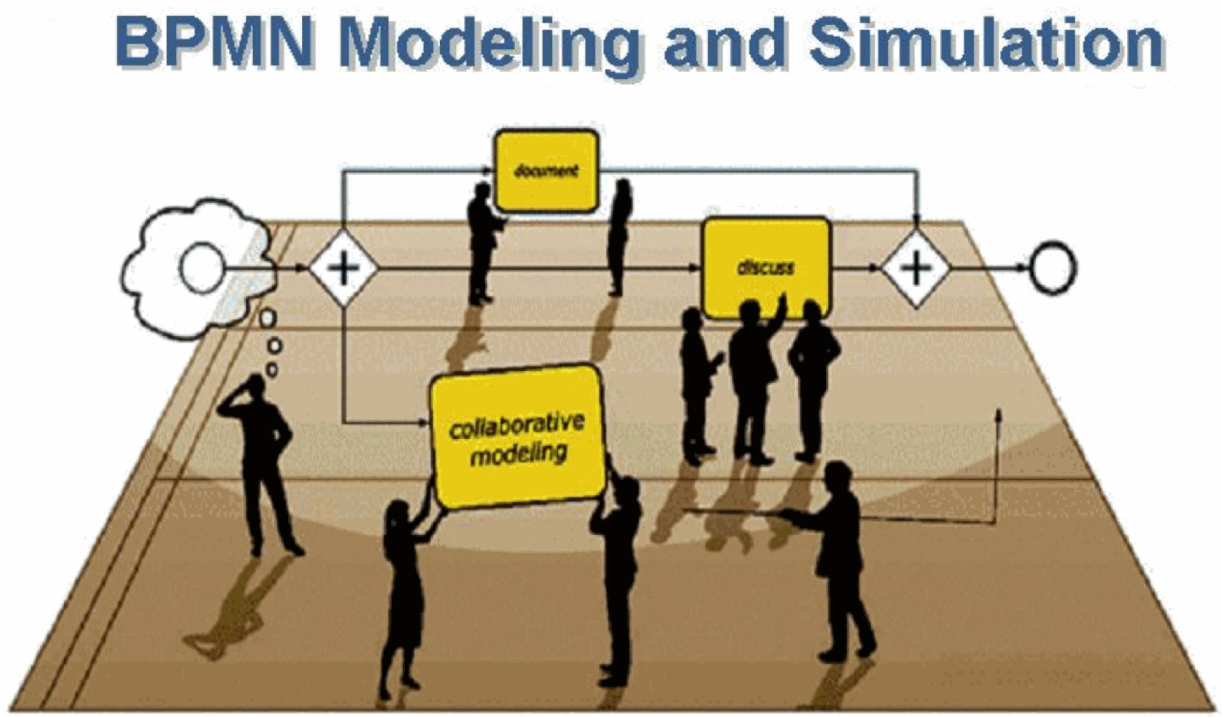
University of Pisa

MSc in Computer Engineering

Supply Chain Operation Management

"Large and complex organizations are a tangible manifestation of advanced technology, more than machinery itself." (J.K. Galbraith)



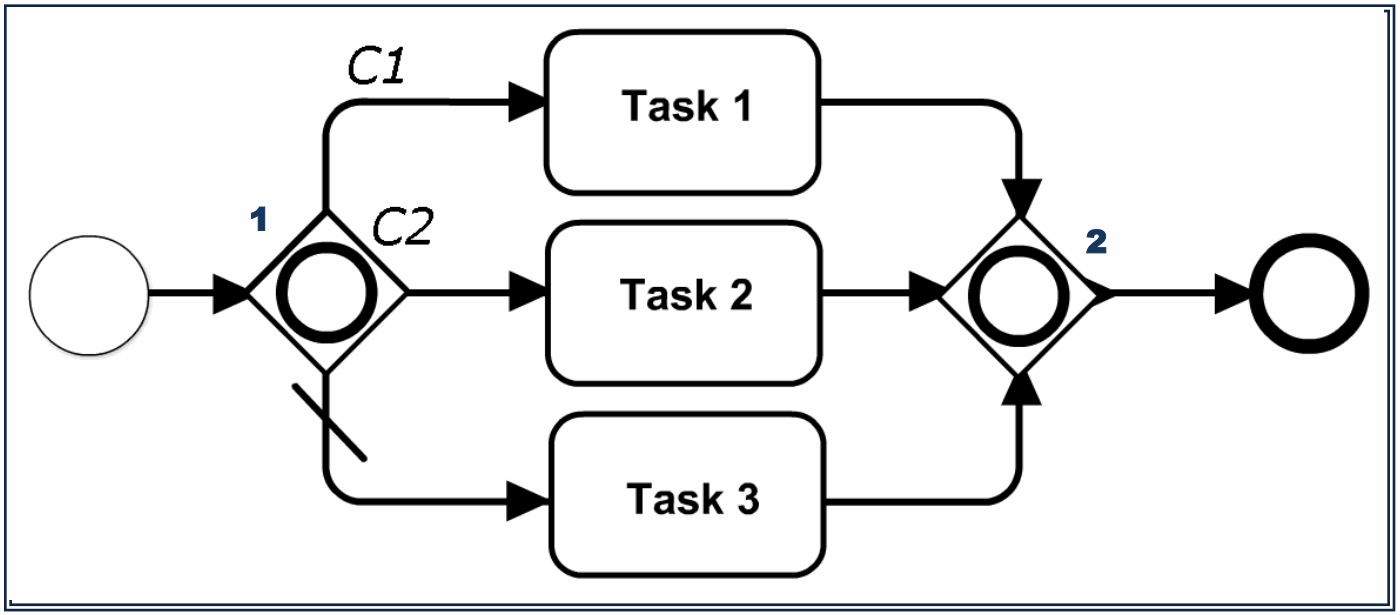
Lectures

Mario G. Cimino, Department of Information Engineering, Center for Logistics Systems

Pisa, March-May 2016, Monday 14.30-17.30, Room: ADInform2

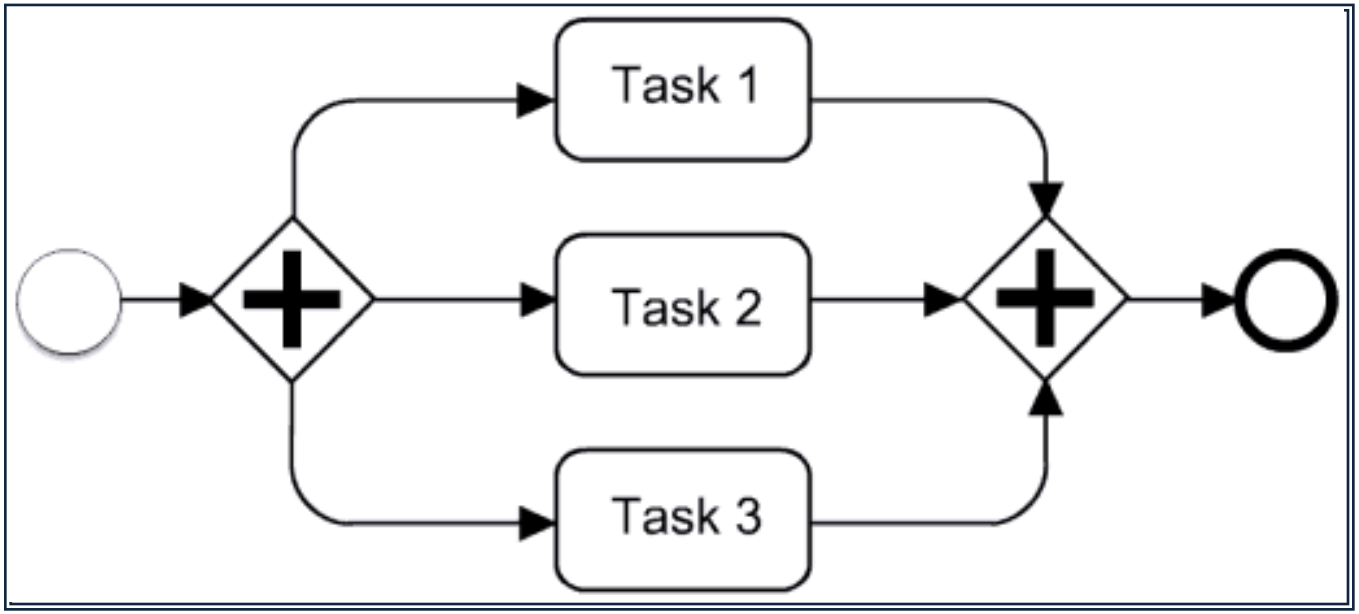
Gateways

- An *Inclusive Gateway*[1] can be used as a decision point where several outgoing sequence flows are possible, they are all constrained by conditions, each outgoing sequence flow with a condition evaluated as being true will be followed. Effectively it might spawn several execution points.
- Used as a merge[2] the Inclusive Gateway will synchronize all the execution points produced upstream but at most one for each incoming Sequence Flow



