

# Représentations des Systèmes à Evénements Discrets

## 1/ Introduction :

### 1.1/ Rappel : information logique

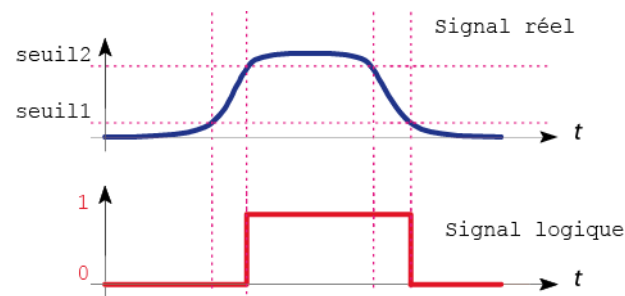
Une information logique est une information qui ne peut avoir que deux états :

- Vrai (Tout, présent, 5 V, 2 bar, etc.) → état logique 1
- Faux (Rien, absent, 0 V, 0 bar, etc.) → état logique 0

Dans la description d'échanges d'informations (avec l'environnement ou dans le système lui-même), les informations sont notées par un nom (simplifié).

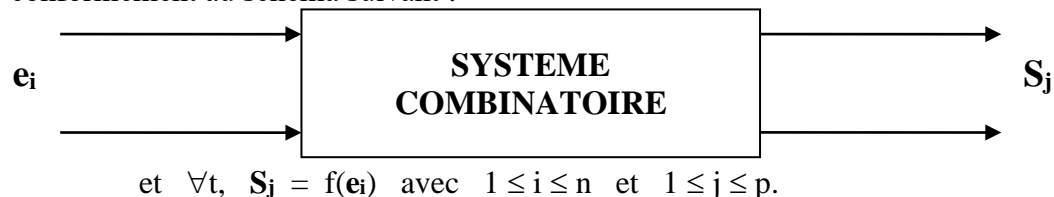
Exemple :

Contenu de l'information	Nom de l'information	Valeur



### 1.2/ Traitement d'une information logique

Un système combinatoire est une unité de traitement de l'information logique qui à une même combinaison des entrées, produira toujours les mêmes sorties. A tout instant, il peut s'exprimer conformément au schéma suivant :



Les  $e_i$  et  $S_j$  sont respectivement des *variables* et des *fonctions binaires* de ces variables (variables et fonctions ne pouvant prendre que les *deux valeurs 0 et 1* par convention).

Afin de représenter et d'étudier de tels systèmes, il est nécessaire de s'appuyer sur un outil mathématique rigoureux : *l'algèbre binaire* (algèbre de Boole).

## 2/ Diagramme de Séquence (Sequence Diagram [sd]) : Description des échanges d'information

Le diagramme décrit le comportement d'un système du point de vue des échanges d'information entre des organes. Autrement dit, il décrit les scénarii d'utilisation :

- à quel moment les acteurs sont sollicités, et quels sont les échanges d'informations ;
- montrer comment les enchaînements se succèdent.

