Extend the value of your core business systems.

Transforming legacy applications into an SOA framework

Contents

Unshackling your core business systems

Facing tremendous pressure to improve responsiveness, global companies are hampered by the very technologies they rely on to run their businesses. Many of their core information technology (IT) systems have been in service for years—even decades. Developed on and optimized for legacy platforms, these systems are difficult to link to new technologies such as Web services and service-oriented architecture (SOA). They also lack the flexibility to adapt quickly to changing business requirements.

The convergence of SOA and mainframe technologies can help enterprises liberate these core business assets by making it easier to enrich, modernize, extend and reuse them well beyond their original scope of design. This paper will introduce the reader to the IBM vision for application modernization through SOA. It will explain the benefits of modular systems, define the steps that are necessary to develop a modularization strategy and roadmap, and describe successful techniques for execution that enable you to:

- *Leverage existing investments and assets*
- *Employ an incremental approach that lowers risk*
- *Maximize flexibility to align your application transformation to business priorities.*

The case for application modernization

Aging systems impose serious roadblocks to business agility. If a business process is going to adapt rapidly as the environment changes, so too must the underlying systems that support it.

Highlights

Although legacy systems embody many of the competitive advantages that enterprises need to succeed, they also are rigid and costly to maintain.

Legacy system value and challenges

It is reported that over five billion lines of COBOL code are added annually[1] on a base of over 200 billion lines of code.[2] Legacy systems support core business processes and provide crucial information for day-to-day operations. They contain business logic that provides competitive differentiation and institutional knowledge in the form of customer, product, supply chain and channel partner data. They also are optimized for performance and scale and embody many of the competitive advantages that enterprises need to succeed. They are, however, costly to maintain, rigid and difficult to use in new ways without unpredictable or even negative results.

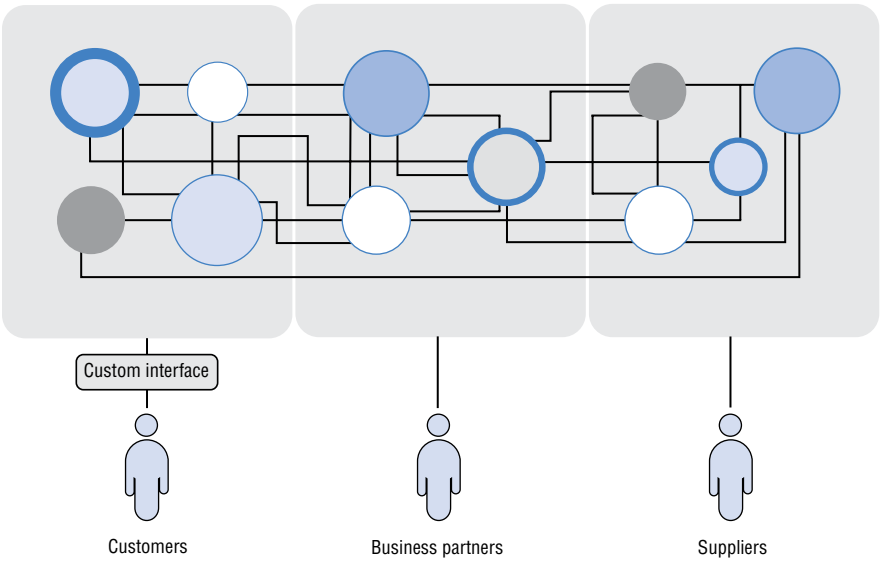

Figure 1. Tightly coupled, complex legacy systems with many point-to-point interfaces limit fast, flexible responses to business changes.

Highlights

What is needed is a prescriptive approach to modernizing this style of system engineering to one that fosters agility. The SOA approach has been shown to provide the value required by businesses today.[3]

Common options

There are options available to the IT executive for making legacy systems more responsive to business needs. The legacy applications can be rewritten, and aged systems can be replaced or retired. However, these options can involve significant investment in capital and implementation and training time. And the business may want to retain many, if not all, of the complex, custom functions that the older systems provide.

