

To Do & Not To Do

By Vladimir Orovic

This article provides a checklist of what to do — and what not to do — when implementing an integration project.

Do ...

1. Think globally and act locally. Plan enterprisewide; implement incrementally.

The key to EAI success is planning at the enterprise level, taking into account the broadest cross-unit and cross-system requirements and long-term business plans. This will ensure that integration process models devised, and the integration framework implemented, can address these broad needs.

However, the EAI implementation itself should occur incrementally, starting with a pilot/prototype, progressing with two systems and then incrementally adding systems to the mix, as required. Risk mitigation and cost control are the two main reasons for this approach. EAI implementations will likely fundamentally change the way enterprises operate. It will make them integrated and agile, real-time structures.

The “big-bang” approach has had spectacular failures due to the size and scope of business and system changes and the inability of organizations to adapt to the new operational model. An incremental implementation approach with enterprise planning is now widely accepted.

2. Define integration framework components.

For most organizations, there’s still no single vendor that will meet all the integration needs. Different components of the integration stack (see Figure 1) need to be considered against the capabilities of different vendor products to determine the right fit for the organization.

3. Focus on business-driven goals with high cost and low technical complexity.

Considering the high cost of EAI implementations, to achieve any positive Return on Investment (ROI), start with the highest-cost business items or those in the “cost of doing business” category. While EAI has a technology infrastructure feel to it, there’s no real technical value to undertaking an EAI implementation unless there’s a clear business problem that’s addressed.

4. Treat the EAI “system” as your strategic application.

Ultimately, EAI infrastructure will become the knitting of the organizational systems and the single most critical point of failure. Planning for security (most information will pass through the middleware), high availability, recovery, etc. is critical.

5. Pursue reusable, template-based approaches to development.

When developing for message-based interconnectivity, patterns can be discovered that will foster reuse. This is particularly true for adapters but applies to business processes, too. Patterns exist at different levels: data acquisition (i.e., ways the data is extracted from one system),

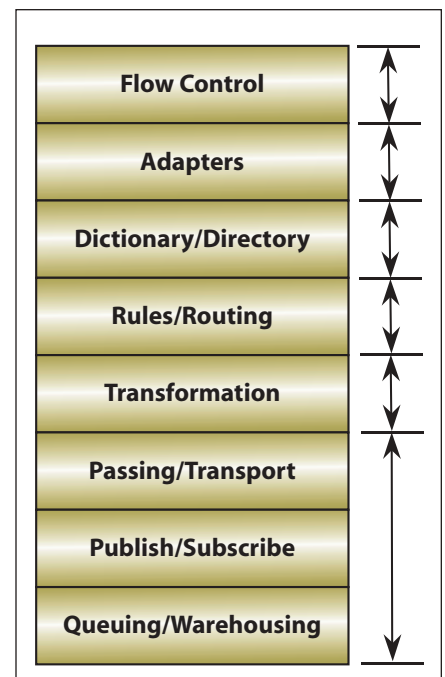


Figure 1

data transformation, workflow, etc. Cost savings can be 20 to 35 percent. Figure 2 provides an example of high-level adaptor architecture for reuse (needs to be drilled down for practical purposes).

6. Use prototyping as the project estimate generator.

There are no existing quality statistics about sizing EAI implementations. EAI implementation is highly tied to the internal processes unique to each organization. The best method for estimating is to prototype one simple business process with two adaptors and a few messages, and measure the effort and cost required. Many organizations have missed their scope and cost by multiple orders of magnitude by not having any idea about what EAI implementation means in their particular environment.

7. Think of integration at different levels of abstraction.

In terms of spatial characteristics, integration can happen at many different levels — listed here by increasing complexity: data, function-to-function, and business process. Not all integration needs to be business-process driven since that carries the highest cost. Level of integration abstraction should be chosen so that it meets the business needs, yet is least complex.

8. Expect to build application logic into the EAI infrastructure.

Not all functional elements that are part of existing systems can be invoked from other systems when integrated processes are required. During EAI implementations, there's often a clear need to integrate some amount of business logic into the middleware. An example in Canada is foreign content calculation on an account basis for a retirement plan, when the account contains multiple instruments, with each being managed by a separate system.

9. Assign project responsibility at the highest corporate level and negotiate, negotiate, negotiate.

EAI touches almost every organizational unit of the enterprise, typically changing process ownership responsibilities and ownership of business objects, and adding a political component to it. To succeed, ownership of the project must be at a management level that exercises control over all the par-

ticipating business units. Moreover, there will be different system owners for each of the systems being integrated. Negotiation becomes a matrix problem at the business process and system level — and the two often don't map one-to-one.

10. Plan for message logging and warehousing.

Not logging or warehousing messages for audit and recovery purposes is among the most common mistakes in creating EAI architectures. Over time, moving to integrated real-time enterprises will mean most business events will be handled through the middleware, which dictates the need for them to be stored for future reference. Messages in the local storage space of each individual application usually expire after a certain period. Planning for this has to happen early in the EAI implementation since message logging and warehousing must be in place from the beginning of middleware operation and because the logic to handle restart has to be built into the first adapters being rolled out. That logic will depend on the warehousing model implemented.

Do Not ...

1. Critique business strategy through the EAI architecture. Instead, evaluate the impact of that business strategy on IT.

