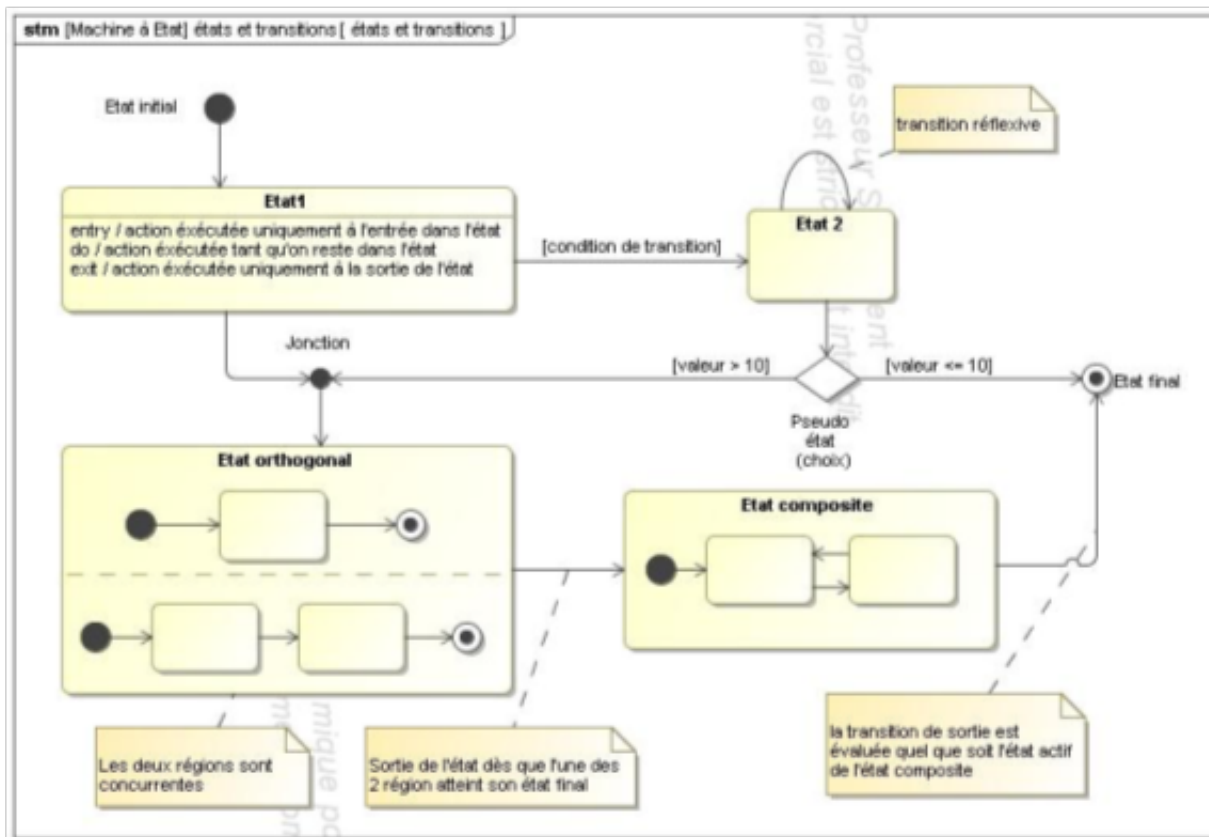


RÉVISIONS GRAPHE D'ÉTATS

Cours

v1.1

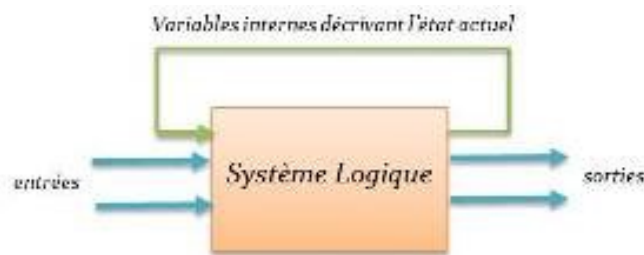
Lycée Chaptal – 45 Boulevard des Batignolles - 75008 Paris



1 Systèmes séquentiels, à évènements discrets et machines d'états

1.1 Introduction

Les systèmes à évènements discrets (SED) permettent de modéliser la succession des différentes étapes d'évolution d'un système. Ce sont des systèmes séquentiels dans lesquels une même cause peut produire des effets différents (par opposition aux systèmes combinatoires).



