WEB-ENABLING MAINFRAME APPLICATIONS: THE CASE FOR A MORE STRATEGIC SOLUTION

Re-engineering legacy mainframe applications through Java programmatic access

Version 1.3

EXECUTIVE SUMMARY

As we begin a new millennium, there is no doubt that the Internet is having a dramatic impact on enterprise computing. Web-enabled E-commerce and eBusiness applications have transitioned almost overnight from industry hype to reality. Not only are they real, they are delivering significant, measurable business benefits. Just a few examples of these successes are:

- A health insurance provider is using the Internet to reduce the cost of processing claims from $3.00 per claim to 70 cents.

- Cisco Systems’ Web-based ordering system generates about $15 million of sales per day (two thirds of its sales).

- A major bank reports that its new online banking system has reduced customer service calls by 40%.

- A semiconductor manufacturer is using a new online ordering system to improve its production forecasting and save its distributors $20 million per year.

With these kinds of results and industry surveys projecting that the volume of eBusiness will increase to $300 - $500 billion by 2002, there is no longer any question that these Web-enabled
applications are strategically important. The only question remaining is how best to deploy these new applications quickly to create a competitive and lasting advantage.

In today’s competitive environment there is no time to reinvent the wheel. The most effective way to speed the deployment of Web-enabled applications is to leverage the information resources that are already in place. Because more than 70% of enterprise data resides on IBM hosts, the mainframe is a natural starting point for building new E-commerce and eBusiness applications. In addition, with the recent legacy investments ensuring Y2K compatibility, most experts expect these mainframe applications to remain viable for years. Industry surveys project that by 2002 nearly one third of all mainframe capacity will be supporting eBusiness applications with legacy access.

Web-to-Host – Tactical versus Strategic

There are essentially three ways to enable legacy mainframe application integration in the new eBusiness environment.

The first is a tactical approach that simply puts a new Web interface on legacy applications. The new user interface can either have the look-and-feel of a traditional character-oriented 3270 or 5250 display or it can be a rejuvenated graphical user interface. The key benefits of this approach are that it can be implemented very quickly at low cost and it can make legacy applications less intimidating and easier to use. The key drawback is that this solution only addresses cosmetic user interface issues. It can only be effective when the design of the legacy applications exactly matches the needs of the new Web users – and this is seldom the case. For example, when an airline deploys a Web-based reservation system, they cannot simply put a graphical interface on their old mainframe applications. Those applications, designed for trained reservations agents, are much too complex to be put directly into the hands of customers. Furthermore, eBusiness applications must function as electronic “guides” to attract customers and employees, lead them through transactions, and give them an incentive to return for their next transaction. Few mainframe applications were designed to meet these requirements.

The second Web-enabling technique is the reengineering of user interfaces. This is a more strategic approach that leverages mainframe applications and enhances those applications to meet new strategic business requirements. User interface reengineering simplifies the screen-to-screen navigation that characterizes all legacy 3270 and 5250 applications.

The third technique is full application reengineering. This includes adding new front-end business logic to legacy applications. This new logic is often needed to transform legacy applications into new eBusiness applications. It also includes a new functional capability – programmatic automation – where the resulting “user” of the legacy data is not a real end-user, but another application program or system. The best example is Enterprise Application Integration or eBusiness Integration, as it is now known, where the requirement for legacy application access can be generated directly from another application such as your CRM or ERP systems.

New front-end business logic not only makes legacy applications easier to use, it can personalize the applications to meet the needs of individual users and help develop ongoing relationships with customers. Personalization is not just a gimmick; it has proven business value. A recent survey of eBusiness sites shows that personalization increased revenues by an average of 50%. One consumer E-commerce site reports that personalization caused 10-30% of visitors to make a purchase compared to a 2-4% rate without personalization. Examples of personalization can be found in new Web-based retail applications that guide users to products that are likely to be of interest based on previous buying patterns. This kind of interaction requires reengineered application logic, not just a new user interface.
True automated programmatic access, on the other hand, enables all of your legacy application logic and data to be available to any other systems in your enterprise. In fact, this architectural approach makes legacy system accessibility easy for your customers, partners, vendors and suppliers as well. Imagine replacing all of your internal manual operator interaction with automated program-to-program communications.

