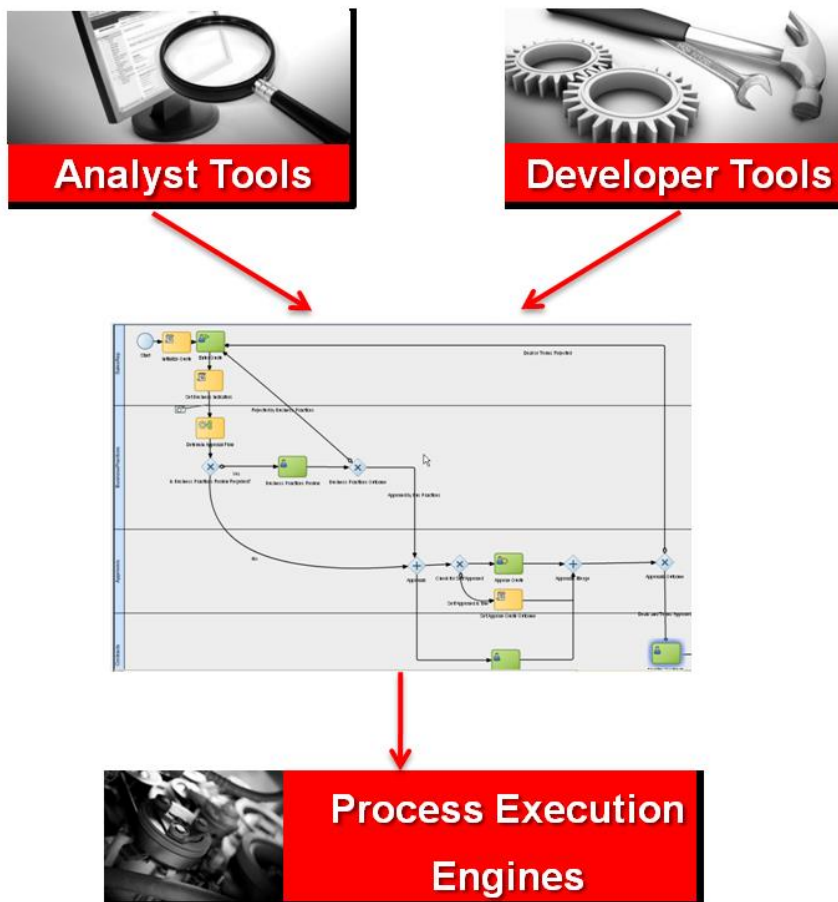


## Quick Overview of BPMN 2.0

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

BPMN stands for **Business Process Modeling Notation** and is a public standard maintained by **OMG**. It describes a business-friendly, flow chart like graphical notation that business process analysts and business users can use to model business processes and has support for process interactions, exception handling, compensation semantics etc. It is widely accepted by both commercial and open source BPMS tooling vendors. It is highly adaptable and can be used to capture all the way from abstract process outlines to detailed process flows to implementation ready processes. One of the main value propositions of BPMN besides being a diagram standard is the precise semantics behind the diagram. The shape, the symbols (also referred to as markers), the borders, the placement of the BPMN diagram elements as well as their properties have well defined meanings and have to be interpreted in the same manner by all tools.

While **BPMN 1.1** comprehensively address process modeling notations, it's failure to address an interchange format (for diagram exchange) has resulted in implementation vendors adopting different standards (**BPML**, **XPDL**, other proprietary formats) to store BPMN process models leading to not only loss of portability across tools but also making it difficult to communicate across the various stakeholders. The vision of BPMN is to have a single specification for notation, metamodel and interchange. In addition, BPMN 2.0 has been expanded to include orchestrations and choreography of process models.



