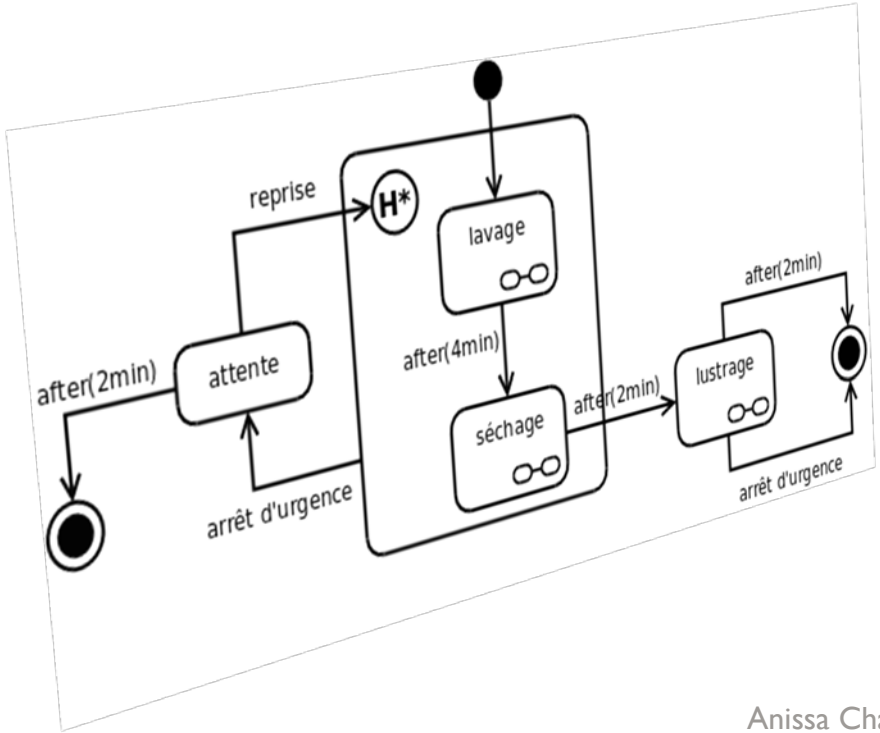
Diagramme d'état-Transition

Mme CHALOUAH Anissa

I.S.E.T Bizerte

Département Technologies de l'informatique



Introduction (1)

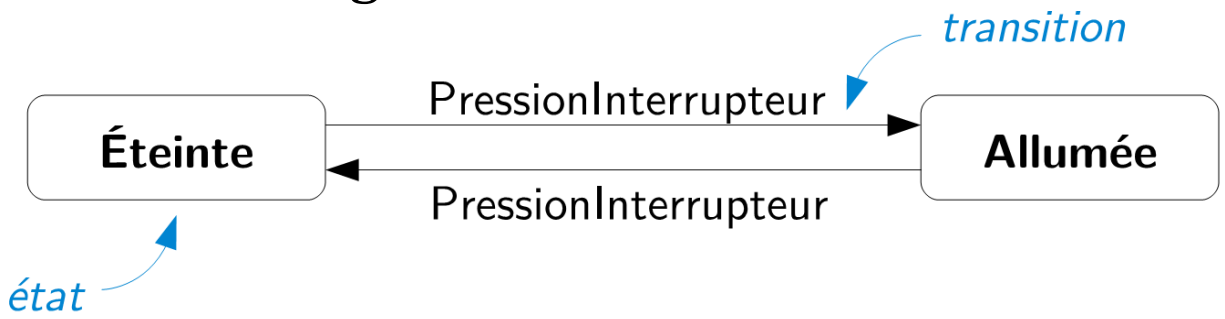
- Les objets d'une classe ne sont pas figés :
  - Ils peuvent évoluer et changer d'état au cours de leur cycle de vie (CV : intervalle de temps entre la création et la suppression de l'objet)
- Un DET est une description des changements d'états d'un objet (ou d'un composant) :
  - en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

Introduction (2)

- Un diagramme d'états – transitions (DET) décrit le comportement dynamique d'une entité (objet, composant,…)

  Comportement décrit par états + transitions entre les états

  - **État** : abstraction d'un moment de la vie d'une entité pendant lequel elle satisfait un ensemble de conditions.
  - **Transition** : changement d'état



Introduction (3)

- Le DET d'une classe est une description des évolutions possibles de ses objets.
- Il donne :
  - la liste des états que peut prendre un objet durant son cycle de vie ;
  - les événements déclenchant les changements d'états ;
  - les éventuelles conditions qu'il doit vérifier avant de changer d'état ;
  - les opérations qui le font passer d'un état à un autre.

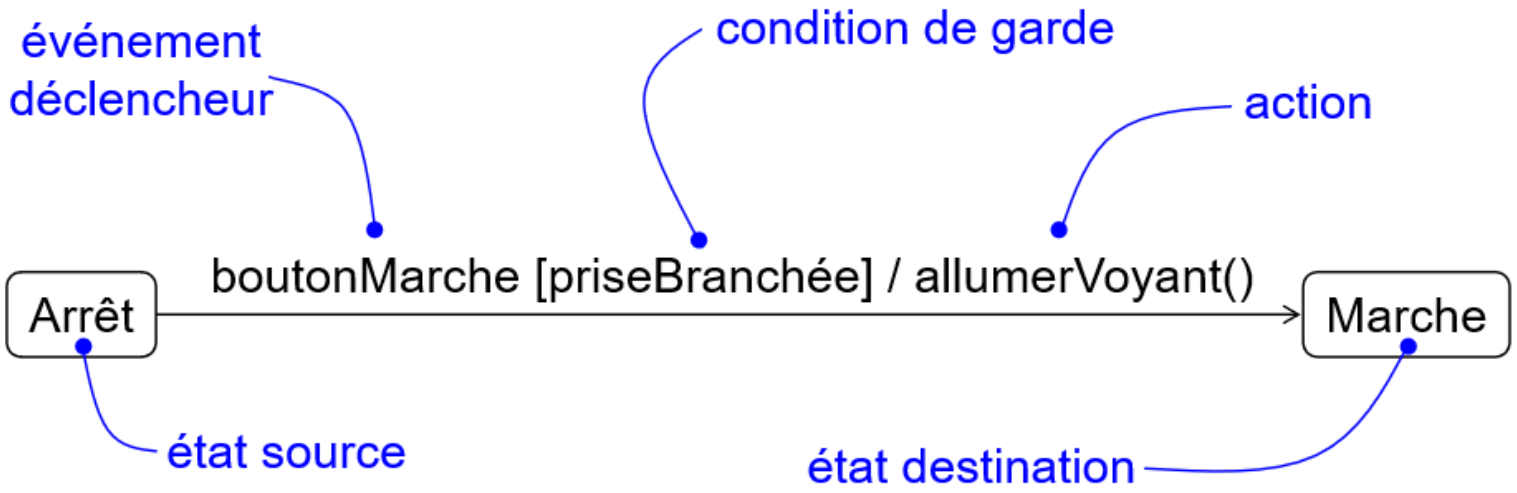
Introduction (4)

- Une classe n’a pas obligatoirement de DET, comme elle peut en avoir plusieurs, selon différentes sémantiques.
- Le modèle dynamique comprend plusieurs diagrammes d'états-transitions
- Chaque diagramme d'états ne concerne qu'une seule classe.

