Communications of the Association for Information Systems

Volume 8 Article 31

6-24-2002

New Developments in Practice II: Enterprise Application Integration

James D. McKeen
Queen's School of Business, Queen's University, jmckeen@business.queensu.ca

Heather A. Smith
Queen's School of Business, Queen's University, hsmith@business.queensu.ca

Recommended Citation
McKeen, James D. and Smith, Heather A. (2002) "New Developments in Practice II: Enterprise Application Integration," *Communications of the Association for Information Systems*: Vol. 8 , Article 31.
DOI: 10.17705/1CAIS.00831
Available at: https://aisel.aisnet.org/cais/vol8/iss1/31

Communications of the Association for Information Systems

NEW DEVELOPMENTS IN PRACTICE II: ENTERPRISE APPLICATION INTEGRATION

James D. McKeen
Heather A. Smith
School of Business
Queen's University
jmckeen@business.queensu.ca

ABSTRACT

The term enterprise application integration (EAI) refers to the plans, methods, and tools aimed at modernizing, consolidating, integrating and coordinating the computer applications within an enterprise. The need to integrate across applications is being driven by customer demand for access to information and the desire of the business for a single point of contact with their customer base. The challenges are significant because of the variety of technologies in need of integration and because integration cuts across lines of business. This paper distinguishes among four different (but related) targets of EAI:

- Data-level integration
- Application-level integration
- Process-level integration
- Inter-organizational-level integration

The paper then discusses the technologies that assist with this integration (the "EAI toolkit") under the following categories:

- Asynchronous event/message transport
- Transformation engines
- Integration brokers
- Business process management frameworks

The paper concludes by outlining six key strategies for managing EAI suggested by a group of senior IT managers from leading-edge firms.

KEYWORDS: application integration, legacy systems, middleware, integration software, messaging software.

EDITOR'S NOTE: This article is the second in a series of articles on new developments in practice coordinated by James McKeen of Queen's University. The present article was originally prepared by the authors for discussion by the IT Management Forum, a group of senior IT managers from 14 Canadian firms that meets regularly to examine advances in the state of the art. The first article in this series (Volume 7, Article 13, July 2001) dealt with Risk Management in Information Systems. Additional articles in this series will appear in CAIS from time to time

I. INTRODUCTION

Enterprise application integration (EAI) – the plans, methods, and tools aimed at modernizing, consolidating, and coordinating the computer applications within an enterprise – is suddenly a hot topic. Given that the need to integrate across applications is an age-old challenge, why is EAI suddenly in the spotlight? For the answer, you need only to look at yourself – and your behavior and expectations – in your role as a customer of any company. Your relationship with your bank, for example, is most likely via a browser where you expect to be provided access to the full range of banking services conveniently integrated on a single screen ("mybank.com") allowing you to query the status of your checking/savings/investment accounts, reconfigure your mortgage, buy/sell stocks/bonds/funds directly, transfer money to other accounts (not necessarily at your branch or even your bank), enact payments (automatically and/or electronically), take advantage of bill consolidation and presentation, and expect these transactions to be done instantaneously. That the transactions cross multiple business lines, require coordination among many applications/databases resident on different technology platforms with different architectures, and must be done perfectly (consider how delighted you would be as a customer if your bank reported your account balances correctly only most of the time!) is of very little concern to you. As a customer, you have come to expect this level of service. There is little doubt that the impetus behind EAI is the business need to respond to customer demand.

EAI, however, is not an easy problem to solve. Perhaps that is why it has been an ongoing, continuous struggle. Consider the following analogy:

> *"Imagine if you didn't have common electric outlets and plugs in your house and every time you bought a new appliance, you had to wire up the appliance to the wires in your wall. And everybody's wires in everybody's walls were different. And everybody's appliance wiring was different. That's really the way it works today with trying to integrate business software applications"* (Koch, 1996)[1] .

Unfortunately, this analogy is not so far-fetched. It is common in most organizations to have multiple applications (custom, legacy, and packaged), multiple platforms, multiple databases, multiple transaction processors, multiple data entry points, multiple versions of the same data, and incompatible business data. This state evolved over time as waves of new technology swept over the landscape. Different groups, operating independently of one another, built application systems at different times. Early programs in areas such as inventory control, human resources, sales automation, and management were designed to run independently, with no interaction among the systems. They were custom-built in the technology of the day for a specific need and were often proprietary systems. As a result, organizations are stuck with incompatible architectures and with hard to maintain (but even harder to eliminate) legacy applications. In addition, starting in the 1970's, organizations embraced a "buy before build" strategy that favors purchased application packages over internal development – a practice that is vulnerable to the proliferation of different standards. The problem is severe. One focus group company (see next paragraph) discovered that 70% of their code consists of interfaces, protocols, and other procedures to link applications. As a result, they spend 30-40% of their development time building interfaces rather than increasing functionality.

To explore how organizations handle these enterprise integration issues, the authors invited a number of senior IT managers from several different organizations to share experiences and best practices. These managers are referred to in this paper as the "focus group" and their firms as "member organizations." Their insights are combined in this paper with research from the literature to present an overview of the key issues.

---

[1] The quote is by Paul Margolis, past Chairman of the Open Applications Group, a consortium of major ERP vendors formed to create open interface standards so that all the member companies products can talk to one another. Quoted in Koch [1996]

Section II describes the integration problem and its genesis. Section III describes the four levels of integration where EAI can be focused. Following that, Section IV presents and categorizes EAI technology. The final section outlines a number of strategies for managing the EAI effort. These strategies are based on the discussions that emerged during the senior manager meeting.