Les variables de sorties d'un **système combinatoire** s'écrivent mathématiquement par la relation suivantes :

$$S(t) = f(E(t))$$

Les variables de sorties d'un **système séquentiel** s'écrivent mathématiquement par la relation suivantes :

$$S(t) = f(E(t), S(t))$$

1.2 Machines d'états (stm) et diagrammes d'états SysML

Pour modéliser le fonctionnement des SED, nous allons utiliser le concept de machines d'états (ou **stm** de l'anglais S**T**ate Machine). L'outil de modélisation SysML permet, grâce au diagramme d'états, de représenter l'évolution d'un des blocs d'un diagramme de définition de blocs (**bdd**) ou d'un diagramme de blocs internes (**ibd**).



Définition *État*

Un état modélise une phase de fonctionnement du système. Pendant cette phase, le système accomplit une ou plusieurs actions et/ou activités, ou attend un évènement. Il peut être actif ou non.



Définition *Diagramme d'états*

Un diagramme d'états du langage SysML donne une représentation graphique des transitions possibles entre les différents états d'une machine d'états.

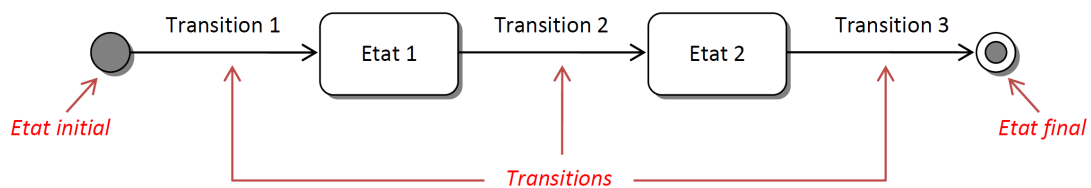
L'objectif de ce diagramme est de décrire les différents états pris par un bloc en fonction des évènements qui lui arrivent. Le passage d'un état à un autre se fait en franchissant une transition.



Définition *Éléments de base d'un diagramme d'états*

Les éléments de base sont donc : l'**état** avec ses **actions** et **activités**, la **transition** et les **événements**.

Les éléments graphiques utilisés dans ce diagramme sont principalement des rectangles aux coins arrondis pour les états et des flèches pour les transitions. Les flèches sont orientées à partir d'un état source et vers un état cible. La transition est franchie (passage de l'état **source** à l'état **cible**) lors de l'apparition de l'événement rattaché à la transition.



1.3 Règles d'évolution

- Le point de départ de la machine d'états est indiqué par l'**état initial** ;
- Les transitions sont franchies en fonction de l'activité des états sources et des transitions ;
- Un seul état par machine d'états ou sous-machine d'états est actif à la fois. Différentes structures de machines d'états (voir plus loin) permettent d'avoir plusieurs états actifs dans des machines d'états distinctes.

2 Les différents types d'états et de structures

2.1 États usuels

Dans les diagrammes d'états, on trouve principalement trois sortes d'états :

- état initial – pseudo-état qui indique un point d'entrée dans un graphe. Il n'y en a qu'un seul par machine d'états ;
- état final – pseudo état non obligatoire, il indique que le système décrit n'a plus d'état actif. Il peut y en avoir plusieurs dans un diagramme d'états. En effet, plusieurs scénarios peuvent être possibles pour mettre fin à un comportement ;
- état générique – état courant, on peut principalement lui rattacher des **activités**, des **actions** ou **activités en entrée et en sortie** d'un état.



Définition *Activités*

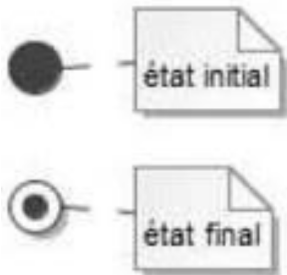
Une activité prend du temps et peut être interrompue par un événement. On la trouve à l'intérieur des états par le mot « **do** ». Une activité est donc dite interruptible. Elle peut se finir naturellement seule (dans ce cas elle est dite finie) ou nécessiter un événement pour se terminer.

Par exemple `do / émettre son` est une activité.



Définition Actions

une action ne prend pas de temps et ne peut pas être interrompue. Son exécution peut par exemple provoquer un changement d'état, l'émission d'un ordre ou un retour de valeur. Ces actions ne sont appelées que par les mots clé « **entry** » ou « **exit** ».



