Migrating to BPEL from Oracle Work Flow

An Oracle White Paper
March 2008
INTRODUCTION

As most readers of this document will be aware, Oracle Workflow is a tool for designing and implementing processes and workflows that runs as PL/SQL in the Oracle database. Oracle has advised its customers using Oracle Workflow to consider migrating to Oracle BPEL Process Manager (BPEL PM), the next generation of process technology from Oracle.

BPEL is a standard language for orchestrating services; in June 2004, Oracle acquired Collaxa, an ISV with the leading BPEL implementation, thereby creating the Oracle BPEL Process Manager (BPEL PM). BPEL PM provides design-time and run-time tools, built around the BPEL standard, that support both system-centric and human-centric workflows (as well as any combinations of the two).

Because Oracle Workflow and BPEL are both fundamentally workflow languages, anything that can be implemented in one can be implemented in the other. However, they have very different models and this paper is intended to provide some guidance for those customers who are familiar with Oracle Workflow and will be designing and implementing BPEL processes, either brand new ones, or ones which will replace existing Oracle Workflow processes.

This paper is primarily focused on customers with the task of re-implementing existing standalone Oracle Workflows in BPEL, so it will spend the most time examining the areas where Oracle Workflow and BPEL models diverge. For example, the control flow and exception handling models are two areas where Oracle Workflow and BPEL are fundamentally different. This paper will focus on these and other areas which developers should be most aware of and describe tips and best practices for migrating Oracle Workflows to BPEL. It will also describe features in BPEL PM which can ease this migration process.
STATEMENT OF DIRECTION

Oracle has announced product obsolescence for Oracle Workflow. In a Statement of Direction published in January 2006, Oracle indicated that new development by customers of human-centric workflow should be done with Oracle BPEL Process Manager. In a new SOD published in January 2007, Oracle is now recommending that customers who have Oracle Workflow workflows in development or production that they plan to continue to enhance in the future should begin re-implementing those in BPEL. The SOD is available at:

BPEL OVERVIEW

Business Process Execution Language, commonly known as BPEL or WS-BPEL, has emerged as the de-facto standard language for process orchestration. BPEL has very rich and comprehensive orchestration semantics, is based on wide industry participation, and is broadly accepted.

BPEL allows composition of Web Services into end-to-end processes, which them selves are available as services. BPEL has built in support for asynchronous transactions, flow control, and compensating transactions. BPEL leverages XPath, XSLT, and XQuery for data manipulation.

The primary components of a BPEL process are:

- Variables – process variables hold data that needs to be passed to process activities or is computed from process activities
- Activities – process activities are of two kind:
  - Control Flow – these activities specify the control flow of a process. Examples include sequence, flow, forEach
  - Processing – these activities specify some processing. Examples include invoke, receive, reply, assign
- Fault and Event Handlers – specify how to handle faults and asynchronous events
- Partner Links – relationship with a partner that either provides services to the process or consumes services provided by the process

Following is a quick description of the important BPEL activities:

- receive - Receive a message. A process typically starts with a receive; a receive may also be used to wait for a message to arrive.
- reply - Reply to a message received
- invoke - Invoke a service
- assign - Update values of variables with new data

“BPEL is the future of the integration space in my view... Why? Because the value is so much higher when you provide not only a way to integrate applications, but also a way to create services from them and put them into business processes.”

— John Rymer, Vice President Forrester Research
• throw - Generate a fault from a business process
• wait - wait for a given time period or until a certain point in time has been reached
• empty - noop
• sequence - defines a collection of activities to be performed sequentially
• while - loop as long as the specified condition is true
• flow - specifies one or more activities to be executed concurrently
• scope - defines a nested scope; variables, partner links, and fault handlers are all scoped

To learn more about BPEL, please read the specification at http://docs.oasis-open.org/wsbpel/2.0/wsbpel-specification-draft.html.


Figure 1: Example BPEL Process
THE ORACLE BPEL PM

The Oracle BPEL PM is a highly scalable and reliable implementation of the BPEL standard. It executes on a J2EE application server and leverages a database to maintain state in a scalable and fault tolerant fashion. Oracle BPEL PM features a rich binding framework enabling connectivity among many other sources to databases both using SQL and PL/SQL. The Oracle BPEL PM includes a rich graphical drag and drop process editor.

Figure 2: Oracle BPEL PM Architecture


The http://otn.oracle.com/bpel site is a great resource and among other things includes a BPEL cookbook.