Sémantique

A un instant *T*, suite à l'arrivée d'un événement *Evt*, ayant les attributs *Att*, et sous certaines conditions gardes, l'objet passe de l'*Etat i* à l'*Etat j* par l'activation de l'action *Action*.

Exemple


événement
déclencheur
condition de garde
action
boutonMarche [priseBranchée] / allumerVoyant()
Arrêt
Marche
état source
état destination

Notion d'état

- Un état = une étape dans le cycle de vie d'un objet
- Chaque objet possède, à un instant donné, un état particulier.
  - Dans un état donné, l' objet satisfait des conditions, réalise des actions, ou il est tout simplement en attente d'événements.
  - L'état d'un objet est déterminé par l'ensemble des valeurs de ses attributs et de la présence de liens avec d'autres objets.
  - Un état se caractérise par sa durée et sa stabilité.

Notion d'état

- Les états sont représentés par des rectangles aux coins arrondis.

- Ils existent en plus deux états particuliers :
  - L'état initial
  - L'état final

État initial

- L'état initial est un pseudo-état qui définit le point de départ par défaut pour l'automate ou le sous-état.
  - Lorsqu'un objet est créé, il entre dans l'état initial.

Etat initial

- Un diagramme d'états a toujours un et un seul état initial pour un niveau hiérarchique donné.

État final

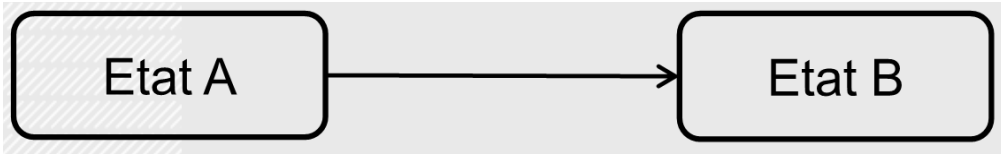
- L'état final est un pseudo-état qui indique que l'exécution de l'automate ou du sous-état est terminée.

Etat final

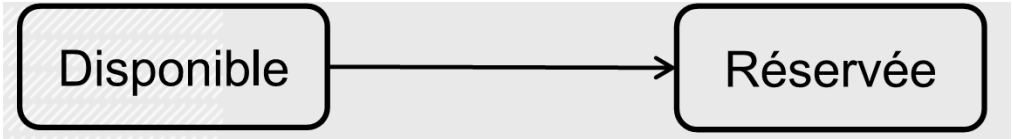
- Un DET peut n'avoir aucun état final ou plusieurs.
- Aucune transition ne peut avoir comme origine l'état final.

Notion de transition

- Une transition indique le passage d'un état (état source) dans un autre (état cible).
- Elle est représentée par une flèche orientée de l'état source vers l'état cible.

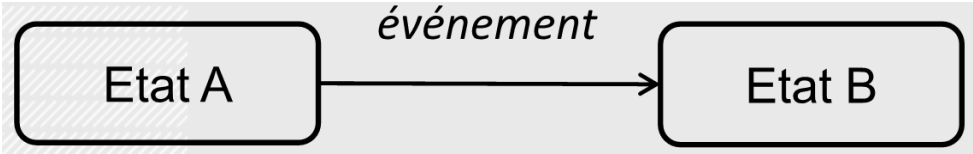


- Exemple : Place de parking



Notion d'évènement

- Une transition est déclenchée par un événement : c'est l'arrivée d'un événement qui conditionne la transition.
- Stimulus qui provoque une (ou plusieurs) transition(s). A chaque stimulus peut correspondre une **action** responsable des modifications de l'objet (les valeurs des attributs)



Syntaxe d'un événement

Nom de l'événement (Nom de paramètre : Type,….)

La description complète d'un événement est donnée par :

- Nom de l'événement
- Liste des paramètres
- Objet expéditeur
- Objet destinataire
- Description textuelle

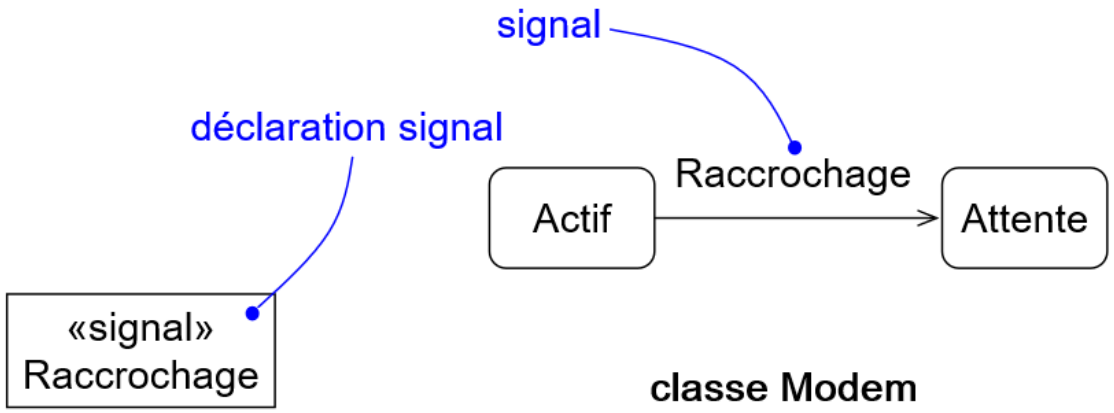
On se limite généralement à donner le nom de l'événement.