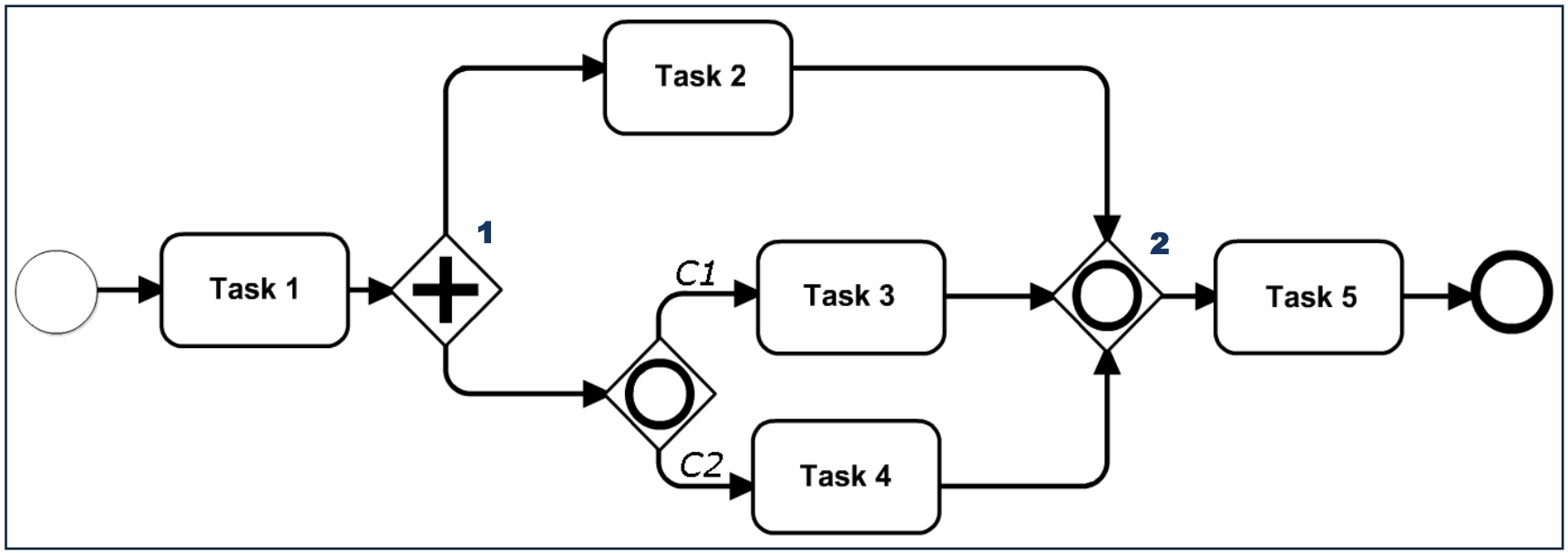
http://www.iet.unipi.it/m.cimino/scom/res/mov06.swf

- A gateway's exit can also be marked with a small diagonal slash as *default* sequence flow. It will be selected automatically, (only) if no condition of the other sequence flows is true. This ensures the actual selection of at least one sequence flow.
- A *Parallel Gateway*[1] provides a mechanism to fork and synchronize flows. There are no conditions associated to this gateway.



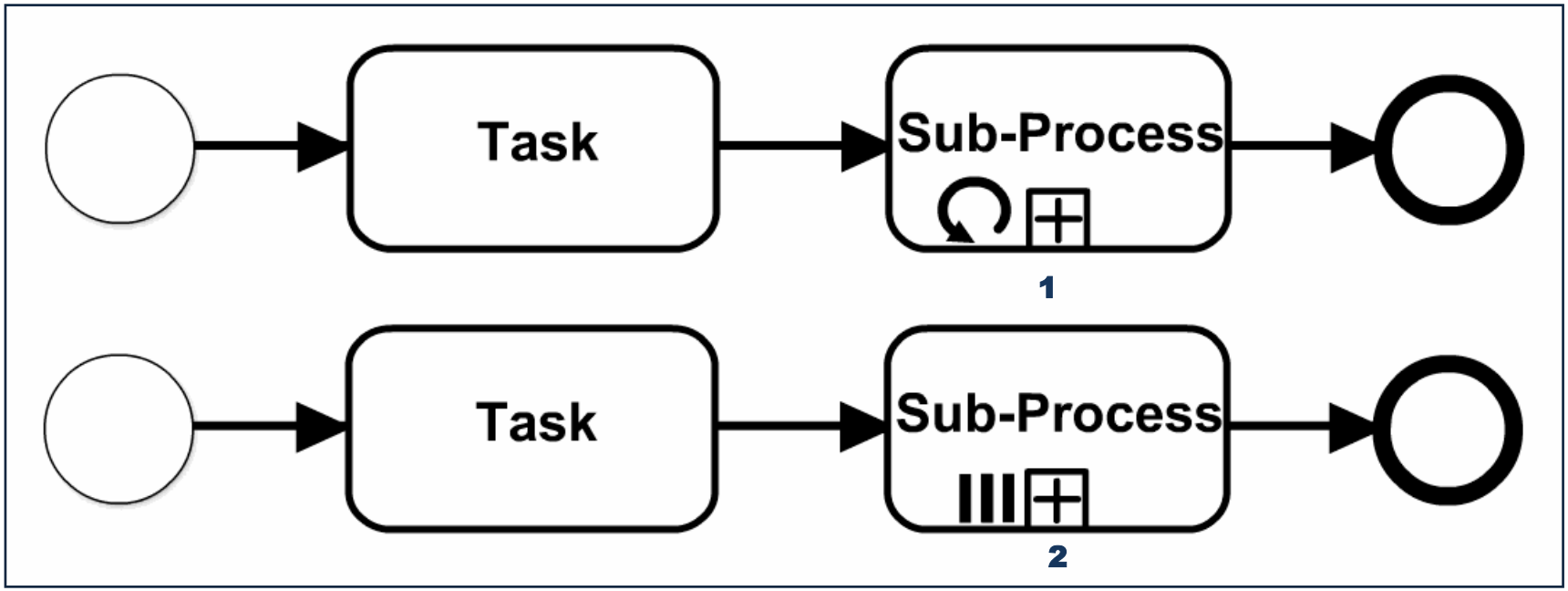
http://www.iet.unipi.it/m.cimino/scom/res/mov07.swf

- In the following model, notice how the first inclusive gateway produces two tokens because both conditions ‘C1’ and ‘C2’ are evaluated as being true. The second inclusive gateway[2] will not only synchronize the token produced by the upstream inclusive gateway, but also the one coming from the upstream parallel gateway
- Exercise: what happens if the second inclusive gateway is replaced by a parallel gateway? Consider a scenario in which *C1* (or *C2*) is false.



