

- 3.4 -

Modélisation comportementale

Exemples dans le cours

Une partie inspirés de

- Software Engineering, 9th ed. - Ian Sommerville
- Object-Oriented Software Engineering: Practical Software Development using UML and Java - Timothy C. Lethbridge, Robert Laganière
- UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. - Martin Fowler
- UML 2 - de l'apprentissage à la pratique - Laurent Audibert, Ellipses, 2009

Perspectives

- *externe*
 - ▶ modéliser le contexte (l'environnement)
- *interaction*
 - ▶ modéliser les interactions avec l'environnement et entre les composants
- *structurelle*
 - ▶ modéliser l'architecture et les données
- *comportementale*
 - ▶ modéliser la dynamique

Modélisation comportementale

- les aspects dynamique du système lors de son exécution
- comment répondre aux stimuli
 - ▶ données
 - ▶ événements

Modélisation dirigée par les données

- adaptée pour les systèmes dont le comportement est dirigé par les données reçues en input
- modéliser les actions impliquées dans le traitement de données en entrée et la génération de données en sortie
 - ▢▢▢▢ ➡ utilisée dans l'analyse des besoins

Diagramme de séquence

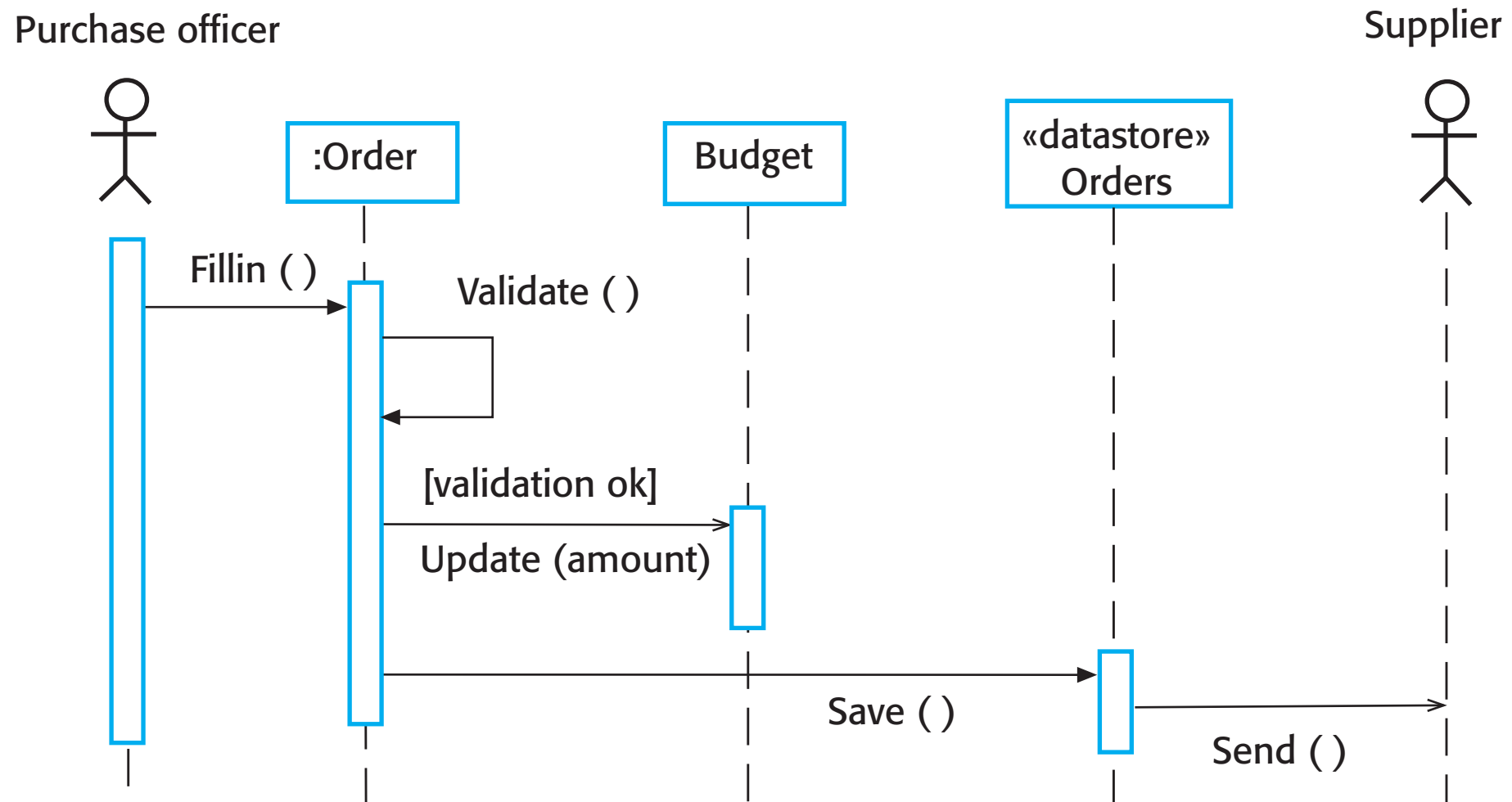
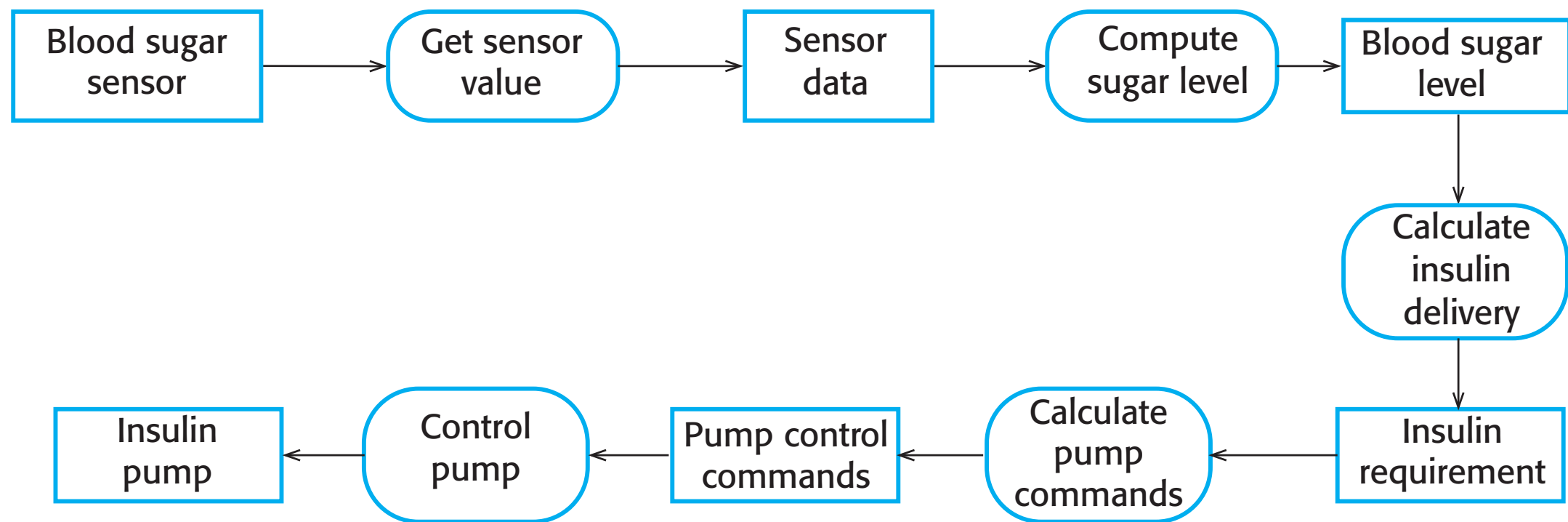


Diagramme d'activités



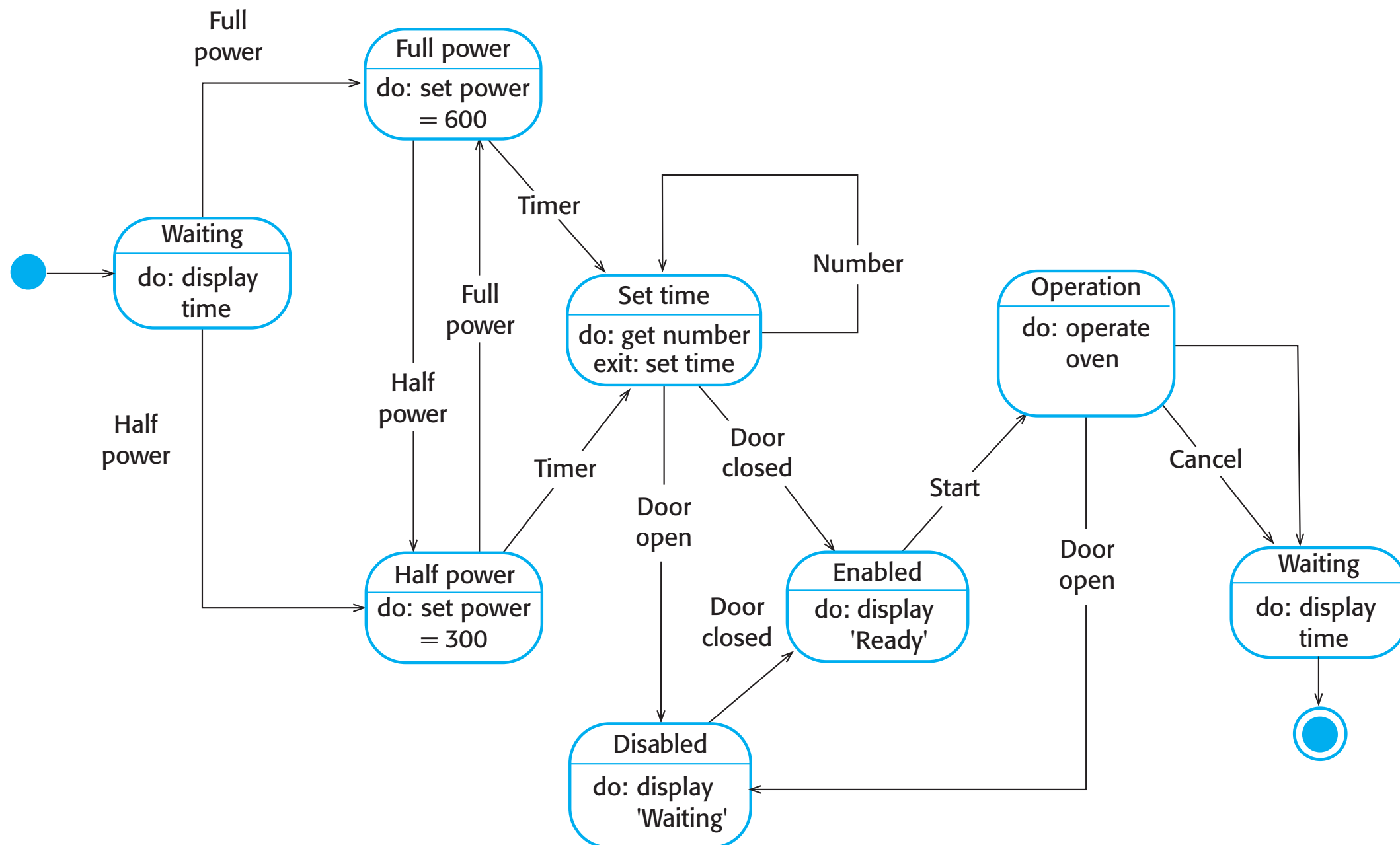
Modélisation dirigée par les événements

- adaptée pour les systèmes dont le comportement est dirigé par les événements internes et externes
- exemple : systèmes temps-réel
- adaptée pour les systèmes avec un nombre fini d'états (les événements déclenchent les transitions)

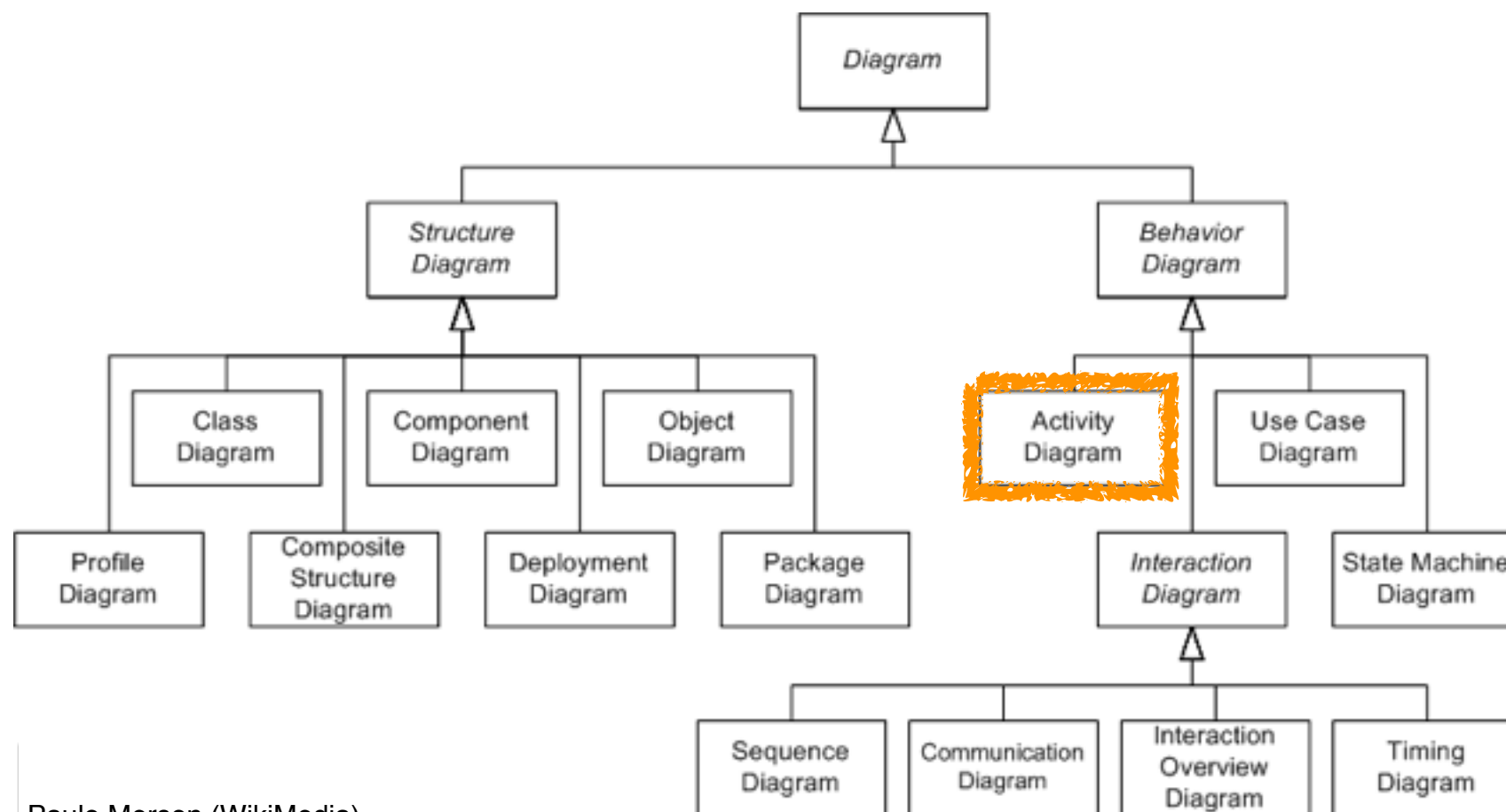
Diagramme d'états

- utilisés pour modéliser le comportement du système en réponse à des événements
- graphes avec
 - ▶ noeuds = états
 - ▶ arcs = événements

