A Picture Of Business Integration

BY CAROL WEISZMANN AND SUSAN MESSENHEIMER

"Business integration is the culmination of successful data and/or application integration," said Jerry Bartlett, chief information officer at Ameritrade.

But too often, noted Jeff Ryan, enterprise architect at The Hartford Financial Services Group, integration solutions are chosen before there's been a clear articulation of integration requirements.

How, then, should developers and enterprise architects approach a business integration project?

What kind of integration do I need?

"You've got to begin by understanding the general nature of the problem and the outcome you're looking to achieve," advised Bartlett. "Is it purely an efficiency problem? If so, focus on the applications and the associated processes utilized. The solution may be as simple as tying several applications together to reduce duplication and rework.

"If," Bartlett continued, "the problem is that different business units are getting different versions of the 'truth' [metrics and financials, for example], then the focus needs to begin with the data sources—how are they being populated, and which applications are using the various data sources? Business leaders must be engaged in order to determine the correct version of the truth."

The requirements of a business integration project must address an assortment of capabilities, ranging from business functionality to runtime qualities such as expected volumes, response time and number of concurrent transactions, to development time qualities, including the volatility of the integration interface. Additional considerations include budget, resources, time frames, existing skill sets, strategic versus tactical concerns, anticipated reuse and so on.

"A thoughtful architecture," said The Hartford's Ryan, "will include blueprints and road maps that provide a portfolio view and anticipate rather than react to the integration needs of projects." At The Hartford, business integration is the rule rather than the exception.

Once project requirements have been delineated, they need to be aligned with the appropriate integration domains or layers.

"Think very hard about whether integration is really necessary," said Guido Sacchi, chief information officer and executive director of shared services at CompuCredit, where an innovative project called XML Gateway pioneers Web services and makes access to external data sources transparent to client applications.

"Interoperability may be all that's needed," he said. "Interoperability is cheaper and faster than integration, and today's solutions for interoperability are robust and reliable."

Key is asking the right questions.

Rhonda Ross, IS specialist at insurance software firm Mitchell International, said organizations need to know who will need access to the information, what specific data they will be using, how they will be interacting with it (read only or read/write), and when this will be occurring (throughout the day or just for monthly reporting, for example).

Mitchell International has used Above All Software's Composite Application Platform to integrate its sales force automation system (salesforce.com), customer support application (Remedy) and custom legacy billing solution (called Universe CIS) into a new composite app deployed within the salesforce.com user interface. Now Mitchell International's sales team can interact with a complete customer profile, enabling them to be more responsive and effective.

> *'That's what it's all about— solving problems and adding business value.'*
>
> —Jeff Ryan, enterprise architect at The Hartford Financial Services Group

Examining the Nature Of the Business Integration Beast

**BY CAROL WEISZMANN
AND SUSAN MESSENHEIMER**

Business integration requirements are typically presented in a couple of ways, reported Jeff Ryan, enterprise architect at The Hartford Financial Services Group.

"One is a storyboard picture of a user interface comprising functionality from disparate applications," he said. "Another is a napkin drawing showing desired interactions between applications."

As Ryan sees it, the architect's job is first to understand these requirements, then match them to the integration domain(s) where they can be addressed most effectively. Setting aside the details of varying viewpoints, these domains, or integration layers, include:

• **Data integration,** focusing on sharing single-source data among multiple applications, or synchronizing data transactions among multiple heterogeneous databases and relying on transaction management and data access components as key elements. Real-time data integration targets real-time distributed databases and data replication for throughput, high availability and disaster recovery purposes. Analytic data integration encompasses batch processes such as extract-transform-load (ETL) for data warehousing, reporting and online analytical processing (OLAP) purposes.

• **Presentation integration,** involving creation of a new user interface that provides access to more than one legacy application, aggregating such desktop functionality as portals and screen scraping. This is accomplished via the interaction of new integration logic with the application programming interface or presentation components of the legacy applications.

• **Application integration,** allowing one application to invoke the functions of another application, hiding application functionality behind abstract interfaces. Middleware that delivers interface definition, communications standards and runtime support provides robust means of achieving functional integration.

• **Web services integration,** tackling the exchange of data on large private enterprise networks or over the public Internet via standard-based application integration. Effectively architected Web services integration enables applications to provide and consume services at a functional and/or data level, resulting in a flexibility and robustness unmatched by proprietary closed systems.

• **Business process integration,** dealing with long-running transactions made up of sequences of events, triggers and human interaction.

But one size does not fit all. "The level and type of integration appropriate to one problem," said Jim Lupton, vice president and director of systems management in ISD at American Fidelity Assurance Company, "may not be so useful for the next problem."

INTEGRATION AND SOA

From a technical perspective, observed Eric Roch, national practice director of business integration at consultant (and IBM business partner) Perficient, integration problems occur when assembling legacy IT assets as components within a service-oriented architecture (SOA), which is the current state of the art for integration efforts.

Why SOA?

"If you use a data integration approach, you're moving whole chunks of data from one system to another," said Sanjay Sarathy, vice president of marketing at Above All Software. The drawbacks: difficult data integrity and synchronization issues, trouble supporting real-time requirements, and a spaghetti-works set of links when more than two apps are integrated.

Meanwhile, enterprise application integration (EAI) has been expensive, code-intensive, intrusive and—because business requirements change and integration needs change with them—ineffective, Sarathy observed. "The connec-

'The level and type of integration appropriate to one problem may not be so useful for the next problem.'

