

Les systèmes à événements discrets

Partie 2 : LES SYSTEMES SEQUENTIELS

Avertissement, programme de CPGE.

La description des systèmes à événements discrets séquentiels se fait grâce à trois diagrammes SysML : le diagramme de séquence (SD), le diagramme d'état (SMD), le diagramme d'activité. Seuls les deux premiers diagrammes sont au programme de CPGE.



Le programme CPGE précise que vous devez être capable de lire ces diagrammes.

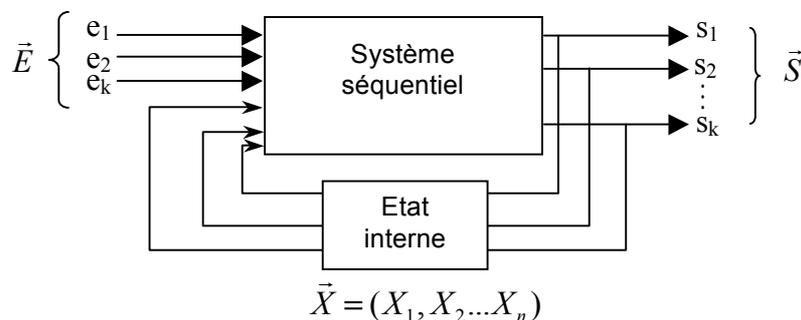
Vous devez donc connaître le vocabulaire SysML de ce cours en termes anglais et français ainsi que les abréviations consacrées.

I. INTRODUCTION – DEFINITION

1.1. Définition d'un système à événements discrets séquentiel.

Un système séquentiel est un système à événements discrets dans lequel l'état des sorties dépend à la fois de l'état des entrées et de ses évolutions passées.

Un système séquentiel évolue **étape par étape**.



Il existe en fait un **état interne** fonction des évolutions passées, donc des sorties. L'état des sorties est donc fonction des entrées, ET **de l'état interne** du système.

L'exemple basique de système séquentiel est la commande d'éclairage (E) par bouton poussoir (m) monostable :

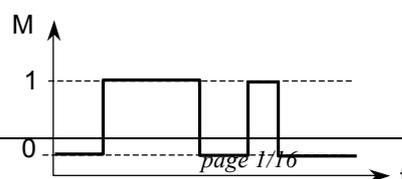
- Lieu non éclairé $E=0$. Appui $m=1 \Rightarrow E=1$.
- On relâche le bouton $m=0$. L'éclairage est maintenu $E=1$.
- On appuie à nouveau $m=1 \Rightarrow E=0$ (extinction éclairage)

Conclusion : pour le même état de l'entrée m ($=0$), il ya 2 états différents de la sortie E .
Idem pour $m=1$: on peut avoir éclairage ou extinction.

1.2. Représentation temporelle d'une variable logique : Chronogramme

Le chronogramme représente l'état logique d'une variable booléenne en fonction du temps. Le temps est représenté en abscisse, l'état de la variable logique booléenne est en ordonnée (0 ou 1).

Exemple : alimentation d'un moteur M . Si le moteur est alimenté on écrit $M=1$, s'il n'est pas alimenté on écrit $M=0$. Le diagramme ci-dessous représente l'évolution temporelle du signal logique M .



Pour un système comportant plusieurs variables logiques, on indique les variables en les positionnant les unes sous les autres.

