

Leverage, evolution, and flexibility are key

Modernizing Legacy iSeries Systems

BY: MIKE MARLOWE



Mike Marlowe leads the pre-sales team at ASNA as a pre-sales engineer and Windows and Web development trainer specializing in both .NET- and COM-based technologies. Mike is instrumental in educating prospects on ASNA's technology message as well as assisting with conceptual application planning and development. Mike also assists with developing and deploying ASNA's training curriculum. marketing@asna.com

IBM iSeries systems are installed in an estimated 250,000 organizations. If yours is one of them, then most likely your group is either interested in or committed to modernization, which could mean you wish to enhance the user interface while maintaining the existing core business logic, or to integrate data on an AS/400 with Windows-centric, SQL-based back-office applications, or to implement a service-oriented architecture (SOA) and offer Web services to better serve your customers and your enterprise's own competitiveness.

Whatever the level of modernization you are currently seeking, the existence of the iSeries presents both challenges and opportunities. The challenge is to determine the best way to bring the enterprise system into the 21st century without incurring unnecessary disruption to the business, while managing costs and achieving a reasonable timeline. The opportunity is to leverage the valuable asset that is the iSeries itself – including its reliability, performance, ease of use, and familiarity – along with the associated assets in existing data, developer-born applications, and staff expertise.

The options for modernization are many. To evaluate and select among the possibilities, an articulated set of goals and an awareness of the assets to be leveraged and

preserved will be valuable. These goals and assets are interrelated, in the sense that among your goals, it is important to consider the ability to *maximize* iSeries assets. You may consider these assets in terms of the literal building blocks of the system: RPG programs, CL programs, DDS display, print files, data files, and so forth. For a more enterprise-wide perspective, however, also consider your assets in terms of the “four pillars” of an organization's IT “ecosystem.” These are:

1. The *people* (your existing programming staff, including RPG programmers, C# people, VB folks, and so forth, as well as anyone else who interacts with, is affected by, or has expertise in the system AND the company's business)
2. The *applications* (which repre-

sent the sum total of your company's IT existence, and are often complex and comprehensive – but whose value erodes when they are not readily extended to other applications, interfaces, and Web services)

3. The *processes* (which depend on IT technologies for support, and benefit from leaner, integrated methodologies – but these require change and adaptation throughout the organization)
4. The *data* (both on the iSeries and on Windows-based or any other system you may already have)

Each of these categories of asset is important not only to the iSeries system but to the organization as a whole. Each represents an investment that can be either lost, preserved, or enhanced through the modernization initiative. All four merit careful consideration as to the way they may be impacted by, respond to, or contribute to the changes you are considering.

With this insight into your organization, your goals, and your assets, you can begin to consider the modernization options, ideally without becoming mired in the plethora of possibilities.

Java/WebSphere or .NET?

Although this oversimplifies the situation, the looming question at the outset is whether to proceed with a Java/WebSphere solution or to consider an alternative – that is to say, .NET – for your Web/SOA realm. IBM's original iSeries Developer Roadmap was certainly intended to steer iSeries shops toward a Java-centric future. The disadvantages to this route, however, have proven so great that even IBM has backed off from its

insistence on Java/WebSphere and begun to recognize a role for third-party solutions.

For large enterprises that can bear the risk and costs of Java projects, the Java/WebSphere route may make sense. For small and midrange companies (which make up the majority of OS/400 shops), budget, staff, and time considerations require them to be more circumspect. They must consider alternatives that decrease the risks and costs of switching to an unfamiliar language, retraining or hiring new programmers, and rewriting reams of code.

Microsoft's .NET platform has emerged as an alternative modernization path for iSeries shops that are looking for a faster, less expensive, and less cumbersome route. Adoption of .NET is particularly of interest to companies that are implementing Web services and SOA. In addition, because it embraces many programming languages, it provides flexibility and advantages in terms of cost and staff considerations – developers can use whatever language is most appropriate for them and the project.

The Options

Although Java/WebSphere and .NET are the routes of choice for many enterprises that are undertaking modernization of iSeries systems, these are not the only alternatives. All of the following are means by which an organization can achieve a degree of modernization (or at least the veneer of it):

- Replace or rewrite applications
- RPG/400 emulation
- RPG/400 screen scrape
- Host-based ad hoc
- Migration to Java/C#
- RPG portfolio transformation

In reality, these options are not mutually exclusive. An organization will choose the path that is best for the job and for the enterprise, and

sometimes that means a *combination* of alternatives. This may mean the use of DB2/400 databases, SQL databases, Windows Servers, and IBM iSeries eServers in any combination according to requirements and goals.