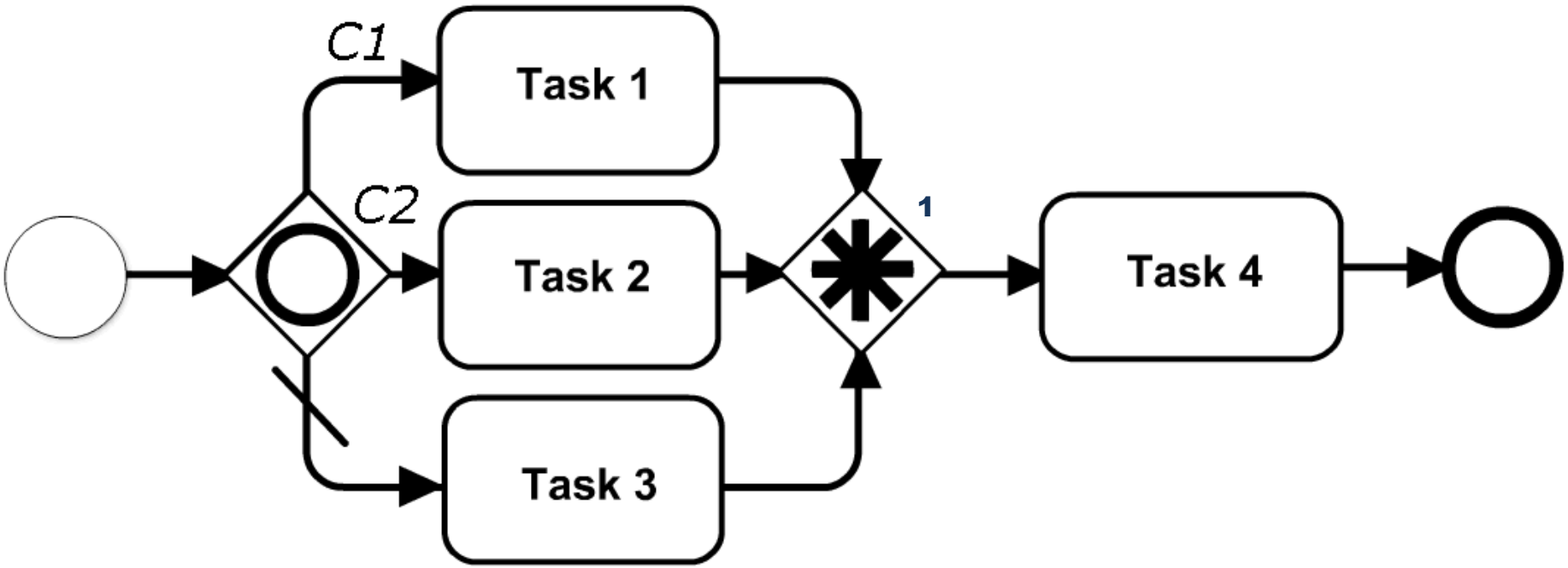
http://www.iet.unipi.it/m.cimino/scom/res/mov08.swf

- A *Loop activity* is an activity repeated sequentially a number of times. It is characterized by a 'Sequential Loop' marker[1] on the Task / Sub-Process. As a result, the sub-process will be instantiated several times sequentially. The number of instances to execute might be: (i) defined at design time, (ii) affected at runtime from some process data, (iii) computed at runtime.
- A *multi-instance activity* is an activity repeated in any order, sequentially or in parallel, whose number is defined in advance. It is characterized by a "Parallel Loop" marker[2] on the Sub-Process.



http://www.iet.unipi.it/m.cimino/scom/res/mov09.swf
http://www.iet.unipi.it/m.cimino/scom/res/mov10.swf

- The *Complex Gateway* addresses complex cases which would require the combination of several other gateways. To avoid this, the behavior of the complex gateway can be scripted using an expression language.



http://www.iet.unipi.it/m.cimino/scom/res/mov11.swf

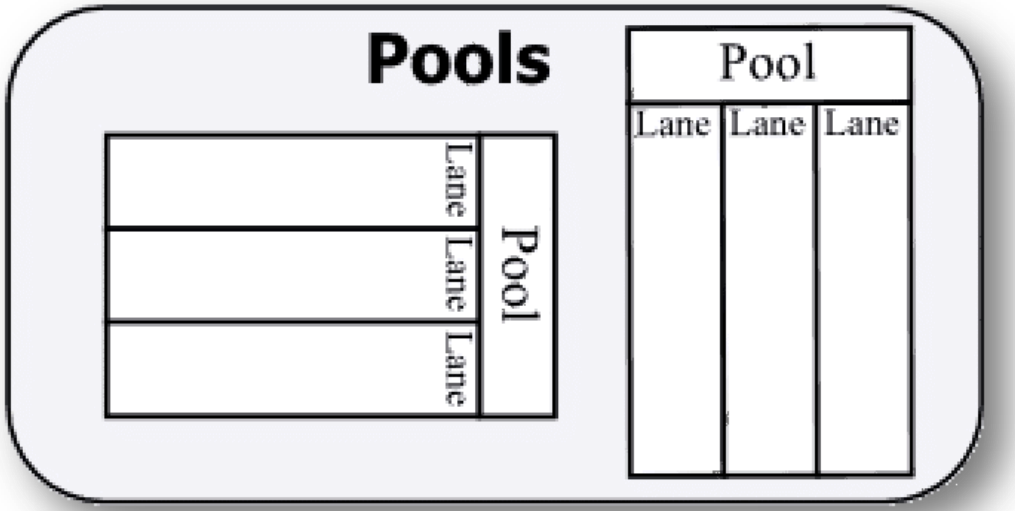
- As a result the complex gateway can be used to handle every situation. However, a good practice is to avoid it since it makes the process models less readable.

- Modeling with (+) or without (–) gateways?
  + it is not possible to model entirely without gateways
  + gateways are more expressive and, sometimes, necessary
  – if gateways are omitted as far as possible, more compact models can be created
  – gateways with several inputs and several outputs may cause misunderstanding

- A *Pool* represents a participant in a business process. It can be a specific entity (e.g. department) or a role (e.g. assistant manager, doctor, student, vendor). A *Pool* is represented by a box.

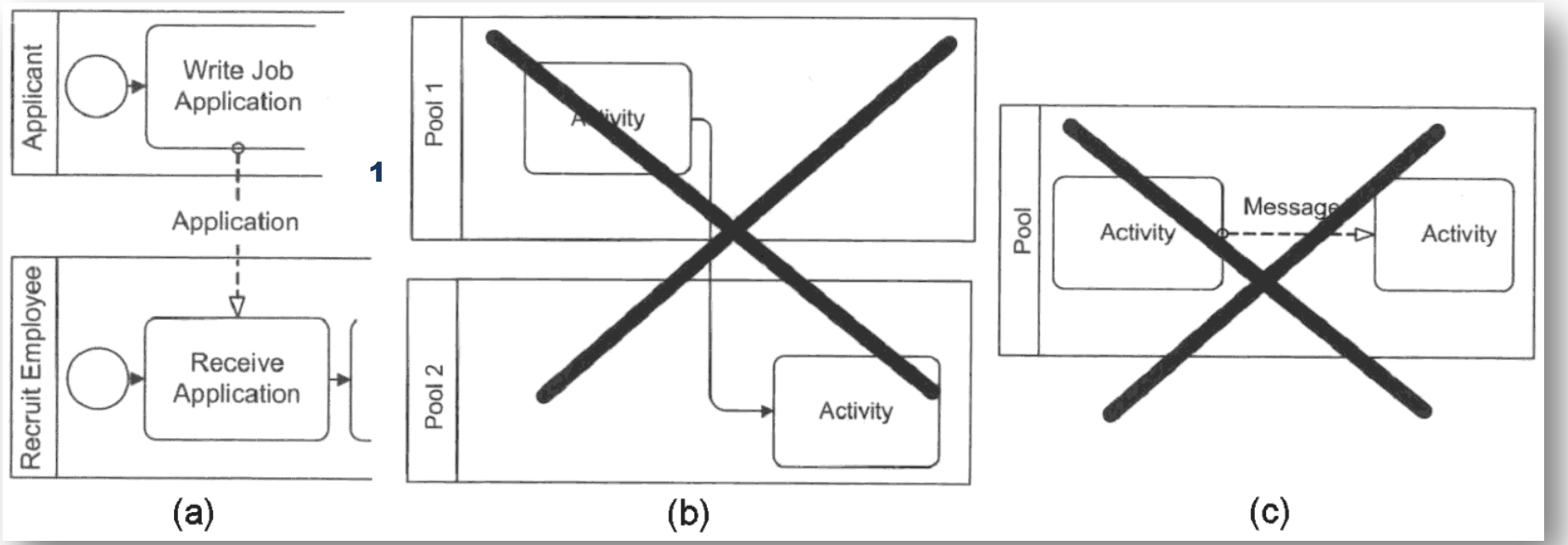
- A *Lane* is a sub-partition within a Pool. For instance, having a pool Department, you may have Department Head and General Clerk as lanes. Same as pools, you can use lanes to represent specific entities or roles who are involved in the process.