Diagramme d'états



Diagrammes d'activités



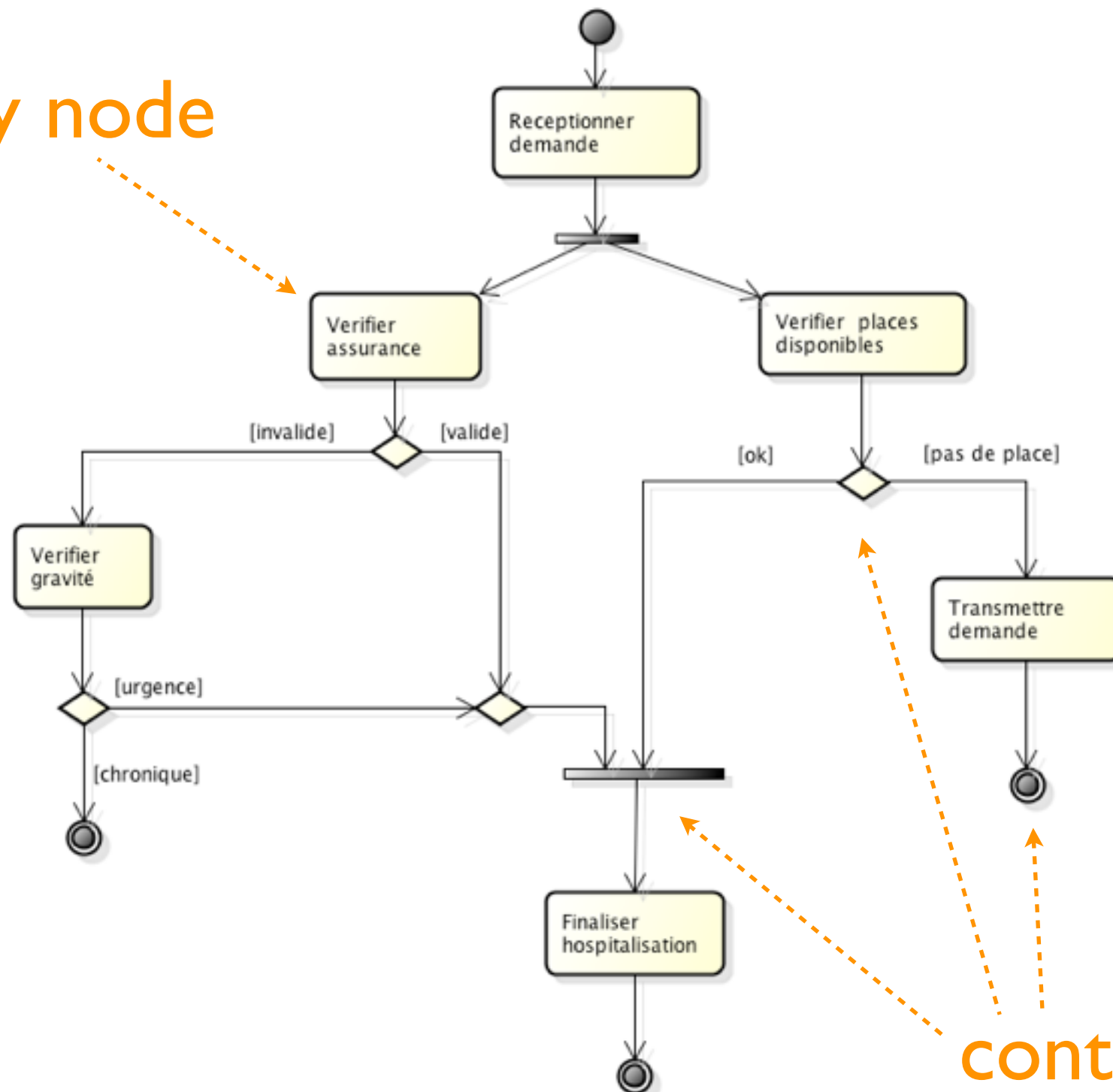
Paulo Merson (WikiMedia)

Diagramme d'activités

- utilisé pour décrire
 - ▶ la succession des étapes qu'un système doit accomplir afin de réaliser une certaine tâche
 - ▶ les use-cases et la façon dont ils sont liés
 - ▶ des activités concurrentes
- *est le plus souvent associé à plusieurs classes*

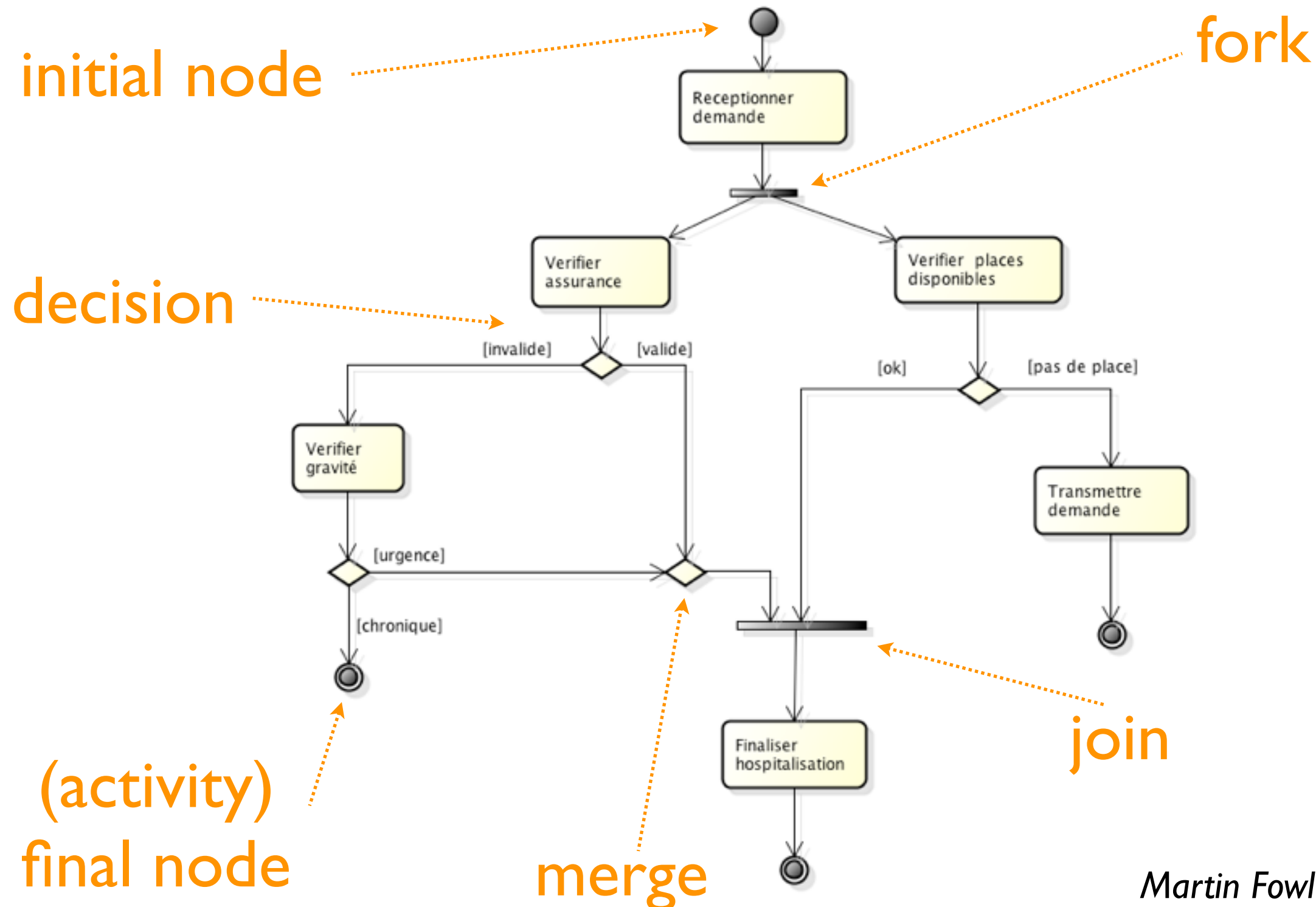
Exemple

activity node



control node

Exemple



Concurrence

Un **fork** a une transition entrante et plusieurs transitions sortantes

- L'exécution se sépare alors en différents fils d'exécution

Un **join** a plusieurs transitions entrantes et une sortante

- La transition sortante s'effectue lorsque toutes les transitions entrantes se sont produites

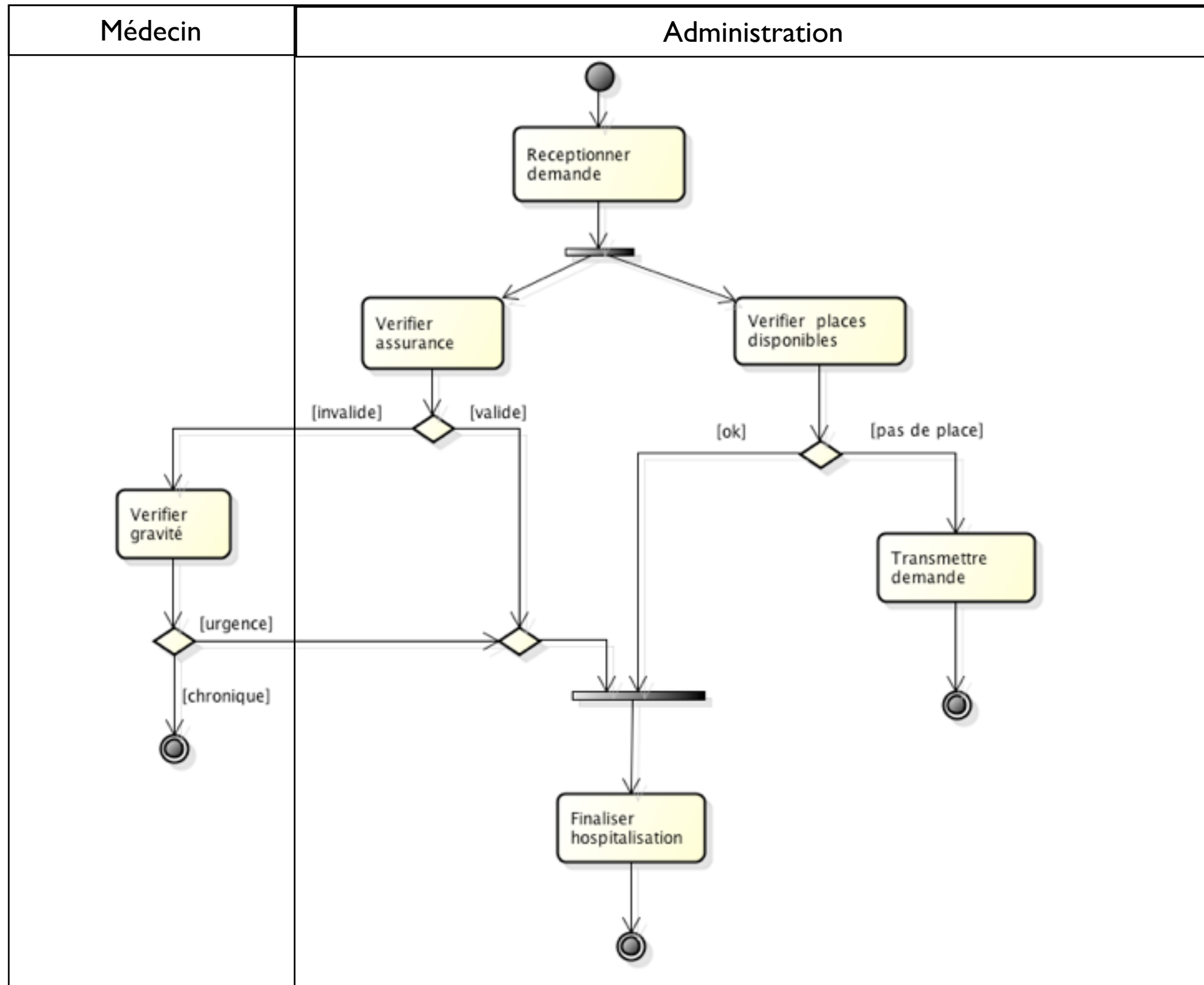
join + fork = rendez-vous

Swimlanes

- Un diagramme d'activités est souvent associé à plusieurs classes
- Le partitionnement des activités à travers les différentes classes peut être explicitement montré à l'aide de *swimlanes*

