

Using BPEL and Workflow to Build Integration Components for the E-Business Suite

A Solbourne White Paper
February, 2008

Using BPEL and Workflow to Build Integration Components for the E-Business Suite

Executive Overview

Oracle Workflow is a key business process management component within the E-Business Suite. Workflow provides management of information routing and data sharing. Oracle's BPEL implementation also provides some of the same basic functionality through its Human Workflow component. BPEL itself also comprises some of the same functionality as traditional E-Business Suite Workflows and can extend the power of the E-Business Suite to integrate with other applications, an enterprise-wide user community, and directly to partner data. In this paper, we will show you how to integrate the E-Business suite with your partners while leveraging the functionality of seeded Workflow processes. We will also demonstrate the power of BPEL and enterprise-wide benefits gained by replacing Workflow with BPEL processes and Human Workflow. Finally, we will demonstrate how BPEL does all of this from an easy-to-use monitoring and management console.

Introduction

The following paper discusses concepts implemented at several clients. Each one of the scenarios discussed cover a real-world problem and how to solve it. The intent of this paper is to demonstrate the nuts and bolts behind each of the solutions. All too many presentations and papers discuss these topics, but leave significant holes in the details behind the technology and implementation of said technology. The goal of this paper is to provide a high level—yet complete—solution for each of the business challenges below.

Assumptions

This paper assumes a working understanding of the following list of Oracle products and technologies. They will only be covered at a higher level, as they pertain to the particular sections of the paper.

- Oracle Workflow Builder
- JDeveloper
- BPEL Process Manager
- Oracle E-Business Suite (EBS) technologies
- Knowledge of the usual EBS acronyms

This paper is a reference document for the OAUG Collaborate08 presentation. Some topics covered in this paper are covered in more depth during the demonstration portion of the presentation.

Finally, the processes discussed in this document are not all inclusive. Where specifics such as PO approvals and cancellations are concerned, there are other document statuses that are assumed to be addressed. The purpose of this document is not to cover every detail or a process, but relate the overall concept.

Scenario 1: PO Approval Extension

Many Oracle EBS customers have requested more information regarding POs, and high on that list are requests for additional notifications. Some common requests are that PO requestors are notified when a PO is approved or received or that vendors get notified when the POs are approved. These are outside the range of the seeded PO Approval Workflow.

In the past, Solbourne has been able to deliver extensions like this by modifying the seeded Workflow, creating PL/SQL components, and performing additional configurations to the server environment. However, modifying the delivered Workflow is problematic because of upgrades. Modifying or creating a custom Workflow require Workflow expertise, are limited to Oracle-based connections if other systems are involved, and will require changes as Oracle moves away from the Workflow product. This type of human interaction is precisely what BPEL's Human Workflow component does best. For those with a Fusion Middleware environment, Human Workflow is the best choice for this type of extension.

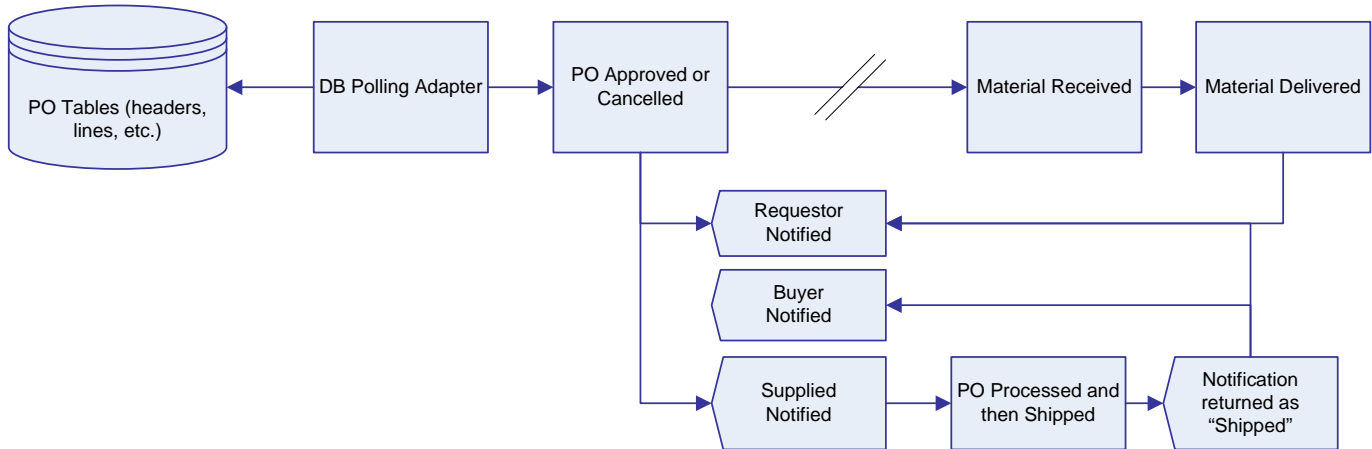
This example will detail how to interact with the following groups:

- Requestors on PO approval
- Vendors on PO approval
- Vendors will notify requestors and buyers at shipment
- Requestors notified on receipt

This will be accomplished using BPEL and Human Workflow after the seeded PO Approval process completes. This ensures that Oracle's long-standing, well-integrated approval process remains untouched and fully upgradeable. It also provides customers with a simple, fully customizable way to manage emails and subsequent responses.

Process and Design Steps

This process will live outside of the seeded PO Approval workflow process. The BPEL process will use a combination of polling database adapters and human workflow responses to interact with the various user groups. A high level overview of the process is defined in the next flow:



The BPEL process starts with a Database Polling adapter. This polling adapter will reference a view that uses the last approved or cancel date (and other dates where appropriate), whichever is greater. When it finds new data, a BPEL process is kicked off with the PO Header ID as input. From this point, the process queries the database to get more information about the PO, including data about the lines, distributions, and supplier. The process will then branch, performing different activities, based on the status of the PO.

