MICRO FOCUS

Application Modernization with SOA

While existing mainframe applications or so-called legacy systems may perform well, they will typically have been designed to operate as part of an infrastructure driven by green-screen character interfaces or batch processing. As new requirements such as new lines of business or regulatory compliance emerge, such systems can no longer operate in relative isolation. A key driver for modernization is the requirement to integrate legacy applications from across or even outside the enterprise while maintaining their original structure and function.

By enabling such systems to be extended to a Service Oriented Architecture (SOA) through Web services, existing mission-critical functionality can be integrated with other systems and act as strategic assets with proven business logic, accessible however and whenever required. Micro Focus SOA Express™ will leverage existing interfaces to mainframe applications, delivering them to an SOA through Web services without code change. This enables the extension of existing enterprise applications to meet new requirements in parallel with the continuation of existing business operations.

white paper

Introduction

Core to the effective operation of any IT department is its ability to both reliably and securely service existing business operations while rapidly and cost effectively adding new capabilities in response to business needs. But, according to Forrester, "The percentage of the IT budget that firms spend on application maintenance is widely believed to range from 60% to 80% and higher. This illustrates the challenges in meeting new requirements in a timely and cost-effective manner can often be challenging.

New requirements from the business are often related to existing business applications in terms of core functionality. The focus may be on a change of delivery channel, user base or organizational structure. More specifically such requirements can include requests to:

- Support a new channel or line of business
- Enable strategic IT assets for use in B2B transactions
- Integrate with other key systems internally or externally to share data and business processes
- Comply with regulatory requirements such as Sarbanes-Oxley, HIPAA and BASEL II
- Address new users, for example through the web or via mobile devices
- Re-organize, re-present or consolidate systems following a Merger or Acquisition.

Many additional business requirements can be addressed, in whole or in part, by leveraging existing enterprise applications. These applications, which may sometimes be considered legacy systems, represent a major corporate asset in terms of the unique business processes they contain and deliver. For mainframe systems that asset can only be exploited in support of new initiatives by providing access to the business logic locked inside COBOL, CICS and IMS applications.

This paper explores how modernization of enterprise applications can unlock these business processes. It examines how modernization compares to rewrite or package alternatives, and outlines considerations, benefits and delivery for SOA specifically as a vehicle for modernization.

Modernization and alternatives

Significant new business requirements frequently arise. These include adding a whole new channel of users (for example via the internet or mobile devices), compliance needs, delivering modern contemporary user interfaces or providing agile access to business information systems. This can prompt some fundamental decisions about the future of an organization's enterprise business systems.

While there are many hybrid and complex possibilities, such initiatives typically face three fundamental options of how to proceed:

1) **Rewrite** – Build a completely new system in the contemporary language of choice to directly meet the emerging business requirements

2) **Package** – Purchase an industry standard package that implements the core functions reflected in the initial business requirement

3) **Modernization** – Extend existing legacy systems to meet the new requirements.

On the face of it, the **Rewrite** option would seem to prove the most compelling option. The new business requirement can be directly implemented in new code by a team of outsourced or newly recruited engineers unburdened by the legacy of previous generations. That is, of course, a simplistic view. In reality, for most existing enterprises a 'rip and replace' strategy can prove a very challenging path to follow for these reasons:

- High Risk – large scale development projects notoriously overrun and complete failure is not unusual. Tying the delivery of key new business requirements to a major rewrite is a very high risk strategy
- High Cost – It can prove very expensive indeed to simply deliver the functionality users already had with the new system, even before pricing and implementing the new requirements
- Slow to Market – A rewrite almost always takes longer than expected as a lot of time is taken re-implementing existing functionality
- Understanding Existing Systems – Often existing legacy systems have evolved over many years. Poor documentation

and staff retirement or promotion makes it difficult to understand and replicate that functionality in a new system

- Integration – Unless it is a 100% replacement, new systems will typically need to be integrated with some existing systems. Data and business results must be consistent across systems

In many ways a **Package** solution would appear to solve many of these issues. A package will contain proven and tested business logic that delivers core functionality. There are certainly scenarios where a package can be an appropriate option, but for functions that represent unique and critical business value, this is unlikely to be the case.

A package will typically provide a sound implementation of the functionality for which it is designed. However, this functionality may only be a subset of the business functions provided by the existing system. As with a rewrite it is still necessary to understand the legacy system in order to determine what functionality will be covered by the package.

This process of analysis and understanding often reveals significant areas of functionality not covered by the package. This can lead to the costly and complex retention of so-called 'Orphaned Applications' that implement the missing functionality. This in turn can lead to an intricate and time-consuming integration requirement adding further cost and risk to the project.

