

CHAPITRE 5: DIAGRAMME D'ETATS- TRANSITIONS

Dr. Ilhem ABDELHEDI ABDELMOULA

Email: ilhemabdelmoula13@gmail.com

Université de Carthage

Enicarthage - École Nationale des Ingénieurs à Carthage

Département informatique

Niveau : 2^{ème} année INFO ING / 1^{ère} Année INFO ING

Semestre : 2

Année universitaire: 2019 - 2020



UTILITÉ DES DIAGRAMMES D'ETATS-TRANSITIONS DET

- Fait partie des **modèles dynamiques**
- Permet de visualiser le **comportement** d'un **objet** à travers ses différents états possibles, en réponse à des événements au cours de son **cycle de vie**, sous forme d'un automate à états finis.
- Décrit :
 - l'enchaînement de tous les **états** d'un objet
 - Les **événements** qui provoquent les changements d'état
 - Les **transitions** qu'ils effectuent



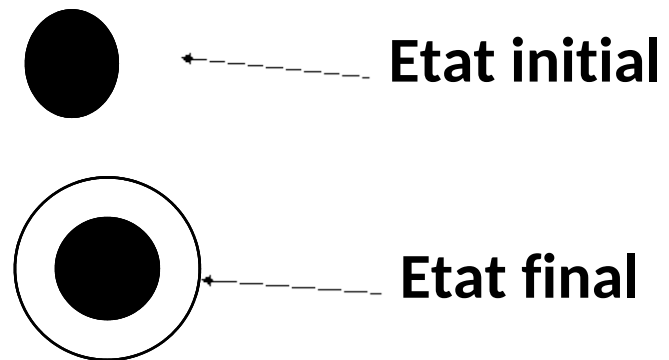
DÉFINITION D'UN ETAT

- Il représente une conjonction instantanée des **valeurs des attributs** d'un objet.
- Il dépend aussi de l'état précédent et de l'événement survenu
- Il se caractérise par **une durée** et **une stabilité** = l'objet est toujours dans un état donné pour un certain temps

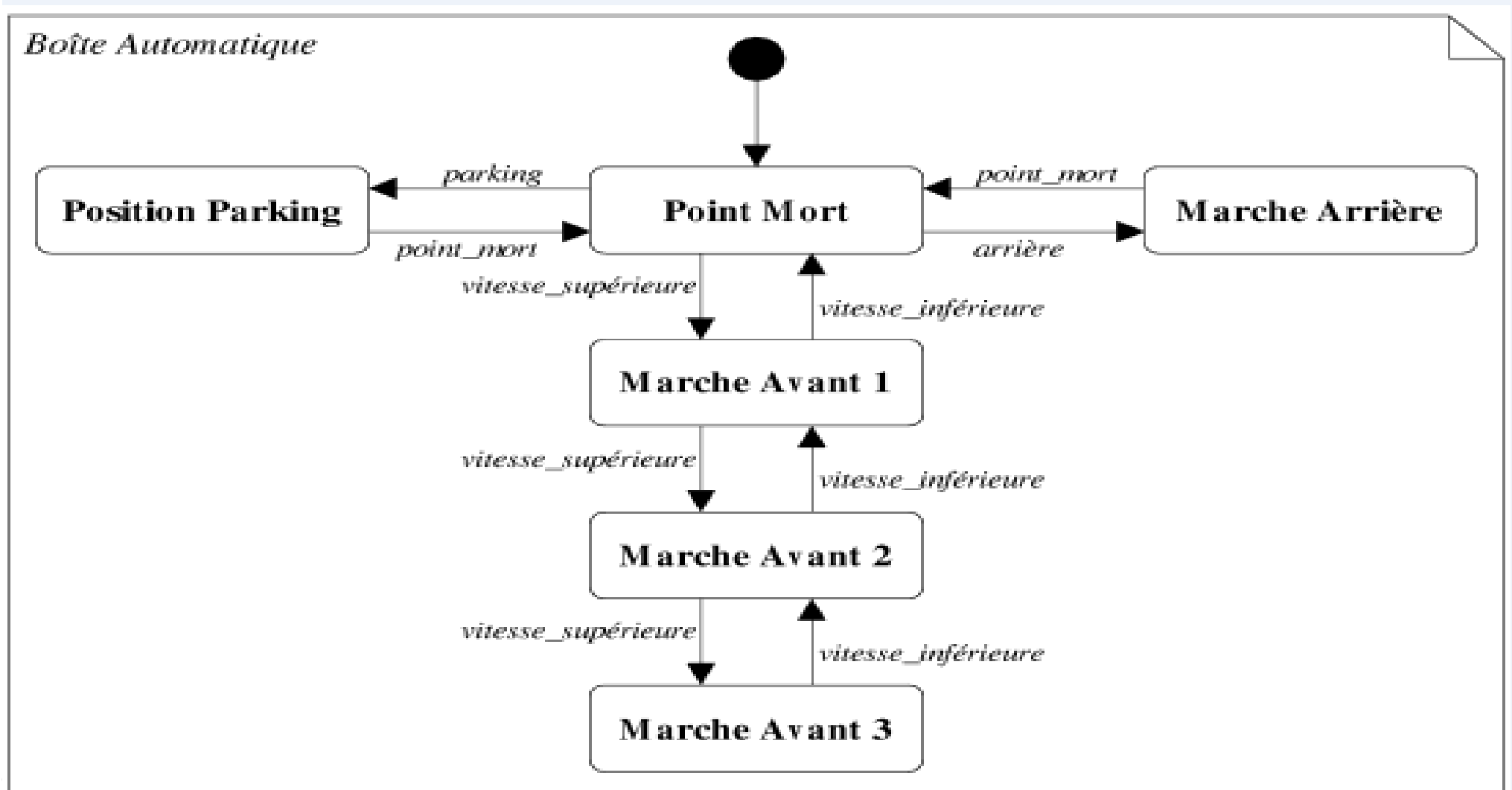


DEUX ÉTATS PRÉDÉFINIS

- Un diagramme état-transition débute toujours par un **état initial** : il est **obligatoire et unique**. Lorsque l'objet est créé, il entre dans l'état initial.
- Il se termine par **un ou plusieurs états finaux** (sauf où le diagramme représente une boucle)



EXEMPLE



FORMALISME

Un état se caractérise par :

- **Nom** : Chaîne de caractères
- **Activités** (opérations **ou actions**) qu'effectue l'objet tant qu'il se trouve dans cet état donné.
 - **Activités** : représentée par : **Do/**
 - **Actions d'entrée/sortie** : Opération instantanée non interrompue que l'objet exécute
 - A l'entrée de l'état : **Entry/**
 - A sortie de l'état : **Exit/**



ACTIONS

Certaines actions peuvent être rattachées à un état au lieu d'une transition:

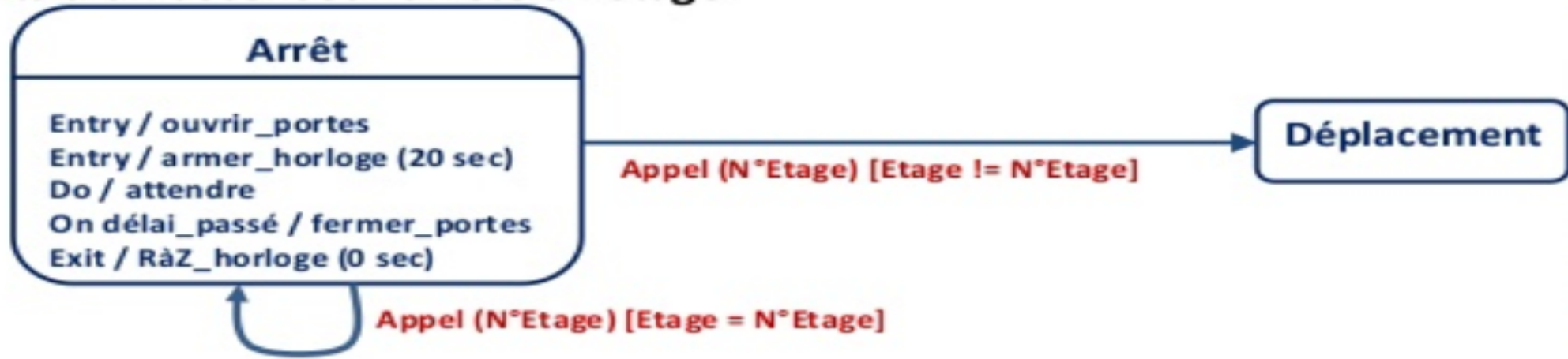
1. Action d'entrée: notée **entry/action (ou entrée/action)** action exécutée systématiquement dès l'entrée dans l'état (quelle que soit la transition qui nous y amène..
2. Action de sortie: notée **exit/action (ou sortie/action)** action exécutée systématiquement à la sortie de l'état (quelle que soit la transition qui nous en fait sortir).
3. Action d'exécution : notée **do/action (ou faire/action)** exécute une activité, càd une opération qui prend un temps significatif lorsque l'objet est dans cet état.
4. Action sur événement interne : notée **event événement/action** action exécutée à suite de la génération d'un événement interne (ne provoque pas de changement d'état).

saisie mot de passe

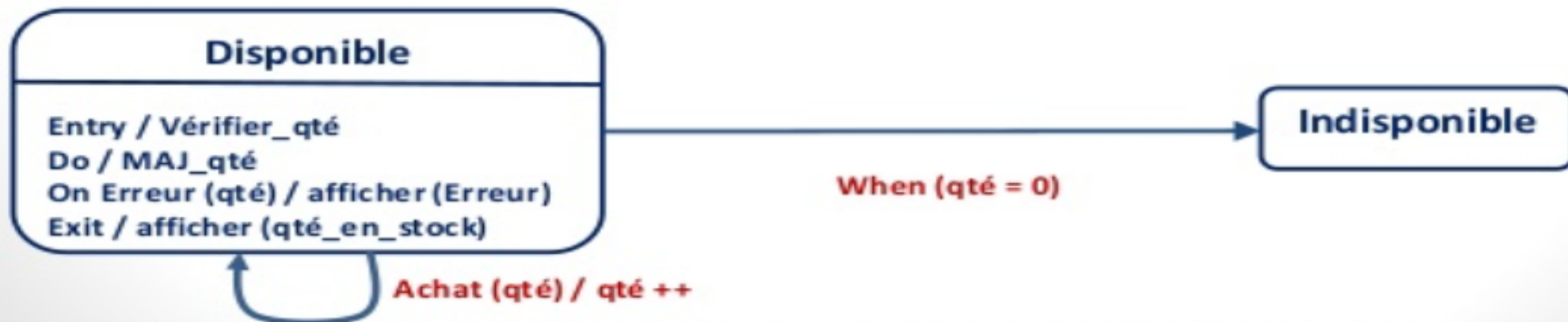
entry/ désactiver l'affichage des caractères
exit/ rendre l'affichage des caractères norm...
do/ saisir un caractère
event saisie d'un caractère/ valider caractère
event after(20s)/ exit

