Sustaining your advantage with business process integration based on service oriented architecture.

By Ed Lynch and Chandra Venkatapathy,
IBM Software Group

October 2005

Contents

Executive summary

Successful businesses are gaining strategic advantage by adopting flexible IT infrastructures and automating their business processes. Many organizations have adopted service oriented architecture (SOA) and business integration technologies to gain this flexibility and automation. IBM delivers a compelling business process management portfolio that enables companies to easily automate business processes based on SOA, while still using existing IT investments. This SOA-based process integration is made possible by a standards-based service component architecture (SCA) that renders IT as reusable services. Companies can create business solutions by wiring these service components and changing them dynamically with minimal skills and resources. This white paper focuses on the assemble and deploy aspects of business process management, using IBM WebSphere® Integration Developer.

Introduction

Business conditions and the competitive nature of an increasingly globalized and interconnected economy are causing many companies to reevaluate the fundamentals of their businesses. An On Demand Business — whose business processes are integrated end to end across the company and with key partners, suppliers and customers — can gain sustainable advantage over its competitors. The strategic advantage of an On Demand Business lies in its streamlined business processes — processes that increase the organization's flexibility and ability to respond quickly to changes in the marketplace.

To help your organization become an On Demand Business, you need to focus on two things: business design (business models and business processes) and underlying technology infrastructure. You must modify your business models and processes to focus on the core competencies of your business and eliminate inefficiencies inherent in the business model. For example, with an order-entry process, you need to eliminate redundant order processing by both manufacturing and finance. This can be easy to say but difficult to implement, unless you have tools that can render complex business processes in such a simple format that bottlenecks and duplication are easy to spot.

ON DEMAND BUSINESS™

Optimizing business models is an important step in becoming an On Demand Business, but it's not enough. You also need to focus on the underlying technology infrastructure. Your IT infrastructure must be capable of instantiating the new business processes and adapting quickly to any future changes in business processes.

Dynamic business models require a flexible IT architecture. SOA can deliver that flexibility. SOA and Web services are quickly becoming the new standards upon which integration applications are being built. Although SOA offers benefits in development and maintenance costs, flexibility is the primary goal. SOA differs from other object-oriented technologies because it separates interface technology from implementation technology, thereby rendering services, which are typically more independent than objects with respect to deployment. These services become the building blocks for composite intra- and cross-enterprise applications.

As an On Demand Business, you represent your business processes as a choreographed sequence of service invocations and instantiate these business processes in an integration server. From there, you can monitor the business processes and dynamically respond and adapt to emerging business conditions.

Barriers to business flexibility
Successful businesses understand that dynamic business models resulting from good business processes and flexible IT infrastructures lie at the heart of their sustainable competitive advantage. But in many cases, businesses find that they can't easily change their business processes without taking on risky IT projects — projects that often miss deadlines and run over budget. Instead of making changes in-house, many organizations acquire commercial off-the-shelf business applications to address short-term requirements. These applications come with their own application architectures and programming paradigms, leaving the business process hard-wired to the application. Businesses adopting this approach can end up with inflexible IT infrastructures and business processes that are disconnected and unmanageable. And worse, as they grow, businesses can find it difficult to make incremental changes or reuse these applications to address other business needs.

With its introduction of new business integration technologies and techniques, IBM offers an alternative. The IBM solution enables businesses to assemble solutions from reusable components — without a massive investment in skills. It provides businesses with the flexibility to respond quickly to changing business conditions — without the need to rip and replace their entire infrastructure. With the IBM solution, businesses can have a real-time view of their operations — without having to overhaul them. This compelling, comprehensive solution is built on SOA and is designed with business flexibility in mind.

Business flexibility with process integration based on SOA
SOA is about transforming your IT infrastructure into a set of flexible, reusable, standards-based services that can be quickly assembled together into business solutions. SOA differs from other object-oriented technologies in that it separates interface technology from implementation technology. This capability enables the integration developers who are building the business solutions to work with services without knowledge of the complex underlying implementation details. These services become the building blocks for composite intra- and cross-enterprise applications.

With the maturity of standards like Business Process Execution Language (BPEL) and the flexibility of Web Services Description Language (WSDL), process integration based on SOA has become a reality. Business flexibility comes from the inherent nature of SOA — the ability to assemble a set of services into a solution and deploy a composite application onto an integration server. If that integration server also has a process choreography engine, composing a business process becomes as simple as wiring services components together to create a process flow.

The WebSphere portfolio delivers a compelling set of offerings to address your end-to-end business integration needs (see Figure 1).
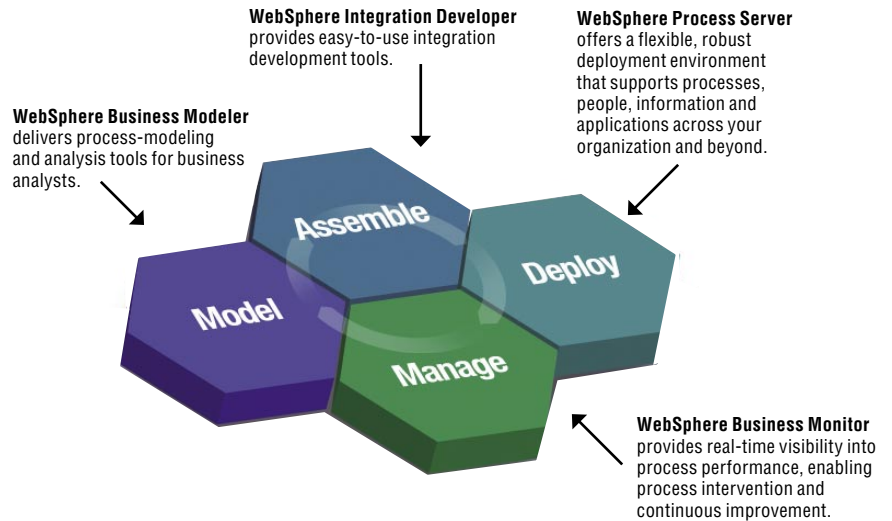
WebSphere Integration Developer
provides easy-to-use integration development tools.

