



## C10 : Modélisation de la chaîne d'information des systèmes

### C10-1 : Modélisation des systèmes à événements discrets

Émilien DURIF

Lycée La Martinière Monplaisir Lyon  
Classe de MPSI  
24 Juin 2025



# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications

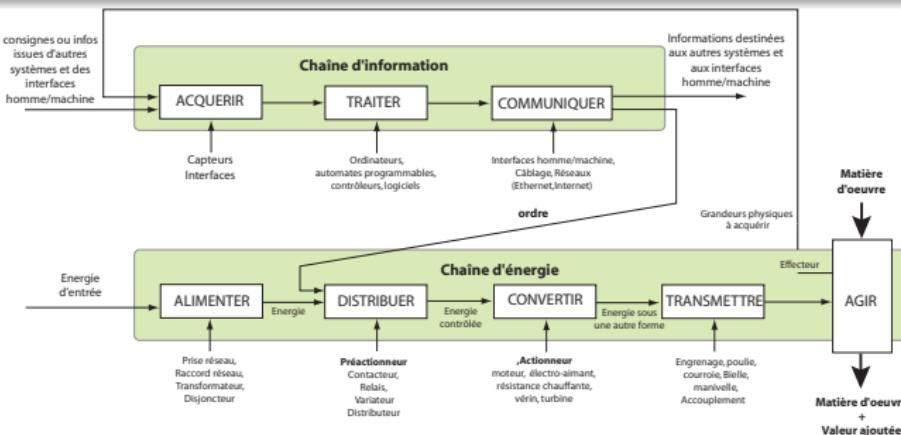


# Système numérique de contrôle-commande

## Système numérique de contrôle-commande

Un **système numérique de contrôle-commande** est une structure programmable (SNCC ou DCS pour *distributed control system* en Anglais) utilisée pour le pilotage d'un procédé industriel ou d'un système mécanique. Un SNCC est composé :

- cartes de commande : fonction **traiter** ;
- interface homme-machine : fonction **communiquer et/ou acquérir** ;
- ensemble de capteurs : fonction **acquérir** ;



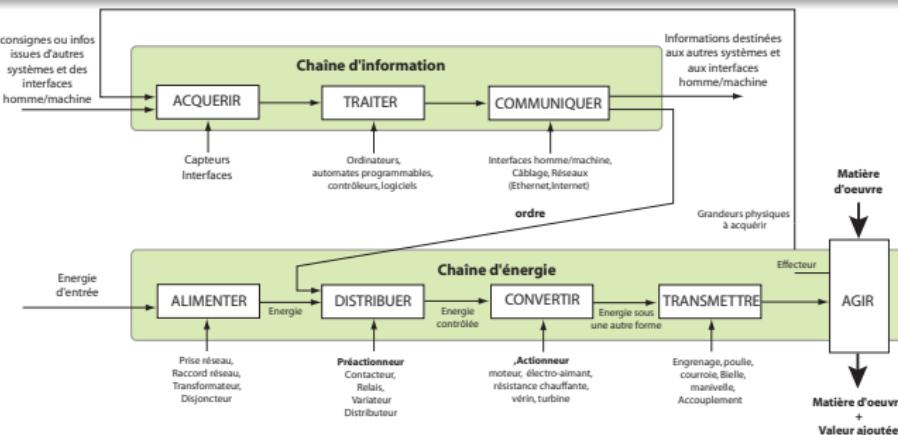


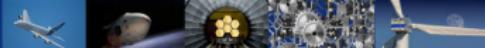
# Système numérique de contrôle-commande

## Système numérique de contrôle-commande

Un **système numérique de contrôle-commande** est une structure programmable (SNCC ou DCS pour *distributed control system* en Anglais) utilisée pour le pilotage d'un procédé industriel ou d'un système mécanique. Un SNCC est composé :

- cartes de commande : fonction **traiter** ;
- interface homme-machine : fonction **communiquer** et/ou **acquérir** ;
- ensemble de capteurs : fonction **acquérir** ;



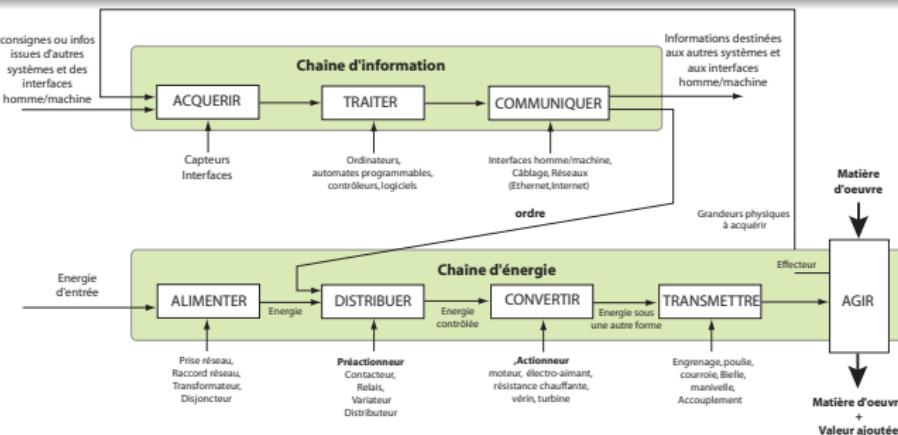


# Système numérique de contrôle-commande

## Système numérique de contrôle-commande

Un **système numérique de contrôle-commande** est une structure programmable (SNCC ou DCS pour *distributed control system* en Anglais) utilisée pour le pilotage d'un procédé industriel ou d'un système mécanique. Un SNCC est composé :

- cartes de commande : fonction **traiter** ;
- interface homme-machine : fonction **communiquer** et/ou **acquérir** ;
- ensemble de capteurs : fonction **acquérir** ;



## Système numérique de contrôle-commande

### Système numérique de contrôle-commande

Un **système numérique de contrôle-commande** est une structure programmable (SNCC ou DCS pour *distributed control system* en Anglais) utilisée pour le pilotage d'un procédé industriel ou d'un système mécanique. Un SNCC est composé :

- cartes de commande : fonction **traiter** ;
- interface homme-machine : fonction **communiquer** et/ou **acquérir** ;
- ensemble de capteurs : fonction **acquérir** ;



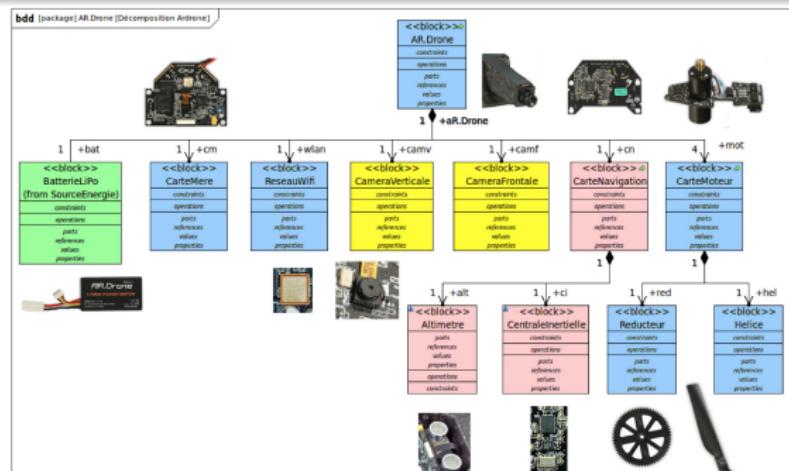


# Système numérique de contrôle-commande

## Système numérique de contrôle-commande

Un **système numérique de contrôle-commande** est une structure programmable (SNCC ou DCS pour *distributed control system* en Anglais) utilisée pour le pilotage d'un procédé industriel ou d'un système mécanique. Un SNCC est composé :

- cartes de commande : fonction **traiter** ;
- interface homme-machine : fonction **communiquer** et/ou **acquérir** ;
- ensemble de capteurs : fonction **acquérir** ;





## Système numérique de contrôle-commande

### Système numérique de contrôle-commande

Un **système numérique de contrôle-commande** est une structure programmable (SNCC ou DCS pour *distributed control system* en Anglais) utilisée pour le pilotage d'un procédé industriel ou d'un système mécanique. Un SNCC est composé :

- cartes de commande : fonction **traiter** ;
- interface homme-machine : fonction **communiquer** et/ou **acquérir** ;
- ensemble de capteurs : fonction **acquérir** ;

### Rôle des informations logiques dans un système

- Un certain nombre d'informations logiques doivent être collectées et traitées sur ce genre de système (Arrêt, détection d'obstacle, etc..).
- L'objet de ce cours est la présentation des bases nécessaires à la manipulation des informations logiques.



# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Informations binaires

### Informations binaires

- Un signal **binnaire ou logique** est une grandeur physique qui ne peut prendre que deux états : "Vrai ou Faux".
- On dit aussi qu'un signal binaire est un signal "Tout ou Rien", "présent ou absent".
- Par convention, on donne le code numérique 1 au signal quand il est à l'état "vrai" et 0 quand il est à l'état "faux".
- En pratique, il ne peut y avoir de discontinuité au niveau d'un signal physique. Le passage de 0 à 1 est appelé "**front montant**" et le passage de 1 à 0 "**front descendant**". Les informations traitées par un système combinatoire sont des signaux binaires.



## Informations binaires

### Informations binaires

- Un signal **binnaire ou logique** est une grandeur physique qui ne peut prendre que deux états : "Vrai ou Faux".
- On dit aussi qu'un signal binaire est un signal "Tout ou Rien", "présent ou absent".
- Par convention, on donne le code numérique 1 au signal quand il est à l'état "vrai" et 0 quand il est à l'état "faux".
- En pratique, il ne peut y avoir de discontinuité au niveau d'un signal physique. Le passage de 0 à 1 est appelé "front montant" et le passage de 1 à 0 "front descendant". Les informations traitées par un système combinatoire sont des signaux binaires.



## Informations binaires

### Informations binaires

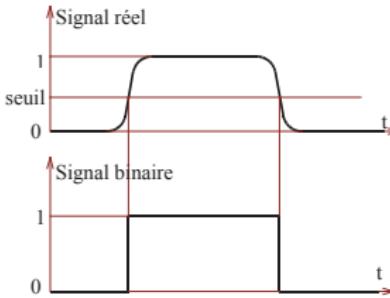
- Un signal **binnaire ou logique** est une grandeur physique qui ne peut prendre que deux états : "Vrai ou Faux".
- On dit aussi qu'un signal binaire est un signal "Tout ou Rien", "présent ou absent".
- Par convention, on donne le code numérique 1 au signal quand il est à l'état "vrai" et 0 quand il est à l'état "faux".
- En pratique, il ne peut y avoir de discontinuité au niveau d'un signal physique. Le passage de 0 à 1 est appelé "front montant" et le passage de 1 à 0 "front descendant". Les informations traitées par un système combinatoire sont des signaux binaires.



## Informations binaires

### Informations binaires

- Un signal **binnaire ou logique** est une grandeur physique qui ne peut prendre que deux états : "Vrai ou Faux".
- On dit aussi qu'un signal binaire est un signal "Tout ou Rien", "présent ou absent".
- Par convention, on donne le code numérique 1 au signal quand il est à l'état "vrai" et 0 quand il est à l'état "faux".
- En pratique, il ne peut y avoir de discontinuité au niveau d'un signal physique. Le passage de 0 à 1 est appelé "**front montant**" et le passage de 1 à 0 "**front descendant**". Les informations traitées par un système combinatoire sont des signaux binaires.





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Systèmes asservis numériques

### Definition

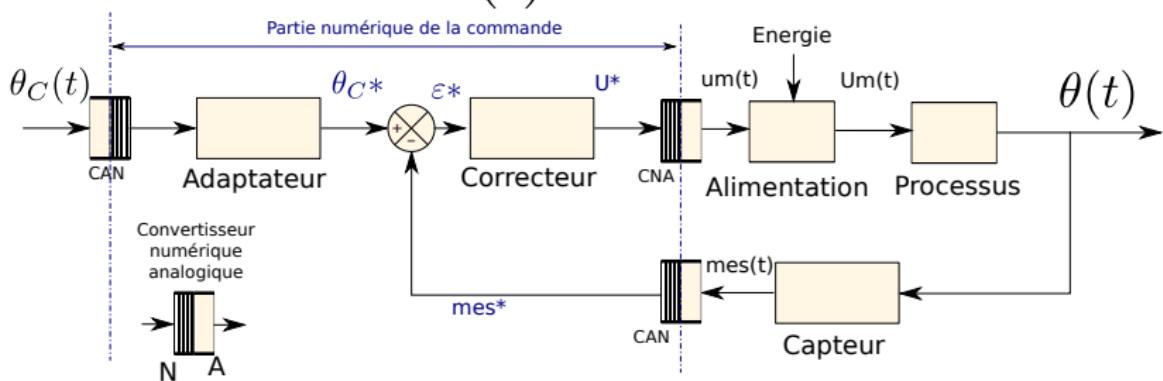
Représentation physique et modèle numérique

Le rôle d'une carte informatique est d'assurer les fonctions de calcul de l'écart et de la valeur de commande. Il faut donc convertir les grandeurs physique continues en grandeurs numériques :

- dans un sens avec un **Convertisseur Analogique Numérique** : CAN ;
- dans l'autre sens avec un **Convertisseur Numérique Analogique** : CNA.



