

Meeting New Business Demands with Existing IBM System i Applications

A pragmatic approach

White Paper

Meeting New Business Demands with Existing IBM System i Applications

A pragmatic approach

September 2006

Magic Software is a trademark of Magic Software Enterprises Ltd. All other product and company names mentioned herein are for identification purposes only and are the property of, and may be trademarks of, their respective owners.

Magic Software Enterprises has made every effort to ensure that the information contained in this document is accurate; however, there are no representations or warranties regarding this information, including warranties of merchantability or fitness for a particular purpose. Magic Software Enterprises assumes no responsibility for errors or omissions that may occur in this document. The information in this document is subject to change without prior notice and does not represent a commitment by Magic Software Enterprises or its representatives.

Contents

Introduction.....	4
The System i As Part of the Larger IT Community.....	5
Demystifying the Buzzwords.....	6
The Benefits of SOA and Composite Applications	7
Industry Trends: Where are the Major Vendors Headed?.....	9
The Advantages of Being a System i Shop.....	10
A Pragmatic Approach.....	12
Next Steps	14
About Magic Software Enterprises	15

Introduction

IBM System i™ users have a right to be proud. Years ago, they made a wise choice to invest in a hardware platform (the AS/400), an operating system (OS/400), a programming language (RPG), and robust business applications that have proved over the years to be easily maintainable, incredibly reliable, and extremely easy to use and understand, delivering a high return on investment.

In fact, a senior IBM System i executive stated at a recent COMMON user conference that 80 percent of *all* IBM server customers have a System i (or iSeries or AS/400) somewhere in the company, running at least one key application. Massive amounts of mission-critical data and innumerable applications vital to these organizations reside securely and reliably in System i environments.

But today, everyone's talking about "modernization," which often translates into messages that OS/400 users must move off the System i platform, abandon the "legacy" applications that still deliver high value to the organization, or, at the very least, adopt much more complex software components and complex languages such as Java. The simplicity, reliability, and ease of use upon which System i users have relied for decades are suddenly in jeopardy.

Will it be possible for OS/400 users to retain the benefits of the environment that has delivered so much value while modernizing those applications so that they can be integrated into today's wider IT community? Is there a way to take advantage of and protect the investment that still has great value to the organization while enhancing those applications to meet new business demands for extensibility, flexibility, agility, return on investment, and communication with customers and business partners?

The System i As Part of the Larger IT Community

Until relatively recently, most IT applications operated as "islands" of computing capability, communicating with each other via the transfer of files and reports. The usefulness of that model, however, has long since passed. It is certainly true that batch processing and near-real-time updates still provide a reliable and viable model for some applications. However, today, thanks in large part to the Internet, many businesses need to provide data in real time (on demand) to users, remote locations, customers, and other points in the supply and service chain. The days of stand-alone islands of computing are fast disappearing.

If a company is going to survive, let alone thrive, establishing a strong competitive edge in a global economy even when it is servicing tight geographies and/or niche vertical markets, it will have to become part of the larger IT community, extending its business processes across both internal and external channels and across the Web. Following are some of the most common types of integration needs:

- Linking product information, inventory adjustments, order processing, credit approvals, and customer discount information to an e-commerce website.
- Enhancing customer service operations and help desks by connecting them directly to information about customers, past calls, problem resolutions, order tracking, and more.
- Synchronizing data between various remote offices.
- Consolidating financial results of different business units.
- Automating order processing and eliminating the need to create and mail or fax hard copies of purchase orders or invoices.
- Meeting the requirements of other companies in the supply chain to automatically exchange documents in standardized electronic form, directly from one application to another.

- Enabling the sharing of data and logic between multiple applications that use that information across diverse databases, applications, operating systems, and geographies.

To meet any of these goals, System i applications will have to move off of the island and into the mainstream. This **can** be done—in a way that will let platform lovers maintain the simplicity, continue to enjoy the reliability, and protect and extend their investment in their proven applications.

Demystifying the Buzzwords

There's certainly no shortage of talk about the need for OS/400 users to "modernize" their applications. The buzzwords abound to the point that many people have come to think of the entire conversation as "just hype." But it isn't. There are real, viable, emerging technologies that are changing the paradigm of the way companies will build and maintain applications and integrate them across diverse environments both now and into the future. Will these technologies require some rethinking on the part of businesses and their IT departments? Yes. But this process will be evolutionary rather than revolutionary, modifying the paradigm over time and allowing for phased integration implementation rather than dramatic changes that would send shock waves through a company or its budget.

Here are some of the terms that are necessary to a discussion of this new technology:

Service Oriented Architecture (SOA): Perhaps better described as a "methodology," SOA seeks to define reusable components or "services"—repeatable business tasks—that contain application logic that performs a specific business function such as "validate customer number" or "insert employee ID number." These services are defined in business terms, without regard to a specific application, operating system, programming

language, or platform. System i is a platform well suited for the delivery of robust and reliable services and reusable components.