Each of the following statuses has a corresponding branch in the BPEL process:

- Approved/Cancelled – notify the requestor and supplier based on the information from the PO. The vendor’s email address will be used for the notification¹.
- Shipped – vendor responds to notification with a shipment acknowledgement, buyer and requestor notified.
- Delivered – requestor notified of the receipt

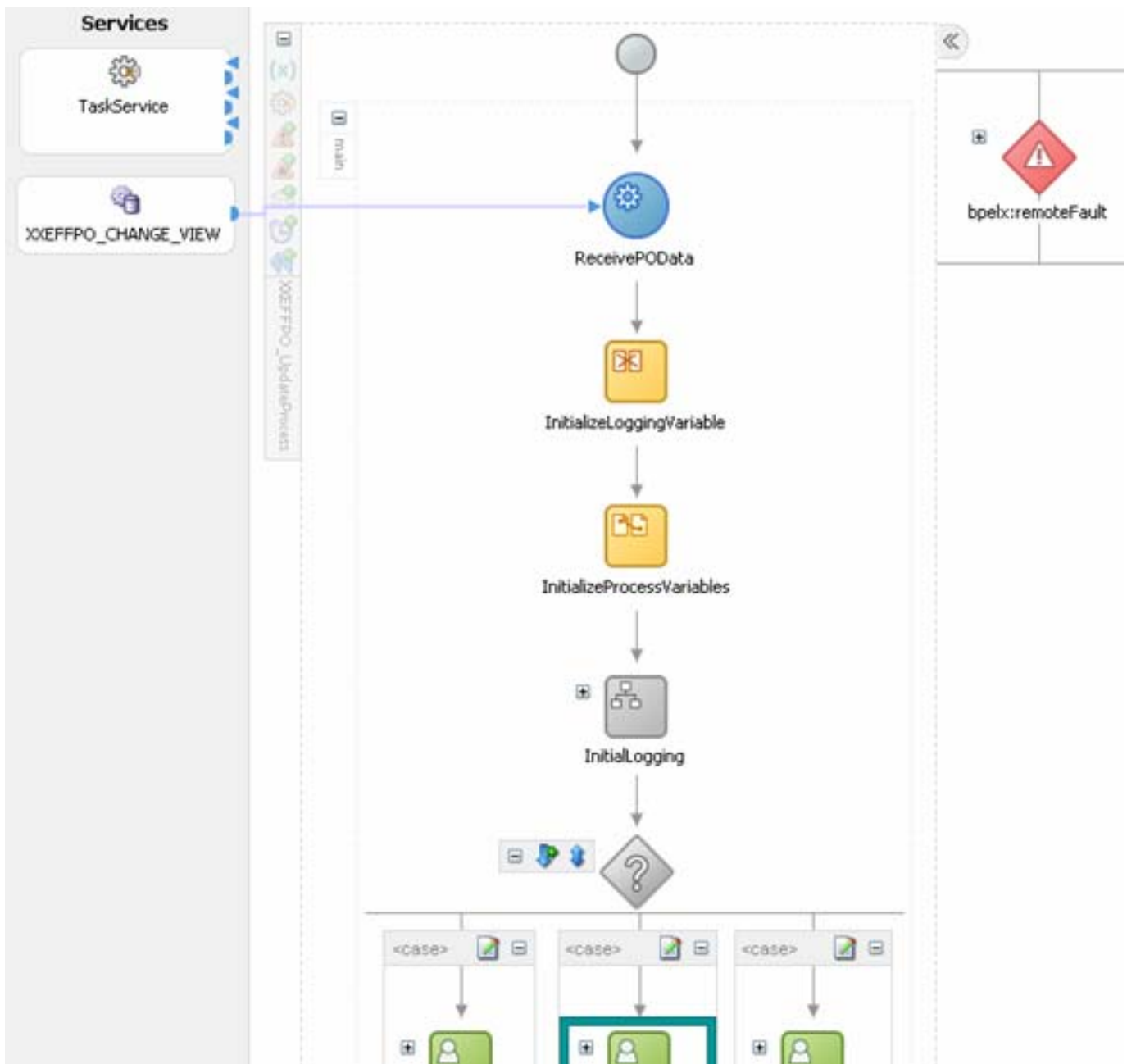
Following are several BPEL process screenshots, including Human Workflow, defining one possible way to implement this solution.

¹ Emails used with Human Workflow must exist either in OID or another directory service. The easiest method to handle this is with a custom directory service with vendor, requestor, and buyer contacts built as a view from the Oracle EBS.

BPEL process which starts with an update to the XXEFFPO_CHANGE_VIEW.

