Lab for the course on Process and Service Modeling and Analysis

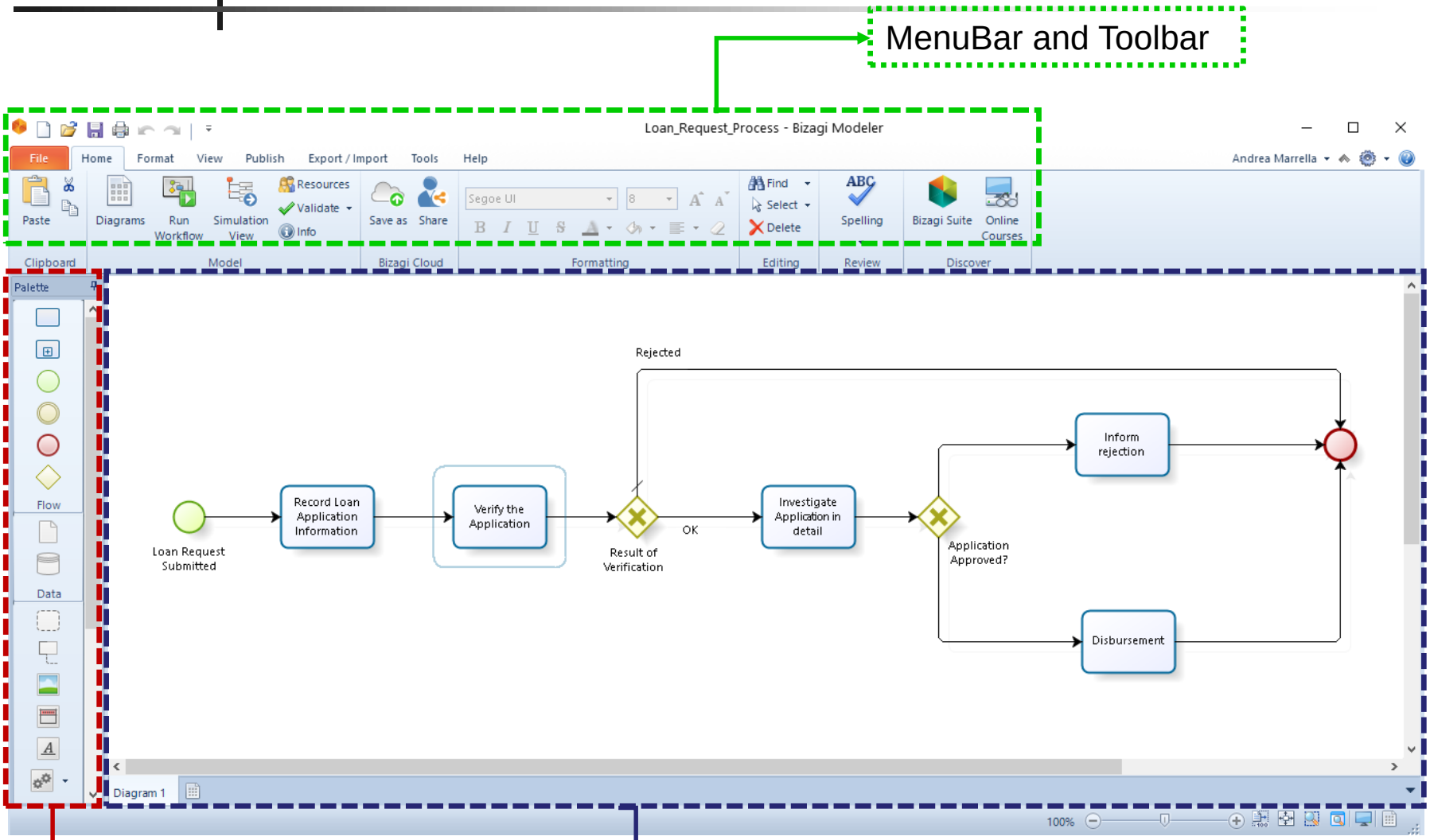
LAB-02
BPMN Basics

Lecturer: Andrea MARRELLA

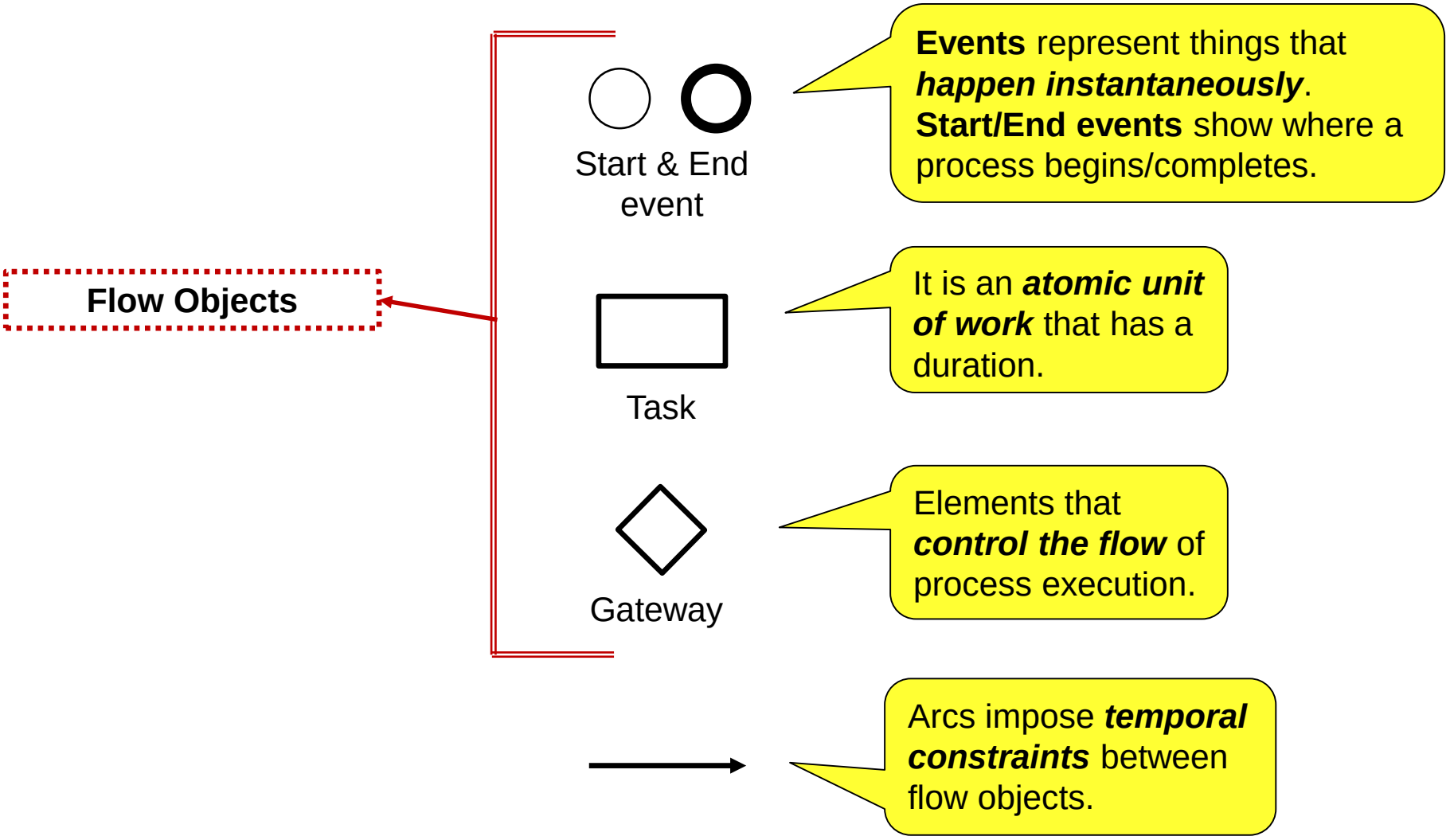
Objectives of this lecture

- Learn how to use Bizagi Modeler
- Recap: BPM Basics and use of gateways
- Checking structural/behavioural correctness of a process with the simulation tool of Bizagi Modeler
- Classroom exercises

