Oracle Fusion Middleware Patterns

Harish Gaur
Markus Zirn

Chapter No.2
"Process-driven SOA Development"
In this package, you will find:

A Biography of the authors of the book
A preview chapter from the book, Chapter NO.2 "Process-driven SOA Development"
A synopsis of the book’s content
Information on where to buy this book

About the Authors

**Harish Gaur** is Director of Product Management for Fusion Middleware at Oracle. In this role, he works closely with strategic customers implementing Service-Oriented Architecture using Oracle SOA technology. Harish is the co-author of *BPEL Cookbook* from Packt Publishing. He holds an engineering degree in Computer Science and an MBA from Haas School of Business, UC Berkeley. Harish is an author and editor for this book.

**Markus Zirn** is Vice President of Product Management for Oracle Fusion Middleware. He is the editor of *BPEL Cookbook*, the author of several articles on SOA and related topics, and a frequent speaker at leading industry and analyst conferences. He holds a master's degree in Electrical Engineering from the University of Karlsruhe, Germany, the University of Southampton, U.K., and ESIEE, France.

For More Information:
Oracle Fusion Middleware Patterns

by Alex Andrianopoulos - Vice President of Fusion Middleware Product Marketing at Oracle

Dain Hansen - Director of Fusion Middleware Product Marketing at Oracle

Enterprise Architecture is seldom simple enough to be built on a single technology—it generally encompasses a wide array of technologies, including SOA, Identity Management, Business Intelligence, Performance Management, Web 2.0, Content Management, BPM, Distributed Caching, Data Integration, and Application Servers.

What are the key business drivers where you need mix of a multiple technologies? What are the best practices (organizational as well as architectural) to meet enterprise needs? How do other organizations approach enterprise challenges that transcend technology and product boundaries? IT Leaders, CIOs, Enterprise Architects are searching for proof points.

This book is an attempt to answer these questions by illustrating 10 unique enterprise solution patterns to solve business needs in three specific areas—process improvement, business visibility, and collaboration and security. Each article will introduce a new pattern, along with an architectural overview of a real-world customer solution developed with Oracle Fusion Middleware.

What This Book Covers

Chapter 1 discusses a new approach to application development and enterprise architecture—Fusion development and Oracle Enterprise Architecture. These principles lay the foundation of any agile and flexible solution IT needs to deliver to the business. Rest of the book is conceptually divided into three sections dedicated for solutions in three areas—process improvement, business visibility and collaboration, and security.

Chapter 1, Building Agile Applications using Fusion Development and Oracle Enterprise Architecture Principles, introduces the concept of Fusion development and Oracle Enterprise Architecture (OEA). These principles serve as a guideline to create flexible and agile solutions. This chapter will demonstrate how British Telecom defined Universal Application Framework using OEA and Fusion Middleware technologies. UniApp framework helped BT to drive consolidation of 46 different applications down to one and implement application development standardization. This chapter also highlights how Pardee Homes, major home builder in California, USA, adopted the Fusion Development approach to assemble its solution, leveraging components of Oracle Fusion Middleware, including Oracle BPA Suite, Oracle BPEL Process Manager, Oracle Web Services Manager, Oracle ADF, and Oracle Application Server.

For More Information:
Section 1: Process Improvement

Businesses often grapple with questions like "How do I improve efficiencies in my core business processes and make it more agile?", "How can I improve the response time of my SOA applications while minimizing hardware costs?", or "How do I extend the value of my ERP applications?". Unfortunately, there is lack of integrated tools to handle entire process lifecycle—modeling, integration, execution, management, and monitoring of business processes. No standardization in application development. Performance of composite applications is generally an after-thought. Next three chapters provide solution patterns to answer these questions through process improvement approach.

Chapter 2, Process-driven SOA Development, highlights how SOA can help IT align with key business processes. SOA reduces the semantic gap between IT and business by introducing a development model that aligns the IT development cycle with the business process lifecycle. This chapter introduces reader to organizational and technical aspects of SOA development. It then describes how Elektro Slovenija, Slovenia's state-owned power distribution company, transformed its procurement process using Oracle BPA and Oracle SOA Suite.