References:
OMG, *BPMN 2.0 Specification*,
pp. 29-41.

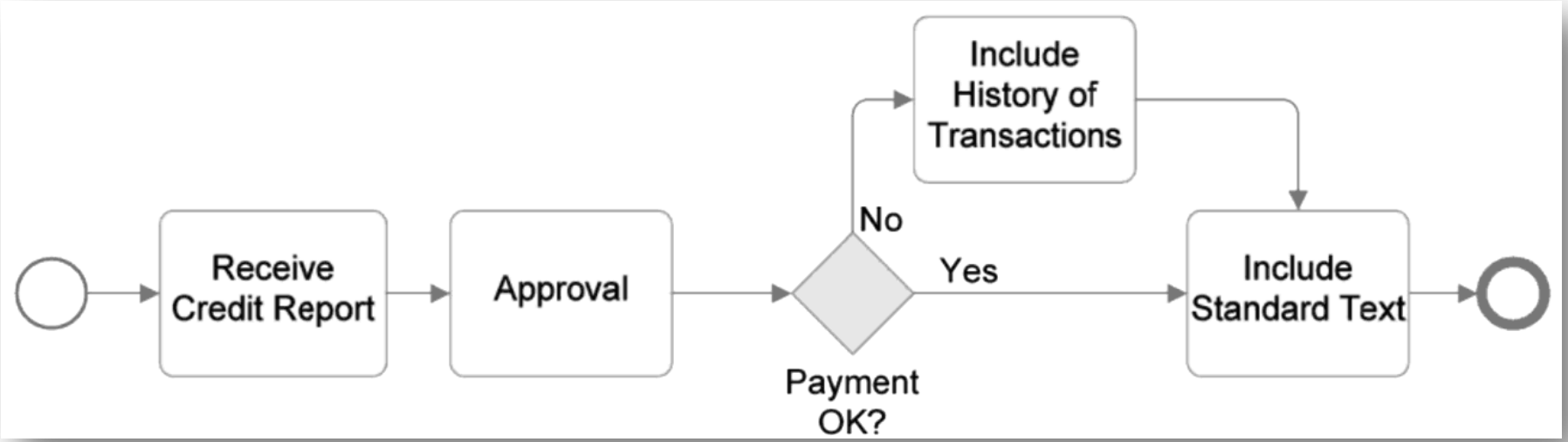


Orchestration and Choreography

- A *private* business process is internal to a pool, i.e., does not interact with external pools.
- A *public* process interacts with other pools via message flows[1], represented as white dashed arrow (a).
- Control flow between pools (b) and message flow within a pool (c) are **forbidden**

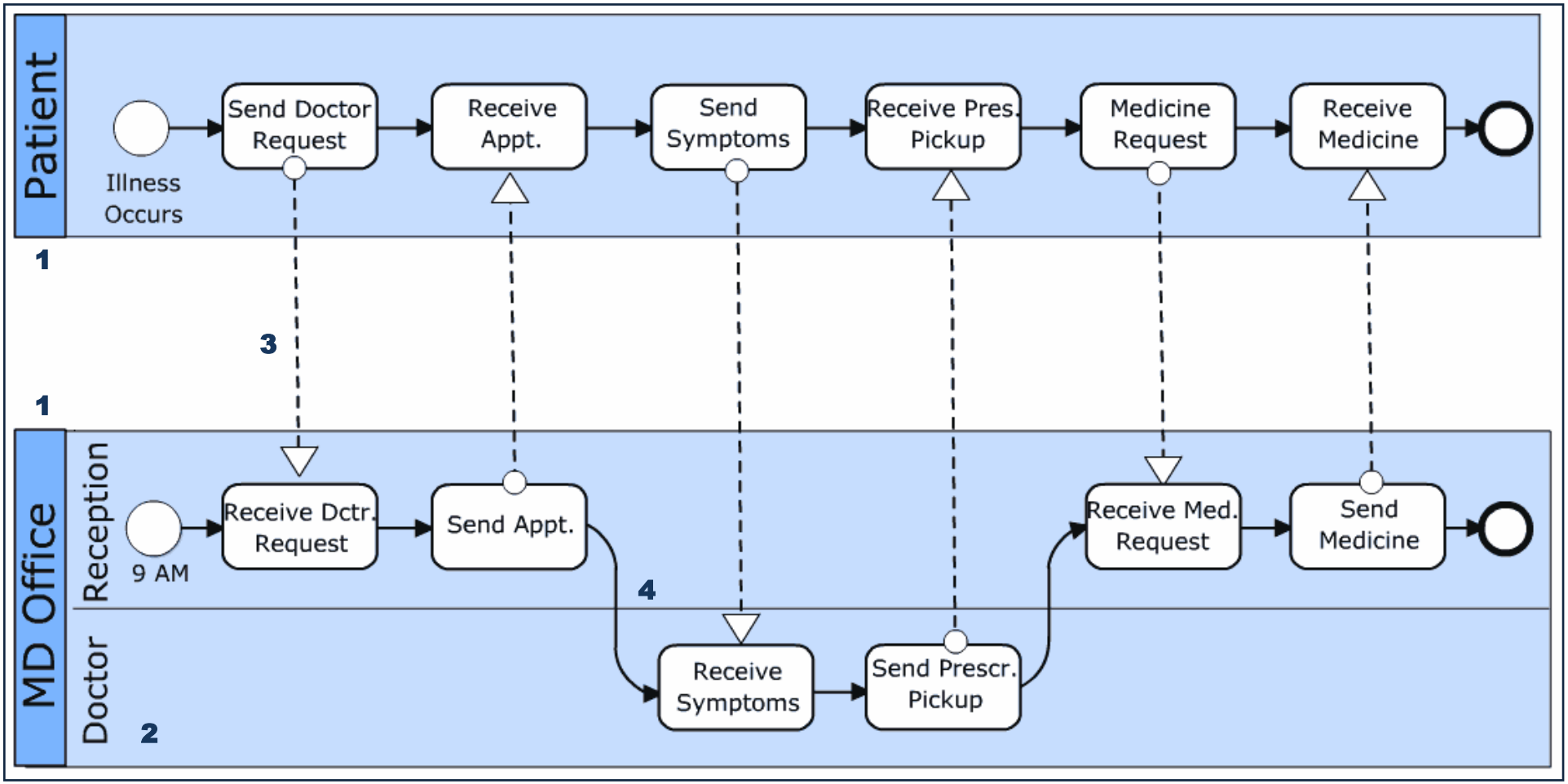


- Considering the centralization/distribution of the control flow, there are two types of BPMN processes: *orchestration* and *collaboration/choreography*.
- *Orchestration*: a process with *central control* of activities (like the control of an orchestra director), modeled as a private process, and containing optionally black-box pools presenting external entities.



- *Collaboration* ("business protocol"): the ordered set of interactions between two or more participants as shown by pools, with their public processes and message flow between them. A collaboration is characterized by *distributed control*. It is also called *choreography*: each individual participant reacts to events generated by the other participants (as in a choreographed dance where dancers react to behaviors of their peers). It can contain optionally some orchestration or just black-box pools.

