BPEL Cookbook: Best Practices for SOA-based integration and composite applications development

Editors
Markus Zirn
Harish Gaur

Authors
Stany Blanvalet, Jeremy Bolie, Michael Cardella, Sean Carey,
Praveen Chandran, Yves Coene, Kevin Geminiuc, Matjaž B. Jurič,
The Hoa Nguyen, Arun Puduval, Lawrence Pravin, Jerry Thomas, Doug Todd

Chapter 5
"Building Rich Internet Applications for Workflow and Process Monitoring"
In this package, you will find:
A Biography of the editors and authors of the book
A preview chapter from the book, Chapter 5 "Building Rich Internet Applications for Workflow and Process Monitoring"
A synopsis of the book’s content
Information on where to buy this book

About the Editors

Markus Zirn [markus.zirn@oracle.com] is a Senior Director of Product Management for Oracle Fusion Middleware. In this role he heads the Strategic Customer Program, where he works with Oracle's leading and most innovative middleware customers. He has been part of the Enterprise Software industry for more than 10 years, including roles as Vice President of Product Marketing and part of the founding team of QUIQ and as a Management Consultant of Booz Allen & Hamilton's Silicon Valley High Tech Practice.

Markus' passion for Service-Oriented Architecture (SOA) and BPEL stems both from practical experience designing and optimizing business processes as part of process reengineering projects and from being part of the advent of "software as a service" before web services became mainstream. He holds a Masters of Electrical Engineering from the University of Karlsruhe and is an alumnus of the Tripartite program, a joint European degree from the University of Karlsruhe, Germany, the University of Southampton, UK, and ESIEE, France.

Harish Gaur [harish.gaur@oracle.com] has more than 10 years of experience in the enterprise software industry. He is currently the Group Product Manager for Fusion Middleware at Oracle. In his current role, he works closely with strategic customers implementing Service-Oriented Architecture using Oracle SOA technology. Harish's expertise in Service-Oriented Architecture (SOA) draws from an extensive hands-on experience with Business Process Management (BPM) and Enterprise Application Integration (EAI).

Before Oracle, he worked as a Solution Specialist with Vitria Technology educating customers about the benefits of Business Process Management. Prior to that, he helped Fortune 500 companies architect scalable integration solutions using EAI tools like webMethods and CrossWorlds (now IBM). Harish holds an engineering degree in Computer Science and is an MBA from Haas School of Business, UC Berkeley. He lives in Fremont, CA with his wife Swati and son Agastya.
About the Authors

Praveen Chandran [Praveen_R01@infosys.com] works in the EAI practice of Infosys Technologies Ltd., focusing on platforms and technologies such as TIBCO, Vitria, and web services/BPEL.

Arun Poduval [Arun_P@infosys.com] also works in the EAI practice of Infosys Technologies Ltd., specializing in similar technologies. Praveen Chandran and and Arun Poduval worked together on Chapter 1.

Lawrence Pravin [Lawrence.Pravin@SierraAtlantic.com] is the Product Manager, Process Integration Packs, Sierra Atlantic Inc. Process Integration Packs deliver end-to-end business process integration solutions between enterprise applications. He has over 10 years of rich experience in packaged applications, and has deep integration expertise with Oracle, PeopleSoft, Siebel, and SAP applications. Lawrence Pravin worked on Chapter 2 for this book.

Yves Coene [Yves.Coene@spacebel.be] currently works for SpaceBel SA in Brussels as Project Manager. He has 15 years of experience in aerospace software projects such as Ariane 5, the International Space Station, F16 MLU, and various other projects for the European Space Agency. Since 2001, he and his team have been responsible for the SSE project for ESA in Frascati, Italy.

The Hoa Nguyen [the.hoa.nguyen@spacebel.be] currently works for the SDC subsidiary of SpaceBel SA in Brussels as senior software engineer. His main interests are J2EE, web services, and workflow development with BPEL. Since 2001, he has been one of the lead engineers of the SSE project team at SpaceBel and is also in charge of SSE software releases and on-site SSE software installations at ESA. The Hoa Nguyen and Yves Coene contributed Chapter 3.

