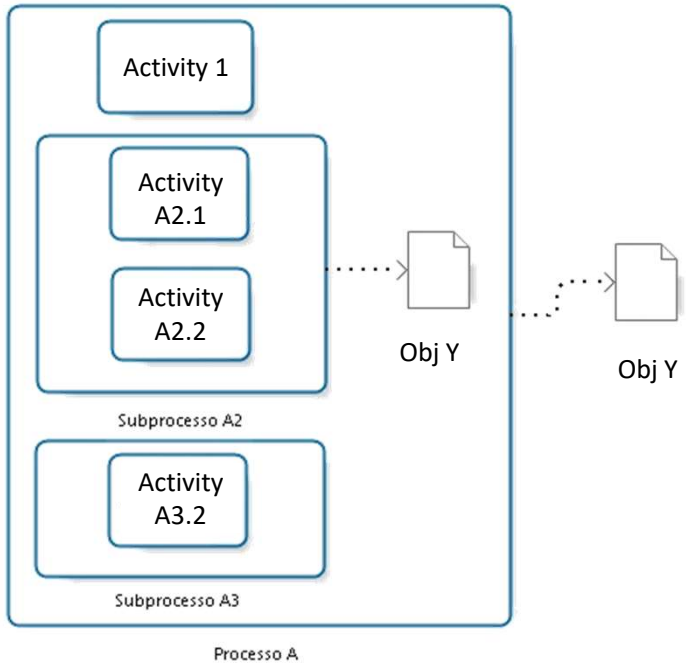
Advanced Concepts of BPMN 2.0

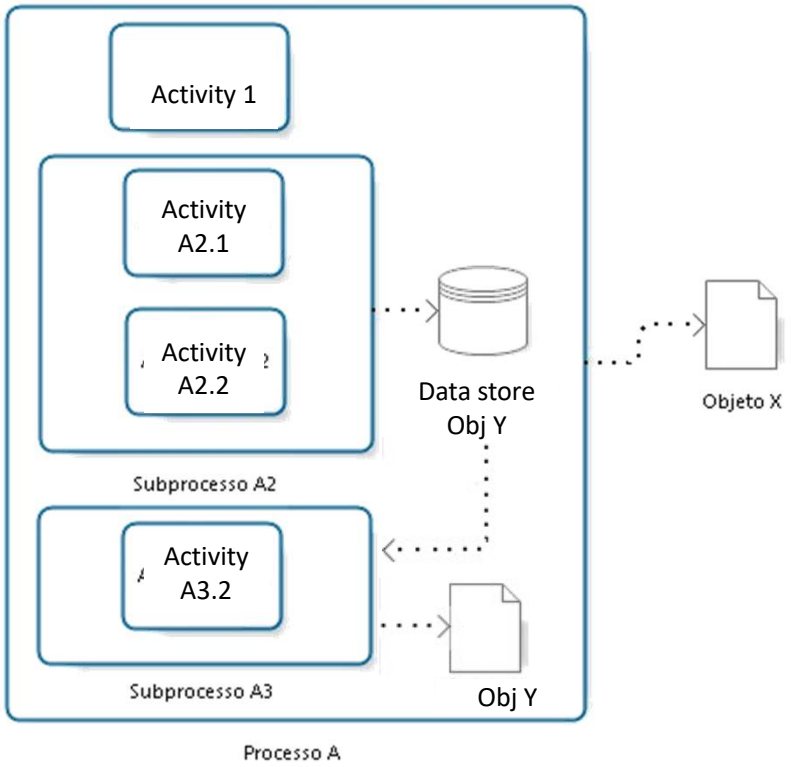
Visibility of *DATA OBJECT*

- Objects are temporary forms of data, their scope of visibility depends on their point of creation.



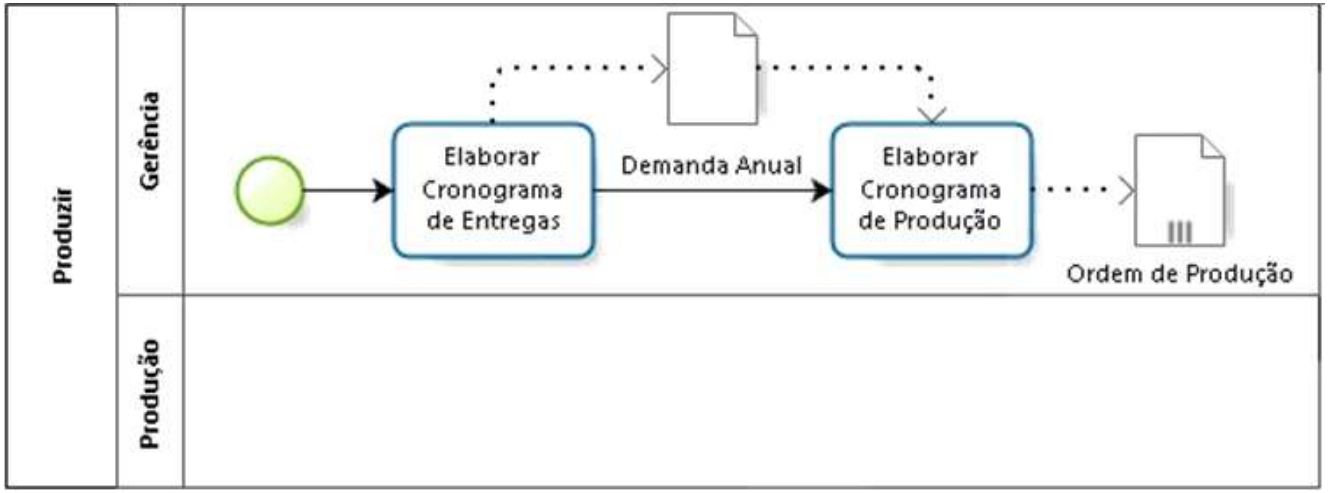
DATA STORE

- They are persistent data, which once created, continue to exist during the process and can be accessed from any other process, subprocess or activity



Data Collection

- It is represented by an internal identifier with three vertical dashes, means a set of data of the same type

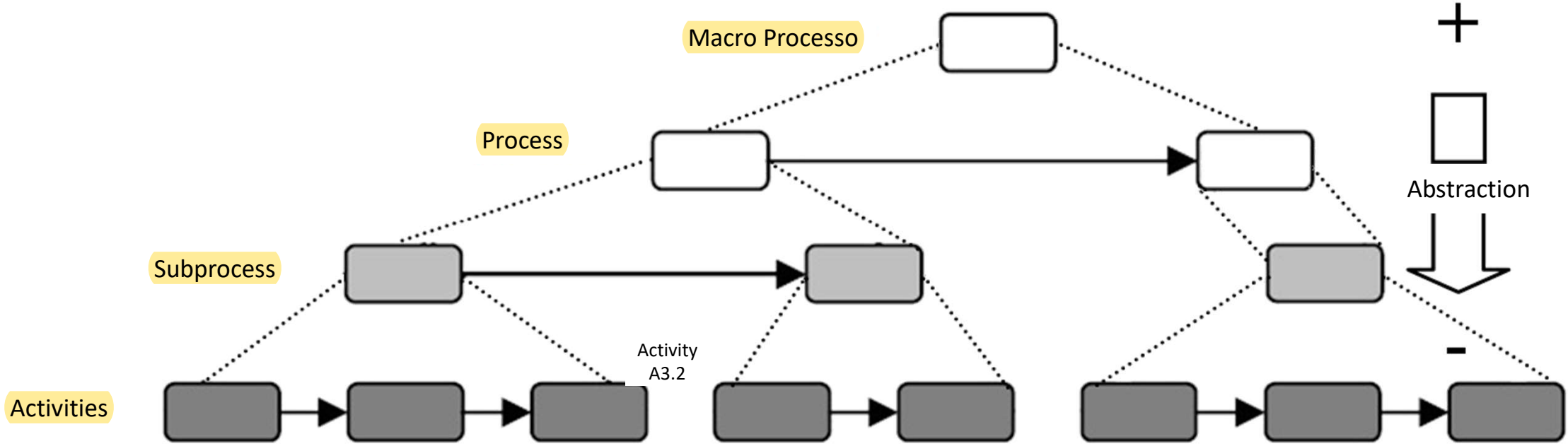


Functional Decomposition

- It's easier to solve a complex problem when you break it down into manageable parts
  - This principle addresses human limitations in dealing with complexity, allowing the designer to focus on one subject or feature at a time.

Abstraction levels

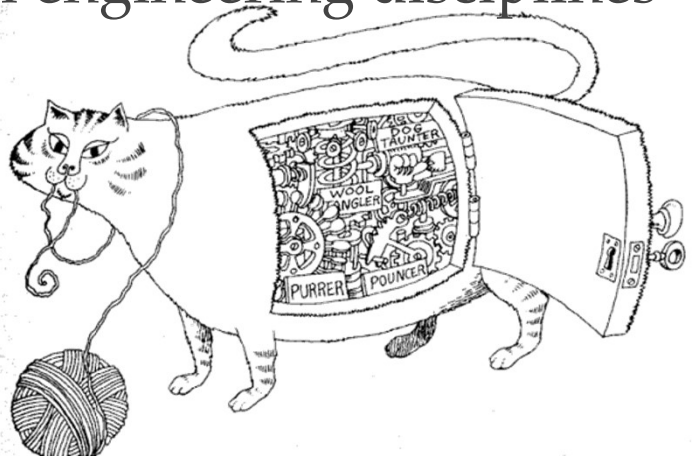
- When considering a modular solution to any problem, many levels of abstraction can be raised.



Modularization

- It is the process of dividing a whole into well-defined parts, called modules, that can be constructed and examined separately and interact in a well-defined way.

- Modularization has become an accepted approach in all engineering disciplines

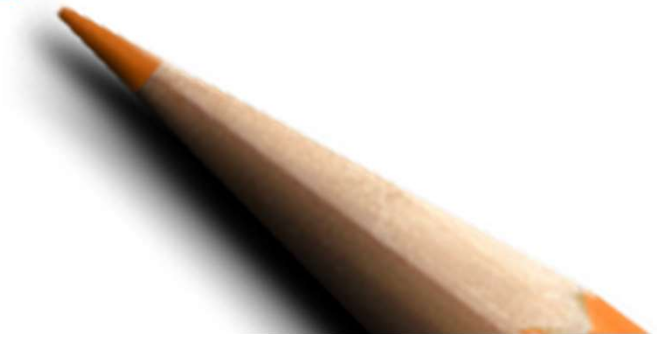


MODULES

- They are essential functional units, self-contained in relation to the whole of which it is part, having standardized interfaces that allow, by means of combinations the composition of the whole.

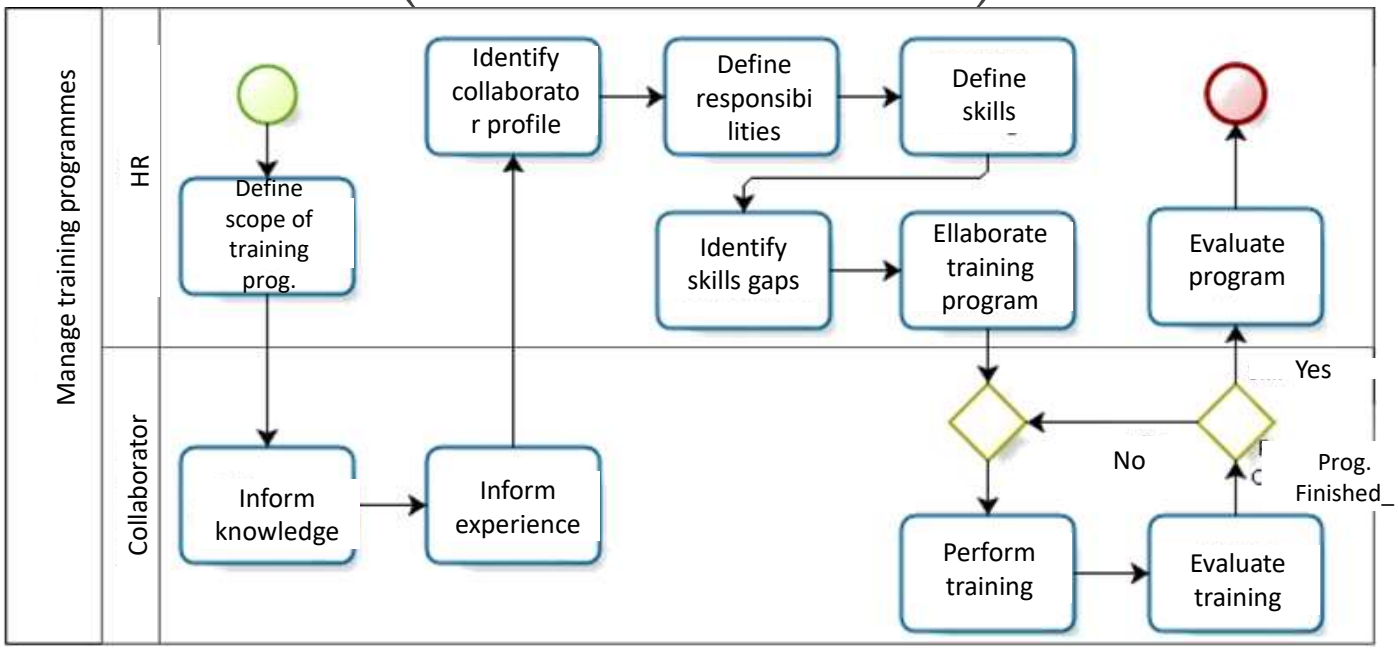
Independent Modules

- Functional independence is achieved by the development of modules (processes) that have a "single purpose" and an "aversion" to excessive interaction with other modules (processes)
- Independence is measured using two qualitative criteria: cohesion and coupling.