Chapter 3, Code-free Application Extensions and Integrations, deals with the most common problem we see in our ERP install base—how do I extend my ERP applications? Fusion Middleware plays a critical role here. This chapter illustrates how you can use SOA, BPM, Web 2.0, and BI to extend and integrate Oracle applications without writing a single line of code. The approach makes these modifications simple, quick to implement, and easy to maintain/upgrade. It offers an innovative and practical solution to radically change the economics of running Oracle applications.

Chapter 4, Data Tier Caching for SOA Performance, is all about improving the performance of composite SOA applications. Performance/scalability issues have grown to be the one of the topmost concerns when building an SOA application. This chapter introduces a new architecture pattern—boosting SOA performance with distributed caching. It discussed how a mid-tier caching strategy can inject high performance into data services as part of a SOA. It also illustrates the approach a major pharmaceutical company took to improve the performance of a composite application using an Oracle Coherence Grid solution with Oracle SOA Suite.

Section 2: Business Visibility

Business constantly struggle with questions like "How do I make sound business decisions with disconnected data?" or "How can I measure my KPIs and improve them in real-time?". Challenge is that there is no unified platform available in the market to handle "what-if" and "how" analysis. BI dashboards are traditionally not actionable by business users and there is no real-time intelligence for on-the-fly decision making. Next three chapters provide user with solution patterns to achieve business visibility.

For More Information:
Chapter 5, Integrated Real-time Intelligence with Oracle's WebCenter, Coherence, and Business Activity Monitoring, describes reference architecture for contextual, real-time business insight that uses BAM and E2.0 to combine information management and analytics in the same context and transaction. A combination of BI and E2.0 allows us to combine information management and analytics in the same context and transaction. Using DJO, a leading global provider of high-quality orthopedic devices as an example, this chapter walks through a real-life example of how this is accomplished using Oracle WebCenter, Oracle Business Activity Monitoring, and Oracle Coherence.

Chapter 6, Achieving Business Insight by Integrating Relational and Multi-dimensional Data, tackles how organizations need to simultaneously address situational questions ("What"), and forward-looking questions ("How") as part of their every day organizational reporting and planning needs. Through the use of a real-world example, this article highlights a compelling business need to integrate relational and multi-dimensional data. We will discuss how the Australia-based Knowledge Global has used the combined power of Oracle BIEE, Oracle Essbase and Oracle Data Integrator to build a carbon/energy monitoring and measurement application. In Chapter 7, Building Intelligent Processes with Insight-driven Agility, discusses how convergence of BI and SOA allows organizations to become agile. Business SOA and BI are natural partners for a changing organization. Processes that are supported by SOA allow the organization to directly execute the business model and better support business change; BI provides measures that inform decisions for strategic and tactical change within an organization. Combining SOA with BI allows you to act on those measures, changing processes, services, and rules to target identified improvement goals. In this chapter, we will show how Motability Operations is creating a platform to monitor and improve vehicle remarketing efforts across multiple channels using Oracle OBIEE, Oracle BPEL PM, and Oracle Business Rules.

Section 3: Collaboration and Security

Last section deals with collaboration and security. If you are grappling with issues like "How do I open up to new communities and channels without compromising my organization's brand, intellectual property, and customer data?", "How do I enable customer self- service in a collaborative environment?", or "How can I automate employee onboarding to achieve compliance?", then you might have realized that there is no effective way to handle collaboration across different contributors. Data could be widely dispersed in structured (database) or unstructured (e-mail, chats, VOIP, and others) format. There is a need for appropriate checks and balances to maintain security and privacy, but tools are ineffective in doing so. Next three chapters provide solution patterns to address business issues around collaboration and security.

Chapter 8, Building Enterprise 2.0 Applications, examines the key building blocks of Enterprise 2.0 architectures and then outlines important integration considerations for
building an Enterprise 2.0 application. The fundamental capabilities of any rich Enterprise 2.0 requires the combination of content management, a Web 2.0 framework, security, and integration with enterprise applications. This chapter also illustrates how Wind River, a software device optimization company, leveraged Oracle WebCenter and Oracle Enterprise Content Management platform to revamp its online customer support portal.