Ingredients for this lab: Bizagi Modeler



Ingredients for this lab: BPMN basic contructs



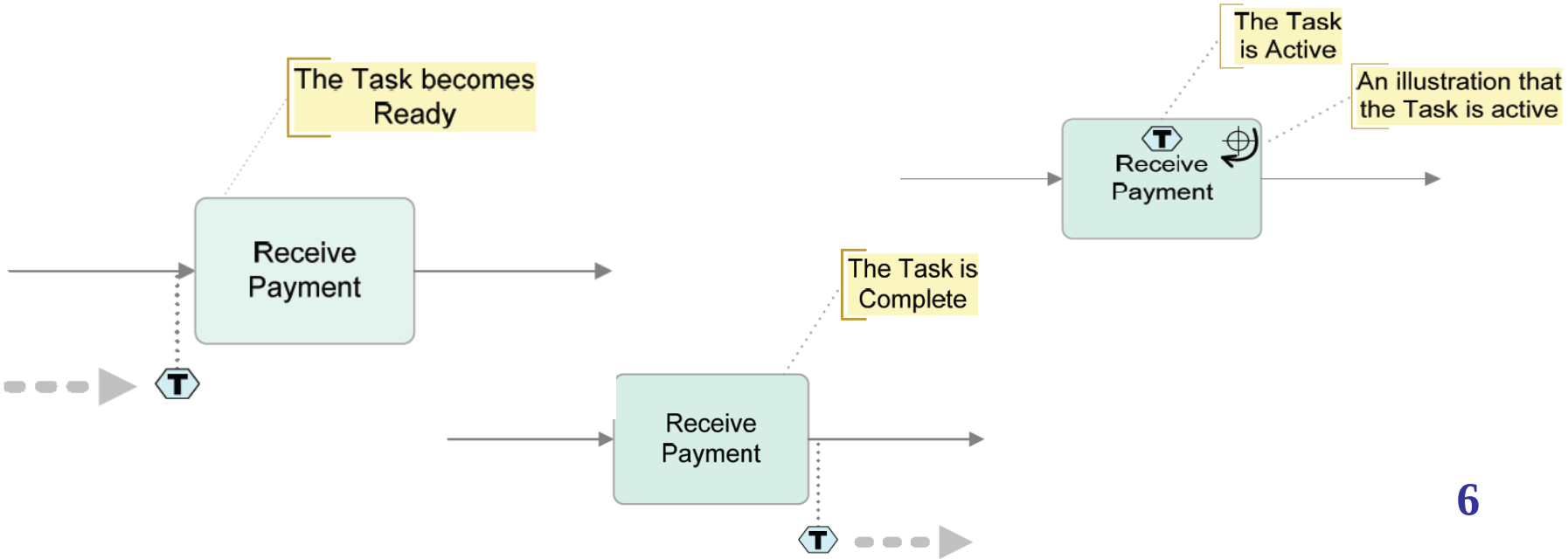
Recap: Connecting Activities

- The **sequence flow** defines the ***order of flow objects*** in a process (activities, events and gateways).
  - In BPMN, each activity can have one or more incoming sequence flow and one or more outgoing sequence flow.
  - Typically, an activity tends to have a ***single incoming*** and a ***single outgoing*** sequence flow.

- Each process must have always at least a ***start event*** (a circle with a thin border), that shows where a process instance can start, and a ***end event*** (a circle with a thick border), for indicating when a process instance completes.
- Starting from a process model, an organization runs a ***number of independent instances*** of this process.

Recap: Activity Behaviour

- Once a process instance has been created, we use the notion of **token** to identify the ***progress*** (or ***state***) of that instance.
- A token is a "***theoretical object***" used to create a descriptive "simulation" of the behavior associated to each BPMN element (it is not currently a formal part of the BPMN specification).
- **A token is created in the start event**, **traverses the sequence flow** and **is destroyed in an end event**. That is, <u>there is no time associated with the token travelling down a sequence flow</u>.



Recap: Gateways in BPMN

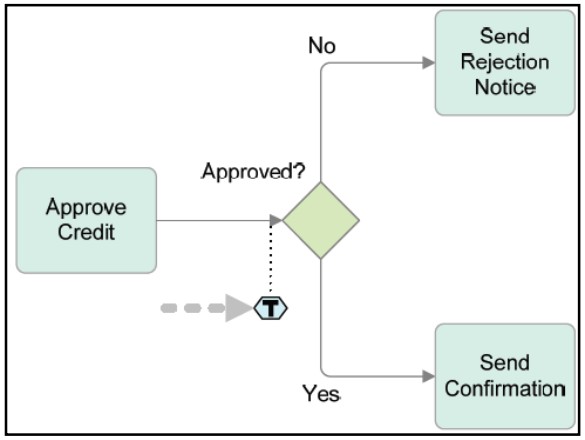
- A ***gateway*** implies that there is a **gating mechanism** that either allows or disallows flow of tokens through the gateway.
- As tokens arrive at a gateway, they can be **merged** together on input, or **split** apart on output depending on the gateway type.
- A ***split gateway*** represents a point where the process flow diverges, while a ***join gateway*** represents a point where the process flow converges.
  - Splits have one incoming sequence flow and multiple outgoing sequence flows (representing the branches that diverge).
  - Joins have multiple incoming sequence flows (representing the branches to be merged) and one outgoing sequence flow.

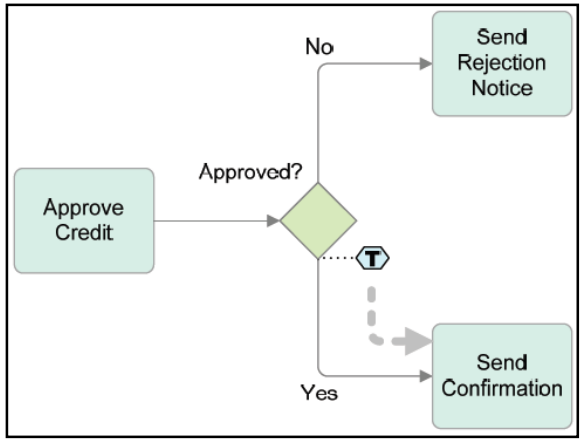
Exclusive Gateways (XOR)

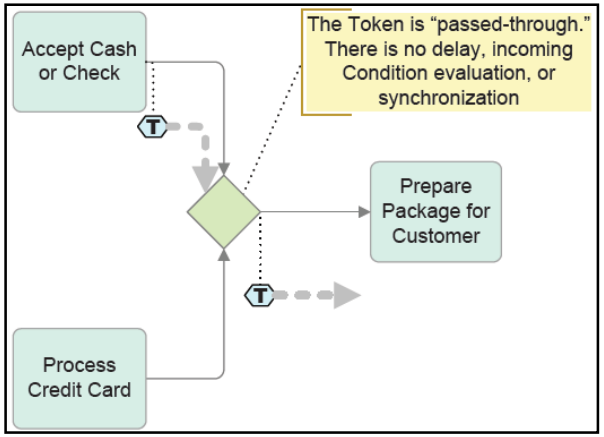
- **XOR Gateways** indicate locations within a business process where the sequence flow can take two or more alternative paths. **Only one of the paths** can be taken.
- The **criteria for the decision** exist as ***conditions on each of the outgoing sequence flow***. **One of those conditions must always evaluate to true**.

When a token reaches a XOR split, there is an **immediate evaluation of the conditions** that are on the gateway's outgoing sequence flow.

