Software Engineering (Other UML Diagrams)

Lina YE

CentraleSupélec

https://www.lri.fr/~linaye/GL.html
lina.ye@centralesupelec.fr

Sequence 3, 2017-2018

Plan

Recall

Behavior Diagrams

- Use case diagram
- Interaction diagram (Sequence, collaboration diagram)
- State machine diagram
- Activity diagram

Interaction Diagrams

Important elements:

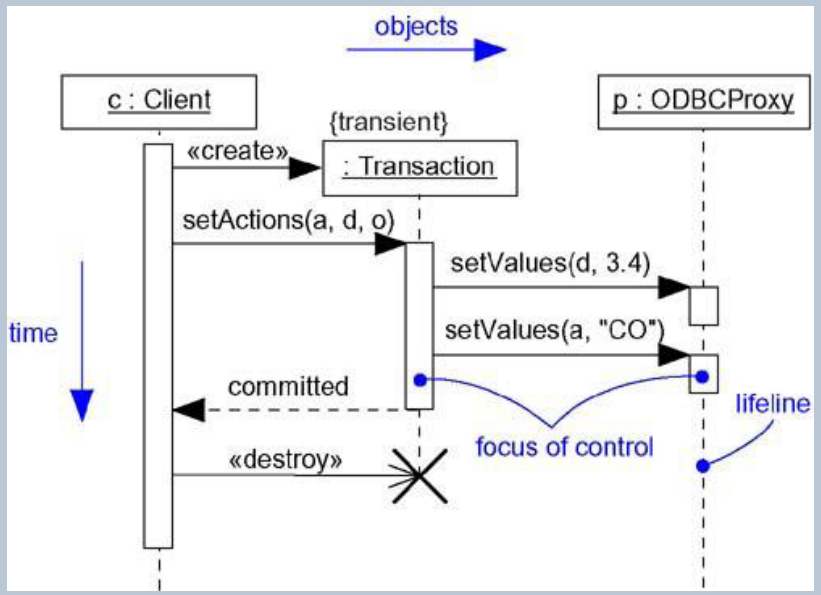
- a set of objects participating to realize a (set of) scenario(s) corresponding to a use case
- communication between objects by sending messages

Sequence diagram

- represent objects horizontally, whose existence is represented by its lifeline (vertical dashed line)
- represent time vertically
- emphasize the time ordering of messages

Sequence Diagram

Example



Interaction Diagrams

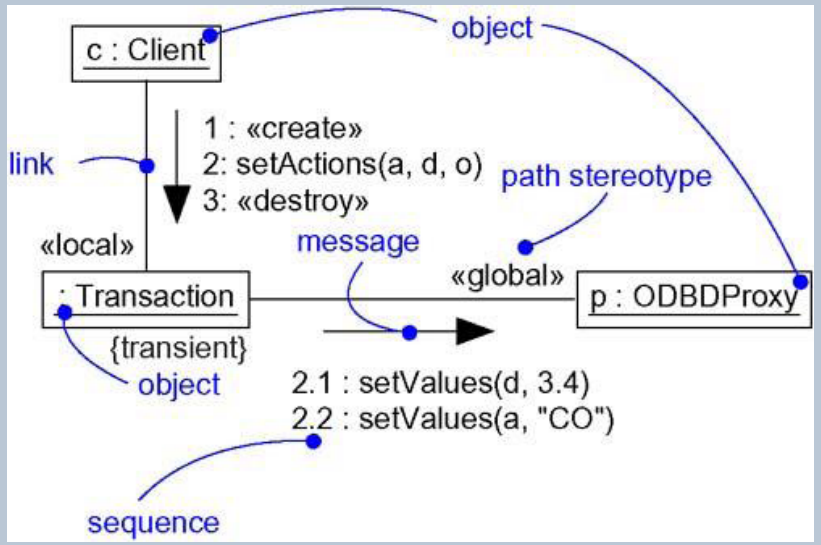
- a set of objects participating to realize a (set of) scenario(s) corresponding to a use case
- communication between objects by sending messages

Collaboration diagram

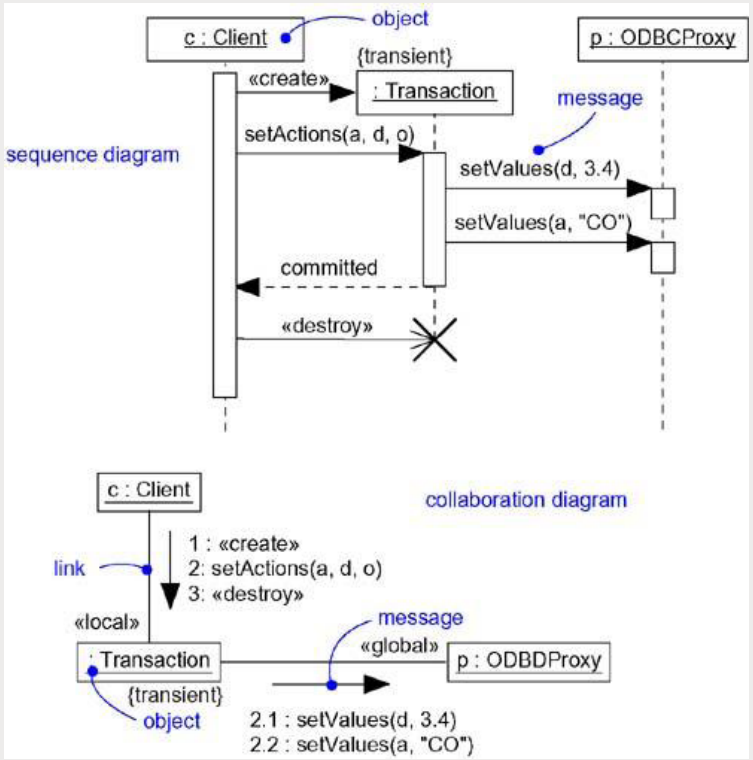
- Graphically it is a collection of vertices and arcs
- Ordering of messages is represented by sequence number
- Emphasize the structural organization of the objects that send and receive messages

Collaboration Diagram

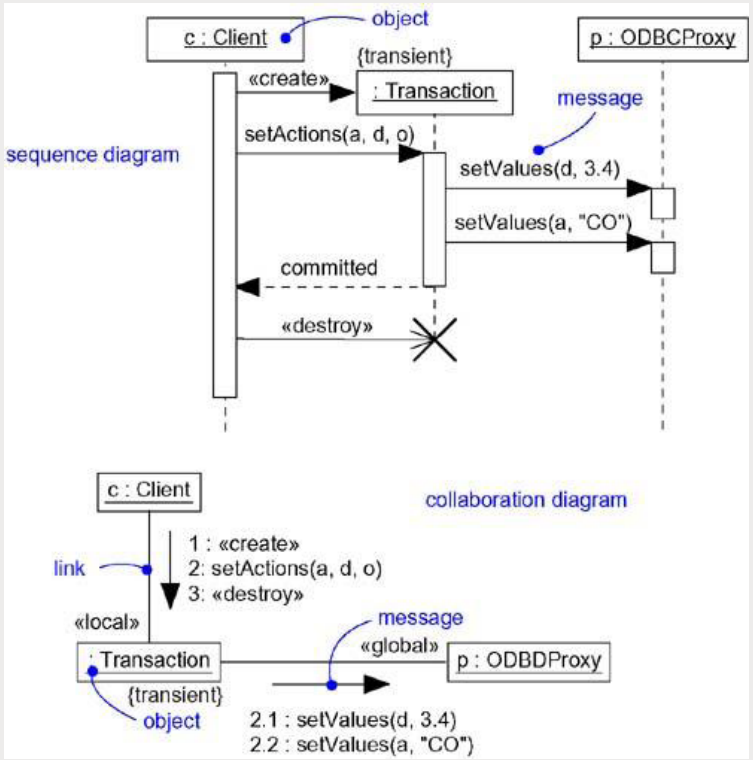
Example



What's is missing in Interaction diagram?



What's is missing in Interaction diagram?



In interaction diagrams, we have no information about how the state of an object changes.

State diagram

Models to describe state-dependent behaviors of an object. An object responds differently to the same event depending on what state it is in.

State diagram

Models to describe state-dependent behaviors of an object. An object responds differently to the same event depending on what state it is in.

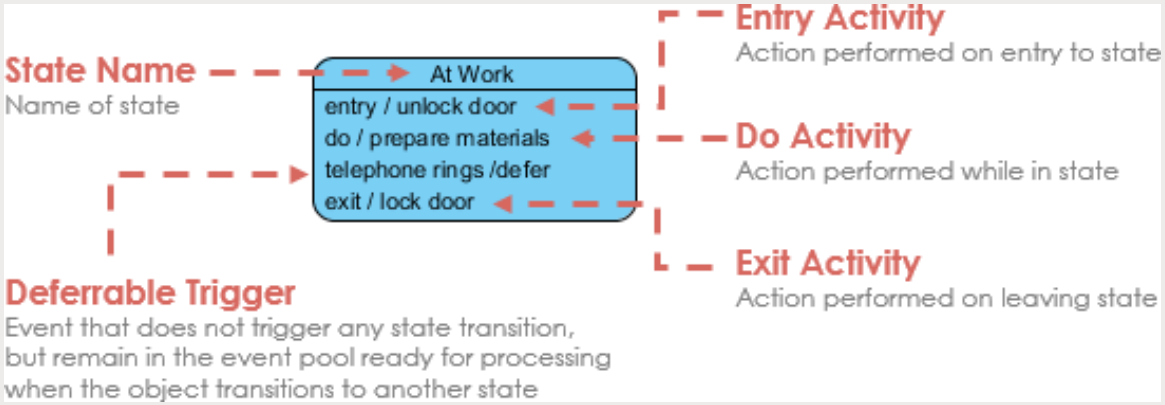
- states (composite, concurrent): an abstraction of the attribute values satisfying some conditions and links of an object (occupy an interval of time).
- transition: move from one state to another and with five components (source state, event, guard, action, destination source).

State

- state: may have a second compartment containing actions or activities performed in this state
  - Entry & exit actions: actions that always occur upon entry into or exit away from a state regardless of transition.
  - Do activity: ongoing behavior which occurs and continues when the object in this state
  - Deferred events: events not handled by the current state but postponed for later processing

State

- state: may have a second compartment containing actions or activities performed in this state
  - Entry & exit actions: actions that always occur upon entry into or exit away from a state regardless of transition.
  - Do activity: ongoing behavior which occurs and continues when the object in this state
  - Deferred events: events not handled by the current state but postponed for later processing

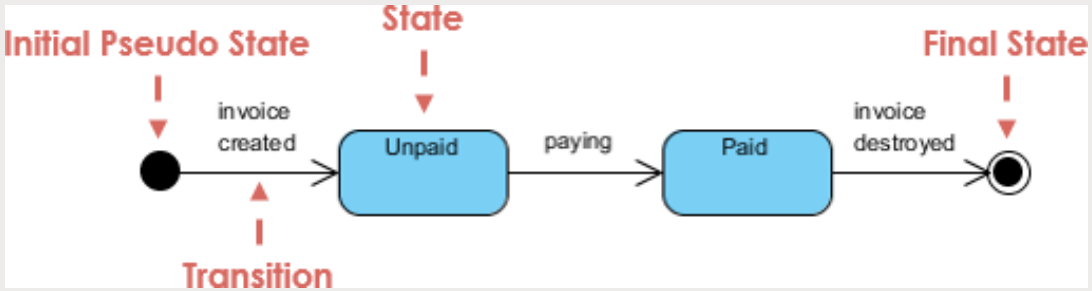


State

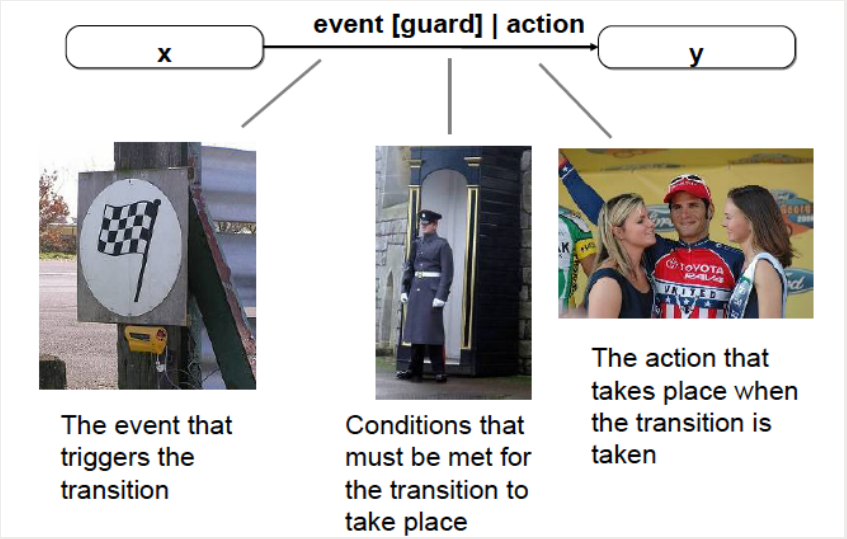
- state: may have a second compartment containing actions or activities performed in this state
- initial state: a transition from this state will go to a first real state (solid circle).
- final state: represent an end state of an object (concentric circle).