Rewriting or replacing aged systems can involve significant investments in capital, as well as implementation and training time, and can result in less robust systems.

Other approaches might include modification of the system in place, which will typically satisfy the immediate need, and may (if the skills and knowledge exist and are available) be delivered in an acceptable timeframe. These short-term approaches, however, do not move IT and business closer to the agility that's really needed.

In 2003, Robert C. Seacord described several options for modernizing legacy systems.[4] SOA concepts were not well known at that time, but the basic ideas, challenges and risks remain true today. In particular, the risks of total replacement, or "big bang" replacement, include a potential lack of deep knowledge of the business logic, as well as the risk that the new system might not be as robust as the old. Legacy systems have been tuned and streamlined over many years, and they most likely perform very well. Leveraging these existing assets in place, where possible, is the best approach to capitalizing on the investment and years of work spent perfecting them.

Leveraging existing core assets wherever possible is the best approach to capitalizing on the years of work spent perfecting them.

Highlights

An SOA, which provides an inventory of structured, loosely coupled reusable services, can help you tap into the business value in current systems and position IT for rapid future changes to the business model.

With an SOA, IT can more quickly add new functions and modify existing processes by simply assembling the services required to meet the new or enhanced business process model.

The SOA alternative

A better option, therefore, is to unlock the business function buried in the legacy systems. By crafting modular, service-based business processes supported by an SOA, you're able to tap into the business value in the current systems and position IT for rapid future changes to the business model.

With an SOA, you have an inventory of structured, loosely coupled services that perform a single function and can be choreographed to build business processes—yet the underlying implementation is isolated from the consumer. The user of the service is only aware of the function being performed, not of the details of how the function is implemented. With this service model in place, IT can rapidly add new functions and modify existing processes by simply assembling the services required to meet the new or enhanced business process model.

Within the construct of an SOA transformation, there are two principle options to unlocking business value and aligning IT function to business process. Leveraging defines the option of exposing function in legacy systems without rewriting the system. Repurposing defines the option of converting existing code to a new platform or enabling technology. An example of repurposing is to convert a COBOL program to Java™ code and execute it as is on a

Highlights

For more information on these benefits, refer to *The IBM Systems Journal*, Volume 44, Number 4, 2005: "Impact of Service Orientation at the Business Level"

researchweb.watson.ibm.com/journal/sj/444/cherbakov.html

process server. While this option does not constitute SOA, it does position the legacy code more favorably for service enablement, because the platform to which the code has been repurposed has better options for enabling business function for SOA participation.

Benefits of an SOA

An SOA can support powerful business capabilities. Overall, a business-centric, SOA approach delivers a host of benefits, including the following:

- *Reduced time to market*
- *Improved business alignment for growth*
- *Reduced costs*
- *Reduced business risk.*

Defining the concept of legacy transformation to SOA

Essentially, SOA and Web services provide a foundation for orchestrating business process in the IT domain. When combined with transformation of legacy systems, these services provide the structure for releasing deeply locked business functions out of legacy code and exposing those functions to the enterprise in a consistent fashion.

Highlights

Essentially, SOA and Web services provide a foundation for orchestrating business process in the IT domain.

