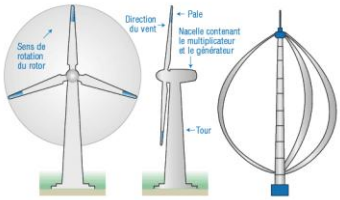
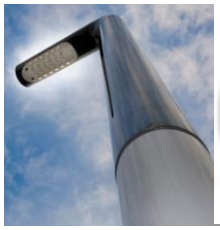




Eléments de modèlisation des systèmes

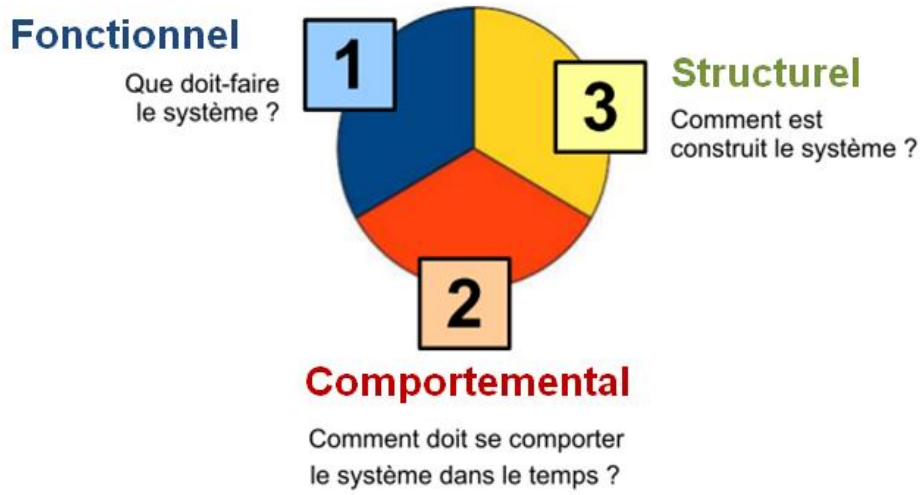
à l'usage des STI2D

1 Décrire un système

Répondre à quelques questions permet de commencer à définir le système, c'est-à-dire :

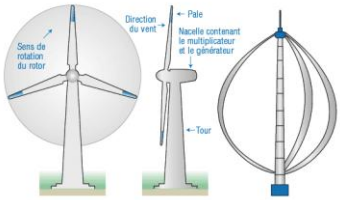
- Que doit-faire le système ?
- Comment le système est-il construit ?
- Comment se comporte le système ?

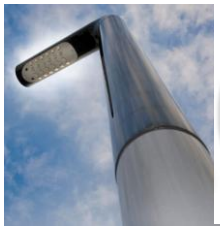
Ces questions permettent de décrire le système selon trois directions :



Pour nous entrainer nous allons décrire un objet du quotidien bien connu la cafetière Nespresso Essenza automatique.

Mais pour commencer visualisons une vidéo :





2 Différents diagrammes disponibles

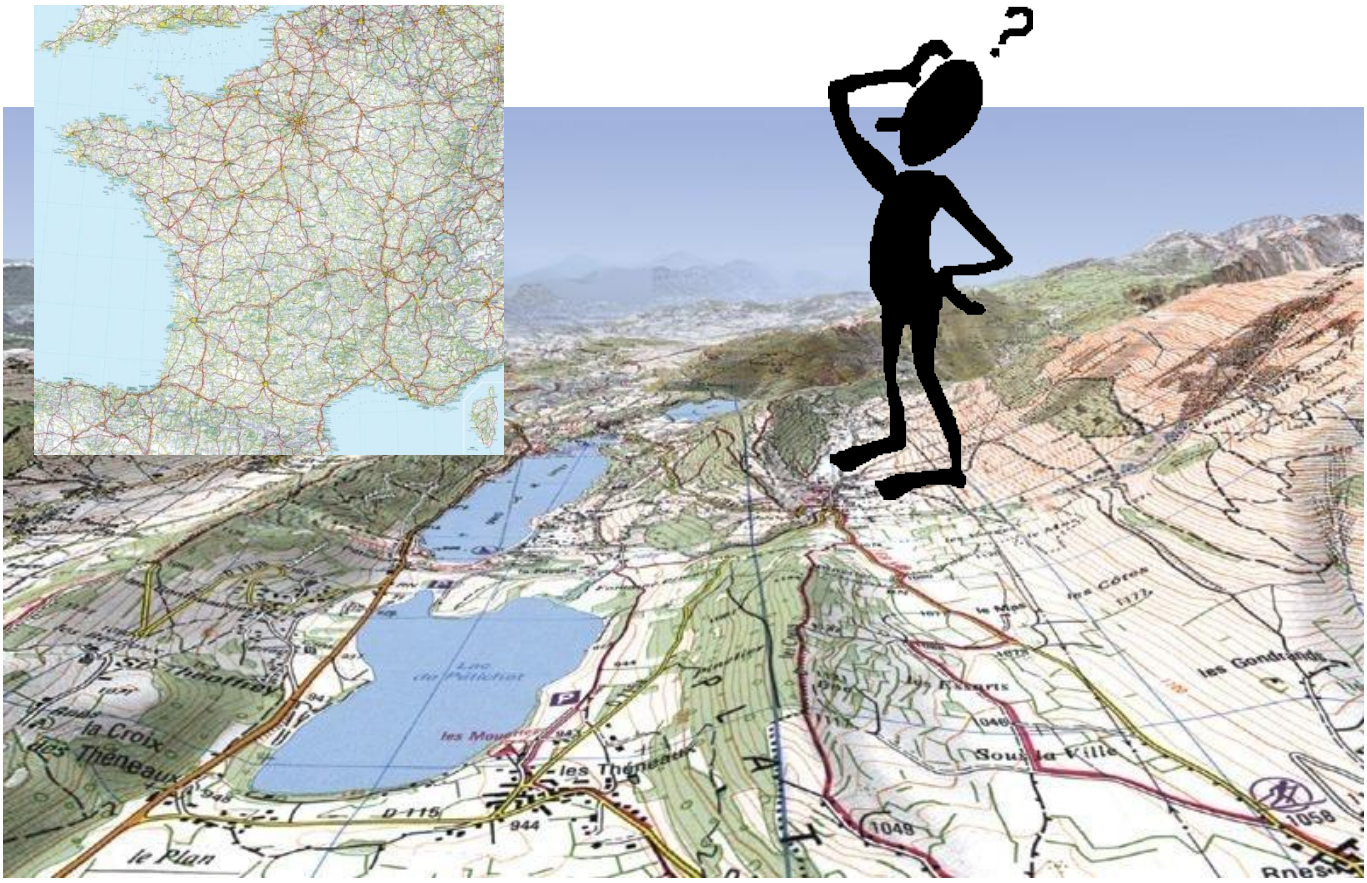
A chaque diagramme son utilité attention il faut privilégier la simplicité avant tout et ne jamais oublier le point de vue avec lequel on veut construire le modèle :