## Systèmes asservis numériques





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Algèbre de Boole

### Algèbre de Boole

Soit un ensemble  $B\{0,1\}$  constitué de deux éléments représentants les deux états logiques vrai ou faux. Cet ensemble est muni d'une structure d'algèbre avec les trois lois suivantes :

Complémentarités : **NON**

$$\begin{aligned} B &\longmapsto B \\ a &\longmapsto \text{NON}(a) = \bar{a}. \end{aligned} \tag{1}$$

$\bar{a}$  est à 1 si  $a$  est à 0 et réciproquement.



## Algèbre de Boole

### Algèbre de Boole

Soit un ensemble  $B\{0,1\}$  constitué de deux éléments représentants les deux états logiques vrai ou faux. Cet ensemble est muni d'une structure d'algèbre avec les trois lois suivantes :

Produit booléen : **ET**

$$\begin{aligned} B \times B &\longmapsto B \\ (a, b) &\longmapsto a \text{ ET } b = a \cdot b. \end{aligned} \tag{2}$$

*a · b est à 1 si et seulement si a est à 1 ET b est à 1.*



## Algèbre de Boole

### Algèbre de Boole

Soit un ensemble  $B\{0,1\}$  constitué de deux éléments représentants les deux états logiques vrai ou faux. Cet ensemble est muni d'une structure d'algèbre avec les trois lois suivantes :

Somme booléenne : **OU**

$$\begin{aligned} B \times B &\longmapsto B \\ (a, b) &\longmapsto a \text{ OU } b = a + b. \end{aligned} \tag{3}$$

*a + b est à 1 si et seulement si a est à 1 OU b est à 1.*



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$

## Propriétés : manipulation de l'algèbre de Boole

	<b>Loi OU</b>	<b>Loi ET</b>
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$

- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$

- Inclusion :

$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  

$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$ .
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$ .
- Inclusion :  
$$a + \bar{a} \cdot b = a \cdot (1 + b) + \bar{a} \cdot b = a + a \cdot b + \bar{a} \cdot b = a + (a + \bar{a}) \cdot b = a + 1 \cdot b = a + b.$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$ .
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$ .
- Inclusion :  
$$a + \bar{a} \cdot b = a \cdot (1 + b) + \bar{a} \cdot b = a + a \cdot b + \bar{a} \cdot b = a + (a + \bar{a}) \cdot b = a + 1 \cdot b = a + b.$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$ .
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$ .
- Inclusion :  
$$a + \bar{a} \cdot b = a \cdot (1 + b) + \bar{a} \cdot b = a + a \cdot b + \bar{a} \cdot b = a + (a + \bar{a}) \cdot b = a + 1 \cdot b = a + b.$$



## Théorème de De Morgan

### Théorème de De Morgan

- Le complément d'un **OU** est le **ET** des compléments :

$$\overline{(a + b)} = \bar{a} \cdot \bar{b}. \quad (4)$$

- Le complément d'un **ET** est le **OU** des compléments :

$$\overline{(a \cdot b)} = \bar{a} + \bar{b}. \quad (5)$$



## Théorème de De Morgan

### Théorème de De Morgan

- Le complément d'un **OU** est le **ET** des compléments :

$$\overline{(a + b)} = \bar{a} \cdot \bar{b}. \quad (4)$$

- Le complément d'un **ET** est le **OU** des compléments :

$$\overline{(a \cdot b)} = \bar{a} + \bar{b}. \quad (5)$$



## Définition d'une fonction par sa table de vérité

### Table de vérité

Une **table de vérité** permet de représenter le comportement logique d'une fonction de plusieurs variables logiques. Pour une fonction de  $n$  variables la table comporte :

- $n$  colonnes plus une pour le résultat.
- $2^n$  lignes pour les différentes combinaisons des  $n$  variables écrites en **binaire naturel** ou **binaire réfléchi** à partir de 0.

$a$	$b$	$c$	$S$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

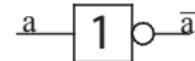
Code binaire naturel

$a$	$b$	$c$	$S$
0	0	0	0
0	0	1	1
0	1	1	1
0	1	0	1
1	1	0	0
1	1	1	1
1	0	1	0
1	0	0	0

Code binaire réfléchi

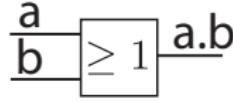


## Opérateurs logiques fondamentaux

Opérateur logique NON																	
Informations	Table de vérité	Symbolisation															
C'est l'opérateur qui transforme une variable logique $a$ en son complément $\bar{a}$ (a barre).	<table border="1"> <tr> <td><math>a</math></td><td><math>\bar{a}</math></td></tr> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </table>	$a$	$\bar{a}$	0	1	1	0										
$a$	$\bar{a}$																
0	1																
1	0																
Opérateur logique ET																	
Informations	Table de vérité	Symbolisation															
Le résultat R de l'opération "ET" entre deux variables binaires $a$ et $b$ est égal à 1 si et seulement si les deux variables sont vraies. On note $a \cdot b = R$ . On lit	<table border="1"> <tr> <td><math>a</math></td><td><math>b</math></td><td><math>a \cdot b</math></td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>-</td><td>-</td><td>-</td></tr> </table>	$a$	$b$	$a \cdot b$	0	0	0	0	1	0	1	0	0	-	-	-	
$a$	$b$	$a \cdot b$															
0	0	0															
0	1	0															
1	0	0															
-	-	-															



## Opérateur logiques fondamentaux

Informations	Opérateur logique OU	Symbolisation															
Table de vérité																	
Le résultat R de l'opération "OU" entre deux variables binaires $a$ et $b$ est égal à 1 si au moins une des deux variables est vraie. On note $a + b = R$ . On lit "a OU b égale R".	<table border="1"><thead><tr><th data-bbox="434 428 530 458"><math>a</math></th><th data-bbox="530 428 625 458"><math>b</math></th><th data-bbox="625 428 830 458"><math>a + b</math></th></tr></thead><tbody><tr><td data-bbox="434 458 530 561">0</td><td data-bbox="530 458 625 561">0</td><td data-bbox="625 458 830 561">0</td></tr><tr><td data-bbox="434 561 530 663">0</td><td data-bbox="530 561 625 663">1</td><td data-bbox="625 561 830 663">1</td></tr><tr><td data-bbox="434 663 530 765">1</td><td data-bbox="530 663 625 765">0</td><td data-bbox="625 663 830 765">1</td></tr><tr><td data-bbox="434 765 530 830">1</td><td data-bbox="530 765 625 830">1</td><td data-bbox="625 765 830 830">1</td></tr></tbody></table>	$a$	$b$	$a + b$	0	0	0	0	1	1	1	0	1	1	1	1	
$a$	$b$	$a + b$															
0	0	0															
0	1	1															
1	0	1															
1	1	1															



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$
- Inclusion :  
$$a + a \cdot b = a \cdot (1 + b) + a \cdot b = a + a \cdot b + a \cdot b = a + (a + a) \cdot b = a + 1 \cdot b = a + b$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$ .
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$ .
- Inclusion :  
$$a + \bar{a} \cdot b = a \cdot (1 + b) + \bar{a} \cdot b = a + a \cdot b + \bar{a} \cdot b = a + (a + \bar{a}) \cdot b = a + 1 \cdot b = a + b.$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$ .
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$ .
- Inclusion :  
$$a + \bar{a} \cdot b = a \cdot (1 + b) + \bar{a} \cdot b = a + a \cdot b + \bar{a} \cdot b = a + (a + \bar{a}) \cdot b = a + 1 \cdot b = a + b.$$



## Propriétés : manipulation de l'algèbre de Boole

	Loi OU	Loi ET
Commutativité	$a + b = b + a$	$a \cdot b = b \cdot a$
Distributivité	$a + (b \cdot c) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (a \cdot b) + (a \cdot c) = a \cdot b + a \cdot c$
Associativité	$a + (b + c) = (a + b) + c = a + b + c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
Élément neutre 0	$a + 0 = a$	$a \cdot 0 = 0$
Élément neutre 1	$a + 1 = 1$	$a \cdot 1 = a$
Complémentarité	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$
Idempotence	$a + a = a$	$a \cdot a = a$

- Double complément :  $\bar{\bar{a}} = a$ .
- Absorption :  $a + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$ .
- Inclusion :  
$$a + \bar{a} \cdot b = a \cdot (1 + b) + \bar{a} \cdot b = a + a \cdot b + \bar{a} \cdot b = a + (a + \bar{a}) \cdot b = a + 1 \cdot b = a + b.$$

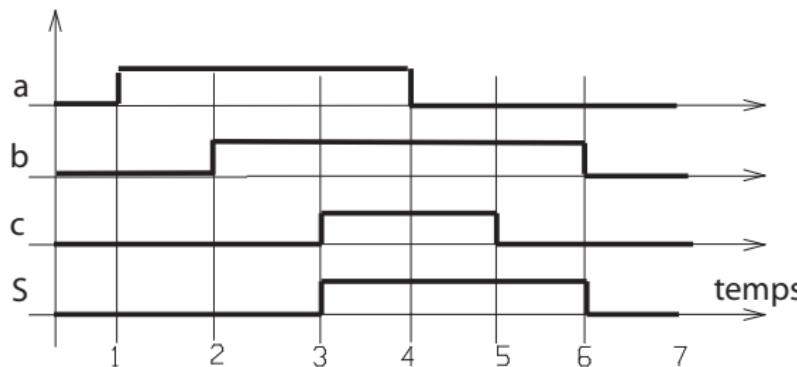


## Chronogrammes

### Chronogramme

Le **chronogramme** est un graphe qui permet de définir l'état d'une variable de sortie à partir de l'état des variables d'entrée définies chronologiquement.

Exemple de chronogramme pour  $S = \bar{a} \cdot c + \bar{a} \cdot b + b \cdot c$



### Remarque

Deux variables ne peuvent pas changer d'état simultanément.

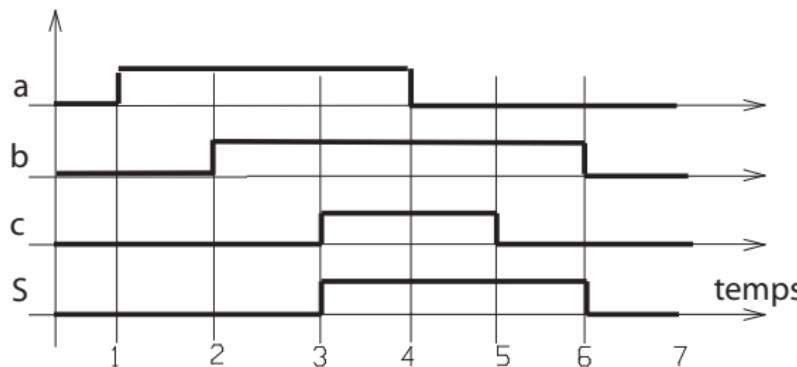


## Chronogrammes

### Chronogramme

Le **chronogramme** est un graphe qui permet de définir l'état d'une variable de sortie à partir de l'état des variables d'entrée définies chronologiquement.

Exemple de chronogramme pour  $S = \bar{a} \cdot c + \bar{a} \cdot b + b \cdot c$



### Remarque

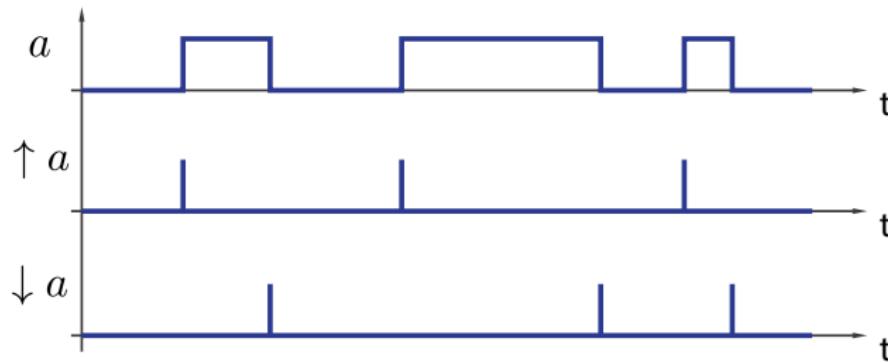
Deux variables ne peuvent pas changer d'état simultanément.



## Notions de fronts montants et descendants

### Fronts montants et descendants

- Le **front montant** d'une variable  $a$  est vrai à l'instant pour lequel la variable passe de son état logique 0 à son état logique 1. Il est noté  $\uparrow a$ .
- Le **front descendant** d'une variable  $a$  est vrai à l'instant pour lequel la variable passe de son état logique 1 à son état logique 0. Il est noté  $\downarrow a$ .
- Un front est d'une durée théorique nulle. Il sera représenté par une impulsion de Dirac sur un chronogramme.

