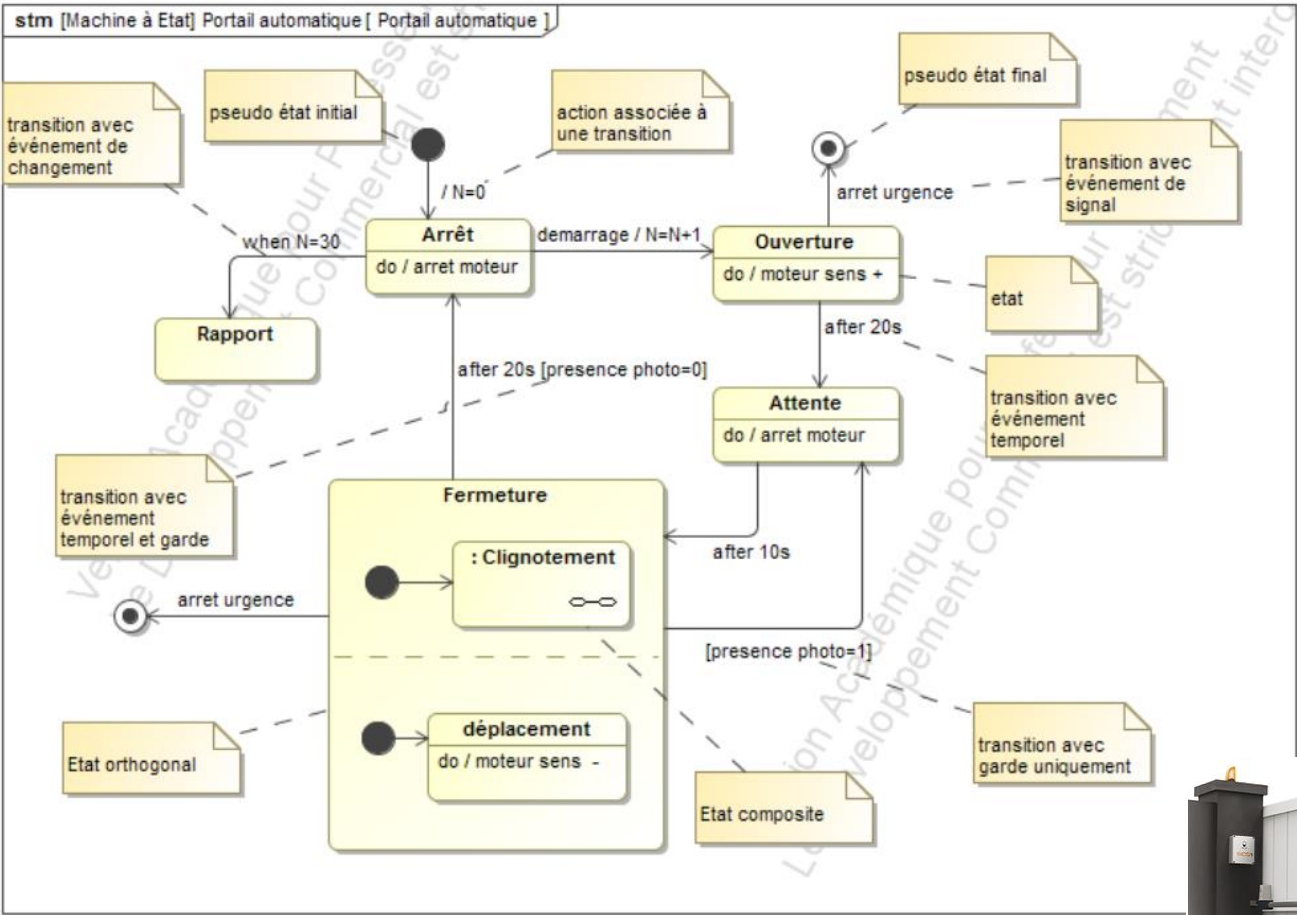


# Systemes à évènements discrets



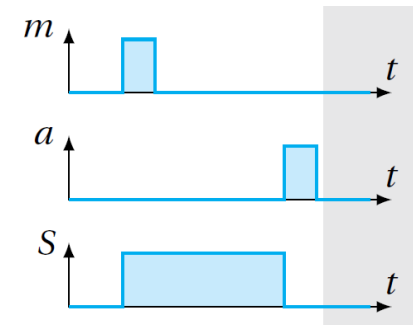
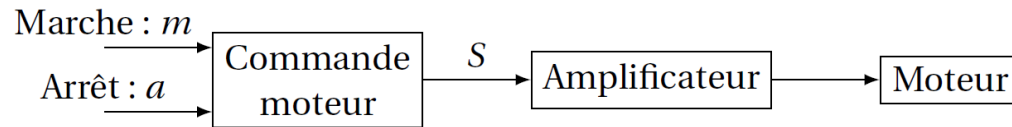
# Systemes à évènements discrets

On appelle **systeme à évènements discrets** un systeme réel qui évolue **par étapes**.

