

ANALYSE ET CONCEPTION ORIENTÉE OBJET (UML)

ANNEE DE FORMATION
2019 - 2020

Animateur : Belkassem ECHCHADLI

UML

Diagramme d'états-transitions

PLAN DE L'UNITÉ

- Définitions
- Élément du diagramme
 - États
 - Événement
 - Transition
- Exemples
- Point de décision
- Étude de cas

DESCRIPTION

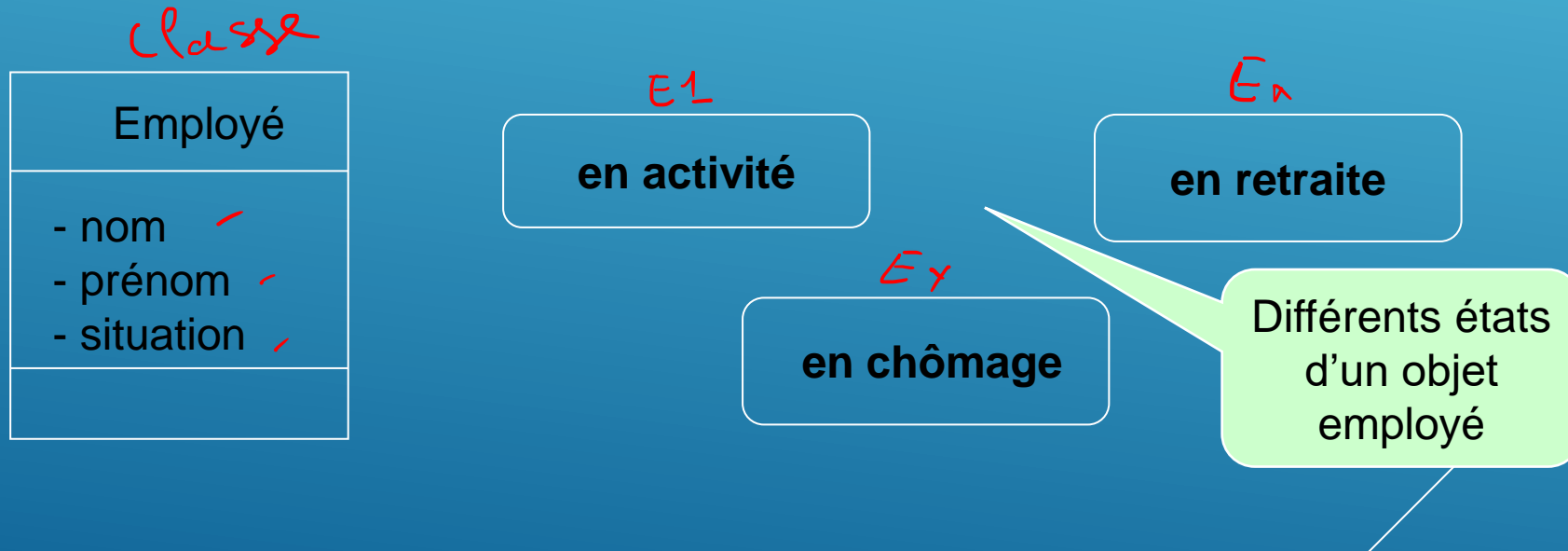
- Le diagramme d'états-transitions décrit le comportement dynamique des objets dans le temps, en modélisant les cycles de vie de chaque instance.
- le diagramme d'état est attaché à une classe (ou à un cas d'utilisation), il présente une classe par rapport à ses états possibles et aux transitions qui la font évoluer.
- Il spécifie les états, les transitions entre ces états et les actions associées aux transitions.

ETAT

objet

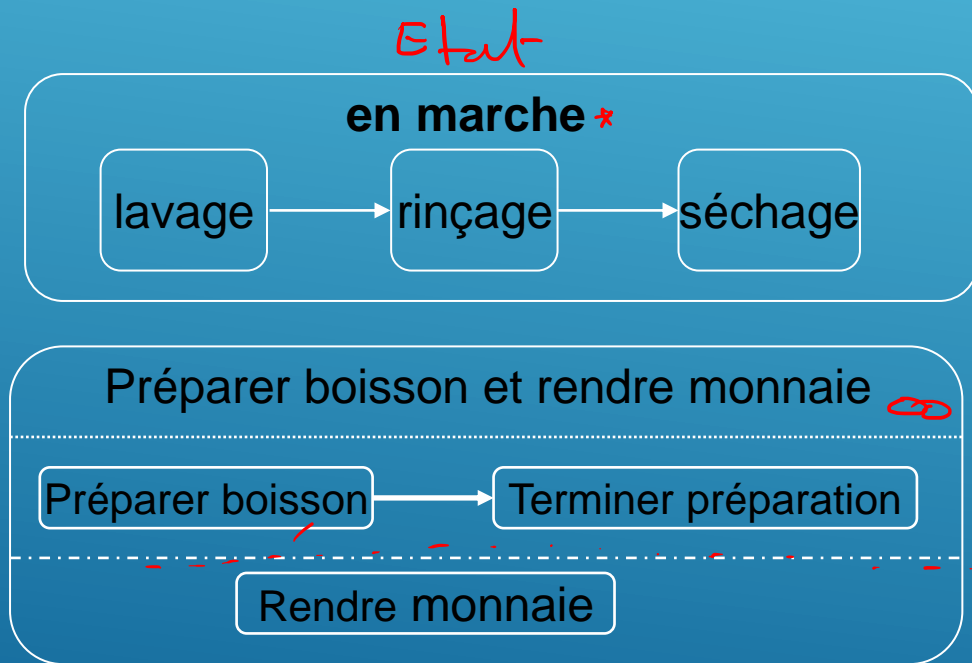
Etat ←
= Σ valeurs
des attributs
instantané

- C'est une situation stable dans la vie d'un objet.
- L'état d'un objet est défini par les valeurs de ses attributs.
- Un état doit avoir une signification dans le domaine.
- Un état se représente graphiquement par un rectangle aux coins arrondis.
- Certains états, dits **composites**, peuvent contenir des sous-états. ?



ÉTAT COMPOSITE

- L'état d'un objet est constitué par le jeu de ses sous états qui sont actifs à un instant donné. Si cette configuration contient plusieurs états, il y a concurrence au sein de l'objet.
- Un état composite est partitionné en plusieurs compartiments séparés par une ligne horizontale qui peuvent recevoir des transitions internes ou des sous-états.



Indication que l'état en marche possède des sous états.

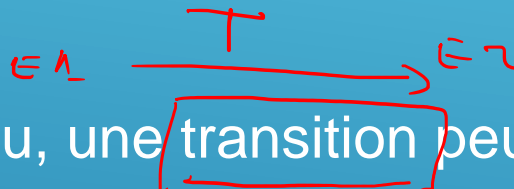
ÉTAT FINAL ET ÉTAT INITIAL

- L'état initial indique l'état de départ, par défaut, lorsque le diagramme d'états-transitions est invoqué.
- L'état final indique que le diagramme d'états-transitions est terminé.
- Il faut toujours un état initial, il peut y avoir plusieurs états de fin.



ÉVÉNEMENT (1/3)

= *changement d'un valeur d'attributs*

- Un objet passe d'un état à un autre suite à un événement, certains événements pouvant ne pas provoquer de changement d'état.
- Un événement est quelque chose qui se produit pendant l'exécution d'un système et qui mérite d'être modélisé.
- Quand un événement est reçu, une transition peut être déclenchée et faire basculer l'objet dans un nouvel état.


```
graph LR; E1 -- T --> E2
```
- Un événement étant une occurrence, on peut donc, si nécessaire, les regrouper en classes.
- Il existe différents types d'événements: call, signal, when, after.