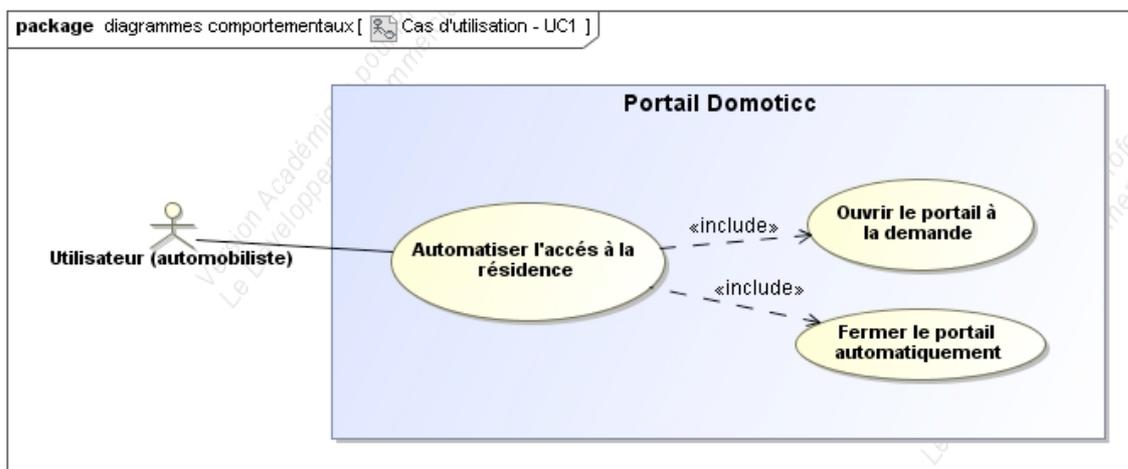
II. LE DIAGRAMME DE SEQUENCE SYSML

2.1. Rappel : les cas d'utilisation d'un système (diagramme UCD)

Le diagramme des **cas d'utilisation** (= **Use Case Diagram**) permet de modéliser le ou les **services rendus** par un système, à un ou plusieurs acteurs.

L'UCD répond à la question « quels services rend le système ».

Pour le portail Domoticc présent dans le laboratoire de Sii, on peut proposer le **diagramme des cas d'utilisation** suivant :



Remarque : ce diagramme pose les éléments du contexte et quelques interactions. L'UCD sous-entend voire « suscite » éventuellement un scénario.

Le **diagramme de séquence** que nous allons expliciter ci-dessous permet d'exposer et d'explicitement un scénario d'utilisation.

2.2. Le diagramme de séquence SysML (= Sequence Diagram, SD)

Le diagramme de séquence décrit le(s) scénario(s) liés au(x) cas d'utilisation.

Le diagramme de séquence documente le diagramme des cas d'utilisation précédemment décrit.

Le diagramme de séquence fait partie des diagrammes SysML de type **comportemental**. C'est un **diagramme temporel** qui représente l'enchaînement séquentiel des **interactions** nécessaires pour assurer un cas d'utilisation.

Boîte noire/boîte blanche

Le diagramme de séquence peut être réalisé pour le système global : le système est alors considéré comme une **boîte noire** et on ne distingue pas ses interactions internes. On ne détaille que les interactions entre le système et un acteur extérieur (bloc, utilisateur, etc.).

On peut aussi décrire les interactions internes au système. On parle alors de **boîte blanche** et on détaille certains blocs contenus dans le système. Tout dépend du niveau de détail souhaité (zoom).