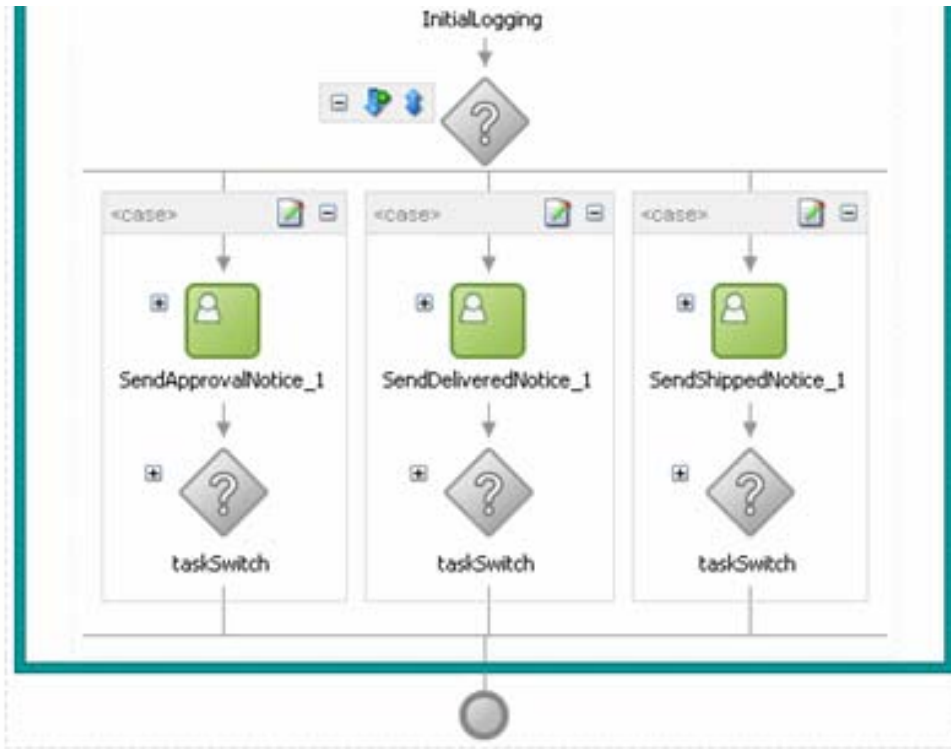
The view which is the source of data for this process is looking at all PO records that have had their Authorization_Status updated. The DB Adapter is set up to poll the view every 5 seconds to check for records. As records are found, the process is kicked off and the DB Adapter, in addition to passing data to this process, updates the table with SYSDATE.

The first part of this process initializes some basic information about this process and logs a line that this process has kicked off, capturing the history of this process, including PO acted upon. Error handling is included in this process and allows the process to be more robust and recover from expected and unexpected errors.



Human workflow elements

Notifications are sent based on status of PO:



Notification settings, based on the outcome of the Human Workflow process:

Expiration and Escalation Policy

Notification Settings

Task Status	Recipient	Notification Header
Expire	Assignees	
Complete	Initiator	
Error	Owner	

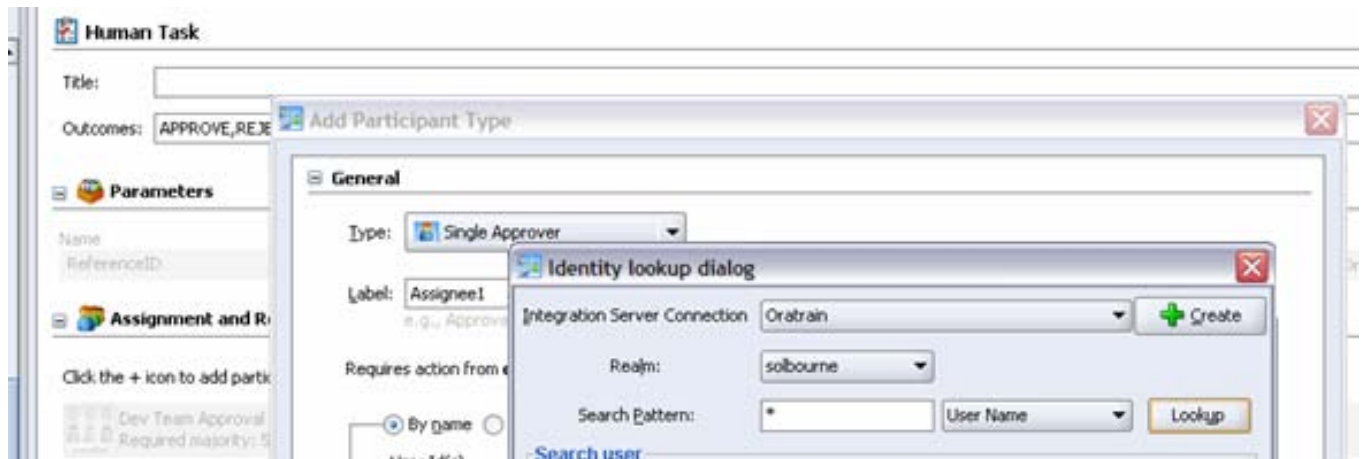
Remind once Day Hour Minutes Before Expiration

Make notifications secure (exclude details)
 Make email messages actionable
 Send task attachments with email notifications

Custom Identity plug-in

An important part of any Human Workflow process is where the user repository is found. Your AS 10g 10.1.3.3 server must be setup to point to a particular set of user information. This user information can be found in Oracle Internet Directory (OID), an LDAP server like Active Directory, or a custom schema. For more information on how to configure integration with a user repository, refer to Solbourne's *Rapid Interface Development for the E-Business Suite and PeopleSoft in a SOA Environment* white paper, available at <http://www.solbourne.com/downloads.html>.

In this case, the approval hierarchy is derived from users in the E-Business Suite. A Java-based custom Identity application has been written and deployed to the AS 10g 10.1.3.3 server that eliminates the need for a more complex OID setup or connection to Active Directory. The 'Realm' below associates this task with the Solbourne users associated with this E-Business Suite instance.