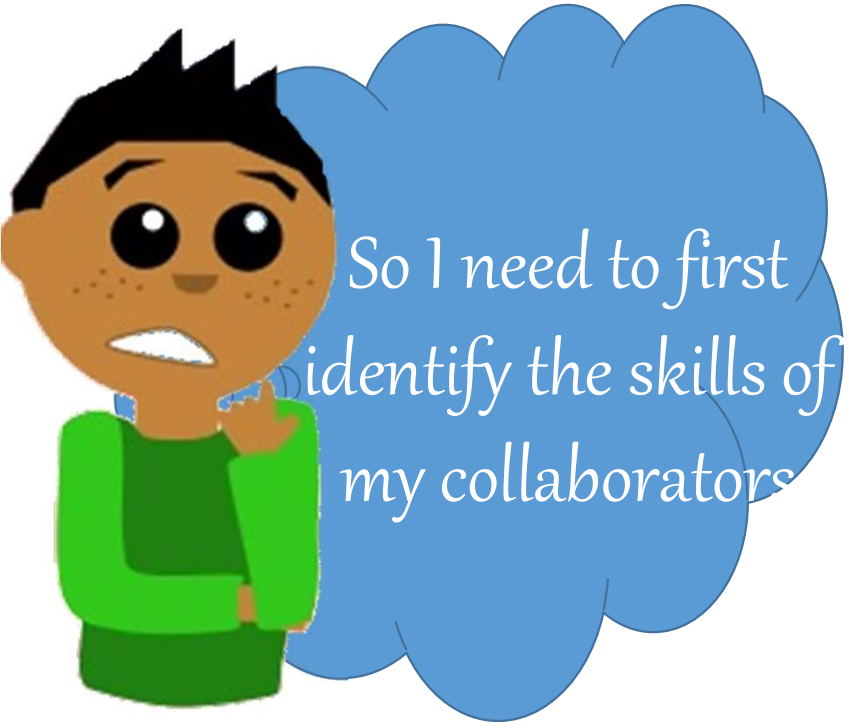
Example

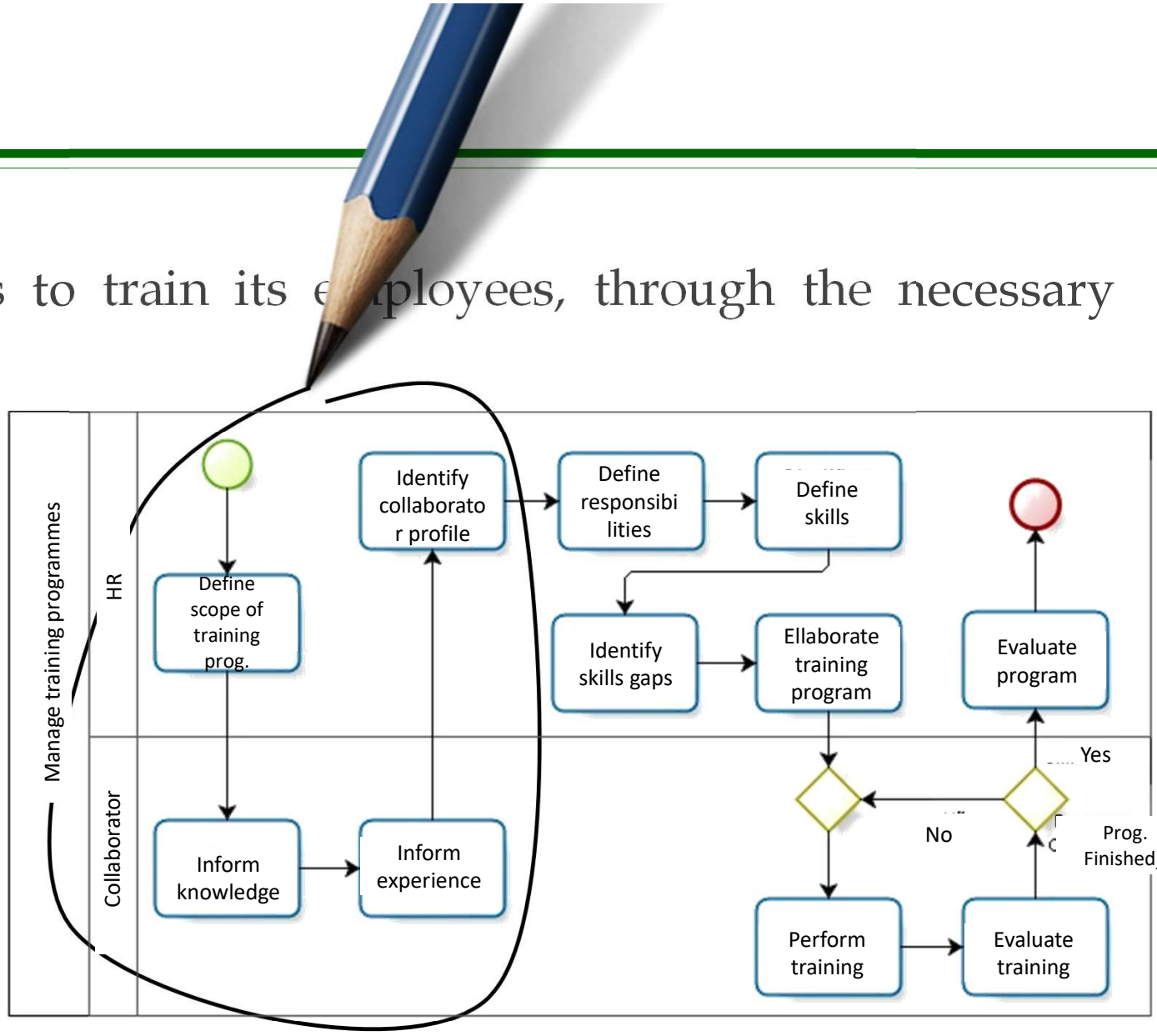
- The natural tendency of most designers, especially the less experienced, is to build models with many elements, which makes them difficult to read (maintain and reuse).



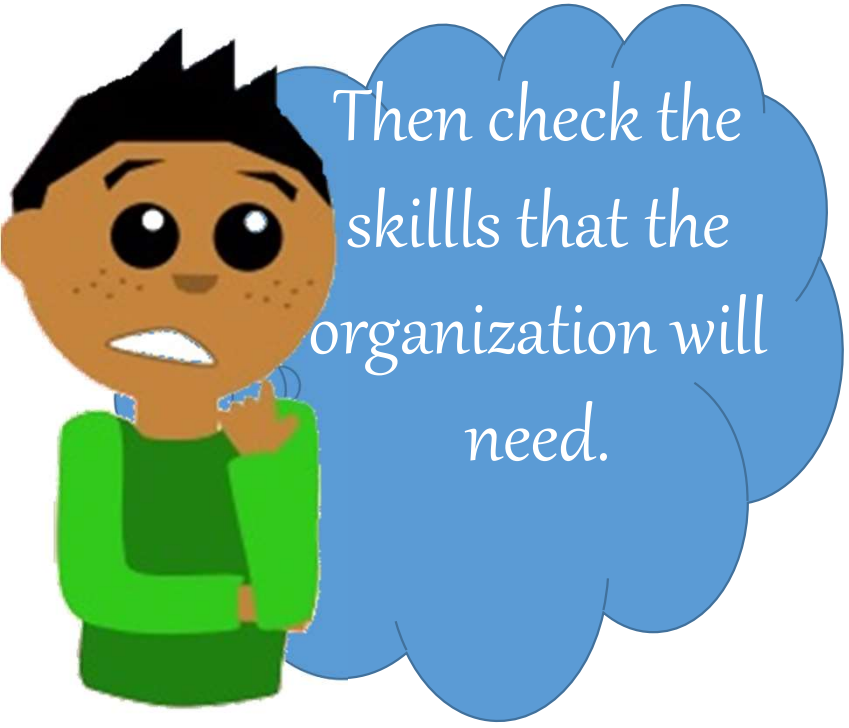
Example

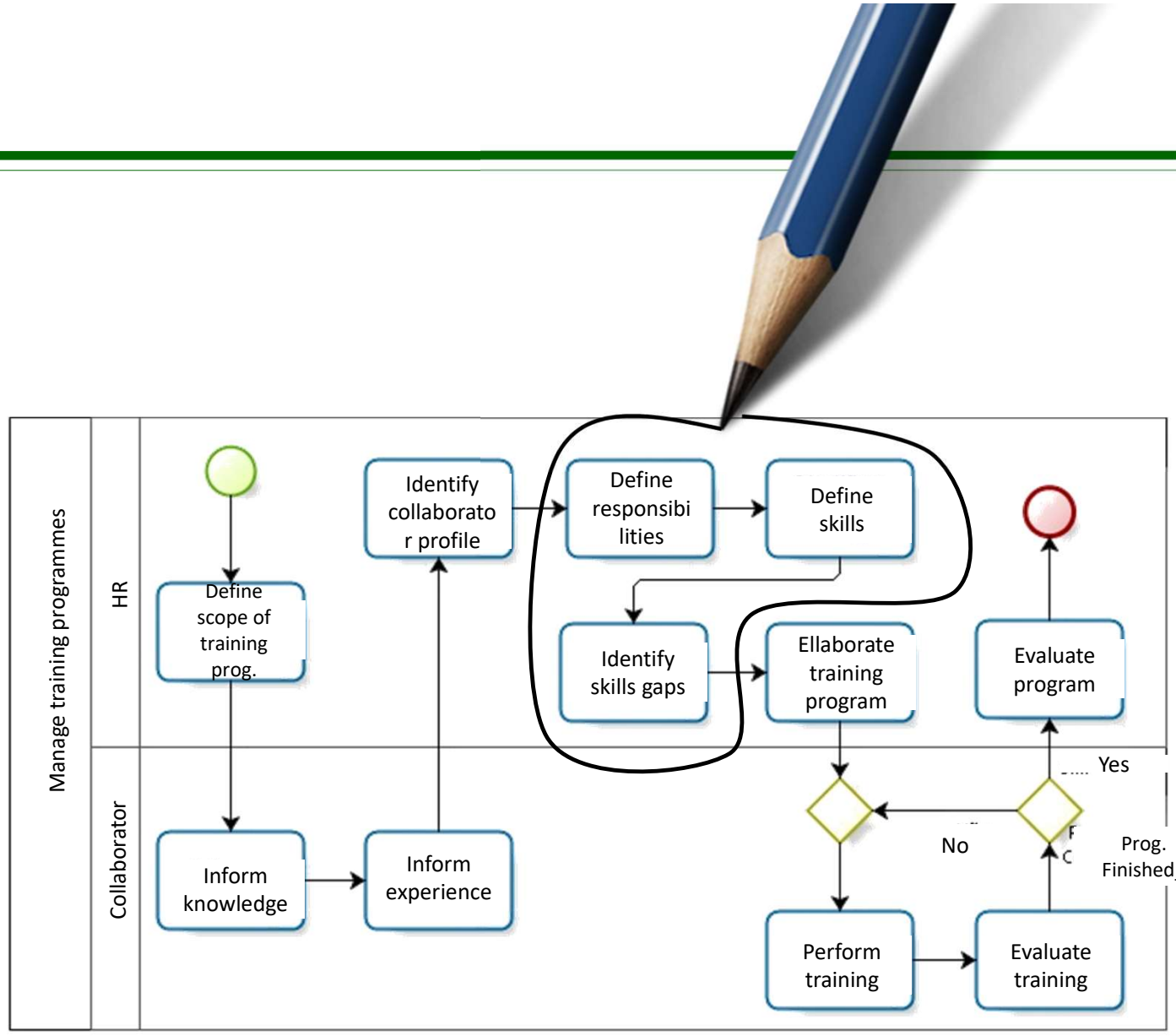
- The organization wants to train its employees, through the necessary training.





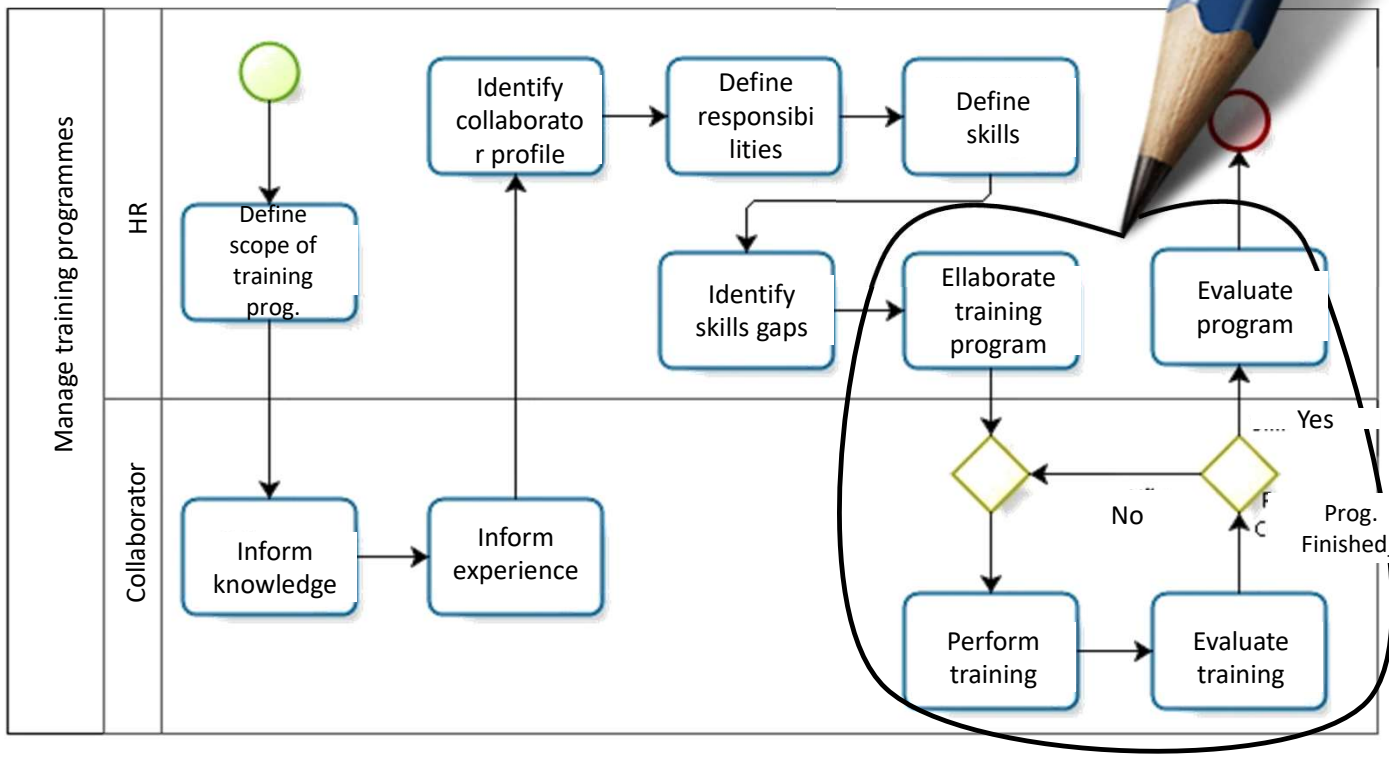
Example


Then check the skillls that the organization will need.