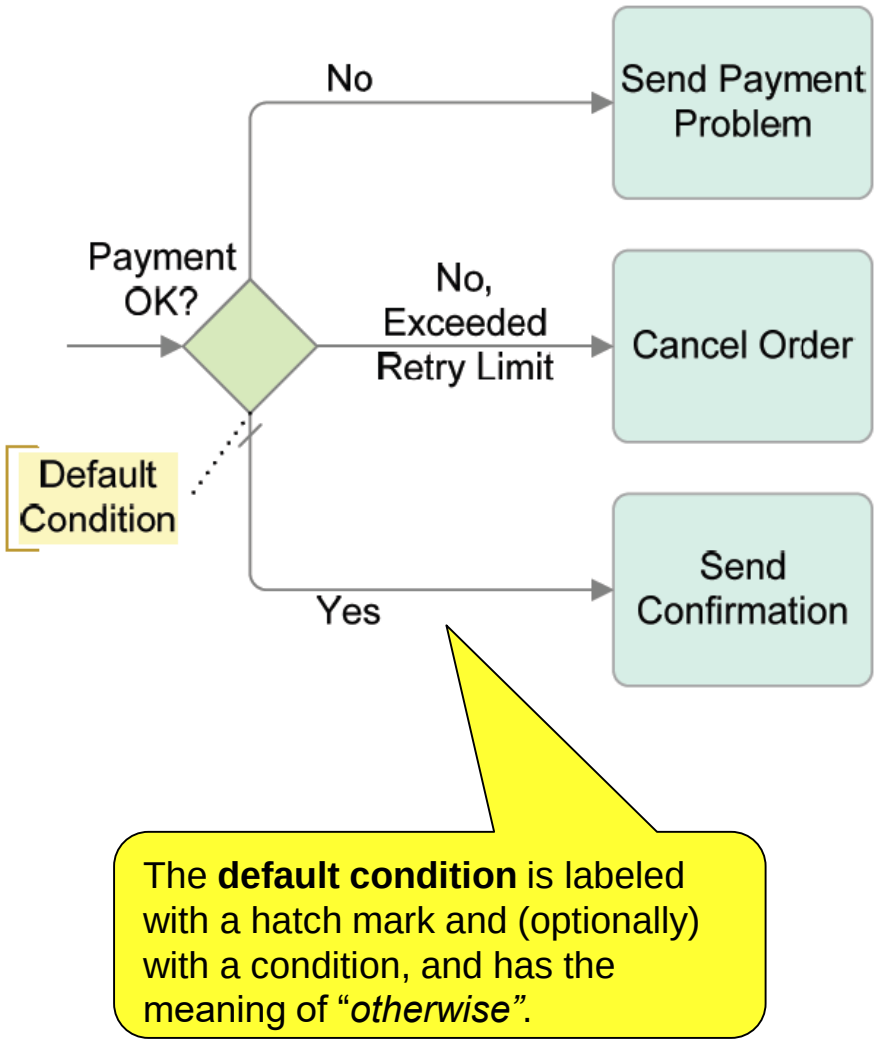
When a token reaches a XOR join, there is **no evaluation of conditions** (on the incoming sequence flow), and immediately moves down the outgoing sequence flow.







Default Conditions

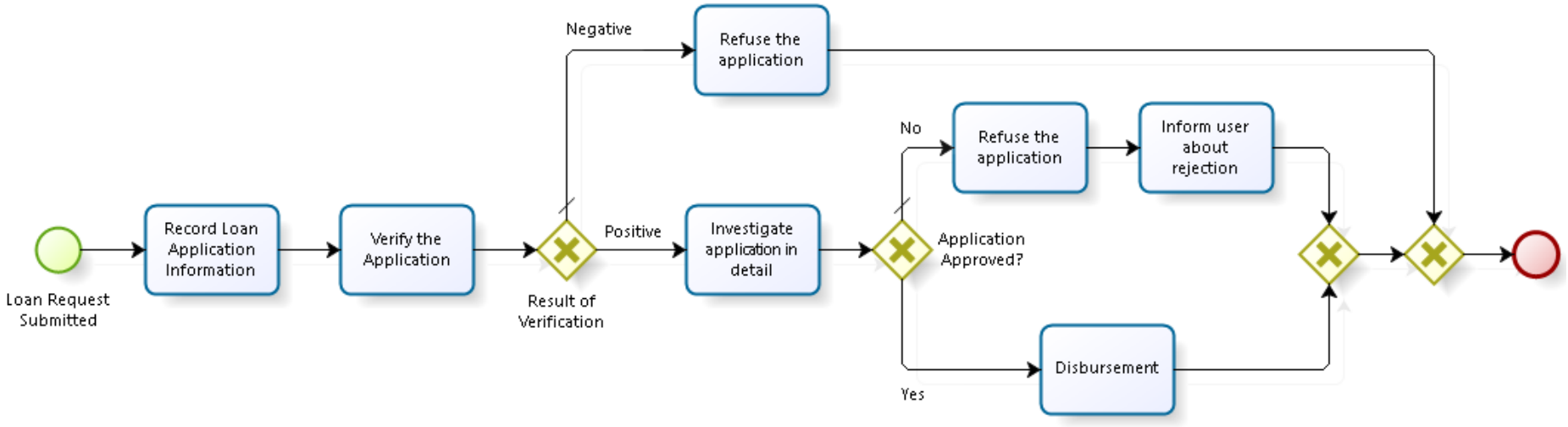


- One way for the modeler to ensure that the process does not get stuck at an exclusive gateway is to use a ***default condition*** for one of the outgoing sequence flow.
- The *default condition* can complement a set of standard conditions to provide an **automatic escape mechanism** in case *all the standard conditions evaluate to false.*
- The *default* is chosen if all the other sequence flow conditions turn out to be false.

Example: Loan Request Process

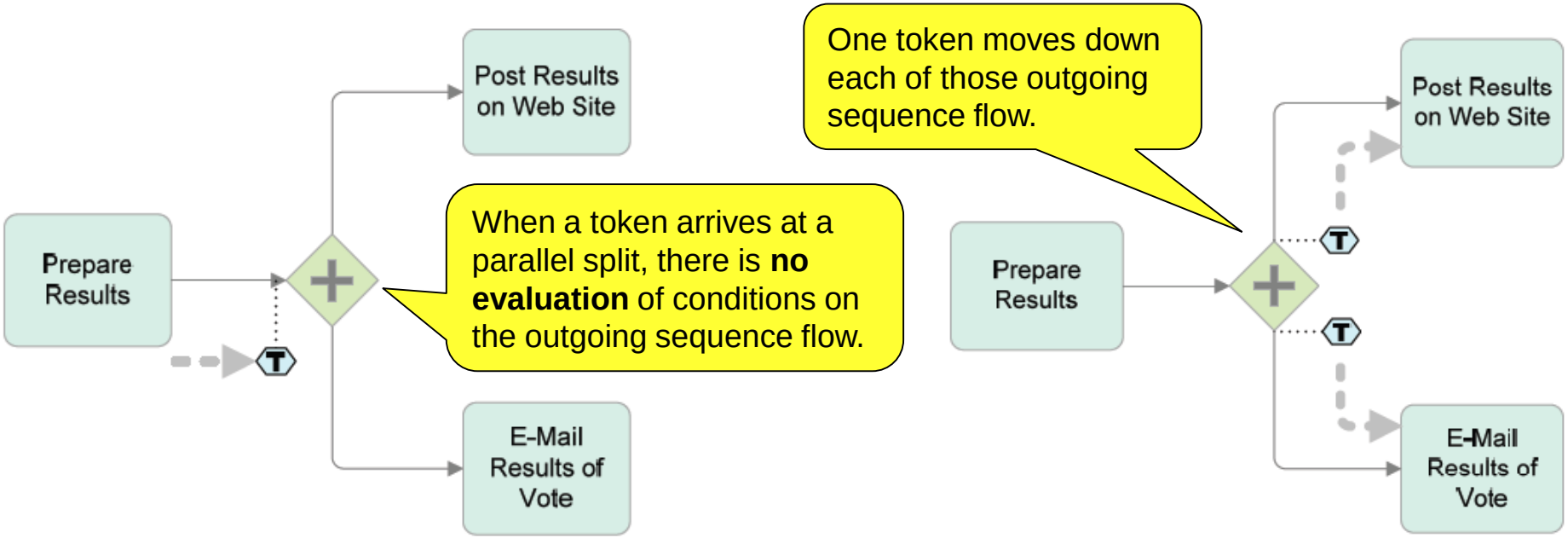
- The Loan Request Process handles the necessary activities to receive, analyze and approve loan applications submitted by customers of a financial entity. A simplified version of this process consists of the following steps.
- When a customer submits a loan request together with the required documents, the information submitted is first recorded in an archive and then the application is verified.
- If the result of the verification is negative, the loan is refused and the process completes. Otherwise, if the result of the verification is positive, the application is further investigated in detail to decide if it can be approved or not.
- Finally, the amount of the loan is disbursed, if approved. If not approved, the loan is refused and the customer is informed about the reasons of the rejection.