II. UNDERSTANDING EAI

The driving factor behind the push to achieve EAI is the need to redirect systems. [Bove 2002]

> *"… In the 1950s and early 1960s, systems were used to reduce costs and headcount by automating rote tasks as part of a predominantly static and structured approach to corporate growth. During the 1980s and '90s, business practices shifted toward the concepts of stability, repeatability, return on long-term investments and economies of scale … In today's' digital economy, these dynamics are passé. Existing and emerging Internet technology enables – and capital markets reward – innovation, decisiveness, rapid response and (over the long haul) excellence in execution. The ability of companies to completely infuse and develop these Internet-ready characteristics into information systems used by all their employees – across the board and in every department, from the loading dock to marketing and sales and to customer service – will separate the winners from the losers."*

What is needed are:

- 24x7 availability,
- instantaneous scalability,
- personalized easy-to-use self-service systems, and
- the fast and unerring reliability in transaction processing that the customer demands.

These goals can only be achieved when applications within the organization work together to route and transform information in response to service requests originating from customers, suppliers and/or employees. This redirection of systems is possible only by integrating the disparate, disconnected applications that are used within organizations to automate business processes.

But how did we get to the point of having disparate, disconnected applications? Was this the result of poor technology choices? Was it a planning failure? Was it a lack of standards? Doesn't EAI represent just another layer of software that in time will contribute to the problem? And, while we are on the topic, whose fault is it anyway? According to the focus group (defined in Section I), business managers often ask these very questions. Finding answers to them is necessary to gain an understanding of the problems associated with integrating business applications within an enterprise. What is readily apparent is that the "problem" did not appear suddenly. It has been in existence since the start of information technology. It just worsens in direct proportion to the number and size of business applications. And, with IT enabling (if not driving) virtually all organizational initiatives today, the application portfolios in most firms grew substantially over the years, sharply increasing the need for (and challenge posed by) integration. It should be pointed out that EAI exists because it:

- embraces the diversity ("heterogeneity") that will always be part of businesses,
- considers IT an inevitable part of large-scale systems design and development, and
- serves as the connection/broker/translator linking autonomously designed applications into a cohesive whole.

Every year organizations launch new applications. New technology (hardware, software, methodology) replaces old technology. New standards replace old standards. It is impossible to

upgrade all applications continually. In fact, it would be imprudent to attempt to do so. As a result, the typical applications portfolio consists of a "mixed vintage" concoction of technology. Superimposed on this situation is the imperative that businesses must continually change to meet the evolving needs of customers, suppliers, and employees. To do so sometimes requires new applications but more often requires integration across existing databases, applications, technology platforms, and lines of business. Applications are almost never autonomous; data and transactions are constantly transferred among different systems within an enterprise and outside to its trading partners. The application integration challenge, arguably the most critical challenge facing IT today, will continue into the foreseeable future. The bright side is that, for those organizations that manage to integrate applications effectively, financial rewards will result from their ability to leverage their investment in information technology effectively.

Standards. Standards play an important role in application integration. Without standards, the ability to integrate across applications is reduced to unique handcrafted solutions. With standards in place (and rigorously imposed), the costs of integration in terms of effort expended and time-to-market are reduced drastically. The adoption of standards (in programming techniques, languages, hardware, software, or methodologies) is therefore crucial to the ability to integrate applications, but it does not obviate the need for application integration. The real integration problem facing IT managers derives less from the standardization of communications than from the volume of communications. Figure 1 depicts a typical (and much simplified) application portfolio. As the number of applications, n, increases, the number of possible interconnections increases as (n-1)*(n-2)/2, which is of the order of n^2. Until recently, these inter-application connections were handled on a one-to-one basis, often hard-coded within applications. The result drove up the level of complexity within the application portfolio to the point that changes to a single application could wreak unforeseen havoc on other applications connected through a vast web of interrelationships.

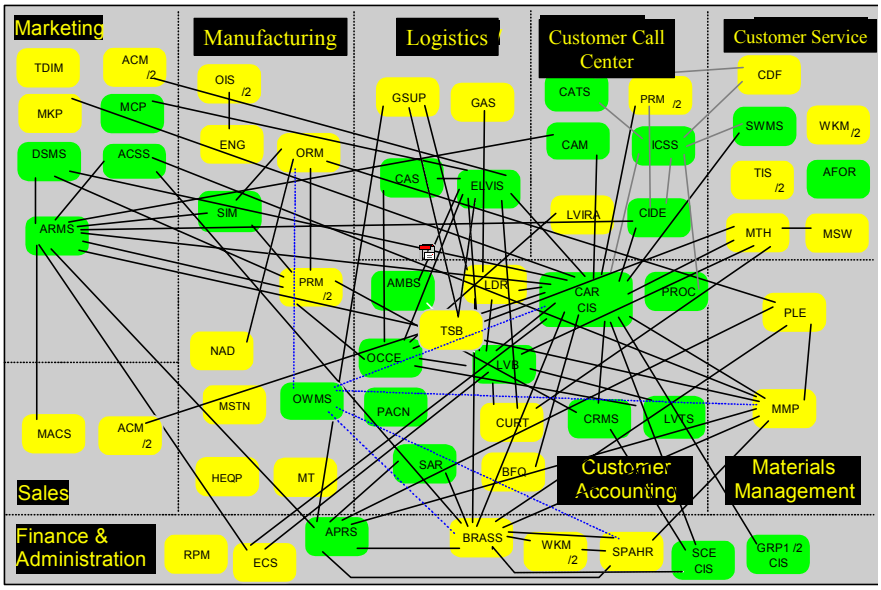

Figure 1. Typical Interactions Among Business Applications

Note that Figure 1 illustrates the number of interconnections. The meaning of the acronyms is not important for understanding the figure.