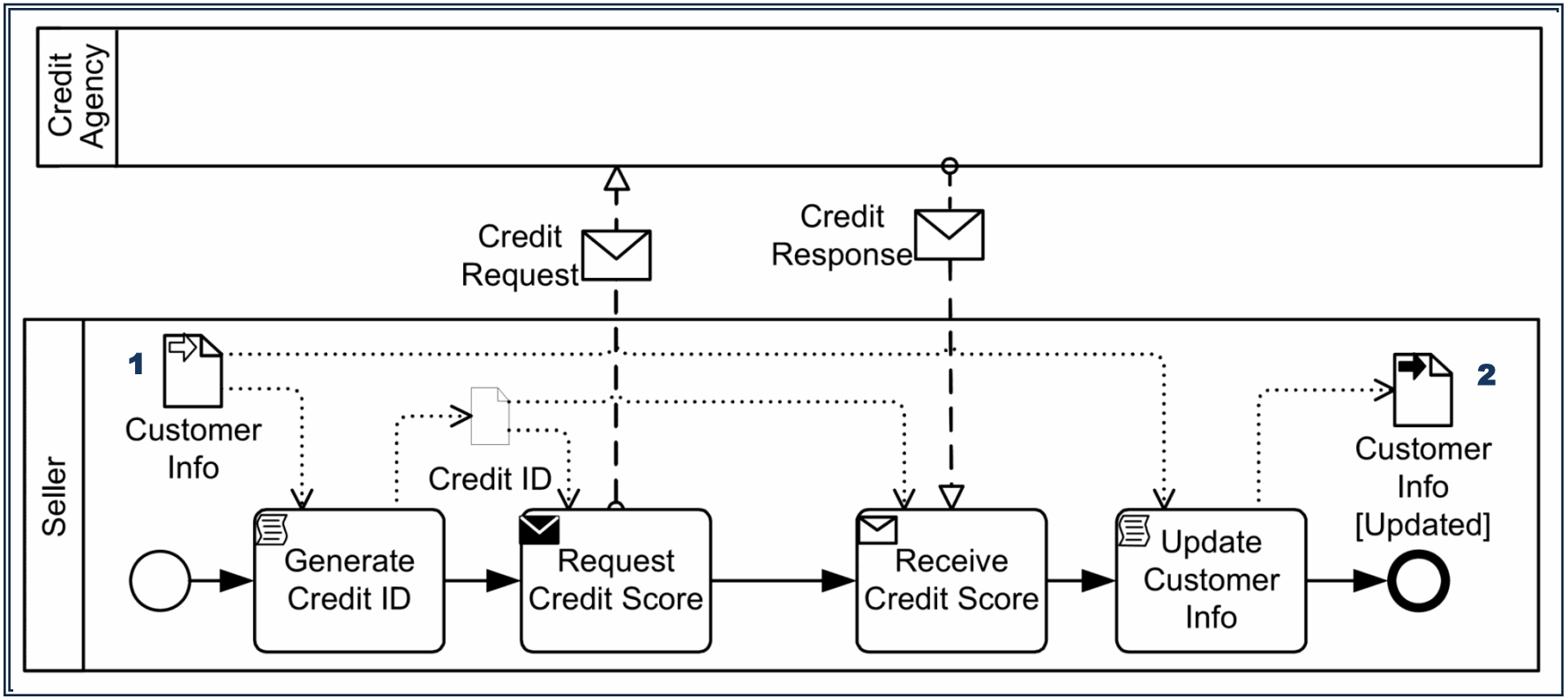
- Collaboration between a patient and a medical office. The processes within pools are abstracted to hide complexity or private information to partners. Note: pool (1), lane (2), message flow (3) and control flow (4). Here, choreography is made by the message flows, whereas orchestration is made by sequence flow, events, tasks.



http://www.iet.unipi.it/m.cimino/scom/res/mov12.swf

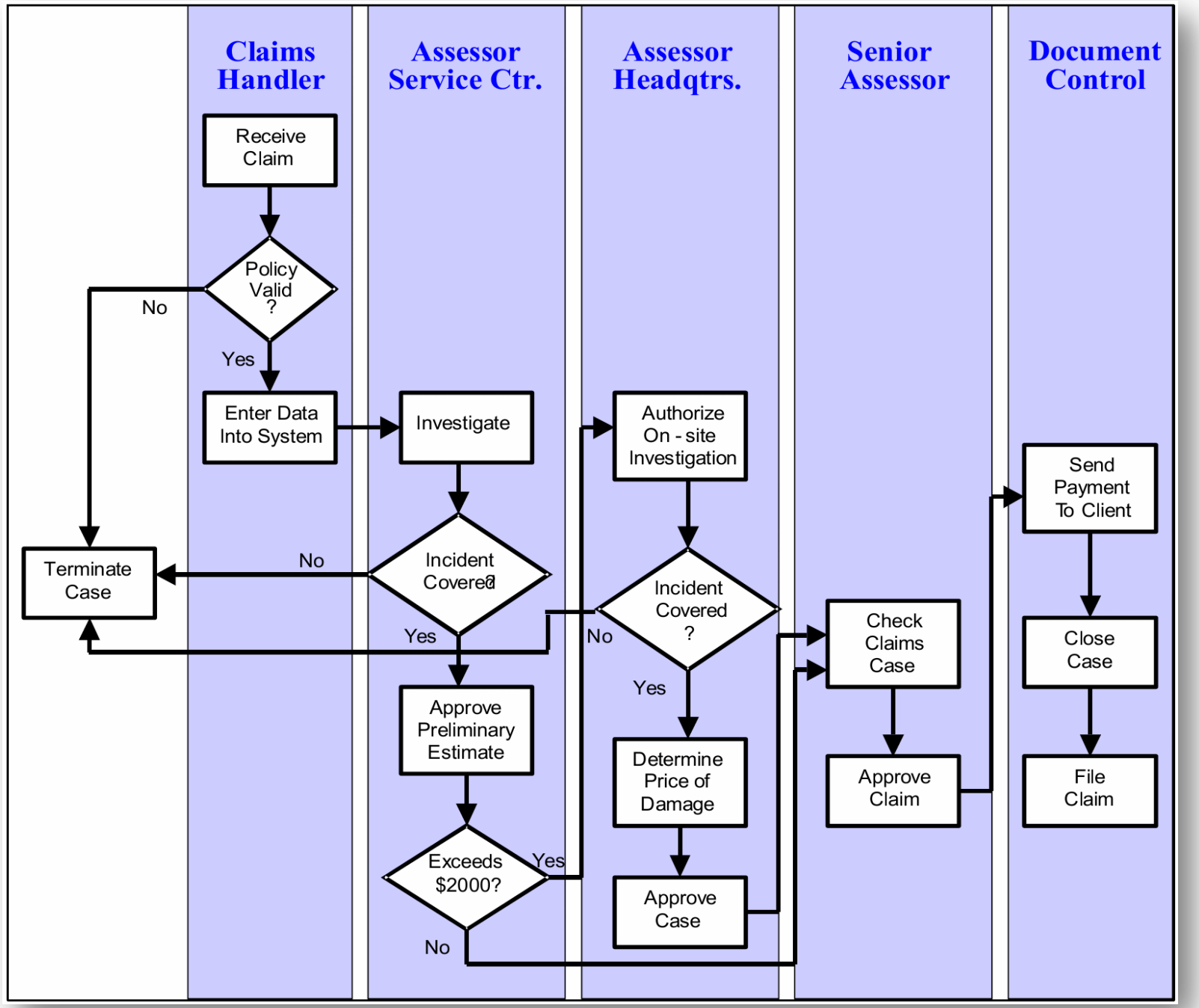
Data objects

- BPMN allows the explicit modeling of data transfer, via *data objects*, *messages* and *data store*. Data objects only exist within a process, whereas data store represent persistent data. UML class diagrams or technical terms diagrams could be used to refine data objects in process models → process view is integrated with data view.
- A directed data association is drawn as a dotted line to model which activity outputs and takes as input a certain data object. Data input[1] and data output[2] can represent a form of dependency between activities.



