

Logique séquentielle

Systemes à Evénements Discrets SED

I. Introduction.

On distingue les systèmes combinatoires et les systèmes séquentiels.

Système combinatoire.

- ⇒ Une même cause produit toujours le même effet.
- ⇒ Un même état des entrées donne toujours le même état des sorties.
- ⇒ Un système est combinatoire si toutes les sorties sont de nature combinatoire.

Système séquentiel.

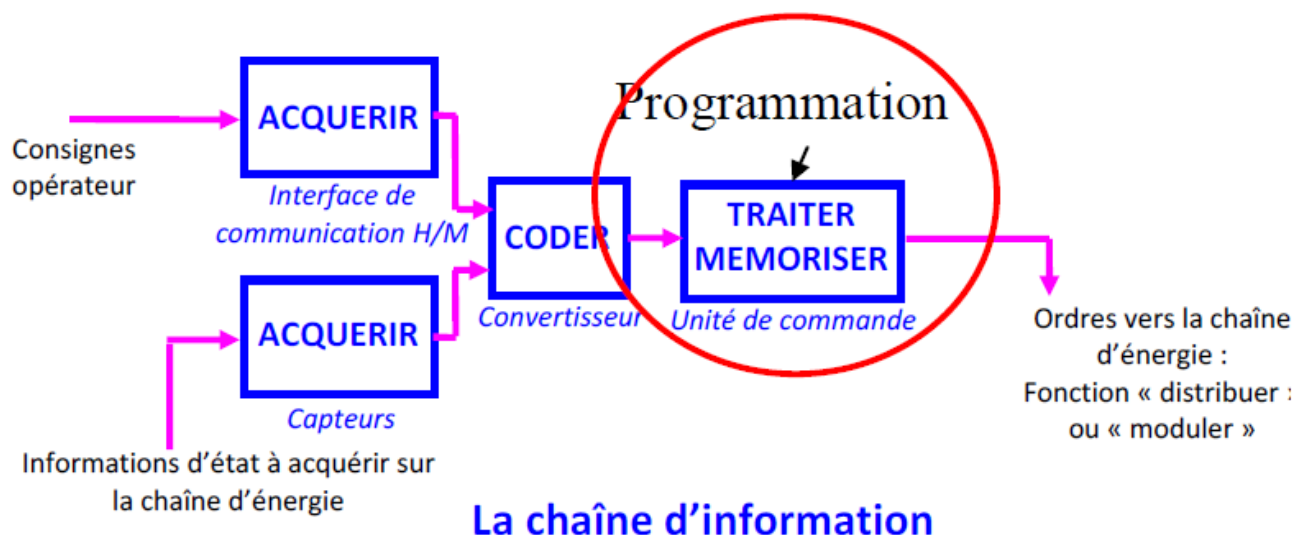
- ⇒ Une même cause peut produire des effets différents.
- ⇒ Un même état des entrées peut donner des états des sorties différents.
- ⇒ On tient compte de l'état du système, c'est un système à mémoire.
- ⇒ Un système est séquentiel si au moins une des sorties est de nature séquentielle.

La description du comportement d'un système séquentiel peut être réalisée notamment par :

- ✓ Un chronogramme.
- ✓ Un algorithme.
- ✓ Un diagramme de séquence Sysml.
- ✓ Un diagramme d'états (ou graphe d'état) Sysml.

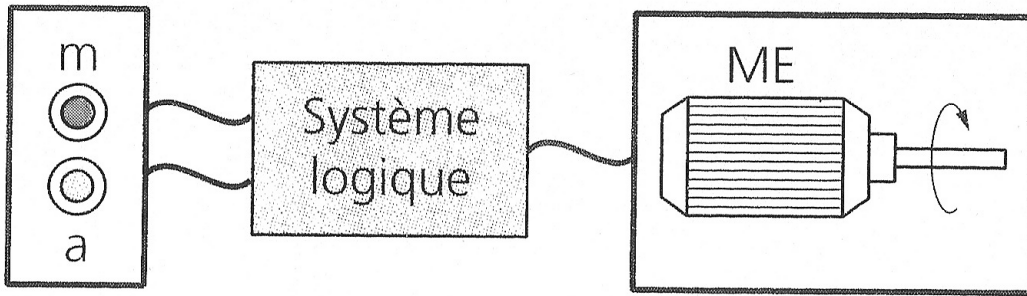
Ces outils sont, à la base, des outils de modélisation du comportement séquentiel.

Ils peuvent aussi servir à la programmation des composants qui réalisent la fonction « Traiter » de la chaîne d'information (microcontrôleur, microprocesseur, automate programmable, puce à circuits logiques programmables...).

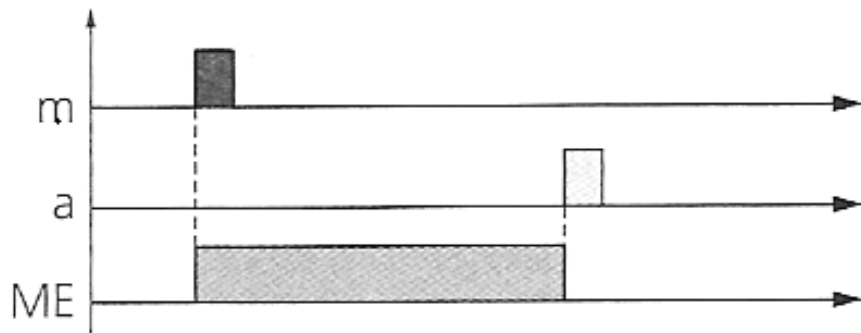


II. Exemple de système séquentiel

Soit un moteur électrique commandé par deux boutons poussoir.



On donne le chronogramme de fonctionnement.



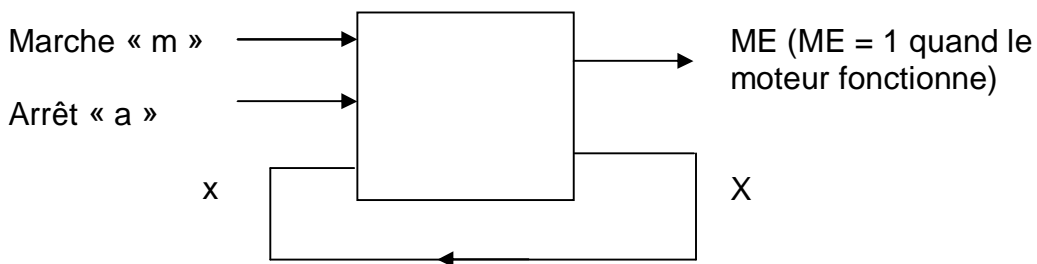
Lorsque les deux entrées sont à 0, le moteur peut être en marche ou à l'arrêt.

Un même état des entrées peut donner deux sorties différentes.

Le système est donc séquentiel.

Nous allons utiliser une variable interne « x » pour caractériser l'état du système.

On note « x » variable interne prise en compte comme entrée et « X » variable interne prise en compte comme sortie.



On peut représenter le fonctionnement sur le tableau de Karnaugh suivant :

On exprime l'évolution de l'état du système ($ME = X$, moteur en fonctionnement), en fonction des 2 entrées (« m » et « a ») et de l'état du système précédent (« x »).

Equation de fonctionnement :

$$ME = X = (m + x) \cdot \bar{a}$$

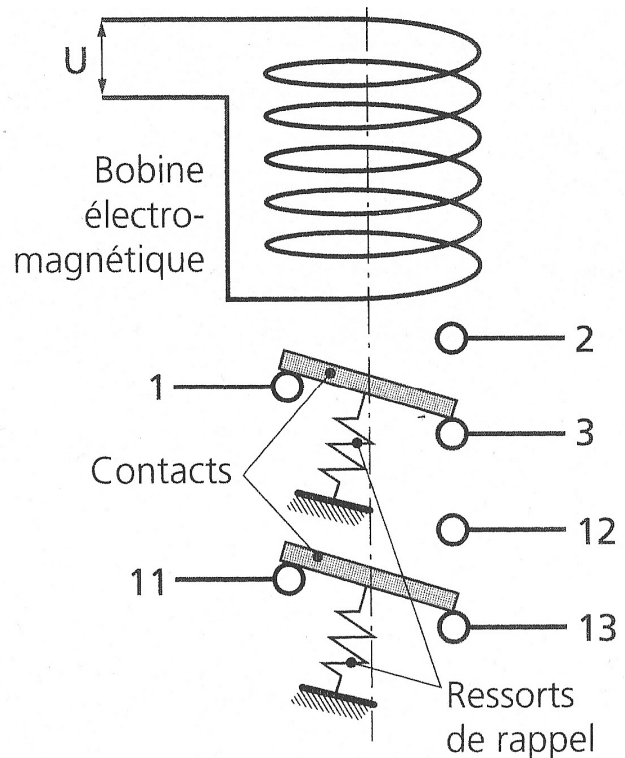
	x		
	0	1	x
a	0	1	m
	0	1	1
	0	0	0

Pour réaliser le fonctionnement de ce système séquentiel, on peut utiliser un relais auto maintenu.

Principe de fonctionnement d'un relais :

Quand on alimente la bobine, le champ magnétique ferme les contacts et le courant circule.

Remarque : Le relais schématisé est un relais à 2 contacts.



Pour réaliser une mémoire, on utilise un relais à deux contacts :

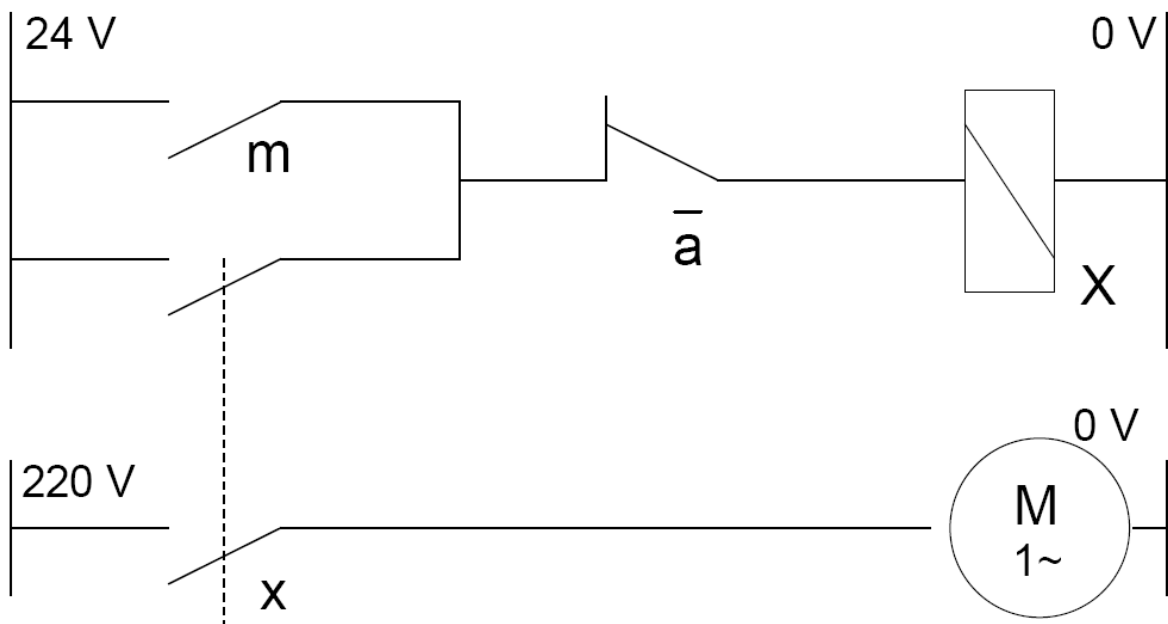
- ✓ Un des contacts assurant l'auto maintient.
- ✓ L'autre contact assurant la séparation du circuit de commande et du circuit de puissance.

La variable « X » représente le passage du courant dans la bobine.

La variable « x » représente l'état du contact associé (état interne du relais).

L'équation de fonctionnement est : $M = X = (m + x).\bar{a}$

C'est une mémoire à effacement prioritaire.



III. Le langage SysML

SysML est un langage graphique, composé de diagrammes qui permettent d'aborder plus facilement les systèmes pluri techniques.

SysML « Systems Modeling Language », veut dire Langage de Modélisation de systèmes.

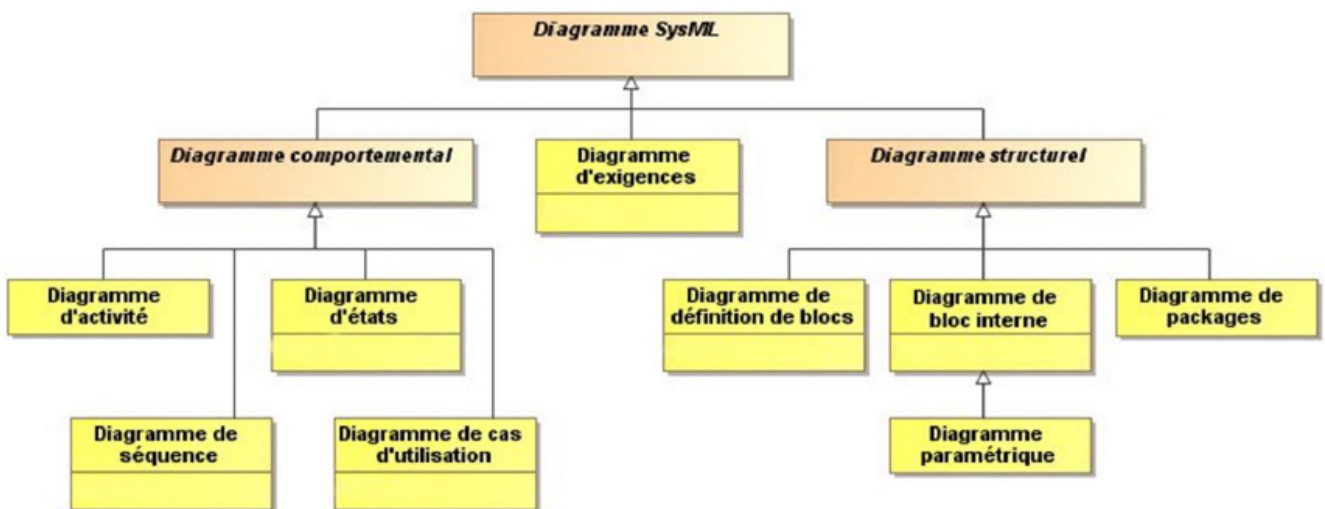
Les diagrammes SysML permettent de représenter :

- ✓ Les exigences du système.
- ✓ Les composants du système.
- ✓ Les flux de toute nature (matière, énergie et information).
- ✓ Le fonctionnement du système.

Les diagrammes SysML :

- ✓ Sont utilisés tout au long du cycle de vie du système (conception, analyse...).
- ✓ Sont communs à tous les champs disciplinaires.
- ✓ Ont leur description propre et sont le plus souvent liés entre eux.
- ✓ Remplacent la plupart des autres outils de description auparavant utilisés.

Il y a 9 diagrammes SysML (tous ne sont pas au programme) :



Nous avons vu dans un cours précédent :

- ✓ Le diagramme des exigences.
- ✓ Le diagramme des cas d'utilisation.
- ✓ Le diagramme de définition de bloc.
- ✓ Le diagramme de bloc interne.

Nous allons voir dans ce cours le diagramme d'états.

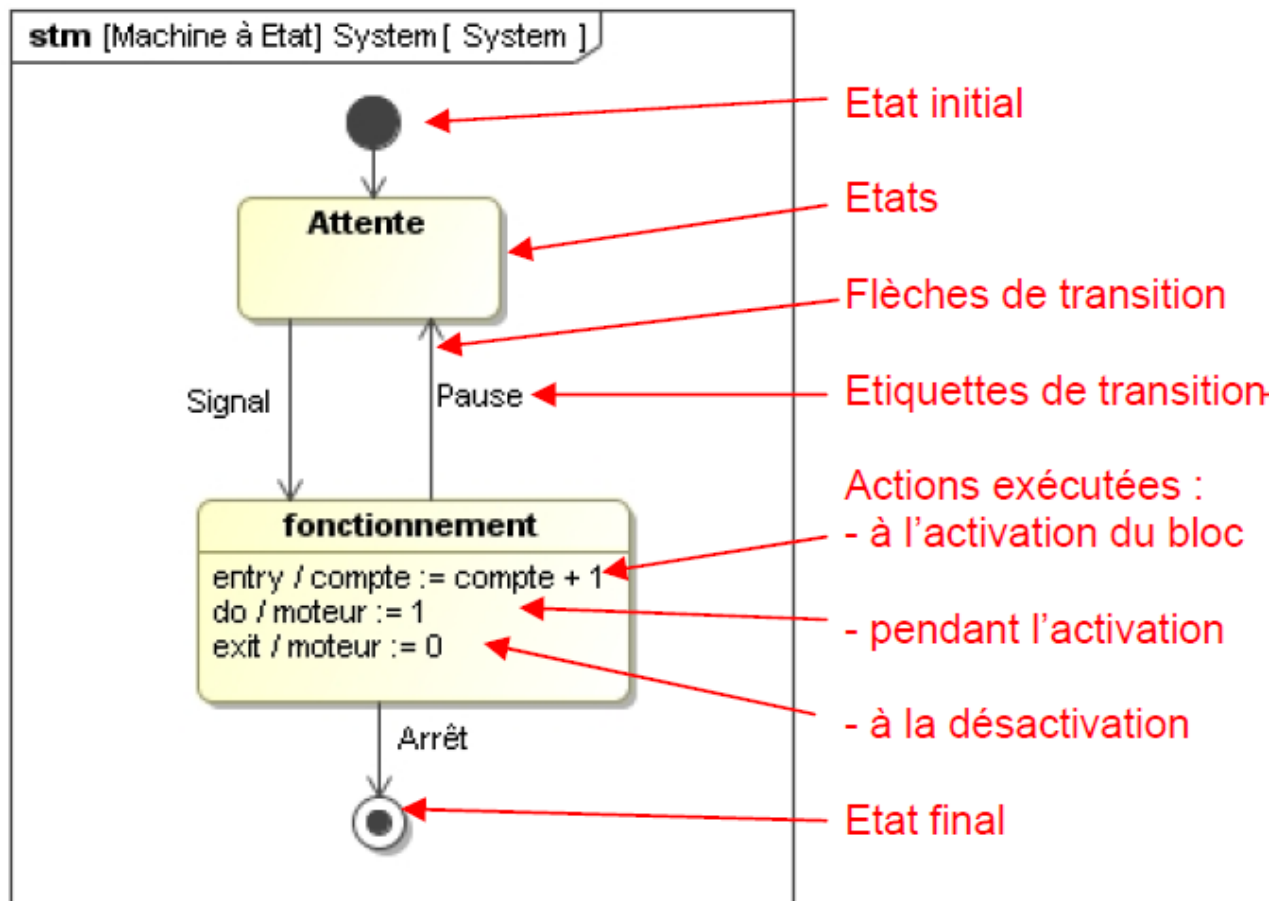
IV. Diagramme d'états / transitions (SysML State Machine Diagram).

« Comment représenter les différents états du système ? »

Structure d'un diagramme d'états :

- ✓ Les différents états du système sont représentés par des rectangles encadrés.
- ✓ Aux états peuvent être associées des actions et/ou une activité.
- ✓ L'état initial est représenté par un rond noir ●.
- ✓ L'état final par un rond avec un point noir ⊙.
- ✓ Les transitions relient les différents états.
- ✓ Les événements et les conditions sont liés aux transitions.

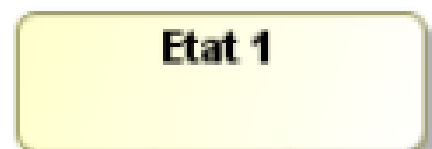
Exemple 1 : Diagramme d'état d'une commande de moteur.



Etat :

Un état représente une période de la vie du système, pendant laquelle il accomplit une ou plusieurs actions, ou attend un (ou des) événement(s).

Cet état peut être actif ou non.



Il n'y a qu'un seul état actif à un instant donné, sauf dans le cas des sous-états parallèles. En plus de la succession d'états « normaux », le diagramme d'états comprend également deux pseudo-états :

- ✓ L'état initial : ● .
- ✓ L'état final: ○ (Il est possible d'utiliser plusieurs états finaux).

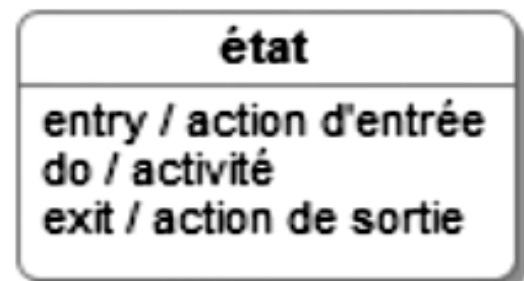
Activité et action :

Activité : Une activité peut être considérée comme une tâche à réaliser. Elle prend un certain temps et peut être interrompue. Elle peut être associée à un état, spécifiée par le mot clé « do ».

Action : Une action est également une unité de comportement.

La différence est qu'une action ne prend pas de temps (considérée instantanée) et ne peut pas être interrompue.

Elle peut être associée à un état, spécifiée par le mot clé « entry » ou « exit ».



Exemples :

- ✓ Activité : Ouvrir une porte, sortir une tige de vérin, accélérer une charge
- ✓ Action : Emettre un ordre à un pré-actionneur, incrémenter une variable, etc...

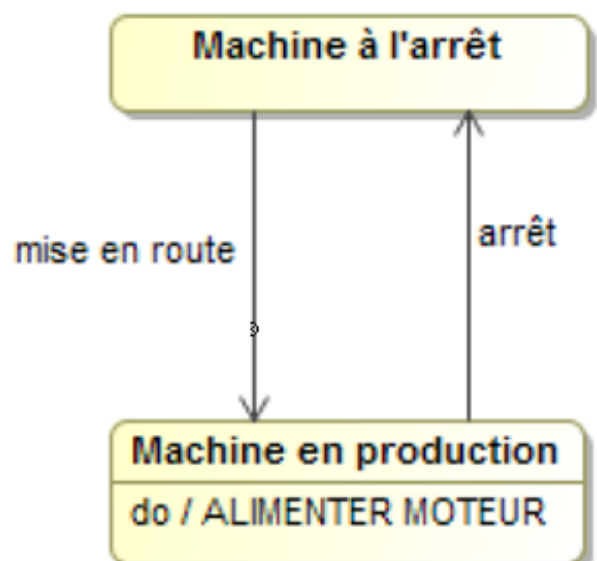
Événement :

Spécification d'une occurrence qui peut déclencher une réaction du système.

Un événement possède une localisation dans le temps, c'est en générale un front montant.

Exemples :

- ✓ Appuis sur un bouton.
- ✓ Détection d'un objet.
- ✓ Changement d'une variable interne « when (N=10) ».
- ✓ Comptage du temps « after (90s) » ou « at(10:00) ».



Transition :

Évènement [condition]/action →

Une transition décrit l'évolution du système.

Elle traduit le passage instantané d'un « état source » vers un « état destination ».

Une transition peut posséder :

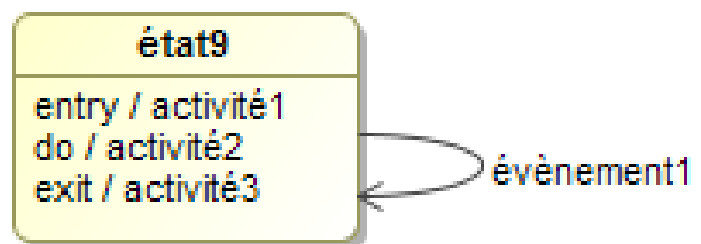
- ✓ Un évènement déclencheur.
- ✓ Une condition.
- ✓ Une action.

Principe :

- ✓ La transition n'est évaluée que si l'état source est actif.
- ✓ La transition n'est franchie que si l'évènement devient vrai (et si l'éventuelle condition est vraie).
- ✓ L'activité « do » de l'état source est stoppée (les actions entry et exit sont toujours exécutées).
- ✓ L'action associée est exécutée pendant le franchissement.

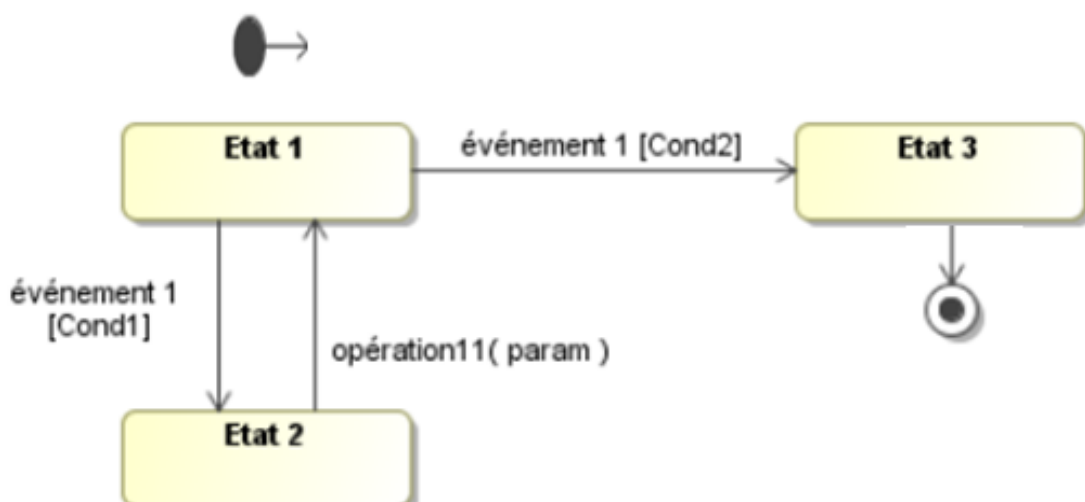
Une transition réflexive est une transition où état source et état destination sont identiques.

Si évènement1 apparaît, l'état9 reste actif, mais l'activité2 est interrompue, l'activité3 est exécutée puis lorsqu'elle est terminée, l'activité1 se réalise et enfin de nouveau l'activité2.

Condition :

Une condition (ou condition de garde) est une expression booléenne qui doit être vraie lorsque l'évènement arrive pour que la transition soit déclenchée.

Elle se note entre crochets.



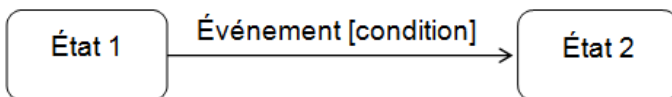
Remarques :

- ✓ Si les deux conditions « Cond1 » et « Cond2 » ne sont pas vraies, l' « événement 1 » est « perdu ».
- ✓ Si les deux conditions sont vraies simultanément, le diagramme est dit « non-déterministe » et on ne peut savoir dans quel état le bloc passera.

Cette situation est à proscrire.

Donc plusieurs transitions avec le même événement doivent avoir des conditions de garde non vraies simultanément.

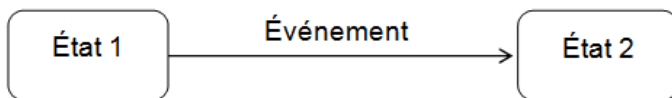
Synthèse sur l'évolution d'un état :



L'état 1 est actif.

A l'occurrence de l'événement la transition est franchie uniquement si la condition est vraie.

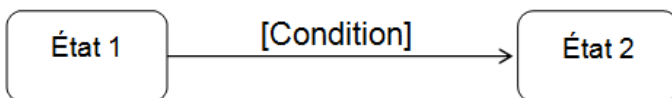
L'éventuelle activité de l'état 1 est interrompue.



L'état 1 est actif.

A l'occurrence de l'événement la transition est franchie.

L'éventuelle activité de l'état 1 est interrompue.



L'état 1 est actif.

Si la condition est vraie la transition est franchie à la fin de l'éventuelle activité.

L'activité doit donc ne pas être continue mais avoir un début et une fin.



L'état 1 est actif.

La transition est automatiquement franchie à la fin de l'éventuelle activité.

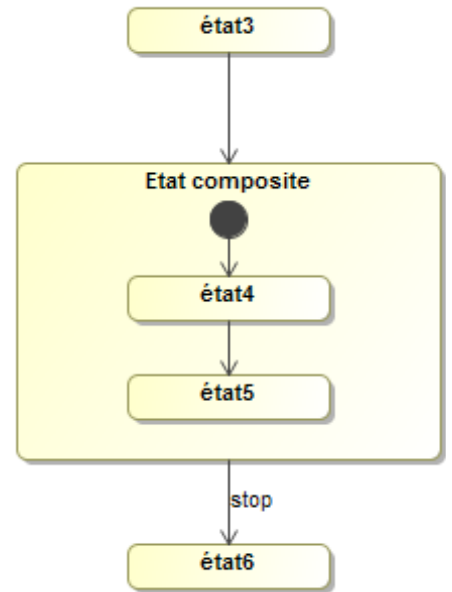
L'activité doit donc ne pas être continue mais avoir un début et une fin.

Etat composite ou super-état :

Un état composite (aussi appelé super-état) contient un diagramme d'état détaillant son fonctionnement séquentiel.

- ✓ L'activation de l'état composite entraîne l'activation du pseudo-état initial. (rond noir)
- ✓ La désactivation de l'état composite (ici évènement stop) entraîne la désactivation de l'état actif (ici état4 ou état5).

L'état composite est donc hiérarchiquement supérieur à l'automate fini qu'il contient.



Exemple : Diagramme d'état d'un radioréveil

Le diagramme d'états fait apparaître des sous-états au sein de l'état « Radio Auto ».

Pour plus de lisibilité et de confort, il est recommandé lorsque le diagramme se complexifie, d'utiliser les états composites.

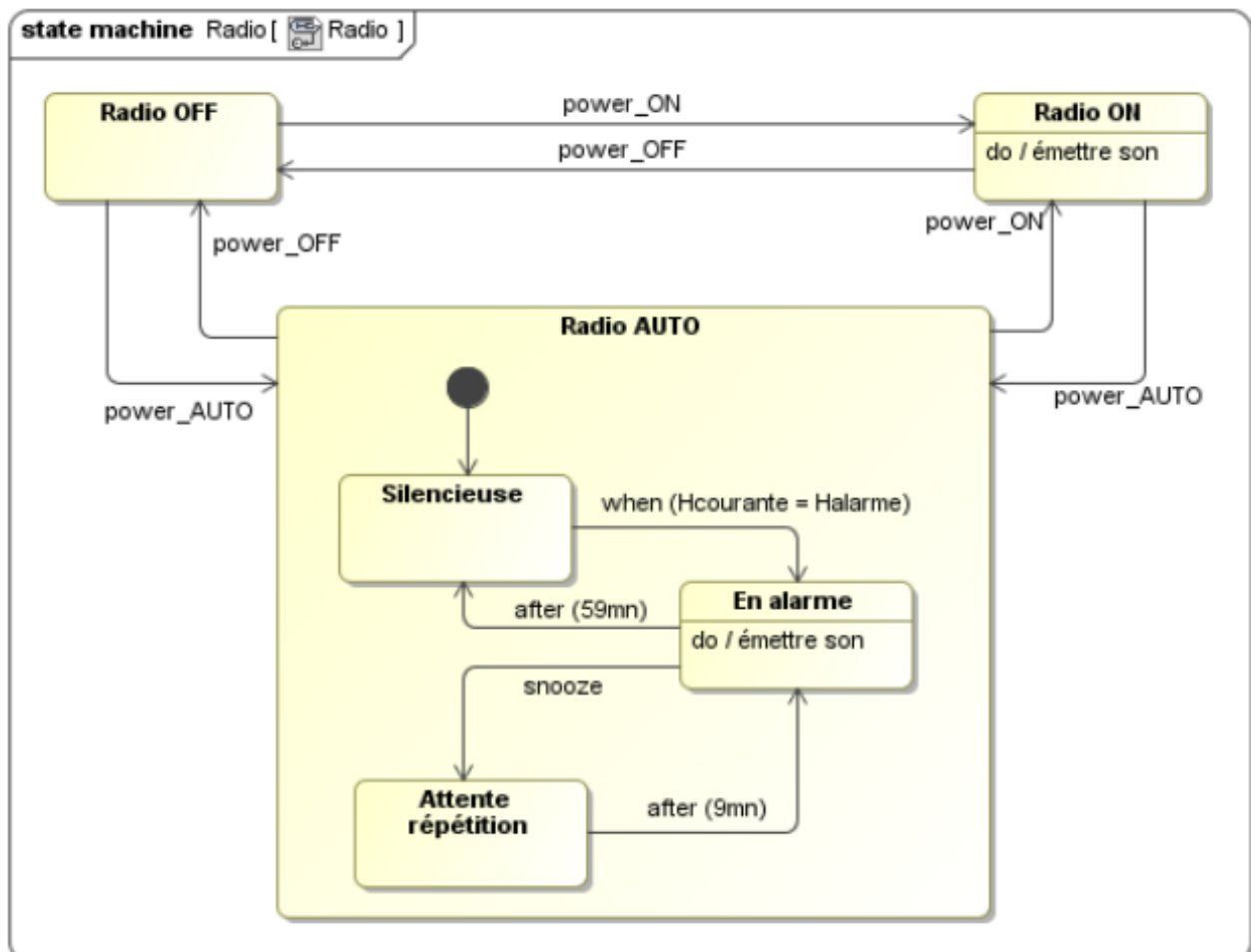
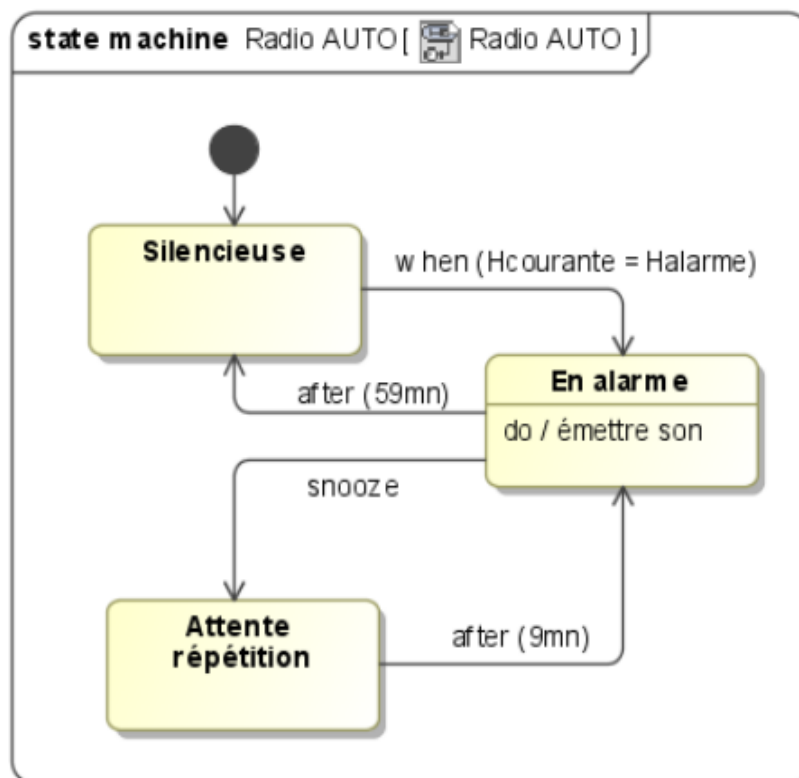
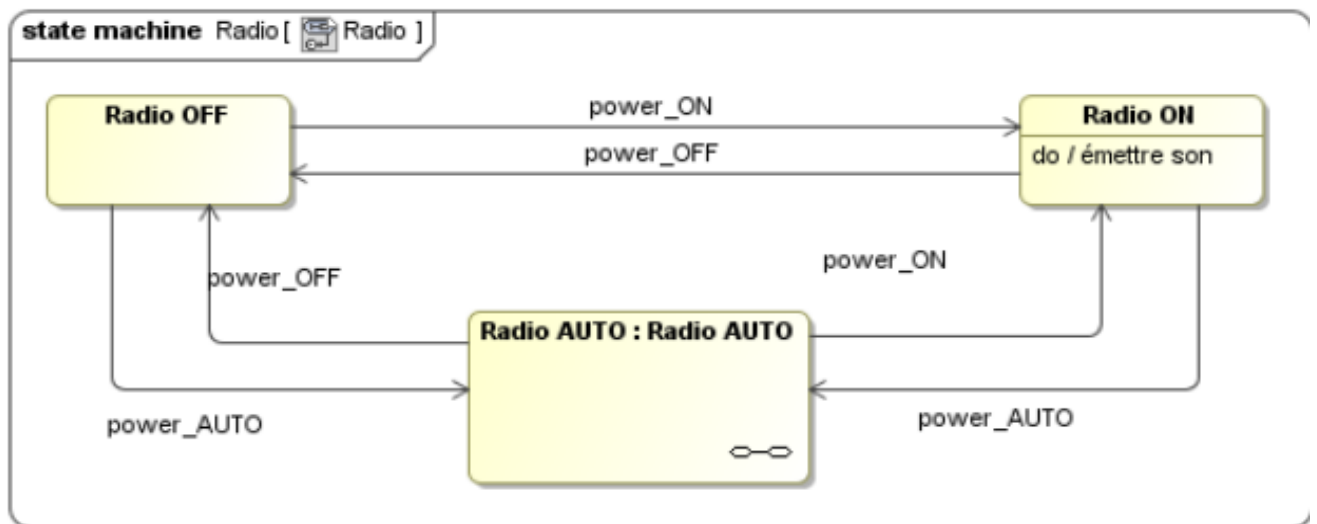


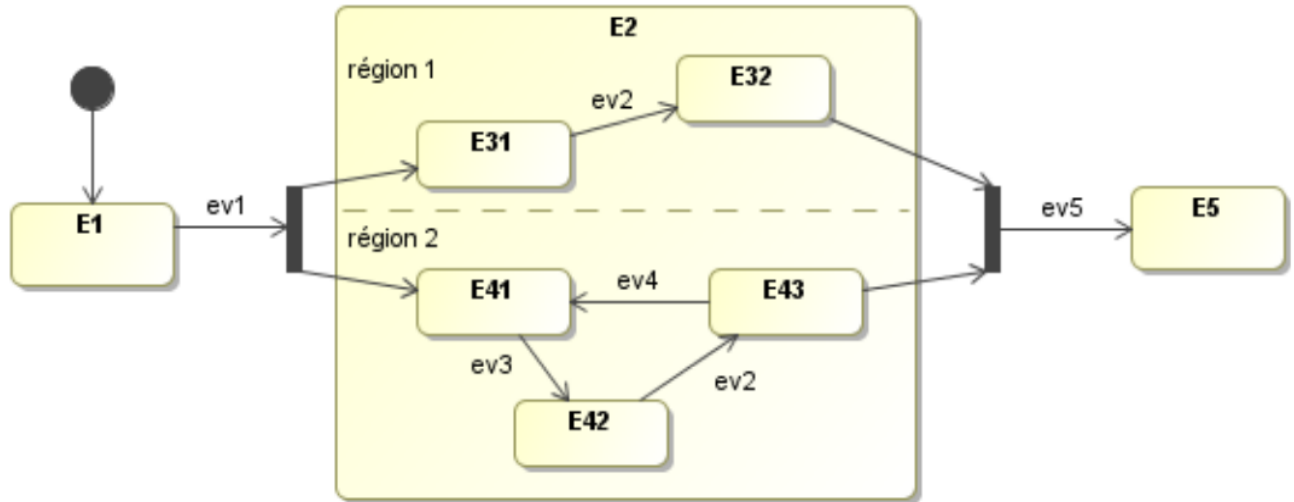
Diagramme d'états d'un radioréveil nécessitant la décomposition des sous états au sein de l'état « radio AUTO ».



Structure parallèle (état orthogonal) :

Un état composite peut contenir plusieurs régions, il suffit graphiquement de le séparer par des traits pointillés.

Elle contient ses propres états et ses propres transitions. Les régions évoluent en parallèle et indépendamment.



Dans l'exemple ci-dessus, à partir de l'état « E1 », quand l'événement « ev1 » arrive, l'élément passe dans l'état composite « E2 ».

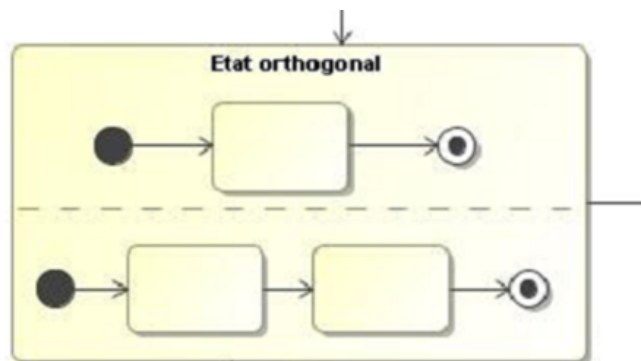
Cela signifie qu'il est à la fois dans les états disjoints « E31 » et « E41 » ;

Ensuite, suivant l'ordre d'arrivée des événements « ev2 », « ev3 » ou « ev4 », chaque région va évoluer indépendamment.

Pour passer à l'état « E5 » (et sortir de l'état composite « E2 »), il faut que l'élément soit à la fois dans « E32 » et « E43 » quand « ev5 » arrive.

Dans l'illustration suivante sont également présentes deux régions en parallèle.

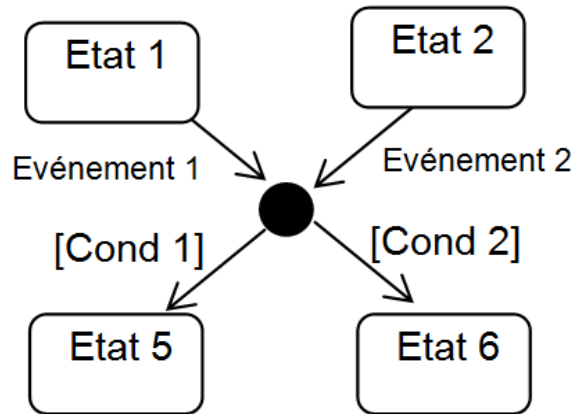
La sortie de l'état orthogonal se fait dès que toutes les régions ont atteint leur état final (équivalent à la représentation précédente).



Notons aussi deux nouveaux pseudos états :

Pseudo état de jonction :

- ⇒ A chaque jonction une et une seule des conditions doit être vraie.



Pseudo état de choix :

Un point de décision possède une entrée et au moins deux sorties.

A chaque choix une et une seule des conditions des être vraie.