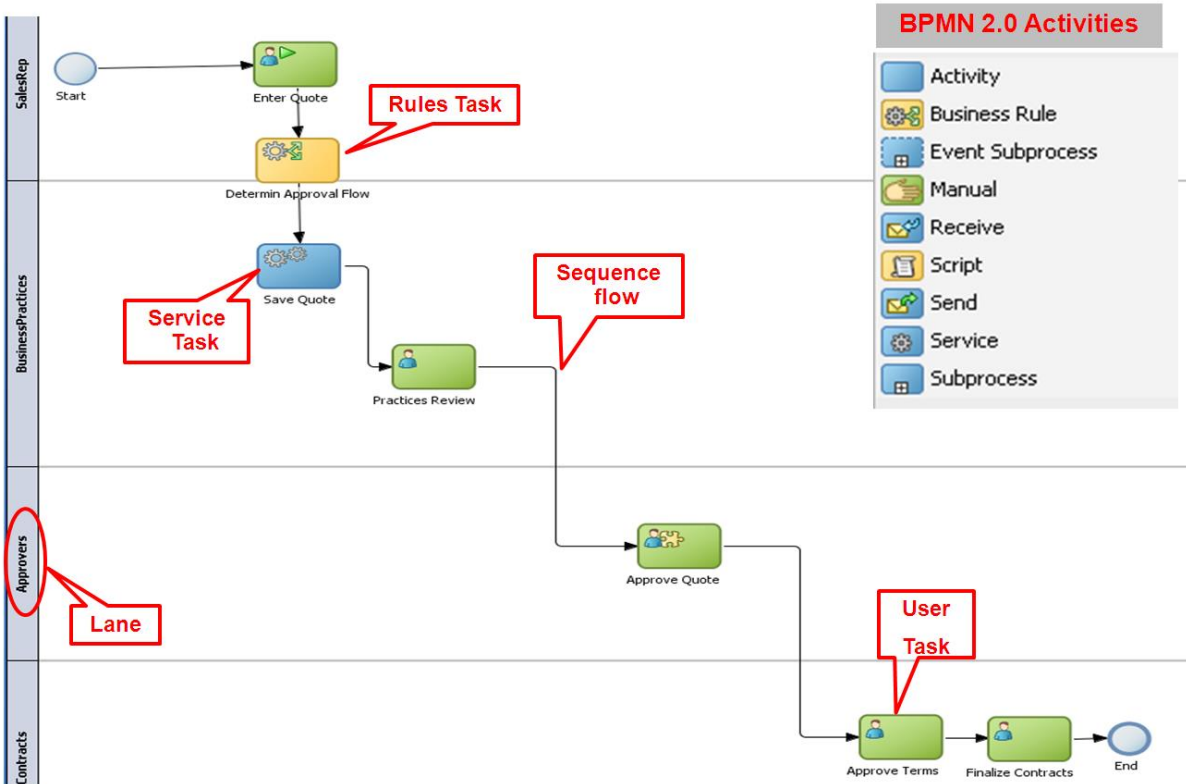
Salient enhancements to **BPMN 2.0** are as follows:

- BPMN 2.0 includes both diagram interchange as well as model interchange (the interchange formats can be either XML or UML) enabling portability of BPMN models across tool vendors.
- Formal execution semantics for all BPMN elements – BPMN 2.0 can not only be used to capture process models but can be used as an implementation model as well.
- Defines an extensibility mechanism for both Process model extensions and graphical extension.
- Refines event composition and correlation
- Extends the definition of human interactions and aligns BPEL4People with the BPMN specification.
- Defines a Choreography model