ATTENTION Qu'est-ce qu'un bon modèle ?

A est un bon modèle de B si A permet de répondre de façon satisfaisante à des questions prédéfinies sur B (d'après D.T. Ross). Un bon modèle doit donc être construit :

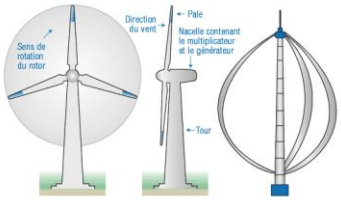
- au bon niveau de détail ;
- selon le bon point de vue.

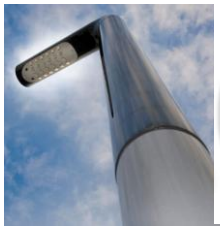
Pensez à l'analogie de la carte routière. Pour circuler dans Toulouse, la carte de France serait de peu d'utilité. Par contre, pour aller de Toulouse à Paris, la carte de la Haute-Garonne ne suffit pas... À chaque voyage correspond la « bonne » carte !



Voilà page suivante les diagrammes disponibles, nous aprendrons à les utiliser au fur et à mesure des descriptions de systèmes vus en exercices, travaux pratiques et simulations, vous les utiliserez de votre coté pour la description de votre projet en spécialité.

Nous retrouvons bien sûr les trois points de vus complémentaires sur le système :





Fonctionnel
- diagramme des exigences
- diagramme des cas d'utilisation

Comportemental
- diagramme d'activités
- diagramme de séquences
- diagramme d'états (états-transition)

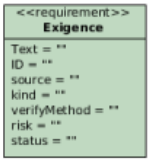
Structurel
- diagramme de blocs
- diagramme de blocs internes
- diagramme paramétrique
- diagramme de paquetage

Ces diagrammes SysML sont succintement présentés dans le tableau ci-dessous, certains ne seront pas utilisés en STI2D (cases rayées).

req : Requirement

exigences

Représenter les contraintes techniques ou non du système.



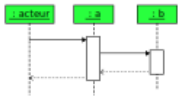
uc : Use Case

cas d'utilisation

Représenter les fonctionnalités attendues du système dans leur contexte.

sd : Sequence Diagram

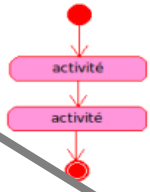
séquence



Décrire les échanges au sein d'un cas ou plusieurs cas d'utilisation.

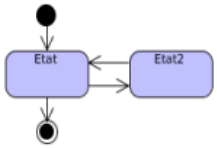
activités

Décrire un enchaînement d'actions lié à un cas d'utilisation.



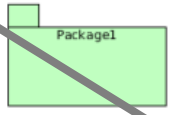
états-transitions

Illustrer les changements d'états d'un système ou sous-système.



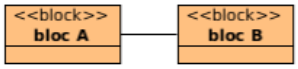
stm : STate Machine

paquetage



Regrouper différents blocs en sous-ensemble (pouvant éventuellement être étudiés séparément)

blocs

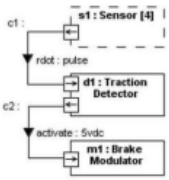


Représenter la structure globale d'un système

bdd : Block Definition Diagram

bloc interne

Illustrer les liens et flux entre les blocs

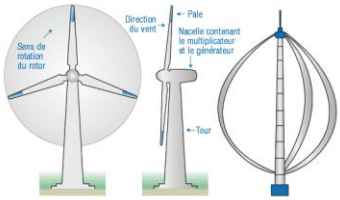


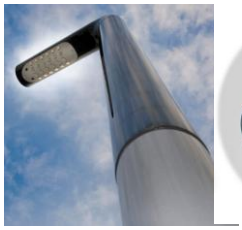
Introduction à UML-sysML
chap 2. outils et mises en oeuvre

ibd : Internal Block

blocs paramétriques

Modéliser le comportement d'un ou plusieurs blocs dans le temps.