Types d'évènements(1)

- Il y a 4 types d'évènements :
  - Évènement de type signal (signal)
  - Évènement d'appel (call)
  - évènement de changement (change)
  - Évènement temporal (after ou when)

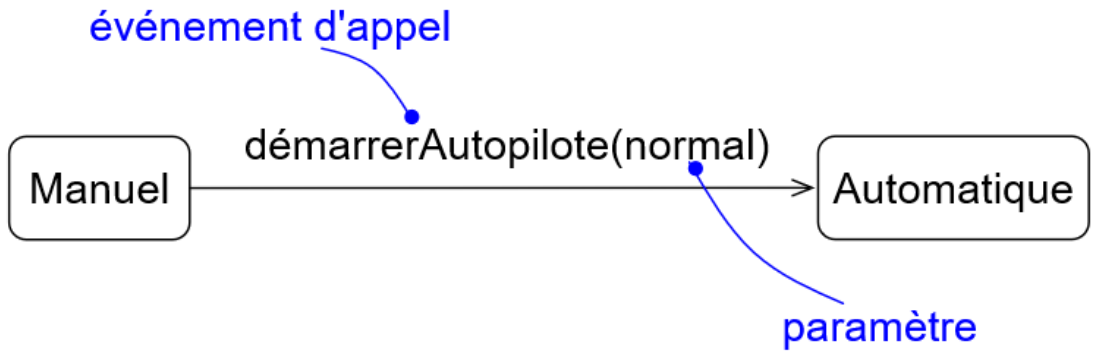
Évènement de type signal

- Est un évènement qui représente la spécification d'un stimulus asynchrone entre objet.
- La réception d'un signal représente un évènement pour l'objet destinataire,



Évènement d'appel (call)

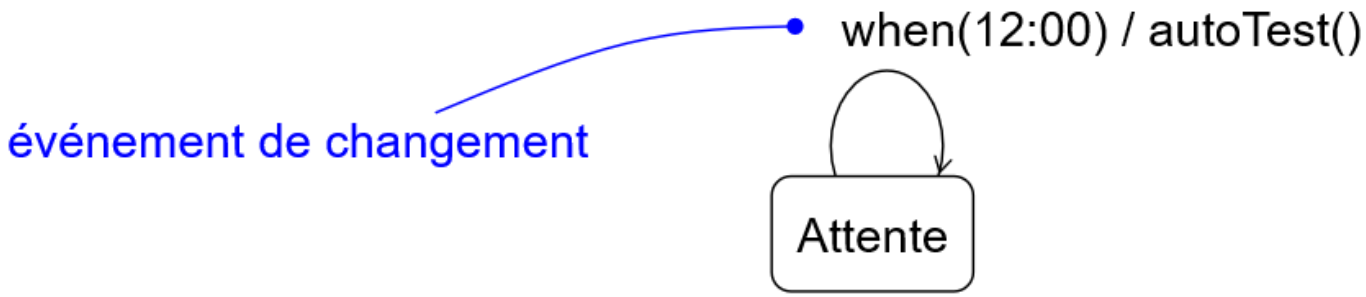
- Représente la réception d'un appel d'une opération par un objet.
- Les paramètres de l'opération sont ceux de de l'évènement d'appel.
- Les évènements d'appel sont des opérations déclarés au niveau du diagramme de classe.



Évènement de changement (change)

- Est généré par la satisfaction d'une condition booléenne sur des valeurs d'attributs.
- Il s'agit d'attendre qu'une condition soit satisfaite

when(condition) évaluée continuellement jusqu'à ce qu'elle soit vraie



Évènement de changement (change)

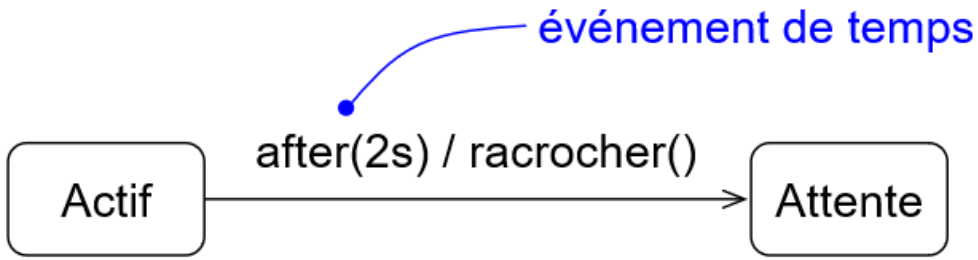
Différence entre une condition de garde et un évènement de changement

- une condition de garde est évalué une fois que l'évènement déclencheur a lieu et que le destinataire la traite. Si elle est fausse la transition ne se déclenche pas et la condition n'est pas réévalué.
- un évènement de changement est évalué continuellement jusqu'à ce qu'il devienne vrai, c'est à ce moment là que la transition se déclenche.

Évènement temporel (after ou when)

- Sont générés par le passage du temps.
- Ils sont spécifiés soit :
  - De manière absolue (date précise) **when**(date = date)
  - De manière relative(temps écoulé) **after**(durée).

Par défaut, le temps commence à s'écouler dès l'entrée dans l'état courant



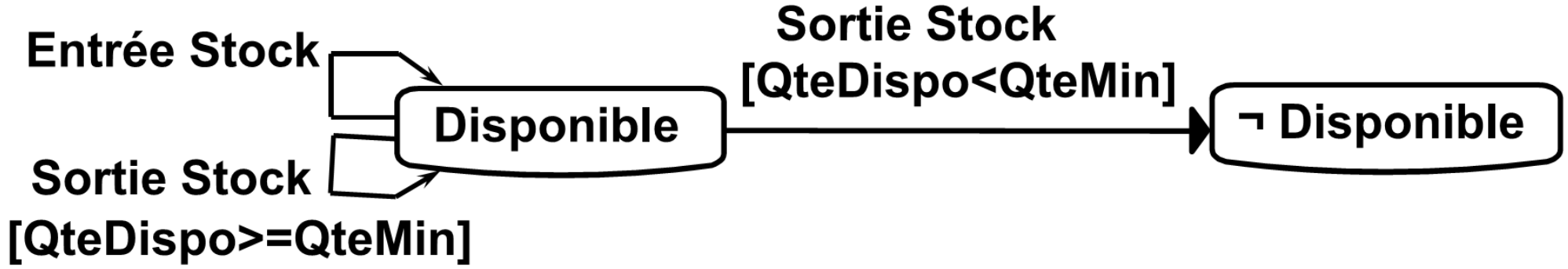
Notion de garde(1)

- Une garde (ou condition de garde) :
  Est une condition booléenne dont dépend le déclenchement d'une transition lors de l'occurrence d'un 'un événement.

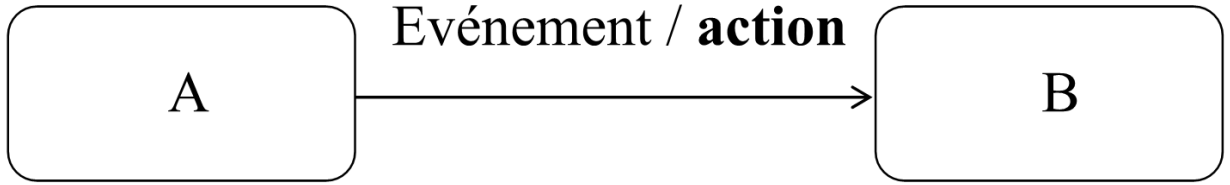
- Est évaluée dès l'arrivée de l'événement de déclenchement.