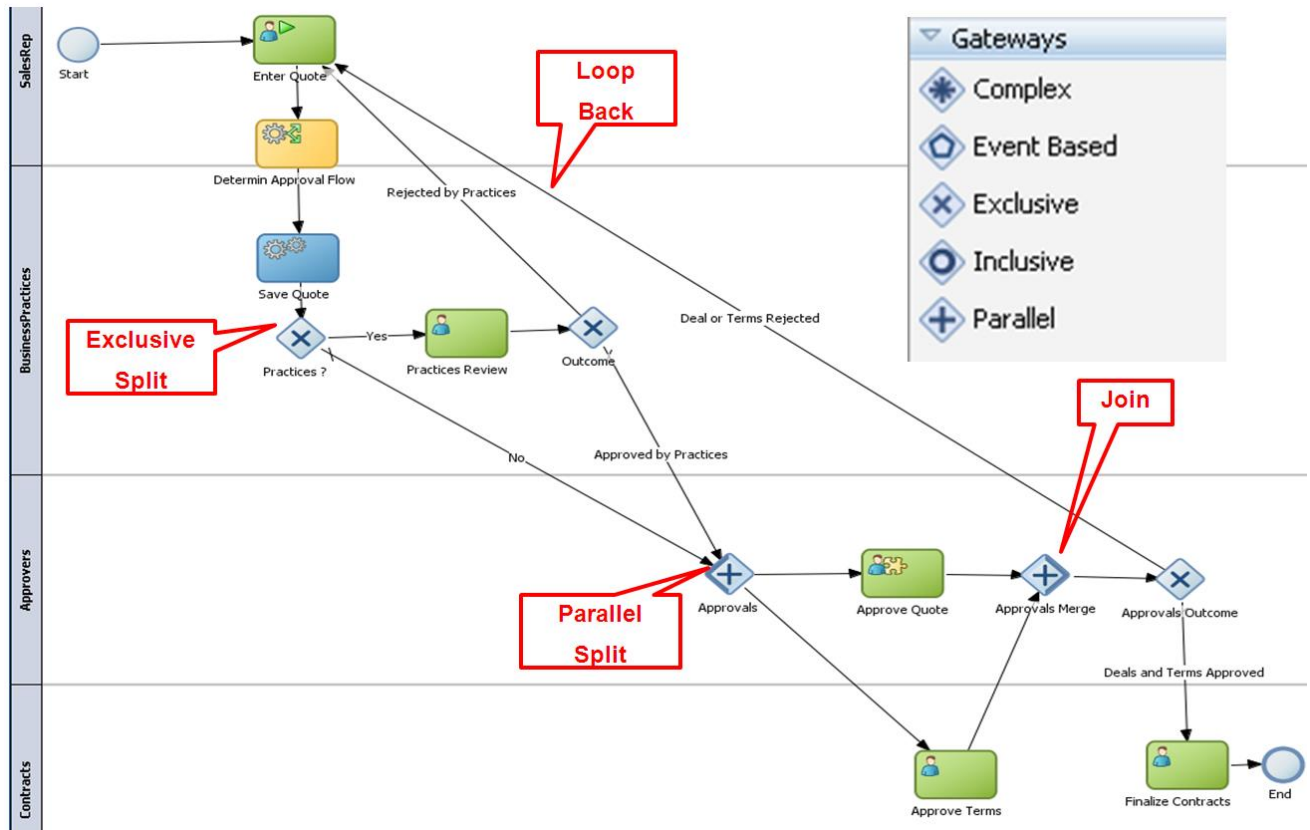
At its heart, BPMN has only 3 main elements also referred to as Flow Objects – Activity (rectangle), Events (circle) and Gateways (diamond). An Activity represents some work done; Gateway represents a decision point or parallel forking or merge or join; Event represents either a trigger generated by the process or received by the process (from external source or from some other part of the process). These **Flow Objects** are linked by connections referred to as Sequential Flows. These Sequence Flows represent the chronological sequence of process steps. The preceding steps pass control to the following step(s) along the connection. The data is also passed along the connection flow.

The **Activity** can be either a **Task** (an atomic process step) or **Embedded Sub-process** (compound process step). The Embedded Sub-Process can be either expanded or collapsed and has access to the process data. BPMN 2.0 supports different flavors of Tasks namely: **User Task** for human step managed by the workflow component of the BPM run-time engine; Manual Task for human step that is not managed by the BPM run-time engine); **Service Task** for synchronous system interactions; **Send Task** and **Receive Task** for asynchronous system interactions; **Script Task** for scripting needs; **Call Task** for invoking another BPMN process (process chaining). The different task types have different symbols or markers to visually distinguish them.

