



## **Service Oriented Architecture and application transformation.**

*IBM transforms an inflexible legacy order entry application  
into an adaptable, high-performance environment*

---

**Contents**

---

- 2 Overview**
- 3 *Identifying business-critical applications for transformation***
- 4 *Building on a framework for SOA***
- 6 *Starting with a complex as-is environment***
- 7 *Defining an incremental process for legacy transformation***
- 9 *Aligning IT processes and business intent***
- 10 *Realizing the to-be target***
- 12 *Shortening development cycles***
- 12 *Providing visibility through key performance indicators***
- 13 *Realizing tangible business benefits***
- 13 *The advantages of employing an integrated software suite***
- 15 *Leveraging lessons learned***
- 15 *About IBM***

**Overview**

IBM faces the same challenges as any enterprise in controlling cost and enhancing the functionality of critical applications. One such challenge was presented by the Customer Order Analysis and Tracking System (COATS). This order entry application is a complex batch system originally designed for a single IBM manufacturing plant. Over time, COATS grew into a shared order entry system serving more than 20 manufacturing plants with diverse needs worldwide.

A critical component of the order fulfillment process, COATS serves IBM customers, IBM Business Partners, IBM sales professionals and other internal organizations. COATS needed to keep pace with rapidly changing business conditions. However, the rigid legacy architecture of the underlying system was difficult to modify, compromising IBM's ability to incorporate new requirements. Batch-induced bottlenecks and conflicting data also affected business responsiveness, resulting in delayed shipments. IBM needed a more adaptable framework to meet the evolving needs of its business.

These characteristics positioned COATS as an ideal candidate for transformation as part of IBM's evolving enterprise application architecture. A key component of this enterprise application architecture is Service Oriented Architecture (SOA), which has been adopted by IBM as a strategic direction for meeting its ongoing requirement that IT solutions be able to quickly adapt to new business challenges posed by customers and competitors. IBM Global Business Services provided the consulting, methodological guidance, design and development to create an SOA-based legacy transformation solution for the COATS order entry application using IBM WebSphere® software and IBM system z™ hardware.

### Solution components

#### Software

- IBM WebSphere Business Modeler
- IBM WebSphere Integration Developer
- IBM WebSphere Process Server
- IBM WebSphere Business Monitor

#### Hardware

- IBM System z

#### Services

- IBM Global Business Services
- IBM Application Management Services

***COATS was selected for an SOA-based transformation because it is a business-critical application that requires frequent updates to accommodate new business initiatives.***

The architecture implemented for COATS has made it possible for the application to undergo an incremental transformation from a rigid legacy system to a highly integrated, dynamic environment – with no disruptions to the business. The benefits that this transformation has produced include:

- *Increasing flexibility to respond to changing business requirements and introduction of new products.*
- *Reducing order transaction processing time from 4 minutes to 10 seconds.*
- *Decreasing application development time and cost by more than 25 percent.*
- *Streamlining integration between systems, enabling real-time transactions that reduce discrepancies in delivery scheduling.*
- *Facilitating on demand changes to the run-time workflow via easily selectable rules.*

### Identifying business-critical applications for transformation

As part of its mission to implement a responsive, enterprise-wide application architecture, IBM charged its various business units to identify and propose projects that would benefit from an SOA-based approach. Of the applications brought forward as fitting this profile, COATS was the first one selected for the transformation.

COATS fulfills hardware orders from IBM customers, IBM Business Partners, IBM sales professionals and other internal organizations. The application sorts and prioritizes these orders, comparing them against manufacturing rules and the customer's installed hardware base. Then, more than 20 times each day, it routes material lists and instructions to appropriate manufacturing facilities. Because COATS is a critical piece of the order fulfillment process, it has to be frequently updated to support new initiatives, new product launches and new business opportunities. Frequent updates meant the COATS environment was complex and costly to maintain and keep current.

**Highlights**

***IBM's internal SOA mission is focused on ensuring that IT development is driven by business processes and gaining flexibility.***

IBM required a more adaptable framework that would facilitate rapid changes, increase application functionality and improve visibility into application performance. A flexible architecture – the type defined by IBM's internal application architecture strategy – was needed to support the ongoing transformation of COATS into a highly adaptable, real-time transaction processing application with increased resiliency.