ÉVÉNEMENT (2/3)

- Un événement de type signal est un message asynchrone circulant entre deux objets.
- Un événement de type call représente la réception de l'appel d'une opération par un objet.
Pr IF
- Les événements de type **call** et **signal** sont déclarés comme suit:

<nomEvenement >(< ListeParametres >)

accélérer (180) *70* Etat

ÉVÉNEMENT (3/3)

- Un **événement de changement** est causé par la satisfaction d'une expression booléenne. La notation UML de ce type d'événement est le mot-clé **when**. *Si*

when (charge de la batterie < limité inférieure)

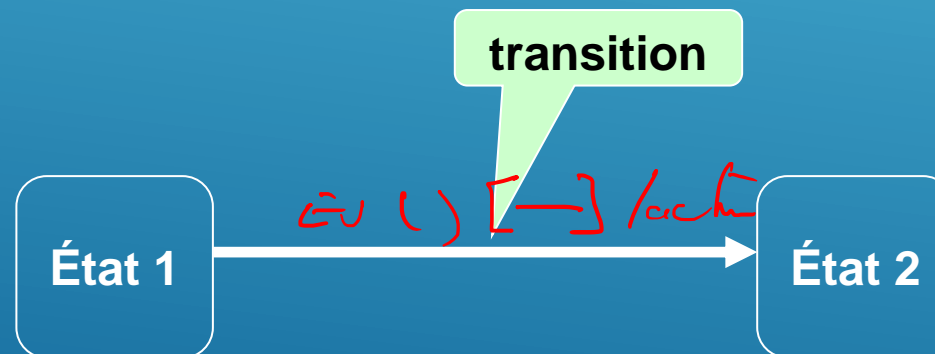
- Un événement temporel de type after est spécifié par :
after (<durée >).

Le paramètre s'évalue comme une durée, par défaut écoulée depuis l'entrée dans l'état courant.

EN after(10 secondes). *ET*

TRANSITION (1/2)

- C'est une relation entre deux états, qui indique qu'un objet peut passer d'un état à un autre quand un certain événement survient.
- Le passage peut être conditionné par une garde (si la condition est vraie alors il y a changement d'état).
- La transition est représentée par une flèche orientée de l'état source vers l'état cible.



TRANSITION (2/2)

- La représentation complète d'une transition est :

nom d'événement (liste de paramètres) [garde] / action

- Garde:** notée entre [], c'est une expression booléenne, paramètre de l'événement déclencheur de la transition entre 2 états.

[moyenne>10]

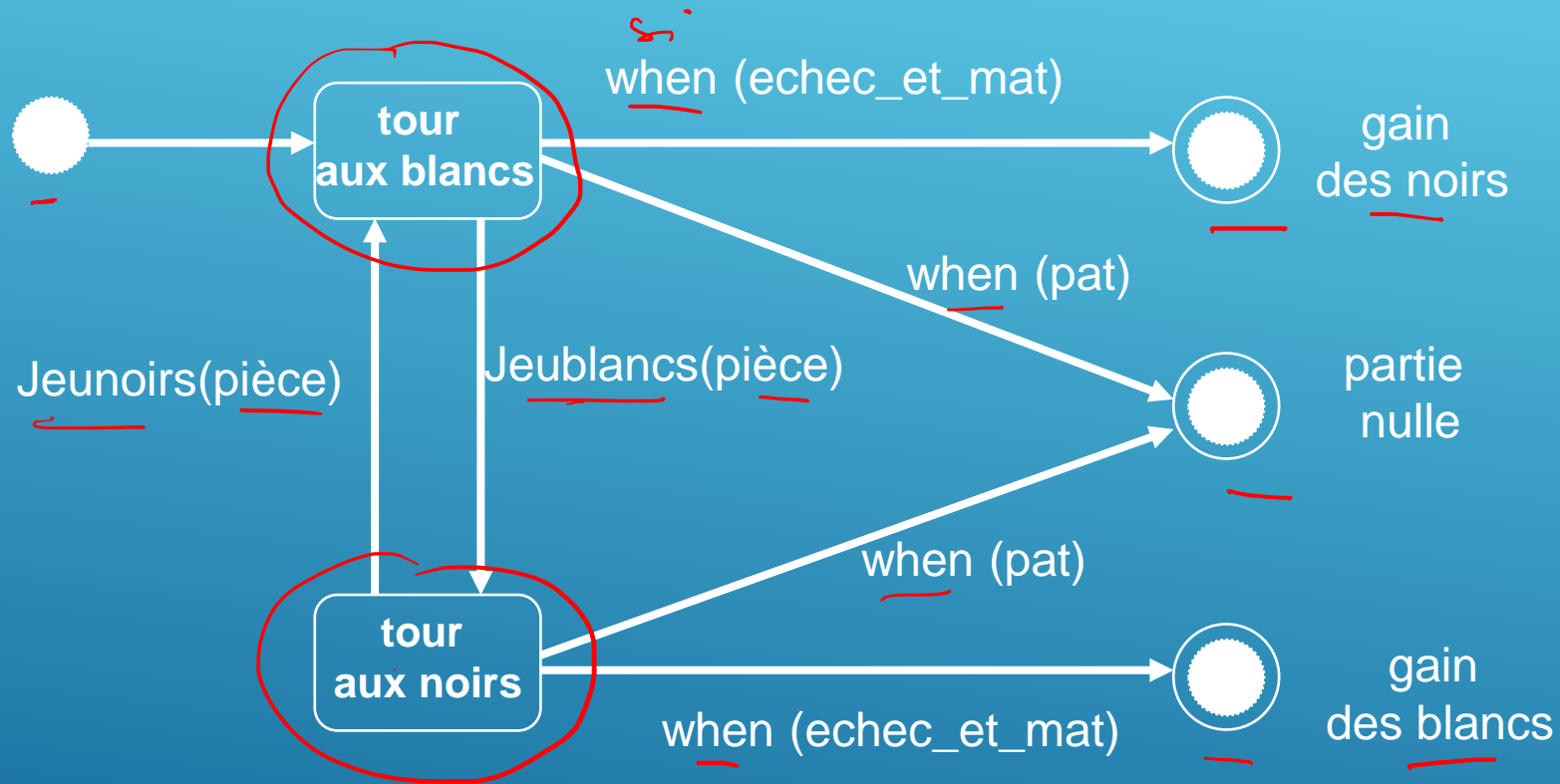
[t=15s]

- Action:** c'est une opération instantanée associée à un événement, qui provoque le changement d'état de l'objet. Elle est notée par un / qui précède le nom de l'action.

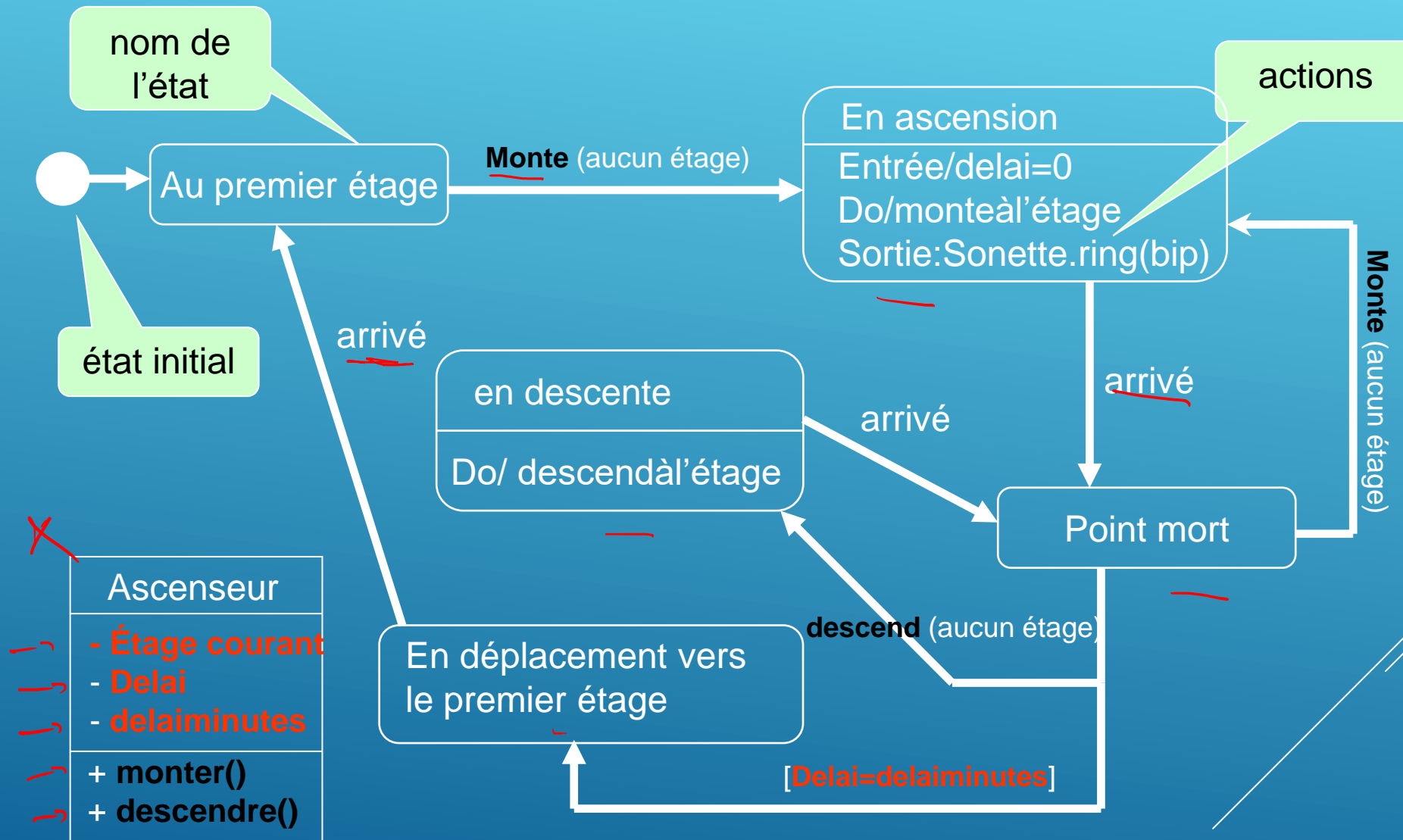
Retrait (montant) [solde>= montant] / solde= solde - montant