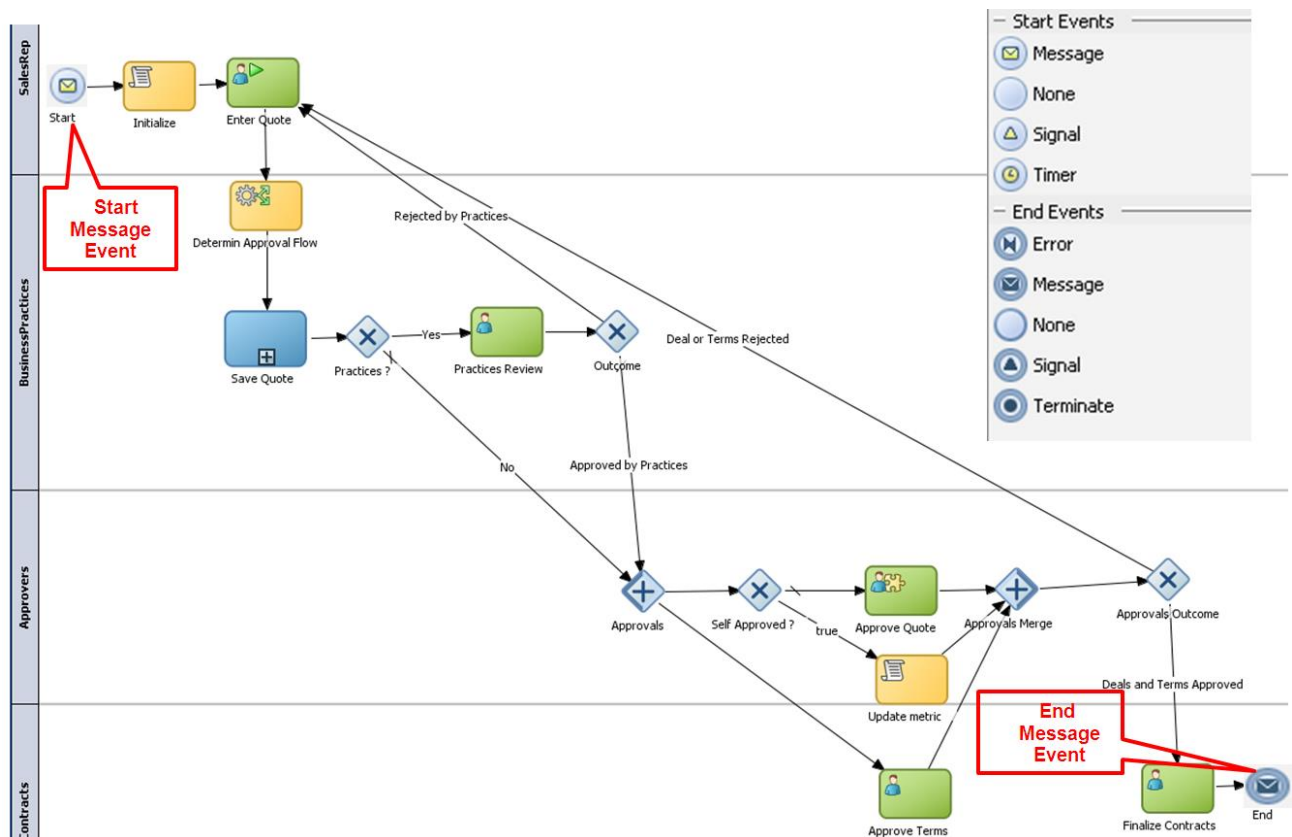
In BPMN, the Lane objects are used to group activities based on the categories (can be human resources or system resources) that they are associated for better visualization purpose. In Oracle BPM Studio, the Lanes are associated with the BPM Role object and the Performer of the User Task is automatically set to the BPM Role object associated with the Lane.