WebSphere Process Server
offers a flexible, robust deployment environment that supports processes, people, information and applications across your organization and beyond.

WebSphere Business Modeler
delivers process-modeling and analysis tools for business analysts.

Assemble

Deploy

Model

Manage

WebSphere Business Monitor
provides real-time visibility into process performance, enabling process intervention and continuous improvement.

Figure 1. Comprehensive SOA offerings from the WebSphere portfolio

Transforming your IT infrastructure to SOA with an SCA

The power of SOA derives from its basic building block, a service. Services normalize access to underlying IT resources like commercial off-the-shelf applications, transaction systems like IBM CICS®, enterprise information system (EIS) assets like PeopleSoft and Siebel, Web services, workflows, structured data stored in databases and so on. Services can be aggregated to form new services, assembled to create new business solutions, choreographed to form automated business processes and updated dynamically without affecting in-flight business processes.

WebSphere software delivers a robust architecture for creating enterprise services — SCA. SCA is designed to help simplify the job of transforming your IT infrastructure into one based on SOA. Built on open standards like BPEL, WSDL and XML, SCA incorporates proven capabilities like workflow, mediation and adapters. The power of SCA lies in its simplicity, flexibility and scalability.

SCA delivers a simple conceptual paradigm that makes working with it very straightforward. This paradigm consists of:

- *A common invocation model that enables all services to be called in the same manner*
- *A standard component model that surfaces enterprise services in a consistent way, regardless of the underlying implementation details*
- *A common composition model that enables you to assemble applications and choreograph business processes by wiring together components*
- *A common user-interaction model that enables humans to easily interact with business process components*
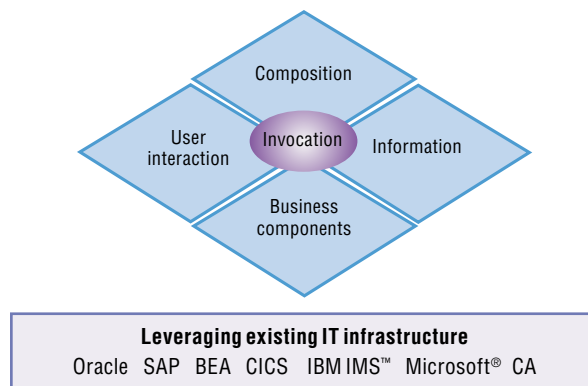

Figure 2. Service-component architecture

The implementation inside service components built on SCA varies from component kind to component kind. It can be a business process described in BPEL, a set of business rules, a Unified Modeling Language (UML)-like business-state machine, a human task, an interface to a transaction system like a CICS system or any other integration artifact. A service component built on SCA is simple but quite powerful. It is simple because it can be described, invoked and managed with a single integration framework. Integration components are created, assembled into composite applications and deployed onto the server (IBM WebSphere Process Server) with a single easy-to-use authoring tool (WebSphere Integration Developer). Business processes built using service components are powerful because they can be dynamically reconfigured after they are deployed. For instance, a business process built with SCA components can be rewired dynamically to new Web services (such as those provided by external service providers) without the need to disrupt the production environment.
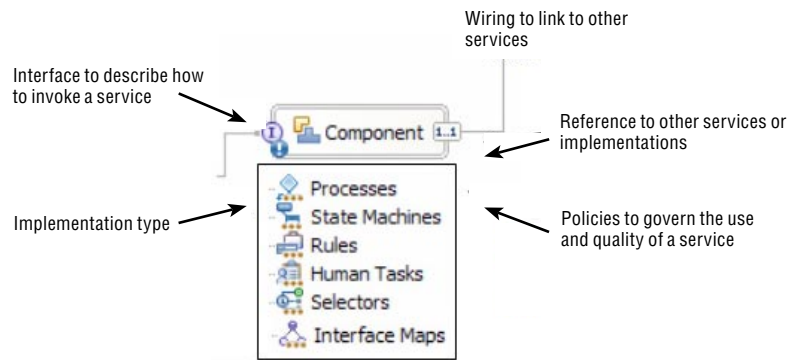
Figure 3. Service component built with SCA

Benefits of SCA

SCA simplifies the development and deployment of your business processes. You start with the set of components you have on hand. You construct other components by wrapping existing non-SCA assets in a component wrapper. And you construct those new components you need with simple, intuitive authoring tools. After you have the right set of enterprise service components, creating new integration solutions can be reduced to the simple act of wiring them together into composite applications. As a result, you can create, manage and assemble service components with minimal programming skill.

