Business Process Modeling

SoftEng
http://softeng.polito.it

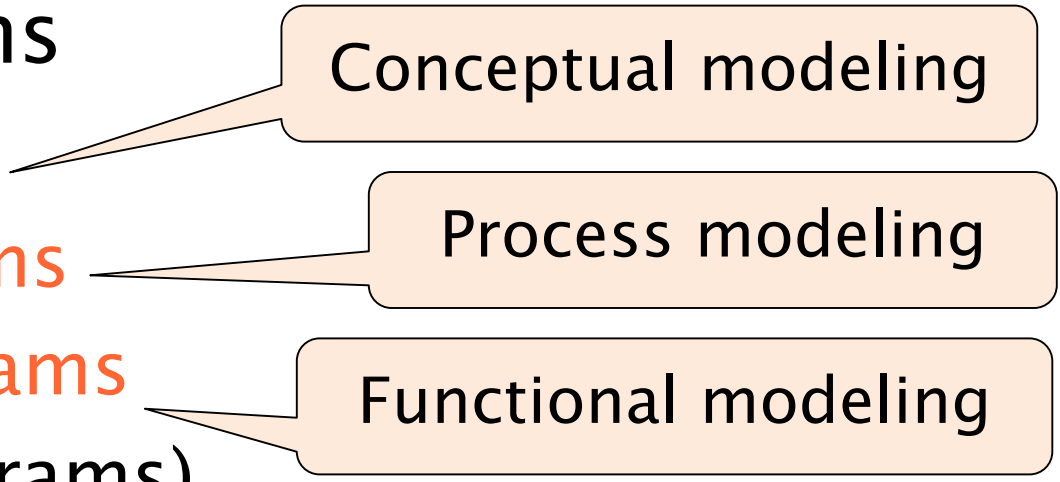
Version 13/10/2016

BP Aspects

- Process flow
  - Process modeling
    - UML Activity Diagrams
    - BPMN
- Information
  - Conceptual modeling
    - UML Class diagrams
    - (Entity-Relationships)
- Interaction
  - Functional modeling
    - Use cases

UML

- Unified Modeling Language
- Standardized by OMG
- Several diagrams
  - Class diagrams
  - Activity diagrams
  - Use Case diagrams
  - (Sequence diagrams)
  - (Statecharts)



Objective

- Describe, as precisely as possible, a process (or workflow)
- Communicate, document, analyze, validate the workflow
- Implement (execute) it
  - Only formal notations allow this step

Issues

- Formal notations
  - Executable
  - But model can be very complex for high level of detail
- Semiformal
  - Not executable
  - But can be starting point for high level analysis

Notations

- Formal
  - UML Activity Diagrams
  - BPMN
  - BPEL
- Semi formal
  - IDEF0
  - Data Flow Diagrams

Process Modeling

UML ACTIVITY DIAGRAM

Goal

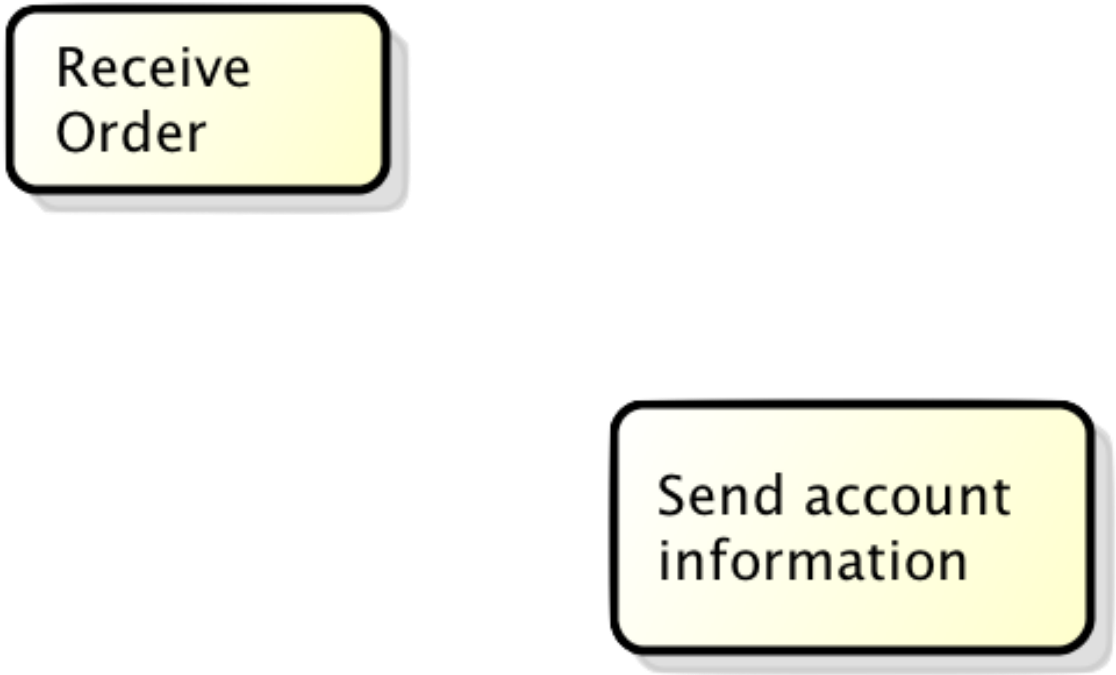
- Capture
  - Activities
  - Rules
  - Responsibilities

Activity Diagram

- Extension of Statechart Diagram used to represent temporal sequence of activities and data flow
- Used to represent workflow process, or the inner service logic of an algorithm or function, process
- Parallel process representation and synchronization (fork – join)

Action

- Represents a task or operation that is performed by either a human or the IS



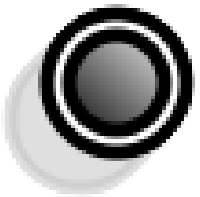
Terminal nodes

- Initial node
  - Represents the starting point of the process execution
  - Create a new token

- Final node
  - Indicate that the processing has completed
  - Destroy all tokens

Semantics

- A token flows through the diagram
- The token is created at the initial node
- The token comply with the process rules
- The token is eventually destroyed at end node



Basic patterns

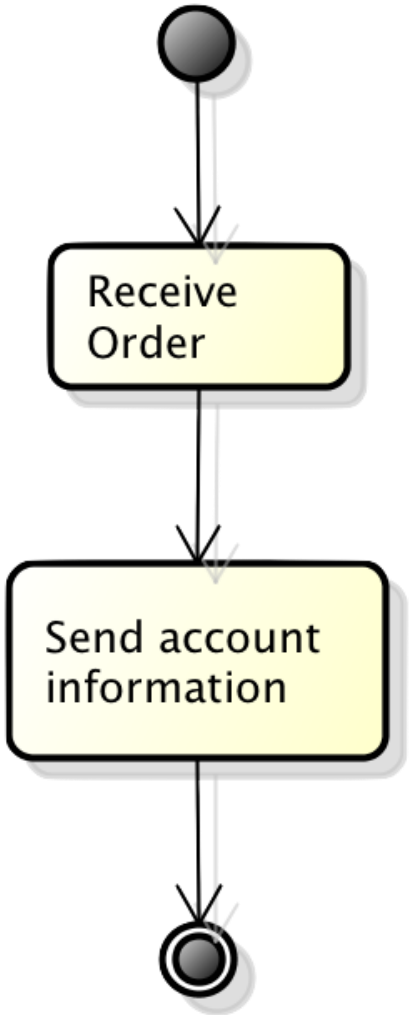
- Sequence
- Parallel split
- Synchronization
- Exclusive choice
- Merge
- Multiple choice

Sequence

- An action in a process is enabled after the completion of a preceding action
  - Aka serialization
- It is the essential building block
  - Can be used to build a series of consecutive actions that take place in turn one after the other

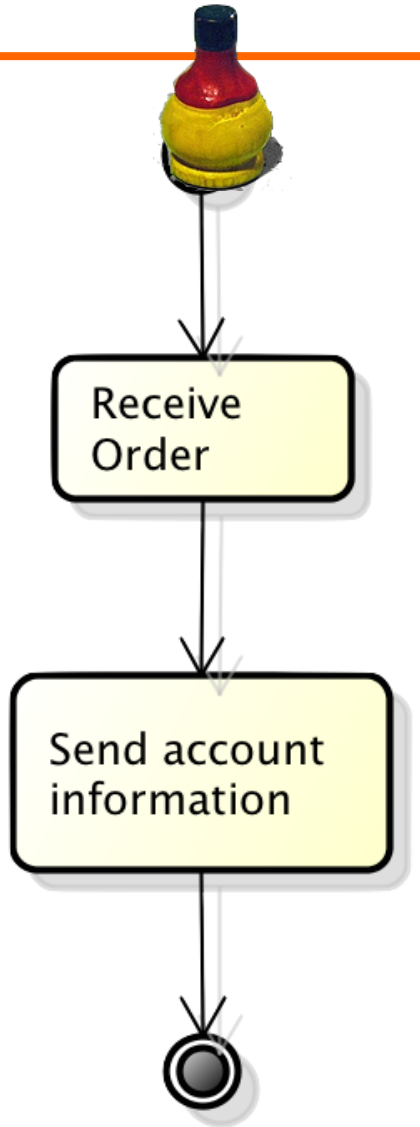
Sequence

- The arc determines the order of execution



Sequence – Semantics

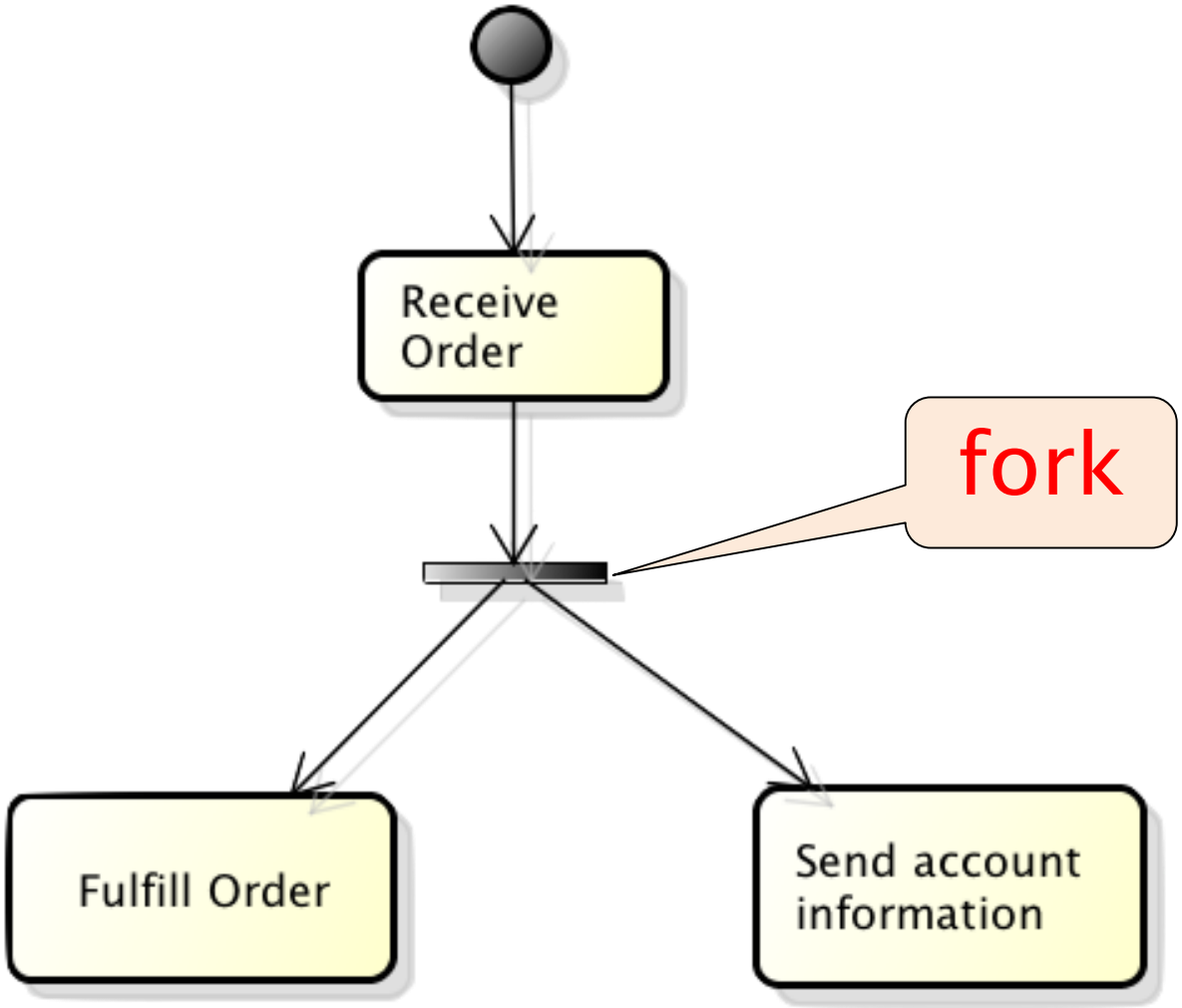
- A token flows through the diagram
- Following the arcs
- Stopping at actions
  - Performing actions



Parallel split

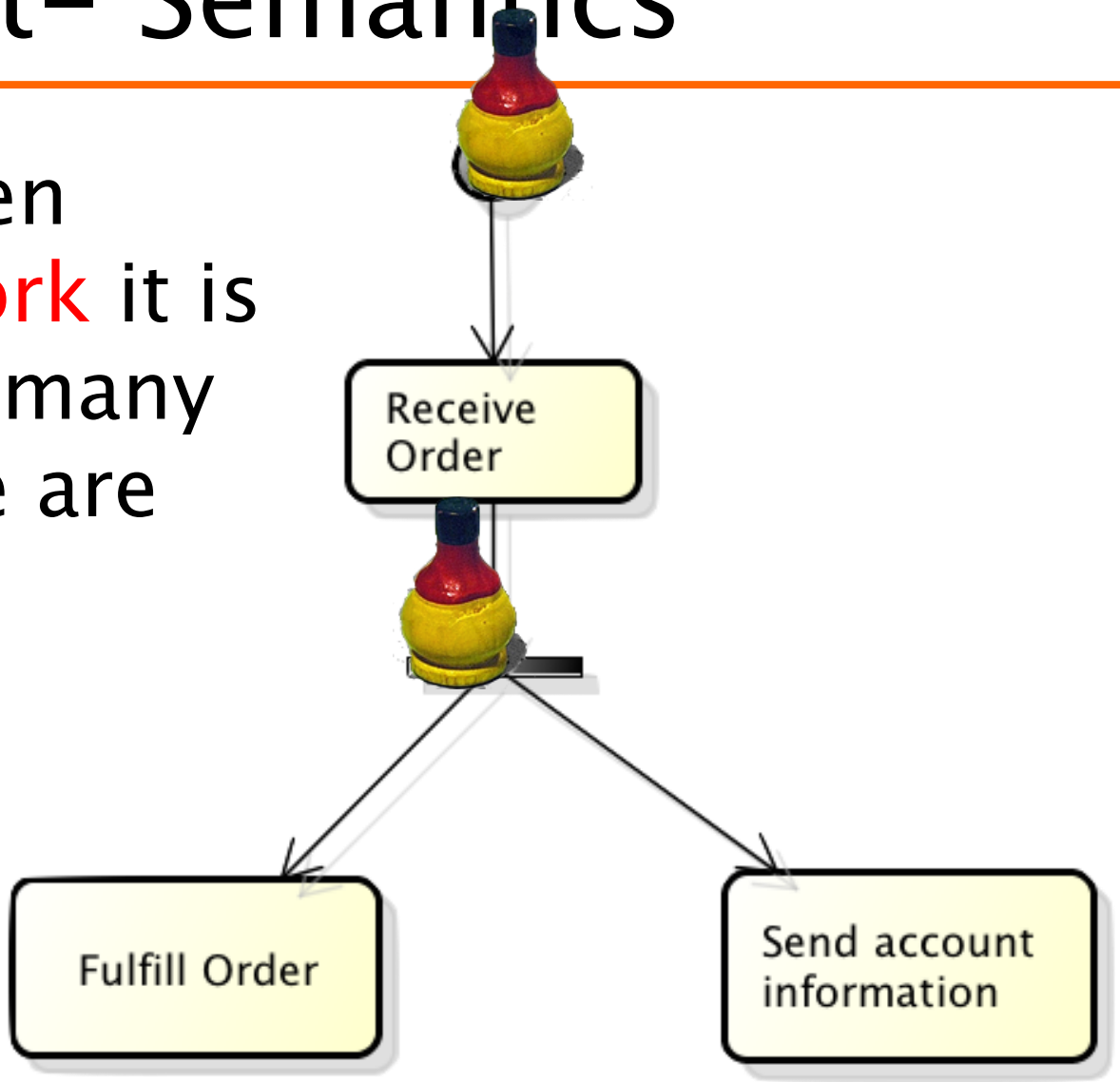
- From a certain point on a thread diverges into several parallel threads that can be executed concurrently
  - Aka fork, AND-split
- Represents
  - Actions taking place at the same time (concurrently)
  - Actions being performed without any specific order
    - Possibly even serialized