Nom de l'état

entry / actions (ou activités) d'entrée
do / activités
exit / actions (ou activités) de sortie

Le nom de l'état est indiqué dans la partie haute du rectangle. La partie basse est réservée aux activités et actions.

Par exemple, `entry / C:=C+1` et `exit / eclairage:= 0` sont des actions.

entry	Ex : <code>entry / activite1</code>	Le front <code>entry</code> apparaît à l'activation de l'état.
do	Ex : <code>do / activite2</code>	L'activité2 s'exécute tant que l'état est actif après la terminaison de l'activité1.
exit	Ex : <code>exit / activite3</code>	Le front <code>exit</code> apparaît à désactivation de l'état. L'activité2 est interrompue, et l'activité3 est exécutée.



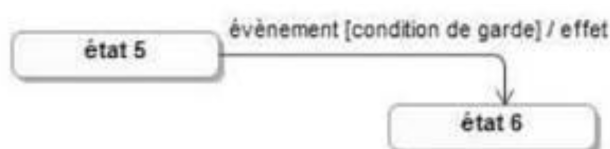
Remarque

Il est possible de ne définir aucun comportement dans l'état, cette structure vide pouvant indiquer par exemple un état d'attente.

2.2 Transitions et événements associés usuels

2.2.1 Structure de base d'une transition

Les possibilités d'évolution entre états sont représentées par des flèches qui indiquent le sens d'évolution.



Ces flèches portent (ou non) un message sous la forme **événement**[garde]/**effet** qui définit la transition avec son événement déclencheur, sa condition de garde et l'effet associé (action), ces termes étant détaillés un peu plus loin.

L'état de départ est appelé l'**état source** et l'état d'arrivée est appelé l'**état cible**. L'activité de l'état source de la transition est nécessaire au franchissement de la transition.

Une transition réflexive entraîne une sortie d'état puis un retour dans ce même état. Lors de l'événement E1 (voir figure ci-dessous), l'état 7 reste actif et :

- L'activité en cours (associée au comportement **do**) est interrompue,
- L'action associée au comportement **exit** est exécutée,
- L'action associée au comportement **entry** est exécutée,
- L'activité associée au comportement **do** est lancée à nouveau.

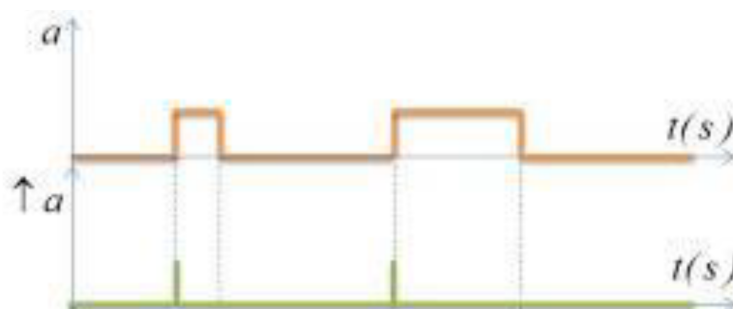


2.2.2 Notion d'événement



Définition Événement

Un événement est un changement qui doit être pris en compte dans la description par diagramme d'états. Un événement se produit à une date précise qui est l'occurrence de cet événement. **La durée d'un événement est nulle et il ne peut pas être mémorisé.** Un événement est le passage de 0 à 1 d'une variable. On peut lui associer la notion de front montant (noté ci-dessous $\uparrow a$).



L'occurrence de l'événement est une condition nécessaire au franchissement.

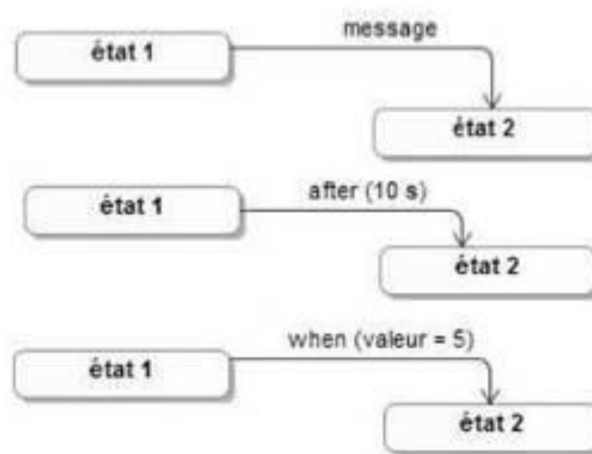


Remarque Front descendant

Pour désigner un événement correspondant au front **descendant** d'une variable **a**, on utilisera l'événement $\bar{a}$.