Deployment is equally straightforward. Deployment to a test environment is as simple as clicking publish. Using the auto-publish feature of the authoring tools, developers can deploy only those elements of the composite application that have changed, which can be quite valuable for debugging business logic later in the development cycle when the business logic has been finalized and the application is getting tuned.

Through SCA, you can build dynamic and adaptive business processes — which, together, can increase IT flexibility. Because interfaces are decoupled from underlying implementation, you can wire new service component implementations into existing business processes without the need to take the process out of production. The rich set of SCA component kinds provided with WebSphere Integration Developer and WebSphere Process Server — business rules, selectors, business-state machines, human tasks and so on — enables you to build business processes that can dynamically respond to changing business conditions. And because every asset built on SCA is fully reusable, development productivity can increase and time-to-deployment can decrease.

WebSphere Integration Developer and WebSphere Process Server deliver on the promise of SOA. WebSphere Integration Developer provides the authoring tool you need to construct business processes and compose business integration solutions. And WebSphere Process Server provides the server you need to automate your business processes and drive your business integration solutions.

Introducing WebSphere Integration Developer
WebSphere Integration Developer, Version 6.0 enables you to author your SOA-based services and choreograph them into business processes that can be deployed on WebSphere Process Server, Version 6.0. Like IT architects, integration developers, the primary users of WebSphere Integration Developer, have a broad understanding of how the system works but do not have — nor do they need — a detailed understanding of what each component does and how it is implemented.

WebSphere Integration Developer complements IBM WebSphere Business Modeler, Version 6.0, and can be used in conjunction with IBM Rational® Software Architect, Version 6.0 and IBM Rational Application Developer, Version 6.0. When combined into a single integrated development environment, these products provide a complete suite of tools to model, simulate, author and deploy composite SOA applications.
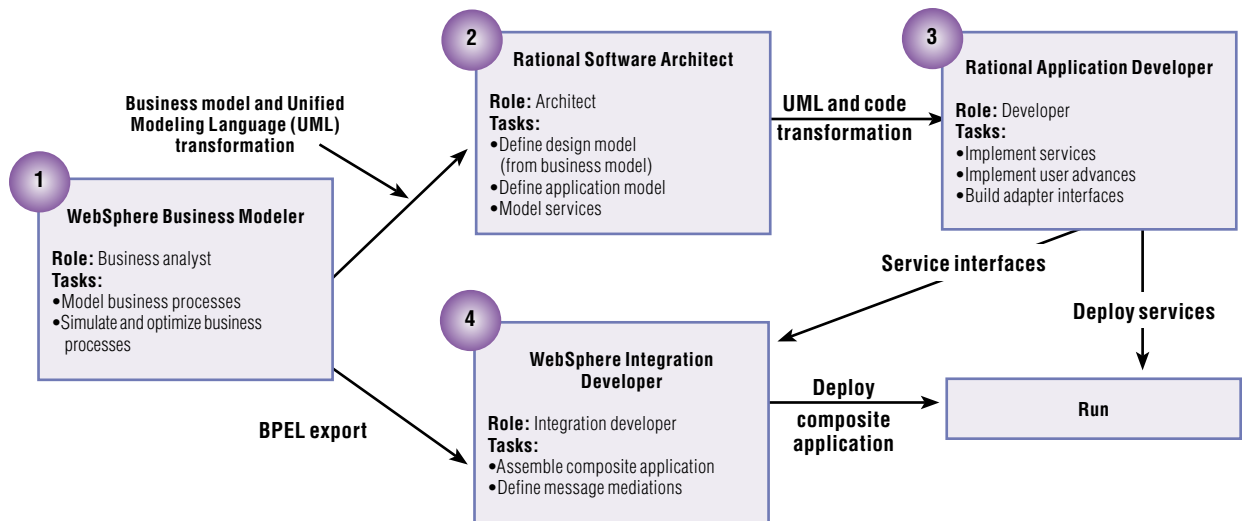
2 **Rational Software Architect**

Role: Architect
Tasks:
•Define design model
 (from business model)
•Define application model
•Model services

Business model and Unified Modeling Language (UML) transformation

UML and code transformation

3 **Rational Application Developer**

Role: Developer
Tasks:
•Implement services
•Implement user advances
•Build adapter interfaces

1 **WebSphere Business Modeler**

Role: Business analyst
Tasks:
•Model business processes
•Simulate and optimize business
 processes

Service interfaces

Deploy services

4 **WebSphere Integration Developer**

Role: Integration developer
Tasks:
•Assemble composite application
•Define message mediations

BPEL export

Deploy composite application

Run

Figure 4. Role-based development enabling optimal use of development resources

Introducing WebSphere Process Server

WebSphere Process Server is the run time for composite applications authored by WebSphere Integration Developer. Based on the robust Java™ 2 Platform, Enterprise Edition (J2EE), Version 1.4 infrastructure and associated platform services delivered by IBM WebSphere Application Server, Version 6.0, WebSphere Process Server includes a wide range of capabilities necessary to meet today's business integration challenges. When combined with IBM WebSphere Business Monitor, WebSphere Process Server enables you to monitor and tune in-flight business processes.