—Jim Lupton, vice president at American Fidelity Assurance Company

A single integration approach, Ross noted, may not be the best solution. "A hybrid that combines some data integration along with other types of integration could be ideal," she suggested.

The Hartford's initiatives, for instance, include many large programs and projects, such as its ICON 2.0 small-agency submission tool, implementing both presentation integration and application integration to enable agencies to better serve small business clients.

STARTING WITH THE BUSINESS

The Hartford's Ryan recommends having a predetermined strategy that aligns its portfolio of initiatives to the several types of integration domains or layers. "Project needs are the first driver," he affirmed.

"Start with understanding the business problem that needs to be solved," counseled Ameritrade's Bartlett. "That's what it's all about—solving business problems and adding business value."

Mitchell International's Ross articulates some of the questions that need to be asked:

What are the specific opportunities

or threats that your enterprise is facing? What strategy is best for handling them?

What approach will deliver the best return on investment?

The answers, she emphasized, will be different for every business. "A company like ours that has been in business for 60 years would start in a different place than a less mature organization," she said. "A company that is global and/or has been through a series of mergers and acquisitions would be facing different challenges than one operating in a single country and with a limited number of systems."

Jim Lupton, vice president and director of systems management in ISD at American Fidelity Assurance Company, believes, "First, the business needs to determine the functionality to be shared across applications and the level of data integration required."

This should be based on current and anticipated business processes and data needs. Picking a specific technical starting point should be driven by these busi-

A Picture of Business Integration

ness requirements.

"Next," Lupton said, "both the high-level business functionality to be shared across applications and the infrastructure ser- vices necessary to accomplish that level of integration should be defined. Selection of a specif- ic application or legacy system for integration should be deferred until the necessary framework, methods, patterns and policies are in place."

AN UNDERLYING STRATEGY

"My recommendation," said CompuCredit's Sacchi, "is to solve the problem from an architecture standpoint: At which integration layer can the problem at hand best be solved? In my view, service-ori- ented architecture [SOA] pro- vides the best trade-off of inte- gration and interoperability."

American Fidelity's Lupton agreed. "Regardless of the strat- egy selected, emphasis should be placed on loose coupling, com- ponent-based integration, and use of an integration services framework."

Those undertaking business integration projects using an SOA/Web services strategy do so because of the business value they derive from this approach, according to Sacchi and Lupton.

"From the business perspec- tive," said Eric Roch, national practice director of business integration at consultant Perfi- cient, "SOA in general can pro- vide benefits such as supply chain optimization and trading partner integration."

To that end, he counseled, "an enterprise should build a services blueprint based on pro- jects in the IT portfolio and business drivers. Funded pro- jects in the IT portfolio should be examined to determine if integration software will reduce costs and create business value."

Such a services blueprint— business-driven, so any architec- ture it defines supports business objectives—acts as a road map for rolling out SOA-based inte- gration. "The road map identi- fies the level of integration needed by defining the architec- ture required to support its pro- jects," said Roch.

INTEGRATION PITFALLS

"The single biggest risk is to undertake an initiative without understanding the business problem and the desired out- come," warned Ameritrade's Bartlett. "Without that under- standing, the technology group will likely not meet major expectations and, as a result, will be viewed as a failure— once again not meeting the business' needs."

American Fidelity's Lupton cautioned against direct cou- pling of applications. "It's com- plex, brittle and inflexible," he argued. "Tight coupling, chaot- ic [or nonexistent] architecture, inefficient application interac- tions, and inefficient data storage and management can be avoided with a carefully thought-out and implemented integration architecture that's solidly based on present and future business needs."

The Hartford's Ryan cited the dangers of "a one-off pro- ject approach resulting in a

Examining the Nature of the Business Integration Beast

tion points between applications become as tough as concrete," he said, "so in many cases, you're better off starting over than attempting to modify it."

SOA, or Web services, on the other hand, enables assembly of composite applications, pulling together all the necessary functionality into an integrated, interactive solution. "And the composite app can be deployed within the user interface of an existing application that users are already familiar with," Sarathy said.

Rhonda Ross, IS specialist at insurance software firm Mitchell International, concurred. "With data integration, raw data is physically moved from one application into another, leaving all of the business rules related to that data behind," she said. "In contrast, with application and Web services integration, it's possible to reuse the data without having to replicate it and physically move it between applications."

In addition, she noted, the integrity of the data can be assured because the same business rules are enforced in all applications using the data.

A Picture of Business Integration

'spaghetti bowl' of integration solutions and a 'junk drawer' of services."

Observed Ross of Mitchell International: "Integration projects often take on a life of their own and grow into projects so complex that by the time a solution is delivered, business needs have changed."

She proposed breaking a business integration project into small parts that can be delivered quickly and provide immediate benefits to the user community. This builds support for the next segment of a multi-phased project plan.

Ross also offered advice about development tools: "Select tools that are easy to use and can be utilized by analysts as well as developers. This increases the speed with which solutions can be delivered and improves the probability of providing solutions that best meet users' needs." ∎

SOA, said Perficient's Roch, "is where you have technical point solutions, such as mainframe and data integration." He pointed to those SOA software vendors emerging from the EAI space as well as the application server vendors (such as IBM and BEA) building SOA integration into their app server stack.

The result, Roch concluded, is that EAI vendors and application server vendors have created an integration software stack.

"These software stacks can be compared," he suggested, "based on an integration software reference architecture that includes such components as portals (which consume services and present them to the user), data integration tools (such as ETL), business process management and messaging. The tools a business will need from the stack depend on the integration problems it must solve to create its SOA." ∎