

# 1 INTRODUCTION

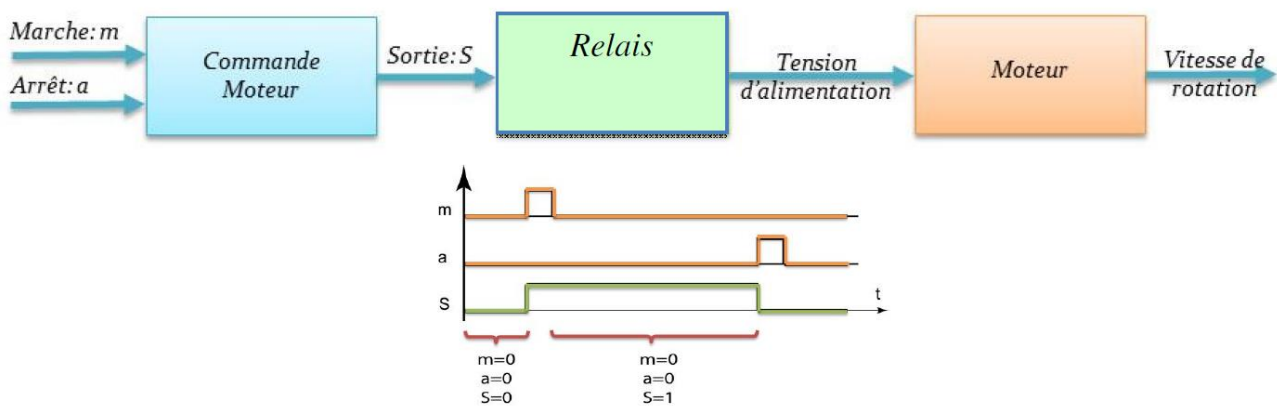
Les systèmes à événements discrets (SED) permettent de modéliser la succession des différentes étapes d'évolution. Ce sont des systèmes séquentiels dans lesquels une même cause peut produire des effets différents. Un effet peut rester maintenu alors même que sa cause a disparu (notion de mémoire).

**Définition :** ...



Exemple de système séquentiel :

- Commande séquentielle d'un moteur électrique On remarque sur le chronogramme ci-dessous que la sortie S peut présenter une valeur différente (0 ou 1) pour une configuration identique des entrées m et a : le système ne peut donc pas être combinatoire. Il faut donc ajouter une variable d'état supplémentaire, notée par exemple s (état antérieur de S).



- Affichage des numéros d'étage dans un ascenseur.

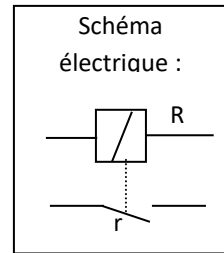
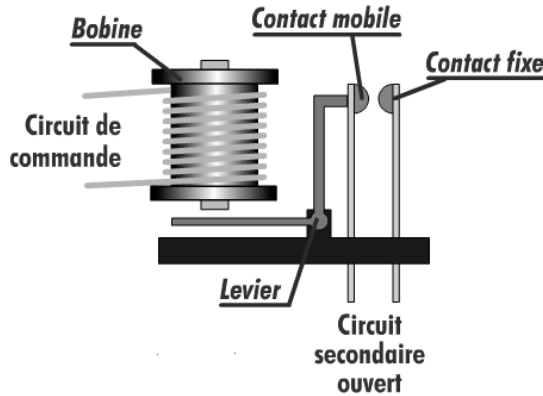
Système combinatoire : l'affichage est correct seulement quand l'ascenseur est arrêté à l'étage  
 Système séquentiel : l'affichage est mémorisé pendant que l'ascenseur se déplace entre 2 étages.



## 2 COMPOSANTS PERMITTANT LA COMMANDE SEQUENTIELLE

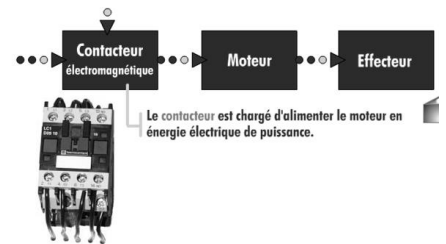
### 2.1 Relais

Un relais est un circuit électrique permettant la commutation d'un circuit en fonction d'un signal extérieur.

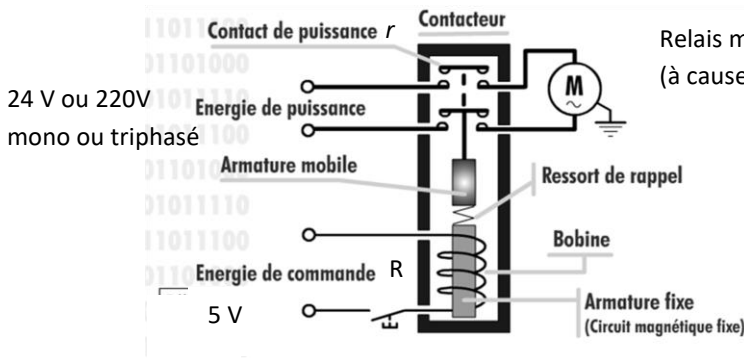


#### 2.1.1 Principe

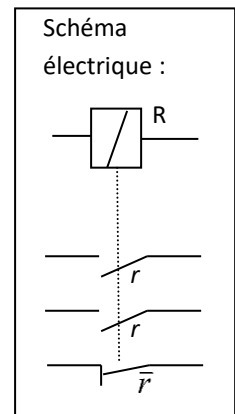
Le relais participe à la commande d'une chaîne électrique (pré-actionneur)