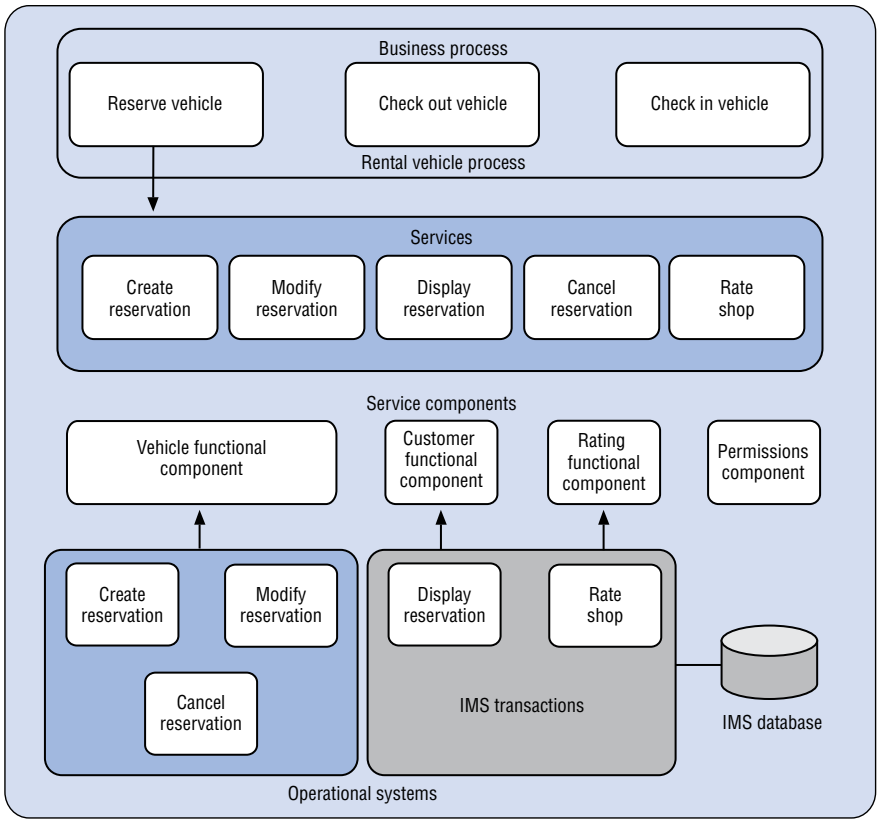

Figure 2. Leveraging Service Oriented Modeling and Architecture (SOMA), the Legacy to SOA method decomposes a business process and identifies services from the existing legacy assets that implement process functions aligned to business goals.

Figure 2 depicts an SOA implementation for a business process with high transaction volumes that requires tight integration with core business function, customers and vendors. The business objective of this particular system is to streamline the rental process to reduce costs and improve service. The SOA

Highlights

Using the Legacy to SOA method, and adhering to the SOA reference architecture, businesses can unlock the business function in legacy assets at the operational level.

An SOA-based legacy application modernization enables process redesign to add new functions and better connect with customers and partners at a lower cost.

reference architecture[5] is used to create the layers of abstraction necessary to expose services and choreograph those services in a business process. Legacy systems are resident at the lowest layer of the architecture, where business function is identified, componentized and made available to the service component layer as callable components. These components are then organized into services and exposed to the upper layers of the architecture for service choreography. Using the Legacy to SOA method, and adhering to the SOA reference architecture, legacy assets at the operational layer can participate in the SOA implementation, and the business can realize the value of business function locked in these systems.

An SOA-based legacy application modernization enables process redesign to add new functions and better connect with customers and partners at lower cost. SOA aligns IT function to business process function and significantly enhances agility by supporting modular systems. For the first time, businesses can easily construct and manage their business processes in real time without investing heavily in custom application development.

The IBM approach

Transforming the legacy code in a renovation project is not a simple matter. The key to successful renovation projects lies in proper execution. An unstructured, silo-based approach merely extends the legacy environment, with all its problems, to include some poorly conceived and implemented Web services.

Highlights

The IBM approach comprises five phases that support and promote best practices for legacy renovation.

Therefore, a prescriptive method for executing application modernization is needed. The IBM approach comprises five phases that support and promote best practices for legacy renovation:

- ***Strategy.*** *Build a set of goals and objectives, and develop a roadmap and set of strategies to achieve those goals.*
- ***Model.*** *Decompose the business process, and construct a service model that aligns with business process goals and requirements.*
- ***Assemble.*** *Create the services and choreograph the business process.*
- ***Deploy.*** *Install the physical environment for services and deploy the service containers.*
- ***Manage.*** *Ensure that the SOA continues to deliver the business value you need, and expand the service model as business requirements evolve.*

As shown in Figure 3, the IBM approach is iterative, as is each major phase.