Parallel split



Parallel split- Semantics

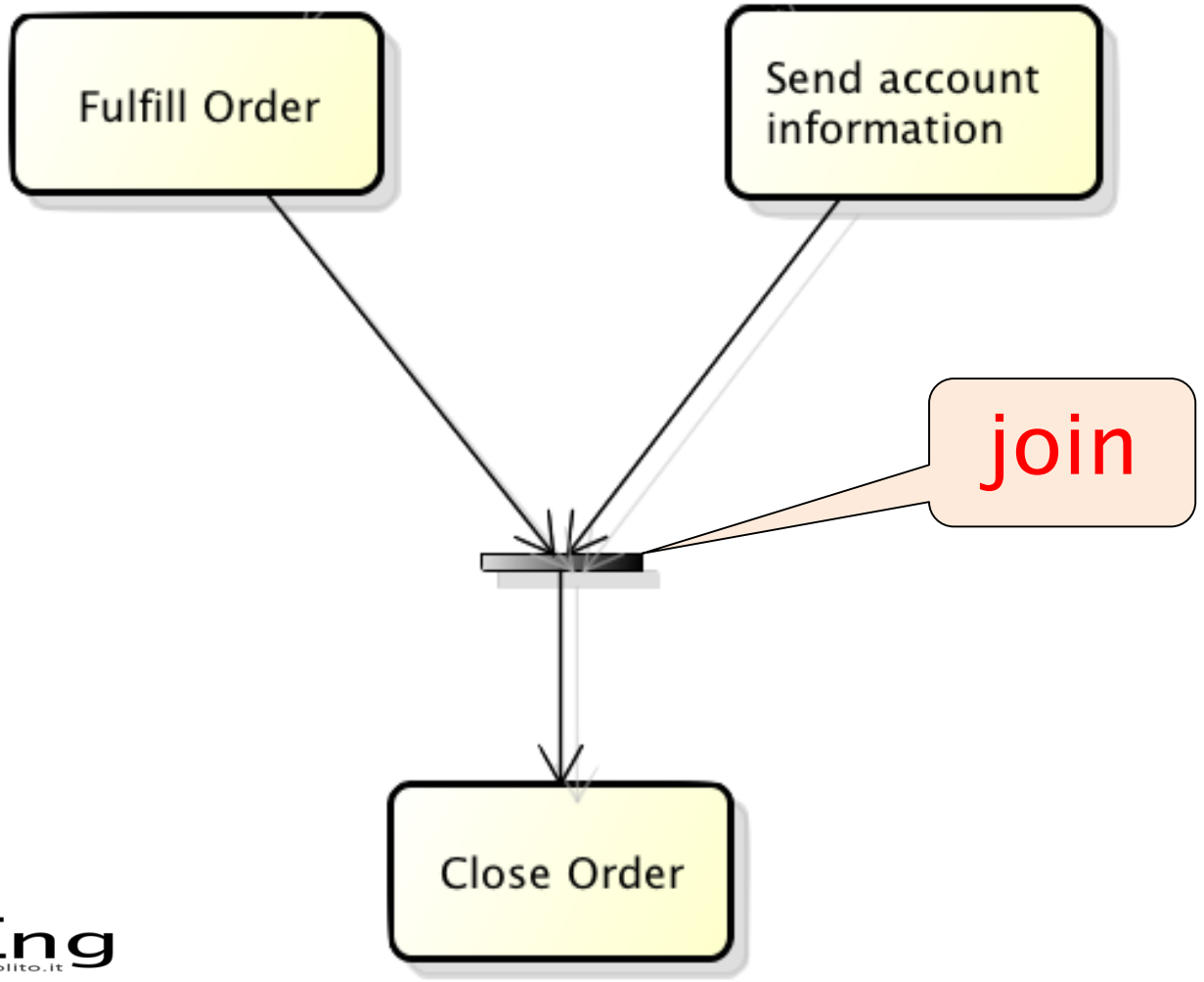
- When the token reaches the fork it is duplicated as many times as there are outgoing arcs



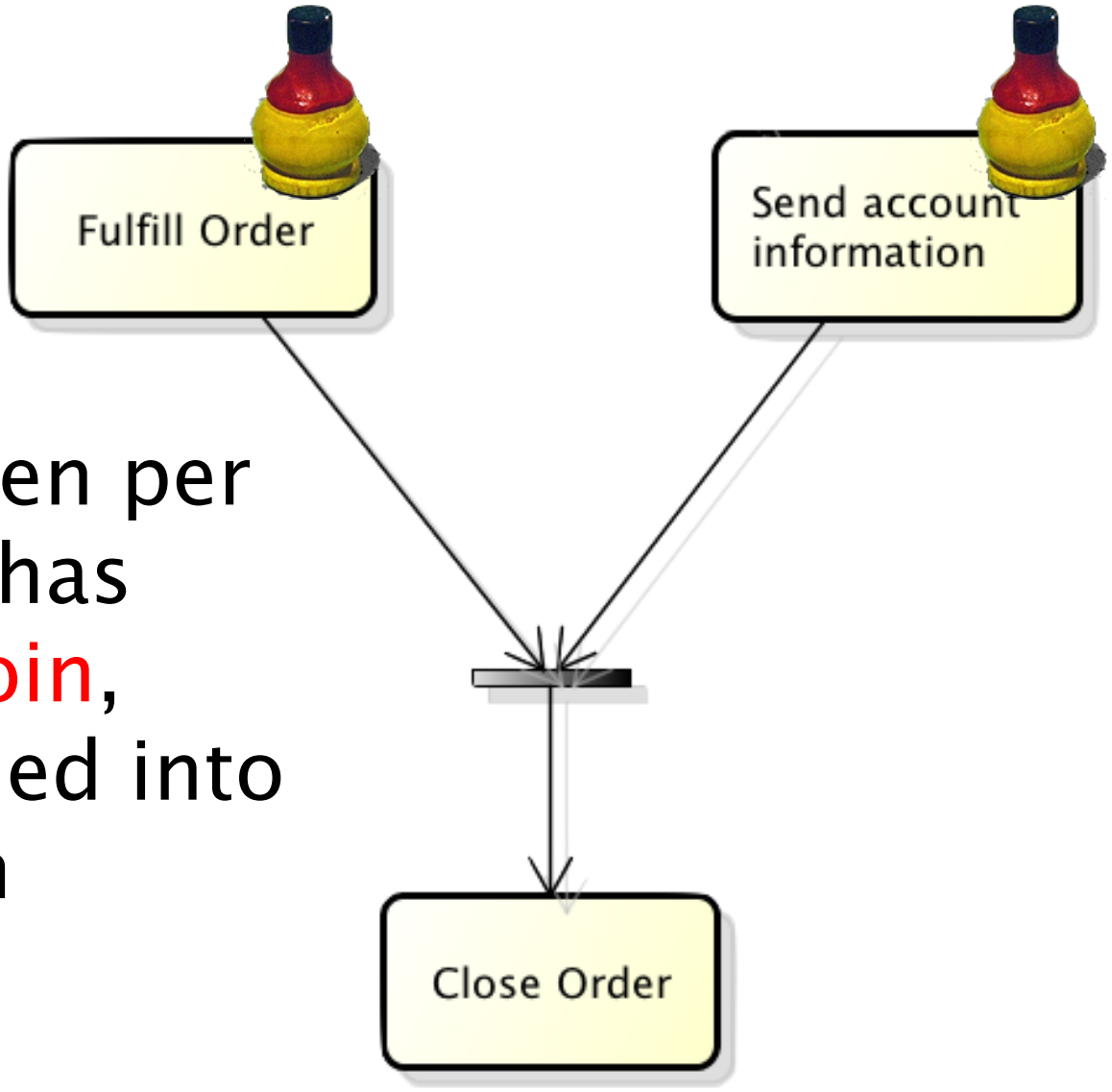
Synchronization

- Define a synchronization point or rendezvous
  - After a group of actions have been executed in parallel
- Before proceeding with further activities all the previous one must be complete

Synchronization



Synchronization- Semantics

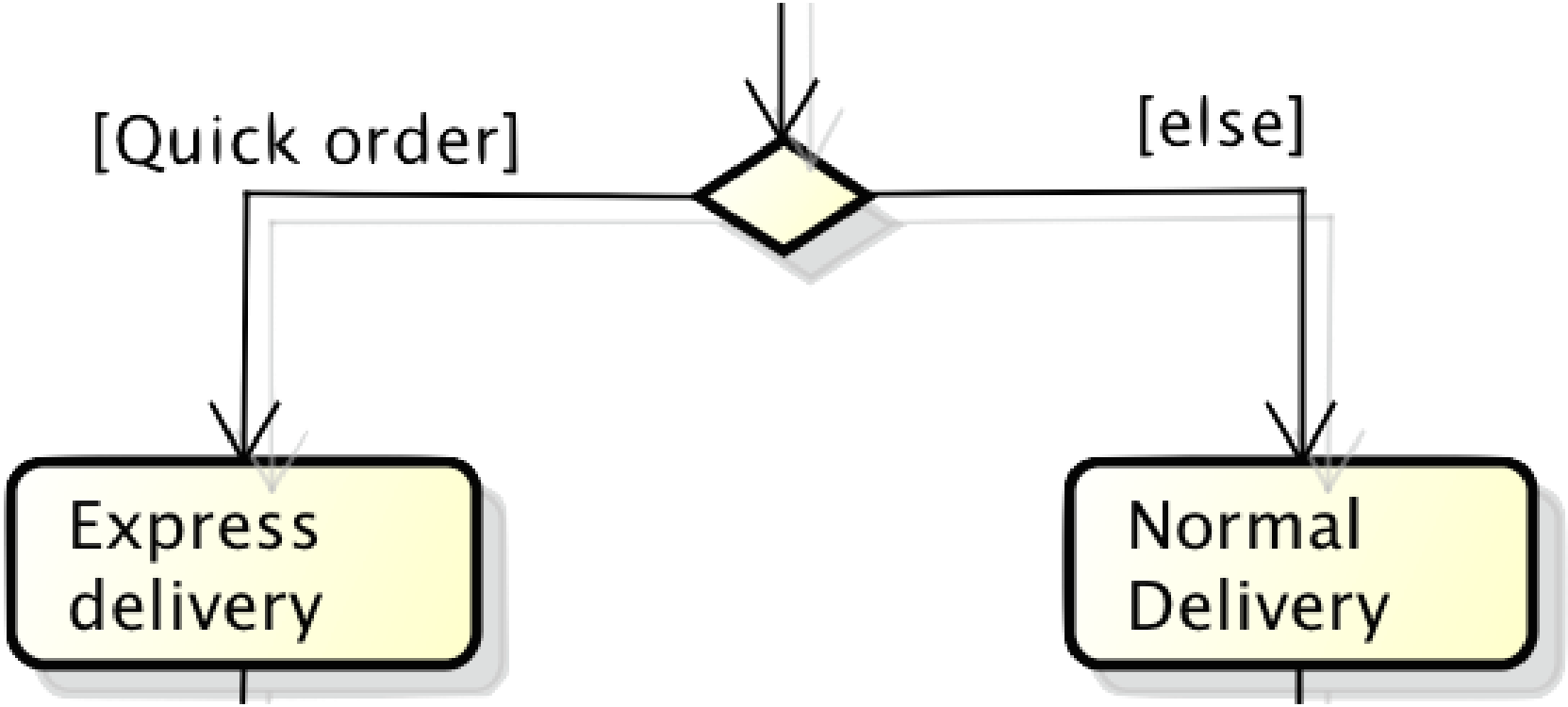


- When one token per incoming arc has reached the join, they are merged into a single token

Exclusive choice

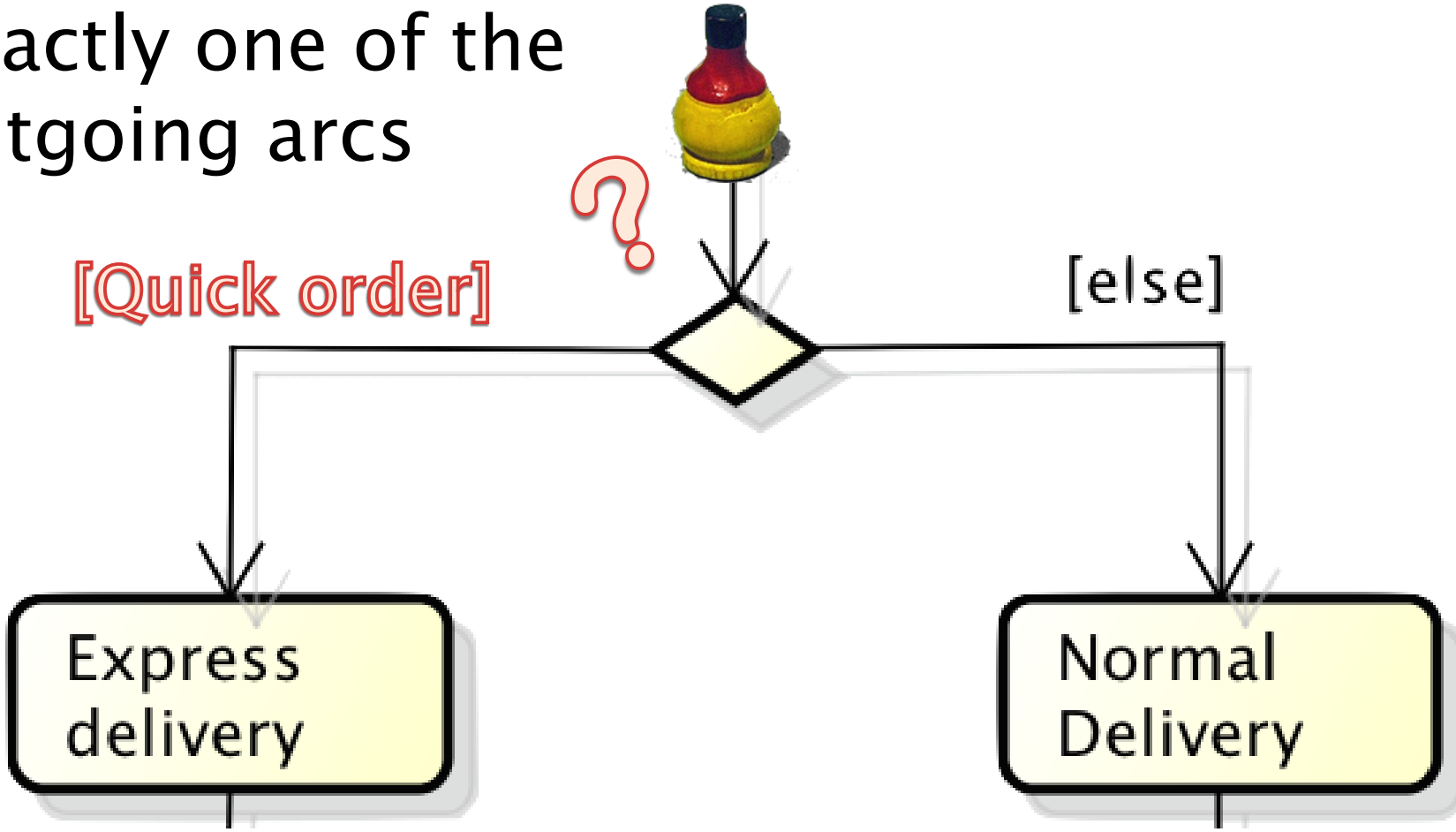
- A diversion of the thread into several alternative paths
  - Exactly one alternative is picked up and followed during execution
  - Aka conditional routing, decision
- Each path is characterized by a guard
  - Represent a condition that, when true, enable the execution of the corresponding path

Exclusive choice



Exclusive choice – Semantics

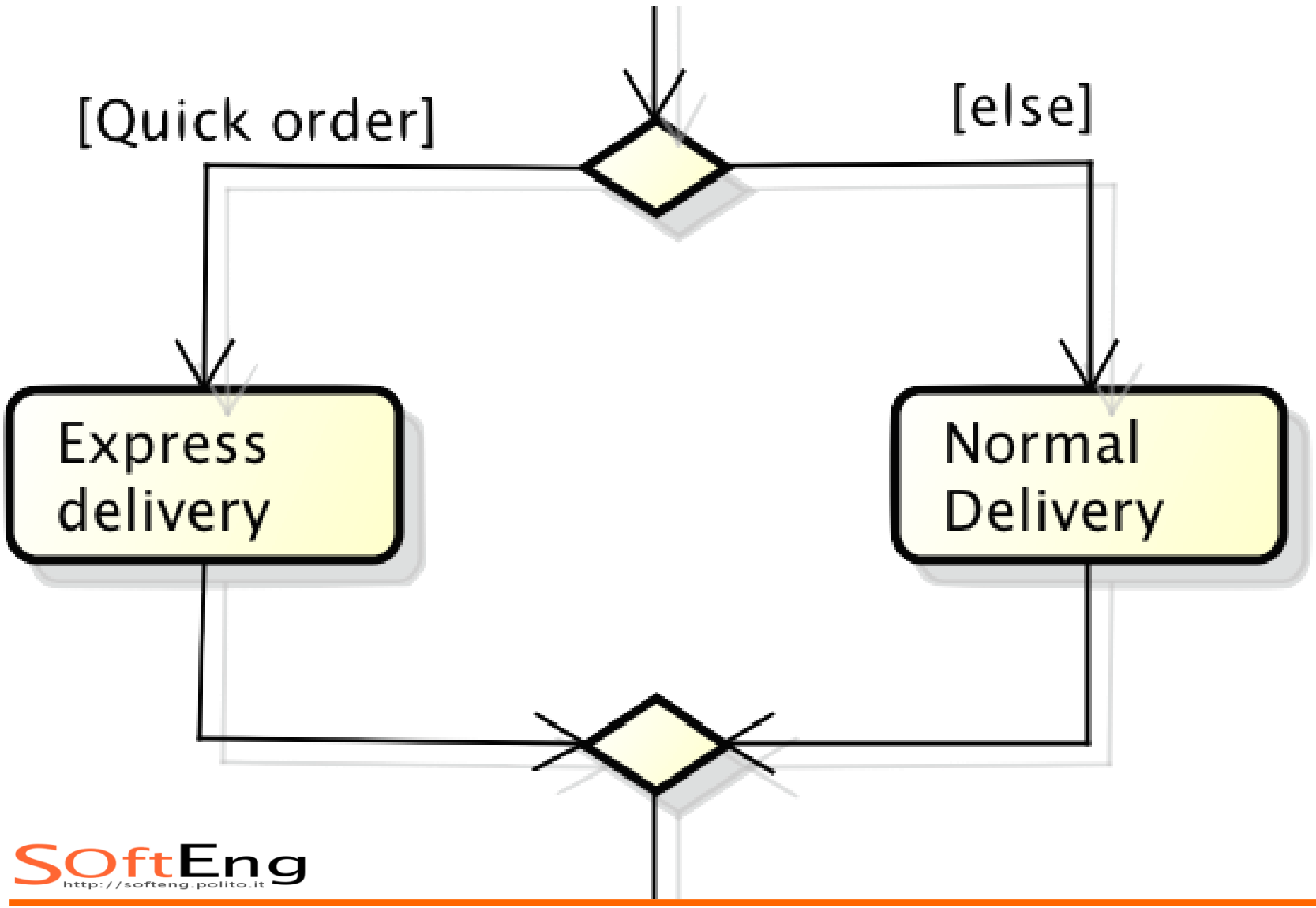
- The token follows exactly one of the outgoing arcs



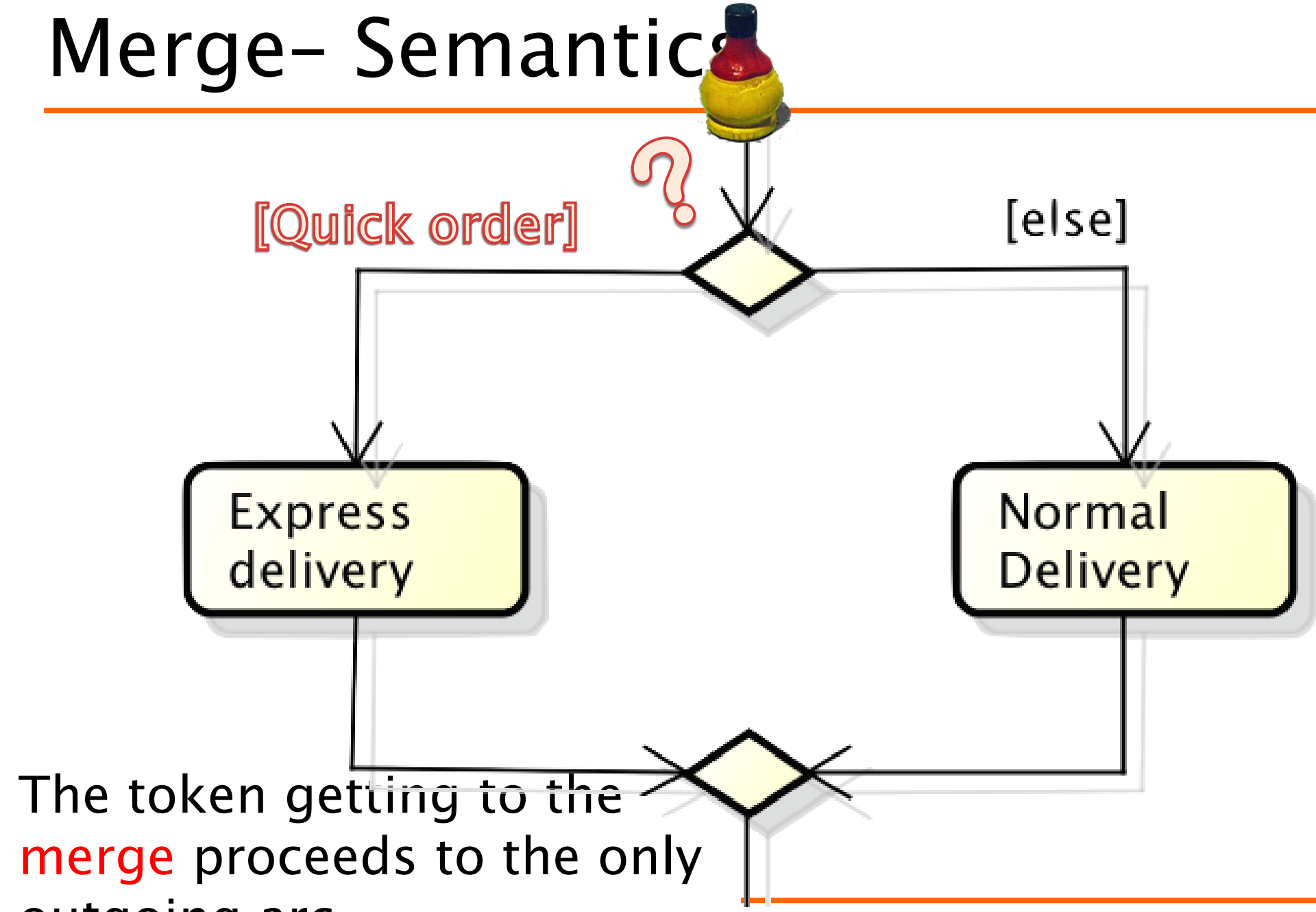
Merge

- The convergence of two or more threads into a single one
  - Any incoming thread activates the outgoing path
  - Aka join
- No synchronization is performed

Exclusive choice



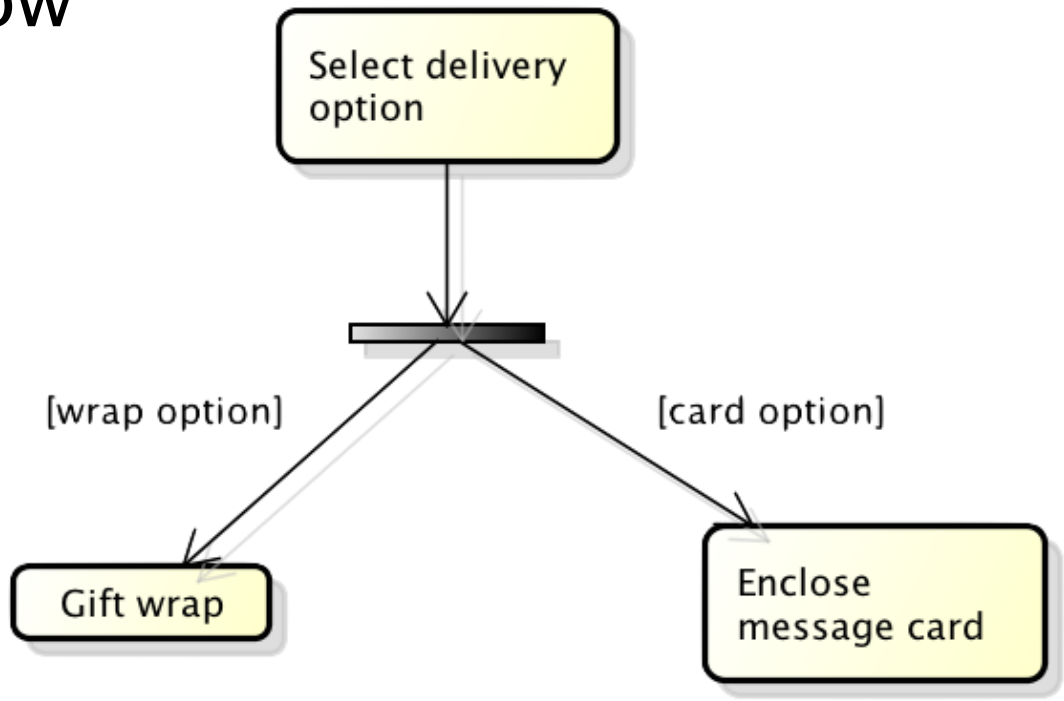
Merge- Semantics



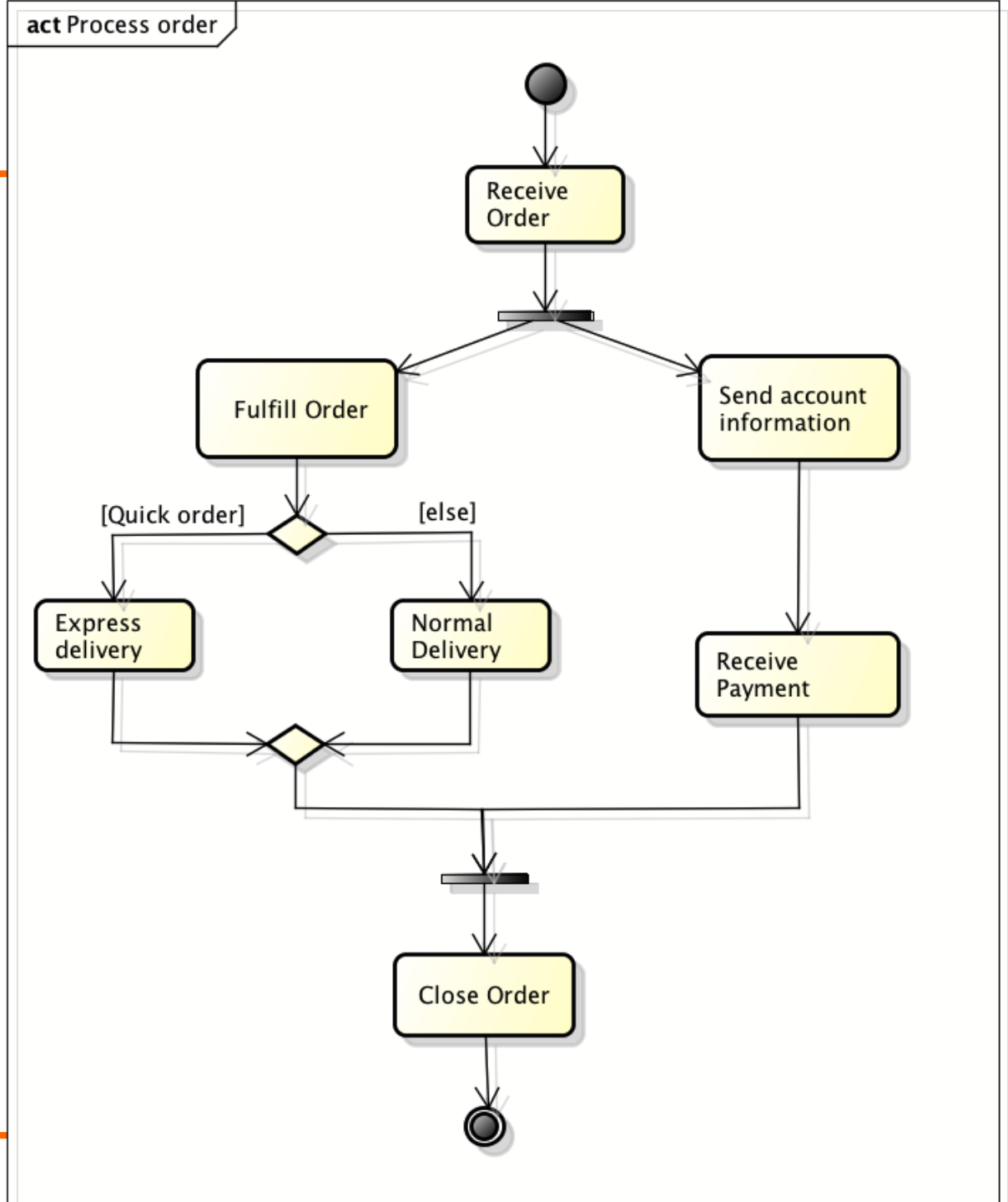
The token getting to the merge proceeds to the only outgoing arc

Multiple choice

- When several paths are available it is possible to chose one or more of them
  - If no path is chosen, we have an abnormal stop to the flow



Example


act Process order
Receive Order
Fulfill Order
Send account information
[Quick order]
[else]
Express delivery
Normal Delivery
Receive Payment
Close Order