Theoretically, the solution to this complexity rests with componentization, decoupling, and standardized interfaces. Application developers have known (and deployed) these techniques for years within applications. It is now accepted practice to invoke standardized routines/objects, calls to databases, and GUI front-ends from within applications. With EAI, the same strategy is

applied, but at a different level. Some of the process (and translation, workflow, and communication) logic is removed from individual applications and reconstituted centrally. Centralization effectively decouples individual applications and orchestrates communication through legitimate channels. This difference is shown as a “pre-EAI” picture in Figure 2 and a “post-EAI” picture in Figure 3 below.

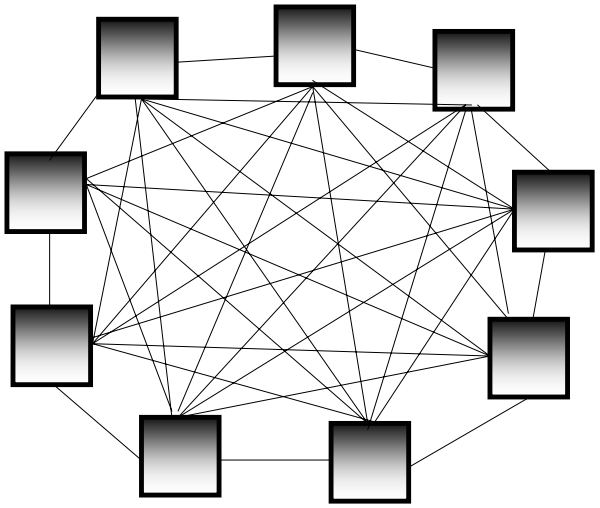

Figure 2. Pre-EAI Communication among Applications

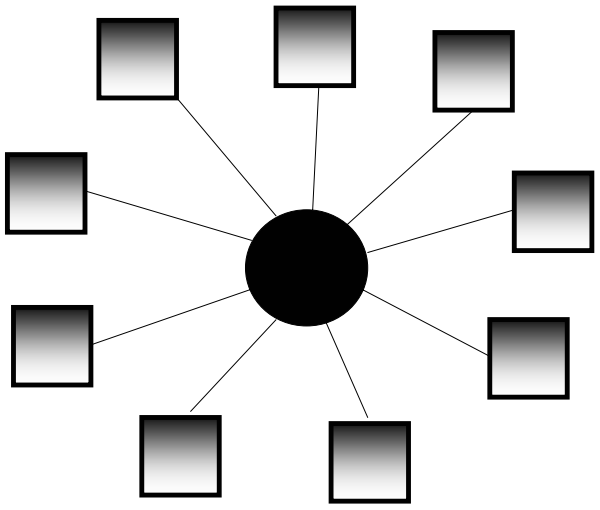

Figure 3. Post-EAI Communication among Applications

It is evident that new software (called middleware, discussed in detail in Section III) plays a key role in the struggle to integrate across existing applications. As with any software product, middleware too ages and becomes “legacy”. The question is legitimately raised whether this additional layer of middleware software is really just paving over the problem and not eliminating it (or even fixing it) and, in time, will become part of the problem rather than the solution. The response by the focus group members was unanimous.

First, they quickly acknowledged that today’s software is tomorrow’s legacy. They were little interested, however, in the search for universal and timeless solutions.

Second, they pointed out that the middleware software offerings comprise some very powerful tools, which enable the integration of a number of otherwise disparate systems with relative ease. As a result, the return on this investment is both substantial and evident.

Third, the middleware itself imposes development standards to which applications must adhere if integration is to be realized.

Fourth, middleware attacks complexity directly by offering tools to manage inter-application communication (and translation and workflow) by decoupling individual applications.

Finally, they argued that, because application integration presents a significant challenge for virtually every organization, the incentive is enormous for software vendors to continue to develop products to address integration. GartnerGroup [see McCoy and Thompson, 2001] estimates the worldwide market for middleware in 2005 will be $11.1 Billion (USD). As a result, middleware tools, as good as they are today, will only improve.

III. THE FOUR TARGETS OF EAI

An analysis of the four possible targets of integration:

- data,
- application,
- process
- Inter-organizational

is useful for managers to help them understand their greatest needs and subsequently to focus better on their EAI initiatives. Experience shows that comprehensive EAI strategies need to focus on more than a single target. For example, application integration without data integration would provide only part of the solution; application integration without process integration would fall far short of the goals of EAI. With full recognition of the interrelationships among these different foci, vendors are beginning to develop tools that span these areas. The categorization that follows, partially based on work by Linthicum [2001], outlines each of the four main focal points for EAI

DATA-LEVEL INTEGRATION

Historically, applications were designed with their own unique data structures making it difficult to share data with other applications (e.g., purchasing uses different product codes than inventory). This problem was solved in the past by hard-coding data format translation programs or by writing file transfer programs to replicate and reconcile data from each application's database. The result was a single "logical" database that all the integrated applications could access. Data warehousing tools soon appeared that facilitated replication (the moving of data between two or more databases while honoring schematic/model differences) and federation (the integration of multiple databases and database models into a single unified view of the databases).