State

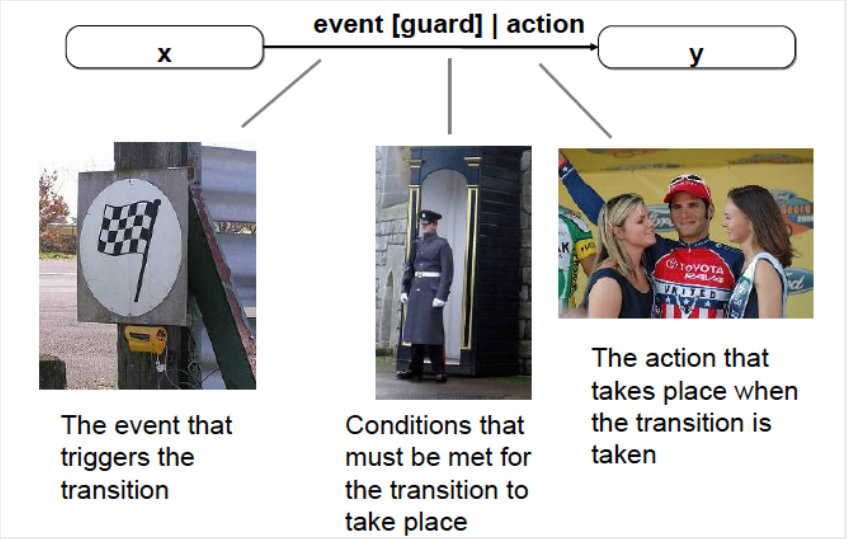
- state: may have a second compartment containing actions or activities performed in this state
- initial state: a transition from this state will go to a first real state (solid circle).
- final state: represent an end state of an object (concentric circle).



Transition


event [guard] | action
x
y
The event that triggers the transition
Conditions that must be met for the transition to take place
The action that takes place when the transition is taken

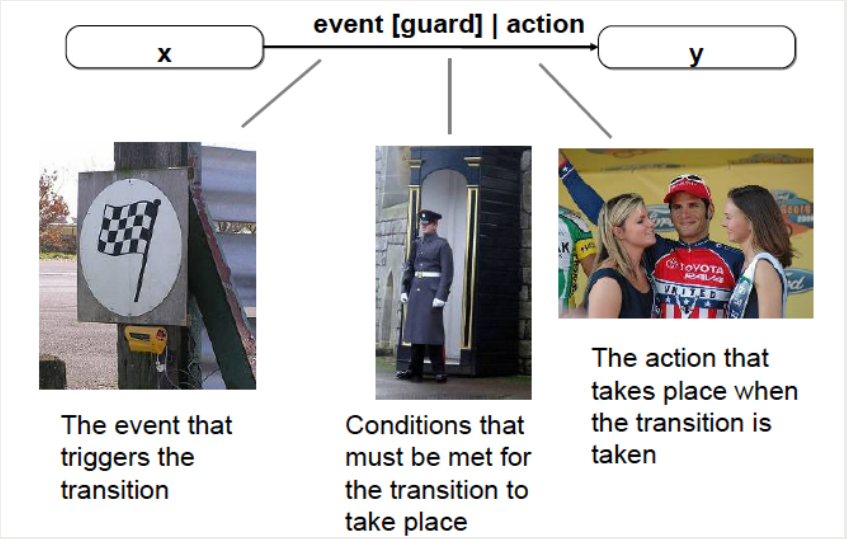
Transition



A transition between two states occurs as follows:

1. the object is in the source state

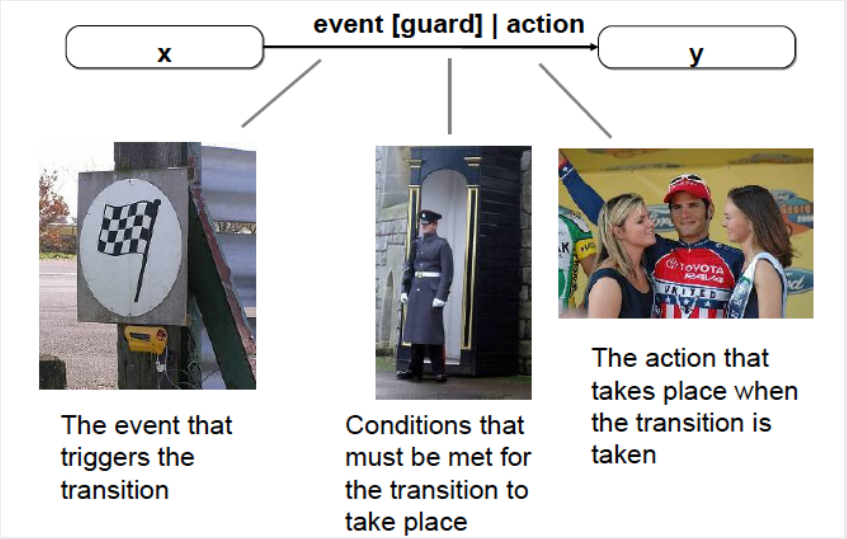
Transition



A transition between two states occurs as follows:

1. the object is in the source state
2. the triggering event occurs

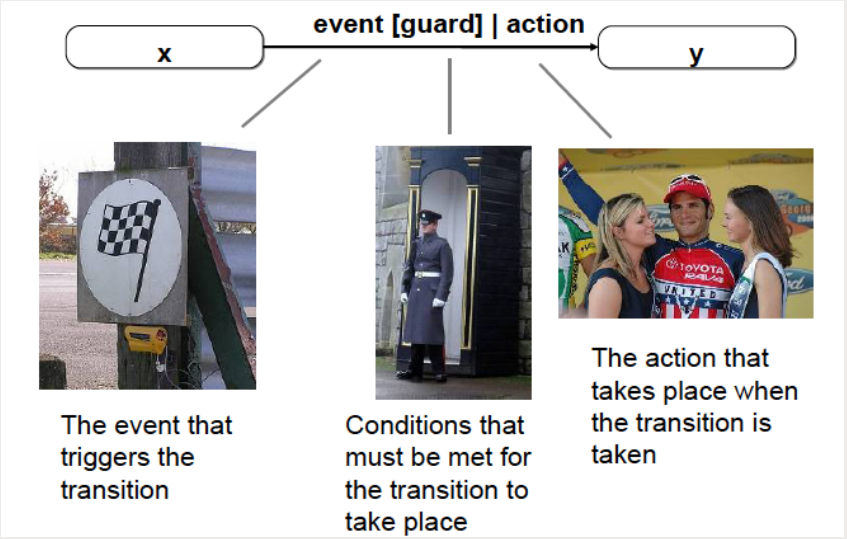
Transition



A transition between two states occurs as follows:

1. the object is in the source state
2. the triggering event occurs
3. satisfy the guard

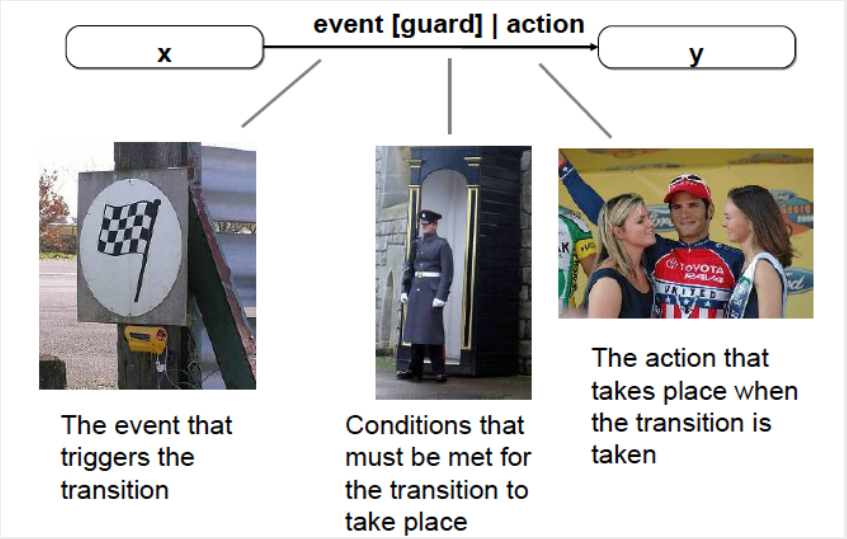
Transition



A transition between two states occurs as follows:

1. the object is in the source state
2. the triggering event occurs
3. satisfy the guard
4. perform the action(s)

Transition



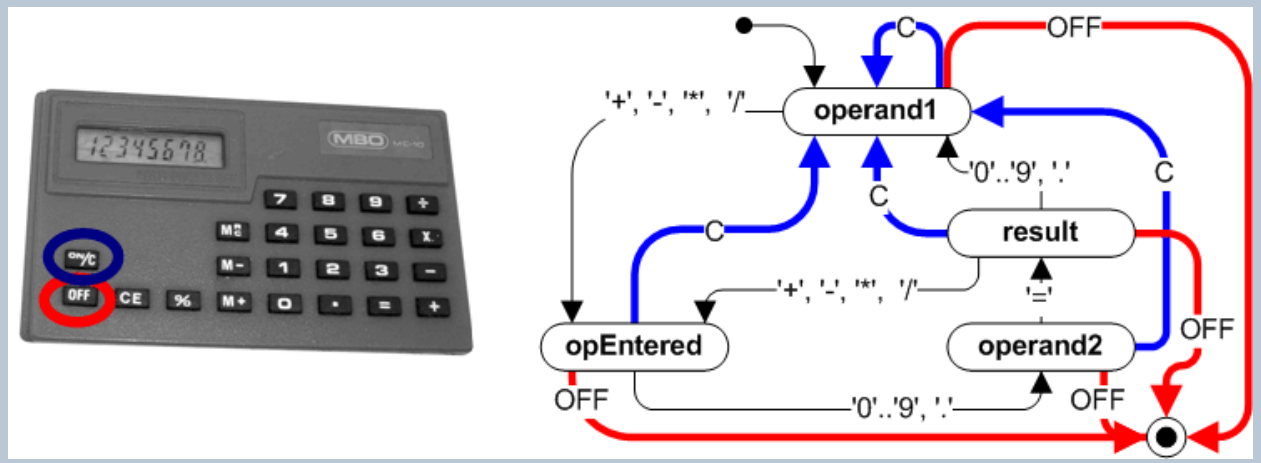
A transition between two states occurs as follows:

1. the object is in the source state
2. the triggering event occurs
3. satisfy the guard
4. perform the action(s)
5. the object enters the destination state

Composite state

A state which has substates (nested states): to simplify complex flat state machines.