**Building on a framework for SOA**

IBM's internal enterprise architecture strategy is built on a set of interrelated principles that are expressed in the IBM Framework for SOA, illustrated in figure 1. From its inception, IBM's internal SOA mission has had a primary focus on ensuring that IT development is driven by business processes. An overarching principle is to exploit the advantages of SOA to create an IT environment that uses services components to decouple business processes from “hard coding” in custom (legacy and new) and third-party applications. Building service components that can be reused across the enterprise helps IBM produce more business capabilities with fewer resources. New business solutions can be developed more quickly through the composition of already deployed services that have been designed for reusability.

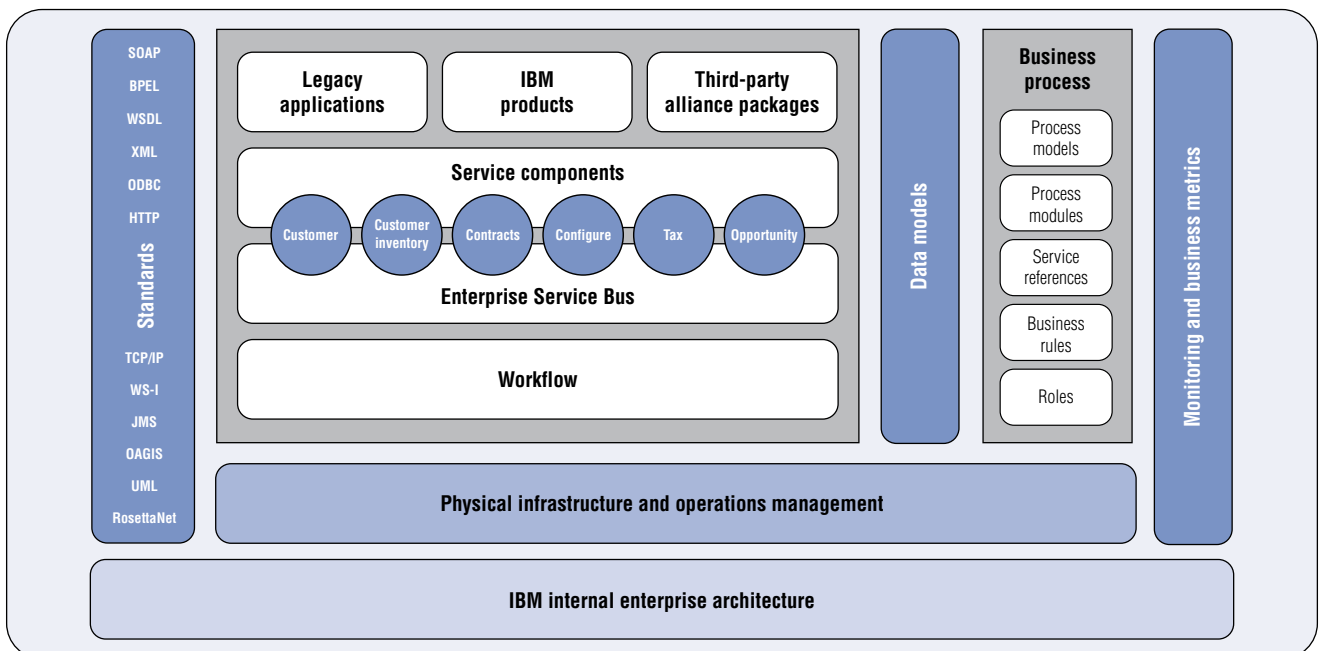


Figure 1: The IBM internal enterprise architecture solution is built on a set of interrelated principles, collectively called the IBM Framework for SOA.

---

Highlights

---

***By separating data and business rules from hard-coded applications, the architecture allows business rules to be readily altered in response to changing business requirements.***

The Enterprise Service Bus (ESB) plays a prominent role in this vision. It allows for the virtualization of services through a middleware intermediary that employs standard protocols to perform transportation services (e.g., protocols for secured and assured delivery), mediation services (e.g., routing, protocol and data transformations) and rudimentary forms of event management. Corporate Web services standards are part of the framework, defining rules for the use of these standards to help enable both SOA and reuse.