Programmatic automation reengineering can also be accomplished without making any changes to the legacy applications. All additional business logic can be implemented on middle-tier servers. These server-side applications act as intermediaries between your legacy applications and any other enterprise system and can synthesize data from mainframes and other sources to create new applications. This eBusiness integration point can seamlessly tie your enterprise applications together or create new applications that forge long-term business relationships with customers.

While reengineering does require an investment in new software development, this investment can be justified by the potential payoff of a strategic business initiative. Simply putting graphical user interfaces on legacy applications cannot be expected to deliver the personalized service that Web users demand nor does it include any programmatic automation capabilities. In today’s environment, competitors are always just a mouse click away. Products such as Red Oak Software’s Stingray SDK, that enables full legacy application reengineering, will become the products of choice for delivering the functionality required for Web-enabling legacy E-commerce solutions and for developing new eBusiness applications.

INTRODUCTION

A common misconception about Web-enabling legacy applications is that it is a single problem with a single solution. In reality, there are a variety of reasons to Web-enable legacy applications that range from tactical cost-saving measures to strategic eBusiness initiatives.

Each Web-enabling project has different business and technical goals, but they generally fall into one of the following categories:

- Enhancement of existing business processes
- Reengineering of business processes for employees and business partners
- Deployment of new customer-centric eBusiness processes
- Enterprise Application Integration (EAI) or Internet Application Integration (IAI) initiatives

Web-enablement can enhance existing business processes by reducing costs or by expanding the number of employees that have access to mainframe applications. A common cost-saving technique is the use of Web-based 3270 terminal emulators to replace the “fat” emulators currently being used. “Fat” emulators have high administrative costs because the software must be installed and maintained on each desktop. “Thin” Web-based emulators reduce the total cost of ownership because they are administered on centralized servers rather than on large numbers of individual desktops.
Existing internal business processes can also be enhanced. For example, a human resources department can Web-enable the mainframe application that maintains the company’s 401k plan. This gives all intranet-connected employees direct access to their account information. Employees gain greater control over their benefits and HR personnel are freed to handle other tasks.

Even greater benefits can be achieved, however, by reengineering the business processes used by employees and business partners. Business processes such as purchasing, billing, and logistics are being reengineered to save time and money. These reengineering schemes typically eliminate the need for intermediaries by providing end users with direct intranet access to applications.

A third category of Web-enablement projects is the development of new customer-centric processes. These are either business-to-business or consumer eBusiness initiatives. Companies are building electronic storefronts to reduce or eliminate the need for sales or support staffs. Examples of customer-centric eBusiness applications include airline reservation systems and on-line brokerage services. These applications require extensive reengineering of legacy applications because they often replace sales and support personnel and, must therefore, build customer relationships while providing easy-to-use services.

The last category – programmatic automation - may offer the most benefits. Almost every CIO and IT manager lists application integration as the area of largest investment in 2000. The reasons are simple. By enabling disparate production applications, within and without the enterprise, to communicate has an enormous payoff for all. The immediate savings include enhanced quality and integrity of shared data, but is quickly followed by leveraged functionality across the enterprise. In fact, eBusiness integration will likely become the cornerstone for all IT activities in major corporations for the next few years.

Connecting your ERP or CRM systems to your legacy production applications conjures up the kind of leveraged payback promised by the industry for all these years. By tying together production, inventory, shipping and accounting into a seamless interchange of data, phenomenal benefits can accrue. Extending this concept to your customers, suppliers and partners through the Internet only compounds the benefits. In fact, this scenario is the very essence of eBusiness - using the Internet to deliver a full range of services and support to your employees, your customers, your partners and your vendors.