Instead of using users in OID or Active Directory, a set of custom views is placed in the E-Business Suite and called from the custom Java program. These views are written especially for the E-Business Suite but could be customized for any database schema.

```
CREATE OR REPLACE VIEW wfis_user_view
AS SELECT
    b.person_id AS userid
    , g.realm AS realm
    , h.user_name AS name
    , b.first_name AS firstName
    , b.middle_names AS middleName
    , b.last_name AS lastName
    , b.last_name||', '||first_name AS displayName
    , ' --- ' AS description
    , f.name AS title
    , nvl((SELECT
        SUP.full_name
        FROM
            PER_PEOPLE_X          PEO
            , PER_PEOPLE_X        SUP
            , PER_ASSIGNMENTS_F    ASS
        WHERE
            PEO.person_id = ASS.person_id
            AND ASS.supervisor_id = SUP.person_id
            AND PEO.employee_number = b.employee_number
            AND PEO.business_group_id = SUP.business_group_id
            AND PEO.business_group_id = b.business_group_id
            AND rownum <= 1), '-- No Manager --')
        AS manager
    , b.email_address AS mail
    , (select pp.phone_number from HR_LOOKUPS hrl, PER_PHONES pp
        where hrl.lookup_code = m.phone_type and hrl.meaning = 'Work')
```

Scenario 2: Migration from XML Gateway to BPEL

Many EBS customers use XML Gateway to integrate with external systems. A common scenario that is used here is a requirement that defines Oracle EBS as the system of record for inventory items. However, an external work-order system can dynamically create items. These items created in the legacy system must be created in Oracle EBS for inventory purposes. This process initiates in an outside system, then, using the item open interface, creates the item in Oracle Inventory.

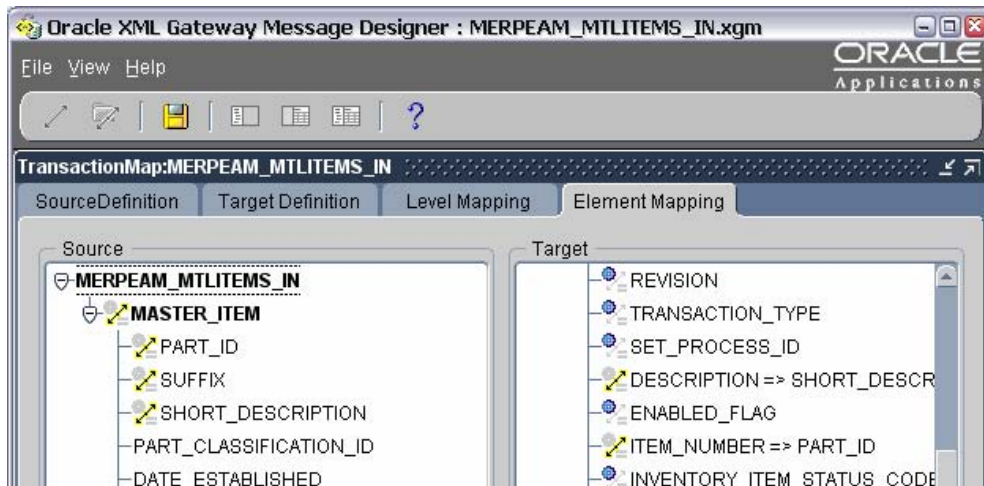
XML Gateway (XMLG) uses workflow as its base for operations. XMLG requires a very specific skill set to manage and maintain. Monitoring and managing an XMLG-based integration requires the administrator to monitor many disparate systems and technologies, and be an expert in all of them. Since workflow is a very good fit for BPEL, it is a logical progression to migrate older web services to BPEL. This scenario covers moving from an older technology (XMLG) to BPEL.

The following process design, flow, and diagrams explain the original components, and what the migration to BPEL entailed.

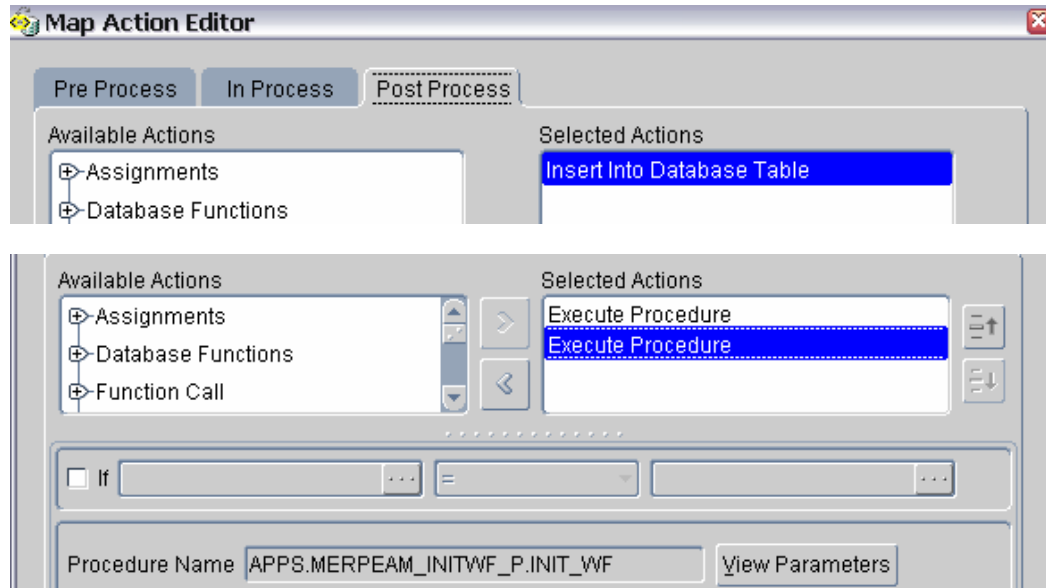
Process and Design Steps

We will start by defining the old XMLG solution. The client's work-order system was connecting to AQ and enqueueing records. A scheduled concurrent process fired off a workflow that looked for data in the queue and invoked an XMLG Message Designer map. The map would then insert the data into a staging table and call the concurrent process to perform the insert:

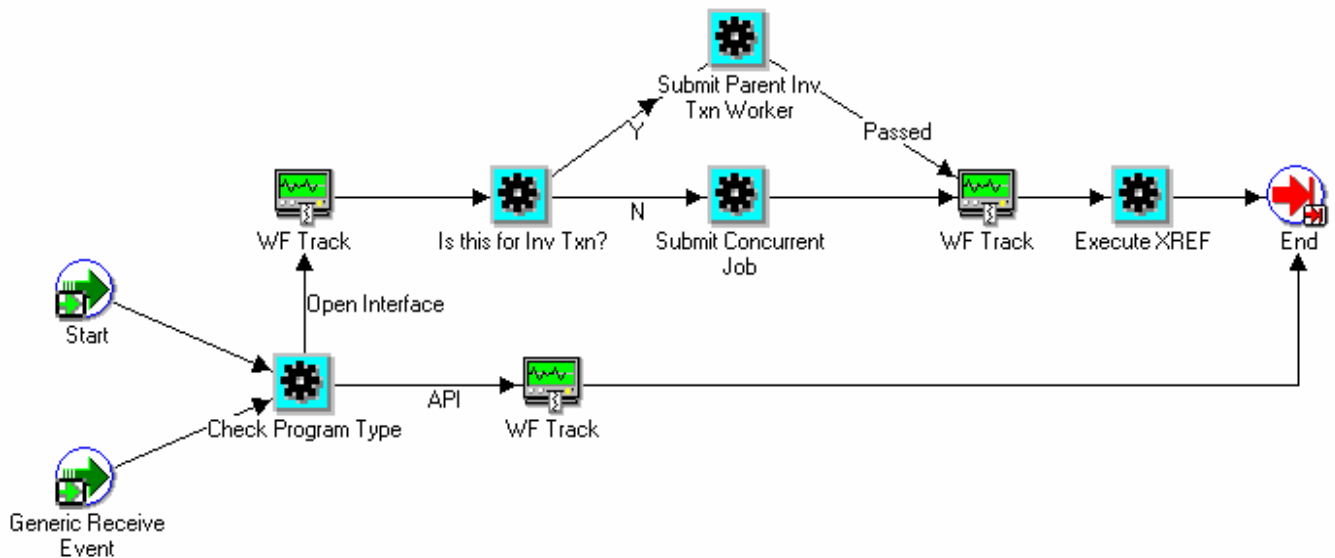
1. XML Gateway listener sees new record enqueued to the XMLG queue. Based on the message's transaction type, an XMLG map is invoked. In this case, it is the MTLITEMS IN map:



- The map's action is to actually insert into the Open Interface table, and after that, invoke the Concurrent Process management workflow:



- Following is the concurrent process workflow that manages the actual call to the open interface program:



The problem with this process is an inadequate notification method to keep administrators up to date on what has failed and why. To monitor this particular process, the administrator would have to keep tabs on the Workflow Transaction Monitor, concurrent request, and open interface failures:

```
SELECT *
FROM mtl_interface_errors
WHERE table_name = 'MTL_SYSTEM_ITEMS_INTERFACE';
```

Refresh Data
Find Requests

Request ID	Name	Parent	Phase
3436381	Import Items		Completed

Results

Script Output Explain Autotrace DBMS Output OWA Output

sults:

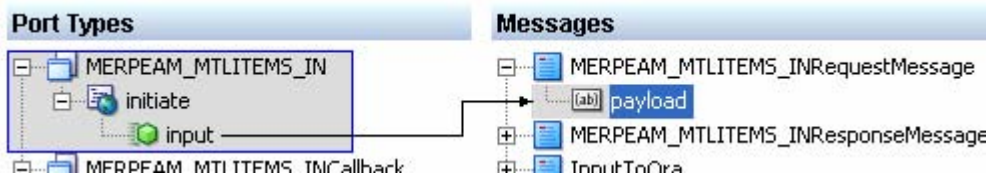
.TE_DATE	ERROR_MESSAGE
3	Item 1234 already exists in the organization V1. Please use a different item name/number.

Inbound Search Results

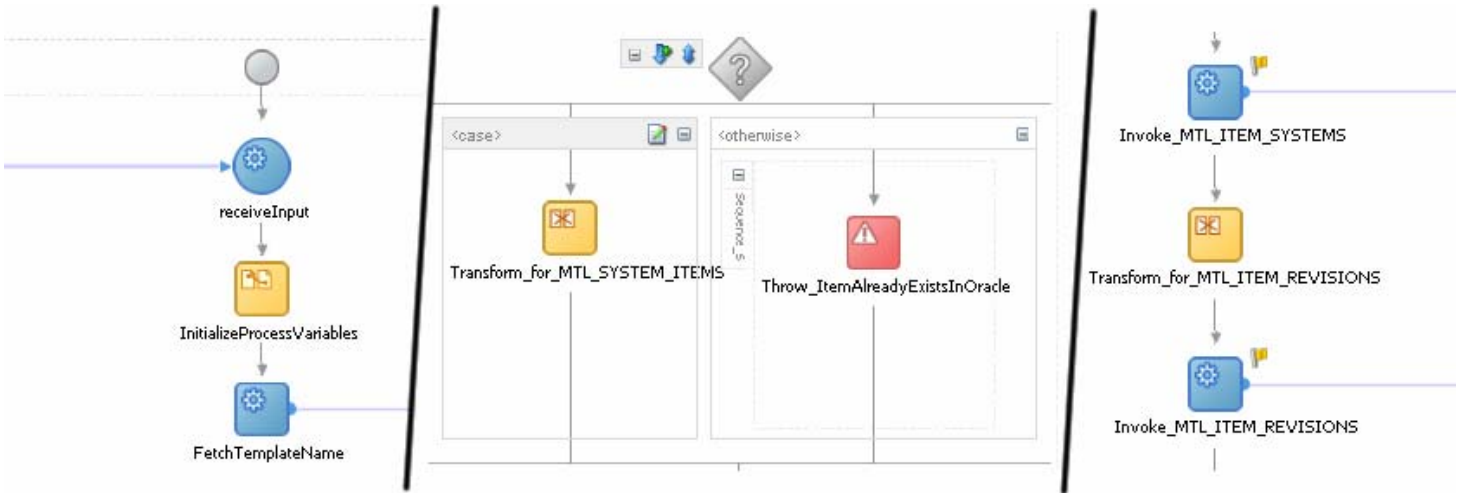
Document ID	Trading Partner Name	External Transaction Type	External Transaction Subtype	Processing Status
99900812341024	N/A	MTLITEMS_IN	MTLITEMS_IN	Success
99900812341025	N/A	MTLITEMS_IN	MTLITEMS_IN	Success

