eBusiness Suite goes SOA

Ulrich Janke
Oracle Consulting
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Presentation Agenda

- Motivation
- Integrated SOA Gateway
- Integration Repository
- SOA Provider
- Service Invocation Framework
- Roadmap
- Q & A
Motivation
Who’s Deploying Integration Solutions

Public | Private | Government

Healthcare | Transportation | Education

High Tech | InfoTech | Manufacturing | Telecom

Federal | State | Local | Defence

Drivers: SOA | Agility | Scalability | Visibility
Oracle E-Business Suite Integration

Business Use Cases

- Order to Invoice Process
- Procure to Pay Process
- Payroll Processing Process
- Hire to Retire Process
- After Sales Service Process
- Sub Contracting Process
Create Supplier Ship & Debit Request
Oracle E-Business Suite Integrated SOA Gateway Use Case

**Inception**
Integration Administrator

1. Identify / Locate API for integration
2. Generate WSDL for Ship & Debit API
3. Deploy Service for Ship & Debit API

**Design**
Integration Developer

1. Locate WSDL URL for SDR: Integration Repository
2. Set SOA Header for SD Service SOAP request
3. Read SD input payload from file
4. Set SD Request Number
5. Invoke EBS API as WS
6. Get SD Request Header ID
7. Create and deploy BPEL process

**Execution**
Integration Administrator

1. Initiate BPEL process from BPEL Console
2. View response in BPEL Console

Trade Management User

1. Verify created SD request in Oracle Trade Management
Interoperability Leverage

- ORACLE FUSION MIDDLEWARE
- ORACLE E-BUSINESS SUITE
- ORACLE PEOPLESOFTE ENTERPRISE
- SIEBEL
- Apache Axis
- .NET WS Client
- 3rd Party Standard WS Client
Key Integration Challenges

- Source of Truth
- Custom Interfaces
- Security Concerns
- Monitor & Manage
- Process Monitoring
- Complexity
- Provide Services
- Consume Services
- Fail-safe Integration
- Process Orchestration
- Event Driven Integration
- Flexible Architecture
Oracle Application Integration Architecture
Pre-built integrations
Integrated SOA Gateway
Oracle E-Business Suite Integrated SOA Gateway

The above reflects Oracle’s current development plans which are subject to change at any time.
Business Benefits
Oracle E-Business Suite Integrated SOA Gateway

Flexible Integration Architecture
- Provide out-of-box Web services
- Consume Web services
- Complements with AIA
- Path to Fusion

Lower Total Cost Of Ownership
- Maximize investment of existing IT assets
- Leverage interoperability with standards based integration

Gain Visibility Into Integration
- Monitor integration transactions
- Comprehensive details about integration transactions

More Interfaces
- Business Events’ – “Subscription Model”
- Composite services – simplify complexity
- More Services
Integration Repository
Integration Repository
Your Snapshot with Customization

- Catalog of all annotated public integration interfaces
- Search / Browse by Product Family / Interface Type
- Part of EBS Integrated SOA Gateway – R12.1.1
Integration Repository
Publishing Custom Interfaces

Custom interfaces in E-Business Suite

- Annotate Interfaces
  - PL/SQL API
  - Concurrent Program
  - XML Message
  - Business Service Object
  - Business Event

- Stand Alone Parser
  - Reads annotated files and generates Integration Repository Loader Files

- Loader File (ildt)
  - FNDLOAD Uploads Loader File to Integration Repository

- Integration Repository
DEMONSTRATION
Integration Repository
Customization of Integration Interfaces
-bash-3.00$ pwd
/tmp/install
-bash-3.00$ ls -l
 total 324
-rw-r--r-- 1 app2176 ems2176 45290 Oct 5 2009 Class-MethcMethodMaker-1.12.tar.gz
-rw-r--r-- 1 app2176 ems2176 207584 Oct 5 2009 Compress-Raw-Zlib-2.009.tar.gz
-rw-r--r-- 1 app2176 ems2176 64039 Oct 5 2009 Compress-Zlib-2.009.tar.gz
-bash-3.00$ gzip -d Compress-Raw-Zlib-2.009.tar.gz
-bash-3.00$ ls -l
 total 932
-rw-r--r-- 1 app2176 ems2176 45290 Oct 5 2009 Class-MethcMethodMaker-1.12.tar.gz
-rw-r--r-- 1 app2176 ems2176 829440 Oct 5 2009 Compress-Raw-Zlib-2.009.tar
-rw-r--r-- 1 app2176 ems2176 64039 Oct 5 2009 Compress-Zlib-2.009.tar.gz
-bash-3.00$ Set Up & Configure