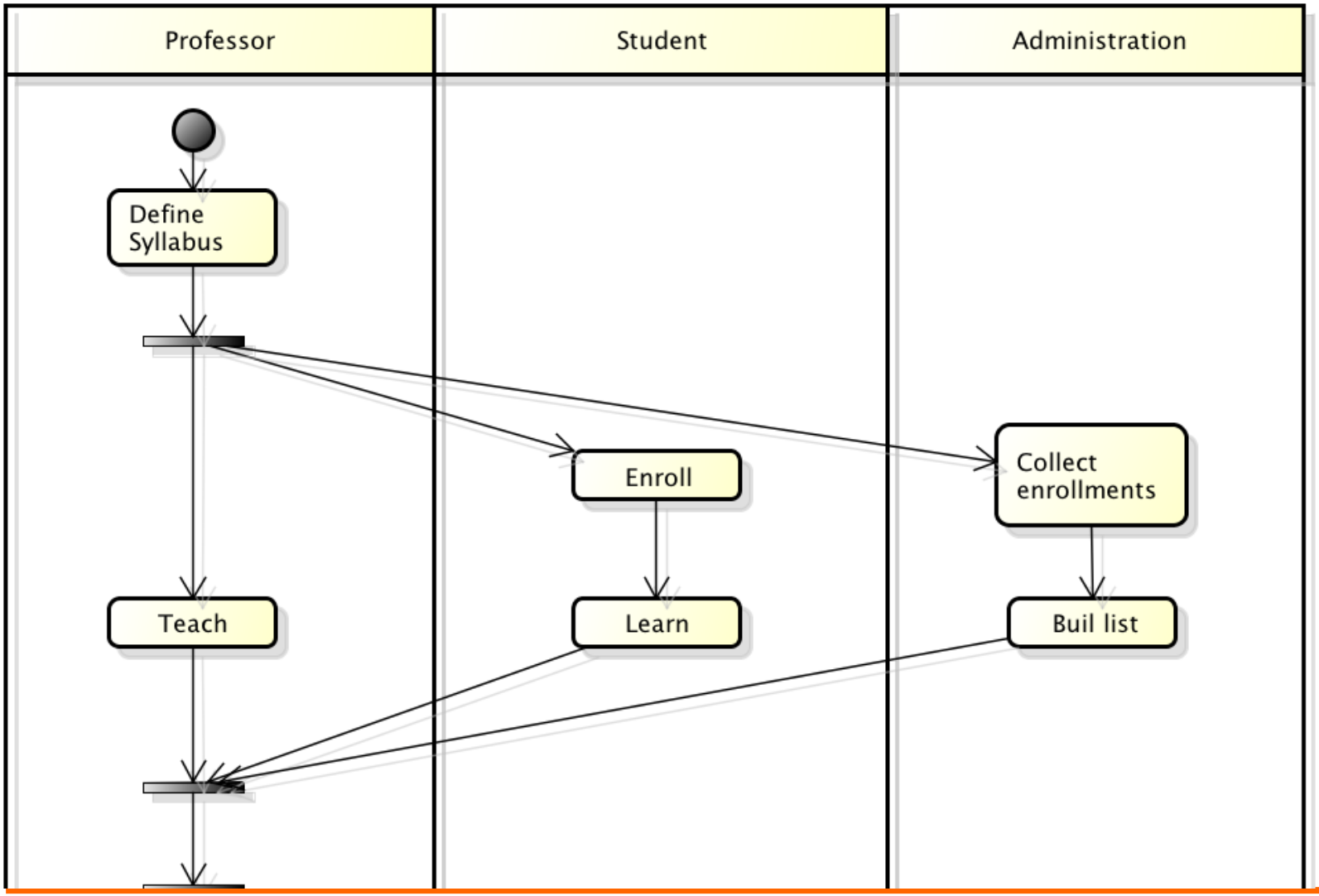
Structured processes

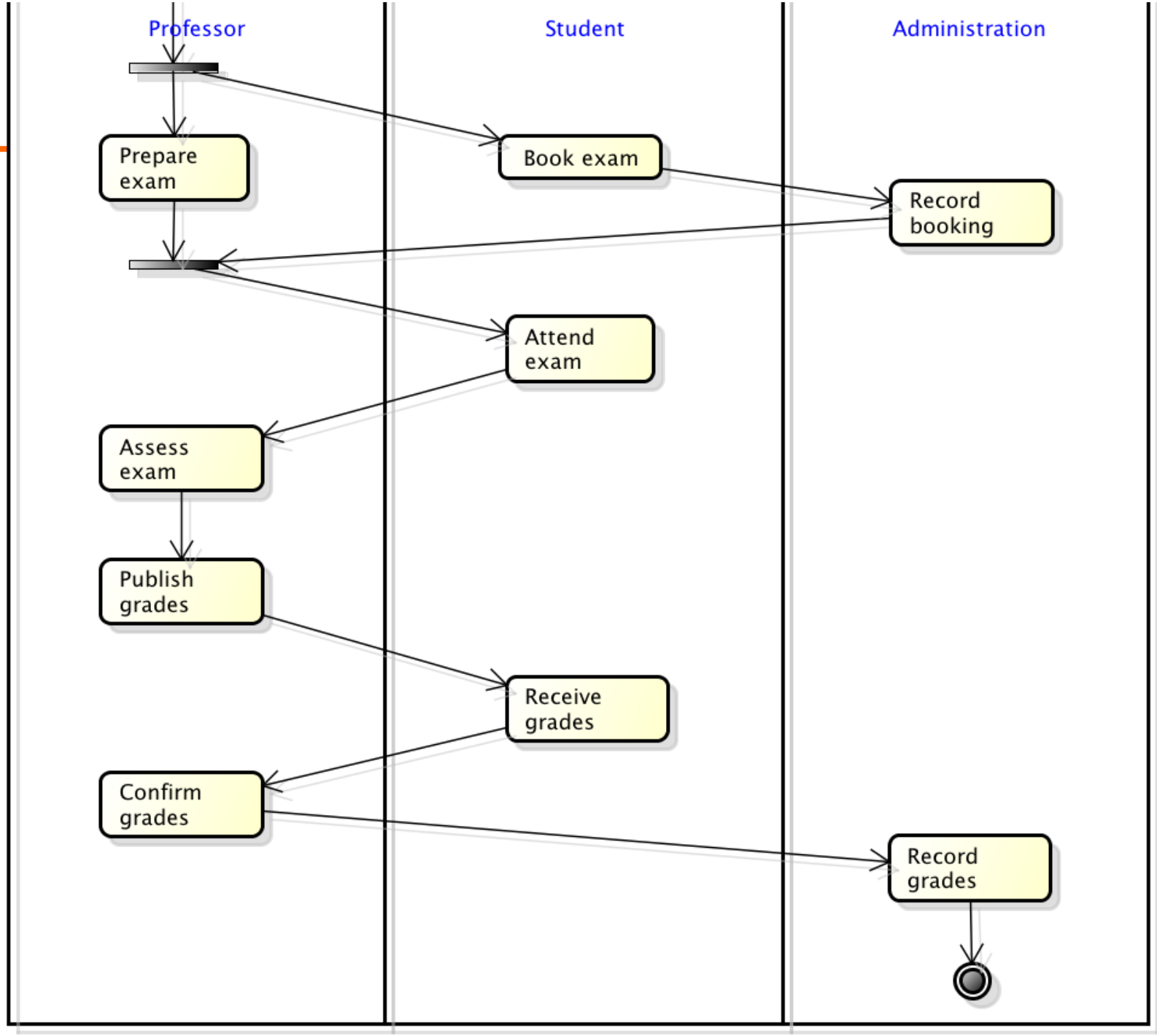
- Each action has exactly one input flow and one output flow
- Fork and Join must be coupled
- Decision and Merge must be coupled

Swimlanes

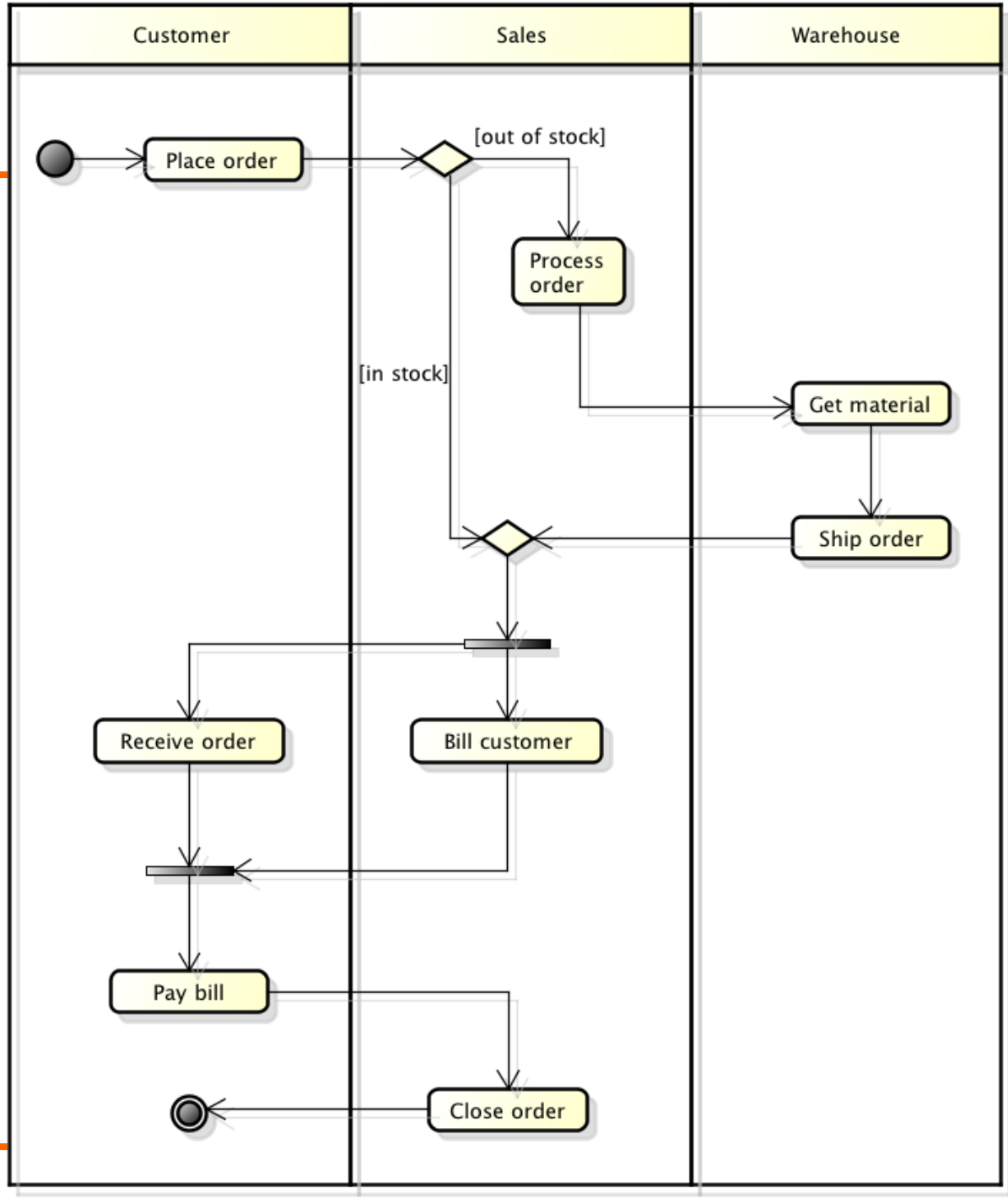
- Actions can be responsibility of different actors or roles
- A swimlane groups together all the activities of a specific actor
  - Assigning responsibilities is not always required
  - Typically it represents a refinement step

Swimlanes – Example




Professor
Student
Administration
Prepare exam
Book exam
Record booking
Attend exam
Assess exam
Publish grades
Receive grades
Confirm grades
Record grades

Swimlanes


Customer
Sales
Warehouse
Place order
[out of stock]
Process order
[in stock]
Get material
Ship order
Receive order
Bill customer
Pay bill
Close order

Prescriptive vs. Descriptive

- Initial goal: understand the currently in place procedure
  - Descriptive
- Next goal: provide guidance for defining IS-supported procedures
  - Prescriptive
- Advice: avoid unnecessary constraints