Relation entre les blocs dans une description SysML

A B

association : relation d'égal à égal entre deux éléments.
- **A utilise B**
- 2 diagrammes : cas d'utilisation, blocs

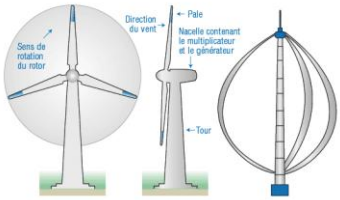
dépendance: deux items indépendants mais dont l'un dépend de l'autre.
- **A dépend de B**
- 4 diagrammes : exigences, cas d'utilisation, blocs, paquetages

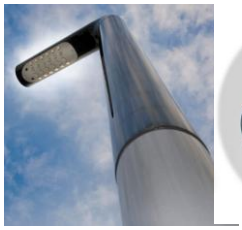
agrégation: un élément est une composante facultative d'un autre.
- **A entre dans la composition de B, sans être indispensable à son fonctionnement**
- 2 diagrammes : exigences, blocs

composition: un élément est une composante obligatoire d'un autre.
- **A entre dans la composition de B et est lui est indispensable**
- 2 diagrammes : exigences, blocs

généralisation: dépendance de type 'filiation' entre 2 items
- **A est une sorte de B**
- 3 diagrammes : cas d'utilisation, blocs, paquetages

conteneur: relation d'inclusion entre deux items
- **B contient A**
- 4 diagrammes : exigences, cas d'utilisation, blocs, paquetages





3 Description des automates séquentiels

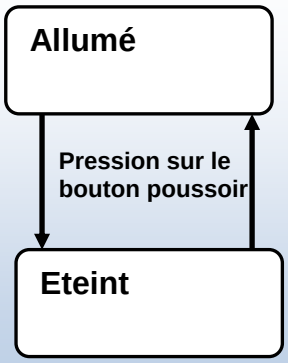
Un système séquentiel peut être décrit avec un automate à nombre d'état finis. Les **diagrammes d'état/transition** sont des graphes permettant de spécifier les évolutions possibles d'un objet (au sens informatique du terme) ou d'un système selon un nombre finis d'états.

Un **état** « mémorise » une situation résultant de l'évolution des variables en entrée et de l'état précédent, l'histoire du système.

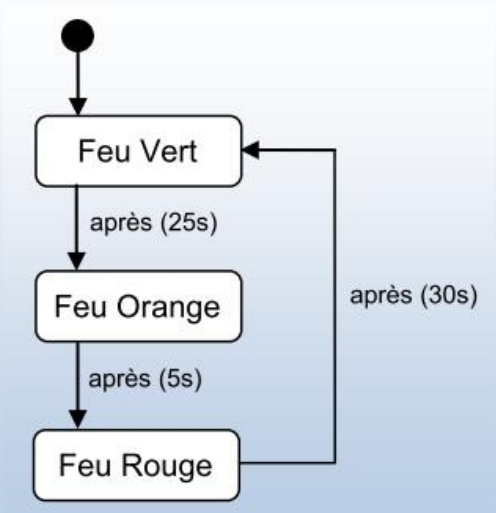
Une **transition** représente le passage instantané d'un état vers un autre.

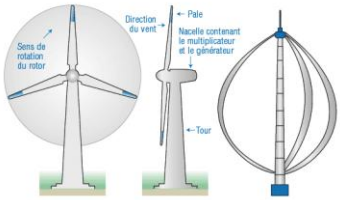
Quelques exemples très simples :

