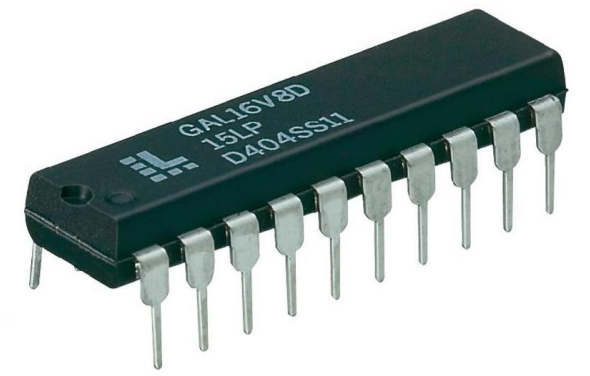
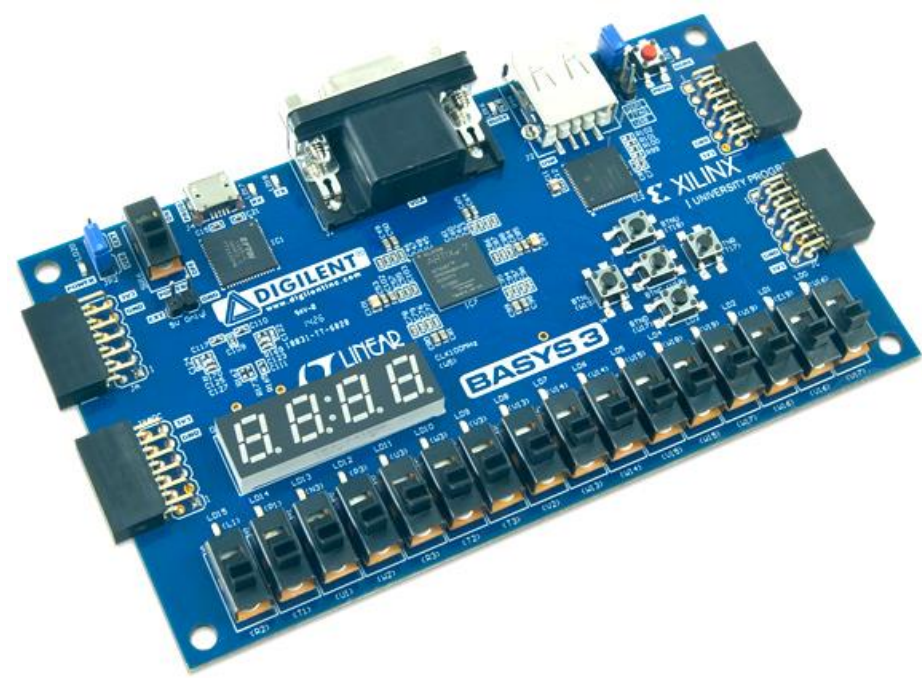


Logique Séquentielle



Logique Séquentielle

Compétences attendues :

- ✓ Interpréter tout ou partie de l'évolution temporelle d'un système séquentiel.
- ✓ Décrire le comportement d'un système séquentiel.

Systeme séquentiel

Définition

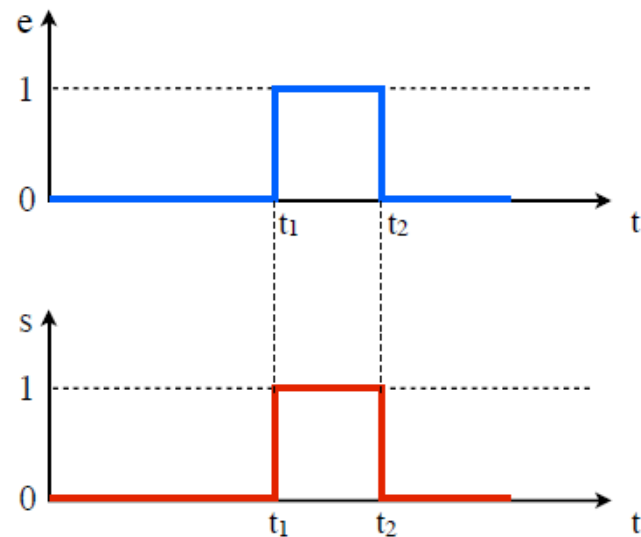
Un système séquentiel, ou à événements discrets, est un système où l'état des sorties S_i dépend de l'état des entrées à l'instant présent, mais aussi de l'histoire de l'évolution des entrées-sorties.

Systeme séquentiel

Exemple : Fonctionnement d'une lampe commandée par un interrupteur ou un contacteur :

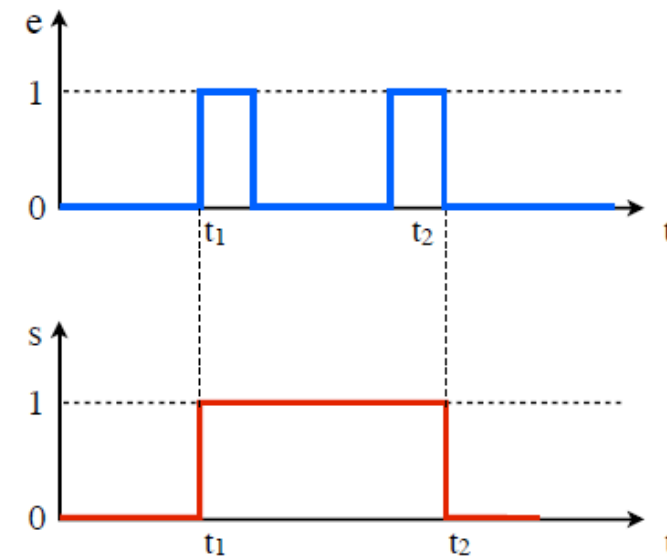
Interrupteur à deux positions (combinatoire)

- 1^e position, la lampe est éteinte ;
- 2^e position, la lampe est allumée.



Contacteur (séquentiel)

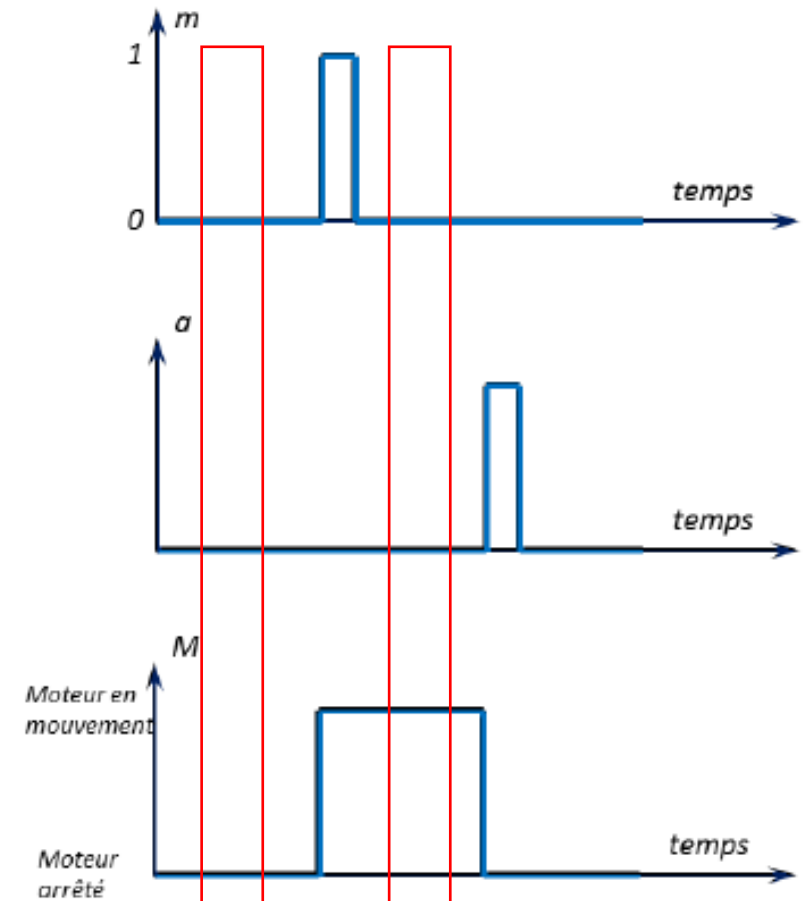
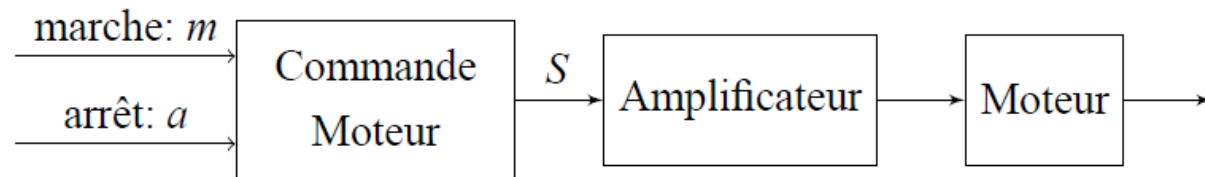
- 1^e impulsion, la lampe s'allume ;
- 2^e impulsion la lampe s'éteint.



Systeme séquentiel

Définition

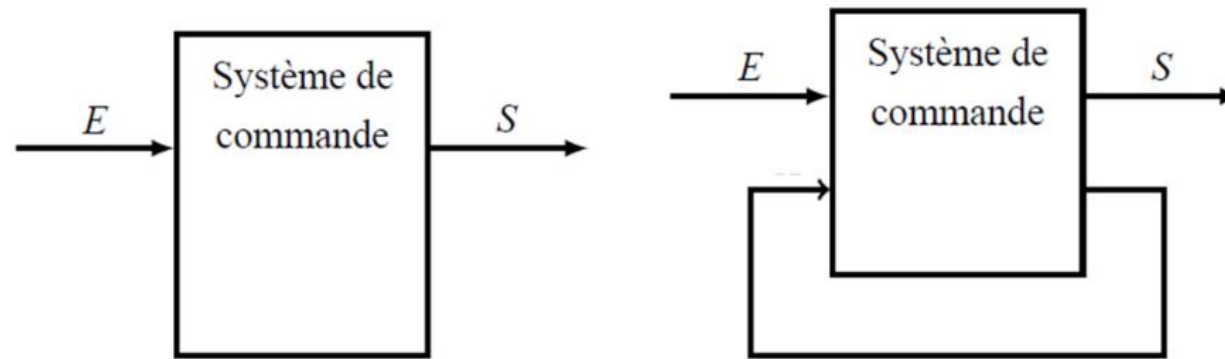
Exemple : Commande d'un moteur électrique par un système séquentiel



Systeme séquentiel

Définition

Systeme \rightarrow mémorise de l'information (état du système)



Représentation d'un système séquentiel par un système combinatoire muni d'un retour d'information sur l'état du système

Systeme séquentiel

Définition

Précision, rapidité, stabilité :

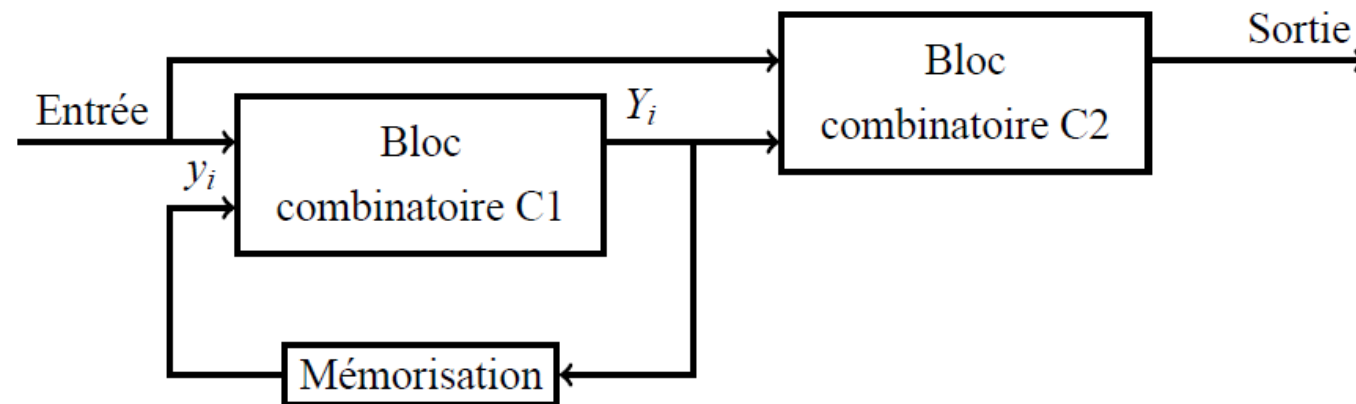
- Systeme respecte **précisément** → CdCF
- Systeme **rapide** → réduire les temps de cycle
- Systeme **stable** → partie commande → ~~situation non prévue~~ (blocage / fonctionnement dégradé)

Systeme séquentiel

Structure d'un systeme séquentiel

Structure d'un systeme séquentiel → 2 blocs fonctionnels combinatoires

Systeme séquentiel → entrées logiques + son état (variables internes = entrées + valeurs mémorisées)



SysML

Drone pour le cinéma

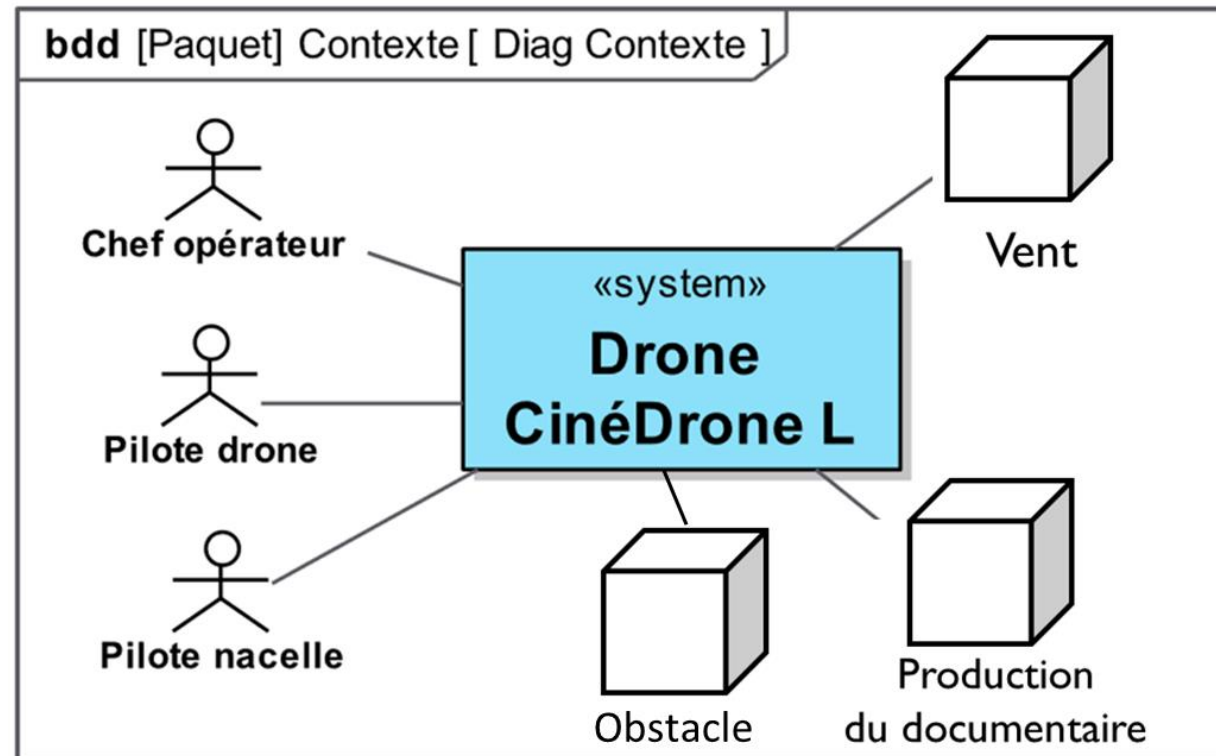


SysML

Diagramme de contexte

EME 1 – verbe d'action – EME 2

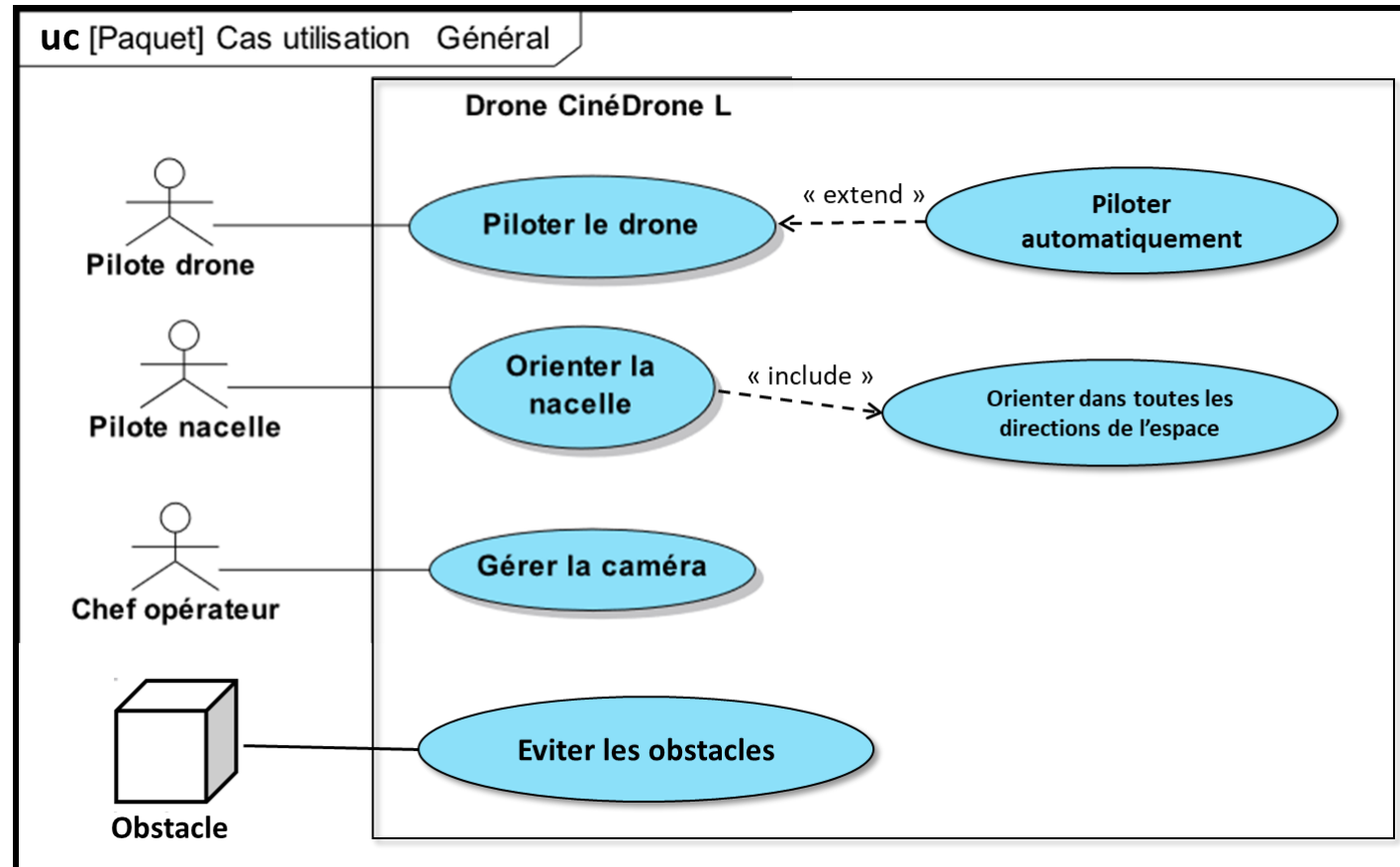
« Quels sont les acteurs et éléments environnants du système ? »