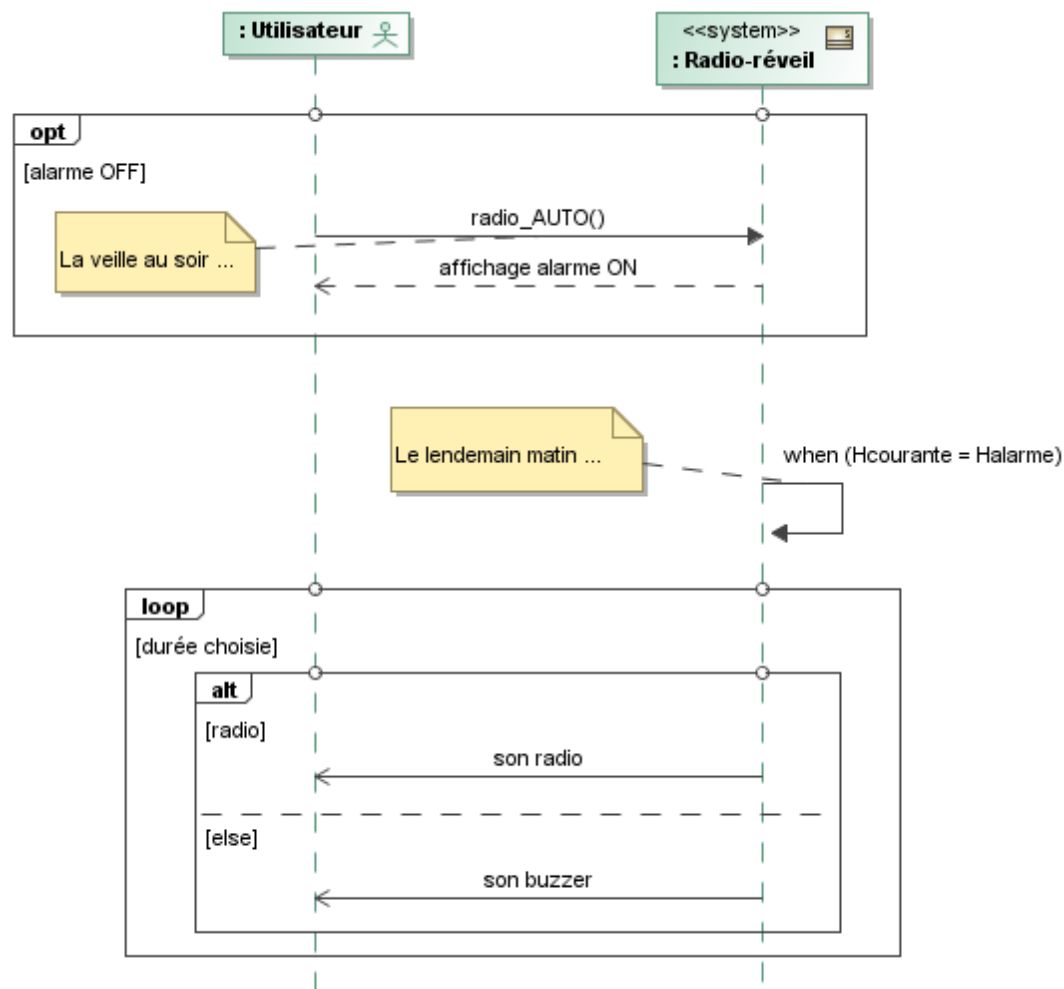
Ce diagramme répond à la question : « quels sont les échanges d'information entre les composants ou les acteur pour un cas d'utilisation ? ». Il décrit les interactions en terme d'échange de messages.

### 2.1/ Définitions

- Ligne de vie : ligne verticale pointillée faisant référence à un composant ou acteur ;
- Message : information transmise. Il y a 4 types de message noté par une variable :
  - le message asynchrone  $\rightarrow$ , pour lequel aucun retour n'est attendu ;
  - le message synchrone  $\rightarrow$  (même notation que pour un message asynchrone), pour lequel la situation de l'envoyeur reste figée en l'attente d'une réponse  $\leftarrow - -$  ;
  - le message réflexif  $\hookrightarrow$ , qui représente un message qui reste interne à l'organe.
- Activité : barre d'activation ou d'exécution = comportement en cours d'exécution
- Fragment : rectangles correspondant à une structure d'algorithme

### 2.2/ Son usage

- Lecture chronologique de haut vers le bas (pas d'échelle de temps)
- Echange d'information dans le système : diagramme de séquence
- Echange d'information du système avec ses interacteurs : diagramme de séquence « système »