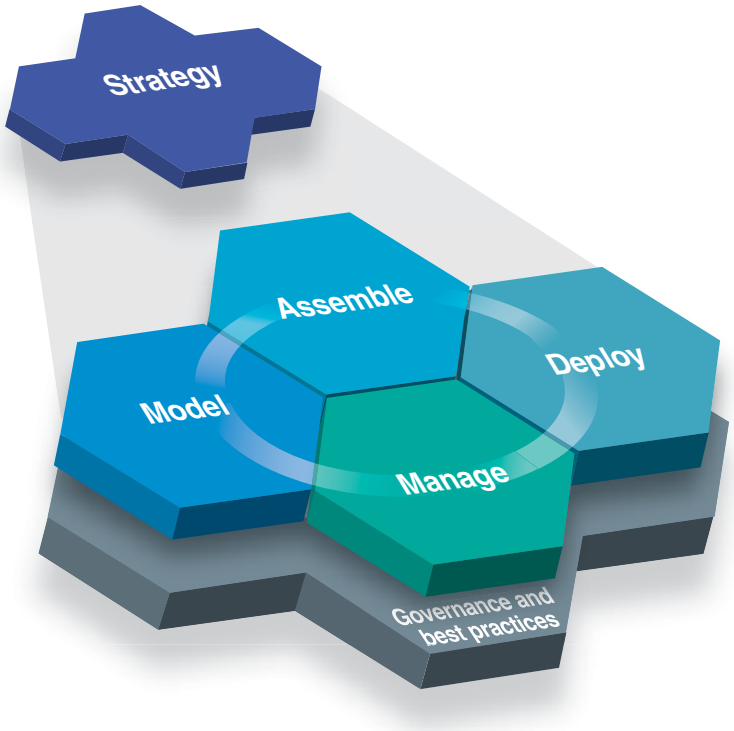

Figure 3. IBM offers a proven approach to application modernization that supports an evolving SOA.

Highlights

The strategy phase is critical to properly aligning business and IT goals with a long-term SOA approach.

Strategy

When evolving legacy systems to SOA, strategy can be narrowed down to a few simple principles.

- ***Understand the broad business and IT goals.*** *Incorporate the process of evaluating the current state, understanding business and IT goals and directions, and defining measurements and key performance indicators (KPIs) aligned to those goals.*
- ***Formulate a strategy.*** *Define the target state at the conceptual level that will allow the organization to achieve its goals, and identify what needs to be done to move from the current to the target state.*
- ***Create a plan for the future.*** *Prioritize required initiatives to transform the business.*
- ***Monitor the plan over time.*** *Collect and evaluate the effectiveness of the initiatives in providing improvements (as measured by the KPIs) in support of the stated business objectives.*

Highlights

The most important first step to a successful renovation project is assessment of the current environment and modeling of a desired end state.

Modeling is the process of identifying the service candidates required from the business process, and aligning them with the goals of the business.

Model

The most important first step to a successful renovation project is assessment of the current environment and modeling of a desired end state. Assessment refers to the process of evaluating the current state of the IT and business environments, determining the suitability of those environments for an application modernization project, and establishing the necessary steps to prepare the system for renovation to SOA.

Modeling is the process of identifying the service candidates required from the business process, and aligning them with the goals of the business. This process seeks to create what is known as a service model, a collection of services to be exposed and made available to those who are implementing business processes.

The IBM approach leverages the Service Oriented Modeling and Analysis (SOMA) method.[6] Our modeling activities align with the existing asset analysis and domain decomposition activities in the SOMA method, as well as with the service specification and service realization decision activities in the SOMA method. Service realization is a key step in the process of leveraging existing assets, where the technical implementation and location of components and services are decided.