“Oracle BPEL allows CapGemini to define a business process that exactly matches what the business needs at any point in time

-- Rick Hymer. VP CapGemini
**Human Workflow in BPEL PM**

Leveraging BPEL’s rich support for asynchronous services, Oracle BPEL PM provides a human workflow service engine that BPEL processes can then call out to. With this architecture, human tasks and manual steps can be incorporated in 100% standard BPEL process flows, just like any other asynchronous service.

![Diagram of Human Task activity in BPEL designer]

**Figure 3: Human Workflow Overview**

In summary, human tasks in BPEL PM are modeled using a Human Task activity, the routing, assignment, notifications, escalation and other metadata is stored in a .task file, and at run time users can log into a Work List application to find and perform work.

Some of the salient features of the Human Workflow component of BPEL PM are:

- **Declarative pattern based routing** – Common patterns such as Management Chain Escalation and Group Voting can be declaratively specified. Multiple patterns can be combined for sophisticated routing patterns.
- **Assignment** – Tasks may be assigned to individuals or groups (roles). Tasks may also be assigned dynamically based on process data enabling plugging in of any other assignment engine or rules based assignment.
- **Nomination, Delegation and Reassignment** – Supervisors and process owners may nominate or delegate tasks. Also, group owners may specify reassignment rules for redistributing work assigned to a group based on various load balancing algorithms. The reassignment feature addresses the scenarios addressed by the Role Resolution feature in Oracle Workflow.
• Declarative Escalations and Notifications – Escalations, notifications, and reminders may be declaratively specified. Rich set of notifications is supported and email notifications may be made actionable allowing some one to complete their task from their email without logging into the Work List application.

• Rich Forms – Rich JSP forms are automatically generated based on the payload specification; these can then be edited in JDeveloper or any other JSP editor.

• Work List Application – Work List application is the application where users find and perform work. The Work List application supports the notion of Supervisors, Group Owners, and Process Owners enabling appropriate capabilities for those roles.

• Comments and Attachments – Participants can add comments or upload attachments.

• Audit Trail – Audit Trail and status of tasks is available within the Work List application.

MIGRATION OUTLINE

Process Migration

Step 1. Classify Workflow Process: The first step in the migration should be to understand your Oracle Workflow Processes and classify them with the goal of identifying the right target technology. Oracle Workflow Processes may fall in the following category:

- Process Navigator Flows: These should be migrated to ADF Task Flows. In Oracle AS11, a new feature - Activity Guides – is planned to target these use cases.
- Page Flows: These should be migrated to ADF Flows
- Simple Deferred Activities: If you are using Workflow to simply perform some DML activities on your tables, then you may want to not convert these to BPEL but rather raise an event, and use ESB to invoke the needed activities on event receipt. If you choose, you can use BPEL as well.
- XML Transaction Flows: If you are using Oracle Workflow to model XML Transactions, you may want to simply use the transformation capabilities of ESB. If you choose, you can use BPEL as well.
- Business Processes: Generally speaking these are the orchestration of system services and human tasks including approvals. This category should be migrated to BPEL as described in this document.

Step 2. Identify Business Services:

Figure 4: Example of a simple deferred activity
• Identify all the PL/SQL procedures and functions you are calling from Workflow. You should be able to call these directly from BPEL using either the DB adapter or Applications adapter. However, if you are accessing or setting item type attributes from within your PL/SQL code you will need to change them as described in section Data Access below.

• Identify all application APIs you are calling from Workflow. You should be able to call them from BPEL using the applications adapter.

Step 3. Identify Workflow Dependencies:

• If you are using any Workflow Tables in your application code, it will need to be modified to use corresponding BPEL APIs.

  Check to see if you are accessing any of the following:
  i. WF_ITEM_ACTIVITY_STATUSES
  ii. WF_ITEM_ACTIVITY_STATUSES_H
  iii. WF_NOTIFICATION_ATTRIBUTES
  iv. WF_NOTIFICATIONS
  v. WF_ITEM_ATTRIBUTE_VALUES
  vi. WF_ITEMS
  vii. WF_ACTIVITY_ATTR_VALUES
  viii. WF_ACTIVITY_TRANSITIONS
  ix. WF_PROCESS_ACTIVITIES
  x. WF_ACTIVITY_ATTRIBUTES_TL
  xi. WF_ACTIVITIES
  xii. WF_LOCAL_ROLES

• If any of your user interfaces are accessing Workflow Monitor UI, the references will need to change to BPEL console.


Step 5. Identify Approval Logic: BPEL Human Workflow enables powerful declarative pattern based approval. You may simplify the approval logic as described in
Approval Flow (Human Workflow) below.