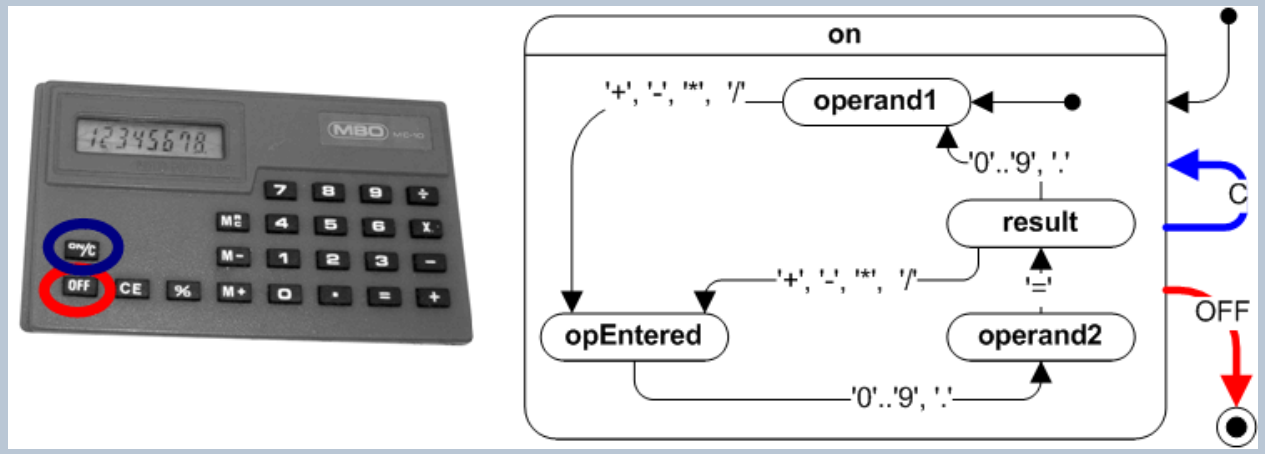
example (simple state)



Composite state

A state which has substates (nested states): to simplify complex flat state machines.

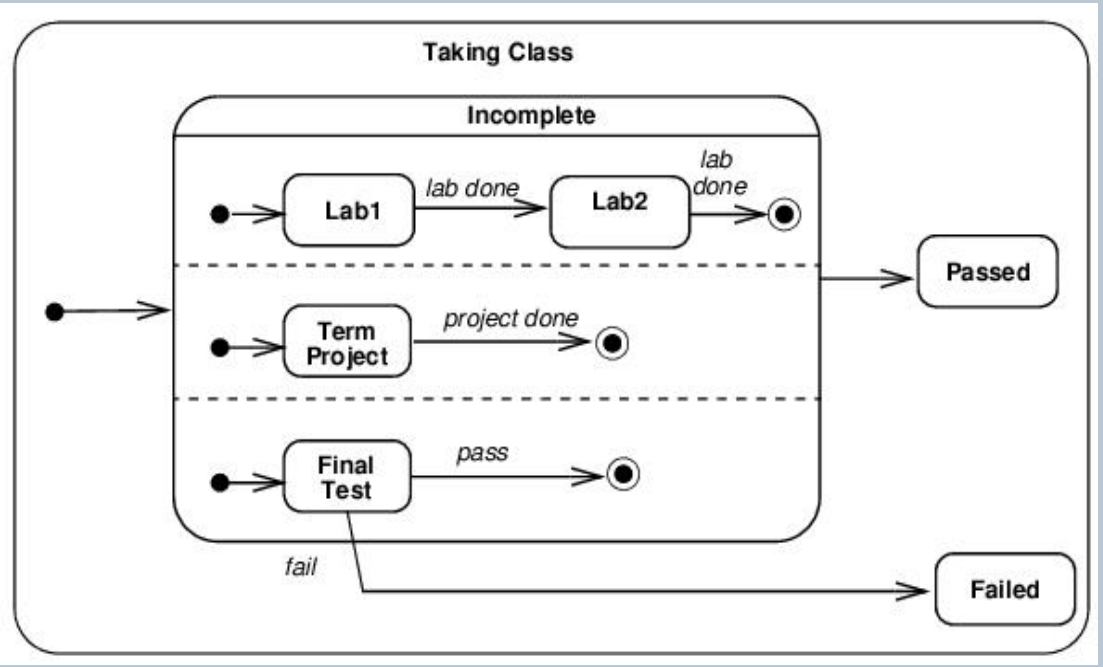
example (composite state)



Concurrent state

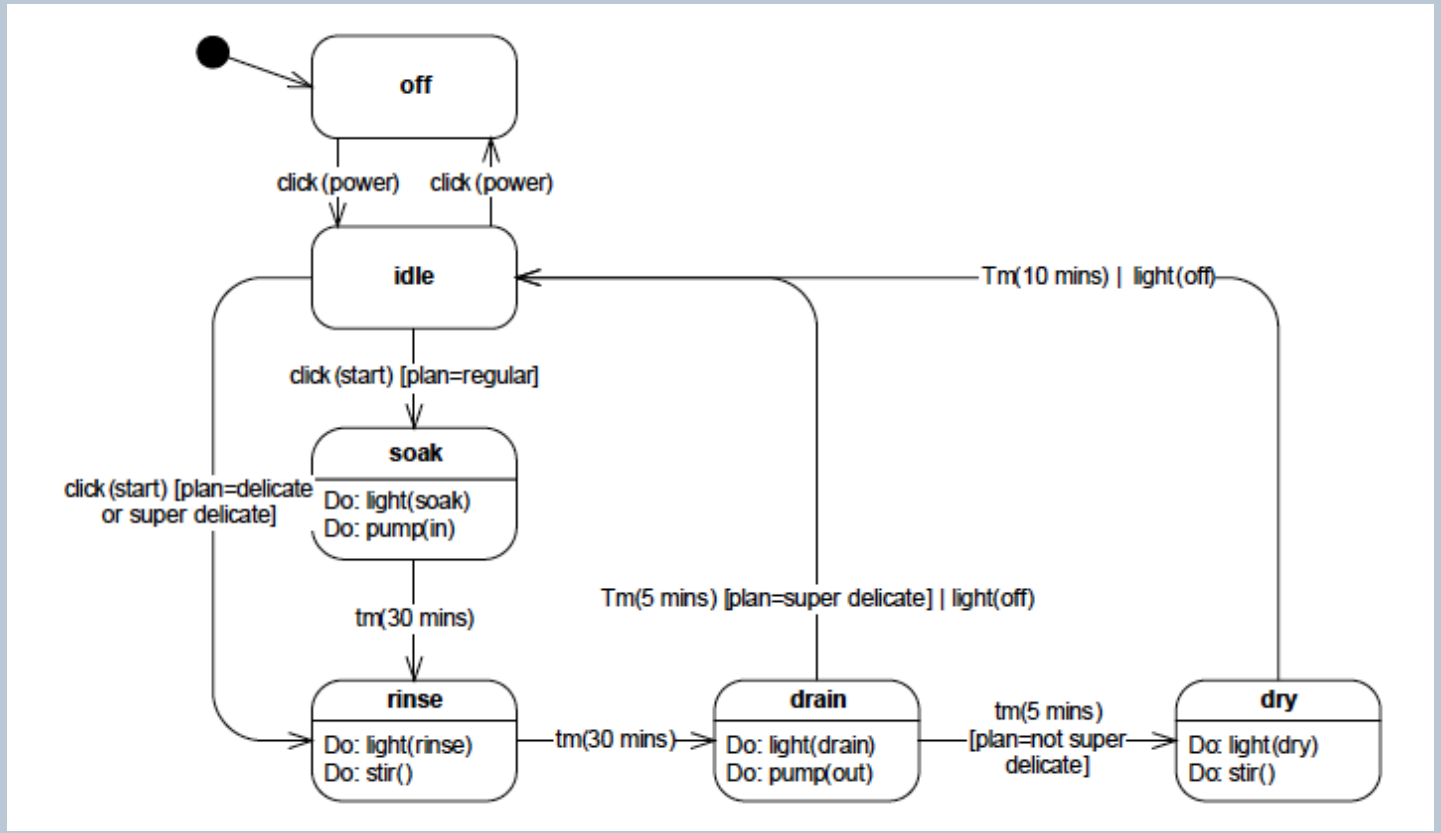
Concurrent substate: one object is simultaneously in several states.

example (concurrent state)