La plupart des organes de commande de ces systemes se **programment en langage C**. Les outils décrits à travers la norme **SysML** permettent de modéliser la structure du programme **indépendamment de tout langage** de programmation.

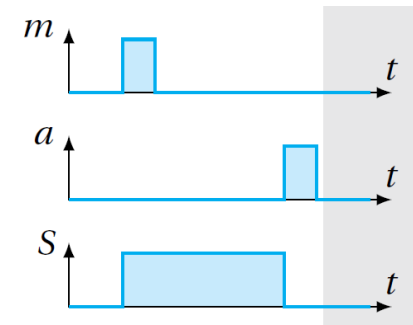
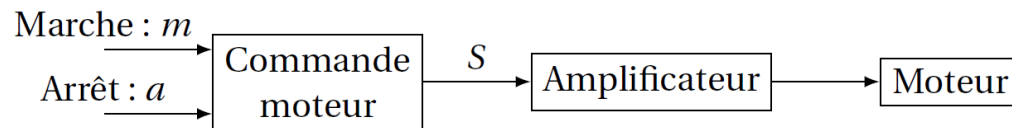
# Systemes à évènements discrets

- **Le chronogramme** : évolution temporelle de variables en fonction du temps. Ces variables prennent en général **2 états** (vrai / faux, 1 ou 0)



# Systemes à évènements discrets

- **Le chronogramme** : évolution temporelle de variables en fonction du temps. Ces variables prennent en général **2 états (vrai / faux, 1 ou 0)**



- **Le diagramme d'état** : états d'un système et évolutions entre ces états suivant les modes de fonctionnement.
- **Le diagramme d'activités** : comportement et processus se déroulant dans un état donné.

- 1. Diagramme d'état**
- 2. Diagramme d'activité en SysML**

# **1. Diagramme d'état**

2. Diagramme d'activité en SysML

# 1. Diagramme d'état

## Définition des états



### **Notion d'état :**

Un état *modélise* une phase du fonctionnement d'un système : il n'y a, par définition, **qu'un seul état actif à chaque instant dans une machine d'état en fonctionnement.**

# 1. Diagramme d'état

## Définition des états



### Notion d'état :

Un état *modélise* une phase du fonctionnement d'un système : il n'y a, par définition, **qu'un seul état actif à chaque instant dans une machine d'état en fonctionnement.**

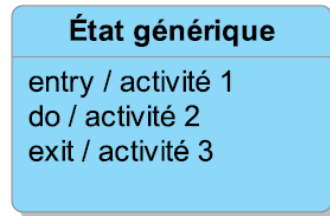
Un *diagramme d'état* du langage SysML donne une représentation graphique des transitions possibles entre les différents états dans une *machine d'état* ***stm***



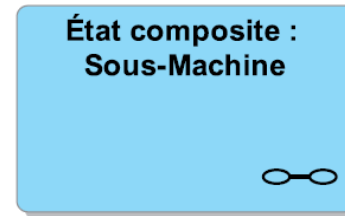
# 1. Diagramme d'état



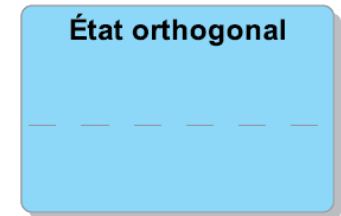
(a) Pseudos-états initial et final.



(b) État générique et comportements.



(c) État composite (sous-machine).



(d) État orthogonal.

## Pseudos-états de départ et de fin (figure a)

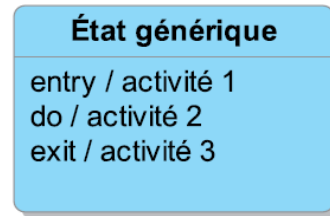
Les « pseudos-états » initial et final indiquent **le début et la fin du fonctionnement** de la machine d'état associée au diagramme d'état.

Il y a toujours **un pseudo-état initial** mais il peut ne pas y avoir de pseudo-état final.

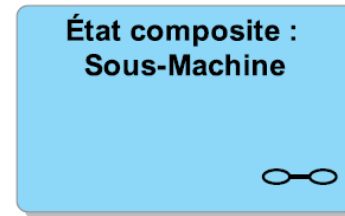
# 1. Diagramme d'état



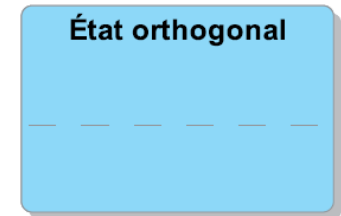
(a) Pseudos-états initial et final.