SysML

Diagramme des cas d'utilisation

« Quels services rend le système ? »



SysML

Diagramme d'exigences

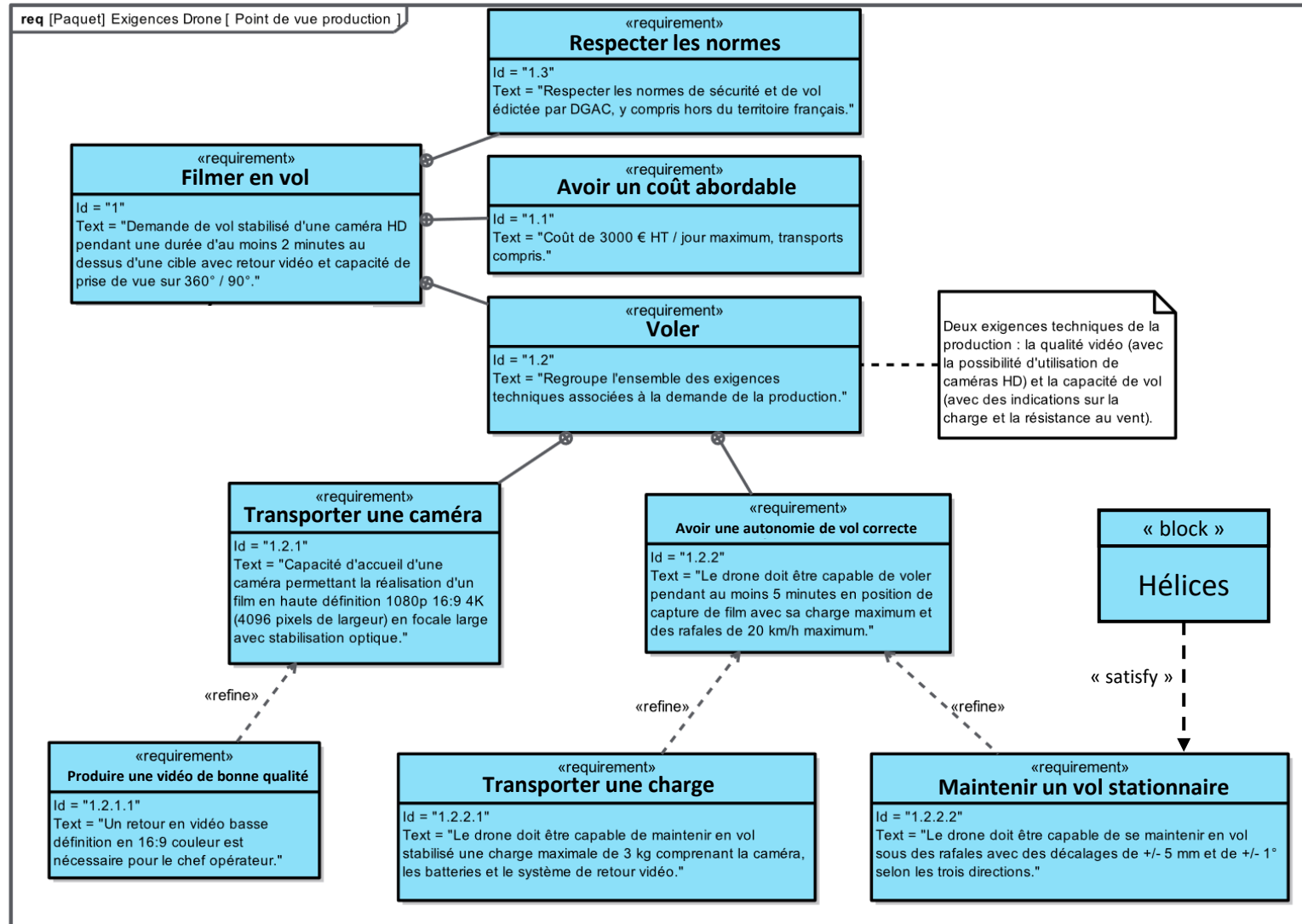
Exigence → spécifier → fonction que doit assurer le système.

Les exigences établissent le contrat entre client et réalisateur du futur système (CdCF).

« Quelles sont les exigences auxquelles le système doit répondre ? »

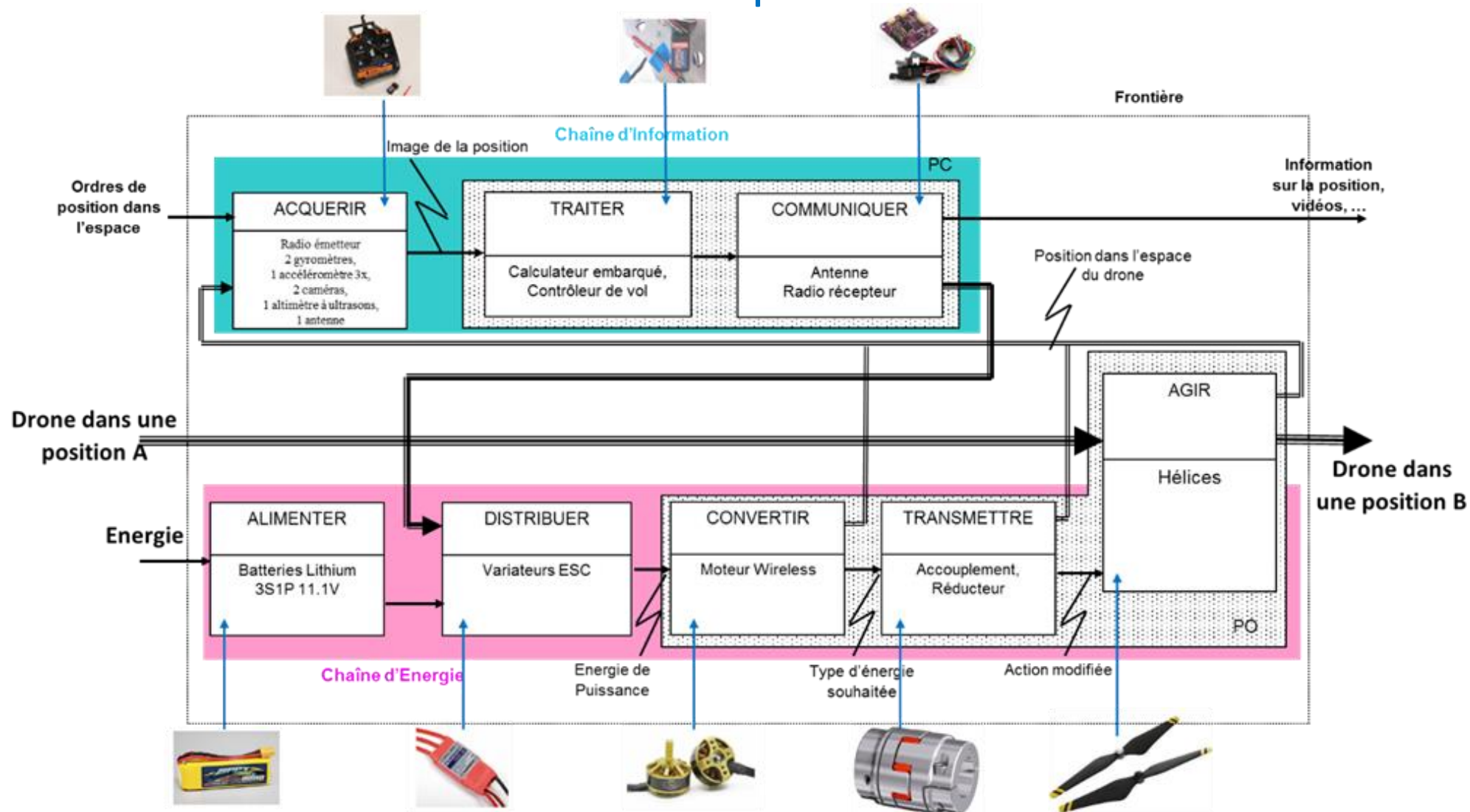
SysML

Diagramme d'exigences



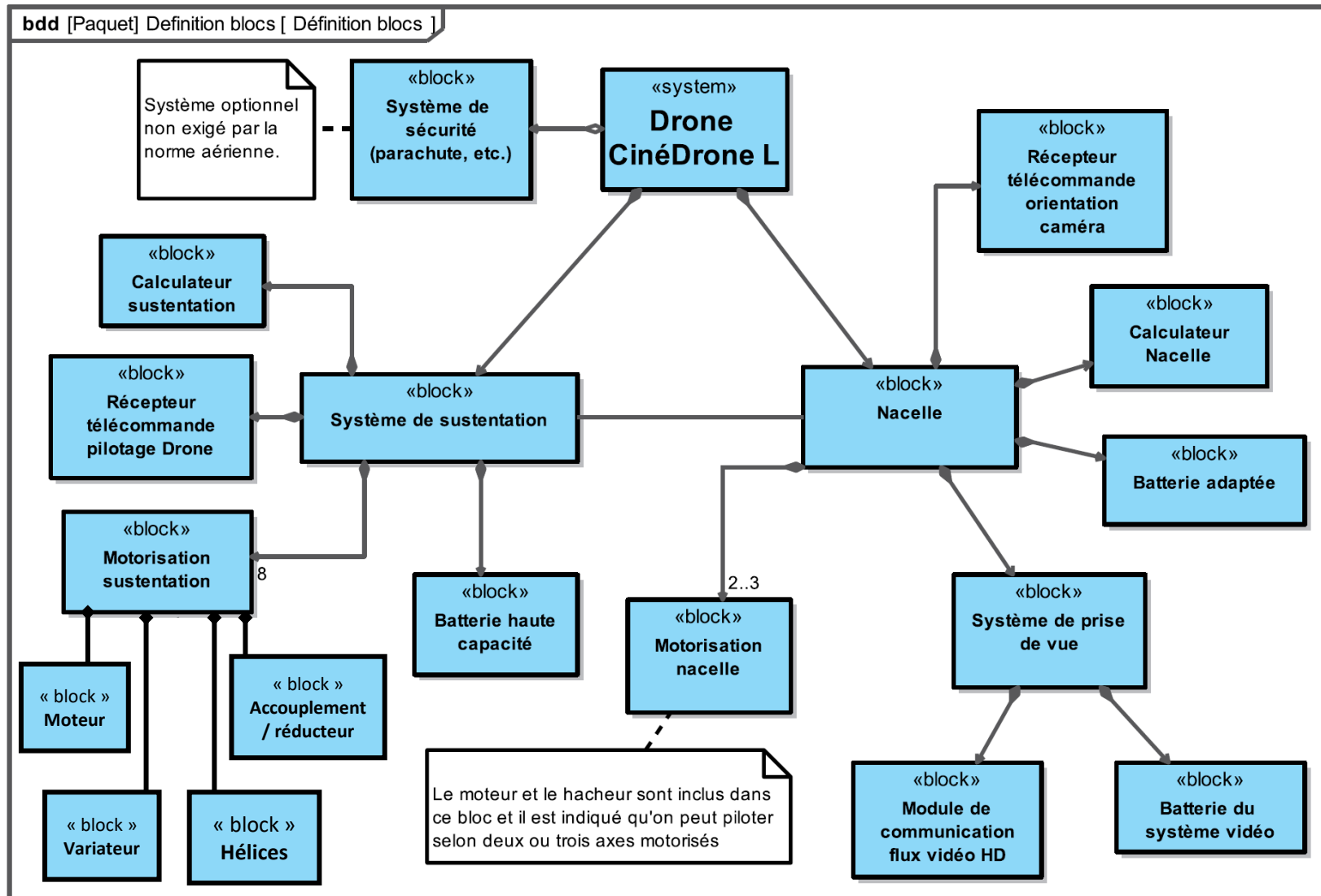
SysML

Chaîne d'information et de puissance



SysML

Diagramme de définition des blocs



SysML

Diagramme de blocs internes

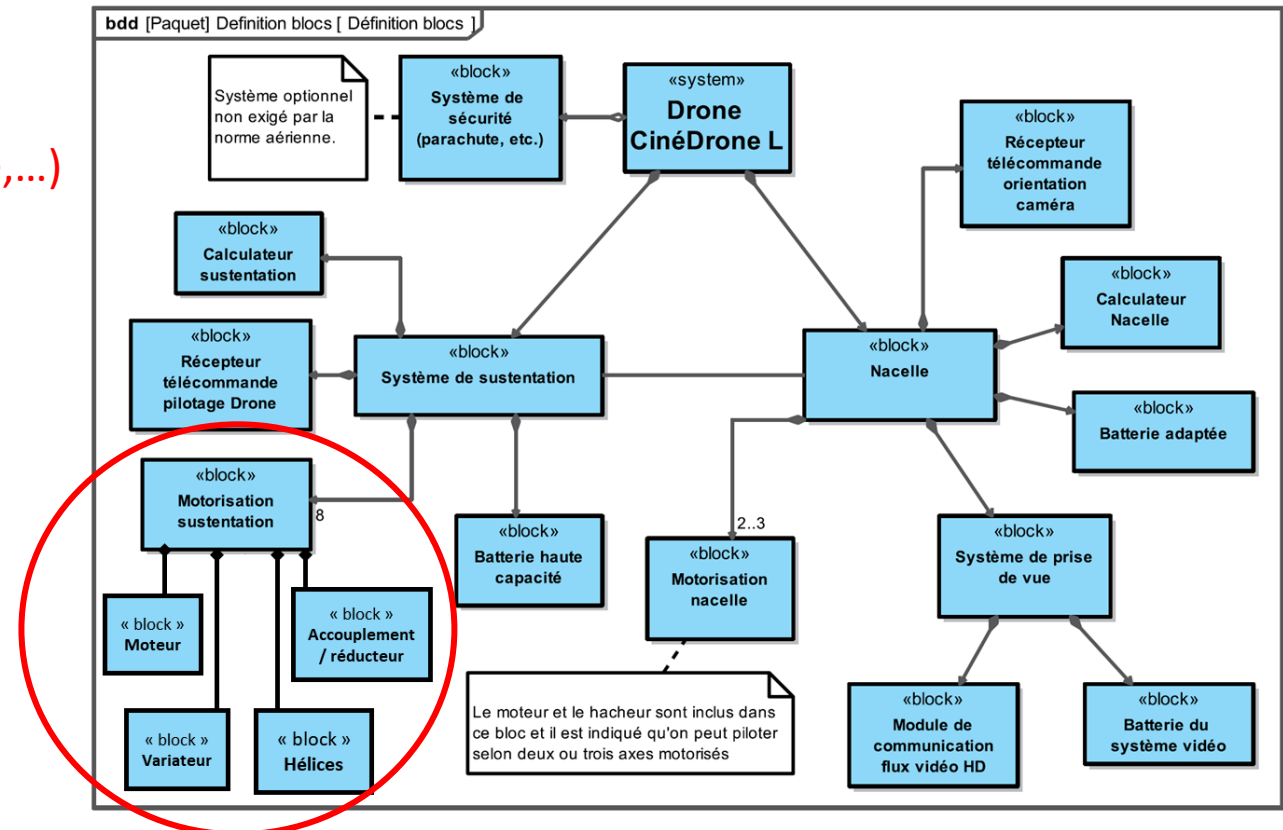
Flux :

- **M**atière (métaux rare, matière composite,...)
- **E**nergie (chaleur, électricité, ...)
- **I**nformation (signaux analogique ou numérique, binaire,...)