#### 2.1.2 Séparation des circuits de commande et de puissance

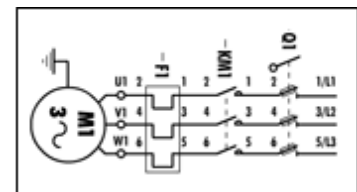


Relais monostable (à cause du ressort de rappel)



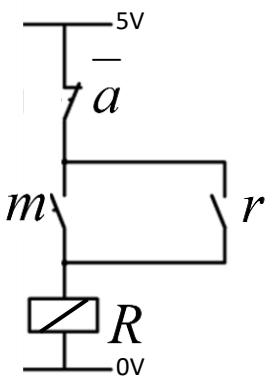
#### 2.1.3 Multiplication des contacts

Exemple schéma électrique d'alimentation d'un moteur triphasé

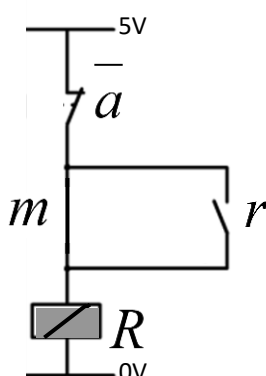


#### 2.1.4 Auto-maintien (fonction mémoire)

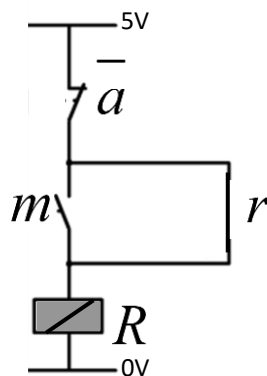
Exemple : Equation logique :  $R = \bar{a} \cdot (m + r)$  priorité à l'arrêt



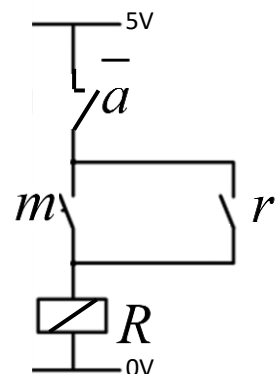
R =



R =



R =

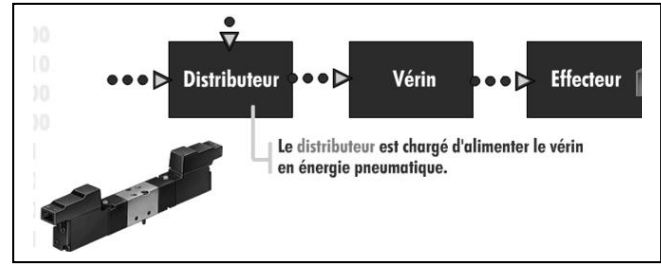


R =

## 2.2 Distributeurs : commande d'une chaîne pneumatique (pré-actionneur)

### 2.2.1 Principe

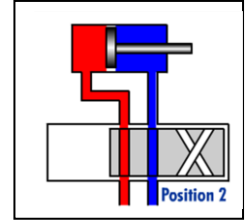
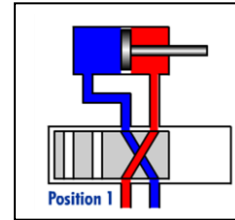
Le rôle du distributeur pneumatique (ou hydraulique) est de distribuer l'air (ou l'huile) vers le vérin en déplaçant le tiroir en translation.



Dénomination du distributeur (ici 4 / 2) :

4 : nombre d'orifices ou de tuyaux raccordés sur le distributeur

2 : nombre de positions possibles du tiroir

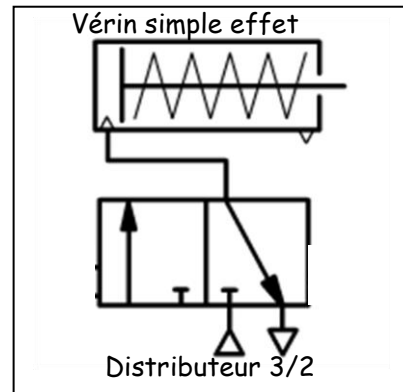


Exemple de câblage pour alimenter :

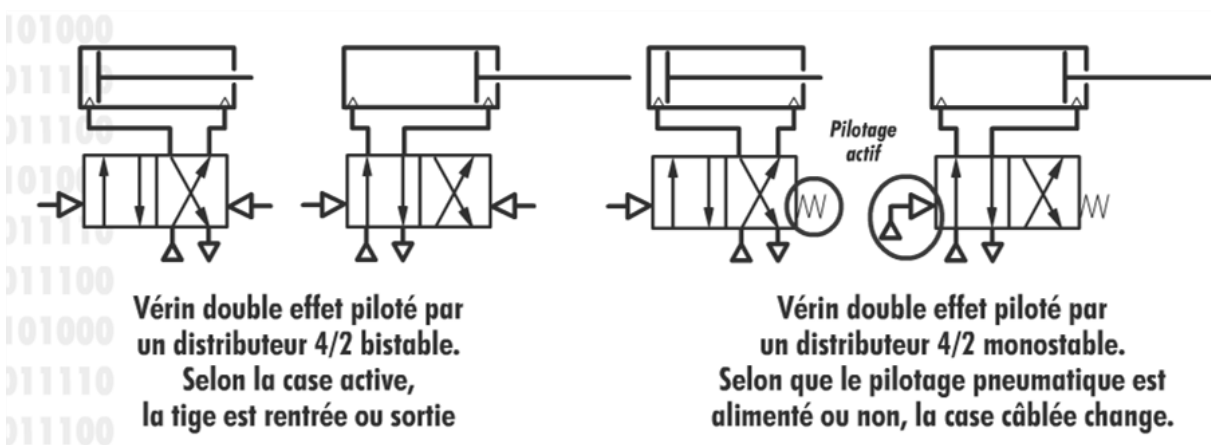
un vérin double effet (deux entrées d'air)

et

un vérin simple effet (une seule entrée d'air)



### 2.2.2 Commande des distributeurs



Pour commander le déplacement du vérin avec le **distributeur bistable**, il faut **deux ordres** : un pour Avancer, et un autre pour Reculer (effet mémoire) ;

Par contre **il suffit d'un seul ordre** (qui sera présent ou non) pour Déplacer le vérin avec le distributeur **monostable** (pour mémoriser l'ordre, il faudra utiliser une autre méthode).