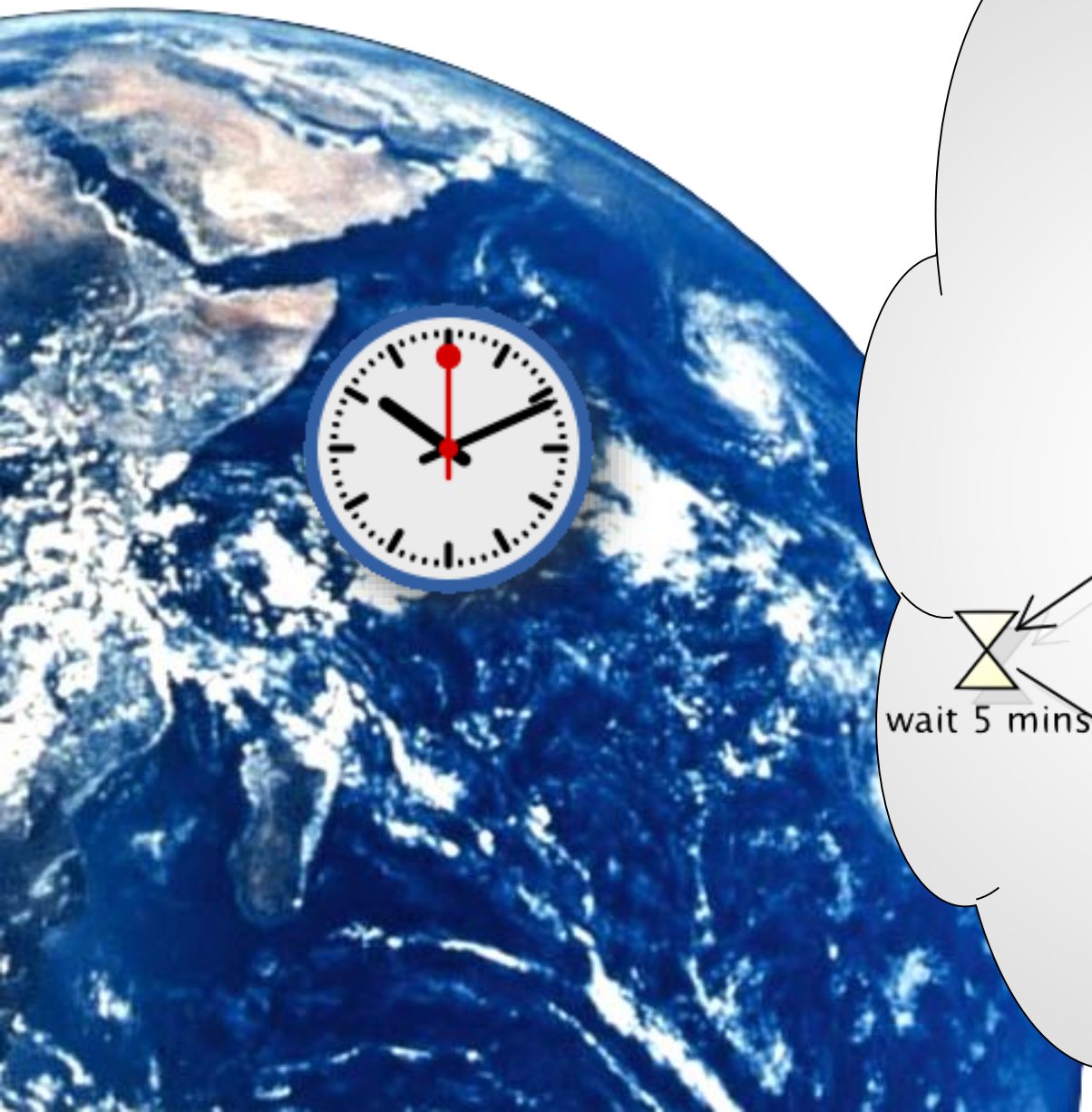
Additional features

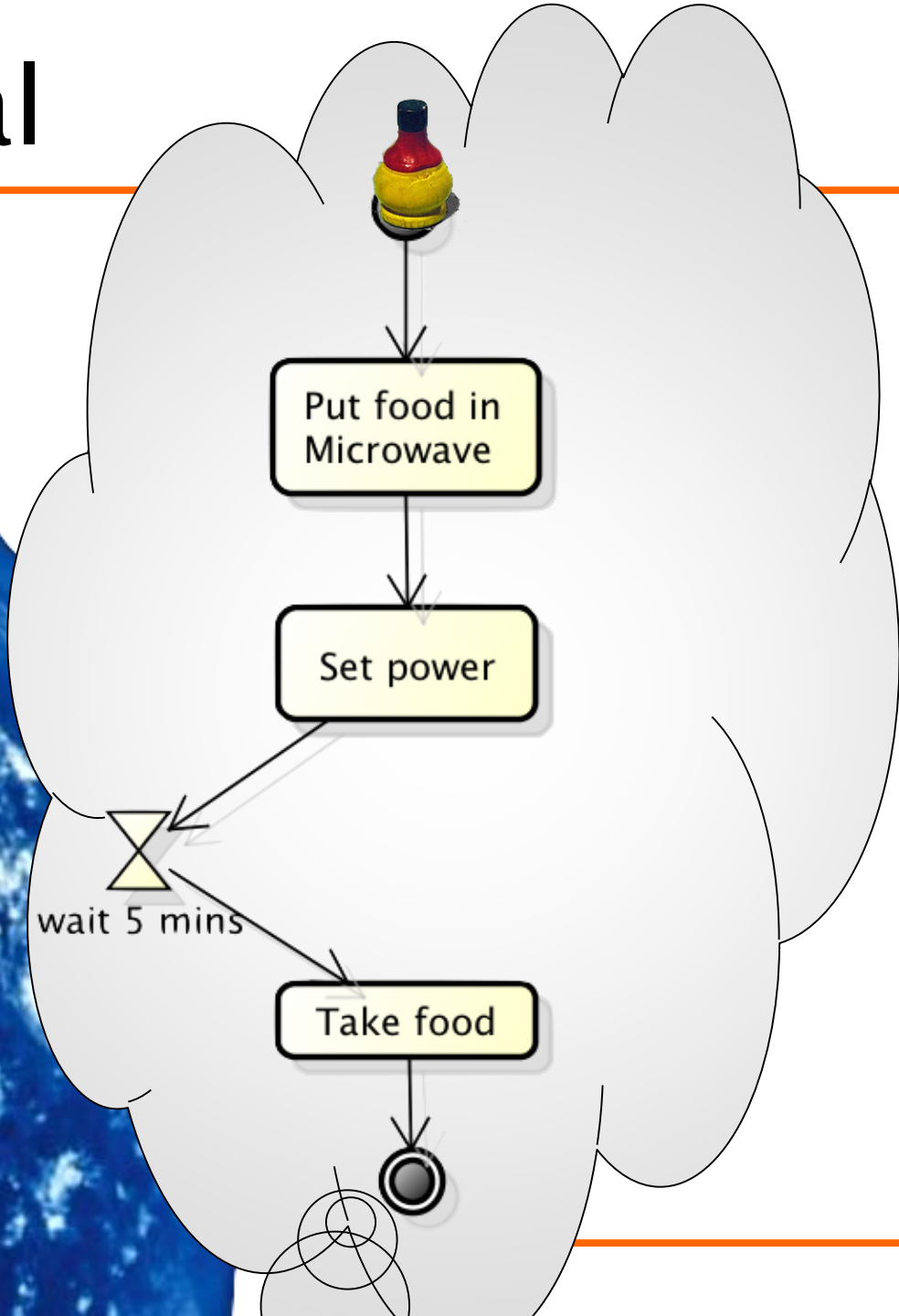
EVENTS, OBJECTS

Signals and Events

- Using an event that origins outside the process as a guard to proceed with the execution of a process
  - Temporal signal
  - Signal acceptance
  - Signal sending

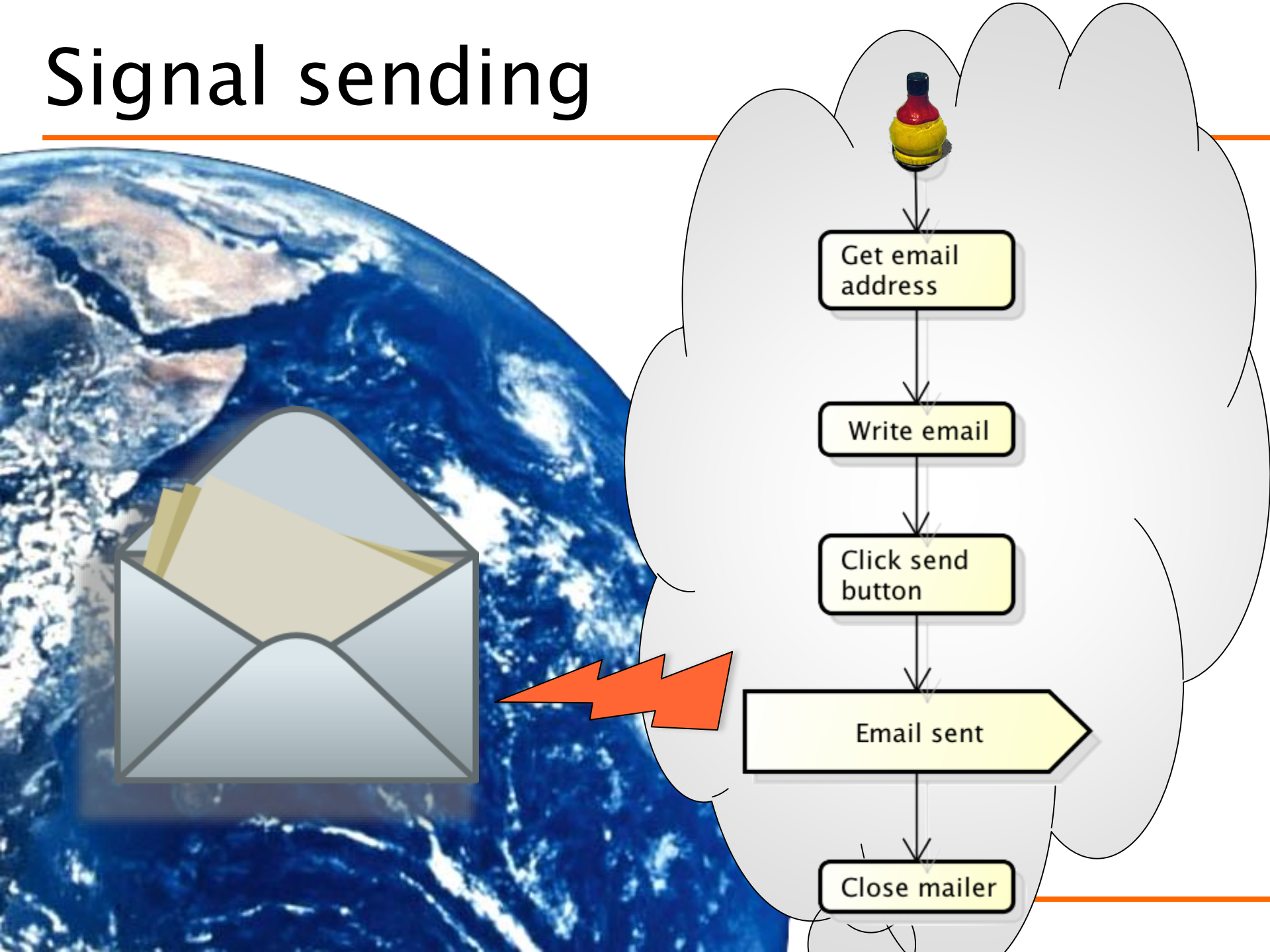
Temporal signal



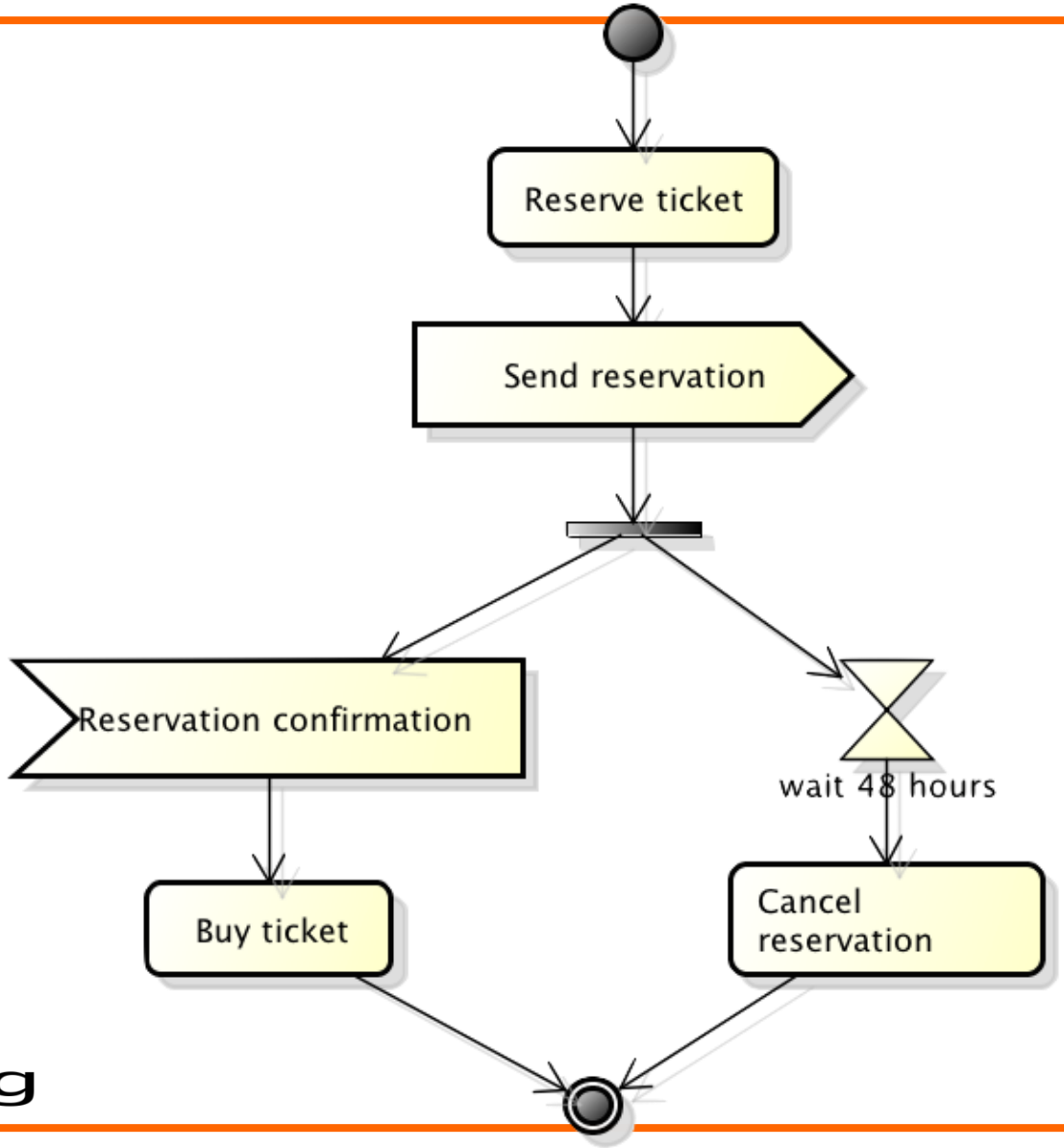


Signal acceptance

Signal sending



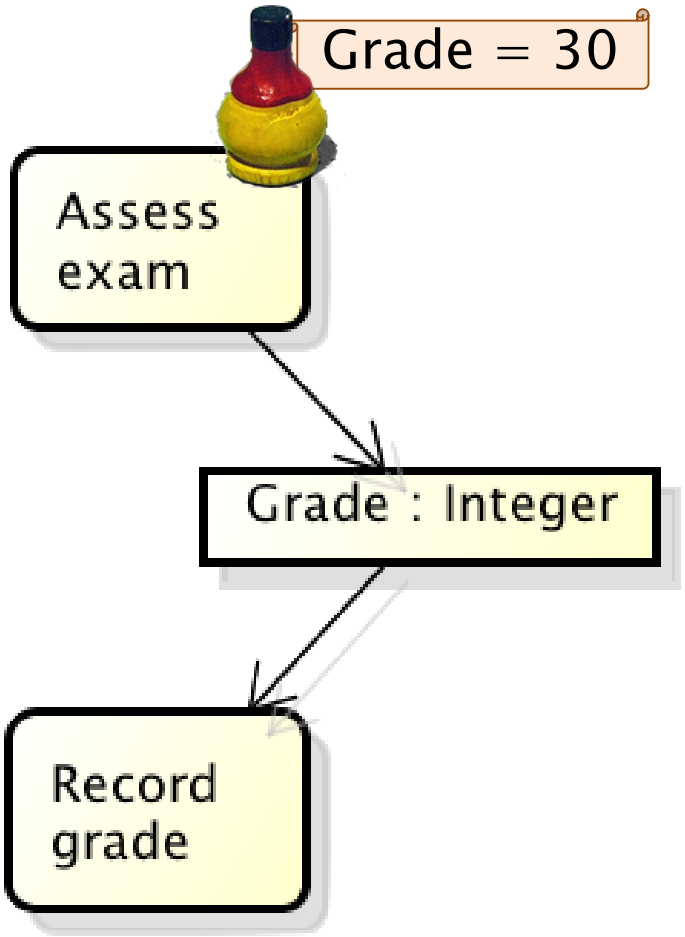
Signals – Example



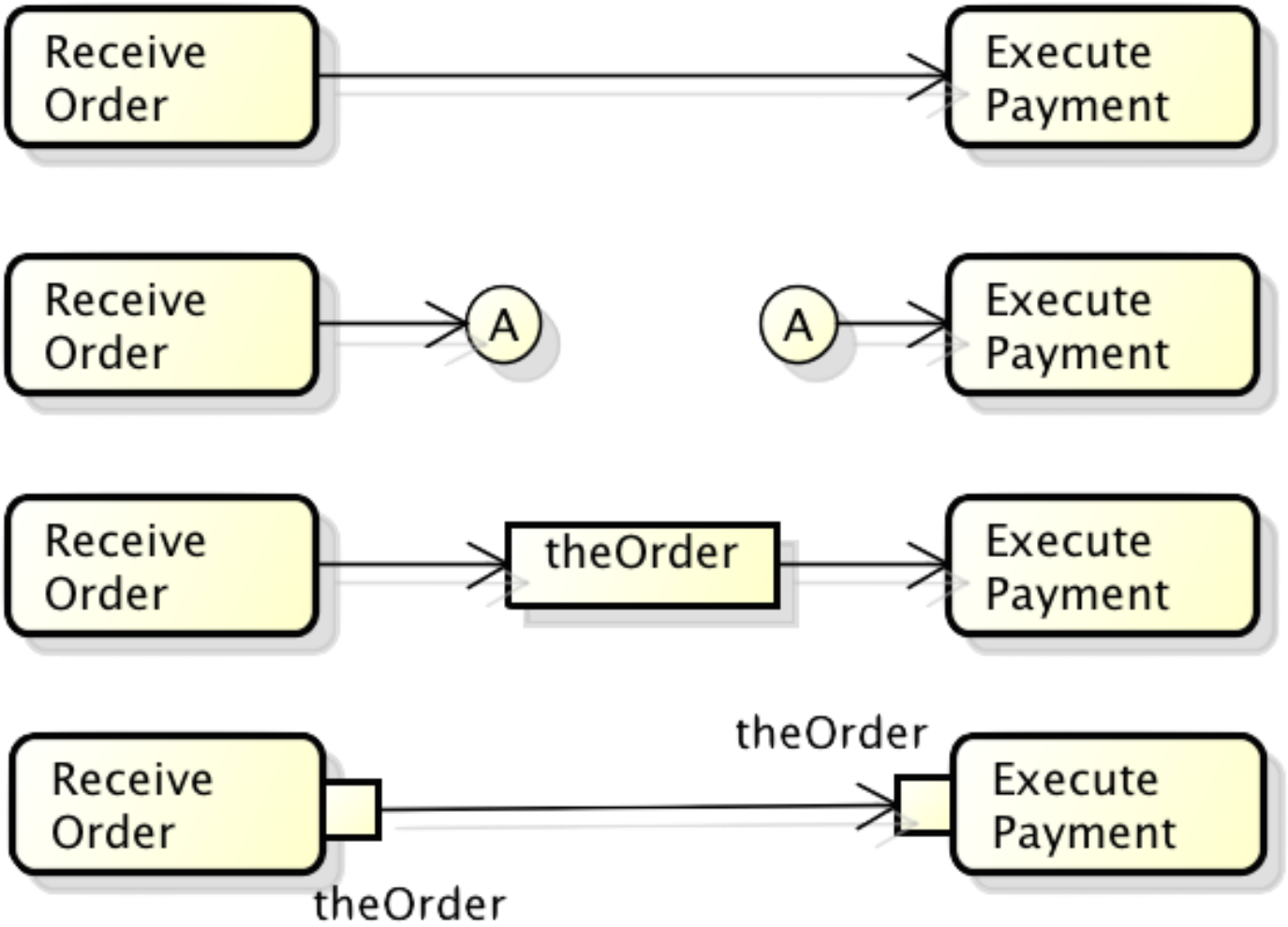
Object nodes

- Indicate that an object node will be available at a specific point in the activity
  - Produced by an action
  - Consumed by an action
- The object is an instance of a class defined in the conceptual model
  - Or possibly a base type