EXEMPLES

Etats d'un ascenseur arrêté à l'étage

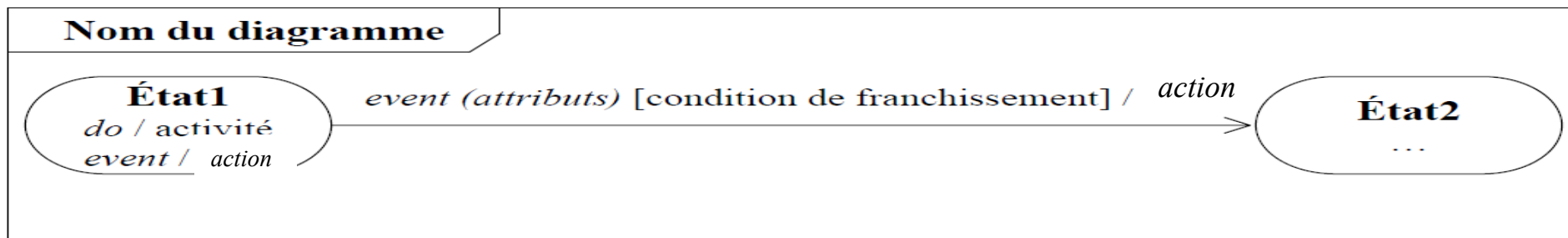


Etats d'un produit disponible en stock



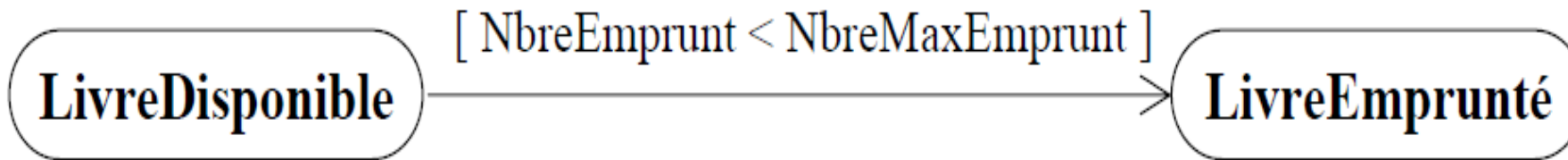
DÉFINITION : TRANSITION

- C'est le passage instantané d'un état vers un autre état souvent déclenché par un événement (*fait externe survenu*); ou peut être sans événement déclencheur (passage implicite) lorsque l'activité d'un état se termine l'objet passe à l'état suivant.
- Une transition est **orientée**;
- **Une transition** peut être **réflexive** => conduire au même état



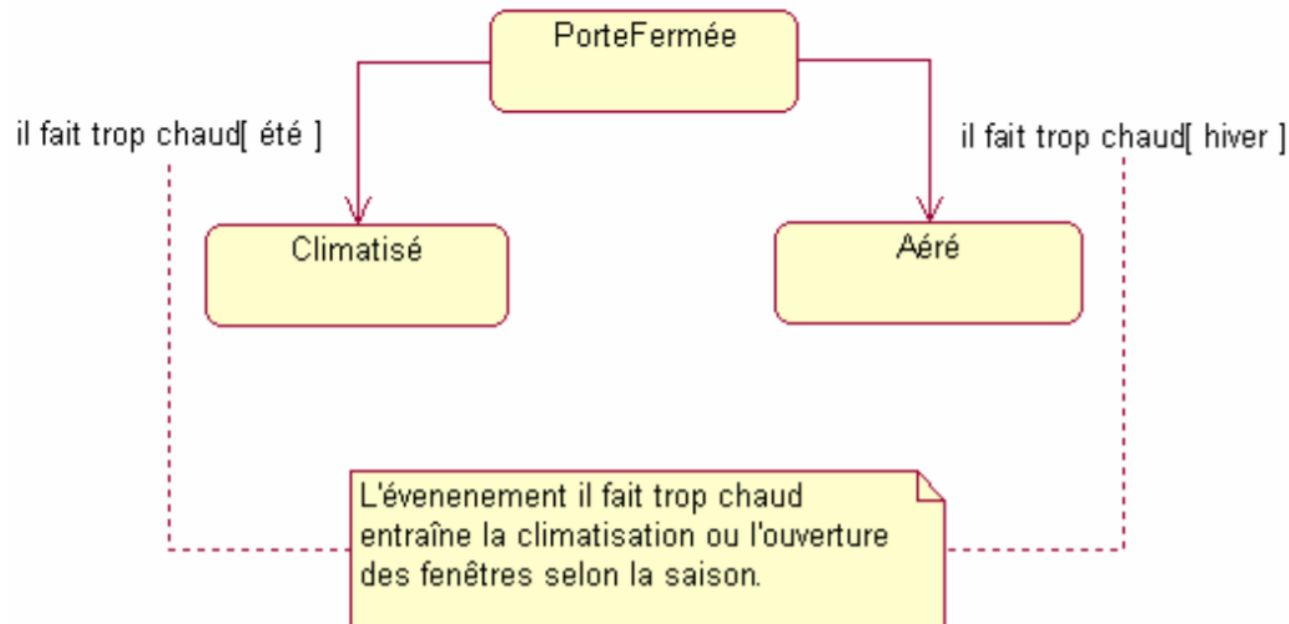
DÉFINITION : TRANSITION

- Une transition conduit toujours à l'appel d'une opération de la classe de l'objet
 - Elle peut avoir des attributs et des paramètres représentés entre parenthèses
- **Garde:** expression booléenne conditionne le déclenchement de la transition
- **Action:** correspond à une opération (définie dans le diagramme de classe), représentée

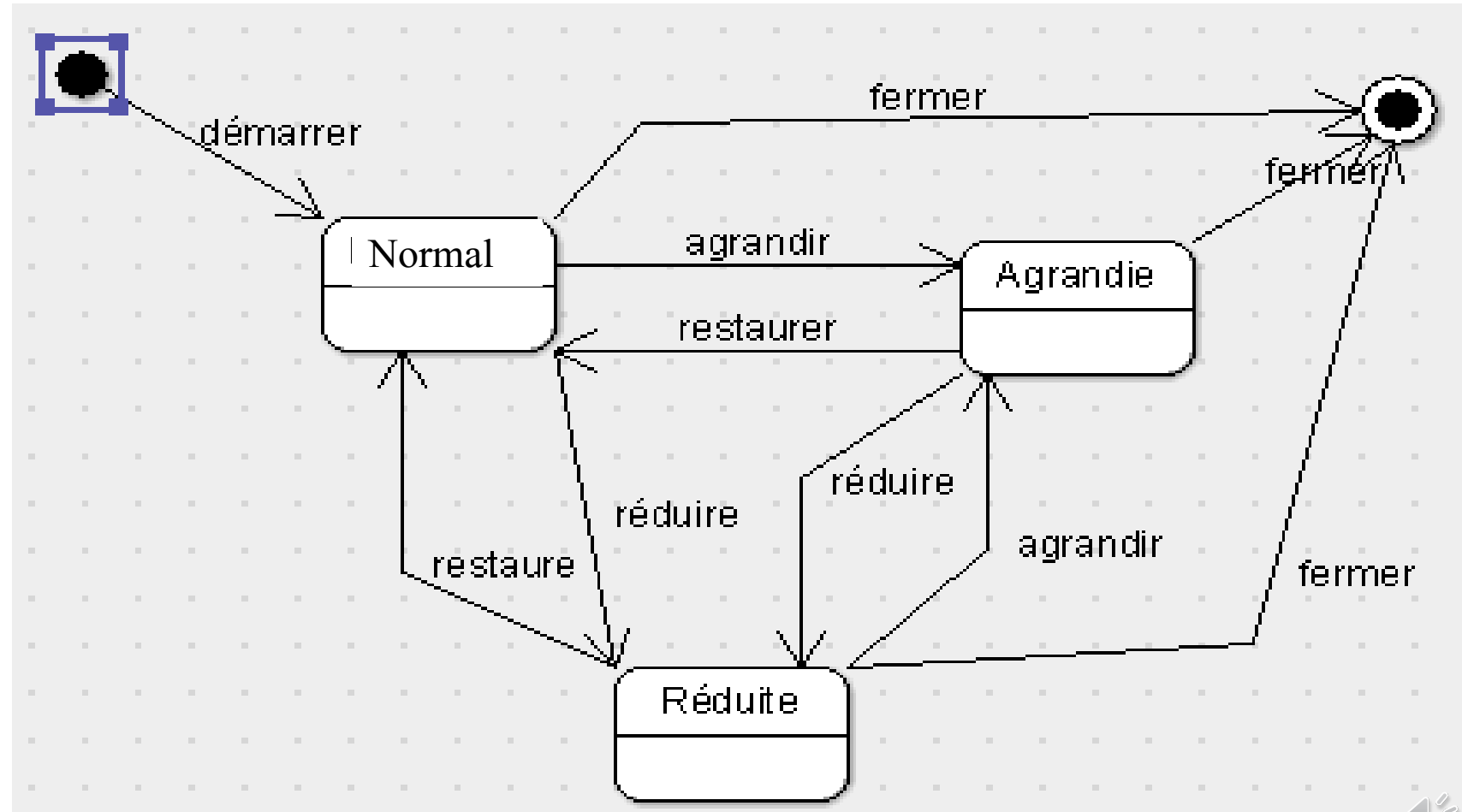
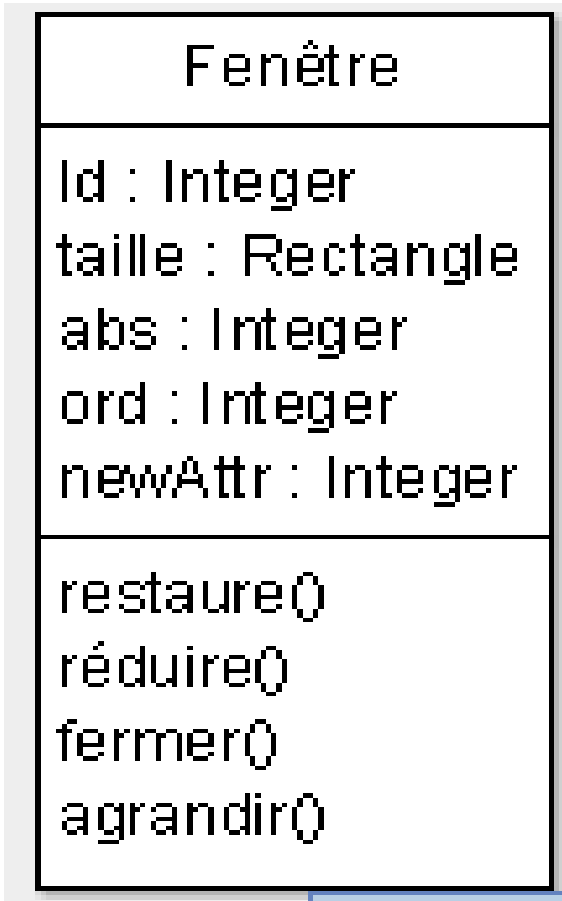