Notion de garde(2)

- Une garde est représentées par :
  - expressions booléennes, exprimées en langage naturel
  - encadrées par des crochets.



Notion d'action



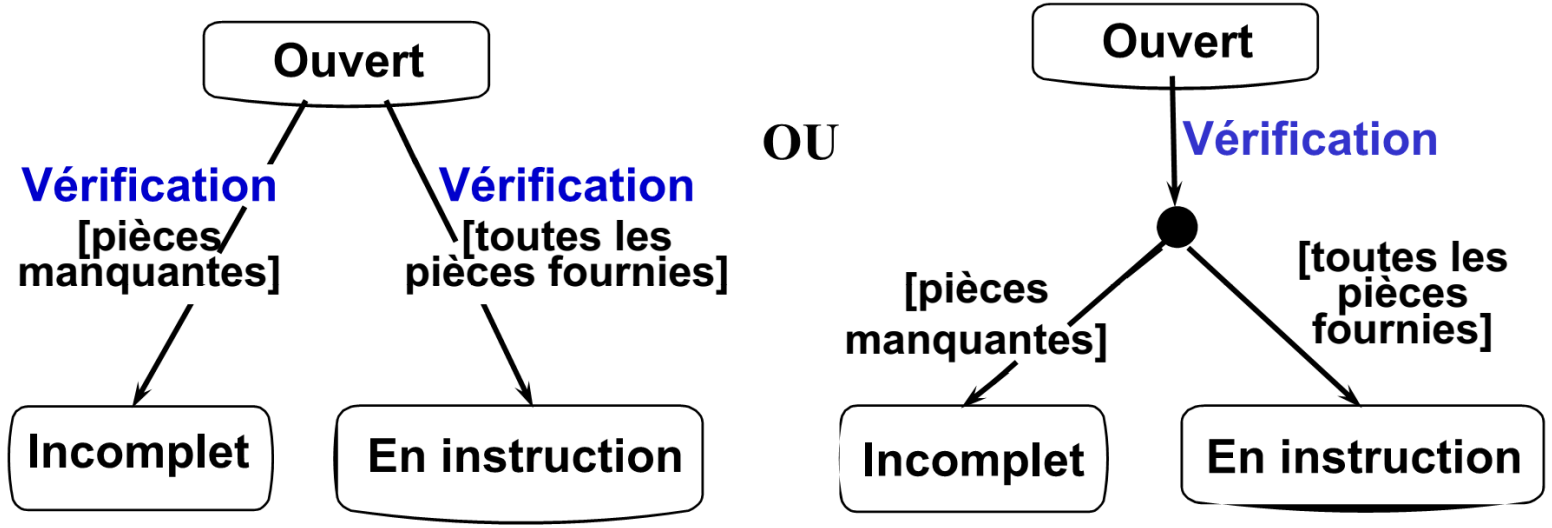
- Les actions spécifiées dans une transition sont les actions à exécuter lors du déclenchement de la transition par l'événement.
- Chaque action est instantanée et atomique, donc non interruptible.
- Une action peut comporter des appels d'opération, la création ou la destruction d'un objet, ….

Points de choix

- Il est possible de représenter des alternatives pour le franchissement d'une transition. On utilise pour cela des pseudoétats particuliers :
  - **les points de jonction** (représentés par un petit cercle plein)
  - **les points de décision** (représentés par un losange).

Points de jonction(1)

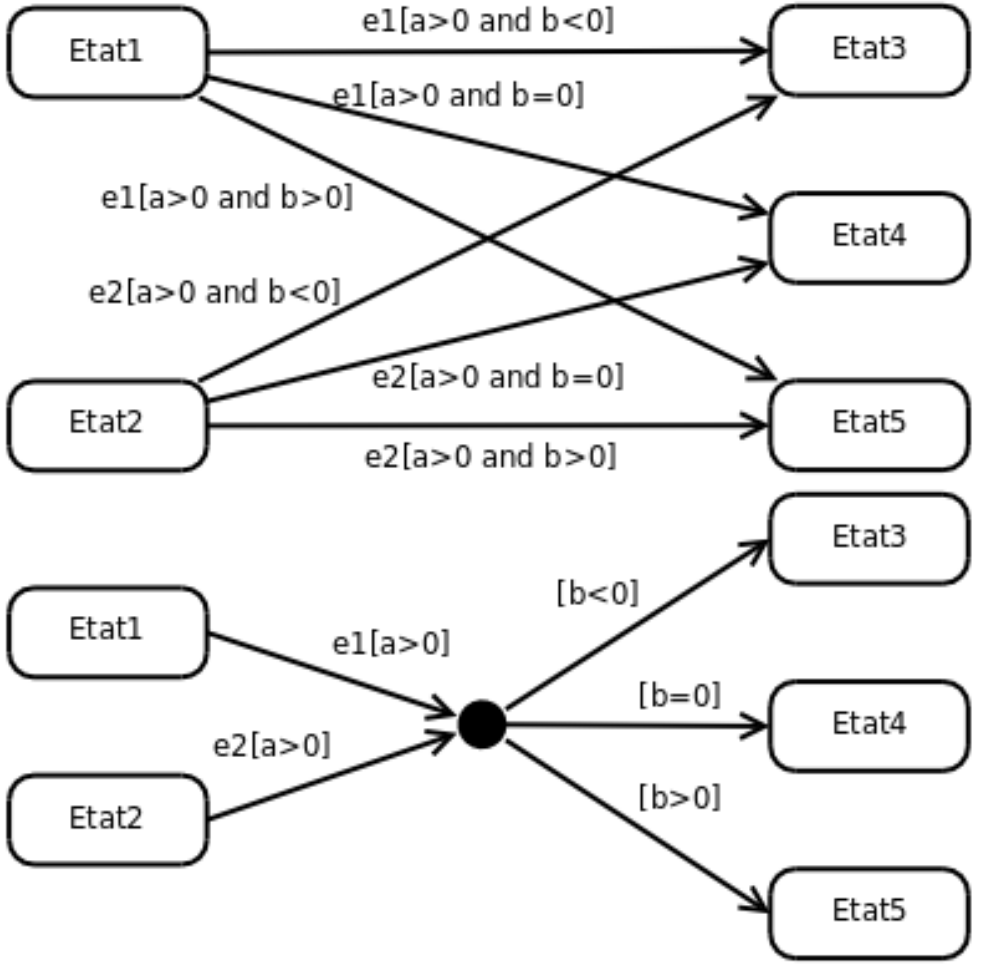
- Les points de jonction sont des artefacts graphiques qui permettent de partager des segments de transition, l'objectif étant d'aboutir à une notation plus compacte ou plus lisible des chemins alternatifs.