(b) État générique et comportements.



(c) État composite (sous-machine).



(d) État orthogonal.

## Comportements associés à un état (figure b)

Des comportements, déclenchés grâce à des événements internes

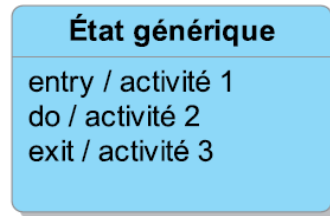
3 types de comportements qui ne sont utilisés **qu'une fois par état**, dans l'ordre suivant : **entry, do et exit**.

**sortie de l'état** : l'activité liée au comportement **do est interrompue** et celle associée au comportement **exit est alors exécutée** (actions liées à la désactivation de l'état : éteindre une LED d'activité par exemple).

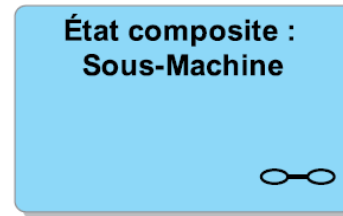
# 1. Diagramme d'état



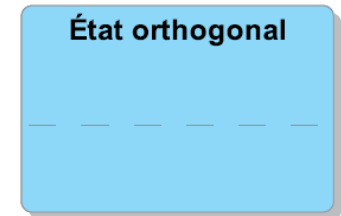
(a) Pseudos-états initial et final.



(b) État générique et comportements.



(c) État composite (sous-machine).



(d) État orthogonal.

## État composite (figure c)

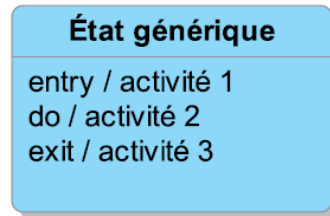
Evolutions internes d'un état avec **un autre diagramme d'état**.

Cette structure de « **sous-machine d'état** » facilite la lecture du diagramme en entrant dans le détail des évolutions internes.

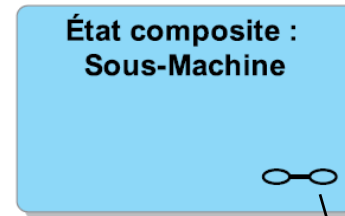
# 1. Diagramme d'état



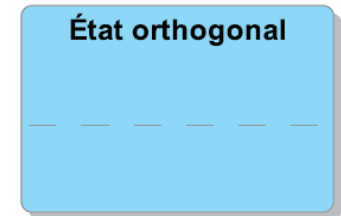
(a) Pseudos-états initial et final.



(b) État générique et comportements.

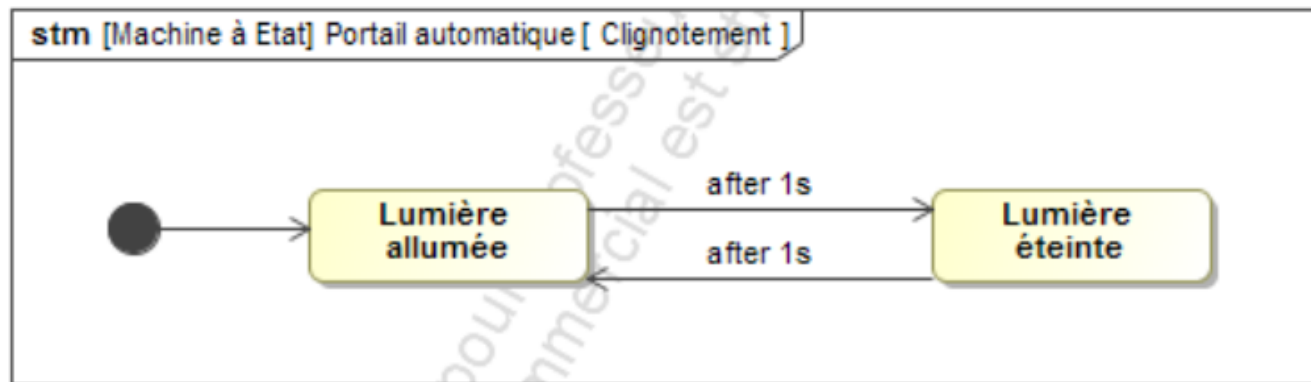


(c) État composite (sous-machine).



(d) État orthogonal.