Workflow is the operational (IT) representation of business process design, interconnecting activities that involve either human staff or IT functions. IBM's architecture strategy calls for using workflow engines rather than standard coding to implement processes that must be readily adaptable to changing business conditions. Data models are separated from application architecture to allow information to be readily accessible across services and workflows. Likewise, business rules are external to hard-coded applications, residing in more accessible and easily updated forms such as business rule engines or processors. As a result, business rules can be readily altered in response to changing business requirements – and often by business analysts rather than by programmers.

---

**Highlights**

---

***An SOA implementation involves transforming process models into workflows that in turn use services.***

As shown in figure 2, an SOA implementation involves developing process models that can then be transformed into workflows, which in turn use services. Distributed components are exposed as these required services. This case study will describe how IBM followed this SOA deployment process to transform the COATS legacy application.

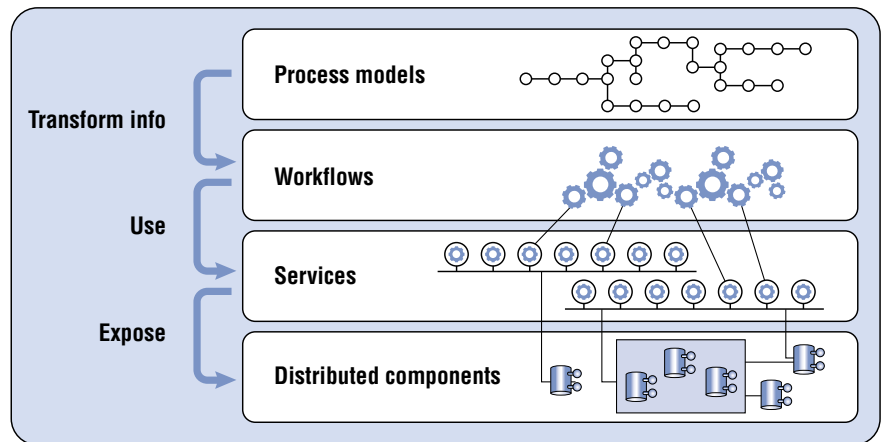


Figure 2: Relationship between models, workflows, services and components

**Starting with a complex as-is environment**

At the outset of the project, COATS was a complex 25-year-old batch system that ran dangerously close to capacity at peak times. It encompassed more than 1.4 million lines of code written in IBM Programming Language One (PL/1), IBM OS/390® Assembler, Java™ and other programming languages. With hard-coded business rules, procedures, logic and data access, COATS had grown in such a manner that it could not easily be adapted to address existing and emerging business demands.

---

### Highlights

---

***The COATS legacy environment required extensive recoding and multiple database updates to implement new functions.***

***As with many complex and enterprise-critical applications, a “rip and replace” upgrade would have been both cost prohibitive and disruptive to ongoing business—leading IBM to adopt an incremental approach.***

Changing a business process often required extensive recoding. Developers had to understand a complex set of point-to-point connections and component interdependencies to anticipate the effects of changes. To accommodate frequent order alterations, including automatic alterations performed by the scheduler system to meet customer delivery dates, multiple databases had to be updated and queried. With COATS’ rigid legacy code, considerable resources in both time and money were spent on new functionality development. Each version could take as long as six months to develop and could entail more than 8,000 developer hours.

COATS’ rigid underlying technology also inhibited application functionality. The application processed order transactions in batches, which could result in delivery-scheduling discrepancies among systems, potentially delaying shipments and revenue streams. The batch processing further limited throughput, making it difficult to support business growth and quarterly spikes in transaction volume. The application displayed a low tolerance for errors. Invalid hardware configurations or conflicting database entries were capable of crashing the system. IBM could monitor for transactional errors only by manually searching through the entire system.

#### **Defining an incremental process for legacy transformation**

With COATS processing an average of 2,500 requests and shipping well over 10,000 parcels a day, IBM made a business decision not to “rip and replace” the application. As with many enterprise-critical applications, a total at-once replacement was cost prohibitive and would have been too disruptive to ongoing business. In the case of the COATS project, funding was limited to the existing COATS maintenance budget. The development team had to balance how much budget could be applied to incremental migration to a new environment and how much could be applied to meeting new requirements for the existing legacy system. Additionally, the resource impact of maintaining dual systems for a period of time was a factor in the decision.