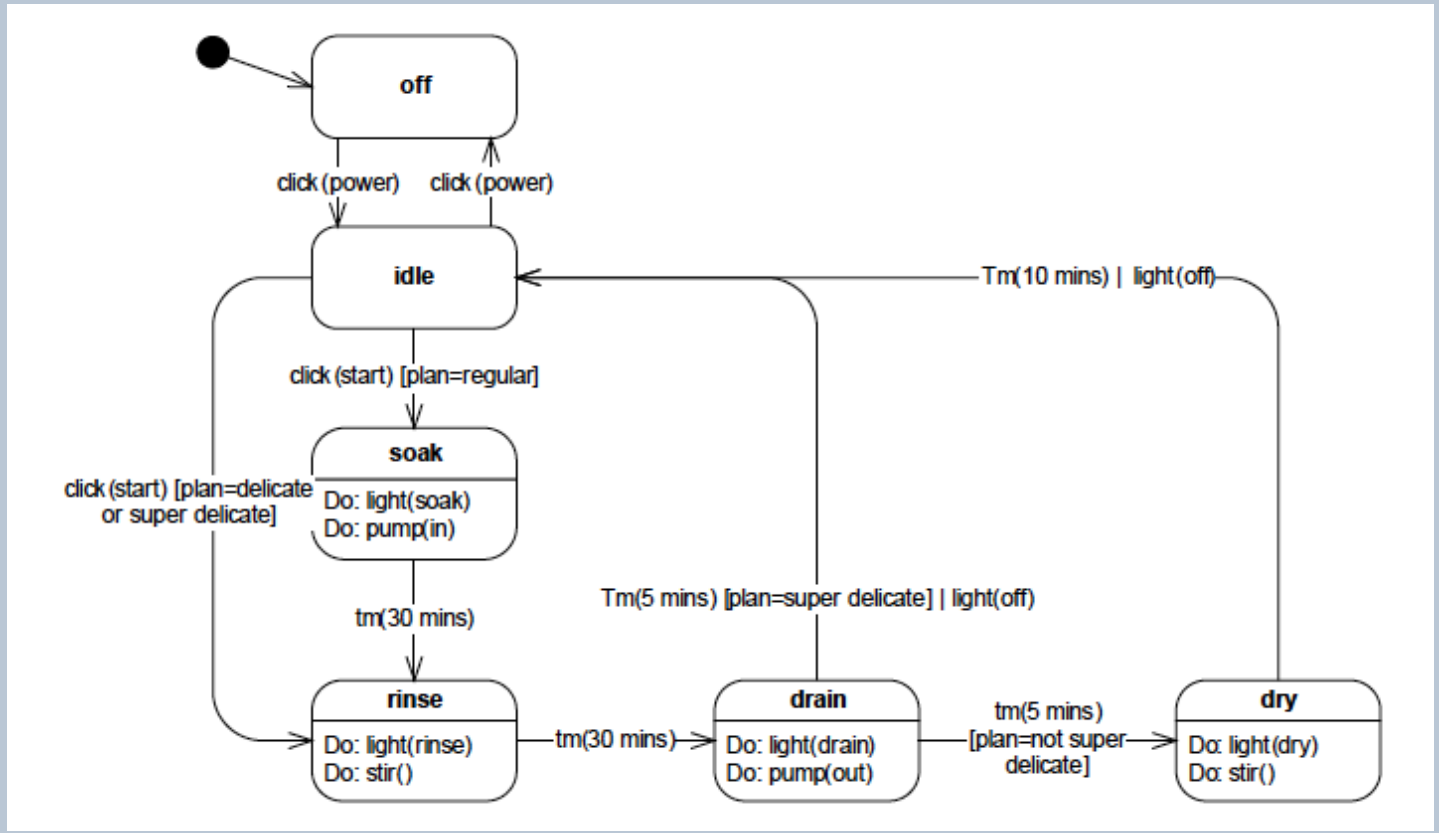
Example

Washing machine



Example

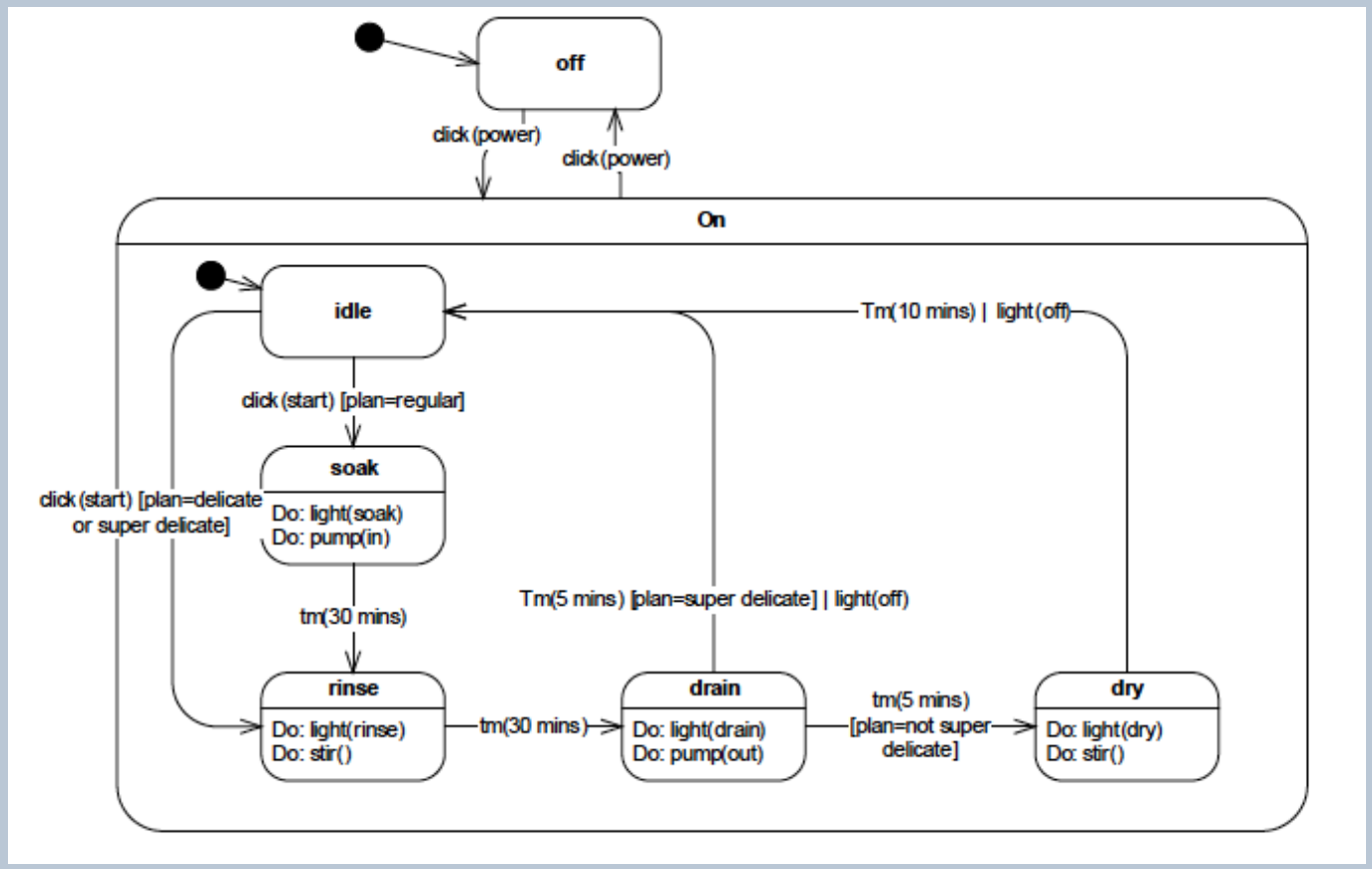
Washing machine



But what if the off button can be clicked at any time?

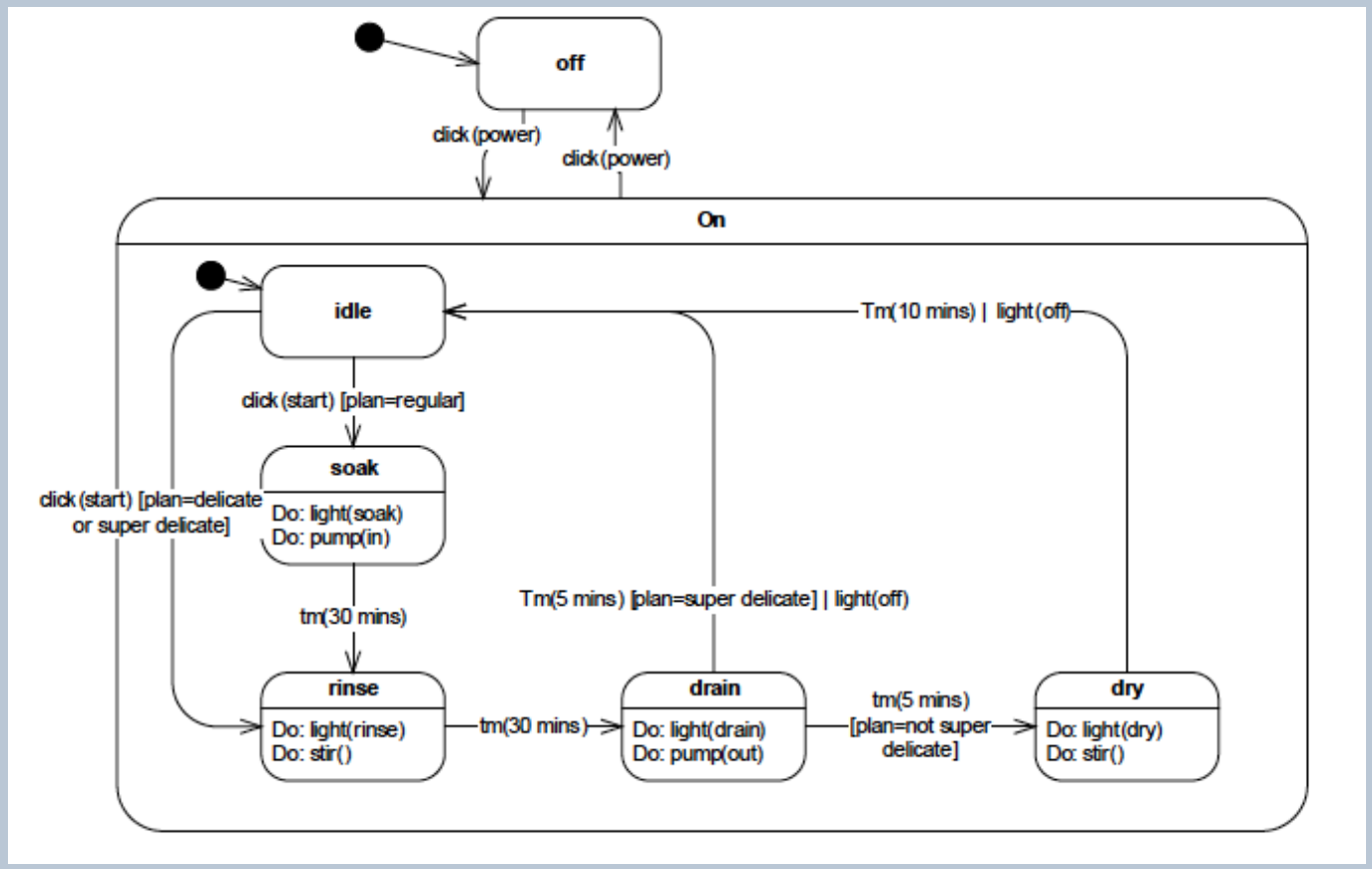
Example

Washing machine



Example

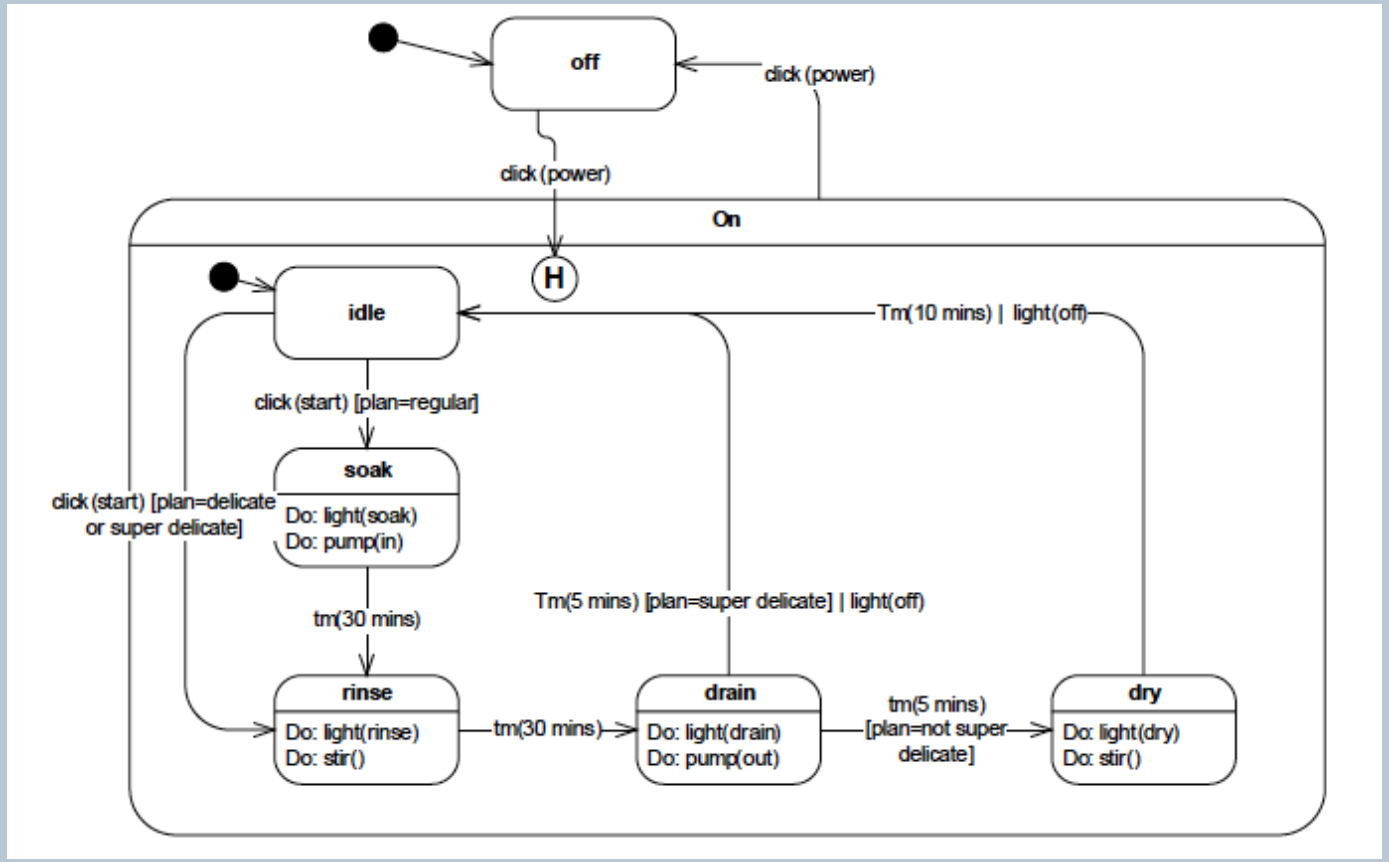
Washing machine



What if we want to come back to the same state we left?