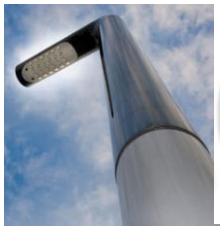
TELERUPTEUR



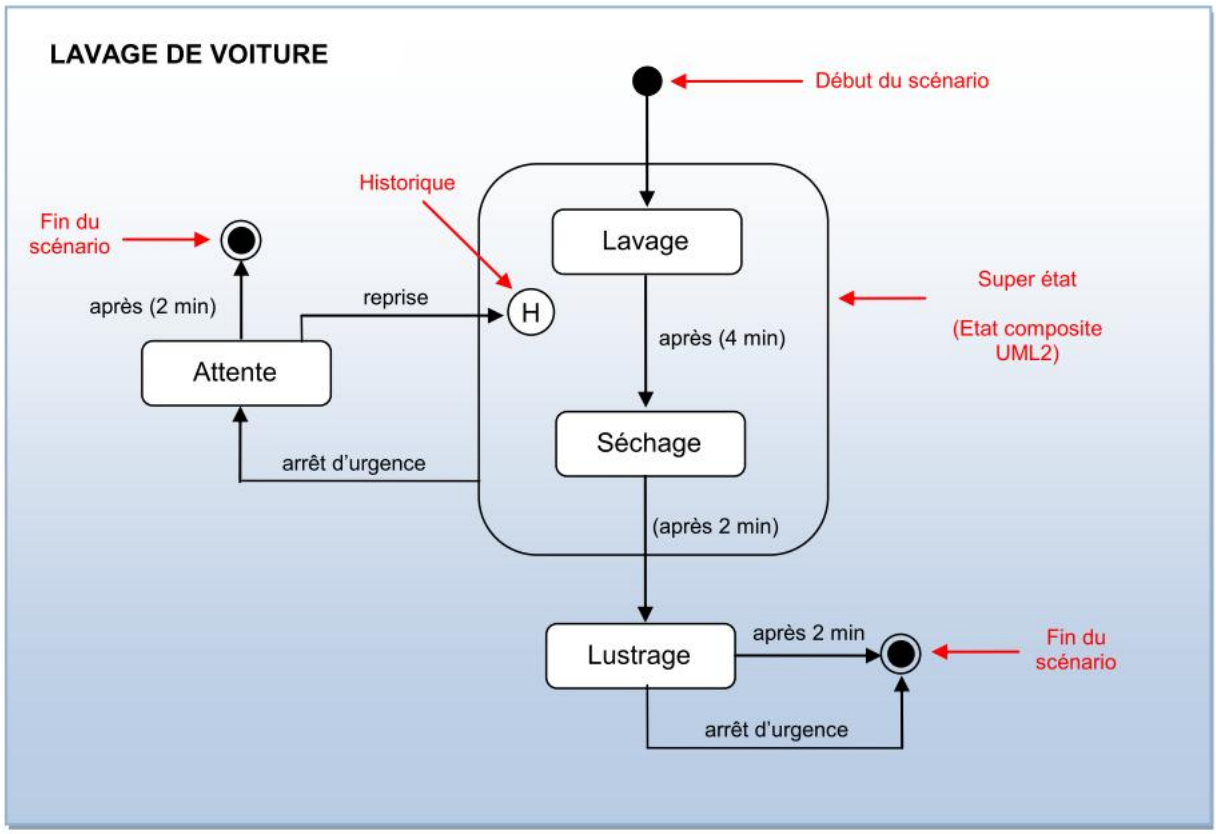
FEU TRICOLORE





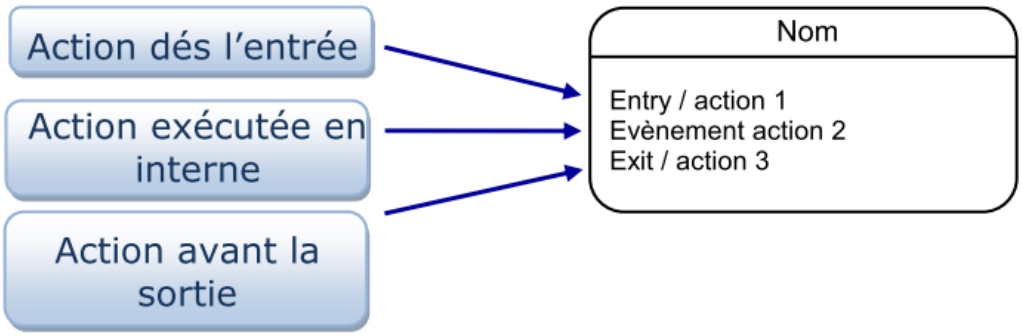


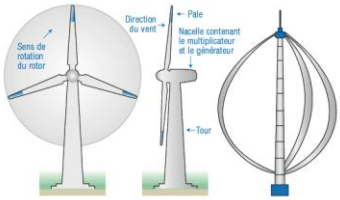
Elements des diagrammes états-transitions de sysML

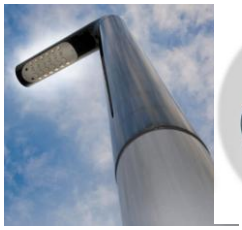


Pour les actions associées à un état :

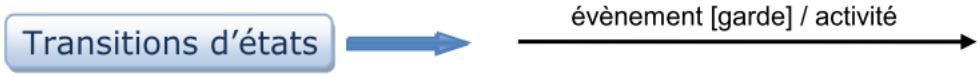
- **entry / action** : action exécutée dès l'entrée dans l'état *(entrée)*
- **exit / action** : action exécutée juste avant la sortie de l'état *(sortie)*
- **on évènement / action** : action exécutée à chaque fois que l'événement cité survient
- **do / action** : action récurrente ou significative, exécutée dans l'état *(faire)* après entry







Propriétés d'une transition :