**Highlights**

*The strategy defined by IBM allows functions of the legacy application to be selectively migrated to the new environment along with new capabilities.*

The team followed an IBM-developed strategy for incremental migration of legacy applications into SOA, illustrated in figure 3. The process begins with defining the new service interface for new applications that need to interact with the legacy application. In the first phase of the migration, this new service interface would be supported by a “pass-through” component that simply routes the incoming request to the legacy application. In phase two, selected functions (gray squares) of the legacy application can be gradually migrated into the component, in addition to new capabilities (white squares) that normally would have been added to the legacy application.

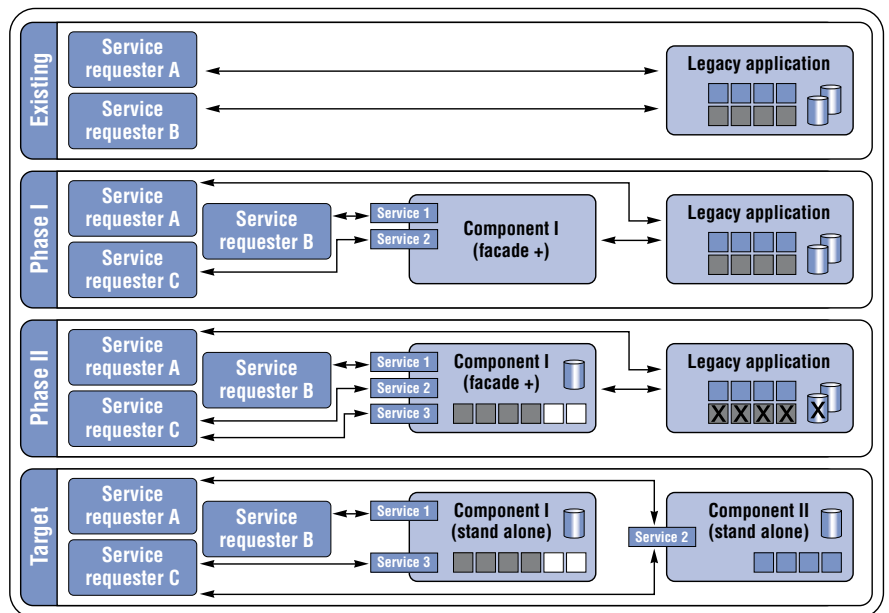


Figure 3: Incremental legacy migration strategy

In the target state, the legacy application has been completely migrated to the new environment, with functions residing in either the first component or a second component, which allows for decomposition into smaller and more manageable parts. The service interface abstraction minimizes the effects of any migration to the new component for the service requesters who are already using (or have themselves migrated to use) the new service interfaces.



---

**Highlights**

---

***Project phases for the incremental migration were defined by the needs of the business—particularly by new product launches.***

***The as-is model of the business process hard coded into COATS provided a starting point for identifying opportunities for improvement as well as existing assets that could be reused.***

Phases for the COATS project were defined based on business requirements—specifically, planned product announcements that would require a change in the order entry system. The time available between announcement dates defined a phase, and the content of the phase was based on the effort that could be completed by the end date. For example, if Product X was coming out on Date X, the end date for the phase would be Date Y. The work content of the phase would be the feature required to support the new product.

**Aligning IT processes and business intent**

The iterative transformation process started with decoupling business processes and data from the enabling IT infrastructure. Business analysts built a model of the as-is business process used in the original COATS application, illustrated in figure 4. With the model in place, the analysts were able to identify opportunities for improvement, propose changes and estimate a projected impact of the changes on the COATS application. In addition, the analysis of the as-is model helped identify existing assets for reuse, also depicted in figure 4.