Example

Washing machine



Activity Diagram

Principles

- Model the processing (adapted to the modeling of control flows and data flows).

Activity Diagram

Principles

- Model the processing (adapted to the modeling of control flows and data flows).
- Take into account choices and iterations.

Activity Diagram

Principles

- Model the processing (adapted to the modeling of control flows and data flows).
- Take into account choices and iterations.
- Useful for graphically describing the behavior of a method or the execution of a scenario of a use case.

Activity Diagram

Principles

- Model the processing (adapted to the modeling of control flows and data flows).
- Take into account choices and iterations.
- Useful for graphically describing the behavior of a method or the execution of a scenario of a use case.
- Interaction diagrams: the flow of control from one object to another
  state diagrams: the flow of control from a state from another for an object.
  activity diagrams: the flow of control between activities.

Activity Diagram

Terminology

- Activity: a processing, where the flow of execution is modeled by nodes connected by arcs (transitions)

Activity Diagram

Terminology

- Activity: a processing, where the flow of execution is modeled by nodes connected by arcs (transitions)
- Transition: the transition from one activity to another, which is triggered as soon as the source activity is over and determines the next activity to be triggered (atomic).

Activity Diagram

Terminology

- Three types of nodes: control nodes; execution nodes; object nodes

Activity Diagram

Terminology

- Three types of nodes: control nodes; execution nodes; object nodes
- Control nodes:
  - decision node: makes a choice between several outgoing flows (diamond)
  - merge node: gathers several incoming flows in a single outgoing flow to choose one incoming flow (diamond)
  - fork node: separates one flow into several concurrent flows (bar)

Activity Diagram

Terminology

- Control nodes:
  - union node (join): synchronize several incoming flows (bar)
  - initial node: start the flow when the activity is invoked (black circle)
  - final node: end the execution of the activity (circled black circle)

Activity Diagram

Terminology

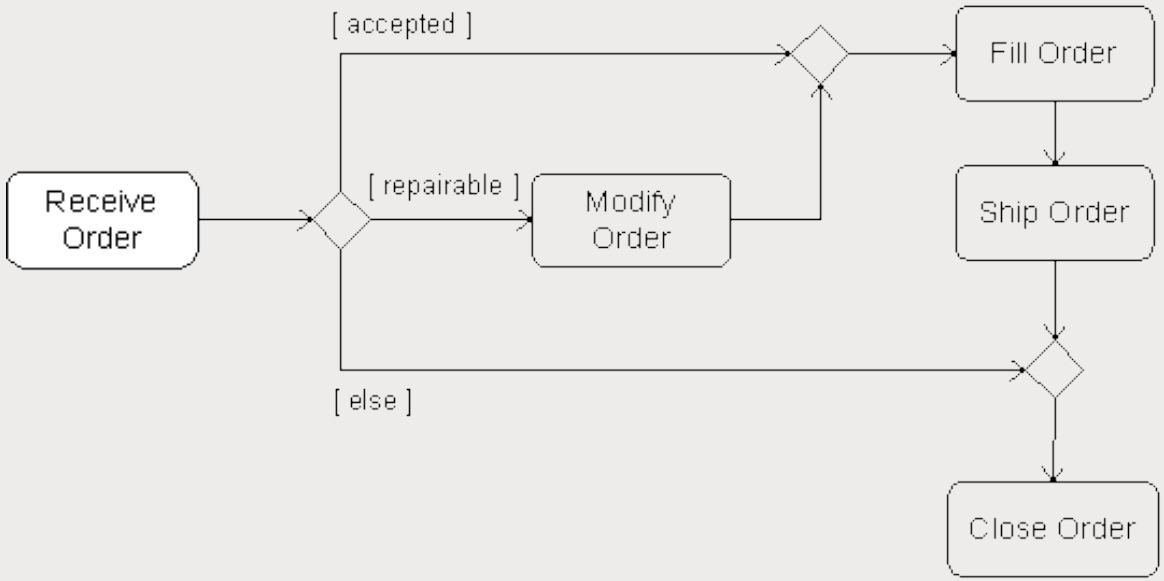
- Control nodes:
  - union node (join): synchronize several incoming flows (bar)
  - initial node: start the flow when the activity is invoked (black circle)
  - final node: end the execution of the activity (circled black circle)
- Execution nodes: activity node that can be executed (rounded rectangle)

Activity Diagram

Terminology

- Control nodes:
  - union node (join): synchronize several incoming flows (bar)
  - initial node: start the flow when the activity is invoked (black circle)
  - final node: end the execution of the activity (circled black circle)
- Execution nodes: activity node that can be executed (rounded rectangle)
- Object nodes: define an object flow (the data), which represents the existence of an object generated in one activity and used by others (rectangle)

Example: order processing

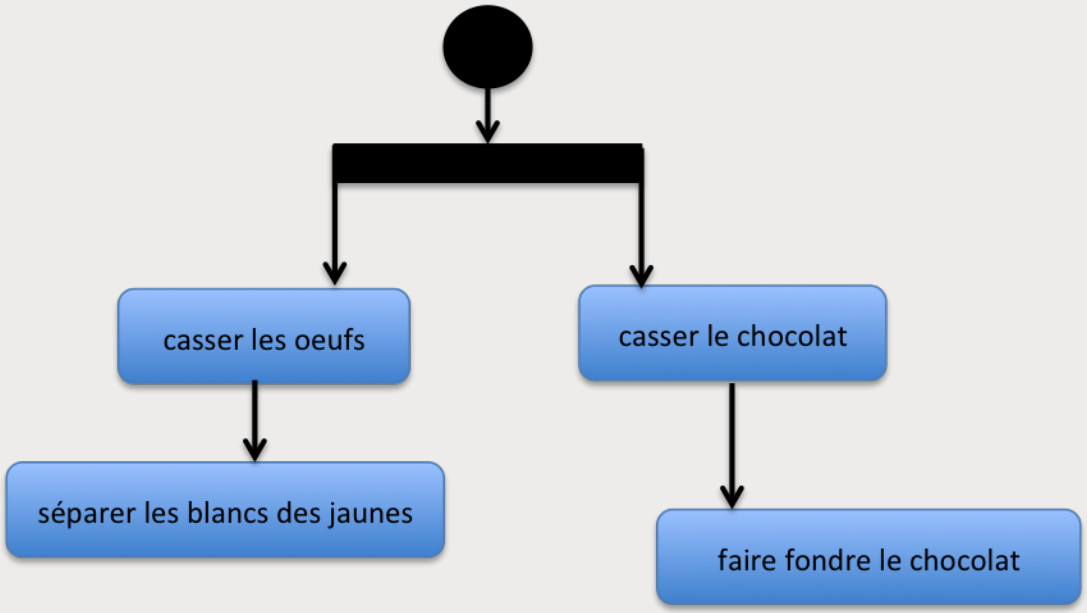


Example

Kitchen recipe

- Start by breaking the chocolate into pieces, then melt it.
- In parallel, break the eggs by separating the whites from the yolks.

Example

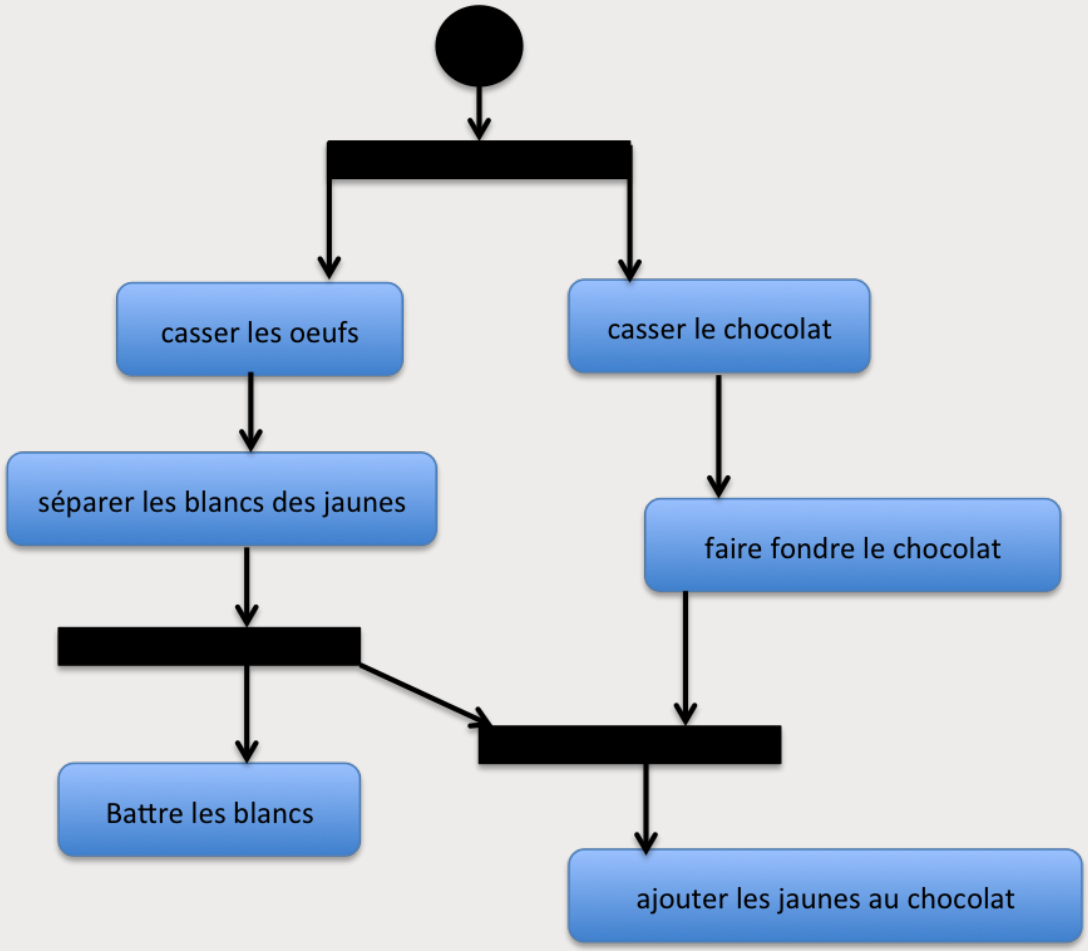

casser les oeufs
casser le chocolat
séparer les blancs des jaunes
faire fondre le chocolat

Example

Kitchen recipe

- Start by breaking the chocolate into pieces, then melt it.
- In parallel, break the eggs by separating the whites from the yolks.
- When the chocolate is melted, add the egg yolks.
- Beat the egg whites until they are stiff.