Kevin Geminiuc [kgeminiuc@yahoo.com] currently works as a senior software architect in Denver. Over the last 15 years, Kevin has worked as a systems architect, technical manager, developer, and hardware engineer. Kevin's technical interests include SOA, RFID, AVL, and genetic software. Kevin contributed Chapter 4 for this book.

Doug Todd [dtodd@enterrasolutions.com] is CTO of Enterra Solutions in Yardley, PA. He has more than 20 years of experience in systems architecture, applications architecture, systems integration, and applications integration with major corporations. Todd is responsible for Enterra's overall IT strategy and tactical implementation, enterprise information architecture, and technology product offerings. Doug Todd worked on Chapter 5.
Jerry Thomas [j thomas@centerstonesoft.com] is Chief Architect at CenterStone Software, which helps many of the world's largest organizations automate and manage their real estate, facilities, personnel, assets, leases, and workplace operations more efficiently. Thomas focuses on CenterStone's enterprise workplace management product and web services, BPEL, and system infrastructure. Prior to CenterStone, Thomas worked as a consultant and held principal development positions at Riverton, ONTOS, and Hewlett-Packard. Jerry Thomas wrote Chapter 6 for this cookbook.

Sean Carey [s c a r e y@s p s c o m m e r c e .c o m] is a Software Architect at SPS Commerce, a leader in hosted EDI. Sean has over seven years of experience in mission-critical e-commerce implementations, and 15 years of industry experience in software design. Sean Carey gave us Chapter 7.

Matjaž B. Jurič [m a t j a z . j u r i c@u n i - m b . s i] holds a Ph.D. in computer and information science, and serves as a content developer and consultant for the BPEL and SOA consulting company BPELmentor.com. He is the author of the book Business Process Execution Language for Web Services from Packt Publishing (ISBN: 1-904811-18-3). He is also the co-author of J2EE Design Patterns Applied, Professional J2EE EAI, Professional EJB, and NET Serialization Handbook, and has contributed to Web Services Journal, Java Developer's Journal, and other publications. Matjaž B. Jurič worked on Chapter 8.

Michael Cardella [m c a r d e@q u a l c o m m .c o m] is a Staff Engineer at Qualcomm CDMA Technologies (QCT). Michael works in the custom applications development team, primarily on web-service- and business-process-related applications. Previously he served as Principal Architect for a leading web services security and management product.

Jeremy Bolie [j b o l i e@q u a l c o m m .c o m] is a Senior IT Manager at QCT, managing the custom applications and Documentum development team. Jeremy has over 10 years of experience with Java and Oracle technologies, and has been involved with web services and Service-Oriented Architectures since the late 1990s. Jeremy Bolie and Michael Cardella worked together on Chapter 9.

Stany Blanvalet [s t a n y . b l a n v a l e t@j a i s y . b e] is a BPEL and J2EE consultant. Previously, working as a Java EE architect, Stany introduced and administered Belgacom's BPEL-based DSL provisioning application, a mission-critical BPEL production system. He is a contributor to the Jaisy-ORABPEL Interface project, an open-source JMX monitoring tool for Oracle BPEL Process Manager. Stany Blanvalet contributed Chapter 10.
Create a real-time workflow and advanced process activity monitoring dashboard by extending Oracle BPEL Process Manager APIs.

More and more organizations are automating their key business processes to increase operational effectiveness. However, even automated processes require manual interaction for two important reasons: to advance a process to the next step (workflow), and to provide real-time process visibility for end users (process monitoring).

Consider a business process for opening a new bank account. First, the customer provides necessary details (name, address, SSN, initial deposit) to open the account. Once the process kicks-off, the customer will want to track the status of the request and respond to any additional queries from the bank. This process requires workflow to enable customer participation, and process monitoring so that the customer can track request status.