Object nodes



Arcs


Receive Order
Execute Payment
Receive Order
A
A
Execute Payment
Receive Order
theOrder
Execute Payment
theOrder
Receive Order
Execute Payment
theOrder

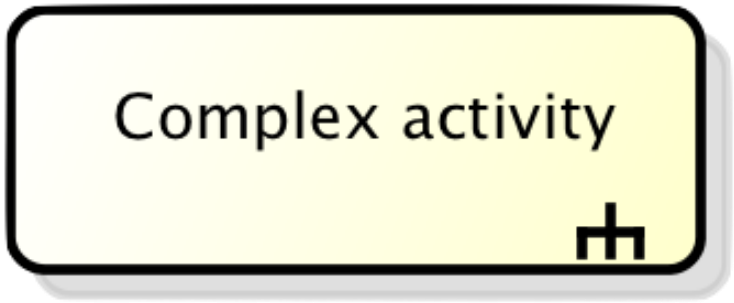
ADVANCED PATTERNS

Complex structures

- Complex activities
- Cycles / loops
- Arbitrary cycles
- Implicit termination
- Multiple choice

Complex action

- Represent a complex (sub-)process a single action
  - Call behavior
- The contents of the complex action can be represented in an additional diagram

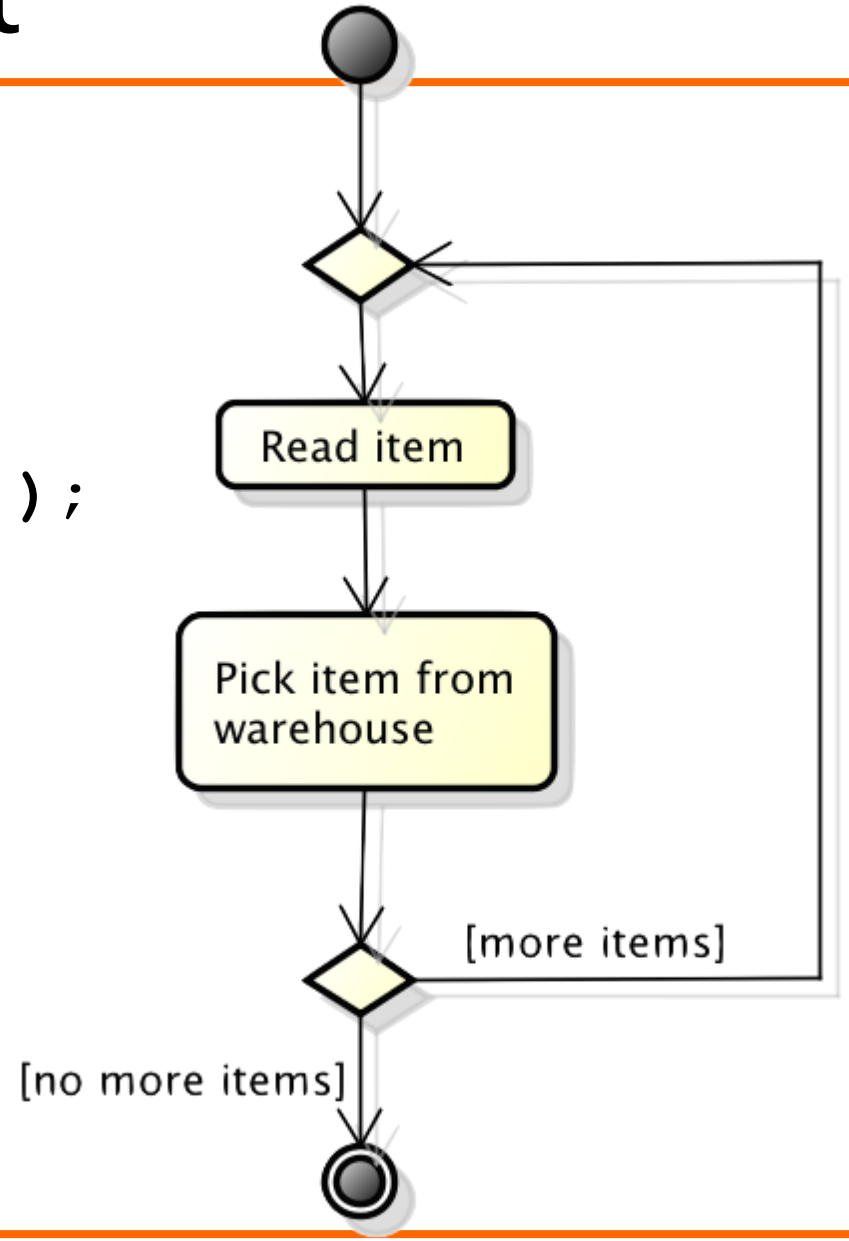


Structured Loop

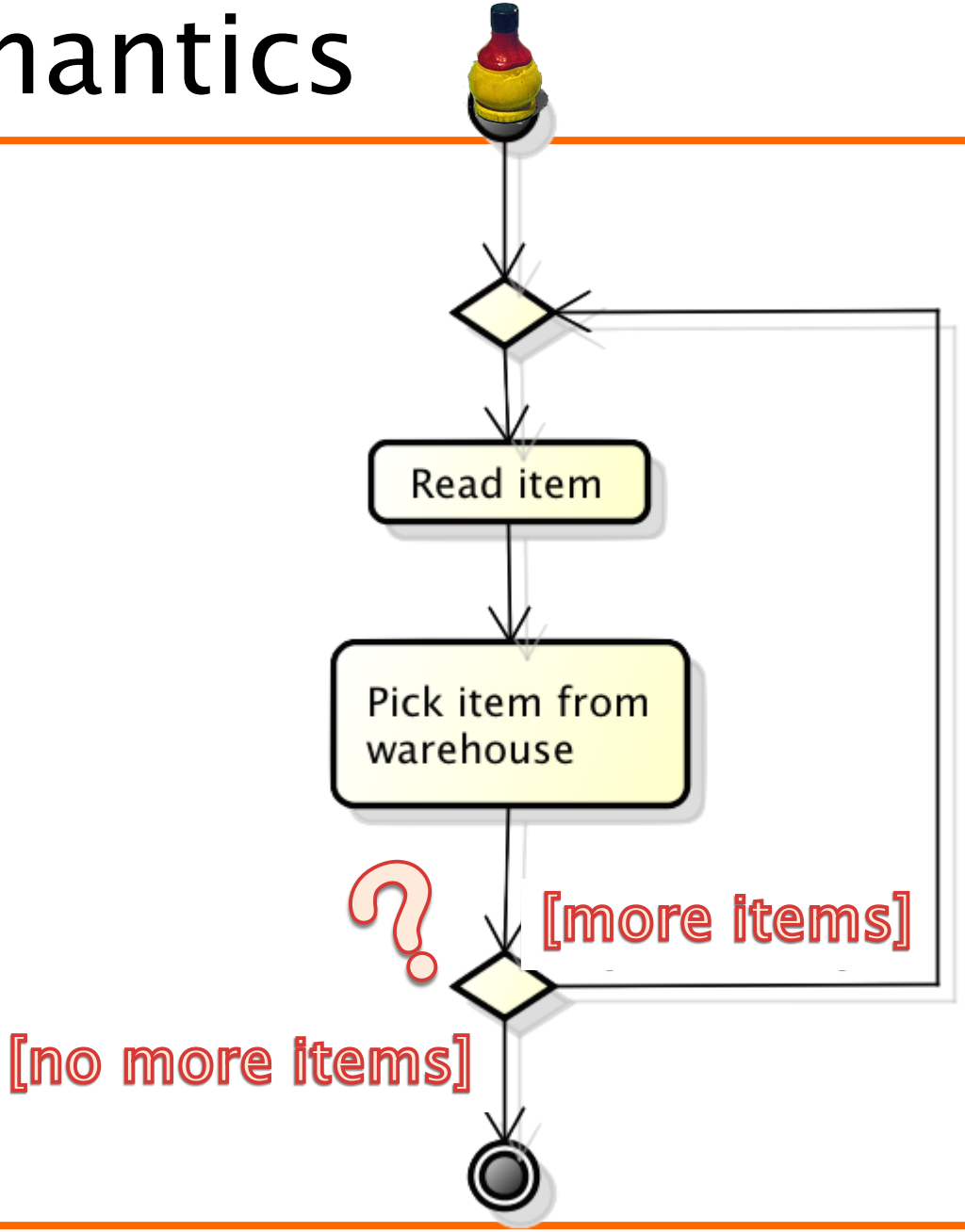
- One or more activities are repeated until a specific condition become true
- Realized by means of decision and merge nodes
  - First a merge node
  - Then a condition

Loop – Repeat

```
do {
   read_item();
   pick_item();
} while( more_items );
```

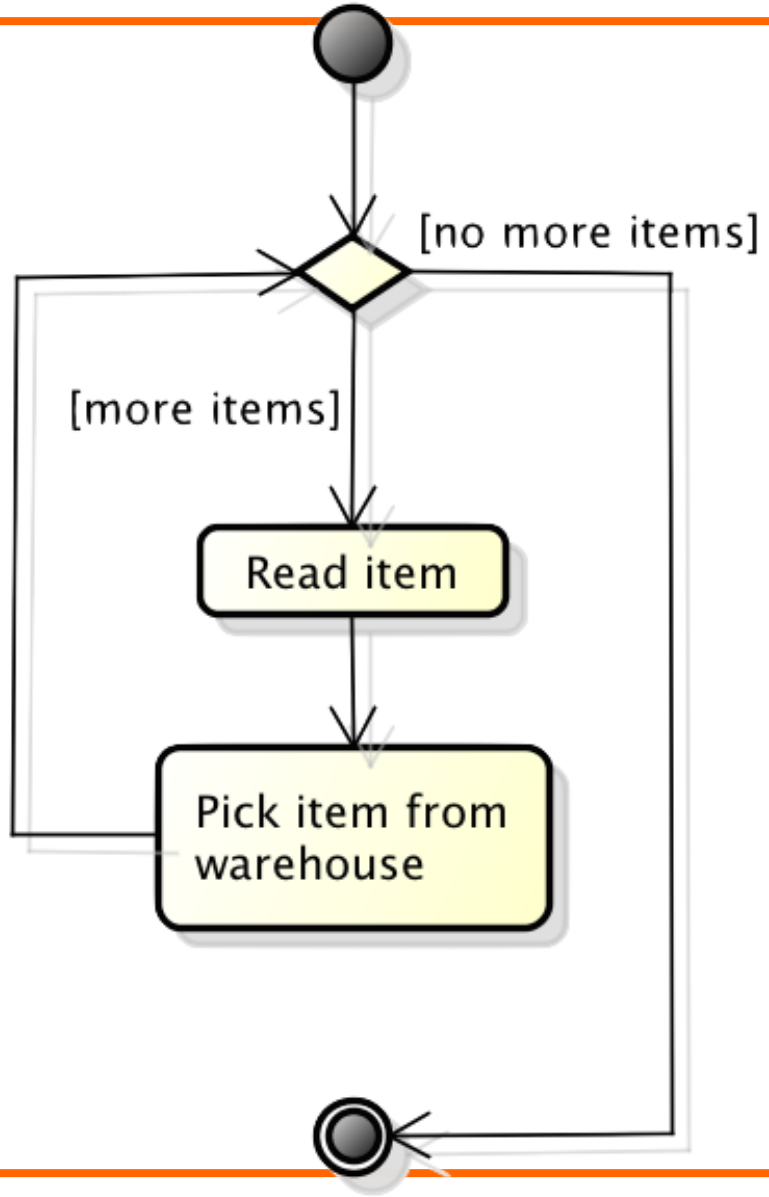


Loop – semantics



Loop – While

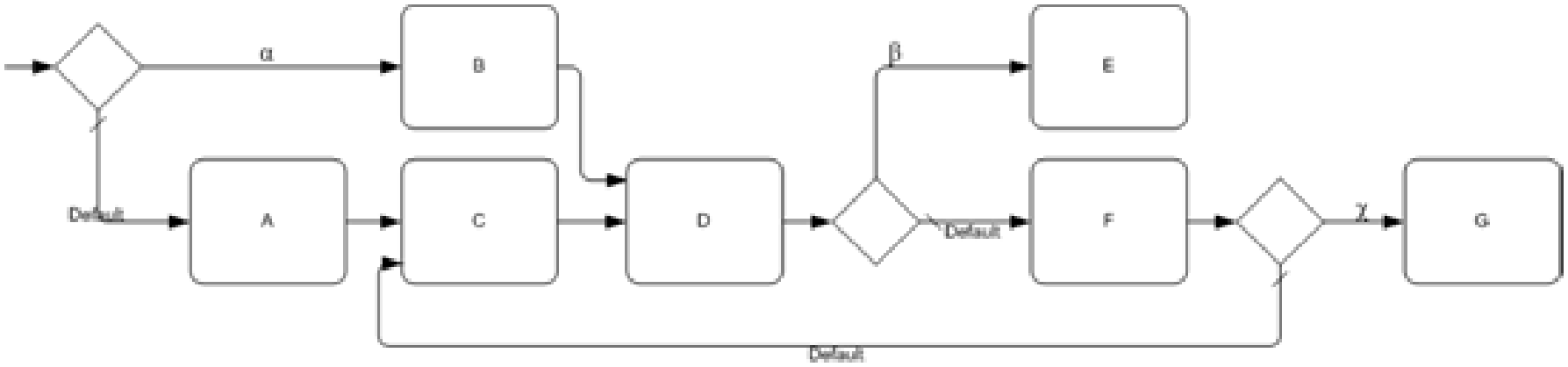
```
while( more_items ){
   read_item();
   pick_item();
}
```