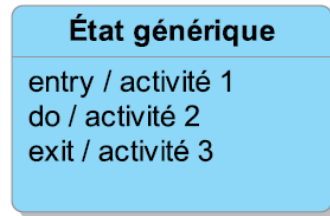
## État composite (figure c)



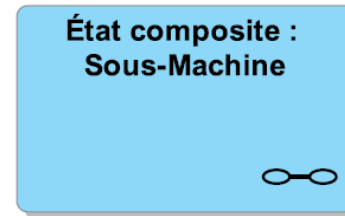
# 1. Diagramme d'état



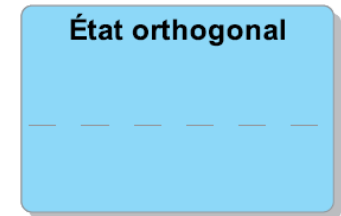
(a) Pseudos-états initial et final.



(b) État générique et comportements.



(c) État composite (sous-machine).



(d) État orthogonal.

## État orthogonal (figure d)

Un état orthogonal possède **plusieurs régions**, séparées par des pointillés, **chacune ayant sa propre description d'état**.

Les différentes régions d'un état orthogonal **fonctionnent toutes en parallèle sans aucune influence** les unes sur les autres.

# 1. Diagramme d'état



## Notion de transition :

Le changement d'état se produit par ***franchissement d'une transition.***

La transition est un **lien orienté** entre deux états.

Elle est représentée par une flèche reliant l'état de départ, appelé *état source*, et l'état d'arrivée, appelé *état cible*.

# 1. Diagramme d'état

La condition de franchissement et l'éventuel **effet qui y est associé** sont indiqués à côté de la flèche en suivant la notation événement **[garde] (*attention aux crochets*)**

# 1. Diagramme d'état

La condition de franchissement et l'éventuel **effet qui y est associé** sont indiqués à côté de la flèche en suivant la notation événement **[garde] (attention aux crochets)**



## Remarque :

Une transition sans condition est dite « **automatique** »

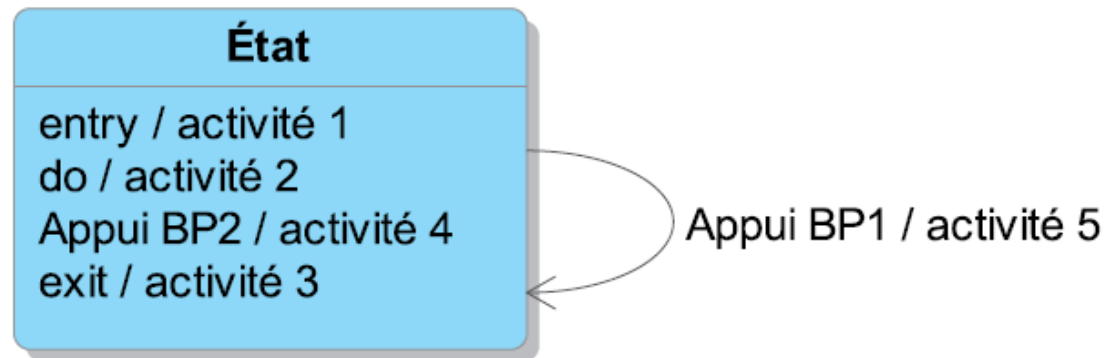
Dans ce cas, la condition prise en compte pour la transition est la **fin de l'activité associée au comportement *do* de l'état amont (ou *entry*)**.



# 1. Diagramme d'état

Le cas d'une transition bouclant sur un état, appelée ***transition réflexive***.

Cette structure permet **d'exécuter à nouveau** les activités associées aux comportements d'entrée (***entry***) et de sortie (***exit***) de l'état.



# 1. Diagramme d'état



## Notion d'évènement :

Un événement correspond à **l'occurrence du changement d'état** d'un élément du modèle ou de la survenue d'un changement extérieur.

Par définition, un événement :

- **est localisé dans l'espace et dans le temps ;**
- **n'est jamais mémorisé.**

À chaque transition est associé un événement unique.

# 1. Diagramme d'état

**L'événement de signal** prend en compte la **réception d'un signal asynchrone** qui peut arriver à n'importe quel moment : *Appui sur un bouton de l'interface homme-machine ou l'arrivée en fin de course d'un mécanisme.*

# 1. Diagramme d'état

**L'événement de signal** prend en compte la **réception d'un signal asynchrone** qui peut arriver à n'importe quel moment : *Appui sur un bouton de l'interface homme-machine ou l'arrivée en fin de course d'un mécanisme.*

**L'événement de changement** prend en compte **le changement d'une valeur interne du modèle** : *compteur qui délivre l'information  $\text{when}(N=30)$  quand 30 appuis ont été effectués.*

# 1. Diagramme d'état

**L'événement de signal** prend en compte la **réception d'un signal asynchrone** qui peut arriver à n'importe quel moment : *Appui sur un bouton de l'interface homme-machine ou l'arrivée en fin de course d'un mécanisme.*

**L'événement de changement** prend en compte le **changement d'une valeur interne du modèle** : *compteur qui délivre l'information  $\text{when}(N=30)$  quand 30 appuis ont été effectués.*

**L'événement temporel** peut être *relatif* ou *absolu* :

**Relatif** :  $\text{after}(T)$  se déclenche après le temps  $T$  passé dans l'état amont.

**Absolu** :  $\text{at}(H)$  se déclenche à la date  $H$  dans un référentiel de temps dont l'origine correspond généralement au démarrage de la machine d'état.