Chapter 9, Automating Enterprise Reporting with WebCenter, SOA, and Oracle Business Intelligence Publisher, explains how to optimize enterprise reporting within a company. It explains how to build an automated reporting platform with SOA, an enterprise reporting tool, and a portal. It then walks through a real-life example of how Arcturus, a leading real estate services company, built a property management reporting solution using Oracle Business Intelligence (BI) Publisher, Oracle SOA Suite, and Oracle WebCenter.

Chapter 10, A Role-based Approach to Automated Provisioning and Personalized Portal, demonstrates how an organization can take a role-based approach to automate provisioning and personalize a portal. The solution should include four key components: a provisioning platform, a role management platform, an access management platform, and a portal. The chapter explores how Schneider National, a multinational trucking company, successfully automated employee on-boarding and personalized its intranet portal using Oracle Role Manager (ORM), Oracle Identity Manager (OIM), and Oracle WebCenter.
Business Process Management and SOA

One of the major benefits of a Service-Oriented Architecture is its ability to align IT with business processes. Business processes are important because they define the way business activities are performed. Business processes change as the company evolves and improves its operations. They also change in order to make the company more competitive.

Today, IT is an essential part of business operations. Companies are simply unable to do business without IT support. However, this places a high level of responsibility on IT. An important part of this responsibility is the ability of IT to react to changes in a quick and efficient manner. Ideally, IT must instantly respond to business process changes.

In most cases, however, IT is not flexible enough to adapt application architecture to the changes in business processes quickly. Software developers require time to modify application behavior. In the meantime, the company is stuck with old processes. In a highly competitive marketplace such delays are dangerous, and the threat is exacerbated by a reliance on traditional software development to make quick changes within an increasingly complex IT architecture.

For More Information:
The major problem with traditional approaches to software development is the huge semantic gap between IT and the process models. The traditional approach to software development has been focused on functionalities rather than on end-to-end support for business processes. It usually requires the definition of use cases, sequence diagrams, class diagrams, and other artifacts, which bring us to the actual code in a programming language such as Java, C#, C++, and so on. SOA reduces the semantic gap by introducing a development model that aligns the IT development cycle with the business process lifecycle. In SOA, business processes can be executed directly and integrated with existing applications through services.

To understand this better, let's look at the four phases of the SOA lifecycle:

- **Process modeling**: This is the phase in which process analysts work with process owners to analyze the business process and define the process model. They define the activity flow, information flow, roles, and business documents. They also define business policies and constraints, business rules, and performance measures. Performance measures are often called Key Performance Indicators (KPIs). Examples of KPIs include activity turnaround time, activity cost, and so on. Usually Business Process Modeling Notation (BPMN) is used in this phase.

- **Process implementation**: This is the phase in which developers work with process analysts to implement the business process, with the objective of providing end-to-end support for the process. In an SOA approach, the process implementation phase includes process implementation with the Business Process Execution Language (BPEL) and process decomposition to the services, implementation or reuse of services, and integration.

- **Process execution and control**: This is the actual execution phase, in which the process participants execute various activities of the process. In the end-to-end support for business processes, it is very important that IT drives the process and directs process participants to execute activities, and not vice versa, where the actual process drivers are employees. In SOA, processes execute on a process server. Process control is an important part of this phase, during which process supervisors or process managers control whether the process is executing optimally. If delays occur, exceptions arise, resources are unavailable, or other problems develop, process supervisors or managers can take corrective actions.

- **Process monitoring and optimization**: This is the phase in which process owners monitor the KPIs of the process using Business Activity Monitoring (BAM). Process analysts, process owners, process supervisors, and key users examine the process and analyze the KPIs while taking into account changing business conditions. They examine business issues and make optimizations to the business process.
The following figure shows how a process enters this cycle, and goes through the various stages:

Once optimizations have been identified and selected, the process returns to the modeling phase, where optimizations are applied. Then the process is re-implemented and the whole lifecycle is repeated. This is referred to as an iterative-incremental lifecycle, because the process is improved at each stage.

**Organizational aspects of SOA development**

SOA development, as described in the previous section, differs considerably from traditional development. SOA development is process-centric and keeps the modeler and the developer focused on the business process and on end-to-end support for the process, thereby efficiently reducing the gap between business and IT.