Modeling from informal diagrams: a claims process at an insurance company

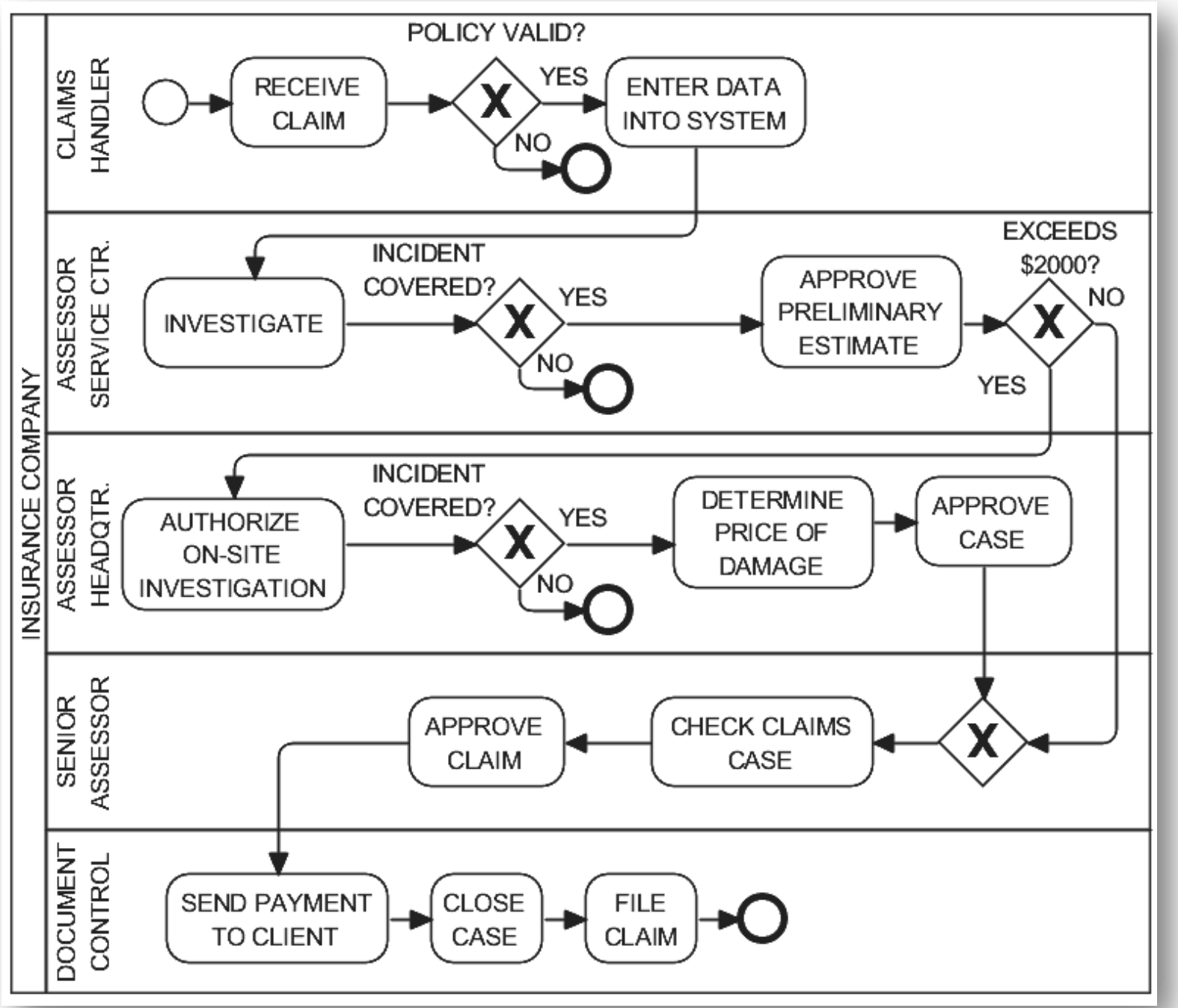
- A personal claims department in an insurance company handles claims made by their clients. The figure on the next page is a non-BPMN process map depicting the personal claims process in terms of swimlanes.
- The first lane corresponds to work done by a claims handler (CH) located at the client's local service center. Upon arrival of a claim, the assessor determines if the client has a valid policy. If no (5% of all cases), then the case is terminated; otherwise (95% of all cases), the assessor enters the appropriate information in the system.
- In the second lane, an assessor located at the service center (ASC) receives data from the claims handler. The assessor first determines if the claim is covered by the client's policy. If not (5% of cases), the case is terminated; otherwise (95% of cases), the assessor approves the preliminary estimate of the damage. If the damage exceeds $2,000 (35% of cases), the claim is sent to an assessor at headquarters for approval; otherwise (65% of cases), it is sent directly to a Senior Assessor.


Claims Handler
Assessor Service Ctr.
Assessor Headqtrs.
Senior Assessor
Document Control
Receive Claim
Policy Valid ?
No
Yes
Enter Data Into System
Investigate
Authorize On - site Investigation
Send Payment To Client
Terminate Case
No
Incident Covered?
Incident Covered ?
Check Claims Case
Close Case
Yes
No
Yes
Approve Preliminary Estimate
Determine Price of Damage
Approve Claim
File Claim
Exceeds $2000?
Yes
Approve Case
No

- Lane 3 corresponds to the assessor at headquarters (AHQ). The assessor first authorizes the on-site investigation of the accident. If the investigation determines that the incident is not covered by the client's policy (2% of cases), then the case is terminated; otherwise (98% of cases), a final price is determined and the case is approved.
- In lane 4, the senior assessor (SA) receives the claim, checks it, completes it, and provides the final approval. Once the claim is approved, it is sent to documentation control.
- Documentation control (DC), in lane 5, is in charge of processing the payment to the client, closing the case, and, finally, filing the claim.