Today, advanced data integration products not only perform required transformations and normalization of data from different applications, but can route and distribute data dynamically based on a set of pre-configured rules. Data-level integration is not just limited to data and databases but also includes distributed objects (i.e., the combination of data, logic and communications within a single entity). By means of a distributed architecture, objects can be combined to execute whole business functions producing an elegant (some would argue the ultimate) integration approach.

APPLICATION-LEVEL INTEGRATION

Beyond data-level integration, diverse applications need to be linked to accomplish specific business processes (e.g., opening a new account might involve credit checking, billing, work processing, and more). In the 1980s, application integration was achieved by the two-tier, client-server computing paradigm that amalgamated the user interfaces of multiple server applications into a single client-based interface, allowing users to interact with the multiple applications from a single screen. Another approach was simply to write hard-coded point-to-point interfaces that allowed the business logic in one application to call the business logic of another

application as if that logic were part of the calling application. This approach came to be known as A2A integration.

Today, A2A integration is facilitated by infrastructure software and application adapters. The most popular type of infrastructure software is referred to as messaging-oriented middleware (MOM). These message brokers transport information ("messages") between applications by identifying, transforming, and routing messages to the appropriate applications on an event-driven, asynchronous basis. They tend to be focused on the back-end operations and process-oriented applications within firms. As organizations attempt to meet the online, 24x7 requirements of Internet customers and partners by adopting zero-latency, straight-through processing, MOM unfortunately often proves inadequate [Bove, 2001]. Instead, enhanced capability is required to manage distribution, backup, load balancing and system capacity to deal effectively with inter-application communication.

Application-specific adapters enable the conversion between different types of applications based on different technologies. Inter-vendor connections (e.g., linking a SAP R/3 application to a Baan application) are relatively straightforward with available adapters. Their advantage is to allow application independence and transparency while facilitating integration. Another integration tool – screen scrapers – allow input data captured in one application to be shared with other applications. In effect, screen scrapers connect many custom or packaged applications. Many of these tools are capable of accounting for differences among schema, content, and application semantics by translating the information moving between the systems in real time.

PROCESS-LEVEL INTEGRATION

The next level of integration is to coordinate the flow of logic among the integrated applications.

> "*This is often achieved by writing a new program that calls the business logic of the integrated applications through their adapters. When this new application is executed, it calls each of the integrated applications in a sequence that corresponds with the flow of an enterprise-wide business process" [Bove, 2001] .*

Some authors refer to process-level integration as "event-oriented" or "transaction-oriented" integration where transactions/events provide the linkage among various applications. Examples of tools used to effect transaction integration include transaction processing (TP) monitors and application servers. This approach enables organizations to create common methods (e.g., transactions) and share those methods among many connected applications.

Recognizing a universal need, software vendors developed a new set of products to address process-level integration specifically. These products – business process automation products and application servers – are based on workflow technology. Application servers access the business logic exposed by adapters to be tied together into a cohesive, end-to-end transaction flow. They are focused on application development (i.e., the front-end) and are particularly good at supporting portal-oriented integration. The business process automation products provide useful management capabilities including tools for IT architects to model the flow of business logic using point-and-click techniques (rather than programming) to create a new enterprise-wide business process from the logic contained in a set of integrated applications [Bove, 2001].

INTER-ORGANIZATIONAL-LEVEL INTEGRATION

Process-level integration typically focuses on bringing together processes within an organization to obtain maximum value while supporting the flow of information and logic among these processes. To do so, a set of easily defined and centrally managed processes is layered on top of existing processes within a set of enterprise applications. The next step is to link processes beyond the organization to include trading partners – both suppliers and customers. Many organizations are already involved with B2B initiatives that share/combine business logic. While sharing business logic is a simple extension of process-level integration, it represents another

level of coordination, negotiation, and complexity and is therefore fraught with management challenges.

> *"The ultimate goal is to bind all trading community systems together in such a way that any application can access any method or any piece of data without delay to support any business process" [Linthicum, 2001].*

IV. THE EAI TOOLKIT

Like any clichéd buzzword, the term EAI now means many different things to many different people. Nowhere is this diversity more apparent than within vendor software offerings – typically referred to as "middleware".

EAI middleware is runtime system software that directly enables interactions among independently designed applications in a distributed computing environment [McCoy and Thompson, 2001]. For the sake of common definitions, the EAI market space as a whole is separated into the following primary categories of software services/products. This categorization is taken from Allen [2001] who notes that products in all these classes of software also include a variety of adapters that provide connectivity to leading applications, databases and middleware. It should be noted that data modeling technology and data warehousing are vital parts of an EAI toolkit particularly for the data-integration level. They are not discussed here as they are beyond the scope of this paper.

- Asynchronous Event/Message Transport: Typically referred to as message-oriented middleware (MOM), these products enable asynchronous routing of business events between applications. That is, they can defer delivery of information about business events until applications are available. This capability facilitates loosely coupled relationships among applications – a fundamental design principle of EAI solutions.

- Transformation Engines: These tools convert data and business events from one format to another. For example, a transformation engine could convert customer records from an ERP system into formats required by a "home-grown" customer service application. These engines are typically batch-oriented and operate at the database table or file level.