THE CHALLENGES OF LEGACY INTEGRATION

Each category of Web-enabling projects focuses on different communities of users and each has very different business objectives. In all cases the goal of Web-to-Host integration projects is to match user requirements with the capabilities and limitations of the legacy applications. These user requirements can be logically grouped into three categories:

- Presentation style / User Interface
- Application Navigation
- Data Accessibility / Information Content

There are two basic presentation styles that can be supported – legacy character-oriented user interfaces and graphical user interfaces. Although the vast majority of end-users prefer graphical
interfaces, those who have been trained to use character-oriented 3270 terminals may prefer to continue using that interface. The 3270 or 5250-style interface is particularly appropriate for users of repetitive, keyboard-intensive applications such as reservations systems or data entry applications.

Screen-to-screen navigation is another key issue. Mainframe-based interactive applications require users to navigate through multiple screens of information to complete a single transaction. This is due to the limited display area of the terminals. In addition, most CICS applications are written in a pseudo-conversational style that conserves host resources by breaking applications into many short-duration user interactions. One of the key challenges of Web-enabling projects is to improve ease-of-use by simplifying or eliminating screen-to-screen navigation.

The final and most critical requirement is to provide end-users with the information that they need. In some cases, information can simply be mapped from a single source, such as a 3270 session, into a format that is suitable for the end user. However, information must often be obtained from multiple sources, integrated and synthesized before it is presented. Information sources can include multiple SNA sessions, local or non-local databases, and non-SNA hosts.

When there is a close match between user requirements and legacy applications, little or no application logic needs to be developed. There are a number of fine web-based terminal emulators that can provide users with either “green screen” or graphical interfaces to these applications.

Yet, in most cases, simple terminal emulation or GUI rejuvenation is insufficient. The bottom line is that few mainframe applications can be adapted to new business purposes without reengineering their application logic.

**TACTICAL VS. STRATEGIC SOLUTIONS**

Web-enabling projects can be categorized as either tactical or strategic solutions. Both types of projects can yield significant benefits, but they differ in their impact on the enterprise and in the level of investment required for deployment.

Tactical solutions provide Web-based users with new interfaces to old applications. Tactical measures are attractive because they are often turnkey solutions that require little investment. For this reason it is often tempting to use tactical solutions to solve strategic problems. One example of a new eBusiness application was a manufacturing company that decided to give its top customers direct access to its mainframe inventory and order entry system. Initially, the “best” solution seemed to be a gateway product that maps the 3270 screens, one-for-one, into equivalent graphical Web pages. This tactical solution was successful - until a competitor rolled out an E-commerce application that was much easier to use. The competitive system required almost no screen-to-screen navigation and it tracked customer buying patterns to offer special promotions and allow customers to reorder with a single mouse click.

The first manufacturer had to scrap its original 3270 emulation solution in order to respond to the new competitive threat. The original solution turned out to be a tactical mistake because it did not have the ability to add new front-end logic, consolidate 3270 screens nor create the required customer tracking databases. This is a classic example of the problems that can occur when tactical solutions are used to solve strategic problems.
Strategic solutions enhance legacy applications by reengineering user interfaces and supplementing the application with new application logic. In some cases, entirely new applications are created using legacy applications as little more than a back-end data source for other systems. It is critical to match project requirements with the capabilities and limitations of Web-enabling solutions.

**TACTICAL WEB-ENABLING SOLUTIONS**

Tactical solutions provide incremental improvements to existing business processes. These solutions typically focus on cost reduction and enabling the growing community of intranet users to participate in existing business processes. The objective is to enable intranet users to access legacy applications, not to reengineer those applications. They take advantage of the fact that most mainframe applications are designed for terminal access. Because terminals and Web browsers are both interactive devices, it is possible to link Web users and mainframe applications without reengineering.

Most tactical solutions are point solutions focused on terminal emulation. They do not support access to other data sources, such as relational databases or non-SNA hosts. They also do not allow customers to reengineer user interfaces or develop new front-end application logic using industry-standard development tools and languages. Although they have a low initial cost, tactical solutions can be very costly in the long run because they are unable to support future requirements beyond terminal emulation. Tactical Web-enabling projects typically employ “green screen” 3270 or 5250 emulators or products that rejuvenate these sessions by mapping them into a Web-like graphical user interface.