Point de jonction (2)

Un point de jonction peut avoir plusieurs segments de transition entrants et plusieurs segments de transition sortants.

Pour emprunter un chemin, toutes les gardes le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.



Point de decision(1)

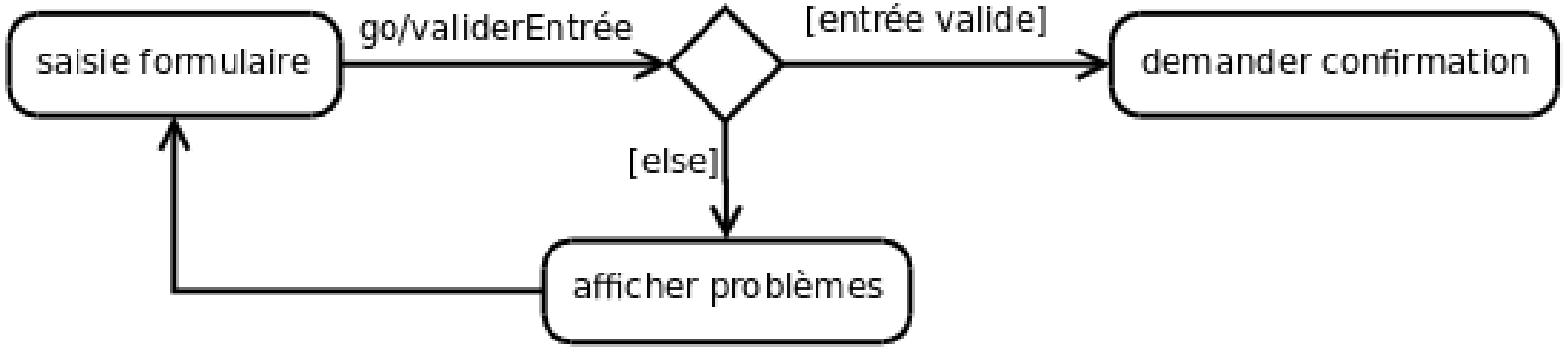
- Un point de décision possède une entrée et au moins deux sorties.
- Contrairement à un point de jonction, les gardes situées après le point de décision sont évaluées au moment où il est atteint.
  - Cela permet de baser le choix sur des résultats obtenus en franchissant le segment avant le point de choix.
- Si, quand le point de décision est atteint, aucun segment en aval n'est franchissable, c'est que le modèle est mal formé.

Point de decision(2)

- Il est possible d'utiliser une garde particulière, notée [else], sur un des segments en aval d'un point de choix.
- Ce segment n'est franchissable que si les gardes des autres segments sont toutes fausses.
- L'utilisation d'une clause [else] est recommandée après un point de décision, car elle garantit un modèle bien formé.

Point de decision(3)

Exemple



ÉTAT COMPOSITE

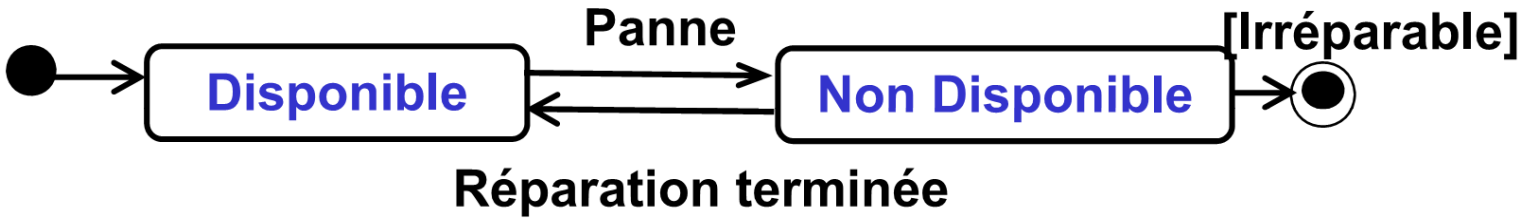
Etat composite : introduction

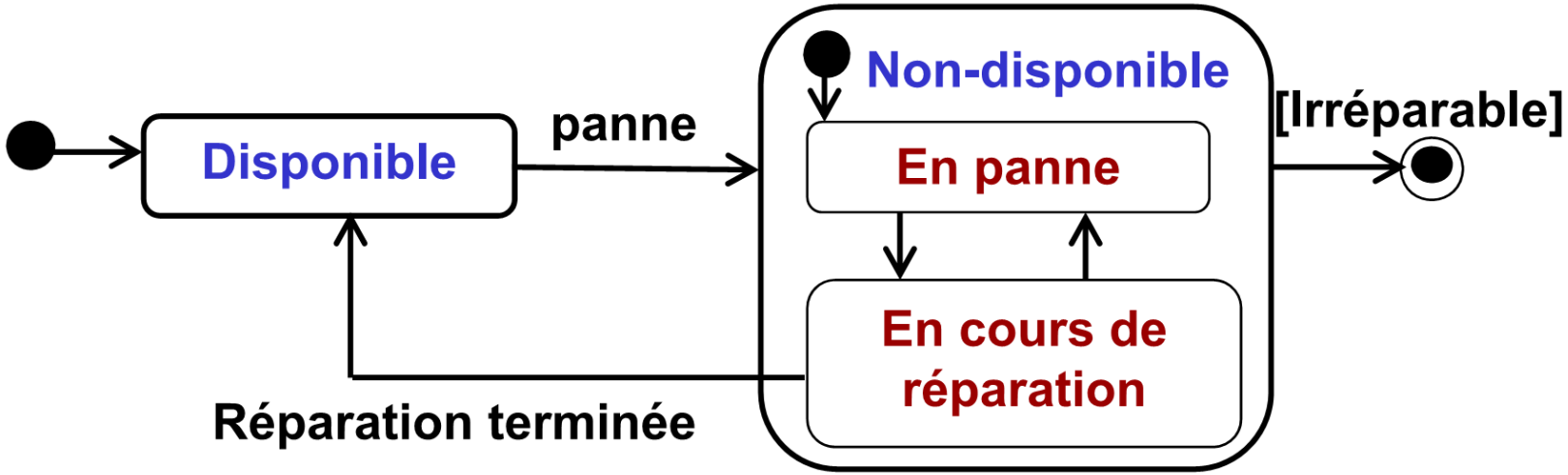
- Si le diagramme d'état transition devient trop complexe, on peut utiliser des états imbriqués pour le simplifier
- Un super-état ou état composite est un état qui englobe d'autres états appelés sous-états
- Les sous-états peuvent être emboîtés à n'importe quel niveau.
- Le nombre d'imbrication n'est pas limité (ne pas abusé sinon problème de lisibilité.