Business challenge: Integration is difficult and expensive

Consider this scenario: Your biggest competitor has just announced that it has moved all of its manufacturing operations to a country with lower labor costs, and you realize that your price points will soon be under attack. Your profit margins are already under pressure from an increasing number of players in the market. What are your strategic options? Do you follow your competitor's strategy and move your manufacturing? Do you change your business model and outsource? Do you change the nature of your offerings and aim at a different customer segment?

Or this scenario: You are the CIO of a major bank. The board of directors has just approved a strategic investment to establish a Web presence so your institution can more effectively compete with other banks that have been attracting your customers. You have been charged with building a portal for your customers that provides access to business applications that run in your IT shop. The portal must provide access to dozens of existing applications and related data, spread across business units. And you have to also add your business partners' applications to the mix. You have 20 business units and a dozen business partners. Your portal must be available on the Web 24x7. You've been given a budget and a deadline.

In many companies today, operations are disjointed, with internal departments operating as functional silos, hoarding information and operating independent of one another. As a result, redundant processes can proliferate. Flexibility is also reduced, hampering an organization's ability to adapt its business processes to address changing market demands.

To become an On Demand Business, you need to focus on two areas: *business design* and the *underlying technology infrastructure.* You need to break down functional silos from a business-process perspective. Extend your business reach to new customers and partners through new business models and new ways of interacting. WebSphere Integration Developer enables you to quickly spot and respond to these inefficiencies, even when they are buried in complex processes. Using visual presentation, the tool shows processes as components linked in a flow from end to end. The "Assembling components" section that appears later in this white paper gives you an idea of how easy it is to see a complex business process and spot areas for improvement.

In an ideal environment, your company would spend time and effort to model your existing business processes as well as any new business processes, examine those processes to determine where bottlenecks exist, modify the processes to achieve optimum efficiency, deploy the modified processes and monitor them to help ensure that they operate as planned. In the real world, you don't have the time or the budget to do that. But you can use WebSphere Integration Developer to model a process, either top-down or bottom-up, to integrate the various elements of your business process.

Besides aligning your business models with your strategic objectives, you must also help ensure that your underlying IT infrastructure can instantiate those business processes. As the CIO, you need to deliver an IT environment that enables people, processes and information to be integrated in a flexible manner. To address increasing budget pressure, you need to increase efficiency by delivering an IT environment that can scale to meet increased demand with minimal impact on capital expenditures. Finally, to extend business reach, you need to make your core applications and the information they control accessible by new business processes.

You know that business integration can be complex and skills-intensive. And the speed of technological change is ever increasing. Your applications and the information they control are probably sitting in islands of automation, many of them hand-coded to work together. Most companies that want to address these challenges see a jumble of components not unlike that in Figure 5.
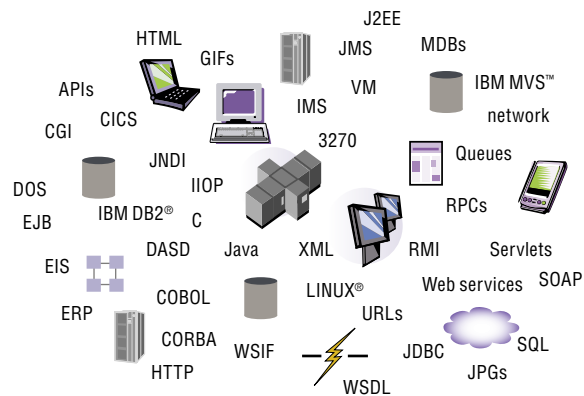
Figure 5. IT complexity as alphabet soup

Are you the only one facing this situation? No. It's a widespread problem that has been many years in the making. As a CIO, you have probably seen some or all of these three behaviors:

- *Business applications recreating function unnecessarily. Programmers write their own logic to perform memory management, database access or any number of things. This behavior leads to logic spread throughout every application that does the same thing (memory management), but probably in a slightly different way, which leads to maintenance and quality issues, and so on.*
- *Programmers building interfaces between service requesters and service providers in a tightly coupled way, when business applications need a service from either within their own code or from another business application. They do this to manage their dependencies and to manage the risk to downstream maintenance. But if the service changes, all requesters are affected.*
- *Programmers spending time creating and maintaining infrastructure code within one or more of your business applications, taking away precious time and budget that could otherwise be spent implementing business logic.*

All three of these behaviors impair agility and impede your ability to react to business change. Programmers don't choose to build things inflexibly; they work this way out of a need to do whatever it takes to keep your infrastructure operating. By adopting an SOA within your IT department and using an authoring solution like WebSphere Integration Developer, reusable components and composable services become artifacts that get produced as a matter of course.

WebSphere Integration Developer is designed to help your programmers (and, thus, your business) by taking SOA and turning it into a tool that your integration developer can use. WebSphere Integration Developer makes this possible by presenting applications and business processes as components. The implementation of the components remains hidden and the components interoperate through interfaces. As a result, the integration developer does not need to have extensive knowledge of the underlying implementation of the components to create an integrated application that uses them. Integration developers, however, are likely to have a broad-based technical knowledge in the integration field because they need some understanding of EIS service assets, business processes and applications coded in Java or other languages.

CIOs are faced with two common patterns of business integration: integration between business units and integration between enterprises.