Handwritten notes: 250 (under montant), 700 - 250 (above solde=), 450 (under solde=)

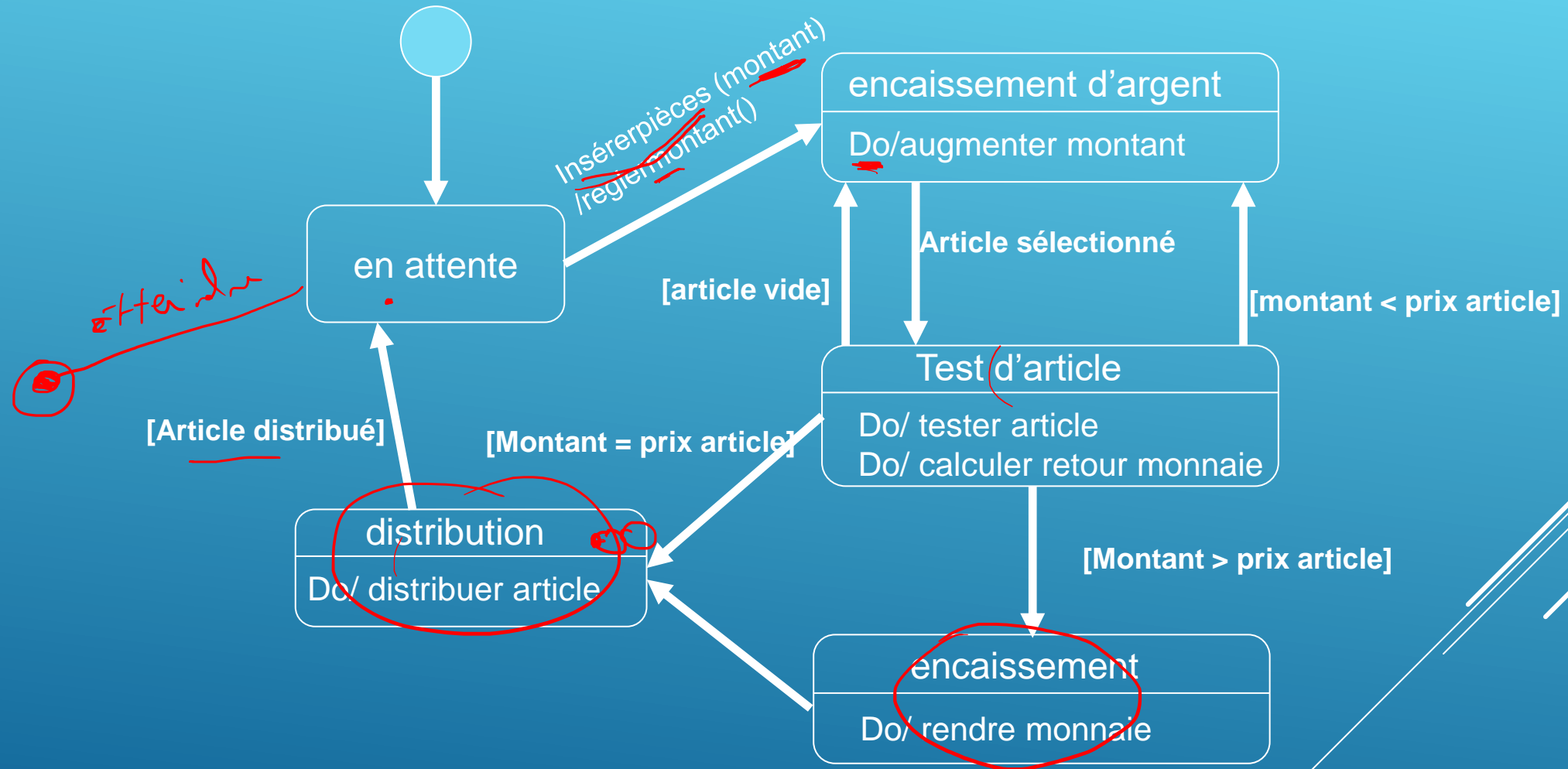
EXEMPLE1: CONTRÔLEUR D'UNE PARTIE D'ÉCHECS