Swimlanes

qui effectue les actions



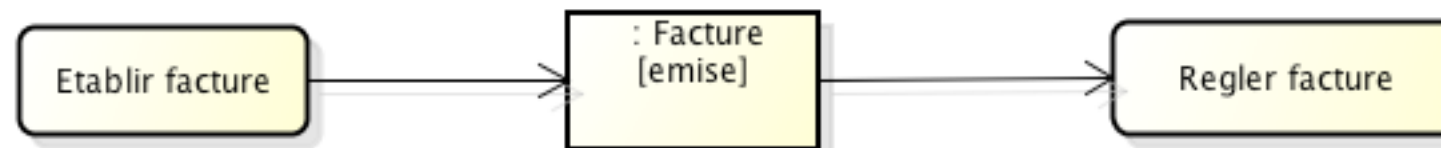
Flot de données



connectors

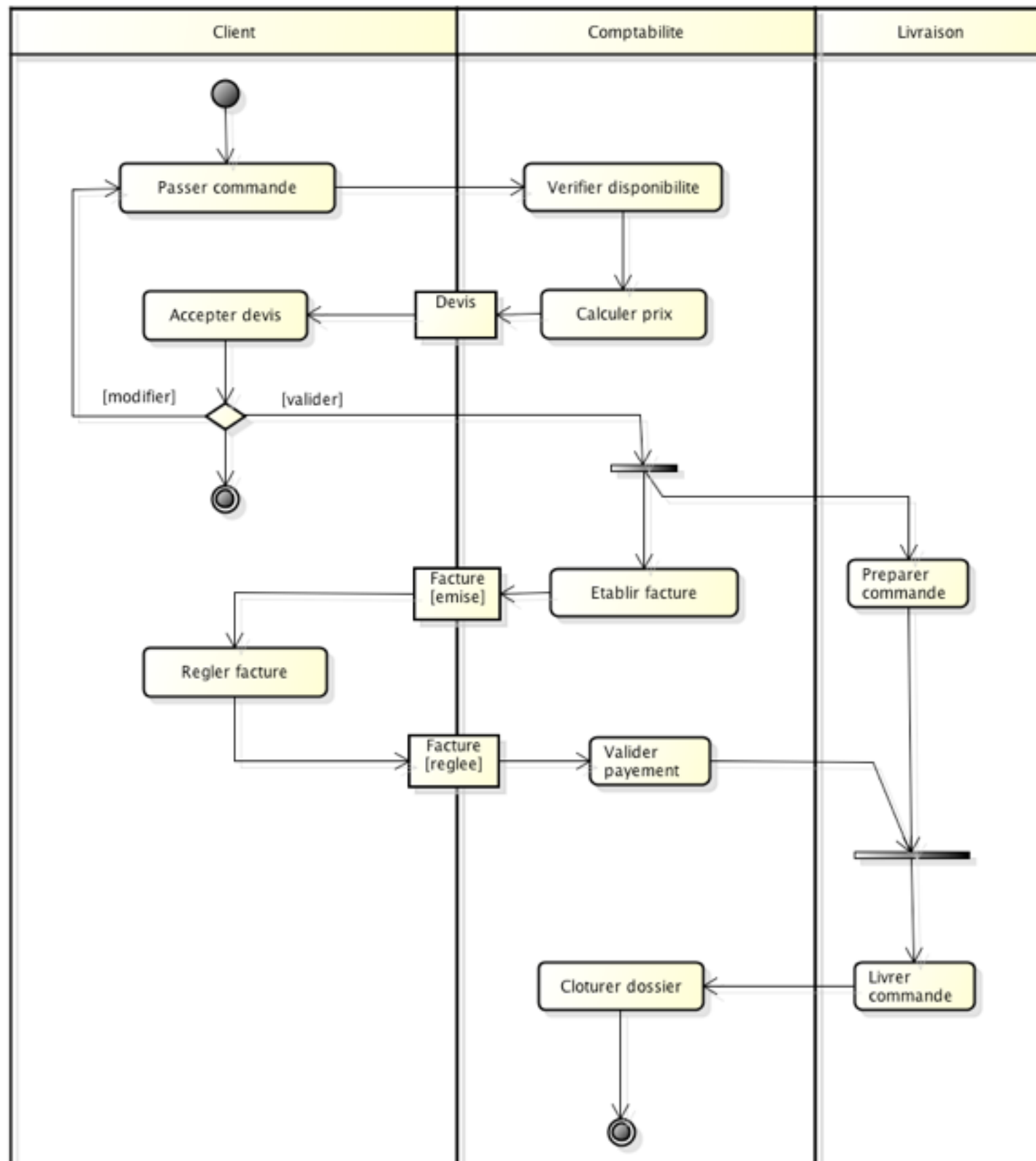


pins

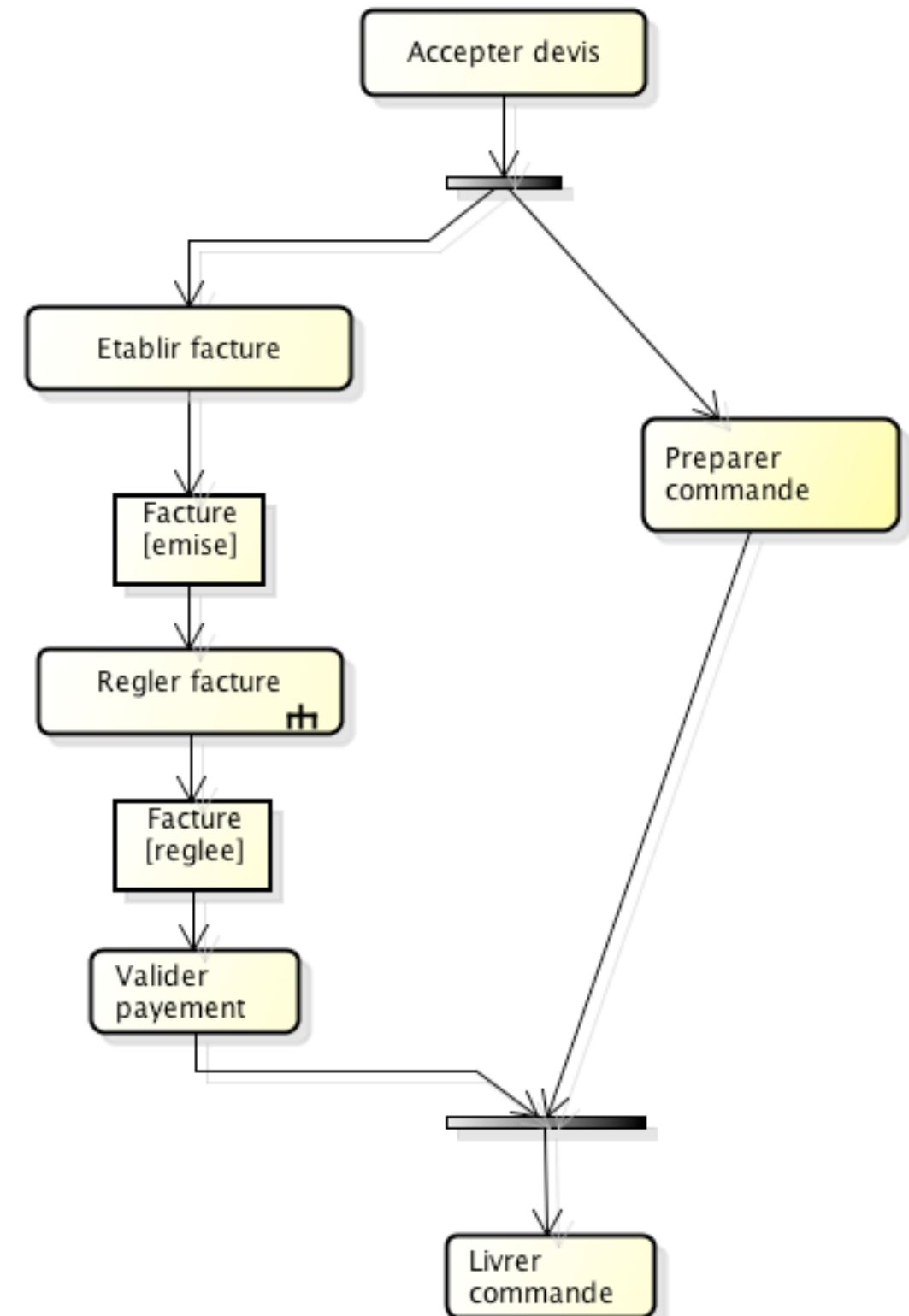
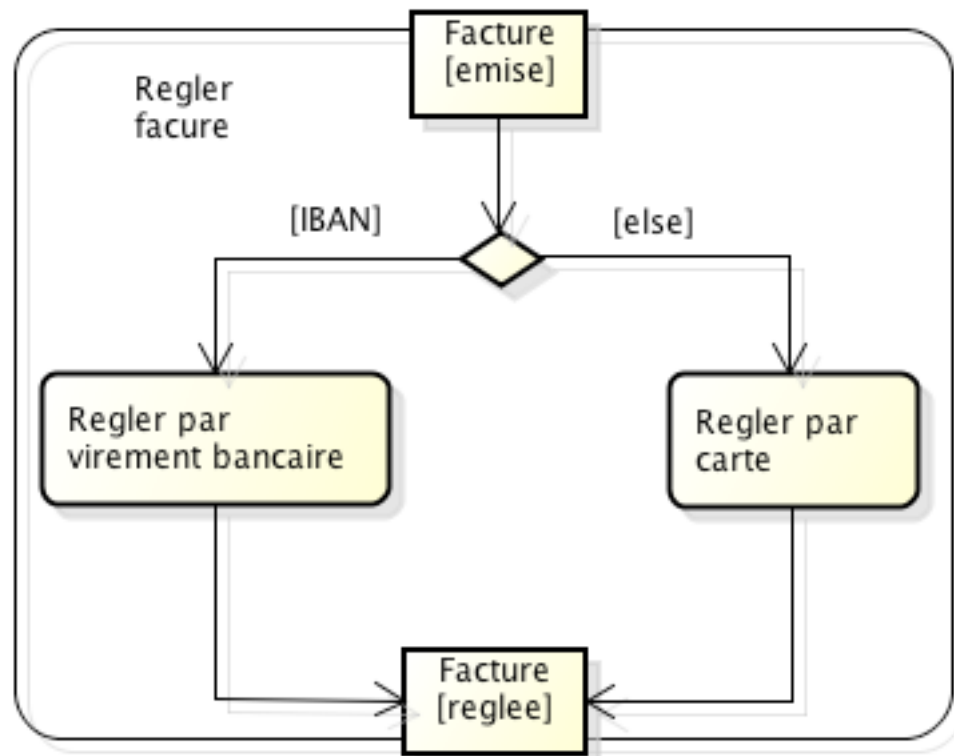


object node

Flot de données

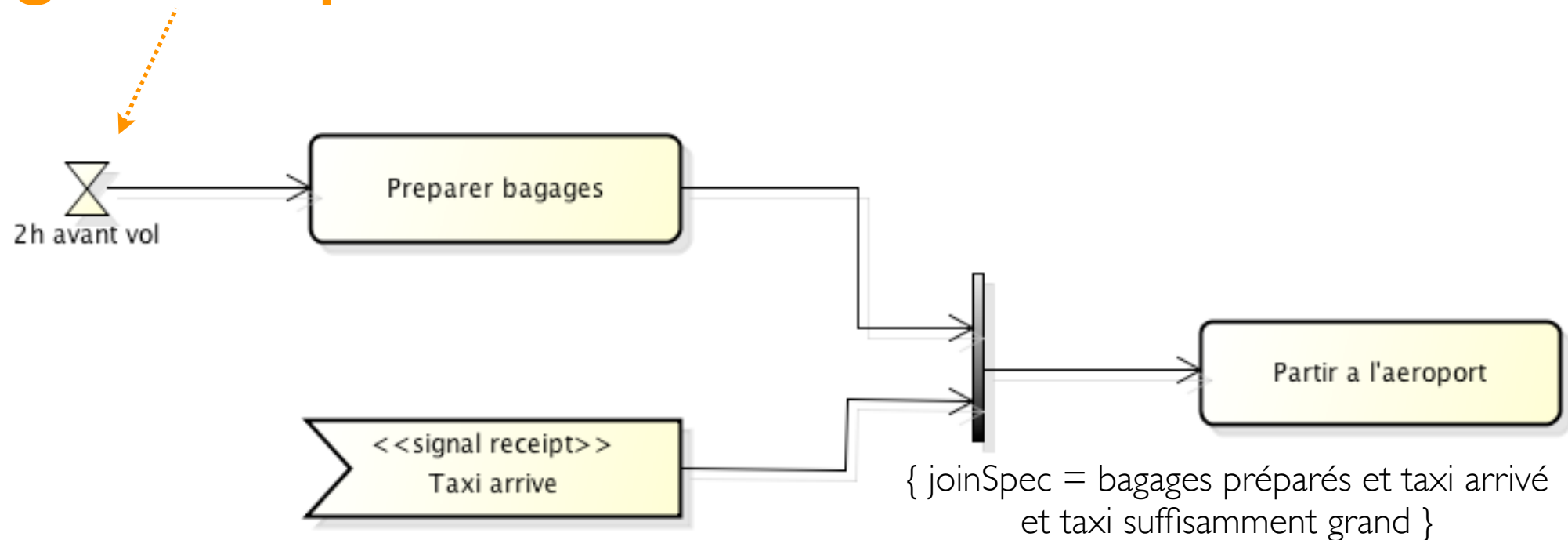


Sous-activités



Signaux - réception

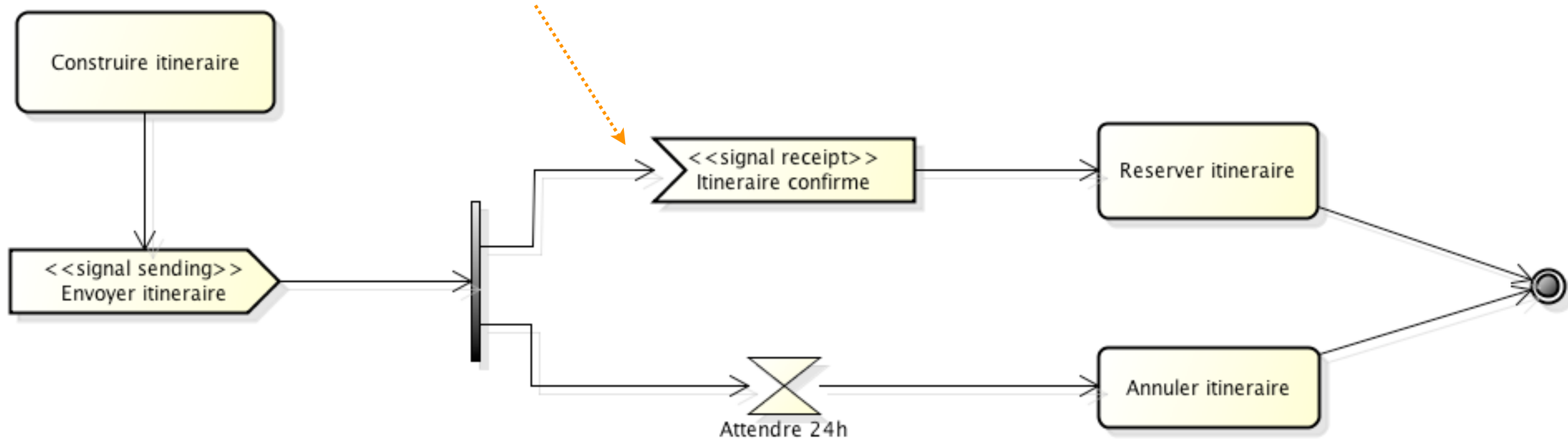
signal temporel



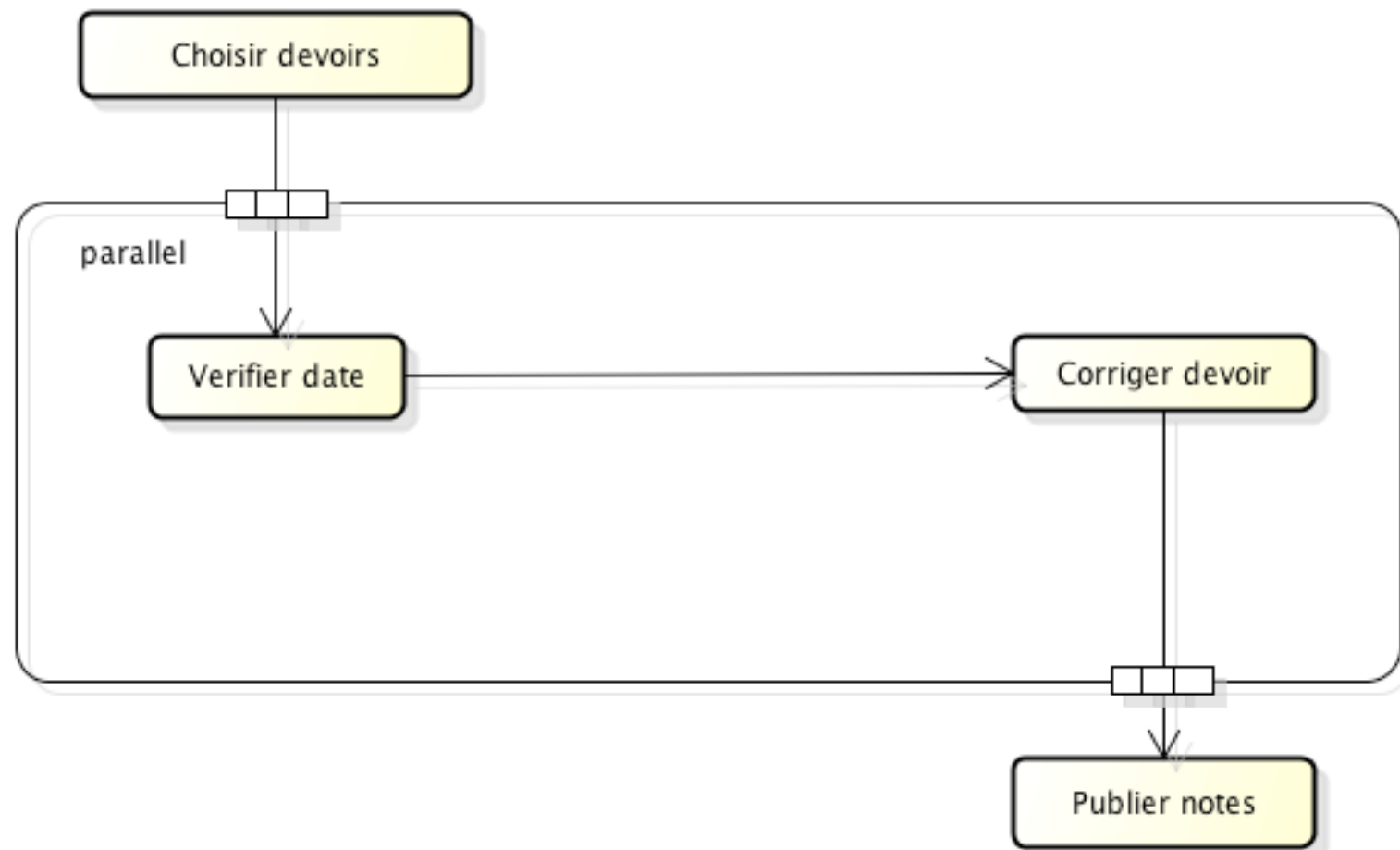
join specification

Signaux - transmission

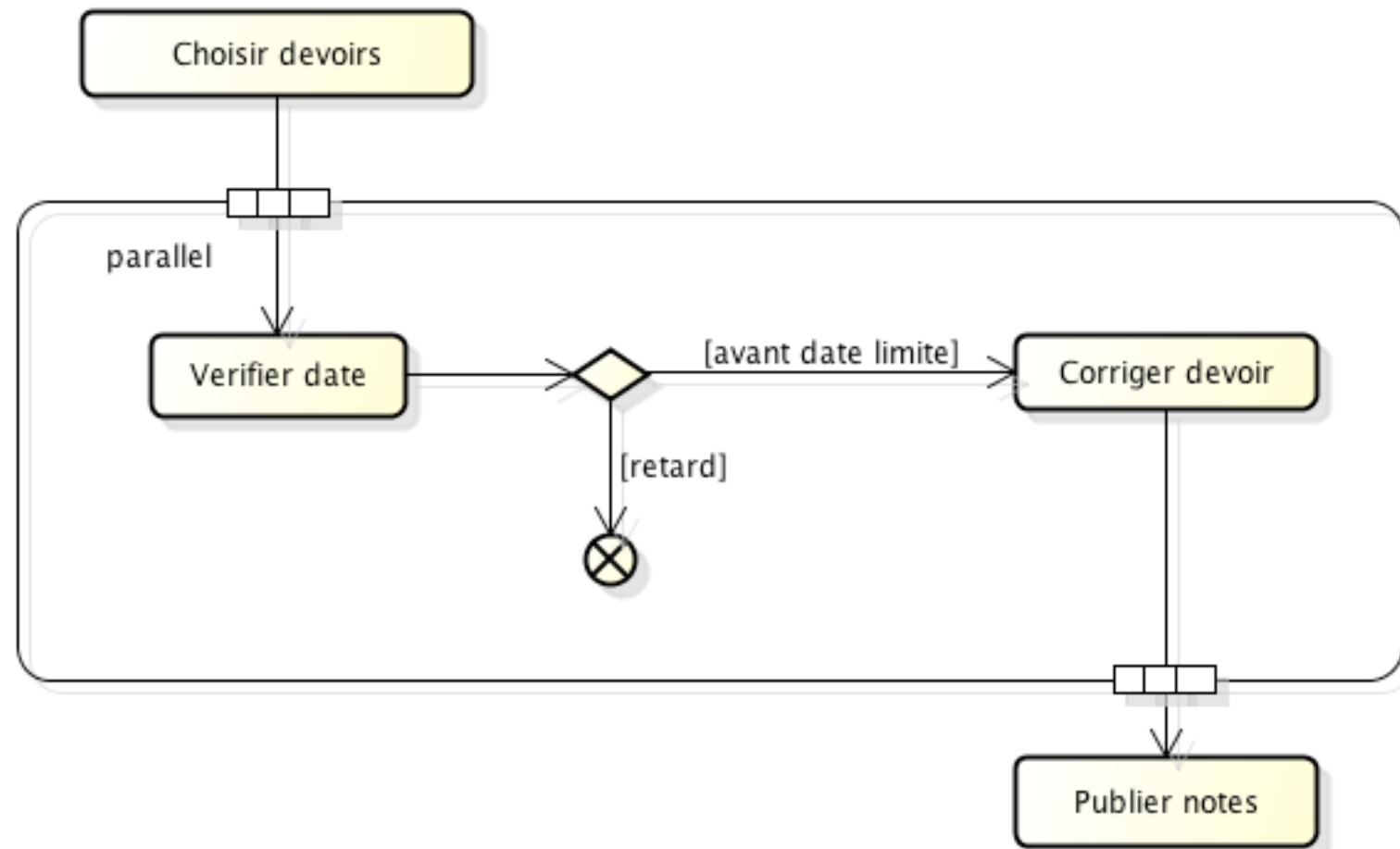
on attend le signal seulement
après activation