- Download and Install Parser Libraries
- Configure the Libraries
- Compile the patch installation
CREATE or REPLACE PACKAGE Custom_OE_Order_PUB AS

PROCEDURE Process_Order
  (  p_org_id IN NUMBER := NULL
      , p_operating_unit IN VARCHAR2 := NULL
      , p_api_version_number IN NUMBER
      , p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE
      , p_return_values IN VARCHAR2 := FND_API.G_FALSE
      , p_action_commit IN VARCHAR2 := FND_API.G_FALSE
      , x_return_status OUT NOCOPY VARCHAR2
      , x_msg_count OUT NOCOPY NUMBER
      , x_msg_data OUT NOCOPY VARCHAR2
      , p_header_rec IN Header_Rec_Type := G_MISS_HEADER_REC
      , p_old_header_rec IN Header_Rec_Type := G_MISS_HEADER_REC
      , p_header_val_rec IN
      , p_old_header_val_rec IN
      , p_Header_Adj_tbl IN
      , p_old_Header_Adj_tbl IN
      , p_Header_Adj_val_tbl IN
      , p_old_Header_Adj_val_tbl IN
      , p_Header_price_Attribute_tbl IN
  )
SET VERIFY OFF
WHENEVER OSERROR EXIT FAILURE ROLLBACK;
WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

CREATE or REPLACE PACKAGE Custom_OE_Order_PUB AS
    /*
    * <Put your long package description here
    * it can span multiple lines>
    * @rep:scope <scope>
    * @rep:product <product or pseudoproduct short code>
    * @rep:lifecycle <lifecycle>
    * @rep:displayname <display name>
    * @rep:compatibility <compatibility code>
    * @rep:businessevent <Business event name>
    * @rep:category BUSINESS_ENTITY <entity name>
    */