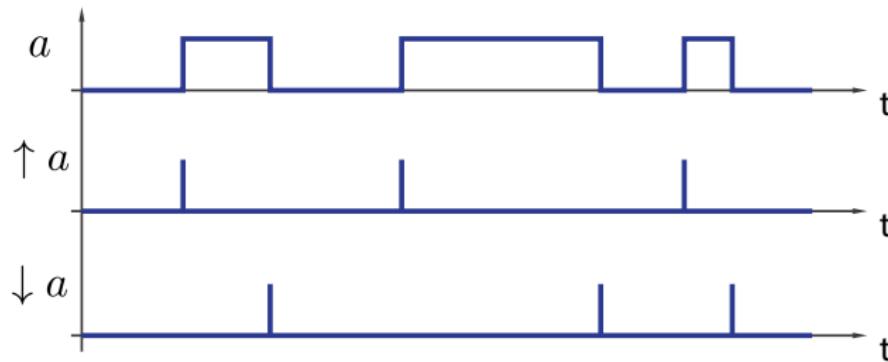




## Notions de fronts montants et descendants

### Fronts montants et descendants

- Le **front montant** d'une variable  $a$  est vrai à l'instant pour lequel la variable passe de son état logique 0 à son état logique 1. Il est noté  $\uparrow a$ .
- Le **front descendant** d'une variable  $a$  est vrai à l'instant pour lequel la variable passe de son état logique 1 à son état logique 0. Il est noté  $\downarrow a$ .
- Un front est d'une durée théorique nulle. Il sera représenté par une impulsion de Dirac sur un chronogramme.

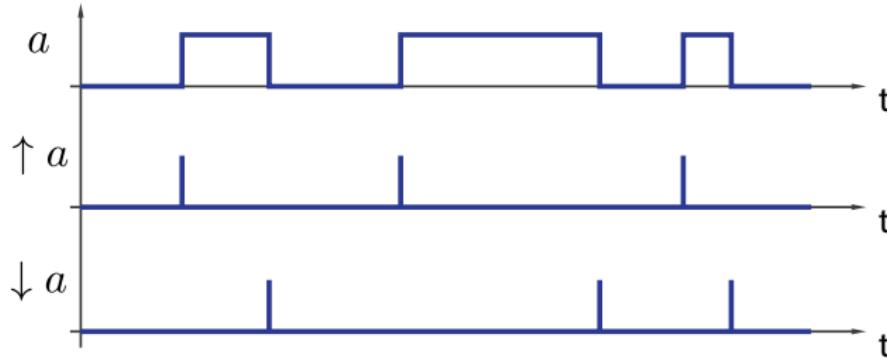




## Notions de fronts montants et descendants

### Fronts montants et descendants

- Le **front montant** d'une variable  $a$  est vrai à l'instant pour lequel la variable passe de son état logique 0 à son état logique 1. Il est noté  $\uparrow a$ .
- Le **front descendant** d'une variable  $a$  est vrai à l'instant pour lequel la variable passe de son état logique 1 à son état logique 0. Il est noté  $\downarrow a$ .
- Un front est d'une durée théorique nulle. Il sera représenté par une impulsion de Dirac sur un chronogramme.



# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Modélisation de la commande des systèmes

### Introduction

- Le chapitre précédent à permis de donner les bases de **l'étude des systèmes numériques** avec la manipulation des informations binaires.
- Ces informations sont traitées par des **cartes de commandes** qui sont généralement insérées dans des **micro-contrôleur**.
- Il faut maintenant faire le lien entre le **cahier des charges** et le **fonctionnement attendu du système** en mettant en relation toutes les informations issues de commandes ou de mesures numériques.
- Dans ce chapitre nous développerons les bases de lecture des **diagrammes comportementaux d'états et d'activités** nécessaires pour la commande et la simulation du fonctionnement d'un **système complexe**.



## Modélisation de la commande des systèmes

### Introduction

- Le chapitre précédent à permis de donner les bases de l'**étude des systèmes numériques** avec la manipulation des informations binaires.
- Ces informations sont **traitées** par des **cartes de commandes** qui sont généralement insérées dans des **micro-contrôleur**.
- Il faut maintenant faire le lien entre le **cahier des charges** et le **fonctionnement attendu du système** en mettant en relation toutes les informations issues de commandes ou de mesures numériques.
- Dans ce chapitre nous développerons les bases de lecture des **diagrammes comportementaux d'états et d'activités** nécessaires pour la commande et la simulation du fonctionnement d'un **système complexe**.



## Modélisation de la commande des systèmes

### Introduction

- Le chapitre précédent à permis de donner les bases de l'**étude des systèmes numériques** avec la manipulation des informations binaires.
- Ces informations sont **traitées** par des **cartes de commandes** qui sont généralement insérées dans des **micro-contrôleur**.
- Il faut maintenant faire le lien entre le **cahier des charges** et le **fonctionnement attendu du système** en mettant en relation toutes les informations issues de commandes ou de mesures numériques.
- Dans ce chapitre nous développerons les bases de lecture des **diagrammes comportementaux d'états et d'activités** nécessaires pour la commande et la simulation du fonctionnement d'un **système complexe**.



## Modélisation de la commande des systèmes

### Introduction

- Le chapitre précédent à permis de donner les bases de l'**étude des systèmes numériques** avec la manipulation des informations binaires.
- Ces informations sont **traitées** par des **cartes de commandes** qui sont généralement insérées dans des **micro-contrôleur**.
- Il faut maintenant faire le lien entre le **cahier des charges** et le **fonctionnement attendu du système** en mettant en relation toutes les informations issues de commandes ou de mesures numériques.
- Dans ce chapitre nous développerons les bases de lecture des **diagrammes comportementaux d'états et d'activités** nécessaires pour la commande et la simulation du fonctionnement d'un **système complexe**.

# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

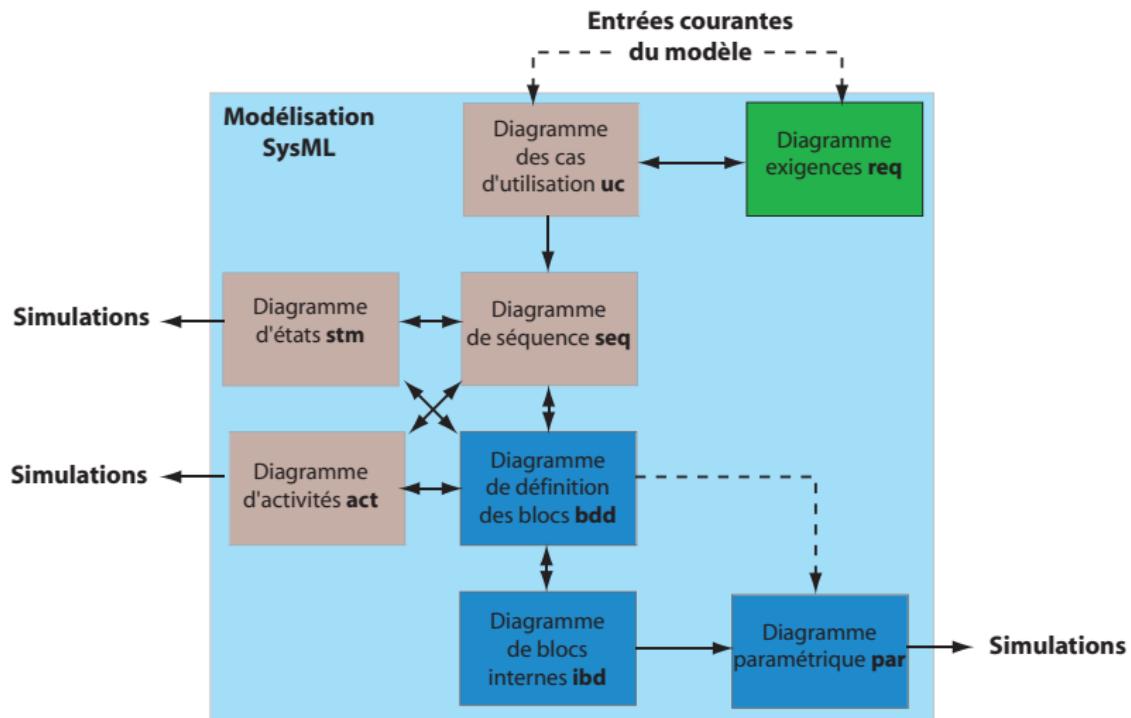
- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Modélisation à l'aide de l'outil SysML





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

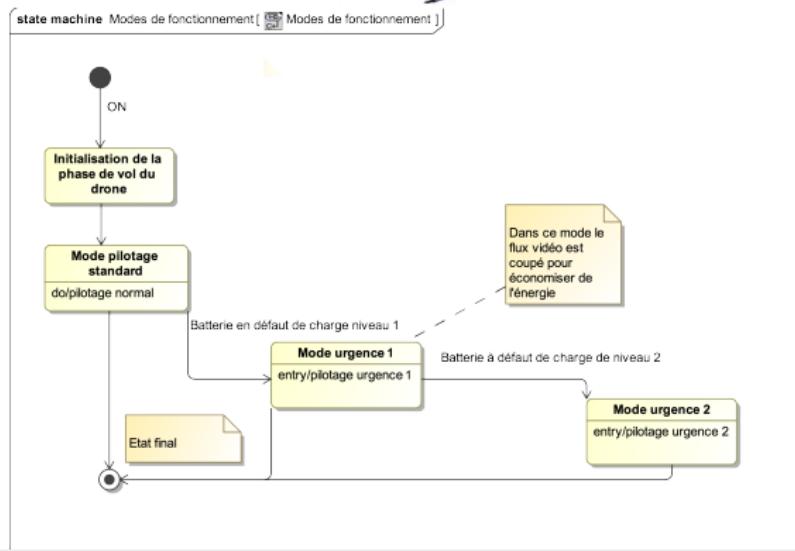
- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications

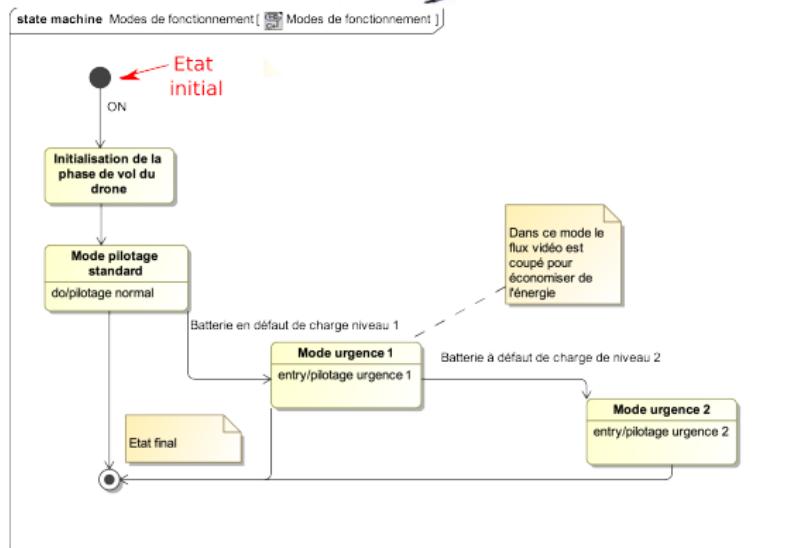


## Diagramme d'états



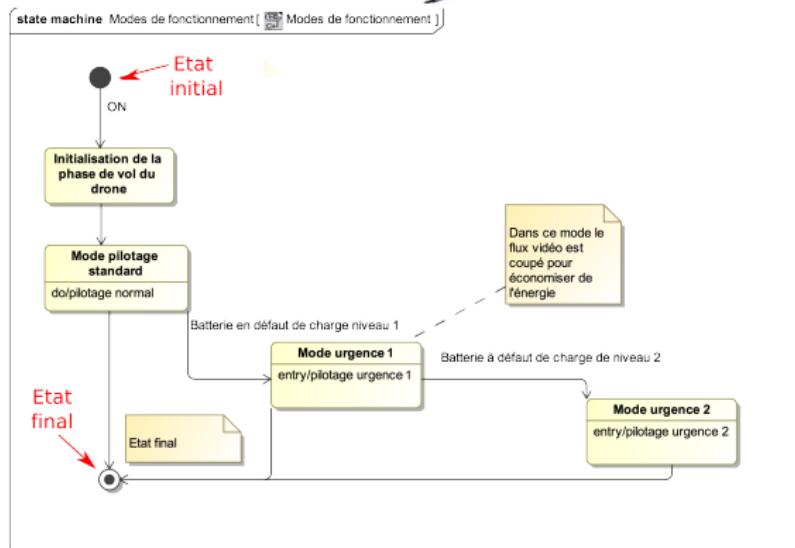


## Diagramme d'états



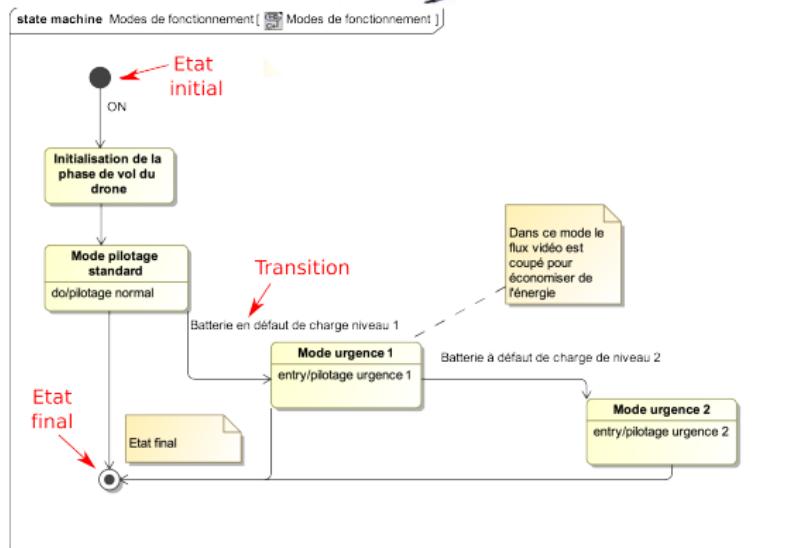


## Diagramme d'états



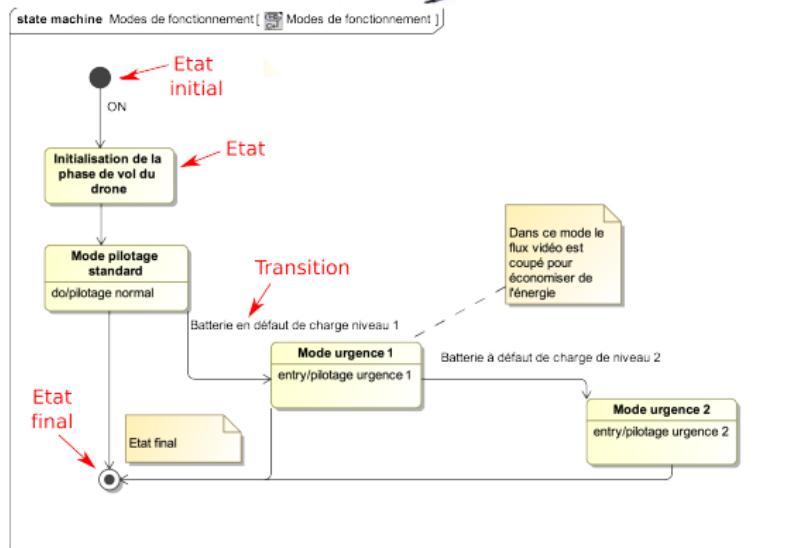


## Diagramme d'états





## Diagramme d'états





## Diagramme d'états

### Diagramme d'état

Le **diagramme d'états** est un diagramme comportemental appelé *State Machine Diagram (stm)* dans le langage SysML. Il est rattaché à un **bloc** (système, sous-système ou composant). Il permet de modéliser les différents états pris par le bloc en fonction d'événements. Les éléments graphiques utilisés dans ce diagramme sont principalement :

- des rectangles aux coins arrondis pour les états,
- des flèches orientées de l'état de départ à l'état cible pour les transitions,
- des occurrences attachées à la transition spécifient les conditions de franchissement.

### Remarque

Chaque bloc ne conduit pas forcément à un diagramme d'états car cela n'est pas toujours possible.

## Diagramme d'états

### Diagramme d'état

Le **diagramme d'états** est un diagramme comportemental appelé *State Machine Diagram (stm)* dans le langage SysML. Il est rattaché à un bloc (système, sous-système ou composant). Il permet de modéliser les différents états pris par le bloc en fonction d'événements. Les éléments graphiques utilisés dans ce diagramme sont principalement :

- des rectangles aux coins arrondis pour les états,
- des flèches orientées de l'état de départ à l'état cible pour les transitions,
- des occurrences attachées à la transition spécifient les conditions de franchissement.

### Remarque

Chaque bloc ne conduit pas forcément à un diagramme d'états car cela n'est pas toujours possible.



## Diagramme d'états

### Diagramme d'état

Le **diagramme d'états** est un diagramme comportemental appelé *State Machine Diagram (stm)* dans le langage SysML. Il est rattaché à un bloc (système, sous-système ou composant). Il permet de modéliser les différents états pris par le bloc en fonction d'événements. Les éléments graphiques utilisés dans ce diagramme sont principalement :

- des rectangles aux coins arrondis pour les états,
- des flèches orientées de l'état de départ à l'état cible pour les transitions,
- des occurrences attachées à la transition spécifient les conditions de franchissement.

### Remarque

Chaque bloc ne conduit pas forcément à un diagramme d'états car cela n'est pas toujours possible.



## Diagramme d'états

### Diagramme d'état

Le **diagramme d'états** est un diagramme comportemental appelé *State Machine Diagram (stm)* dans le langage SysML. Il est rattaché à un bloc (système, sous-système ou composant). Il permet de modéliser les différents états pris par le bloc en fonction d'événements. Les éléments graphiques utilisés dans ce diagramme sont principalement :

- des rectangles aux coins arrondis pour les états,
- des flèches orientées de l'état de départ à l'état cible pour les transitions,
- des occurrences attachées à la transition spécifient les conditions de franchissement.

### Remarque

Chaque bloc ne conduit pas forcément à un diagramme d'états car cela n'est pas toujours possible.



# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Diagramme d'états : états

### États

Un état représente une situation d'une durée finie durant laquelle un système ou sous-système (on parlera alors d'**objet**) :

- exécute une activité,
- satisfait à une certaine condition,
- ou bien est en attente d'un évènement.

### Remarque : règle de syntaxe

Un état peut être partitionné en plusieurs compartiments séparés par une ligne horizontale. Le premier compartiment contient le nom de l'état et les autres peuvent recevoir des transitions internes, ou des sous-états quand il s'agit d'un état composite. Dans le cas d'un état simple, on peut omettre toute barre de séparation.



## Diagramme d'états : états

### États

Un état représente une situation d'une durée finie durant laquelle un système ou sous-système (on parlera alors d'**objet**) :

- exécute une activité,
- satisfait à une certaine condition,
- ou bien est en attente d'un évènement.

### Remarque : règle de syntaxe

Un état peut être partitionné en plusieurs compartiments séparés par une ligne horizontale. Le premier compartiment contient le nom de l'état et les autres peuvent recevoir des transitions internes, ou des sous-états quand il s'agit d'un état composite. Dans le cas d'un état simple, on peut omettre toute barre de séparation.



## Diagramme d'états : états

### États

Un état représente une situation d'une durée finie durant laquelle un système ou sous-système (on parlera alors d'**objet**) :

- exécute une activité,
- satisfait à une certaine condition,
- ou bien est en attente d'un évènement.

### Remarque : règle de syntaxe

Un état peut être partitionné en plusieurs compartiments séparés par une ligne horizontale. Le premier compartiment contient le nom de l'état et les autres peuvent recevoir des transitions internes, ou des sous-états quand il s'agit d'un état composite. Dans le cas d'un état simple, on peut omettre toute barre de séparation.

## Diagramme d'états : états

### États

Un état représente une situation d'une durée finie durant laquelle un système ou sous-système (on parlera alors d'**objet**) :

- exécute une activité,
- satisfait à une certaine condition,
- ou bien est en attente d'un évènement.

### Remarque : règle de syntaxe

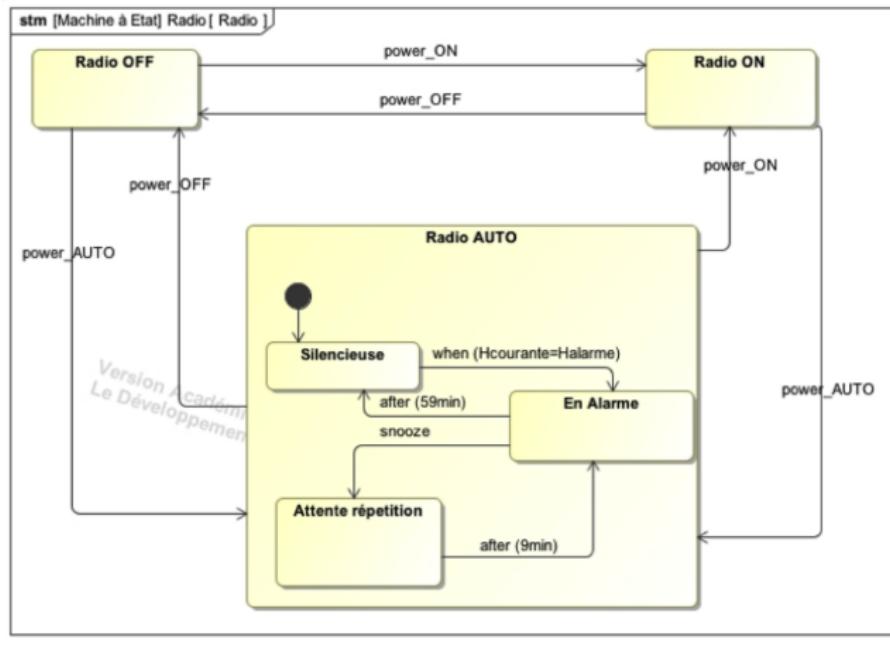
Un état peut être partitionné en plusieurs compartiments séparés par une ligne horizontale. Le premier compartiment contient le nom de l'état et les autres peuvent recevoir des transitions internes, ou des sous-états quand il s'agit d'un état composite. Dans le cas d'un état simple, on peut omettre toute barre de séparation.



## Diagramme d'états : états composites

### État composite

Un état composite peut contenir ou envelopper des sous-états.



## Diagramme d'états : états composites

### Remarque

On peut représenter en abrégé un état composite. Ceci permet d'indiquer qu'un état est composite et que sa définition est donnée sur un autre diagramme.

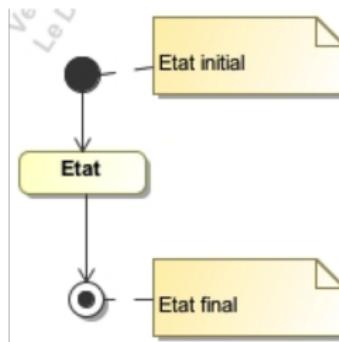
associer client et commande



## État initial, état final

### État initial, état final

- Un **état initial** est un pseudo état qui indique l'état de départ par défaut, lorsque le diagramme d'états ou l'état enveloppant est invoqué.
- Un **état final** est un pseudo état qui indique que le diagramme d'états-transitions, ou l'état enveloppant est terminé.



# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

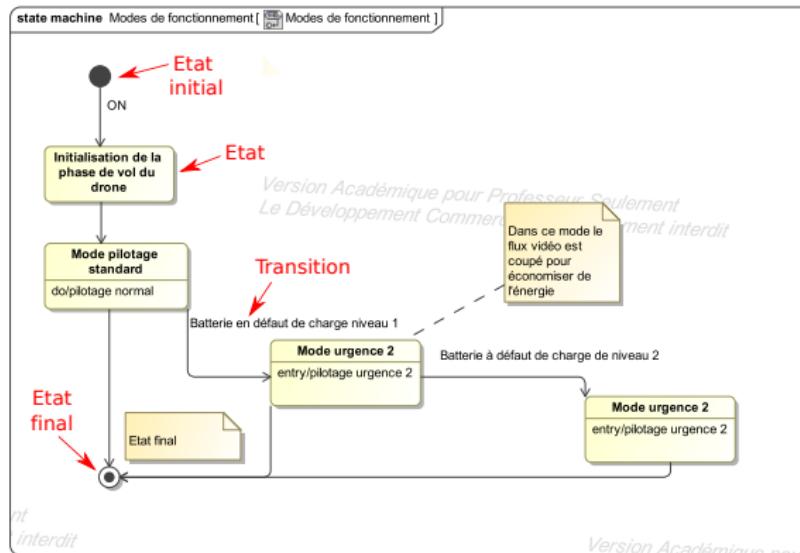
- Définition
- Exemples d'applications



## Transition

### Définition

Une **transition** représente le passage d'un état à un autre.





## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2*;
- **Syntaxe** : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.

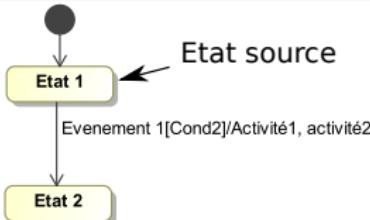




## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2*;
- Syntaxe : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.

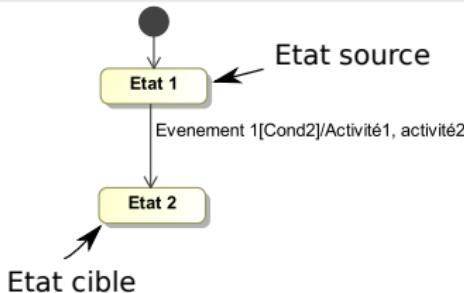




## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2*;
- Syntaxe : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.

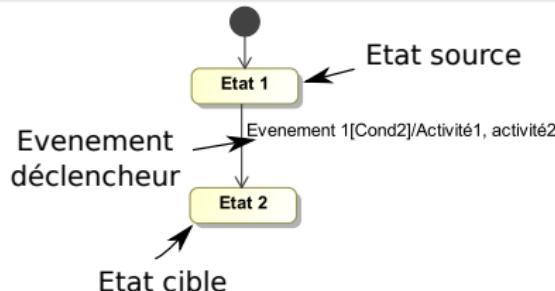




## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2*;
- Syntaxe : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.