The success of the SOA development cycle relies on correct process modeling. Only when processes are modeled in detail can we develop end-to-end support that will work. Exceptional process flows also have to be considered. This can be a difficult task, one that is beyond the scope of the IT department (particularly when viewed from the traditional perspective).
To make process-centric SOA projects successful, some organizational changes are required. Business users with a good understanding of the process must be motivated to actively participate in the process modeling. Their active participation must not be taken for granted, lest they find other work "more useful," particularly if they do not see the added value of process modeling. Therefore, a concise explanation as to why process modeling makes sense can be a very valuable time investment.

A good strategy is to gain top management support. It makes enormous sense to explain two key factors to top management—first, why a process centric approach and end-to-end support for processes makes sense, and second, why the IT department cannot successfully complete the task without the participation of business users. Usually top management will understand the situation rather quickly and will instruct business users to participate.

Obviously, the proposed process-centric development approach must become an ongoing activity. This will require the formalization of certain organizational structures. Otherwise, it will be necessary to seek approval for each and every project. We have already seen that the proposed approach outgrows the organizational limits of the IT department. Many organizations establish a BPM/SOA Competency Center, which includes business users and all the other profiles required for SOA development. This also includes the process analyst, process implementation, service development, and presentation layer groups, as well as SOA governance.

Perhaps the greatest responsibility of SOA development is to orchestrate the aforementioned groups so that they work towards a common goal. This is the responsibility of the project manager, who must work in close connection with the governance group. Only in this way can SOA development be successful, both in the short term (developing end-to-end applications for business processes), and in the long term (developing a flexible, agile IT architecture that is aligned with business needs).

**Technology aspects of SOA development**

SOA introduces technologies and languages that enable the SOA development approach. Particularly important is BPMN, which is used for business process modeling, and BPEL, which is used for business process execution.

---

*For More Information:*

BPMN is the key technology for process modeling. The process analyst group must have in-depth knowledge of BPMN and process modeling concepts. When modeling processes for SOA, they must be modeled in detail. Using SOA, we model business processes with the objective of implementing them in BPEL and executing them on the process server. Process models can be made executable only if all the relevant information is captured that is needed for the actual execution. We must identify individual activities that are atomic from the perspective of the execution. We must model exceptional scenarios too. Exceptional scenarios define how the process behaves when something goes wrong—and in the real world, business processes can and do go wrong. We must model how to react to exceptional situations and how to recover appropriately.

Next, we automate the process. This requires mapping of the BPMN process model into the executable representation in BPEL. This is the responsibility of the process implementation group. BPMN can be converted to BPEL almost automatically and vice versa, which guarantees that the process map is always in sync with the executable code. However, the executable BPEL process also has to be connected with the business services. Each process activity is connected with the corresponding business service. Business services are responsible for fulfilling the individual process activities.

SOA development is most efficient if you have a portfolio of business services that can be reused, and which includes lower-level and intermediate technical services. Business services can be developed from scratch, exposed from existing systems, or outsourced. This task is the responsibility of the service development group. In theory, it makes sense for the service development group to first develop all business services. Only then would the process implementation group start to compose those services into the process. However, in the real world this is often not the case, because you will probably not have the luxury of time to develop the services first and only then start the processes. And even if you do have enough time, it would be difficult to know which business services will be required by processes. Therefore, both groups usually work in parallel, which is a great challenge. It requires interaction between them and strict, concise supervision of the SOA governance group and the project manager; otherwise, the results of both groups (the process implementation group and the service development group) will be incompatible.

For More Information:
Once you have successfully implemented the process, it can be deployed on the process server. In addition to executing processes, a process server provides other valuable information, including a process audit trail, lists of successfully completed processes, and a list of terminated or failed processes. This information is helpful in controlling the process execution and in taking any necessary corrective measures. The services and processes communicate using the Enterprise Service Bus (ESB).

The services and processes are registered in the UDDI-compliant service registry. Another part of the architecture is the rule engine, which serves as a central place for business rules. For processes with human tasks, user interaction is obviously important, and is connected to identity management.

The SOA platform also provides BAM. BAM helps to measure the key performance indicators of the process, and provides valuable data that can be used to optimize processes. The ultimate goal of each BAM user is to optimize process execution, to improve process efficiency, and to sense and react to important events.