Integration between business units
Technological advances are enabling — and the necessities of business efficiency are forcing — formerly autonomous business units to interconnect and cooperate as they are driven together by corporate goals. A sales unit wants to maximize the number of units in inventory while a manufacturing unit wants to keep the number of units on hand as low as possible, preferring just-in-time manufacturing. By integrating your sales department's customer relationship management (CRM) system with manufacturing's enterprise resource planning (ERP) system with your supply chain management (SCM) system, you can help reduce inventory costs, which can result in higher profit margins.

Integration between enterprises
The forces driving integration across business units also drive integration across enterprise boundaries, as partnerships and takeovers require all these parties to share applications and data. This integration between enterprises is driven by economic necessity. As virtual enterprises emerge, closer ties among corporations mean strategic focus on core competency and less time and administrative cost spent processing transactions.

The right technology enables enterprises to be linked in mutually beneficial areas. For example, an automobile manufacturer can set up an integrated supply chain with a tire supplier so that when the stock of tires is low, an order is automatically placed with the supplier. Regardless of whether you are reworking your business processes from the top down, or integrating your applications and knitting together business processes from the bottom up, WebSphere Integration Developer is a tool your developers can use. It doesn't matter whether you start by addressing your business design or by addressing your technology infrastructure; you must address both areas to complete the transformation to On Demand Business.

Developing a business solution: Point. Click. Integrate.

Essentially, WebSphere Integration Developer is a tool that separates business logic (components such as service assets, Web services, business rules, workflows and structured data stored in databases) from implementation. Integration developers work at a higher level of abstraction than other developers and spend their time manipulating the components to get the optimal business logic — your real business goal — rather than getting lost in low-level implementation. The top layer in Figure 6 is where your integration developer and business analysts spend their time.
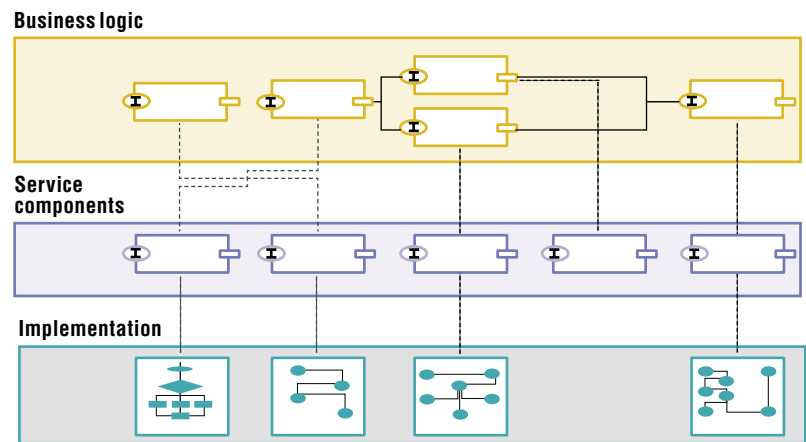
Figure 6. Layers of business integration

At the heart of WebSphere Process Server lies the SCA, which presents all elements of a business process in a service-oriented way. Everything in WebSphere Process Server is a component. These components have an interface and can be wired together to form a module (which has an interface), so you can change virtually any part of a WebSphere business-integration solution without affecting the other parts. For example, a human task can be replaced with a business rule without touching the business process. The next sections of this white paper discuss typical tasks an integration developer might perform using WebSphere Integration Developer.

Setting up project-related information

To manage its complexity, a project is divided into logical divisions called *modules*. Inside modules are components (see Figure 7), the basic building blocks of an integrated application. A component could be a business-state machine that handles incoming requests to your order-entry system, a business process for processing these orders or a set of business rules for providing your partners price discounts depending on the volume of orders they place.
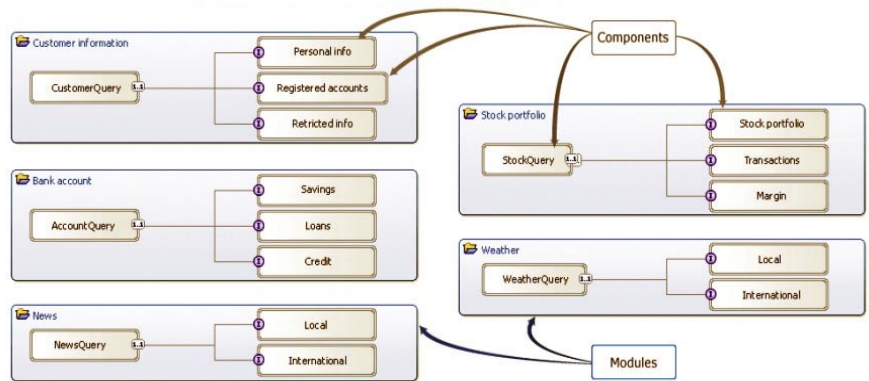
Figure 7. Modules and components

Creating and mapping business objects

Business objects contain business data used by components. They can be generic or application-specific, meaning they are used by a particular application within the solution. Relationships between business objects can be defined. The business object editor creates and modifies business objects (see Figure 8).
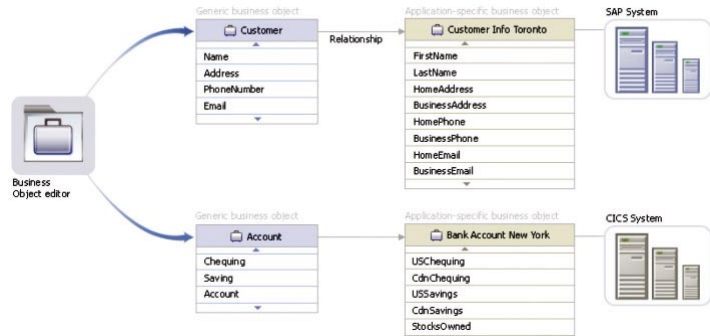
Figure 8. Business object editor