### 2.3/ Structuration :

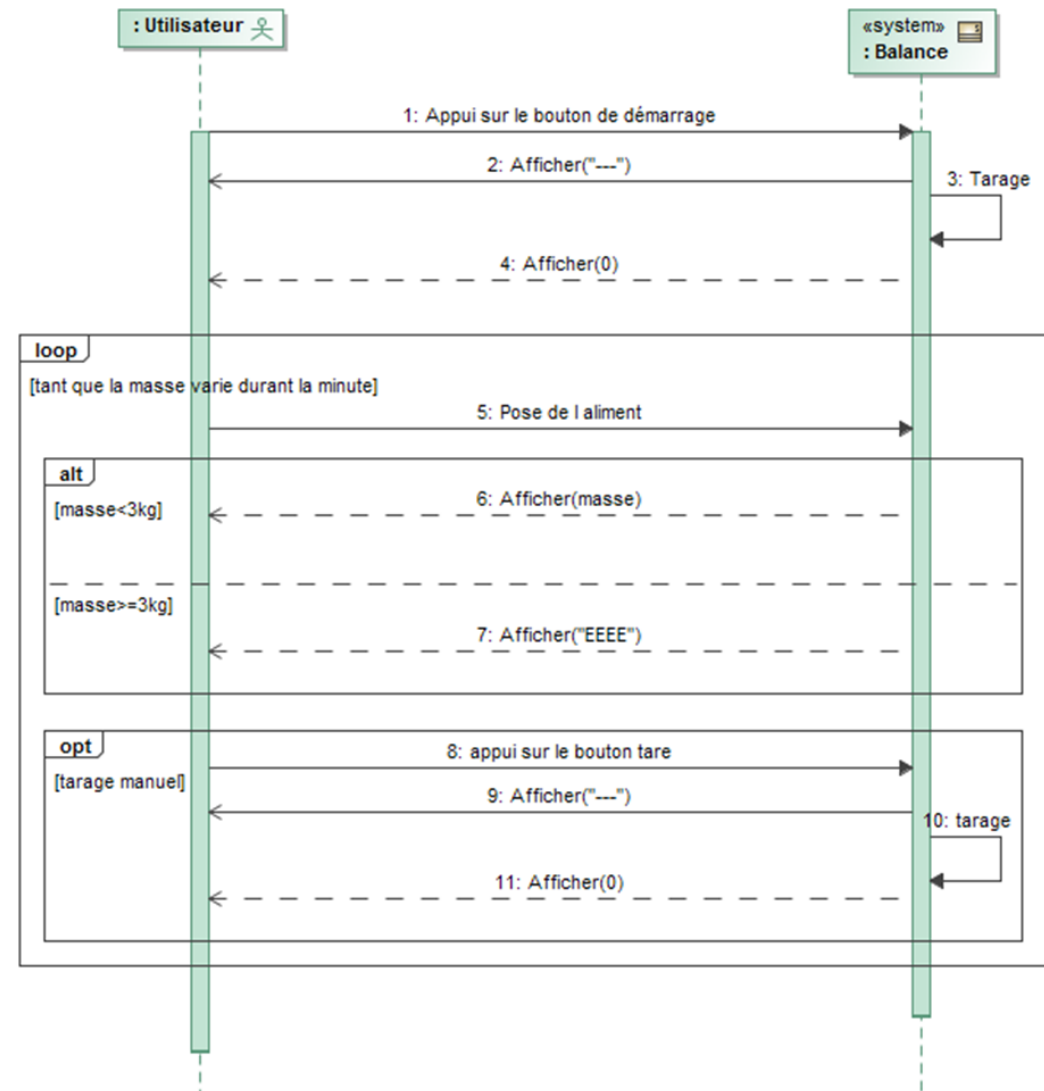
#### a) Structure algorithmique

Les variantes dans un scénario sont décrites par des éléments appelés fragments combinés.