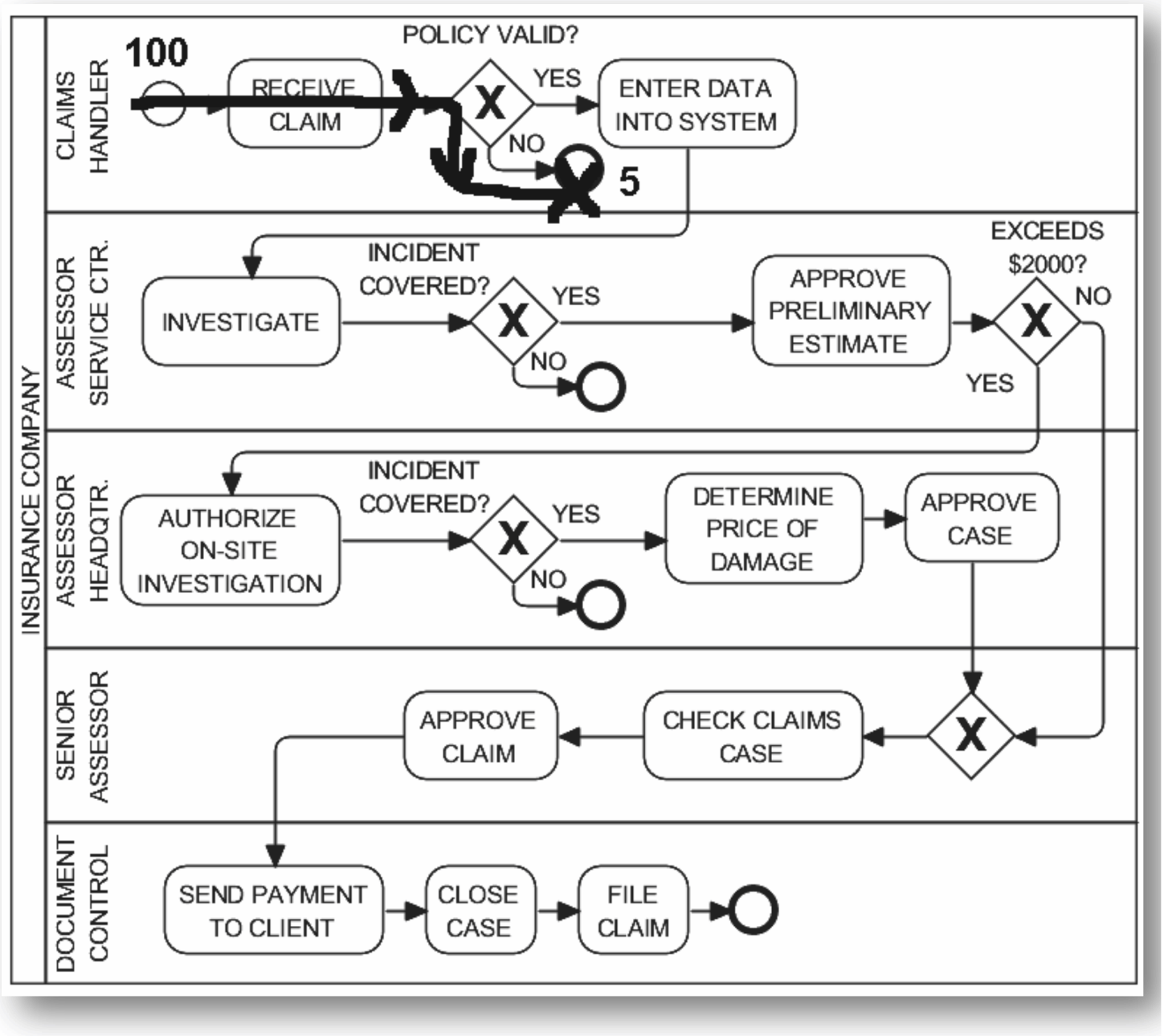
✓ Create a BPMN model

✓ Given 100 starting tokens, determine the number of ending tokens for each scenario (path), considering the aforementioned branching proportion (percentage of cases) for each gateway.


INSURANCE COMPANY
CLAIMS HANDLER
POLICY VALID?
RECEIVE CLAIM
YES
NO
ENTER DATA INTO SYSTEM
ASSESSOR SERVICE CTR.
INVESTIGATE
INCIDENT COVERED?
YES
NO
APPROVE PRELIMINARY ESTIMATE
EXCEEDS $2000?
NO
YES
ASSESSOR HEADQTR.
AUTHORIZE ON-SITE INVESTIGATION
INCIDENT COVERED?
YES
NO
DETERMINE PRICE OF DAMAGE
APPROVE CASE
SENIOR ASSESSOR
APPROVE CLAIM
CHECK CLAIMS CASE
DOCUMENT CONTROL
SEND PAYMENT TO CLIENT
CLOSE CASE
FILE CLAIM

BP Modeling

S1)
POLICY NOT VALID:
100 x .05 = **5**



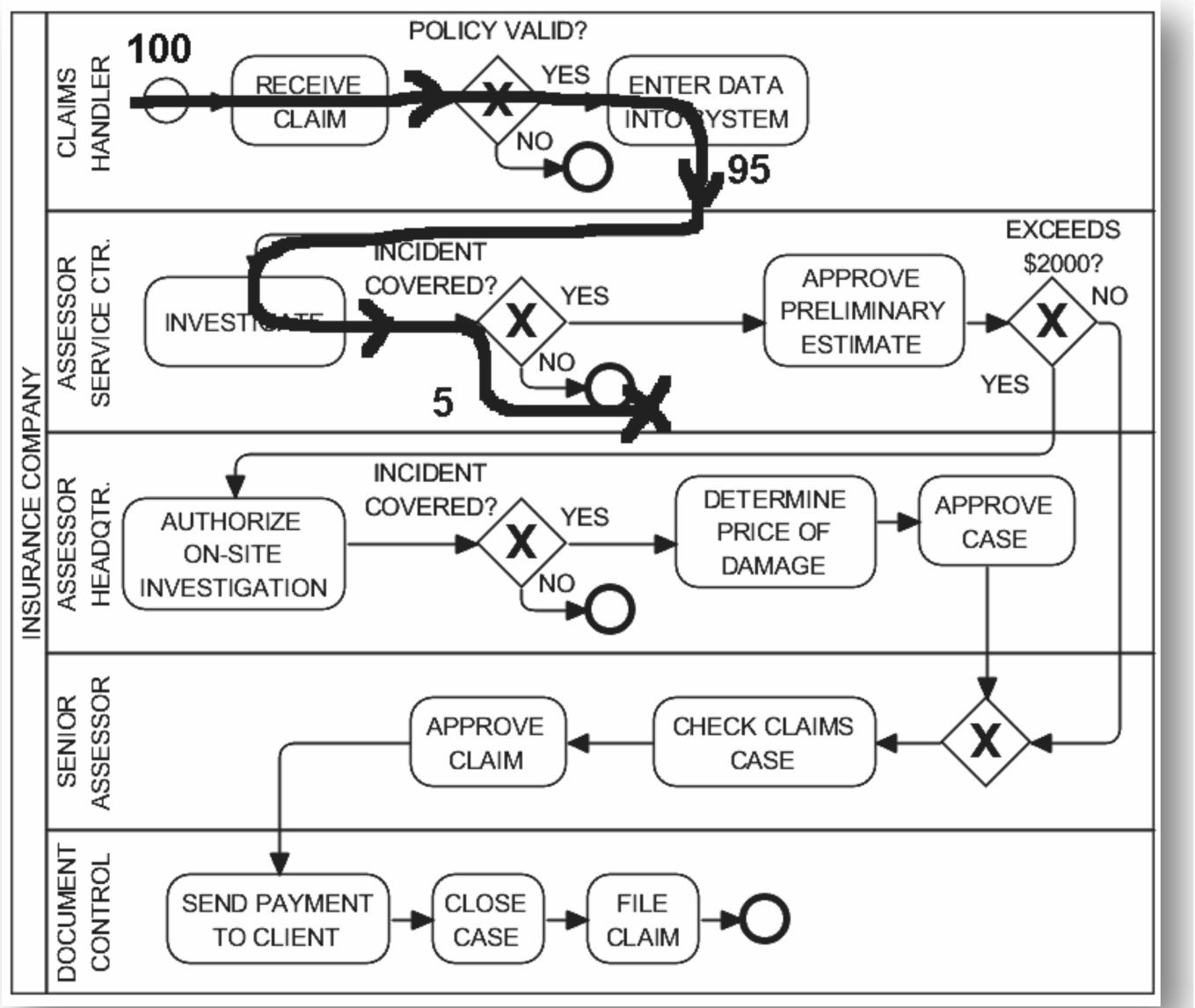
BP Modeling

S2)
POLICY VALID &
INCIDENT NOT COV.:
100 x .95 &
95 x .05 = **5**

Note that the exact result is 4.75 ≈ 5.

Given a number of tokens arriving to a decision node:

(i) apply the percentage covered by the decision variable
(ii) add the operator “&”
(iii) report the next number of tokens rounded to the nearest integer.