# 1. Diagramme d'état



## Notion de garde :

La garde est une **condition booléenne associée** au franchissement d'une transition.

À l'occurrence de l'événement, **la valeur vraie de la garde est une condition supplémentaire** à l'événement pour que la transition soit franchie.

# 1. Diagramme d'état



## Notion de garde :

La garde est une **condition booléenne associée** au franchissement d'une transition.

Un événement est obligatoirement associé à une transition, alors que **la garde est optionnelle**.

Si la condition de garde n'existe pas, elle est considérée comme toujours vraie.

# 1. Diagramme d'état



## Notion de garde :

La garde est une **condition booléenne associée** au franchissement d'une transition.

### **Exemple :**

*Lors de l'appui sur un bouton :*

-



# 1. Diagramme d'état



## Notion de garde :

La garde est une **condition booléenne associée** au franchissement d'une transition.

### **Exemple :**

*Lors de l'appui sur un bouton :*

- **L'événement modélise** le fait de savoir s'il vient d'être **enfoncé** (changement).
- La garde modélise le fait de savoir s'il est enfoncé (persistance).

# 1. Diagramme d'état



## Notion de garde :

La garde est une **condition booléenne associée** au franchissement d'une transition.

### **Exemple :**

*Lors de l'appui sur un bouton :*

- **L'événement modélise** le fait de savoir s'il vient d'être **enfoncé** (changement).
- **La garde modélise le fait de savoir** s'il est enfoncé (persistance).

# 1. Diagramme d'état



## Notion d'effet :

Un effet est **un comportement** (action, envoi d'un message, mémorisation, etc.) **accompli lorsque la transition est franchie.**

# 1. Diagramme d'état

## Règles d'évolution :

- **point de départ** indiqué par **le pseudo-état initial**

# 1. Diagramme d'état

## Règles d'évolution :

- **point de départ** indiqué par **le pseudo-état initial**
- **transitions** franchies en fonction des **événements et des gardes** : la suite des états actifs est donc fonction des événements qui se produisent.

# 1. Diagramme d'état

## Règles d'évolution :

- **point de départ** indiqué par le **pseudo-état initial**
- **transitions** franchies en fonction des **événements et des gardes** : la suite des états actifs est donc fonction des événements qui se produisent.
- **Un seul état peut être actif à la fois dans chaque machine ou sous-machine d'état.**  
Dans les états orthogonaux (machine d'état indépendante) **chaque région a un état actif** : il existe donc plusieurs états actifs situés dans des machines d'état **distinctes**.

1. Numérisation

**2. Diagramme d'activité en SysML**

## 2. Diagramme d'activité en SysML



Malgré la proximité de syntaxe entre le diagramme d'état et le diagramme d'activités, **la sémantique associée à leurs éléments graphiques est différente**



## 2. Diagramme d'activité en SysML



Malgré la proximité de syntaxe entre le diagramme d'état et le diagramme d'activités, **la sémantique associée à leurs éléments graphiques est différente**