Elles sont de type :

- Opt : (optionnel) ce qui est contenu dans le bloc est exécuté si la condition est vérifiée ;
- Alt : (alternatif) le 1<sup>er</sup> bloc est exécuté si la condition est vérifiée, sinon c'est le second bloc ;
- Loop : (boucle) répète la séquence tant que la condition est vérifiée,
- Par : (parallèle) permet d'exécuter deux blocs de séquences en parallèle,
- Ref : (référence) pour intégrer un diagramme de séquence dans un autre diagramme de séquence.

#### b) Application : balance de cuisine



### 3/ Diagramme d'états (*state machine [stm]*)

Le diagramme d'états représente le comportement d'un organe ou d'un système en termes d'états reliés entre eux par des transitions déclenchées par des événements.

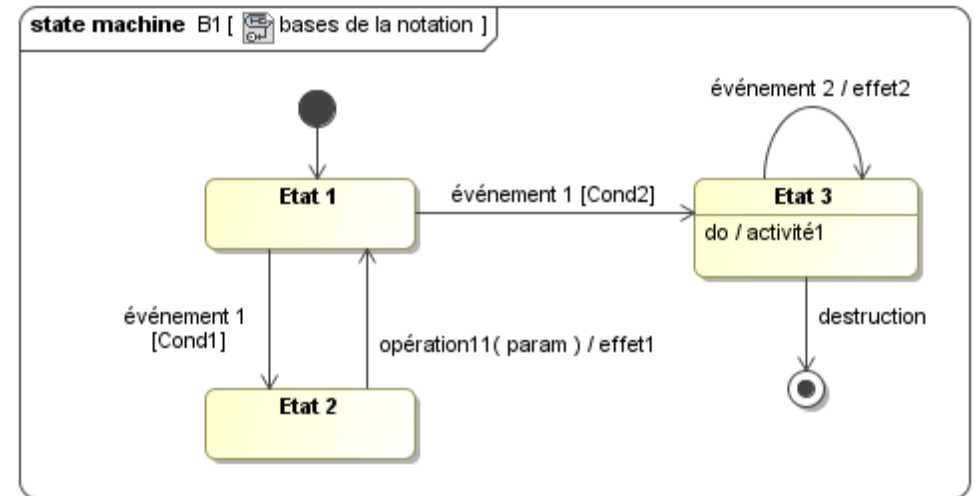
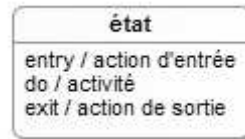
#### 3.1/ Eléments de base

##### a) Etat

Un état représente les situations communes durant la vie d'un organe ou système pendant laquelle :

- il exécute une certaine activité ;
- il satisfait une certaine condition *interne ou externe au bloc* (par exemple, une situation d'attente)

A un état, on peut principalement rattacher une activité, une action d'entrée et une action de sortie.



**Dans un diagramme d'états, le système ne peut être que dans un seul état.**

##### b) Transition

Une transition lie deux états (ou parfois un même état) : une transition indique le passage d'un état à l'autre si l'évènement nécessaire est présent et/ou si la condition est vraie.

Le passage d'un état à un autre est instantané.

Si la transition lie l'état à lui-même, elle est qualifiée de transition réflexive. Son franchissement implique la réinitialisation de l'état.

Elle est représentée par une flèche entre les deux états, complétée d'une expression descriptive du passage d'un état à l'autre, formée de la variable associée à un évènement et d'une condition booléenne, dite de garde, notée entre crochets : nom\_evènement[condition]

##### c) Etat initial et final

###### i. Etat initial



Pour indiquer l'état initial du diagramme, on utilise un symbole particulier appelé Etat initial. Il est relié au reste du diagramme par une transition. Aucun effet ne peut y être associé. Il ne peut n'y en avoir **qu'un seul** dans le diagramme.

###### ii. Etats finaux



De même, pour indiquer la fin du comportement, on peut utiliser des états finaux (il peut y en avoir plusieurs).