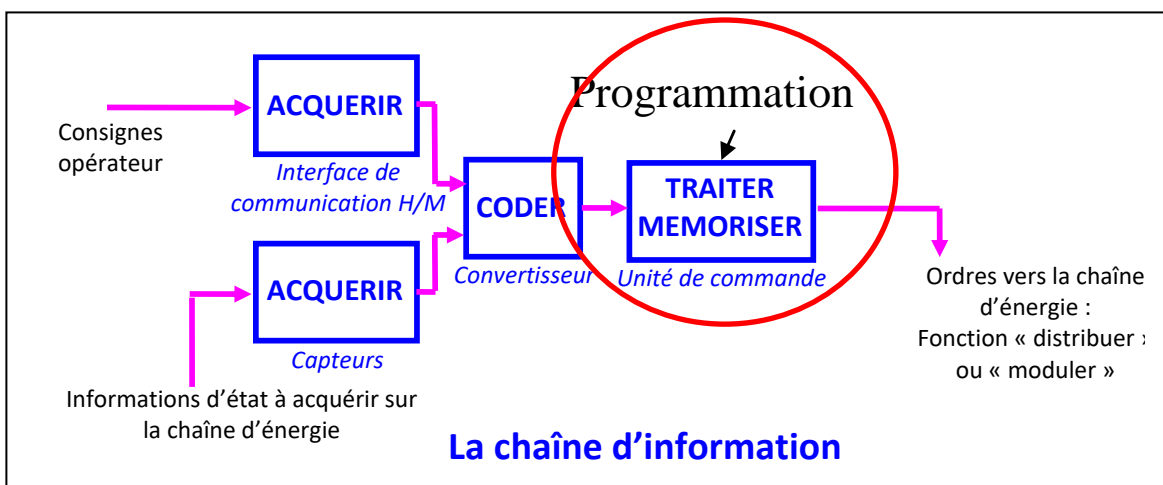
### 3 DESCRIPTION DES SYSTEMES A EVENEMENTS DISCRETS (SED)

#### *Comment décrire (et programmer) le comportement séquentiel d'un système ?*

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

- L'outil « diagramme d'activité » Sysml ; (hors programme, ne décrit pas les causes d'évolution)
- L'outil « diagramme de séquence » Sysml ;
- L'outil algorithmique (ou algorigramme) ;
- L'outil « diagramme d'états » Sysml (ou graphe d'états) ;