Etat composite : objectifs

- Hiérarchiser les états
- Structurer les comportements complexes
- Factoriser les actions
- Apporter plus de clarté au diagramme d'état-transition.

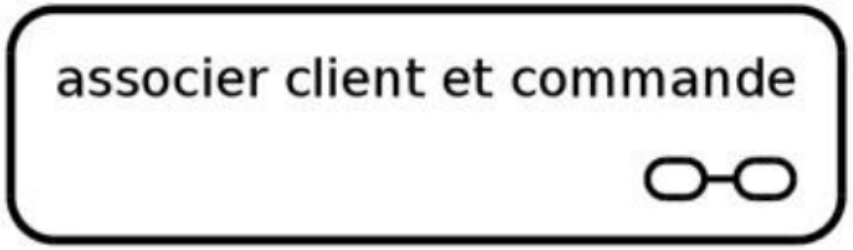
Etat composite : Exemple





Etat composite : Notation abrégée

- L'utilisation des états composites permet de définir une spécification par raffinement successifs.
- Une notation abrégée permet d'indiquer qu'un état est composite et que sa définition est donnée dans un autre diagramme.

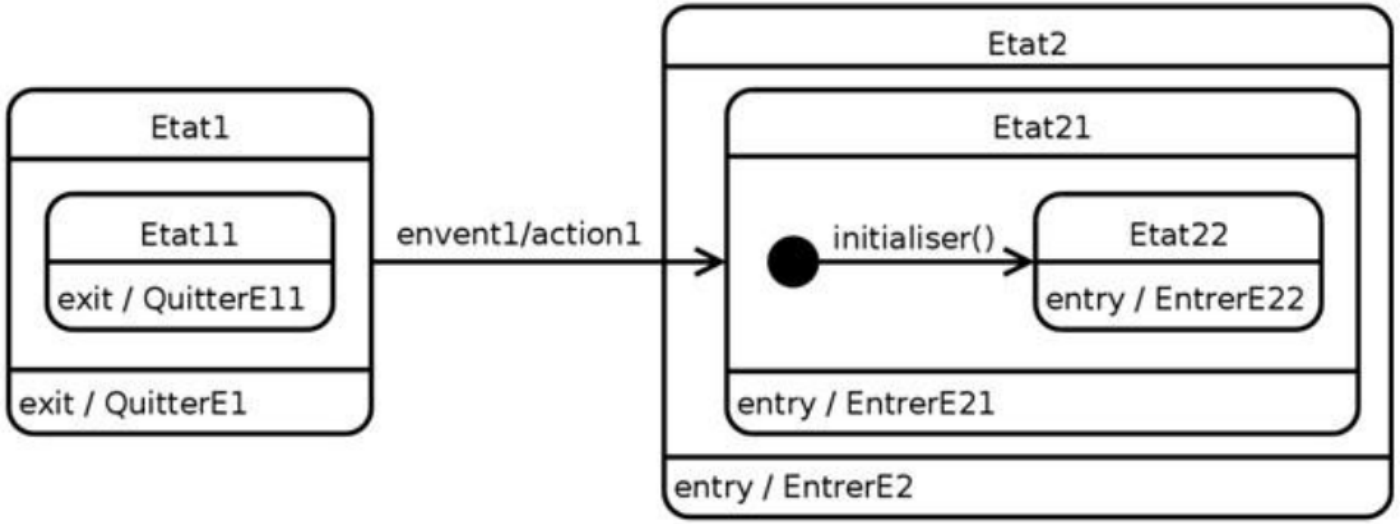


Notation abrégée d'un état composite (masquage des sous-états)

Etat composite et transition

- Les transitions peuvent avoir pour cible la frontière d'un état composite. Elle sont alors équivalentes à une transition ayant pour cible l'état initial de l'état composite.
- Une transition ayant pour source la frontière d'un état composite est équivalente à une transition qui s'applique à tout sous-état de l'état composite source.
  - Cette relation est transitive et peut traverser plusieurs niveaux d'imbrication.
- Si une transition ayant pour source la frontière d'un état composite ne porte pas de déclencheur explicite, elle est franchissable quand l'état final de l'état composite est atteint.

Etat composite et transition



- Depuis l'état Etat1, la réception de l'évènement Event1 produit la séquence d'activité : Quitter E11, Quitter E1, action1, EnterE2, Entrer21, initialiser(), Enter E22 et place le système dans l'état Etat22

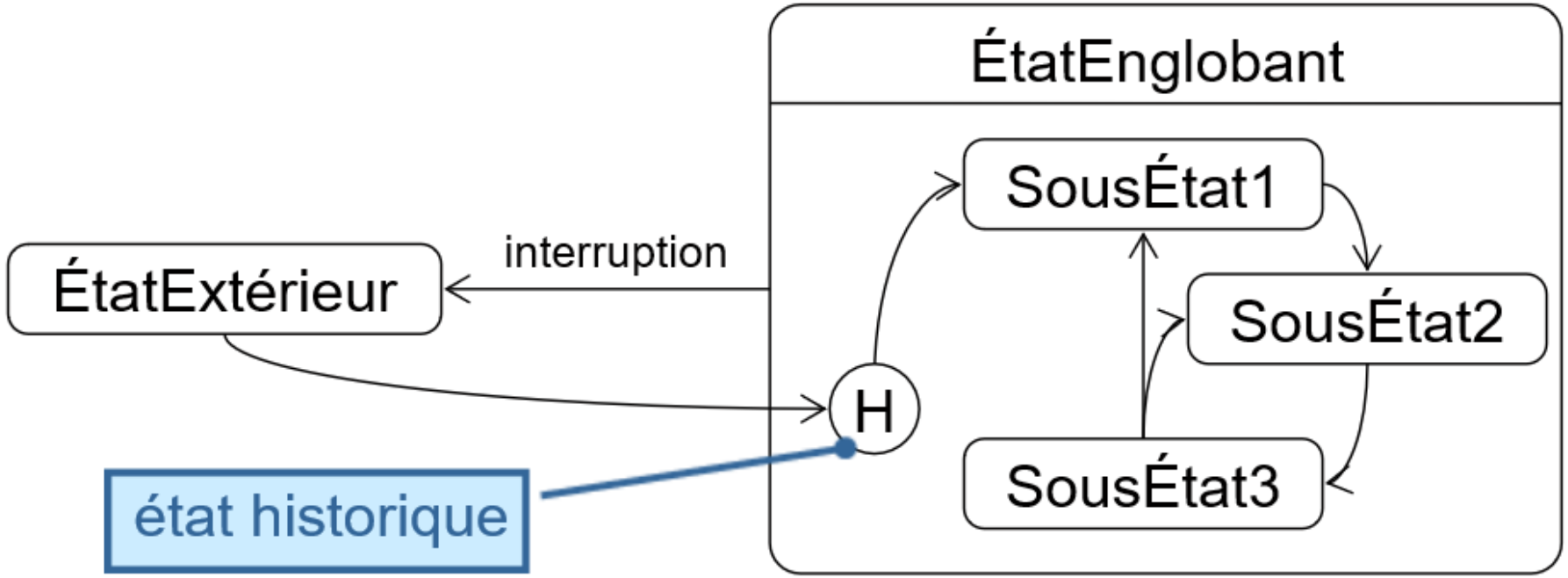
État historique(1)