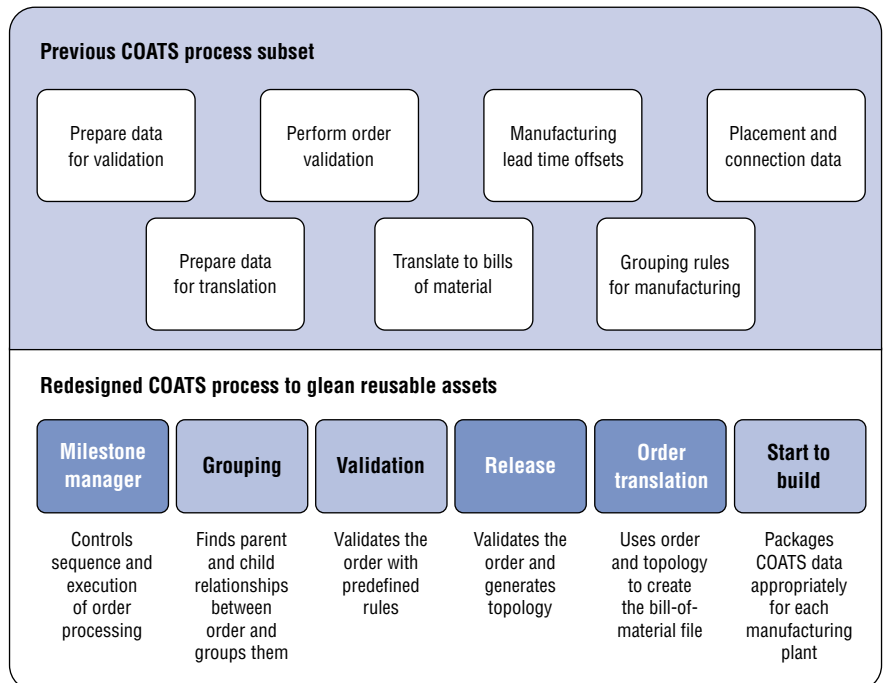


Figure 4: COATS' legacy processes redesigned into reusable assets

---

Highlights

---

***Service Oriented Modeling and Architecture provides a structured, repeatable method for identifying areas of the as-is model that would most benefit from transformation to SOA and developing specifications for the to-be service.***

To identify and specify business services, the team adopted a repeatable method that has been formalized to become Service Oriented Modeling and Architecture (SOMA). SOMA is a method used to enable target business processes through the identification, specification and realization of business-aligned services. SOMA creates continuity between business intent and IT implementation by extending business characteristics into IT analysis and architectural decisions. The analysis and modeling are not specific to IT technology or products, but they establish a context for making technology- and product-specific decisions.

To finalize the specifications for the business processes, services and components, the team used the project objectives considered critical by the business stakeholders to define the decision criteria for identifying areas of the as-is model that would most benefit from the transformation.

**Realizing the to-be target**

The IBM team created a flexible architecture that supports ongoing transformation of COATS. The SOA, shown in Figure 5, provides standardized connections and is highly adaptable with increased resiliency and real-time transaction processing.

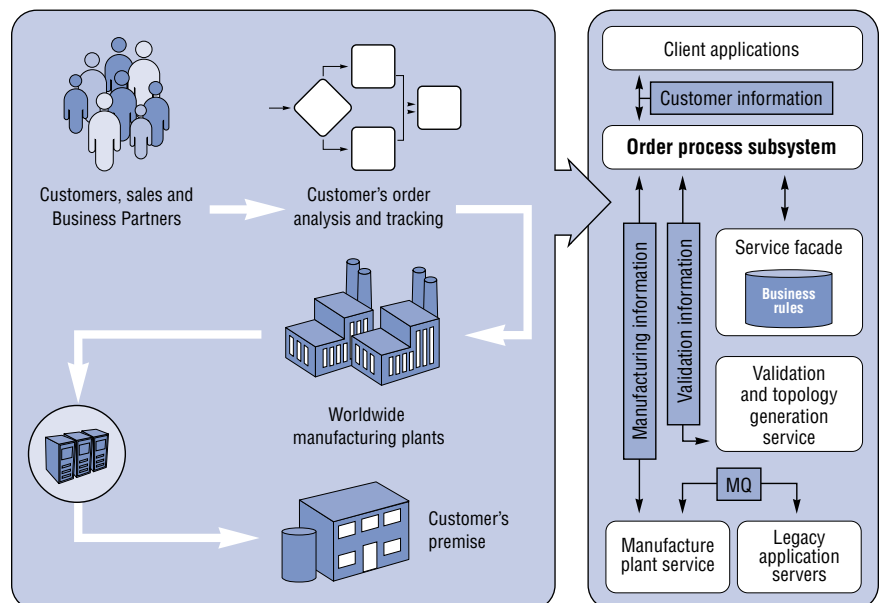


Figure 5: COATS' architecture overview

---

### Highlights

---

***To promote flexibility, scalability and reuse, the architecture for COATS externalized business rules, componentized functions and standardized connections between business processes and IT requirements.***