Oracle BPEL Process Manager facilitates basic workflow capabilities and process activity monitoring. But just as important, by extending its exhaustive API interfaces for interacting with processes, instances, and workflow, it is possible to build a single, rich internet application (RIA) that enables advanced workflow and process activity monitoring. This advanced workflow capability could enable zero-latency communications between user and process, whereas advanced process activity monitoring could transmit real-time process status information to the workflow so that appropriate actions could be taken.

In this chapter of The BPEL Cookbook, I will present a sample business scenario that requires real-time workflow and advanced process activity monitoring capability. We will then explore the architecture of a Macromedia Flex-based RIA that leverages the BPEL Process Manager API to meet those goals, and then learn how the RIA initiates processes and integrates workflow. You will also learn how to visually depict the process paths available as well as those actually taken by the current instance. (This functionality is very valuable when examining processes in retrospect.) Finally, you will get an introduction to audit trail tracking using the API interfaces.
RIA Background

An RIA is a hybrid of web application and traditional desktop application, typically built using "interactive" technologies including JavaScript, Flash, Java Applets, XML User Interface Markup Language (XUL), Extensible Application Markup Language (XAML), JavaServer Faces, and Swinglets, often using an Asynchronous XML and JavaScript (AJAX) approach. RIAs can potentially improve the usability and effectiveness of online applications as well as increase developer productivity. According to Gartner Research, by 2010, at least 60% of new application development projects will include RIA technology of some kind.

Our example here comprises a BPEL workflow interface and process monitoring console that offers end users a unified, feature-rich dashboard. Note that it is important to differentiate this application from the Oracle BPEL Console or true Business Activity Monitoring (BAM); although a certain level of functionality overlap exists, the RIA target user community is vastly different than that for the latter two applications. The BPEL Console is more suited for administrators and developers who debug, deploy, and maintain process instances, and BAM appeals to process owners (executives, LOB managers, and so on) and IT managers who want to measure key performance indicators (KPIs) and then take action to refine the process. In contrast, the example RIA application described here is designed with end users in mind: customers, partners, or suppliers who want to track the status of a specific business interaction and don't care about KPIs or process improvement.

Sample Process

The sample business process, "New Account Request", is a multi-organization process that creates accounts for a financial service intermediary, its financial partners (insurance companies and their financial advisors), mutual fund companies, and clearinghouses. Here we will focus on the first two steps of this process.

1. The process accepts the customer's SSN and then extracts and displays customer information (name, address) from the database.
2. The customer verifies this data, adds the deposit amount, and clicks on Submit.

Clearly customers will need a workflow interface to enter the SSN, verify name and address, and input the deposit amount. In addition, they will need the ability to visually monitor the progress of the request.

In the next section, I will offer an overview of the development aspects of building such an application. Later, you will see how the New Account business process executes using this application.

Building the Application

As explained previously, you will design this application with two primary goals:

- Provide a rich unified user interface for users to track the process status and provide input as and when required (RIA front end)
- Offer advanced process monitoring (BPEL Process Manager API back end)
RIA Front End

The RIA user interface is the "face" of the application; it runs and manages the New Account business process. This presentation layer is built using Macromedia Flex technology and the ActionScript scripting language. It accepts the user input and passes it to the BPEL process. It also displays the current status of the process.

The following figure shows the RIA front end for the New Account business process:

Let's review its four main components and their key capabilities:

- **Workflow Bar**: The Workflow Bar allows the user to view the workflow steps in the New Account process at the business level. Every step in the Workflow Bar is exploded in the BPEL Panel.

- **BPEL Panel**: The BPEL Process Panel offers an exploded view of every step of the New Account process. The panel dynamically reads the process model and the instance audit trail of a given process instance to combine the process paths available with the paths actually taken by the current instance. The panel also polls the audit trail to determine what action is pending, visually delineating where the current process is at that time. Any nodes of the process previously executed can be clicked on to view their respective payload. This functionality gives operations and compliance managers real-time visibility into the past, current, and potential future state of the current process.