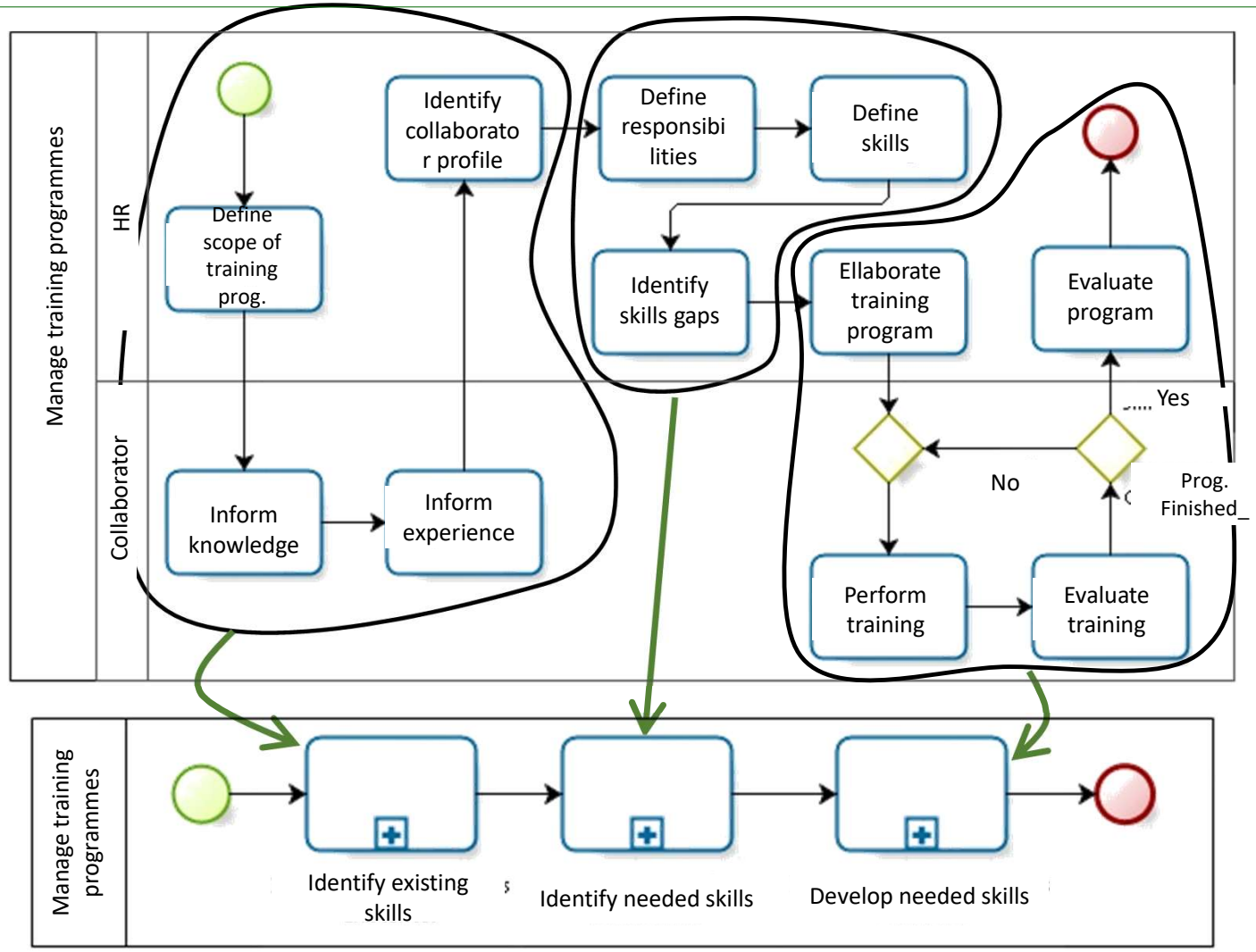

Manage training programmes
HR
Collaborator
Define scope of training prog.
Identify collaborato r profile
Define responsibi lities
Define skills
Identify skills gaps
Ellaborate training program
Evaluate program
Yes
No
Prog. Finished
Inform knowledge
Inform experience
Perform training
Evaluate training

Example


Now I need to compare with existing ones

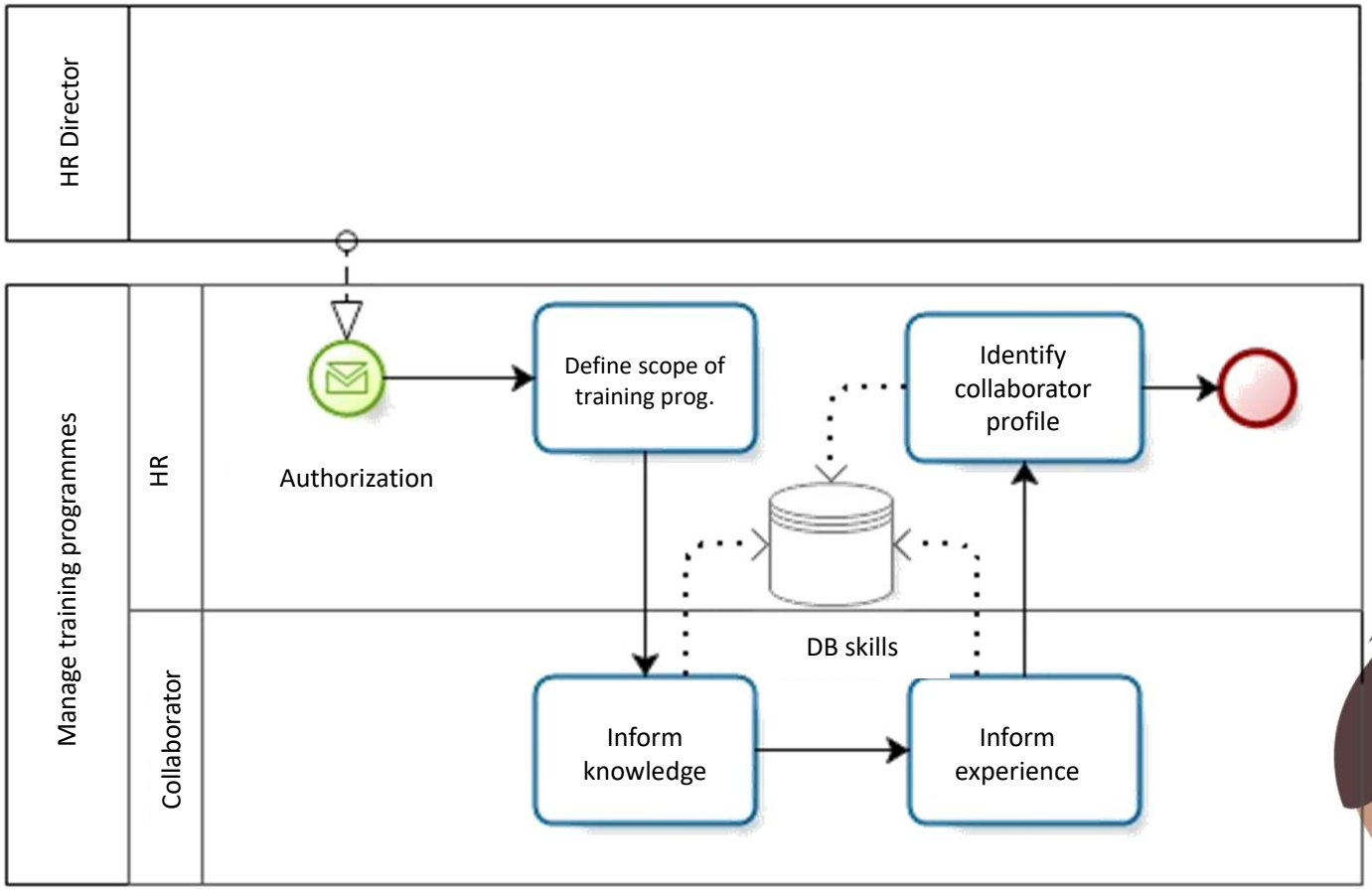

Manage training programmes
HR
Collaborator
Define scope of training prog.
Identify collaborator profile
Define responsibilities
Define skills
Identify skills gaps
Ellaborate training program
Evaluate program
Inform knowledge
Inform experience
Perform training
Evaluate training
Yes
No
Prog. Finished

So…


Manage training programmes
HR
Collaborator
Define scope of training prog.
Identify collaborato r profile
Define responsibi lities
Define skills
Identify skills gaps
Ellaborate training program
Evaluate program
Yes
No
Prog. Finished_
Inform knowledge
Inform experience
Perform training
Evaluate training
Manage training programmes
Identify existing skills
Identify needed skills
Develop needed skills