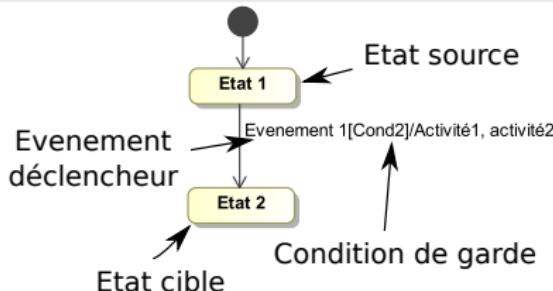




## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1 puis activité 2*;
- Syntaxe : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.

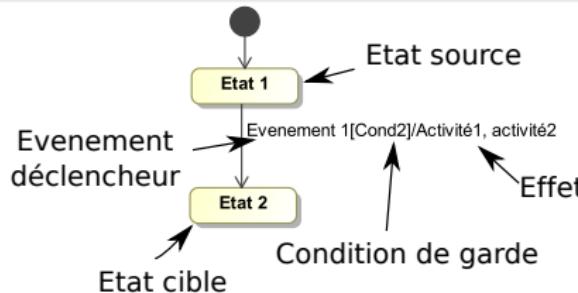




## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2* ;
- Syntaxe : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.

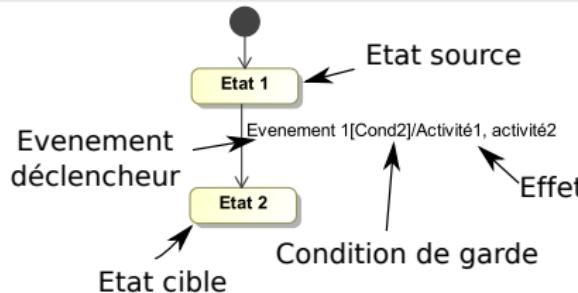




## Transition externe

### Transition externe

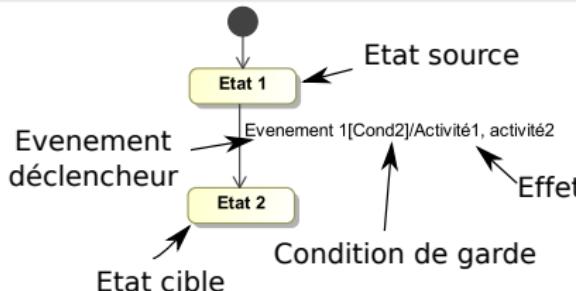
- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2* ;
- **Syntaxe** : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.



## Transition externe

### Transition externe

- Une **transition externe** décrit la réaction d'un objet lorsqu'un évènement se produit. En règle général une transition possède :
  - un état source : *Etat 1*,
  - un état cible : *Etat 2*,
  - un évènement déclencheur : *Evenement 1*,
  - une condition de garde : *Cond2*,
  - un effet : *activité 1* puis *activité 2* ;
- Syntaxe : évènement [garde] /effet.
- Elle est représentée par une ligne terminée par une flèche. Elle relie un état source à un état cible. Une transition ne peut être franchie que si son état source est actif.



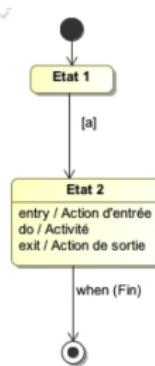


## Transition interne

### Transition interne

Une transition interne est notée à l'intérieur d'un état. Elle est précédée de l'un des mots clé suivants :

- **entry/action d'entrée** : entraîne la réalisation d'une action à l'entrée de l'état,
- **do/Activité** : entraîne la réalisation d'une activité lorsque l'état est actif,
- **On évènement/activité** : entraîne la réalisation d'une activité lorsqu'un évènement se réalise,
- **exit/Action de sortie** : entraîne la réalisation d'une action à la sortie de l'état.



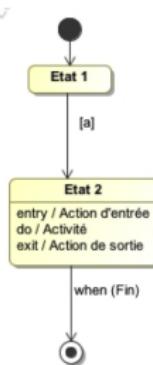


## Transition interne

### Transition interne

Une transition interne est notée à l'intérieur d'un état. Elle est précédée de l'un des mots clé suivants :

- **entry/action d'entrée** : entraîne la réalisation d'une action à l'entrée de l'état,
- **do/Activité** : entraîne la réalisation d'une activité lorsque l'état est actif,
- **On évènement/activité** : entraîne la réalisation d'une activité lorsqu'un évènement se réalise,
- **exit/Action de sortie** : entraîne la réalisation d'une action à la sortie de l'état.



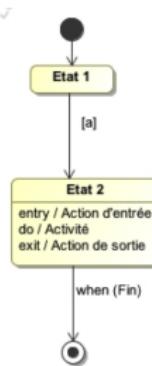


## Transition interne

### Transition interne

Une transition interne est notée à l'intérieur d'un état. Elle est précédée de l'un des mots clé suivants :

- **entry/action d'entrée** : entraîne la réalisation d'une action à l'entrée de l'état,
- **do/Activité** : entraîne la réalisation d'une activité lorsque l'état est actif,
- **On évènement/activité** : entraîne la réalisation d'une activité lorsqu'un évènement se réalise,
- **exit/Action de sortie** : entraîne la réalisation d'une action à la sortie de l'état.



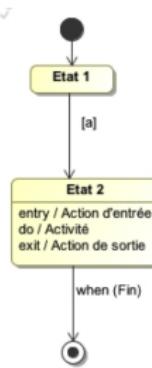


## Transition interne

### Transition interne

Une transition interne est notée à l'intérieur d'un état. Elle est précédée de l'un des mots clé suivants :

- **entry/action d'entrée** : entraîne la réalisation d'une action à l'entrée de l'état,
- **do/Activité** : entraîne la réalisation d'une activité lorsque l'état est actif,
- **On évènement/activité** : entraîne la réalisation d'une activité lorsqu'un évènement se réalise,
- **exit/Action de sortie** : entraîne la réalisation d'une action à la sortie de l'état.



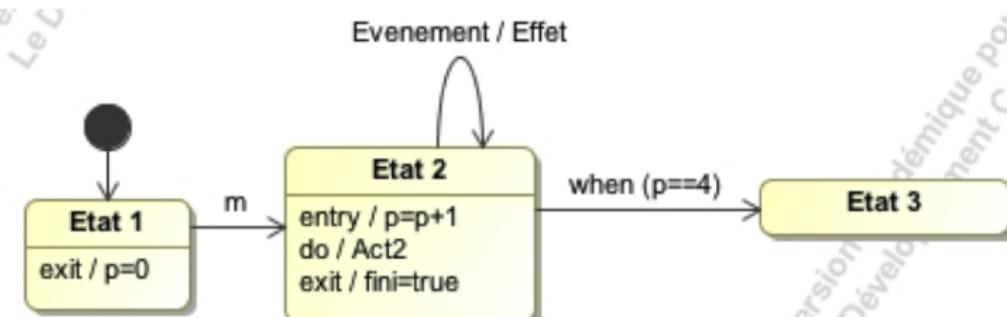


## Transition propre

### Transition propre

Une transition propre concerne un état propre. Elle relie donc un état à lui-même.

- Dans l'exemple ci-dessous, lorsque **Etat2** est actif, si **Evenement** se produit, alors **Effet** (activité de courte durée) est déclenchée puis **Etat2** est de nouveau actif.
- Chaque fois que **Evenement** se produit, la réactivation de **Etat2** fait que  $p$  est incrémenté.
- Le fait de sortir et d'entrer dans **Etat2** peut avoir des effets néfastes comme l'interruption puis le redémarrage de **Act2**.



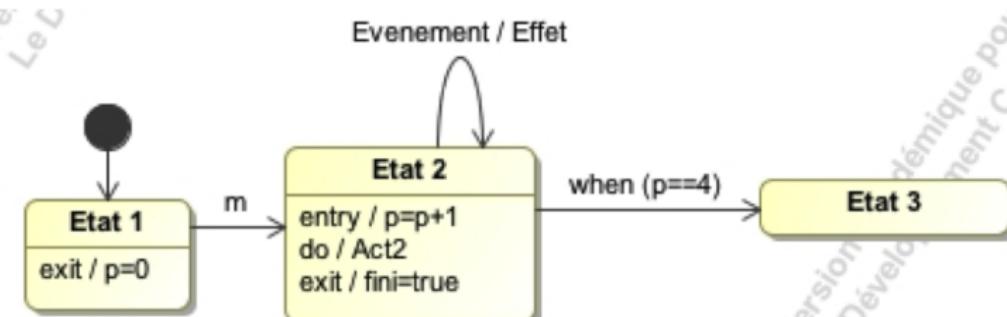


## Transition propre

### Transition propre

Une transition propre concerne un état propre. Elle relie donc un état à lui-même.

- Dans l'exemple ci-dessous, lorsque **Etat2** est actif, si **Evenement** se produit, alors **Effet** (activité de courte durée) est déclenchée puis **Etat2** est de nouveau actif.
- Chaque fois que **Evenement** se produit, la réactivation de **Etat2** fait que **p** est incrémenté.
- Le fait de sortir et d'entrer dans **Etat2** peut avoir des effets néfastes comme l'interruption puis le redémarrage de **Act2**.



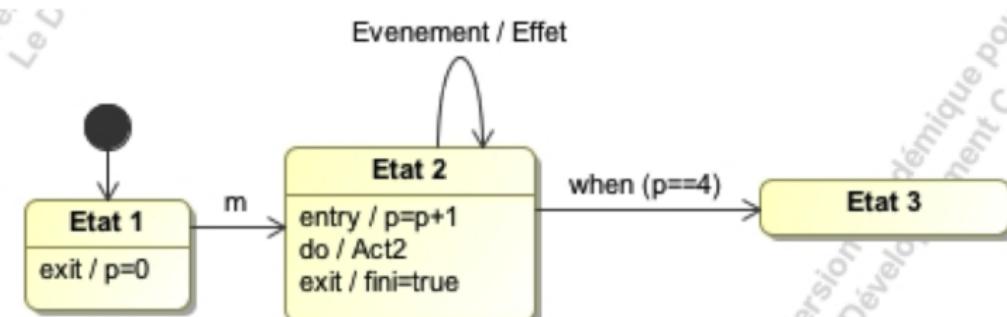


## Transition propre

### Transition propre

Une transition propre concerne un état propre. Elle relie donc un état à lui-même.

- Dans l'exemple ci-dessous, lorsque **Etat2** est actif, si **Evenement** se produit, alors **Effet** (activité de courte durée) est déclenchée puis **Etat2** est de nouveau actif.
- Chaque fois que **Evenement** se produit, la réactivation de **Etat2** fait que **p** est incrémenté.
- Le fait de sortir et d'entrer dans **Etat2** peut avoir des effets néfastes comme l'interruption puis le redémarrage de **Act2**.



## Évènement

### Évènement

Un évènement se produit pendant l'exécution d'un système.

- Un évènement se produit à un instant précis et est dépourvu de durée.
- Quand un évènement est reçu, une transition peut être déclenchée et faire basculer l'objet dans un nouvel état.
- On peut diviser les événements en plusieurs types explicites et implicites : signal, appel, changement et temporel.



# Événement

## Types d'évènements

- Un événement de type signal : appui sur un bouton.
- Un événement de type changement : when (< condition\_booléenne >) Un évènement de changement est évalué continuellement jusqu'à ce qu'il devienne vrai et c'est à ce moment là que la transition se déclenche.
- Les événement temporels sont générés par le passage du temps.
  - Un événement temporel spécifié de manière relative : after (< durée >),
  - Un événement temporel spécifié de manière absolue : when (< date >) ou at (< date >) dans le logiciel MagicDraw.



# Événement

## Types d'évènements

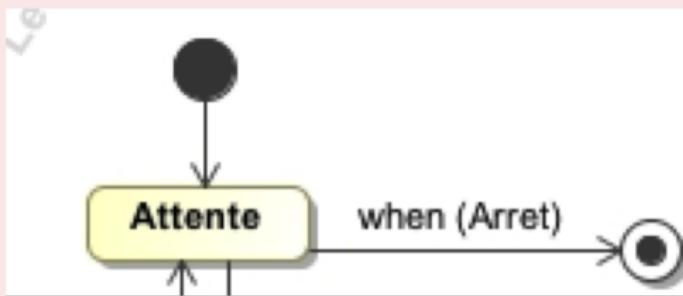
- Un événement de type **signal** : appui sur un bouton.
- Un événement de type **changement** : when (< condition\_booléenne >) Un évènement de changement est évalué continuellement jusqu'à ce qu'il devienne vrai et c'est à ce moment là que la transition se déclenche.
- Les événement temporels sont générés par le passage du temps.
  - Un événement temporel spécifié de manière relative : after (< durée >),
  - Un événement temporel spécifié de manière absolue : when (< date >) ou at (< date >) dans le logiciel MagicDraw.



## Événement

### Types d'évènements

- Un événement de type signal : appui sur un bouton.
- Un événement de type changement : **when (< condition\_booléenne >)** Un événement de changement est évalué continuellement jusqu'à ce qu'il devienne vrai et c'est à ce moment là que la transition se déclenche.
- Les événements temporels sont générés par le passage du temps.
  - Un événement temporel spécifié de manière relative : **after (< durée >)**,
  - Un événement temporel spécifié de manière absolue : **when (< date >)** ou **at (< date >)** dans le logiciel MagicDraw.