Using BPEL, the concurrent monitoring can be made much more streamlined. And with the use of logging and error handling, the entire process can be made much easier to manage. Following are screen shots (and verbiage) explaining various aspects of the BPEL process used to replace the XMLG/workflow process.

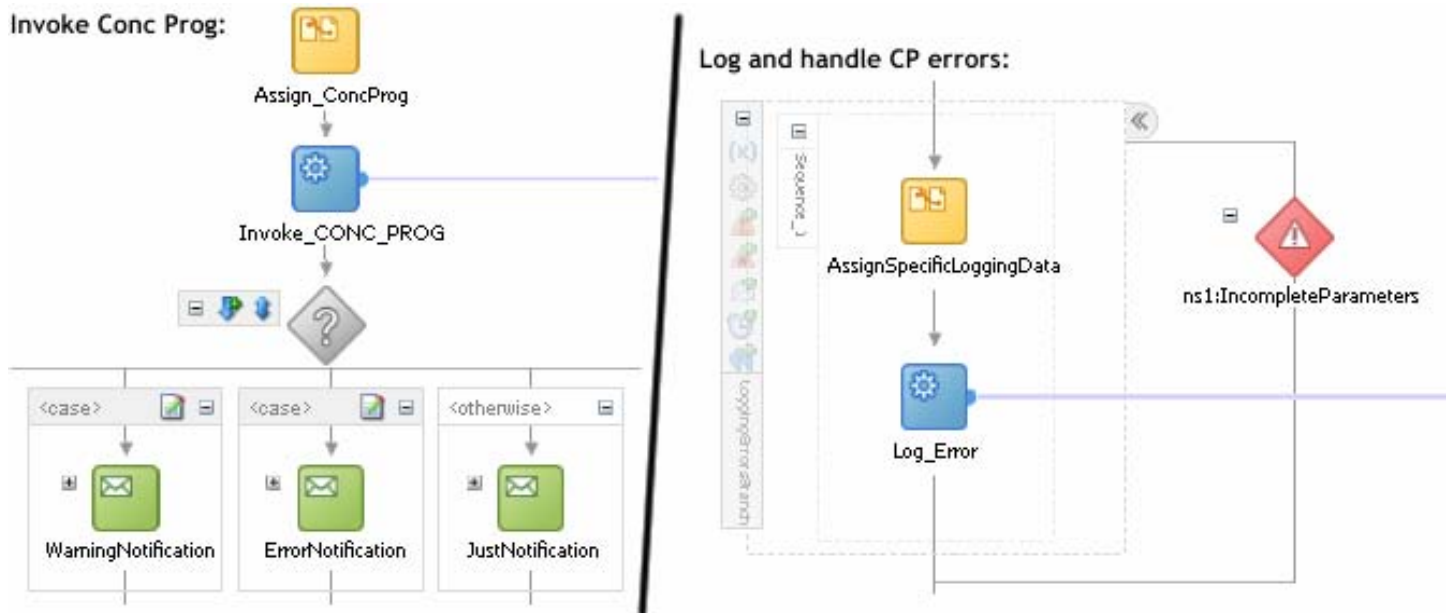
1. The first step was to migrate to a true Service Oriented Architecture SOA, allowing for greater administrative and maintenance flexibility. Although XML Gateway created a web service that could be accessed by external systems, Oracle's SOA Suite allows for this BPEL process to be kicked off via SOAP, WSIF, Oracle AQ, IBM MQ Series, JMS, or a custom web service.




- From the initiation, the payload is received, the item template is fetched, some basic data validation is done, and transformations are started on the data to prepare it for entry in the E-Business Suite Open Interface Tables:



- After all transformation and data validation is completed, the concurrent program that actually executes the open interface is invoked. All errors and results are handled and the administrators are notified accordingly:



The Enterprise Foundation Framework (EFF) dashboard makes this process much more refined. Through the use of process logging, data is stored in database tables, then accessed via the Oracle Applications Framework “dashboard” page. From this page, error monitoring from beginning to end, resubmission, and detailed process statistics are available. Following is a screen shot of the EFF dashboard:



[Diagnostics](#) [Home](#) [Logout](#) [Preferences](#)

[Search for Interface Process Log >](#)



Process Run ID: 140013

Resubmit Process as:

Process Log Details

Master Interface Instance ID	Interface Name	Target System	Oracle E-Business Suite
BPEL Instance Id 140013	Interface Description	Archive File Name	MERPEAM_MTLITEMS_IN_140013_2008-02-29.xml
Process Run Id 140013	Interface Status	Archive File Path	
Request Id			
Creation Date	Notification List		
29-Feb-2008 14:13:35	Source System		
	Maximus FA Suite		

Log Details

Parent Request Id	Sub Request Id	Step Number	Step Name	Step Description	Step Status	Status Text	Request Log	Request Output	Creation Date	Notify User
		1	Initializing	Incoming Item Data	Initializing	New data received from FA Suite			29-Feb-2008 14:13:35	
		2	Data Validation	Data validation	Normal	FA Suite passed validation			29-Feb-2008 14:13:42	
	3436716	3	Item Import	Item Import	Warning	Importing items from FA Suite into E-Business Suite Inventory			29-Feb-2008 14:14:14	
		4	Interface Completion	Interface MERPEAM_MTLITEMS_IN has completed	Normal	Interface Completed: 02/29/2008 at 14:56:37			29-Feb-2008 14:14:15	