GARDE

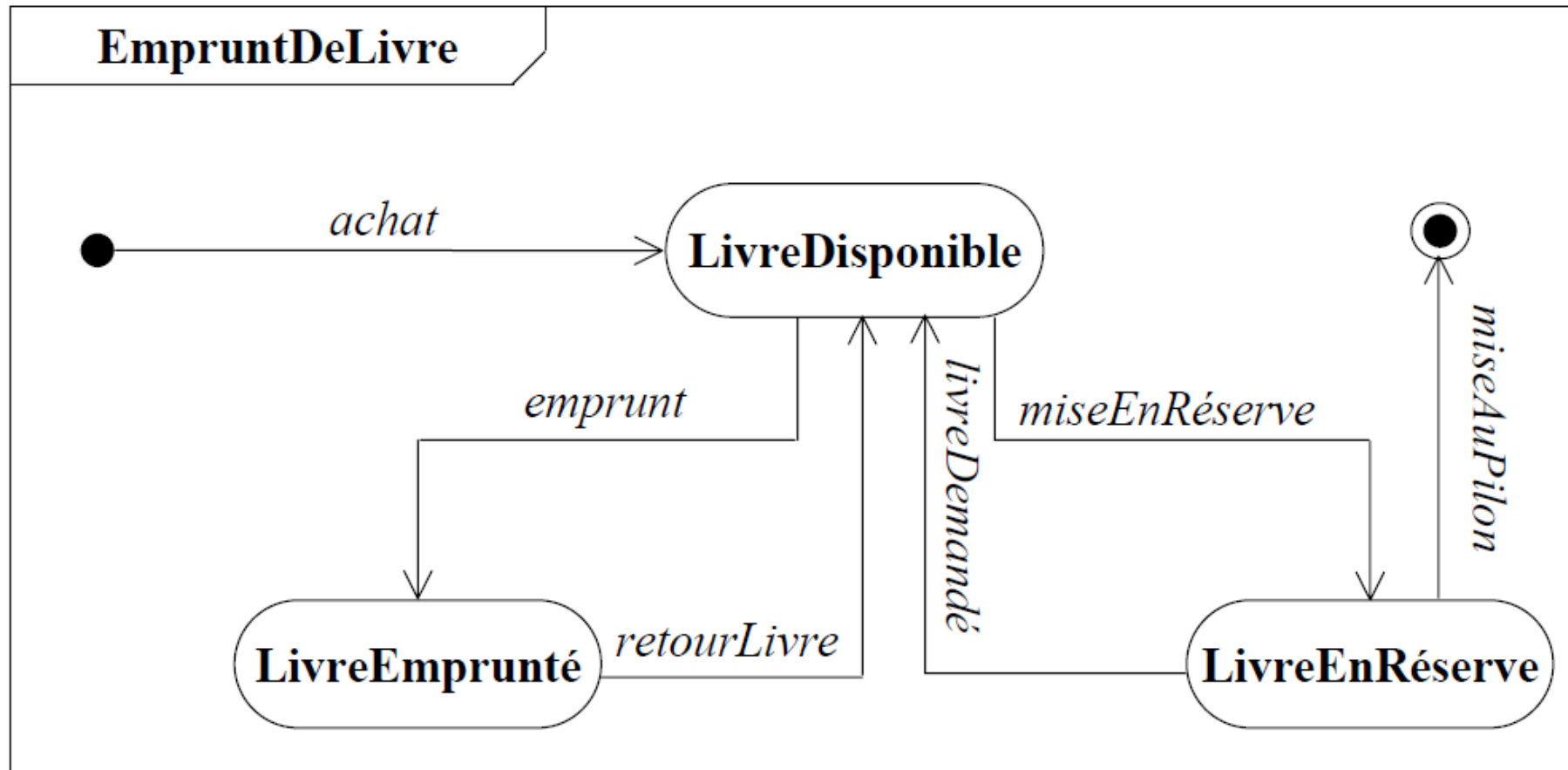
- Une condition booléenne dont dépend le déclenchement d'une transition à la réception d'un événement,
- évaluée uniquement quand l'événement est déclenché,
- peut contenir des attributs de l'objet ou des paramètres de l'événement associé



EXEMPLE 1: DET DE L'OBJET FENETRE

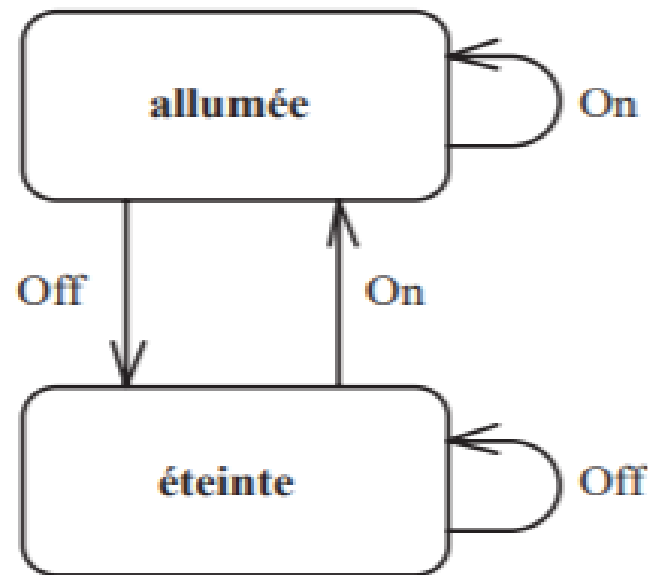


EXEMPLE 2: DET DE L'OBJET **LIVRE**

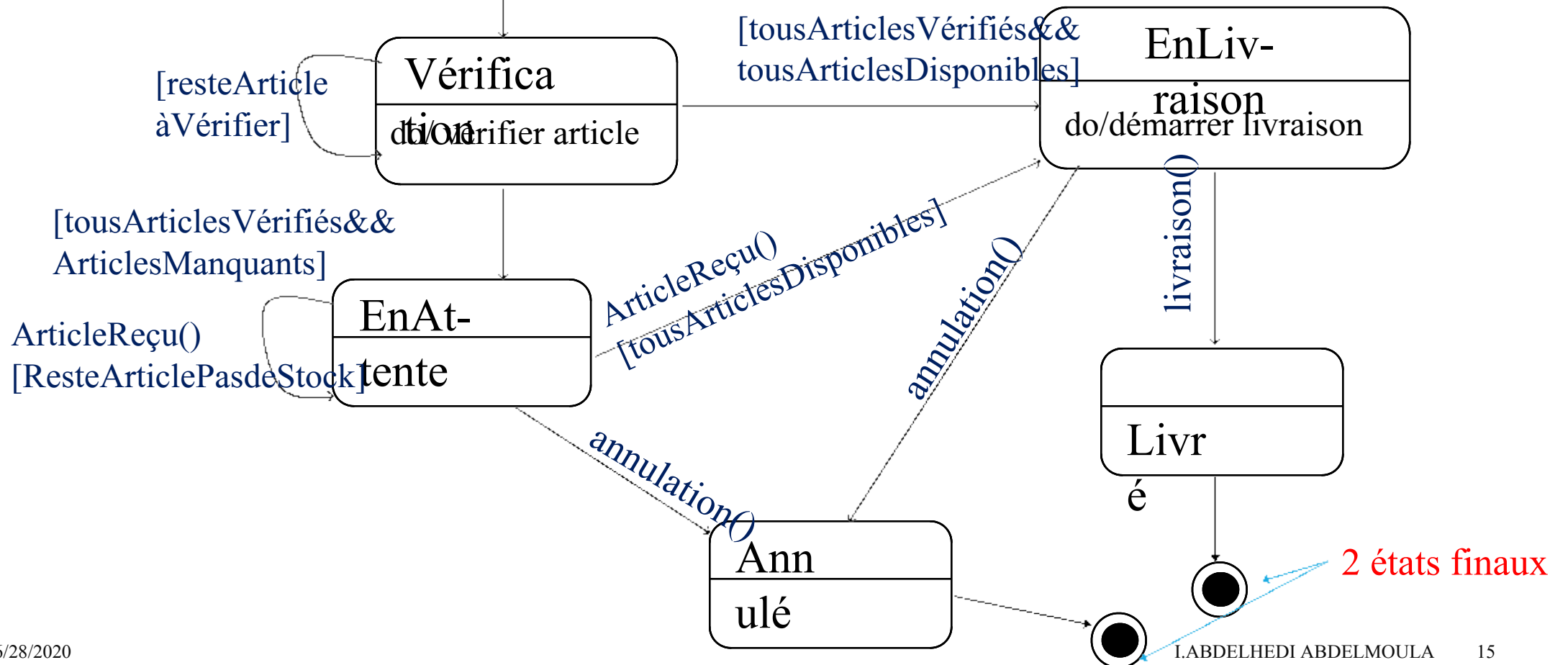


EXEMPLE DET D'UNE LAMPE

Considérez une lampe munie de deux boutons-poussoirs : une pression sur On allume la lampe et une pression sur Off l'éteint. Une pression sur On ne produit pas d'effet si la lampe est déjà allumée ; la réaction d'une instance de Lampe à cet événement dépend de son état interne.