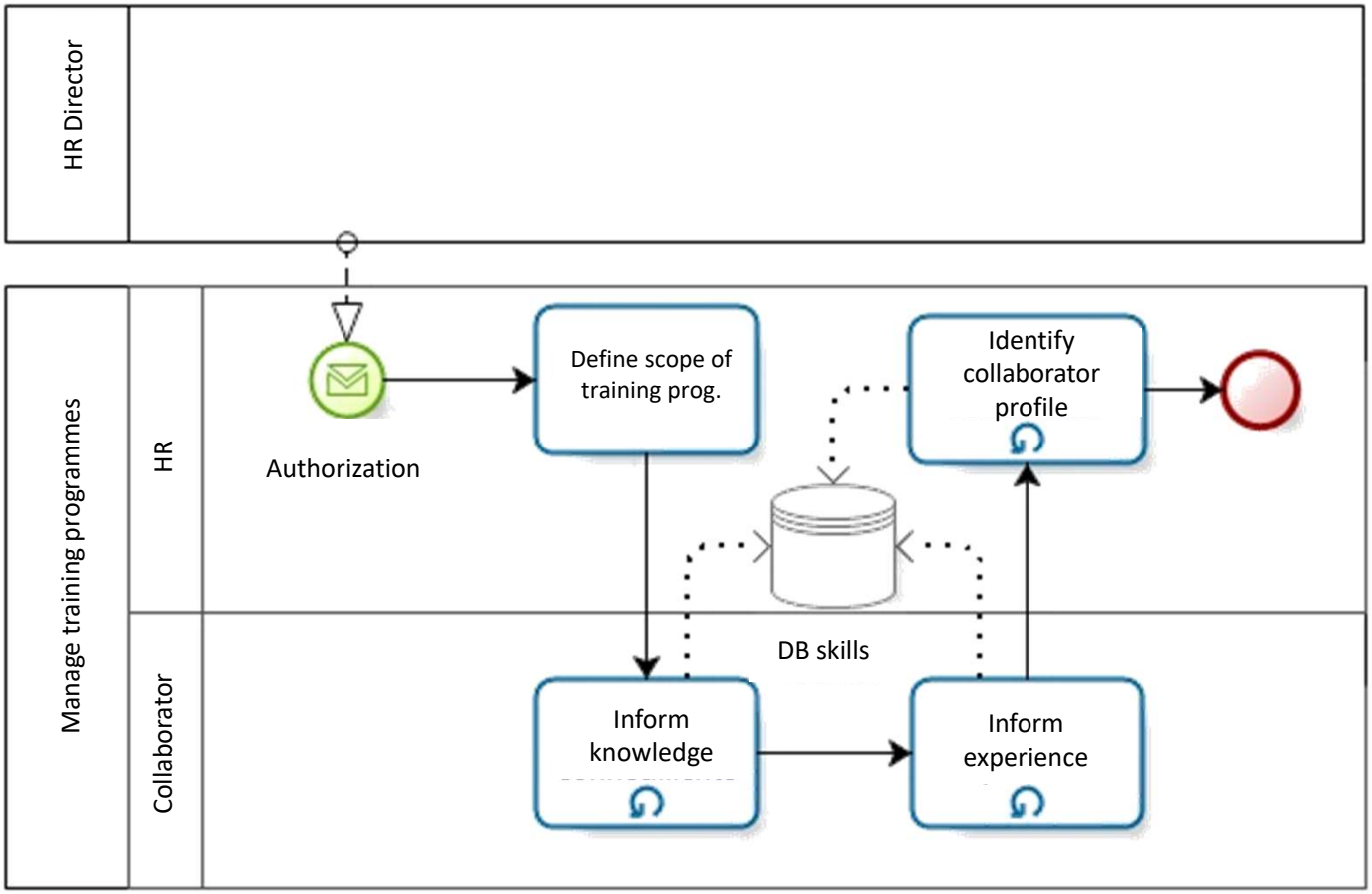
EXAMPLE

Identify existing skills



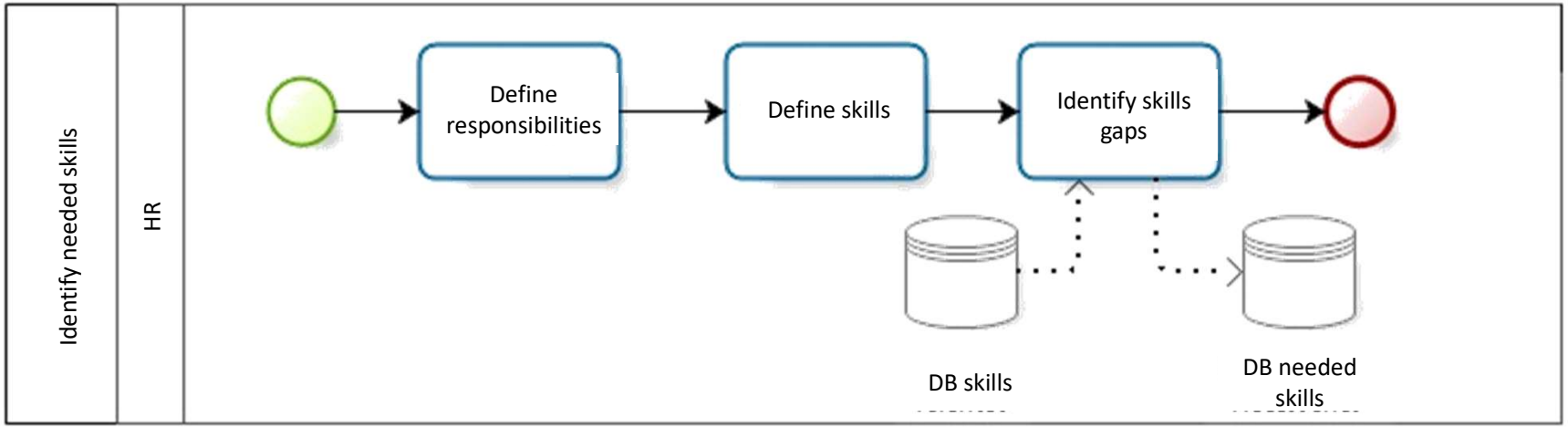
EXAMPLE

Identify existing skills

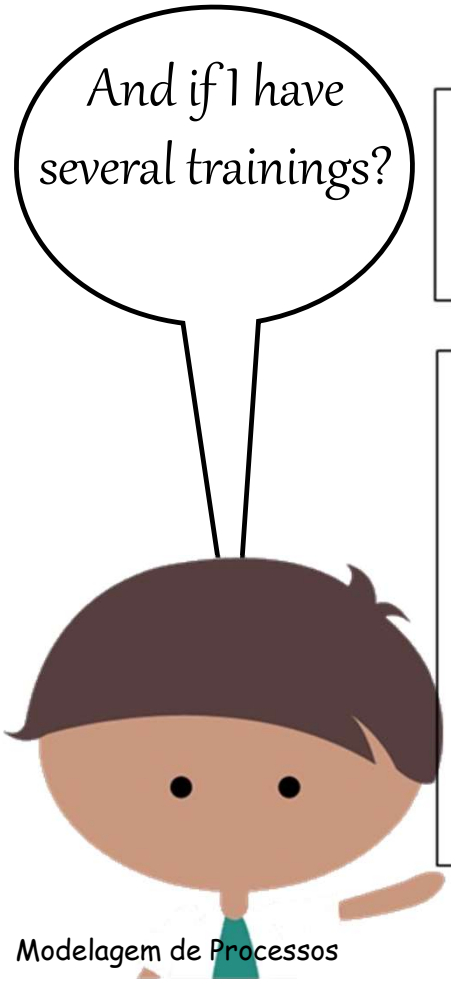


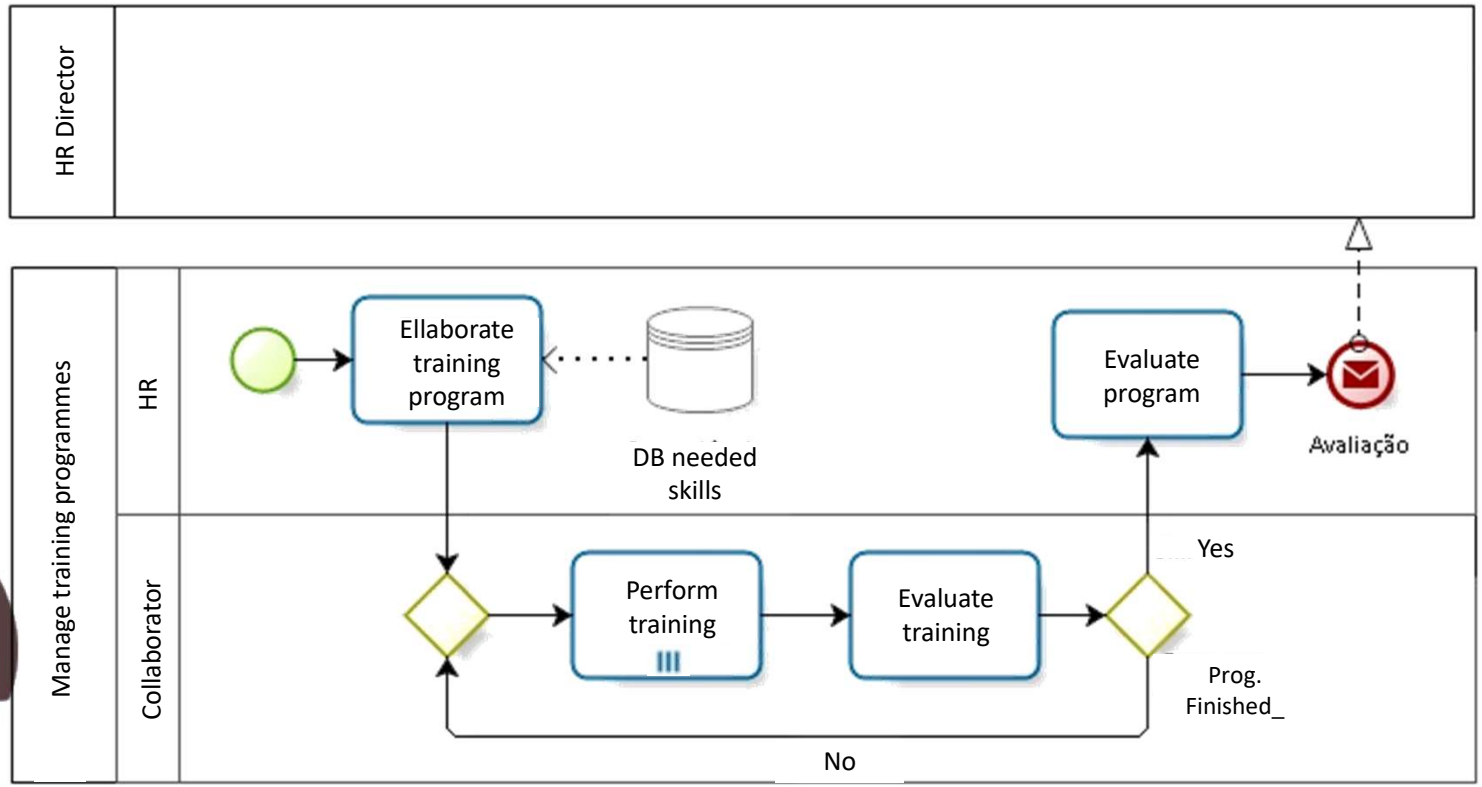
EXAMPLE

Identify needed skills



Example

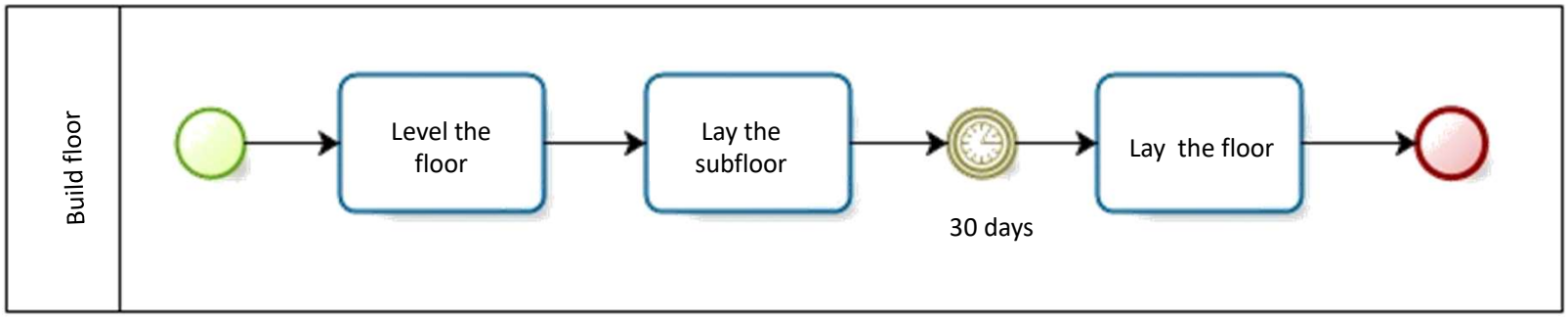

And if I have several trainings?
Modelagem de Processos


HR Director
Manage training programmes
HR
Collaborator
Ellaborate training program
DB needed skills
Evaluate program
Avaliação
Perform training
Evaluate training
Yes
Prog. Finished_
No


Develop missing skills

Events

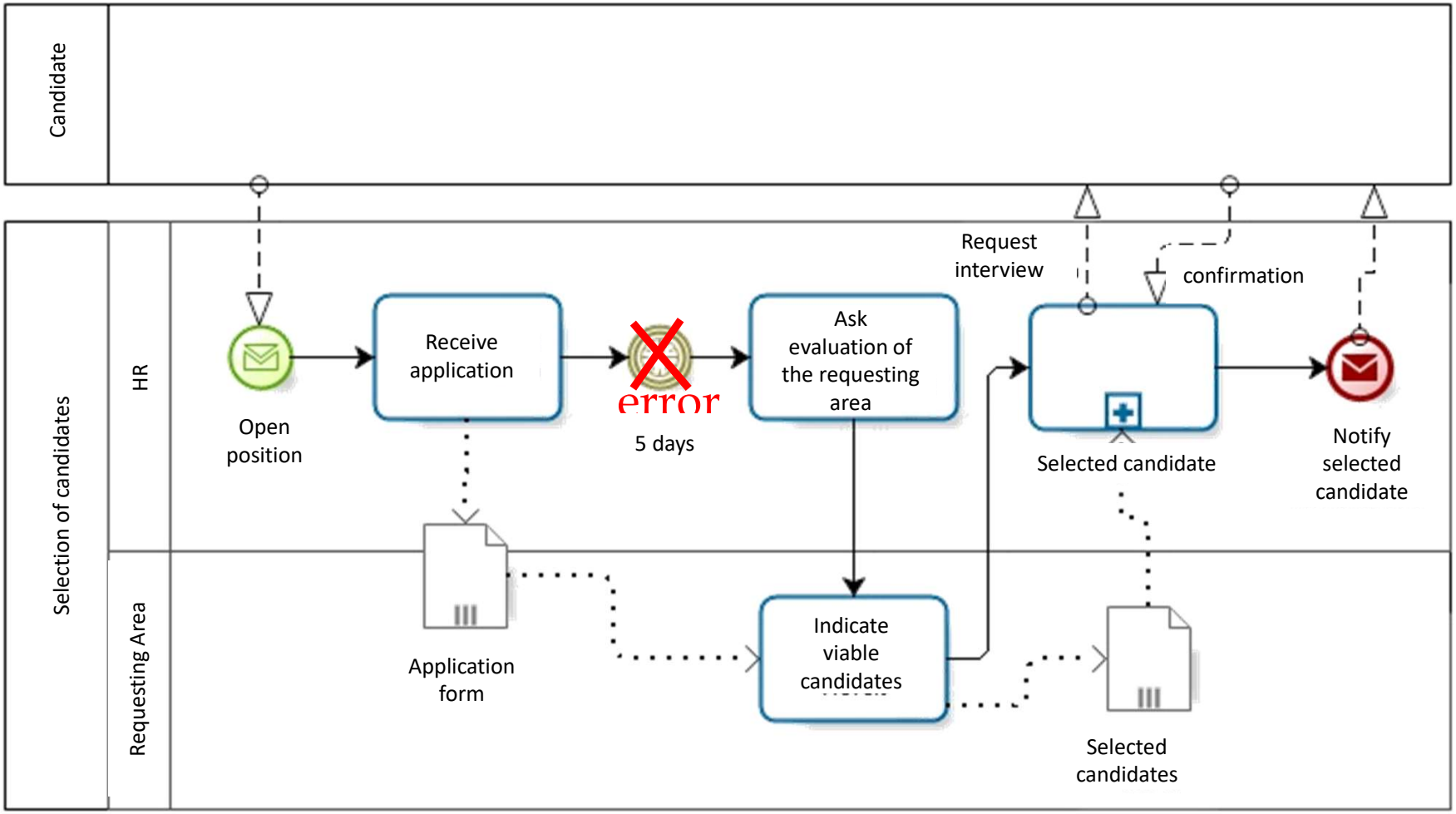
- Imagine a process that you need to determine the time between activities, such as a construction process.