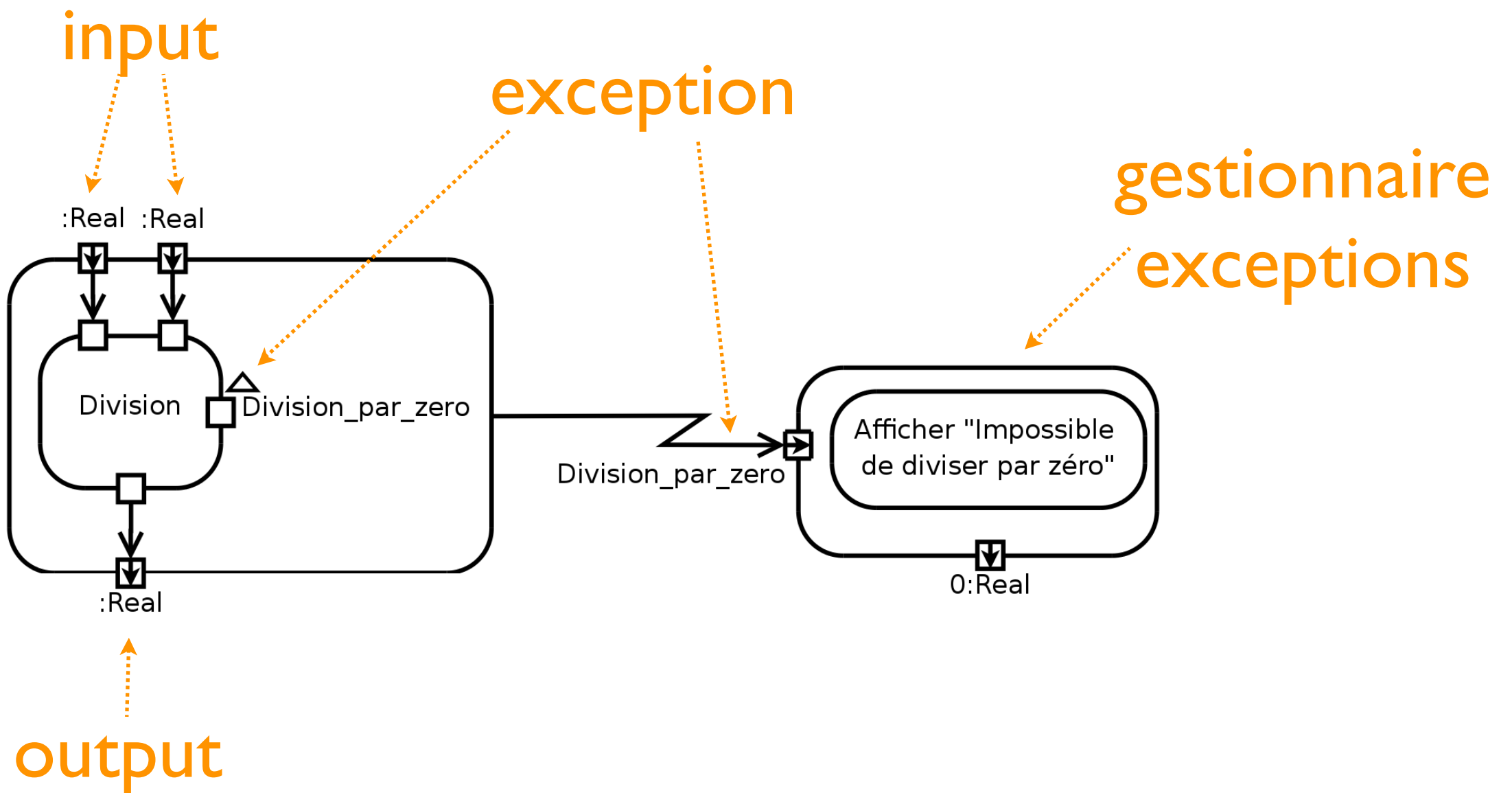
Expansions



Fin de flot

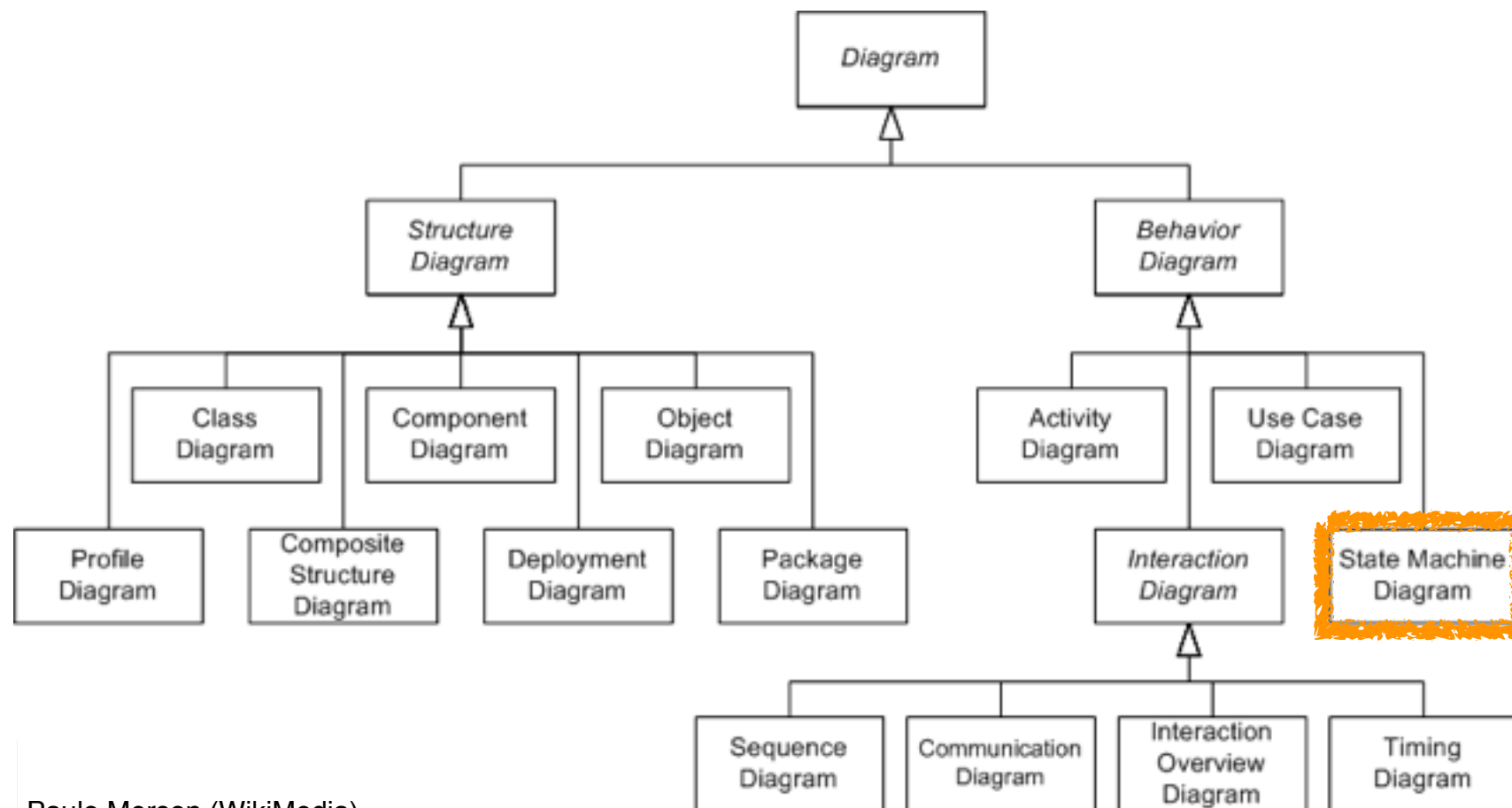


Exceptions



Questions ?

Diagrammes d'états



Paulo Merson (WikiMedia)

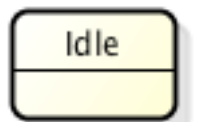
Diagramme d'états

Le comportement d'un système ou d'un objet.

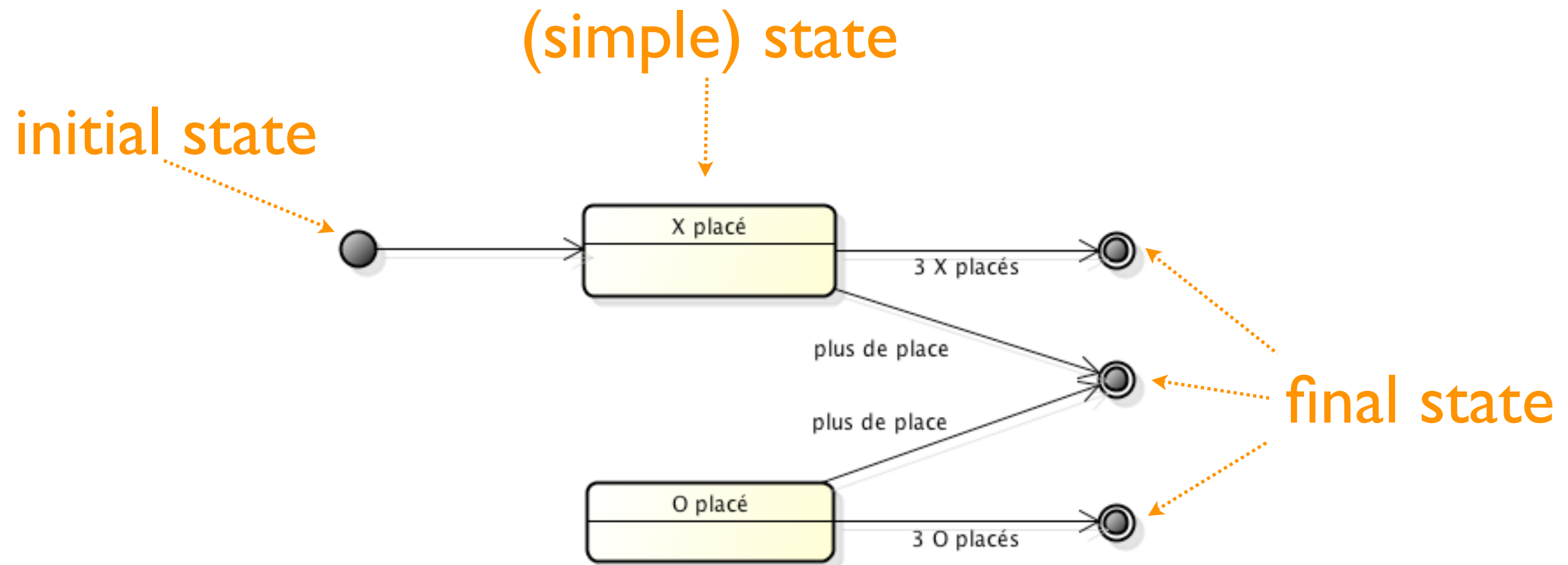
- a tout instant, un système ou un objet se trouve dans un certain état
- certains événements vont provoquer des changements d'états
- un diagramme d'état est un graphe dans lequel les états sont des nœuds et dont les arcs représentent les transitions.

Les états

- Un état se représente à l'aide d'un rectangle arrondi contenant le nom de cet état
- États spéciaux:
 - état initial (un disque noir)
 - état final (un disque noir entouré d'un cercle)



Tic-Tac-Toe



Transitions

- Une transition représente un changement d'état en réponse à un événement
- Cette transition est considérée instantanée
- L'étiquette associée à une transition est l'événement causant ce changement d'état