Diagramme de définition de bloc (bdd)

PARTIE MOTORISATION

« Comment les blocs interagissent-ils ? »



SysML

Diagramme de blocs internes

« Comment les blocs interagissent-ils ? »

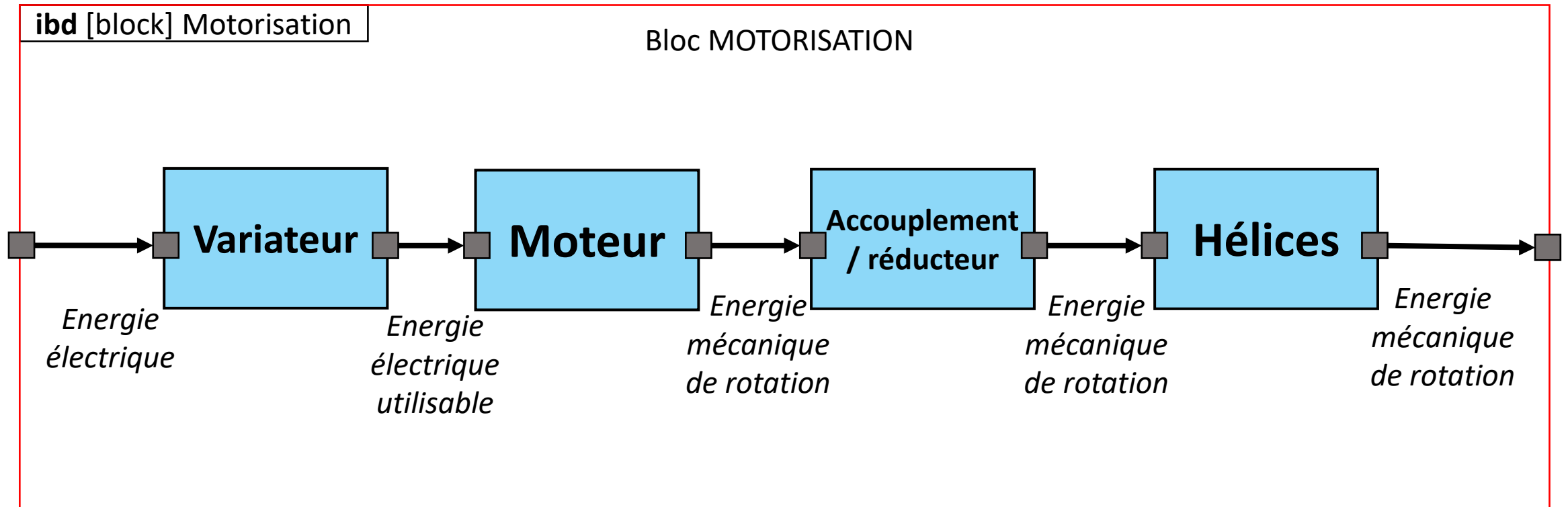
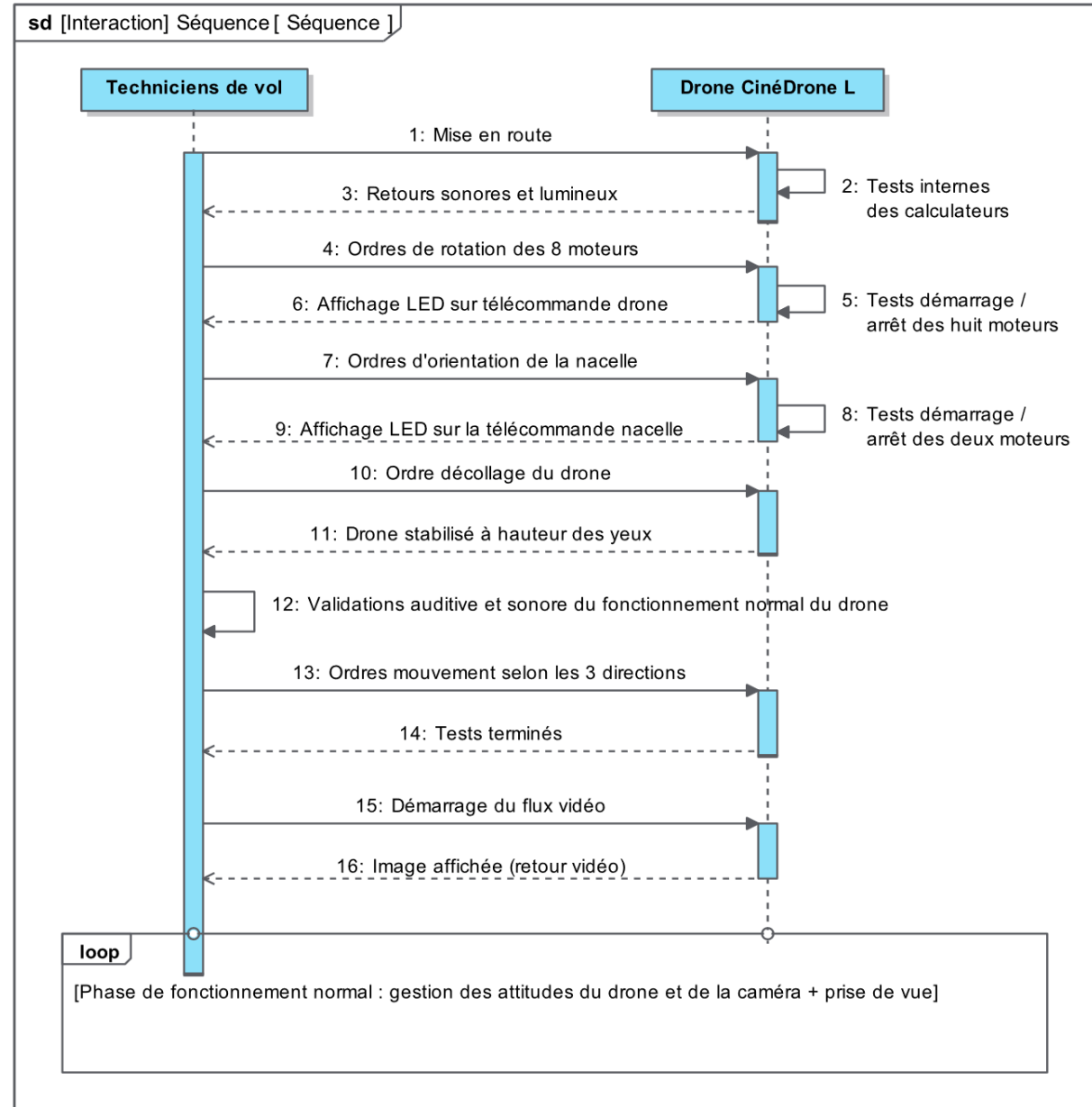


Diagramme de séquence (sd)

A tout cas d'utilisation correspond au moins un diagramme de séquence.

« Comment est réalisé ce cas d'utilisation ? »

Diagramme de séquence (sd)



« Comment est réalisé ce cas d'utilisation ? »

Diagramme de séquence (sd)

Les différents échanges :

- Message **asynchrone** : pas de réponse attendue
- Messages **synchrones** : réponse attendue
- Message **réflexif** : synchrone par essence

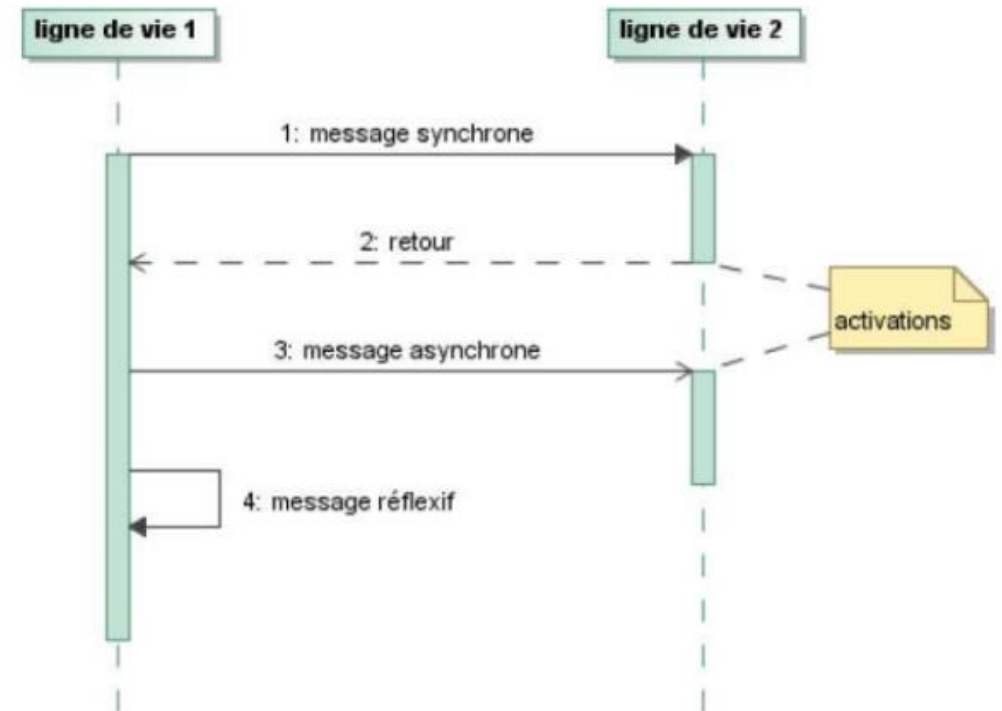


Diagramme de séquence (sd)

Fragments combinés :

- **loop** (boucle) : le fragment s'exécutera plusieurs fois, la **condition de garde** explicitant jusqu'à quand.
- **opt** (option) : le fragment ne se déroule que si la condition est vraie.
- **alt** (alternative) : seul le fragment dont la condition est vraie se déroule.

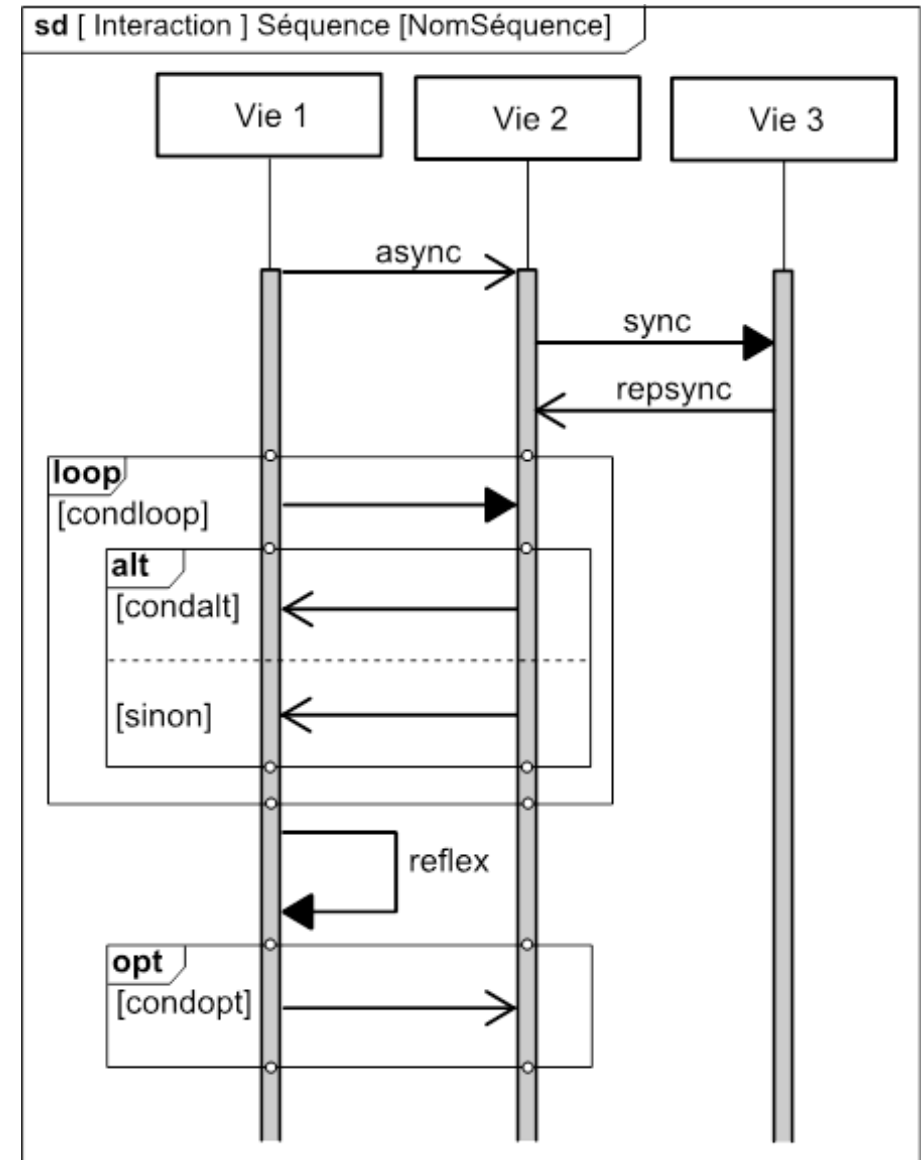


Diagramme de séquence (sd)

Exemple : Réglage du Radio Réveil

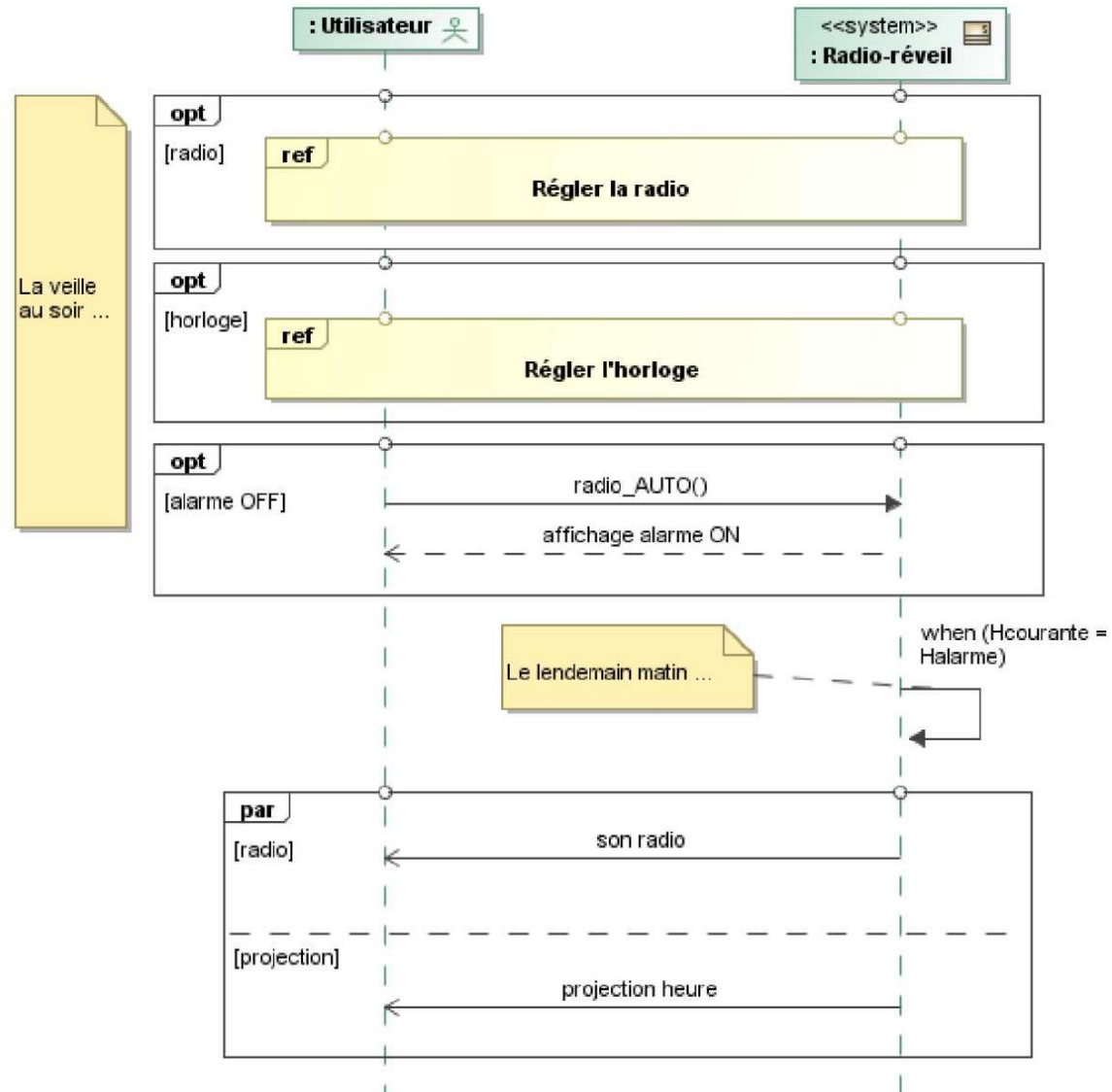


Diagramme d'états (stm)

Diagramme d'états → décrire le fonctionnement d'un programme → machine d'états.

Il montre les différents états pris par le système (ou un sous-système) en fonction des interactions.

« Comment représenter les différents états du système ? »

Les **états** sont **encadrés**, les **évènements** sont représentés avec des **flèches** et un texte les décrit.