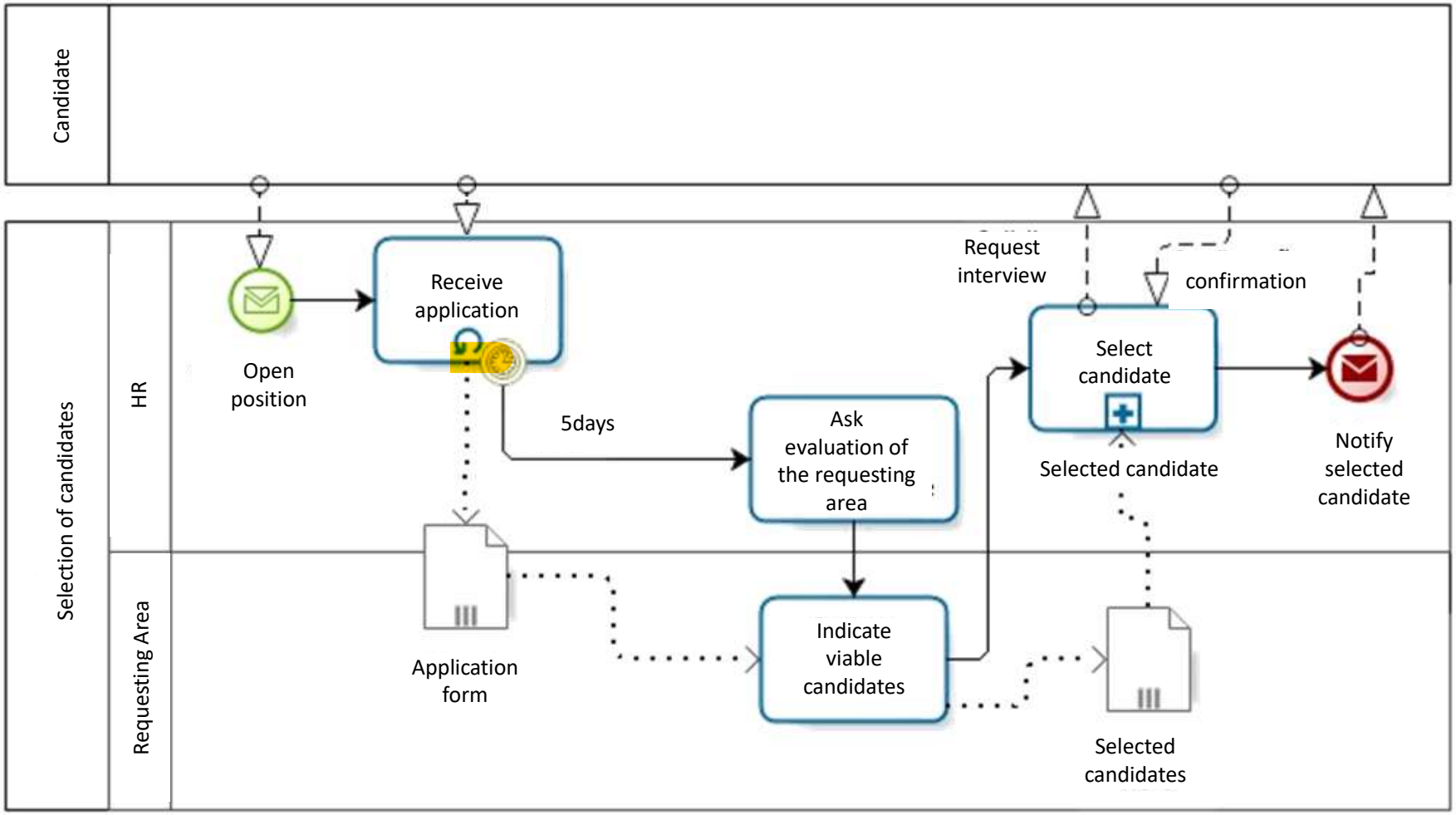
EVENTS

- Imagine another hypothesis: to determine the maximum time for performing an activity, not the time interval between activities.
  - For example, a process of selection of candidates in an HR department, which involves receiving the candidates' files no later than 5 days.

EVENTS

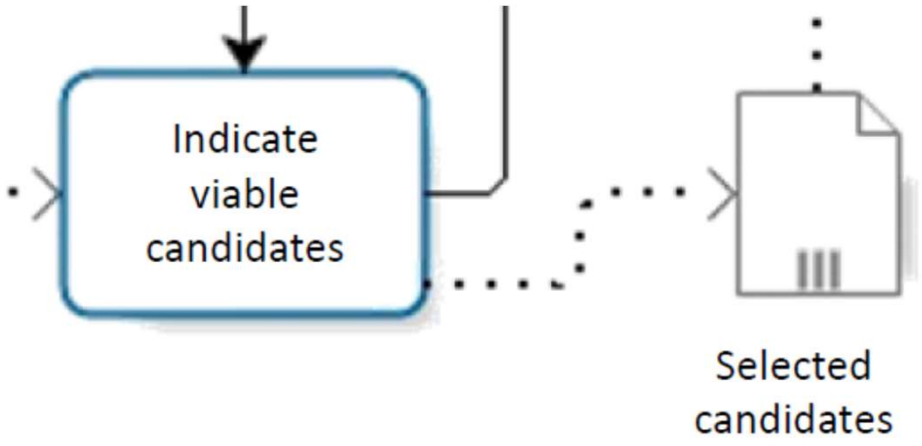

Candidate
Selection of candidates
HR
Requesting Area
Open position
Receive application
5 days
error
Ask evaluation of the requesting area
Request interview
confirmation
Selected candidate
Notify selected candidate
Application form
Indicate viable candidates
Selected candidates

Interruptive Events



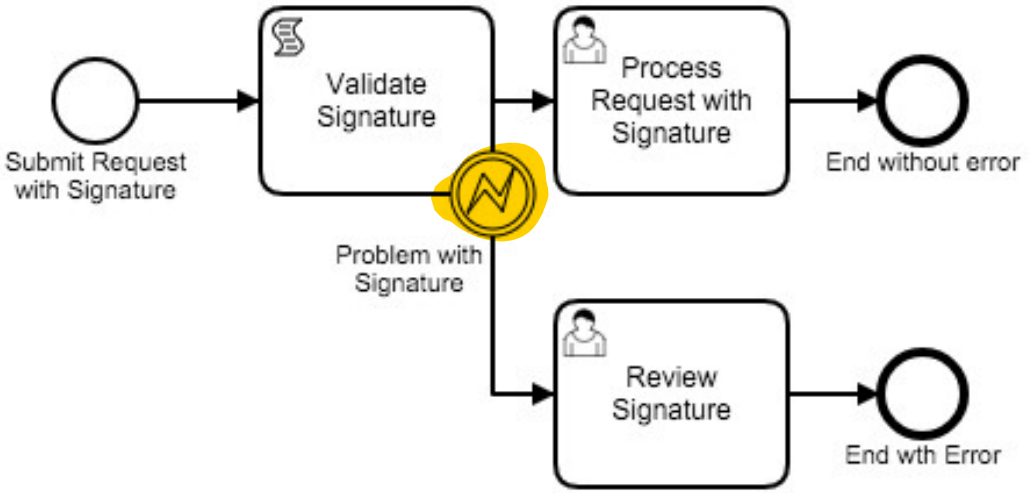
Data Collection

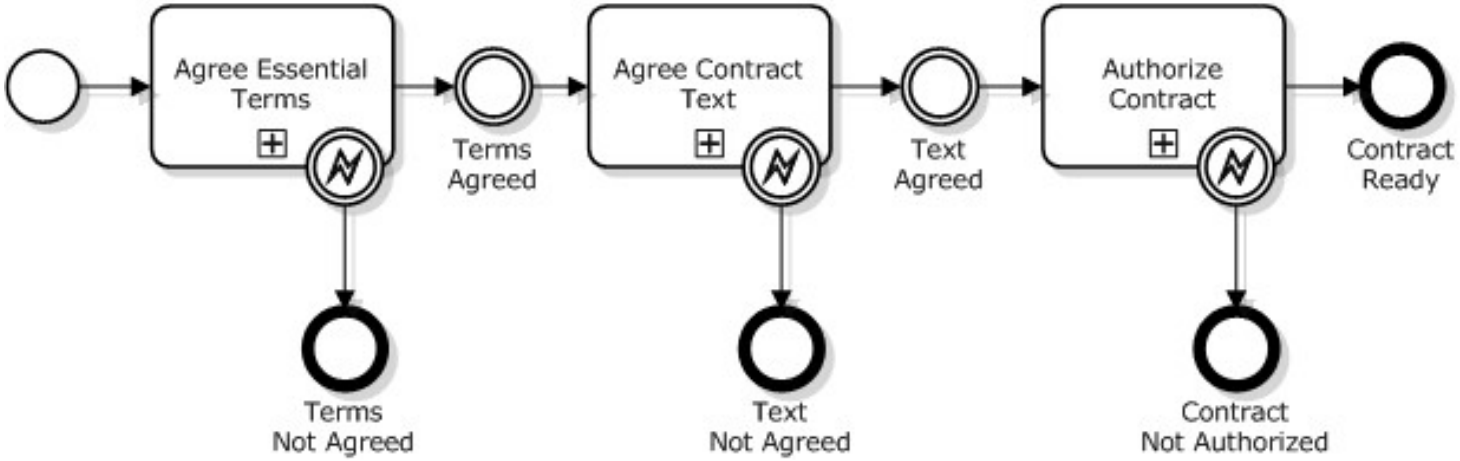
- It is represented by an internal identifier with three vertical dashes, means a set of data of the same type



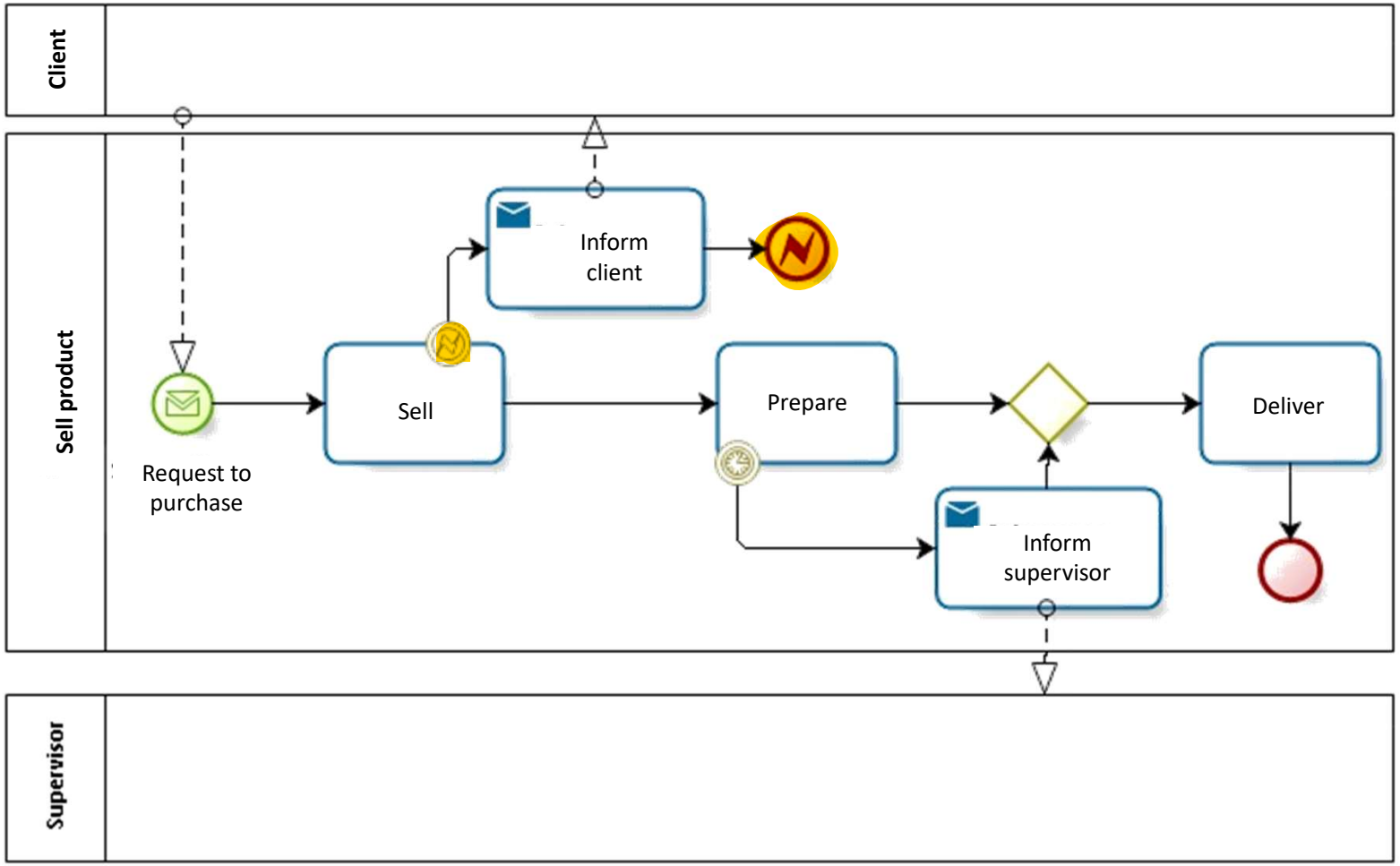
Error Event

- It deals with occurrences outside the normal or desired flow
  - So far, one way to handle these exceptions is through gateways added right after an activity, identifying something like “OK?" or “Not OK?"
  - But, the right thing is to identify them by an edge event that can be triggered at any time during the execution of the activity