EAI drives the need for clear definition of business strategies and processes. Often, natural reaction is to critique the business strategy and processes defined as part of EAI. This creates enormous tension between the partici-

pating organizational units and the business and IT function. In successful EAI projects, business processes and strategies are evaluated to ensure that the systems and architectures in place can support them rather than the other way around. This is a major cause of EAI project delays.

2. Purchase more than you need for a given phase.

The speed of change in EAI standards and middleware software capabilities is so rapid and vendor competitiveness is so strong that, to mitigate risks and reduce costs, it makes sense to purchase middleware products only for the phase in progress. Further, if presented with a long-term buying plan, vendors will tend to spread their discount over a greater time while ensuring (or better said, promising to ensure) "just in time delivery" of product versions with richer functionality.

3. Substitute EAI for data warehousing.

One of the most common mistakes in EAI implementation is to assume that the message warehouse that's part of the middleware is the same as the data warehouse. A second common mistake is attempting to feed the data warehouse using messaging. Consider the first case. Usually, the message warehouse contains all the data needed for warehousing, but the structure of that data doesn't lend itself to data mining and reporting by relevant data elements. In the second case, the costs of building message-based, near-real-time interfaces, where the temporal characteristics of data don't require real-time

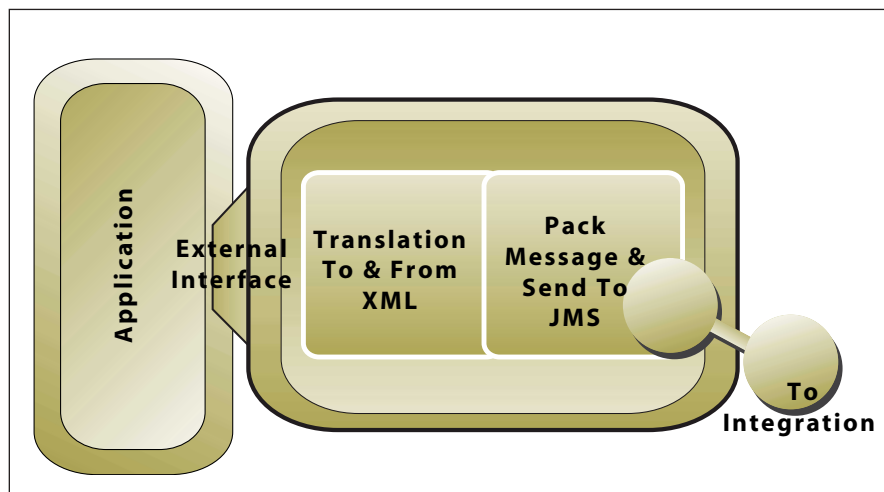


Figure 2

integration, adversely affect the ROI and unnecessarily increase the complexity of the integration architecture.

4. Force usage of near-real-time, message-based integration unless it's absolutely mandatory.

The cost and complexity of integration increase with migration toward real-time connections. The possibility that middleware software won't keep up with the business needs also increases with the move toward reduced latency of interaction. Near-real-time, message-based integration should be used only when business needs dictate (i.e., not for feeding the data warehouse). Increasing latency of connections can save 30 to 50 percent of project costs.

5. Assume existing process models will suffice for process integration; they're not the same.

Three key elements to keep in mind:

- Business processes do not map one-to-one onto system processes.
- Existing workflows deal with data level integration whereas EAI is about process-driven integration.
- Existing business process maps usually span only one business area.

To meet EAI needs, real business integration processes need to be defined. This is especially important in intracompany integration scenarios.

6. Plan to change your business processes as part of the EAI implementation.

Documenting existing integrated business processes and automating them as part of EAI is a formidable task. Changing them as part of EAI implementation adds a layer of complexity that can destroy the project. If executed properly, EAI infrastructure will automate key business processes, allowing for relatively easy changes in the future.

7. Assume that all relevant knowledge resides within the project team.

EAI requires cross-organizational process definition. It's impractical to create such a huge team to contain all resources with relevant elements of knowledge from a business and system perspective. The EAI team should continuously reach out to the broader organ-

ization when validating business and technical architecture.


8. Be driven by centralizing any enterprise-level business objects as part of EAI implementations.

Companies are often driven by the desire to centralize their core business objects in the integration layer as part of EAI implementation. This has proven unrealistic because it adds complexity to the integration and increases scope and cost. It also requires significant, usually impractical, changes to the existing application to adapt to the new "centralized" data model. Finally, it increases complexity to the adaptors. The way to go is to create a metadata layer as part of the integration infrastructure that allows real-time cross-referencing of key common business objects between different applications.

9. Be intrusive into the existing applications.

This creates maintenance issues with existing applications. All interfacing with the integration layer should be encapsulated in the adapter in a way that's transparent to the inner workings of the application.

10. Use ad hoc process and message modeling techniques.

Structured modeling becomes critical when dealing with cross-organizational issues. All participants must use the same set of modeling methods that fosters top-down analysis and implementation. Otherwise, it becomes a bridge built from two sides of the river that never meets in the middle. 

There's no real technical value to undertaking an EAI implementation unless there's a clear business problem that's addressed.

About the Author



Vladimir Orovic is managing director with Insurance Technology Group, a company specializing in research and advisory services for the insurance industry.

He has more than 15 years of IT experience and has held senior management positions with large consulting organizations, focusing on enterprise business process and technology integration strategies for Fortune 500 organizations. Voice: 416-214-3445, e-Mail: vladimir.oroovic@insurancetg.com; Website: www.insurancetg.com.