LEVERAGING LEGACY SYSTEMS IN MODERN ARCHITECTURES

by Len Erlikh

The business value locked in legacy systems combined with the huge investment companies have already made in their development create a powerful incentive to leverage these existing systems rather than to re-create them from scratch. It is crucial for these mission-critical systems to be seamlessly integrated into each facet of an organization's technological infrastructure as well as into the new "extended enterprise" created by e-business solutions. The problem is that the majority of these legacy systems were built on top of monolithic, homogeneous platforms. As a result, they don't possess the characteristics necessary to function with distributed, loosely coupled, component-based architectures that comprise modern technology systems.

Although many companies have rapidly and enthusiastically adopted distributed architectures, many more are stuck with mainframe-based, mission-critical systems that continue to isolate them from the systems of their partners, suppliers, and customers. Indeed, International Data Corporation (IDC) estimates there are more than 10,000 large IBM mainframe sites worldwide with 300 billion lines of legacy code still in use. Most of these companies want to transform their applications to meet new business demands, but because legacy systems tend to be unwieldy, monolithic, and inflexible, many such firms regard modernization as somewhere between improbable and impossible. Reeling from the Year/2000 debacle and saddled with years of application backlog, the most they can hope for is to keep the legacy system alive.

However, keeping it alive is getting more expensive. According to an informal industry poll, 85 to 90 percent of the IS budget goes to legacy system operation and maintenance. It's also becoming harder to find qualified personnel to do necessary system maintenance. The majority of U.S. colleges and universities have removed >

COBOL and other legacy language courses from their curricula, thus effectively cutting off the future supply of skilled industry resources. All of this makes it difficult to add new functionality and keep up with business requirements.

CLASSIFICATION OF LEGACY SYSTEMS

The first question any firm should ask is, "Is this system worth the work?" Figure 1 shows the region where legacy transformations make the most sense. Transformation strategies vary according to where in the region a system falls.

TRASH

You should typically replace a low-quality legacy system that offers generic industry solutions with off-the-shelf Enterprise Resource Planning (ERP) packages. Accounting, payroll, and human resources systems are good candidates for ERP replacement.

NURTURE

In contrast, a high-quality legacy system that provides a competitive advantage is worth nurturing unless external business pressures dictate change. Risk management, logistics, and insurance rating systems typically offer the best strategic advantages.

RECYCLE

The best candidates for modernization are systems that are low quality but are mission-critical and provide competitive advantage to an organization. Here the best strategy is to recycle the systems by preserving their embedded business rules and organizing the rules into new workflows.

MODERNIZE

The second best candidates for modernization are high-quality systems with standard functionality. These systems are the most likely candidates for integration with e-business and other modern enterprise solutions, and in most cases should be modernized by extending their use to an Internet platform. Thus, the systems remain on their present platform but are easily accessible through well-defined Application Programming Interfaces (APIs).

THE FOUNDATION OF MODERN IT ARCHITECTURES

The complex nature of today's IT systems requires sound technical architectures that can stand up to the increasingly rigorous demands placed upon them. The harsh reality that many companies are facing is that much of the technology they've spent millions of dollars implementing over the course of decades will not scale up to the high demands of more modern enterprise and e-business systems. Robust, high-throughput, flexible architectures are needed to support the mission-critical nature of such systems. The four key properties of the modern IT architectures are described as follows.