Example

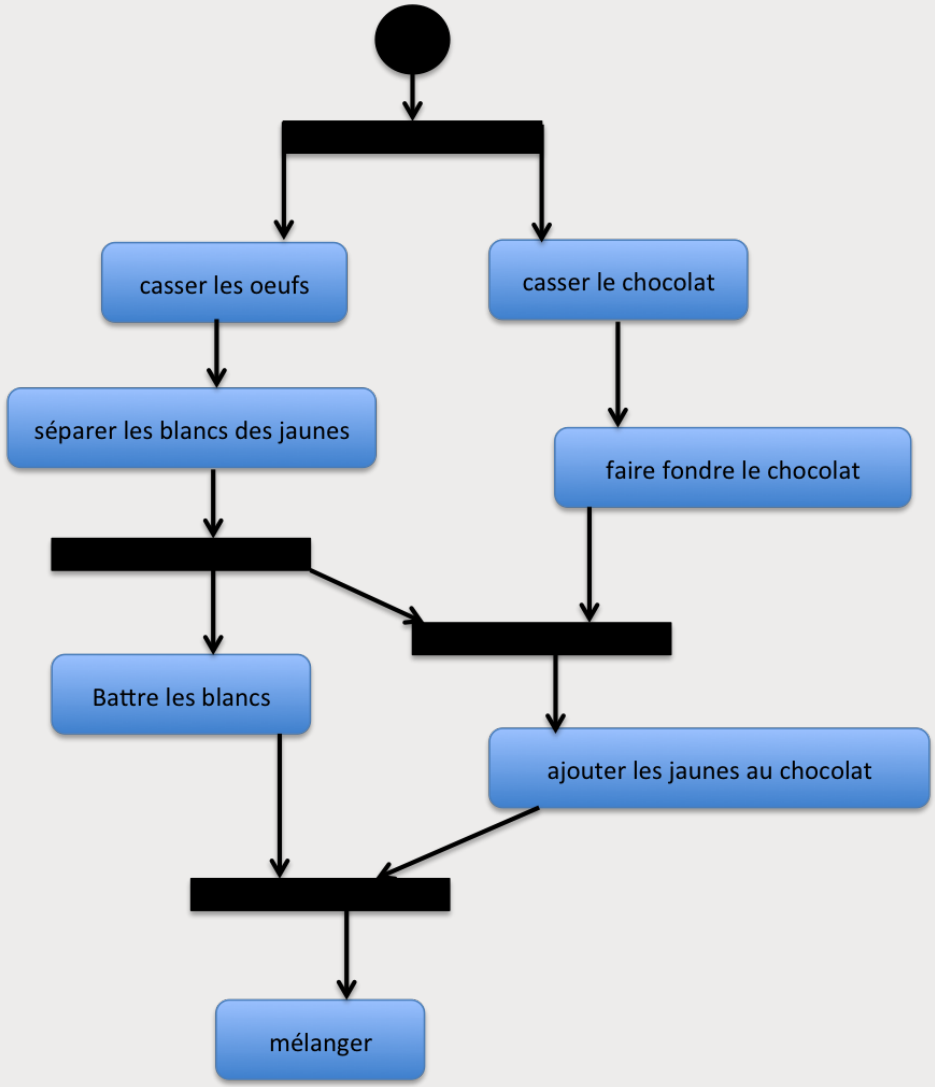

casser les oeufs
casser le chocolat
séparer les blancs des jaunes
faire fondre le chocolat
Battre les blancs
ajouter les jaunes au chocolat

Example

Kitchen recipe

- Start by breaking the chocolate into pieces, then melt it.
- In parallel, break the eggs by separating the whites from the yolks.
- When the chocolate is melted, add the egg yolks.
- Beat the egg whites until they are stiff.
- Incorporate them gently into the chocolate mixture without breaking them.

Example

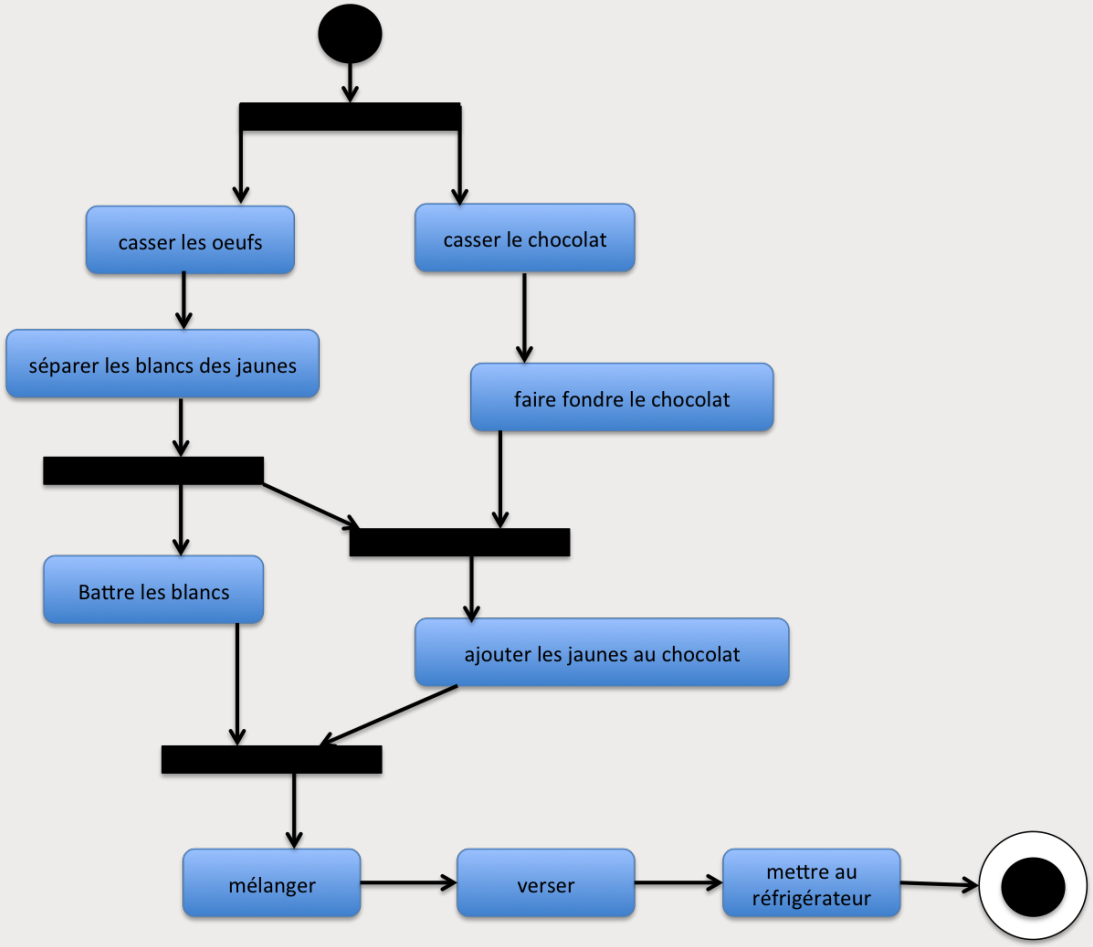

casser les oeufs
casser le chocolat
séparer les blancs des jaunes
faire fondre le chocolat
Battre les blancs
ajouter les jaunes au chocolat
mélanger

Example

Kitchen recipe

- Start by breaking the chocolate into pieces, then melt it.
- In parallel, break the eggs by separating the whites from the yolks.
- When the chocolate is melted, add the egg yolks.
- Beat the egg whites until they are stiff.
- Incorporate them gently into the chocolate mixture without breaking them.
- Pour into individual ramekins.
- Refrigerate before serving.

Example


casser les oeufs
casser le chocolat
séparer les blancs des jaunes
faire fondre le chocolat
Battre les blancs
ajouter les jaunes au chocolat
mélanger
verser
mettre au réfrigérateur

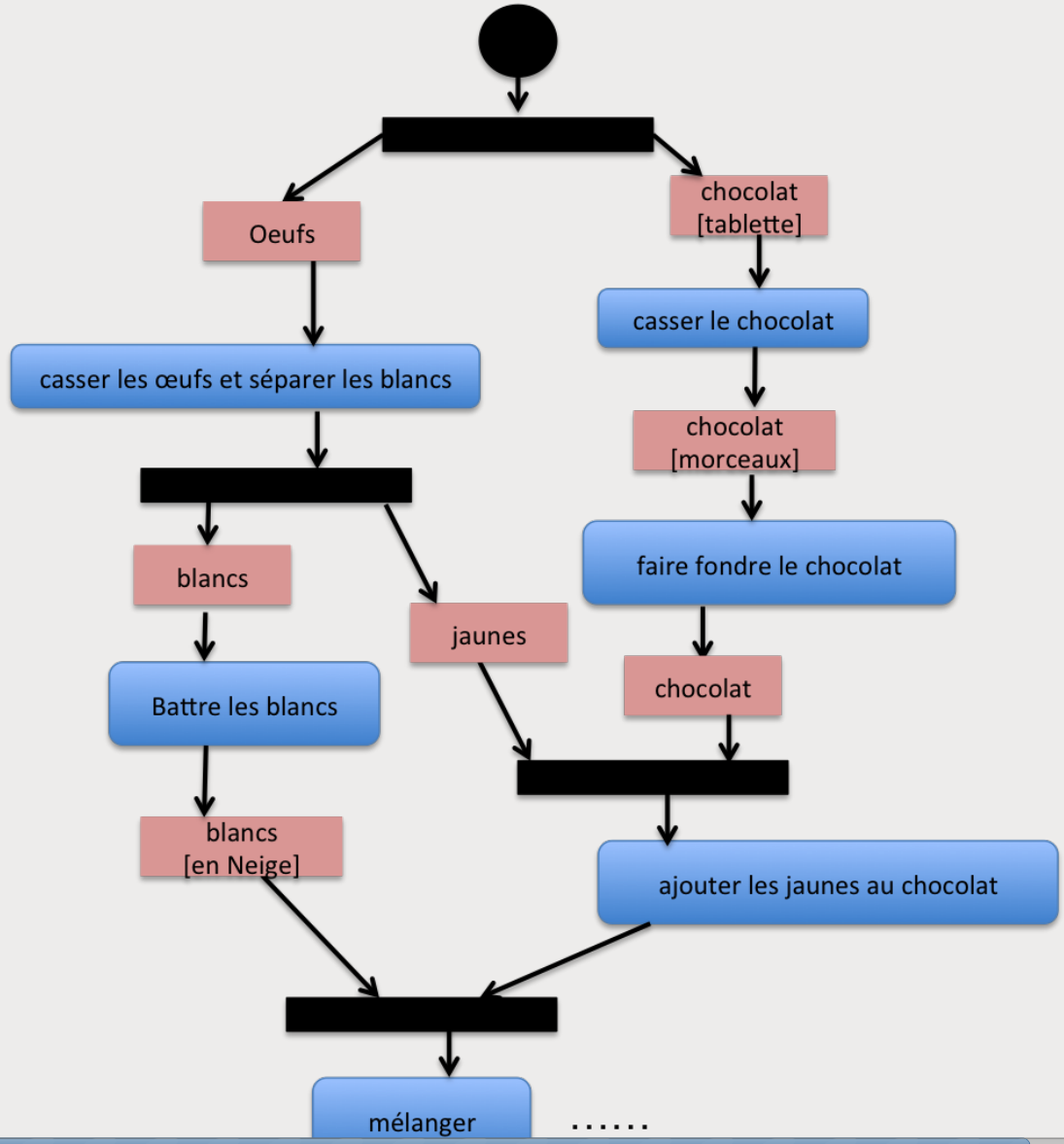
Example

Recette de cuisine

- Start by breaking the chocolate into pieces, then melt it.
- In parallel, break the eggs by separating the whites from the yolks.
- When the chocolate is melted, add the egg yolks.
- Beat the egg whites until they are stiff.
- Incorporate them gently into the chocolate mixture without breaking them.
- Pour into individual ramekins.
- Refrigerate before serving.