- **Workflow Forms**: Workflow Forms correspond to the workflow steps within the New Account BPEL process. This Flex "view stack" communicates with the audit trail of the business process, changing the view stack form to match the context of the BPEL process.

- **Zones**: Zones provides alerts and feedback to assist the workflow and process activity users perform their respective jobs (not discussed here).

These panels offer superior usability experience and interact with the BPEL API. Now let's see which back end API functions are leveraged to execute the RIA front end.

**BPEL Process Manager API Back End**

The `BPELService` class facilitates interaction between the RIA front end and the process activity monitoring capabilities in the BPEL Process Manager API. As shown in the figure that follows and described in the subsequent table, `BPELService` uses five interfaces within the BPEL Process Manager API for process, process instances, and tasks to deliver on the promise of workflow-enabled process activity management and monitoring.

![BPEL Process Manager API Back End Diagram](image-url)
Now that you understand the basic components of RIA and BPELService, let's see how they work together to offer a rich end-user experience.

**Running the New Account Process**

As explained previously, the customer is first presented with an input screen in the workflow panel to enter the SSN. As soon as the user clicks on Submit, a "New Account Application" XML document is created using ActionScript and passed to the BPELService server-side Java class via Flex's Remote Object Architecture. (This enables Flex to speak directly to Java objects such as BPELService.) The Java component uses an instance of the IDeliveryService interface to initiate the BPEL process. The initiateProcess Java method, shown below, then returns the instance reference ID of the newly created business process instance to the Flex client. This reference ID is used in subsequent operations for targeting the correct business process instance.

```java
/**
 * This function initiates an instance of a BPEL PM process based on the
 * process name. The function sets the function id and the reference id for
 * future use.
 * @param xmlRequest
 * @param strBusinessProcess
 * @param strBusinessProcess
 * @return the initiating response message for the process
 */
public String initiateProcess(Document xmlRequest, String strBusinessProcess) {
    System.out.println("Initiate" + strBusinessProcess);
    IDeliveryService deliveryService = getDeliveryService();
    NormalizedMessage nm = new NormalizedMessage();
    nm.addPart(PAYLOAD, xmlRequest.getDocumentElement());
    NormalizedMessage responseNm = null;
    try {
        responseNm = deliveryService.request(strBusinessProcess, PROCESS, nm);
    }
    return responseNm;
}
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>Allows the user to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDeliveryService</td>
<td>Invoke instances from processes deployed on a BPEL process domain</td>
</tr>
<tr>
<td>IBPELProcessHandle</td>
<td>Explore a BPEL process deployed on a BPEL process domain</td>
</tr>
<tr>
<td>IInstanceHandle</td>
<td>Perform operations on an active instance</td>
</tr>
<tr>
<td>ITask</td>
<td>Interact with a task that has been instantiated in a process domain</td>
</tr>
<tr>
<td>IWorklistService</td>
<td>Search for and complete tasks that have been instantiated from the process domain</td>
</tr>
</tbody>
</table>
} catch (Exception e) {
    e.printStackTrace();
    throw new RuntimeException("Could not initialize process.");
}
Map payload = responseNm.getPayload();
Document xmlResponse = getXMLDoc(
    XMLHelper.elementToString((Element) payload.get(PAYLOAD)));

/*
 * Sets the Instance reference ID for integrating active BPEL
 * instances.
*/
setInstanceReferenceId(strBusinessProcess, xmlResponse);
return XMLHelper.elementToString(xmlResponse.getDocumentElement());

Enabling Workflow

As shown in the following figure, customer data is retrieved from an external system via the BPEL
process and presented in the workflow panel. The user then verifies the data, enters the deposit
amount, and clicks on OK.

The workflow capabilities are based on the ITask and IWorklistService interfaces provided by
BPEL Process Manager. The RIA correlates the tasks in the IWorklistService with the workflow
screen needed to complete the given task. When proper form validation has occurred, the user
interface makes a remote object call to the completeTask method, shown next, to send the data
captured in the UI to the BPEL process for further processing.
**/  
* This function gets the user input from workflow zone, relates it to a task  
* and completes the task. The function uses instance reference id for  
* identifying process instance.  
*/