BAM ensures that we start optimizing processes where it makes most sense. Traditionally, process optimization has been based on simulation results, or even worse, by guessing where bottlenecks might be. BAM, on the other hand, gives more reliable and accurate data, which leads to better decisions about where to start with optimizations. The following figure illustrates the SOA layers:
Case study: Process modeling

So far, we discussed the theory. Now let’s take a look at a case study of an end-to-end procurement process from Elektro Slovenija, Slovenia’s state-owned power distribution company. The procurement process was implemented using a full set of Oracle tools—Oracle Business Process Analysis PA Suite for the modeling, SOA Suite (BPEL Process Manager, ESB, Rules Author, WS Manager, Application Server) with JDeveloper, and Service Registry for the implementation, along with Oracle BAM for the business activity monitoring. The interrelationship is shown in the following figure:

As a state-owned company, Elektro Slovenija must conform to strict regulations regarding procurement. The process starts with an order request form. First, a decision needs to be made as to whether the order will be collected with other similar orders for a joint purchase (for example, for office materials), or as an individual order. The order value influences the process as well. Orders smaller than €4,200 are the most simple and require that three offers are collected and that a purchase order is issued. For orders up to €12,000, a negotiation process takes place and a contract is issued. For larger orders, a special commission is created to carry out the ordering process, which differs depending on the type of order (product, service, or real estate). Several roles are involved in the process, including the order creator, the person responsible for the contract, the head of the procurement unit, and the commission for larger orders.

For More Information:
Modeling the process was the first challenge in the project. The company had already established an SOA Competency Center, and top management already had a good understanding of BPM and SOA. This simplified the situation, in that it was not very difficult to motivate business users to participate. In our experience, the group that models the process should include people in the following roles:

- Process owner, who will verify the process flow and make decisions about possible changes in the process.
- One or two process owner assistants with a solid understanding of the process. These individuals will do the actual modeling.
- Moderator, who will ask questions and lead the meeting.
- Process modeler, who is experienced in detailed SOA modeling.

The process in the BPA Suite has been modeled on six levels. It includes 24 sub-processes, and consists of 230 activities, of which more than 100 are human tasks. The process involves 25 different roles, implements more than 40 business rules, and defines seven key performance indicators. The following figure shows the BPA Suite application displaying the top-level process diagram:

It is worth mentioning that the BPA Suite has good support for processes with human tasks (such as our example process). In addition, BPA Publisher can be used to share the process definition with other interested parties in order to foster collaboration.

Once the process has been designed, BPA Suite can check the model for semantic validity in order to identify parts of the process that may have been incorrectly modeled, as illustrated in the following screenshot:

For More Information:

Case study: Process implementation

Three parallel development activities were started in order to develop the composite application for end-to-end support. They are:

- Development of the BPEL executable process
- Development of business services
- Development of the presentation layer—the user interface

In this case, the same team of two developers developed the BPEL executable process and the business services. One of those developers was also the process modeler. This simplified the organizational aspects and reduced the communication overhead.

The following subsections briefly describe the development of the BPEL executable process and the business services. However, the user interface is not SOA-specific, so it will not be discussed.
Development of BPEL executable process

The BPEL executable process can be developed manually, or it can be automatically translated from the BPMN model. We will use the latter approach, because with automatic translation we ensure consistency between the process model and the executable model. This approach also reduces effort, as it does not require developing BPEL from scratch. Finally, automatic translation can be applied in both directions, from BPMN to BPEL and vice versa, as we will see later. To enable seamless translation, several guidelines should be followed by the BPMN design (see the Guidelines on BPMN to BPEL mapping section covered later in this chapter). BPMN to BPEL transformation is very complex, and the same BPMN construct can be mapped to BPEL in different ways.

If you follow the guidelines, the transformation from BPMN to BPEL (from BPA Suite to JDeveloper blueprint) is straightforward. While the technology is relatively new, it works well enough for production use on complex processes, such as this procurement process. Our experience with the transformation has been mostly positive. The following screenshot shows the JDeveloper BPEL Blueprint perspective.
The BPEL code has to be amended in JDeveloper. The most important task here is to connect the BPEL process with the actual business services. Please note that the WSDL interfaces of the business services can be used in the BPA Suite already. This means that we do not have to wait until the development phase (in JDeveloper, where it is obvious that we will use WSDL interfaces). This way we can make the process model conformant with business service interfaces in the design phase, which can simplify BPEL development. Such an approach, however, requires some additional skills from the process modelers.