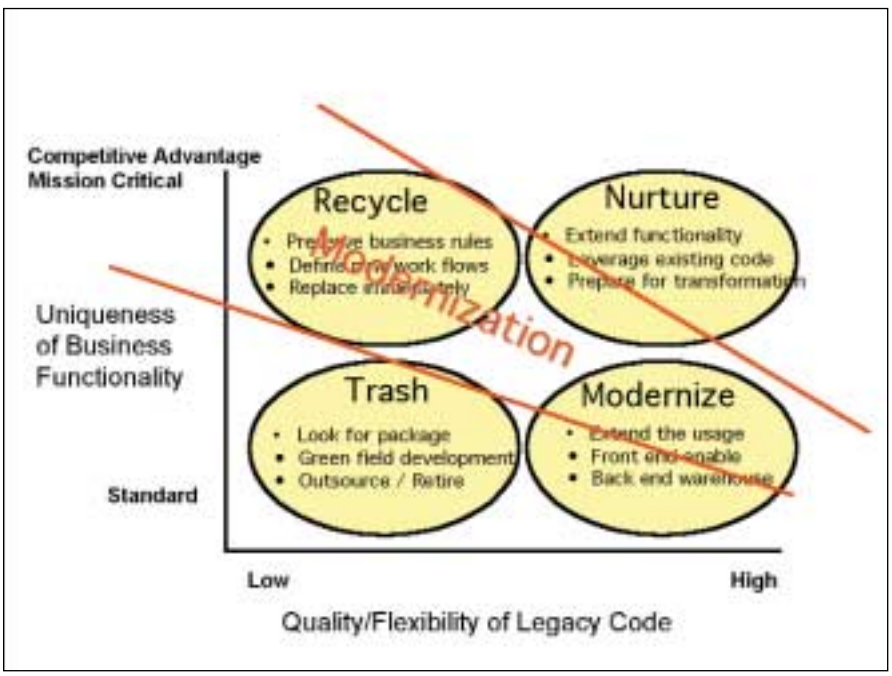

Figure 1 — Asset Extension Alternatives

DISTRIBUTED & HETEROGENEOUS

In order for new business initiatives to be effective, the mission-critical business processes companies have counted on for decades need to be seamlessly linked to all the key systems distributed within an enterprise and even across multiple enterprises. Due to the magnitude and the span of such systems, they invariably incorporate a variety of hardware and software platforms, as well as both newly developed and legacy systems. The resulting solutions are conglomerations of existing islands of automation that will work as a unified whole.

COMPONENT-BASED ENCAPSULATION

Component-based encapsulation is the best way to build such complex systems. This approach allows for more precise management and control over individual business processes by encapsulating and publishing them for use by other components. This divide-and-conquer approach allows for rapid, multi-team, concurrent development so that each large-scale business component turns into a manageably sized, self-contained undertaking.

EAI-BASED COUPLING

Enterprise Application Integration-(EAI-) based coupling allows for further independence and encapsulation of individual system components. Although each business component still publishes a clearly defined set of APIs, the modern approach is to move away from rigid program-to-program connectivity toward more loosely coupled message- or event-based connectivity. A variety of EAI products further de-couple business components by moving systemwide integration into hub-and-spoke or publish-and-subscribe architectures. Finally, Web services and XML-based solutions represent the state-of-the-art in loosely coupled modern architectures.

CONVERGENCE OF INDUSTRY STANDARDS

Industry standards are finally starting to converge into two competing solutions. On one side there is a Microsoft-centric solution with the .NET and COM/DCOM/MTS set of standards supported by C++, Visual Basic (VB), Open DataBase Connectivity (ODBC), and Active Server Pages (ASPs). On the flip side, the Java camp is offering the Java 2 Enterprise Edition (J2EE), Enterprise JavaBeans (EJBs) set of standards supported by Java Data Base Connectivity (JDBC) and JavaServer

Pages (JSPs). The Web services and XML standards transcend both the Microsoft and Java camps, and as a result have the brightest future.

The big problem for many companies is that the mission-critical legacy business processes that need to be integrated with these new-world systems and unified EAI frameworks do not exhibit any of the aforementioned four modern IT architecture properties. These processes are built for homogenous platforms, are monolithic by nature, and do not comply with connectivity standards. Therefore, they fit with modern EAI architectures about as effectively as a square peg in a round hole.

MODELS FOR LEGACY SYSTEMS MODERNIZATION

Legacy systems modernization focuses on the ways in which legacy systems can be integrated into the modern IT architectures. Legacy modernization has evolved over time into a number of distinct approaches. To cater to specific organizational needs, let's examine three approaches, which have to be examined from three perspectives — business benefits, costs, and time-to-market.

Eighty-five to 90 percent of the IS budget goes to legacy system operation and maintenance.

NEW ACCESSIBILITY MODEL

Screen Scraping

Screen scraping creates a Graphical User Interface (GUI) that is a thin layer front-end to text-based legacy screens. Screen scraping is the least expensive and fastest way to extend your legacy system accessibility to the Internet. Companies can achieve initial results in days, sometimes hours, with no modifications or disruptions to the underlying legacy system. Nevertheless, instant gratification has a price.