Creating interfaces

Components can be implemented in many ways, for example, as a Java application or a BPEL process. Interfaces are a common way to access any component independent of its internal implementation. The interface editor creates these interfaces (see Figure 9).
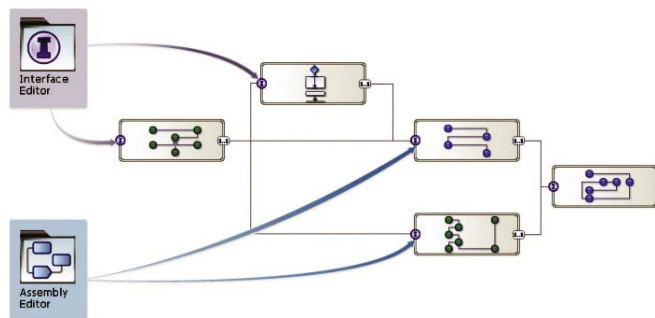
Figure 9. Interface editor

Creating a business process

A key function of an integrated application is to create business processes. The business process editor takes some of the complexity out of business-process creation by enabling you to create the process visually and then generating the underlying BPEL code for you (see Figure 10).
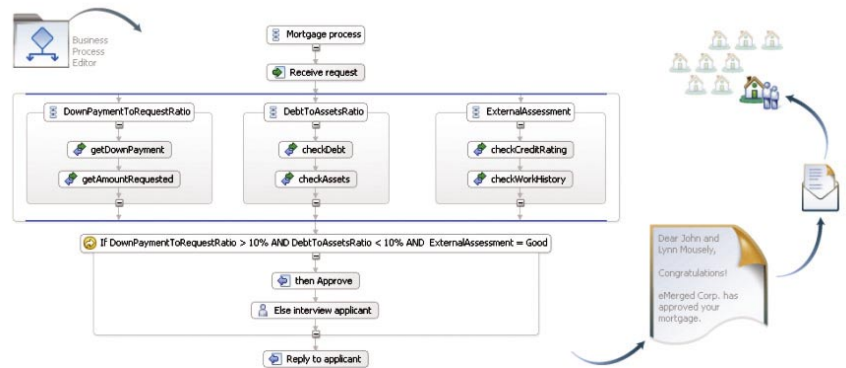
Figure 10. Business process editor

Specifying human tasks

People can be involved in business processes. One example might be to approve a purchase that is above a preset limit. The human task editor adds human intervention into a business process (see Figure 11).
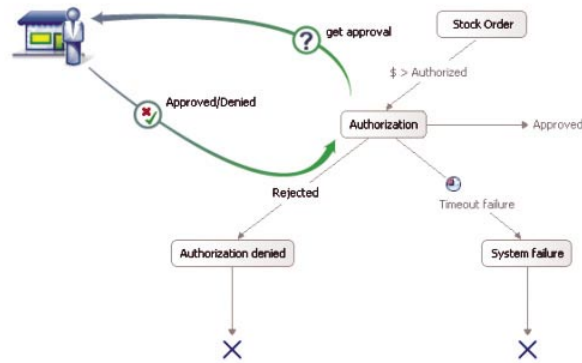
Figure 11. Human task editor

Creating a business-state machine

Some business processes can best be represented as business-state machines where the current state reveals what preceded it and what is to come. A business-state machine is an event-driven business transaction in which external activities trigger changes that guide the transaction from one discrete mode to another. Business-state machines are useful when modeling cancellations and timeouts. Cancellations can be modeled as interrupts that end long-running processes; timeouts help ensure that a state is not maintained indefinitely while waiting for an event that might never occur. The business-state machine editor creates business-state machines (see Figure 12).
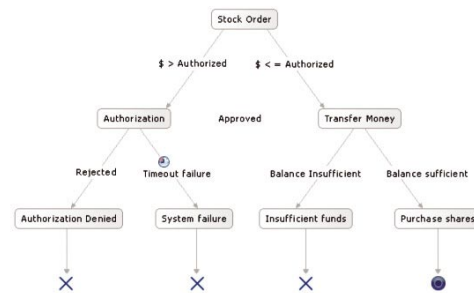
Figure 12. Example of business-state machine editor

Defining activities

Activities are routines used repeatedly by components. For example, an application could have standard financial routines to calculate mortgage payments, personal loan payments and car payments. Activities are built by the visual snippet editor and shared by components through a library (see Figure 13).
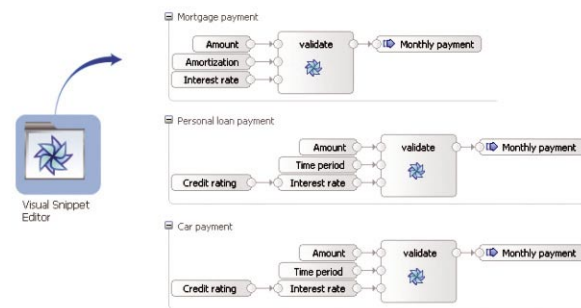
Figure 13. Visual snippet editor