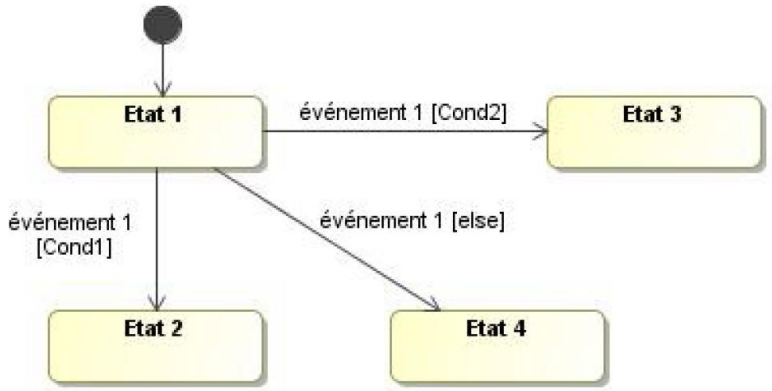
En plus des états de départ (au moins un) et d'arrivée (nombre quelconque), une transition peut comporter les éléments facultatifs suivants :

- Un évènement
- Une condition de garde
- Une liste d'actions

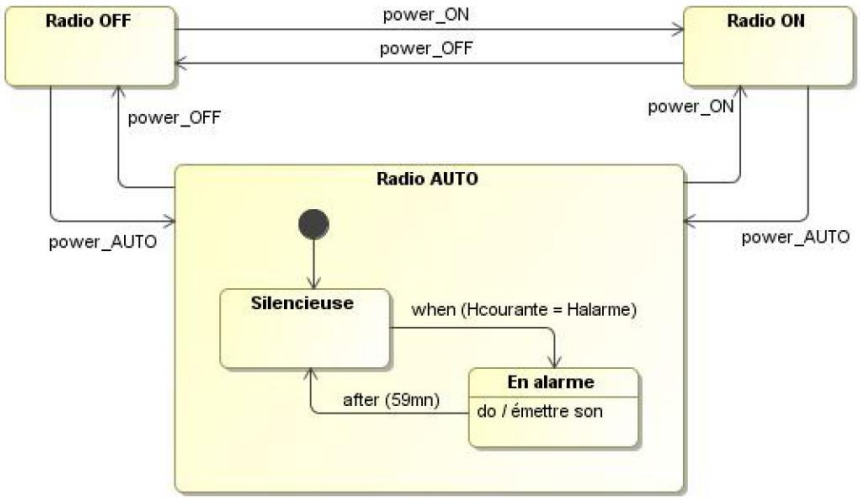
Quand l'événement se produit alors que les états de départ sont actifs et que la condition de garde est vraie alors les actions seront déclenchées.

Une garde (guard) est une condition qui doit être satisfaite pour que la transition ait lieu. Elle s'écrit entre crochets [X]

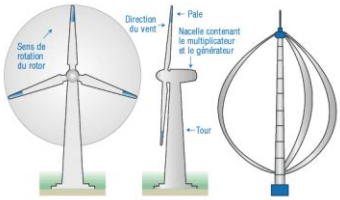
Descriptions à commenter utilisation du else dans les conditions :

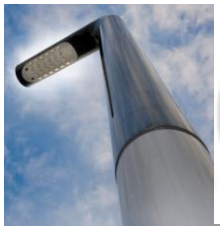


Un radio réveil[1] :