***Customer orders flow through a sequence of business services, such as validation and topology generation, before being converted to bills of materials that are forwarded to the destination manufacturing plants.***

***The SOA solution for COATS has reduced the development time and cost for new releases by 25 percent, producing an average per-cycle savings of US\$115,000.***

In this implementation of SOA, business rules are externalized and legacy system functionality is componentized to promote flexibility, scalability and reuse. This allows IBM to model, compose and choreograph standardized, reusable business processes and related IT services in a rapid implementation of easily adaptable, rule-driven workflows. The rules governing workflow can be quickly adjusted and implemented across systems for integrated, real-time transaction flow. Transaction flows are tailored dynamically to accommodate the parameters of individual customer contractual agreements and the specifications of individual customer orders.

The system flow begins with client applications that interact with the order process subsystem. The order process subsystem receives requests to process groups of customer orders and then forwards those orders to other services. It can handle multiple channels including business-to-business (B2B) messaging systems for both real-time and batch processing.

Upon receiving the orders, the order process subsystem marks them for validation and forwards the orders to the validation and topology generation service. This service categorizes the order and uses specialized validation services for the different categories of orders. The manufacturing orders that pass the validation step are then sent as a group to the manufacture plant service to be converted to bills of materials and forwarded to the manufacturing plants. This service interacts with service adapters for existing legacy or Business Partner applications that actually deliver the bills of materials to the destination plant.

#### **Shortening development cycles**

The adaptable SOA designed and implemented by IBM has made it possible for the COATS application to undergo an incremental transformation from a rigid legacy system to a highly integrated, dynamic environment that reacts with speed to business changes. IBM estimates that a wide range of improvements in the development cycle has decreased the time and cost of new COATS application releases by 25 percent. This shortens the cycle time from 26 weeks to fewer than 20 weeks, and it reduces the costs per cycle from US\$465,000 to less than US\$350,000.

---

### Highlights

---

***Automated process mapping, elimination of recoding, reusable design, enhanced collaboration and lowered reliance on programming skills all contribute to the development savings that IBM has realized.***

***Key performance indicators give business analysts the information they need to evaluate business processes and monitor business unit performance objectives.***

***The COATS transformation has reduced order transaction processing time significantly—from 4 minutes to 10 seconds.***

The savings can be attributed to several factors related to the productivity and collaborative capabilities of developers, including automated process mapping, elimination of recoding, reusable designs and streamlined communication among multiple teams. With abstract business logic, the development team can now change the rules that govern the run-time environment dynamically instead of spending months trying to pinpoint, reprogram and test changes in the legacy code. In fact, business analysts can compose business processes, reducing the need for developing skills in outdated programming languages.

#### **Providing visibility through key performance indicators**

COATS improvements extend beyond development efficiencies to enable business process management. After a deployable, working build of an application is developed, business analysts can monitor the live performance of the system by tracking key performance indicators (KPIs). The KPIs are defined by stakeholders prior to the implementation of the workflow and give a business view into the system by delivering information such as an average number of orders flowing through the system per hour; the number of invalid orders handled by the system; and the average time for an operator to resolve an invalid order. Ultimately, the information gathered from monitoring the application can be used to make decisions about changes in business processes to meet business unit performance objectives.

#### **Realizing tangible business benefits**

Several incremental releases of the new SOA-based solution for COATS have resulted in improved business performance and flexibility. In addition to reduced development cost and faster development turnaround time, the COATS transformation has reduced order transaction processing time from *4 minutes to 10 seconds*. With the application handling an average of 2,500 requests daily (about 300 per hour at peak times), this time savings will add up to approximately 150 hours daily speed-up in the handling of orders per day, once transformation is completed.