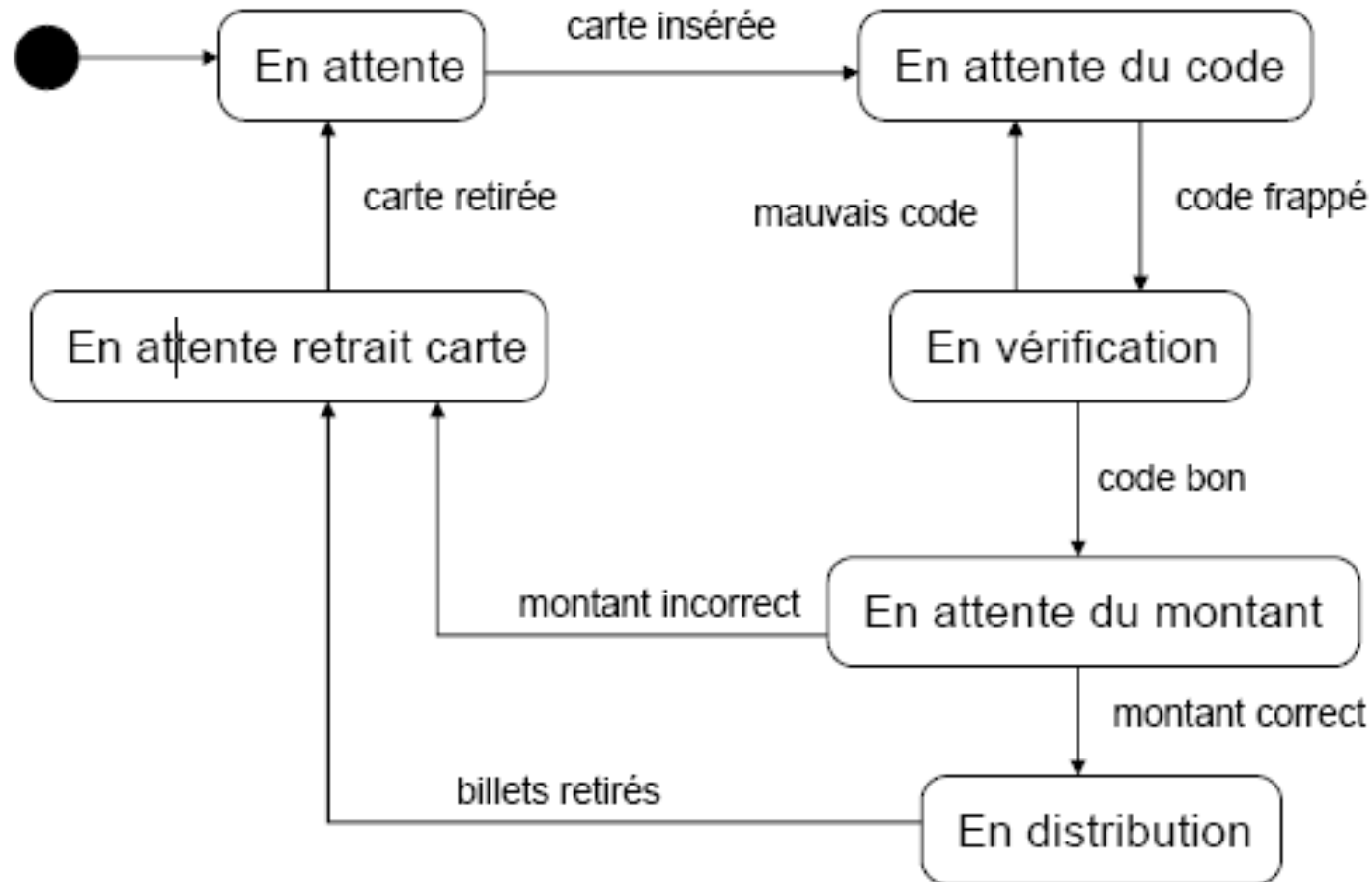
EXEMPLE 3 : DET DE L'OBJET **COMMANDE**



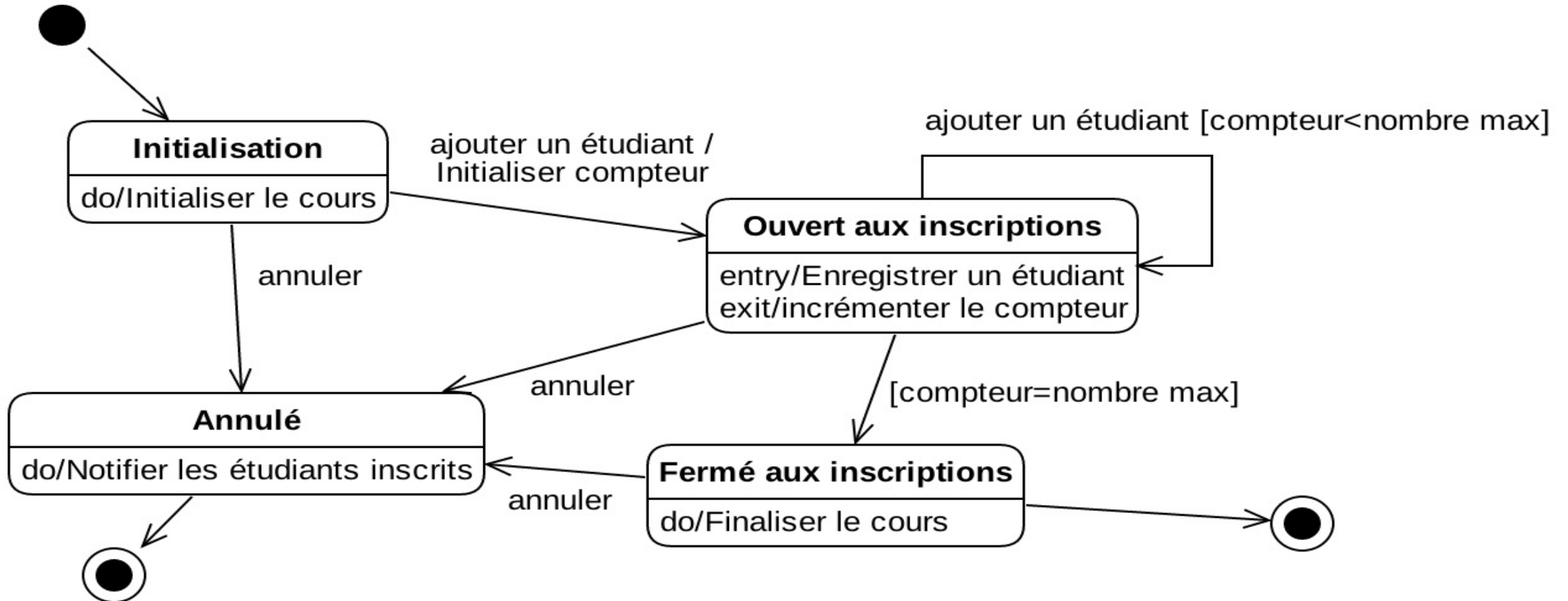
EXEMPLE 4: DET DE L'OBJET

DISTRIBUTEUR GAB

Cyclique – pas
d'état final

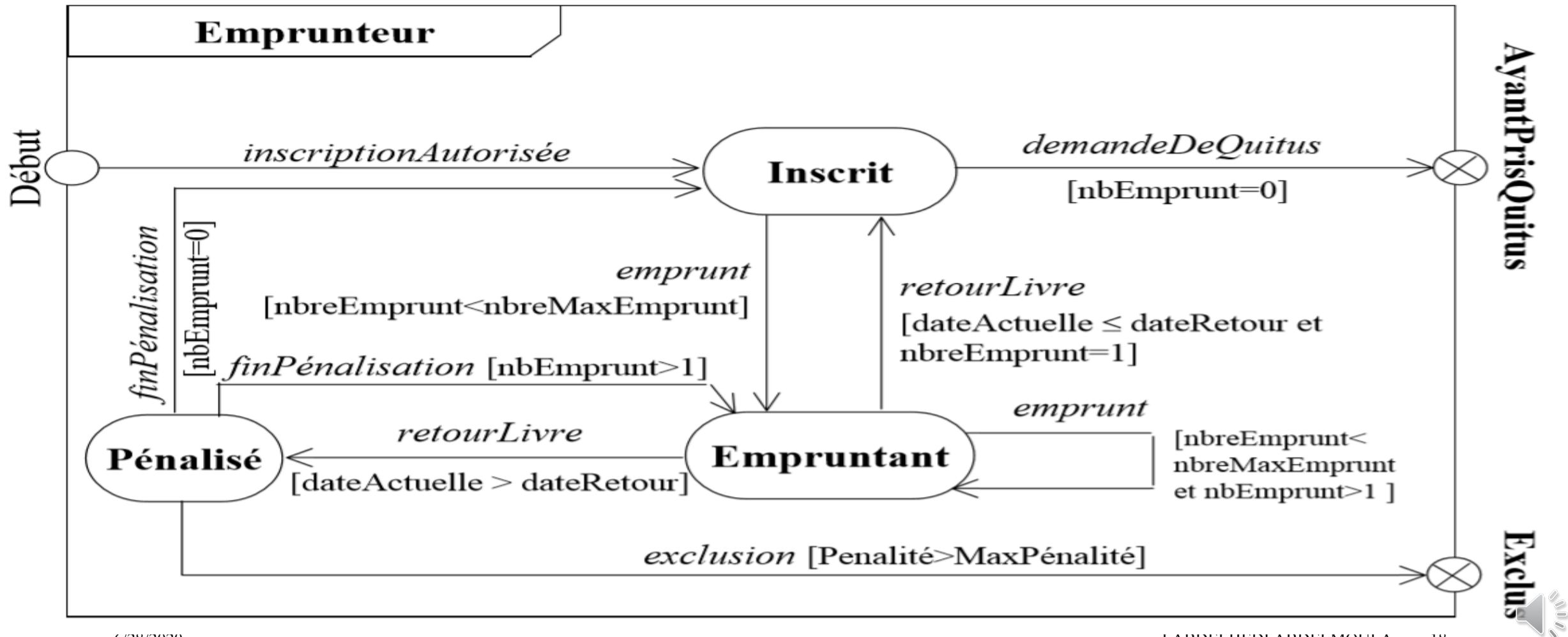


EXEMPLE 5: DET DE L'OBJET COURS

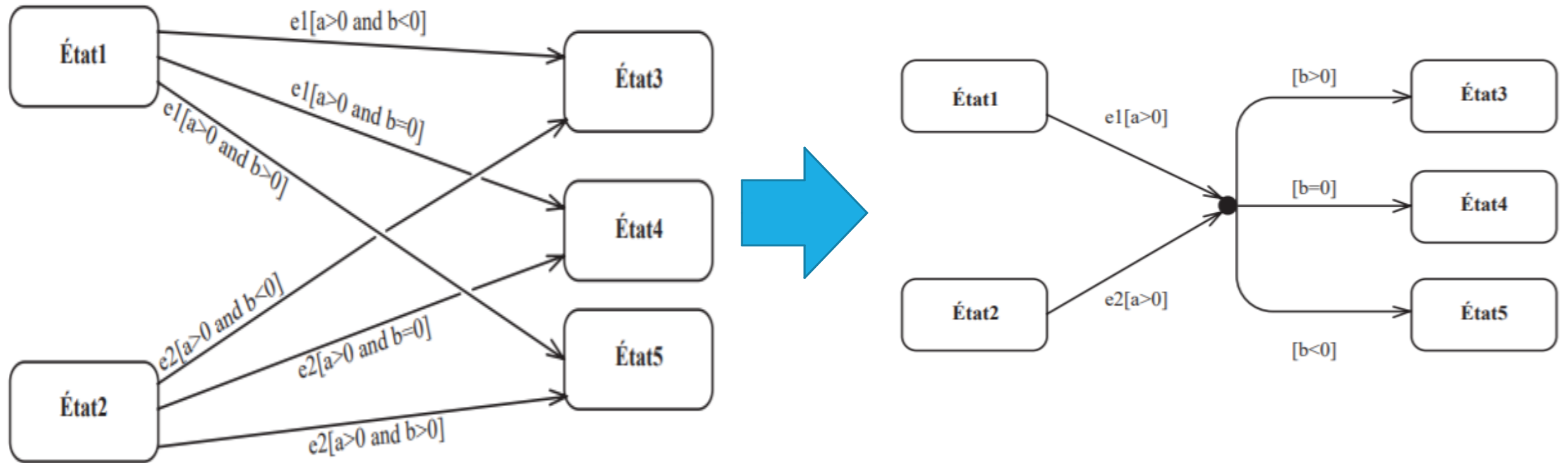


EXEMPLE 6: DET DE

UN MODIET EN ADDITIONNEL ID



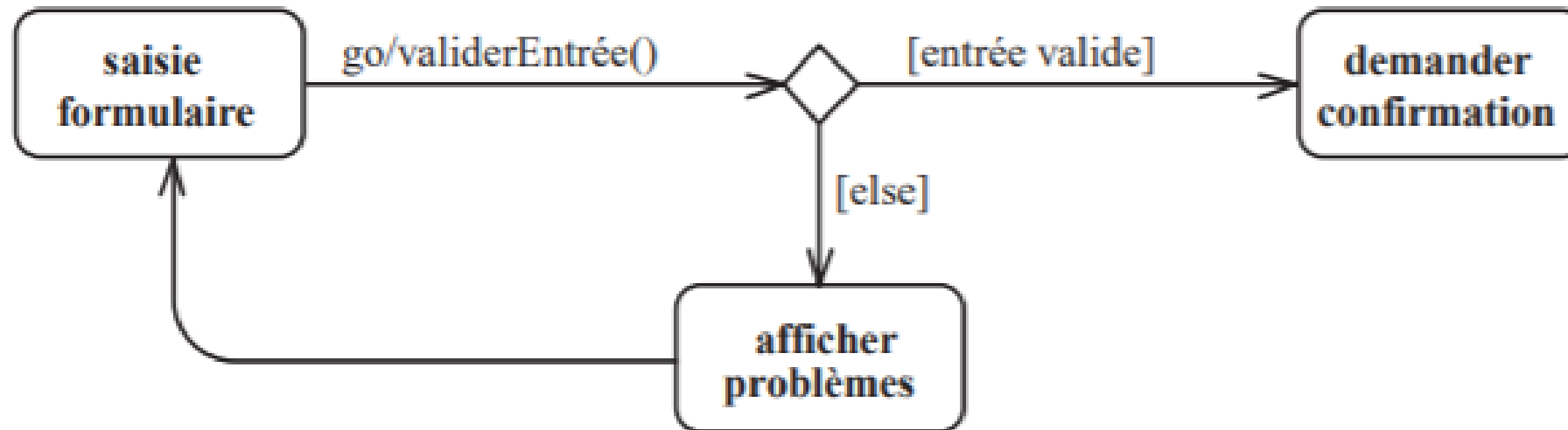
POINT DE JONCTION = PARTAGER DES SEGMENTS DE TRANSITION