Les transitions sont liées aux évènements et sont réalisées lorsque les évènements associés ont lieu.

Point de départ



Point de fin



Diagramme d'états (stm)

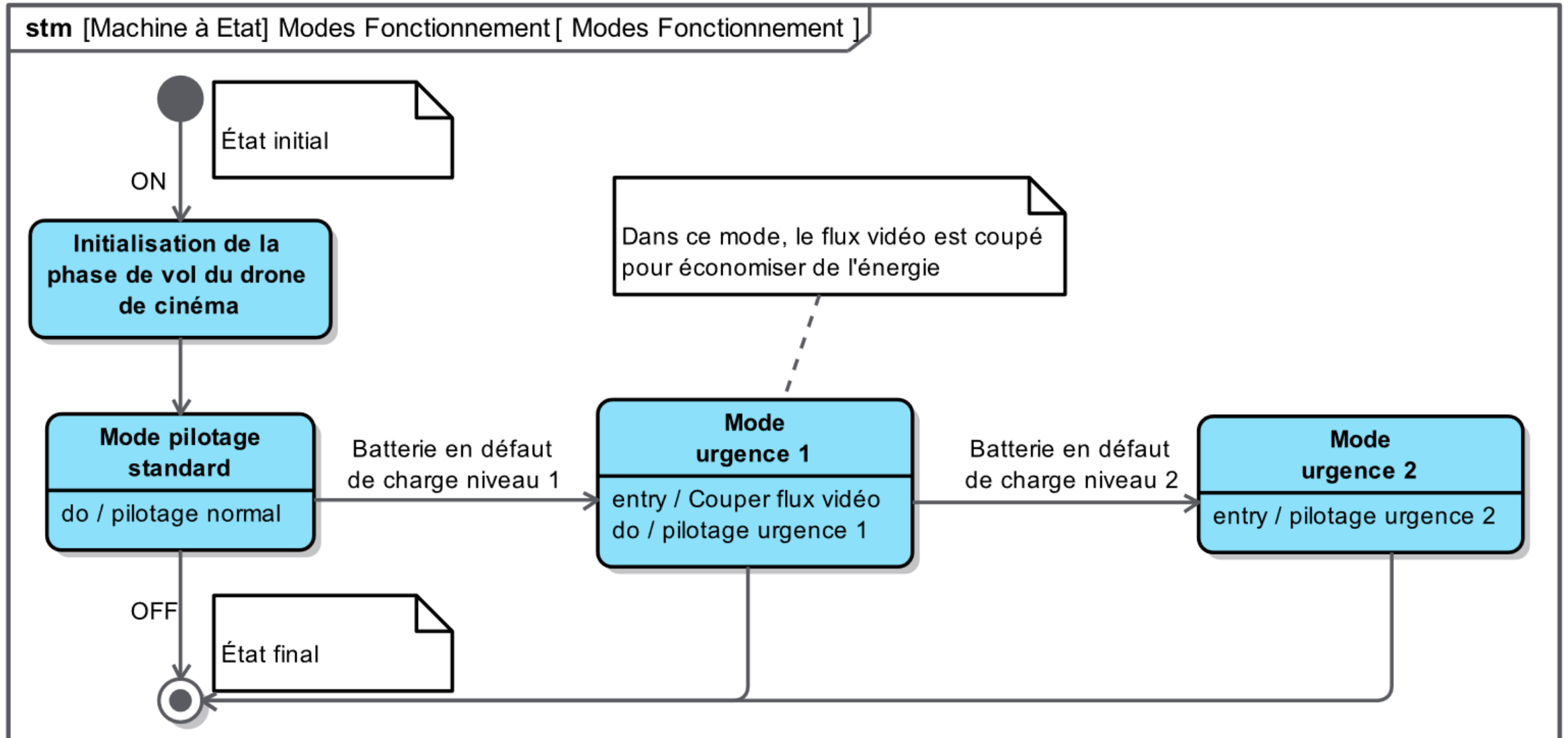
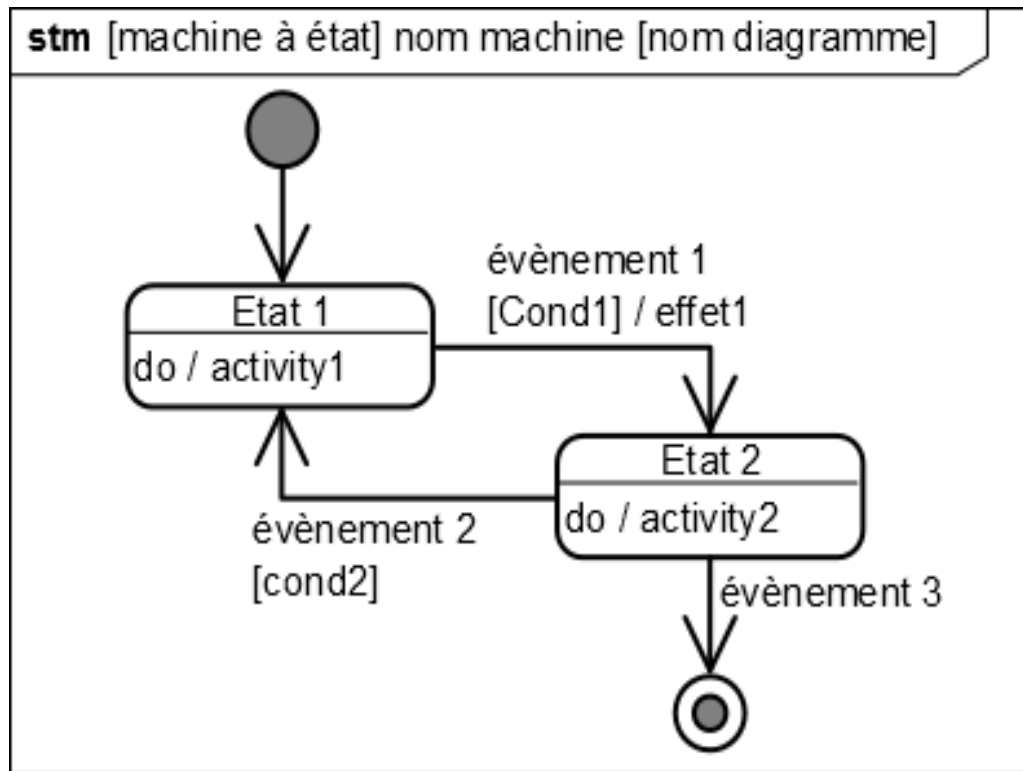


Diagramme d'états (stm)

Diagrammes d'état / transition :



Machines d'état → algorithmes

Etat (situation **dans la vie d'un bloc**) :

- Il satisfait une certaine condition
- Il exécute une certaine activité
- Il attend un certain évènement

Une transition possède :

- Un évènement déclencheur
- Une condition de garde
- Un effet

Diagramme d'états (stm)

Diagrammes d'état / transition :

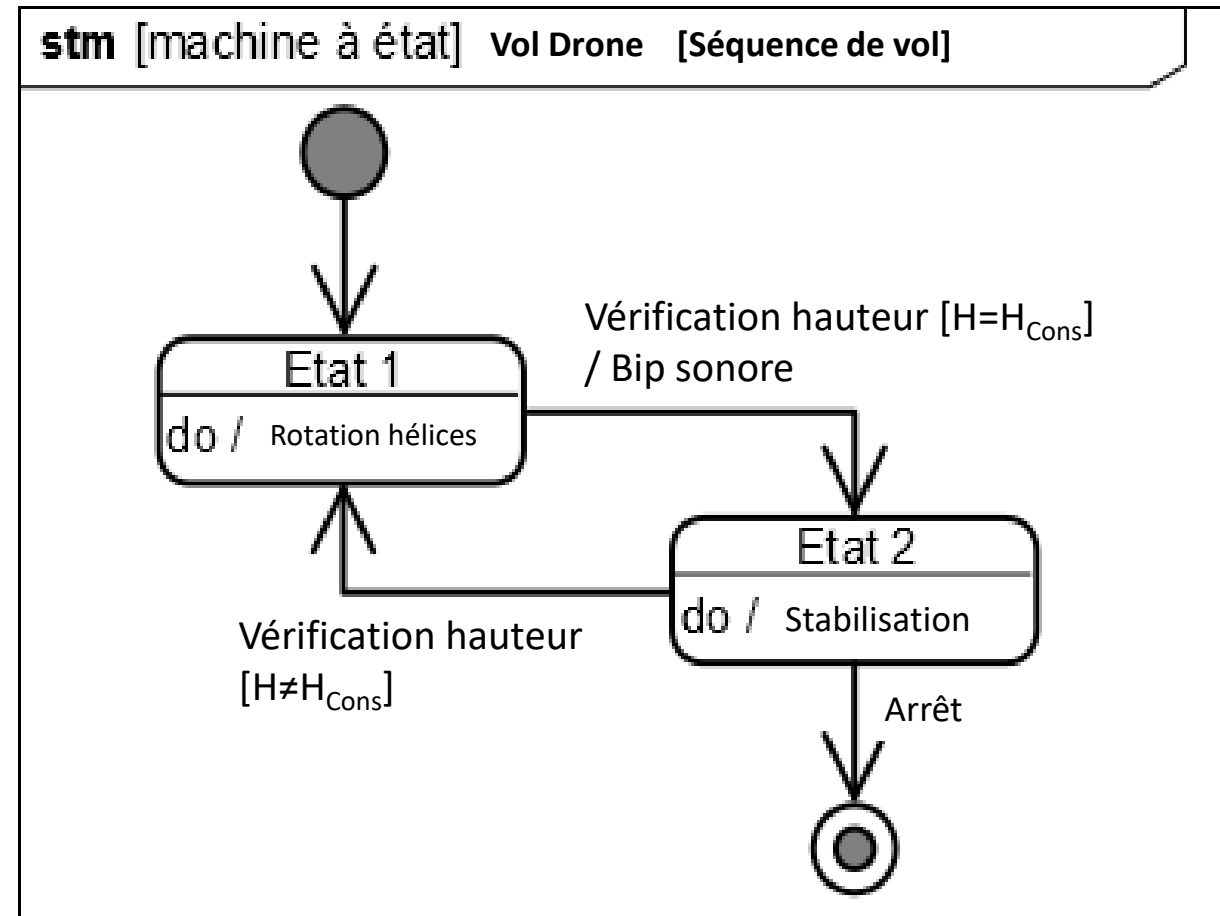


Diagramme d'états (stm)

Diagrammes d'état / transition :

4 types d'événements associés à une transition :

- le message (signal event)
- l'événement temporel (time event) (*after* / *at*)
- l'événement de changement (change event) (*when*)
- l'événement d'appel (call event)

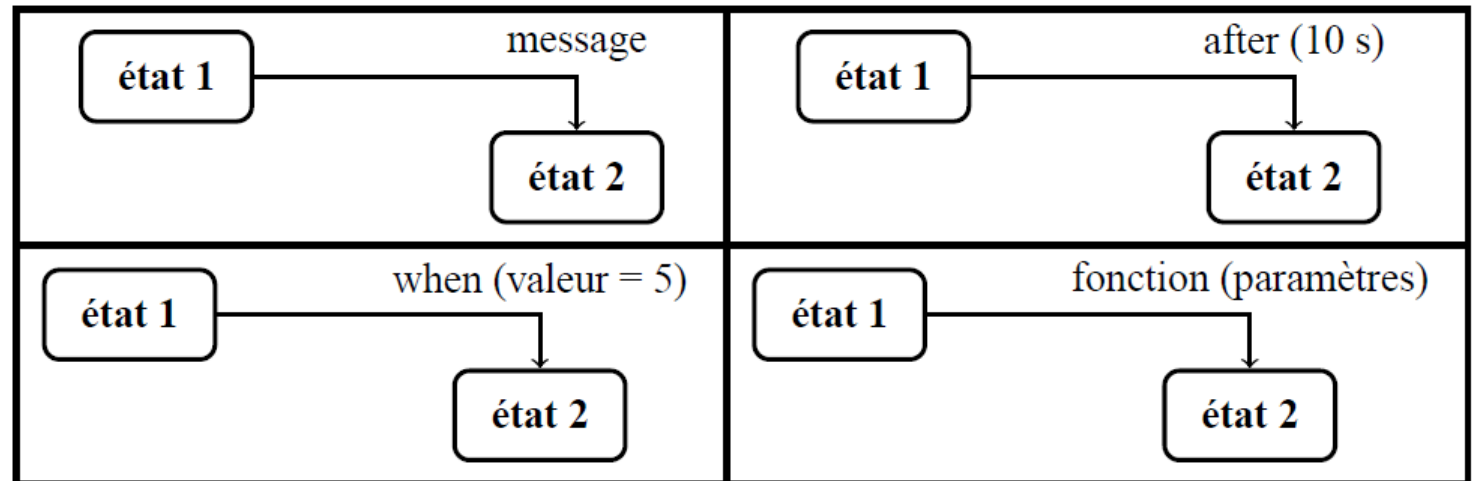
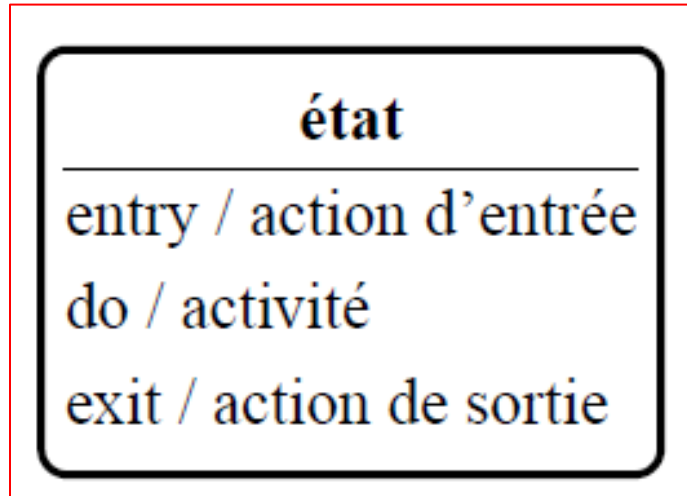


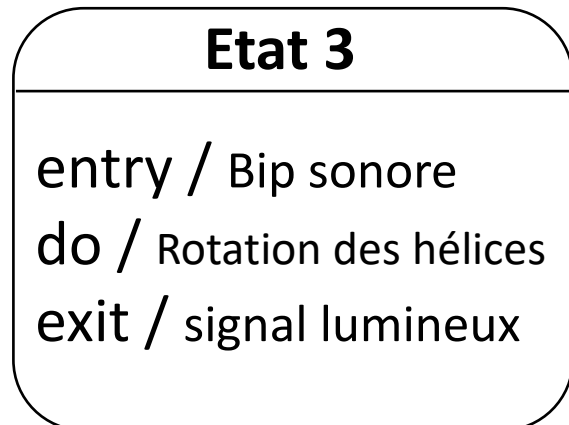
Diagramme d'états (stm)

Diagrammes d'état / transition :



Entrée dans l'état → déclenche → activité associée à *entry*

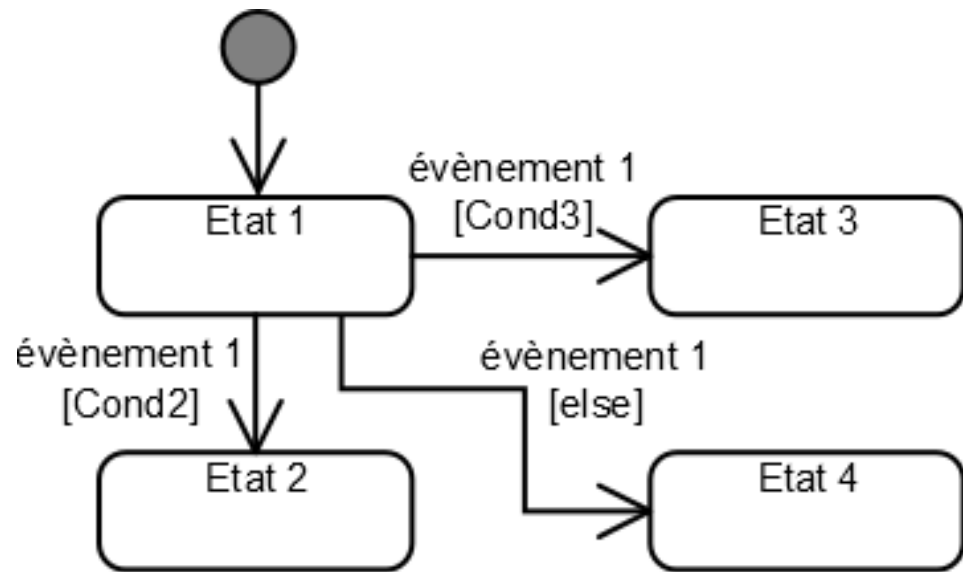
Activité terminée → *do* commence



Sortie de l'état → *do* s'arrête + *exit* commence

Diagramme d'états (stm)

Diagrammes d'état / transition :



Changement d'état → franchissement **instantané** d'une transition entre un état source et un état cible

Condition de franchissement : **événement [garde] / effet**

La transition n'est franchie que si l'évènement ET la condition sont vrais.

Attention : Il faut veiller à ce que les transitions issues d'un même état source soient « exclusives » : il ne doit y avoir qu'une seule possibilité d'évolution à partir d'un état.