```java
public String completeTask(String strInstanceReferenceID,  
String strActivityLabel, Document payload) {  
System.out.println("completeTask-" + strInstanceReferenceID + "-"  
+ strActivityLabel);  
String strStatus = "OK";
try {
   IWorklistService worklist = getWorklist();  
   // get task reference
   ITask task = getTask(strInstanceReferenceID, strActivityLabel);
   // set task payload  
   task.setAttachment(payload.getDocumentElement());
   worklist.completeTask(task);
} catch (Exception e) {
   e.printStackTrace();
   strStatus = ERROR_STRING;
}
return strStatus;
}
```

The code places the task-specific data into the task attachment provided by the `ITask` interface. This interface not only sends the pertinent data back to the BPEL process, but also places that data into the BPEL process audit trail.

Next, I'll introduce the other important aspect of your RIA application: process monitoring.

## Monitoring Process Activity

During the first two steps discussed earlier—initiating the process and enabling the workflow—process states are visually updated via `IProcessHandle` and `IInstanceHandle` API interfaces.

Oracle BPEL Console provides a web-based interface for deploying, managing, administering, and debugging BPEL processes. It's an administrative tool designed using JSP pages and servlets that call the BPEL Process Manager API. Consequently, you can easily develop your own RIA console using the API to provide a business-level, process-monitoring interface.

This console provides visual feedback for a specific instance of a process and even provides a link into the audit trail data associated with a given step in a process. However, the primary advantage of the RIA console lies in its ability to perform actual versus planned analysis. The console displays all the possible steps a process could take during its execution; it also renders the path that was in fact taken during execution. Users can click on any process node to retrieve the BPEL payload and render the payload information in a dialog box.

Thanks to visually demarcated ideal-versus-actual process paths and the ability to drill down at the payload level, end users can analyze the process and identify the trouble spots on their own, without IT intervention. The following figure depicts an example of actual-versus-ideal process path analysis; the "actual" path is haloed in green (since the figure is in grayscale, the path has been highlighted using a box).
Next we’ll examine how processes are rendered to enable this analysis and how to retrieve the audit trail data and depict it visually within an RIA application.

Rendering the Process

The Flex BPEL Panel component makes calls to the IProcessHandle interface to retrieve the XML representation of the business model in its entirety. The BPEL Panel uses the IInstanceHandle interface to gather the XML representation of the current instance’s audit trail. The BPEL Panel then compares and merges the two XML representations to create a user interface that can portray the entire model and the path taken.

Flex is particularly adept at such operations by virtue of its native XML capabilities. Flex can easily merge two XML representations by using a combination of Flex repeater controls and ActionScripts. Together, they have the ability to convert XML into an array and then utilize the inherent array-handling capabilities to facilitate the merging of the XML.

The Process Model XML actually serves as the basis for the BPEL Panel as it contains all possible nodes. With each step of the Process Model, the Process model ID is compared to the audit trail model ID. When the IDs of the process and audit match, the results of that node are copied from the audit model into the process model. Determining what path was taken is as simple as interrogating which nodes in the model have actual instance data associated with them.
The following code snippets show the `getProcessModel` and `getInstanceAuditTrail` methods.

`getProcessModel` method:
```java
/**
 * This function gets the debugger XML model of a given BPEL process.
 * The function returns the xml model.
 *
 * @param strProcessID
 *            the business process name.
 * @return the xml process model.
 */
public String getProcessModel(String strProcessID) {
    System.out.println("getProcessModel - " + strProcessID);
    String strProcessModel = "";
    try {
        IBPELProcessHandle process = getLocator().lookupProcess(strProcessID);
        // Returns the debugger XML model of this BPEL process.
        strProcessModel = process.getProcessModel();
    } catch (Exception e) {
        e.printStackTrace();
        strProcessModel = ERROR_STRING;
    }
    return strProcessModel;
}
```

`getInstanceAuditTrail` method:
```java
/**
 * This function gets the XML audit trail of a given BPEL process.
 * The function returns the xml model.
 *
 * @param strProcessID
 *            the business process name.
 * @return the xml process model.
 */
public String getInstanceAuditTrail(String strInstanceReferenceID) {
    System.out.println("getInstanceAuditTrail - " + strInstanceReferenceID);
    String strResponse = "";
    try {
        IInstanceHandle instance = getInstance(strInstanceReferenceID);
        // Returns the XML representation of the audit trail of this
        // instance.
        strResponse = instance.getAuditTrail();
    } catch (Exception e) {
        e.printStackTrace();
        strResponse = ERROR_STRING;
    }
    return strResponse;
}
```

**Viewing Audit Trail Data**

Audit trail data is available to each node of a process that has already executed. This functionality is similar to that in the BPEL Console. The Flex user interface accesses the *details data* for the node of the process chosen in the user interface (see the following figure):
Audit trail data can be extracted using the following ActionScript 2.0 code:

```actionscript
bpe1Object.bpelDetails = eventData.event.details.data;
```

The Flex interface polls for changes in the status of the current business process by tracking activity in the audit trail to visually delineate what step is pending in the business process. When a change is detected, the Flex application requests a new current instance trace using a remoting call to retrieve the audit trail. The instance trace is then used to display data on the input screens associated with each BPEL process user task. Flex client input screens are associated with corresponding BPEL process steps using activity Correlation IDs.

**Conclusion**

The value of a business process lies in its ability to offer superior visual depiction of process performance and enable task integration. The application described here offers tremendous aesthetic appeal to end users.

As I've demonstrated, RIAs can offer a loosely-coupled presentation-tier approach for delivering new SOA applications. By combining an RIA front end with an SOA back end, enterprises can lower the management and deployment costs of internet applications.
Service-Oriented Architecture (SOA) is attracting a lot of buzz from every realm of the IT industry. Propelled by standards-based technologies like XML, web services, and SOAP, SOA is quickly moving from pilot projects to mainstream applications critical to business operations. One of the key standards accelerating the adoption of SOA is Business Process Execution Language for web services (BPEL). BPEL was created to address the requirements of composition of web services in a service-oriented environment. In the past two years, BPEL has effectively become the most significant standard to elevate the visibility of SOA from IT to business level. BPEL is not only commoditizing the integration market, but it is also offering organizations a whole new level of agility—the ability to rapidly change applications as per changing business landscape. BPEL enables organizations to automate their business processes by orchestrating services within and across the firewall. It forces organizations to think in terms of services. Existing functionality is exposed as services. New applications are composed using services. Communication with external vendors and partners is done through services. Services are reused across different applications. Services everywhere!

What this book covers

Section 1: Service-Oriented Integration
Integration has long been a thorn in the flesh of any company. Whether you attribute it to the proprietary nature of the integration products or to the cost of procuring, implementing, and maintaining these products, integration has been dreaded by the IT community. SOA promises to alleviate this everyday problem by introducing a simple concept: don't integrate applications, rather integrate services. This combined with a standards-based approach lowers the total cost of ownership. It represents the sweet spot for SOA investment. Organizations are leveraging SOA to solve a variety of everyday integration problems, thereby making SOA a mainstream technology. In the first section, we introduce different SOA integration scenarios to inspire integration architects and developers.

Chapter 1: Extending Enterprise Application Integration: This chapter focuses on very common business problem i.e. siloed applications and segregated data glued together using proprietary integration solution. How can we best leverage SOA to add value on top of existing integration infrastructure? By service-enabling existing data-integration processes, business processes could be easily automated by orchestrating underlying services. Infosys, a leading systems integrator, has helped many of its customers leverage their existing EAI investment, and explains you how to do exactly this. This chapter takes an example of broken customer data synchronization between Siebel and SAP, and outlines a strategy to automate this process by integrating with proprietary integration solutions like TIBCO and webMethods.

Chapter 2: Service-Oriented ERP Integration: Driven by the business requirements of different departments, countries, and subsidiaries, many organizations end up with multiple ERP systems. The result is data fragmentation and manual processes. This, in turn, leads to poor customer service and loss of revenue. The problem is how to address this problem without re-architecting the entire solution. Sierra Atlantic, a leading consulting firm specializing in integration technologies, encountered a similar issue with its client. In this chapter, Lawrence Pravin, Architect at Sierra Atlantic, takes
an example of a broken sales order creation process. He walks you through a step-by-step approach to automate it across PeopleSoft HR and Oracle E-Business Suite using BPEL in a service-oriented approach.

