



Oracle BPEL “Nuts and Bolts”

Paper 743

presented by

John Jay King

King Training Resources

john@kingtraining.com

Download this paper from:

<http://www.kingtraining.com>

Copyright @ 2009, John Jay King



- Understand the importance of BPEL to SOA
- Learn how Oracle's BPEL fits into Service Oriented Architecture (SOA) and Oracle Fusion Middleware
- Understand Oracle's BPEL Implementation
- Be able to discuss issues involved in process orchestration



- John King – Partner, King Training Resources
- Providing training to Oracle and IT world for over 20 years
 - Databases: Oracle, DB2, SQL Server, more...
 - Languages: PL/SQL, Java, C#, COBOL, more...
 - Operating Systems: Linux, Unix, Windows, z/OS, more...
 - Tools: XML, HTML, JavaScript, more...
- Leader in Service Oriented Architecture (SOA) design and implementation
- Home is Centennial, Colorado (Denver) – I like to hike and drive in the mountains



- Business Process Execution Language for Web Services (BPEL or BPEL4WS)
 - Language created to compose, orchestrate, and coordinate web services
 - Processes are “composed” of existing services
 - BPEL is the result of over ten years of collaborative effort in Business Project Management by Microsoft and IBM
 - Today, BPEL products are supported by Oracle (including BEA and Siebel), IBM, Microsoft, SAP, and many others
 - BPEL is an open industry standard managed by OASIS

<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.html>



- BPEL extends Web Services
 - BPEL provides both synchronous and asynchronous interactions
 - BPEL supports long-running interactions
 - BPEL allows a well-defined mechanism for creating process definitions
- While usually depicted graphically; BPEL is defined using XML



- BPEL uses an XML language to specify and describe business processes in two specific ways:
 - Definition of executable processes that may be orchestrated
 - Definition of message exchanges or abstract business protocols
- BPEL uses web service XML technologies:
 - XML (eXtensible Markup Language)
 - WSDL (Web Service Description Language)
 - XPath (XML Path Language)
 - XML Schema



- Service Oriented Architecture (SOA) represents a new way of looking at the relationship between “the business” and IT
- Today’s organization must be agile enough to innovate to meet rapidly changing needs
- In most organizations, changes that do not fit current IT infrastructure are accomplished slowly:
 - Significant modifications required
 - Infrastructure additions necessary
- SOA ties IT Service development to business processes rather than to a specific infrastructure



- With SOA all or part of a business process is represented as a Service or set of Services that may be executed securely in a standardized way
- SOA Services may be stand-alone or combined with other Services to address new business issues
- Services are loosely-coupled communicating via coarse-grained messages
- The ability to orchestrate existing Services to meet new needs provides the agility modern organizations require to meet new and future needs



- Generally, services use one or more software components to satisfy some business process
- Service messages are generally coarse-grained (e.g. Purchase Order)
- Component and component tasks that make up a service are often focused in nature and usually require fine-grained messaging (e.g. Purchase Order Line Item)
- Each Service is generally represented by one or more software components; frequently Web Services today
- Other Services might represent existing legacy components or non-SOA implementations



- A key strength of SOA is simplicity
- Basic principles guiding SOA:
 - Standard set of enterprise service definitions *described in a registry*
 - Central management of service definitions *(enabled in-part by a registry)*
 - Loose coupling



- Central management of service definitions ensures that
 - Duplicate services are not created
 - Developers follow organization standards
 - Developers can find (and use) services



- In order to interact successfully with a service, you must know at least two things:
 - What you expect to get from the service
 - What information you have to provide the service
- A well-defined “contract” from the service provider spells out business and technology requirements for using a service (the “interface”) and how to invoke the service
 - A service contract reflects specific business knowledge and is the basis for sharing and reusing services
 - Maintenance of service “contracts” becomes critical over time
 - Contracts are stored in a service registry



- All the Services in the world are **useless** unless:
 - We know what they are named
 - We know where to find them
 - We know the expected inputs and outputs
 - We trust them to work as specified in their contract



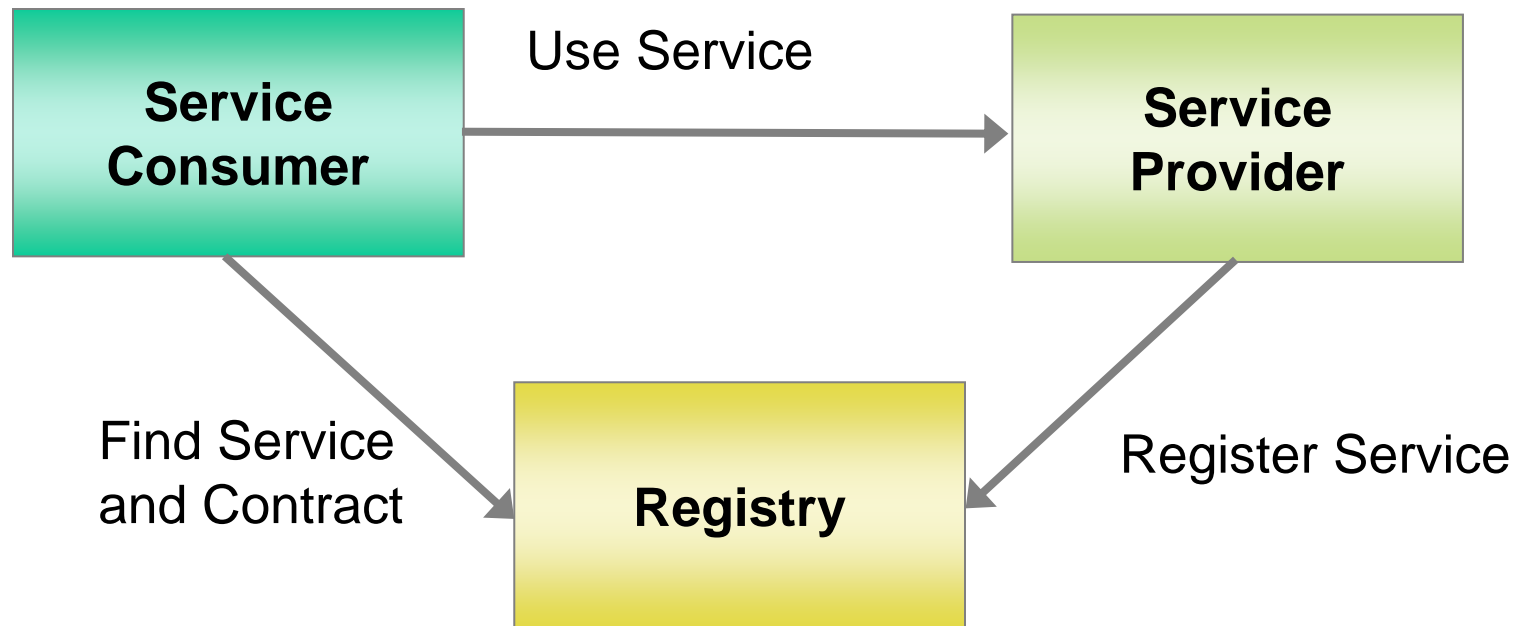
- Governance is needed to:
 - Make sure multiple services don't provide the same functionality
 - Understand who is responsible for a given service
 - Prioritize and control change requests
 - Determine that services conform to standards
 - Ensure that contracts are accurate
 - Provide a level of comfort that advertised services work and can be accessed as described by their contract
 - Be sure that services are cataloged and can be located