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

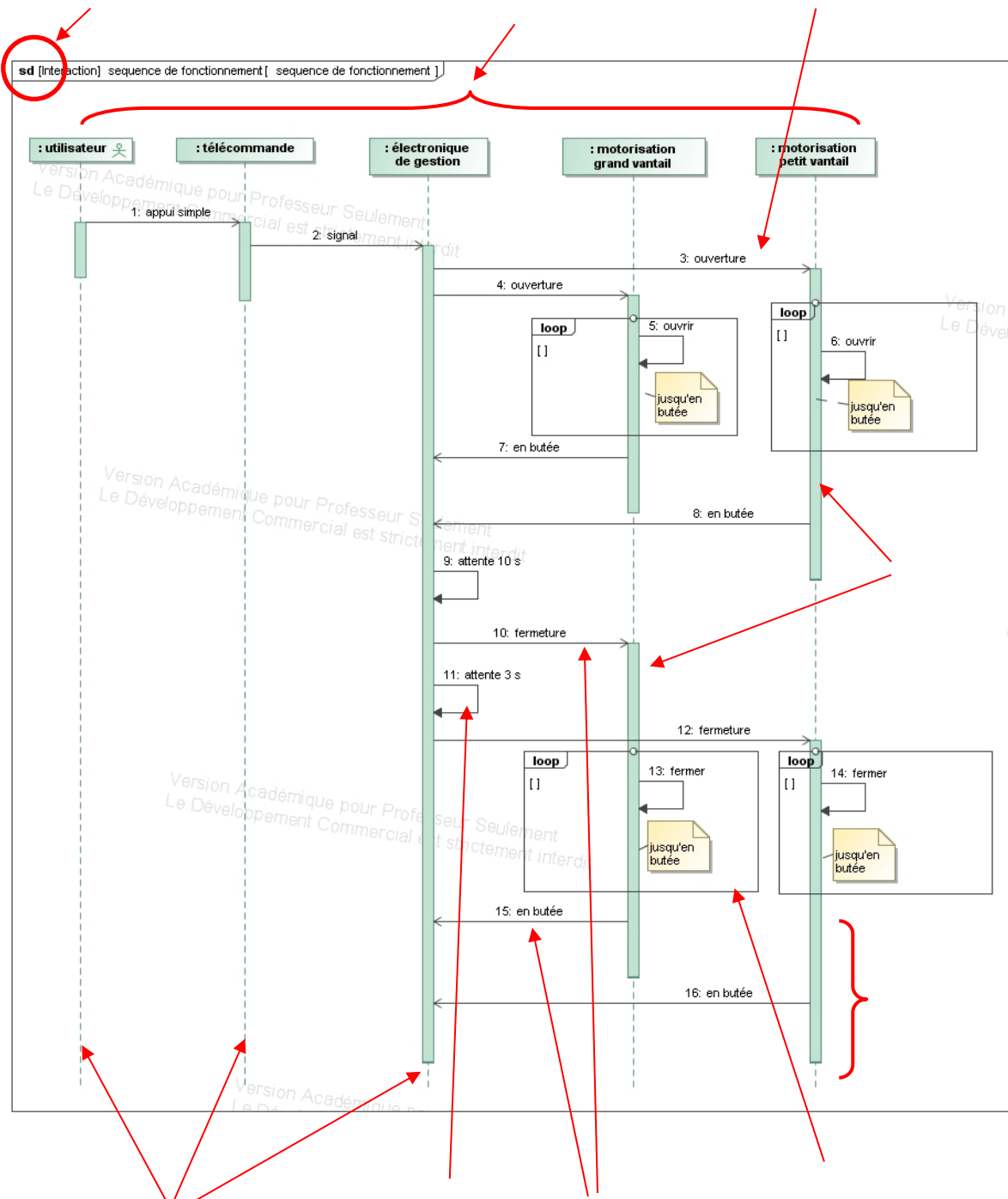


#### 3.1 Description comportementale par le « Diagramme de séquences »

(Le diagramme de séquence est un des outils de la norme « Sysml »)

**a) Objectif :** décrire dans l'ordre chronologique les interactions entre les acteurs et composants d'un système, ou entre les composants eux-mêmes ;

**b) Syntaxe de base :** (sur l'exemple = fonctionnement d'un portail télécommandé)

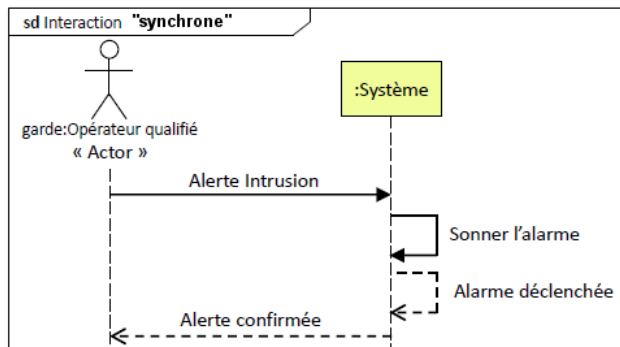


(\*) dans le cas où « l'électronique de gestion » participe à l'action de la boucle « loop », cette boucle doit recouvrir aussi la ligne de vie de « l'électronique de gestion ».

Exemple de question à laquelle on peut désormais répondre : quel est la chronologie de fermeture des vantaux ? .....

**c) Compléments de syntaxe du diagramme de séquences :**

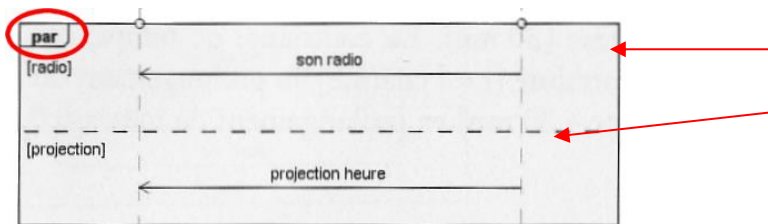
- les messages dits « synchrones » : l'émetteur est bloqué, en attente d'une réponse du récepteur :



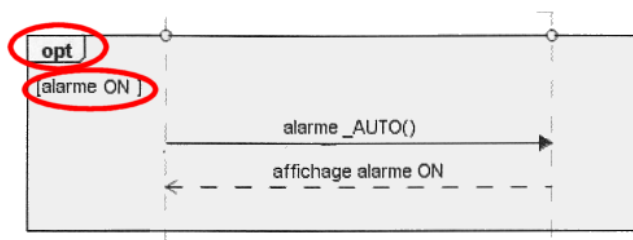
- l'action **maintenue** (boucle « loop ») : la durée peut être précisée



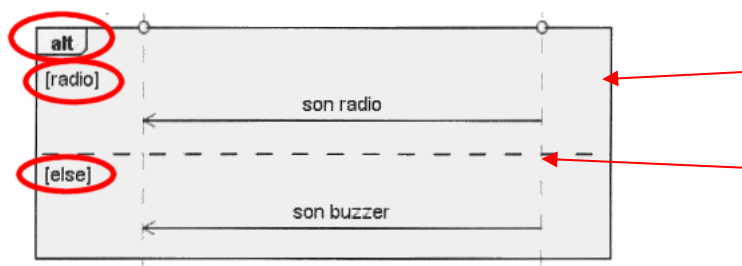
- les interactions en « parallèle » : **les interactions sont exécutées simultanément** :



- les interactions optionnelles : **l'interaction n'existe que si la condition fournie est vraie** :



- les interactions « alternatives » : **seule l'interaction possédant la condition vraie s'exécute** :