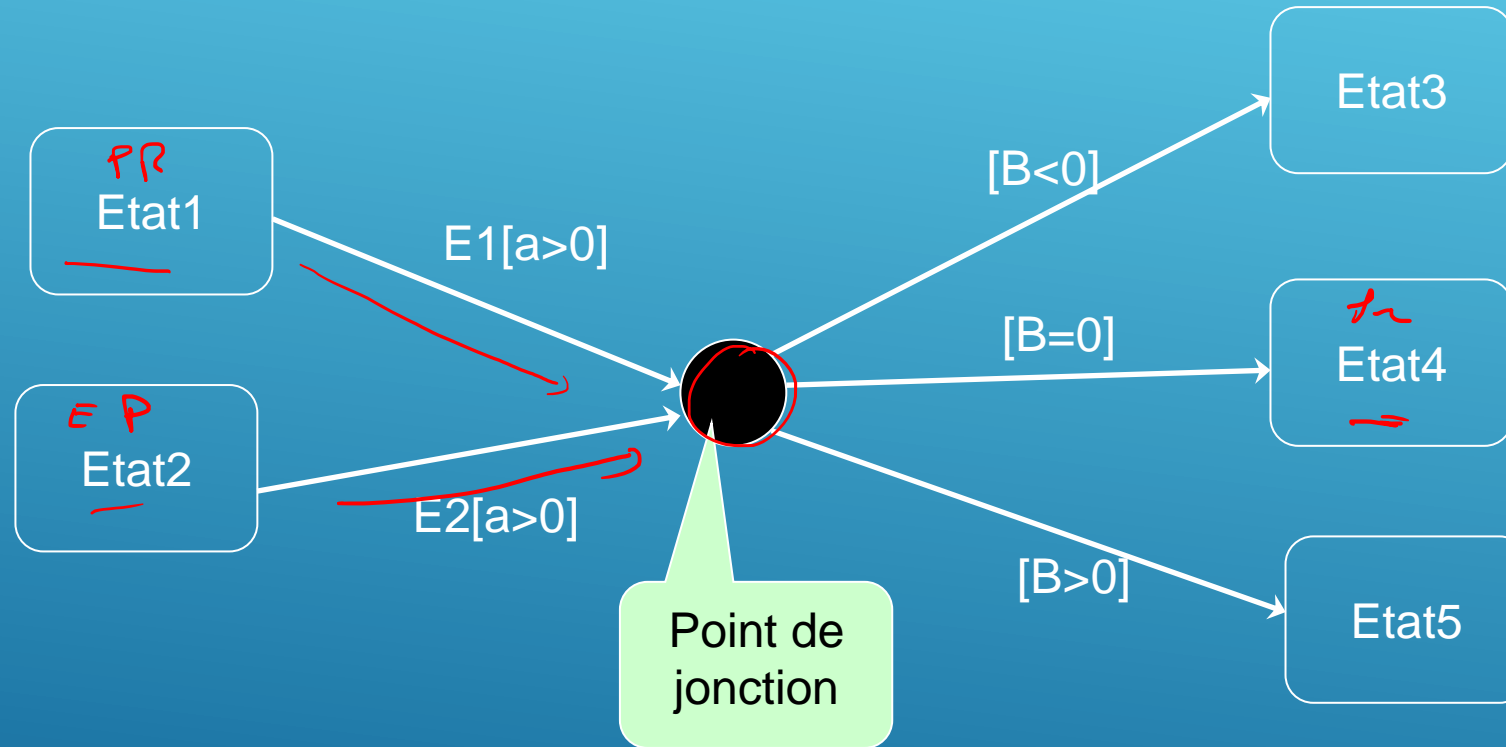
EXEMPLE2



EXEMPLE3: DISTRIBUTEUR DE BOISSONS

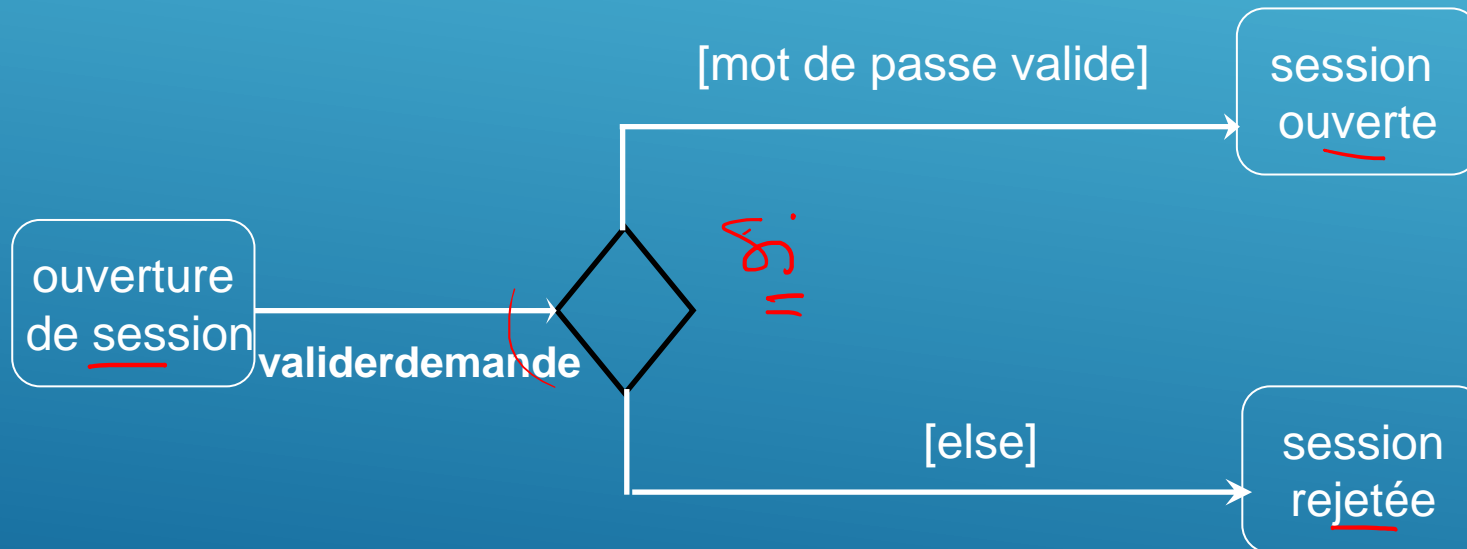


POINT DE JONCTION



POINT DE CHOIX

- C'est un point de décision permettant de simuler un choix (**si-alors-sinon**) qui entraîne deux états différents. Il est représenté par un losange vide.
- Les gardes après le **point de choix** sont évaluées au moment où il est atteint.



ÉTUDE D'UN PUBLIPHONE À PIÈCES

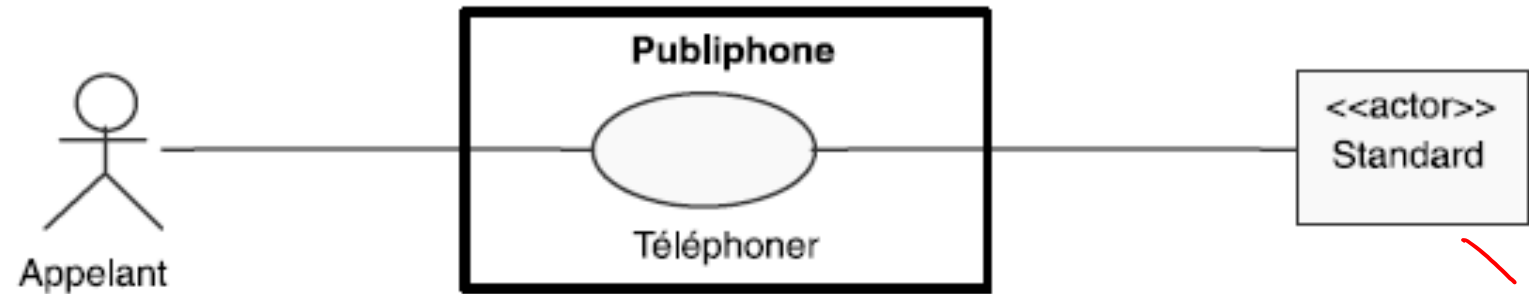
► Cette étude de cas concerne un système simplifié de Publiphone à pièces.

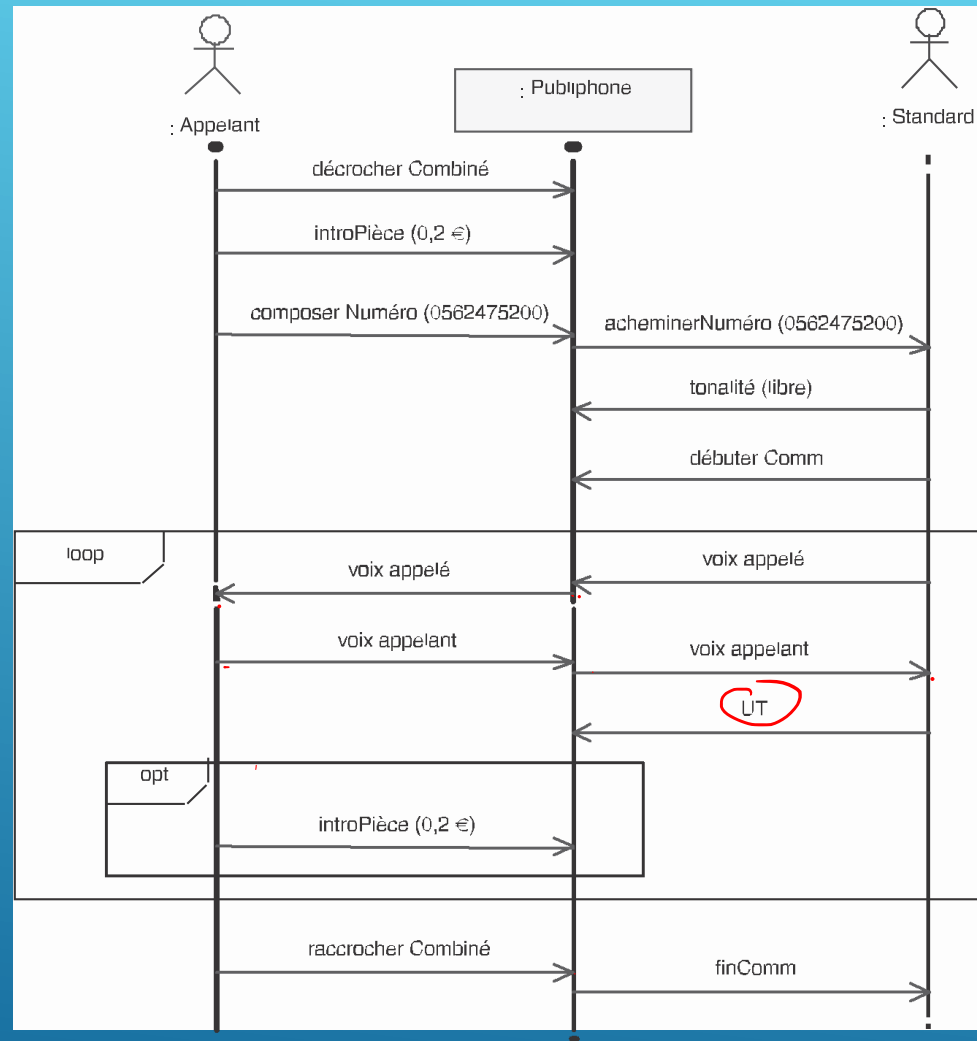
1. Le prix minimal d'une communication interurbaine est de 0,2 euros.
2. Après l'introduction de la monnaie, l'utilisateur a 2 minutes pour composer son numéro (ce délai est décompté par le standard).
3. La ligne peut être libre ou occupée.
4. Le correspondant peut raccrocher le premier.
5. Le Publiphone consomme de l'argent dès que l'appelé décroche et à chaque unité de temps (UT) générée par le standard.
6. On peut ajouter des pièces à tout moment.
7. Lors du raccrochage, le solde de monnaie est rendu.