- What enables SOA?
 - Services
Software components or sets of components
 - Service Providers
Location (server) where Services are available
 - Service Consumers
Software actually using Services (often user-facing)
 - Service Registries
Contains “contracts” describing available services
 - Messaging
Communications between Service and Service Consumer



- The **Contract** describing the **service**, its inputs and outputs, location, and method of invocation is placed in the **Registry** by the **Service Provider**
- The **Service Consumer** locates a **Service** using the specifications found in the service's **contract** from a **Registry**
- **Service Consumers** use **Services** provided by a **Service Provider** to perform all or part of some business function



- Service Component Architecture (SCA) describes a model for building SOA applications and systems via component-based applications that either provide or consume functional components via Service-Oriented interfaces
 - SCA separates SOA application development into two parts:
 - Implementation of service components that provide/consume services
 - Orchestration of sets of service components to create new applications
 - SCA was originally created by a collaboration of industry organizations including: BEA Systems, Inc., Cape Clear Software, IBM Corporation, IONA Technologies PLC, **Oracle**, Red Hat Inc., Rogue Wave Software, SAP AG, Siemens AG, Software AG, Sun Microsystems, Sybase, TIBCO Software Inc., and Zend Technologies
- <http://www.osoa.org/display/Main/Service+Component+Architecture+Home>
<http://www.oasis-openca.org/sca>



- So how does Oracle's Fusion fit in?
- Oracle uses the title "Fusion" to unify its SOA-directed offerings and highlight the integration features incorporated in their products; the acquisition of BEA has added strength to Oracle's SOA offerings
- Two major legs of Oracle Fusion Architecture identified so far are:
 - Oracle Fusion Middleware
 - Oracle Fusion Applications



- Oracle outlines five core principles to Fusion Architecture:
 - Model Driven Following business processes
 - Service and Event-Enabled Loosely-coupled, modular, and flexible
 - Information-Centric Providing complete, actionable information
 - Grid-ready Scalable via low-cost hardware
 - Standards-based Based upon open standards allowing easy interaction with other products



- Oracle Fusion Middleware builds on the solid Java EE and open-source architecture of Oracle App. Server
- SOA emphasis on business processes in Oracle Fusion Middleware leads to better coordination between Oracle's IT groups and business units
- Oracle Fusion Middleware comes complete with over 250 adapters to existing application systems including (but not limited to): Oracle E-Business Suite, PeopleSoft, JD Edwards, and Siebel
- Oracle's purchase of BEA provides some of the best legacy and ERP adapters available



- Oracle Fusion Middleware includes the SOA Suite:
 - BAM (Business Activity Monitoring) providing real-time access to business performance information
 - BPEL (Business Process Execution Language) Process Manager for defining and executing business processes
 - Business Rules Engine to manage business rules
 - Web Services Manager for security (Oracle Directory, Active Directory, LDAP) and management
 - ESB (Enterprise Service Bus) to provide routing/messaging
- JDeveloper provides a unified SOA Suite toolset
- The Oracle Service Registry, Oracle Portal, and other products are also available



- Fusion Applications are the next generation of Oracle's Applications products
 - Oracle E-Business Suite
 - PeopleSoft
 - JD Edwards Enterprise
 - JD Edwards World
 - Siebel
 - Retek
 - more...
- Rather than “stitching together” disparate technologies, Fusion uses a service-oriented architecture to make the functionality of the various tools available



- Oracle Fusion Applications rely heavily on Oracle Fusion Middleware; the opposite is not true
- An organization may use Oracle Fusion Middleware and its wide array of tools even if Oracle Fusion Applications are not installed
- Oracle Fusion Middleware's reliance on industry standards (like SOAP, WSDL, and UDDI) and SOA makes it an excellent choice no matter how applications are supported in an organization

*Fusion Middleware is already good, but will get better and better since it is the lynchpin that thousands of Oracle's developers are building Fusion Applications with. **We all win!***



- BPEL consists of specific steps:
 - Invoking web services <invoke>
 - Waiting for client to invoke a web service via a message <receive>
 - Generating responses for synchronous operations <reply>
 - Manipulating variables <assign>
 - Signaling faults and exceptions <throw>
 - Pausing for selected time <wait>
 - Ending the process <terminate>



- BPEL provides common programming constructs:
 - Sequence <sequence>
 - Flow <flow>
 - Path selection <pick>
 - Case construct <switch>
 - Looping <while>