Another consequence of shortfalls in the functionality of packages is the need for customization. Not only is this an expensive activity, but the ability to upgrade in the future is impacted, negating much of the benefit of using a package.

For many organizations the discussion is almost academic and the notion of even considering the large scale replacement of legacy systems is simply unrealistic. Existing enterprise applications embody an investment often built up over several decades, using business knowledge and optimization to meet the needs of the business. These systems deliver not only the core functions of the business, but also the aspects of the business that make it unique.

Modernization addresses new requirements through the extension of existing assets. By ensuring that these extensions do not touch or affect these existing systems, all of the distinctive competence and intellectual property of the application is retained. Modernization is not a zero effort or automatic approach. It is still necessary to understand existing systems before they can be extended and legacy code must be integrated with modernized elements of the system. Nevertheless, a modernization approach means efforts can be focused on new initiatives rather than recreating existing code, resulting in a very different type of project with these benefits:

- Low Risk. A modernization project can leverage the existing interfaces of the legacy application leaving the back end code unchanged while new interfaces into the code are created. This approach, combined with the much smaller scope of a modernization project compared to a Rewrite, significantly lowers the risk.
- Quick to Market. This more contained, low risk project speeds the time to market of the new business requirements, also potentially supporting an incremental delivery if required.
- Reduced Cost. A modernization project will typically cost much less than a rewrite
- Consistency. Since modernization reuses legacy business logic, existing systems and new initiatives will produce consistent results and require maintenance of only one set of core business functions.
- Leveraging and broadening existing skills. A modernization project leverages the skills of mainframe programmers in meeting new business requirements. These programmers often combine deep understanding of the existing systems with a level of business understanding gained over many years.

The remainder of this paper will look specifically at how Service Oriented Architecture (SOA) can be utilized as a strategic route to modernization.

Modernization through SOA

The extension or modernization of existing enterprise applications to new users or channels may often require that key functionality is presented through a more modern Graphical User Interface (GUI), through mobile devices or rendered through a browser based environment. This in turn may lead to technical requirements such as integration with J2EE or Microsoft .NET.

It's important to note at this stage that none of these requirements actually require the adoption of an SOA or

related technologies such as Web services. The choices of SOA and modernization are essentially separate decisions. Where a very specific or tactical solution is required SOA may not always be the appropriate choice. This section seeks to explore the benefits of using SOA as part of a modernization project. It examines how a tactical implementation can be leveraged to build a service framework that provides a strategic modernization that is agile, maintainable and adaptable to meet future requirements.

SOA provides a way of encapsulating business processes as standards based services. SOA can be tremendously valuable in a modernization project as it liberates a business process from the language it is written in and the platform it is running on. This level of reuse and extensibility can deliver significant business and IT benefits:

- Improved agility through the reuse of existing business processes. Future requirements can repeatedly reuse services previously exposed regardless of the platform of delivery
- Reduced risk as re-usable services delivering consistent results through proven business logic
- Reduced cost through a more efficient development process. Reuse reduces development effort while infrastructure, language protocols and complexities are hidden from developers, enabling them to focus on delivering business services quickly rather than debugging code and communication errors. Specifically, middleware software and integration costs are mitigated by the use of a single strategic architecture.
- The extension of existing mainframe transactions, without requiring any code changes, allows a quick and easy exposure of existing assets and their reuse across the enterprise
- Improved Time to Market through rapid development enabled by reusing existing business processes in a standard SOA infrastructure.

In summary SOA can unlock the business logic encapsulated within existing enterprise applications. This ability to access the functionality of these applications enables them to continue to fulfill their potential as critical strategic business applications within the organization.

Legacy to SOA Considerations

The creation of services from legacy components raises a number of questions and challenges, although the process can prove to be relatively straightforward. Some of the questions concern infrastructure issues that would arise in any SOA implementation, such as security, governance and scalability. Others are more specific, at least in part, to the notion of building services from existing code, as opposed to building a service from a service description from scratch.

Given that legacy systems were designed to be operated primarily from a green-screen interface or as batch operations, the business functionality within those applications is intertwined with the way in which it is accessed. Consequently, a fully automated approach of simply building services from each 'nugget' of business functionality within a multi-million line enterprise application is unlikely to prove either realistic or effective in creating services that are valuable or useable.

While there may be no silver bullet, legacy to SOA enablement is a practical and cost effective proposition. The most successful implementations combine the standard disciplines inherent in any non-trivial SOA project with an understanding of the legacy application and the interfaces available to it. A systematic requirements analysis provides structure and scope for an SOA enablement project and should include:

- Definition of the business (or strategic) requirement. Who will use the services? What interface is required by the potential service users?
- Deployment Platform Infrastructure. This may be as straightforward as the use of a corporate standard IBM Websphere platform already proven for service deployment. In other cases a separate evaluation of deployment choices, costs and benefits may be required. This may be a strategic choice, not governed by the fact that some services are extended from existing applications.
- Understanding of Existing Systems. Application Portfolio Management (APM) and Program Understanding and Analysis tools are invaluable for constructing the necessary information to facilitate the discovery process and make informed strategic and tactical decisions about

which applications, and which interfaces within the applications, to build as services.