What are the ingredients handled?

Example

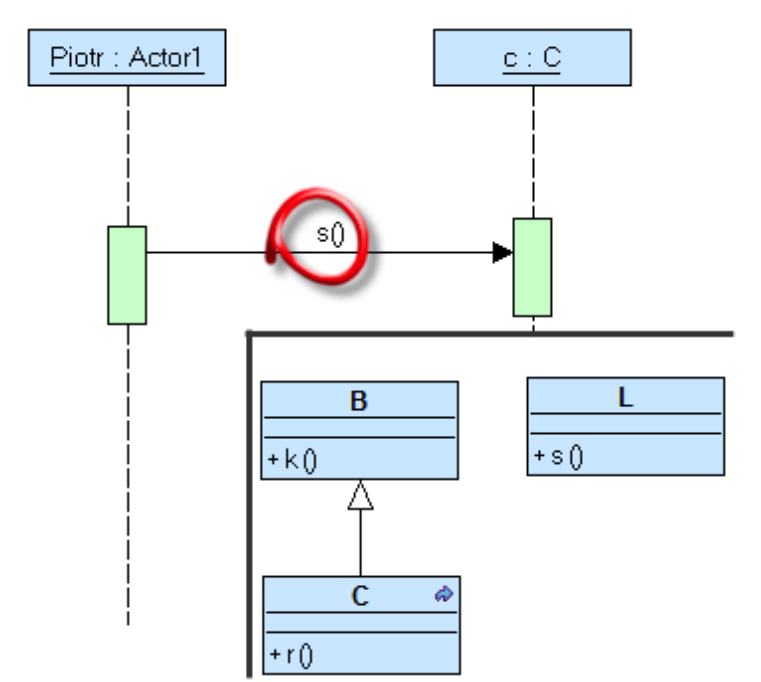

Oeufs
chocolat [tablette]
casser le chocolat
casser les œufs et séparer les blancs
chocolat [morceaux]
faire fondre le chocolat
blancs
jaunes
chocolat
Battre les blancs
blancs [en Neige]
ajouter les jaunes au chocolat
mélanger
.....

Consistencies

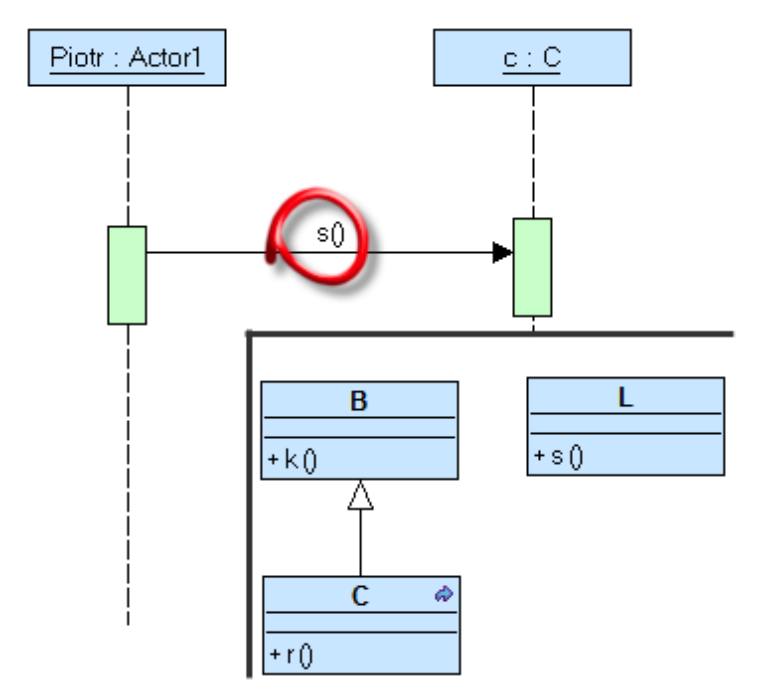
Different diagrams describe the same system from different point of view, so assure consistencies between them

- Class diagram
- Use case diagram
- Interaction diagram
- State diagram
- Activity diagram

Inconsistency example

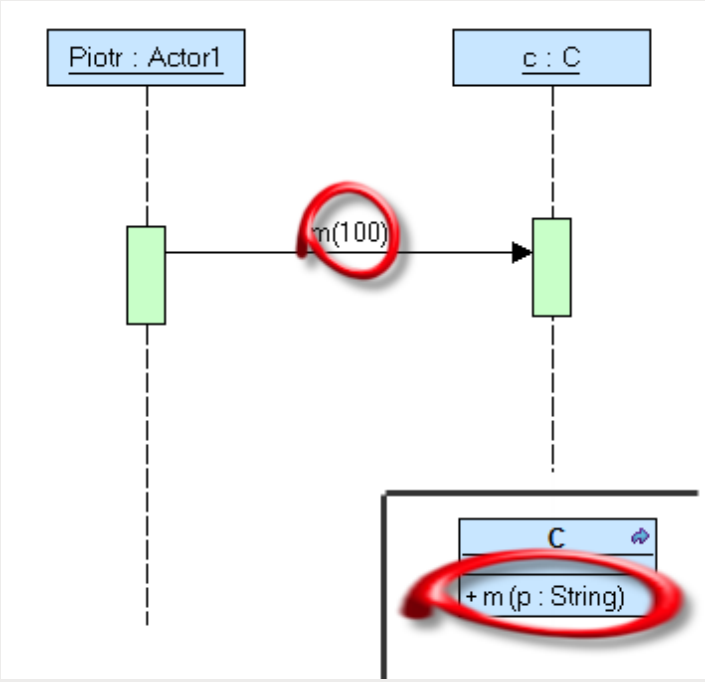


Inconsistency example

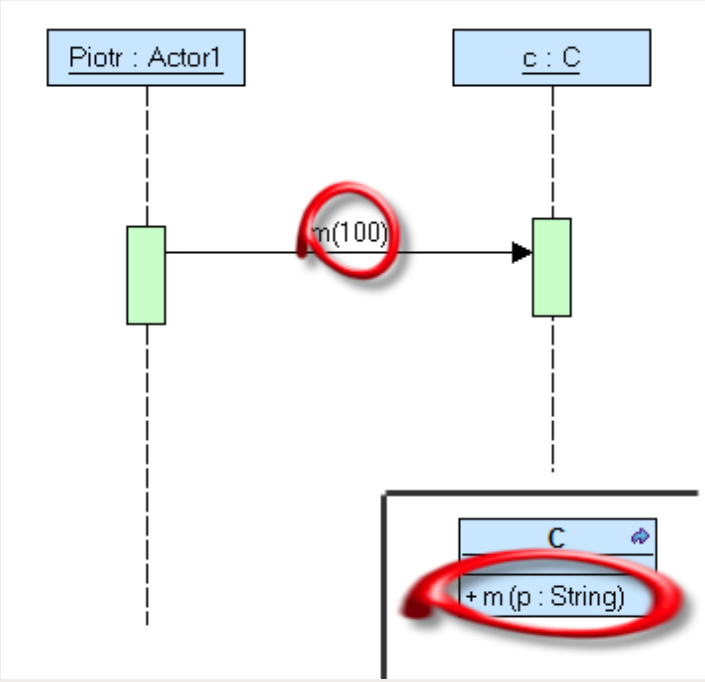


Call of an operation that is not declared in a target class.

Inconsistency example

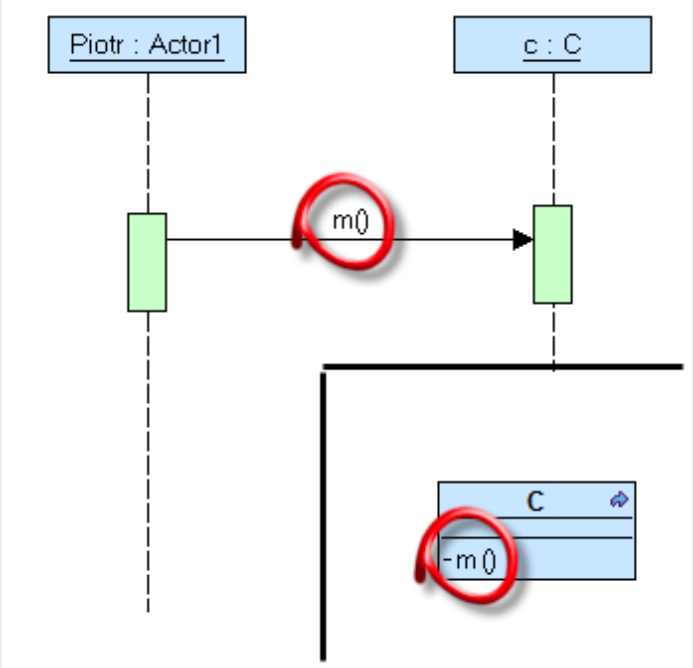


Inconsistency example

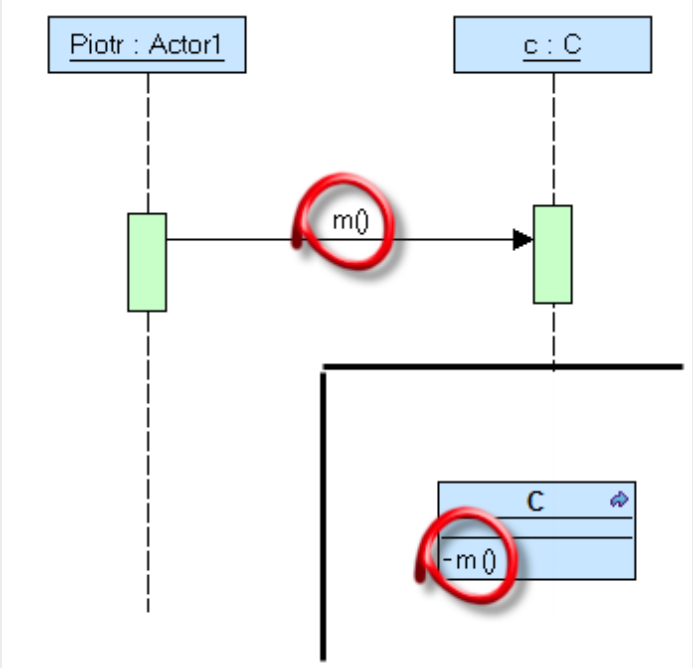


Inconsistency between a type of an argument and a type of a parameter.