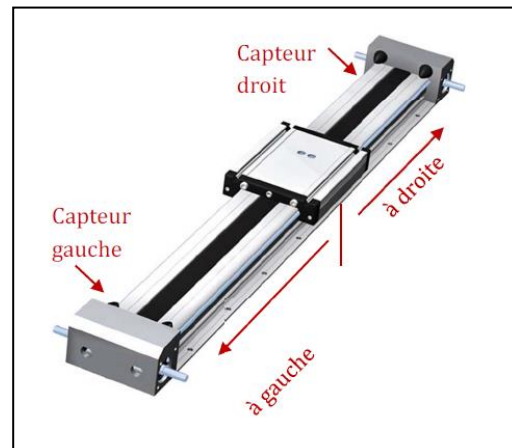
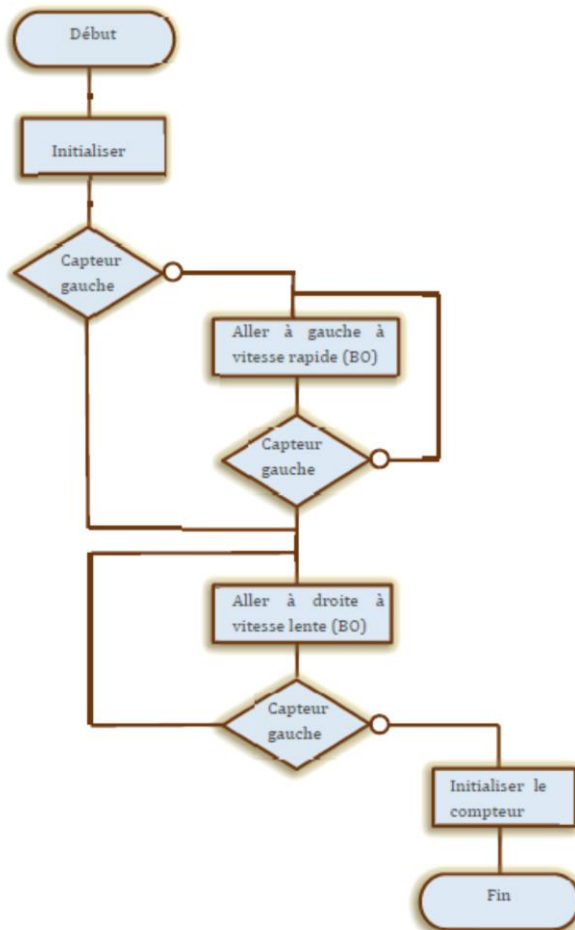
### 3.2 Description par « l'algorithme »

Syntaxe générale :

SYMBOLE	DÉSIGNATION	SYMBOLE	DÉSIGNATION
	début ou fin d'un algorithme		<b>Test ou Branchement conditionnel</b> décision d'un choix parmi d'autres en fonction des conditions
	symbole général de « traitement » opération sur des données, instructions, ... ou opération pour laquelle il n'existe aucun symbole normalisé		<b>sous-programme</b> appel d'un sous-programme
	entrée / sortie de données		<b>Liaison</b> Les différents symboles sont reliés entre eux par des lignes de liaison. Le cheminement va de haut en bas et de gauche à droite. Un cheminement différent est indiqué à l'aide d'une flèche

Exemple : l'axe linéaire à codeur incrémental ;

Description de la procédure de prise d'origine :



Conseil : « décomposer pour mieux résoudre »

Il est préférable de réaliser de petits algorithmes (sous programmes) appelés parfois « macros » ou « fonctions » décrivant chacun une tâche simple, puis de faire un algorithme plus général qui fait appel à ces sous-programmes.

### 3.3 Description et programmation par l'outil « Graphe d'état »

(Le graphe d'états est un des outils de la norme « Sysml »)

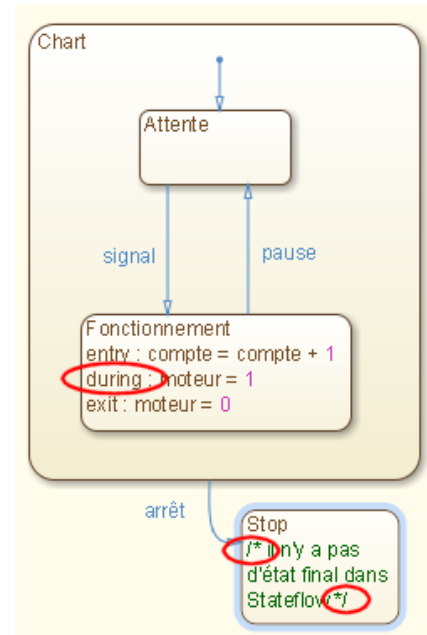
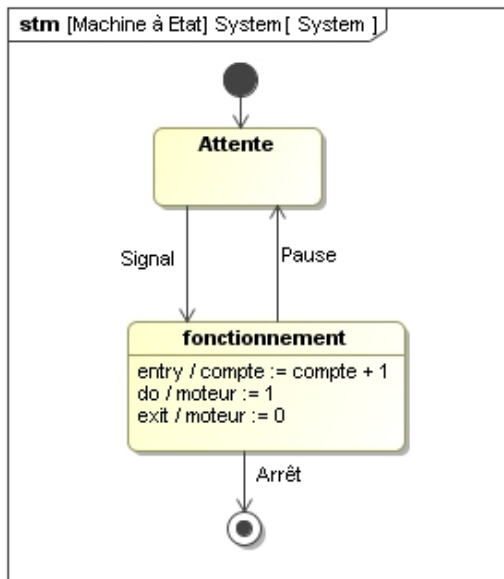
Le « graphe d'états » ou « diagramme d'états » ou *state machine* « stm » permet de décrire l'évolution d'un système en traçant de manière codifiée, l'interaction entre les événements qui se produisent (à l'extérieur ou à l'intérieur du système) et son comportement décrit par les actions qu'il effectue en conséquence.