Diagramme d'états (stm)

Diagrammes d'état / transition :

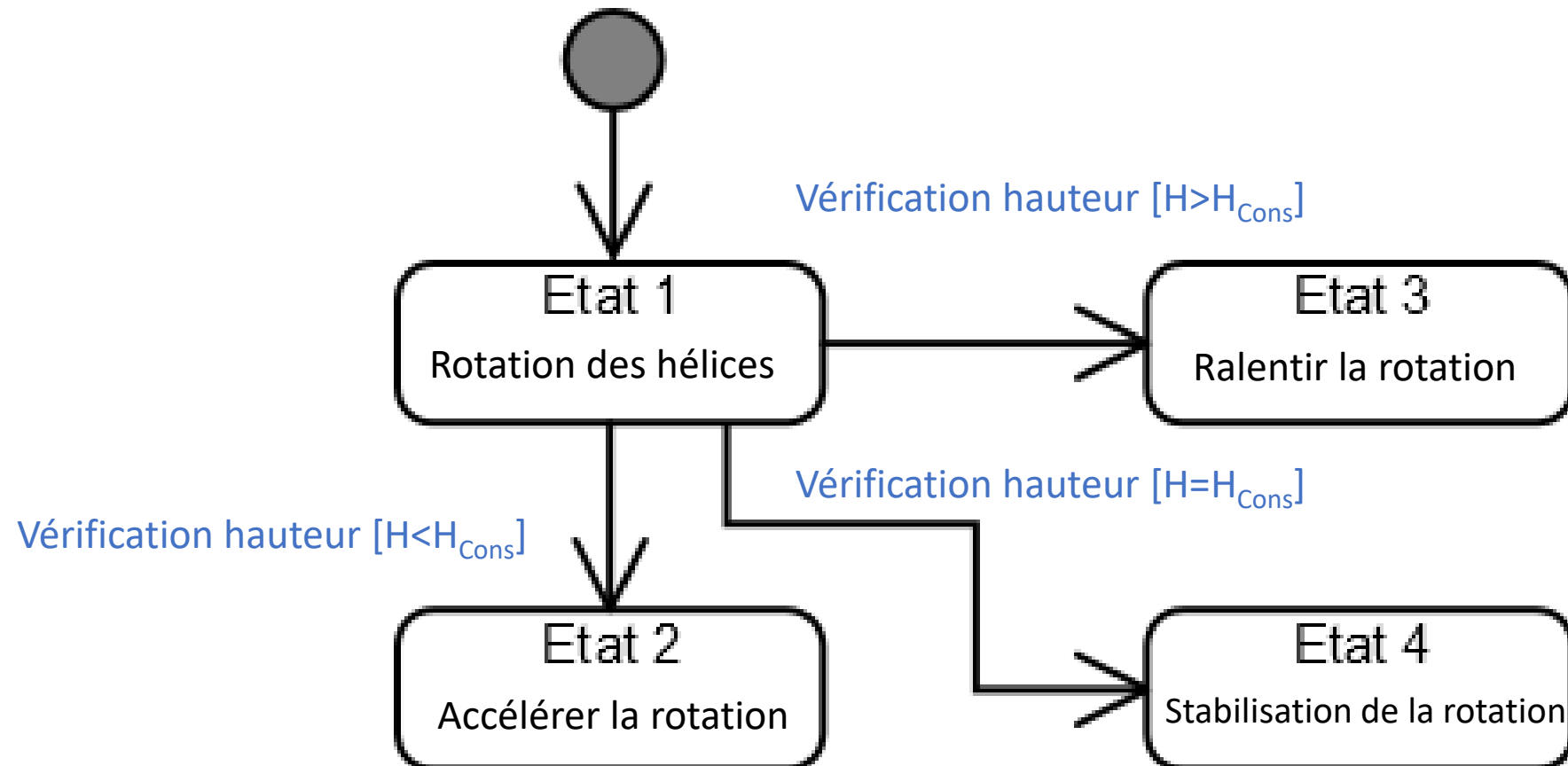
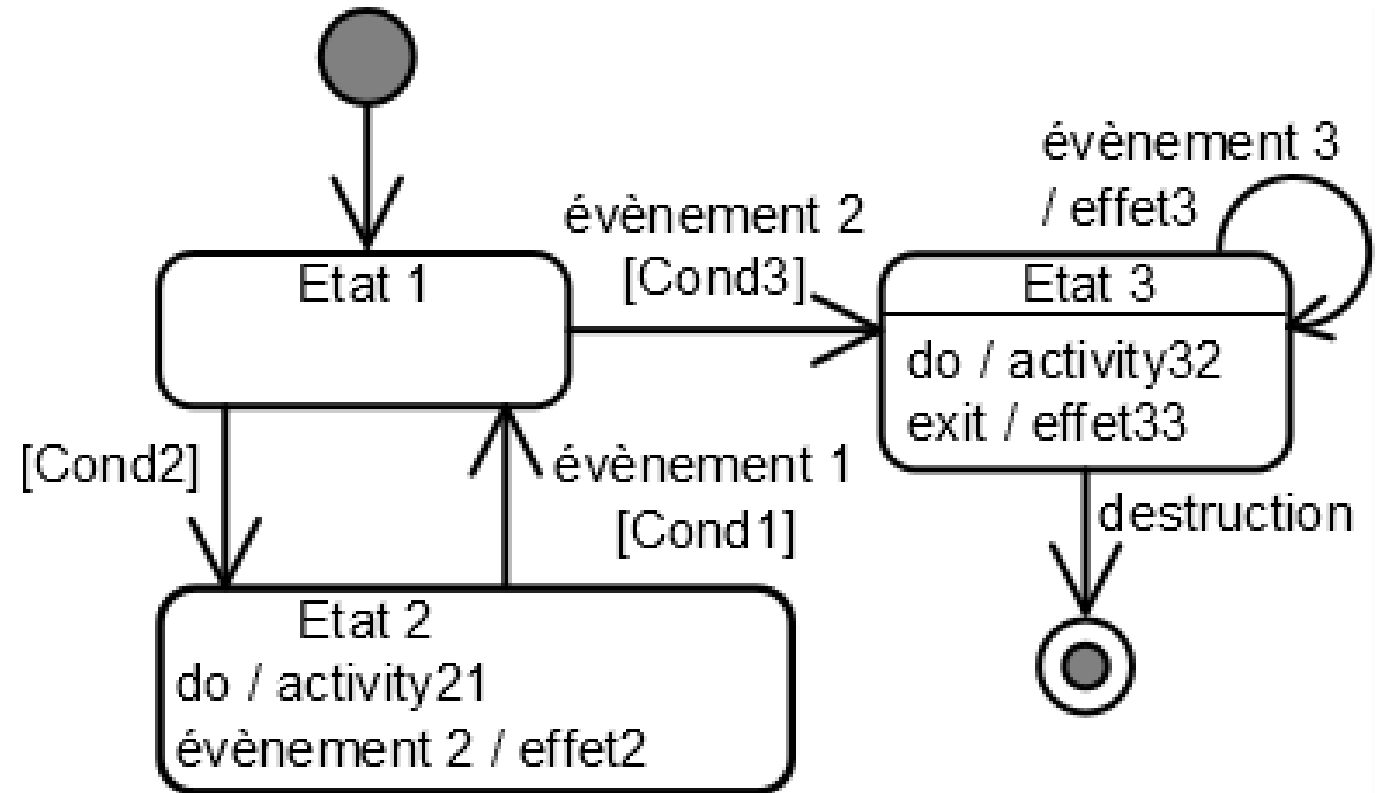


Diagramme d'états (stm)

Diagrammes d'état / transition :



Evènement 3 = Transition réflexive

→ Sortie de l'état 3 + Retour immédiat dans celui-ci.

→ Interruption de l'activité32 → Réalisation effet33 + effet3 → Redémarrage de l'activité32.

Diagramme d'états (stm)

Diagrammes d'état / transition :

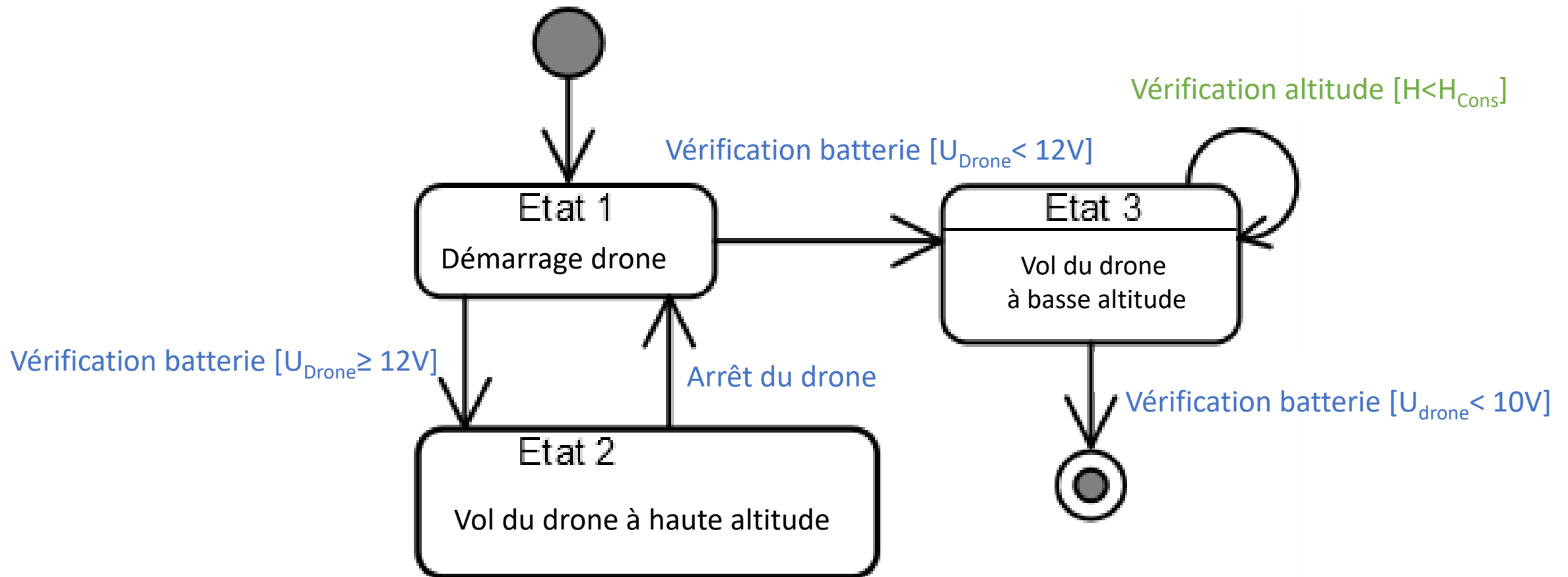
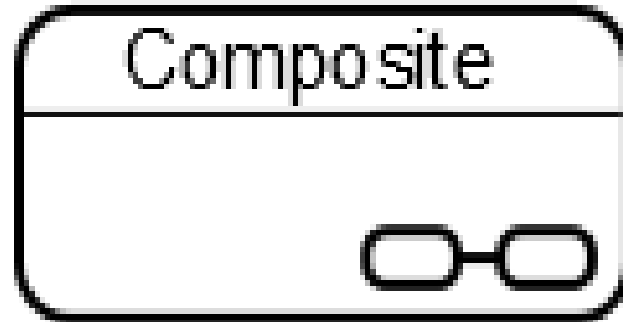


Diagramme d'états (stm)

Etat composite :



Etat composite (lunettes) \approx macro-état

→ Evolutions internes avec un autre diagramme d'état (il contient une machine d'état)

Diagramme d'états (stm)

Etat composite :

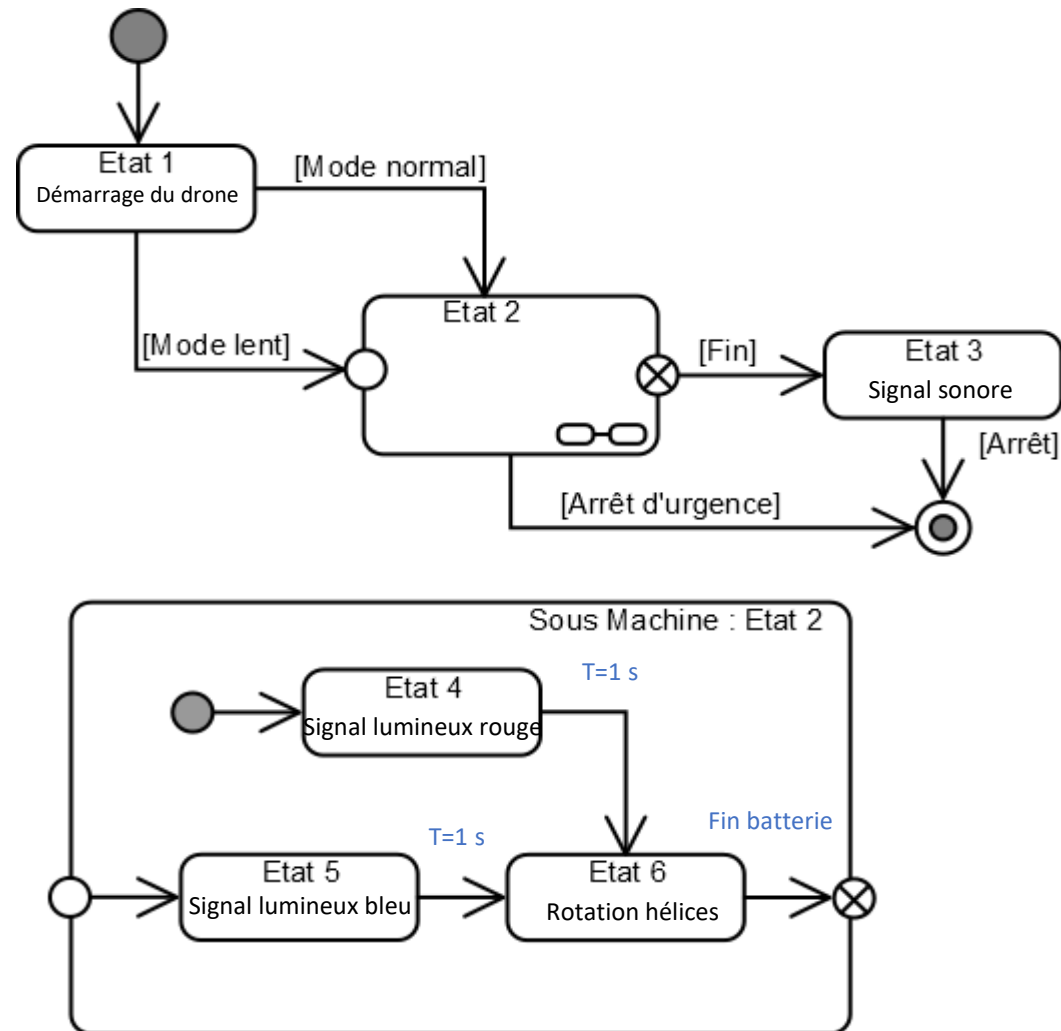


Diagramme d'états (stm)

Etat composite :

Exemple : Radio Réveil

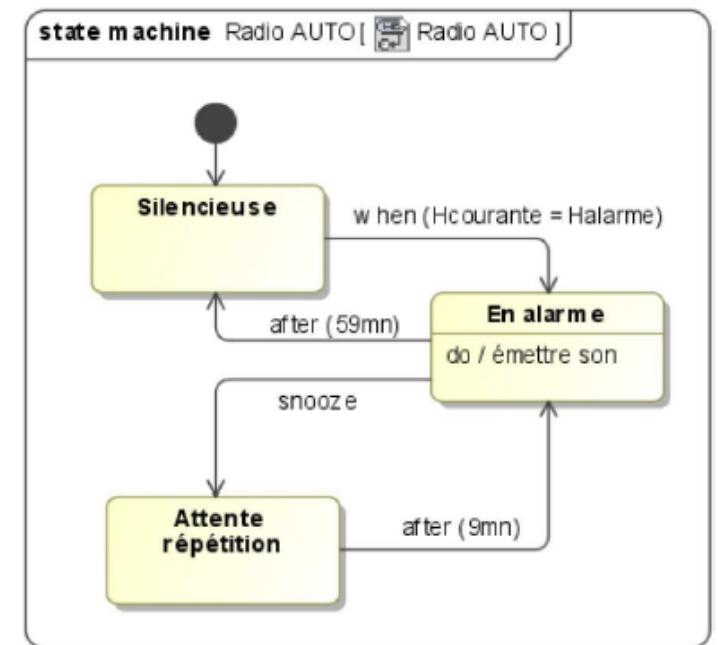
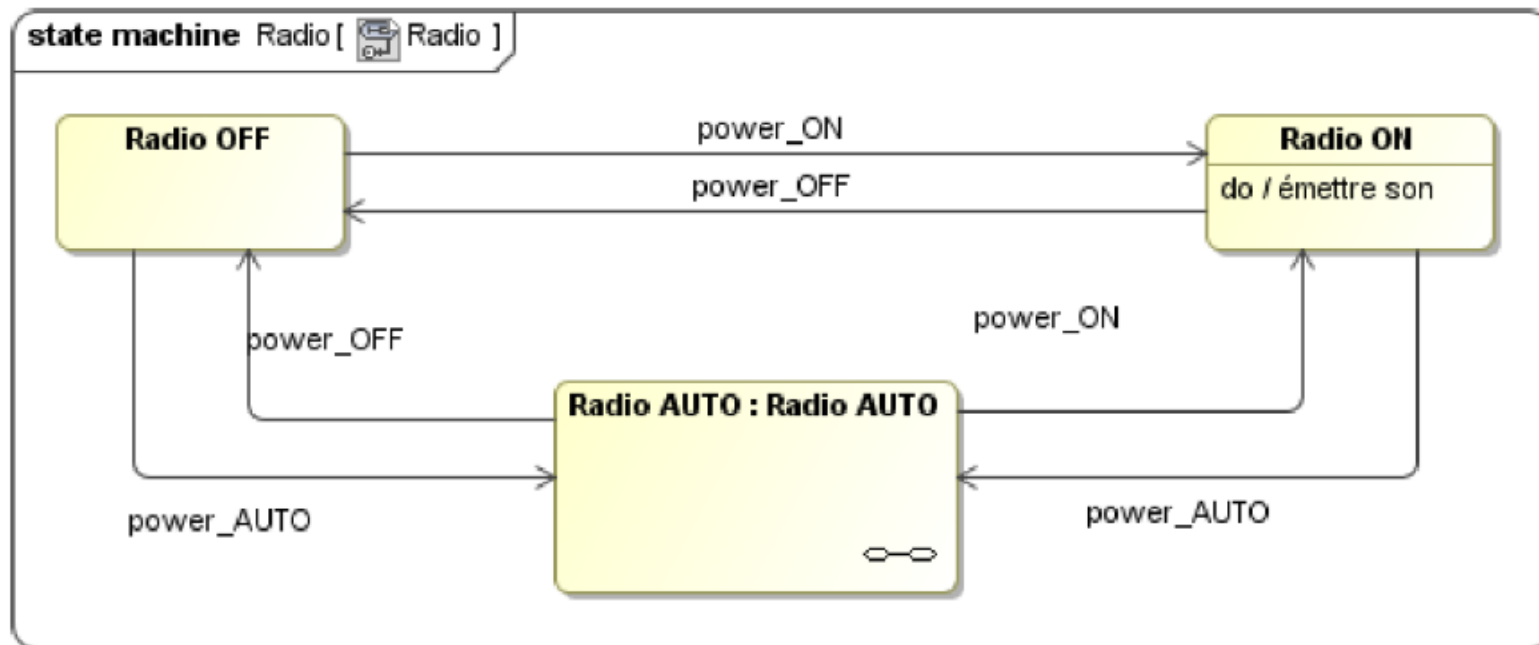