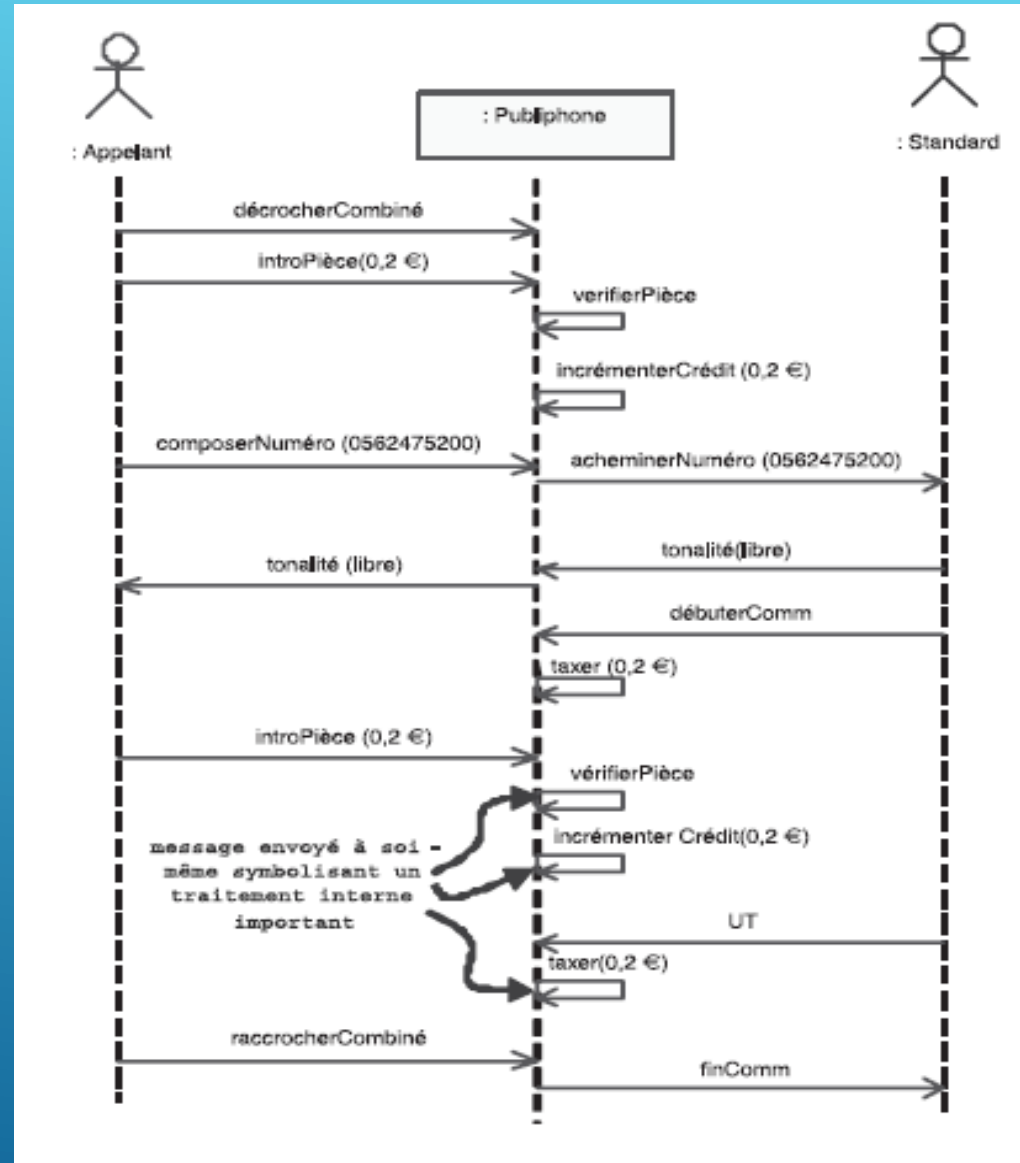
À partir de ces six phrases, nous allons progressivement :

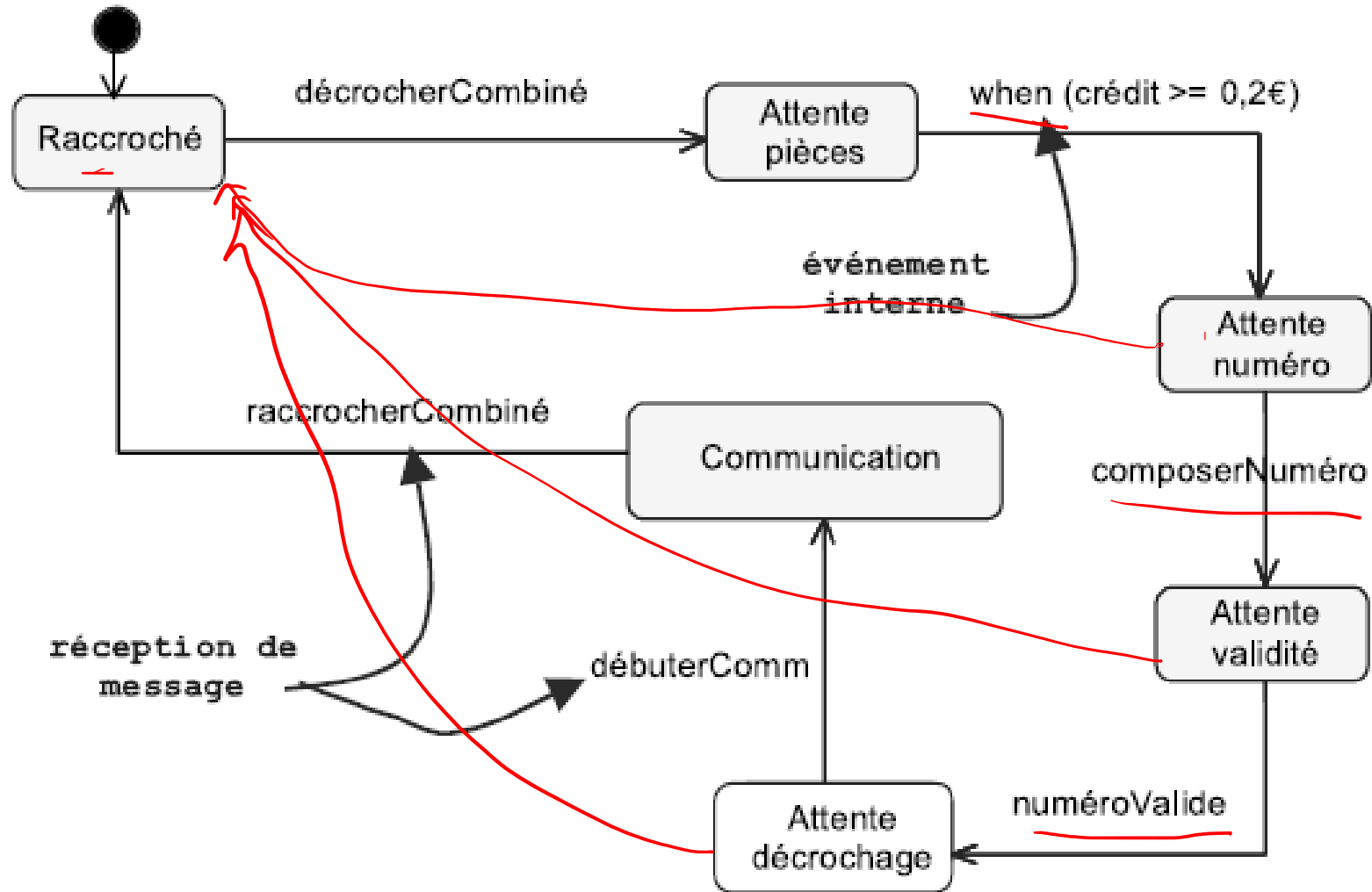
- Identifier les acteurs et les cas d'utilisation.
- Construire un diagramme de séquence système.
- Construire le diagramme d'état de transition avec plusieurs versions.