Composite applications: These are applications that are assembled using both components from multiple existing applications and added functionality developed specifically as services such as “read customer order,” “validate customer status,” “calculate customer discount,” or “apply discount to order.” The component parts of a composite application are created independently of each other, without knowledge or consideration of the different information models being used. In fact, they are platform-agnostic, meaning that the component providing the service may reside on any platform capable of providing the service.

Web Services: The use of standard protocols (such as WSDL, SOAP, and UDDI) to “wrap” SOA services so that they can be shared with other applications across the Web. This term is often used synonymously with “SOA,” although Web Services is really an enabling mechanism for SOA.

The Benefits of SOA and Composite Applications

The greatest benefits realized by SOA are reusability, flexibility and agility. In the business environment today, companies that remain competitive are able to respond quickly to rapidly changing environments. In some situations, these changes can be anticipated and planned for, such as industry compliance standards or the requirement to attach RFID tags on product shipments. But other changes might be necessitated by unexpected events that change the business climate in a moment: the announcement by a competitor of a new product or a price reduction; a major natural disaster such as a hurricane or tsunami; or even an averted terrorist threat that would decrease international travel and/or alter security requirements. It is also true that many business events that demand change in business processes and IT systems are driven by

internal decisions to improve customer satisfaction, increase efficiency, drive revenue and improve competitiveness. Organizations that cannot respond quickly, adjusting their business processes to address these changes, will fall behind. Integration frameworks that allow orchestration of an existing reusable library of services bring the benefit of agility.

The second immense benefit of SOA is productivity, which translates directly into cost savings. By managing changes to the business processes rather than maintaining the way that those processes are addressed in dozens of individual applications, companies save significant expense over time. Each time a service is reused, cost savings accrue from not having to insert it into, or maintain it in, another application. Each time a business process is changed, money is saved by not having to touch the applications that use that process.

It is not only the manner of implementing the changes that brings cost-savings; the business change itself is often the source of tremendous economic and competitive advantage. By responding quickly to business changes, an organization reaps the benefits sooner and for a more sustained period of time. Organizations that are bound by approaches that require long lead times will simply not be able to remain competitive. A sound ROI analysis should incorporate all of these benefits: more efficient IT changes, faster time to value, and capacity to act on a greater number of business initiatives.

This is already happening in the System i community. In fact, many organizations have already begun SOA implementations based on System i. For example, Paul Leone, CIO of adidas Canada utilized a SOA architecture to bring supply chain efficiency resulting in an immediate 400% ROI for his company. IBM saw such value in his story that they placed two-page advertisements in *Business Week*, *Forbes*, *Fortune* and several other IT publications explaining how Leone selected IBM System i and an integration

suite provided by Magic Software Enterprises, in order to achieve these real business benefits.

Industry Trends: Where are the Major Vendors Headed?

Just about all of the major vendors have made significant commitments to SOA, giving credence to the belief that SOA is not just a passing buzzword, but truly an architecture that they are investing in and building their offerings around.

In fact, within the past year, IBM has joined with sixteen other companies, including Sun Microsystems, SAP, Oracle, Red Hat, Siebel Systems, and BEA, in a group that is collaborating on the development of standardized specifications for SOA—specifically Service Component Architecture (SCA) and Service Data Objects (SDO). Such standards, which the collaboration group intends to turn over to an independent standards body when complete—will be important in ensuring uniform access to data across diverse environments.

While standards-setting groups have sometimes been notoriously unproductive in terms of final product, indications are that this group is dead serious about reaching its goals, and each company involved seems to recognize that SOA is the future and it must participate or be left without a voice.

Analyst groups, such as Gartner, point to the importance of an emerging software sector known as "integrated services environments." The advantages of an integrated services environment come through a consistent approach to the design and deployment of services. Any integrated services environment suitable to System i environments will quite naturally take advantage of RPG applications by automating calls to RPG and exposure of RPG applications as services. Unfortunately there are

scores of software tools and layers that can sometimes be required to achieve a "complete stack" of SOA technology.

Vendors offering a unified studio environment for SOA design should be given careful consideration. However, in addition to a unified studio environment, it is advisable to seek solutions with a library of included services from which business processes can be built. Solutions that charge extra fees for additional services or components can become prohibitively expensive and tend to create unnecessary barriers to business process integration. Look for a vendor that offers a full suite of included adapters for databases, messaging, and more. Insist on a solution that runs natively on the System i (regardless of which operating system you select). And check for support of native System i features such as spool files, data queue, program calls, user space, DB2/400, etc.

The Advantages of Being a System i Shop

In moving forward with a SOA-oriented implementation, many System i users actually have a few distinct advantages over their counterparts on other computing platforms.

First, System i shops, because of the stability of the core applications they run, have felt less need to adopt technologies such as Java, .NET, and even web-facing tools. In the case of looking at an SOA integration initiative, this means that some of them have not yet spent money and resources on interim tactics that got them only part of the way towards the goal. SOA is much more than putting a pretty front-end on your applications and taking them to the Web, so organizations that have already done that still have to take the next steps. On the other hand, companies that are still looking at green screens can now, with a well-defined SOA project, jump directly to the process of defining their services, reusing components that are embedded in their existing applications, and move more directly to reach the end objective.