Diagramme d'états (stm)

Divergence et convergence de séquences parallèles :

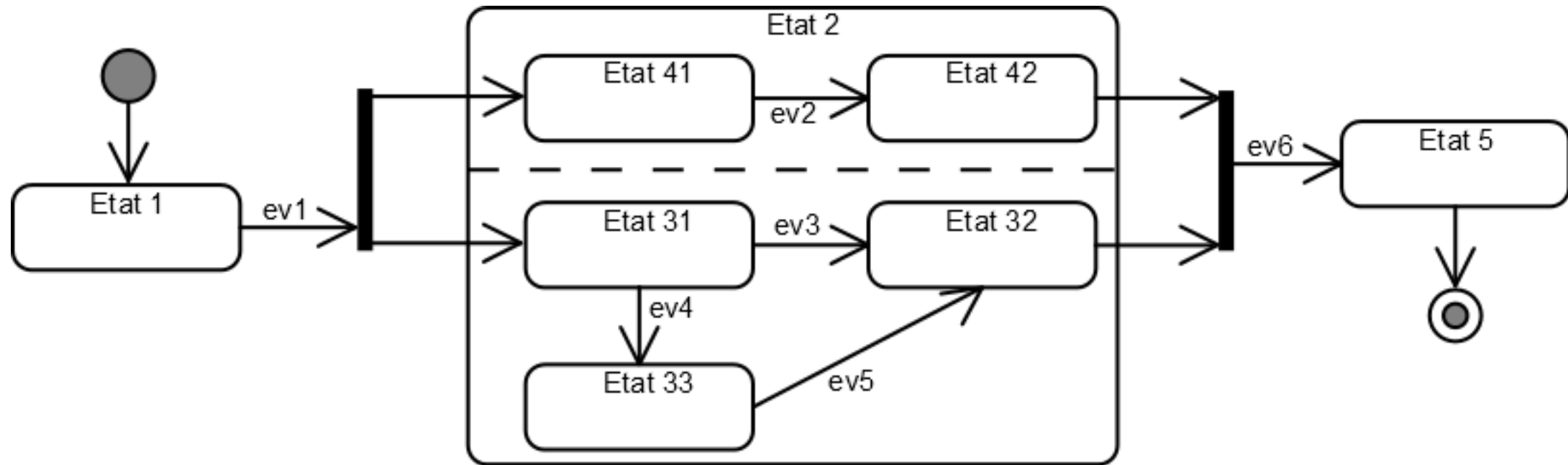
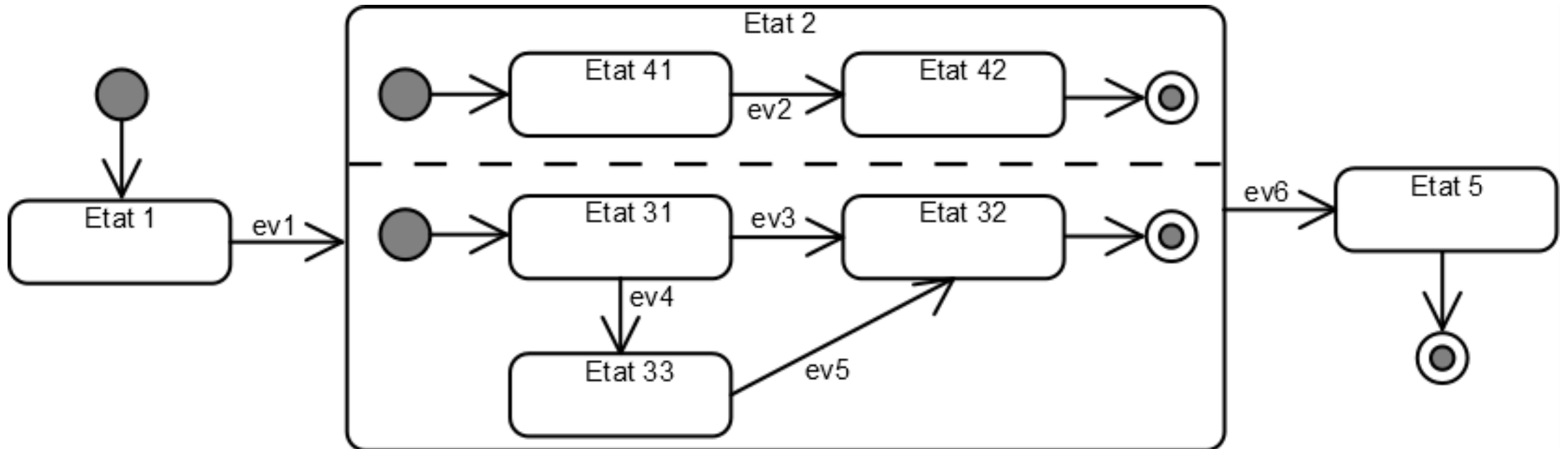


Diagramme d'états (stm)

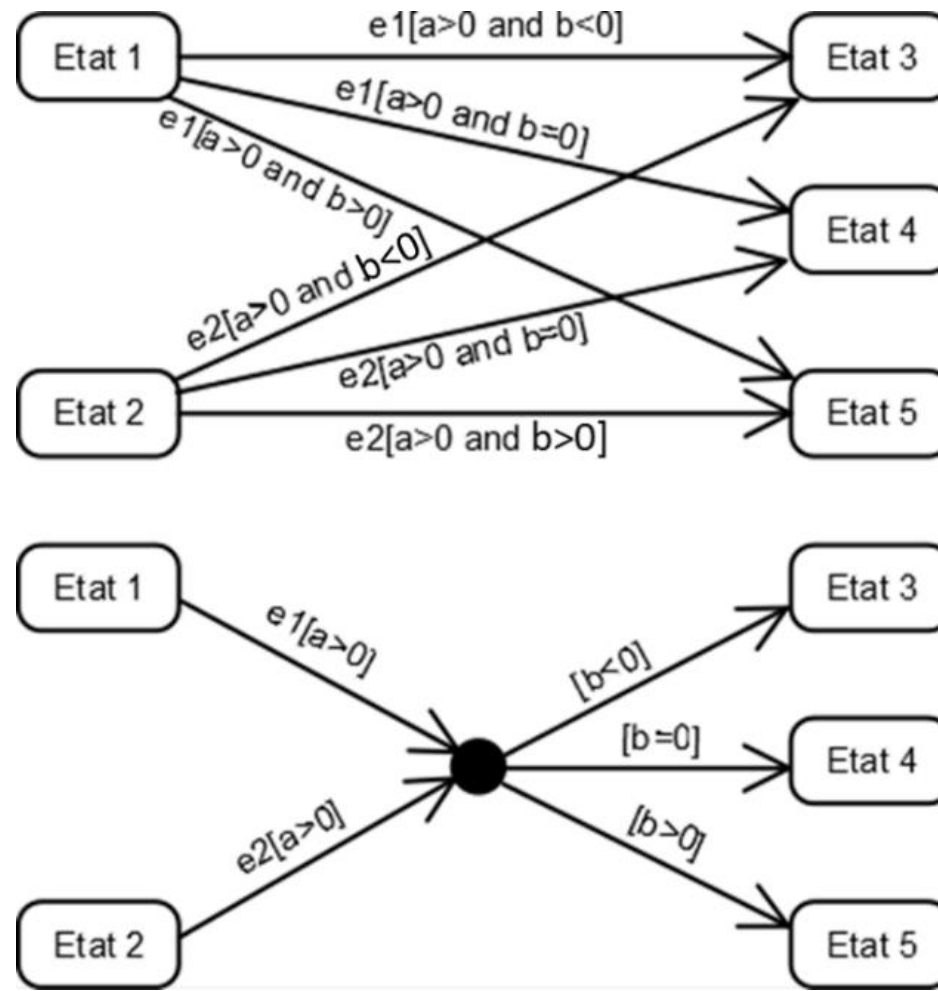
Transitions dans un état composite :



Une transition qui atteint la bordure d'un état composite (Etat2) est équivalente à une transition qui atteint toutes les étapes initiales.

Diagramme d'états (stm)

Points de jonction :



Points de jonctions → pseudo états → Simplification des graphes

Diagramme d'activités (act)

Diagramme → **algorithme** (= un flux de contrôle).

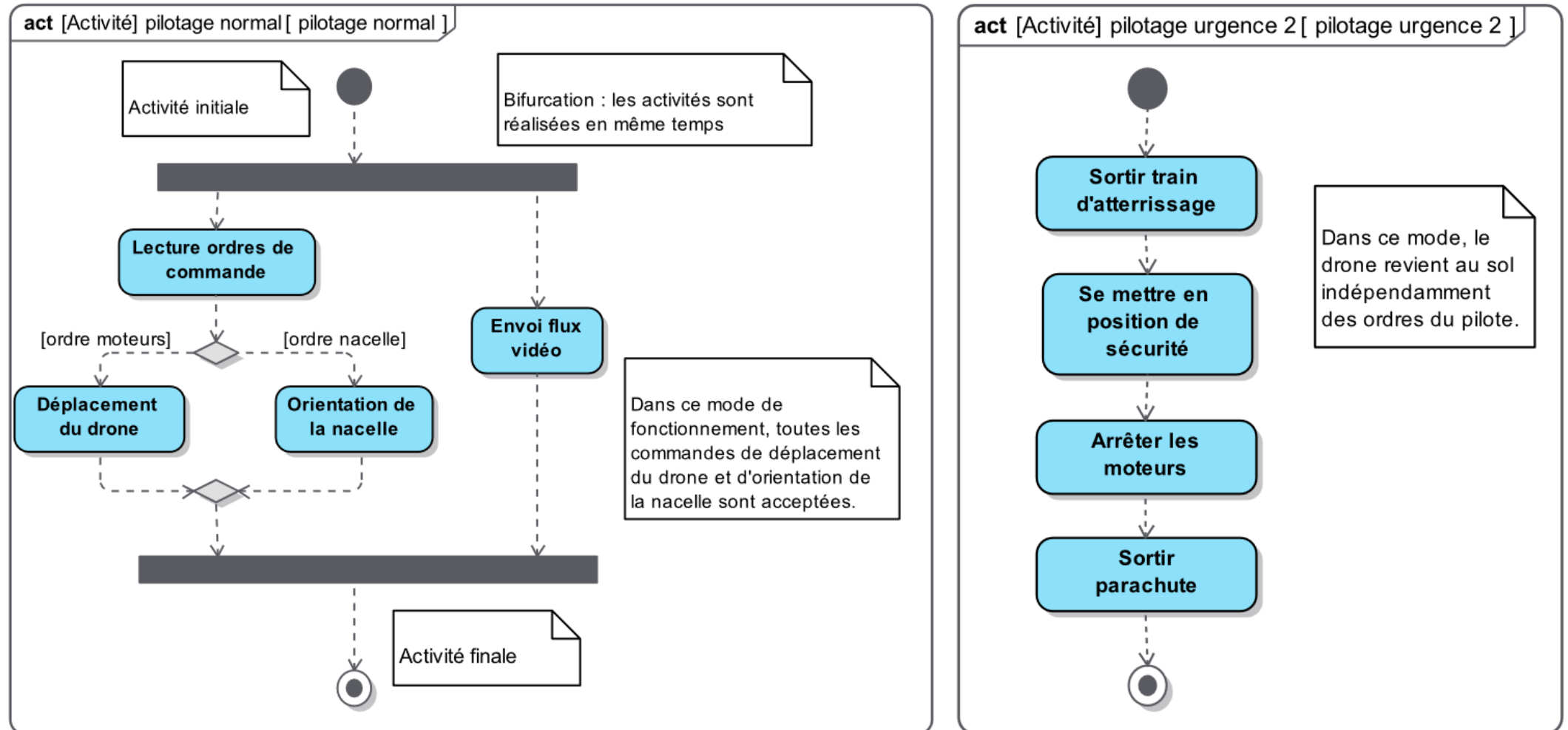


Diagramme d'activités (act)

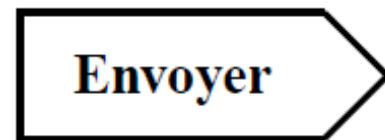
Signaux et événements :

Activité → Recevoir / Emettre des signaux

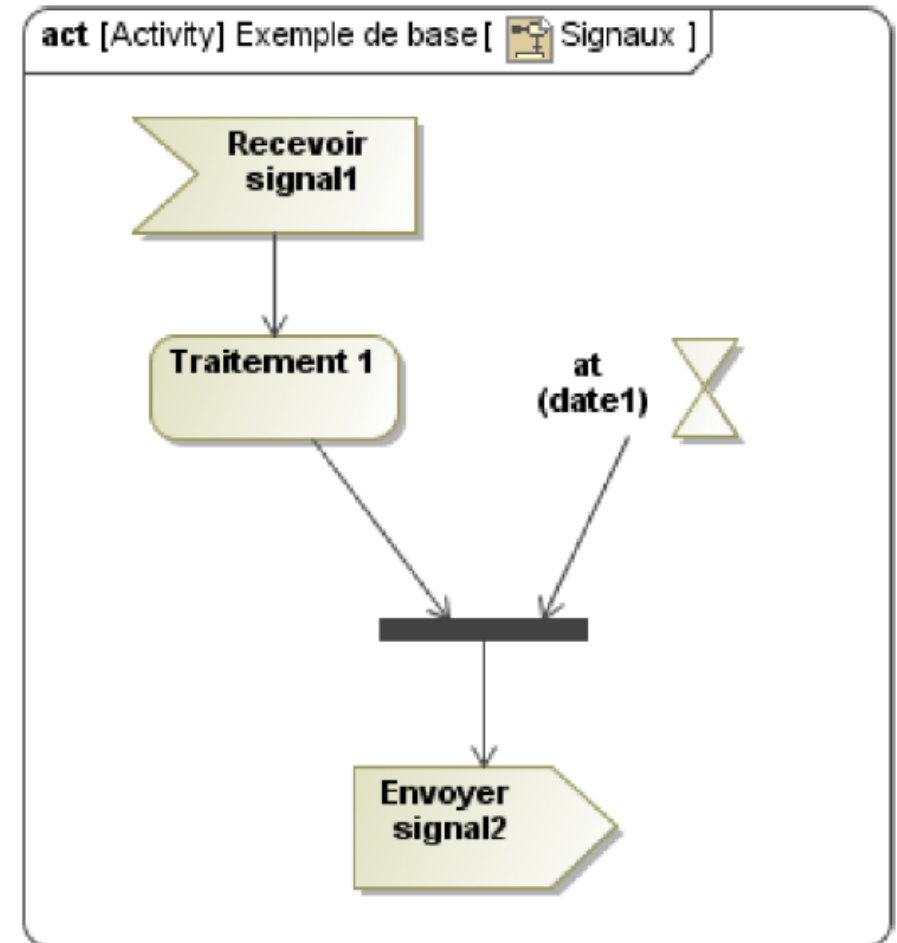
- accept event action :



- send signal action :