ALORS)

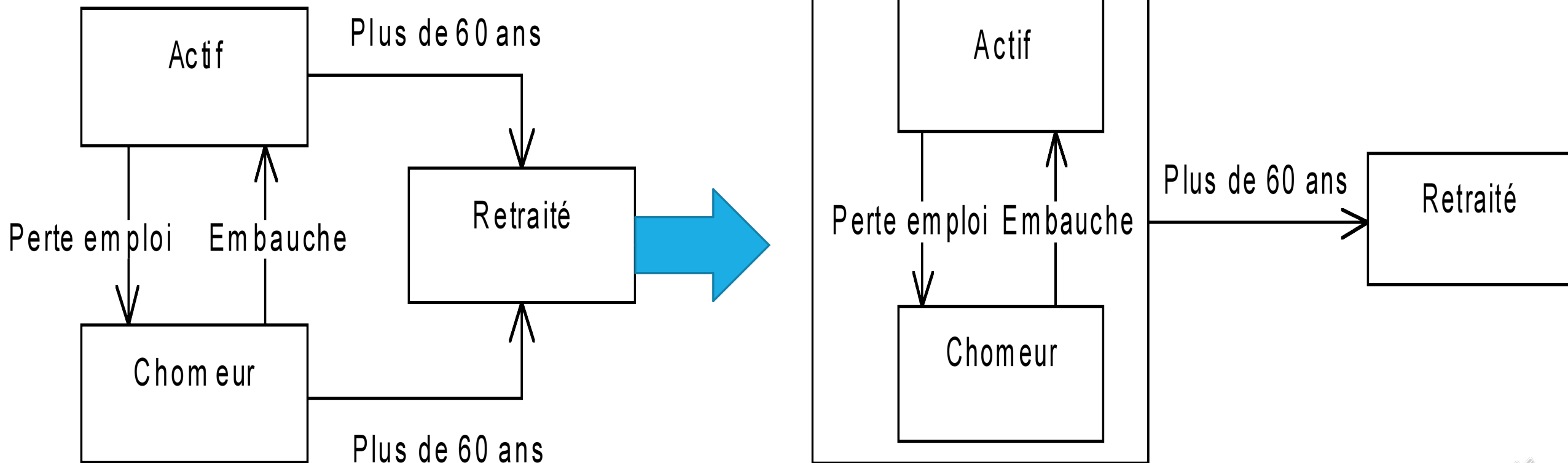
= CHOISIR UNE AUTRE

TRANSITION ALTERNATIVE

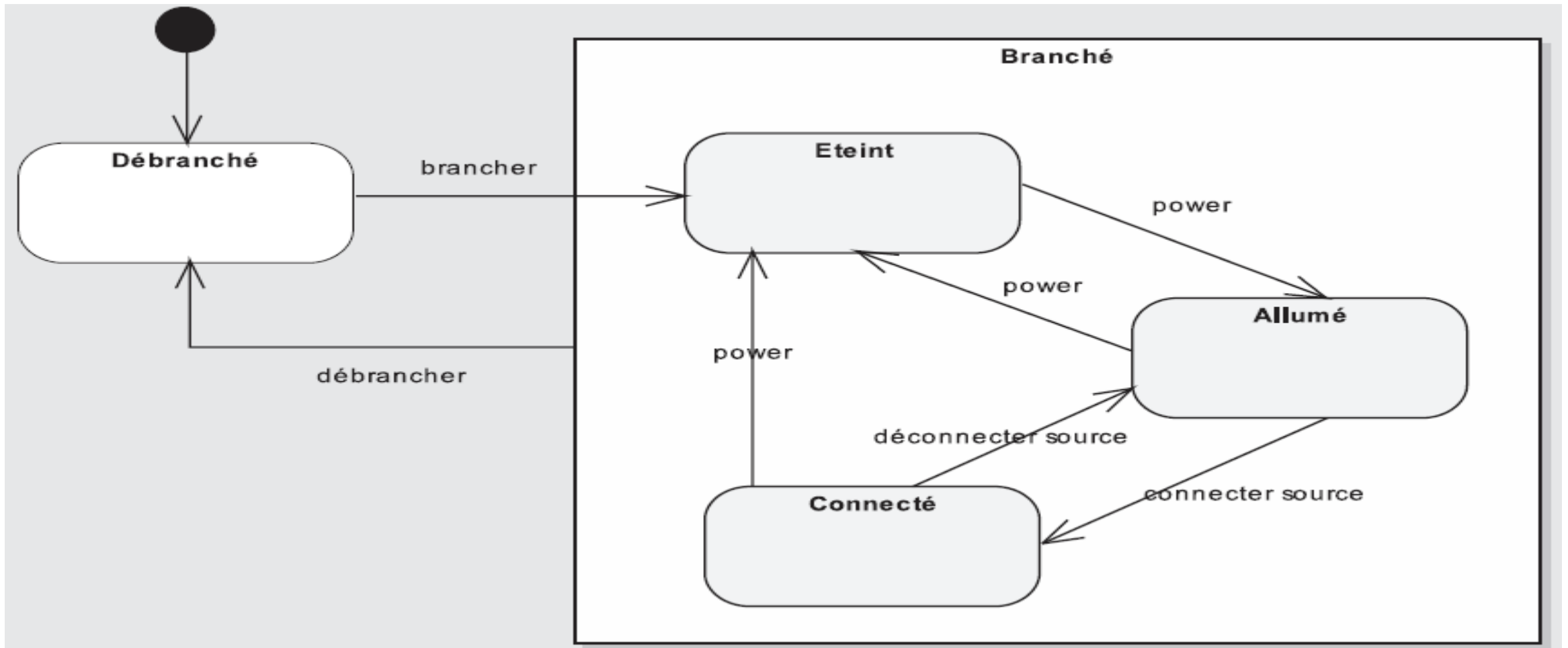


ÉTAT COMPOSITE OU IMBRIQUÉ

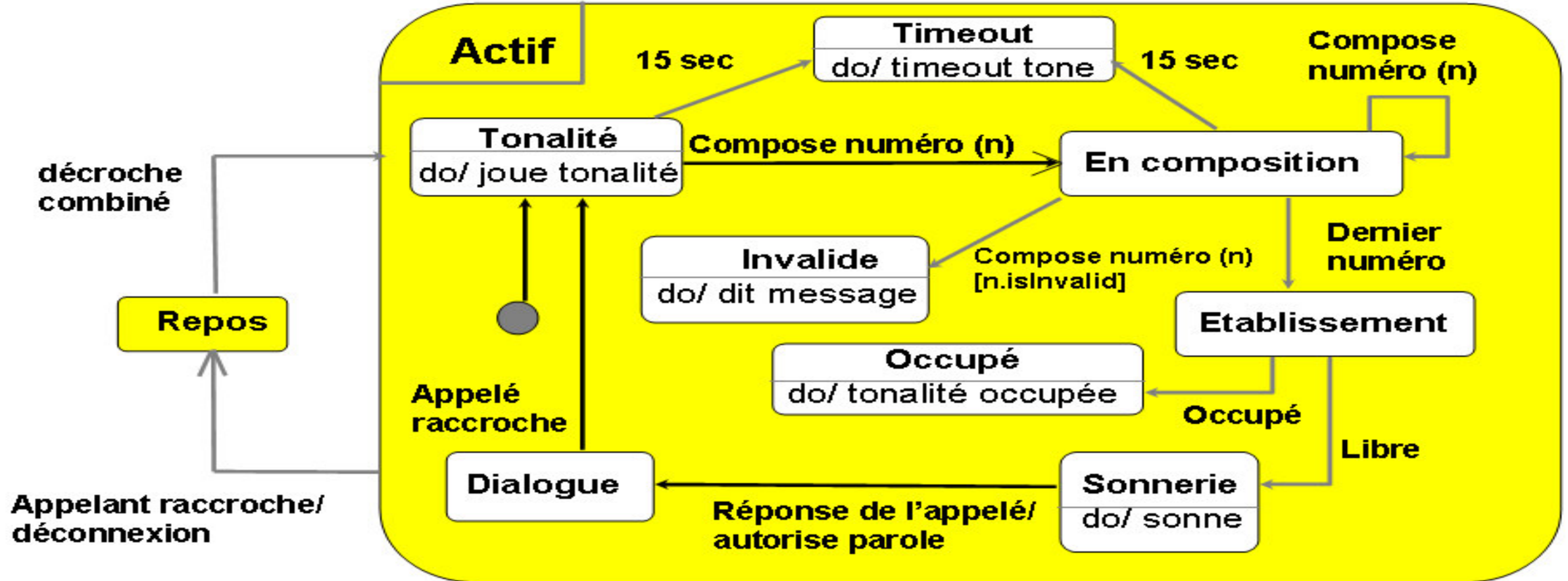
- Un état qui est décomposé en deux ou plusieurs sous états.
- La décomposition est réursive



EXEMPLE 1 : DET DE L'OBJET VIDEOPROJECTEUR



Exemple 2 : états du système téléphone (diagramme avec décomposition d'état)