Specifying business rules

Business services often use business rules to define their interactions in a business environment. For example, a rule could establish the amount of credit that customers can receive based on their credit ratings. Business rules are implemented by a business rules group. Within that rules group, they can be expressed as a set of *if-then* statements or decision tables (see Figure 14).
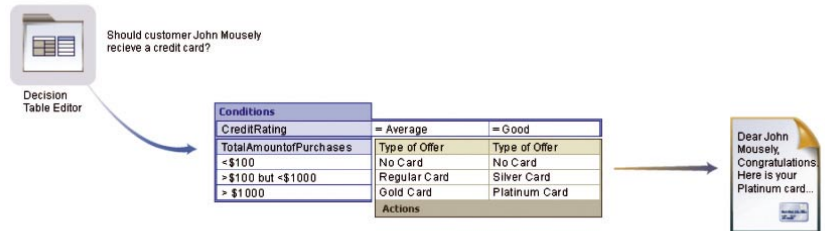

Figure 14. Business-rule decision tables

Importing and exporting applications

Applications on EISs are discovered by the enterprise metadata discovery wizard and then represented as components in the integrated application (see Figure 15). Imported applications act like a local application when invoked. Exported applications provide notification from an external application. Transforming and modernizing enterprise applications can be difficult and time-consuming. With this feature, you can selectively expose your enterprise application as part of enterprise business processes with speed and efficiency to make reuse a reality.
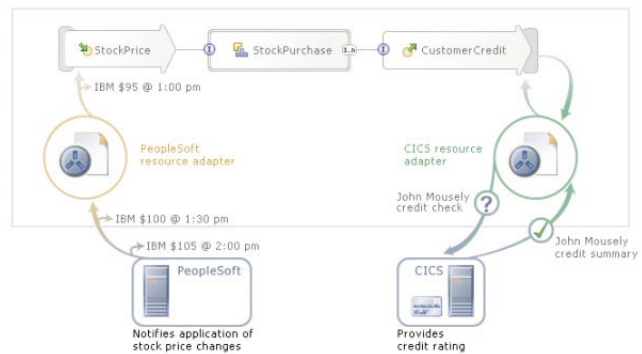
Figure 15. Enterprise metadata discovery wizard

Assembling components

Components are assembled into an integrated application in the assembly editor (see Figure 16). Alternately, components can be created within the assembly editor. This flexibility means you can implement business services first and assemble them later, or create shell components that you can assemble first and implement later.
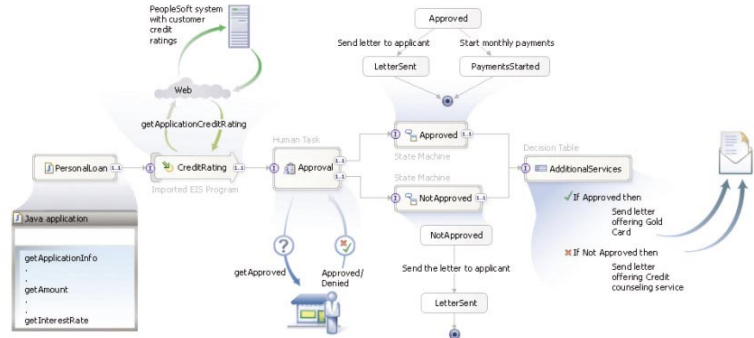

Figure 16. Assembly editor

Mapping interfaces

The output from one component might not match exactly the input requirements of a subsequent component. The interface mapping editor enables you to alter the data between components either to match the input requirements that follow or to allow other operations such as logging data (see Figure 17). For example, the output could be a set of fields of various data types, but the next component could be a human task with a person expecting a readable letter. The interface mapping editor could create the letter.
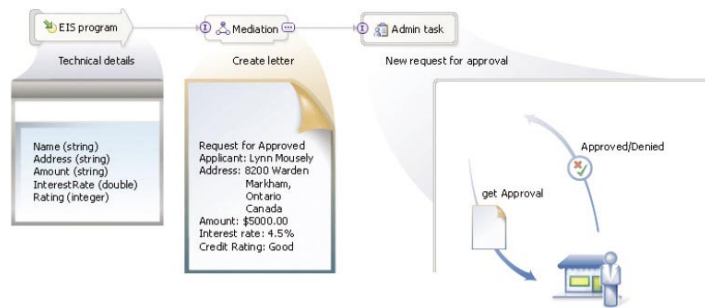
Figure 17. Interface mapping editor

Debugging a module or component

Using the integration debugger, you can debug at several levels of granularity, such as debugging the entire module or limiting yourself to a component (see Figure 18).
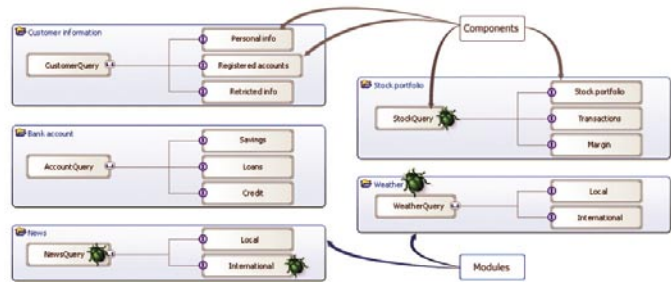