In addition to connecting BPEL with services, other important activities of this phase include development of ESB mediations, registering services in the Service Registry, entering business rules into the Rules Author, deploying the process, and developing the BAM dashboard (which we will talk about later).

**Development of business services**

The procurement process had to be integrated with several existing applications, particularly the business information system applications, including accounting, general ledger, and inventory. Not all applications had service interfaces; therefore, some new interfaces had to be developed and business logic exposed through services. When exposing existing applications as services, it is crucial to have support from the original developers. Otherwise, service developers will waste a lot of time struggling through legacy code. In this case, existing applications have been Oracle Forms applications. The procurement process also deals with quite a few documents, including the order request, tender documentation, offers, a contract, purchase order, and several others. To manage electronic versions of the documents, an e-document content management system has been used. Although Oracle Universal Content Management would be the obvious choice, the company already had another content management system deployed, which had service interfaces. This made the task of integrating it quite straightforward. When developing business services, it is important to follow SOA patterns and, in particular, the loose-coupling principle. It is also important to develop reusable business services.

**BPMN to BPEL Round-tripping**

An important part of SOA development, particularly in real-world projects, is the ability to round-trip changes between the BPMN model in BPA Suite and BPEL in the JDeveloper blueprint representation. There are two important aspects of round-tripping:

- How changes in the BPMN model can be propagated to BPEL without losing implementation changes already done in BPEL
- How changes done in BPEL can be propagated back to the BPMN model, to keep them both in sync

For More Information:
We have been pleasantly surprised by the maturity of Oracle tools. In both scenarios, the round-tripping experience has been positive:

- When updating the BPA process model, we have been able to propagate changes to the JDeveloper blueprint without losing previous implementation changes to the BPEL.
- When updating the BPEL in JDeveloper, we have been able to propagate the changes back to the BPA model, where the business people had the choice to accept or decline the changes. The following screenshots illustrate an example where a Save Order request activity has been added in BPEL (in JDeveloper) and the change has been propagated to the BPMN model in BPA Suite.
Round-tripping is very important for real-world development, as it is the key to iterative SOA development, which guarantees short development cycles and easy modifications to existing composite applications. Round-tripping also keeps the model (BPMN) and the executable code (BPEL) in sync. Our experience with Oracle tools revealed only very minor problems, such as fault handlers that did not synchronize properly between BPA and JDeveloper. In our opinion, the approach taken by Oracle, where the modeling and implementation tools are separate, is better than the approach that uses the same tool for both.

**Guidelines on BPMN to BPEL mapping**

Let’s take another look at the BPMN to BPEL mapping. We learned that in BPA Suite not all BPMN constructs map equally well to BPEL. Therefore, a process modeler must be aware of the specifics related to the BPMN to BPEL mapping in order to get the most efficient model.

The basic rules for mapping are:

- All BPMN activities are mapped to BPEL scopes
- Start events are mapped to receive or pick activities, depending on the type of trigger
- End events are mapped to reply, invoke, throw, or other activities, depending on the type
- Gateways are transformed to different BPEL activities, such as pick, switch, or flow

For More Information:
Process-driven SOA Development

- Business data is mapped to variables
- Subprocesses are mapped to invoke activities
- Each BPMN subprocess becomes a separate BPEL process

Particular care must be exercised when dealing with the cycles. BPMN supports arbitrary cycles, but BPEL does not. Therefore, we must convert all cycles to while loops (structured cycles). This, however, can change the process model considerably, at least from the visual perspective (the process behavior is unchanged if the appropriate conversion is performed). As a result, business people might have trouble understanding the converted model. The lack of support for arbitrary cycles can also be a problem if the process has many interleaved human tasks.