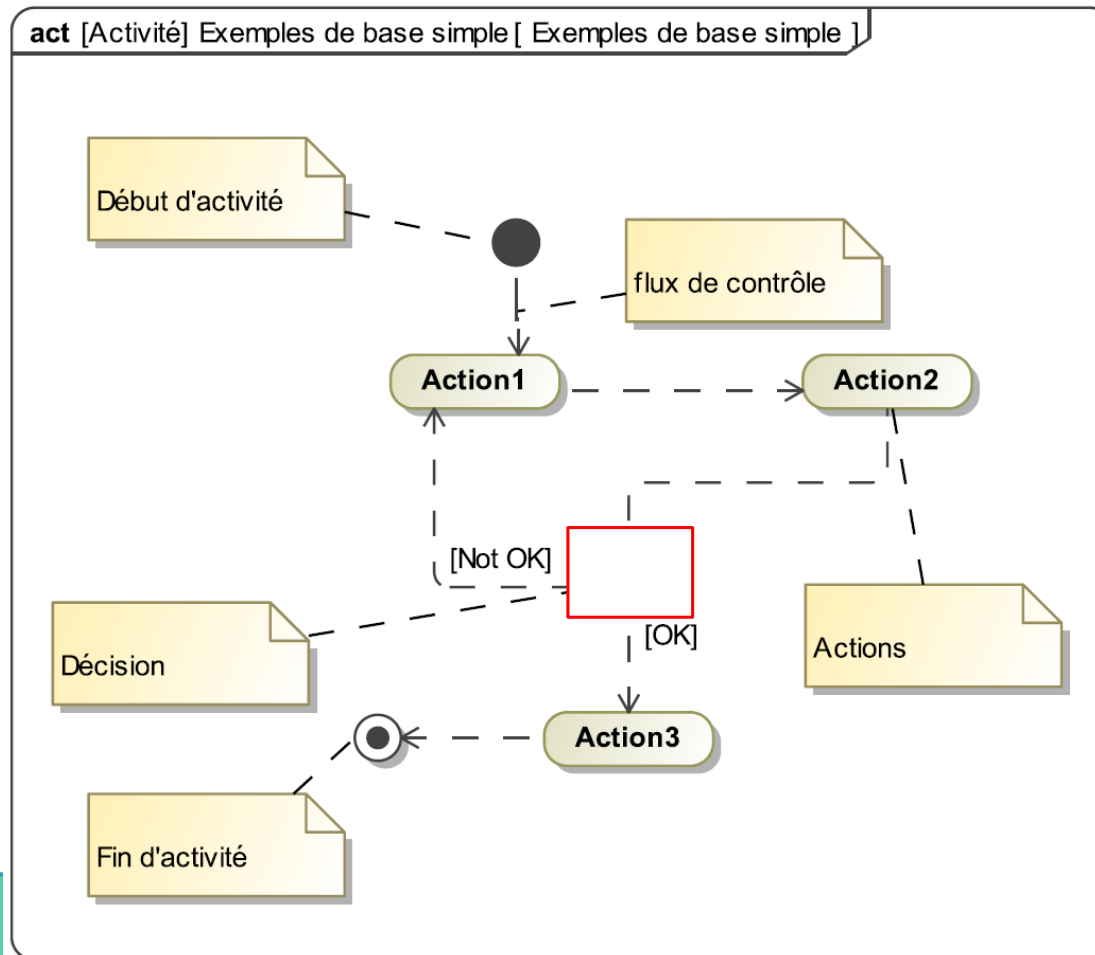
### **Activité :**

Une activité correspond à **une suite d'actions élémentaires** : elle peut notamment être enclenchée par un comportement interne à un état.

Pour simplifier, le diagramme d'activité peut être assimilé à un **algorigramme** : il décrit un déroulement d'actions qui, une fois engagé, va à son terme.