- Additional or Replacement Usage. A legacy to SOA enablement exercise will often require that the existing application continues to operate unchanged while its functionality is utilized for services to an additional group or channel of users.
- Utilization of Mainframe Interfaces. Host applications provide a variety of interfaces that tools such as Micro Focus SOA Express™ can exploit, aggregate and tailor to build services that meet the business requirement.

Delivering Services with SOA Express™

The process of building services from an existing mainframe application can essentially be defined in four steps: Discovery, Definition, Generation and Deployment.

The significance of the **Discovery** stage will vary according to the type of project that is being undertaken. For example, a tactical and contained extension of a small number of services from a well-understood application will need only a brief analysis of the interfaces of the application in question. This can be provided within the SOA Express™ product itself. A more strategic and wide ranging project to modernize existing applications may well have to be preceded by an enterprise wide project using, for example, APM tools to assess assets across the enterprise. These tools provide the required level of information for appropriate decisions on retention, modernization, consolidation, etc.

As the process moves from Discovery to **Definition** and the service requirements become clear, it is necessary to determine how the SOA requirements match the mainframe code. The structure and transaction flow of the existing enterprise applications may well not match the business functionality or services that need to be exposed to meet new strategic requirements. SOA Express™ enables encapsulation of CICS and IMS transactions, screens and application workflow, and allows aggregation of multiple transactions into services without change to the original business logic. In this way, a degree of liberation is possible in building services from the original application structure and the captured service interfaces can be narrowed and tailored to render the required business function.

Once an interface has been captured and mapped, SOA Express™ provides a choice of **Generation** options including the ability to create Web services. SOA Express™ creates WSI standard Web services, including appropriate support files for deployment into the application server environment of choice. As a result, existing business processes can be delivered as services, across or outside the enterprise, without changing the back-end mainframe applications. The deployed Web services can be consumed without any knowledge or understanding of the back end mainframe infrastructure and technology.

SOA Express™ focuses on providing a choice of deployment platforms and making the **Deployment** process as straightforward as possible. Required middleware components and support files for the Web services are generated to represent the service interface described earlier. Finally, the generated services should be deployed. Services can be deployed to a variety of application server platforms including IBM Websphere (on or off the mainframe), BEA Weblogic, Oracle Application Server and .NET.

Conclusion

As with most trends in IT, SOA should be examined with a critical perspective to determine where it adds the most value. The strength of SOA lies in providing a standards based language and a platform neutral way of consuming and delivering business services, within and beyond the enterprise.

SOA provides a valuable mechanism for reusing the powerful functionality in legacy assets such as mainframe applications. COBOL remains a strong language for business processing, it is fast, maintainable and can process corporate data stores as or more effectively than most other languages.

Modernization using SOA can also help to overcome the insular and contained nature of 3270 mainframe applications. The ability to combine the retention of these valuable applications with the capability of defining and deploying services from them provides a compelling and cost effective alternative to packages or rewrite.

An SOA approach to application modernization delivers considerable business benefits. Time to market of new developments is accelerated, costs are kept to a minimum and risk is avoided.

Much of the hype around SOA has focused on building and deploying new code and services in J2EE or .NET. However for many larger organizations today, the real value of SOA could prove to be as a vehicle for unleashing the power and capabilities of their longstanding investment in mainframe systems.

About Micro Focus

Micro Focus provides innovative software that allows companies to dramatically improve the business value of their enterprise applications. Micro Focus Enterprise Application Modernization and Management software enables customers' business applications to respond rapidly to market changes and embrace modern architectures with reduced cost and risk. For additional information please visit www.microfocus.com.

www.microfocus.com

Micro Focus Worldwide

Australia..... 1800 632 626
Austria..... 0800 293 535
Belgium 0800 11 282
Canada .....+1 905 824 7397
France..... 0800 835 135
Germany..... 0800 182 5443
Ireland..... +353 1 469 3121
Italy..... 800 784 420
Japan..... +81 3 5793 8550
Luxembourg..... 800 23743
Netherlands..... +31 23 5689 138
Norway ..... +47 22 91 07 20
Sweden..... +46 8 21 99 21
Switzerland..... 0800 564 247
United Kingdom ... 0800 328 4967
United States..... 1 877 772 4450
Other Countries .. +44 1635 32646