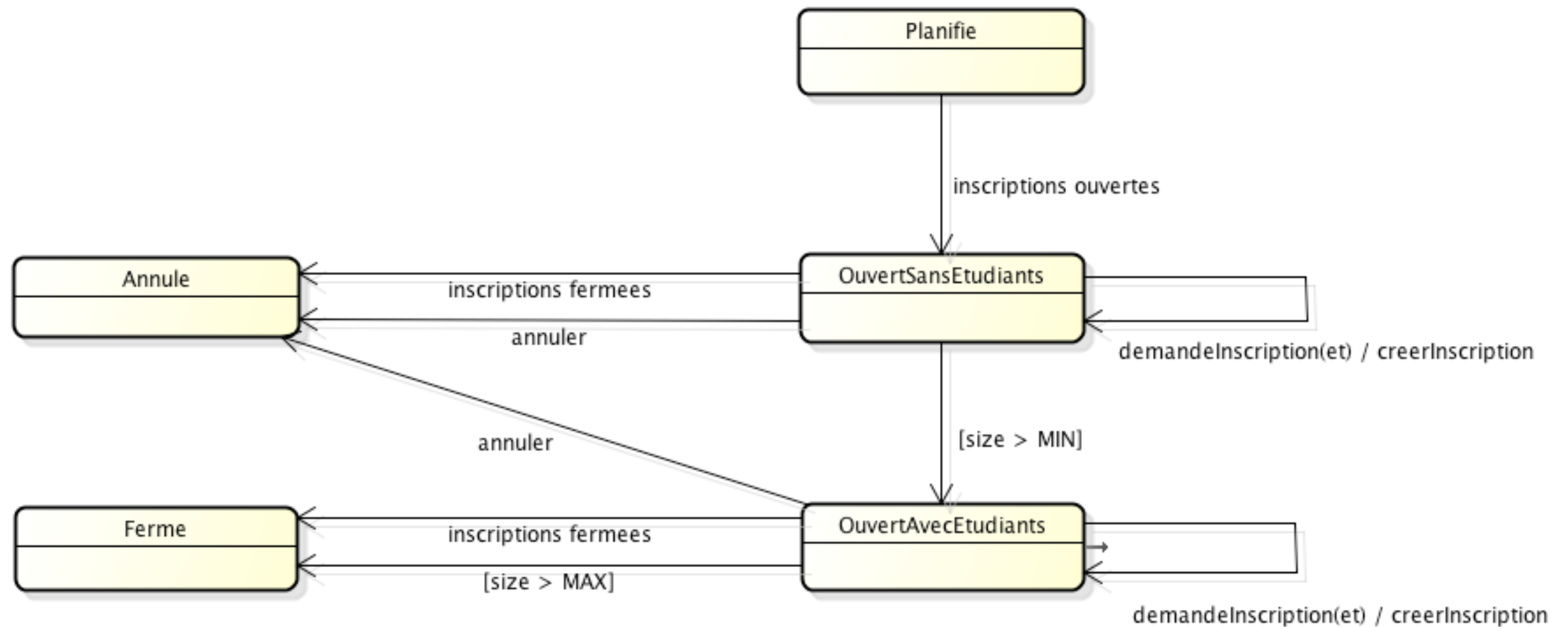
événement [garde] / action

Événement

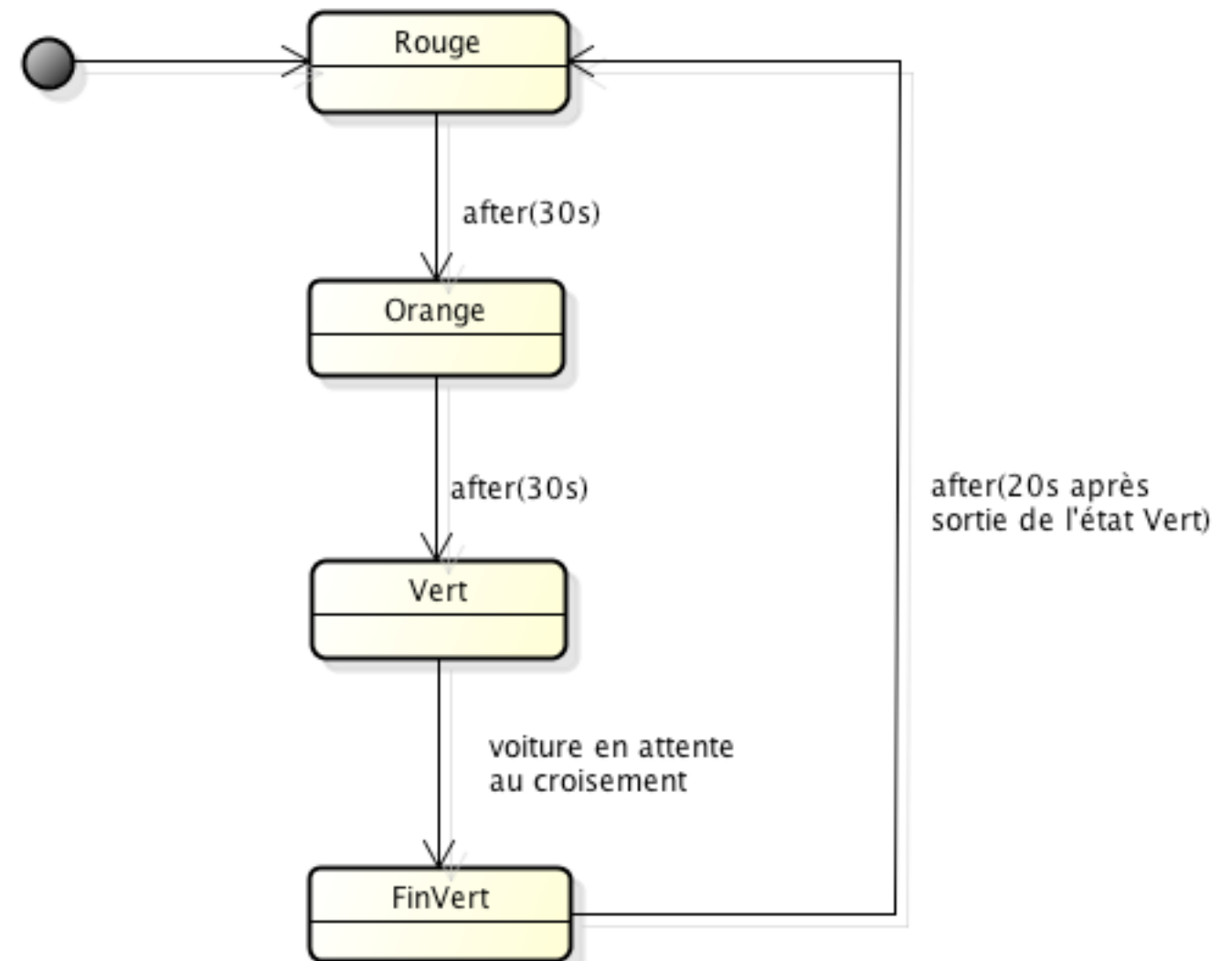
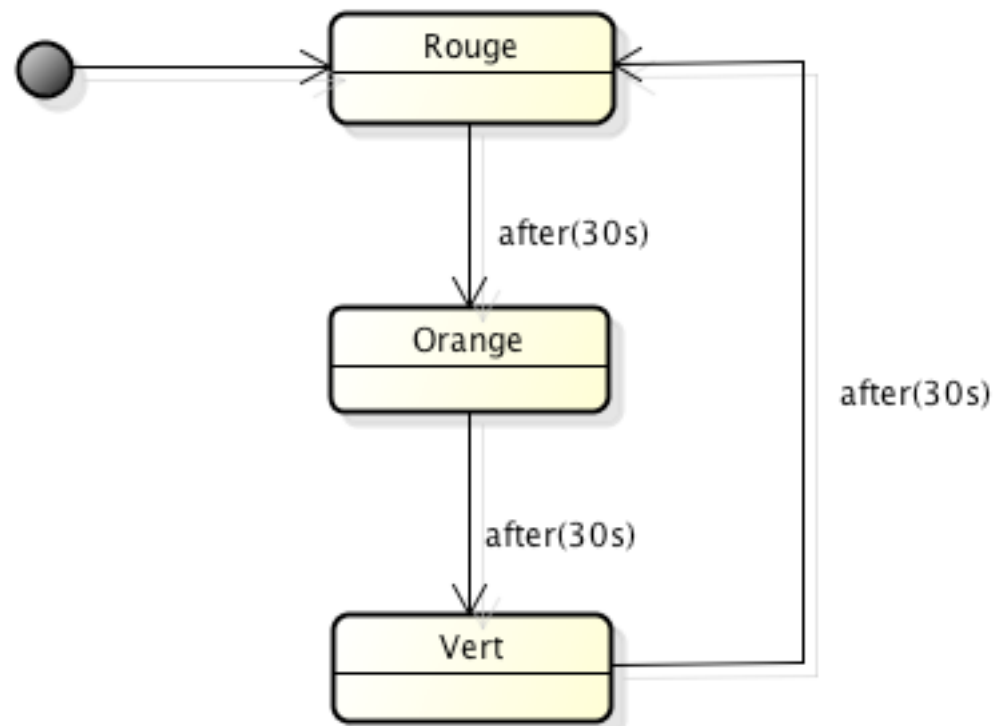
Précise qu'il s'est passé quelque chose de significatif.

- réception d'un signal, envoyé par un autre objet ou un autre acteur (envoi asynchrone),
- appel d'une opération sur l'objet récepteur (appel synchrone),
- passage du temps (`after(t)`, `when(date = . . .)`),
- changement dans la valeur d'une condition (`when(cond)`)

Exemple



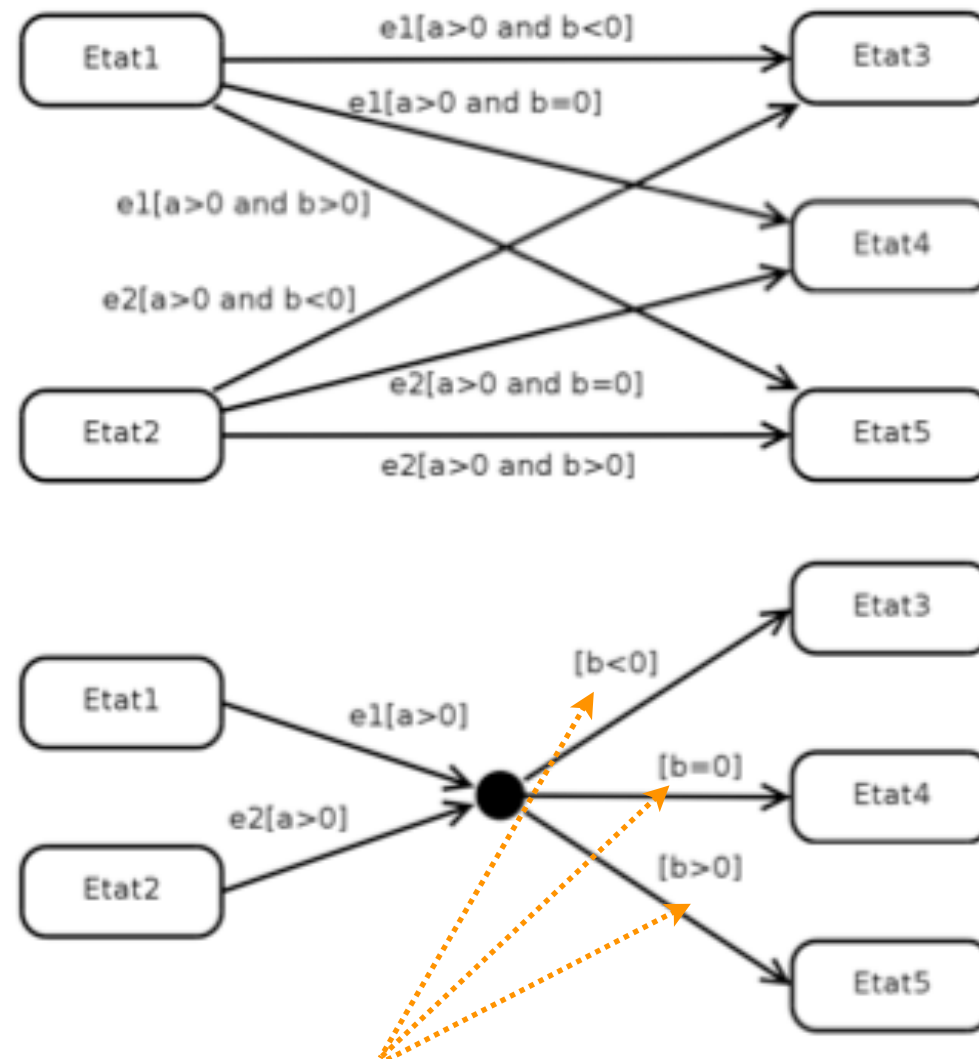
Événements temporels



Points de choix

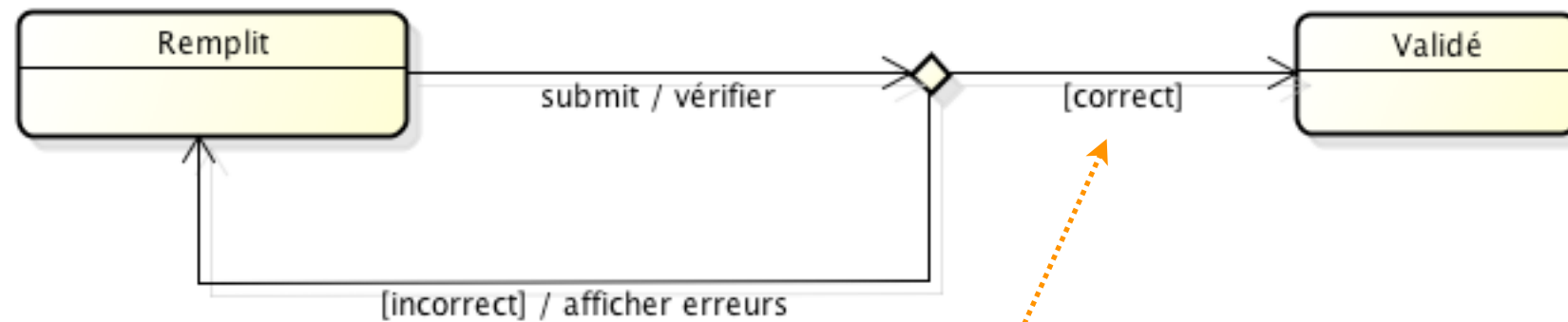
- *points de jonction*
 - ▶ partager des segments de transition
- *points de décision*
 - ▶ gardes évaluées au moment de la décision

Points de jonction



évaluation dès le déclenchement de
la transition

Points de décision



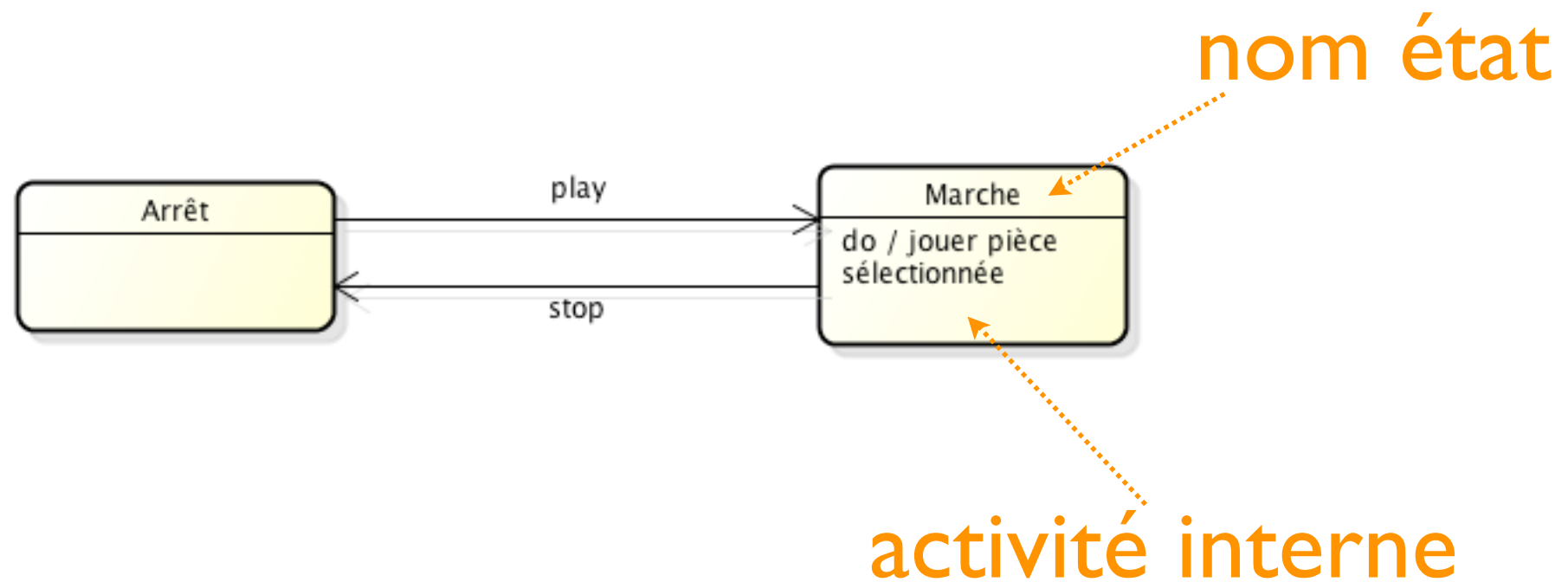
évaluation après le franchissement
du point de décision

Activités

Une activité peut être lancée lorsque le système se trouve dans un état

- Une activité a une durée
- En réponse à la terminaison de l'activité, le système peut effectuer un changement d'état
- Les transitions peuvent aussi se produire lorsque l'activité est en cours:
 - ▶ l'activité se termine
 - ▶ le changement d'état s'effectue