**“GREEN SCREEN” EMULATION**

Web-based “green screen” emulators allow 3270 and 5250 users to migrate from legacy SNA networks to IP-based intranets or extranets. These products are implemented as Java applets or ActiveX controls that are downloaded into browsers to provide the character-oriented look-and-feel of these terminals. These Web-based emulators are an alternative to conventional “fat” 3270 emulators.

The only real benefit of Web-based emulators is reduced administrative costs. Administration is simplified because emulators are installed and configured on central Web servers rather than on individual desktop systems. The emulators are downloaded, on demand, to end user desktops. These solutions also leverage enterprise intranets that can provide lower cost connectivity than legacy SNA networks, but are viable solutions only for users who need a “green screen” interface.

**USER INTERFACE REJUVENATION**

Green screen emulation provides 3270 and 5250 users with an intranet migration path, but this is not a viable, long-term solution for the majority of users either. Most intranet and extranet users have never used a mainframe terminal – nor do they want to learn. To support these users, 3270 and 5250 data streams have to be mapped into a more familiar and user-friendly graphical interface.

Many Web-enabling products employ screen-scraping techniques to selectively extract fields from 3270/5250 screens and position then on the graphical interface. Graphical elements are usually added to make the interface more inviting and easier to use. For example, graphical controls such as radio buttons and checkboxes can replace traditional 3270 text-based menus. To simplify screen-to-screen navigation, PA, PF, and ENTER key functions can be replaced by descriptive hyperlinks.
3270 screen-scrapers use two basic techniques to present information to end-users. Some products leverage the Web’s native GUI by mapping the terminal data streams directly into HTML. Other products employ Java applets or ActiveX controls that run within browsers and control the user interface. GUI rejuvenation solutions can be deployed quickly and at minimal cost, but they are only successful when there is a close match between host application design and end user requirements. The information content provided by host applications must match the information needs of the users and the host applications must have an interaction style that is suitable for the target user community.

If there is a mismatch of information content or interaction style, then tactical Web-enabling solutions are not likely to be successful. Strategic reengineering is the approach that can resolve this problem.

STRATEGIC WEB-ENABLING SOLUTIONS

In contrast to tactical solutions that simply replace the 3270 or 5250 user-interface with a graphical equivalent, strategic Web-enabling solutions can be used to reengineer user interfaces or entire legacy applications. This involves the development of new front-end application logic – usually on middle-tier servers. These server-side applications use emulation technology to eliminate the need for changes to any mainframe applications or other software.

As eBusiness becomes the primary focus in all organizations, requirements for access to legacy application logic and data will undoubtedly grow. Most core business applications continue to reside where they were developed, on the IBM 390 or AS/400. These applications, originally designed for and delivered to the user via terminal, can now be accessed by emulation technology, which has progressed to the point where most experts consider it mature. As such, it can be relied upon to efficiently and effectively deliver legacy system access. As eBusiness integration requirements increase, the need for programmatic access to legacy applications will also grow. Rather than traditional Web-to-Host connectivity, such implementations will be marked by programmatic interface requirements. In this scenario, Application Servers and/or EAI infrastructure products will need programmatic access to those custom or proprietary applications running on mainframes and AS/400s. Using an automated toolset to develop Java terminal emulation programs for access to this application logic and the resulting underlying data will become an even more strategic initiative.

Strategic solutions should also leverage existing development tools and skills. Yet, the closed development environments of proprietary tools and languages limit many Web-enabling products. In contrast, open development environments such as Java have two advantages. First, they can leverage the customer’s current development skills, tools, and languages. Second, open development environments can make best use of new tools, languages, and object-oriented programming technologies as they become available. Because strategic applications have long life cycles, they must be able to leverage new technologies and quickly adapt to changing business requirements.

Platform independence is another requirement for strategic applications. IT managers who have experienced the evolution from mainframe-centric computing to client-server and, most recently, to network-centric computing models, understand the importance of building platform-independent applications.