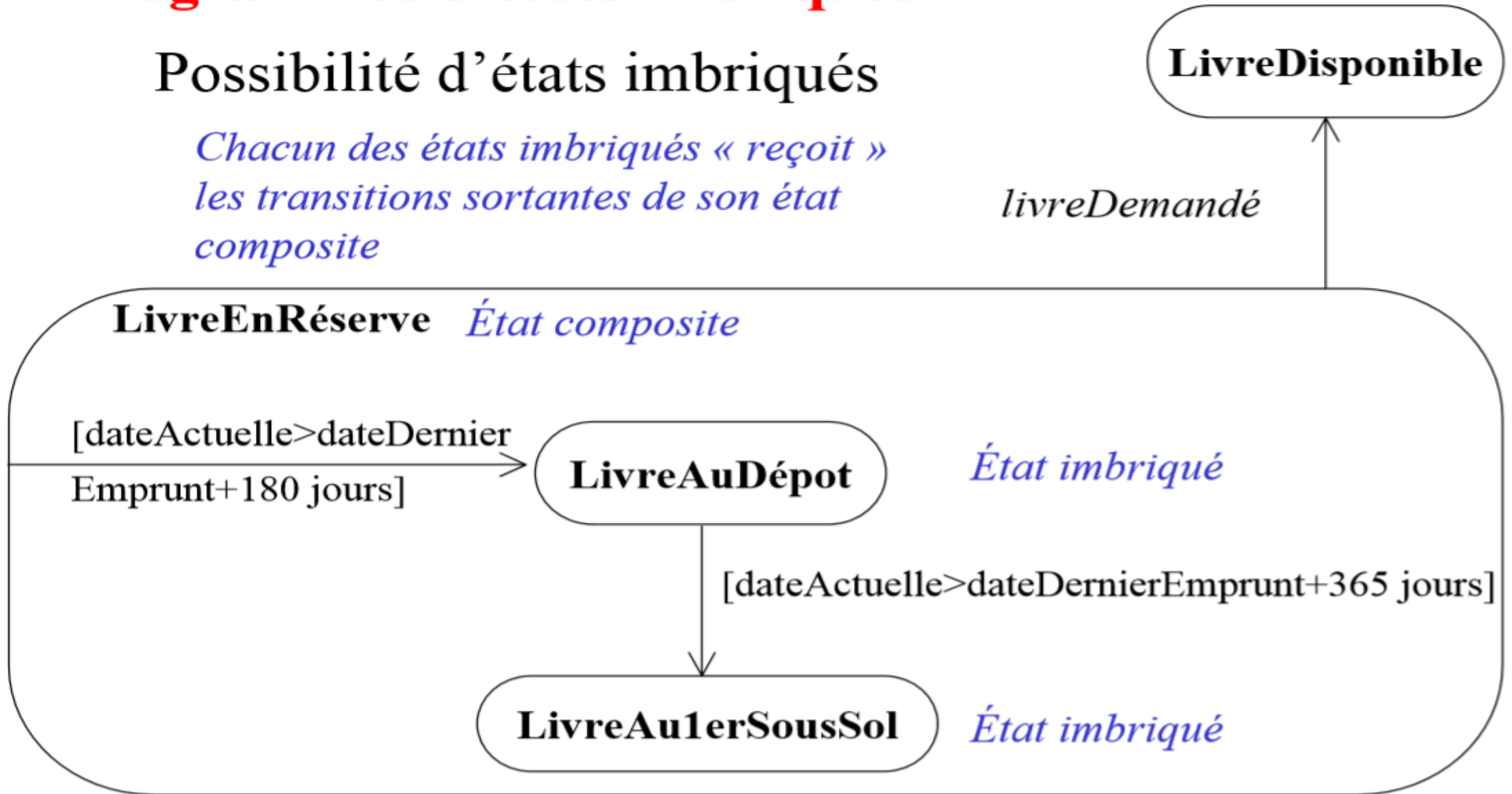
Les événements des transitions correspondent souvent dans ce cas à des actions sur le système



Diagrammes d'états imbriqués

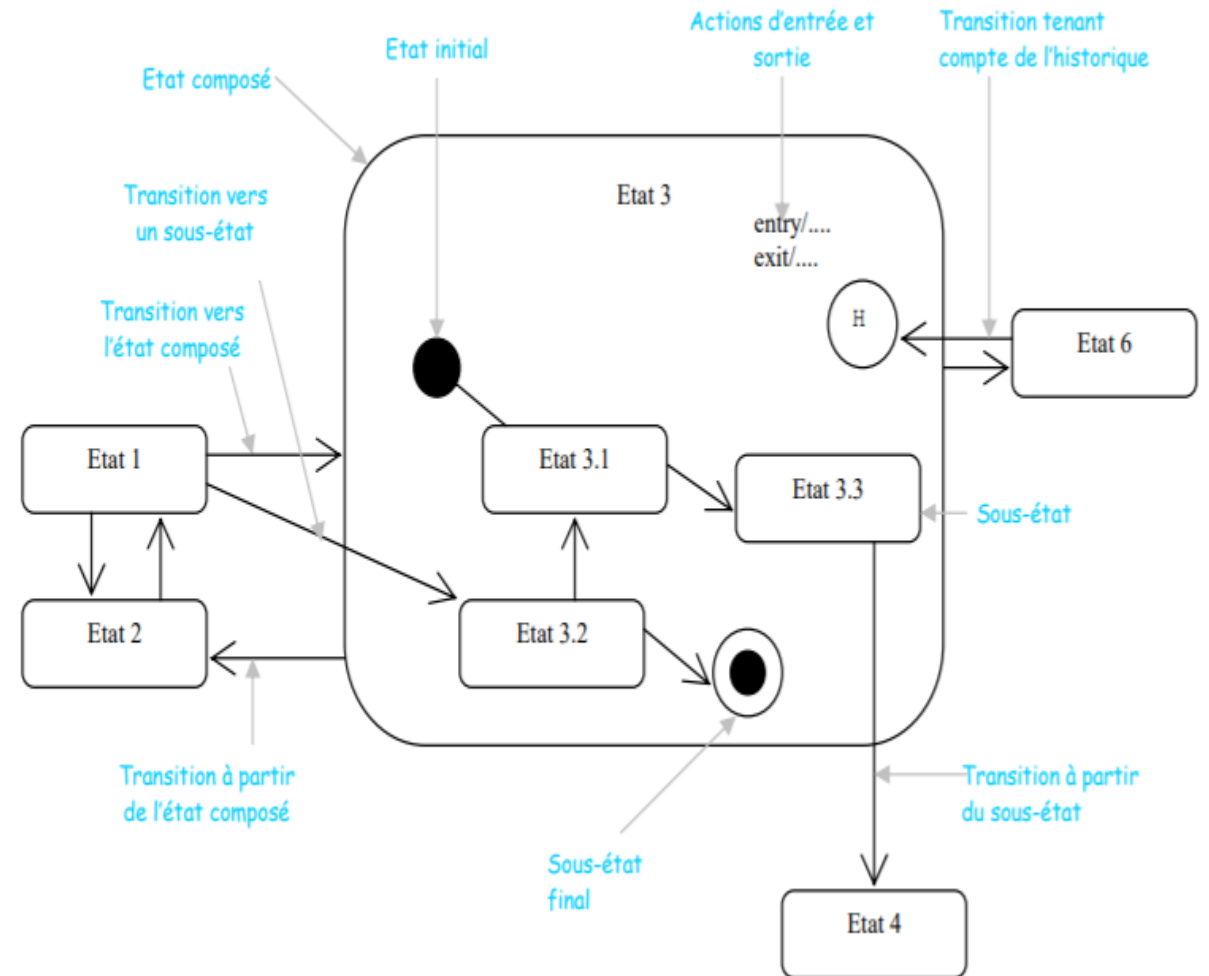
Possibilité d'états imbriqués

*Chacun des états imbriqués « reçoit »
les transitions sortantes de son état
composite*



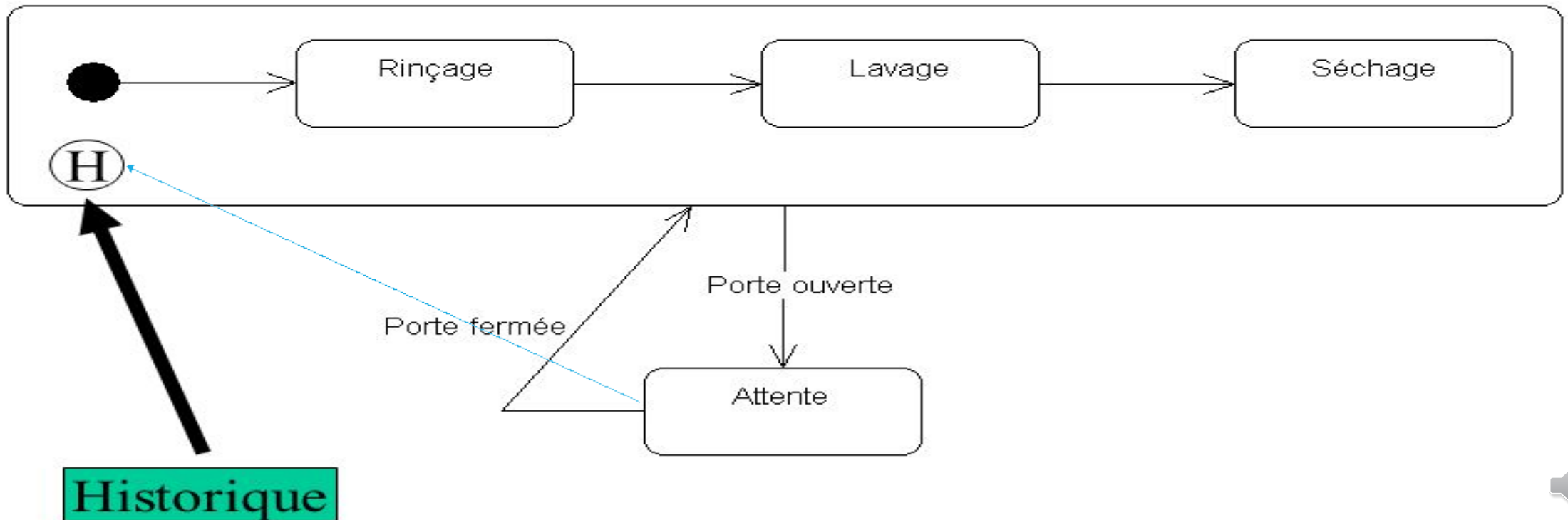
ETAT HISTORIQUE – AUTOMATE A MÉMOIRE

- Lorsqu'on rentre dans un état composé, l'action de l'automate repart de son état initial.
- Dans certaines situations, on a besoin que l'automate emboîté se souvienne du dernier sous-état où il était avant de le quitter pour la dernière fois ceci sera modélisé à l'aide d'état historique.
- Un automate emboîté avec historique (marqué par un cercle contenant la lettre H) repart à partir du dernier sous-état où il était si la transition entrante dans l'état composé aura pour cible le symbole H.
- Dans tous les cas, quand un automate emboîté atteint son sous-état final, il perd l'historique et se comporte comme si rien ne s'était passé.