Il existe trois types principaux d'évènements associés à une transition :

- l'évènement de signal (**signal event**) : un message asynchrone (cela signifie que le destinataire ne l'attend pas forcément. Par exemple appui sur un bouton) est arrivé ;
- l'évènement temporel (**time event**) : un intervalle de temps s'est écoulé depuis l'entrée dans un état (mot clé « **after** ») ou un temps absolu a été atteint (mot clé « **at** ») ;
- l'évènement de changement (**change event**) : une valeur a changé de telle sorte que la transition est franchie (mot clé « **when** »).



2.2.3 La condition de garde

La condition de garde est la deuxième condition de franchissement.



Définition *Condition de garde*

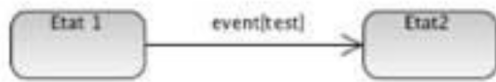
Il s'agit d'une expression booléenne encadrée de crochets, évaluée lorsque l'état précédant la transition est actif ET que l'évènement déclencheur se produit. Si la condition de garde est vraie, la transition est alors franchie, sinon elle ne l'est pas et l'évènement est perdu.



Remarque *Différence fondamentale entre évènement et condition de garde*

Un évènement est parfaitement daté dans le temps (cf front montant). Une condition de garde n'est pas datée, elle doit être vraie (niveau logique 1) à l'instant où l'évènement survient pour que la transition soit franchie. Si la condition de garde est fausse lors de l'évènement, il faudra attendre le prochain évènement pour espérer évoluer.

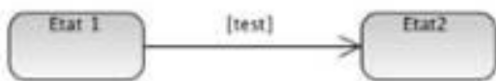
Exemples de transition :



À l'occurrence de event, test est évalué et la transition est franchie uniquement si test est vrai. L'éventuelle activité est interrompue. Si test n'est pas vrai, event est « perdu » et il faut attendre une seconde occurrence de event pour éventuellement franchir la transition si cette fois test est vrai.



À l'occurrence de event, la transition est franchie, sans condition. L'éventuelle activité est interrompue.



Si test est vrai, la transition est immédiatement franchie dès la fin de l'éventuelle activité (qui doit donc être une activité finie). S'il n'y a pas d'activité associée à l'état 1, la transition est franchie immédiatement si test est vrai.



Transition de complétion : est immédiatement franchie dès la fin de l'éventuelle activité (idem). Equivaut à [1].

2.2.4 Notion d'effet



Définition *Effet*

Un effet est un comportement (action ou activité) accompli lorsque la transition est franchie.

L'action (ou activité) associée à l'effet est exécutée AVANT d'exécuter les actions (ou activités) associées à l'état cible.

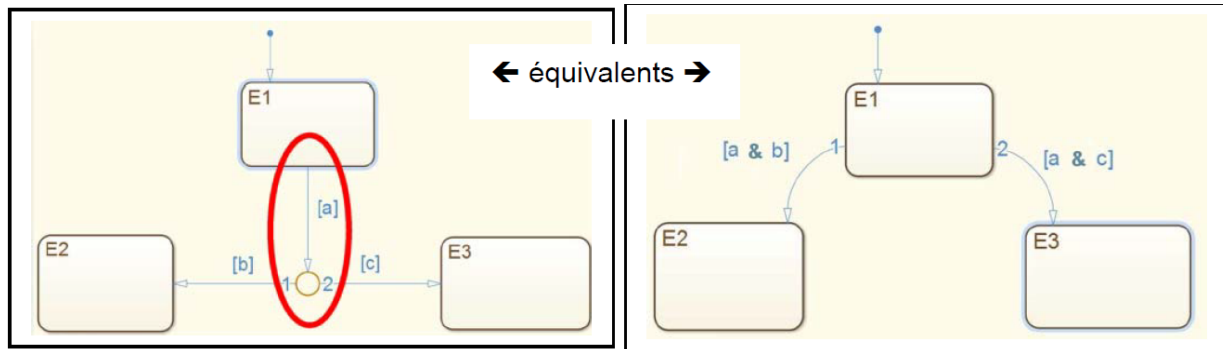
2.2.5 Transition basée sur un état interne

On peut utiliser l'activité d'un état comme variable dans une condition de garde. Par exemple, la condition de garde `[in Etat1]` est vraie (`[in Etat1]=1`) si l'état intitulé « Etat1 » est actif, et vaut 0 sinon.