3.3.1 **Eléments de base**

**a) Syntaxe**

Sysml

cas Matlab-Stateflow



**b) Les états**

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 » (voir plus loin).

Mots réservés dans les états :

Actions effectuées	Sysml	Cas Matlab-Stateflow
à l'activation de l'état	<b>entry /</b>	<b>entry :</b> ou <b>en :</b>
en continu tant que l'état est activé	<b>do /</b>	<b>during :</b> ou <b>du :</b>
à la désactivation de l'état	<b>exit /</b>	<b>exit :</b> ou <b>ex :</b>

**c) Les transitions**

Une transition représente le passage instantané d'un « état source » vers un « état destination » (« instantané » car on considère négligeable la durée d'un cycle du calculateur). On trouve les événements déclencheurs d'évolution dans les « étiquettes de transition » ; Dans l'usage, les « étiquettes de transition » sont appelées « transitions »

Syntaxe de l'écriture des transitions :

Sysml	Cas Matlab-Stateflow
<b>événement [condition] / action</b>	<b>événement [condition] {action sur condition} / action sur transition</b>

**Principe :**

- La transition n'est évaluée que si l'état source est actif ;
- La transition n'est franchie que si l'événement est vrai **et** si la condition est vraie ;
- L'action (lorsqu'elle est présente) est exécutée pendant le franchissement (sauf action sur condition de matlab).

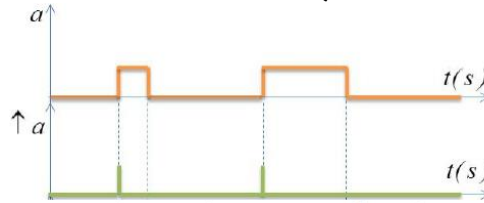


Exemples possibles de transitions :

événement ou condition	Sysml	Cas Matlab-Stateflow
1 variable devient vraie	$a := 1$	$a = 1$
1 variable devient vraie	$a$	$a$
2 variables vraies	$a.b$	$a \& b$ ou $a \&\& b$
L'une ou l'autre vraie	$a + b$	$a   b$ ou $a    b$
Le complément est vrai	$/ a$	$! a$
Test : 2 variables sont égales ?	$a = b$	$a == b$
Condition temporelle de retard	<b>after(2 s)</b>	<b>after(2,sec)</b>
Condition temporelle	<b>at(13 :00)</b> (horaire)	<b>at(10,ticks)</b> (tick = cycle de calcul) déclenchée par l'activation de l'état source
Activité d'un autre état	<b>[in ETAT]</b>	<b>in(ETAT)</b>

Un **évènement** est un changement qui doit être pris en compte dans la description par diagramme d'état.

Un évènement se produit à une date précise qui est l'occurrence de cet évènement. **La durée d'un évènement est nulle et il ne peut pas être mémorisé.** Un évènement 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'évènement est une condition nécessaire au franchissement.

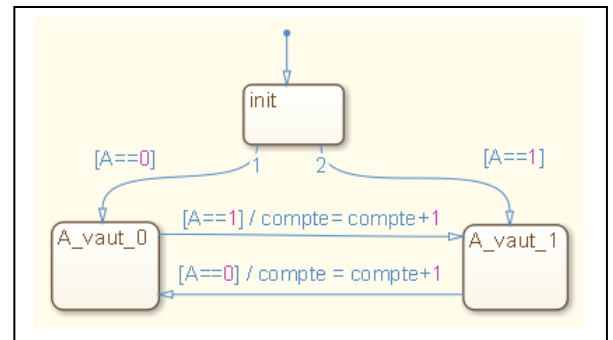
Pour désigner un évènement correspondant au front descendant d'une variable  $a$ , on utilisera l'évènement  $\bar{a}$  (dans certains logiciels comme Matlab, on n'associe à la définition de l'évènement les qualificatifs « rising » ou « falling »).

**Exemple de graphe avec action exécutée pendant le franchissement :**

*l'objectif est de compter le nombre de fronts montants et de fronts descendants qui se produisent au cours du temps :*



Nota : la présence d'actions dans les transitions nuit à la lisibilité des graphes d'états ; donc sauf cas très particuliers on évitera l'utilisation de l'action au franchissement.

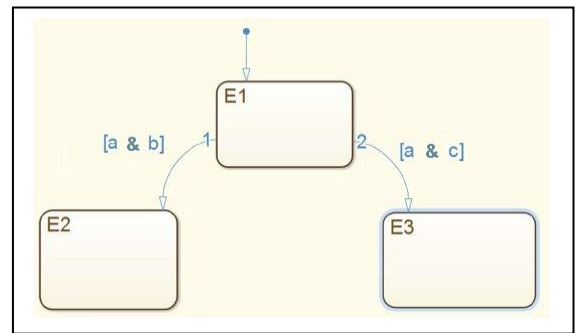
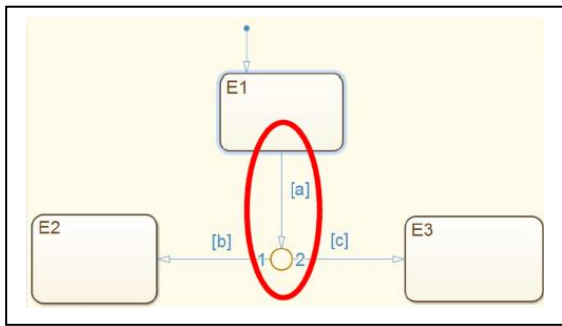


**d) les syntaxes particulières**

Transition avec « point de décision » (ou « jonction ») :

Sysml	Cas Matlab-Stateflow

La jonction est un **pseudo-état** utilisé par exemple pour factoriser des expressions booléennes associées à des conditions.

