Session 3:

Business Process Modeling (BPMN):
Overview, Activities, Flows

Richard Welke
Director, CEPRIN
Professor, CIS
Robinson College of Business
Georgia State University
Atlanta, GA
Learning objectives

- To provide an overview of the basic ideas of modeling with BPMN
  - The ideas of orchestration and choreography and how they distinguish the process you’re focused on versus the other processes that process interacts with
  - The three basic shapes used and their role
  - Meaning and types of flows
  - Where the data is in a BPMN-modeled process
- To gain a deeper understanding of the types of activities/tasks that are used in BPMN
Benefits of BPMN

- Open standard (OMG)
- Multi-vendor support
- Business-friendly
  - Only three basic shapes
  - But specializations within each shape
- Explicit treatment of event-based process behavior
  - Allows capturing of common condition and exception handling
- Distinguishes sequence flows and message flows
  - Fits well with Service-oriented Architecture support
- Supports an end-to-end view of the process
  - Collapse/expand sub-processes to hide detail
  - Allows for multiple, distinct business processes interaction (choreography)
Pools

- Each “pool” is associated with a single process that is:
  - Separately owned, managed & independently running
- The main pool is the process you wish to execute & manage
  - It’s the implied “domain of control” (called an “orchestration”)
- A BPMN diagram can contain multiple pools

Activities under the control of an “employee”
Generally don’t know what these are & “black-box” them

Activities under the control of “Vacation Request” process
Here you provide details
Swimlanes (lanes)

- A **swimlane** represents a **role** at some level of abstraction:
  - Organizational unit
  - Position or job title (role)
  - More generally, common grouping of functional capabilities
- Used to visualize often complex interactions between units and roles within a process

Manager Lane

Activities performed by or the responsibility of a “Manager”

HR Lane

Activities performed by or the responsibility of “HR”
Orchestration & choreography

**Orchestration**
- The managed flow of control **within** a business process
- Represented by **sequence (control) flows** (solid lines with arrows)

**Choreography**
- Pattern of messages **between** separate, distinct processes (pools)
- Coordination of processes – done by **message flows** (dashed lines with arrows at one end and small circles at the other end)

Managed process 1 (Approver’s process)

Managed process 2 (Booking process)
Orchestration & choreography II

- External (to the main process) pools can be abstract processes
  - While you can show their contents, they’re generally “black boxes” (if you have no control over them)
- Gray-boxing may be useful to show dependencies and suggest “To-be” alternatives
  - Gray-box means some activity details are provided

This process (pool) is considered “external” to the main process you’re focusing on. Therefore it is typically left as a “black-box” (no detail)

The process you’re focusing on is Travel Booking below

- Travel Request
- Options
- Option Selection
- Confirmation & Invoice
- Propose Options
- Send Options
- Receive Selection
- Create Confirmation and Invoice
- Confirm & Invoice
What’s “flowing”?  

**Within** a process (Orchestration)
- A transaction (the thing that initiates the process)
  - Abstract transaction; NOT the data of the transaction
- Transaction moves from object to object in the BPMN diagram using **Sequence flow** (flow of control)
  - Target activity *enabled* to start when preceding activity ends
  - Actual start will depend on available resources needed to perform activity

**Between** separate process pools (Choreography)
- **Message flow**: signal passed between pools
  - Request/response, or unsolicited signal
  - NOT a flow of control
  - Process responds to signal via a message event

**Data/document flow**
- If explicitly shown, then as annotation only
- Represented in most modelers as parameters of individual activities
A “token” that moves from object to object and “enables” that object to begin its work; the token is the initiating transaction.
Where’s the data?

If Flow within a BPMN modeled business process (the lines between activities, etc.) denotes sequence (only)

...then, where’s the data?

Answer:

- Messages into the process contain the external data for use within the process pool (e.g., forms, DB accesses)
  - Can be explicit (as in message flows), or
  - Implicit, such as the invocation of an external service
- Activities transform the data, and create or modify the data
- All data for a particular instance of a process (one end-to-end execution) is kept in a data store for use by any other action in that process (transaction data)
- In different words – data doesn’t “flow”
- It is captured when it arrives or is created, kept in a data store, and made available to any subsequent activities, gateways, etc. associated with that transaction

See next slide for a graphic of this …
Data “flow” in a typical BPMS

Basic concept
- When data is available at/from one activity it’s available to the process designer for all following activities to use (not just the next sequential one)
- Different modeling tools and model standards adopt different approaches to how this is represented (and where)
BPMN Overview

- Basic shapes
- Specializations of shapes
- Sub-processes & repetitions
- Events
- Gateways
- Artifacts (annotation symbols)
Three basic BPMN flow objects

- **Activity**
  - A step in the process
  - Represents work/action performed
  - Called a “task” if it’s the smallest unit of work represented

- **Gateway**
  - Controls flow branching, merging, and parallel actions
  - Pure logic; result *based on data previously obtained/analyzed*