First, screen-scraped systems can offer only the same functionality as the original system. If you start with a poor, rigid legacy system, you end up with a poor, rigid screen-scraped system.

Second, most legacy systems were developed for highly trained internal end users. Offering the same functionality to an untrained audience can create problems related to difficulty of use, and thus lead to avoidance. The focus of screen scraping is to jazz up the user interface rather then modify core process models or offer business benefits. In other words, with screen scraping it is fundamentally impossible to teach

your legacy systems new tricks. Nevertheless, screen scraping is a viable alternative for organizations merely looking to offer Web-based access to existing business functionality on their current platform.

Legacy Wrapping

Legacy wrapping builds callable APIs around legacy screens and transactions, providing alternative access into existing systems. Legacy wrapping is a permutation of the screen-scraping theme. This approach offers somewhat improved flexibility in managing legacy process flows and eliminates the dependencies on text-based screens as the only points of legacy connectivity. However, legacy wrapping still falls short of true legacy integration, since its non-invasive approach offers no way to fundamentally change the basic, rigid structure of the legacy system.

Legacy wrapping is currently the most popular way to integrate legacy systems into EAI frameworks. Most of the EAI vendors advocate this approach, since it allows them to focus more on the communications/connectivity aspects of the solution and stay away from the sticky issue of truly understanding and restructuring legacy systems for a better fit. However, most of these vendors are beginning to appreciate the need and value of true legacy integration. This realization sometimes comes from understanding the internal problem, but more often it comes from increased customer demand for a complete end-to-end solution.

Case in point is a well-known regional telecommunications provider. This organization was prepared to spend millions of dollars to implement a company-wide EAI solution based on the latest technology in business process automation. It did not take long for the company to figure out that its legacy systems, which were written in COBOL and resided on an IBM mainframe, were crucial to its operations and that the EAI solution it was using provided at best an awkward, bubble-gum-style integration of these systems into the new EAI framework. In the end, the EAI vendor was asked to complete the picture with a viable legacy modernization solution — a solution for which the vendor had to look outside the parameters of its own offerings to more specialized partners.

NEW PLATFORM MODEL

Companies often face "burning platform" issues, such as rapidly vanishing Wang or Unisys mainframes, which spur them to re-deploy legacy systems to a new architecture, usually the Internet. The idea is to keep the legacy system in a platform with the same language (i.e., COBOL to COBOL) or create a carbon copy of the legacy system on a new language platform (such as COBOL to Java or C++). Relative to screen scraping, the new platform approach costs more and takes longer to create, but at least it gets the company off the existing platform and extends accessibility via an Internet-based architecture. Again, however, the new system is only as good as its predecessor. What was suitable on a monolithic, host-centric homogeneous architecture can be totally inappropriate for a new distributed heterogeneous architecture. A 10,000-line COBOL program is commonplace, but a 10,000-line Java program is a nightmare. Legacy migration falls at the opposite end of the legacy modernization spectrum from a benefit vs. cost/time-to-market standpoint. Legacy migration moves the entire function of the legacy system to a new technical architecture — both hardware and software. This approach not only greatly broadens and expands system accessibility, it allows the legacy system's assets to be leveraged on the modern platform and extends the system to support new business models. This last element is extremely important, since typically it is a new business model that drives corporate technology initiatives.

The real value of a legacy system is in its business knowledge, not its technical structure.

The business benefits of re-deploying legacy applications to a modern platform and the potentially significant enhancements to the underlying business functionality cost significantly more than the screen-scraping approach. However, there are certain risks associated with re-deploying an entire application to a new platform. For example, the new platform may be relatively immature compared to the legacy platform (e.g., in terms of robustness, transaction throughput, and security). Depending on a company's specific needs and IT structure, this option may offer the greatest return on investment. For example, the implementation of a new business model, the need to move off of a proprietary platform that has become too difficult or expensive to maintain, or even a sour vendor relationship can be significant factors that support complete legacy migration.

NEW BUSINESS MODEL

The new business model approach truly transforms the legacy application into a collection of reusable business components suitable for a particular target platform.