Solution


Negative
Refuse the application
No
Refuse the application
Inform user about rejection
Record Loan Application Information
Verify the Application
Positive
Investigate application in detail
Application Approved?
Loan Request Submitted
Result of Verification
Disbursement
Yes

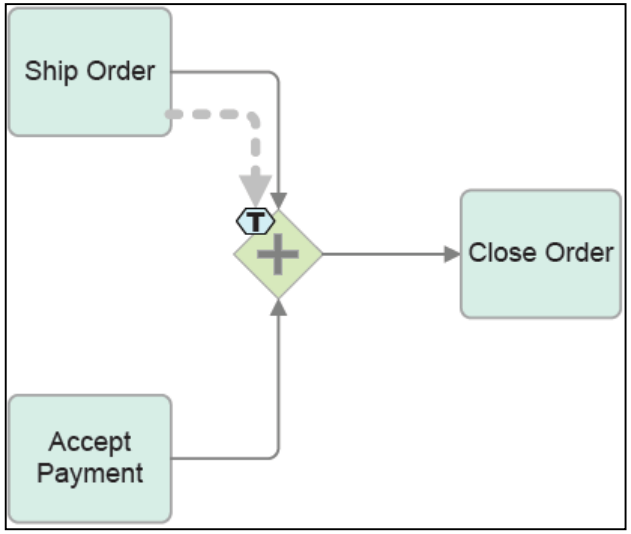
Parallel Gateways (AND)

- **Parallel Gateways** indicate locations within a business process where the sequence flow can take two or more parallel paths. A **parallel split** creates **parallel paths**.
- It creates a number of tokens **that is equal** to the number of outgoing sequence flow.

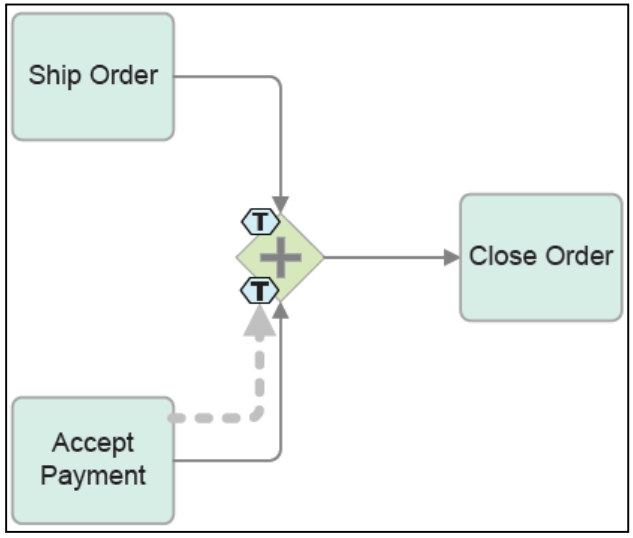


Parallel Gateways – Merging Behaviour

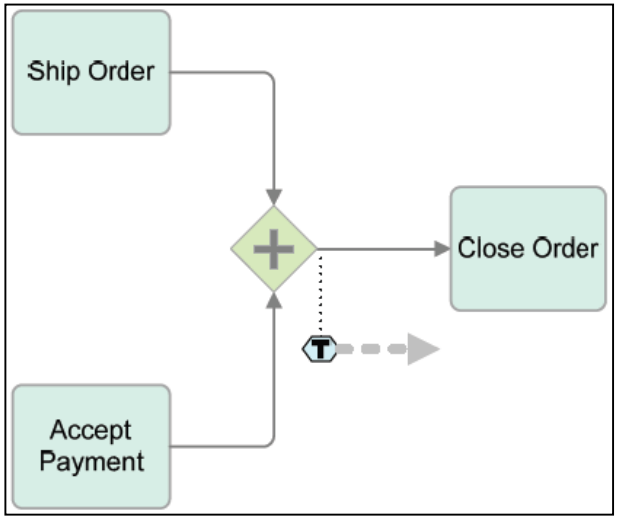
- To synchronize the flow, the parallel gateway will wait for a token to arrive from each incoming sequence flow.



When the first token arrives, there is no evaluation of a condition for the incoming sequence flow, but the **token is "held" at the gateway and does not continue**.



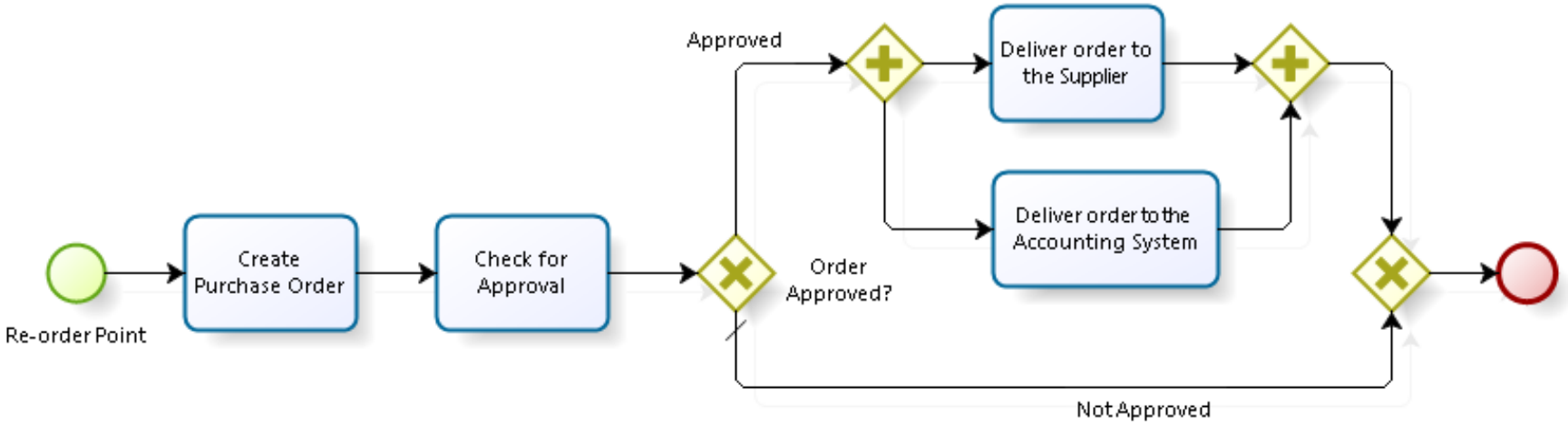
When all the tokens are arrived, they are merged and one token moves down the outgoing sequence flow.