- **Event**
  - Def’n: A signal (message) that something has happened
  - Can start, pause, resume, interrupt or redirect a process or activity
    - *Catch*: Inbound events; *Throw*: Outbound events
Activity

- Work performed in a BP (business process)
- Task
  - Atomic activity (indivisible work unit)
  - Task types (specializations)...
    - BPMN: Service, User, Send, Receive ...
    - Tool: depends on actions supported
- Sub-process
  - Compound activity that may be decomposed into tasks and sub-processes
  - Can be shown expanded or collapsed
    - “+” sign shown within activity to indicate it’s expandable
    - Two kinds: In-line and external (re-usable)
Task specializations

Modeling tools allow for a set of activity/task “specializations”
(OMG has a number of these specializations)

- Typically “tailored” to the underlying process execution engine of the modeling tool vendor
  - Properties needed to specify their behavior
  - How they “behave” when process execution is done

- Specializations are enumerated in the tool palette
- Or “right click” on object and select its specialization
Task specializations (BizAgi) part I

- **Plain task** is a generic placeholder; used when first scoping out a process.
- **User task** designates a man-machine task involving both a user and the use of a computer (e.g. receive digital form to complete).
- **Manual task** means the task is non-automated – useful when manual work done – e.g., “fill a tire”.
- **A service task** is when the process automatically calls an external service to get some work done (see also: reference task – executes another BP).
**Task specializations (BizAgi) part II**

- **Script**: Used when a script (e.g., Java) is specified for how the task is to operate.

- **Send**: Creates and sends a message to an external pool (similar to a “throw” message event).

- **Receive**: Receives and processes a message sent from an external pool (similar to a “catch” message event).

- **Re-Use**: Places an external sub-process in a diagram.
  - Clicking on the “+” opens up a new diagram where the sub-process details are provided.
Sub processes

- **Compound, decomposable activity**
  - Displayed as a collapsed activity (with + sign)
  - Type I: Embedded in same pool; expands when clicked on within the existing diagram
  - Type II: Independent subprocess that opens a separate business process diagram
    - Allows for separate creation and re-usability

- **Strict containment semantics**
  - Sequence flow self-contained
  - One entry point, one exit point
Separate (not embedded) sub-process

- Generally used for more complex sub-processes
- Also, used when parts of standardized processes are to be re-used
  - Make each “re-use” process a named sub-process
  - Can then use it in any other business process as well

Example:

Click on the + in the “Register Participant” activity and it expands to a full, separate BP diagram with the process details regarding “register participants”

Note: This typically opens a separate diagram
Repeating activities

- Applicable to tasks or subprocesses
- Two types:
  - **Standard loop**
    - Sequential iteration of activity
    - Key: Basis for exiting is evaluated after each iteration (not known in advance)
    - Like a “Do-While” construct in programming
  - Key: basis for stopping
  - **Multi-instance loop**
    - Key: Iteration for “N” times where “N” is known before entering
    - All must complete for activity to complete
    - Like a “For-Each” construct in programming
FAQ’s

Q: When do I begin using “Task Specializations”?
A: Use the “plain task” when you’re initially building a process diagram
   - An exception to this would be Send/Receive tasks as these will be specifically linked to in/out messages to the process

   Then, refine your diagram with how these tasks are currently (As-Is) or should be (To-Be) performed

Q: I don’t see specializations for loop type
Gateways

Control of process sequence flow
- “Take this path, that path, all paths”
- Which path determined by type of gateway & conditions
- Conditional basis for flow determined elsewhere

Types ("split" or "fork"):
1. Exclusive (X-OR) – only one path taken
2. Independent (OR) – multiple paths can be taken
3. Parallel split/fork -- all paths out are always taken
4. Event-based – first path taken for which the corresponding event occurs

Also used for merging ("merge" or "join")
- Synchronization of all valid paths created by split
- Best to show explicitly until you’re better at BPMN
Event (Level 1: start/end only)

- BPMN definition:
  - “A signal that something has happened”
  - Different types differentiated by border style
    - Start (all processes have one and only one)
    - Intermediate (along process flow or activity border)
    - End (all processes have at least one)
  - An icon within the event circle indicates its type
    - Most common intermediate are:
      - Message ("throw/catch"), Timer, Error
Start/End events

Start event
- Thin border
- Usually just one per process or sub-process
  - Indicates “Process starts here”
- May specify the trigger (e.g., a message arrives) by its icon

End event
- Thick border
- Often more than one in a process or sub-process
  - Implicit join of end events on all enabled paths

Terminate event
- Like an end event but when encountered, any and all processing on that process thread ends
Intermediate event

Key differentiator from other process description models

Uses (and types)
- **Wait** for an event
- **Abort and redirect** on event
- **Throw/catch** an exception

Semantics depend on where event occurs in diagram
- Drawn in a sequence flow, can mean
  - Send the signal ("throw")
  - Wait for the signal ("catch," then proceed)
Artifacts

- Way to present additional information not related to sequence or message flow
  - No BPMN-defined semantics

- Data object
  - Linked to connector/activity by association line

- Text annotation
  - Linked by association line (dotted line, no arrows)