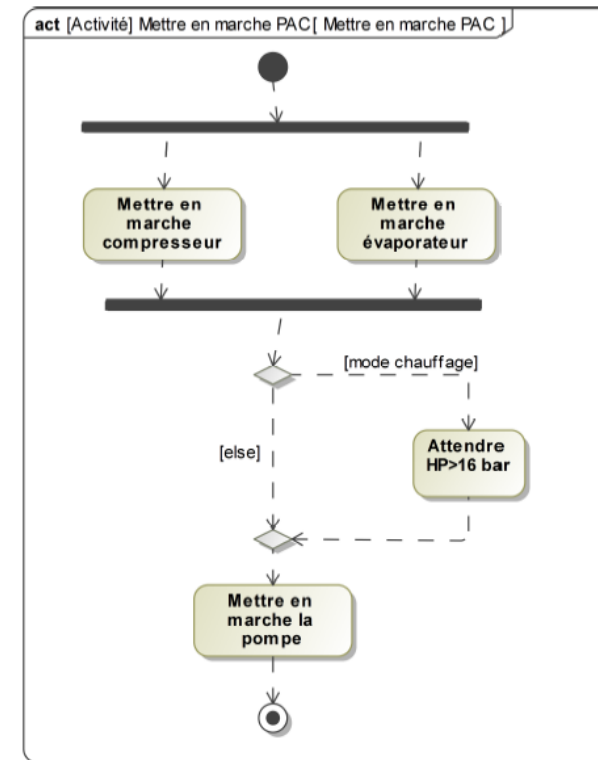
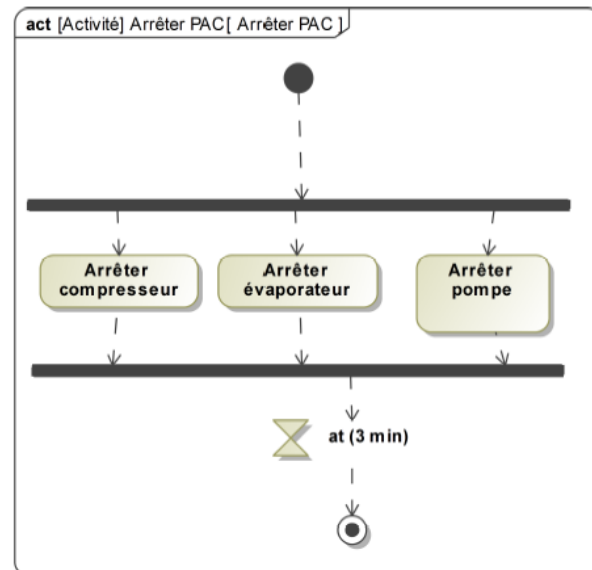
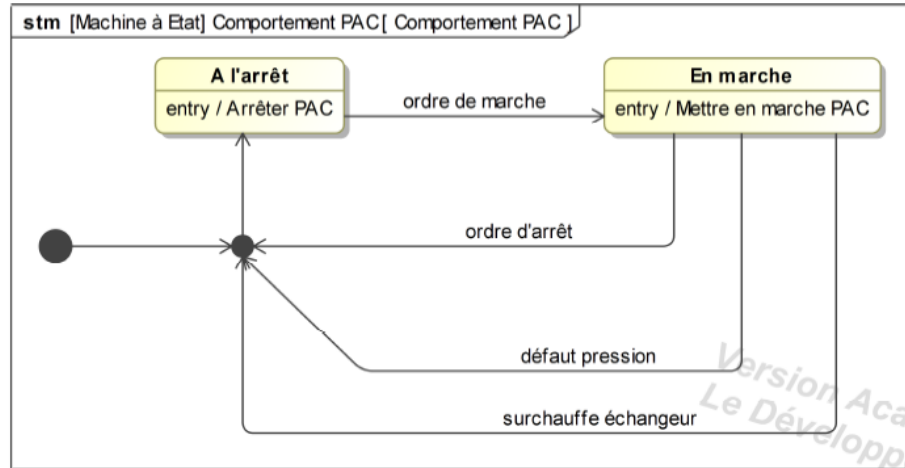
- accept time event :



Différences entre le diagramme d'état et le diagramme d'activités

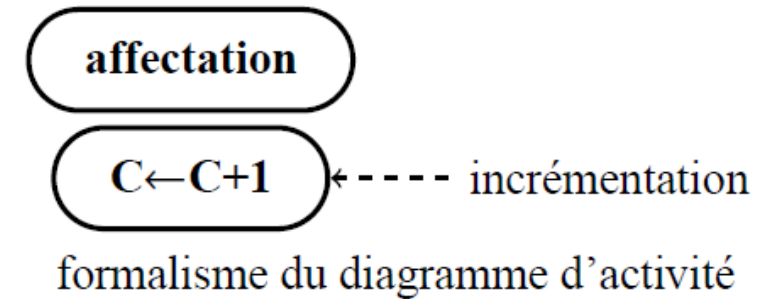
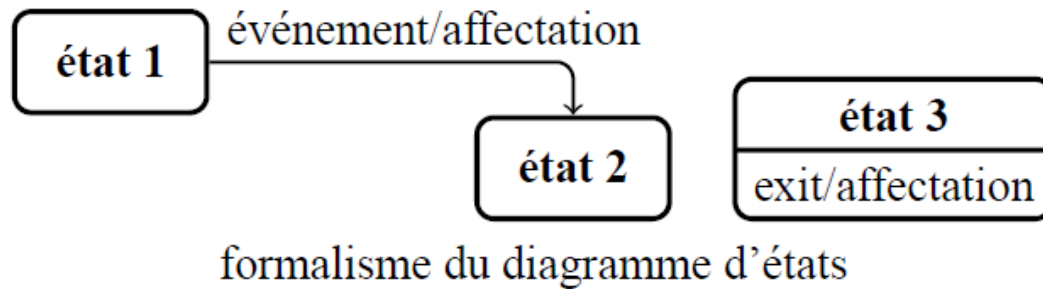
	Diagramme d'états (stm)	Diagramme d'activités (act)
Evènement associé à une transitions ?	✓	✗
Rattachement à :	1 bloc	Plusieurs blocs
Pilotage des évènements ?	✓	✗
Mise en évidence de l'élément associé à la tâche ?	✗	✓

Différences entre le diagramme d'état et le diagramme d'activités



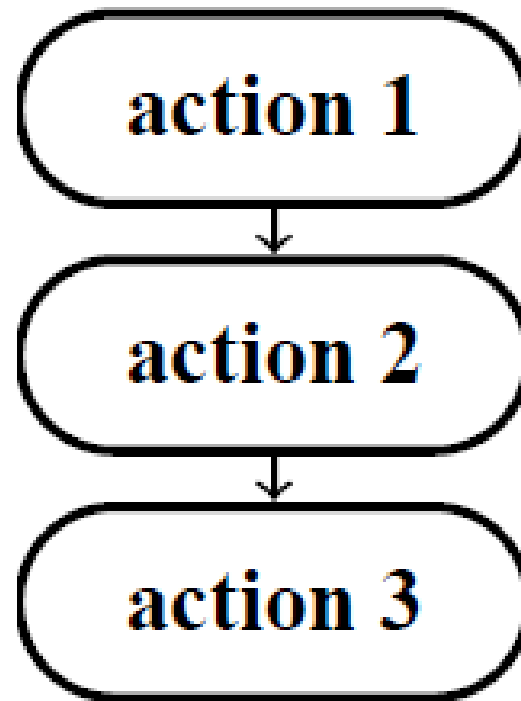
Les structures algorithmiques de base

L'affectation



Les structures algorithmiques de base

Le groupe ou bloc d'instructions



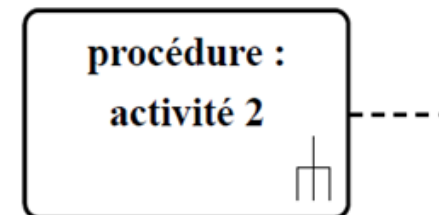
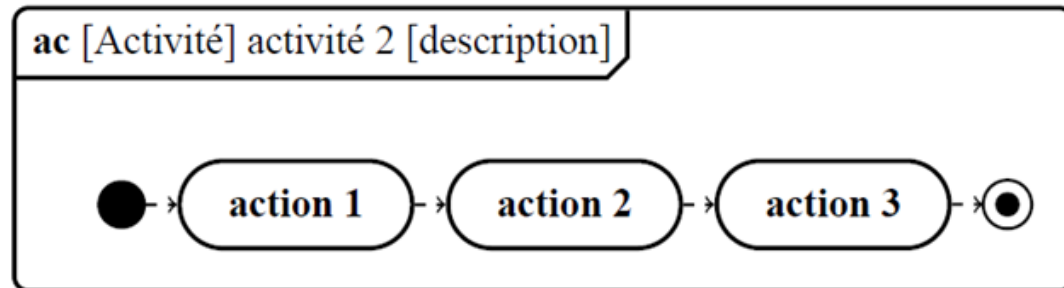
Les structures algorithmiques de base

Fonctions et procédures

Décomposition d'un algorithme → fonctions et procédures :

- Partitionner une problématique générale
- Réutilisation de sous-programmes (tâches élémentaires)

Procédure → succession d'instructions (ne renvoie rien)

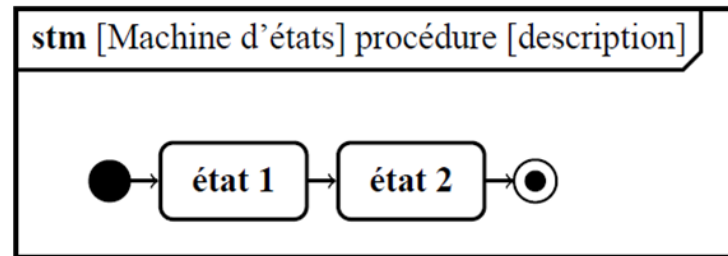


La procédure est définie dans le diagramme de l'activité 2.

Les structures algorithmiques de base

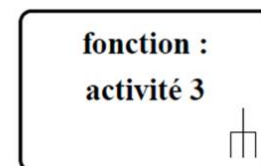
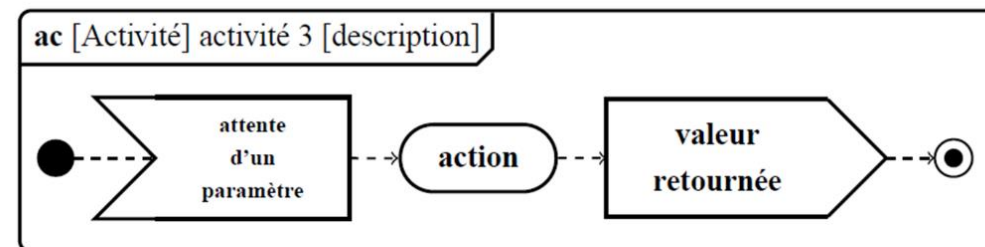
Fonctions et procédures

Etats composites d'un diagramme d'états :



La procédure est définie dans le diagramme d'états **procédure**.

Fonction → retour d'une valeur, d'une liste, d'un objet, ...



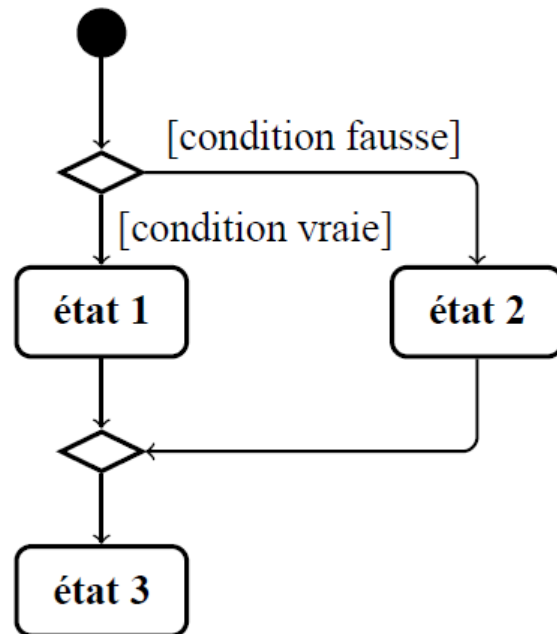
La procédure est définie dans le diagramme de l'activité 3.

Les structures algorithmiques de base

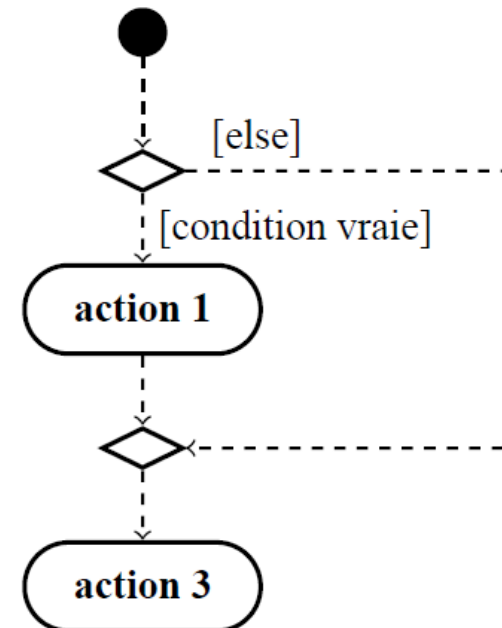
Les structures conditionnelles et bouclées

Structure alternative avec saut

Si , alors faire , sinon faire



Formalisme du diagramme d'états



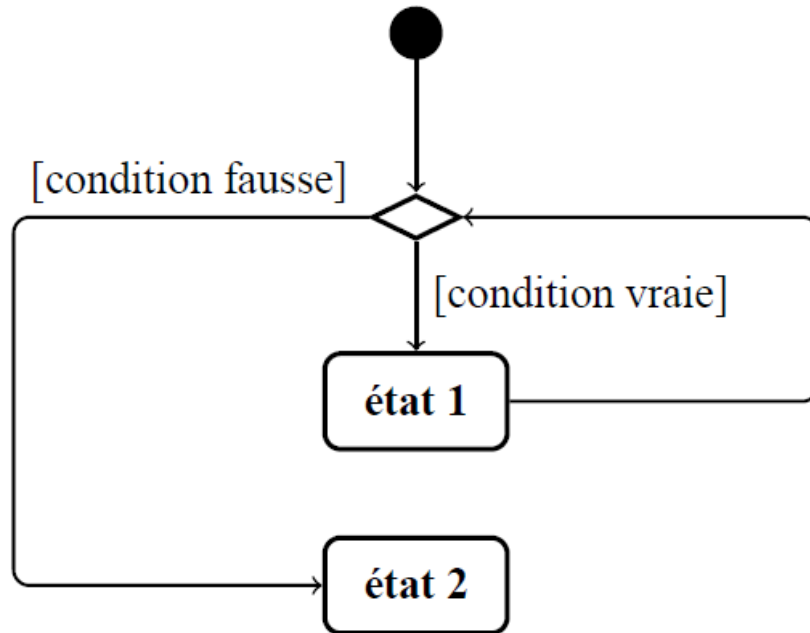
Formalisme du diagramme d'activités

Les structures algorithmiques de base

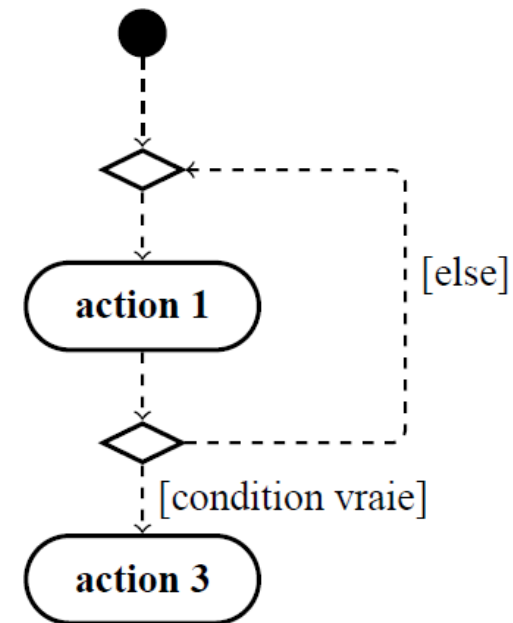
Les structures conditionnelles et bouclées

Structures répétitives (itératives)

Tant que condition vraie, faire ... / Répéter ... jusqu'à condition vraie



Formalisme du diagramme d'états



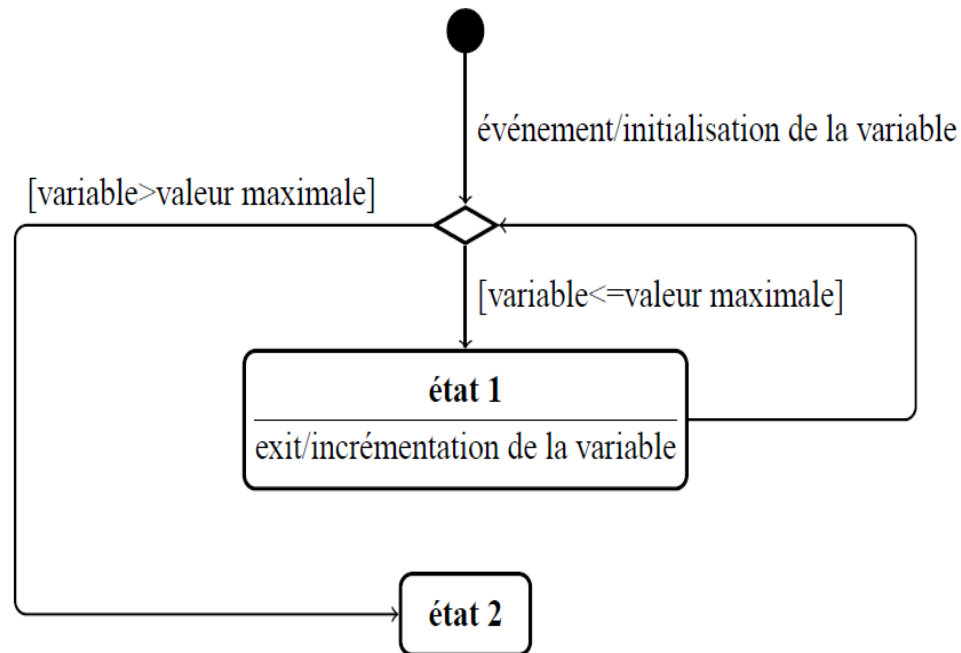
Formalisme du diagramme d'activités

Les structures algorithmiques de base

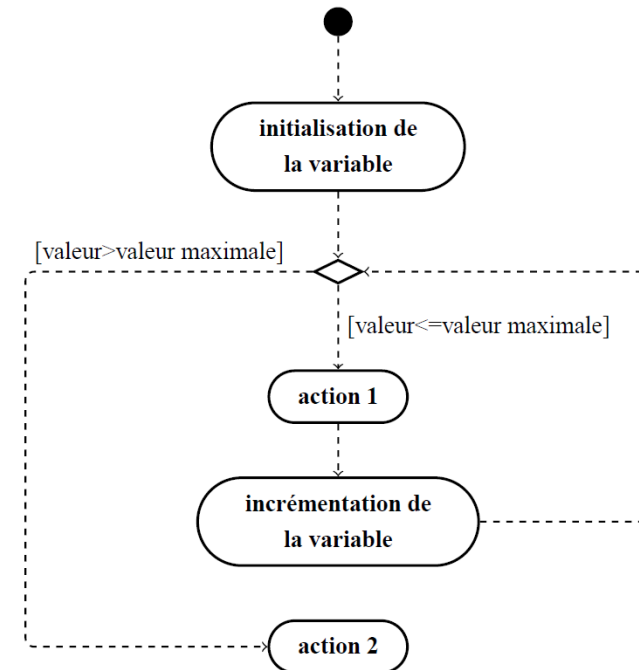
Les structures conditionnelles et bouclées

Structures répétitives (itératives)

Pour variable = valeur initiale, jusqu'à valeur maximale, faire ...