There are special considerations required when identifying and specifying services from existing legacy systems. The top-down modeling activity considers the business process and seeks to break down the process into small pieces that can be aligned with a service. Generally, the legacy environment is not being considered during this step. Only the business process itself, looked at logically, is considered in top-down analysis. The objective is to isolate business function with respect to the overarching business process. Later, service candidates identified in the top-down decomposition of business processes will be reconciled with candidates from bottom-up modeling to create a comprehensive service model for the domain.

Highlights

Bottom-up modeling, which analyzes existing systems to identify service candidates, is a critical activity in application modernization projects.

Bottom-up modeling is a required and critical activity in application modernization projects. During this process, we analyze existing systems to identify service candidates. Some care must be taken to limit the scope of bottom-up modeling to those system components that are likely to yield services in the business domain targeted for renovation. Proper scoping of the project during the initial stages of the renovation, as specified in the method, will provide the necessary inventory from the legacy assets. The method is iterative so that the entire process can be refined to achieve complete use of available legacy assets.

The final result of this analysis is a service model that identifies candidate services aligned to the business process steps derived from the top-down model. The service model will have a mix of services: those that are well matched to existing assets, those that are a partial match and those from top-down analysis that have no match in the existing base and will need to be constructed from scratch. Once the service model is defined, a process called Service Realization Decisions is conducted. In this vitally important step, technical implementation details of components and services are decided, and allocation of components to services is performed. In this step, legacy systems are leveraged to create functional and technical components, or decisions are made to repurpose or replace existing assets to achieve the goals of the business process and service model.

In addition, at this point, data usage is rationalized within the service model, and an information architecture is defined for the business domain.

Highlights

The assemble phase involves designing and constructing low-level services from service components.

Numerous service-enabling techniques and associated technological components are used to complete the various steps of the assemble phase.

Assemble

Once a service model is created, we can embark on the task of low-level services design and construction from service components. Service components are the building blocks of the exposed services in the model. They execute the actual business function and can be made up of technical and functional components. This is the layer of enterprise components that is responsible for realizing functionality and maintaining the quality of service (QoS) for the exposed services. Functional components supply the business capabilities required, while technical components provide operational capabilities such as database access, queue support, authentication and authorization, error handling, auditing and logging.

Service components accumulate these technical and functional components and build a deployable service. The service itself does not contain any function. It is merely the interface. The service is what is exposed to the system, and it defines the inputs and outputs of the service component. Numerous techniques are available to extract business function from existing assets, to wrap existing assets to form a functional component, or to coordinate existing assets through messaging or databases to form a service component. During the assemble phase, the appropriate service-enabling technique and associated technological components are chosen. The IBM Legacy Renovation method links with other IBM methods for services development or services integration application renovation, providing the modeling and design input to development activities in these methods. These methods work together to form an end-to-end capability for SOA transformation.

Highlights

In the deployment phase, the renovated and constructed components and services are released into a production operating environment.

During the manage phase, you maintain the operational environment and the policies expressed in the assembly of the composite applications deployed to that environment.

The management phase also involves tuning the operations environment to meet the business objectives expressed in the business design, and measuring effectiveness.

Deploy

In the deployment phase, the renovated and constructed components and services are released into a production operating environment. In addition to the actual deployment of the composite applications, this phase involves creating the hosting, or run-time, environment. This means addressing any application resource dependencies, operational conditions, capacity requirements, and integrity and access constraints. You also need to carefully evaluate the techniques you will employ for ensuring system availability, reliability, integrity and efficiency. With scalable systems and software from IBM that support multiplatform IT infrastructures, you can optimize the SOA run-time environment to reliably run composite applications while providing the flexibility to make updates dynamically in response to changing business requirements.