[1] Pascal Roques, SysML par l'exemple, Eyrolles

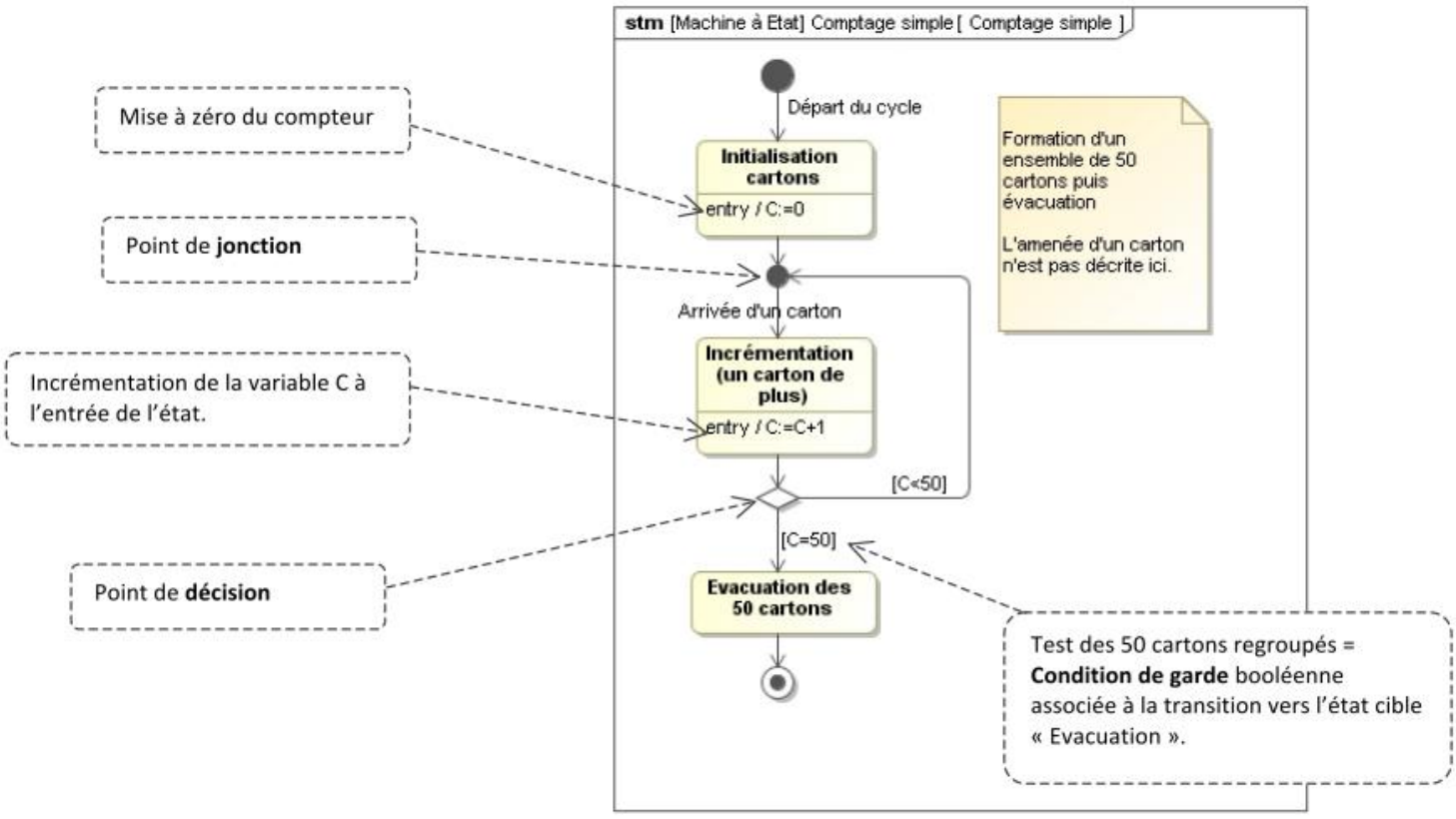




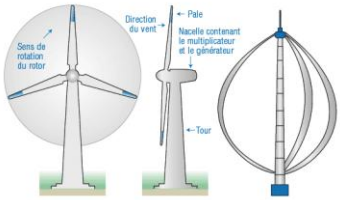
Description de fonctions avancées[2] :

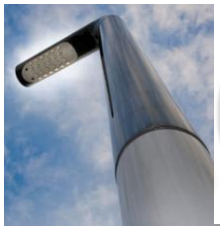
Comptage et reprise de séquence : c'est une illustration de prise de décisions à partir de variables internes à l'automate. Dans l'exemple ci-dessous un décompte numérique accumulé dans la variable C :

Cahier des charges : à la demande du départ cycle par l'opérateur, on doit compter 50 cartons, puis les évacuer. Le système d'approvisionnement de chaque carton n'est pas l'objet de l'étude ici. Seul le système de comptage doit être décrit. Seul l'évènement d'arrivée d'un carton déclenche donc le comptage.



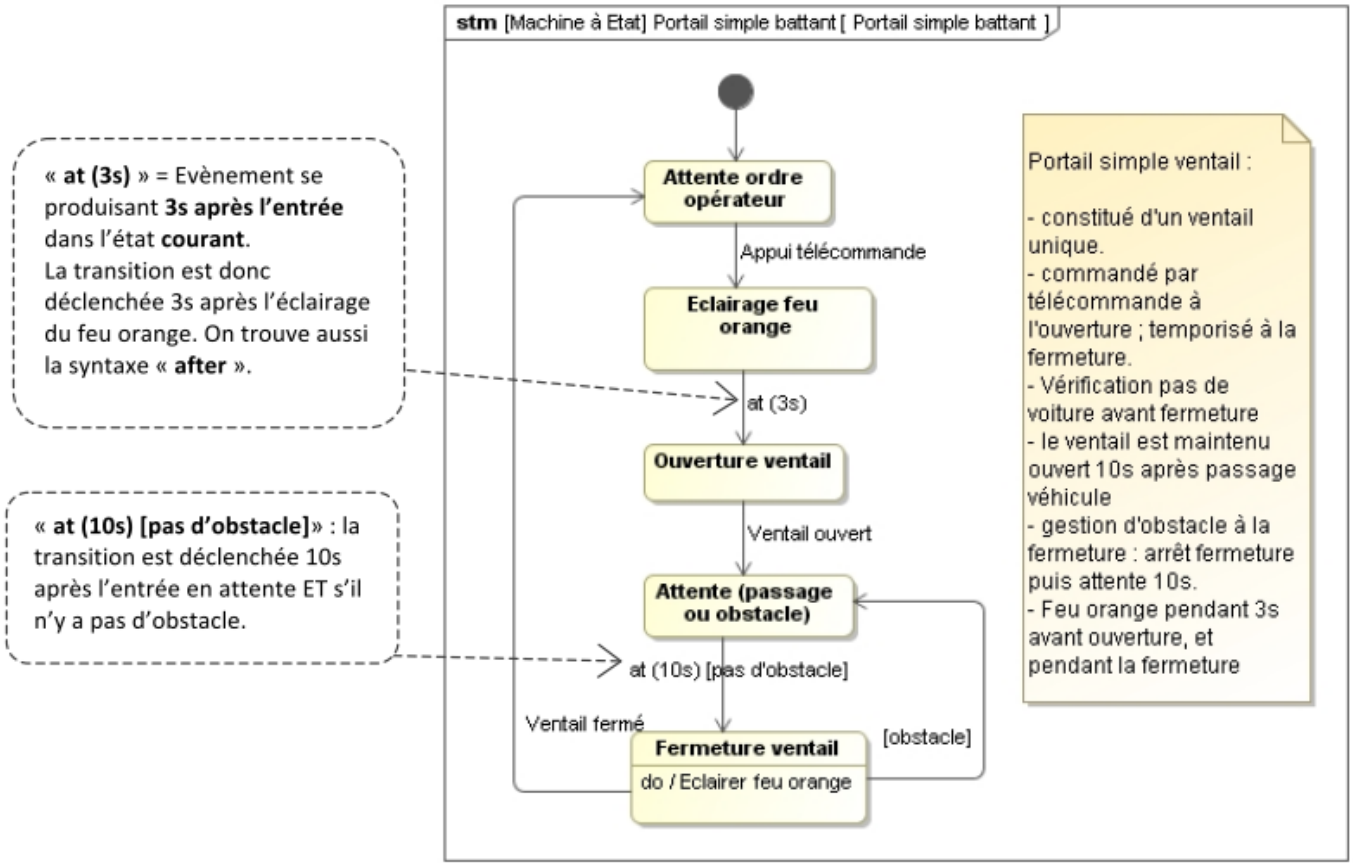
[2] D'après D.JOLIVET Les systèmes à évènements discrets SII sur internet Cours-séqu-2014-SysML-DJ.pdf