Second, an effective SOA implementation team, by definition, must include both business leaders and IT leaders. An in-depth understanding of the business is necessary in defining which services are key to the operation of the business and the way in which these services should be constructed and integrated across the enterprise. Additionally, technical expertise is important in how these services will be integrated within the technical environment. In most large, mainframe-based companies, business users have little understanding of the technical issues, while many on the IT staff are not close to, and have little understanding of, the business issues. Not so in System i shops, where many of the programmers and systems staff have quite an extensive understanding of the business aspects of the applications. Indeed, it is not uncommon to find System i developers who came out of the business areas. The ever critical dialogue between business and IT, upon which the success of an SOA integration project depends, is likely to run quite smoothly in the majority of System i installations. Here again, selection of a studio environment that connects business process models to the execution of those processes is vital. An approach that gives business analysts tools that allow them to orchestrate services with little or no reliance on programmers can be beneficial. And even the programmer benefits from these tools through a direct method of orchestrating existing application logic rather than a mandate for a complete rewrite.

Finally, remember that SOA is the enabling mechanism that will integrate logic and data into the wider internal and external IT environment. With that in mind, consider just how much business logic and data is already embedded in applications and databases on the System i. For many companies—even those that rely more heavily on other platforms—the System i may, indeed, be the key element of an SOA implementation. The "sizzle" may be up front on the Wintel and other servers, but the real "meat" of the business often resides on the System i.

These advantages actually give System i users a very good chance of being able to: 1) realize some early successes very quickly, and 2) minimize the cost and learning curve necessary to approach SOA when compared with their business associates on other platforms.

A Pragmatic Approach

While this may all sound good in theory to many System i companies, it's understandable that some degree of skepticism may prevail. After all, the complexities of Java environments or .NET sometimes seem overwhelming to a shop accustomed to the simplicity and ease of use of the System i environment.

These skeptics are right to be wary of the high-end, overkill solutions that are not necessary and will not give System i users what they want: a way to retain the platform with its many advantages at the same time that they move to deliver an IT solution that will let the company grow and be flexible in meeting business needs. A more pragmatic approach is in order.

Magic Software Enterprises understands this.

Twenty years ago, Magic Software invented a fundamentally new application development and deployment technology by introducing a metadata-driven, rule-based platform and development framework. Since then, Magic has observed two trends: 1) Organizations have an increasing need to communicate between departments internally as well as with external partners, streamline business processes, reduce cost, and increase revenue; 2) The demand for technology enablement is no longer coming exclusively from the IT department, but from the business units as well. Ensuring that integration projects meet the objectives of the business requires increasing involvement from the business analysts. However,

application development and integration of systems is not part of their core skill sets.

Taking advantage of its existing development platform, Magic Software enhanced its already-existing ability to implement application integration layers, create SOAs, and manage end-to-end processes. The result is one technology stack with two products—the eDeveloper™ composite application platform, and the iBOLT™ business integration platform—that delivers a unified and highly scalable development and deployment environment.

This offers System i users an attractive alternative to their two previous options of either remaining on the island, disconnected from the rest of the enterprise and supply chain, or replacing their System i applications with expensive new applications designed for another platform or written in more complex languages.

This alternative strategy is to implement a framework that can:

- Invoke RPG (and other language) programs with full support for passing and receiving parameters;
- Integrate and expose System i program processes;
- Connect System i databases and files with high-speed gateways;
- Enable programmers to quickly and easily integrate 5250 terminal-interfaced applications with other applications in the enterprise;
- Deliver a straightforward process for providing automated programming access to legacy applications;
- Assemble composite applications from components of existing and proven applications, extending their value and building on the investment already made;
- Provide an incremental, risk-free migration path from green-screen legacy applications to modern GUI applications; and

- Deliver a means to design, test, and monitor using a code-free, visual process design interface.

Next Steps

If your company has a significant investment in its core business applications, you should take a look at SOA as a means to protect and extend this investment and meet the changing business needs of your organization.

What if your company itself does little or no application development but, rather, uses software purchased from an ISV? In that case, a business integration platform is ideal, because it allows your existing applications to be service enabled.

One way or the other, SOA will impact your company. Your applications, developed internally or purchased from another company, will need to integrate with the wider community. SOA is the route that will get you there.

About Magic Software Enterprises

Magic Software Enterprises is a software provider of enterprise application development, deployment and integration technology.

The company's service oriented platform allows small and medium-size enterprises to rapidly develop, change and deploy business solutions integrated with existing and legacy systems. Magic Software enables enterprises to increase their agility and rapidly adapt to business changes by aligning their IT with their business operations and accelerating the evolution to a service oriented architecture. Our products are built upon 20 years of R&D and customer experience. Magic technology, products and professional services are available through a global network of subsidiaries, distributors and over 2500 Magic Solutions Partners in approximately 50 countries. More information about Magic Software can be found at: www.magicsoftware.com.