Note

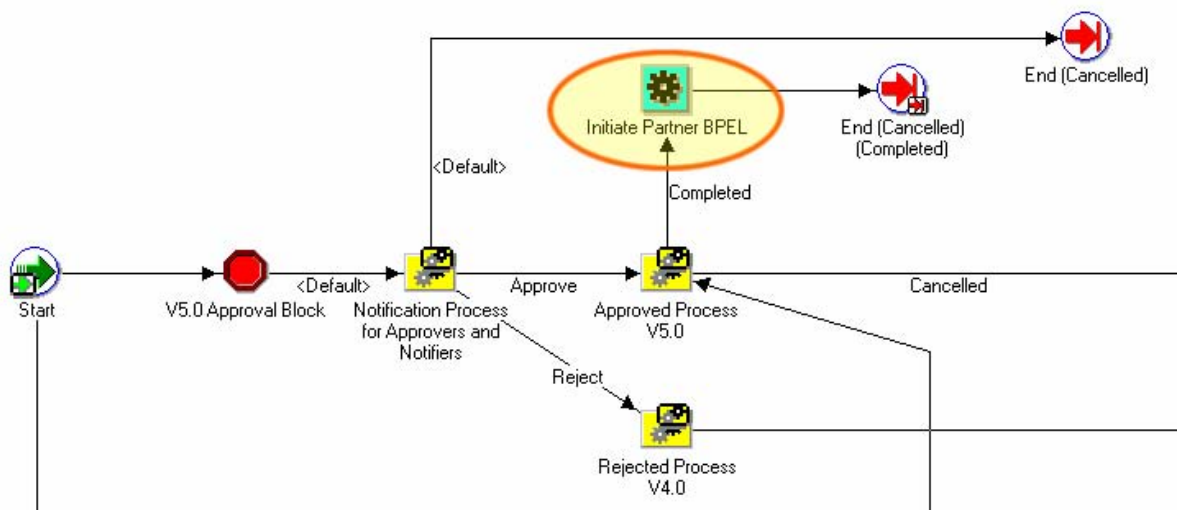
Scenario 3: Data-push to Partner Web Services

There are many times when employee information needs to be disseminated to a variety of partners. This need ranges from health insurance providers to unions and 3rd party payroll administrators. In this example, we will be doing real-time integration with a 3rd party payroll administrator.

In this example, the client has Self-Service HR installed. Personal information changes require workflow approval.

Implementation Detail Steps

1. This process will start with an employee changing their personal information using Self-Service HR. After the final approval completes in the seeded workflow, a BPEL process is initiated within the Workflow. Note that the “Initiate Partner BPEL” node is new. This invokes custom PL/SQL used to start the BPEL process:

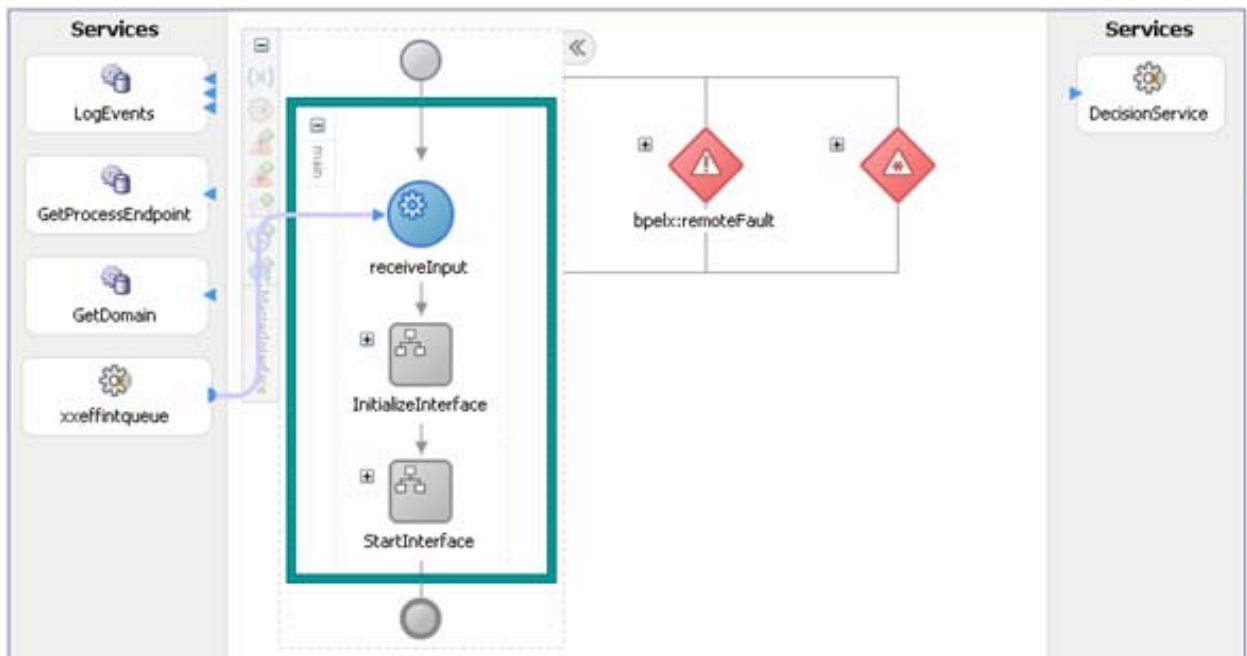


2. Custom PL/SQL compares the old data to the new data (using date-tracking functionality). If a name change has occurred, the data is loaded into a PL/SQL Object and this Object is passed to a BPEL process using Oracle’s AQ:

```
657 for cr_read_data in c_new_person_data
658 loop
659   l_payload := xxeffhr_person_update_o(
660     person_header => l_header,      --update with each read procedure
661     marital_status => cr_read_data.marital_status,
662     emp_no        => cr_read_data.emp_no,
663     effective_date => cr_read_data.effective_date,
664     term_date     => cr_read_data.term_date,
665     job_id        => cr_read_data.job_id,
666     manager_id    => cr_read_data.manager_id,
667     assignment_id => cr_read_data.assignment_id
668   );
669   dbms_aq.enqueue(
670     queue_name      => l_queue_name, -- IN
671     enqueue_options => enqopt,      -- IN
672     message_properties => msgprop,  -- IN
673     payload         => l_payload,    -- IN
674     msgid          => enq_msgid); -- OUT*/
675 end loop;
676
677 update xxeffint_dbpolling
678 set last_read_date = sysdate
679 where table_name = l_table_name;
```

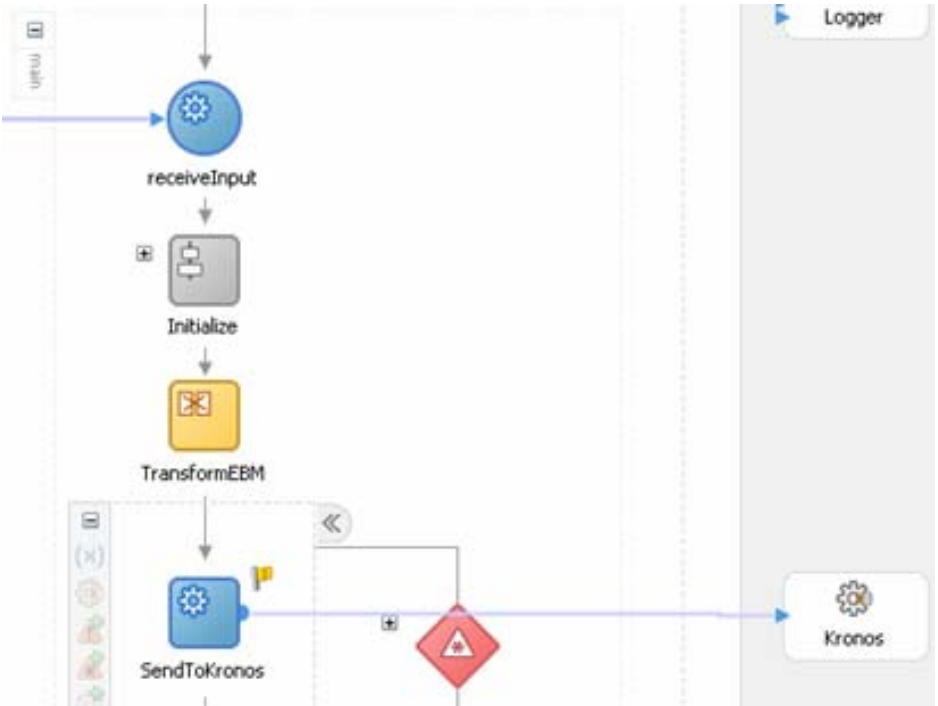
3. A single BPEL process listens on that queue for any HR person related activity – once a message lands on the queue, this process receives the data and, based on the payload, determines it's a person update for the third party payroll system. This type of 'decision service' fills two roles. First, it makes dequeuing messages from this queue a singular operation. That's all this process does, abstracting the connection process from the primary business process. Second, it streamlines the development process, and overhead on the DBMS, by allowing one queue to process multiple messages.

After logging a single line to note the beginning of the payroll system update, the process kicks off the main business process and finishes. This type of transaction is known as a 'fire and forget' type of transaction, in which a reply is not expected.



4. The main BPEL process receives the set of data, or message, from the decision service. This message is a prescribed set of data for person updates, that is not *necessarily* exactly what the partner application is looking for, but is the full range of information one would need to understand any updates to a person in the E-Business Suite. This type of message is known as an Enterprise Business Message, a subset of the larger person data set (Enterprise Business Object).

This main BPEL process, known as an Enterprise Business Service, is responsible for now transforming the data to match the needs of the partner application, including deriving any missing values and validating or modifying the data to match the correct format expected by the partner. After transforming the data, it will then be distributed to the partner site.



When the partner confirms the receipt of the data (through synchronous or asynchronous reply), a notification is sent to the employee and the HR rep to confirm the change with the partner application and company. In this way, the business process of updating the employee data is closed.



Conclusion

As demonstrated by the examples above, the strengths of both BPEL and Workflow can be used to extend the return on an E-Business Suite investment. Through the use of BPEL, a standards-based, robust, user-friendly tool can be used to manage interfaces with other systems and interactions with users.

BPEL is flexible enough to be incorporated in and around existing business processes. As demonstrated, Oracle's well-integrated components can be extended to fulfill very specific, business driven needs, or broader, more-industry focused requirements.



Using BPEL and Workflow to Build Integration Components for the E-Business Suite
February 2008

Authors: Ryan Stefani, Solbourne
Chad Coffman, Solbourne

Solbourne
4001 Discovery Drive, Suite 210
Boulder, CO 80303
U.S.A.

Inquiries:
Phone: +1-303-417-2800
Fax: +1-303-417-2820
www.solbourne.com

Copyright © 2008, Solbourne. All rights reserved.
This document is provided for informational purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. The Solbourne name, logo, related marks and all images on this document are owned by Solbourne and are registered and/or used in the U.S. All trademarks, service marks and trade names not owned by Solbourne referenced in this document are the property of their respective owners.