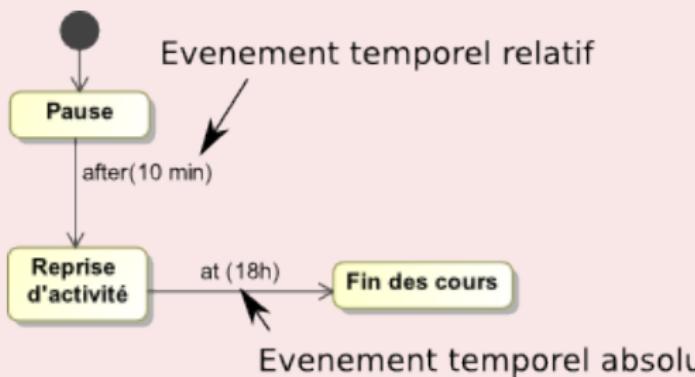




## Événement

### Types d'évènements

- Un événement de type **signal** : appui sur un bouton.
- Un événement de type **changement** : **when (< condition\_booléenne >)** Un événement de changement est évalué continuellement jusqu'à ce qu'il devienne vrai et c'est à ce moment là que la transition se déclenche.
- Les événements **temporels** sont générés par le passage du temps.
  - Un événement temporel spécifié de manière relative : **after (< durée >)**,
  - Un événement temporel spécifié de manière absolue : **when (< date >)** ou **at (< date >)** dans le logiciel MagicDraw.



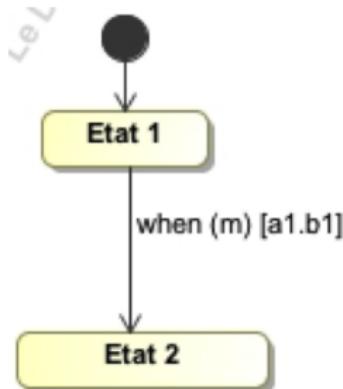
## Condition de garde

## Condition de garde

Une condition de garde est représentée entre crochet : [garde]

Il s'agit d'une expression logique sur les attributs de l'objet associés au diagramme d'états ainsi que sur les paramètres de l'événement déclencheur.

- La condition est évaluée uniquement lorsque l'événement déclencheur se produit.
  - Si l'expression est fausse à ce moment là, la transition ne se déclenche pas.
  - Si elle est vrai, la transition se déclenche et ses effets se produisent.





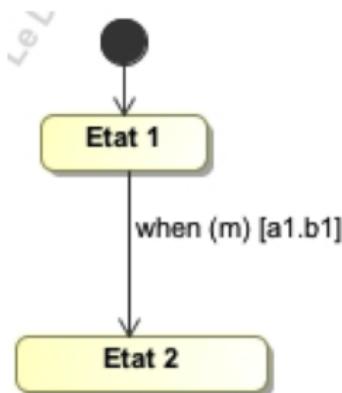
## Condition de garde

### Condition de garde

Une **condition de garde** est représentée entre crochet : [garde]

Il s'agit d'une expression logique sur les attributs de l'objet associés au diagramme d'états ainsi que sur les paramètres de l'événement déclencheur.

- La condition est évaluée uniquement lorsque l'événement déclencheur se produit.
- Si l'expression est fausse à ce moment là, la transition ne se déclenche pas.
- Si elle est vrai, la transition se déclenche et ses effets se produisent.





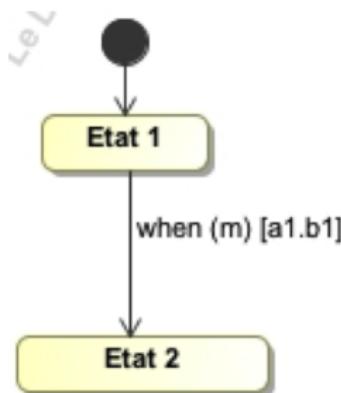
## Condition de garde

### Condition de garde

Une **condition de garde** est représentée entre crochet : [garde]

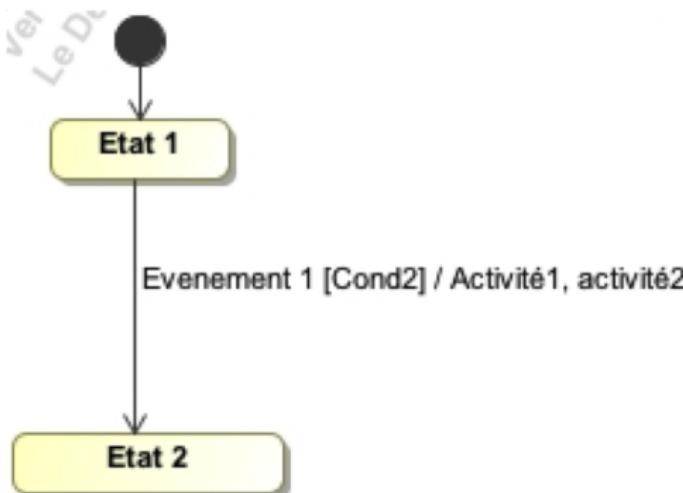
Il s'agit d'une expression logique sur les attributs de l'objet associés au diagramme d'états ainsi que sur les paramètres de l'événement déclencheur.

- La condition est évaluée uniquement lorsque l'événement déclencheur se produit.
- Si l'expression est fausse à ce moment là, la transition ne se déclenche pas.
- Si elle est vrai, la transition se déclenche et ses effets se produisent.



## Effet d'une transition

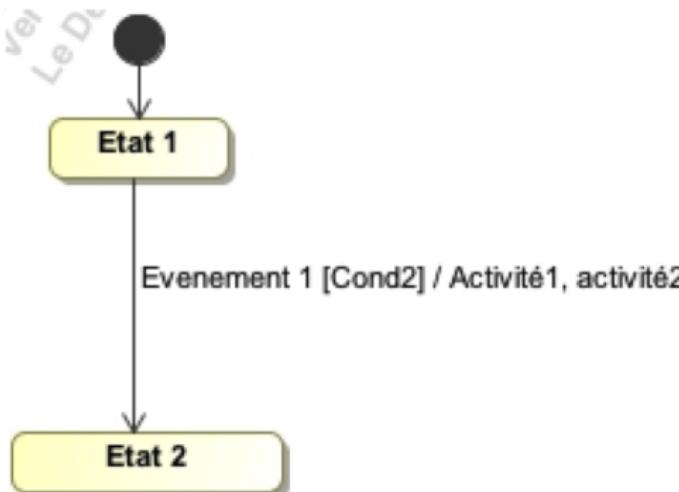
- Lorsqu'une transition se déclenche, son effet spécifié par "**/activité**" dans la syntaxe, s'exécute. Il s'agit généralement d'une activité qui peut être :
  - une opération primitive comme une instruction d'assignation ;
  - l'envoi d'un signal ;
  - l'appel d'une opération ;
  - une liste d'activités, etc.
- La façon de spécifier l'activité à réaliser est laissée libre (langage naturel ou pseudo-code). Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.





## Effet d'une transition

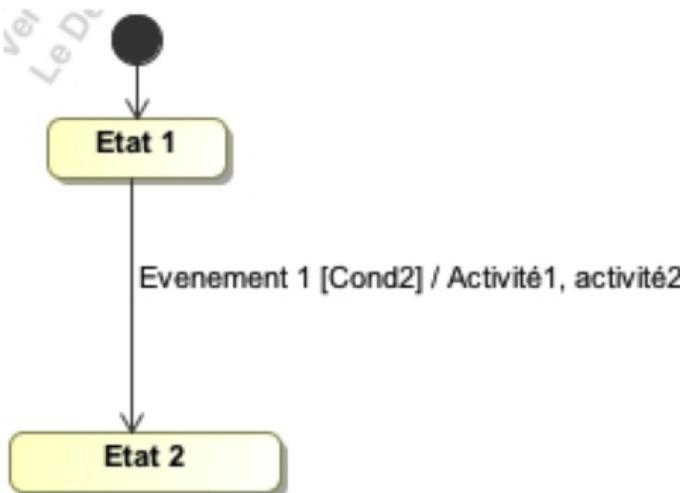
- Lorsqu'une transition se déclenche, son effet spécifié par "**/activité**" dans la syntaxe, s'exécute. Il s'agit généralement d'une activité qui peut être :
  - une opération primitive comme une instruction d'assignation ;
  - l'envoi d'un signal ;
  - l'appel d'une opération ;
  - une liste d'activités, etc.
- La façon de spécifier l'activité à réaliser est laissée libre (langage naturel ou pseudo-code). Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.





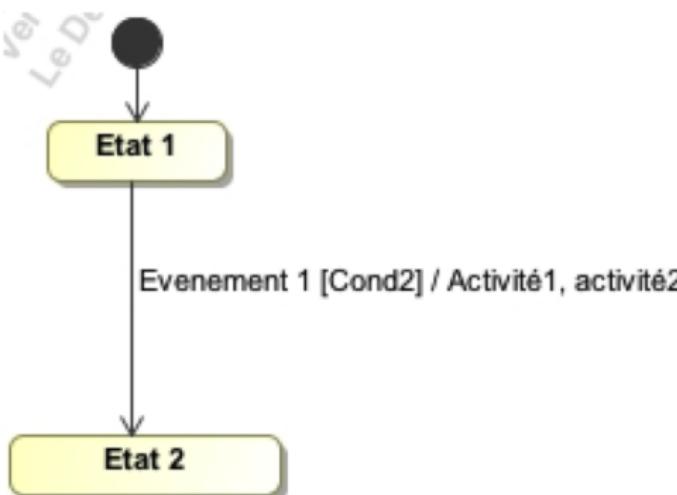
## Effet d'une transition

- Lorsqu'une transition se déclenche, son effet spécifié par "**/activité**" dans la syntaxe, s'exécute. Il s'agit généralement d'une activité qui peut être :
  - une opération primitive comme une instruction d'assignation ;
  - l'envoi d'un signal ;
  - l'appel d'une opération ;
  - une liste d'activités, etc.
- La façon de spécifier l'activité à réaliser est laissée libre (langage naturel ou pseudo-code). Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.



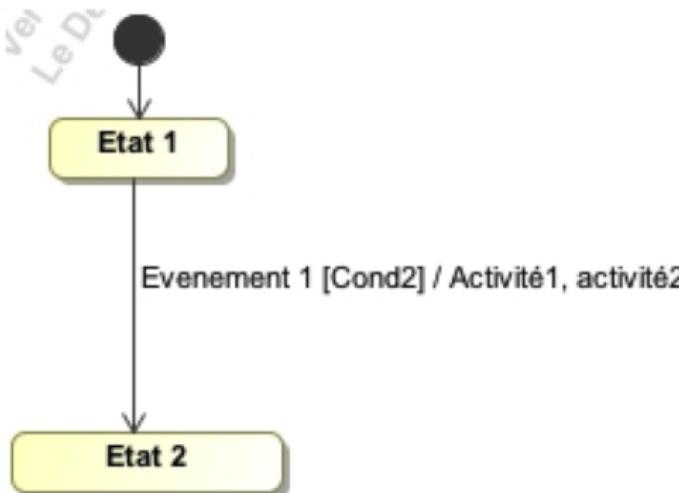
## Effet d'une transition

- Lorsqu'une transition se déclenche, son effet spécifié par "**/activité**" dans la syntaxe, s'exécute. Il s'agit généralement d'une activité qui peut être :
  - une opération primitive comme une instruction d'assignation ;
  - l'envoi d'un signal ;
  - l'appel d'une opération ;
  - une liste d'activités, etc.
- La façon de spécifier l'activité à réaliser est laissée libre (langage naturel ou pseudo-code). Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.



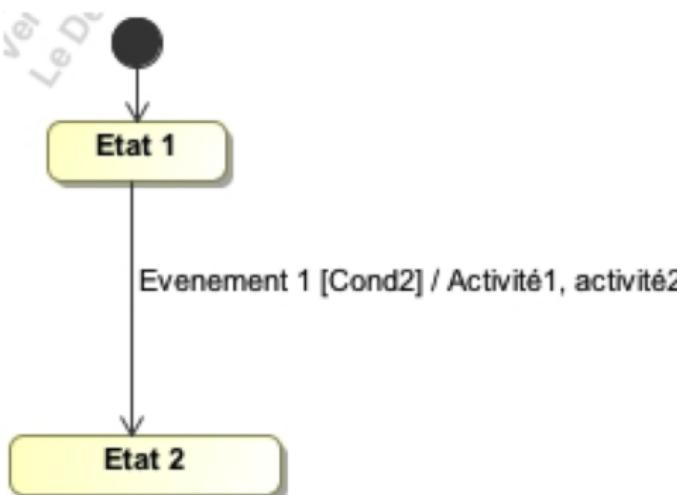
## Effet d'une transition

- Lorsqu'une transition se déclenche, son effet spécifié par "**/activité**" dans la syntaxe, s'exécute. Il s'agit généralement d'une activité qui peut être :
  - une opération primitive comme une instruction d'assignation ;
  - l'envoi d'un signal ;
  - l'appel d'une opération ;
  - une liste d'activités, etc.
- La façon de spécifier l'activité à réaliser est laissée libre (langage naturel ou pseudo-code). Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.