Exercise: Purchase Order Generation

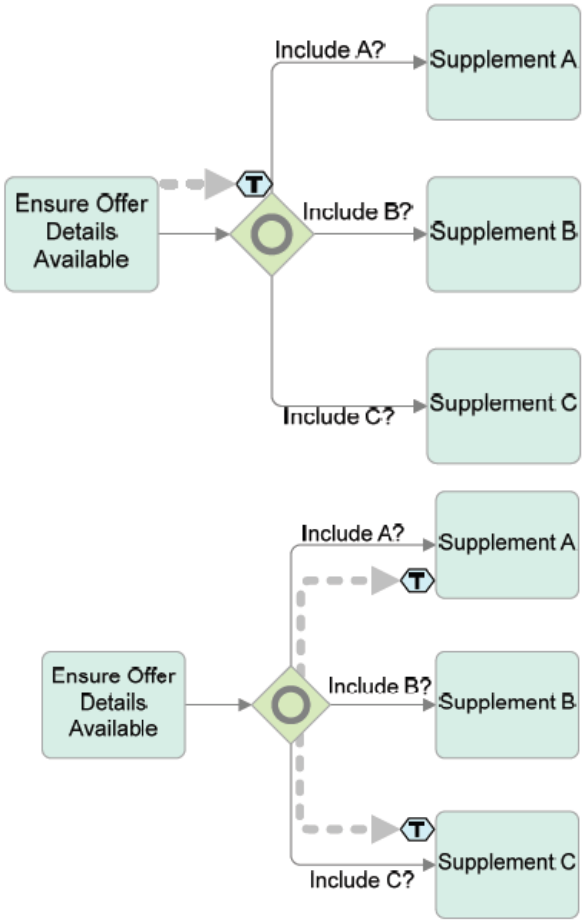
- The goal of this process is to automatically generate purchase orders according to the raw materials inventory levels and to manage their approval, record them in the company accounting system and deliver them to suppliers. The process is organized as follows.
- When the inventory level reaches a re-order point, the process starts and immediately a purchase order is generated and checked for approval or rejection.
- If the order is approved, it will be delivered to the supplier and its details are sent to the account system for being recorded. These two last activities can be performed in any order. Finally, the process completes.

Solution


Approved
Deliver order to the Supplier
Deliver order to the Accounting System
Re-order Point
Create Purchase Order
Check for Approval
Order Approved?
Not Approved

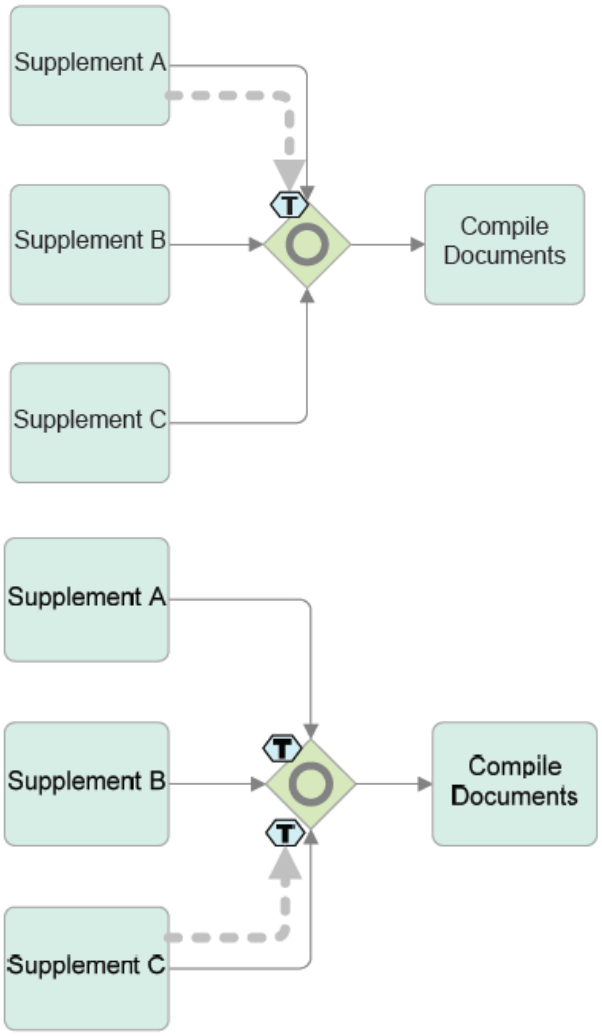
Inclusive Gateways (OR)

- Inclusive gateways support decisions where **more than one outcome is possible** at the decision point based on the conditions of the outgoing sequence flows.



- In terms of token semantics, this means that the **OR-split** takes the input token and generates a number of tokens **equivalent** to the number of output conditions that are true.
- Every condition that evaluates to true will result in a token moving down that sequence flow. **At least one condition *must evaluate to true***.

Inclusive Gateways
Merging Behaviour



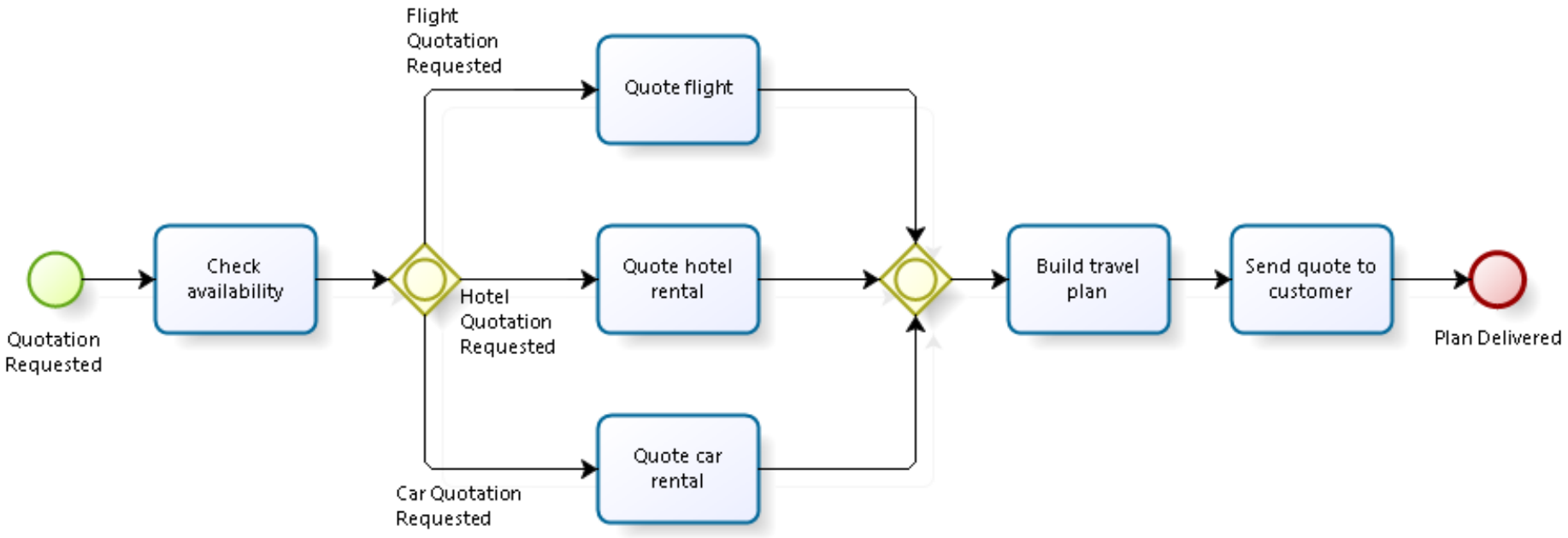
- When the first token arrives at the gateway, the gateway will "***look upstream***" for each of the other incoming sequence flow to see if there is a token that might arrive at a later time.
- Thus, the gateway **will hold the first token** that arrived in the upper path **until the other token** from the lower path **arrives**.

- When **all the expected tokens have arrived** at the gateway, the **process flow is synchronized** (the incoming tokens are merged) and then a token moves down the gateway's outgoing sequence flow.