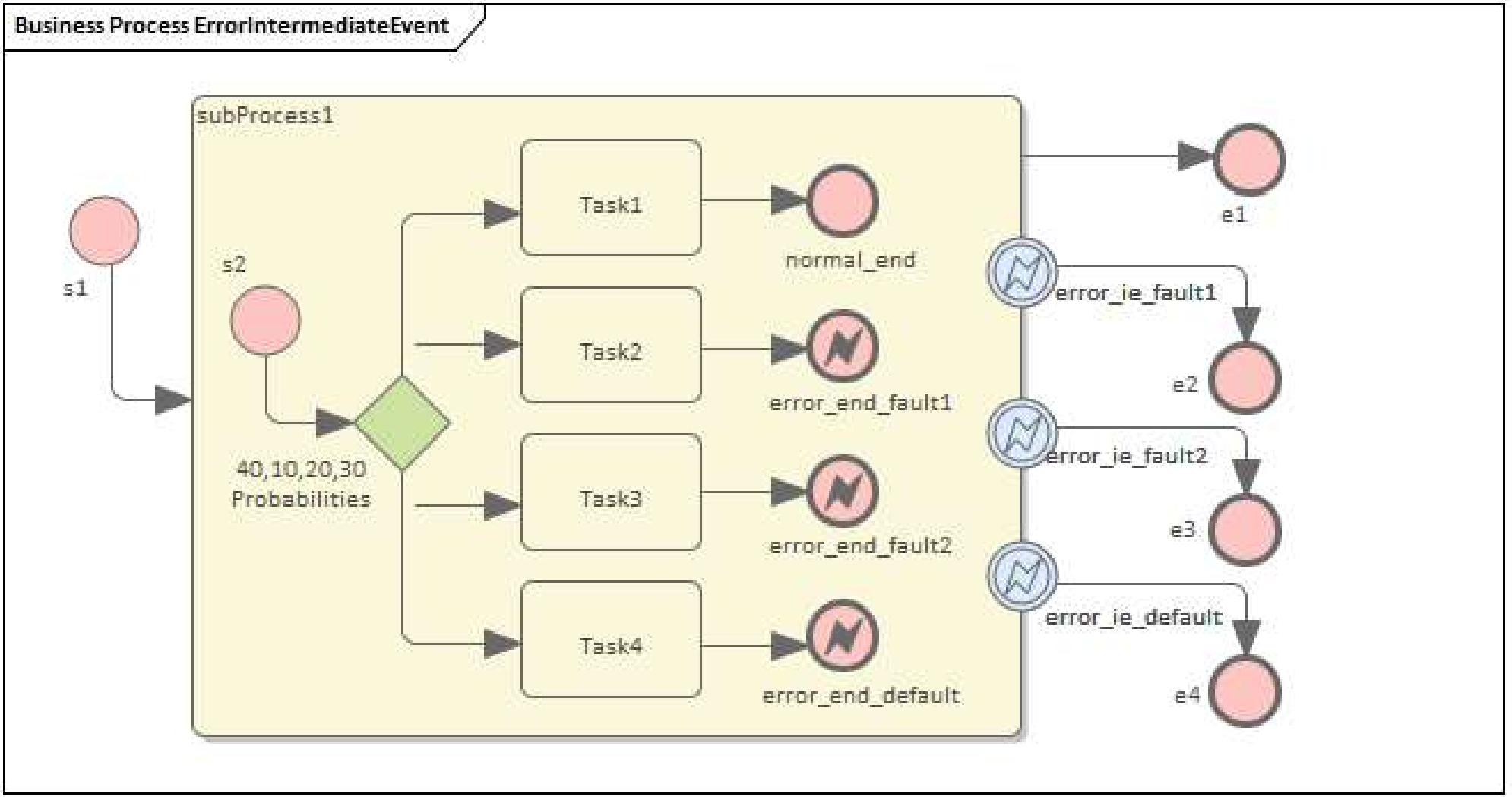

Submit Request with Signature
Validate Signature
Process Request with Signature
End without error
Problem with Signature
Review Signature
End wth Error


Agree Essential Terms
Terms Agreed
Agree Contract Text
Text Agreed
Authorize Contract
Contract Ready
Terms Not Agreed
Text Not Agreed
Contract Not Authorized

Error Event


Client
Sell product
Supervisor
Inform client
Request to purchase
Sell
Prepare
Deliver
Inform supervisor

Error


Business Process ErrorIntermediateEvent
subProcess1
s1
s2
40,10,20,30
Probabilities
Task1
Task2
Task3
Task4
normal_end
error_end_fault1
error_end_fault2
error_end_default
e1
error_ie_fault1
e2
error_ie_fault2
e3
error_ie_default
e4

Default flow

The default Sequence Flow is taken (a token is passed) only if all the other outgoing Sequence Flows from the Activity or Gateway are not valid (i.e., their conditions are false).

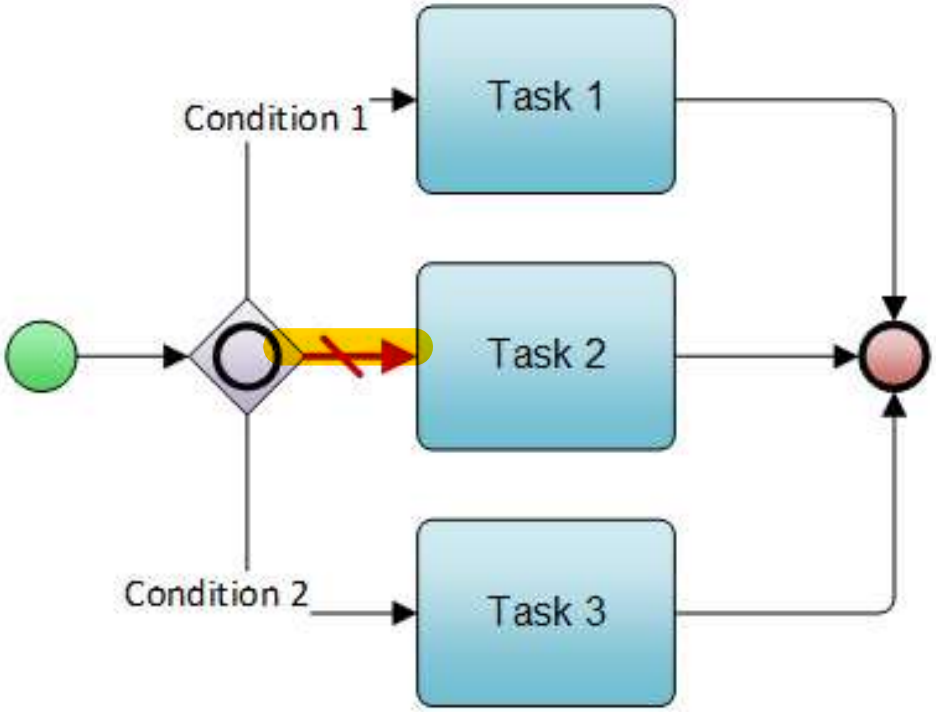
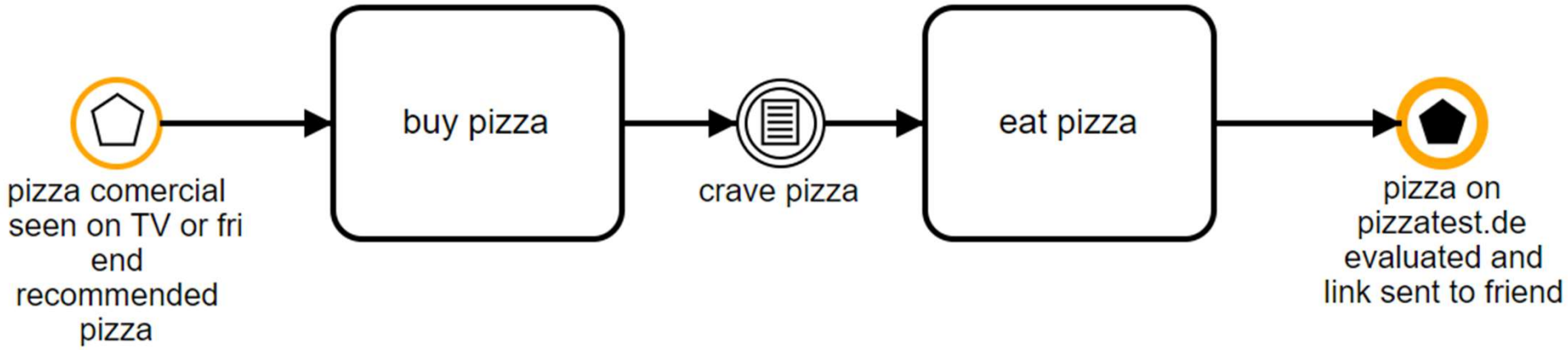


Figure 5: Default Flow (red arrow)

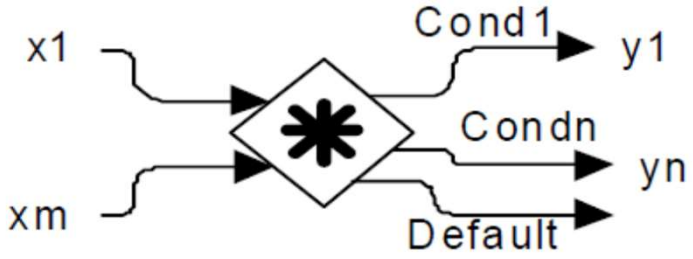
Multiple event

- We can use the multiple event to summarize several events with a single symbol.
- The diagram below applies the multiple event to our pizza scenario.
- In the example, we try a new pizza after having seen it on TV or after a friend recommended it.
- After eating it, we will rate the pizza on Pizzatest.de and in turn inform our friend if we also recommend this pizza.

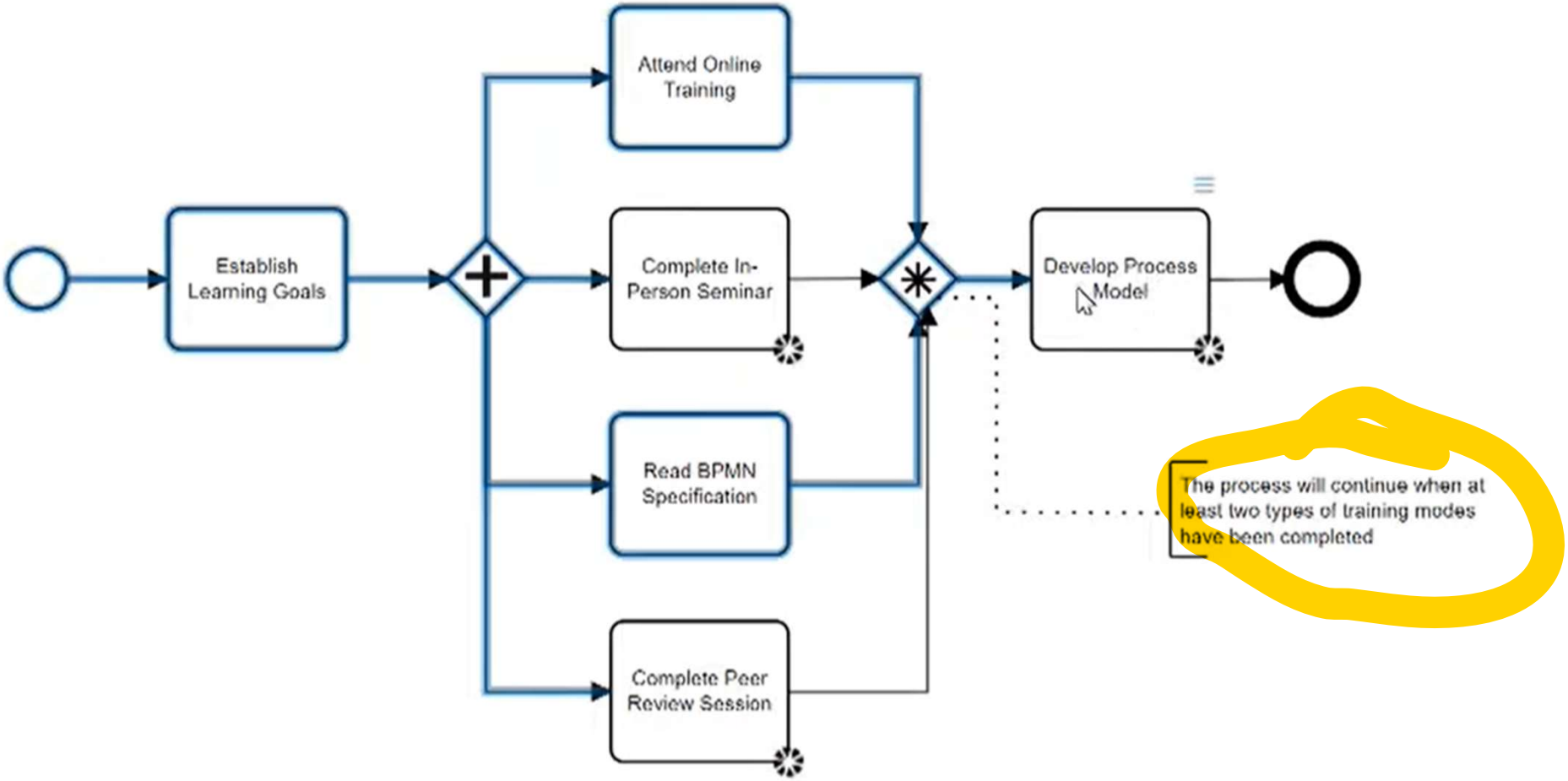


Complex Gateways

- The Complex Gateway facilitates the specification of complex synchronization behavior, in particular race situations.
- The diverging behavior is similar to the Inclusive Gateway. Each incoming gate of the Complex Gateway has an attribute activationCount, which can be used in an Expression as an integer-valued variable.
- This variable represents the number of tokens that are currently on the respective incoming Sequence Flows.
- The Complex Gateway has an attribute activationExpression. An activationExpression is a boolean Expression that refers to data and to the activationCount of incoming gates.
- For example, an activationExpression could be x1+x2+…+xm >= 3 stating that it needs 3 out of the m incoming gates to have a token in order to proceed. T