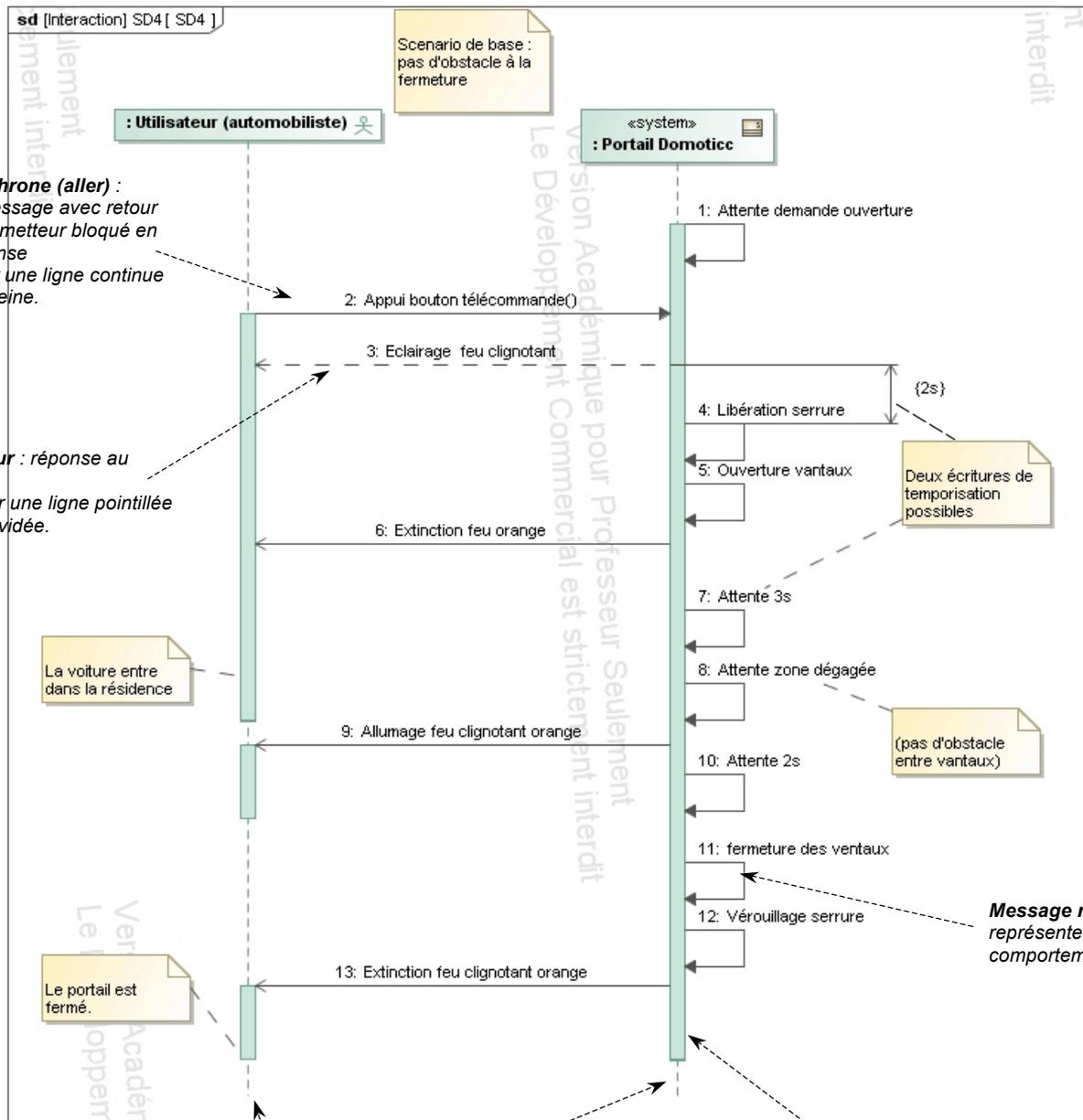
Scénario

Le SD présente un **scénario** possible lors du fonctionnement du système. On peut donc réaliser autant de SD qu'il existe de scénarios d'un système : scénarios de réussite, scénarios d'échec (ex : on place sur une balance électronique maxi 3kg, un objet pesant 4kg), scénarios de dysfonctionnement (plantage d'un appareil électronique).

En fait on réalise les SD des scénarios que l'on souhaite montrer.

Dans un premier temps on réalise souvent un **diagramme de séquence système (DSS)** présentant le système comme une « boîte noire ». On y représente deux lignes de vie : l'utilisateur et le système.

Observons le cas d'utilisation principal du portail « Automatiser l'accès à la résidence ». Nous pouvons proposer le diagramme de séquence suivant.



Ligne de vie : ligne verticale en pointillés, représentant l'existence d'un élément participant au SD. Deux l. de v. ici : l'utilisateur et le portail. La 1^{ère} l. de v. située à gauche est généralement un acteur. Les lignes de vie déclenchent une activité. Il peut y avoir plus de deux lignes de vie.