- Integration Brokers: Also known as message brokers, these tools provide the ability to route and manipulate business events intelligently between multiple applications and data stores. For example, an integration broker could receive order requests from a Web-based application and route those requests to one or more target applications based on information in the order. The source event would be transformed into the format expected by the destination applications. Therefore, by definition, integration brokers include data transformation services. Integration brokers support an event paradigm and deal with individual records, rows, or autonomous business events.

- Business Process Management Frameworks: These tools, which enable business logic to be separated from process flow logic, consist of two distinct categories: process automation and workflow. Process automation products provide a framework that allows multiple disparate software components to participate in an integrated business process flow. Workflow products support a similar paradigm but focus on process steps performed by human interactions with the system. As business events move through a process flow, they require routing and transformation services. Therefore, by definition, process automation/workflow frameworks also include either their own integration broker services or the ability to use the services of an external integration broker. Most of these products also contain management tools that monitor application processing and assist in identifying bottlenecks.

It should be noted that the ability of EAI tools to manage business processes is as much a business issue as a technology issue. According to one focus group member,

> *"EAI software can help to automate task delivery/process control. This being the case, process and business knowledge AND business influence is critical to the success of efforts that leverage EAI process capabilities. Choosing not to leverage process capabilities (like business ware) could point to the fact that organizations are not exploiting the full functionality/potential of EAI technology".*

V. MANAGEMENT STRATEGIES FOR EAI

The following strategies for managing EAI were suggested by focus group members based on their experiences. All strategies are designed to be "near-term" – that is, they can all be started immediately with the expectation of yielding results within months.

1. CRAFT A CORPORATE INTEGRATION STRATEGY

Enterprise application integration does/will not happen by itself. Nor does it stand much of a chance if left up to one or two courageous individuals within the firm. Focus group members suggested that someone within IT must first recognize the need for a concerted effort to approach EAI. They also suggested that this individual often resides within the IT architecture group (although not necessarily so). Once identified, this individual becomes the EAI champion. His/her first task is to gain the support of the senior IT executive team for the EAI initiative to begin. The next order of business is to craft an EAI strategy. According to focus group members, such a strategy has seven key steps performed in roughly the same order as they appear in the sidebar.

Sidebar 1
Seven Key Steps to Develop a Corporate Integration Strategy

1. Target strategic applications
2. Become an EAI expert
3. Identify "status quo" costs
4. Build the business case for EAI
5. Estimate resource requirements
6. Create a plan
7. Sell EAI to senior management

Step One – Target Strategic Applications

Identify the applications that require integration. Of these, decide which are strategic. For the strategic applications, conduct interviews with IT leaders and determine the specific integration needs for these applications. Non-strategic applications will benefit from EAI efforts eventually.

Step Two – Become An EAI Expert

Based on an analysis of these applications, identify the most appropriate levels to focus the integration effort and EAI toolkit best suited to meeting those needs. This step requires the champion to learn about EAI vendors, products, services, and the experiences of other organizations already investing in EAI. Someone has to become the in-house EAI expert.

Step Three – Identify The "Status Quo" Costs

Based on these strategic applications, conduct scenario planning to assess the costs of the "status quo" ("let's keep on building applications the same way") approach. These costs should represent rough estimates of additional development, maintenance and time-to-market costs associated with this approach.

Step Four – Build The Business Case For EAI

Again, based on these strategic applications, map out the advantages of the planned EAI toolkit. Focus group members pointed out that vendors can (and should) play a role in this exercise. Be specific in terms of the exact functionality that EAI products can provide and express these deliverables in business terms. Omit the "this will increase our productivity" arguments. Instead, base the argument on such ideas as "this will enable our customer service reps to query the status of customers' orders, payments and delivery schedules in real time".

Step Five – Estimate Resources

An EAI initiative will require people, tools, new skills, education, and procedural and structural changes (more on this in item 4) within IT. First, identify the individuals who will be directly involved with the EAI implementation effort. Then estimate the acquisition costs for the EAI toolkit (see item 2), training costs and effort needed to bring everyone in IT up to speed working with new EAI tools, the necessary architectural changes to the technical platforms and the system installation and configuration.

Step Six – Create a Plan

Identify the necessary tasks to integrate the previously identified strategic applications. Combine these tasks into a workable plan specifying the logical order of the work, a realistic timeframe for these changes to be achieved, and a preliminary benefits delivery schedule.

Step Seven – Sell the EAI Strategy to Management

An EAI strategy will require senior IT management's full endorsement. As with other major initiatives within IT, EAI must be presented convincingly to senior management. This presentation should be based on a thoroughly prepared business case – deploying the standard pro-forma business case adopted by your organization. As would be expected, business cases are typically derived from the ability to integrate disparate business applications to provide new functionality. As one focus group member pointed out, EAI can also "accelerate the business by reducing the time it takes to deliver meaningful functionality to the business". He claimed that, in his company, EAI software release cycles approach (if they are well planned) the same speed with which web solutions are deployed (i.e., 3-4 month increments). So, time-to-market should definitely be a part of the business case presented to senior management.

2. ASSEMBLE THE EAI TOOLKIT

The assembly of an EAI toolkit requires a deliberate strategy. It begins with an analysis of the firm's strategic applications to suggest the key focus of the integration effort (i.e., data, application, process, or inter-organizational). This analysis, in turn, suggests the most appropriate categories of EAI tools. This understanding enables you to categorize the various EAI software vendors and helps you to discuss their respective product offerings more intelligently.