- BPEL XML syntax is “clunky” and error prone
- BPEL tools are:
 - Graphical (pretty):
 - Paint-by-the-numbers
 - Abstractions of BPEL resulting in BPEL
- Some BPEL tools provide ESB-like features enabling them to fulfill integration roles
- Some tools extend the basic BPEL functionality
- BPEL tools allow the creation of Composite Processes; a collection of coordinated service invocations and related activities that provide useful business process functionality



- Oracle BPEL Process Manager and Oracle (BEA) Aqualogic
- Microsoft BizTalk
- IBM WebSphere Process Server
- IBM AlphaWorks BPWS4J



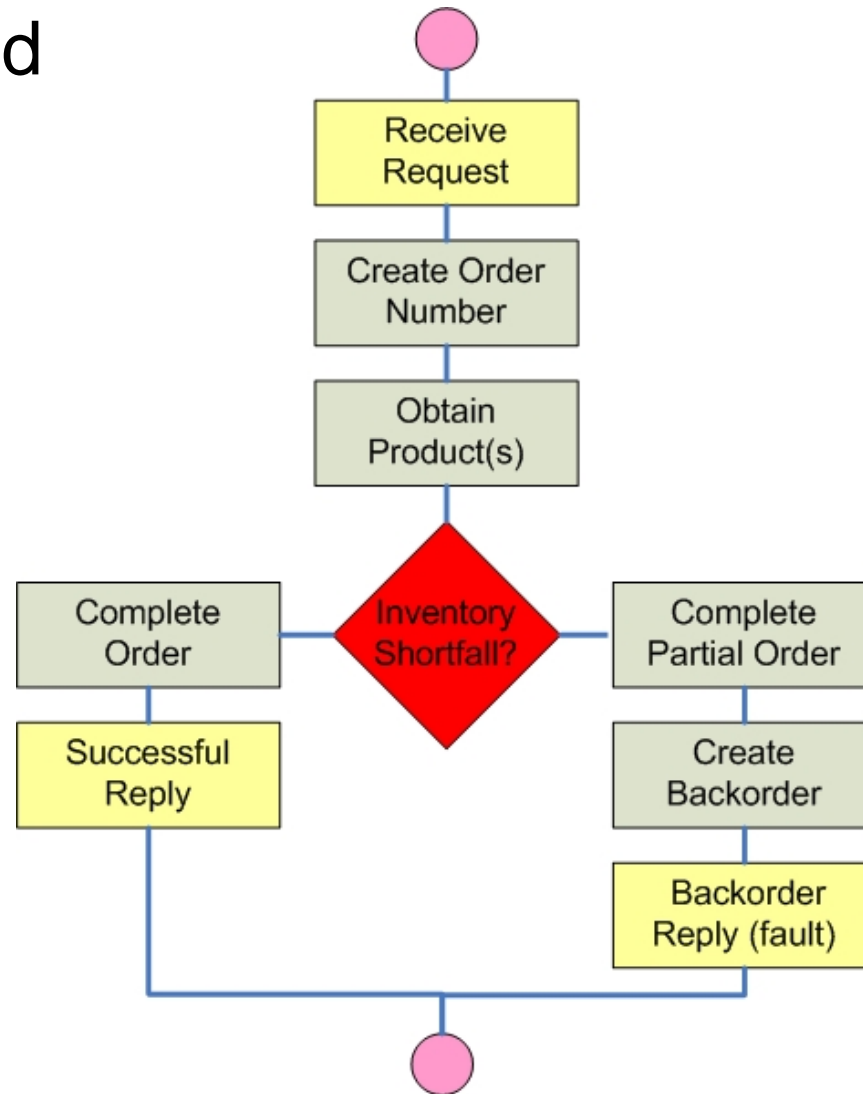
- Oracle provides two types of BPEL support:
 - Oracle BPEL Process Manager
 - Execution environment for BPEL processes
 - Supports BPEL version 1.1
 - Used to monitor, manage, and deploy BPEL processes
 - Designed for Oracle Application Server and OC4J
 - Versions available for JBoss and BEA (Oracle) WebLogic Server
 - Oracle BPEL Designer
 - Development environment for BPEL processes using GUI
 - Develops standard BPEL interoperable with other BPEL tools
 - Works with Oracle JDeveloper
 - Available as plug-in for Eclipse platform



- Oracle's BPEL tool began as the “Collaxa BPEL Designer & Web Service Orchestration Server” tool acquired by Oracle in 2004
 - Fully-compliant BPEL implementation
 - Has been “converged” with BEA BPEL offering
 - Oracle has added Human Workflow and Database Interface services
 - Easy-to-use modeler
 - Scalable
 - Flexible
 - Monitoring available



- Services are composed to support a complete Business Process





- Create Application using JDeveloper
- Create a SOA Composite project:
File->New->Project->SOA Tier->SOA Composite
- Define the BPEL process with input and output processes
- Use the GUI to complete the composition; assign values to input/output variables
- Save and deploy the application
- Use the SOA Console to review and test composite services



- Partner Links
- Variables
- Activities
 - Service Activities
 - Structure Activities
 - Fault Activities
- Expression Language
- Correlation Set
- Scope
- Fault Handler
- Compensation Handler
- Event Handler



- BPEL processes are called by clients and also use services themselves
- Services and clients that interact with BPEL processes are referred to as “Partners”
 - partnerLinks define BPEL process partners
 - Reference partners are services invoked by a process
 - Interface partners allow a process to be invoked by a client
 - BPEL processes have at least one partnerLink



- Variables are used:
 - Create request messages used for service invocation
 - Hold state information between BPEL activities
 - Receive response messages returned from services invoked



- Steps in a BPEL process are called “Activities”
- Three major groups of activities are used:
 - Service Activities Communicate with partners
 - Structure Activities Program logic (sequence, looping, if-then-else...)
 - Fault Activities Error-handling activities
- Each Activity type is usually represented by an icon in BPEL tools



- XPath is used by BPEL processing and evaluation statements like While and Switch