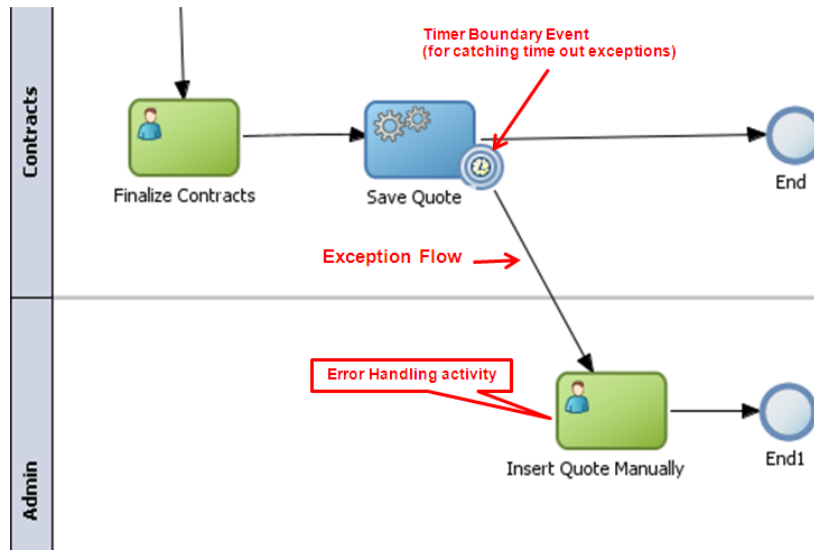
The **Gateways** are used for conditional data splits, conditional merge, parallel forking and parallel joins. The conditional data splits can be exclusive (XOR Gateway – one and only one path can be taken) or inclusive (OR Gateway – one or more paths can be taken. The XOR Gateway is also used for exclusive conditional merges and the OR Gateway for inclusive conditional merges). The parallel forking (AND) is used to indicate parallel paths. The AND Gateway is also used for joining parallel paths.



The Events can occur at the beginning (Start Events) or at the end (End Events) or in the middle (Intermediate) of the process. The Events can be of catch (receive trigger) or throw (send trigger) type. The Start Events are always of catch type and the End Events are always of throw type. The Intermediate Events can be either of throw or catch type. Similar to Activities, there are various flavors of Events to denote the type of trigger. The Events can be of different flavors – Message Events for inter-process communication, Signal Event for publish/subscribe paradigm, Error and Timer Events for exception handling and wait semantics etc.



Certain Intermediate Events of catch type can be attached to the boundary of an activity (Task or subprocess) and can listen to internal as well as external triggers. These events are referred to as **Boundary Events**. When the boundary event is triggered, the process execution flows through the outgoing path also referred to as exception flow. The Boundary Event is active and listening to trigger only as long as the activity to which it is attached is active. The boundary event can be of 2 types – interrupting and non-interrupting. The non-interrupting boundary events have a dashed border to distinguish them from interrupting boundary events. In the interrupting case, the process execution proceeding on the normal sequence flow is interrupted and instead proceeds on the exception path. In other words, the normal execution flow and the exception flow are mutually exclusive. In the non-interrupting case, the process execution continues on the normal sequence flow and in parallel also executes the exception flow. Care must be taken when the exception flow is merged into the main flow of the Process. In most cases, the exception flow should end with its own End Event. Oracle BPM 11g supports various Boundary Events like Message Boundary Event, Signal Boundary Event, Error Boundary Event and Timer Boundary Event. It supports both interrupting and non-interrupting Boundary Event types.



BPMN 2.0 also supports a new construct called Event Subprocess. The **Event Subprocess** is contained within a process or a subprocess. It is not applicable to a Task. It is marked with a dotted line boundary and triggered by a Start Event. The Start Event is followed by activities. It does not have incoming or outgoing sequence flows. It can be collapsed or expanded. When collapsed, it has the marker of the Start Event. Types of Start Events for Event Subprocess are Message, Error, Timer and Signal. The Event Subprocess has access to the data of its parent scope. Just like boundary events it listens for external signals but instead of transferring outside of the activity it runs within the activity (process / subprocess). Oracle BPM 11g supports both interrupting and non-interrupting flavors of Event Subprocess.