> *"Understanding the various categories of EAI services will help you refine the EAI requirements specific to your enterprise, which will keep you from buying the whole house when all you need is a kitchen sink"* [Allen, 2001].

Allen [2001] provides an excellent check list for dealing with vendors complete with questions to ask and tests to be undertaken covering vendor negotiations from needs determination through vendor "bake-offs" to implementation. Focus group members agreed that no single vendor provides a complete package. Managers are left to select the best offering and augment it with other products/services. They are put in the "general contractor" mode. The group

felt that the following criteria [McLean Report, 2002] provide a good starting point for evaluating EAI systems.

- Internal and/or external integration. Depending on specific needs, some EAI projects need to integrate internal systems, while others require integration with customer or supplier systems.

- Business process management. EAI software must support this feature to manage changes to business processes when they happen (which can be very frequent). Without this functionality, it will take a long time to re-map databases and data flows to applications when business processes change.

- Security. Security is especially important when dealing with entities outside your own organization. Data encryption and user authentication (and program authentication for automatic data exchanges) are both necessary features.

- Ease of use. This criterion is especially important if your business users will be maintaining business process or workflow information and must use the EAI toolkit frequently.

- Technology management functionality. No matter how good the rest of the package is, the application will be useless if you can't tell quickly if all available network bandwidth is being used or the application is in an endless loop that will fill up all available storage. The EAI toolkit must tell, in detail, what it is doing with the infrastructure and notify you when there are problems.

3. DEPLOY "HUB AND SPOKE" DESIGN

Figure 3 in Section II depicts a classic "hub and spoke" design. The three organizing principles with this design are:

- Don't connect anything directly to anything.
- Design applications to be autonomous and don't allow them to share databases directly.
- All knowledge of interconnections is removed from the source (and target) and placed within the hub.

A number of the focus members indicated that they had adopted the hub and spoke architectural model. They described its operation as follows.

Applications are empowered to create messages encapsulating additions (and/or changes and/or deletions) to their business objects. A broker then routes and distributes these messages to the various integrated applications. At this point, the broker transforms the data into the appropriate representation for the destination. It is the broker that contains the logic to assist in the execution of the business process workflow.

Adopting hub and spoke architecture greatly reduces the complexity of one-to-one integration by organizing all the communication, transformation, and process workflow within the hub where it can be managed effectively. This architecture achieves operational simplification and facilitates change. Individual applications can be replaced with relative ease since much of their logic, communication, and translation functions were removed. The final advantage of this architecture is its ability to leverage reusability substantially.

4. CREATE AN INTEGRATION CORE COMPETENCY TEAM

The adoption of hub and spoke architecture presents new challenges for application development. Because many of the functions normally resident within applications are now physically removed to the hub, application development is affected significantly. One focus group

member suggested that application developers must undergo a mindset change from that of "developer" to that of "integrator".

Within this new architecture, the hub is recognized not just as a sophisticated messenger but as a full-functioning organizational asset complete with hardware and software. As such, it must be scaled and must operate within a "release-controlled" environment much the same as other computing environments. Two member organizations created central groups (called integration core competency teams) for each hub. These groups manage both the business functions and the software and hardware associated with the hub. They take responsibility for the following activities:

- maintaining all integration documentation
- performing (and assuming ownership for) the detailed design of each interface
- constructing the middle pieces of the interface
- establishing best practices
- performing broker marketplace evaluations, and
- administering all middleware software products.

Each hub should have its own team. In addition, a small team of architects should be appointed to manage across hubs. One member organization appointed an enterprise IT Architectural Council responsible for developing standards, preferred architecture(s), and policies to govern their implementation and use. These standards and policies are then pushed down to the system architects and deployed on an enterprise-wide basis. In this organization, it is mandatory for all application development teams to use "hub" people for all interfaces. Also, they placed responsibility for the delivery of all cross-platform projects with the hub people. In essence, the hub people are the managers for these projects.

The adoption of a different architectural design also affects the structure of the IT department. In a way, the structure of IT mirrors the technology. As soon as the hub and spoke architecture model is adopted, we see differences being drawn between "hub" people (with interface and integration responsibilities) and "spoke" people (with more traditional application development responsibilities). There are good reasons for this division of responsibility:

1. The role of application development is changing from that of "developer" to that of "integrator".
2. EAI tools are special-purpose tools and IT professionals need to be trained in their use.
3. Hub people need to develop "process" skills.

In one member organization, they created "business process" analysts to reflect these skills. In 2002, individuals with "hub" skills are few in number as IT organizations did not commit significant staff to these roles. As the deployment of EAI spreads, this shortage is likely to disappear.

REINTEGRATE LEGACY APPLICATIONS

A key part of enterprise application integration involves legacy systems for no better reason than the fact that legacy systems predominate. With legacy systems, anything less than wholesale replacement involves deconstruction – dismembering the application into its three main components/layers of:

- presentation (the user interface),
- business logic (the rule-based reasoning) and
- communications (the data and inter-application linkage).

Within typical legacy systems, these three components are inextricably bound together, making it complex, time-consuming, and even impossible to separate and/or distribute the layers

for integration purposes. Decoupling each of these components/layers enables the application to be reconfigured into a series of common, sharable tasks that paves the way towards integration. When re-integrating these legacy applications, there is a continuum of approaches based on the degree of application invasiveness. Dubbed the "five R's", these strategies are:

- refacing,
- repurposing,
- restructuring,
- re-engineering and
- replacing [Anonymous, 2001].