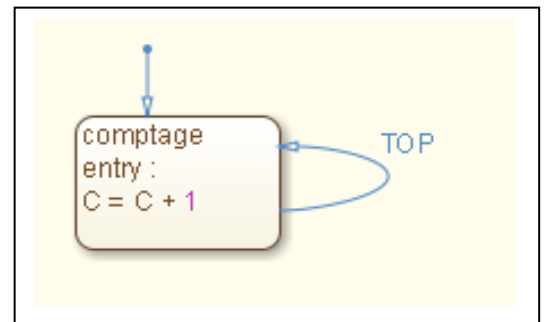


Exemple : Lorsque la condition [a] est vraie, la jonction devient active, mais E1 est désactivé seulement si [b] ou [c] deviennent vrais.

**Transition réflexive :**

L'état source et l'état pointé est le même ; il est désactivé puis aussitôt réactivé au pas de calcul suivant.

Exemple ci-contre : transition réflexive utilisée en comptage (de TOP)

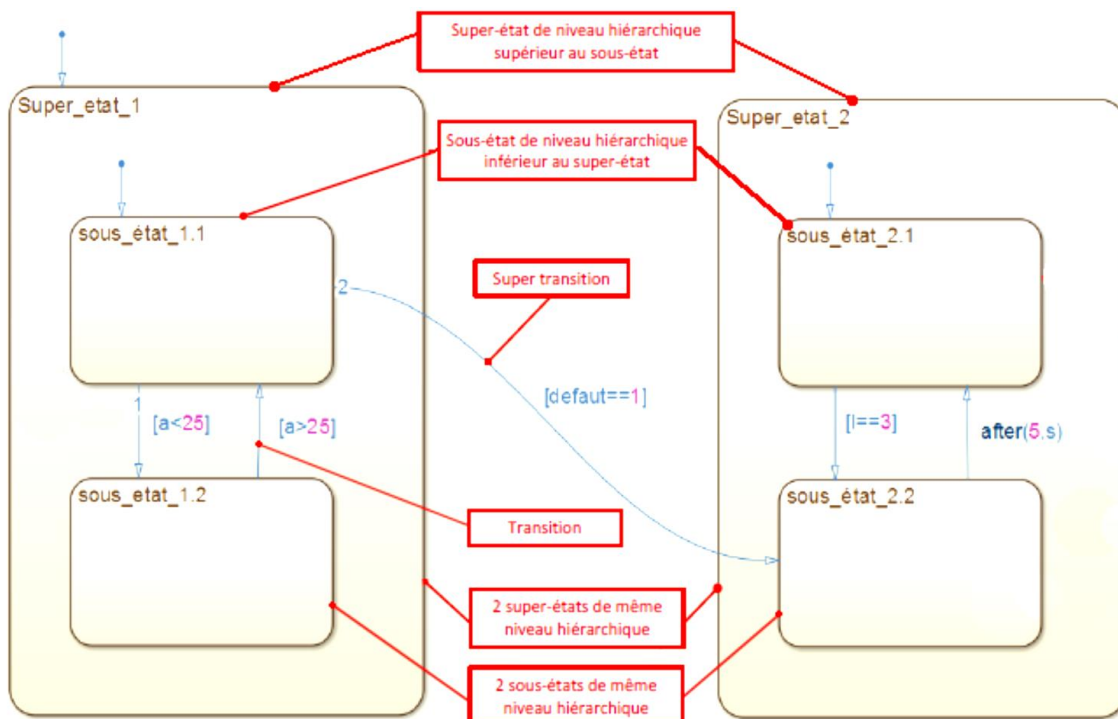


**3.3.2 Les états composites**

**a) Définition :** un état composite contient un ou plusieurs états.

**Appellations :** l'état composite peut être appelé « **super-état** » ; il contient des « sous-états » ;

**Exemple :**



Intérêt : Ils permettent :

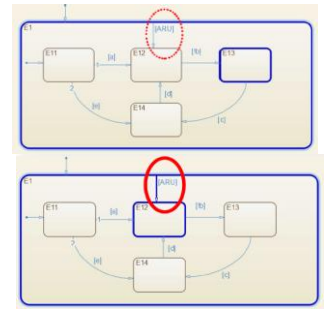
- de donner de la lisibilité en simplifiant la description des évolutions
- de gérer par zones séparées, les différentes parties d'une machine ;
- de fournir la possibilité de hiérarchiser les évolutions. ....

Représentations abrégées dans lesquelles les sous états sont masqués :

Sysml	Cas Matlab-Stateflow
 <p>état composite</p>	 <p>Subchart</p>

symbole

cadre épais



**b) Règles**

- un état composite contient un état initial qui est activé lorsque l'état composite est activé ;
- un sous-état ne peut être actif que si l'état composite dans lequel il se trouve l'est.
- Il ne peut y avoir qu'un seul état actif dans le sous-état et dans l'ensemble du graphe, sauf dans le cas des « structures à évolutions simultanées » (voir plus loin).

**c) les syntaxes particulières**


**Transition interne :** elle relie le cadre d'un état donné au cadre d'un état de niveau hiérarchique inférieur. Cette transition de niveau hiérarchique supérieur est évaluée prioritairement.

*Exemple : cas de forçage dans un état particulier, Lorsque la condition ARU est vraie, l'état E12 s'active*

**Etat « History » :**

il s'agit d'un pseudo-état qui mémorise l'activité d'un super-état (composite).

