

Lab 11: Process Instantiation

Important notes:

- The key point here is what are the start events that are required to execute the business process? They could be simply singular event, or multiple events combined with \wedge , combined with XOR, combined with \vee , or even combinations of these.
 - In the latter case, need to check where is the last connection point that is just before execution of process instance, no know which is necessary and which one could be not-present.
 - Both XOR and \vee are dealt similarly, since any of the incoming branches can activate the gateway to go on for next stage, while \wedge gateway must wait for all branches to be fulfilled to accomplish its logic.
- It is given to you a sequence of arrival events e_j . It is required to check for process instance factory with respect to them.
 - The number between brackets is considered collaboration ID of process instance; i.e. something like ID of different instances
 - Example: this ID can represent different clients of the same company (i.e. following same business process but with different information).
- Process Instance Factory is creating Process instance with the collaboration ID mentioned.
 - If this process instance is already created once, it is ignored in later mentions.
 - After creating the process instance, they begin to check for other events required to activate the business process. All required events are being subscribed in this step.
 - While going on with the time t_i , any of the required events are mentioned –regarding the same process instance- they are being unsubscribed (i.e. not waiting for them any more) and check for next step.
 - Once all required events are now collected, the process can be activated for execution.
 - If further events appeared for the same process instance, they are simply not used. “Ignore” state here is useless. They are just skipped without mentioning.

Example 1:

Consider that the process model in Figure 1 is deployed to an execution engine where “Delivery notification received”, “Delivery planning completed” and “Shipment is relevant for freight” are three types of events. We symbolize them with E1; E2; E3 respectively.

Consider the following sequence arrival of events.

t1 : e3(1000)
 t2 : e1(1001)
 t3 : e1(1000)
 t4 : e2(1001)
 t5 : e3(1001)

Show the actions of the process instance factory as well as the process instances in response for the events occurrence above.

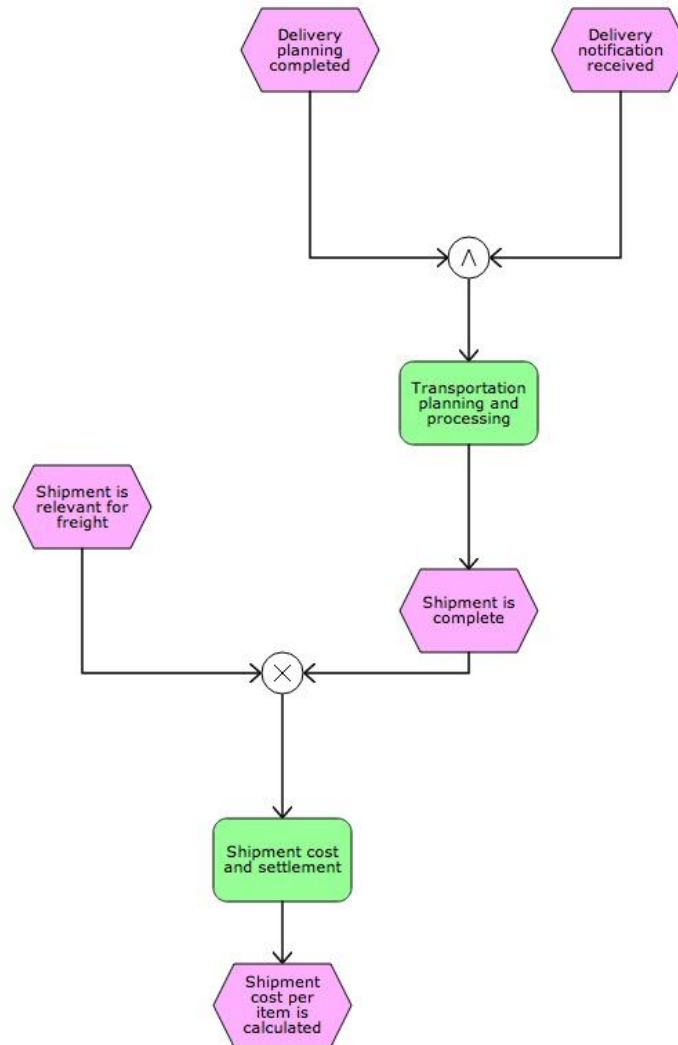


Figure 1

Solution:

- At time t1 the process instance factory creates a new process instance p_{1000} with an initial activation of $e3(1000)$.
 - At the same point in time p_{1000} turns into state running as it has all necessary input collected, because $E3$ is exclusive to $E1 \wedge E2$.
- At time t2 the process instance factory creates a new process instance p_{1001} with initial activation $e1(1001)$.
 - At the same time p_{1001} subscribes for event $E2$.
- At time t3 the event instance $e1(1000)$ is ignored by the process factory. (p_{1000} already created)
- At time t4 the event instance $e2(1001)$ is ignored by the process factory. (p_{1001} already created)
 - However, the process instance p_{1001} consumes the event and unsubscribes from $E2$ and turns into state running.
- At time t5 the event instance $e3(1001)$ is ignored by the process instance factory. (p_{1001} already created)
 - Ignore $e3$ completely, since it is not needed

Note that: you can use p_1 and p_2 instead of p_{1000} and p_{1001} they are both correct in meaning.

Example 2:

Consider that the process model in Figure 2 is deployed to an execution engine. Consider the following sequence arrival of events.

t1: e2(1000)
 t2: e3(1001)
 t3: e4(1000)
 t4: e5(1002)
 t5: e5(1001)
 t6: e1(1002)
 t7: e1(1000)
 t8: e3(1000)
 t9: e3(1002)
 t10: e2(1001)
 t11: e4(1001)
 t12: e3(1001)

Show the actions of the process instance factory as well as the process instances in response for the events occurrence above.

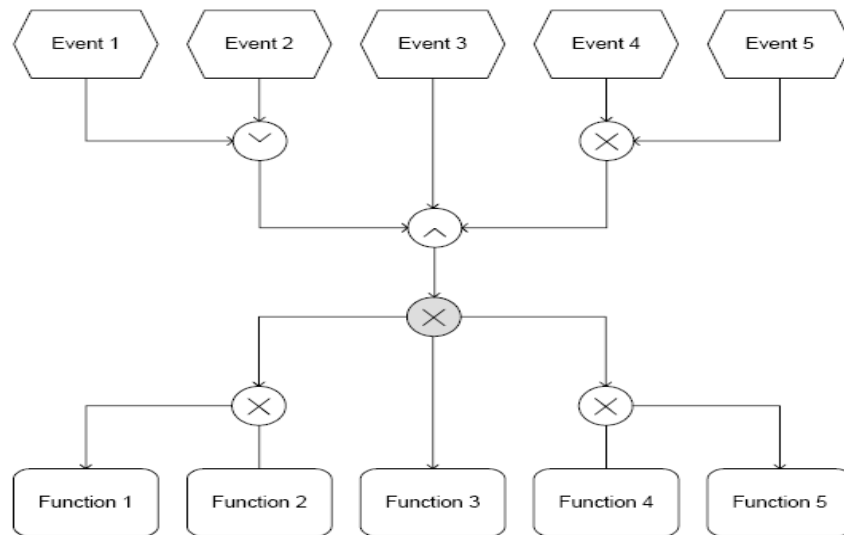


Figure 2

Solution: Process factory

- At t1: e2(1000) → Factory creates p1000, p1000 subscribes in e3, e4, e5
- At t2: e3(1001) → Factory creates p1001, p1001 subscribes in e1, e2, e4, e5
- At t3: e4(1000) → Factory ignores creation of p1000, p1000 unsubscribes from e4, e5 (still waiting for e3)
- At t4: e5(1002) → Factory creates p1002, p1002 subscribes in e1, e2, e3
- At t5: e5(1001) → Factory ignores creation of p1001, p1001 unsubscribes from e4, e5 (still waiting for (e1 OR e2))
- At t6: e1(1002) → Factory ignores creation of p1002, p1002 unsubscribes from e1, e2, (still waiting for e3)
- At t7: e1(1000) → Factory ignores creation of p1000, (still waiting for e3)
- At t8: e3(1000) → Factory ignores creation of p1000, p1000 unsubscribes from e3, business process is **activated** [process instance **p1000**]
- At t9: e3(1002) → Factory ignores creation of p1002, p1002 unsubscribes from e3, business process is **activated** [process instance **p1002**]
- At t10: e2(1001) → Factory ignores creation of p1001, p1001 unsubscribes from e1, e2, business process is **activated** [process instance **p1001**]
- At t11: e4(1001) → Factory ignores creation of p1001.
- At t12: e3(1001) → Factory ignores creation of p1001.