- Si un sous-état d'un état composite est atteint puis abandonné prématurément, il peut être utile de revenir à l'état composite au sous état même qui était actif en dernier.
- Un état historique est donc utilisé pour réaliser ceci.

État historique(2)

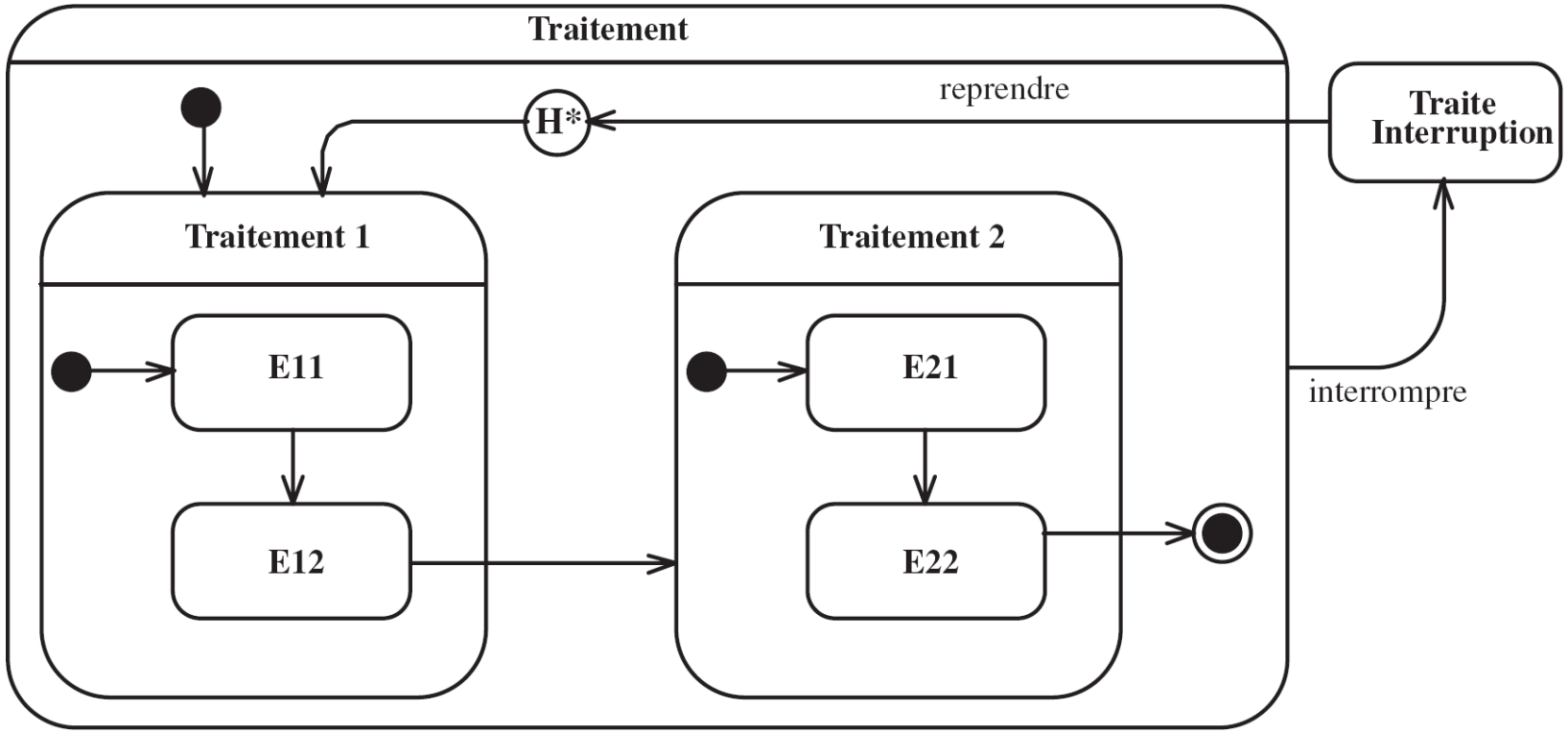
- Un état historique est un pseudo-état qui mémorise le dernier sous-état actif d'un état composite.
- Graphiquement, il est noté par un H cerclé.
- Une transition ayant pour cible l'état historique équivaux à une transition qui a pour cible le dernier état visité de l'état englobant.
- H* désigne un historique profond, cad un historique valable pour tous les niveaux d'imbrication.

État historique : exemple



- L'état historique permet de revenir au dernier sous-état visité lors du retour à un état englobant