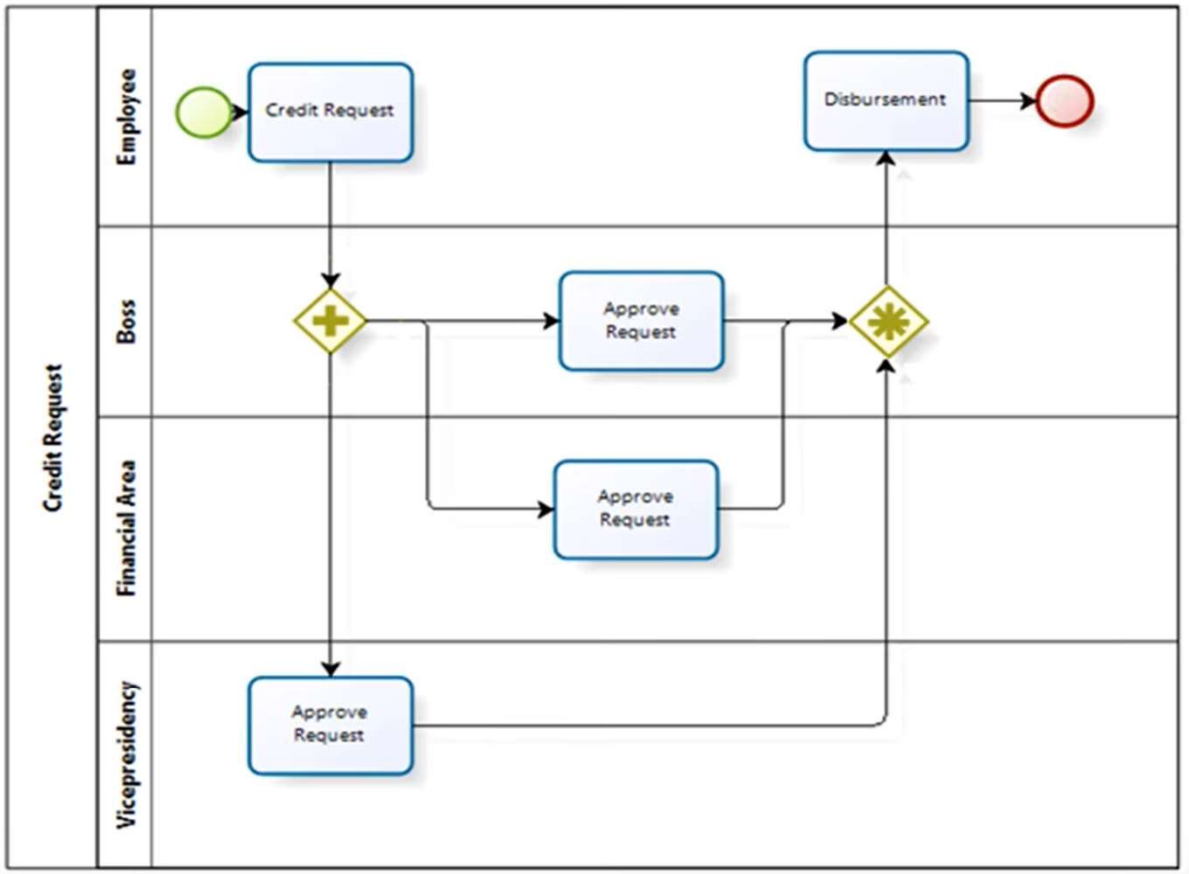
Complex Gateway



Loan request process

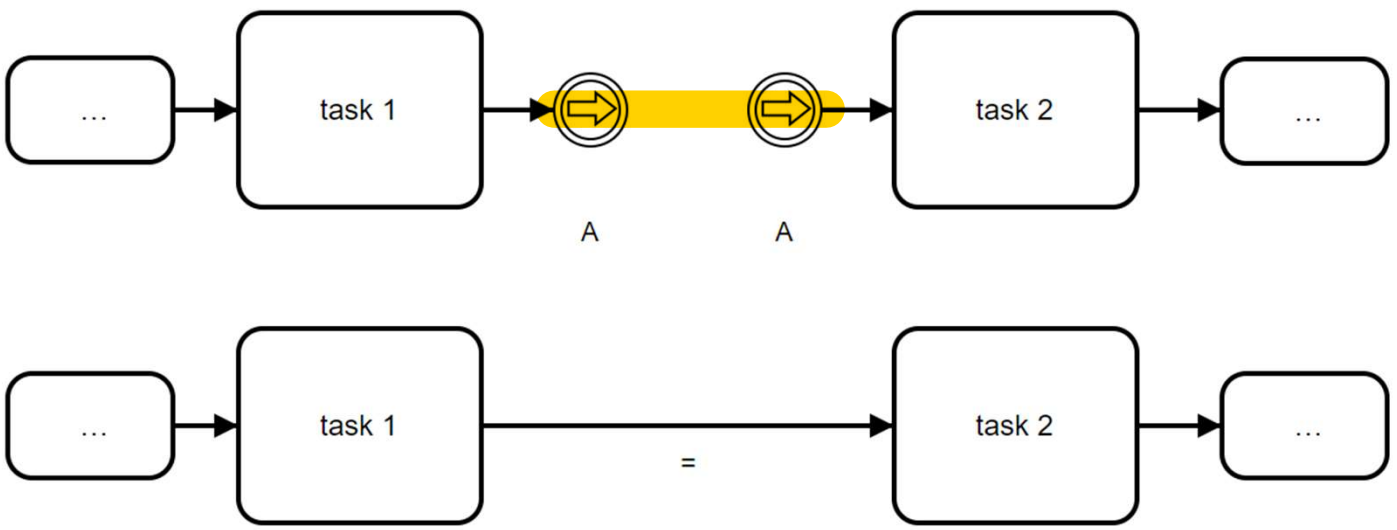
An employee requests a loan from the company. This must be approved by his or her boss, the financial area or the vice-presidency. When at least two of the three approve the request, the money is given to the employee.

Loan request process



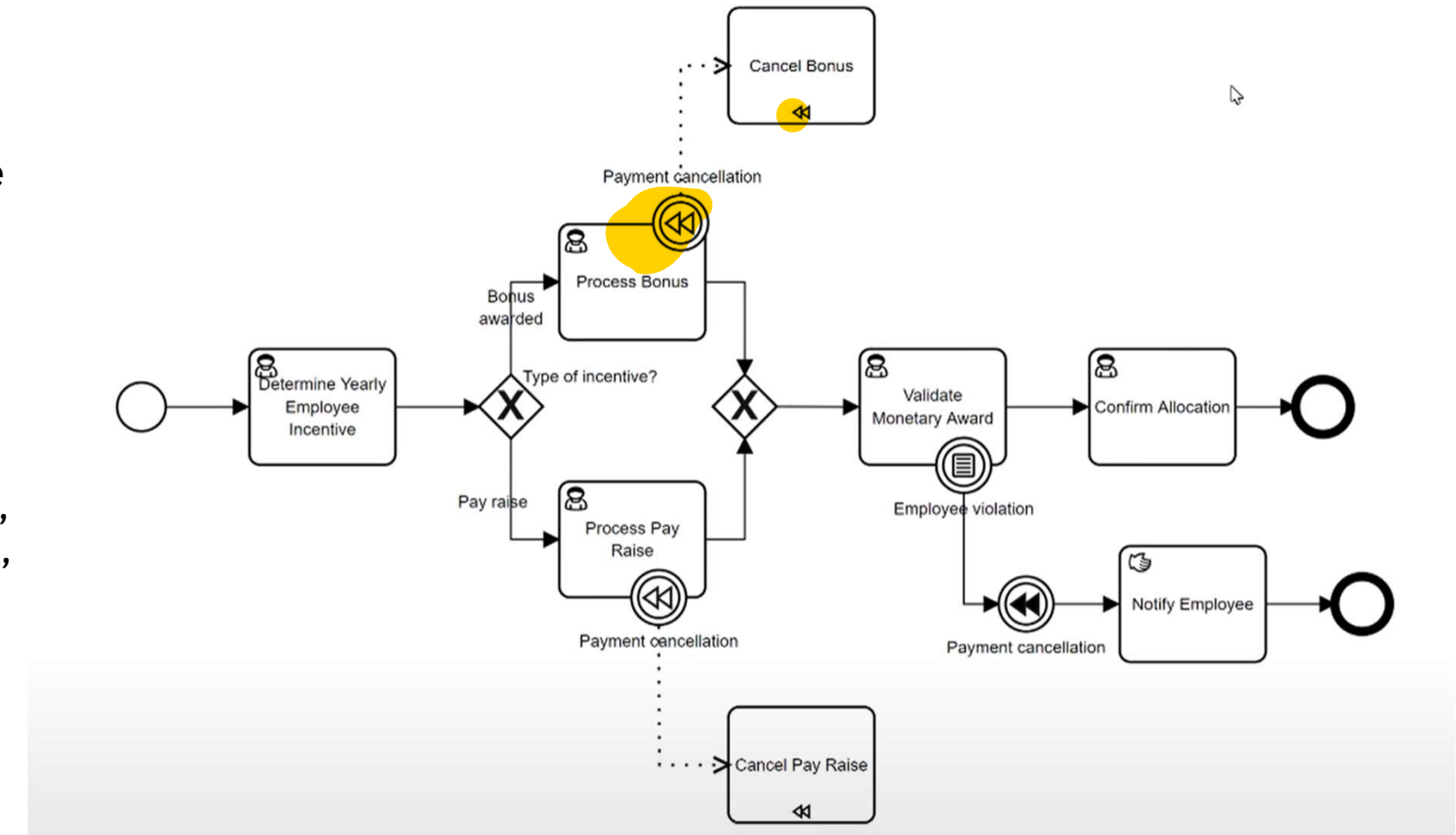
Link

- The link event is a special case. It has no significance related to content, but it facilitates the diagram-creation process. As shown below, you can draw two associated links as an alternative to a sequence flow.
- Here, “associated” means there is a throwing link event as the “exit point,” and a catching link event as the “entrance point,” and the two events are marked as a pair – in our example by the designation “A.”.

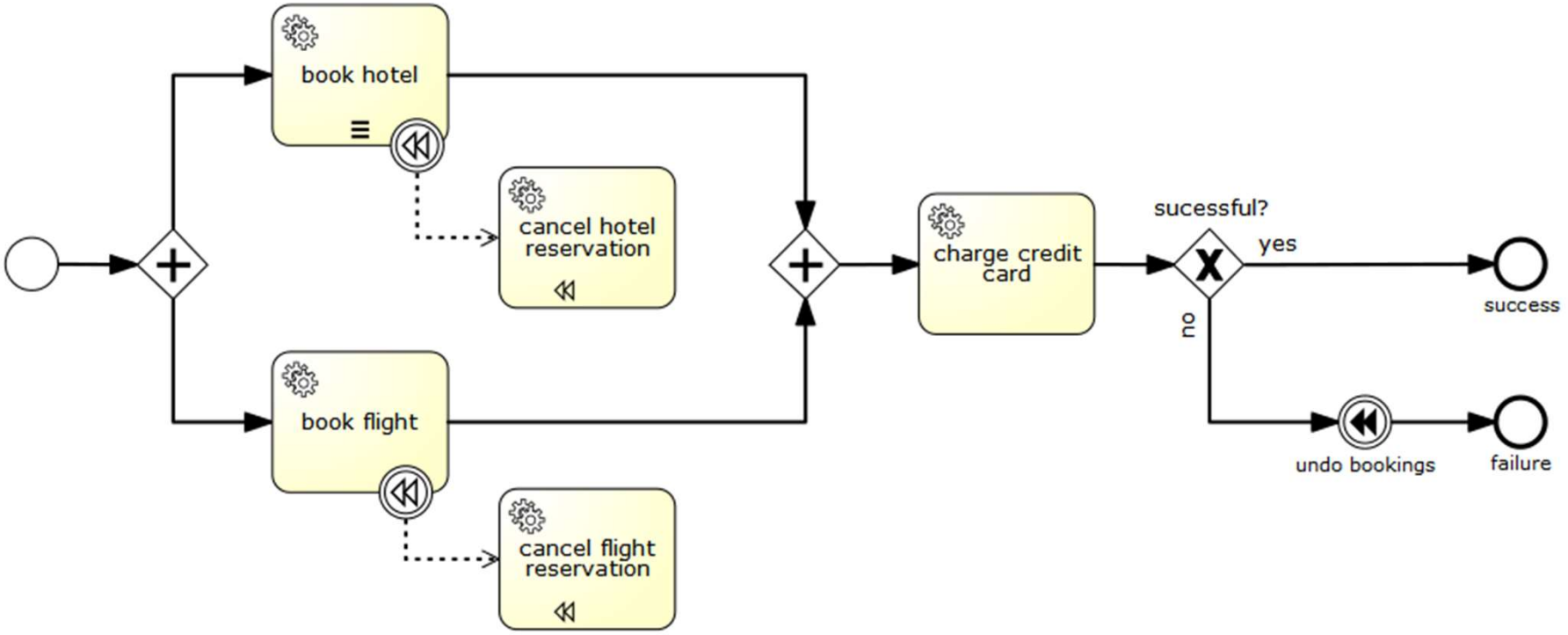


Compensation

- We execute tasks in our processes that sometimes have to be cancelled later under certain circumstances.
- Typical examples are: Booking a train or airline ticket, Reserving a rental car, Charging a credit card, Commissioning a service provider