## d) Effets

## iii. Dans un état

Le déclenchement de comportements associés à un état se fait par mot clé :

Entry : lorsque le système atteint l'état concerné

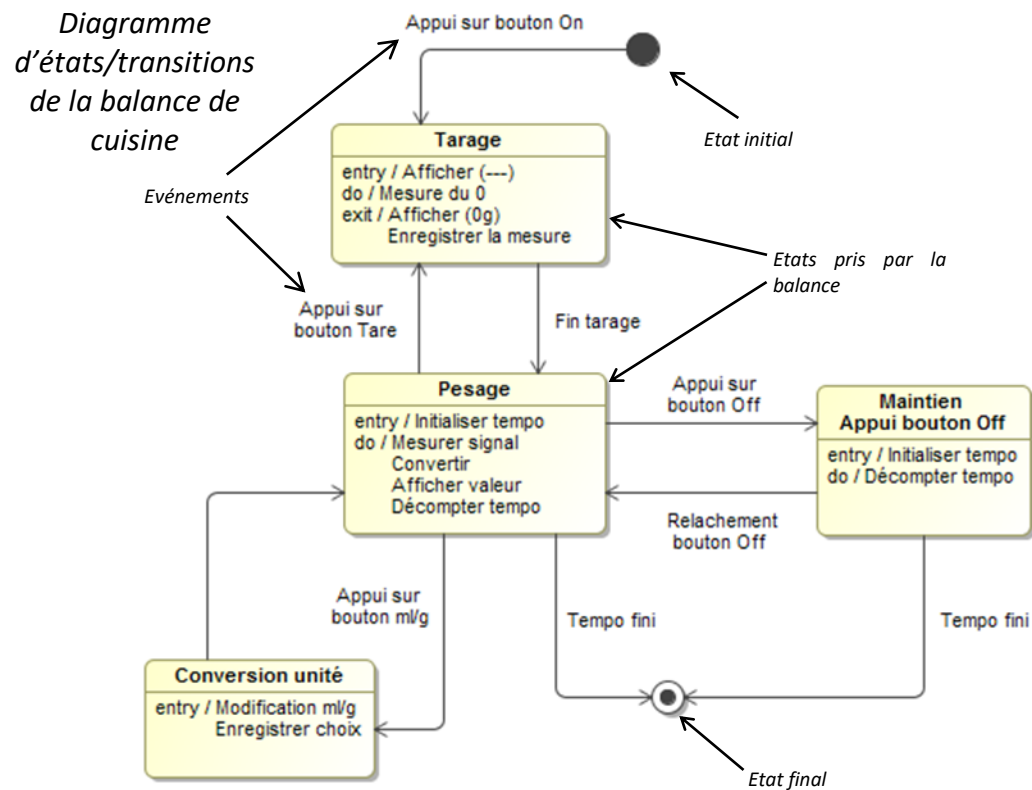
Exit : lorsque le système quitte l'état concerné

Do : pendant que le système est dans l'état concerné

<événement> : lorsque le système est dans l'état concerné et que l'événement apparaît.

## iv. Associé à une transition

Un effet peut être associé avec une transition. Il est noté à la suite de la condition de passage, séparer par un slash /.