That said, let's consider each of the modernization options in turn, and its effects on the "four pillars" mentioned above – people, applications, processes, and data. As you read about each one, keep in

certain that the new system will facilitate evolution toward that vision.

- *Don't compromise.* The iSeries is robust and stable – your target platform should be no less. Organizations that have "grown up" with an iSeries appreciate its safe environment of almost never-fail hardware (iSeries systems rarely crash) and resistance to hacking or security issues (few iSeries systems are hacked or

“Whatever the level of modernization you are currently seeking, the existence of the iSeries presents both challenges and opportunities”

mind not just these four pillars but also these good pieces of advice:

- *Remember the meaning of modernization.* IBM's definition of "modernization" is: "the enhancement, conversion, or rewriting of iSeries applications to include new technological capabilities that did not exist when the applications were originally developed." This is the target, but it must be achieved in the context of your business's specific circumstances and goals.
- *Know where you are going.* Consider the evolution of processes, the ease of integration, and the way people will fit into this ongoing evolution. As you develop plans for modernization, base these on a vision for the future of your business, and be

become prone to virus attacks). Given that the applications running on the iSeries are the critical back-office applications of an enterprise – ERP, manufacturing and supply chain, and so forth – it is reasonable to maintain high standards for stability and security. Such solutions can be found.

REPLACE OR REWRITE APPLICATIONS

"Rip and replace" means reinventing entire applications. Enterprises that have chosen this route – typically in an attempt to duplicate applications in Java – have usually found it to be an exercise in futility. In reality, an organization that is replacing its systems is not engaging in modernization but in an expensive,

time-consuming, and highly disruptive effort to create an entirely new system. This is not an entirely sensible way to approach the challenges of minimizing disruption and cost—or to take advantage of the opportunity to preserve, leverage, and enhance the investment existing on the iSeries system and related assets.

RPG/400 EMULATION

“Lift and shift” emulation leaves the application exactly as the original, nonmodern application, except that now it exists in a proprietary environment. Again, this option is not entirely sensible. It is demonstrably an expensive lateral move, and once accomplished, this sort of a platform shift is no closer than before to exploitation of modernization technologies such as Web services.

RPG/400 SCREEN SCRAPE

Screen scraping (or “Web facing”) really isn’t so much a method of Web-enablement as it is a look-and-feel “upgrade.” It has been very popular among users seeking to make applications and data available through a graphical interface or to convert host screens to a Web-like appearance, without altering the applications or processes. The resulting Web pages are limited in functionality and cannot be enhanced. If any changes to the application are made, all of the affected screens have to be re-scraped – again, without the true benefits of Web enablement. As a result, business applications cannot evolve, and even simple modifications become risky, cumbersome, and costly.

An expensive option that rarely returns its value, screen scrap-

ing fails to leverage or enhance the organization’s iSeries assets. Programmers simply pass the legacy source code through a tool that creates the output. The data, which remains on the iSeries, is not modernized in any way, and the legacy source code and applications also remain. Only the interface is modern. Screen scraping is at best a stop-gap measure; eventually, true modernization will have to be achieved through some other means.

HOST-BASED AD HOC

In a host-based “ad hoc” system, both Java and .NET are avoided. The iSeries remains central, preserving existing data assets. However, these solutions are extremely proprietary, placing the enterprise and its IT evolution at the mercy of the vendor or vendors involved with the solution. In this sense the “application” pillar is compromised, as is the “people” pillar, since the programmers will be largely dependent on an external party to maintain and enhance the system. This solution may be attractive to a small shop with a very small programming staff that would be overwhelmed by a change of platform. However, the disadvantages of this route suggest that alternatives should be explored if modernization for Web services is an eventual goal.