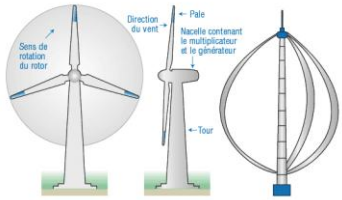


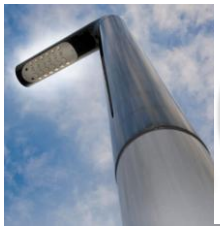


Prise en compte du temps : réaliser des activations d'états pendant des durées déterminées.

Dans la commande d'un portail l'ouverture doit débuter après un délai pendant lequel le voyant d'avertissement orange clignote.





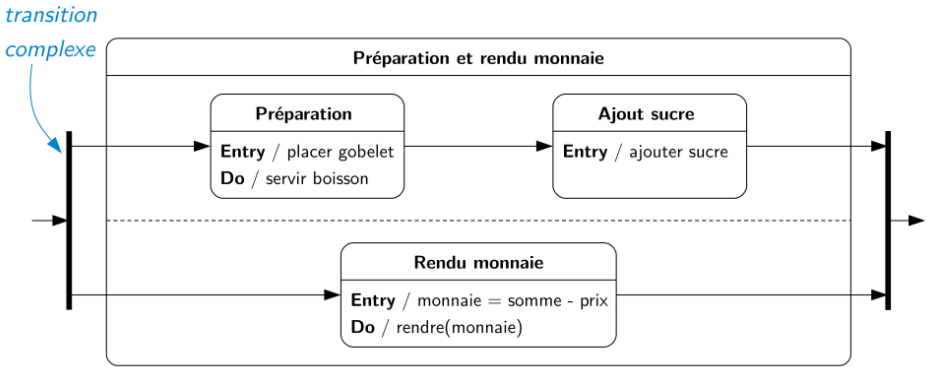


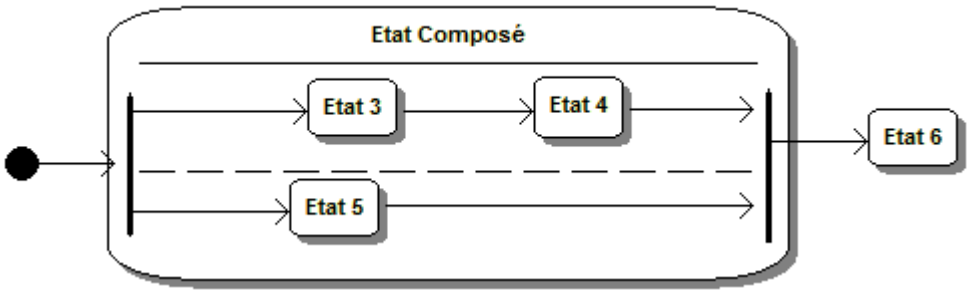
Compléments sur les graphes états transitions du SysML[3]

États composites

État orthogonal : État composite dans lequel plusieurs états sont actifs simultanément (concurrence/parallélisme)

État actif global = un état actif par région





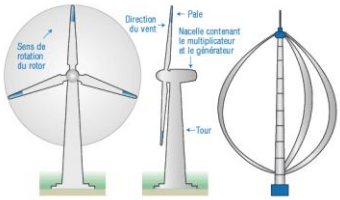
STI2D_SYSML_Construction_et_exemples.docx

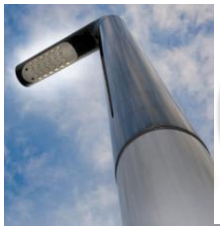
La concurrence a été induite ici par une **barre** de **synchronisation** (*représentation de points de synchronisation ou « points de rencontre »*). Les **transitions** associées à une **barre** de **synchronisation** sont **franchies** en même temps.