Activité : représente le travail de l'élément support de la ligne de vie

2.3. Généralisation : ce qu'il faut savoir sur la syntaxe du diagramme de séquence

... et qui n'a pas été défini dans l'exemple précédent.

Définition d'un message : élément de communication unidirectionnel entre deux lignes de vie qui déclenche une **activité de l'émetteur**. La réception du message provoque une activité du récepteur. Le **message retour** s'indique en pointillés. Cela signifie qu'il s'agit de la réponse directe au message précédent.

Alors qu'un message **synchrone** attend un retour de la part du récepteur, le **message asynchrone** n'attend aucun retour et son émetteur. Un message asynchrone se représente par une flèche évidée.

Attention : un message est le plus souvent à durée nulle.

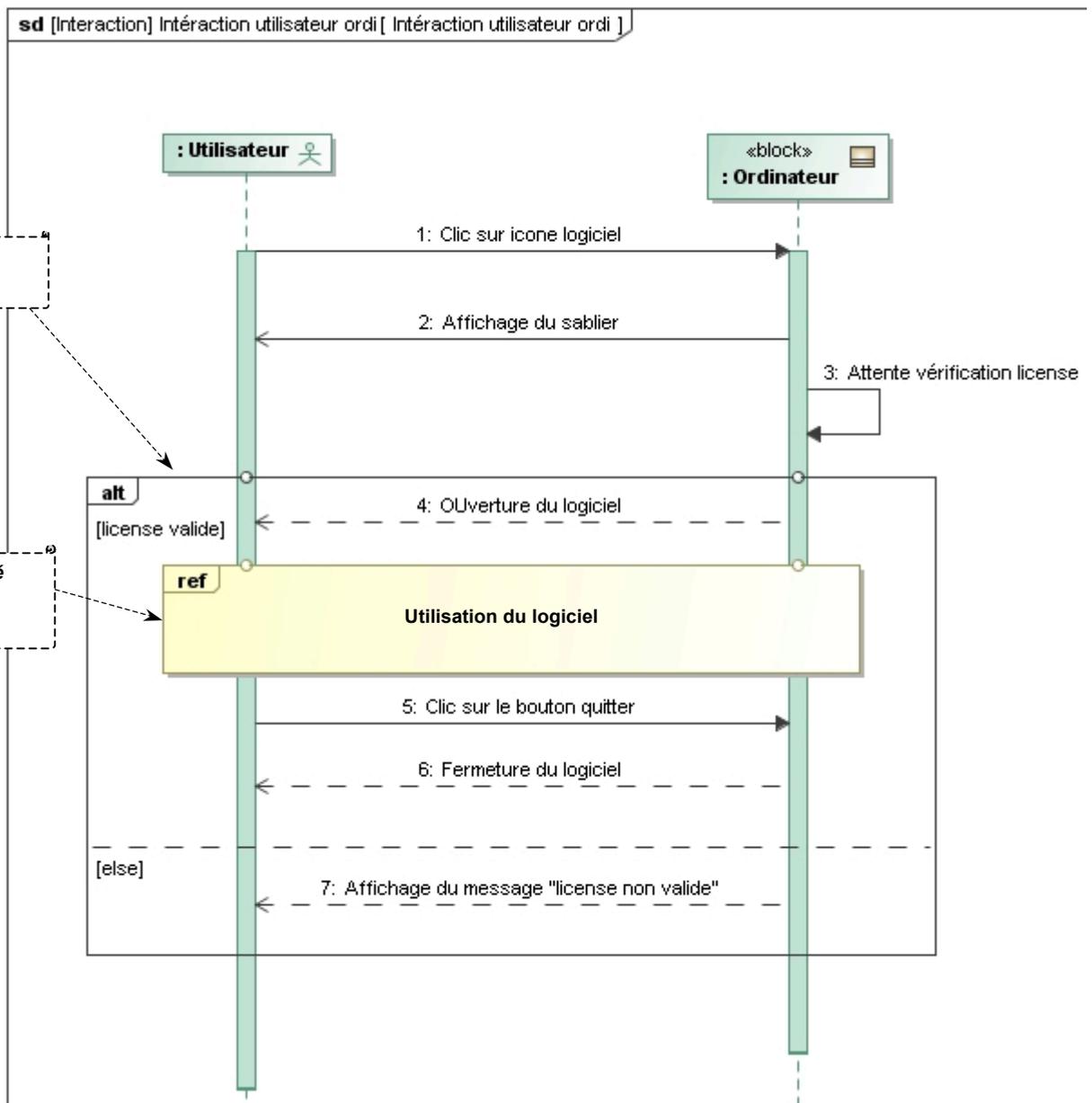
Définition d'une activation : les bandes verticales le long de la ligne de vie représentent une période d'activation de celle-ci.