- Some BPEL processes are long-running and may have multiple instances executing at once
 - A unique key is attached to messages to make sure they go to the correct instance
 - Since some services provide different operations they need additional keys for each type of message
 - Correlation Set is the term used to refer to the key field(s) used by the various messages that are used for a service



- BPEL allows the grouping of activities into “Scopes”
 - Every BPEL process begins with a “Global Scope”
 - Scopes provide a context for a group of activities
 - A scope may be associated with various “handlers” (fault handlers, compensation handlers, event handlers)



- Various handlers might be associated with a BPEL process
 - Fault Handler
 - Compensation Handler
 - Event Handler



- Service invocation or execution errors are called “faults”
- BPEL allows faults that occur within a given scope to be associated with fault handlers specific to the scope’s activities
- When faults occur
 - Scope execution stops
 - If a fault handler is available it is executed
 - If a fault handler is not available the fault is passed to the scope’s parent scope



- Compensation handlers are necessary because
 - Some processes are long-running
 - Sometimes handling of a fault requires that the work of one or more scopes must be reversed
 - Compensation handlers are defined as part of a scope and describe the “undo” activities to be performed if an unhandled fault occurs



- BPEL event handlers are defined as part of a scope to handle messages that are received when a process is “blocking” (waiting) due to the execution of a Receive or Pick (or Receive Choice) activity
 - An event handler describes what type of message it can receive
 - An event handler includes activities to be performed upon receipt of an expected message



Oracle BPEL Console v10.1.3.1.0 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://teqram3.kinghome.private.johnjayking.com/BPELConsole/default/index.jsp

ORACLE Enterprise Manager 10g BPEL Control

Manage BPEL Domain | Logout | Support
Logged to domain: default

Dashboard BPEL Processes Instances Activities

Deployed BPEL Processes	In-Flight BPEL Process Instances		
Name	Instance	BPEL Process	Last Modified ↑
CreditRatingService			
StockQuoteService			
TaskActionHandler			
TaskManager			

Deploy New Process

Oracle BPEL Console v10.1.3.1.0



- The Oracle BPEL Console provides a browser-based management capability supported by JSPs and Servlets on the Application Server hosting BPEL
- BPEL Console includes:
 - Management and administration of BPEL processes
 - Debugging for processes
 - Audit trails and process history
 - GUI representation of process flows



Oracle JDeveloper - CustomerService.jws : CustomerService.jpj

File Edit View Search Navigate Run Debug Refactor Versioning Tools Window Help

Applications Navig... Connections CreditFlow.bpel CreditRatingService.bpel CreditRatingService.wsdl

Applications

- CreditRatingService
 - CreditRatingService
 - Resources
 - bpel.xml
 - CreditRatingService.bpel
 - CreditRatingService.wsdl
- CustomerService
- MyBPELApp

Services

client

receiveInput

Service

Component Palette

Process Activities

- Pointer
- Assign
- Compensate
- Decide
- Email
- Empty
- Event

Property Inspector

Help Center

- Working with Oracle WebCenter Framework
- Getting Started with Oracle JDeveloper
- Cue Cards
- Working with Service Oriented Architecture

CreditRatingService.bpel - Structure

BPEL Structure

- Variables
- Message Types
- Correlation Sets
- Schemas
- Partner Links
- Activity Structure
- Properties
- Property Aliases
- Sensor Actions
- Sensors
- Test Suites

Diagram Source History

BPEL Messages - Log

XPath	Type	# of Errors
/process/partnerLinks/partnerLink	Partner Link	1
/process/sequence/switch/case/sequence/reply	Reply	1
/process/sequence/switch/otherwise/sequence/reply	Reply	1