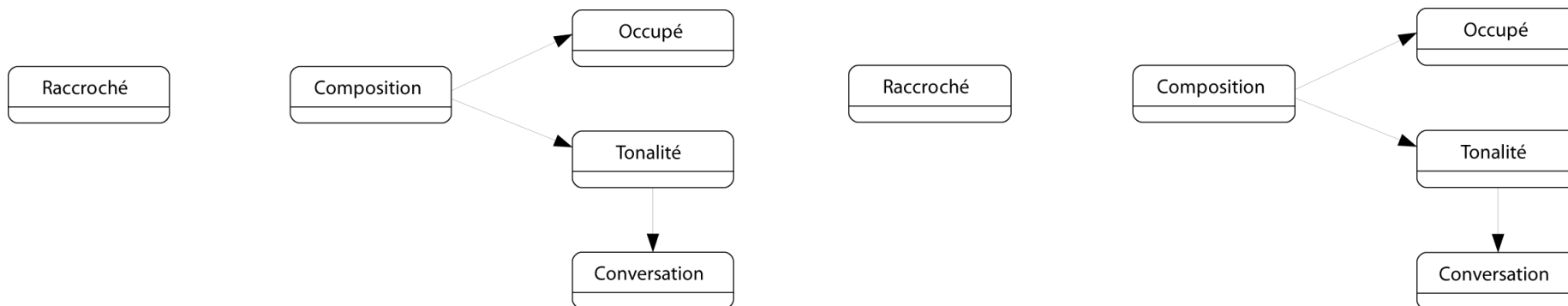


### 3.2/ Structuration hiérarchique : regroupement et raffinement

#### a) Principe

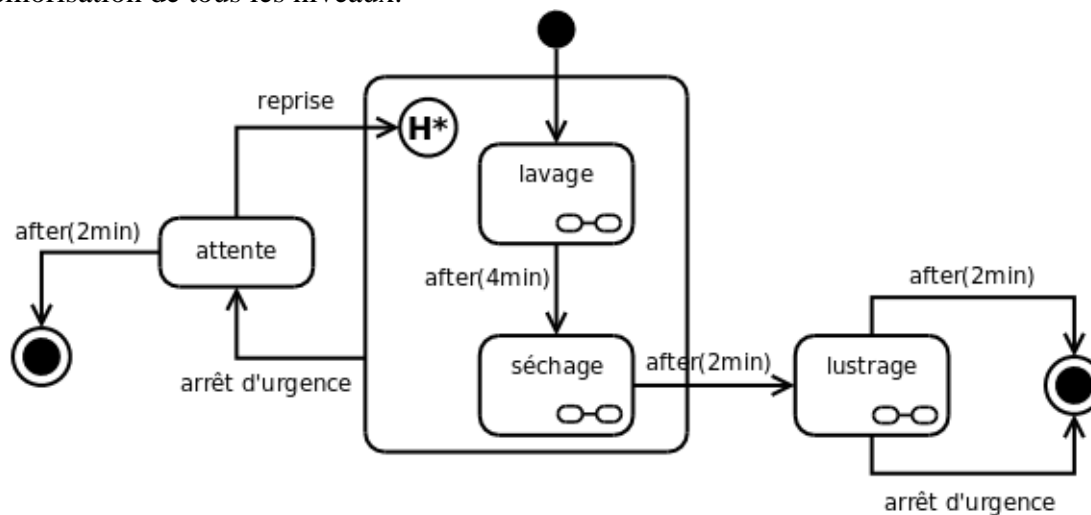
La structuration se fait en utilisant un état composite, appelé aussi appelé super-état.

L'objet n'est que dans un seul sous-état à la fois.