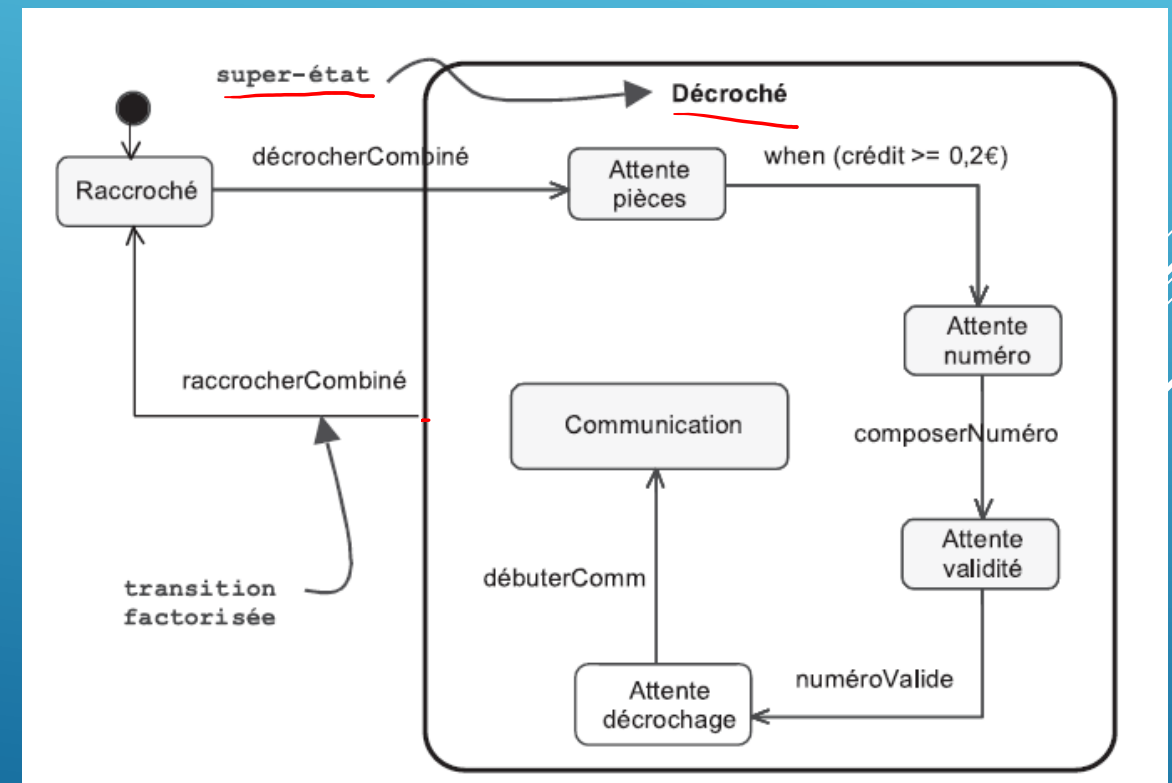
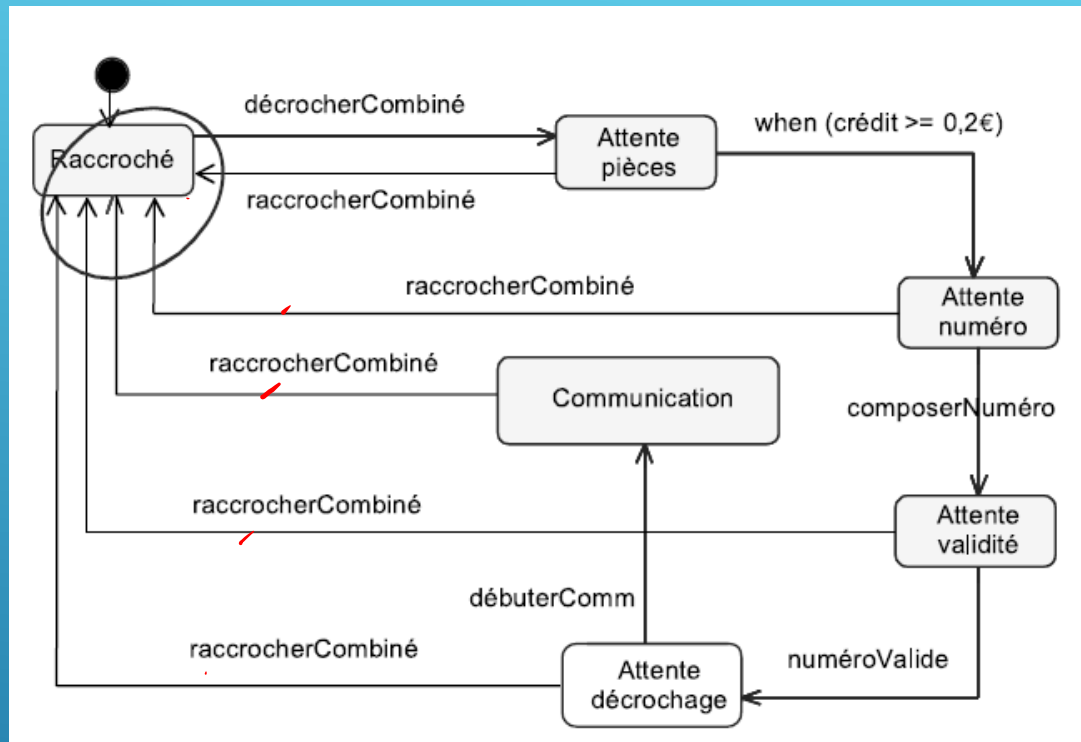
Figure 5-6.
*Diagramme de cas
d'utilisation
du Publiphone*

