BPMN 2.0 versus BPEL

Meera Srinivasan, Director, Product Management, Oracle Business Process Management

With its Visio like simplicity and support for swimlanes, BPMN 1.1 has been widely adopted as the standard for business process modeling among both business user community as well as vendors. Vendors positioned BPMN as the business model for business-IT collaboration and translated BPMN 1.x models to BPEL and XPDL models for business process execution. This resulted in impedance mismatch and classic business-IT gap as there were translation issues and the models looked different in the business and IT worlds.

With the latest version 2.0, BPMN is no longer restricted to just business friendly representation of business processes and has been elevated to an execution language similar to BPEL. BPMN 2.0, with its explicit meta model, interchange format, precise execution semantics behind the various shapes and symbols, support for inter-process communication, transaction, compensation and error handling semantics supports all features critical to making a business process executable. With BPMN 2.0, the model is the executable and the model drives the implementation. There is no translation and the round-tripping problems associated with BPMN to BPEL are completely eliminated.

Oracle BPM Suite 11g is the first BPMS platform that supports BPMN 2.0 at design-time and run-time. The Oracle BPM Studio, the design-time component of Oracle BPM Suite 11g platform supports BPMN 2.0 process modeling, analysis and implementation. The Oracle BPM Studio 11g component is a layering on top of SOA Composite Editor and includes BPEL Editor, Rules Editor, Task Editor and Adapter Editors. In addition to Oracle JDeveloper based BPM Studio, the BPM Suite 11g platform supports a web-based, role-based BPMN 2.0 process modeling experience through the BPM Process Composer component. BPM Process Composer provides collaborative process modeling capabilities to the business audience thus empowering business stakeholders to participate in the definition and requirements specification of the business processes.

The process engine in Oracle BPM Suite 11g supports both BPEL and BPMN 2.0 natively. BPEL processes compile into a BPEL activity set and BPMN processes compile into a BPMN activity set, both of which execute on the same process core. This architecture provides consistent experience, behavior, and performance/scalability regardless of the usage of BPMN or BPEL. The Oracle BPM Project can contain both BPMN and BPEL models. The BPMN system process steps can invoke BPEL processes and the BPEL process can invoke the BPMN processes exposed as a Service.

BPMN and BPEL are complementary standards.

BPMN 2.0 is great for the following:

- The world of business processes has changed dramatically over the past few years. Processes can be coordinated from behind, within and over organizations natural boundaries. A business process now spans multiple participants and coordination can be complex. BPMN is suited for less structured business processes as it does not restrict the process model to a block structure and supports all flavors of processes including ones that have dynamic and complex human interactions.
- With its support for swimlanes and a business-friendly visual representation, it is suited when business people are involved in the business process development lifecycle. BPMN facilitates providing a business view that is easy for business people to understand and provides a more detailed level for process implementers.
- BPMN is targeted at users, vendors and service providers that need to communicate business processes in a standard manner. BPMN processes are not restricted to execution and they can be used for process documentation and to promote shared understanding of business processes across the enterprise.

BPEL on the other hand is more suited for the following:

- Developer models and implements typically in integration scenarios. In addition, BPEL is sufficient for processes involving simple workflows.
- Developers prefer programming language kind of experience where they can design in XML if needed and has Java like try-catch exception semantics.
- Ideal for composing low level services to expose coarse-grained Business Service that can in turn be referenced from business processes.