

Matching Integration Technologies to Business Problems

Data consistency and multistep process integration are two predominant business drivers for integrating independent applications. Each requires a different middleware technology.

Core Topics

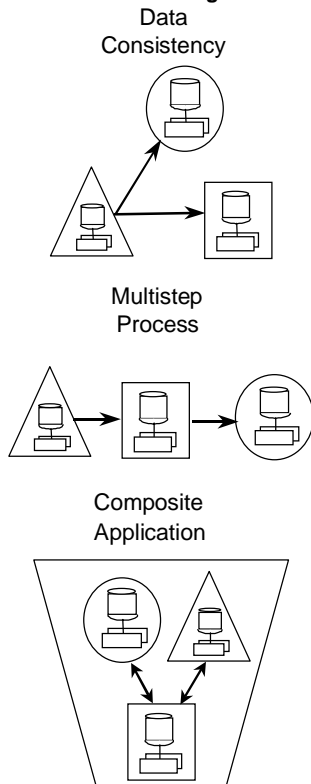
Software Infrastructure: Application Integration Strategies; Data Topology and DBMS Middleware

Key Issues

What strategies can enterprises pursue to improve data consistency and quality in a distributed computing environment?

What will be the most successful ways of integrating new, purchased and legacy applications?

Figure 1
Three Forms of Integration



Source: GartnerGroup

As vendors hurry to take advantage of the surge in interest in EAI, buyers become more confused about their application integration problems, and vendors lose their technological differentiation. We distinguish two major business drivers for application integration — data consistency and business process re-engineering — and identify the most appropriate technology for each (see Figure 1).

The Problems: The business driver behind multistep integration (see Note 1, Figure 1, Note 2 and Note 3) is the recognition that stovepiped applications — typically transaction processing (OLTP) — automate individual steps in a larger business process. Goals of this integration pattern are varied, including re-engineering the business processes for manageability and quality, reducing input errors, or speeding up the related steps for a zero-latency strategy. This pattern may or may not involve human intervention. It is a common application pattern, usually accomplished with a batch data transfer. In all multistep integration designs, the applications are logically interdependent; each step is driven by the work of another system earlier in the process.

The goal of a data consistency integration pattern is to get redundant data, stored in multiple independent systems, to “agree on the facts,” since there is no practical way to re-engineer the applications and eliminate the redundancy. While data consistency is important for both OLTP and DSS, we find it to be a primary goal within DSS architectures. Redundant and overlapping data creates many problems, such as complicating consolidated reporting and hampering new applications, e.g., customer relationship management. Data consistency management is predominantly accomplished by batch data transfer today (although messaging is increasingly being

GartnerGroup

Note 1

Composite Applications

Interactive, composite applications represent another integration driver and the most closely knit integration pattern. They always work in real time (ZLE). Composite applications are rapidly growing in popularity, as enterprises seek to provide more-customer-focused marketing and service. A composite application generally appears to be a new Web application to the end user, but, behind the scenes, it may invoke one or more mainframe transactions, and/or calls to packaged Unix or Windows NT applications.

Note 2

Data Consistency Management

This refers to the ability to improve the consistency of data elements stored in the physically disparate databases, often referred to as “synchronizing” them from a semantic perspective and sometimes from a syntax perspective as well. Data synchronization is an application process, not a technology. DBMS replication as well as message queuing middleware can be used to develop an application-specific synchronization process. When applications are logically integrated using a multistep design pattern, in most cases data consistency is also improved as a result.

Note 3

Resulting Improved Data Consistency Example

Many times, an address change event may need to be sent to 10 OLTP application systems. Integration brokers can improve the quality, read *data consistency*, of these 10 systems by sending this update program-to-program rather than inserting it directly into the databases, potentially circumventing application-level validation logic. Customer name and address data is some of the most critical and frequently referenced data.

Note 4

Example Integration Broker Products

Neon Software's Neonet
TSI Software's Mercator

Note 5

Example ETT Products

Ardent Software's DataStage
Informatica's PowerCenter/PowerMart

Acronym Key

DBMS	Database management system
DSS	Decision support system
EAI	Enterprise application integration
ETT	Extraction/transformation/transport
OLTP	Online transaction processing
ZLE	Zero-latency enterprise

considered). In data consistency patterns, the application environments are logically independent.

Potential Solutions: Integration brokers (see Note 4) are the most appropriate integration technology for multistep process integration patterns. By capturing System A's output and routing it into receiving Application(s) B, C, etc., a broader business process that is well-defined, repeatable, with sequenced steps is automated. This requires the semantics of the participating applications to be reconciled. Within the integration broker, logic and business rules are developed that transform the syntax and semantics of the sending application's data, messages or transactions into inputs that are semantically and syntactically consistent with those of the receiving systems. Three critical concepts are: 1) participating systems are *tightly dependent* (thus, if the integration technology is “down,” the receiving system may not function), 2) integration is done at the *application level* to preserve the transactional integrity of the receiving system, and 3) both systems are *process centric* in their design.

ETT tools (see Note 5) are more appropriate for data consistency management. ETT tools integrate data from multiple systems, reconciling redundant and overlapping data into a single, consistent set for delivery to one or more DSSs. Because decision making is heuristic — it is not a well-defined, repeatable process — DSS databases are *data centric* rather than *process centric* in their design. Data must be complete, consistent, timely and relevant to create business intelligence information quality. Therefore, data originating in two or more applications must be reconciled into a semantically consistent set. ETT tools enable business rules to be defined for syntax and semantic transformation. When data consistency is the sole motivation for integrating previously independent applications — the systems are otherwise independent — then *data* integration technology, i.e., ETT tools, rather than *process* integration technology should be used. Three critical concepts are: 1) participating systems are *loosely coupled* (thus, if the integration technology is “down,” the receiving system will continue to function, albeit out of sync), 2) integration is done at the *database level*, and 3) the receiving system is *data centric* in its design.

Bottom Line: Different integration problems require different integrating technologies. Although ETT tools and integration brokers have similar features and functionality, they bring different strengths to the specific problems they best address. Although the temptation to apply one technology to the other's problem domain may be strong (since most users will need both types of integration pattern), in practice this requires significant customization and extensions to be developed to supplement off-the-shelf functionality.