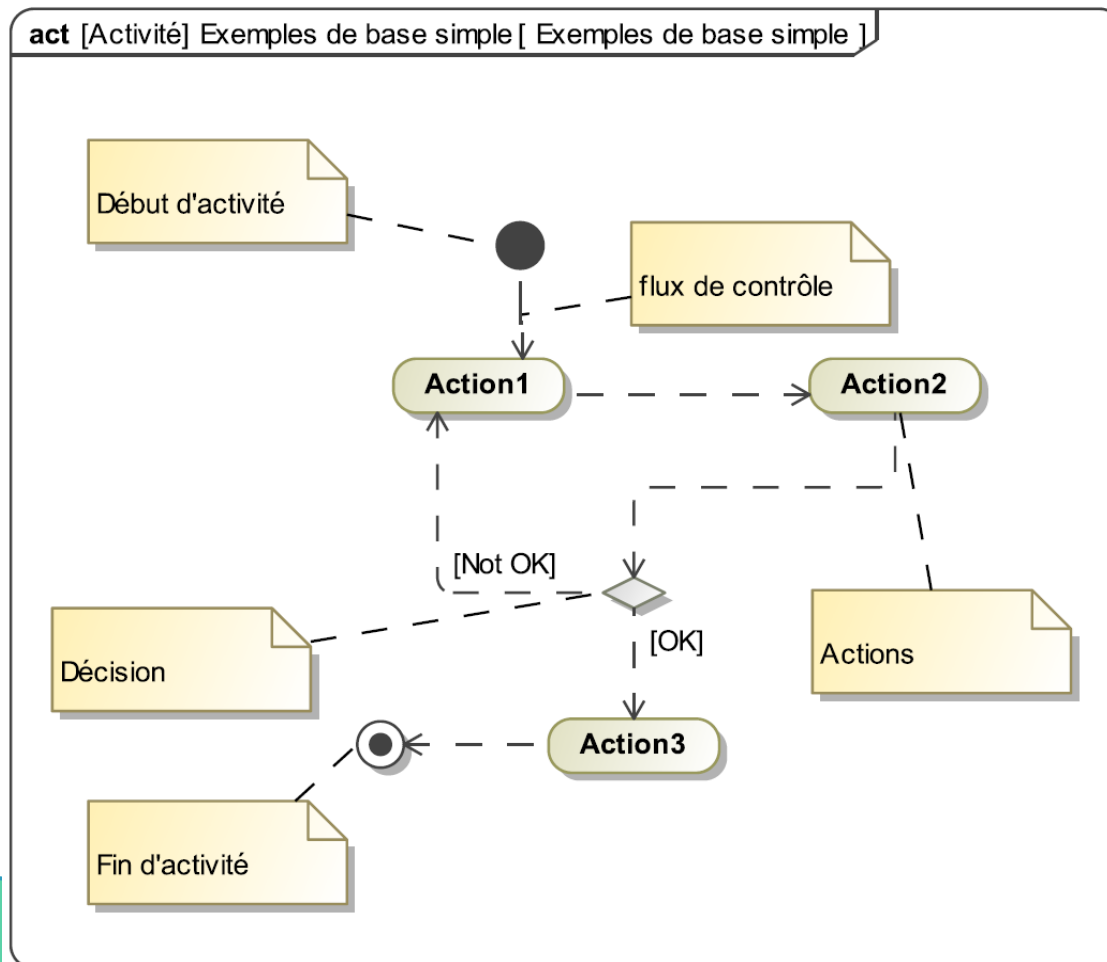
## 2. Diagramme d'activité en SysML

**choix à faire entre plusieurs conditions exclusives** : on utilise le symbole de branchement conditionnel. Les conditions sont notées entre crochets en sortie du branchement conditionnel (on peut utiliser le mot clé [else])