---

Highlights

---

***Sense-and-respond performance metrics enable IBM to reallocate system resources whenever transaction volumes outpace system resources.***

***Using an integrated software suite with compatible components can contribute to the success of an SOA project.***

The operational business processes surrounding COATS have also improved. The application can now handle an increased order capacity and has the ability to monitor and correct its own performance in response to mounting errors. Whenever transaction volumes outpace system resources, IBM can reallocate system resources accordingly. These sense-and-respond performance metrics and operational business rules can be adapted easily without changing the underlying code. In addition, the SOA implementation enabled IBM to offload many of the transactions from the legacy system, reducing the amount of data that is updated through batch processes. An increasing amount of real-time data transfer ultimately reduces order and delivery discrepancies among systems, improving customer service levels and boosting satisfaction.

**The advantages of employing an integrated software suite**

There is a plethora of software available to support business directives. However, choosing an integrated suite with compatible components is essential to a project's success. The service-oriented methodologies described in this paper were combined with IBM WebSphere software to enable integrated control over the COATS application—from design and development to testing and deployment—and from business process modeling to IT workflow modeling and implementation. An integrated software suite enables concurrent, collaborative and iterative development that reduces the amount of time it takes to launch new releases. The tools employed by the IBM team included the following:

- ***IBM WebSphere Business Modeler software supports the creation of reusable process models that correlate underlying IT, data, rules, tasks, roles and metrics in an integral operational design. It supports business process modeler identification and extraction of valuable business logic from the COATS legacy code, and enables that logic to be easily maneuvered as business requirements change.***

---

---

Highlights

---

---

- **IBM WebSphere Integration Developer software** enables IBM developers to directly import and efficiently transform the COATS business process operational model, from WebSphere Business Modeler software, into the reusable IT components, Web services and workflow needed for the COATS on demand capabilities. These new COATS components and Web services can interoperate with remaining legacy functions and other strategic applications.
- **IBM WebSphere Process Server software** directly imports the completed COATS IT model, including reusable components, Web services and workflow, forming an integrated, paperless development environment—from business process modeling through IT workflow modeling to production. This software provides automated, rule-driven threading of IT component Web services, customized to the requirements of each transaction.
- **IBM WebSphere Business Monitor software** provides the vehicle for real-time capture, tracking and reporting of the business process and Web services-enabled application operational performance.

***The success of the COATS transformation project depended in large part on employing a phased, incremental migration path that helped minimize business disruptions while helping secure early buy-in and support.***

#### **Leveraging lessons learned**

Employing a phased, incremental migration from legacy application to SOA environment was a key factor behind the success of the COATS transformation project. Following is a sampling of lessons learned throughout the transformation process of a legacy application to on demand business:

- *Use incremental implementation of services to achieve early buy-in and a nondisruptive migration path while managing expectations. Coexistence of services with legacy code supports the gradual transition of system functions to the new architecture.*
- *Align business and IT architectures through service modeling.*

- *Follow an iterative design and incremental development roadmap employing modeling, design and integration patterns.*
- *Create early Business Process Execution Language (BPEL) process models.*
  - *Do not include activities detail—they should be documented in use cases.*
  - *Create a data model at the same time you create the business process model.*
- *Ensure an appropriate skill base exists to accomplish the project.*
- *Design and validate Web services against a common data model.*
- *Establish milestones to interlock development artifacts. For example, ensure that integration specialists sign off on the business process model before it is converted to BPEL.*

#### **About IBM**

IBM is a global computer products and services company, with close to US\$90 billion in sales, approximately 320,000 employees and operations spanning more than 150 countries and including thousands of clients, Business Partners and suppliers. As one of the world's largest IT companies, IBM strives to invent, develop and manufacture the industry's most advanced information technologies, including computer systems, software, networking systems, storage devices and microelectronics. Its worldwide network of services professionals provides the extensive expertise clients need to create business value using IBM's leading technological solutions.

#### **For more information**

To learn more about the many ways IBM can help you build and manage an SOA, please contact your IBM representative or IBM Business Partner. Or visit us at:

[ibm.com/soa](http://ibm.com/soa)



© Copyright IBM Corporation 2007

IBM Corporation  
Software Group  
Route 100  
Somers, NY 10589  
U.S.A.

Produced in the United States of America  
03-07  
All Rights Reserved

IBM, the IBM logo, OS/390, System z and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

This case study illustrates how one IBM customer uses IBM products. There is no guarantee of comparable results.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.