Strategic applications must also scale to support large numbers of users and high transaction rates. This is particularly true of Internet-based applications targeted at consumers. With millions of potential customers, these applications have unprecedented scalability requirements. They must also be manageable and secure. IT professionals are struggling with these requirements as applications become
increasing distributed. What is needed is a scaleable, manageable, and secure platform for distributed computing. An industry-standard technology has emerged to address these strategic application requirements – the Java programming language.

Java enables developers to write an application one time and then port it easily to any platform that has a suitable Java Virtual Machine (JVM). Java is now implemented on virtually all application platforms ranging from IBM mainframes, to UNIX servers, to desktops. This allows Java applications to migrate more easily between many of the application platforms used in large enterprises.

Of particular interest to IBM mainframe users is the fact that IBM is fully supporting Java technologies on its mainframes and all other key application platforms. Java is also the language of choice for most application server vendors as well as the eBusiness integration vendors.

OTHER STRATEGIC SOLUTION REQUIREMENTS

Strategic Web-enabling projects may require full application reengineering for a variety of reasons, but the two most common are:

- Simplify or eliminate navigation between 3270 or 5250 application screens
- Access information from multiple mainframes and other client/server data sources

While tactical solutions can simplify screen-to-screen navigation by using descriptive hyperlinks, strategic solutions can often go a step further and use front-end logic to combine information from related terminal applications and present it to the user as a single Web page. The best way to simplify screen-to-screen navigation is to eliminate it entirely. New applications commonly require access to a variety of data sources including multiple 3270/5250 sessions, local and remote relational databases, and applications running on non-IBM hosts. Tactical Web-enabling solutions that only provide access to a single 3270 or 5250 session are not suitable for building these types of applications.

RED OAK SOFTWARE’S STINGRAY SDK

Red Oak Software is committed to developing strategic Legacy and Web integration products based on open and industry-standard Java technologies. Red Oak’s Stingray SDK product is a “best of breed” solution for reengineering all legacy 3270/5250 applications.

Stingray SDK meets all of the requirements for strategic Web-enabling solutions while also including full support for client-side terminal emulation and rejuvenation applets. Stingray SDK also leverages the application development skills that are inherent in most corporations. Its open development environment allows customers to use any Java integrated development environments.

The host session Recording feature of the SDK dramatically increases development productivity by automatically generating the Java code needed to access legacy applications. The SDK’s Playback feature enables the developer to perform in-line testing as well. It allows the developers to concentrate on application and user interface reengineering rather than the “plumbing” needed to access host data. Together the components of Stingray provide a complete solution and upgrade path - for those users
initially seeking a short-term tactical solution as well as for those who initially require a more strategic approach using the Stingray SDK.

Because Stingray is all Java, emulation solutions developed with the SDK achieve a high degree of platform independence. Applications based on Stingray SDK can run on any on any server – from NT to Unix to Linux or any client supporting a browser. Stingray provides full scalability with management, transaction, and security services.

Above all other benefits, Stingray facilitates development of sophisticated server-side applications that can easily be integrated with all application server products and with ActiveWorks™, the eBusiness integration solution from Active Software. It is in this automated programmatic environment that Stingray delivers its maximum value.

CONCLUSION

Tactical Web-enabling solutions can be very effective in simple 3270 or 5250 replacement applications, but IT executives must anticipate whether these applications will evolve into more complex applications in the future. Any future requirement to reengineer the user interface, add new business logic or access data from sources other than 3270/5250 sessions can render these tactical solutions obsolete.

Experienced IT executives also know that applications can have unexpectedly long life cycles. Testimony to this fact is that many IT departments still maintain applications that were deployed as “quick and dirty” solutions 10 or 15 years ago. Strategic Web-enabling products such as Stingray SDK will help ensure that today’s eBusiness applications do not become liabilities in the future.

Like the mainframes themselves, strategic applications have very long life cycles. It is critical to base these solutions on open industry standards such as Java to ensure that they can evolve to meet ongoing business requirements and take advantage of new platforms and tools as they become available.

Red Oak is committed to keeping Stingray on the leading edge of Web-to-Host and automated programmatic eBusiness solutions.