Example: Travel Plan Quotation

- This process handles request for quotes of travel plans made by customers of a travel agency.
- When a customer requests a quote, a travel agent must verify the availability and calculate the costs of each of the services the customer included in the request (flight, hotel, car rental).
- When completed, a travel plan is built and the quote is sent to the customer.

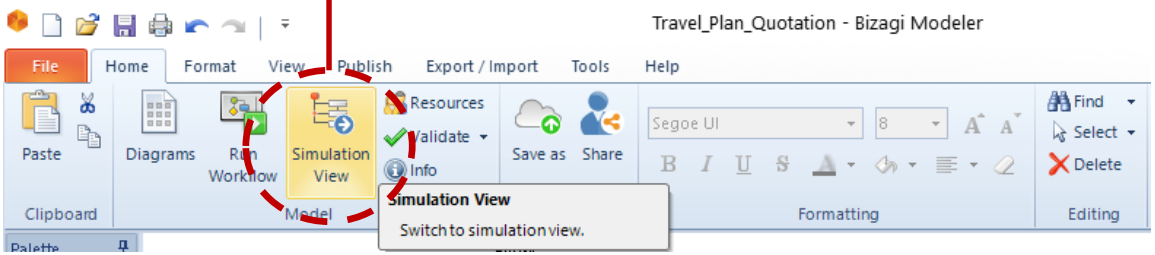
Solution


Flight
Quotation
Requested
Quote flight
Check
availability
Quote hotel
rental
Hotel
Quotation
Requested
Build travel
plan
Send quote to
customer
Quotation
Requested
Plan Delivered
Quote car
rental
Car Quotation
Requested

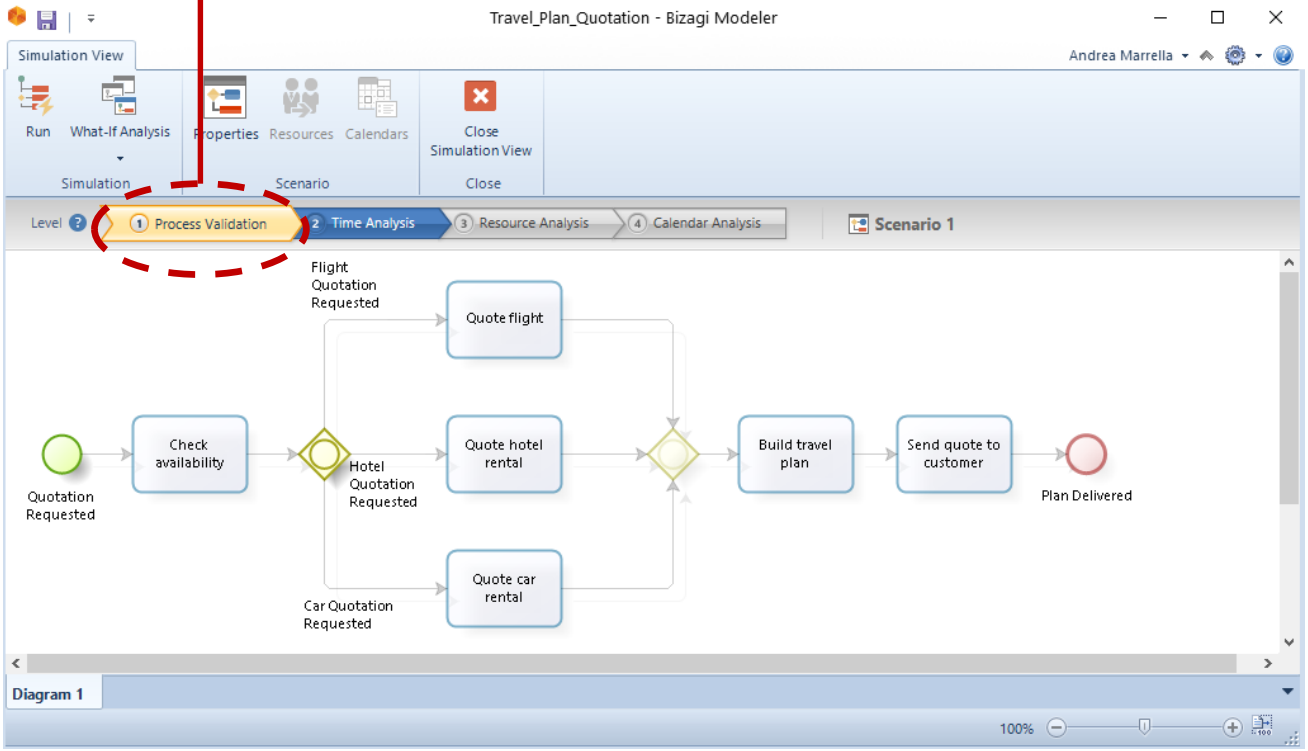
Simulation in Bizagi Modeler

- Bizagi Modeler provides an internal tool for ***simulating*** the flow of tokens inside a process under different configurations.
- It can be used to:
  - evaluate the performance of a model over long periods of real time;
  - reduce the chances of failure to meet specifications;
  - prevent under or over-utilization of resources (including people and money);
  - **check** the **structural** and **behavioural** correctness of the process.

Simulating the flow of tokens \1

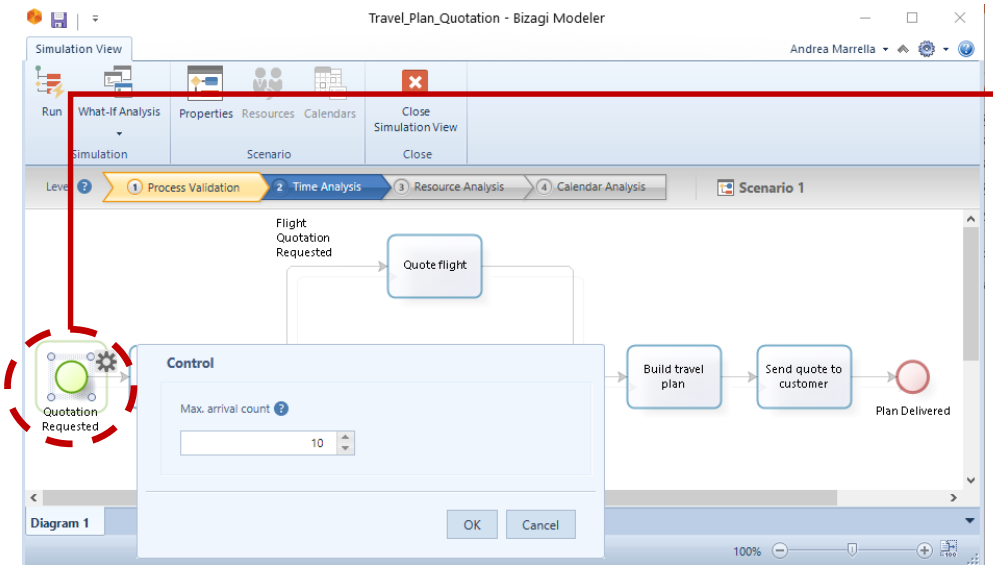


1. click the **Simulation View** button on the ribbon



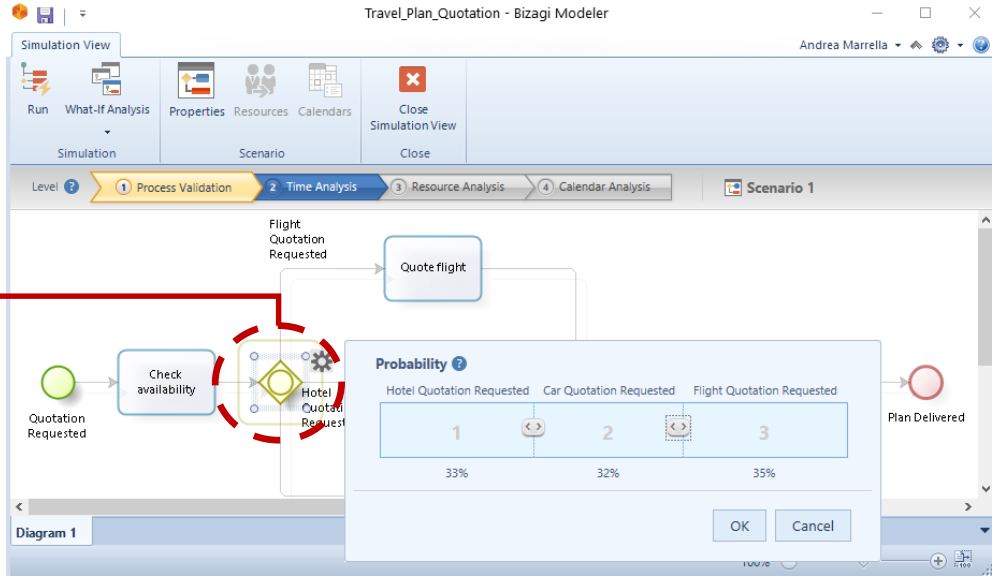
2. Make sure that the **Process Validation** shape is selected and highlighted