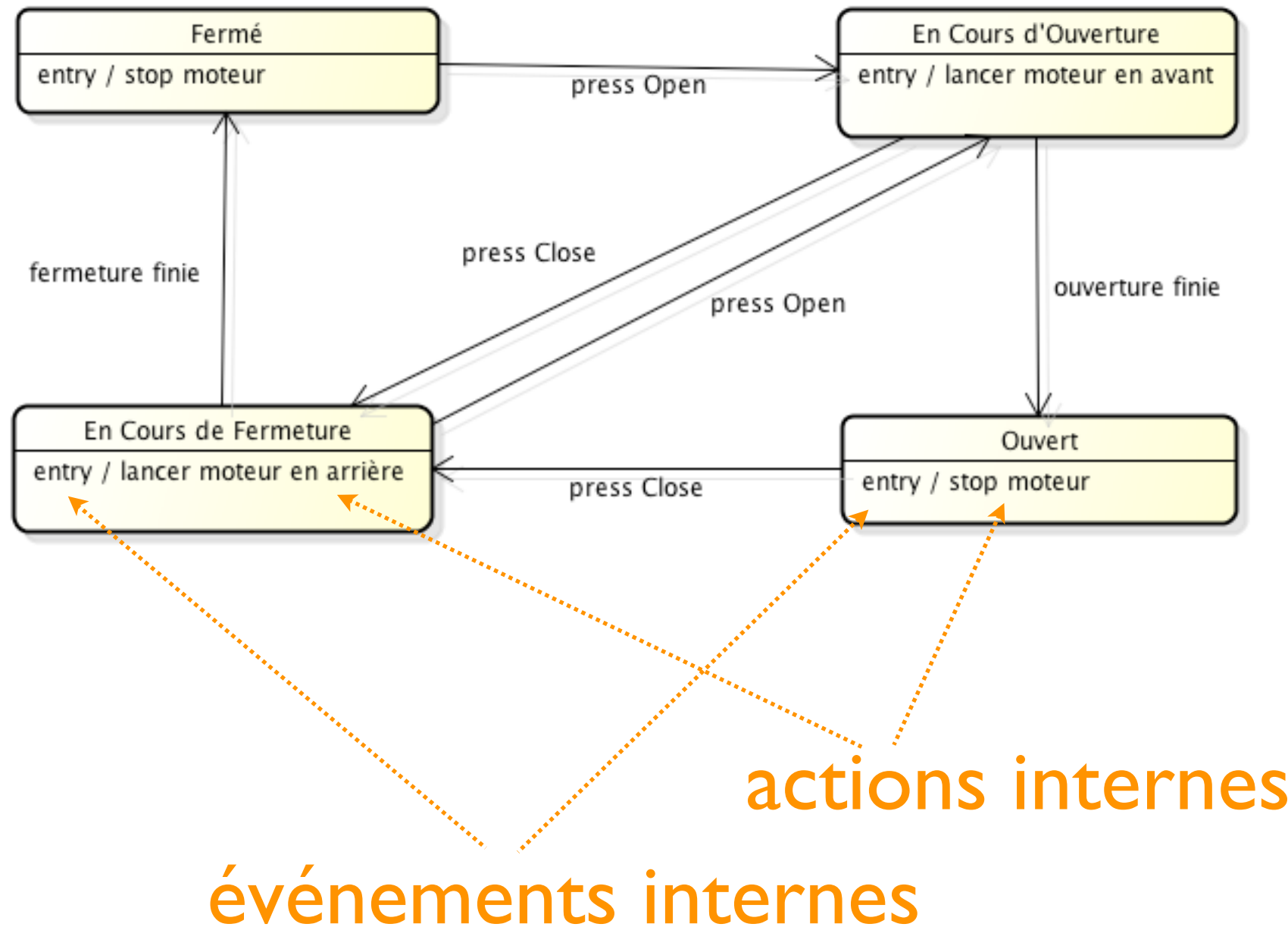
Etats avec activités



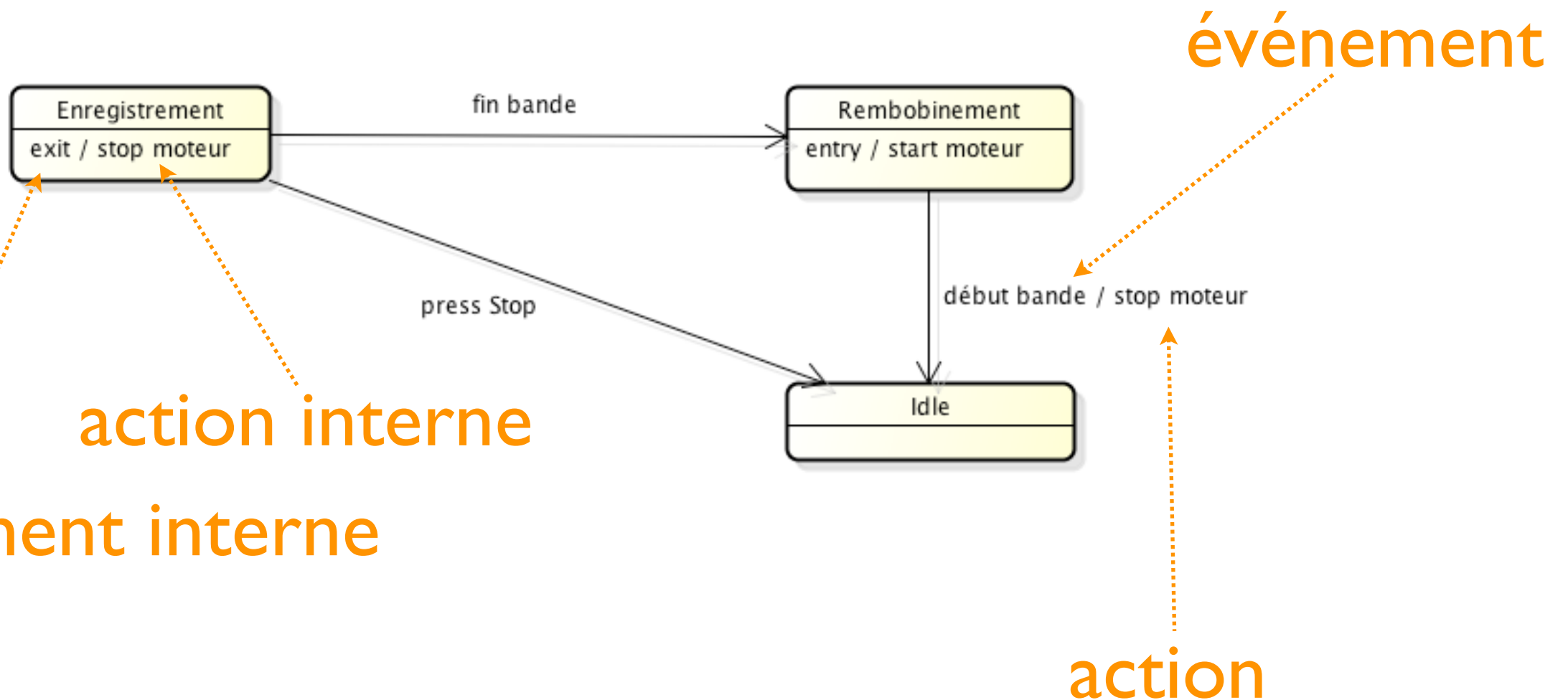
Actions

- Une action se réalise de façon instantanée
 - ▶ Lorsqu'une transition se produit
 - ▶ Lorsque le système entre dans un certain état
 - ▶ Lorsque le système reste dans un certain état
 - ▶ Lorsque le système sort d'un certain état

Activités internes



Événements et Actions

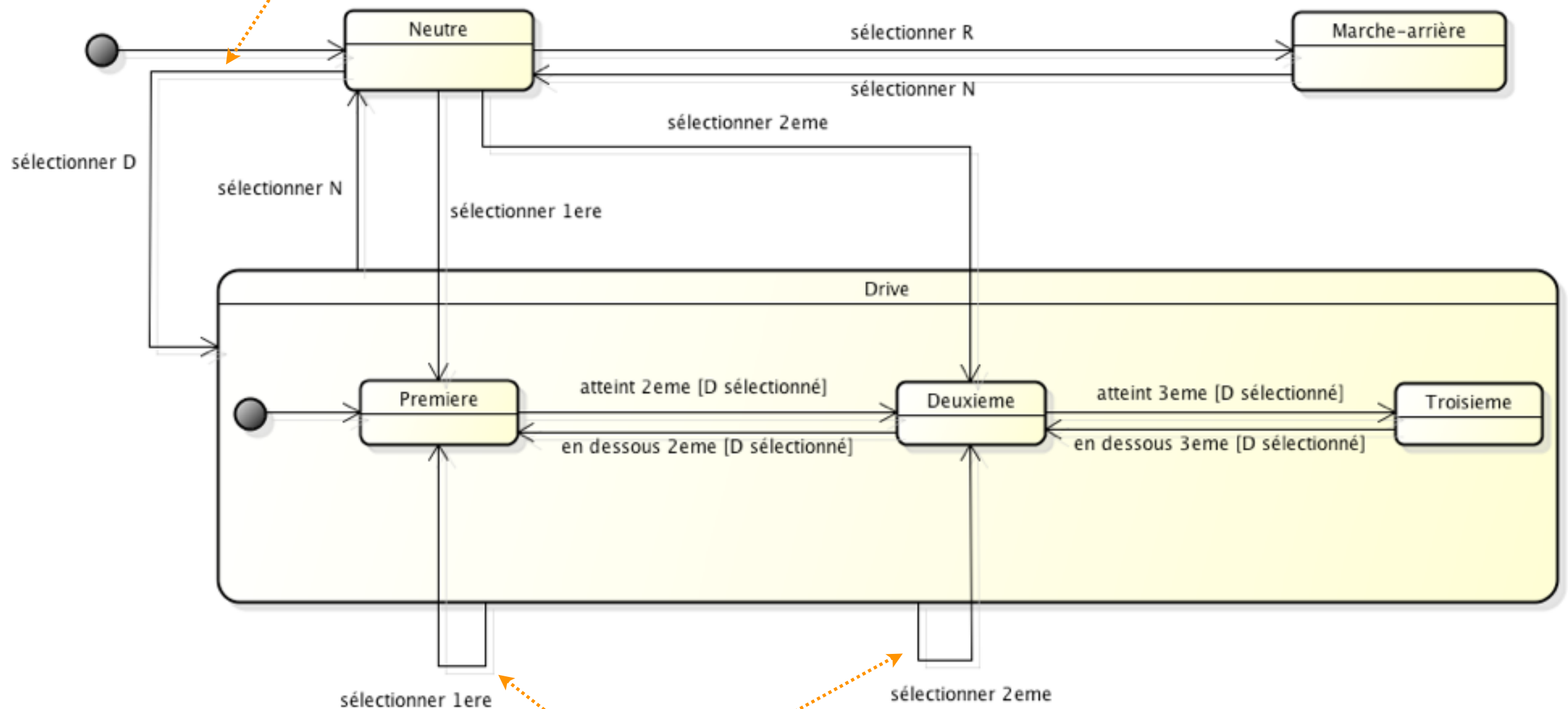


Activités internes

- do: activity
- entry: entry action
- exit: exit action

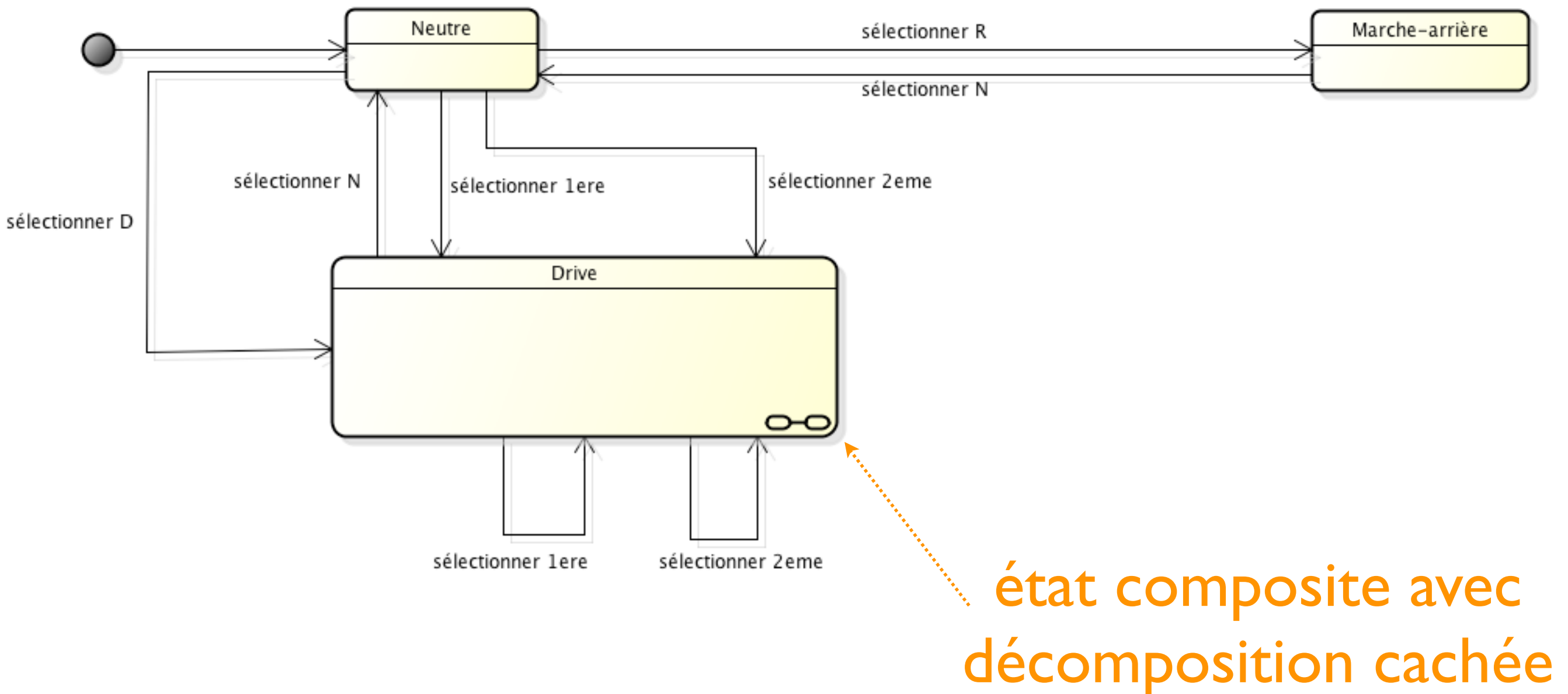
Etats composites

transition vers l'état initial (imbriqué)