Arbitrary cycles

- Loop that is unstructured or not block structured.
- That is, the looping segment of the process may allow more than one entry or exit point.
- Important for the visualization of valid, but complex, looping situations in a single diagram

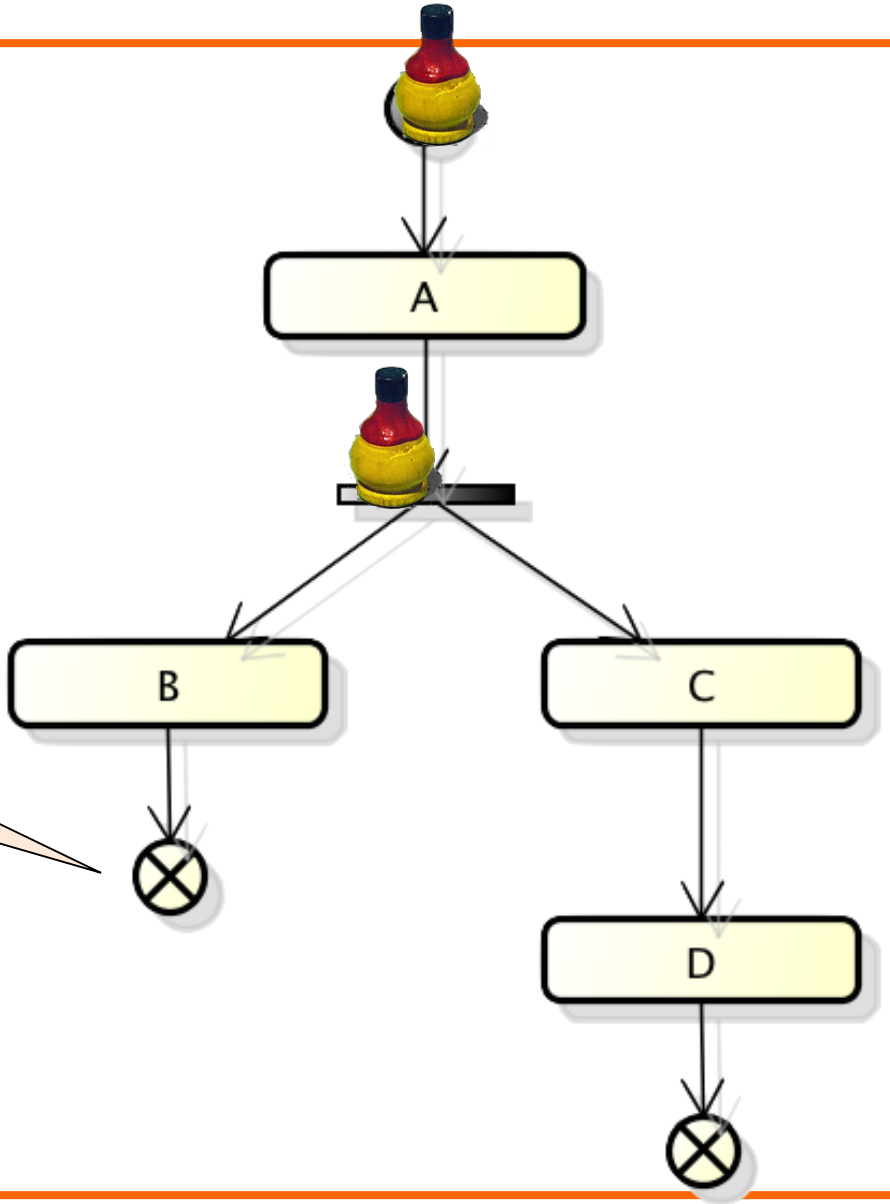
Arbitrary cycles



Implicit termination

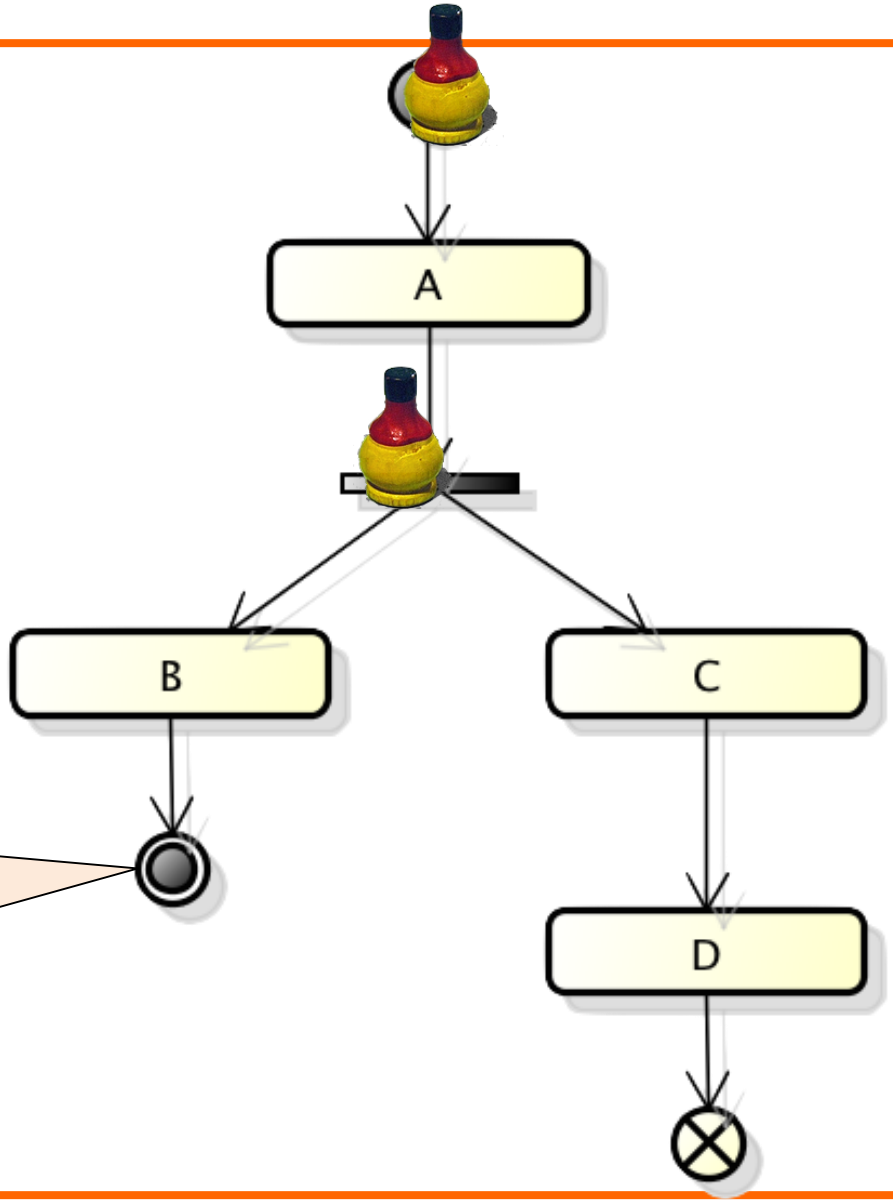
- A specific path of a process can be concluded without other parallel paths be required to end as well.
- The normal case require the whole process to end when any end node is reached.

Implicit termination – semantics



Only one path is terminated

Explicit termination – semantics

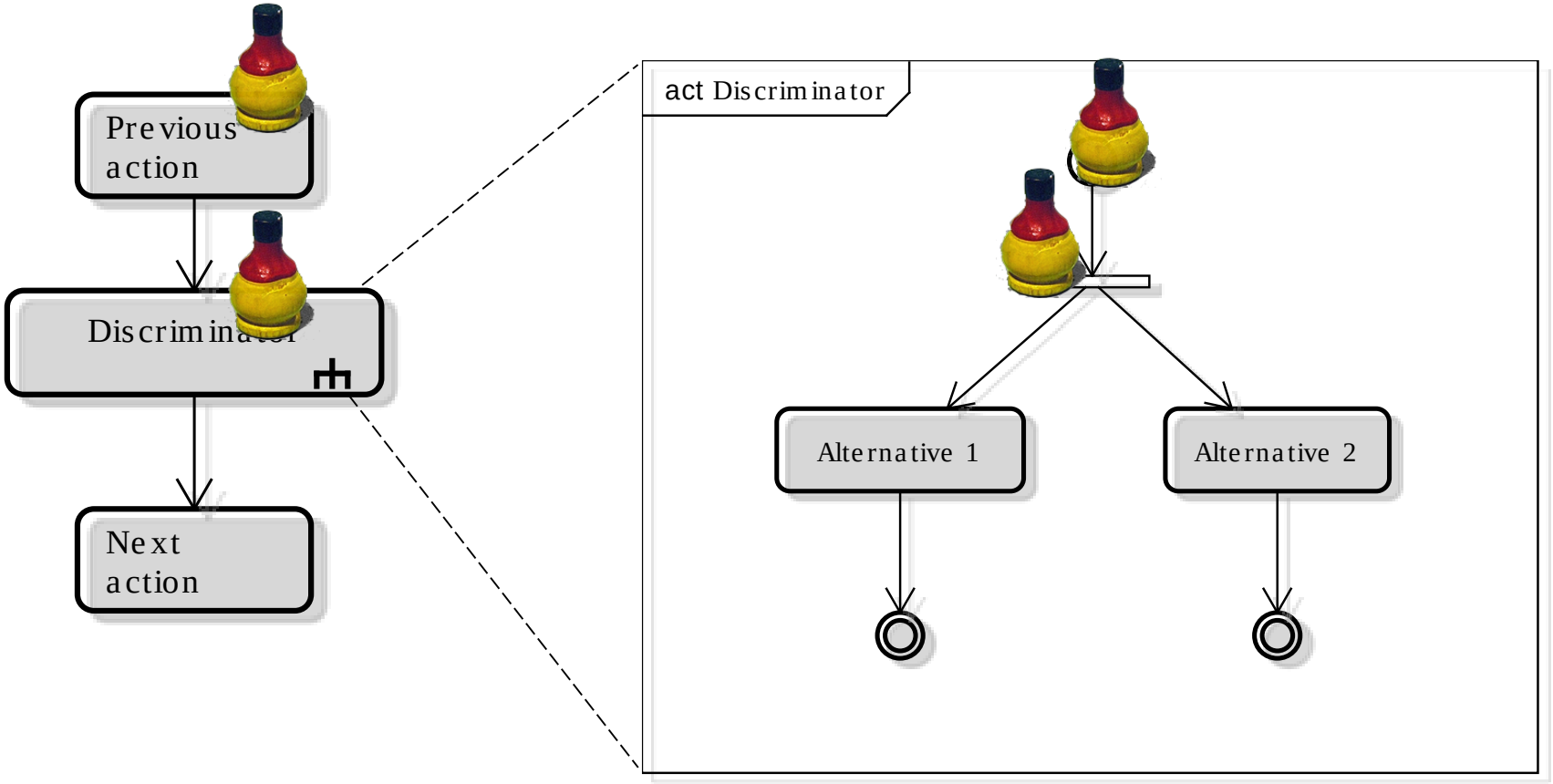


The process is terminated, i.e. ALL path are terminated

Structured Discriminator

- Convergence of two or more branches such that the first activation of an incoming branch results in the subsequent activity being triggered and subsequent activations of remaining incoming branches are ignored.

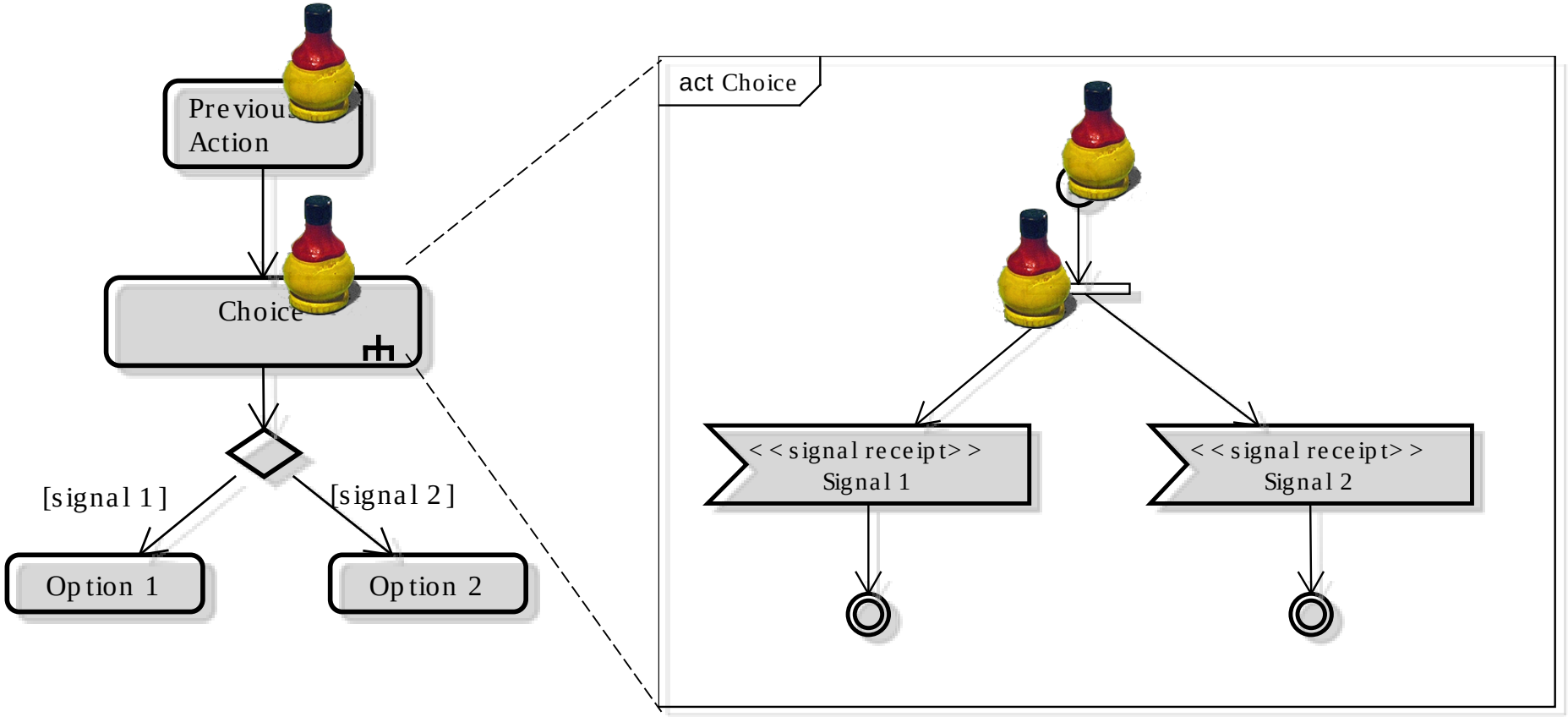
Structured Discriminator



Deferred choice

- A divergence point in a process where one of several possible branches should be activated.
- The actual decision on which branch is activated is made by the environment and is deferred to the latest possible moment.