## Effet d'une transition

- Lorsqu'une transition se déclenche, son effet spécifié par "**/activité**" dans la syntaxe, s'exécute. Il s'agit généralement d'une activité qui peut être :
  - une opération primitive comme une instruction d'assignation ;
  - l'envoi d'un signal ;
  - l'appel d'une opération ;
  - une liste d'activités, etc.
- La façon de spécifier l'activité à réaliser est laissée libre (langage naturel ou pseudo-code). Lorsque l'exécution de l'effet est terminée, l'état cible de la transition devient actif.





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

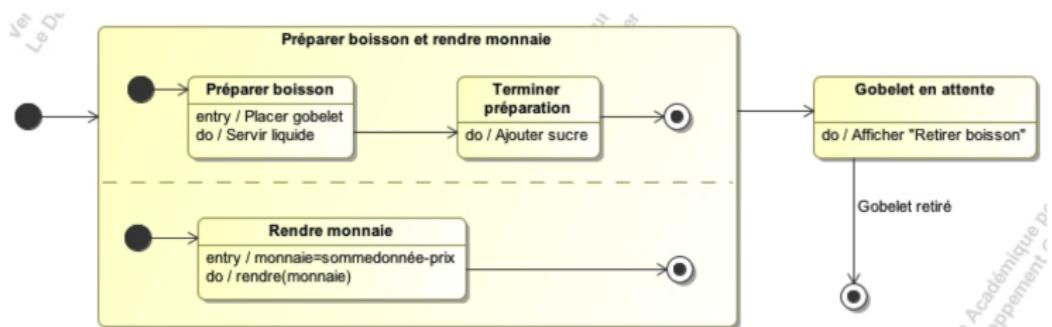
## 6 Diagramme d'activités

- Définition
- Exemples d'applications

## États orthogonaux

### États orthogonaux

- Un état **orthogonal** est un état composite comportant plus d'une région représentant chacune d'elle un flot d'exécution. Ces états permettent de décrire efficacement les mécanismes concurrents.
- Chaque région peut posséder un état initial et final. Une transition qui atteint la bordure d'un état composite orthogonal est équivalente à une transition qui atteint les états initiaux de toutes ses régions concurrentes.
- Toutes les régions concurrentes d'un état composite orthogonal doivent atteindre leur état final pour que l'état composite soit considéré comme terminé.

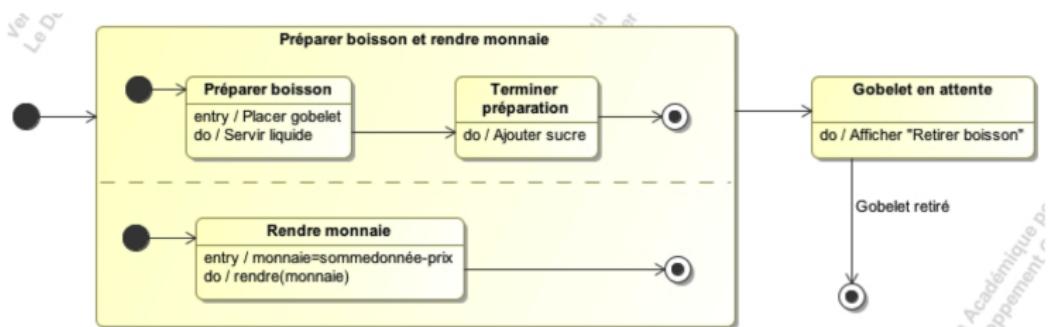




## États orthogonaux

### États orthogonaux

- Un état **orthogonal** est un état composite comportant plus d'une région représentant chacune d'elle un flot d'exécution. Ces états permettent de décrire efficacement les mécanismes concurrents.
- Chaque région peut posséder un état initial et final. Une transition qui atteint la bordure d'un état composite orthogonal est équivalente à une transition qui atteint les états initiaux de toutes ses régions concurrents.
- Toutes les régions concurrentes d'un état composite orthogonal doivent atteindre leur état final pour que l'état composite soit considéré comme terminé.

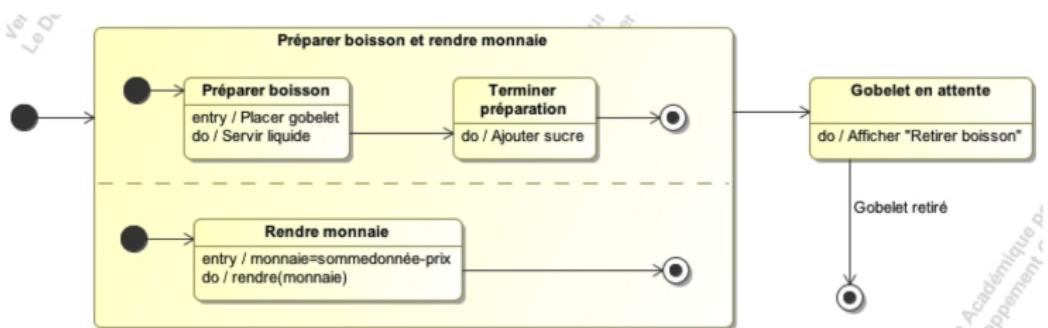




## États orthogonaux

### États orthogonaux

- Un état **orthogonal** est un état composite comportant plus d'une région représentant chacune d'elle un flot d'exécution. Ces états permettent de décrire efficacement les mécanismes concurrents.
- Chaque région peut posséder un état initial et final. Une transition qui atteint la bordure d'un état composite orthogonal est équivalente à une transition qui atteint les états initiaux de toutes ses régions concurrentes.
- Toutes les régions concurrentes d'un état composite orthogonal doivent atteindre leur état final pour que l'état composite soit considéré comme terminé.



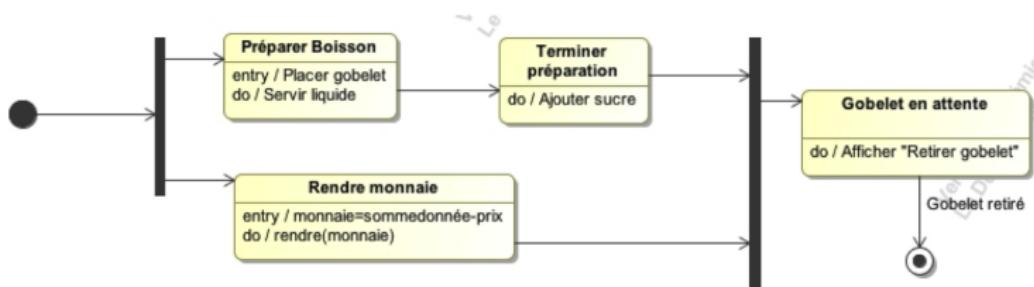


## Transitions complexes ou concurrentes

### Transitions complexes ou concurrentes

Les **transitions complexes ou concurrentes** sont représentées par une barre épaisse et peuvent éventuellement être nommées.

- Dès le franchissement de la bifurcation les états immédiatement suivants sont activés.
- L'union ne peut être franchies que lorsque toutes les activités des états immédiatement précédents sont terminées.





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

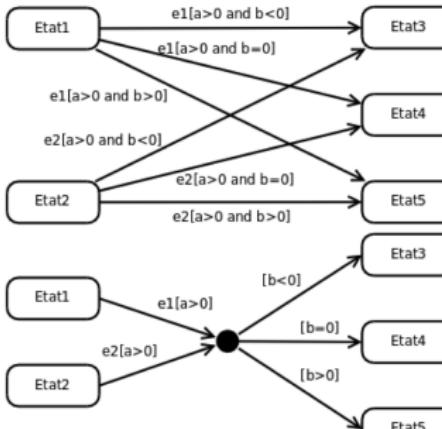
- Définition
- Exemples d'applications



## Point de jonction

### Point de jonction

- Un **point de jonction** permet de partager des segments de transition.
- Un point de jonction peut avoir plusieurs segments de transition entrante et plusieurs segments de transition sortante. Par contre il ne peut avoir d'activité interne ni des transitions sortantes dotées de déclencheurs d'événements.
- Il ne s'agit pas d'un état qui peut être actif au cours d'un laps de temps fini.
- Toutes les gardes le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.

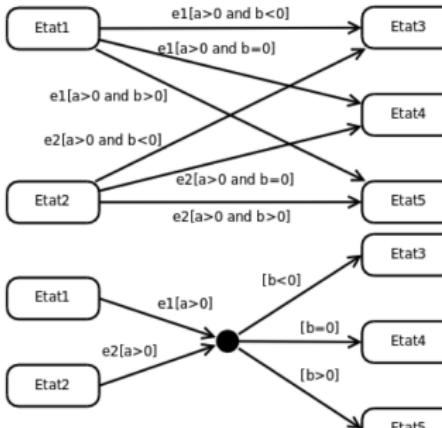




## Point de jonction

### Point de jonction

- Un **point de jonction** permet de partager des segments de transition.
- Un point de jonction peut avoir plusieurs segments de transition entrante et plusieurs segments de transition sortante. Par contre il ne peut avoir d'activité interne ni des transitions sortantes dotées de déclencheurs d'événements.
- Il ne s'agit pas d'un état qui peut être actif au cours d'un laps de temps fini.
- Toutes les gardes le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.

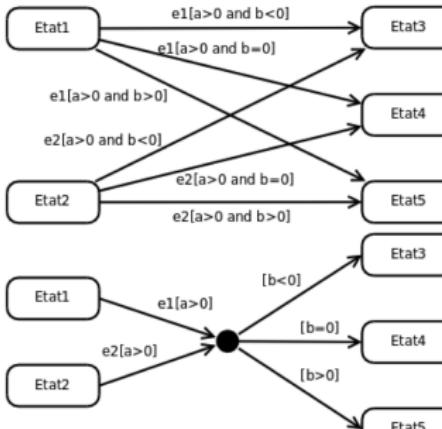




## Point de jonction

### Point de jonction

- Un **point de jonction** permet de partager des segments de transition.
- Un point de jonction peut avoir plusieurs segments de transition entrante et plusieurs segments de transition sortante. Par contre il ne peut avoir d'activité interne ni des transitions sortantes dotées de déclencheurs d'événements.
- Il ne s'agit pas d'un état qui peut être actif au cours d'un laps de temps fini.
- Toutes les gardes le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.

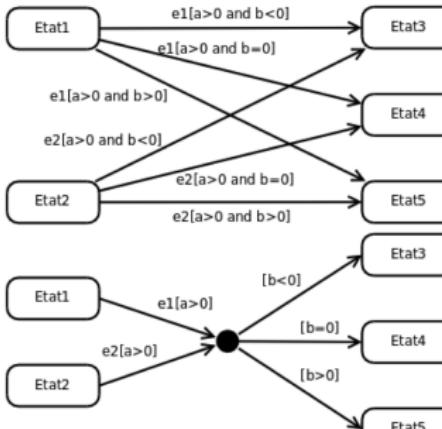




## Point de jonction

### Point de jonction

- Un **point de jonction** permet de partager des segments de transition.
- Un point de jonction peut avoir plusieurs segments de transition entrante et plusieurs segments de transition sortante. Par contre il ne peut avoir d'activité interne ni des transitions sortantes dotées de déclencheurs d'événements.
- Il ne s'agit pas d'un état qui peut être actif au cours d'un laps de temps fini.
- Toutes les gardes le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.

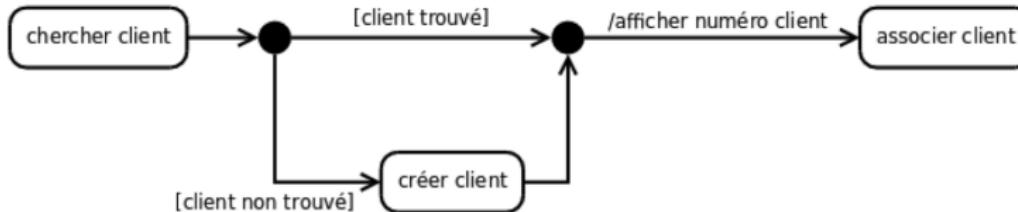




## Point de jonction

### Point de jonction

- Un **point de jonction** permet de partager des segments de transition.
- Un point de jonction peut avoir plusieurs segments de transition entrante et plusieurs segments de transition sortante. Par contre il ne peut avoir d'activité interne ni des transitions sortantes dotées de déclencheurs d'événements.
- Il ne s'agit pas d'un état qui peut être actif au cours d'un laps de temps fini.
- Toutes les gardes le long de ce chemin doivent s'évaluer à vrai dès le franchissement du premier segment.