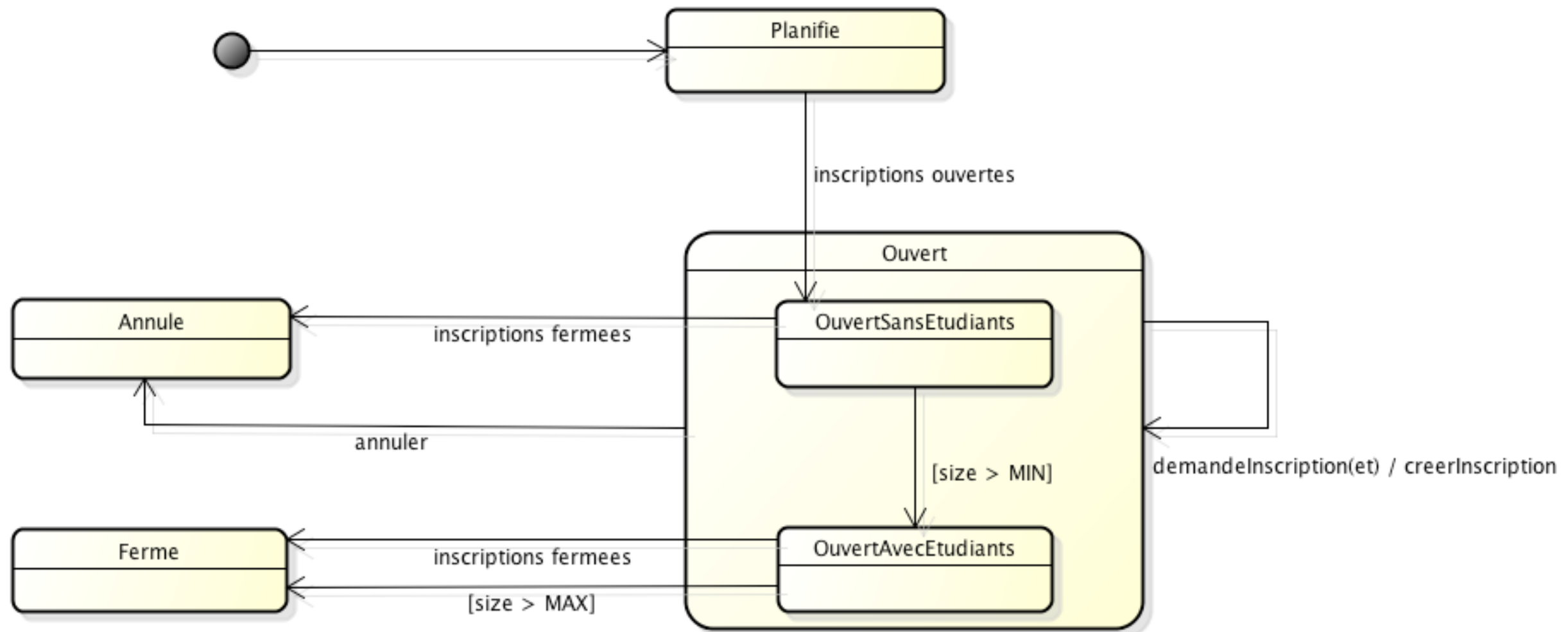


transitions possibles depuis tout état imbriqué

Etats composites



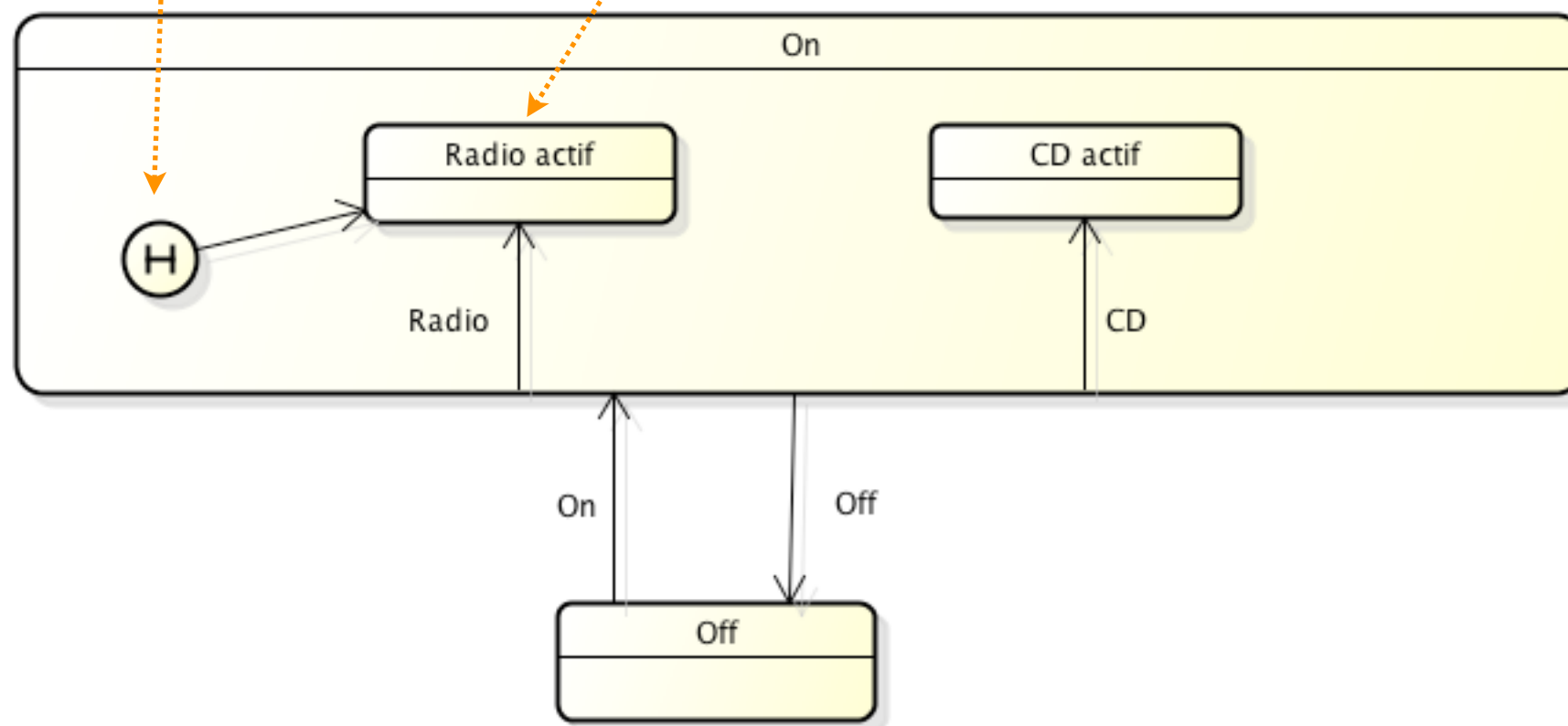
Etats composites



Etat historique

état historique

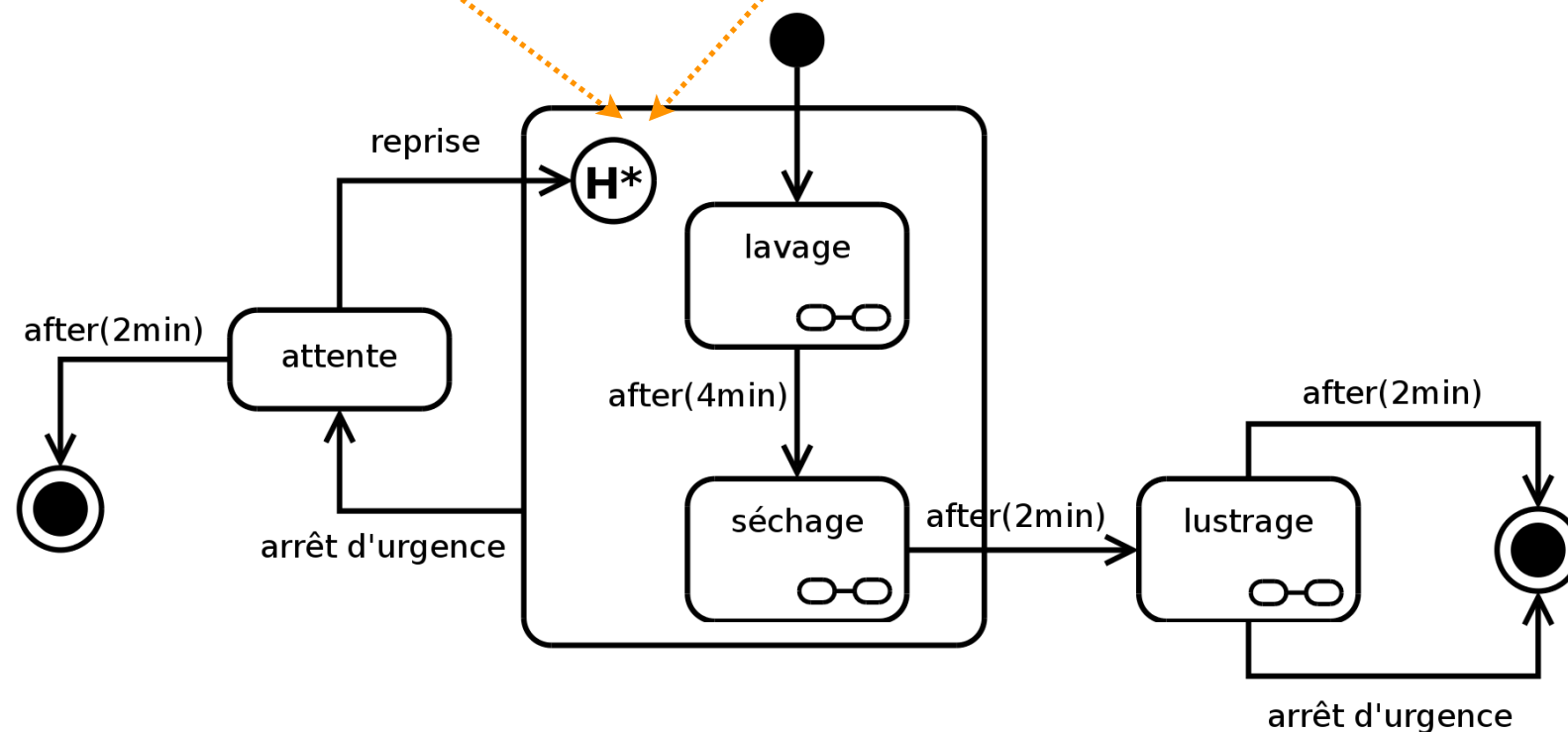
état historique par défaut
(si « On » n'a jamais été actif)



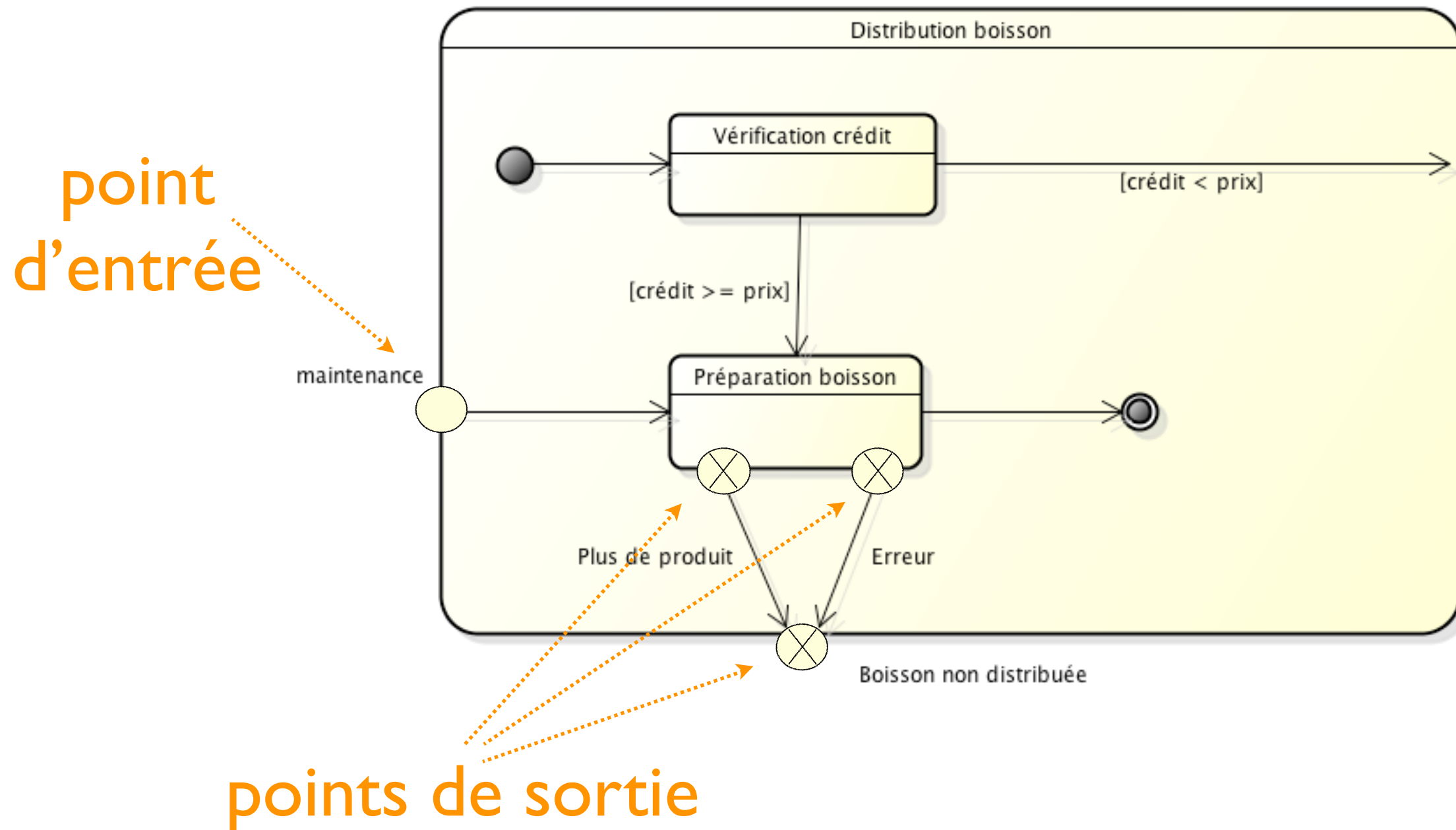
Etat historique

état historique (profond)

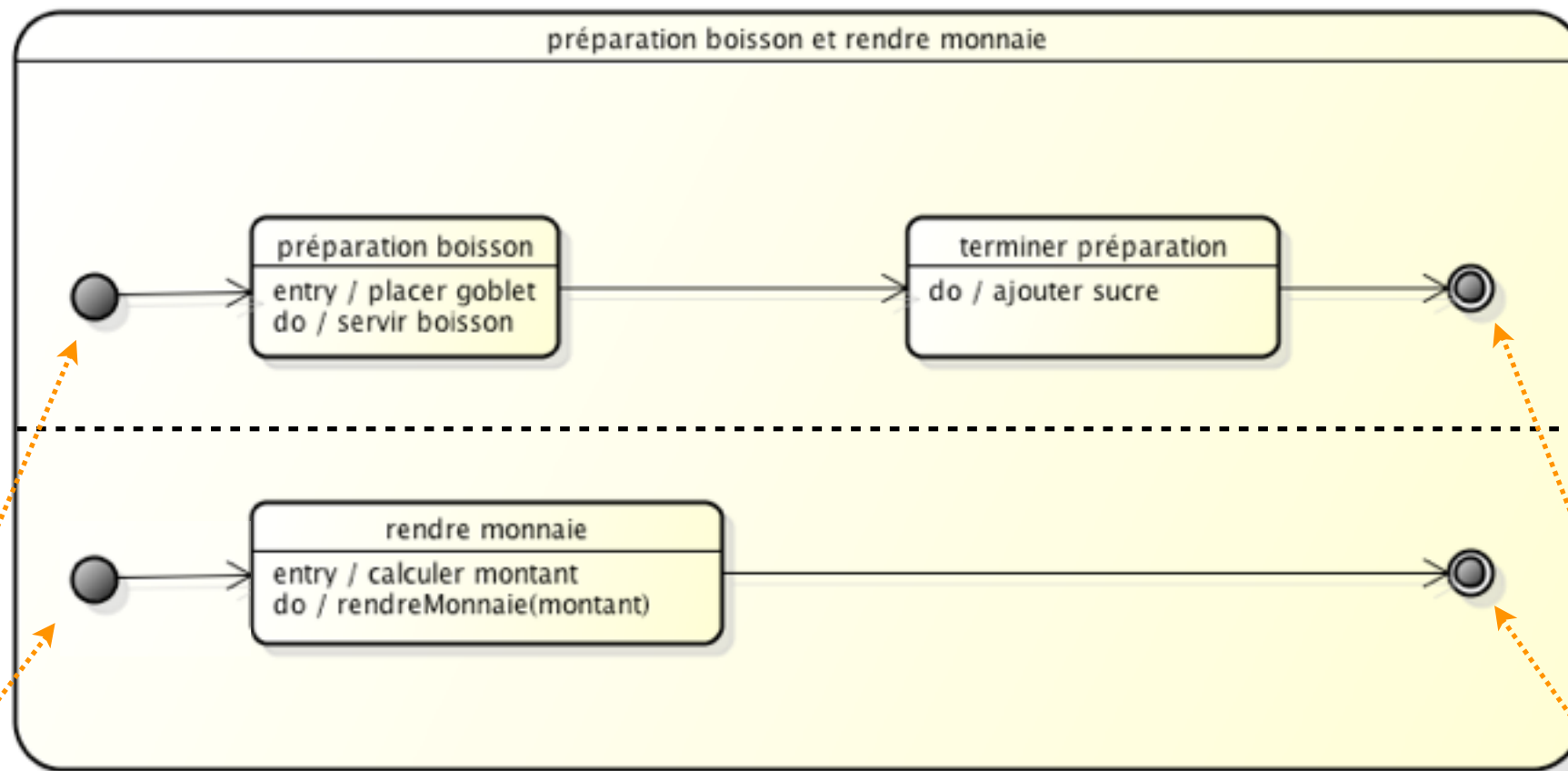
cible = le dernier (sous-)état
visité de l'état englobant