Depuis l'état initial, on atteint **directement** l'état « **Etat 3** » et l'état « **Etat 5** » de l'état **composé**. Ils se dérouleront en **parallèle** (*concurrence*).

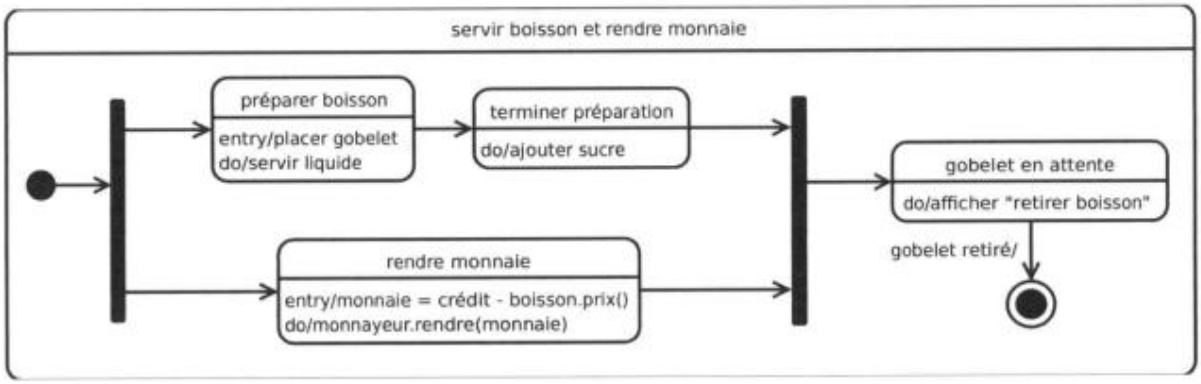
L'état **composé** sera **terminé** lorsque chaque **région** aura atteint son état **final** respectif (*soit l'état « Etat 4 » et l'état « Etat 5 »*) et que les **transitions** associées pourront être **franchies**. On se retrouvera alors dans l'état « **Etat 6** ».

[3] Extrait de Delphine LONGUET, UML Diagrammes états transitions, Polytech, Paris-Sud





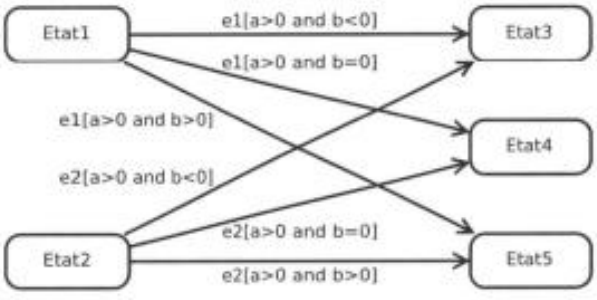
Dans le cas d'utilisation de transition complexe l'état orthogonal est devenu inutile.

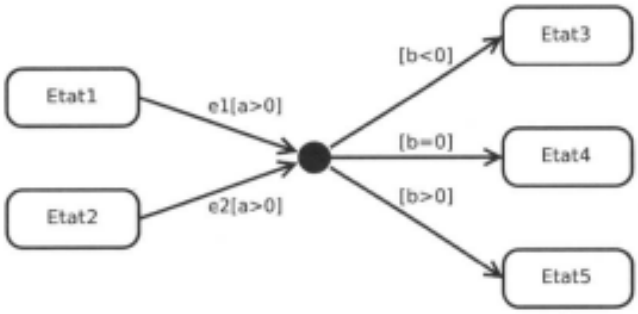


Pour activer l'état gobelet en attente il faut que les deux branches concurrentes soient achevées.

Utilisation de points de jonction

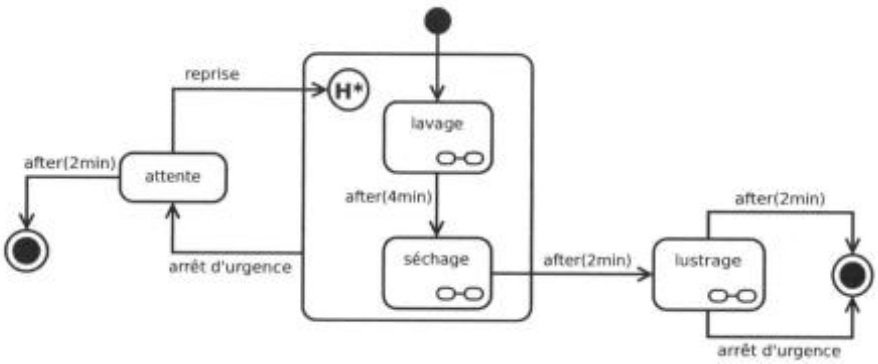
Pour simplifier la représentation des automates :

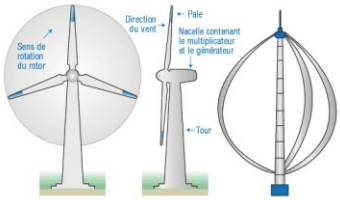


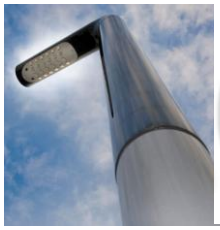


Etat composite

Pour une description hiérarchique des diagramme états-transitions







5 Synthèse de la syntaxe des graphes états transitions :