Figure 18. Integration debugger

Testing a module or component

Similar to debugging, you can test at several levels of granularity within the test environment (see Figure 19). You can test the entire module or narrow your testing to a component, which should reduce your test effort because it is focused on specific problem areas.
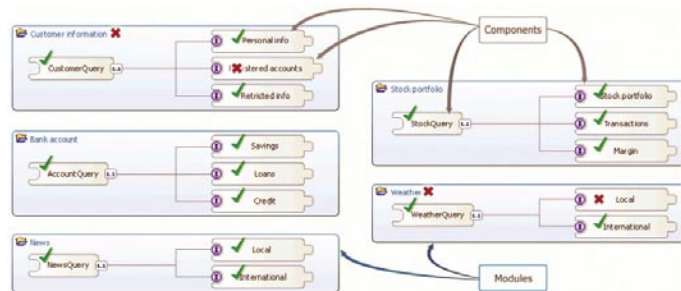
Figure 19. Test environment

Deploying a business-integration application

Deploying an application to a production environment is the end result of development in WebSphere Integration Developer (see Figure 20). Business processes are assembled from service components that represent underlying IT. These flexible service components can be easily deployed and managed in WebSphere Process Server. The deployment tools enable you to deploy your modules simply and easily, handling the low-level packaging of the artifacts for you.
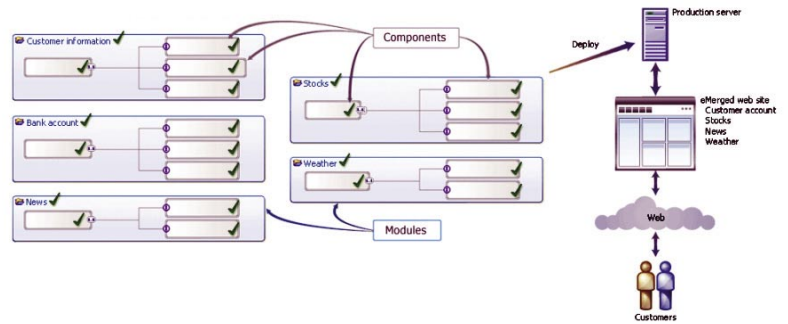
Figure 20. Deployment using WebSphere Integration Developer

Skills acceleration

Along with tools, editors, navigation assistants and property sheets, the WebSphere Integration Developer IDE delivers an extensive help system and a comprehensive set of skills accelerators (see Figure 21). The help system provides contextual help and a rich set of cheat sheets − recipe books to help you complete common tasks. Out of the box, the Welcome page introduces newcomers to the concepts and tasks used throughout WebSphere Integration Developer, and provides a series of access methods to help accelerate developers' levels of productivity. Some developers learn best by reading; others learn by watching videos and listening to teachers; and still others learn by playing with the tools and working with samples. Regardless of their preferred modality of learning, developers can find themselves productive quickly.
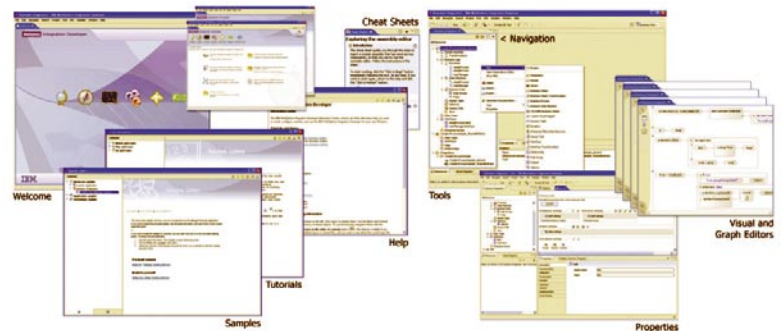

Figure 21. WebSphere Integration Developer includes a variety of educational tools to help you get started quickly.

Summary

Sustaining competitive advantage is about business flexibility and adaptability of business processes. This white paper is designed to demonstrate the benefits of WebSphere Integration Developer in helping organizations increase business flexibility by transforming to SOA.

Business flexibility is one of the key attributes for a successful business. But streamlining business processes has sometimes been a difficult, costly and in some cases, virtually impossible goal, for some companies. Barriers to business flexibility lie in the multitude of processes that are implemented and managed with a multitude of programming paradigms and applications that don't necessarily work together efficiently. Even though SOA delivers on the promise of On Demand Business, many companies have struggled to transform to SOA while searching for the right adoption model. Today, adopting the SOA model is made simpler by the WebSphere business process management portfolio.

Companies can't afford to replace their entire IT infrastructures. They need to use what they have, and combine those assets with new technologies. IBM helps make this goal possible through the standard-based SCA that transforms existing IT assets to reusable service components. You can 'assemble' solutions from service components with minimal skills and without the complexities associated with underlying IT.

WebSphere Integration Developer is the tool you need to create and manage business processes. It delivers the flexibility of assembling solutions by wiring the reusable service components together to build composite business applications. It helps deliver the business flexibility gained from changing and managing business processes deployed in WebSphere Process Server with advanced features like business rules, selectors and business-state machine.

For more information

To learn more about how IBM WebSphere Integration Developer can help you get started with SOA, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/integration/wid

To join the IBM WebSphere Global Community, visit:

www.websphere.org

IBM®