

eAI

JOURNAL

www.eaijournal.com

Turning Process Into Profit

- Wiring the Enterprise for Real-Time
- The Enterprise Portal Advantage
- Securing the Mobile Economy
- Killer Content



enterprise

integrity

By DAVID MCGOVERAN



Architected Simplicity

Active Software, BEA, Forté, IBM, NEON, STC, TIBCO, Vitria, and numerous others offer a software stack for various markets, including Enterprise Application Integration (EAI), B2B, B2C, e-business, e-commerce, and so on. Evaluating these for completeness and usability is a difficult task. Starting this month we'll separate the software stack into some standard components examined from the bottom up. Call it a "quick evaluation guide." Next month we'll use the guide to discuss performance and scalability.

Architecture — A uniform distributed architecture permits all components to be used together seamlessly. Architectures have evolved from simplistic point-to-point, hub, multi-hub, and finally peer-to-peer or service-oriented, descriptive of the routing that is taken by messages between interconnected software systems in an attempt to improve scalability and performance.

Integrated Design/Development/Deployment Environment — A familiar IDE (integrated design, development, debug, and deployment environment) permits the vendor to develop all components in a uniform and cohesive manner, and the customer to extend product functionality while leveraging existing facilities. The IDE is necessary for Adapter SDKs, building new applications, and integrating multi-environment (COM, CORBA, ASPs, etc.) software functions.

System Support Facilities — System support facilities include management and monitoring, multi-threading, multi-processing, load balancing, fault tolerance, public key encryption, and the like. As part of the core architecture, they should be accessible to all components. The product and its implementations come to have a high degree of uniformity, evidenced by a standard API, and upgrades are easier.

Intelligent Adapters (a.k.a. connectors) — Adapters (by any name) are rapidly becoming the primary and most powerful component in integration suites. Early adapters did little more than provide data and message formatting. Today, these components are instrumented as many levels, providing system management and monitoring, metadata interfaces, error recovery and logging, encryption, and much more. Because functionality varies so much, spend some time comparing adapters: They are not a checklist item. The wrong adapter can mean lots of development and high support costs.

Transport — Each of the major products offers some form of native transport, a method of communicating messages. It may be synchronous or asynchronous, with subcategories such as RPC, request/reply, fire and forget (datagram), or publish/subscribe. If asynchronous, then queue management is required. An implementation may very well use multiple transports, requiring adapters between the product's native transport and some pre-existing one. Offering a native transport avoids the potential cost of a third-party product, and the need for additional selection, deployment, support, and compatibility.

Routing/Message Broker — Each of the major products offers a mechanism for flexibly routing messages. Most now pro-

vide request/reply, publish and subscribe routing, as well as both content-based and logical address routing. Content-based routing introduces a conditional element, which can (in principle) be based on anything that can be incorporated as part of the content of the message. Pre-defined routing can be extremely efficient where it is applicable.

Re-Formatting — Because aspects of the message that do not pertain to the content may need to be altered between source and destination, each of the major products offers some means of altering message format (re-formatting). Note that this is distinct from content transformation. Most often re-formatting is done within the adapter, although some products perform such functions within a central broker.

Transformation — Altering message content between source and destination requires a transformation engine, perhaps as simple as inline translation. Event-driven content merge and split, and XML support, sometimes involving multiple records from one source but separated in time, and sometimes involving multiple sources, is increasingly mandatory. Transformation rules can be extremely complex, using content interpretation, metadata (such as a record header or trailer), remote function calls, or simply field position, type, and length. Transformation engines may be hub-based, but the trend adapter-based, providing multi-source to multi-destination transformation and multi-record to single-record transformation without programming. All reformatting and transformation rules need to be maintained in an open, distributed repository.

Business Process Support — Business processes represent sequences of business events and activities, each of which may be implemented under the control of either personnel or software. Support ranges from simple workflow management to complex business process automation. A workflow or process definition controls the routing of messages, and drives destination-based transformation. Unlike the point-to-point routing of a message broker, business process support introduces a higher level of routing control end-to-end. Sophisticated business process support will include hierarchies, simulation, analysis, optimization, transactions, recovery, graphical design, and both forward and reverse engineering at the click of a mouse. A high-performance process definition and instance status repository are required.

A B2B, B2C, or EAI product without sufficient investment in an architectural stack like the one above can still be valuable, but functional benefits must outweigh higher costs of learning, poor usability, deployment, maintenance, and inflexibility in the face of unforeseen business and technology challenges. When it comes to sophisticated enterprise integration, a rich features set implemented on architected simplicity provides the most stable enterprise integrity. **EM**

David McGoveran is president of Alternative Technologies, Inc. He has more than 20 years' experience with mission-critical applications and has authored numerous technical articles on application integration. e-Mail: mcgoveran@alternativetech.com.