Interfaces



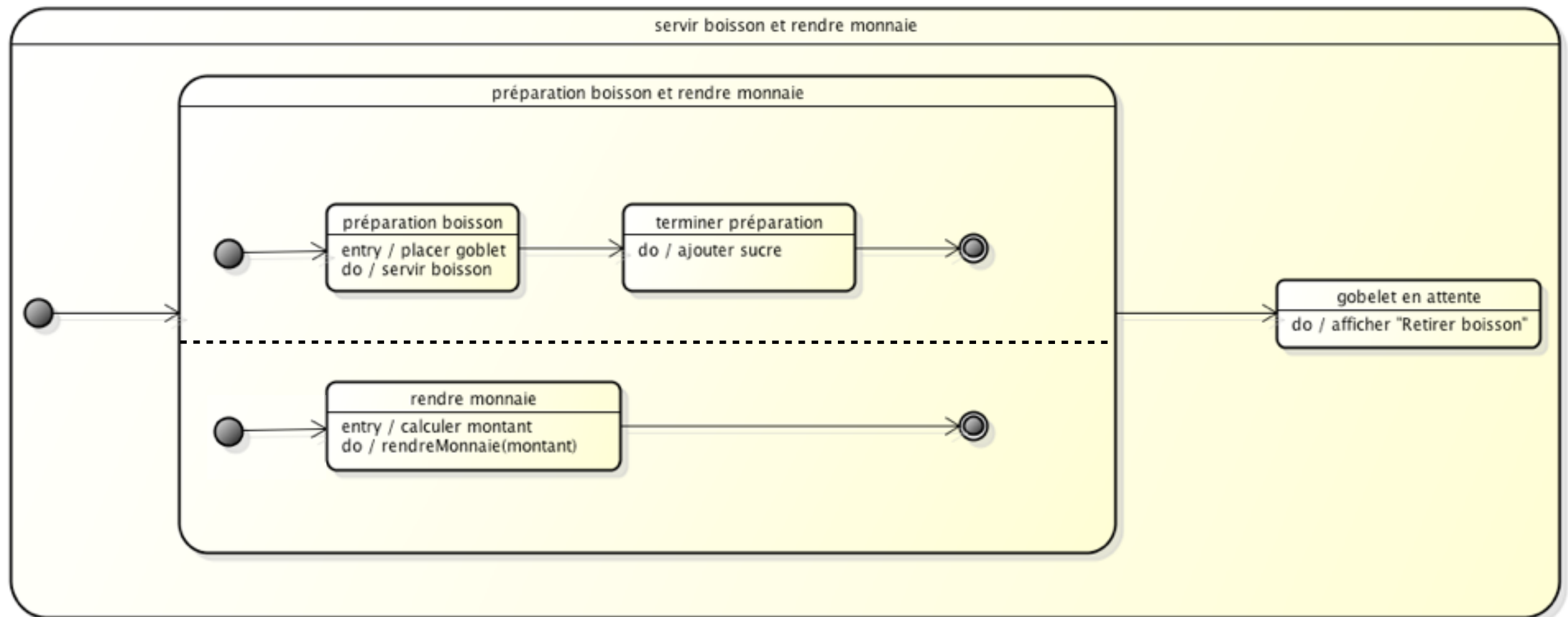
Etats orthogonaux



atteints en parallèle

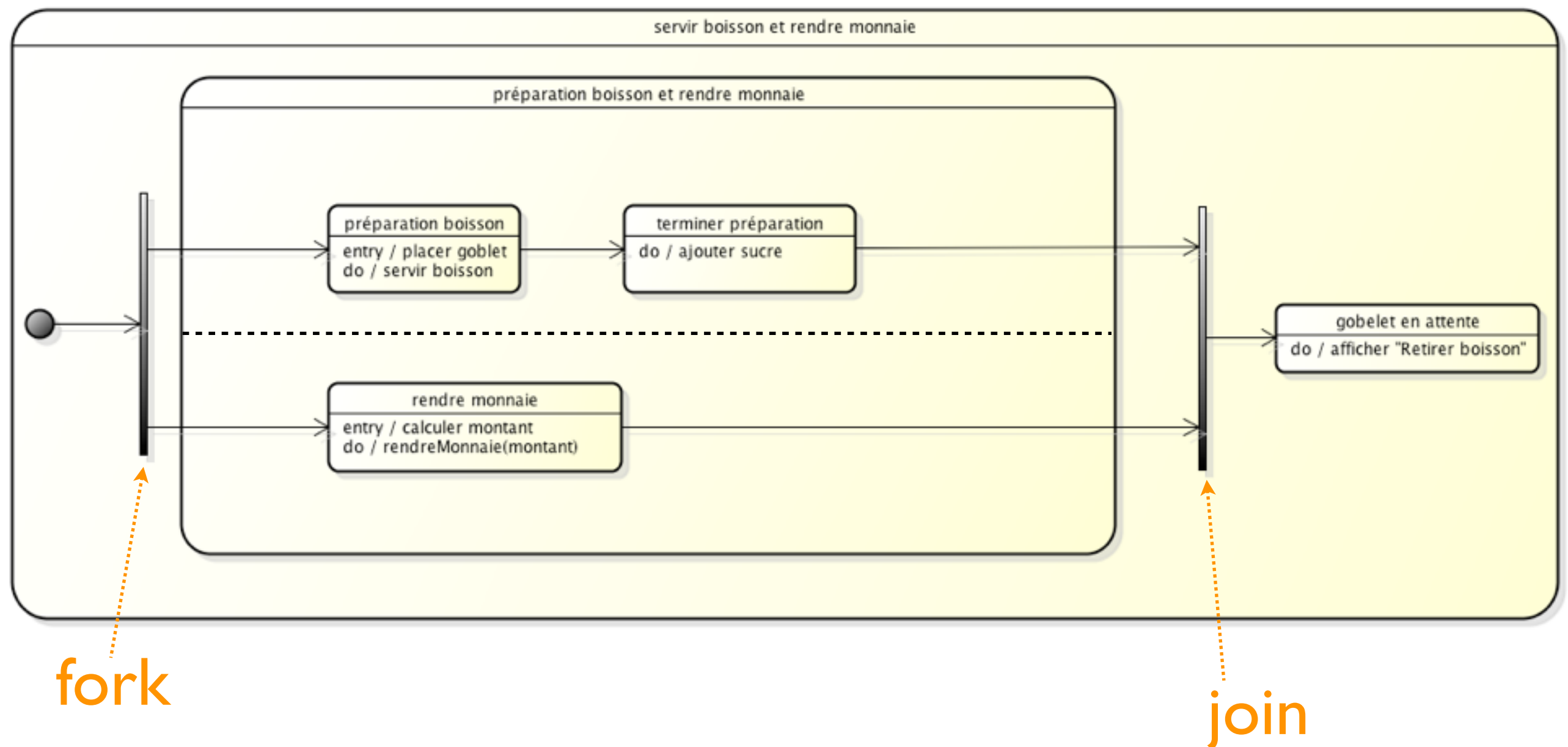
atteints avant de
terminer l'état

Etats orthogonaux

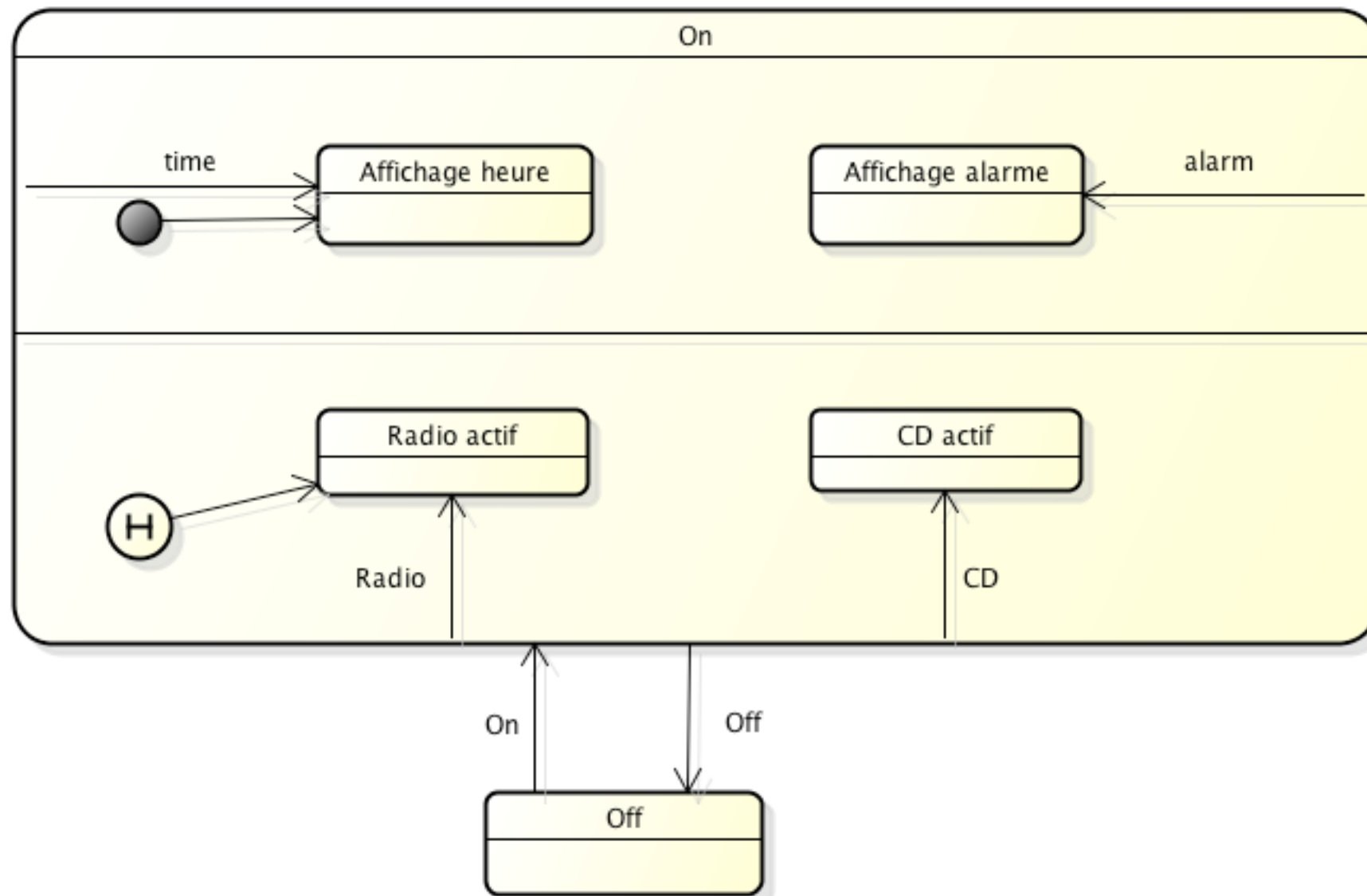


Etats orthogonaux

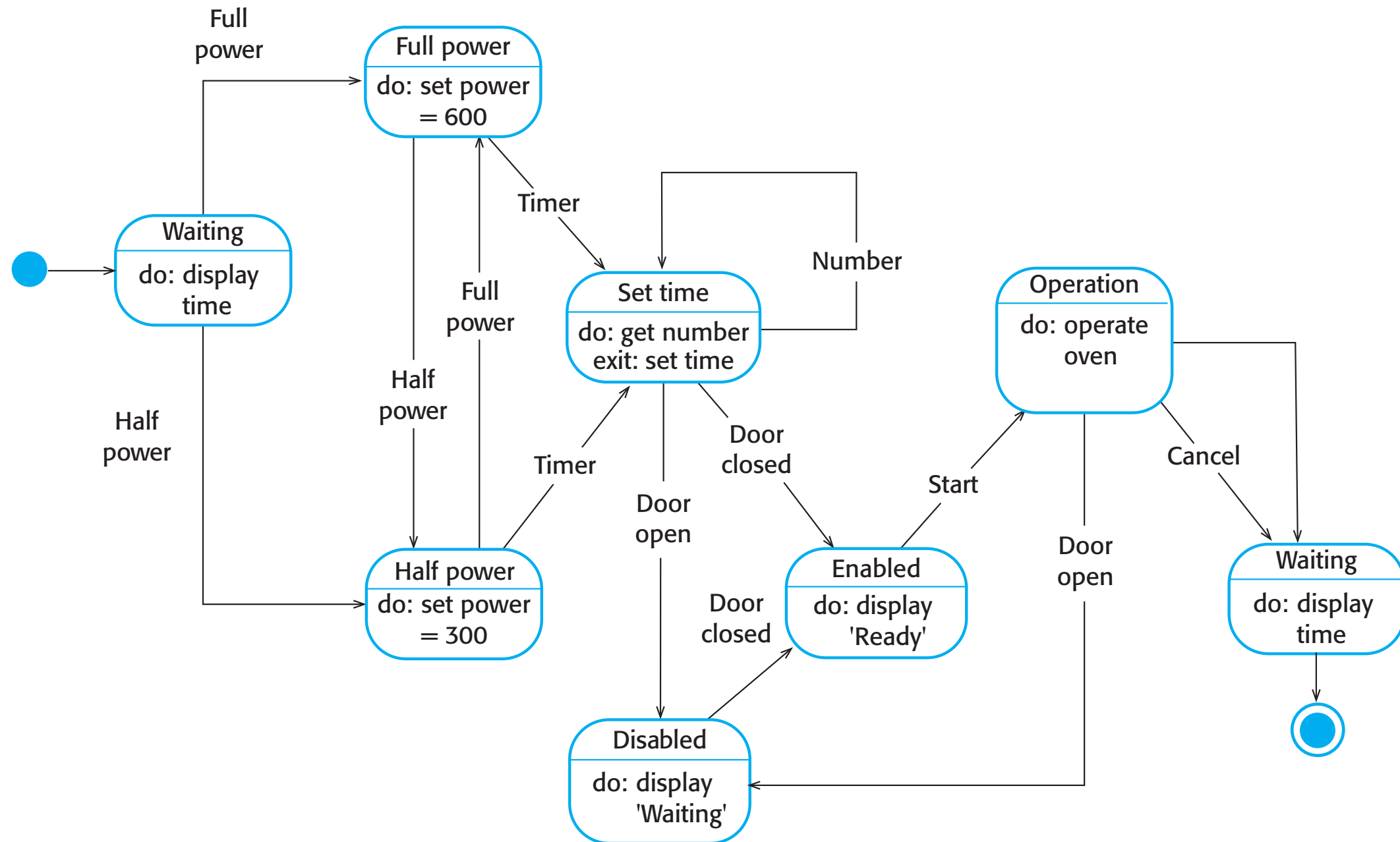
- fork / join -



Etats orthogonaux



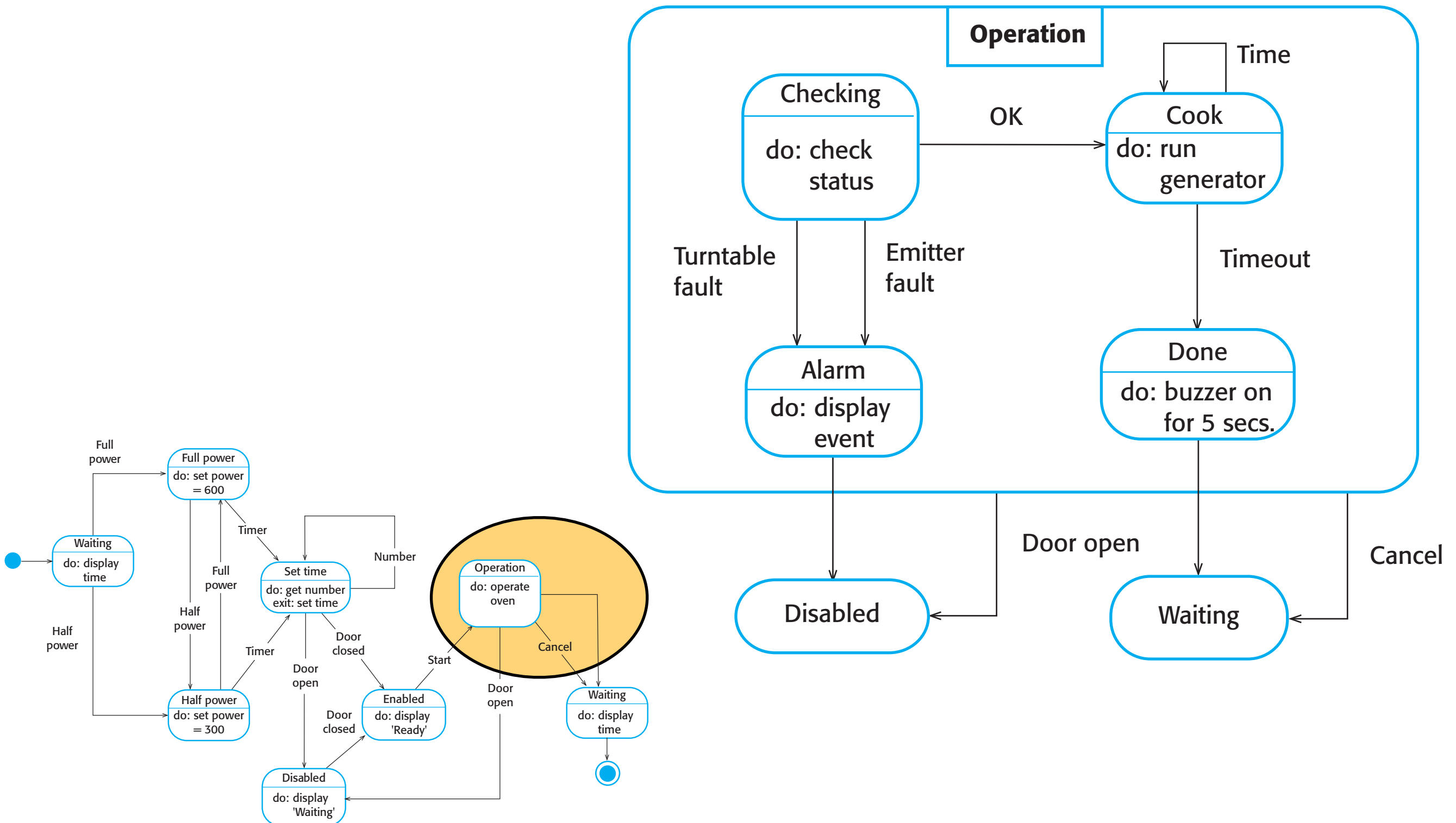
Four aux micro-ondes



State	Description
Waiting	The oven is waiting for input. The display shows the current time.
Half power	The oven power is set to 300 watts. The display shows 'Half power'.
Full power	The oven power is set to 600 watts. The display shows 'Full power'.
Set time	The cooking time is set to the user's input value. The display shows the cooking time selected and is updated as the time is set.
Disabled	Oven operation is disabled for safety. Interior oven light is on. Display shows 'Not ready'.
Enabled	Oven operation is enabled. Interior oven light is off. Display shows 'Ready to cook'.
Operation	Oven in operation. Interior oven light is on. Display shows the timer countdown. On completion of cooking, the buzzer is sounded for five seconds. Oven light is on.

Stimulus	Description
Half power	The user has pressed the half-power button.
Full power	The user has pressed the full-power button.
Timer	The user has pressed one of the timer buttons.
Number	The user has pressed a numeric key.
Door open	The oven door switch is not closed.
Door closed	The oven door switch is closed.
Start	The user has pressed the Start button.
Cancel	The user has pressed the Cancel button.

Etat «Operation»



Questions ?