediately. By contrast, all forms of batch processing must save feedback from failed transactions in files or queues for later delivery to a program or user that can correct and re-enter the transaction. Tiers B and C hold the presentation, business rules and flow-control unique to each application and access mode.

Service-oriented configurations are appropriate for many functions that are common between multiple applications. However, they rarely span the wide range of application types shown in Figure 1. Moreover, they will never be a universal solution, fully replacing traditional batch and other C/S topologies. For additional explanation, see FSIR Research Note SPA-401-071, Jan. 20, 1997.

**Bottom Line:** A service-oriented architecture is a style of application partitioning and targeting (placement). It assumes multiple software tiers and usually has thin clients and fat servers (i.e., little or no business logic on the client), but it is more than that. It organizes software functions into modules in a way that maximizes sharing application code and data.

**Core Topic**
Software Infrastructure: Data and Process Topology

**Key Issue**
How should data and process be centralized or distributed around the enterprise?

**Strategic Planning Assumption**
Service-oriented topologies will account for more than one-third of new, mission-critical operational applications by 2001, up from less than 10 percent in 1996 (0.7 probability).
Note 1

Benefits of Service-Oriented Configurations

- Enables sharing of some business logic and data I/O logic among many applications of the same or different operating modes.
- Automatically provides data sharing through the act of logic sharing.
- Shields the developers of mode-specific and application-specific presentation, flow control and business logic from having to know how the back-end service functions are coded.

Acronym Key

C/S  Client/server

EDI  Electronic Data Interchange

GUI  Graphical user interface

I/O  Input/output

LAN  Local-area network

OLTP  Online transaction processing

PC  Personal computer

WAN  Wide-area network