Formalisme du diagramme d'états



Formalisme du diagramme d'activités

Algorithme et pseudo-code

Définition : Un algorithme est une série d'actes ou d'opérations élémentaires qu'il faut exécuter en séquence pour accomplir une tâche quelconque, en suivant un enchaînement strict.

Algorigramme et pseudo-code

Le pseudo-code

Mots et symboles du pseudo-code	Opérations réalisées
Début	Début de l'algorithme, permet de le nommer
Fin	Fin de l'algorithme
Faire	Exécution d'une opération
Entrer	Acquisition ou chargement d'une donnée
Sortir	Edition ou sauvegarde d'un résultat
←	Affectation d'une valeur à une variable
Symboles d'opérateur	Opérations arithmétiques ou logiques
Aller à	Branchement inconditionnel
Si...alors...[sinon]	Branchement inconditionnel
Selon cas...[autrement]	Branchement conditionnel généralisé
Itérer...sortir si...	} Répétition conditionnelle
Tant que...faire...	
Répéter...jusqu'à ce que...	
Pour...de...à...	Répétition contrôlée

Algorigramme et pseudo-code

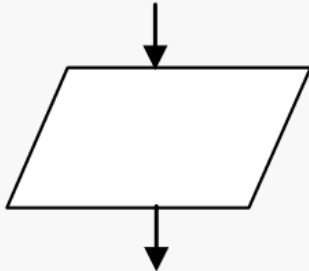
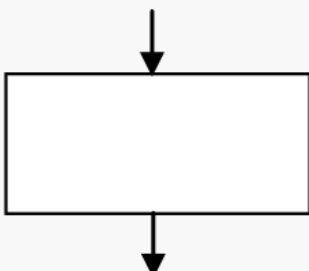
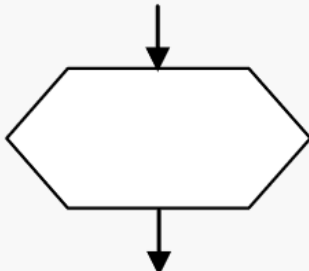
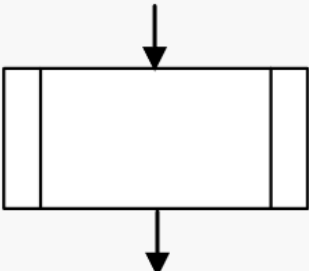
Algorigramme

Définition : L'algorigramme est la représentation graphique de l'algorithme, il permet de représenter chaque opération élémentaire au moyen d'un symbole graphique normalisé.

Algorigramme et pseudo-code

Algorigramme

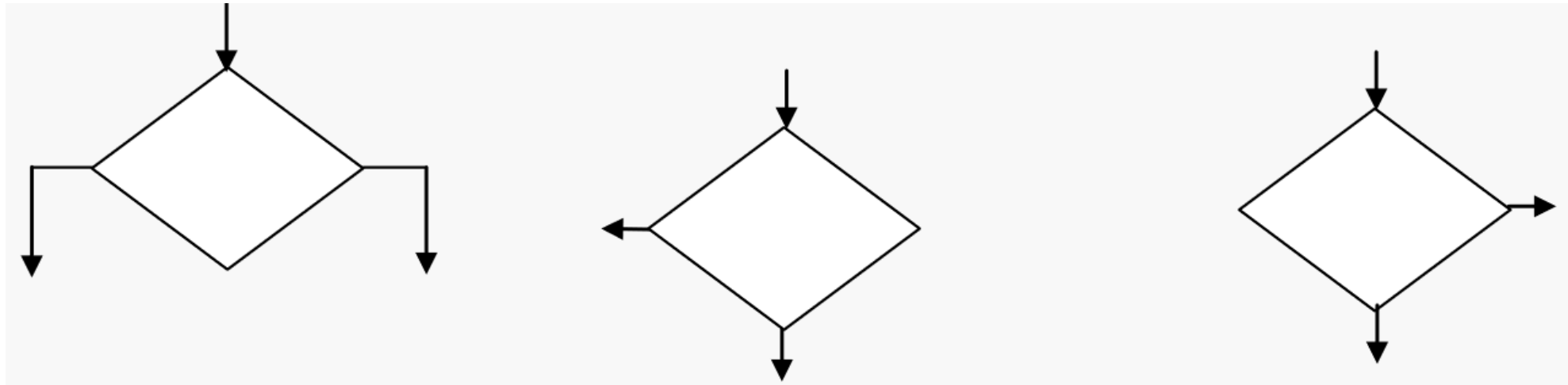
Symboles de traitement

			
Opération de mise à disposition d'une information à traiter ou enregistrement d'une donnée à traiter	Opération ou groupe d'opérations sur des données, instructions ou opération pour laquelle il n'existe aucun symbole normalisé.	Opération de préparation ou d'organisation	Groupe d'opérations considéré comme une seule opération sous programme.

Algorithme et pseudo-code

Algorithme

Symboles de test logique



Algorigramme et pseudo-code

Règles de construction

- Centrer l'algorigramme sur une feuille.
- Construire l'algorigramme afin que sa lecture s'effectue verticalement.
- Les lignes de liaison entre symboles ne doivent pas se couper (utiliser un symbole de renvoi O).
- Une ligne de liaison doit toujours arriver sur le haut et au centre d'un symbole.
- Les commentaires sont à placer de préférence à droite et les renvois de branchement à gauche.

Algorithme et pseudo-code

Structures classiques

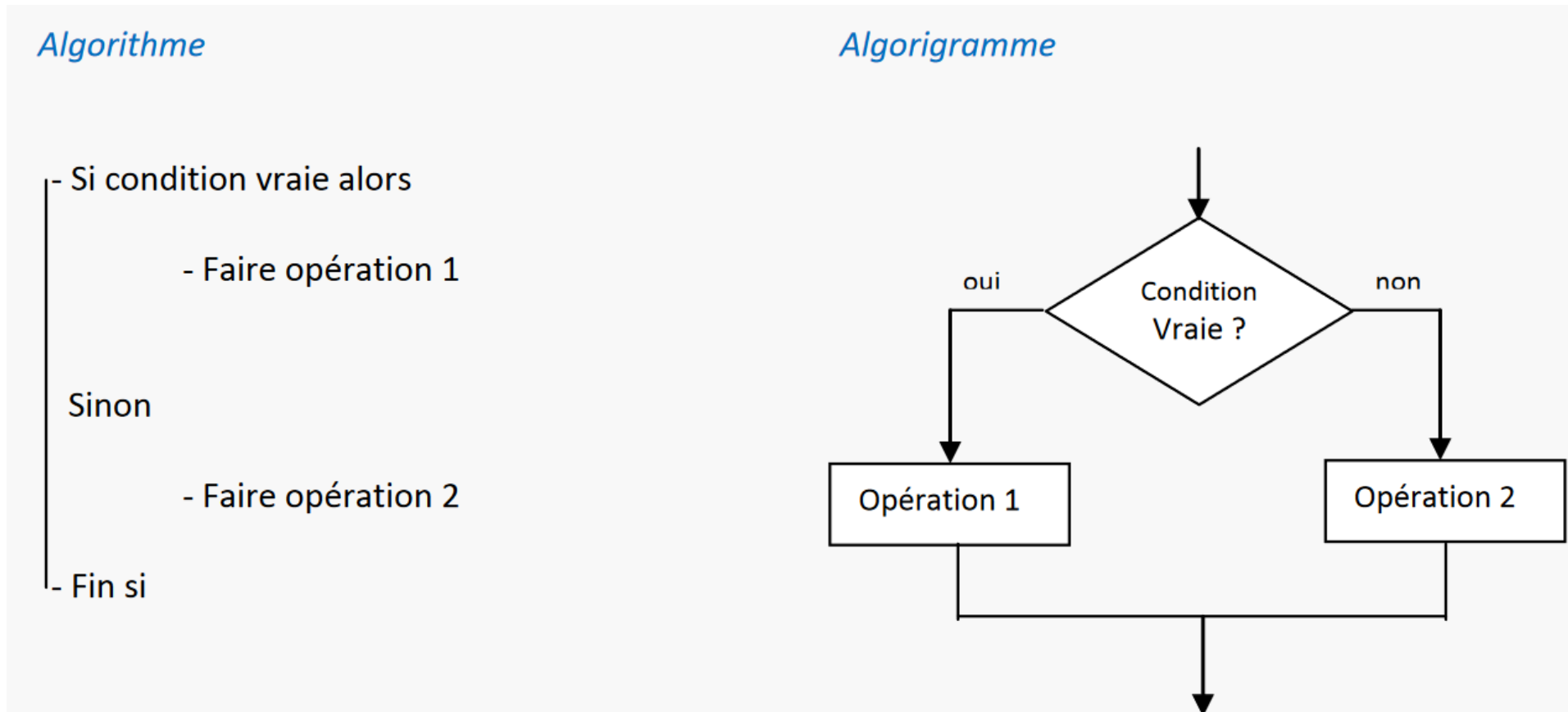
Structure séquentielle ou linéaire



Algorithme et pseudo-code

Structures classiques

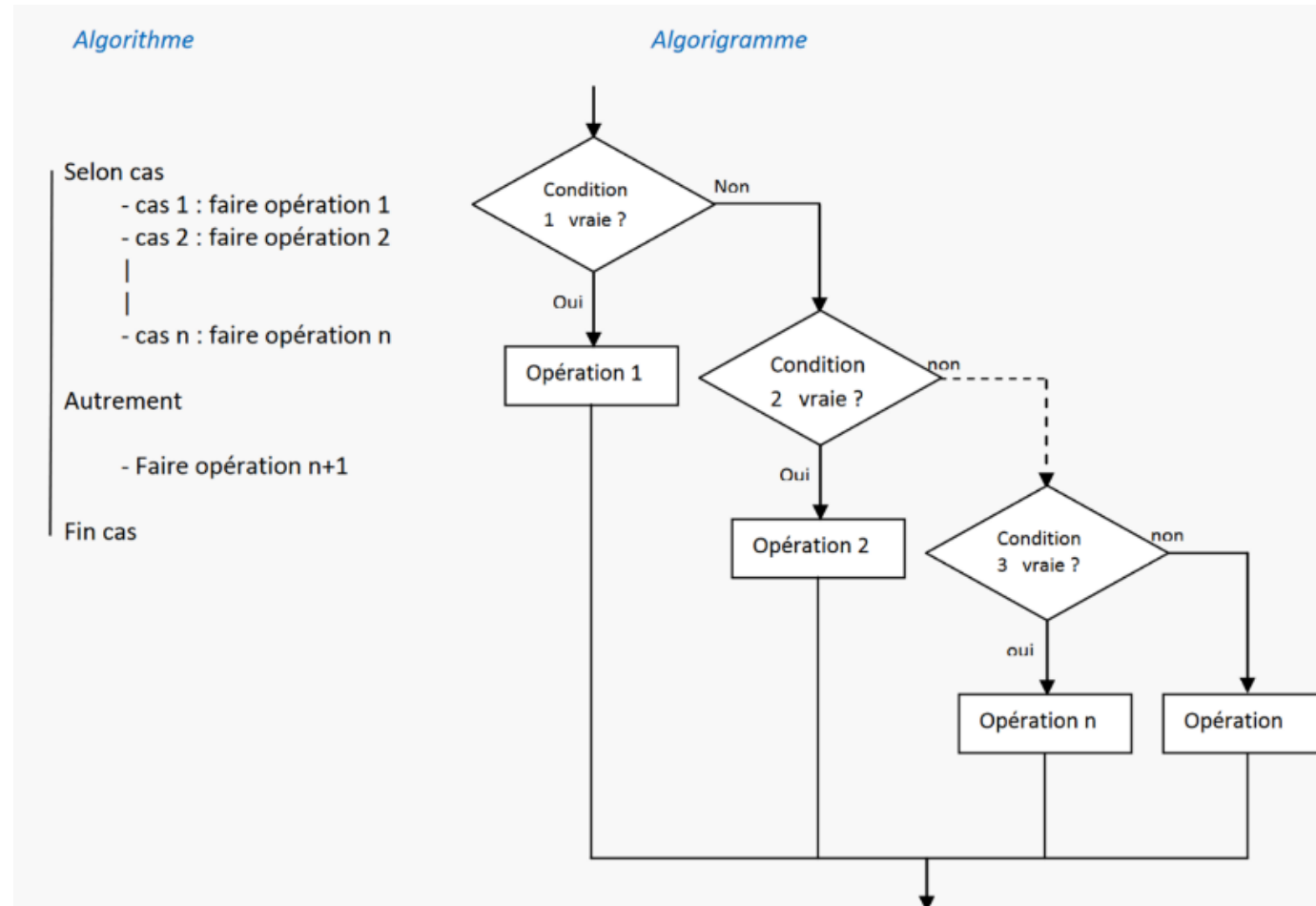
Structure conditionnelle (sélection simple)



Algorithme et pseudo-code

Structures classiques

Structure conditionnelle (sélection multiple)

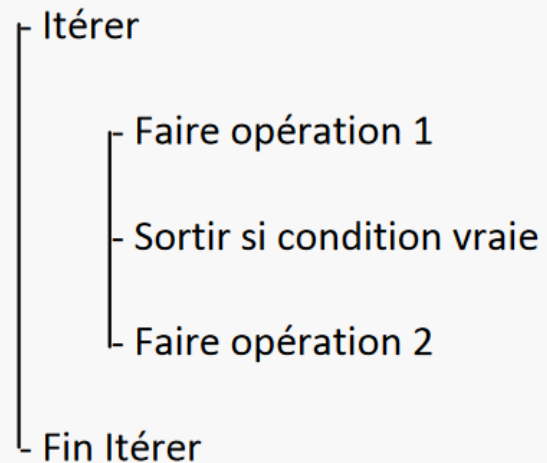


Algorithme et pseudo-code

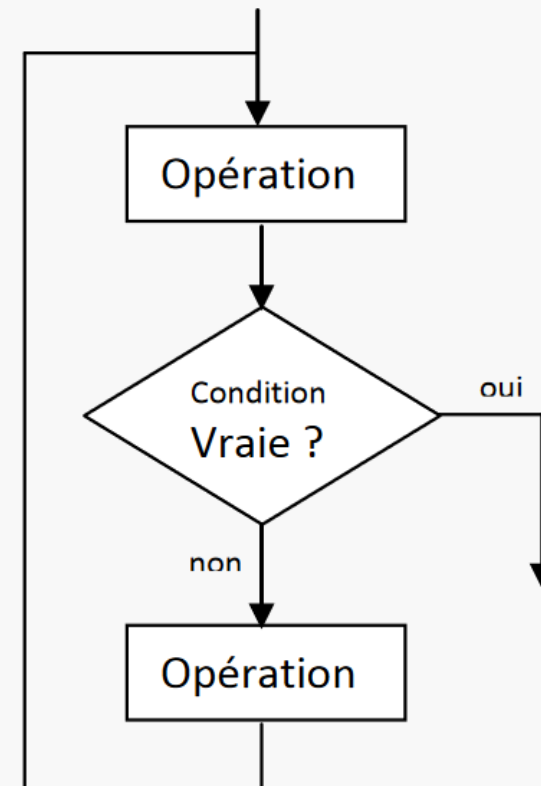
Structures classiques

La structure itérative ou de répétition

Algorithme



Algorithme



Algorithme et pseudo-code

Structures classiques

Cas d'une répétition contrôlée

Algorithme

Pour I de I1 à I2
Faire opération
Fin pour

Algorithme