Errors: 3 Warnings: 0 Last Validated On: 10 Apr 2008 11:47:54 GMT

Validation Errors Log Messages Search Results

Messages BPEL Messages

Source BPEL

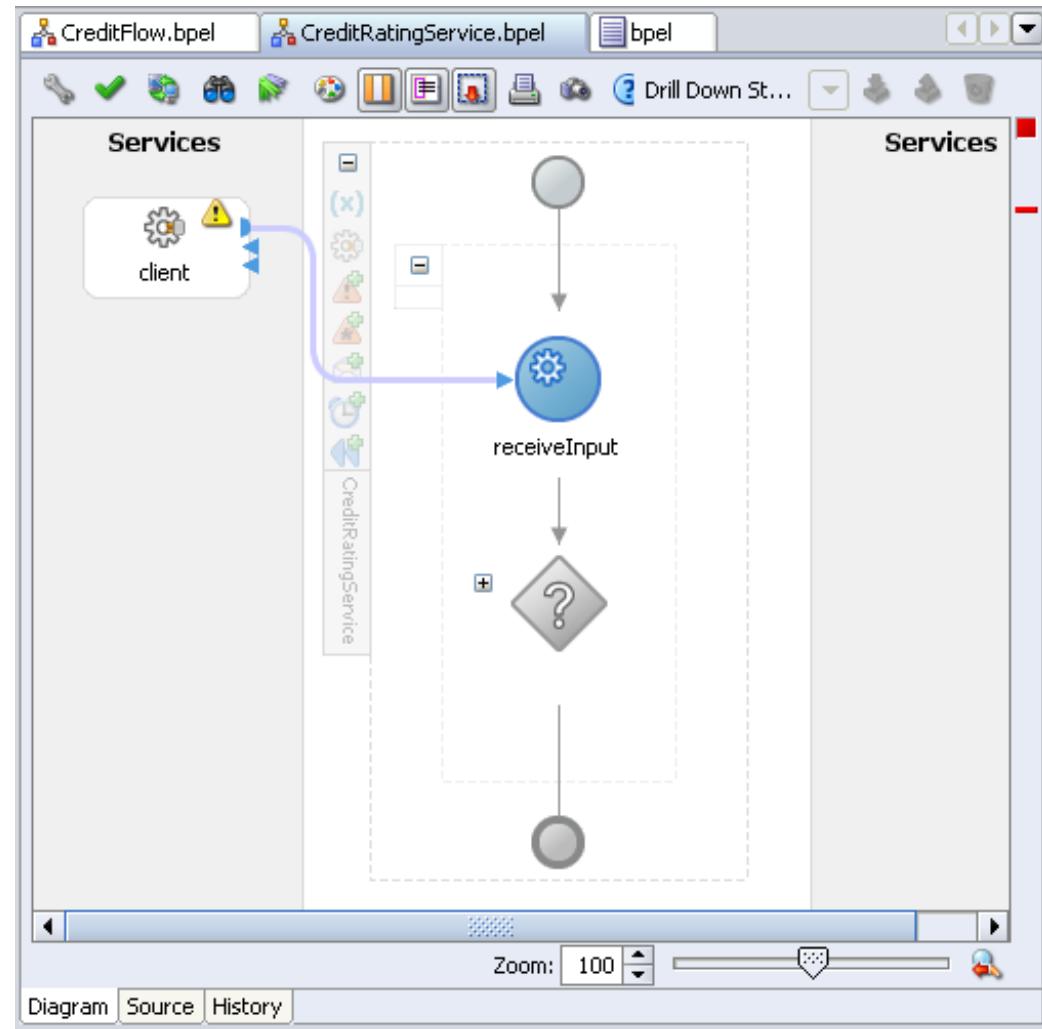
BPEL editor



- BPEL Designer is built-in to Oracle JDeveloper and available as a plug-in to Eclipse
- BPEL Designer allows:
 - Graphical development of BPEL processes (BPEL XML code is generated automatically)
 - Activities, PartnerLinks, Services, and Faults may be added using drag-and-drop

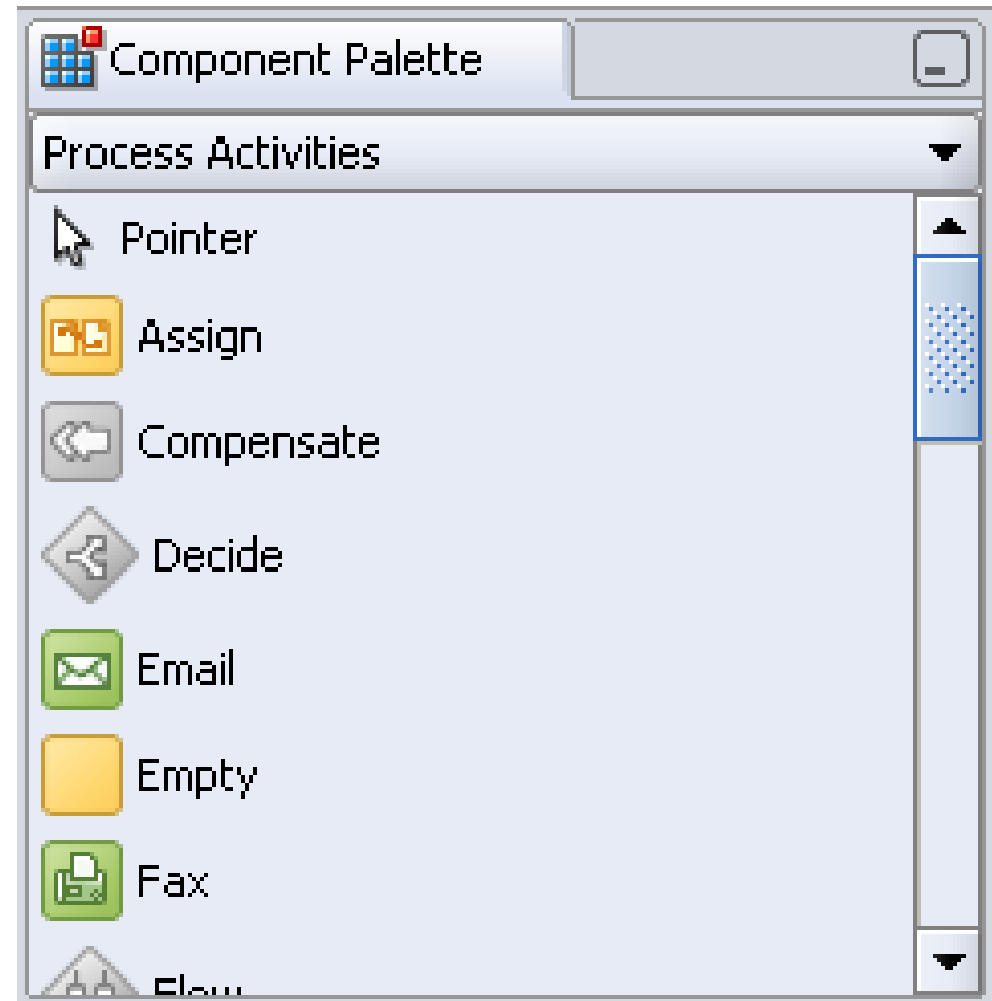


- Processes may be combined in the BPEL Designer to create new processes





- The component palette provides visual representations of Activities, Faults, and other BPEL components





- BPEL XML source may be viewed on the “Source” tab

```
<!-- CreditRatingService BPEL Process -->
<process name="CreditRatingService"
  targetNamespace="http://services.otn.com"
  suppressJoinFailure="yes"
  xmlns:tns="http://services.otn.com"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-pr
  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/busine

<!-- List of services participating in this BPEL process --
<partnerLinks>
  <!-- The 'client' role represents the requester of this s
  <partnerLink name="client"
    partnerLinkType="tns:CreditRatingService"
    myRole="CreditRatingServiceProvider"/>
</partnerLinks>

<!-- List of messages and XML documents used as part of thi
<variables>
  <!-- Reference to the message passed as input during init
  <variable name="input"
    messageType="tns:CreditRatingServiceRequestMessage"/

  <!-- Reference to the message that will be sent back to t
```