Step 6.  Define Data Shapes (XSDs): Understand the shape of the data variables that you will need and define the XML Schema Definitions (XSDs)

Step 7.  Define BPEL Process Flow: Understand the current flow and define it in BPEL. Section Control Flow below discusses the semantic differences between OWF flow and BPEL flow and how to map OWF Flow to BPEL flow.

Step 8.  Fill out BPEL Details: Add BPEL details per the mapping outlined in section
Constructs Mapping below.

Step 9. Test: Use BPEL Test Framework to test your BPEL processes

Data Migration

Due to the differences in the OWF process and migrated BPEL process, it is not recommended to attempt migrating run time data from Workflow Tables to BPEL Runtime Tables. The value of the run time historical data is primarily for analytic purposes; therefore, instead plan on migrating runtime data to the appropriate analytic warehouse schema.
CONCEPTS MAPPING

Control Flow

Graph vs. Block Structured
The Control flow in Oracle Workflow is essentially a Graph with cycles allowed. Any node in the control flow may have multiple incoming and outgoing transitions.

BPEL has block structured control flow, where most nodes have single incoming and outgoing transitions. Only specialized control flow nodes such as Flow (parallel forking), Switch (conditional branching), and Pick (Or join for events) may have multiple incoming and outgoing transitions.

Note: It is tempting to assume that BPEL Links may be used to design Graph control flows. However, BPEL links are a dependency mechanism and are not recommended for designing control flow.

Loops
In Oracle Workflow, loops are implicitly designed as cycles in the graph. A Loop Counter activity may be used to control the allowed number of iterations.

In BPEL, loops are explicitly modeled using looping constructs such as While and forEach. The specification of iterations is configured as attributes of these looping activities.

Transitions / Branching
In Oracle Workflow, multiple transitions may be drawn from a node based on its result lookup as well as Any, Default, and Timeout. Parallel branching is implicitly defined if more than one transition is enabled (Any in parallel with Result based transition).

As discussed above, branching is explicitly designed in BPEL. Also conditional branching vs. parallel branching is explicitly designed using the appropriate constructs (Switch vs. Flow and Pick).

Mapping OWF Control Flow to BPEL Control Flow

1. Understand the logic of your OWF control flow.
2. Identify the loops and looping conditions. Model these in BPEL using While.
3. Identify the fan-out points. Understand whether the fan-out is conditional branching or parallel branching. Identify the matching fan-in points. Model in BPEL using Switch or Flow.
4. Identify any fan-in point not matched in the previous step. These must be because of re-use of a single activity. Duplicate it in BPEL.
Data Access

In Oracle Workflow, invoked PL/SQL code may access the Item attributes using WF_ENGINE APIs. Also, each activity returns a result suitable for result based transitioning.

```plaintext
procedure UpdateStatus ( itemtype in varchar2,
  itemkey in varchar2,
  actid in number,
  funcmode in varchar2,
  resultout out varchar2) is
  l_po_number varchar2(20);
  l_po_status varchar2(20);
begin
  if ( funcmode = 'RUN' ) then
    l_po_number :=
    wf_engine.getitemattrtext(itemtype, itemkey, 'PO_NUMBER')
    l_po_status :=
    wf_engine.getactivityattrtext(itemtype, itemkey, actid,
      'PO_STATUS')
    PO_PKG.UpdateStatus(l_po_number, l_po_status);
    resultout := 'COMPLETE:<result>';
    return;
end if;
end;
```

Figure 5: Data access in OWF within PL/SQL using WF_ENGINE API

In BPEL, the invoked activity can not access process data. Any data it needs should be passed in as its input parameter and any data it needs to set should be returned as its output parameter. Note that the output of a BPEL activity is not limited to lookups.
Approval Flow (Human Workflow)

In Oracle Workflow, Approvals were modeled using complex loops, wait activities, notification activities, timeouts, etc.

Figure 6: Approval Flow in Oracle Workflow

Human Workflow component of the BPEL PM enables most such flows to be modeled as a single user task with declarative metadata as shown in Figure 3.

Figure 7: Task metadata captures routing, escalations, notifications etc.
Error Handling

Error handling in Oracle Workflow is specified in an Error process. OWF does not allow errors to be returned to callers.

Standard BPEL Error Handling – Fault Handlers and Compensation

Error handling in BPEL is more sophisticated providing a wider array of choices. Fault Handlers in BPEL allow errors to be caught within the process and appropriate corrective actions to be applied. BPEL also supports the notion of compensation where activities done prior to the catching of fault may be undone as part of the fault handling. Processes may recover after applying the fault handling and continue or propagate the fault to the callers.