Inconsistency example

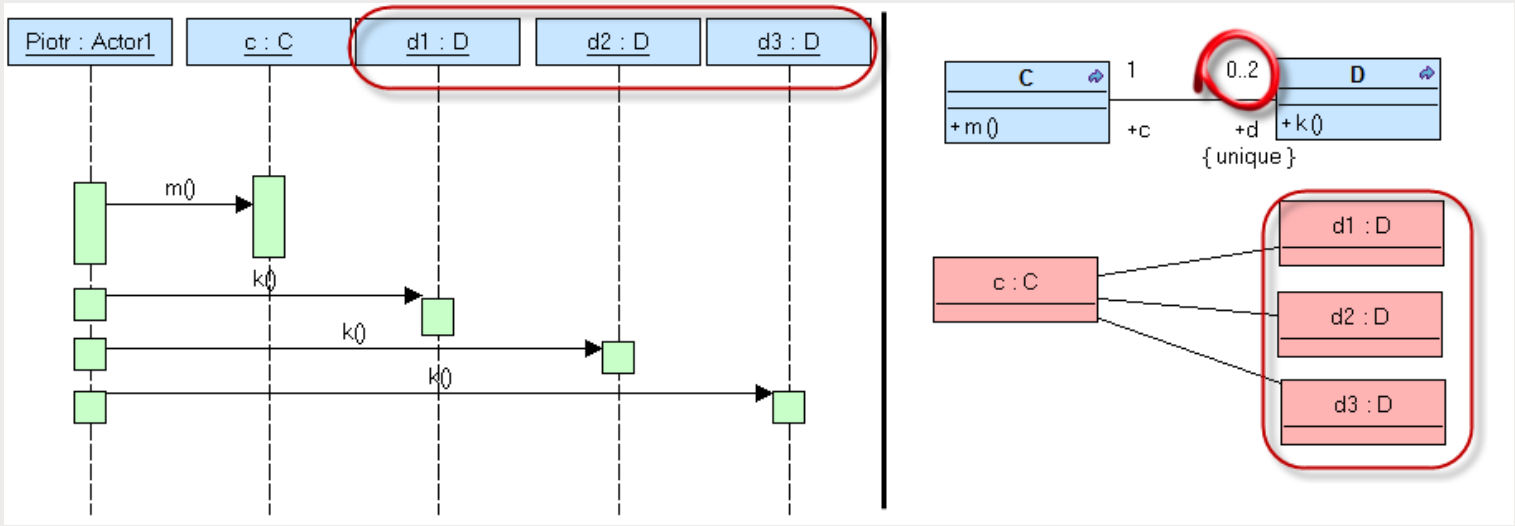


Inconsistency example

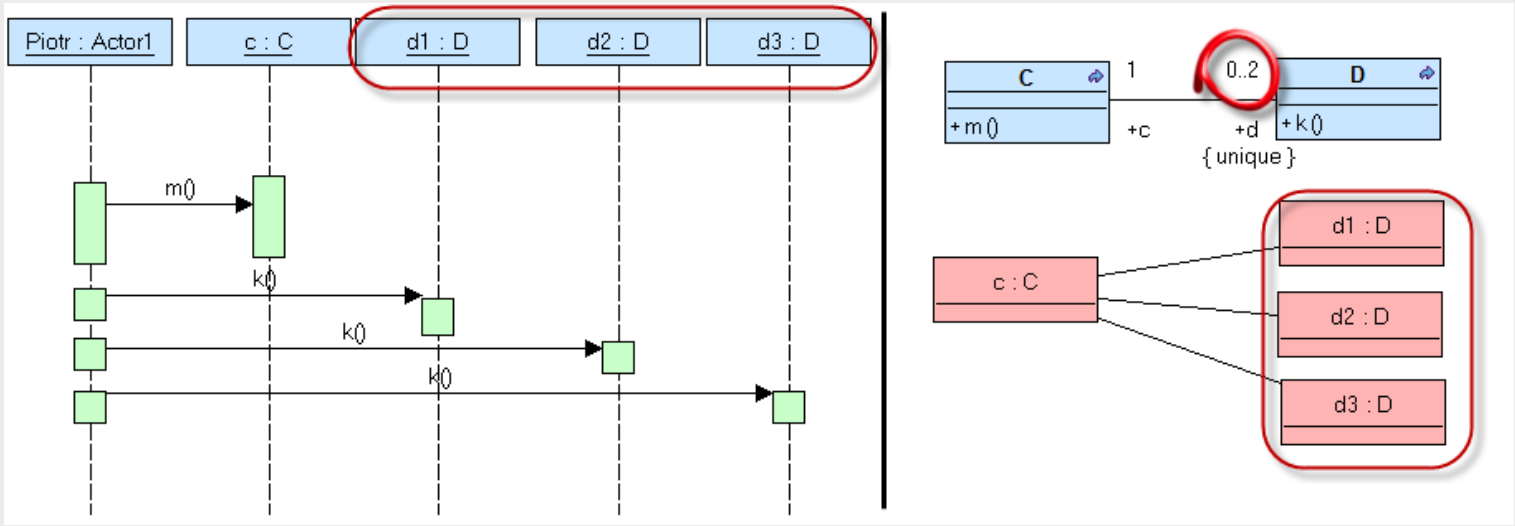


The call to a private operation. Private operations are recognized by the minus sign.

Inconsistency example

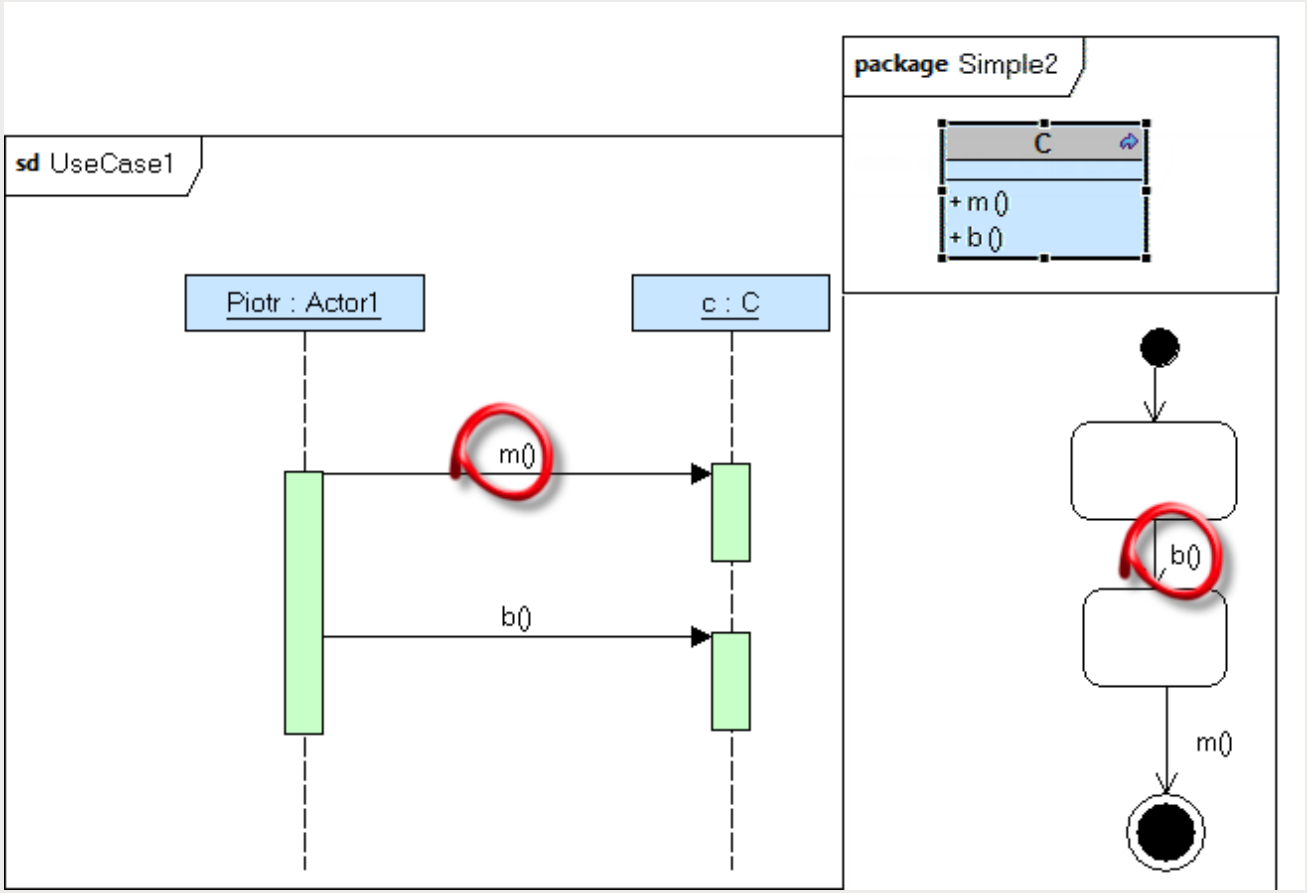


Inconsistency example

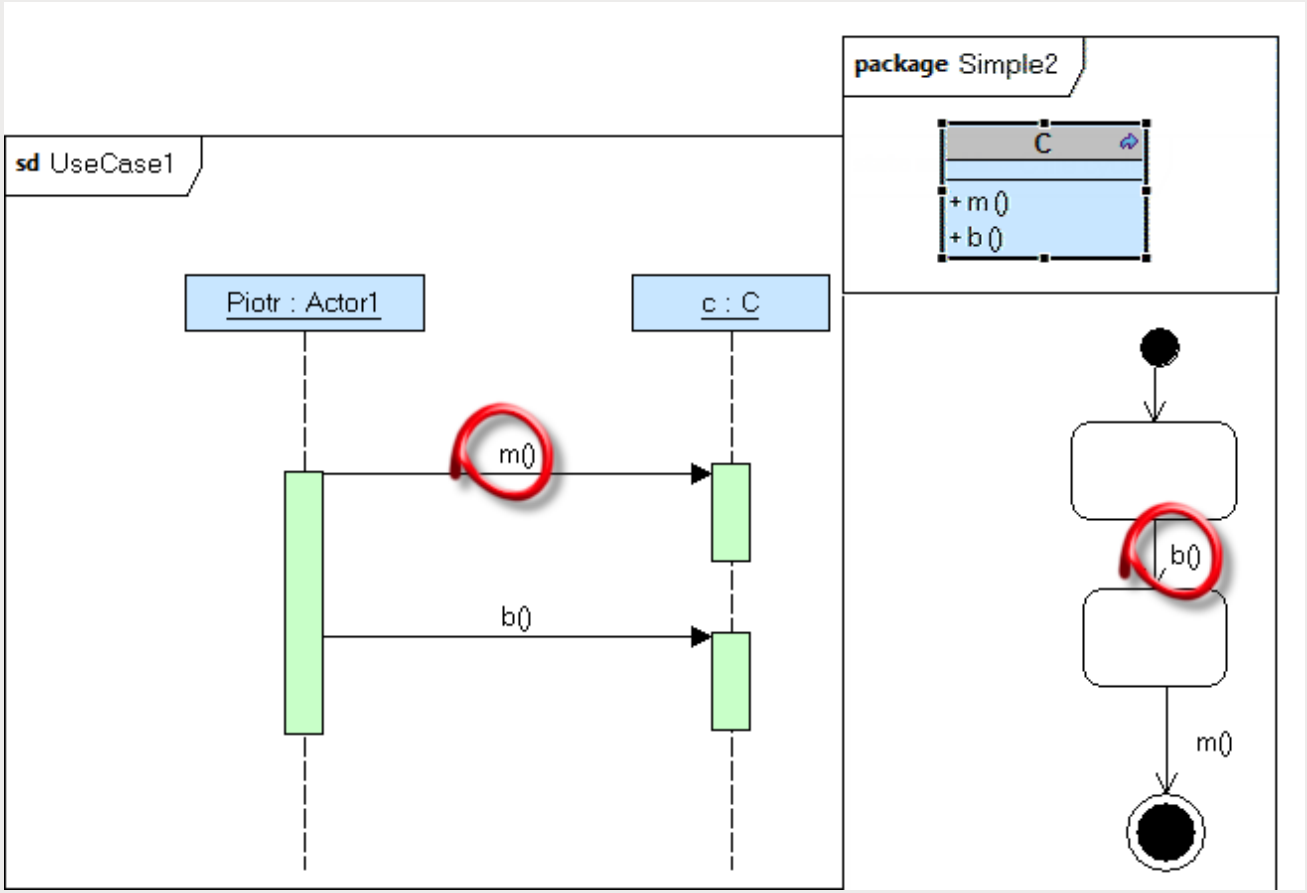


Inconsistent number of objects w.r.t. an association's end multiplicity.

Inconsistency example



Inconsistency example



The order of the messages in the scenario's sequence diagram cannot be realized by the state machine order of triggers.