Chapter 3: Building the Service Value Chain: Not all integrations are limited within the enterprise. Processes interact with applications, people, and partners. You might have heard the term Business-to-Business (B2B) frequently in the past few years. How can organizations build a network of services spanning multiple organizations? The European Space Agency built such a network of web services across more than 20 partners in nine different countries. The primary purpose of this network is to offer Earth Observation and Geographic Information System services to consumers. This chapter presents an initial strategy of how to architect and design a service-oriented partner-friendly network using web services and BPEL for orchestration. The focus is on four important aspects of network design: defining partner relationships, enabling partner provisioning, offering a central registry of available services, and empowering partners and end users.

Section 2: Building Modern Applications

SOA represents an evolution in the way applications are architected and built. Functions, objects, and modules were some of the artifacts used to build applications in the 90s. In essence, SOA has captured many of the old architectures and refined them to provide a new approach in building applications to meet modern business needs. Modern businesses demand faster response time i.e. the ability to meet new business requirements as fast as possible in the most economical way. Modern applications are built with these requirements in mind. Composite Application Development, Service-Oriented Development of Applications (SODA), and Agile Programming are different but related paradigms of building such modern applications in an "incremental" fashion. This second section continues the charter to inspire architects (this time application architects) to build modern service-oriented applications.

Chapter 4: A Services-Oriented Approach to Business Rules Development: Organizations have processes, and processes have rules. Processes need to be automated. Rules need to be defined intuitively. BPEL automates process and a rules engine automates policies. These rules essentially drive the processes. IT organizations have so far struggled to delineate business processes from rules, leading to operational inconsistency and duplication. Policy Studies Inc. provides an approach to segregate rules from processes, and offers a blueprint to expose rules as services for building cleaner applications. Using BPEL and a rules engine, PSI has built a shared services platform to perform Medicare eligibility processing for different states. Kevin Geminiuc, former Architect at PSI, explains the development strategy to integrate BPEL with a rules engine resulting in a solution that is more agile and flexible. With this approach, it is possible to change a business process without touching policies. Policies can be changed graphically without affecting the business processes.

Chapter 5: Building Rich Internet Applications for Workflow and Process Monitoring: As we discussed before, processes interact with applications, people, and partners. How can we build an application that enables business users to interact with processes seamlessly? Applications should be built to enhance end-user experience. Enterra Solutions marries the world of SOA with the world of Web 2.0. In this chapter, Doug Todd, CTO of Enterra, presents a strategy to extend BPEL workflow in a rich user interface and build an application that not only automates processes, but also ups the ante in terms of aesthetic appeal. It also represents a unique approach to customize a platform, which is SOA ready.
Chapter 6: Building BPEL Process on the Fly: BPEL provides an opportunity to bring business and IT together. Business can help define the key processes, and IT provides the necessary infrastructure to run and automate those processes. Some might argue that BPEL is too technical to be handed over to analysts for process modeling. CenterStone software, a provider of workplace management solution, addressed this very concern by building a custom designer geared towards property managers to define processes for workplace management. CenterStone devised an approach to convert the processes designed using the custom designer into BPEL processes. This chapter will inspire you to build applications, which will facilitate tighter integration with your business counterparts. Jerry Thomas, Chief Architect at CenterStone Software, takes you into the guts of this approach by explaining how process definition can be stored in the database and how XQuery and Saxon parser can help you to build an executable BPEL process from its higher-level process definition.

Section 3: SOA Techniques