2.3 Autre symbole utile



On peut trouver dans les graphes d'états les points de décision, qui doivent être suivis de deux transitions sous forme de condition de garde complémentaires, donc non compatibles. Il est d'ailleurs souvent utile d'utiliser la condition de garde `[else]`. La durée d'exécution est nulle et aucune activité ni action n'y est associée.



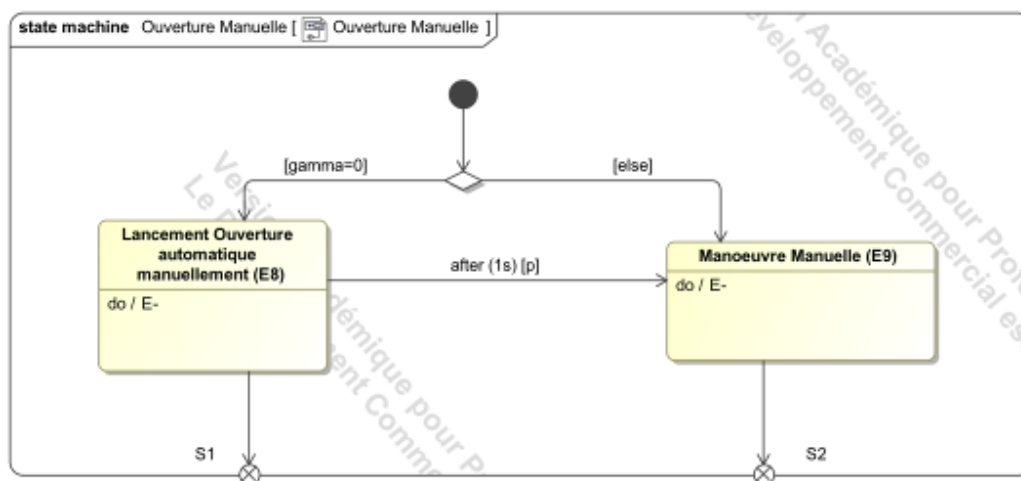
3 Structures et fonctions avancées

3.1 État composite

Lorsque le comportement à décrire :

- risque d'être peu lisible car trop complexe ;
- fait apparaître des parties "séparables" de l'ensemble.

Alors on extrait ces parties pour en faire des sous-graphes séparés, hiérarchiquement inférieurs au graphe principal. On englobe ces sous-graphes dans un état appelé **état composite**. Un état composite est composé de plusieurs sous-états internes. Un état composite possède un état initial.

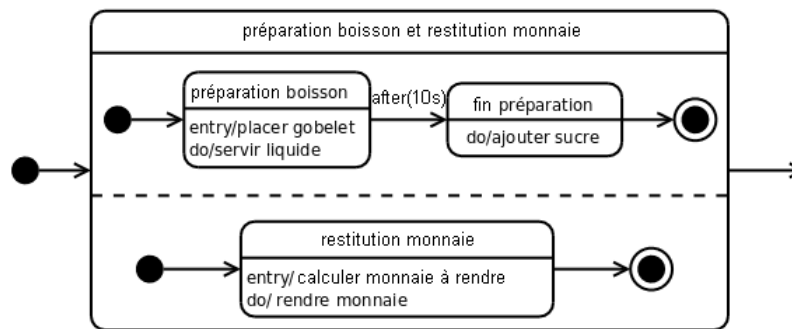


3.2 Notion d'état orthogonal



Définition *État orthogonal*

Un état orthogonal possède plusieurs régions, séparées par des pointillés et possédant chacune sa propre description d'états. Les différentes régions d'un état orthogonal fonctionnent toutes en parallèle sans aucune influence les unes sur les autres.



Remarque

Ainsi, avec un état orthogonal, il est possible d'avoir plusieurs états actifs à un instant t donné. Cela permet de réaliser plusieurs tâches en parallèle. Chaque région doit **obligatoirement** avoir un état initial (qui sont activé lors de l'activation de l'état orthogonal), et peut avoir un état final ou non.

La désactivation de l'état orthogonal entraîne la désactivation de l'ensemble des états de chaque région. Cette première est obtenue, par exemple, lorsque l'un des états finaux d'une région est actif, et que l'éventuelle transition de sortie est vraie. Ainsi, l'utilisation d'une transition basée sur les états internes de chaque région peut-être pertinente, afin de synchroniser la sortie de l'état orthogonal.