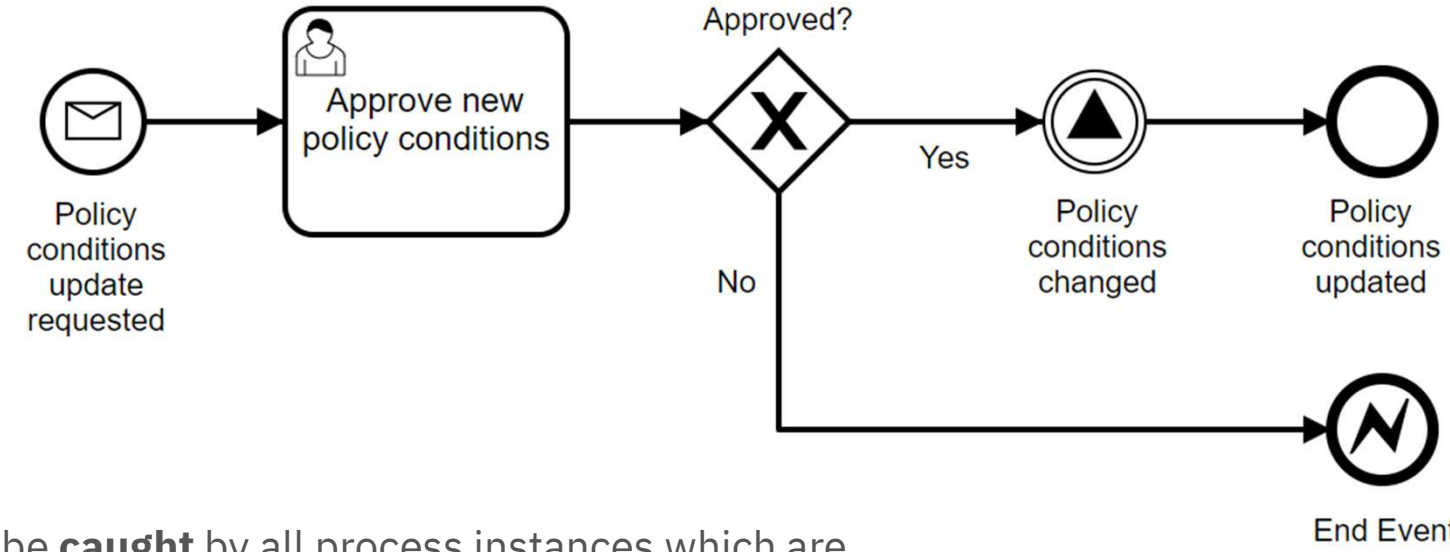
Exercise: Cancelling hotel and flight



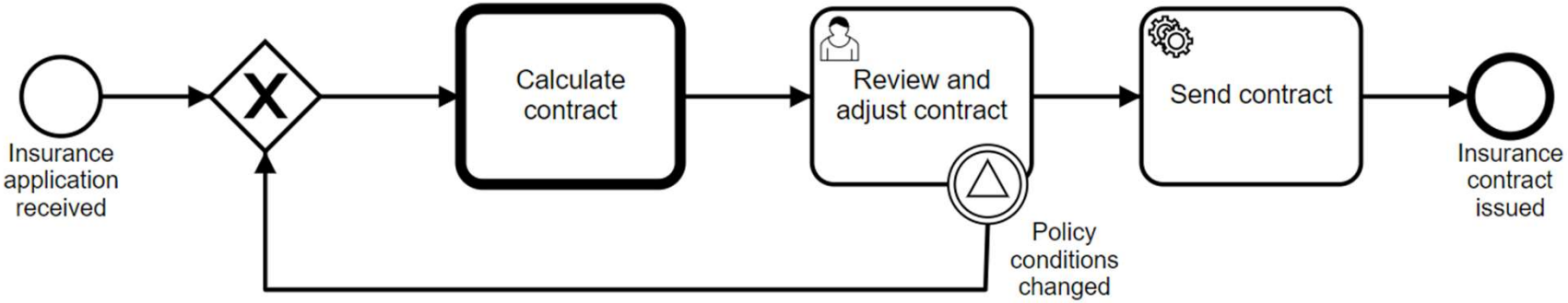
Signal event

- Signals are similar to messages, which is why you can model them in BPMN as events just as you can with messages.
- The symbol for a signal is a triangle. The essential difference between a signal and a message is that a signal is an event of global scope (broadcast semantics) and is delivered to all active handlers.
- A message is always addressed to a specific recipient.
  - An e-mail contains the e-mail address of the recipient, a call starts with dialing the telephone number, and so on.) It is relatively undirected. Anyone who receives the signal and wants to react may do so.

After the changes have been reviewed by a human participant, a signal event is **thrown**, signaling that a policy has changed:

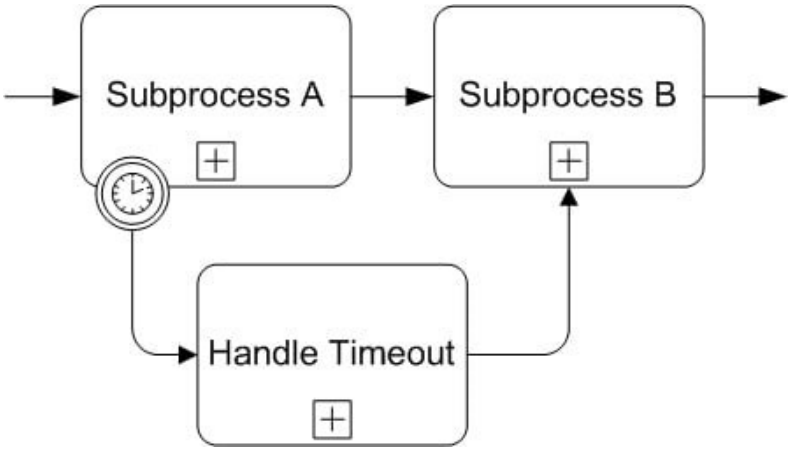


This event can now be **caught** by all process instances which are interested. The following is an example of a process subscribing to the event.

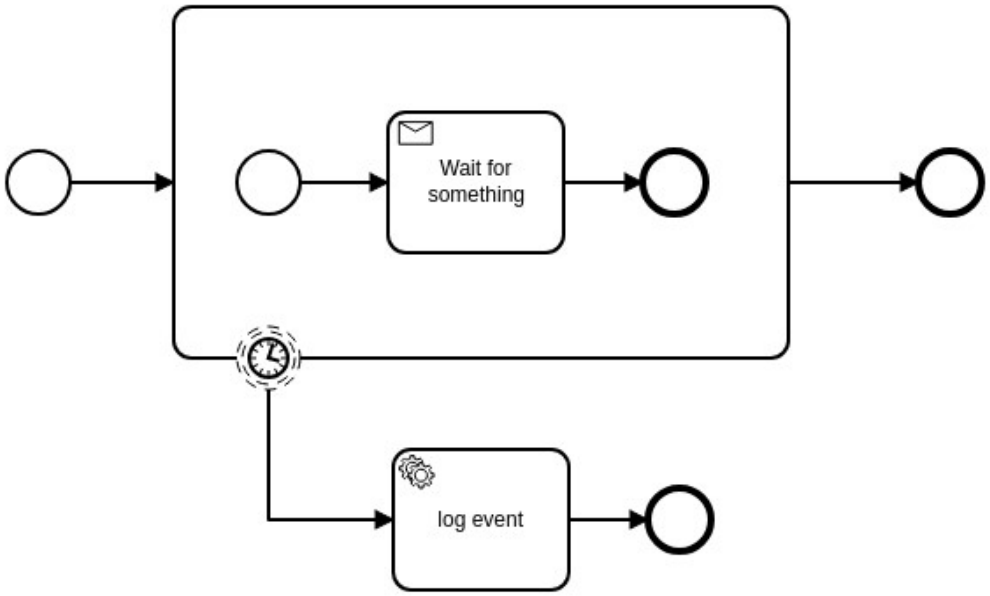


Non-interrupting events

- In the case of an Interrupting Event, the activity which was being performed will immediately be canceled.
- An Interrupting Event is shown by a solid circle or two solid circles around the event icon



- In the case of a Non-Interrupting Event, the activity which was being performed will continue in parallel along with the new flow that was initiated by the boundary event.
- The current activity will NOT be cancelled (or interrupted).
- A Non-Interrupting Event is shown by a dashed circle or two dashed circles around the event icon depending on whether the event was a start event (one dashed circle) or an intermediate event (two dashed circles).



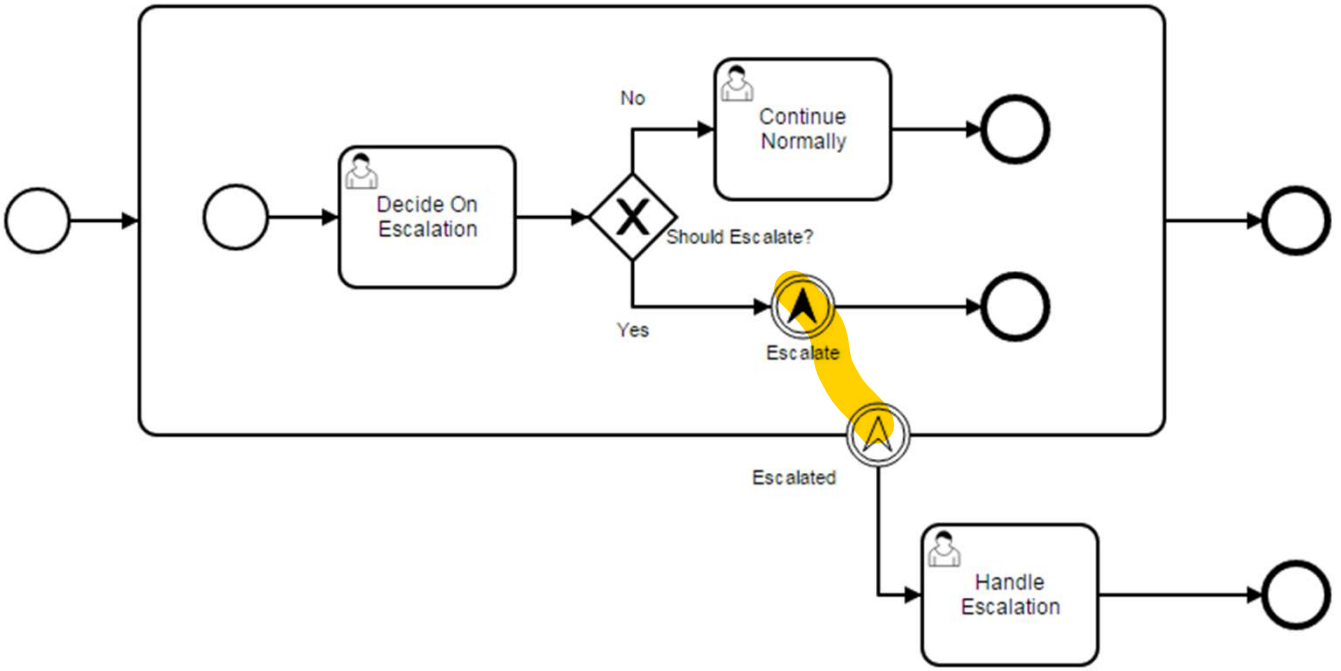
Escalation

- This type of Event is used for handling a named Escalation. If attached to the boundary of an Activity, the Intermediate Event catches an Escalation.
  - For an Escalation Event that interrupts the Activity to which it is attached, the boundary of the Event is solid
  - For an Escalation Event that does not interrupt the Activity to which it is attached, the boundary of the Event is dashed.

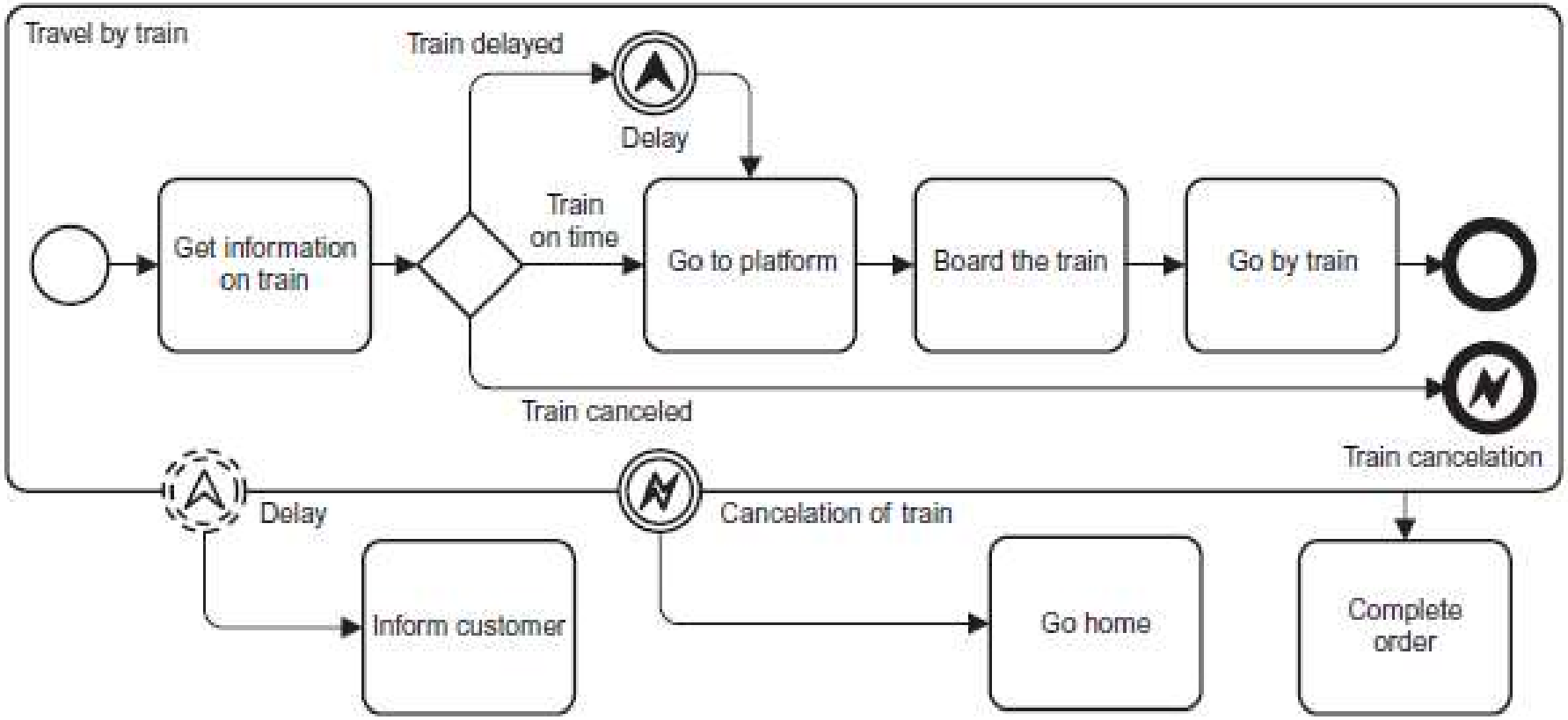
Interrupting

Non-Interrupting

Escalation event



Escalation


Travel by train
Train delayed
Delay
Get information on train
Train on time
Go to platform
Board the train
Go by train
Train canceled
Train cancelation
Delay
Cancelation of train
Inform customer
Go home
Complete order

BIBLIOGRAFIA

- OMG: Business Process Model and Notation - BPMN, Object Management Group, Document Number: formal/2011-01-03, 2011, http://www.omg.org/spec/BPMN2.0.
- WHITE. S. A.; MIERS, D.; BPMN Modeling and Reference Guide: Understanding and Using BPMN, Future Strategies Inc., Florida, USA, pp. 216.
- SHAPIRO, R.; WHITE, S. A.; PALMER, N.; MUEHLEN, M.; ALLWEYER, T.; GAGNÉ, D.; *et al.*, BPMN 2.0 Handbook, Edited by Layna Ficher, http://www.bpm-guide.de/wp-content/uploads/2011/09/BPMN-2.0-Handbook-Camunda.pdf.