Refacing: Standard terminal emulation screens are replaced with a graphical user interface (GUI) for each application. The approach is non-invasive and the underlying legacy application is not modified. Overall functionality is not altered but, due to the refacing, integration among enterprise applications is facilitated by the use of common front-ends.

Repurposing: Like refacing, repurposing uses a GUI for the presentation layer. Unlike refacing, repurposing integrates the screen-based business logic of many legacy applications and allows changes to the workflow of legacy applications to facilitate the underlying business processes. Even though the underlying applications remain unchanged, repurposing can build in new business logic for various purposes such as updating data sources, performing calculations, triggering events, and otherwise automating tasks that can result in significant process streamlining and enhanced productivity.

Restructuring: This process separates the presentation and business logic components within an application. Once separated, these components can be wrapped with new interfaces and integrated into any number of new refacing or repurposing applications. Restructuring allows an organization to leverage an application procedure whose logic is proven to work while preserving data integrity in the communication layer. Only through restructuring can the desired application procedure be invoked directly

Re-engineering: Reengineering entails rebuilding the entire application. Built to exact specifications, this approach provides the organization with a chance to build in the integration necessary for it to meet its business and technology needs.

Replacing: Sometimes it is best to procure an off-the-shelf solution to replace the host application. This approach allows the organization to choose what operating platform the application will run on, as opposed to being bound by the original.

The first two approaches are non-invasive, requiring no modification to host applications and hence provide the quickest remedy. For this reason, and if possible, refacing and repurposing are preferred solutions. By contrast, the last three approaches are invasive, requiring modification and even replacement of host applications. As can be seen from Figure 4, each of the approaches requires greater time to implement and greater costs. What is not apparent from Figure 4 is that restructuring, re-engineering, and replacing often provide benefits not possible with refacing for example. Decisions regarding the integration of legacy systems should be based on a cost-benefit analysis in alignment with the organization's overall integration strategy. One-off decisions about individual legacy applications provide short-term solutions at best. Applications should be mapped onto an integration strategy as part of overall application portfolio management.

NEXT STEPS: THE "COLLABORATIVE ENTERPRISE"

Technology is enabling organizations to become intricately connected in many ways. By linking systems with trading partners, an organization can check a supplier's inventory directly, check the status of in-process orders, preview pricing structures, choose delivery options, and

interface ordering systems directly with a supplier's fulfillment systems. As these "electronic fingers" bond organizations, they create significant value by enabling the "instantaneous"

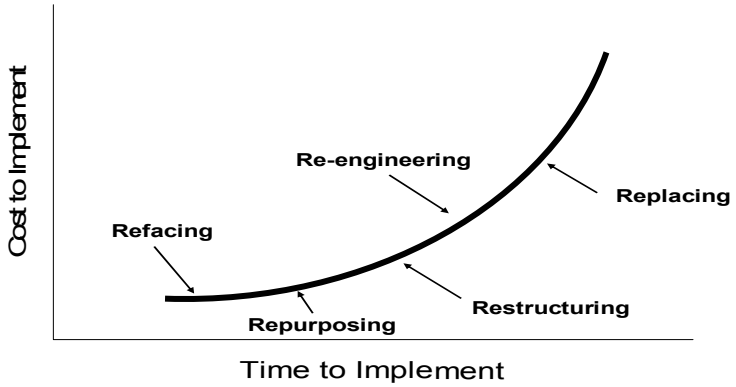

Figure 4. Cost/Time Tradeoffs for "5 R"s

movement of data, products and services thus eliminating the "float" between an action (e.g., the placement of an order) and its realization (e.g., the processing of that order). To do so requires collaboration between trading partners, hence the term "collaborative enterprise". Customers are demanding it, competitors are doing it, the business needs it, executives are asking for it, and the burden of delivery rests on IT's shoulders.

The technological challenge presented by the collaborative enterprise is enormous. It requires the integration of application systems across firms, thus elevating the integration challenge to a new level. Even within organizations, where they control the technology, integration presents a daunting task. Beyond their boundaries, organizations have little control over the technology of partner organizations. Furthermore, current software solutions deployed on the edges of enterprises are expensive, do not scale, and are complex to implement across enterprises [Donato et al, 2001]. Donato et al. argue that:

- inter-organization integration will require a new technology architecture that makes integration much less difficult, less expensive and less time-consuming.
- the appropriate technology is "web services" – an emerging technology architecture that could make integration as easy as plugging an appliance into the electrical grid. Web services is based on the notion of building applications by discovering and collaborating with network-available services – the just–in-time integration of applications.

If web services is successful, collaborating enterprises will be able to plug applications and business processes into a service grid that is ubiquitously accessible and affordable to most companies. Web services are designed to enable a collaborative environment that is decentralized, dynamic, and diverse. This emerging architecture will enable companies to reap unprecedented productivity rewards from a more focused and integrated set of business processes and partners. Indeed, machine-to-machine execution of loosely coupled business processes by enterprises and their partners will touch every company, customer and employee. It will redefine enterprises, and even industries, by facilitating agility [Donato, 2001].

While it is difficult to know when web services will become a reality, it is easy to see the transitions that would be required. Web services would revolutionize the definition of "enterprise" architecture, changing the enterprise from a self-contained set of applications into a nexus of services, data flows, and business process interfaces shared by other enterprises. Web services would create a market economy for services operating much like an electricity grid, where

organizations draw from (or supply to) the grid [Hagel and Brown, 2001]. With such a marketplace, EAI represent the "table stakes". Firms not advanced in terms of EAI would simply not be ready to "plug and play" and would lose opportunities to competitors who were more advanced in EAI development. The advantages of collaborative enterprises are such that it is likely that every effort will be made to make web services a reality.