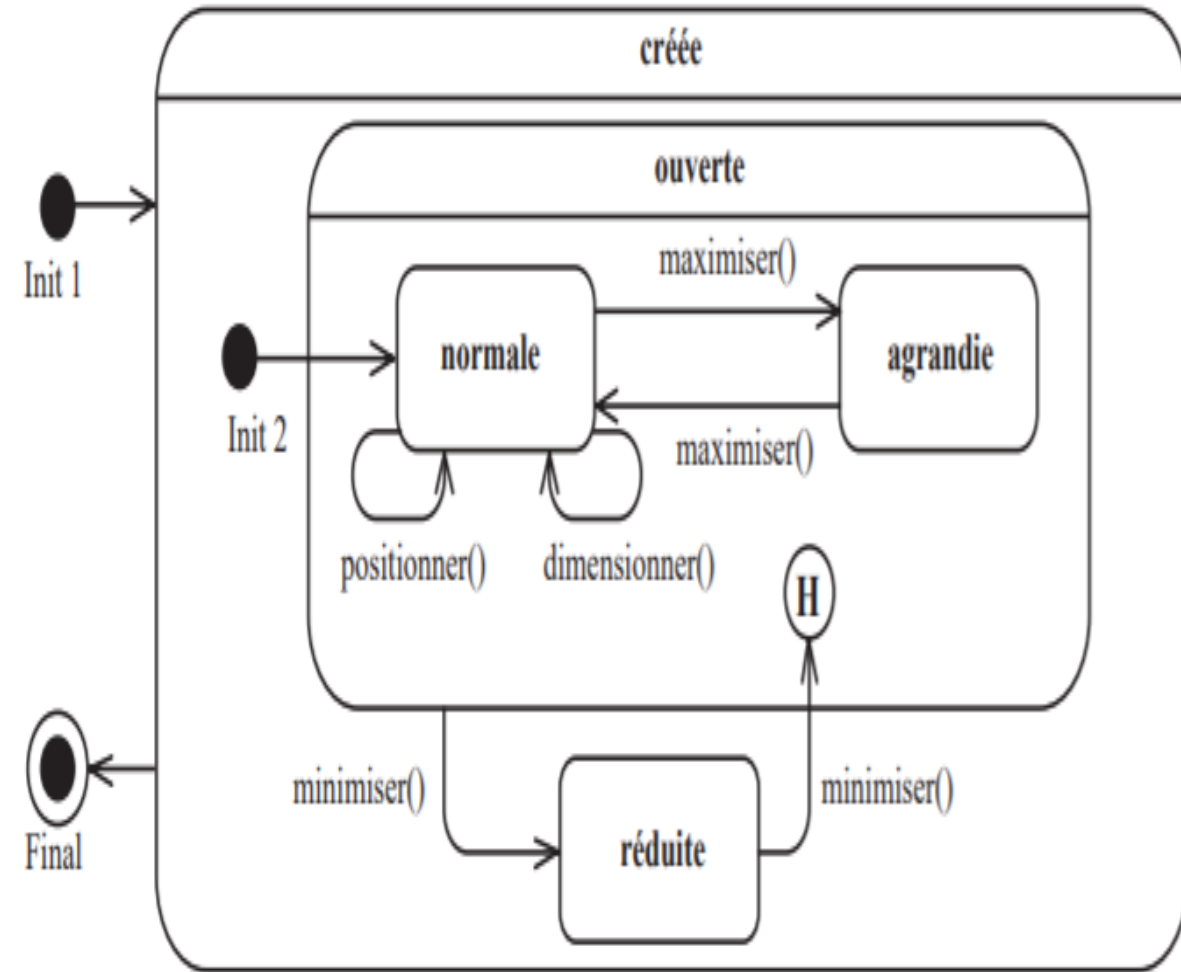
ETAT HISTORIQUE – AUTOMATE A MÉMOIRE

- Par défaut, un automate n'a **pas de mémoire**
- La notation **H** offre un **mécanisme pour mémoriser** le dernier sous-état qui l'englobe
- **Exemple** : cycle de lavage d'un lave vaisselle



EXEMPLE 1: DET DE L'OBJET FENETRE

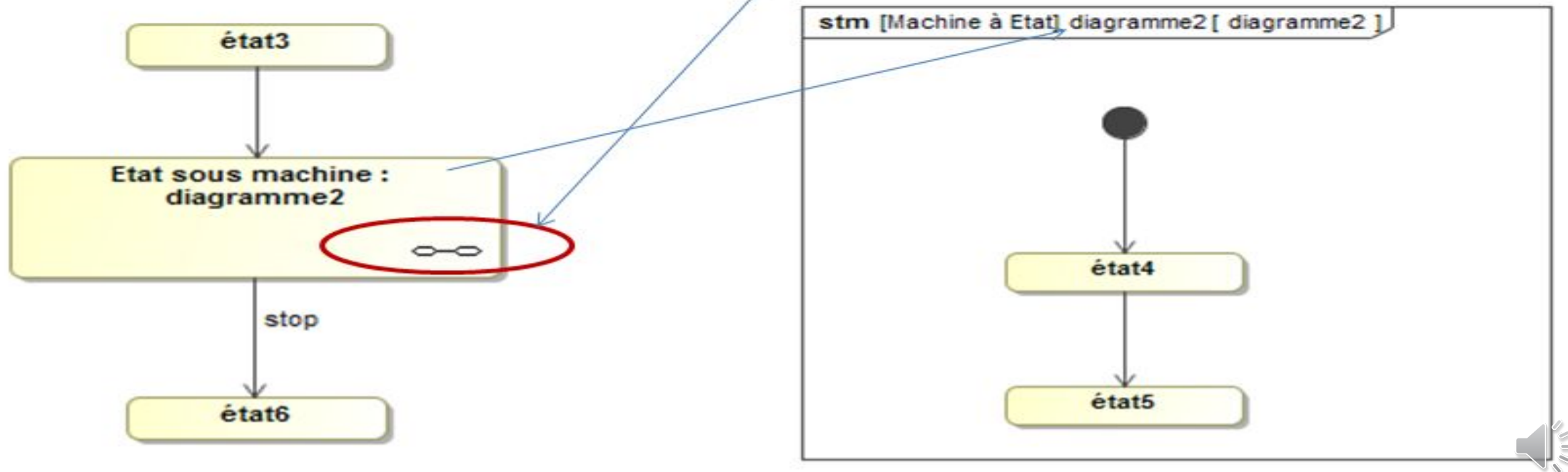
- Après sa création, l'objet FENETRE répond à stimulus de 3 boutons placés dans l'angle.
- Elle Peut avoir trois états : réduite, normale, agrandie.
- Lorsqu'elle est à l'état normal, elle peut être déplacée et redimensionnée.
- Lorsqu'elle est agrandie, elle occupe toute la surface. Il ne peut ni déplacée et redimensionnée. Lorsqu'on quitte l'état réduite, on retrouve son dernier sous-état lorsqu'elle était à l'état ouverte.
- Init 1 et Init 2 désignent les points d'entrée. Une nouvelle instance sera initialisé à l'état créée, dans le sous-état ouverte et dans le sous-état normale.
- Le pseudo-état historique permet de retrouver la fenêtre dans son état précédent (normale ou agrandie) quand on quitte l'état réduite.
- L'état final correspond à la destruction de l'instance (la fin de sa vie)



ETAT SOUS MACHINE

Etat sous machine : état instance d'un automate fini (diagramme d'états).

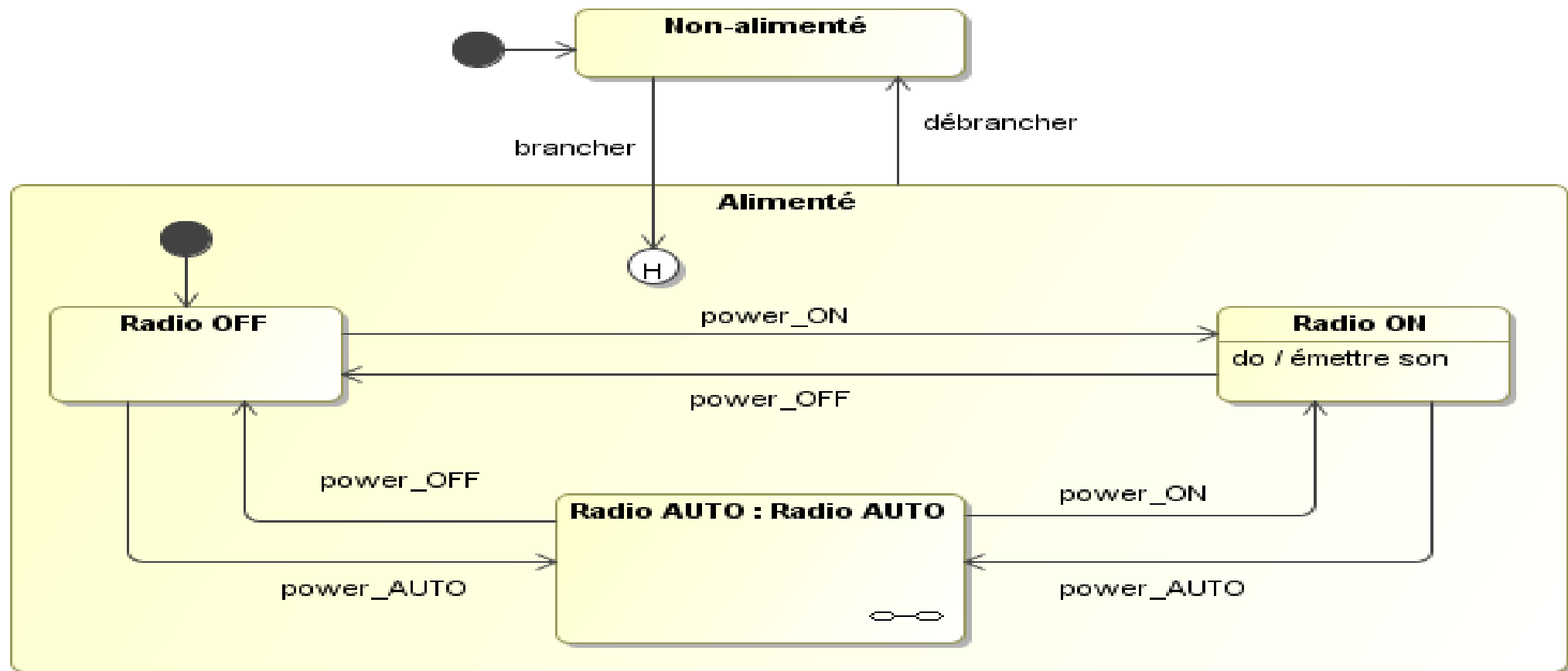
- Le diagramme d'états d'un état sous machine est dessiné dans un autre diagramme d'état : une icône représentant 2 états + une transition permet de le signaler.



EXEMPLE 1: DET DE

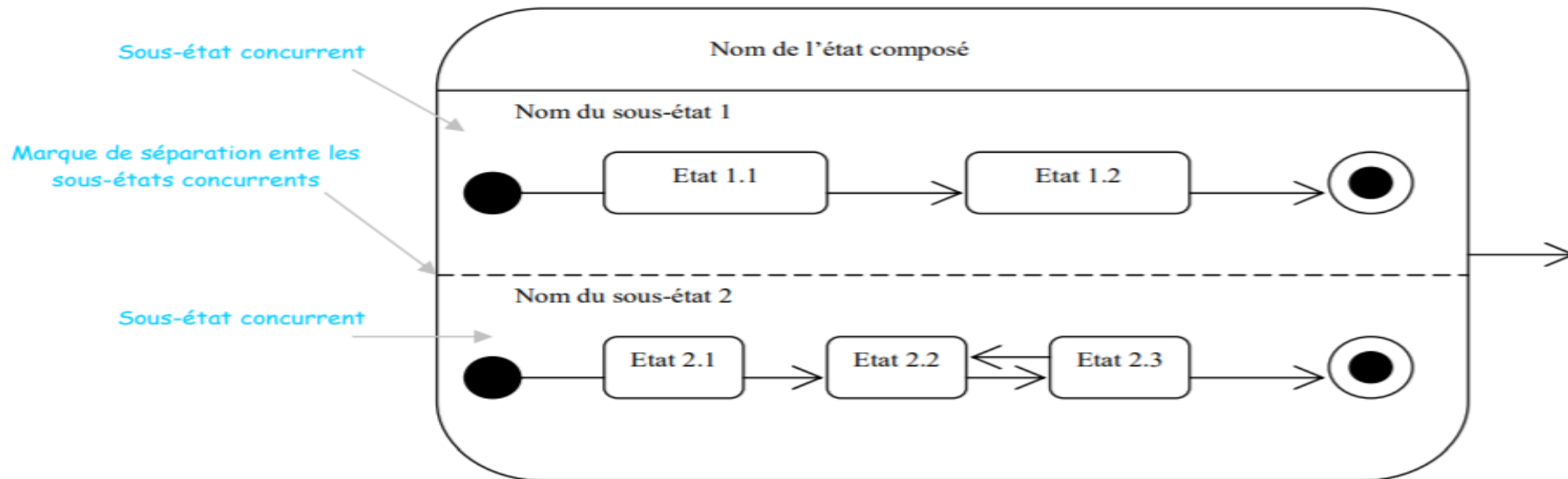
L'OBJET RADIO

state machine Radio [ Radio]