Legacy componentization is the next generation of legacy modernization approaches. Legacy components dramatically enhance the legacy migration approach. Legacy components are based on the most recent developments in the area of EAI. With this approach, a legacy application is partitioned into a collection of functional components that can subsequently be deployed on multiple platforms of choice — including the existing legacy platform — and are integrated into the overall enterprise architecture using appropriate functional components.

Legacy applications are fundamentally monolithic systems designed and developed for homogenous platforms. The real value of a legacy system is in its business knowledge, not its technical structure. Componentization unlocks and partitions the business knowledge inherent in a legacy system and allows for the re-packaging of these components, distributed on multiple platforms, to build new business applications.

We are now seeing a renewed proliferation of competing technical platforms such as multiple flavors of Unix, Windows/NT, good old mainframes, and aspiring Linux. Each of these platforms offers specific benefits that make it uniquely qualified within an overall technical solution. There has been an increased tendency to include mainframes into the platform mix for e-business initiatives. Mainframes offer robust server environments with high transaction throughput and data integrity.

Legacy components provide for their deployment on any of the popular platforms, including the mainframe. Leaving part of the legacy system on the mainframe reduces the overall cost of the modernization project and exploits a robust, time-proven environment.

Legacy components facilitate true EAI-based coupling, which allows for further isolation and encapsulation of individual systems. Each resulting system has a well-defined, single business purpose and is largely independent in its operations from internal implementation of other systems. Each component publishes a clearly defined set of APIs and facilitates the move away from rigid program-to-program connectivity to-ward more loosely connected message- or event-based approaches. A variety of EAI products further decouple components by moving system wide integration into hub-and-spoke, publish-and-subscribe, or business process automation architectures:

- **Hub-and-Spoke** — works just like a post office where all communications are sent to a central location, which in turns knows where and how to deliver them to the intended recipients.
- **Publish-and-Subscribe** — works just like a radio so that a radio station broadcasts the news on a published frequency and the listeners tune in to catch the latest information from the airwaves.
- **Business Process Automation (BPA)** — is the next generation of the hub-and-spoke architecture where the hub takes an active role in coordinating key business process flows. Finally, XML-based solutions define state-of-the-art in loosely coupled EAI architectures.

EXAMPLES OF LEGACY SYSTEMS MODERNIZATION

A major U.S. healthcare provider turned its mission-critical batch systems for claims processing into a true message-driven Business-to-Business (B2B) solution. The system, which started out as 2 million lines of rigid IBM mainframe batch code with 24-hour response turnaround, was restructured into an elegant EAI solution that deploys modernized legacy components as callable CICS business servers over IBM MQSeries and XML with the response time reduced to only a few seconds.

A major government agency managed to completely move its 30-year-old logistics system away from the mainframe to a distributed application server platform using a combination of the legacy migration and e-componentization approaches. The agency managed to preserve its investment in its core legacy business processes while altering the hard-wired logistics systems into a robust hub-and-spoke EAI architecture with several benefits as enumerated previously. The new system dramatically reduced the change control issues and complexity-driven cost of maintenance that used to plague the old system.

The underlying message from these examples is the willingness of organizations to dig into the large-scale, mission-critical legacy system to mine out the locked business value and to effectively restructure the core business processes into a modern component-based architecture. These organizations needed to go beyond the capabilities offered by non-intrusive screen scraping and wrapping techniques to build long-lasting solutions with a true fit into the modern EAI architectures.

THE UPSHOT

The integration of legacy business processes with modern business initiatives is the key to achieving a truly streamlined, efficient, and cost-effective infrastructure. There is a wealth of business benefits and knowledge buried within most companies' legacy systems that need to be discovered, mined out, and leveraged. True component-based legacy modernization is the key to getting the most value from what already works while enabling technological and strategic evolution.

Companies that rely on their mission-critical legacy business processes for their day-to-day operations need to adopt a cogent, financially viable approach to integrate these vital processes. The organizations that can best leverage these distinctive assets will be the winners in an increasingly competitive and dynamic industry landscape.

About the Author

Len Erlikh is co-founder and chief technical officer of BridgeQuest, Inc. He has 20 years of software industry experience in developing commercially successful software products.

Voice: 718-714-6740
e-Mail: len.erlikh@bridge-quest.com
Website: www.bridge-quest.com