Session 4: BPMN Gateways

Richard Welke
Director, CEPRIN
Professor, CIS
Robinson College of Business
Georgia State University
Atlanta, GA
Some reflections on BPMN modeling

- The model you draw reflects the “semantics” of how you wish the process to execute.
- You don’t get to determine model semantics:
  - They’re decided by BPMN specifications.
  - Think of it as a way of “programming” a third-party execution controller (called a BP Execution Engine).
  - When I grade your results I do it based on how a BPMN “engine” would execute your process, not your intent.
- The overall objective is “shared meaning”:
  - What you draw is uniformly interpreted in the same way by all persons looking at your process model.
  - This should be true for *any* model used to transmit meaning & achieve shared understanding.
Gateways

- Used to “split” or “fork” the sequential flow between activities into separate paths.
- And also, to bring the flows back together (“merge” or “join”).
  - Need to “re-join” using a matching gateway, if common, downstream tasks or gateways associated with one or more of the split streams.
  - Don’t have to (re-) join if each has a separate processing path and end event.
  - BUT … process doesn’t end until all valid paths completed.

- Criteria for join gateway depends upon the type of gateway that’s used.
- Many different choices in BPMN; we’ll focus on the more commonly used.
Exclusive OR ("X") gateway

- eXclusive OR gateway is used when only one path can be taken for each transaction being processed
- Key points:
  - The basis for the choice of path is achieved BEFORE the gateway is entered
  - The gateway can only perform Boolean (T/F) logic on pre-existing data
  - X-OR Gateways are best labeled as a T/F question on each outgoing path
- To recombine the paths another "X-OR" gateway is used
- **Best practice** is to specify a "default" path
  - (i.e., the ‘true’ path where, if the other path(s) evaluate “false,” then this path is taken)

BizAgi Note:
To get the “default” symbol, in Gateway properties, go to Advanced tab, and under “condition expression” column, drop down on the row for the path and select “Default”
Sidebar: Gateway logic

- Gateways only perform Boolean routing logic (T/F)
- Based on pre-determined state (external or activity)
- In other words, not this …

Why?
What’s wrong with this?

Answer: Haven’t performed pre-processing to know if the order is valid or not before gateway

But, this …
Sidebar: What is Boolean logic?

- An expression that evaluates to being either true or false (binary)
- \( A < B \)?
- Boolean expressions:
  - \( X = A > B \)
  - \( Y = A \neq C \)
  - \( Z = X \text{ and } Y \)

The data (numeric, logical and textual) used in these expressions come from prior data either entering the process (e.g. via messages) or from activity processing creating new data values.

Gateway evaluates these expressions when entered to determine which outgoing path(s) are true.

If \( A = 9, B = 5, C = 9 \)

What is \( Z \)?
**More on Exclusive (X-OR) gateway**

Only one path taken (hence “eXclusive”)

- Use label on gateway to specify condition (phrased as a question)
- Use labels on flows to indicate outcome
- For execution
  - Default path indicated by back-slash (good practice)
  - Done by right-clicking on the gateway and selecting properties|advanced, then drop-down for the path name
OR gateway

- Also called “Inclusive OR” gateway
- Differs from “eXclusive-OR”
  - Each path is evaluated for meeting the gateway condition and thus, **Multiple** paths can be “true” and taken concurrently
    - (Recall with the exclusive OR – only one path “true” is possible)
  - Exiting paths **must** be re-combined with a merge “OR” gate
- Example below, both A & C paths could be taken
  - All active tokens (paths) must arrive at the merge OR gateway before process can proceed

![Diagram showing an OR gateway example](image-url)
Parallel (AND) gateway

- Used to create required “concurrent” paths
  - Each path executes independently (concurrently) with other paths (sometimes called a “split” or “fork”)
  - Creates a “token” for each split path
  - All tokens must be joined (merged) before ending the path (and process) unless a termination is encountered
  - The Flow out of its joining AND gate doesn’t occur until **all** preceding paths are complete
More on parallel split (AND) gateway

Process sequence flow splits into concurrent segments
- In diagram all downstream paths are all enabled unconditionally
  - No Boolean logic, all paths must be “true”

Each path token must reach end event for process to complete
- (unless it is a termination event)

Parallel segments don’t have to rejoin
- Join gateway *is* needed if parallel paths need to be synchronized before proceeding further

Note: “Concurrency” semantics are poorly/under-defined in BPMN
- Common data changed?
- Each BPMS makes own rules
Event gateway

- Every path out of this gateway has an event on it.
- The intermediate event is the “condition” that must be met to proceed on that path.
  - Gateway “logic” is based on which event happens first.
  -Exiting paths re-combined with an “X-OR” gate – why?
  - What happens if none of these events occur? Possible?
- It’s the only “future looking” gateway.
  - (events are in future)
More on (exclusive) Event gateway

- Only one path taken
  - Based on event occurrence, not data condition
  - Waits on *first event that satisfies condition*
  - **Best practice:** cover situation of no event satisfying
    - Typically done using a “timer” event
  - Uses an *intermediate* event on each gate (path out)

Example (use case)
- Message A: Normal resp.
- Message B: Exception resp.
- Timer: Timeout if no response after 1-Day
Complex gateway

- A set of conditions are evaluated and outgoing path or paths chosen on this basis
  - Combinations of Or, X-or and Event gateway conditions
  -Exiting paths re-combined using a complex gateway
  -Merge behavior (i.e., basis upon which incoming tokens are assessed to determine proper exit) can also be complex
Merge and join concepts: “tokens”

Based X-OR gateway logic, emits one (exclusive) token on one of the paths out of the gateway (e.g. P-2)

Exclusive “merge” waits for one token on its incoming paths; when received, initiates outgoing path

Based on “OR” gateway logic, emits one or more tokens on outgoing paths

The “OR” merge gateway logic waits for all tokens emitted before proceeding to the outgoing sequence flow
Exclusive (X-OR) “merge”

- Merge exclusive alternative incoming sequence flows into one outgoing sequence flow

Technically unnecessary in BPMN – merge of X-OR without gateway using task means same thing

Task 5 also acting as an implicit X-OR merge gateway
Inclusive OR decision gateway alternatives

- Each path evaluated independently
  - All paths evaluating “true” are taken
  - At least one must be “true”
  - Can use “default” to ensure at least one path taken (ME&CE)

Note:
- Inclusive OR not supported in all BPMN modeling tools
Inclusive OR decision alternative

- Representation without gateway
  - Condition 1: Order > $10,000
  - Cond. 2: Domestic client
  - Cond. 3: Default (everything else)
Sidebar: How to get the little diamonds?

1. Right click on the path you wish to make conditional; click on properties

2. Click on the Advanced tab

Change the “Condition Type” from none (default) to “Expression”
Alternative representation (AND Gateway)

Formally ...

Evaluate initial customer order request

Check the client’s credit

Determine availability in inventory

Create shipping options

Alternative allowed

Note:
Here we’re using implied gateways ...
Until you’re familiar with the proper use of gateways, you should initially avoid these simplifications.
Some general guidelines when modeling...

- When first drawing a BPMN diagram, matchup forks and merges with same gateway type.
- Think of as parens ( … ) for every opening paren there should be a corresponding closing one of the same type.
  - You can simplify later where appropriate.
- Clearly label both the criteria of the gateway as well as all non-default paths out of the gateway.
- Don’t overload a gateway for merging, e.g.: