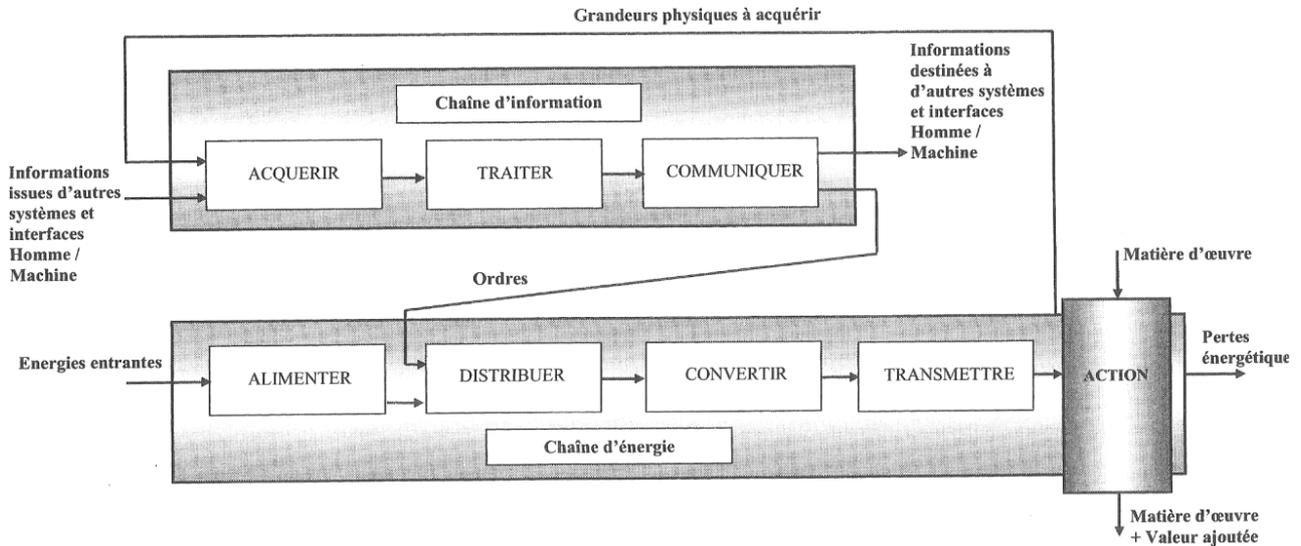


COURS LOGIQUE 1 : LOGIQUE COMBINATOIRE

I. Variable logique.

Rappel : structure d'une chaîne fonctionnelle d'un système automatisé.



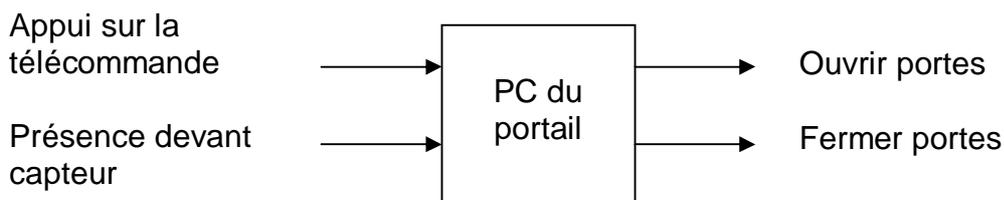
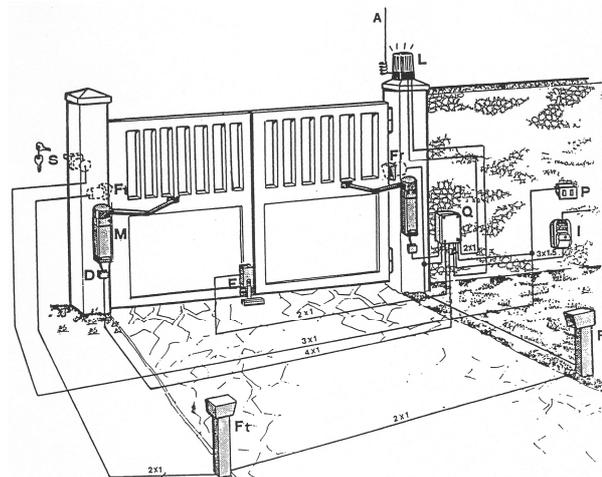
- Les ordres et les informations peuvent être :
- ✓ Analogique (par exemple une tension variable)
 - ✓ Logique (0 ou 1, vrai ou faux)
 - ✓ Numérique

Exemple de système logique :

Le portail automatisé. Pour simplifier, on s'intéresse aux éléments suivants :

- ✓ Les 2 portes, avec les 2 moteurs
- ✓ La télécommande
- ✓ La cellule photo électrique

Entrées/sorties de la Partie commande (PC) du portail : (les entrées sont les informations, les sorties sont les ordres)



Les entrées et les sorties sont sous la forme tout ou rien (1 ou 0) (vrai ou faux), on les appelle des variables logiques.

L'objet de ce chapitre est de modéliser le fonctionnement des PC

II. ALGEBRE DE BOOLE. Il permet de traiter des problèmes de logique.

1. Opérateurs logiques de base.

Opérateur égalité

Table de vérité

a	S
0	0
1	1

Equation logique : $S = a$

Opérateur ET

Table de vérité

a	b	S
0	0	0
0	1	0
1	1	1
1	0	0

Equation logique : $S = a.b$

Opérateur OU

Table de vérité

a	b	S
0	0	0
0	1	1
1	1	1
1	0	1

Equation logique : $S = a + b$

Opérateur NON

Table de vérité

a	S
0	1
1	0

Equation logique : $S = \bar{a}$

Opérateur OU exclusif

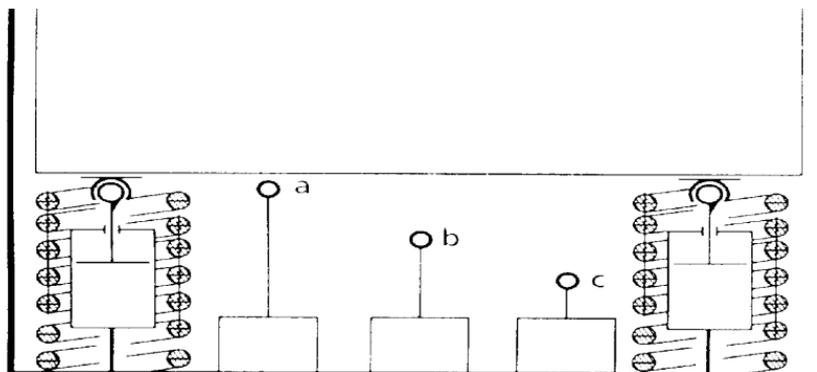
Table de vérité

a	b	S
0	0	0
0	1	1
1	1	0
1	0	1

Equation logique : $S = a \oplus b$

2. Exemple de problème de logique.

Etude d'un monte charge.



Un monte-charge doit permettre le levage de masses comprises entre 20 et 80 kg. Pour cela, il comporte une plate-forme reposant sur des ressorts.

Selon l'importance des charges à soulever, trois contacts réglables sont mis en circuit. Les contacts passent à 1 lorsque qu'ils sont en contact avec la cuve du monte-charge.

Cahier des charges

- ✓ À vide, aucun des contacts n'est activé. le monte-charge peut fonctionner.
- ✓ Pour des charges comprises entre 5 et 20 kg, le monte-charge ne peut fonctionner. Le contact « a » est actionné.
- ✓ Pour les charges comprises entre 20 et 80 kg, le monte-charge doit fonctionner. « a » et « b » sont actionnés.
- ✓ Pour des charges supérieures à 80 kg, le monte-charge ne peut fonctionner. Les contacts « a », « b » et « c » sont actionnés.

Problème posé : Il faut modéliser le comportement attendu en équations afin de réaliser la partie commande.

Question : Déterminer l'équation booléenne de la sortie S assurant l'autorisation de fonctionnement de ce monte-charge.

Le monte-charge doit fonctionner (S passe à 1) à vide (cas $a=0$ et $b=0$ et $c=0$) ou pour les charges comprises entre 20 et 80 kg (cas $a=1$ et $b=1$ et $c=0$)

On peut écrire l'équation de la sortie $S = \bar{a}.\bar{b}.\bar{c} + a.b.\bar{c}$

3. Propriétés de l'algèbre de BOOLE.

Commutativité	$a.b = b.a$	$a + b = b + a$
Associativité	$a.(b.c) = (a.b).c$	$a + (b + c) = (a + b) + c$
Distributivité	$a.(b + c) = a.b + a.c$	$a + (b.c) = (a + b).(a + c)$
Éléments neutres	$a.1 = a$	$a + 0 = a$
Élément absorbant	$a.0 = 0$	$a + 1 = 1$
Complément	$a.\bar{a} = 0$	$a + \bar{a} = 1$
Idempotence	$a.a = a$	$a + a = a$
Théorème de De Morgan	$\overline{a.b} = \bar{a} + \bar{b}$	$\overline{a + b} = \bar{a}.\bar{b}$

Exemples d'utilisation de l'algèbre de Boole pour simplifier des expressions logiques.

$$(c + b).c = c.c + b.c = c + b.c = c.(1 + b) = c$$

$$a.(\bar{a} + b) = a.\bar{a} + a.b = a.b$$

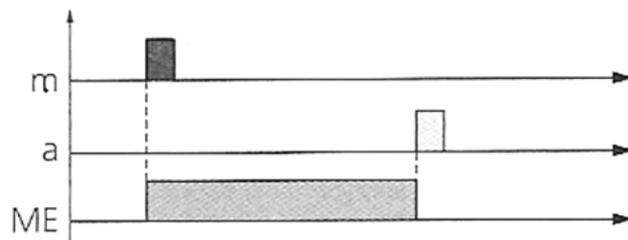
I. REPRESENTATION DES FONCTIONS LOGIQUES

On peut représenter une fonction logique par :

1. Une phrase (S est vrai si a et b sont vrais)
2. Une équation logique ($S = a.b$)
3. Une table de vérité

a	b	S
0	0	0
0	1	1
1	1	0
1	0	1

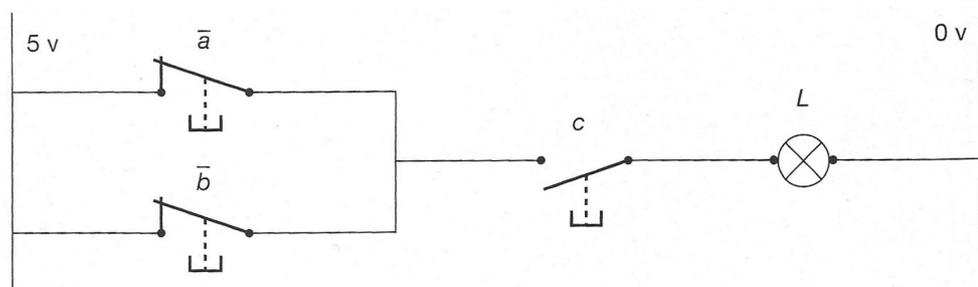
4. Un chronogramme



5. Un schéma électrique.

Ils sont réalisés par des contacts électriques commandés manuellement (poussoir) ou électriquement (relais).

Exemple 1.



On distingue deux sortes de contacts : « à ouverture » (Non) » et « à fermeture » (Oui).

- ✓ Si $c = 1$, le courant passe dans le contact. Si $c = 0$, le courant ne passe pas.
- ✓ Si $a = 0$, le courant passe dans le contact. Si $a = 1$, le courant ne passe pas.

La fonction représentée est : $L = c.(a + b)$

- ✓ La fonction « et » est réalisée par des contacts en série.
- ✓ La fonction « ou » est réalisée par des contacts en parallèles.

6. Un logigramme.

Un logigramme est une association d'opérateurs logiques décrivant une équation logique.

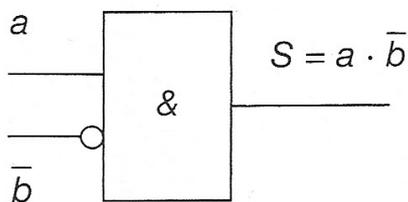
Liste (non exhaustive) des opérateurs logiques :

Symbole	Equation	Opérateur	Symbole	Equation	Opérateur
	$S = a$	Identité		$S = \bar{a}$	Non
	$S = a \cdot b$	ET		$S = \overline{a \cdot b}$	NAND
	$S = a + b$	OU		$S = \overline{a + b}$	NOR
	$S = a \oplus b$	OU exclusif		$S = \bar{a} + b$	a implique b

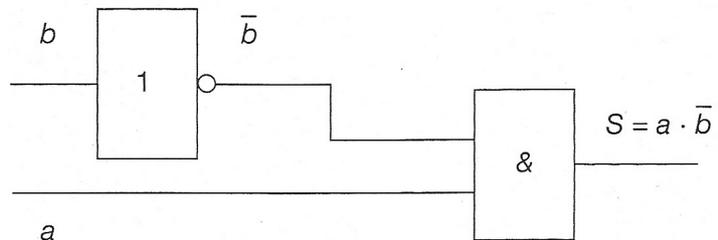
Le logigramme d'une fonction logique n'est pas unique, il dépend des contraintes technologiques imposées.

Exemple : On veut réaliser la fonction $S = a \cdot \bar{b}$

Solution 1 : En utilisant toutes les fonctions logiques disponibles



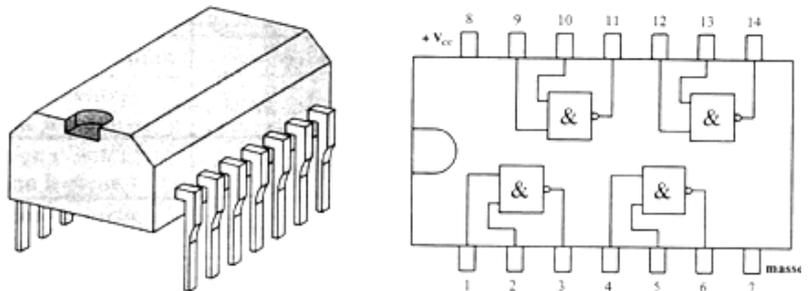
Solution 2 : En utilisant uniquement les fonctions de base ET, OU, NON



Dans certains cas, on se voit imposer l'utilisation unique des cellules NOR ou NAND

En effet, toute fonction peut être réalisée en utilisant uniquement des cellules NOR ou NAND (cellules dites universelles).

Cela permet de réaliser une fonction logique en utilisant qu'un seul _____ de cellules.



Il faut alors réorganiser la fonction (en utilisant le théorème de De Morgan) pour faire apparaître que des NOR ou que des NAND.

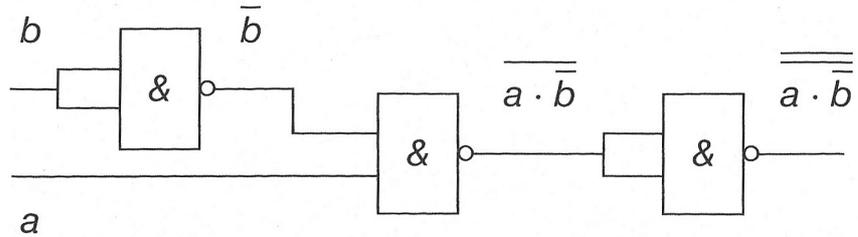
Remarque.



Solution 1 :

En utilisant que des NAND.

$$S = a \cdot \bar{b} = \overline{\overline{a \cdot \bar{b}}} = \overline{\overline{a} \cdot \overline{\bar{b}}}$$

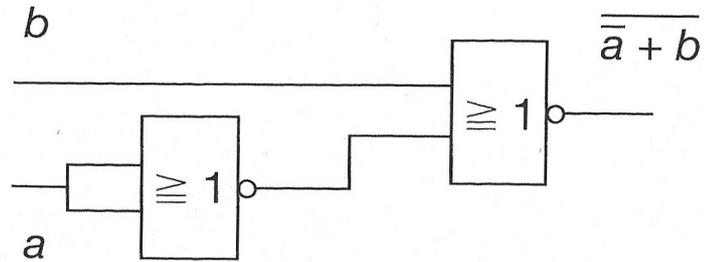


Solution 2 :

En utilisant que des NOR.

$$S = a \cdot \bar{b} = \overline{\overline{a \cdot \bar{b}}} = \overline{\overline{a} + \overline{\bar{b}}}$$

$$S = \overline{(a + a) + b}$$



COURS LOGIQUE 1 : CODAGE

Historique

- ✓ 35000-20000 ans av JC Des entailles sur des os
- ✓ 6000 ans av JC Des objets utilisés pour compter des dizaines et des centaines
- ✓ Grec et Romain Utilisation des chiffres romains (problème les calculs)
- ✓ 5 ème siècle Utilisation du zéro pour positionner les chiffres (en Inde)
- ✓ 12 ème siècle Utilisation du zéro en Europe

I. Code binaire « naturel ».

Le système de numérotation communément utilisé est le système décimal.

Les systèmes automatisés (et informatiques) utilisent le système binaire naturel (0 ou 1).

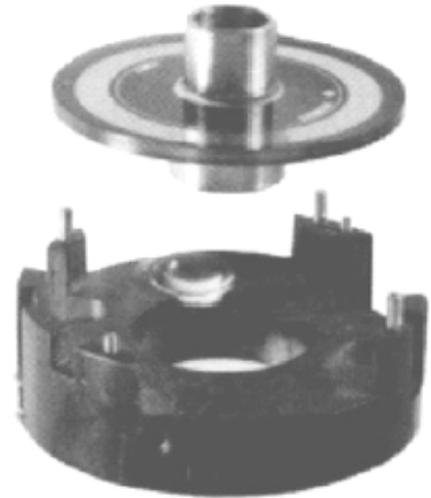
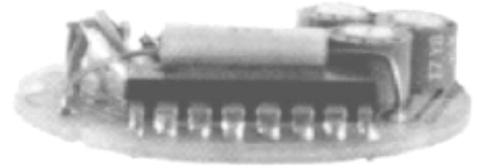
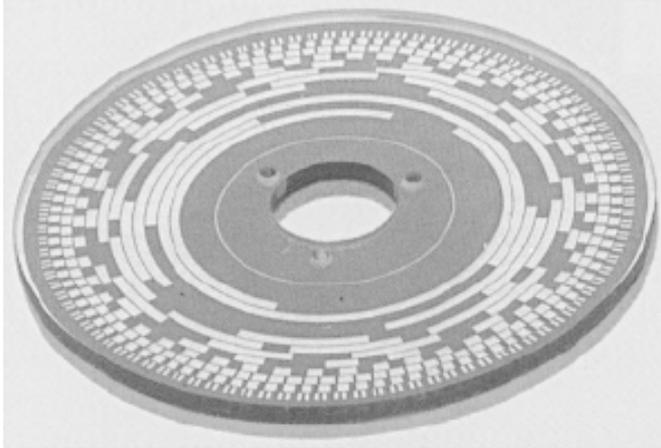
Base 10 (Décimal)	Base 2 (Binaire)	Base 16 (Hexadécimal)
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9

Base 10 (Décimal)	Base 2 (Binaire)	Base 16 (Hexadécimal)
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13

Le « codeur » numérique de position angulaire

Tous les codeurs de ce type sont construits autour d'un disque divisé en « n » pistes concentriques.

Chaque piste comporte des « cases » qui peuvent être laissées transparentes ou rendues opaques.



III. Transcodage.

Le transcodage consiste à passer d'un code à un autre.

Exemple.

Passage du code binaire réfléchi (code Gray) au code binaire naturel.

On cherche à exprimer (abc) en fonction de (xyz).

Décimal N	Binaire naturel (a b c)	Binaire réfléchi (x y z)
0	0 0 0	0 0 0
1	0 0 1	0 0 1
2	0 1 0	0 1 1
3	0 1 1	0 1 0
4	1 0 0	1 1 0
5	1 0 1	1 1 1
6	1 1 0	1 0 1
7	1 1 1	1 0 0

On trouve après observation attentive :

$$a = x$$

$$b = x \oplus y$$

$$c = x \oplus y \oplus z$$

Systemes à Evénements Discrets SED

I. 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.

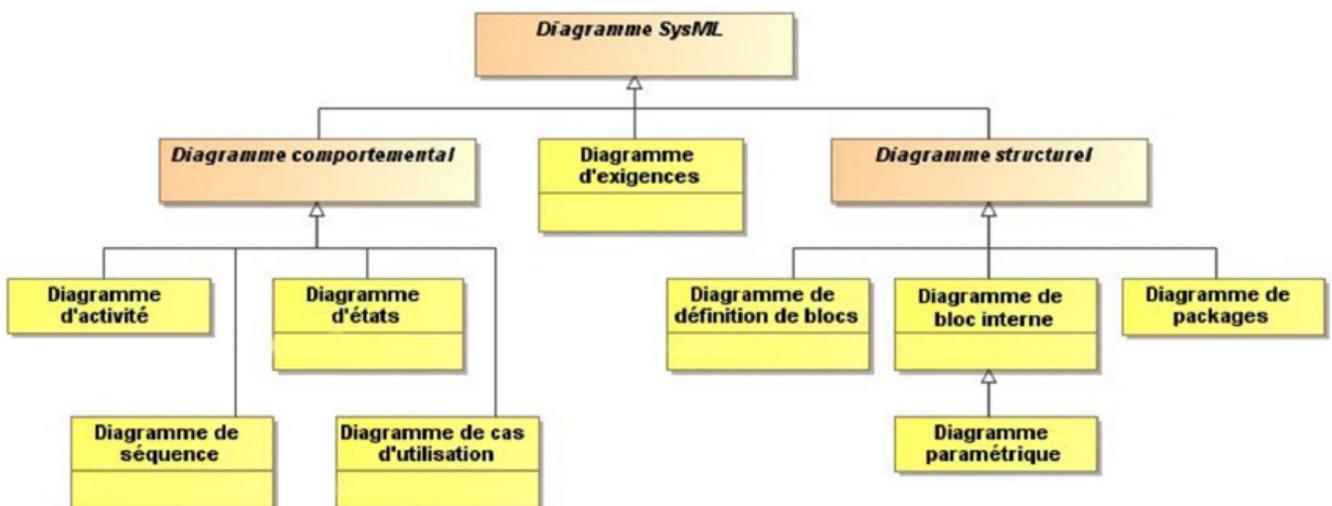
Les diagrammes SysML permettent de représenter :

- ✓ 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.

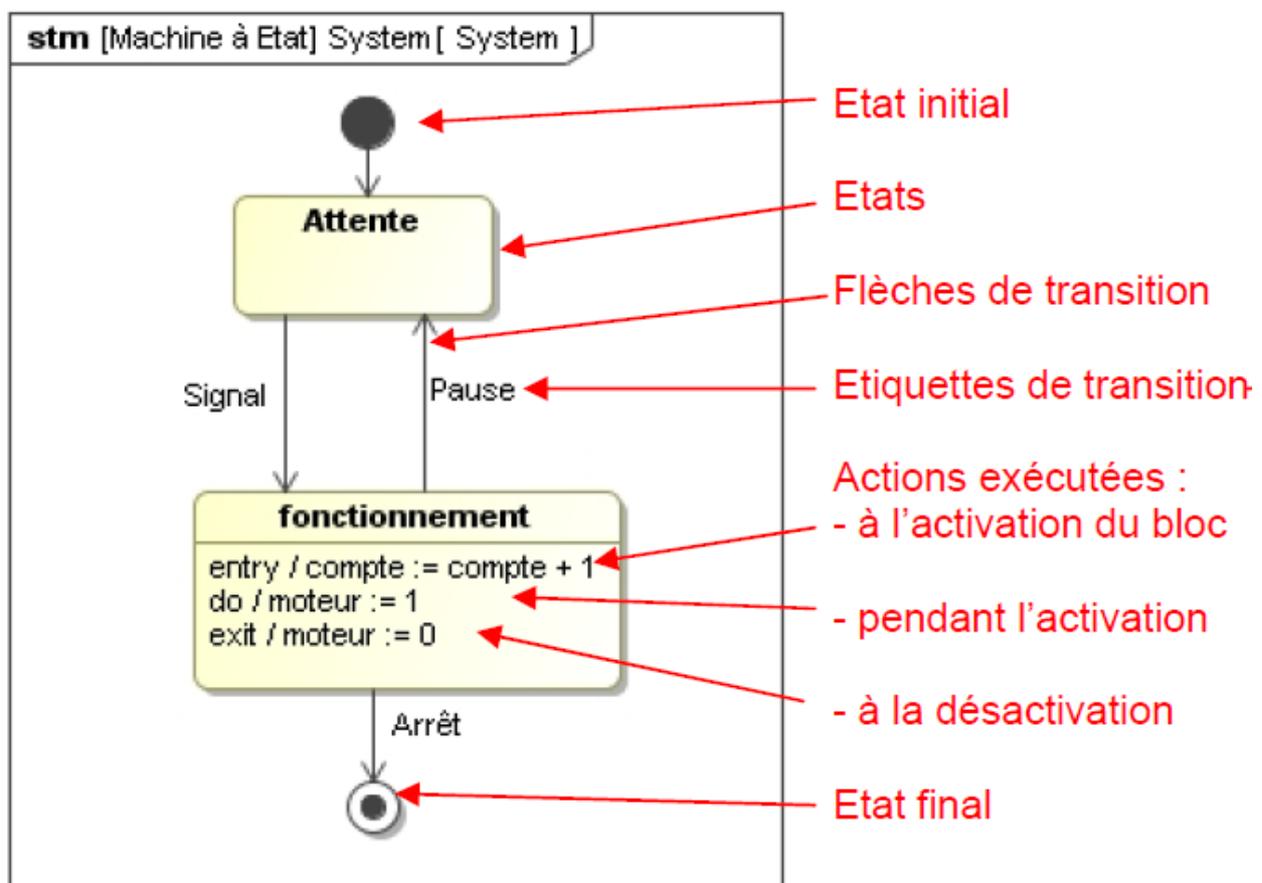
Nous allons voir dans ce cours le diagramme d'états.

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

Structure d'un diagramme d'états :

- ✓ Les états du système sont représentés par des rectangles.
- ✓ Aux états peuvent être associées des actions et 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, des conditions et des actions peuvent être liés aux transitions.

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



Etat :

Un état représente une période de la vie du système, pendant laquelle il peut accomplir des actions, une activité et il attend un ou des événements.

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.



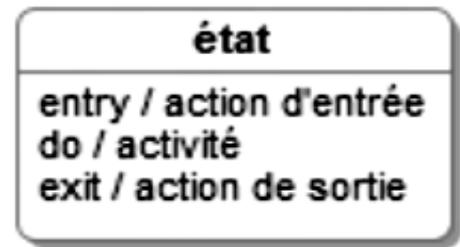
Il existe deux états spécifiques : L'état initial : ● et l'état final: ⊙ (Il est possible d'utiliser plusieurs états finaux).

Activité et action :

Activité :

Une activité est associée à un état, spécifiée par le mot « do ».

Elle prend un certain temps et peut être interrompue.



Action :

Une action est associée à un état, spécifiée par le mot « entry » ou « exit ».

Une action est instantanée, elle ne peut pas être interrompue.

Exemples :

- ✓ Activité : Faire tourner un moteur « M », alimenter un vérin « V »...
- ✓ Action : Mettre le moteur en marche « M=1 », incrémenter une variable « c=c+1 »...

Transition :

Evénement [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 :

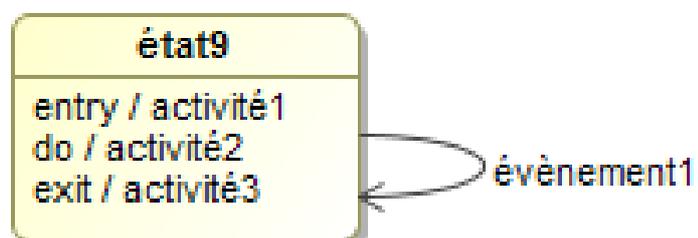
- ✓ Un événement déclencheur.
- ✓ Une condition.
- ✓ Une action.

Principe :

- ✓ La transition peut être franchie que si l'état source est actif.
- ✓ La transition n'est franchie que si l'événement devient vrai (passage de 0 à 1), 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 événement1 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.



Événement :

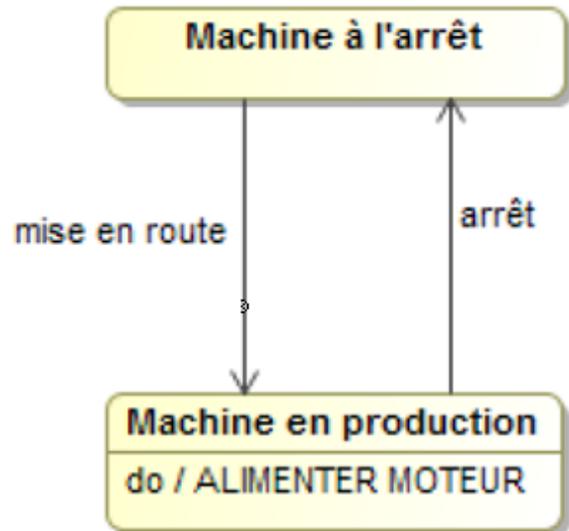
Variable associée à une transition, elle permet le passage de la transition lors de son changement d'état, c'est-à-dire du passage de 0 à 1.

Si l'événement est noté \bar{a} ou $!a$, le passage de la transition se fera lors du changement d'état de 1 à 0.

On parle de front montant et de front descendant.

Exemples :

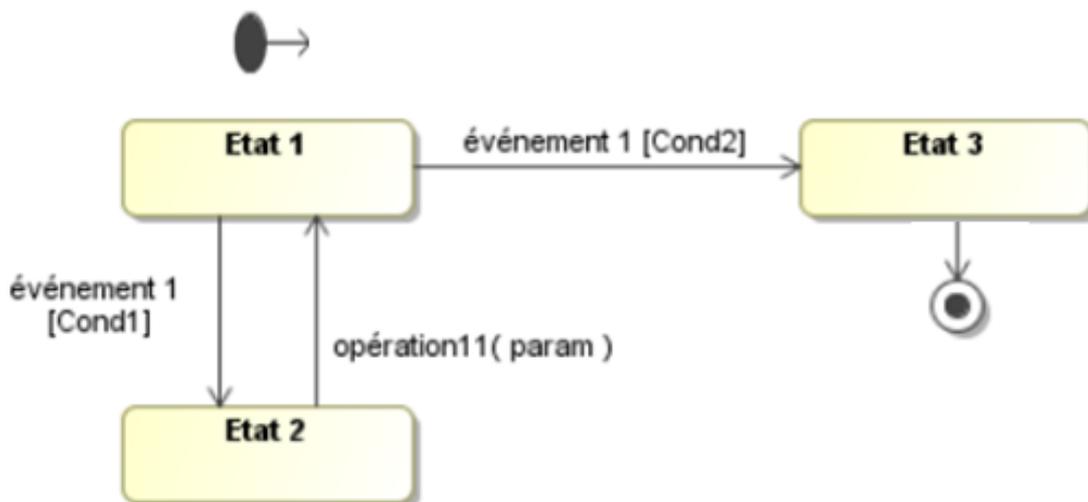
- ✓ Appui sur un bouton.
- ✓ Détection d'un objet.
- ✓ Changement d'une variable interne « when (N=10) ».
- ✓ Comptage du temps « after (90s) » ou « at(10:00) ».



Condition :

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

Elle se note entre crochets.



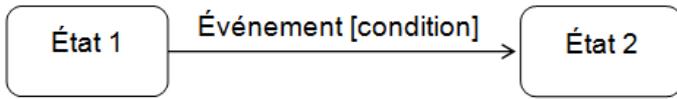
Remarques :

- ✓ 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.

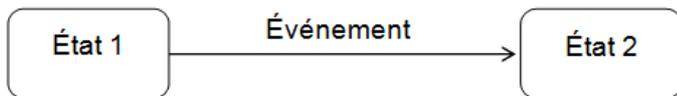
Synthèse sur l'évolution d'un état :



L'état 1 est actif.

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

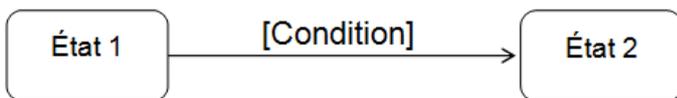
L'éventuelle activité de l'état 1 est interrompue.



L'état 1 est actif.

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

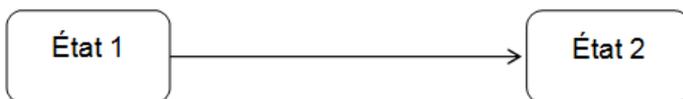
L'éventuelle activité de l'état 1 est interrompue.



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.

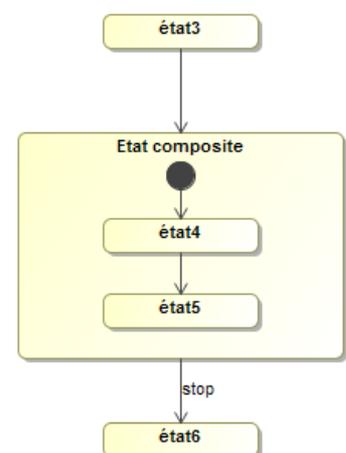
L'activité doit donc ne pas être continue mais avoir un début et une fin.

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 événement 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 radoréveil

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.

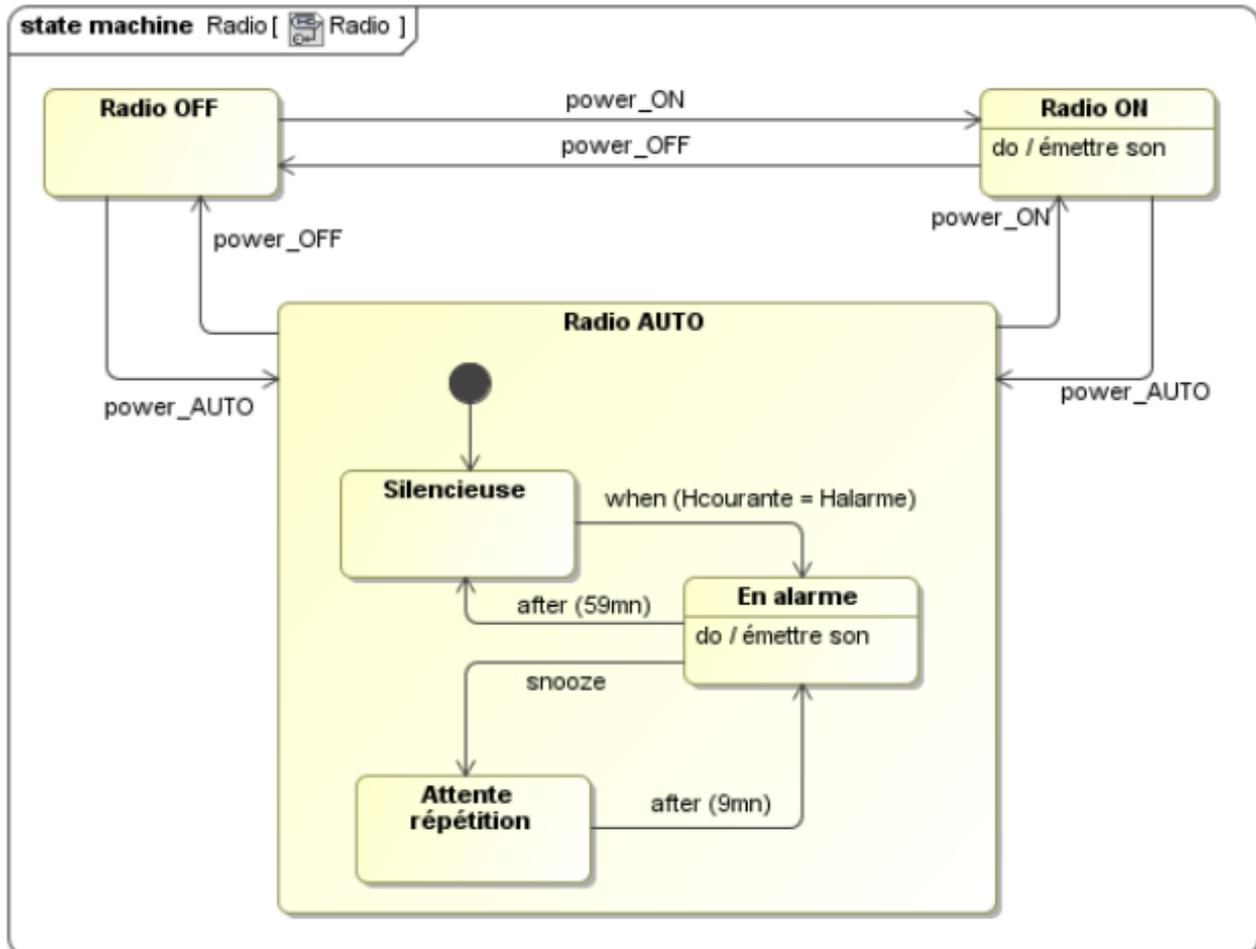
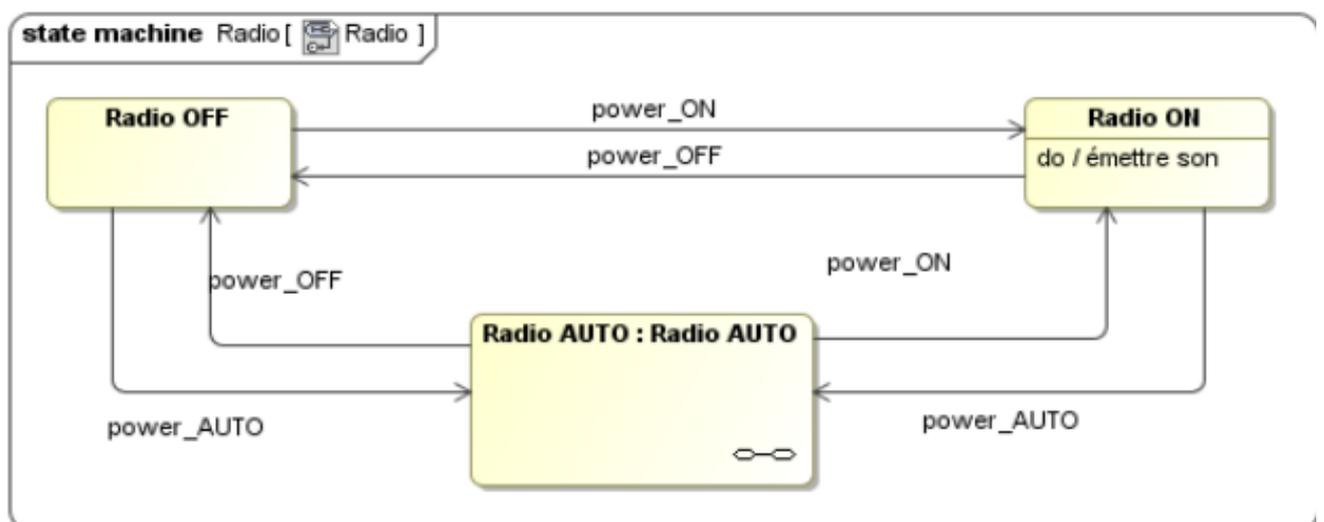
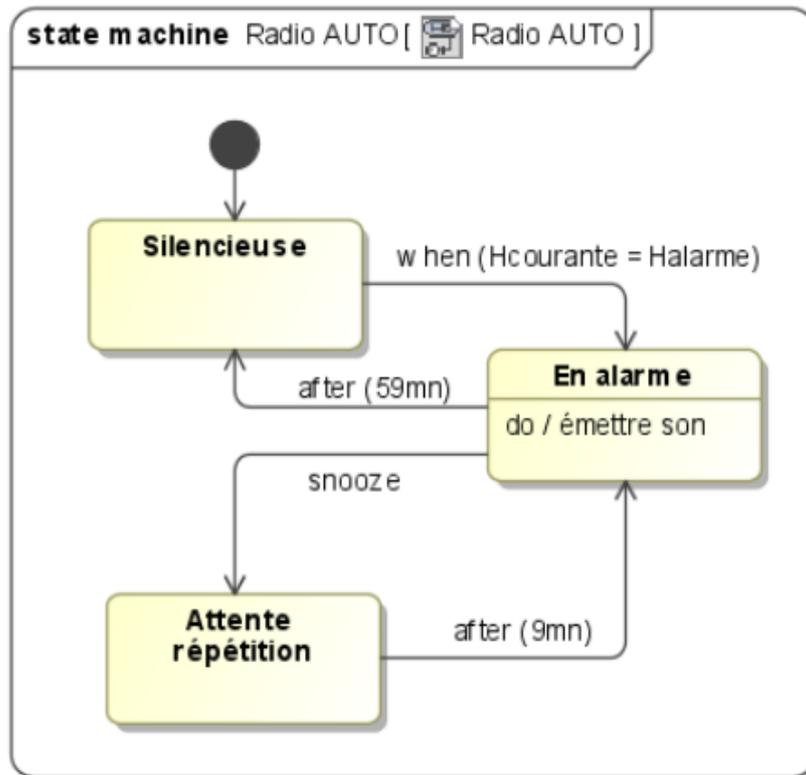


Diagramme d'états d'un radoréveil nécessitant la décomposition des sous états au sein de l'état « radio AUTO ».

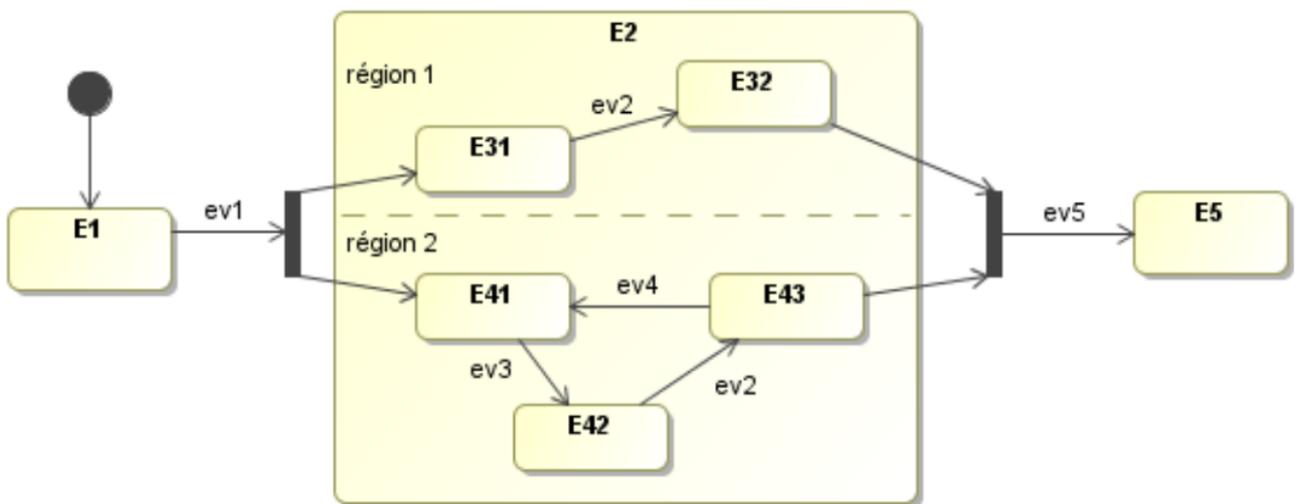




Structure parallèle (état orthogonal) :

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 ».

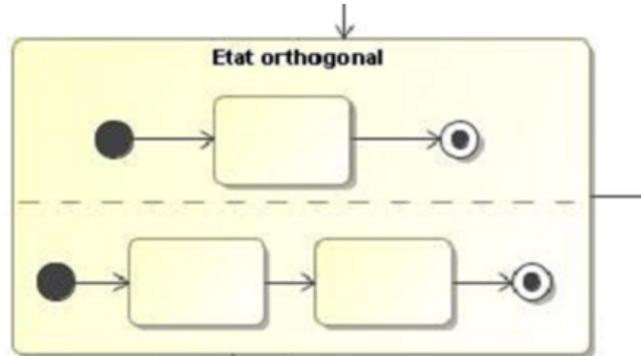
Cela signifie qu'il est à la fois dans les états disjoints « E31 » et « E41 » ;

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.

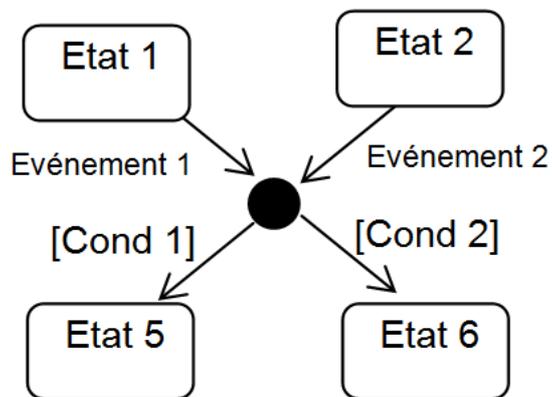
La sortie de l'état orthogonal se fait dès que toutes les régions ont atteint leur état final (équivalent à la représentation précédente).



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.