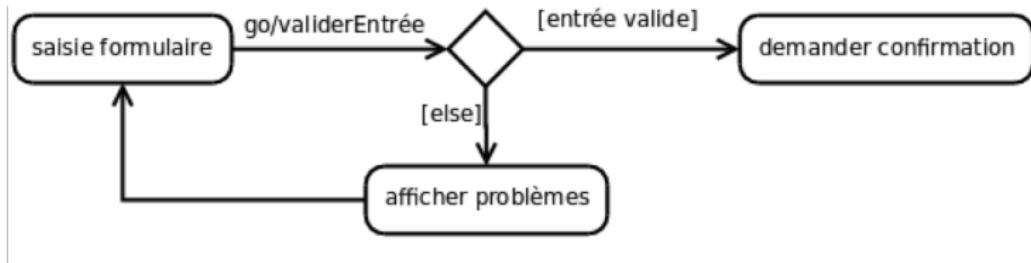




## Point de décision

### Point de décision

- Un point de décision possède une entrée et au moins deux sorties.
- Contrairement à un point de jonction, les gardes situées après le point de décision sont évaluées au moment où il est atteint.
- Cela permet de baser le choix sur des résultats obtenus en franchissant le segment avant le point de choix. Si, quand le point de décision est atteint, aucun segment en aval n'est franchissable, c'est que le modèle est mal formé.
- Condition exclusive avec la garde : [else], sur un des segments en aval d'un point de choix. Ce segment n'est franchissable que si les gardes des autres segments sont toutes fausses.

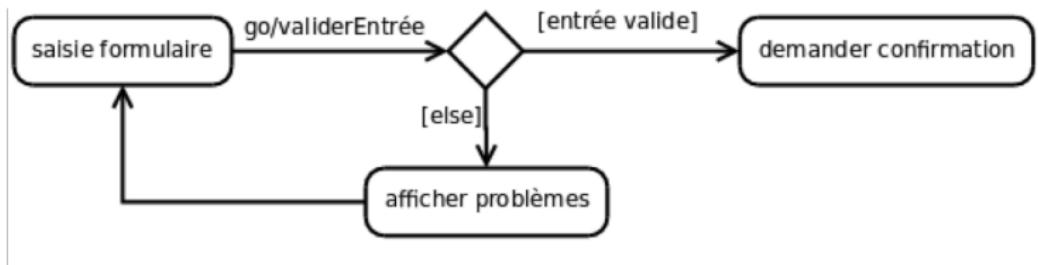




## Point de décision

### Point de décision

- Un **point de décision** possède une entrée et au moins deux sorties.
- Contrairement à un point de jonction, les gardes situées après le point de décision sont évaluées au moment où il est atteint.
- Cela permet de baser le choix sur des résultats obtenus en franchissant le segment avant le point de choix. Si, quand le point de décision est atteint, aucun segment en aval n'est franchissable, c'est que le modèle est mal formé.
- Condition exclusive avec la garde : **[else]**, sur un des segments en aval d'un point de choix. Ce segment n'est franchissable que si les gardes des autres segments sont toutes fausses.

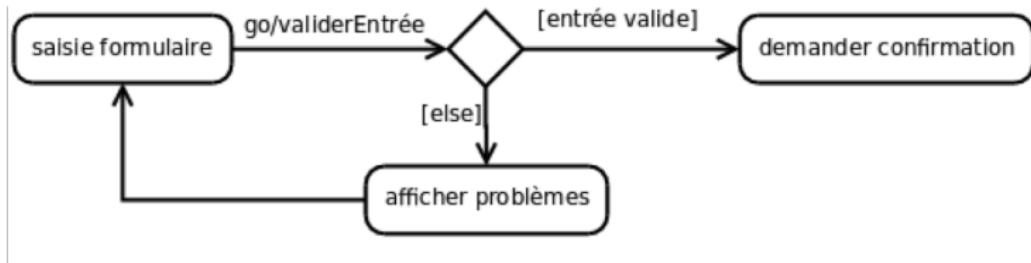




## Point de décision

### Point de décision

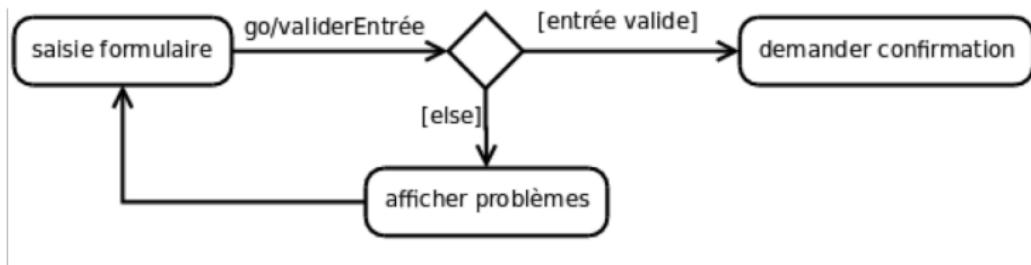
- Un point de décision possède une entrée et au moins deux sorties.
- Contrairement à un point de jonction, les gardes situées après le point de décision sont évaluées au moment où il est atteint.
- Cela permet de baser le choix sur des résultats obtenus en franchissant le segment avant le point de décision. Si, quand le point de décision est atteint, aucun segment en aval n'est franchissable, c'est que le modèle est mal formé.
- Condition exclusive avec la garde : [else], sur un des segments en aval d'un point de décision, ce segment n'est franchissable que si les gardes des autres segments sont toutes fausses.



## Point de décision

### Point de décision

- Un **point de décision** possède une entrée et au moins deux sorties.
- Contrairement à un point de jonction, les gardes situées après le point de décision sont évaluées au moment où il est atteint.
- Cela permet de baser le choix sur des résultats obtenus en franchissant le segment avant le point de choix. Si, quand le point de décision est atteint, aucun segment en aval n'est franchissable, c'est que le modèle est mal formé.
- Condition exclusive avec la garde : **[else]**, sur un des segments en aval d'un point de choix. Ce segment n'est franchissable que si les gardes des autres segments sont toutes fausses.





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



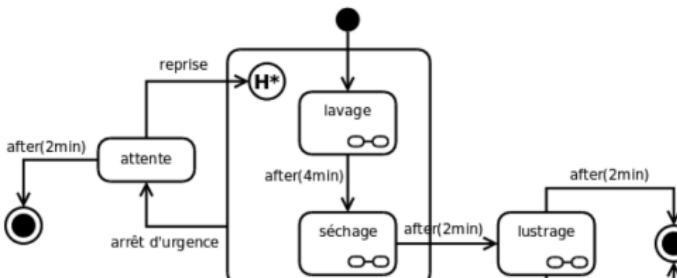
## État historique

### État historique

Un état historique est un pseudo-état qualifié d'*état historique plat*, qui mémorise le dernier sous-état actif d'un état composite. Graphiquement il est représenté par un cercle contenant un **H**.

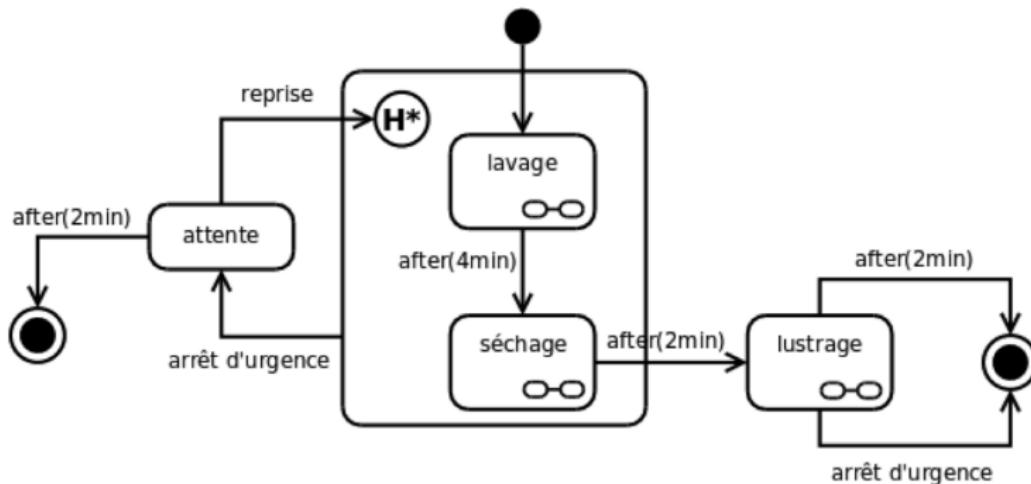
- Une transition ayant pour cible l'état historique est équivalente à une transition qui a pour cible le dernier état visité de l'état englobant.
- Un état historique peut avoir une transition sortante non étiquetée indiquant l'état à exécuter si la région n'a pas encore été visitée.

Il est également possible de définir un *état historique profond* représenté graphiquement par un cercle contenant un **H\***. Il permet d'atteindre le dernier état visité dans la région, quel que soit son niveau d'imbrication, alors que l'état historique plat limite l'accès aux états de son niveau d'imbrication.





## État historique





# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Diagramme d'activités : définition

### Diagramme d'activité

Le **diagramme d'activité** est un diagramme comportemental appelé *Activity Diagram (act)* dans le langage SysML. Il permet de modéliser le déroulement d'un processus sous la forme d'une activité correspondant à la décomposition séquentielle d'actions aussi appelées tâches. Les éléments graphiques utilisés dans ce diagramme sont principalement :

- des rectangles aux coins arrondis pour les états,
- des flèches orientées de l'état de départ à l'état cible pour les transitions,
- des occurrences attachées à la transition spécifient les conditions de franchissement.

### Remarque

- Ce diagramme ne possède aucun événement associé aux transitions entre actions : la fin d'une action implique automatiquement le passage à la suivante, donc dans un ordre déterminé d'actions menant à un résultat. Lorsque le processus est enclenché il va à son terme selon un ordre précis.
- Ce diagramme permet aussi de représenter des **algorigrammes**, c'est à dire un flux de contrôle.
- Ce diagramme ne figure pas explicitement dans le programme.



## Diagramme d'activités : définition

### Diagramme d'activité

Le **diagramme d'activité** est un diagramme comportemental appelé *Activity Diagram (act)* dans le langage SysML. Il permet de modéliser le déroulement d'un processus sous la forme d'une activité correspondant à la décomposition séquentielle d'actions aussi appelées tâches. Les éléments graphiques utilisés dans ce diagramme sont principalement :

- des rectangles aux coins arrondis pour les états,
- des flèches orientées de l'état de départ à l'état cible pour les transitions,
- des occurrences attachées à la transition spécifient les conditions de franchissement.

### Remarque

- Ce diagramme ne possède aucun événement associé aux transitions entre actions : la fin d'une action implique automatiquement le passage à la suivante, donc dans un ordre déterminé d'actions menant à un résultat. Lorsque le processus est enclenché il va à son terme selon un ordre précis.
- Ce diagramme permet aussi de représenter des **algorigrammes**, c'est à dire un flux de contrôle.
- Ce diagramme ne figure pas explicitement dans le programme.

# Plan

## 1 Introduction

- Rôle d'un système de commande
- Informations binaires et numériques
- Systèmes asservis numériques

## 2 Manipulation de l'information binaire

- Bases de logiques combinatoires

## 3 Introduction : système à évènements discrets

- Modélisation de la commande des systèmes
- Modélisation à l'aide de l'outil SysML

## 4 Diagramme d'états : les fondamentaux

- Introduction
- États
- Transition

## 5 Diagramme d'états : pour aller plus loin

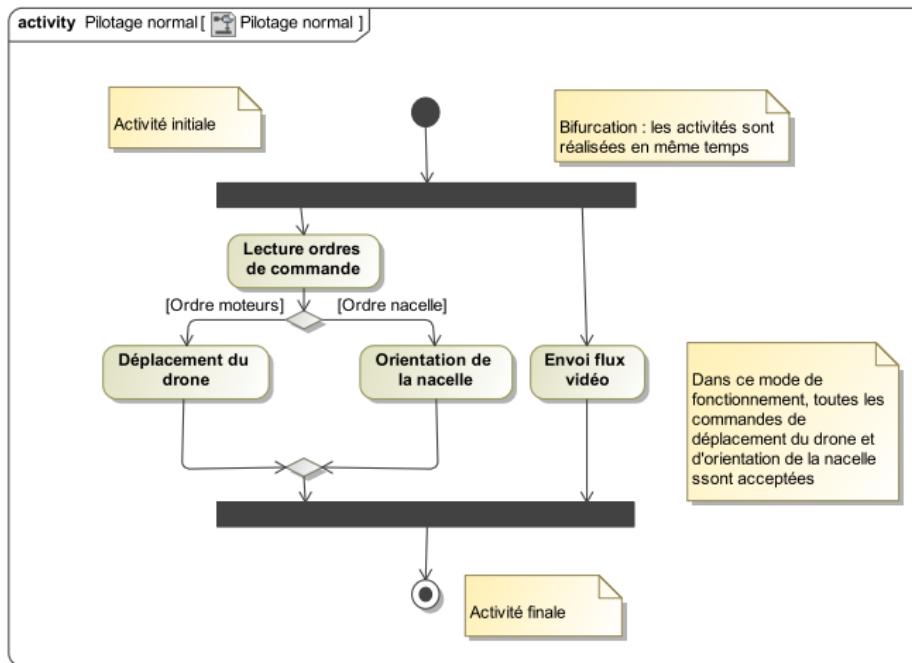
- Concurrence
- Points de choix
- État historique

## 6 Diagramme d'activités

- Définition
- Exemples d'applications



## Diagramme d'activités : exemples





## Diagramme d'activités : exemples