BPEL PM Error Hospital – Policy Based Error Handling

While the standard BPEL error handling is powerful, it requires explicit modeling of fault handling and does not allow for manual recovery. Oracle BPEL PM will add policy based error handling in a 10.1.3.1 patch to complement the BPEL capabilities. This feature will enable specification of error handling policies including automatic retries and suspend on error. Suspended processes may be corrected and restarted either manually or by some error handling processes. Thus this feature will address the requirements addressed by OWF error handling.

Monitoring and Administration

Oracle Workflow includes an Administrator Monitor component that lets users view and administer run time workflows. The corresponding functionality is provided by the BPEL console. Administration of human workflow tasks including suspending and error fixing is done within the Worklist application, which exposes administration screens when a user with Administrator role logs in. User’s calendars are also managed within the Worklist application.

BPEL PM also provides a rich set of Administration APIs that are documented at http://download-east.oracle.com/docs/cd/B31017_01/integrate.1013/b28986/toc.htm.
### CONSTRUCTS MAPPING

#### Overview

The following table provides an overview of how Oracle Workflow constructs map to BPEL PM:

<table>
<thead>
<tr>
<th>Oracle Workflow</th>
<th>BPEL PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Invoke</td>
</tr>
<tr>
<td>Sub-process</td>
<td>Process</td>
</tr>
<tr>
<td>Receive Event</td>
<td>Receive</td>
</tr>
<tr>
<td>Send Event / Raise Event</td>
<td>Sensors</td>
</tr>
<tr>
<td>And</td>
<td>Flow</td>
</tr>
<tr>
<td>Or</td>
<td>Pick (not a direct map)</td>
</tr>
<tr>
<td>Block</td>
<td>Receive</td>
</tr>
<tr>
<td>Comparison Activity</td>
<td>Switch</td>
</tr>
<tr>
<td>Wait</td>
<td>Wait</td>
</tr>
<tr>
<td>Defer</td>
<td>Invoke (Asynchronous) + Receive</td>
</tr>
<tr>
<td>Launch Process</td>
<td>Invoke</td>
</tr>
<tr>
<td>NOOP</td>
<td>Empty</td>
</tr>
<tr>
<td>Loop Counter</td>
<td>- Loops modeled explicitly -</td>
</tr>
<tr>
<td>Role Resolution</td>
<td>- Run time Worklist rules -</td>
</tr>
<tr>
<td>Notification</td>
<td>Human Task</td>
</tr>
<tr>
<td>Vote Yes/No</td>
<td>Human Task</td>
</tr>
<tr>
<td>Master/detail – wait for flow and continue</td>
<td>Receive and Invoke</td>
</tr>
<tr>
<td>Get Monitor URL</td>
<td>BPEL Admin API</td>
</tr>
<tr>
<td>Activity result based transition</td>
<td>Switch</td>
</tr>
<tr>
<td>Timed Transition</td>
<td>Pick</td>
</tr>
<tr>
<td>Lookups</td>
<td>- Not needed -</td>
</tr>
</tbody>
</table>
Function Activity - PL/SQL

Workflow function activities calling PL/SQL code may be mapped to BPEL:

   Step 1. Drag and drop either the DB adapter or the Applications adapter to the Partner Links section on the BPEL canvas
   Step 2. Follow the wizard to configure the adapter
   Step 3. Use an Invoke activity to invoke the service created in the above steps

However, if the PL/SQL code uses WF_ENGINE APIs it will need to be changed as discussed in section Data Access above.

Sub-processes

There is no difference between a BPEL process and sub-process. Any BPEL process may be invoked from another BPEL process as a sub-process. Therefore, your Oracle Workflow sub processes will map to BPEL processes. You may follow naming conventions to identify certain BPEL processes as sub-processes.

Note: There is an overhead involved in invoking a process; therefore, you will need to balance maintainability/reuse with performance.

Events

Receive Event Activity

The Receive event activity in Oracle Workflow maps to Receive activity in BPEL.

Send Event / Raise Event

Sensors are the BPEL technology to raise events. You can also raise Apps Business Events using the Apps Adapter.

Standard Activities

And Activity

The OWF And activity should be modeled with BPEL Flow and FlowN constructs. Note that the Flow and FlowN constructs specify the forking as well as synchronization points, unlike the And activity which simply specifies the synchronization point.

The Flow and FlowN constructs execute multiple branches concurrently and synchronize on completion.

Or Activity

Modeling of Or joins in BPEL is not straightforward other than the case where the merging branches are all message or event handlers.
The Pick activity in BPEL waits for multiple events – messages and alarms – and proceeds on receiving the first; essentially an Or join limited to events.