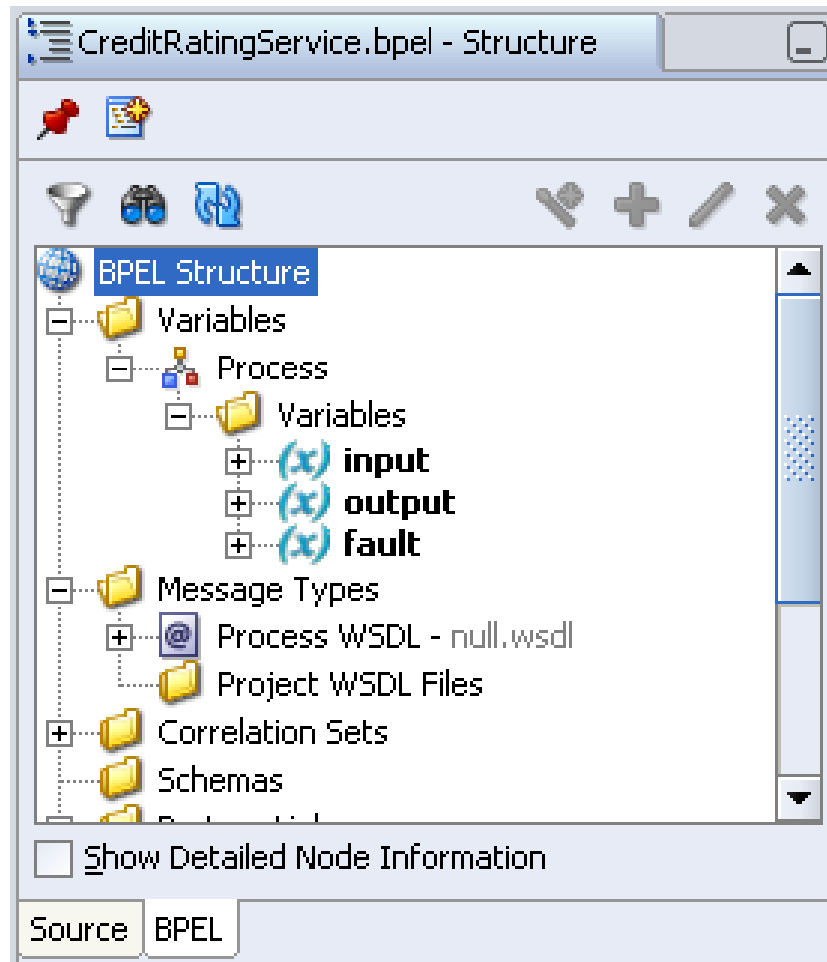
Applications Navig... Connections

Applications

- CreditRatingService
 - CreditRatingService
 - Resources
 - bpel.xml
 - CreditRatingService.bpel
 - CreditRatingService.wsdl
- + CustomerService
- + MyBPELApp



- The Structure View shows all of a BPEL processes components





- Oracle uses its skills in the database world to store schemas that support BPEL in the database including
 - “Dehydration” where process state is stored
 - Non-Oracle database products are supported by Oracle BPEL Process Manager including:
 - Oracle Database Lite
 - IBM DB2 UDB
 - Microsoft SQL Server

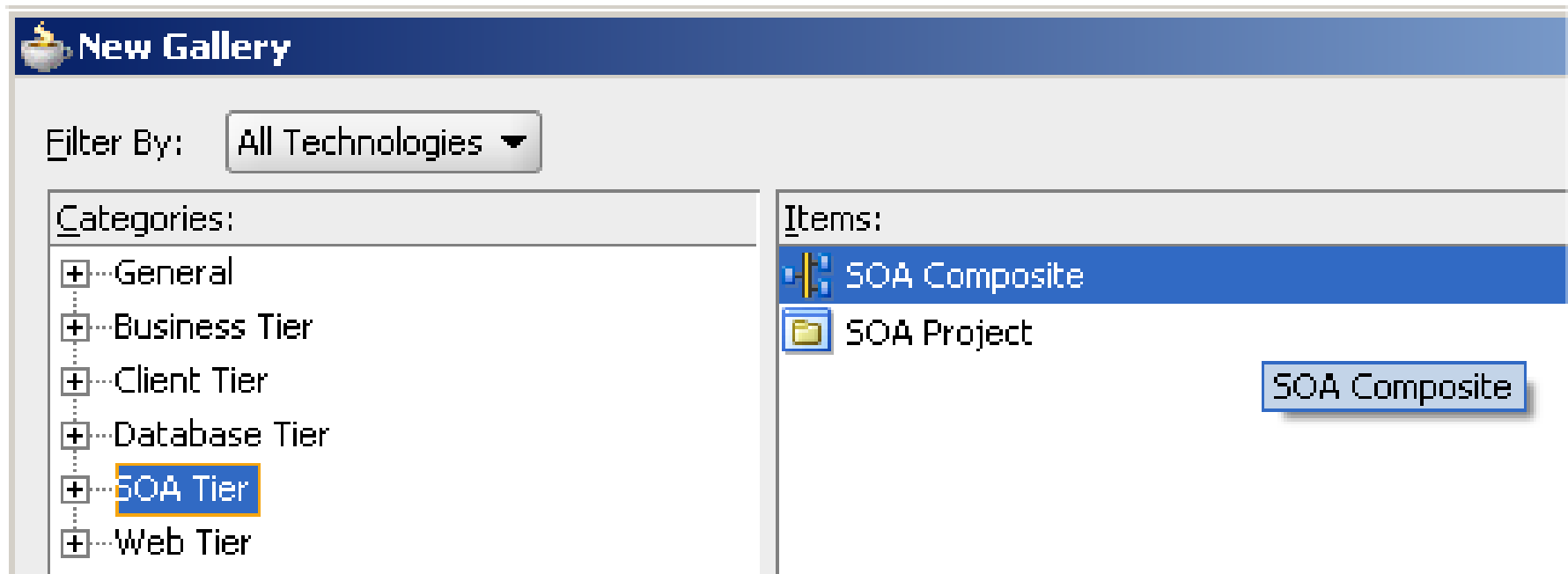


- To create a composite process:
 - Create Application using JDeveloper
 - Create a SOA Composite project
File->New->Project->SOA Tier->SOA Composite
 - Define the BPEL process with input and output processes
 - Use the GUI to complete the composition; assign values to input/output variables
 - Save and deploy the application
 - Use the SOA Console to review and test composite services