PROCEDURE Process_Order
    ( p_org_id IN NUMBER := NULL
    , p_operating_unit IN VARCHAR2 := NULL
    , p_api_version_number IN NUMBER
    , p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE
    , p_return_values IN OUT VARCHAR2
    , p_action_commit IN OUT NUMBER
    , x_return_status OUT NUMBER
    , x_msg_count OUT NUMBER
    , x_msg_data OUT VARCHAR2
    , p_header_rec IN OUT
    , p_old_header_rec IN OUT
    , p_header_val_rec IN OUT
);
```
1 SET VERIFY OFF
2 WHENEVER OSERROR EXIT FAILURE ROLLBACK;
3 WHENEVER SQLERROR EXIT FAILURE ROLLBACK;

4 CREATE or REPLACE PACKAGE Custom_Package
5   /
6   * This is a custom API which allows users to perform process order on sales orders in the Order Management system.
7   * @rep:scope public
8   * @rep:product ONT
9   * @rep:lifecycle active
10  * @rep:displayname Process Order API
11  * @rep:category BUSINESS_ENTITY ONT_SALES_ORDER
12  */
13
14 PROCEDURE Process_Order
15  ( p_org_id IN NUMBER := NULL,
16   , p_operating_unit IN VARCHAR2 := NULL,
17   , p_api_version_number IN NUMBER,
18   , p_init_msg_list IN VARCHAR2 := FND_API.G_FALSE,
19   , p_return_values IN VARCHAR2 := FND_API.G_FALSE,
20   , p_action_commit IN VARCHAR2 := FND_API.G_FALSE,
21   , x_return_status OUT NOCOPY VARCHAR2,
22   , x_msg_count OUT NOCOPY NUMBER,
23   , x_msg_data OUT NOCOPY VARCHAR2,
24   , p_header_rec IN Header Rec,
25   , p_old_header_rec IN Header Rec,
26   , p_header_val_rec IN Header Rec,
27   , p_old_header_val_rec IN Header Rec,
28   , p_Header_Adj_tbl IN Header_Adj_tbl
29 )
30 /
31 ```
SET VERIFY OFF
WHENEVER QSEVERR EXIT FAILURE ROLLBACK;
WHENEVER SQLERROR EXIT FAILURE ROLLBACK;
CREATE or REPLACE PACKAGE Custom_OE_Order/pub AS
/
/*
 * This is a custom API which allows users to perform process order on sales orders in the Order Management system
 * @rep:scope public
 * @rep:product ONT
 * @rep:lifecycle active
 * @rep:displayname Process Order API
 * @rep:category BUSINESS_ENTITY ONT_SALES_ORDER
 */
/**
 * <Put your long procedure description here
 * it can span multiple lines>
 * @param <param name 1> <param description 1>
 * @param <param name 2> <param description 2>
 * @scp:scope <scp>
 * @rep:product <product or pseudoproduct short code>
 * @rep:lifecycle <lifecycle>
 * @rep:displayname <display name>
 * @rep:compatibility <compatibility code>
 * @rep:businessevent <Business event name>
 */
PROCEDURE Process_Order
  p_org_id IN NUMBER,
  p_operating_unit IN VARCHAR2,
  p_api_version_number IN NUMBER,
  p_init_msg_list IN VARCHAR2,
  p_return_values IN VARCHAR2,
  p_action_commit IN VARCHAR2
AS
Use this procedure to process Sales Order. Process Order supports action requests that allow users to execute actions such as booking, hold application and removal, automatic attachments, fulfillment set application and removal, match and reserve for Configured items, get ship method and freight rates (both individually and together), and the linking and delinking of Configured items.

- **p_org_id** Input OU Organization Id
- **p_operating_unit** Input Operating Unit Name
- **p_api_version_number** API version used to check call compatibility
- **p_init_msg_list** Boolean which determines whether internal message tables should be initialized
- **p_return_values** Boolean which determines whether ids should be converted to values for populat
- **p_action_commit** Not used
- **p_return_status** Return status of API call
- **p_msg_count** Number of stored processing messages
- **p_msg_data** Processing message data
- **p_header_rec** Input record structure containing current header-level ID information for an o
- **p_old_header_rec** Input record structure containing old header-level ID information for
- **p_header_val_rec** Input record structure containing current header-level value informati
- **p_old_header_val_rec** Input record structure containing old header-level value informati for p
- **p_Header_Adj_tbl** Input table containing current header-level ID information for price a
- **p_old_Header_Adj_tbl** Input table containing old header-level ID information for price adjus
- **p_Header_Adj_val_tbl** Input table containing current header-level value information for psc
- **p_old_Header_Adj_val_tbl** Input table containing current header-level value information for psc
- **p_Header_price_Attribute** Input the price at which a particular item is being pric
- **p_old_Header_Price_Attribute** Input the price at which a particular item was pricin
- **p_Header_Adj_Attribute** Input the attribute information for an item between
- **p_old_Header_Adj_Attribute** Input the attribute information for an item betw
- **p_Header_Scredit_tbl** Input table containing current header-level ID information f
- **p_old_Header_Scredit_tbl** Input table containing current header-level value infor
Run the Integration Repository Parser

Custom PL/SQL API Annotation
Generate ILDT files

Custom PL/SQL API Annotation

- Generate the ILDT files
Run FNDLOAD to load the ILDT files into the Integration Repository

Custom PL/SQL API Annotation

- Run FNDLOAD to load the ILDT files into the Integration Repository
The Integration Repository is an integral part of Oracle E-Business Suite. It provides a complete catalog of Oracle E-Business Suite's Business Service interfaces. The tool lets users easily discover and deploy the appropriate business service interface from the catalog for integration with any system, application, or business partner.
Search by Interface Source = Custom and Interface Type – PL/SQL

Custom PL/SQL API Annotation

- Search by Interface Source = Custom and Interface Type – PL/SQL