Simulating the flow of tokens \2

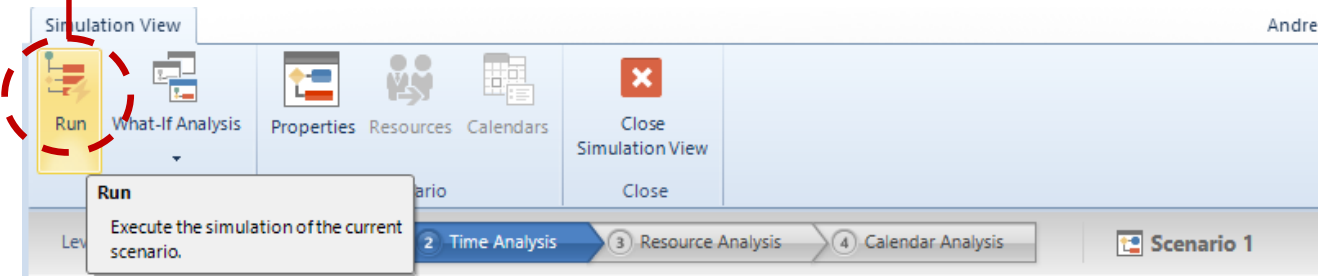


3. Select the Start Event of the process and click the ***Gear icon*** on the pie menu. It allows to define the number of token instances (***max arrival count***) the process will generate. The simulation will complete when *max arrival count* is reached.

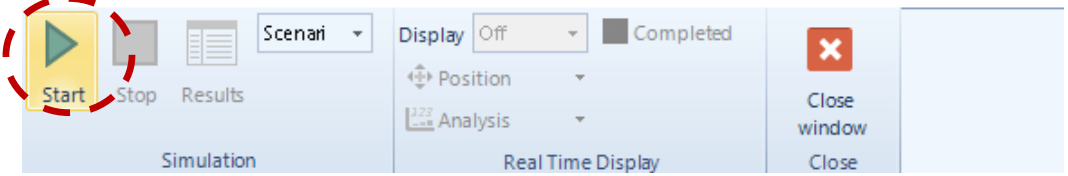
4. Inclusive and exclusive gateways have ***activation probabilities***. Probabilities are values between 0 and 100%. If you do not define probabilities for the paths, they will be equally distributed.



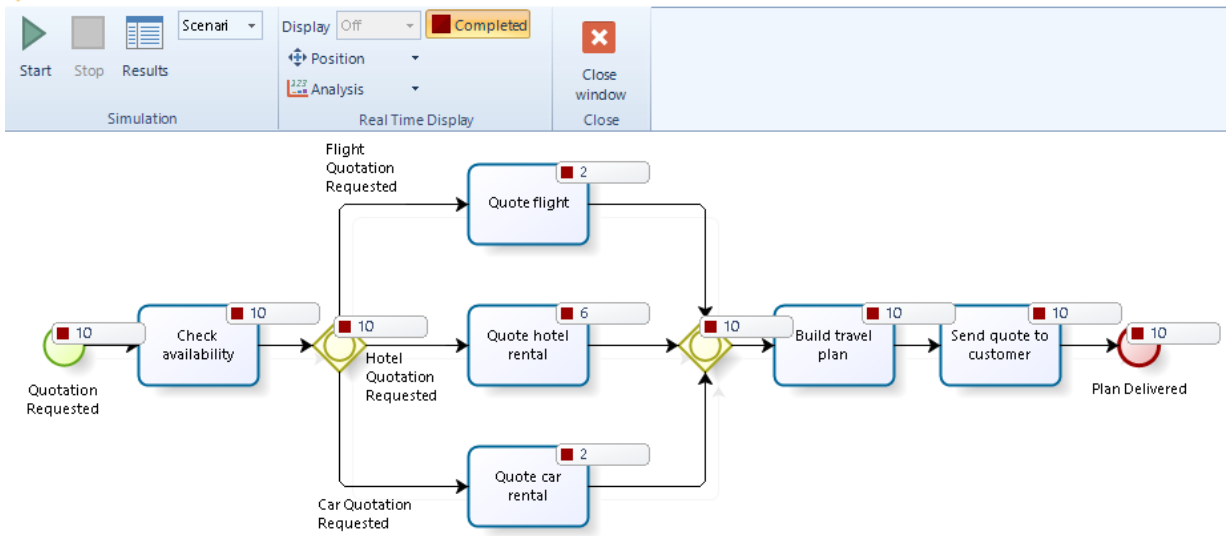
Simulating the flow of tokens \3



5. Click the ***Run*** button to open the simulation view.



6. Click the ***Start*** button to run the simulation.



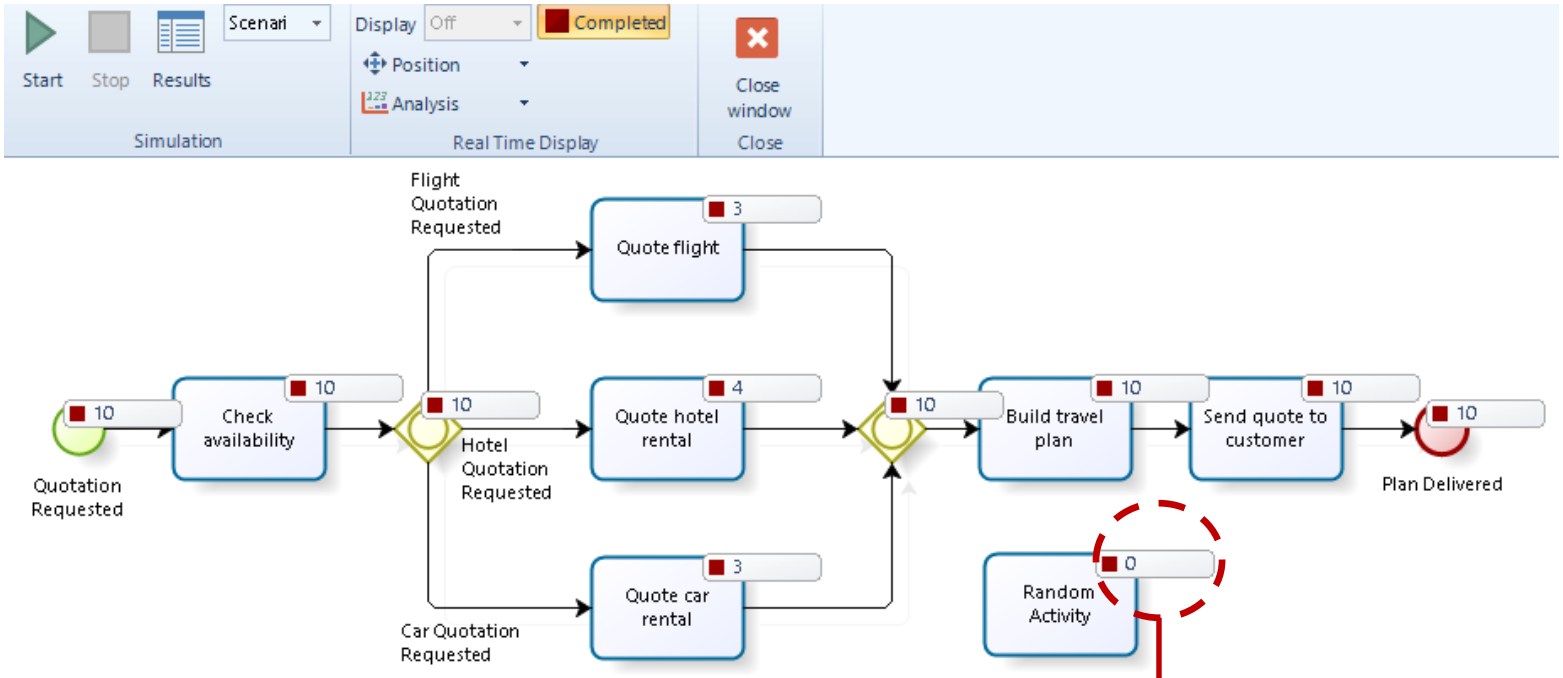
When running a simulation, the following data will display:

- Number of completed tokens.
- Number of token instances created.
- Number of instances that activate each shape.
- Number of finished instances.

Checking structural correctness through simulation in Bizagi Modeler

- Start events must not have incoming arcs, end events must not have outgoing arcs.
  - **Not allowed by the graphical editor of Bizagi Modeler**!
- Gateways must have exactly one incoming and at least two outgoing arcs (splits) or at least two incoming and exactly one outgoing arcs (joins).
  - **In this case, the Simulation tool of Bizagi Modeler will not be launched and an appropriate feedback is returned to the user**!
- All nodes are on a path from a start to an end event (i.e. no dangling arcs or disconnected nodes).
- Activities must have at least one incoming and one outgoing sequence flow.
  - **In this case, the Simulation tool of Bizagi Modeler will be able to simulate the flow of tokens inside the process, but "isolated" flow object.**

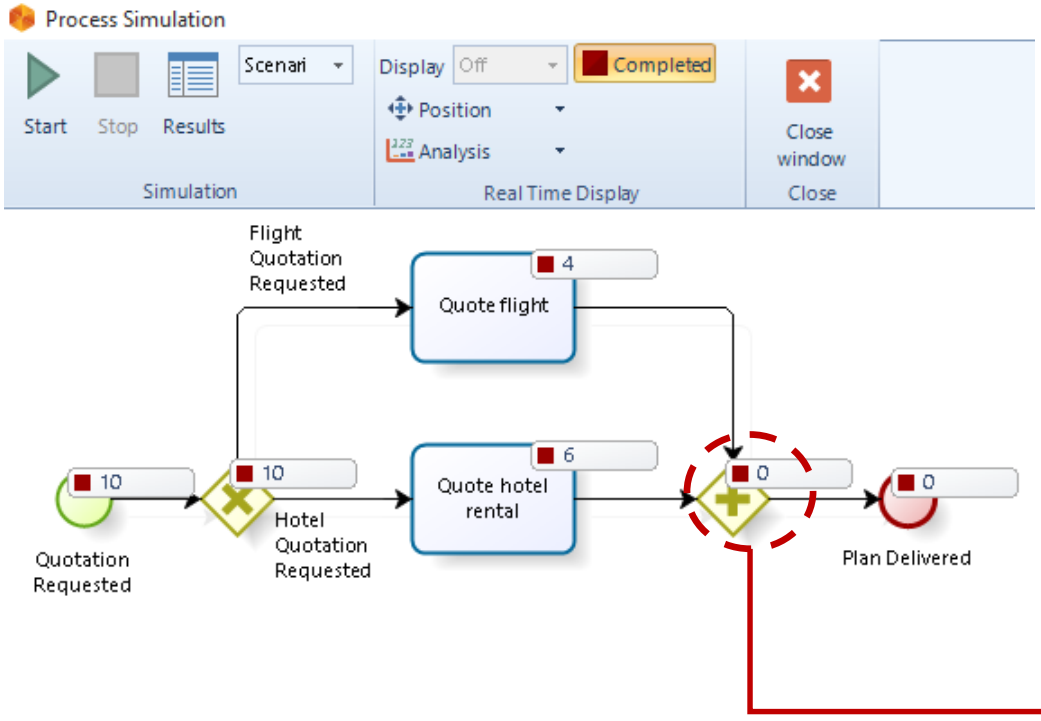
Checking structural correctness
Example



If a node is not on a path from a start to an end event, **no token** will flow into it.

Checking behavioural correctness through simulation in Bizagi Modeler

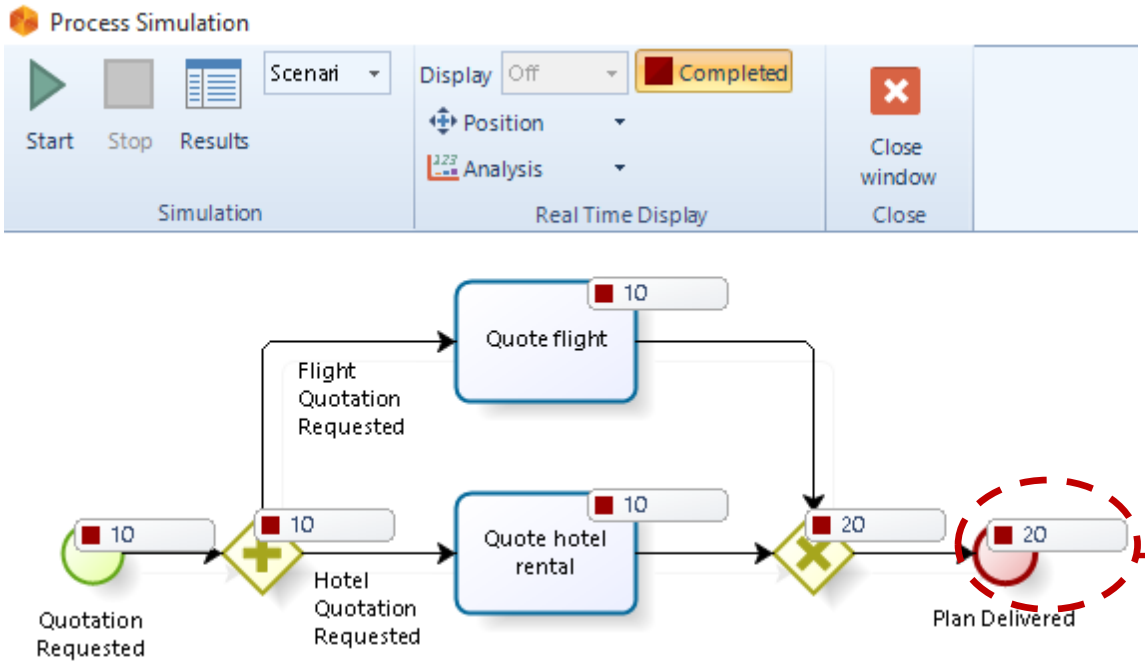
- **option to complete**: any running process instance must eventually complete, i.e. there are no *deadlocks* or *livelocks*



Deadlock - **No token** flows into the parallel join.

Checking behavioural correctness through simulation in Bizagi Modeler

- **proper completion**: at the moment of completion, each token of the process instance should be in a different end event. Therefore, it should not be any other task still running when the process completes.



The number of process instances completed is greater than the number of tokens generated in the start event.

Checking behavioural correctness through simulation in Bizagi Modeler

- **no dead activities**: any activity can be executed in at least one process instance