These pages illustrate one of Oracle's example composite processes



- Create a SOA Composite project
File->New->Project->SOA Tier->SOA Composite



Define Inputs & Outputs



Create BPEL Process

BPEL Process

A BPEL process is a service orchestration, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

General

Name: SOAHelloProcess

Namespace: http://xmlns.oracle.com/SOAHelloApp/SOAHelloComposite/SOAHelloProcess

Template: Asynchronous BPEL Process

Expose as Composite Service

Input: SOAHelloApp/SOAHelloComposite/SOAHelloProcess}SOAHelloProcessProcessRequest

Output: SOAHelloApp/SOAHelloComposite/SOAHelloProcess}SOAHelloProcessProcessResponse

Buttons: Help, OK, Cancel

Assign Activity



Oracle JDeveloper 11g Technology Preview 3 - SOAHelloApp.jws : SOAHelloComposite.jpr : C:\jdevinstance\mywork\SOAHelloApp\SOAHelloComposite\SOAHelloProcess...

File Edit View Search Navigate Run Debug Refactor Versigning Tools Window Help

Application Navigator

SOAHelloApp

Projects

- SOAHelloComposite
 - SOA Content
 - testsuites
 - xsd
 - xsl
 - .designer
 - SCA-INF
 - composite.xml
 - SOAHelloProcess.bpel
 - SOAHelloProcess.componentType
 - SOAHelloProcess.wsdl

Structure

SOAHelloProcess.bpel

- Variables
- Message Types
- Correlation Sets
- Schemas
- Partner Links
- Activity Structure
- Properties
- Property Aliases
- Sensor Actions
- Sensors

Services

References

client

main

receiveInput

callbackClient

Component Palette

BPEL

- BPEL Activities and Components
 - Activities and Components
 - BPEL Process
 - Business Rule
 - Human Task
 - Mediator
 - BPEL Activities
 - Assign
 - Bind Entity
 - Compensate
 - Email
 - Empty
 - Fax
 - Flow
 - FlowN
 - IM
 - BPEL Services

Design Source History

BPEL Messages - Log

Property Inspector

XPath	Type	# of Errors
/process/sequence/invoke	Invoke	0

Errors: 0 Warnings: 1

Last Validated On: 24 Feb 2008 07:29:49 GMT

Validation Errors Log Messages Search Results

Messages Extensions SOA Configuration BPEL Messages

Added to project, SOAHelloComposite.jpr.

BPEL editor

Adding Activity



Oracle JDeveloper 11g Technology Preview 3 - SOAHelloApp.jws : SOAHelloComposite.jpr : C:\jdevinstance\mywork\SOAHelloApp\SOAHelloComposite\SOAHelloProces...

File Edit View Search Navigate Run Debug Refactor Versigning Tools Window Help

Application Navigator

SOAHelloApp

Projects

- SOAHelloComposite
 - SOA Content
 - testsuites
 - xsd
 - xsl
 - .designer
 - SCA-INF
 - composite.xml
 - SOAHelloProcess.bpel
 - SOAHelloProcess.componentType
 - SOAHelloProcess.wsdl

- Application Resources
- Data Controls
- Recently Opened Files

Structure

SOAHelloProcess.bp...

- SOAHelloProcess.bpel
 - Variables
 - Message Types
 - Correlation Sets
 - Schemas
 - Partner Links
 - Activity Structure
 - Process - SOAHelloProcess
 - Properties
 - Property Aliases
 - Sensor Actions
 - Sensors

- Show Detailed Node Information

Source BPEL

Services

client

main

receiveInput

Assign_1

callbackClient

References

Component Palette

BPEL

- BPEL Activities and Components
 - Activities and Components
 - BPEL Process
 - Business Rule
 - Human Task
 - Mediator
 - BPEL Activities
 - Assign
 - Bind Entity
 - Compensate
 - Email
 - Empty
 - Fax
 - Flow
 - FlowN
 - IM
 - BPEL Services

Design Source History

BPEL Messages - Log

Property Inspector

SOAHelloProcess.bpel

XPath	Type	# of Errors
/process/sequence/invoke	Invoke	0

Errors: 0 Warnings: 1 Last Validated On: 24 Feb 2008 07:29:49 GMT

Validation Errors Log Messages Search Results

Messages Extensions SOA Configuration BPEL Messages

Added to project, SOAHelloComposite.jpr.

BPEL editor



Assign [Close]

⚠ Errors: 1

General | **Copy Operation** | Sensors | Annotations

[+] [Edit] [Delete] [Up] [Down]

From	To

- Copy Operation...
- Append Operation...
- Insert-After Operation...
- Insert-Before Operation...
- CopyList Operation...
- Remove Operation...
- Rename Operation...

Copy Operation Expression



Expression Builder

Expression:
concat('Hello ',)

Insert Into Expression

BPEL Variables

- Variables
 - Process
 - Variables
 - inputVariable
 - payload
 - client:SOAHelloProcessProcessRequest
 - client:input

Functions

Advanced Functions

- create-nodeset-from-delimited-string
- element-available
- function-available
- generate-guid
- generate-id
- lookup-xml
- system-property

Content Preview:
bpws:getVariableData('inputVariable','payload','/client:SOAHelloProcessP

Description:
Variable XPath expression

Help OK Cancel



Create Copy Operation

From

Type: Expression

Expression:

```
concat('Hello ', bpws:getVariableData('inputVariable','payload','/client:SOAHelloProcessProcessRequest/client:input'))
```