Regardless, focus group members felt that now is the time to take significant action in terms of integrating enterprise applications. This effort provides immediate benefits to the firm while paving the way towards the web services where integration transcends organizational boundaries. For this reason, organizations should view their EAI efforts in light of the broader goal of enabling the collaborative enterprise – that is, integrating their organization's technology within a web of partner organizations to facilitate the free flow of data and functionality. Specifically, they should do the following:

- Craft an EAI strategy that explicitly recognizes the eventuality of an inter-organizational architecture (e.g., web services).
- Use this strategy to guide internal EAI efforts. For example, opt for industry standards over in-house standards where possible. This choice will increase the chances of being inter-organizationally integrated.
- Join industry associations involved in establishing architectural standards. The "best case" scenario is that this involvement provides an opportunity to influence the selection/adoption of standards; the "worst case scenario" is that you receive early notification of industry–wide decisions regarding the adoption of standards. These options are "no-lose".
- Assess key business processes to identify any/all that might provide value to other organizations (i.e., best-in-class processes). Investigate the potential market for these processes both within and beyond your industry.
- Investigate the market for best-in-class offerings that might complement (or replace) current business processes.
- Pilot web services as early as possible with non-critical processes. When web services become a reality, your firm should be ready to move.

VI. CONCLUSION

The need to integrate applications across the enterprise is at a crucial stage. Due to customer/supplier demands for instant access and 24x7 service, EAI changed from a "nice to have" to a "do or die" situation. This article described the dimensions of the integration problem, outlined the challenges it presents for IT, and (based on the cumulative insights of a number of senior IT professionals from leading-edge firms) presented a number of strategies for dealing with EAI effectively. As with many organizational challenges, EAI can be seen as a problem or as an opportunity. Given the impact that EAI will have on the marketplace, it is arguably best to approach EAI as an opportunity to create enhanced value for both customers and suppliers.

Editor's Note: This article was received on March14, 2002 and was published on June 27, 2002

REFERENCES

EDITOR'S NOTE: The following reference list contains the address of World Wide Web pages. Readers who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that
1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.
4. the authors of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Allen, D. (2001), "Assembling Your EAI Toolkit", http://eai.ebizq.net/str/allen_1.html

Anonymous (2001), "Approaches to Enterprise Application Integration Involving Legacy Host Applications: The Five R's", http://www.attachmate.com/article/0,1012,3163_1_3858.00.html

.

Bove, T. (2001), "EAI: Providing Stability in the Whirlwind of e-Commerce", http://eai.ebizq.net/str/bove_1.html, August

Donato, C., Durchslag, S. and Hagel III, J. (2001) "Web Services: Enabling the Collaborative Enterprise", http://e-serv.ebizq.net/wbs/donato_1a.html

Koch, C. (1996), "The Integration Nightmare: Sounding the Alarm", *CIO Magazine*, November 15. http://www.cio.com/archive/111596_alarm_content.html

McCoy, D. and Thompson, J. (2001), Gartner Dataquest AV-14-0031, June, pp. 1-17.

Hagel, J. III and Brown, J. S. (2001), "Your Next IT Strategy", *Harvard Business Review*, (79)9, October, pp. 105-113.

Linthicum, D. (2001), "Chasing the Tail of the Beast", *Software Magazine*, www.Softwaremag.com , February, pp. 1-9.

McLean Report (2002) "Enterprise Application Integration (EAI): Software Buyer's Guide", January, http://www.mcleanreport.com/current_issue.asp#2

Pinkston, J. (2001), "The Ins and Outs of Integration: How EAI differs from B2B", EAI Journal, August, pp. 48-52.

ABOUT THE AUTHORS

James D. McKeen is Professor of MIS at the School of Business, Queen's University at Kingston, Canada and is the Founding Director of the Queen's Centre for Knowledge-Based Enterprises. He received his Ph.D. in Business Administration from the University of Minnesota. His research interests include IT strategy, user participation, the management of IT, and knowledge management in organizations. His research is published in the *MISQ*, *JITM*, *CAIS*, the *Journal of Systems and Software*, *the International Journal of Management Reviews, Information & Management*, *CACM*, *Computers and Education*, *OMEGA*, *Canadian Journal of Administrative Sciences, JMIS*, *KM Review*, and *Database*. He currently serves on the Editorial Board of the Journal of End User Computing and was the MIS area editor for the *Canadian Journal of Administrative Sciences* for seven years. Jim and Heather Smith are currently writing a book on the topic of IT Strategy.

Heather Smith is Senior Research Associate with Queen's University School of Business, specializing in IT management. A former senior IT manager, she is a founder and co-facilitator (with James D. McKeen) of the IT Management Forum, the CIO Brief, and the KM Forum, which facilitate inter-organizational learning among senior executives, and co-author with James D. McKeen of *Management Challenges in IS: Successful Strategies and Appropriate Action.* She is also a Research Associate with the Lac Carling Conference on E-Government, the Society for Information Management, and Chair of the IT Excellence Awards University Advisory Council. Her research is published in a variety of journals and books including *CAIS, Journal of Information Technology Management, Database, CIO Canada,* and the *CIO Governments Review.*

Communications of the Association for Information Systems

ISSN: 1529-3181