#### b) Usage des historiques

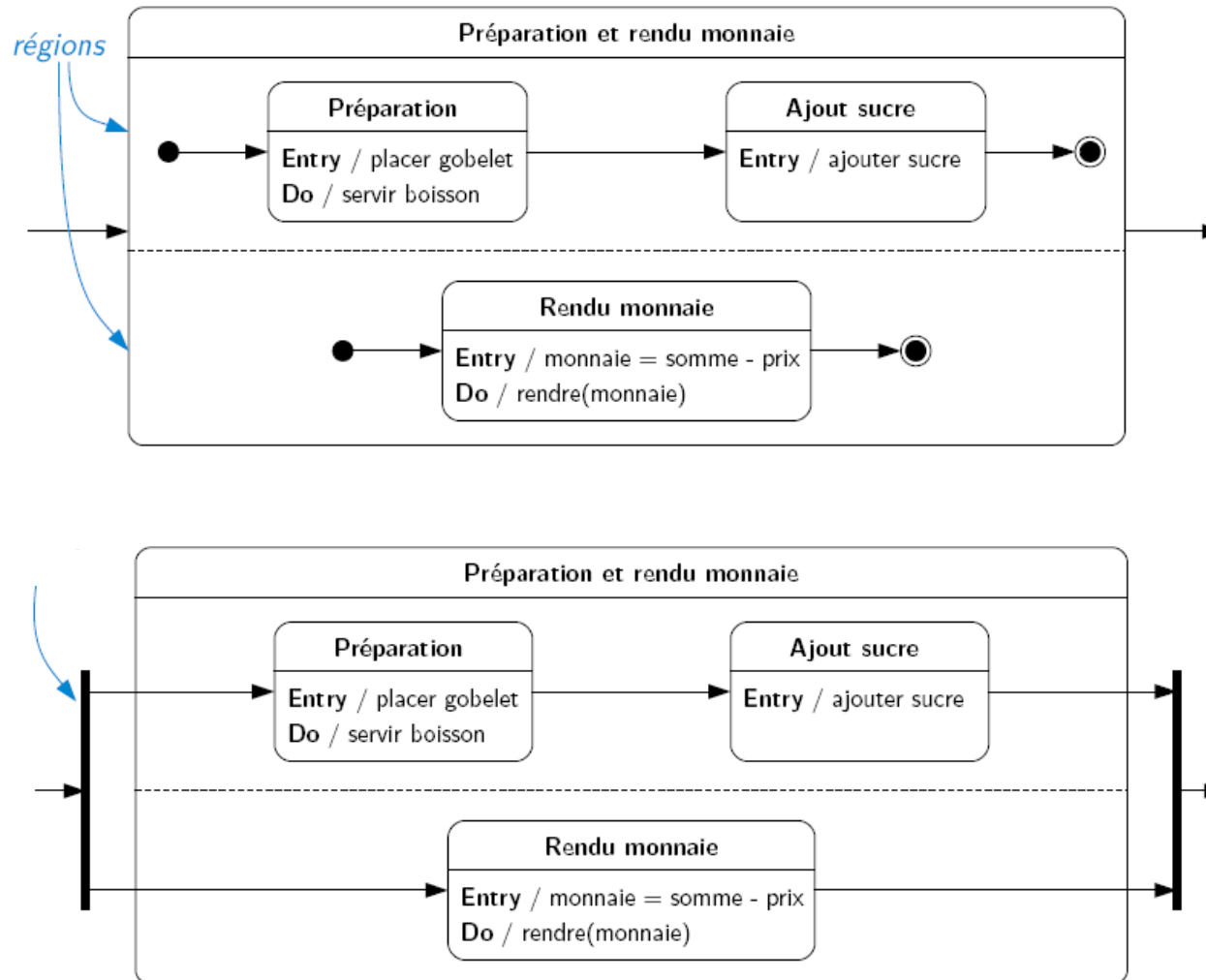
Dans certaines applications, il est nécessaire de se souvenir de l'état d'un super-état. On utilise alors le symbole :  $\textcircled{H}$  pour une mémorisation sur un seul niveau, ou  $\textcircled{H^*}$  pour une mémorisation de tous les niveaux.



a) Régions orthogonales

On qualifie de régions orthogonales (séparée par un trait pointillé) deux régions d'un état composite qui décrivent des sous-espaces indépendants du comportement du système.

Chaque région doit avoir un et un seul état actif.



### 3.3/ Compléments

#### a) Transition sur condition numérique

- when

L'évènement est émis dès qu'une expression algébrique passe de faux à vrai suite à un changement de valeurs d'un ou plusieurs attributs

- after

L'évènement associé est émis à l'instant retardé de celui après que le système soit entré dans l'état concerné.

- at

L'évènement est émis lorsque l'instant voulu a été atteint

#### b) Chemin de transitions

- *Choise* : mélange
- *Jonction* : décision