By now, it is our sincere hope that integration and application architects are "inspired" to take on the SOA challenge. All the learning and encouragement might have invigorated you to apply SOA in your IT environment. However, as you start implementing SOA, you will need to "equip" yourself with best practices, which facilitate efficiency in design, development, and management of your SOA implementation. Best practices will accelerate your path to SOA success and help deliver on the traditional SOA promises, i.e. promote reusability by leveraging existing investment or increase business agility through flexible business processes. In this section, peers will offer you tips and tricks that add value in different stages of your implementation. This third section introduces you to four such "best practices" to derive maximum benefit from your investment. The step-by-step guides attempt to make it easy for you to adopt these proven techniques.

Chapter 7: Making BPEL Processes Dynamic: The benefit of agility has been belabored exhaustively in the industry. We decided to go back to basics and offer you a very simple technical insight into how SOA enables business agility. Agility is directly correlated to the ability to quickly respond to business changes. By using dynamic partner links, processes can effectively change their behavior to adapt themselves to external business conditions and thereby offer flexibility. SPS Commerce, provider of hosted EDI solutions, has built an SOA-enabled platform for enabling seamless exchange of EDI documents between different entities. In this chapter, SPS Commerce will explain the significance of dynamic partner links and walk you through a step-by-step guide to implement partner links in a sample loan-processing scenario. This approach will enable you to quickly add/delete service providers participating in a business process without actually changing the process.

Chapter 8: Using WSIF for Integration: Organizations operate in a heterogeneous environment. Applications are built using different technologies, from different vendors and different implementers. As you start building a process, you will realize that the underlying applications are not necessarily web services. More often than not, they are either .NET applications or applications built using J2EE, i.e. either purchased applications or home-grown Java Applications. Matjaž Jurič, author of the Packt book Business Process Execution Language for Web Services (ISBN: 1-904811-18-3) presents a strategy to integrate with Java resources using Web Services Invocation Framework (WSIF). Matjaž, professor of Computer Science at the University of Maribor, argues that although it is possible to expose these applications as web services and use them in the process, accessing these resources natively can improve the application performance significantly.

Chapter 9: BPEL with Reliable Processing: The success of any service in the SOA world depends upon its degree of reusability, which in turn depends upon the quality of service offered. As you run your SOA applications, many things could go wrong. Network connections may be lost, participating applications may go down, incoming data might be corrupted, etc. These external interruptions can degrade the quality of a particular application. How can you design an application that can withstand all these failures and still emerge as a winner? Qualcomm encountered this specific issue while leveraging SOA to build the Entitlement Provisioning application. Presented in the fashion of step-by-step tutorial, Jeremy Bolie, IT Manager, and Michael Cardella, Architect at Qualcomm, share with you a strategy to design a reusable BPEL process capable of offering any service and defeating rectifiable errors.

Chapter 10: Managing a BPEL Production Environment: The last chapter in this cookbook deals with an important aspect of any application—maintenance. More dollars are spent in maintenance and enhancement of an application than the combined amount spent during its design, development, and testing phases. Once the application is deployed into production, the real work begins. Belgacom, one of the leading telecommunications companies in Belgium, has automated DSL service provisioning and diagnosis using BPEL. Having been in production for a long time, Belgacom has vast practical experience in managing a BPEL infrastructure. In this chapter, Stany Blanvalet, former Architect at Belgacom, explains various strategies for managing a BPEL production environment. This is a must read for all the BPEL administrators.

Where to buy this book
You can buy BPEL Cookbook: Best Practices for SOA-based integration and composite applications development from the Packt Publishing website:

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.