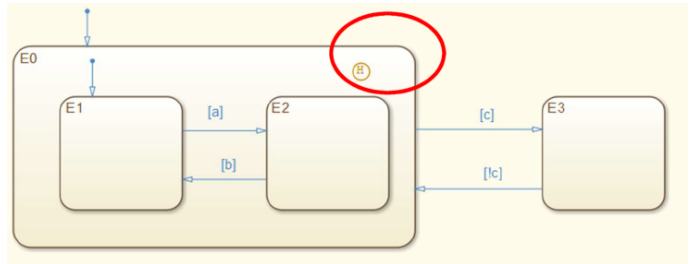
Lors d'une désactivation, puis d'une transition retour vers ce super-état, il retrouve son état antérieur.

Notation :  représente le « pseudo-état History » placé à l'intérieur du super-état ;

*Exemple : voir ci-contre.*

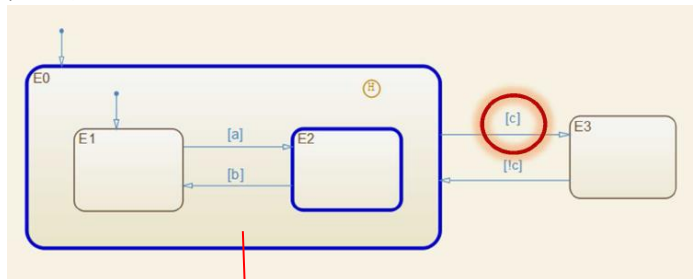
Explication :

1- le symbole « H » est présent dans le super-état E0 (composite) :



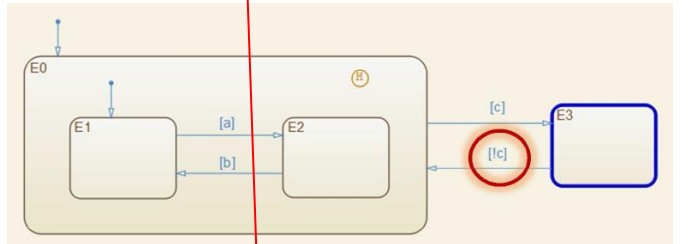
2- on suppose que l'évolution conduit à l'activation de l'état E2 ;

puis que la condition « c » est vraie :

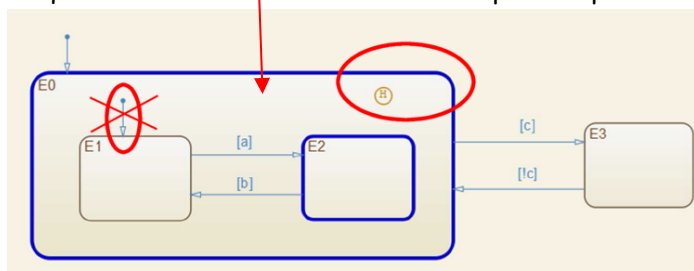


3- l'activation de E3 s'effectue normalement ;

puis on suppose que la condition « !c » devient vraie :

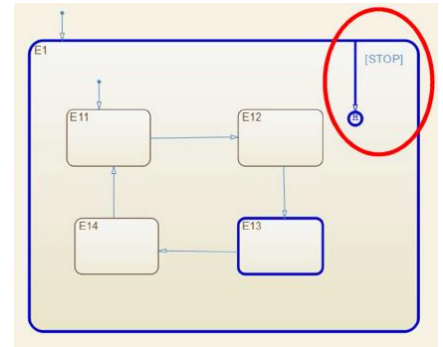


4- c'est l'état E2 mémorisé par « H » qui s'active, au lieu de E1 qui aurait été activé si « H » n'avait pas été présent.



Application particulière ci-contre :

Le « figeage » du super-état E1 - La présence d'une transition interne vers le pseudo-état history conduit ici au figeage (ou blocage) du graphe de l'état composite lors de l'activation de « STOP » ; dès que la condition « STOP » n'est plus vraie, l'état composite E1 reprend son mode d'évolution normal.



3.3.3 Les structures à évolutions simultanées : sous-états « parallèles »

Exemples et syntaxe

Sysml	Cas Matlab-Stateflow
<p style="text-align: center;"><b>ligne pointillée</b></p>	<p style="text-align: center;"><b>cadres pointillés</b></p>

**Définition :** des sous-états sont dit « parallèles » lorsqu'ils sont actifs simultanément ;

**Vocabulaire :**

- Le super-état (composite) qui contient les états « parallèles » est dit « orthogonal ».
- Des états parallèles sont aussi appelés « concurrents » (du mot anglais « concurrently » qui signifie « simultanément »)
- Lorsqu'un super-état (composite) n'est pas « parallèle », il est simplement « exclusif » (il ne doit y avoir qu'un seul état actif dans le sous-graphe).

**Conditions d'activation et de désactivation :**

Sysml	Cas Matlab-Stateflow
<p>Toutes les régions parallèles d'un état composite orthogonal doivent atteindre leur état final pour que l'état composite soit considéré comme terminé.                  La synchronisation est alors automatique et la transition de sortie de l'état composite est déclenchée. <b>ou</b>                  Il est également possible de représenter ce type de comportement au moyen de transitions « concurrente » constituées de barres de synchronisation "fork" et "join".                  Le graphe ci-dessous est une représentation équivalente à la précédente :</p>	<p><b>Exemple 1 :</b></p> <p>La présence d'une transition par défaut affectée à E11 ou E12 n'a pas lieu d'être du fait de la simultanéité de l'activation des états.                  Lorsque la transition est franchie par le fait que l'événement survient et que la condition est vraie, E11 et E12 sont simultanément désactivés.</p>