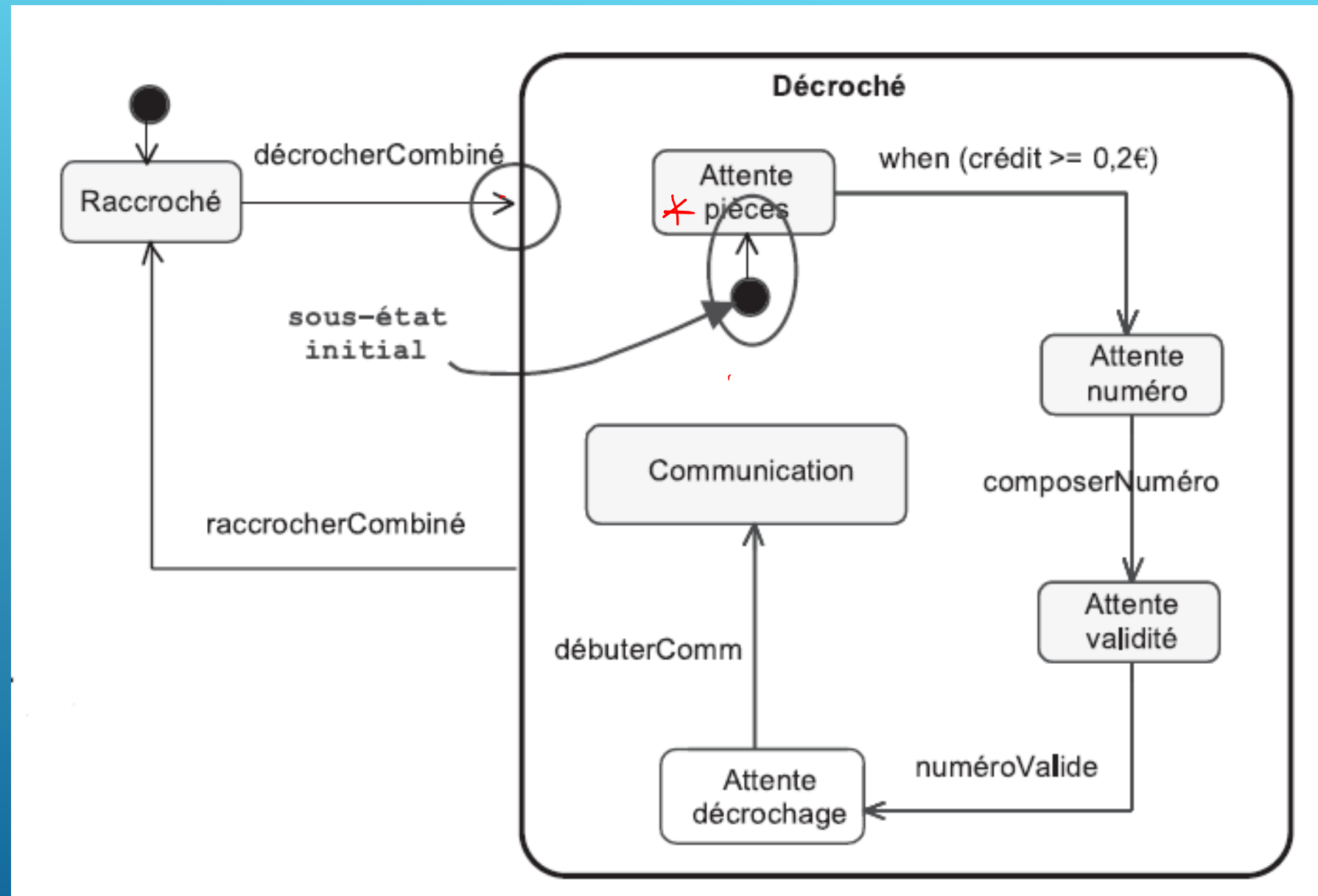


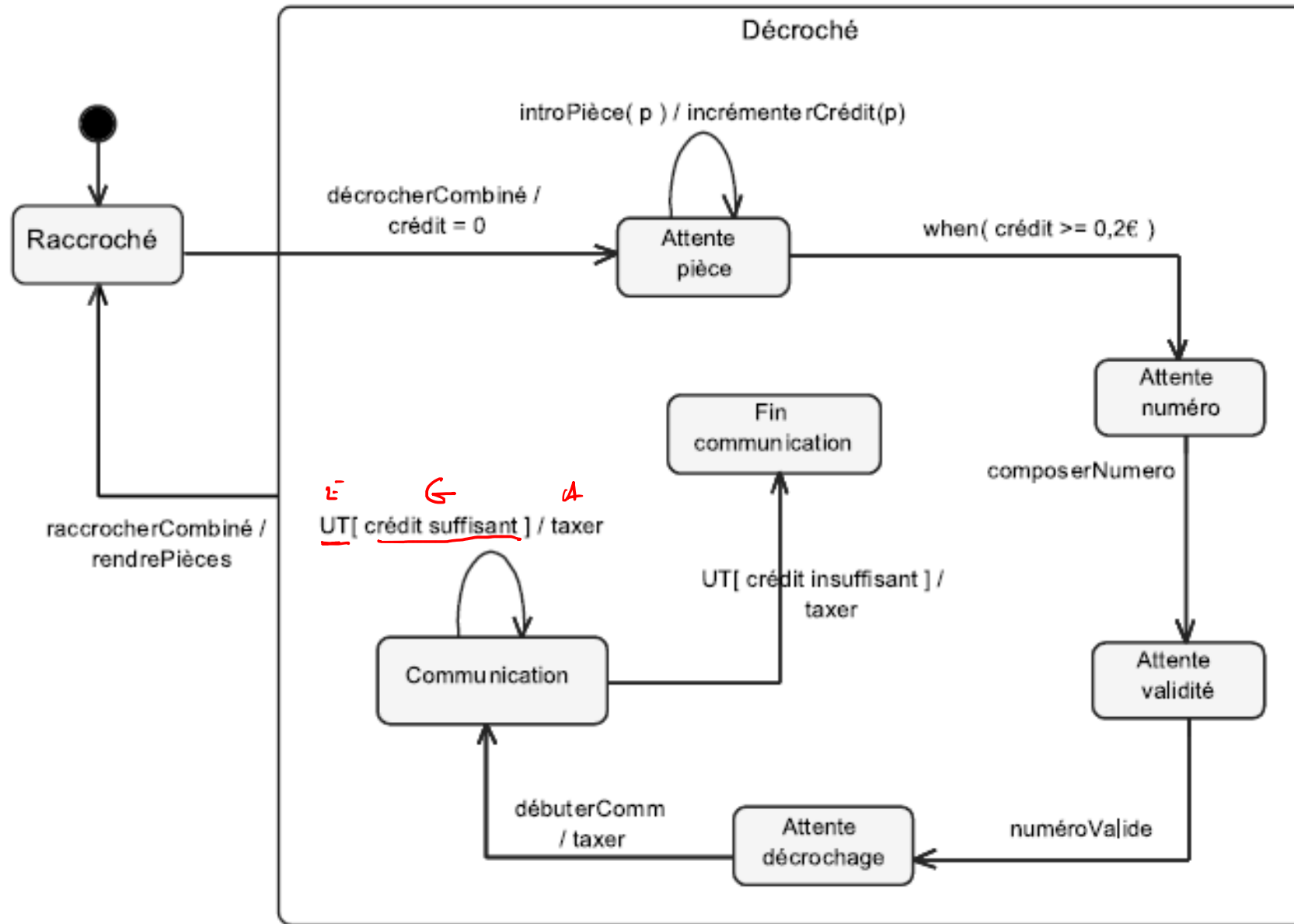


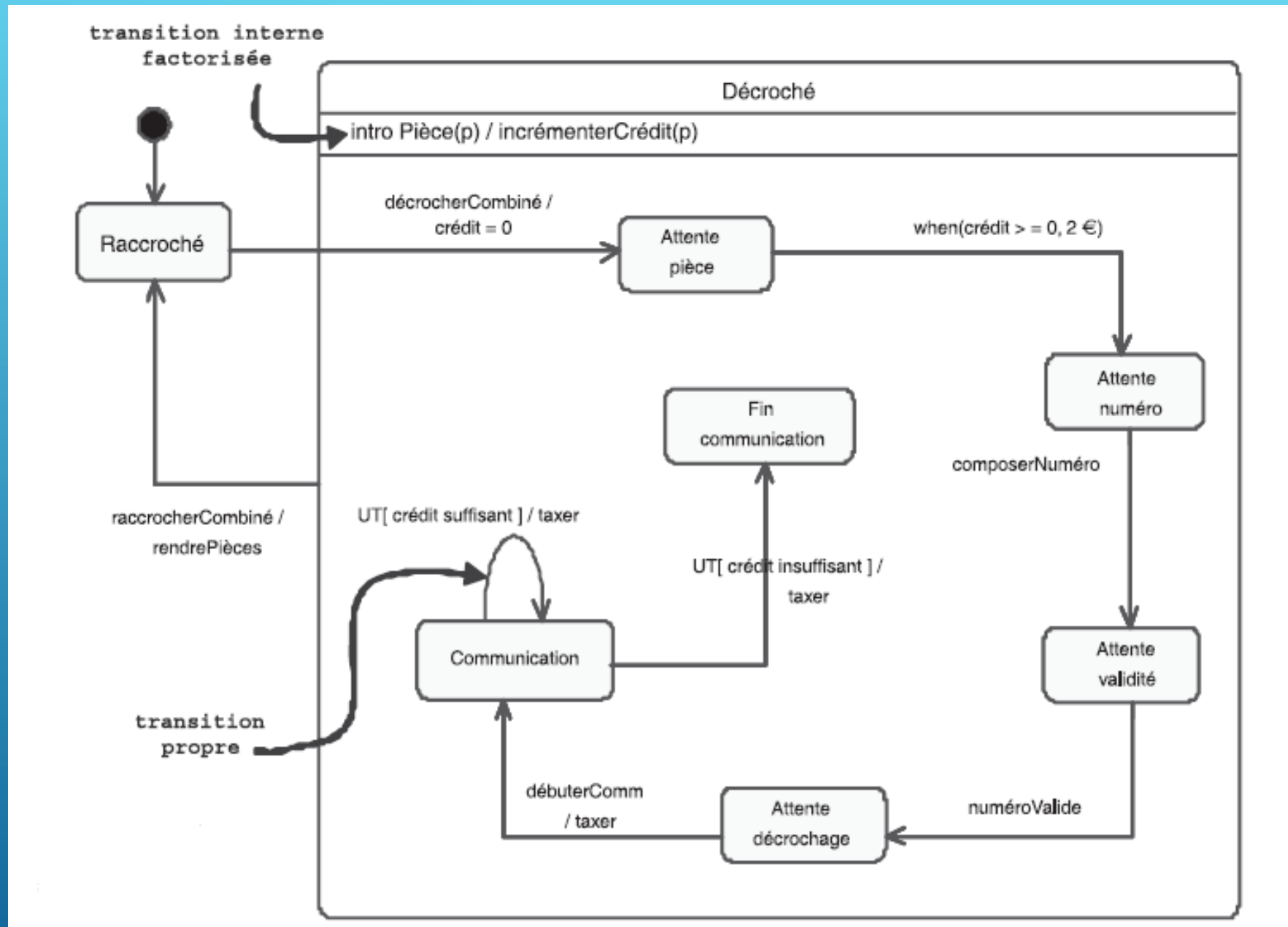


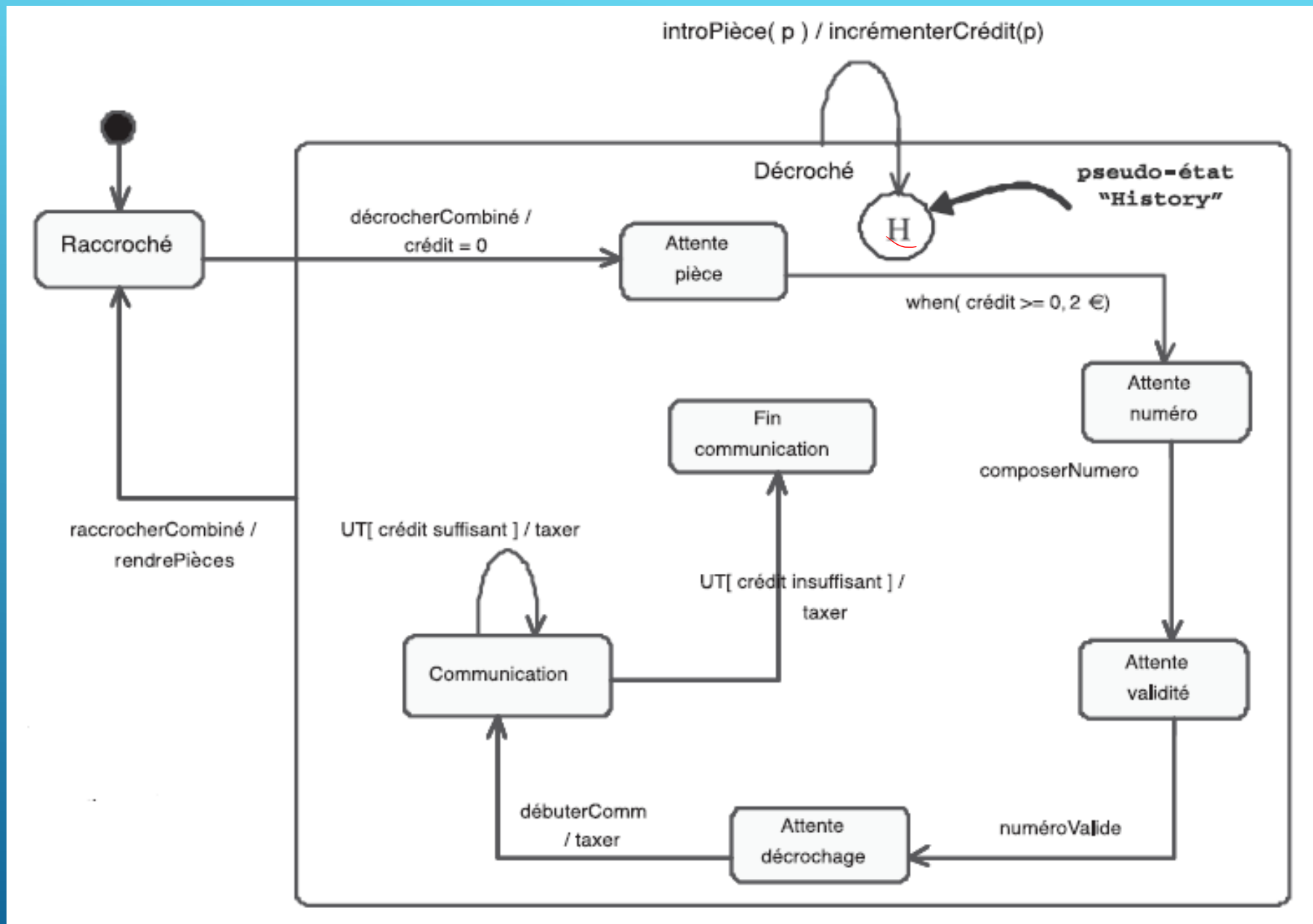