Deferred choice



Free Tools

- Argo UML
- Astah community
- Star UML (win)
  - Graphical editors
  - Some support to translate to java
  - No support to execute activity diagrams

References

- OMG UML web site
  - http://www.uml.org
- M. Fowler, UML Distilled, Addison-Wesley
- N. Russell et al. WORKFLOW CONTROL-FLOW PATTERNS A Revised View
  - http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf
- N.Russel et al. On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling
  - http://www.workflowpatterns.com/documentation/documents/UMLEvalAPCCM.pdf
- Workflow Patterns web site
  - http://www.workflowpatterns.com

IDEF

IDEF

- Integrated Computer-aided Manufacturing Definition
- Approach of choice in the 1990s (have been around for over 25 years)
- Only one compliant with Federal Information Processing Standards (FIPS)
  - FIPS Publication 183

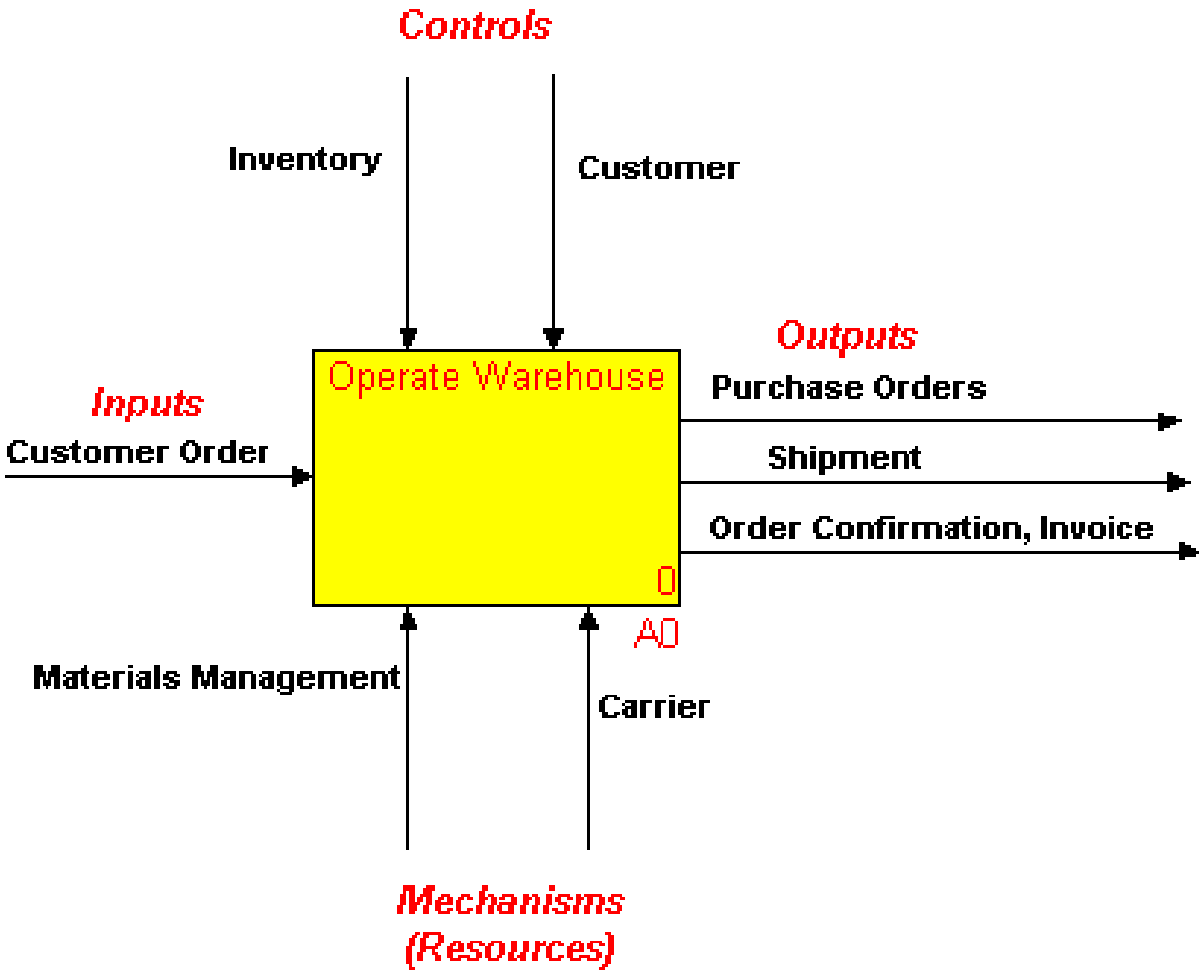
IDEF

- IDEF refers to a group of methods, each of which fulfills a specific purpose
  - IDEFØ, for example, is used to model an organization's functions
  - IDEF1x is used for DB modeling

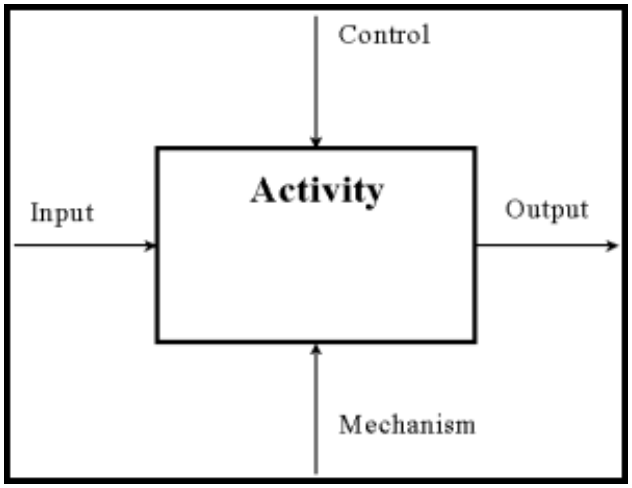
IDEFØ

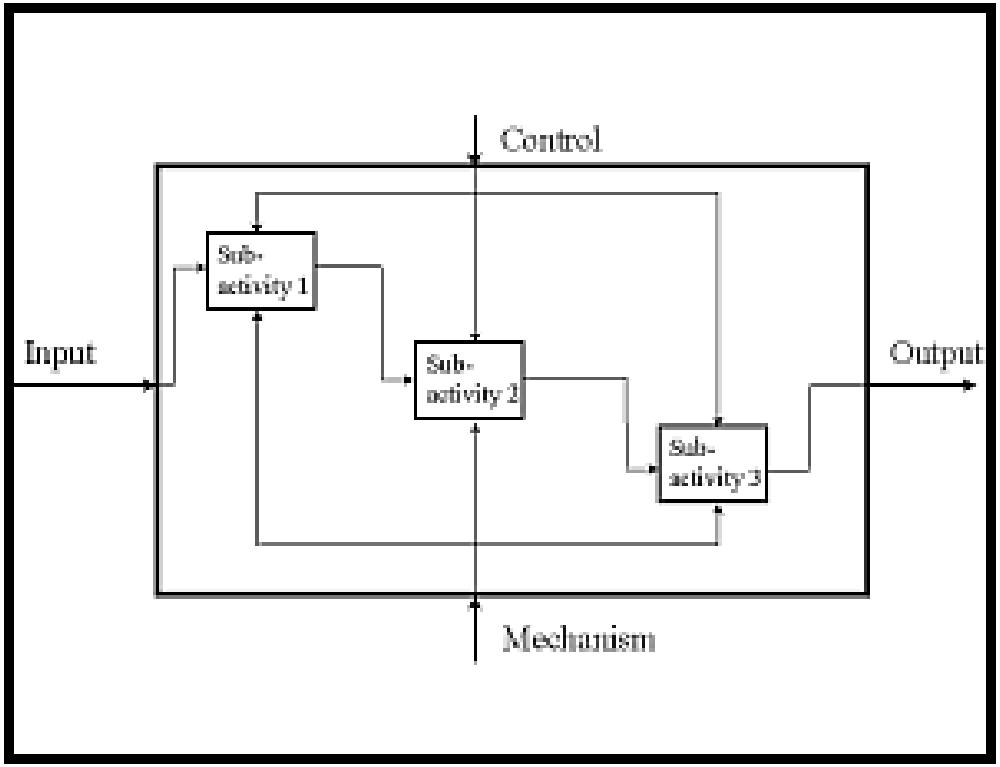
- “Box and arrow" graphics
  - function as a box
  - interfaces to or from the function as arrows entering or leaving the box
- Context diagram (main)
- Constraint diagrams (sub)
- Decomposition

IDEFØ – Example



IDEFØ – Decomposition





DFD

DFD

- Data Flow Diagram
  - Yourdon and Coad
  - Gane and Sarson

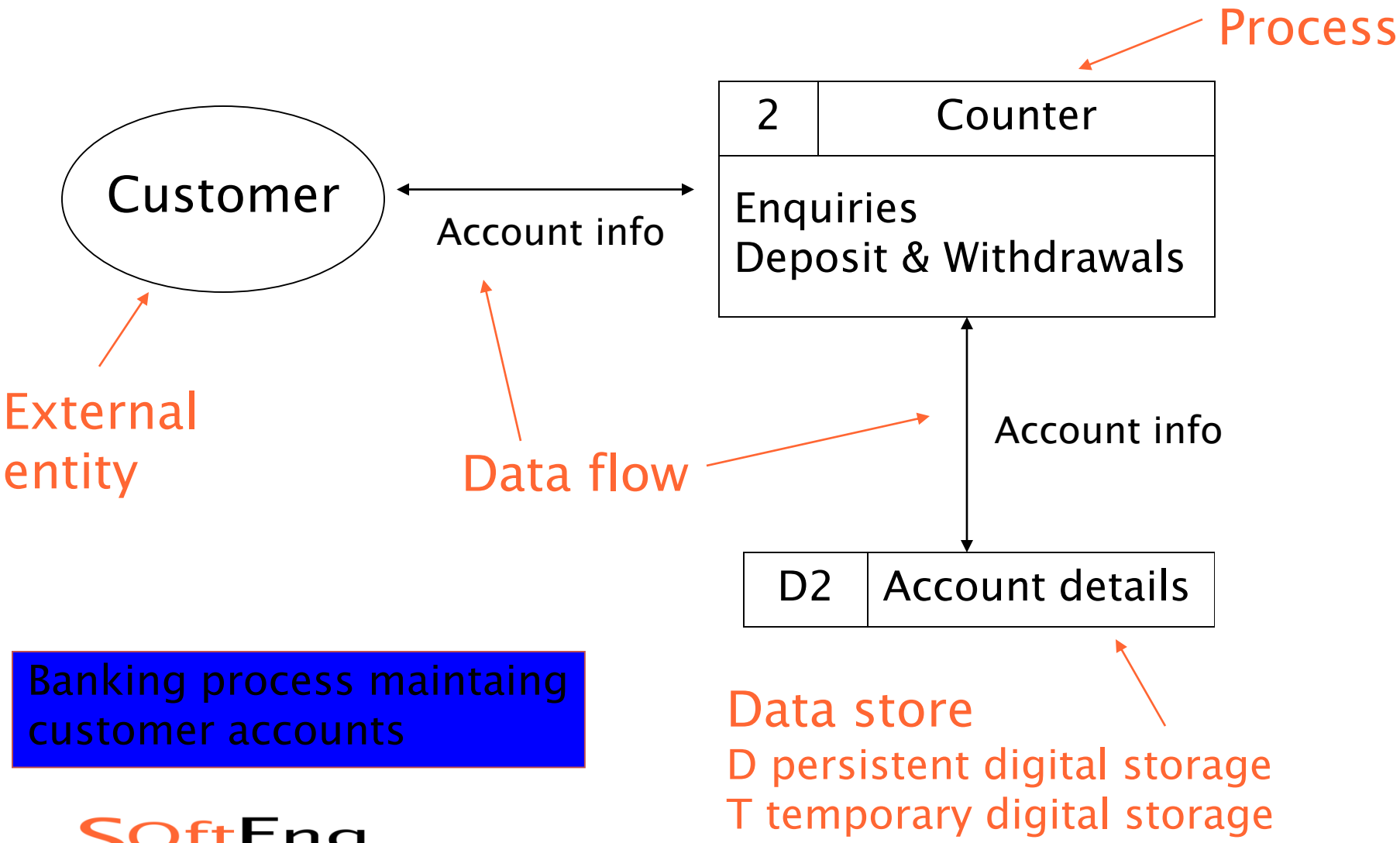
DFD – Decomposition

- Initially a context diagram is drawn, which is a simple representation of the *entire system* under investigation
- This is followed by a level 1 diagram, which identifies *major business processes* at a high level
- These processes can then be analyzed further with level 2 diagrams
- And so on…

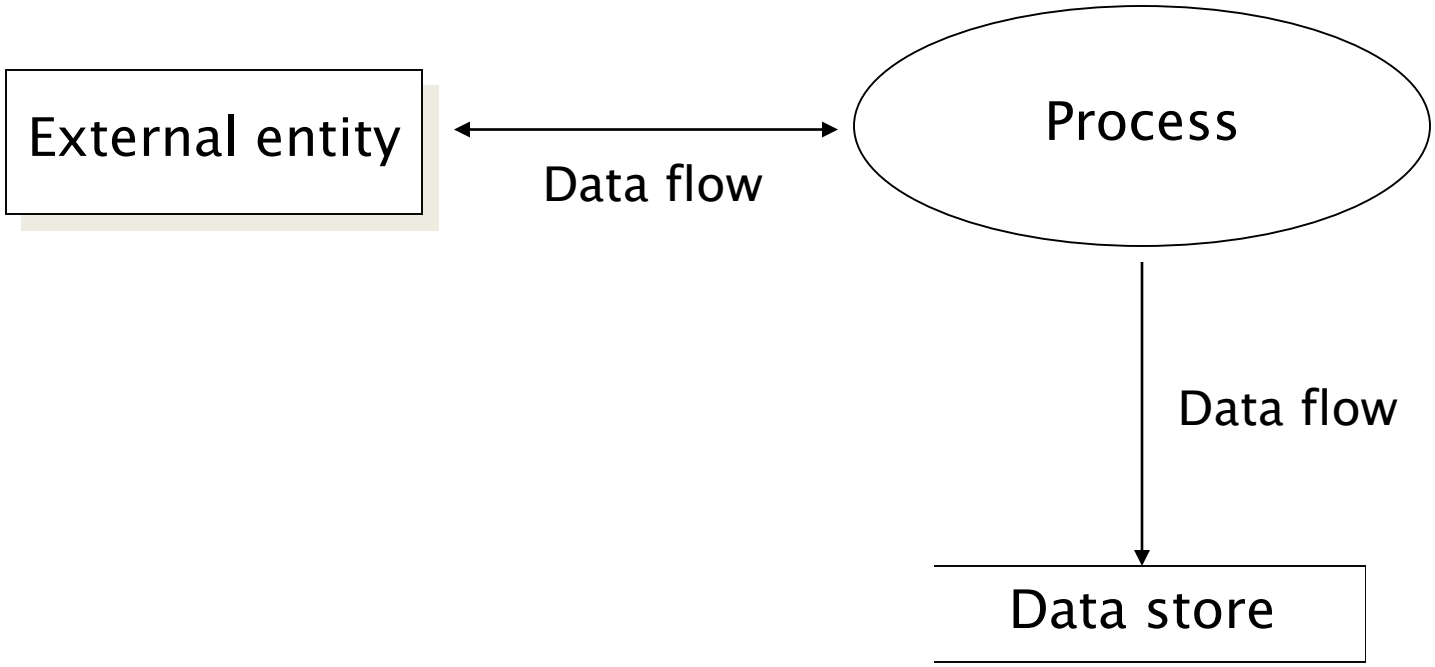
DFD – Objects

- Process
  - A process is a unit of work that operates on the data
- Data flow
  - A data flow is a named flow of data through a system of processes
- Data store
  - A data store is a logical repository of data
- External entity
  - An external agent is a source or destination of data

DFD – Example


Process
2
Counter
Enquiries
Deposit & Withdrawals
Customer
Account info
External entity
Data flow
Account info
D2
Account details
Banking process maintaing customer accounts
Data store
D persistent digital storage
T temporary digital storage

DFD – Example II



DFD – Rules

- Data flows are allowed between
  - different external entities
  - processes and external entities
  - processes and data stores
- Data flows are not allowed between
  - external entities and data stores
  - one data store and another