**Block Activity**

The Receive activity in BPEL maps to Block activity in Oracle Workflow. To signal completion of the activity (CompleteActivity), invoke the operation corresponding to the Receive activity.

**Comparison Activity**

You can perform comparisons using XPath expressions in the condition expression of the conditional transition constructs such as Switch. (Also see Activity Result Based Transitions below)

**Wait Activity**

The OWF Wait activity maps to the Wait activity in BPEL, which waits for a certain period of time or until a certain point in time is reached.

**Defer Thread Activity / Defer**

Oracle Workflow provides a Defer Thread activity as well as activity cost based deferring of activities to perform heavy processing in the background. BPEL has first class support for long running processes and does not require explicit deferring of threads. To achieve the functionality of Deferring, invoke the service as an asynchronous service paired with a Receive to retrieve the results. The BPEL process is automatically dehydrated any time it is waiting for a message to arrive or an event to happen.

**Launch Process Activity**

Processes are launched just like any other service using the Invoke activity in BPEL. (Also see Sub-processes above)

**NOOP Activity**

The NOOP activity in OWF maps to the Empty activity in BPEL.

**Loop Counter Activity**

See modeling of loops described above.

**Role Resolution Activity**

Role resolution is handled differently in BPEL PM; instead of modeling it in the process diagram, role resolution is achieved by specifying Group rules in the Work List application. This rule based resolution is more powerful and enables different rules to be specified for different periods of time or based on other criteria.
Notification Activity
The Human Task activity should be used in BPEL PM to assign and route tasks. The human workflow functionality includes sophisticated pattern based routing as well as a rich set of notifications and reminders.

Vote Yes/No Activity
Group voting is one of the routing patterns supported by the human workflow component of Oracle BPEL PM. This pattern routes work to a group of people in parallel and can be configured to proceed when the outcome is determinate instead of waiting for all responses.

Master-Detail Coordination Activities – Wait for Flow and Continue Flow
In current version of Oracle BPEL PM, Master-Detail coordination may be achieved using Receive and corresponding Invoke:

- The Wait for Flow functionality may be achieved by using a Receive activity in BPEL. As discussed above, Receive waits for matching message to arrive
- The Continue Flow functionality may be achieved by using the Invoke corresponding to the Receive. Such an Invoke when executed will cause the waiting process to receive the message and continue.

Future version of Oracle BPEL PM may have enhanced support for this pattern.

Get Monitor URL Activity
Such an activity is not needed in the BPEL PM model. Instead, the clients needing to use such URLs should use BPEL PM Admin APIs.
Transitions

Activity Result Based Transitions

In BPEL, conditional transitions are modeled using explicit Switch activity. The conditions are modeled as XPath expressions. Note that this is more comprehensive functionality because:

- It enables activities to return any desired results instead of simple lookups
- Branching conditions may be based on not only the returned results from the immediate activity but also from any other activity as well as other data available to the process.

![Switch activity example]

Figure 8: Example of a Switch activity

The above picture shows conditional branching to three branches, with the third one being the fall back default (otherwise); the condition associated with the second branch is also shown.

The usage of Switch is illustrated in the sample references Switch shipping with Oracle BPEL PM.
**Timed Transition**

In BPEL, time outs may be achieved by using the Pick activity. Pick waits for one or more messages to arrive or alarms to fire; it continues when the first message is received or alarm is fired.

![Figure 9: Using Pick to timeout](image)

Note that such time out can only be used in the context of timing out the receipt of a message. However, this should be sufficient as in BPEL any long running activity should be broken into an Invoke and a Callback.

The onAlarm construct of BPEL may also be applied to any activity or scope in BPEL to do some processing if an alarm is fired in parallel to the main flow.

The usage of Pick is illustrated in the sample references\Pick shipping with Oracle BPEL PM.

**Lookups**

Lookups were used in Oracle Workflow to do activity result based transitions. In BPEL, transitions are instead model with XPath condition expressions as discussed above. Therefore, lookups are not needed in BPEL. You can use process variables to store the results of activities, which may be any XML document and not just lookups.
CONCLUSION

While there are differences between Oracle Workflow and BPEL, once the concepts are understood, migrating Oracle Workflow processes to BPEL would be a matter of mapping one’s concepts and constructs to the other’s.

Customer benefits of migrating to standards based functionality of BPEL PM include:

- Comprehensive Functionality
- Knowledge Portability
- Toolset Extensibility and “Hot Pluggable” products
- Interoperability
- Vendor Portability

Oracle Applications may have the largest number of Oracle Workflows. They also will be migrating to the BPEL PM to realize the above benefits.