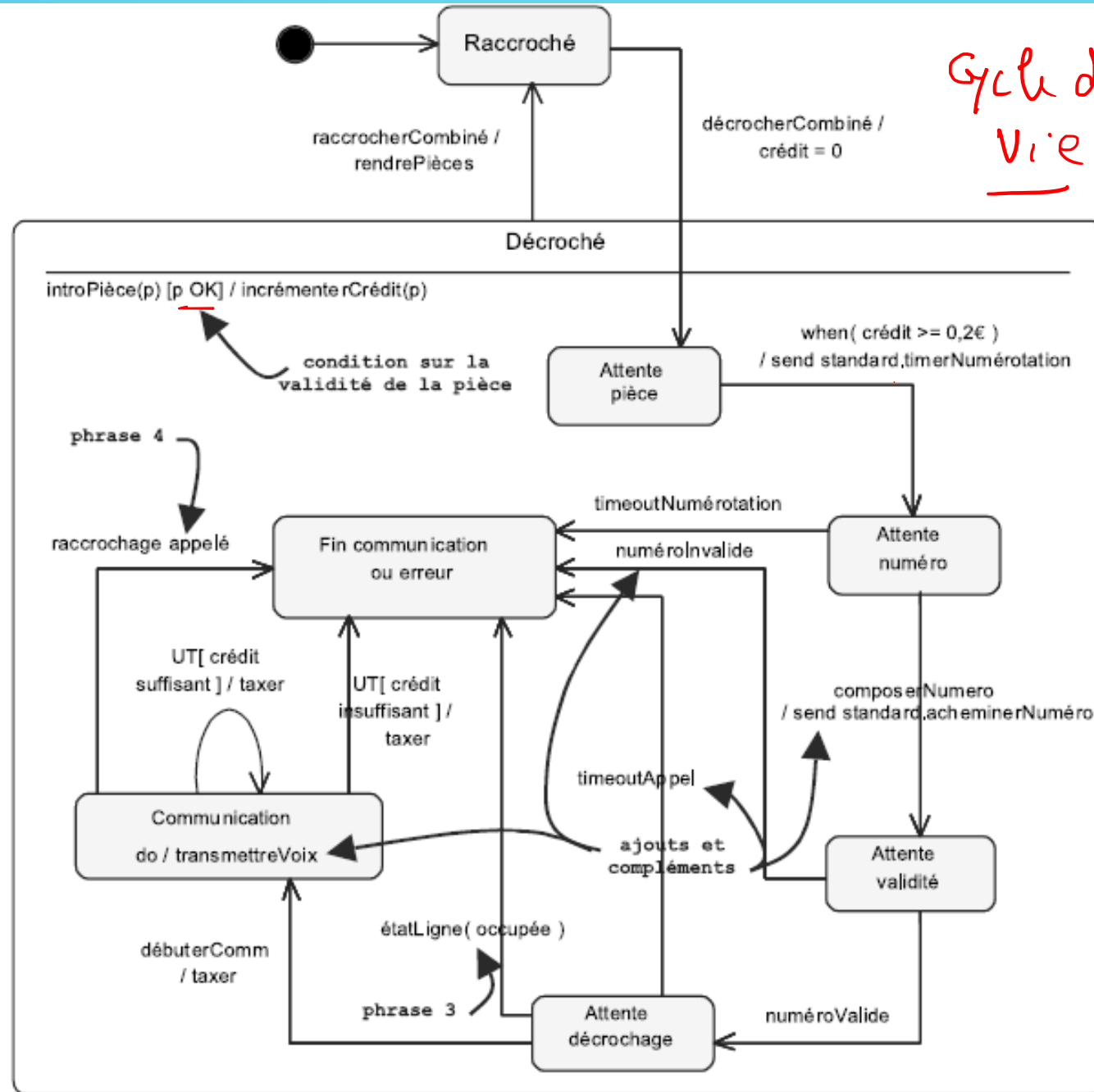












*Gycle de
vie d'un
objet*

Merci De Votre Attention

Animateur : Belkassem ECHCHADLI