To

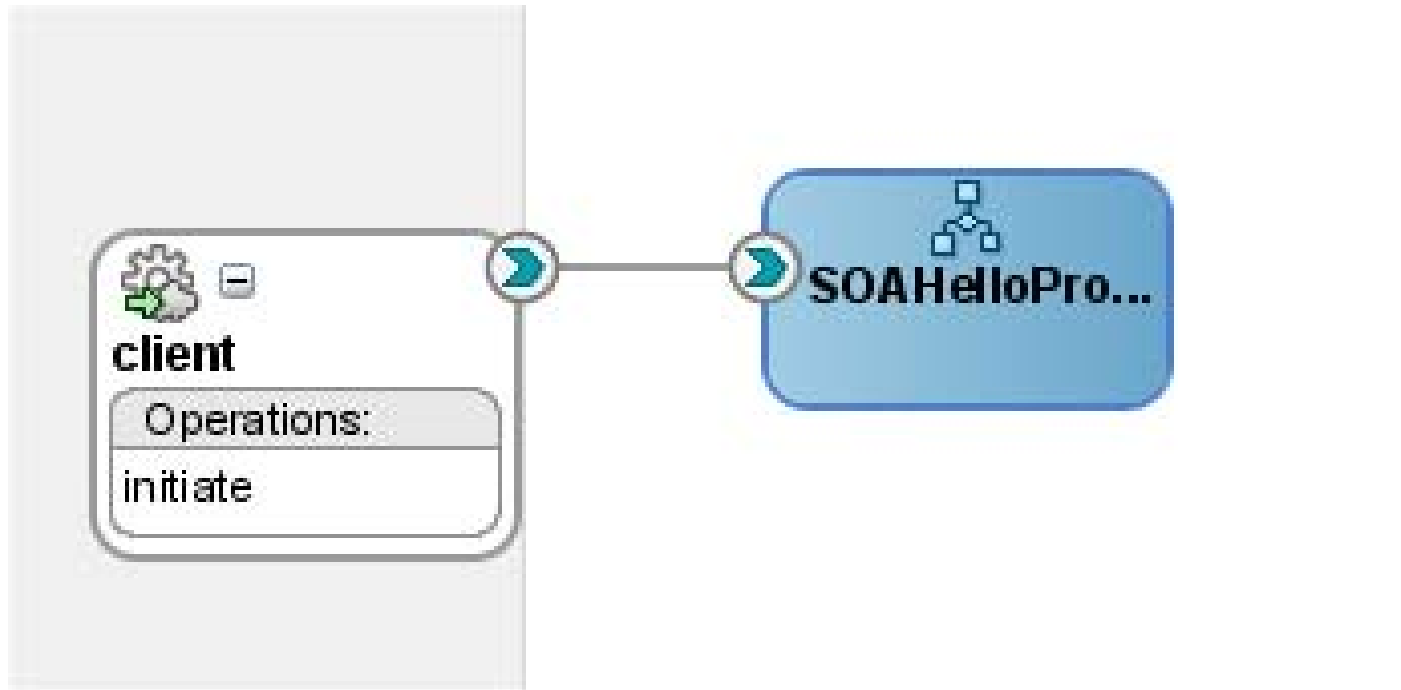
Type: Variable

- Variables
 - Process
 - Variables
 - inputVariable
 - outputVariable
 - payload
 - client:SOAHelloProcessProcessResponse
 - client:result

Show Detailed Node Information

XPath:

Help OK Cancel





- Open the SOA Console to test composite

<http://localhost:8988/SOAConsole/>



- Oracle's BPEL Process Manager provides a complete standards-based solution to creating and orchestrating services and composite services
 - Oracle BPEL Console tool is used to manage BPEL services and compositions
 - Oracle BPEL Designer is used to create services and the code that supports them
 - Oracle's Workflow capabilities add a richness and realism that is often missing from BPEL process flows
 - Oracle's database provides robust support for BPEL



Training Days 2010

Save the dates!

February 17-18 2010!



Save the Date: April 18-22 2010

Las Vegas, Nevada!



- Web Architecture
- Oracle Tools
- Professional Development
- Web and Java
- Essbase
- Application Express
- Web and Not Java
- Hyperion
- Business Intelligence and Data Warehousing
- Database Server
- BEA
- Best Practices
- Third Party Tools

Monterey, CA
June 21–25, 2009

**SPOTLIGHT ON
DEVELOPERS**

www.odtugkaleidoscope.com



Oracle BPEL “Nuts and Bolts”

Paper 743

To contact the author:

John King

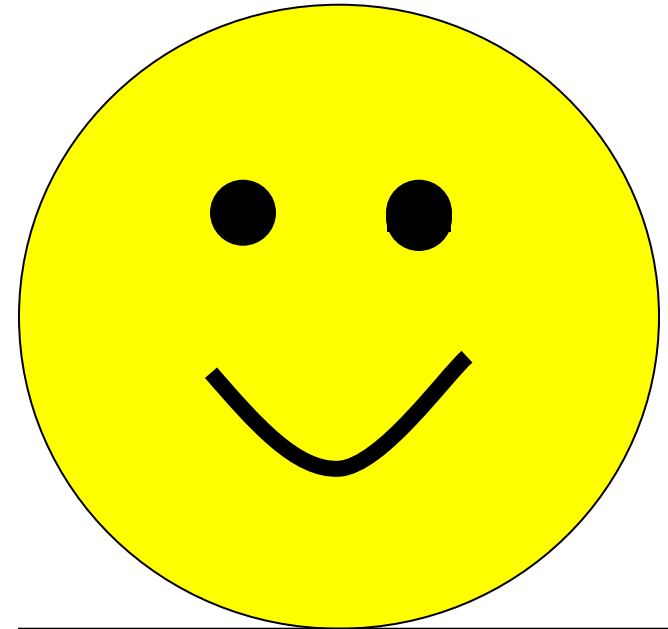
King Training Resources

6341 South Williams Street

Littleton, CO 80121-2627 USA

1.800.252.0652 - 1.303.798.5727

Email: john@kingtraining.com



Thanks for your attention!

Today's slides are on the web:

<http://www.kingtraining.com>