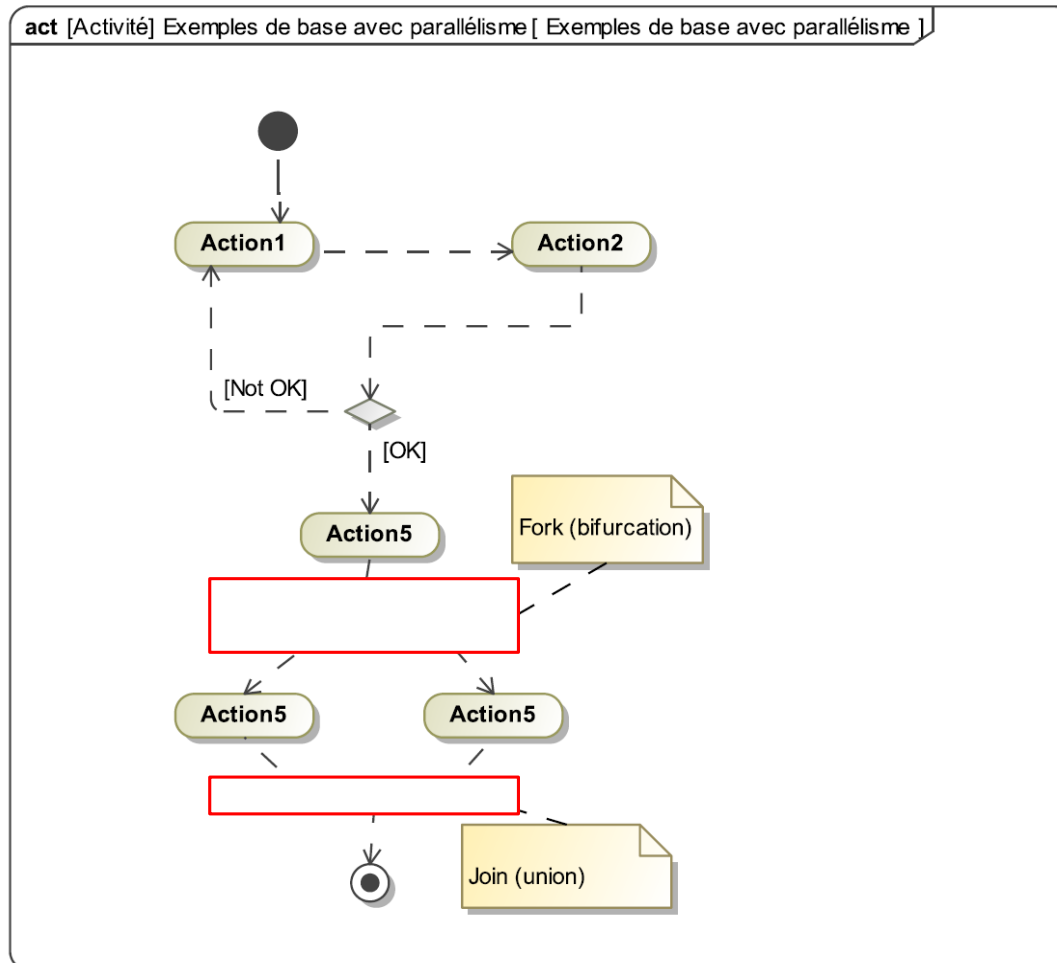
## 2. Diagramme d'activité en SysML

**choix à faire entre plusieurs conditions exclusives** : on utilise le symbole de branchement conditionnel. Les conditions sont notées entre crochets en sortie du branchement conditionnel (on peut utiliser le mot clé [else])



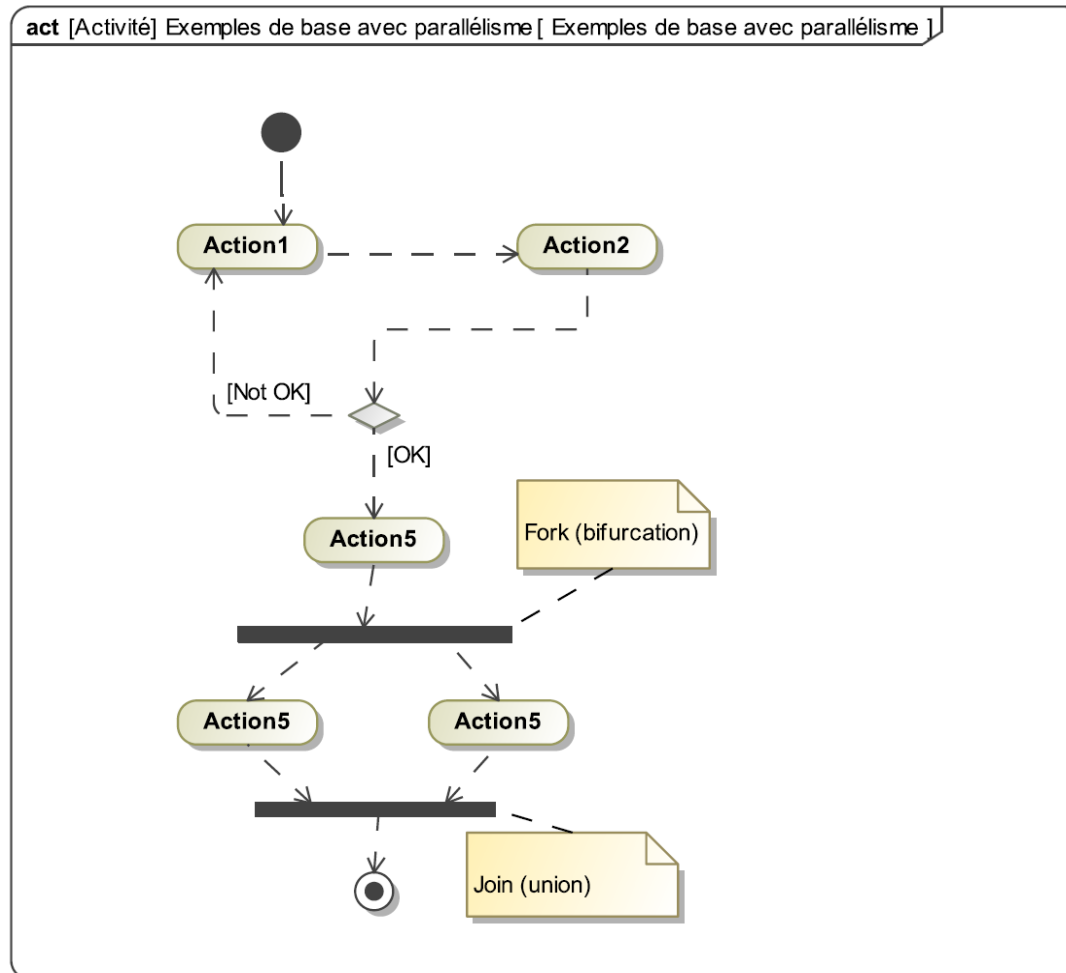
## 2. Diagramme d'activité en SysML

**Actions en parallèle ou synchronisées :** on utilise des symboles particuliers appelés Fork et Join comme indique sur la figure b (vrai également pour les diagrammes d'état lors de la sortie des états orthogonaux).



## 2. Diagramme d'activité en SysML

**Actions en parallèle ou synchronisées :** on utilise des symboles particuliers appelés Fork et Join comme indique sur la figure b (vrai également pour les diagrammes d'état lors de la sortie des états orthogonaux).



**FIN**