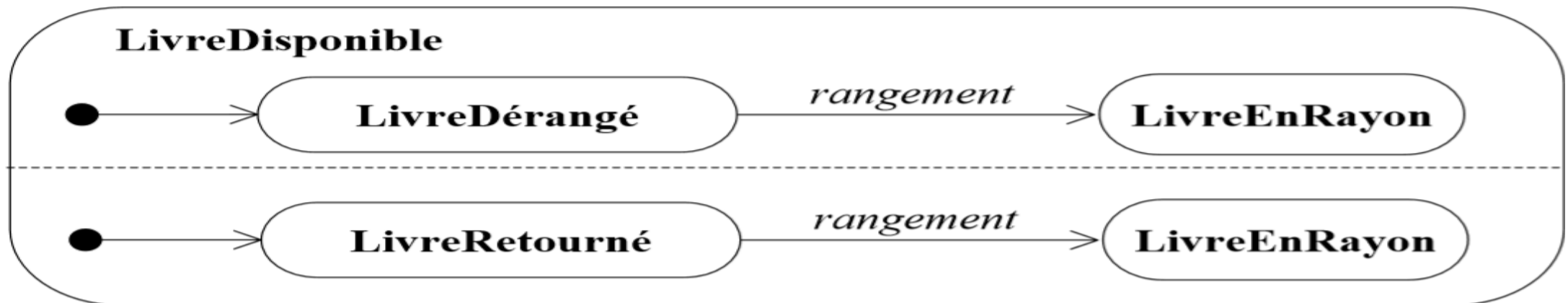


ÉTATS CONCURRENTS

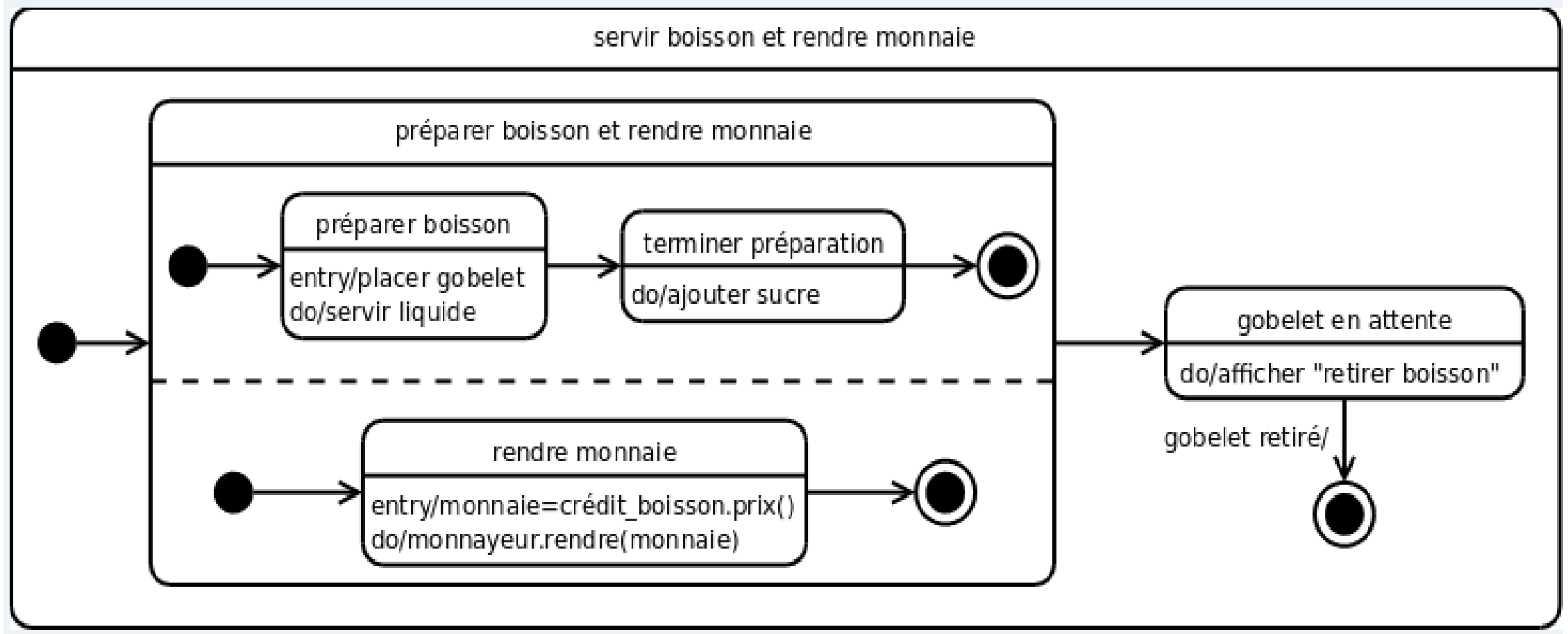
- Un état concurrent est un **état composite** qui représente l'**exécution de plusieurs automates** s'exécutant indépendamment
- On utilise un séparateur en pointillés ou l'objet peut être simultanément dans plusieurs états concurrents



EXEMPLE : ÉTATS CONCURRENTS

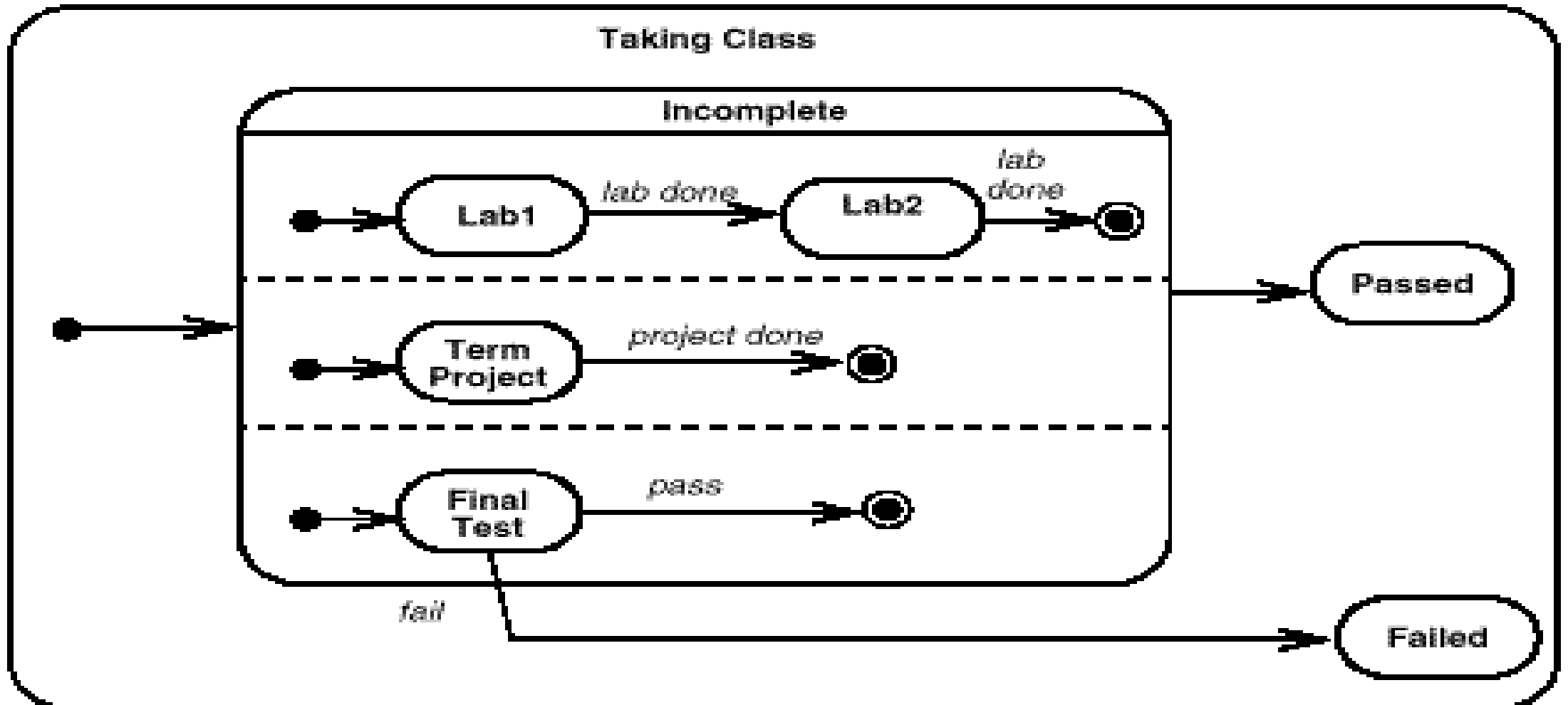


EXEMPLE 1: D'ÉTATS CONCURRENTS



EXAMPLE 2: DES SOUS-ÉTATS

CONCURRENCE



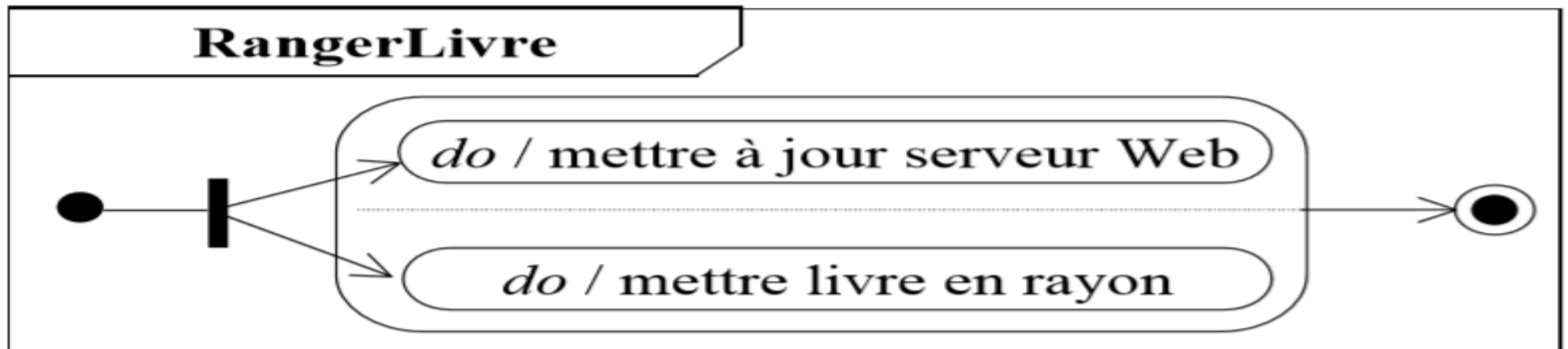
TRANSITIONS

CONCURRENTES

Synchronisation du contrôle des activités concurrentes

Possibilité pour un même objet d'exécuter des activités concurrentes

- ⇒ Pas de synchronisation des activités
- ⇒ Mais division du contrôle des activités
- ⇒ Et synchronisation du contrôle



TRANSITIONS

CONCURRENTES

- **Fork** = création de deux états concurrents
- **Join** = supprimer la concurrence (barre de synchronisation)

Pour pouvoir franchir la **barre de synchronisation**, toutes les tâches concurrentes doivent être achevées