BP Modeling

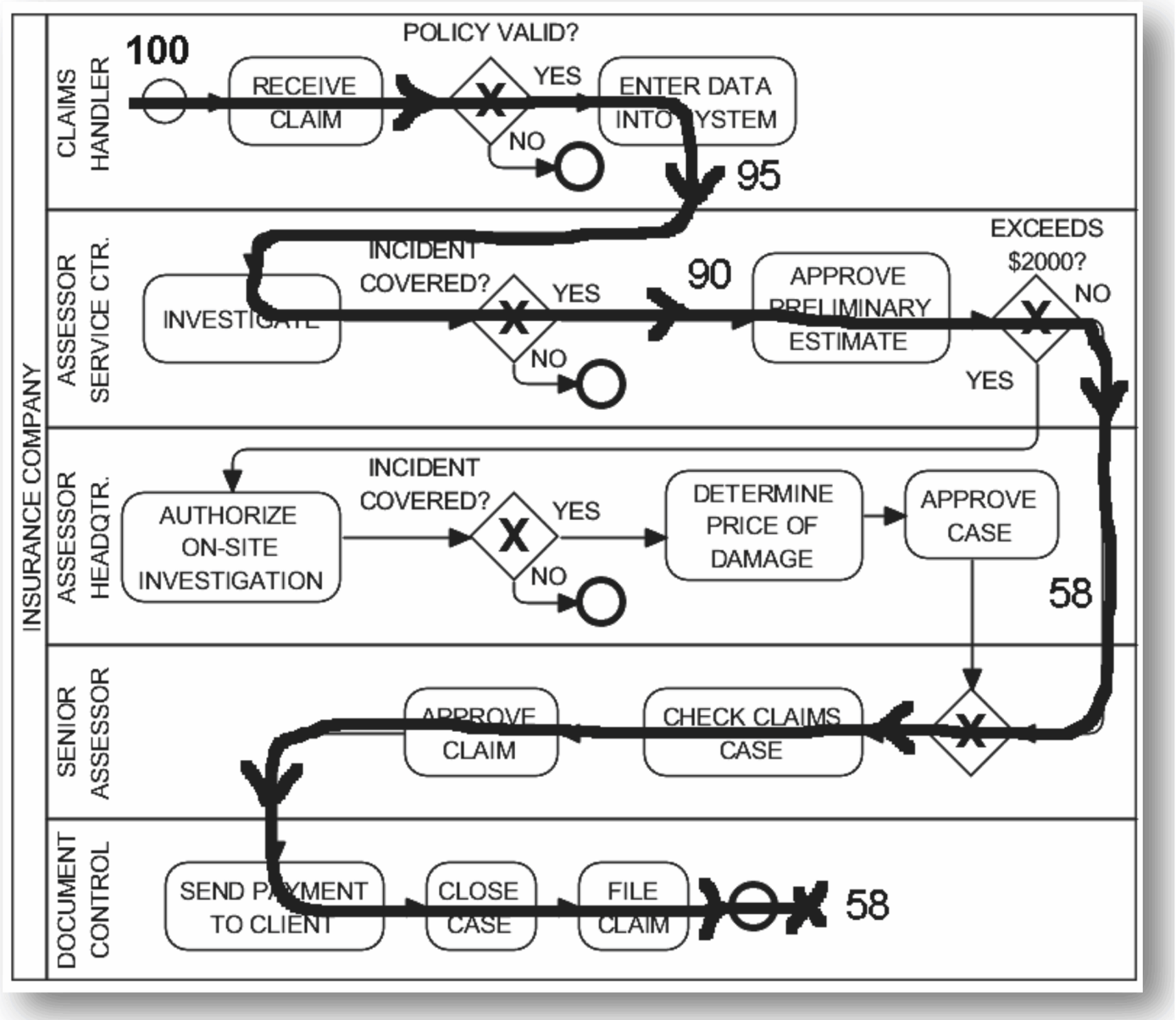
S3)
POLICY VALID &
INCIDENT COV. &
NOT EXCEEDS 2K:
100 x .95 &
95 x .95 &
90 x .65 = **58**

Note that the exact result is 58.5 ≈ 58.

If the decimal part is .5, you can round-up | round-down, thus adding | removing a half token.

But in the opposite scenario you have to make the opposite choice, removing | adding the half token.

This to avoid removing | adding a half token in both scenarios.

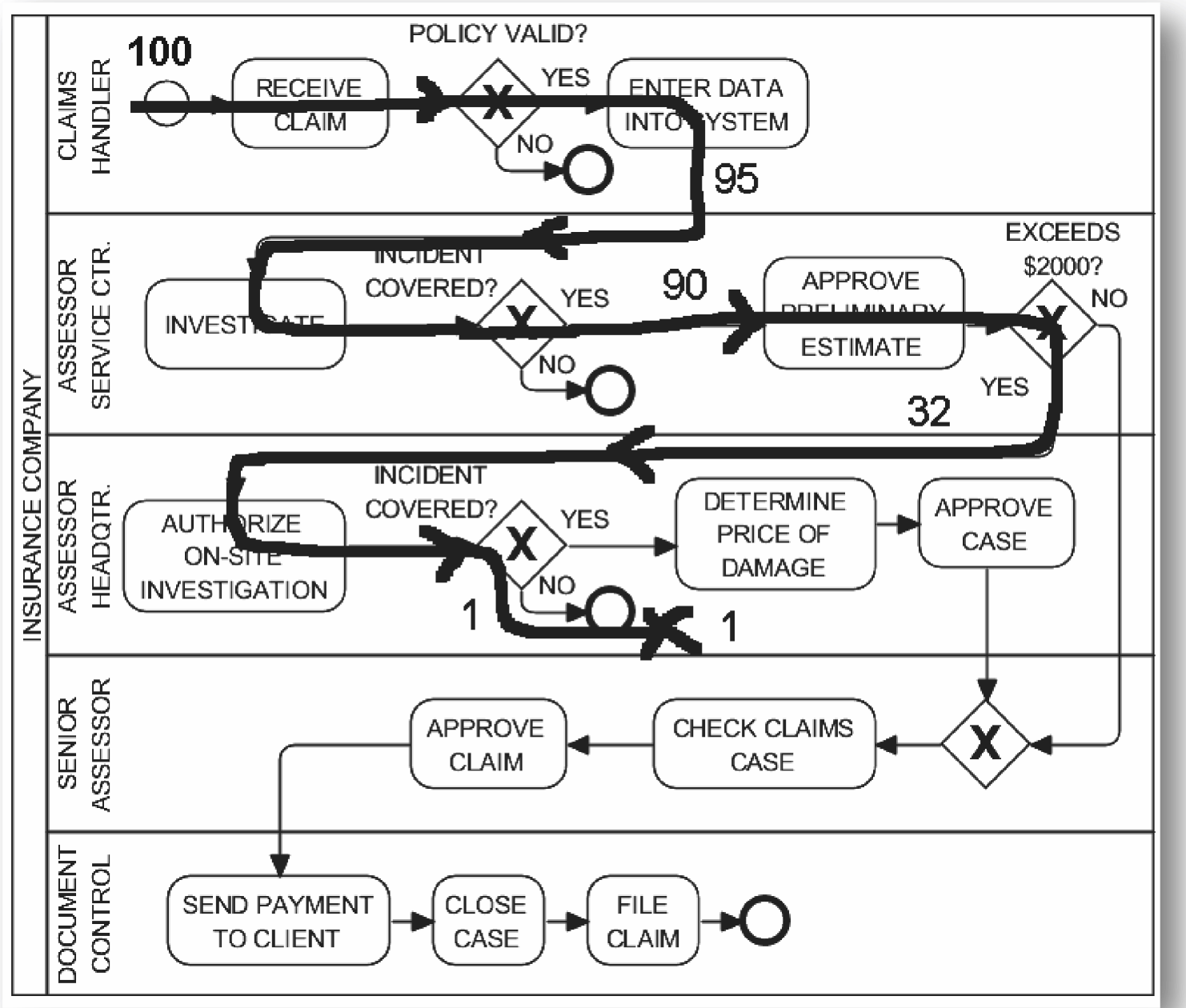


In brief: include the half-token in a unique scenario, to avoid a whole token disappears/appears.

BP Modeling

S4)
POLICY VALID &
INCIDENT COV. &
EXCEEDS 2K &
INCID. NOT COV.:
100 x .95 &
95 x .95 &
90 x .35 &
32 x .02 = **1**

Note that the exact result of 90 x .35 = 31.5 ≈ 32.



BP Modeling

S5)
POLICY VALID &
INCIDENT COV. &
EXCEEDS 2K &
INCIDENT COV.:
100 x .95 &
95 x .95 &
90 x .35 &
32 x .98 = **31**

S1 + S2 + S3 +
S4 + S5 =
5 + 5 + 58 +
1 + 31 = **100**