Manage

Legacy renovations to SOA result in a mix of service and non-service components in the system, all of which need to be monitored and managed. During the manage phase, you maintain the operational environment and the policies expressed in the assembly of the composite applications deployed to that environment.

This phase also includes managing the business model—tuning the operational environment to meet the objectives expressed in the business design, and measuring success or failure to meet those objectives. This feedback loop requires the ability to easily track and measure composite application service performance against defined metrics (KPIs) and to adjust accordingly.

IBM system management software offers a composite application manager that enables the KPIs identified during the strategy phase to be monitored at run time. You gain real-time visibility into business processes for ongoing improvement and innovation.

Highlights

SOA governance is an extension of IT governance that focuses on the lifecycle of SOA services and composite applications.

SOA governance helps to mitigate many of the business risks inherent in SOA adoption.

Underpinning all of these lifecycle phases is a governance framework that provides guidance and oversight.

Governance

Adoption of an SOA raises new issues in IT decision rights and measurement and control. *SOA governance* is an extension of IT governance that focuses on the lifecycle of SOA services and composite applications. It enables organizations to maximize the business benefits of SOA, which include increased process flexibility, improved responsiveness and reduced IT maintenance costs.

SOA governance also mitigates many of the business risks inherent in SOA adoption by establishing decision rights, guiding the definition of appropriate services, managing assets, and measuring performance and realized business value. Effective SOA governance systems help you answer such questions as:

- *What are the common business services that are needed?*
- *Which services can be shared and under what rules and circumstances?*
- *Who makes a decision on whether a service can be accessible to other applications?*
- *Who owns the data, and is there agreement to allow the service access to the data?*
- *Who should fund the shared service? Who owns it?*
- *Who's responsible for initiating and approving changes?*
- *How is the business going to promote reuse of enterprise assets and shared business services?*
- *How do we measure the business value achieved through services creation and reuse?*

Defining the approach that works for you

When you're ready to modularize your legacy systems, having the right approach is critically important. There are multiple points of entry to successful application modernization. IBM has the business knowledge and technical skills required to help you define and execute the approach that is best for your business.

IBM provides flexible combinations of expertise, tools and technologies that enable SOA infrastructure and IT innovation based on your needs at each stage of the modernization process.

Our global consultant teams offer broad technical skills and extensive business experience in renovating systems to SOA. They facilitate active engagement across your organization to ensure broad input and buy-in on goals. Our data and code analysis and modeling tools include a model-driven integration framework that automates much of the renovation. You're able to improve the accuracy and reduce the cost, time and resources needed to discover, analyze and document assets and business process components within your legacy systems.

The advantage of working with IBM is that you gain access to all the right skills, methodologies and tools when you need them for a comprehensive, end-to-end solution. In addition, we can provide an integrated global delivery process that promotes the highest degree of quality with an emphasis on accountability, governance, project quality management and real-world metrics—all for a more competitive price.

For more information

To learn more about the IBM Legacy to SOA offering and other IBM Global Services offerings, contact your IBM representative or visit:

ibm.com/soa

IBM Global Services
Route 100
Somers, NY 10589
U.S.A

Produced in the United States of America
09-06

---

1 *From the Dustbin, COBOL Rises,* Stephanie Wilkinson, *eWeek,* May 28, 2001.
2 *Micro Focus Moves Mainframe Apps to .Net,* Daryl K. Taft, *eWeek,* November 4, 2003.
3 *IBM SOA foundation—providing what you need to get started,* ibm.com/software/soa/pdf/SOA_g224-7540-00_WP_final.pdf.
4 *Legacy Systems—Transformation Strategies,* William M. Uhlrich, Prentice Hall PTR, 2002, p.17.
5 *Modernizing Legacy Systems,* Robert C. Seacord, *et al.,* Addison-Wesley, 2003, p. 10.
6 *Service-Oriented Modeling and Architecture,* ibm.com/developerworks/webservices/library/ws-soa-design1.

G507-1950-00