To avoid structured cycles in simpler scenarios, we can do some refactoring for decisions with multiple outputs. Such decisions should be decomposed into multiple decisions. For example, the following scenario does not map appropriately to BPEL:

![Diagram](image1)

Therefore, we should decompose the decision into two decisions as follows:

![Diagram](image2)

The activities in cycles, which repeat, are duplicated in the generated BPEL. This can be avoided by grouping the repeating activities and modeling them as subprocesses. Then, only the invoke activity for calling the subprocess will be repeated in the generated BPEL.

Exercise caution with the end events, which do not actually end the BPEL process, as you might expect. In BPMN decisions we can influence how the generated BPEL switch will be generated. If we do not define the default flow for a condition, an otherwise is generated within the BPEL switch activity. This can be useful, particularly for human tasks where the BPEL process should check for a variety of possible human task outcomes.

**Process execution**

After successful development, the composite application is deployed to the process server. Oracle SOA Suite and BPEL Process Manager in particular provide adequate tools to control and manage the process execution. Particularly, the added fault policies in 10.1.3.4 are very useful in production use, as they allow system administrator intervention if a fault occurs in the process. The procurement process is a long-running process that includes several human tasks. So the process must not be terminated if a fault occurs. Fault policies support this and allow a separation of the system fault handling from the process implementation. WS Manager is a useful tool in the process execution phase. It simplifies supervision of the process and services, and allows usage monitoring and even SLA compliance control.

Processes that include human tasks also require an identity management system, with all the roles and users that participate in the process. In our case, we used Active Directory, which can be integrated with the SOA Suite. Alternatively, we could have used Oracle Identity Management.

**Process monitoring using BAM and optimizations**

Legislative changes in the past made several modifications necessary to the procurement process. In addition, the company's efforts to increase efficiency, which placed a high priority on the ability to systematically identify process bottlenecks, further increased the need for process modification.
To support this, we have implemented BAM support. This includes the definition of sensors to monitor the process execution and the development of a BAM dashboard. The following screenshot shows an example dashboard developed for the process:

BAM has proved particularly important for process owners, who are able to monitor the process execution in real time. In our experience, developing a BAM dashboard is quite simple, at least for monitoring simple KPIs.

The final challenge was to import the BAM data back into the BPA modeling tool. This can be done using custom JavaScript, but it's quite time consuming. Future versions of BPA Suite will improve on this and make things easier.

BAM data is a useful starting point for process optimizations. So far, the procurement process is only in its first iteration, and has not yet been optimized. However, BAM data has already been used to identify possible optimization points. Even in its first iteration, the procurement process has demonstrated several benefits—the workload on the employees is reduced by 25% to 35%, as some process activities have been automated, and the visibility of the process is considerably improved. It is possible to track the execution of the process instances and to figure out which activity is occurring within a selected process instance. This reduces gap times, which leads to faster execution. In future iterations, more process steps will be automated, which will result in even higher benefits.
Summary
As a real-world example, the case study has proved the most important benefits of a process-centric SOA approach—better alignment between business and IT, faster development cycles, fewer errors, and most important, much faster development cycles, which guarantee considerable reduction in the time required to fulfill business requirements.

Related to better alignment between business and IT, the following important benefits of process-centric development also emerged:

- Better understanding of business requirements
- End-to-end business process support
- Process monitoring and execution control, which provides valuable business performance indicators

These benefits would have been very difficult to achieve without an SOA approach.

We also learned that the process-centric approach to SOA development is both a technical and an organizational challenge. It requires the setting up of competent groups for process modeling, process implementation, service development, and the development of the user interface. It also requires the setting up of necessary supervision, through SOA governance and project management. For smaller projects, the same small development group can take over several roles. Finally, SOA development requires new skills to be successful.

The benefits of the complete SOA process lifecycle, spanning modeling, implementation, execution, monitoring, and optimization, are numerous and well worth the necessary investment in the required knowledge and products. We believe that covering the full process lifecycle shows the real value of SOA, which will be even more obvious when changes, modifications, and optimizations are made to the process.

The main message of this case study is that it is possible to develop a complete end-to-end process for a complex real-world process. The process-centric approach to SOA development with full lifecycle support encompasses enormous opportunities for companies to improve and optimize not only their IT support but also their operational efficiency through process automation.

For More Information:
Further Reading:


Where to buy this book


Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

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