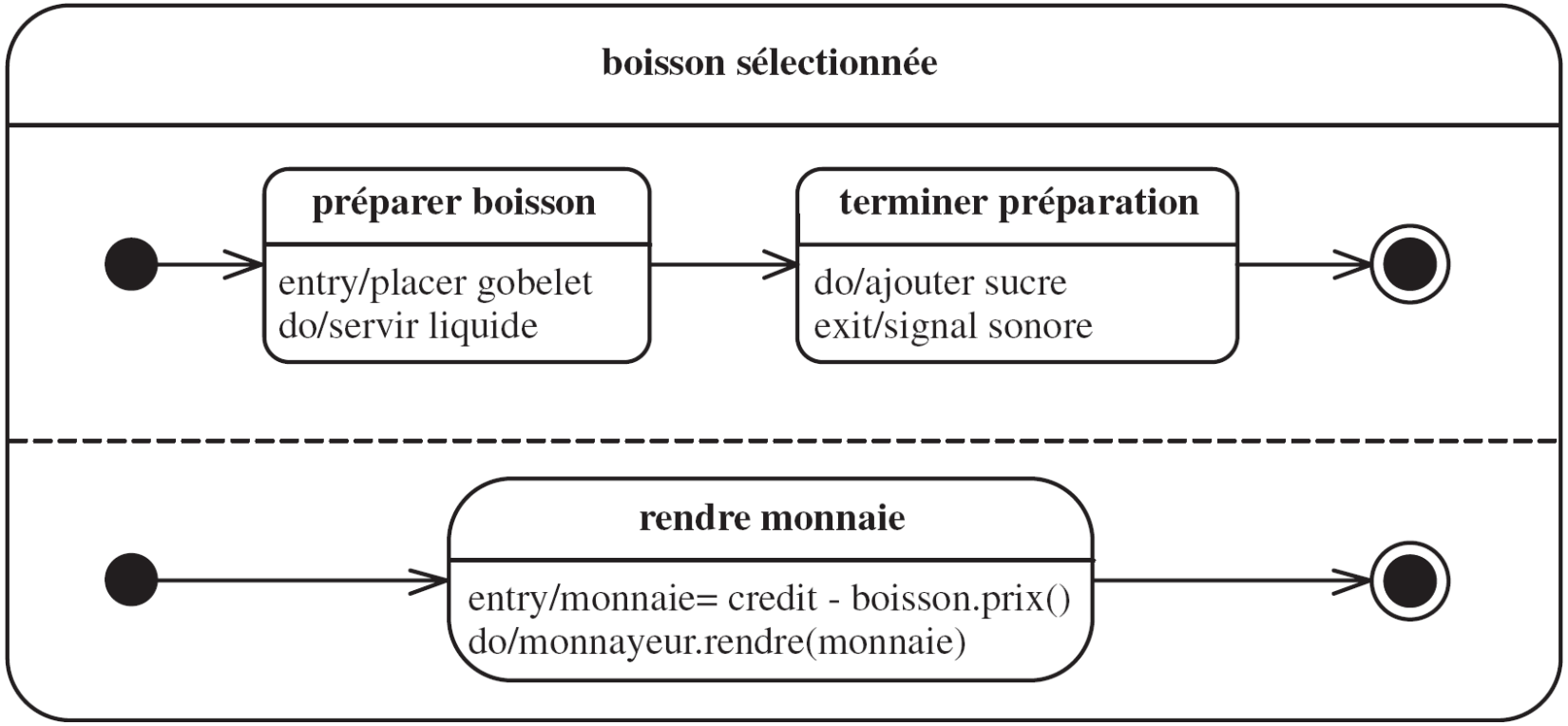
État historique profond : exemple



États concurrents

- Un état composite peut comporter plus d'une région : régions concurrentes séparées par une ligne pointillée.
- Chaque région représente un flot d'exécution : elles sont exécutées en parallèle.
- Permet de décrire des automates parallèles : plusieurs régions, chacune représente un flot d'exécution
- Un évènement peut déclencher une transition dans plusieurs sous-automates
- Sortie possible quand tous les sous-automates sont dans un état final

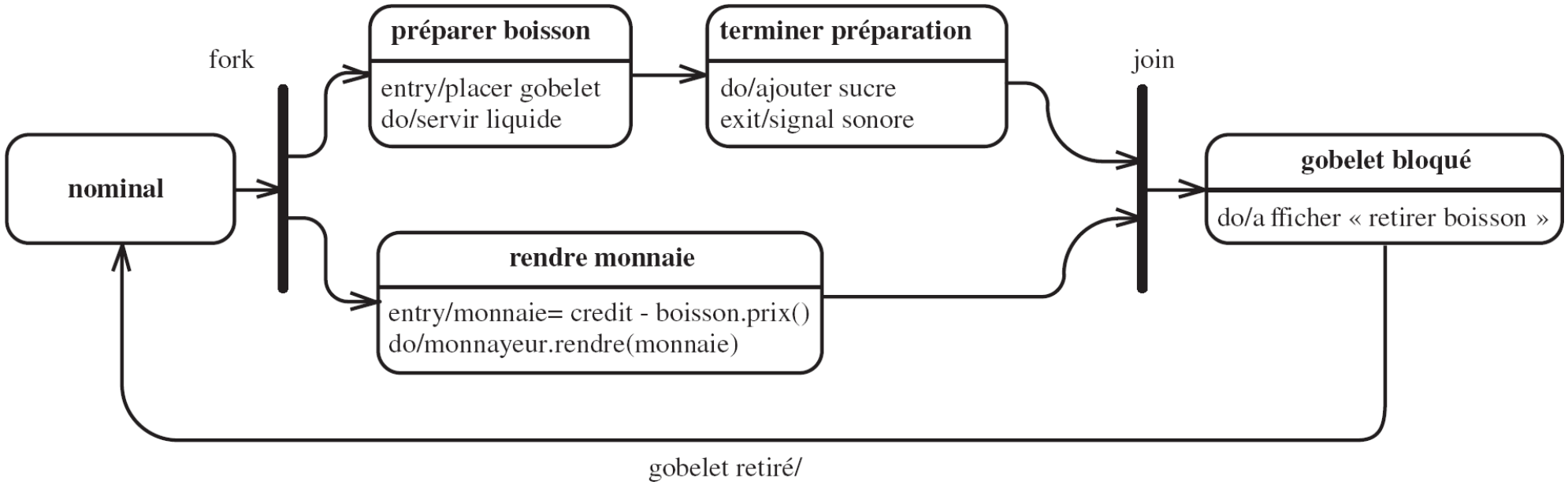
États concurrents : Exemple



Transitions concurrentes

- Une transition fork correspond à la création de deux états concurrentes.
- Une transition join correspond à une barrière de synchronisation qui supprime la concurrence.
  - Pour pouvoir continuer leur exécution, toutes les tâches concurrentes doivent préalablement être prêtes à franchir la transition.

Transition concurrente



UTILISATION DES DIAGRAMMES ÉTATS-TRANSITIONS

En phase d'analyse :

- Description de la dynamique du système vu de l'extérieur
- Synthèse des scénarios liés aux cas d'utilisation
- Événements = action des acteurs

En phase de conception :

- Description de la dynamique d'un objet particulier
- Événements = appels d'opérations

Bibliographie

- Cours “modélisation de conception orientée objet des système d’information”
  - Mohamed Amine CHAÄBANE ISAA-SFAX
  - RafikBOUAZIZ FSEG Sfax
  - FaïezGARGOURI ISIM Sfax
- http://laurent-audibert.developpez.com/Cours-UML
- Cours “Génie logiciel 2”
  - Julie Dugdale
- Cours “UML”, Polytech Paris-Sud
  - Delphine Longuet

Bibliographie

- Cours “Introduction à UML2”
  - Pierre gérard
- http://slideplayer.fr/slide/2994661/