Custom PL/SQL API is available in the Integration Repository.

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Internal Name</th>
<th>Product Family</th>
<th>Product</th>
<th>Interface Type</th>
<th>Web Service Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Order API</td>
<td>CUSTOM_OE_ORDER_FUB</td>
<td>All</td>
<td>All</td>
<td>PL/SQL Custom</td>
<td>All</td>
<td>This is a custom API which allows users to perform process order on sales orders in the Order Management system.</td>
</tr>
</tbody>
</table>
Custom PL/SQL API details for the Package level

This is a custom API which allows users to perform process order on sales orders in the Order Management system.
Custom PL/SQL API Annotation

- Custom PL/SQL API details with the parameters and procedures

**Full Description**

Use this procedure to process Sales Order. Process Order supports action requests that allow users to execute a variety of actions such as booking, hold application and removal, automatic attachments, fulfillment set application and removal, match and reserve for Configured items, get ship method and freight rates (both individually and together), and the linking and un-linking of Configured items.

**Signature**

**Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Precision/Size</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PОРG_ID</td>
<td>NUMBER</td>
<td>In</td>
<td></td>
<td>NULL</td>
<td>Input OU Organization Id</td>
</tr>
<tr>
<td>P_OPERATING_UNIT</td>
<td>VARCHAR2</td>
<td>In</td>
<td></td>
<td>NULL</td>
<td>Input Operating Unit Name</td>
</tr>
<tr>
<td>P_API_VERSION</td>
<td>NUMBER</td>
<td>In</td>
<td></td>
<td>NULL</td>
<td>API version used to check call compatibility</td>
</tr>
<tr>
<td>P_INIT_MSG_LIST</td>
<td>VARCHAR2</td>
<td>In</td>
<td></td>
<td>FND_API.G_FALSE</td>
<td>Boolean which determines whether internal message tables should be initialized</td>
</tr>
<tr>
<td>P_RETURN_VALUES</td>
<td>VARCHAR2</td>
<td>In</td>
<td></td>
<td>FND_API.G_FALSE</td>
<td>Boolean which determines whether PK should be converted to values for population in the output value parameters</td>
</tr>
<tr>
<td>P_ACTION_COMMIT</td>
<td>VARCHAR2</td>
<td>In</td>
<td></td>
<td>FND_API.G_FALSE</td>
<td>Not used</td>
</tr>
</tbody>
</table>
SOA Provider
Provide Web services
SOA Provider
Generate, Deploy Service

1. Generate WSDL
2. Deploy Service
3. Invoke Service

Oracle
Composite Services – BPEL
Published Integration Repository

Coarse-grained abstracted service within which multiple finer-grained services are bonded together to execute in a series

• Simplifies business services implementation requiring multiple APIs
• Provides flexibility in business integration architecture
• Catalog of BPEL composite service
SOA Monitor

Insight into SOA Provider transactions

- Provides an interface to audit / monitor integrations
- Provides easy to configure built-in administrative tool
  - Full Details Captured – Including Request & Response Payloads
  - On / Off Switch: Control from GUI
  - Configurable: In-Memory Cache Size, Flush Interval
  - Purge Stale Data
SOA Monitor
How does it work

**PROCESS**

1. **SOA Provider receives SOAP Request**
   - SOA Monitor captures SOAP Request details

2. **SOA Provider processes SOAP Request**
   - Error & exception details

3. **SOA Provider sends SOAP Response**
   - SOA Monitor captures SOAP Response details
Service Invocation Framework
Service Invocation Framework
Invoke And Consume Web services

- PL / SQL
- Workflow
- Forms
- OA Framework

Invoke Web service
Web Service Request
Web Service Response
Optional Transform-In & Transform-Out

External Web services
Fire Wall
- Reservation Service
- Packaged Applications Web services
- BPEL Process as Service
Service Invocation Framework

1. Create / Use Invoker Event
   - 2.1 Enter WSDL
   - 2.2 Select Service
   - 2.3 Select Port
   - 2.4 Select Operation
   - 2.5 Enter Subscription Parameters

2. Create Invoker Subscription

3. Test Service Invocation

Create Event
A business event is an occurrence in an internet or intranet application or program that might be significant to other programs.

Create Event Subscription
An event subscription is a registration indicating that a particular event is significant to a particular system. An event subscription specifies the event to be monitored.
Service Invocation Framework

Key Features

- Supports Synchronous Request-Response
- Supports One-way/Notification
- SOAP Request XSL Transformation (Outbound)
- SOAP Response XSL Transformation (Inbound)
- Supports custom input headers
- Callback to EBS using BES
- Supports WS-Security
- Manage errors using BES Error Handler process
- Testing Framework
Service Invocation Framework
Demo Use Case Steps

1. Deploy the “Create SDR” BPEL PM Process
2. Create / Use Invoker Event & Subscription
3. Provide XSL Transform for Request & Response
4. Deploy the Workflow process for Response processing
5. Create Callback Event & Subscription
6. Test Service Invocation & verify the WS Response / Notification

For Sync Request / Response
Out-of-the-box Web services

Roadmap

- SOA Governance Solutions
- Advanced Customization
- Advanced Exception Handling And Diagnostics
- SAML Token Support
- Java APIs (Document Style)
Out-of-the-box Web services
SOA Governance

Oracle FMW SOA Governance Suite

- Analyze & Model
- Design
- Develop
- Deploy & Secure
- Retire Services
- Monitor Policies & SLAs
- Manage Change

Enterprise Manager
System Monitoring
Web Services Manager
WS Policies Security
Enterprise Repository
SOA Lifecycle Governance
Registry
UDDI

S = Services