MIGRATION TO JAVA/C

Language converters attempt to translate RPG source code to C# or Java. It may be claimed that an RPG programmer can recognize the “ghost” of the original code, but in fact the converted code is neither native RPG nor native Java or C#. This code presents challenges for both current and future developers. It may be impossible for RPG programmers – who understand the application – to maintain. Interestingly, C# and Java programmers typically find the converted code just as un-maintainable. This

“The challenge is to determine the best way to bring the enterprise system into the 21st century without incurring unnecessary disruption to the business, while managing costs and achieving a reasonable timeline”

is because C# and Java are object-oriented languages, whereas RPG is procedural. The translation results in an abomination to C# and Java language constructs.

RPG PLATFORM TRANSFORMATION

From the foregoing brief descriptions of modernization options, it may seem that there is no good alternative. Every option presents disadvantages that, depending on the organization's specific goals and circumstances, may be prohibitive, and often a combination of alternatives, and even of platforms, is the best solution to meet the challenges and opportunities of the enterprise's specific case. In many cases, however, a more integrated and flexible solution can be found in new tools that have emerged out of the demand for iSeries modernization.

It's now possible to migrate an existing system to .NET with little reengineering, while preserving scalability and positioning the migrated application to implement Web services and SOA. The most promising solution may be an enhanced version of RPG that is both MSIL compliant and fully object oriented. By migrating to such a language, the organization can leverage the skills and knowledge of its engineering staff – allowing the RPG programmers, who possess intimate knowledge of the applications and processes, to develop new Windows and Web-based applications. In addition, because the new language is MSIL compliant, RPG programmers and MSIL programmers (VB, VS, C#, and so forth) can work together on the same projects. The language bridges platforms both on the technological and the human level.

For implementation of Web services and service-oriented architectures, the availability of an MSIL-compliant RPG language provides a vital link. It ensures that the RPG applications participate in

.NET, which in turn supports all of the standards – WDSL, XML, and so on – that enable Web services and SOA. Through such a language, RPG applications can fully participate in Web services and be modernized to and included in an SOA, while retaining the skills and knowledge of existing programming resource (RPG as well as OO), and without rewriting or re-hosting the applications. In other words, the iSeries and each of the pillars that supports both the IT system and the organization – the RPG programming support staff, the critical iSeries business applications, the inherent business processes they support, and the data that resides on the iSeries – can be transformed into a modernized, Web-enabled system, thanks to tools that bridge the iSeries/RPG and .NET.

In addition, an SOA, once implemented through an MSIL-compliant language, provides an elegant path to modernization that no other technology or methodology has accomplished to date. By encapsulating business functions performed in the legacy environment inside Web services, a company can now leverage the investment in the business system while making it available to other heterogeneous environments. Furthermore, using the SOA model, a company can extend those parts of their legacy application that they need over a time frame that makes the most business sense to them.

Checking the “MAP”

In 2004, Microsoft teamed with nine iSeries service providers and ISVs in an alliance – the “Midrange Alliance” – intended to encourage iSeries shops to consider a variety of alternatives when they modernize. The project was spearheaded by ASNA – a longtime iSeries vendor that markets extension, modernization, and migration software products, including ASNA Visual RPG, an MSIL-compliant, fully object-oriented RPG language, and

its Monarch product, an iSeries/400 to .NET replatforming analysis and transformation solution. The Midrange Alliance, which now has 19 members, provides resources for organizations planning modernization efforts on their Web site, including case studies, white papers, and pertinent information on IBM, Microsoft, and third-party products. Although the alliance may have a reputation for offering migration-only options, members, which include ASNA and LANSA, another founding member, also provide solutions for organizations that want to retain sensible loyalty to the iSeries.

Summary and Evaluation

For the six modernization options listed earlier, we can now provide a concise summary: the first two, replace or rewrite applications and RPG/400 emulation, are not sensible approaches to modernization, if they could be considered modernization at all. Screen scraping creates the look of modernization without doing so in reality, and in fact results in programs that resist modernization. Host-based ad hoc systems bind the organization to proprietary, non-standard technologies and remove maintenance, enhancements, and capabilities from the organization to the vendor. Language conversion results in the loss of the original RPG developers (and their “tribal knowledge”) and in code that is difficult to maintain for RPG and Java or C# developers alike.

Platform transformation tools, such as an MSIL-compliant RPG language, offer the greatest promise for meeting the challenges of modernization: bringing the enterprise system into the 21st century while avoiding unnecessary or costly disruption to the business. In short, it provides the best means of managing costs and achieving a reasonable timeline while making the most of opportunities that leverage an organization's assets. 