Fragments combinés : utilisés pour montrer des variantes dans un même scénario. Les fragments combinés doivent être utilisés avec parcimonie car les diagrammes de séquence ne sont pas des algorigrammes. S'il y a trop de variantes (donc une grande variété de scénarios en fait) il vaut mieux faire plusieurs diagrammes de séquences.

Il existe plusieurs types de fragments combinés. Les cinq principaux sont :

- **Opt** : le contenu du bloc est exécuté si la [condition] est respectée
- **Alt** : le premier des blocs est exécuté si la [condition] est respecté, sinon c'est le deuxième. Il peut y avoir plus de deux compartiments, les [conditions] doivent être complémentaires bien sûr.
- **Loop** : répète la séquence tant que la [condition] est respectée
- **Ref** : intègre un diagramme de séquence dans un autre diagramme de séquence
- **Par** : exécute deux blocs de séquence en parallèle

Illustrons deux de ces fragments combinés dans l'exemple ci-dessous... bien connu !



2.4. Si vous devez réaliser un diagramme de séquence...

Pour représenter un diagramme de séquence je vous propose la méthode suivante.
Crayon/gomme/brouillon obligatoires !

- a. Bien identifier et comprendre **le scénario**. Si le scénario n'est pas clairement exposé, ou si vous l'avez mal identifié, vous risquez de noyer votre SD dans des détails inutiles (fragments combinés *alternatifs* inutiles par exemple), ou écrire un scénario inadéquat.
- b. Représenter les lignes de vie en traits pointillés et indiquer le nom des blocs ou acteurs. Soit les lignes de vie sont imposées, soit vous devez les identifier. Cela n'est pas difficile, mais il peut s'avérer quand vous réaliser le diagramme, qu'une nouvelle ligne de vie est nécessaire. Ou bien un bloc est trop global et il faut le détailler : il faudra alors refaire le diagramme de séquence.
- c. Représenter la succession de messages entre les ldv. Ne vous préoccupez pas trop du type de message dans un premier temps (synchrone/asynchrone). Surtout écrivez en traits fins car vous aurez besoin de gommer.
- d. Représenter les fragments combinés simultanément aux messages. Cela n'est pas facile car il faut parfois anticiper. Il ne doit pas y avoir trop de fragments combinés. Si c'est le cas, si le scénario semble complexe (nécessité de fragments imbriqués, de séquence alternatives) : peut être le scénario a-t-il mal été identifié ?
- e. Prenez du recul, relisez, vérifiez la cohérence et la pertinence de la séquence.
- f. Recopier au propre en différenciant les types de messages (synchrone/asynchrone).

III. LE DIAGRAMME D'ETAT SYSML (State Machine Diagram = SMD)

3.1. Définition

Le **diagramme d'état**, encore appelé **machine d'état** (SMD), montre les différents **états** pris par un système ou un sous système en fonction d'**interactions avec le milieu extérieur ou internes**.

Il répond à la question : « comment représenter les différents états du système ? ».

Définition d'un état. Un état est une situation durant la vie d'un **bloc**, décrite par un enchaînement de différentes activités :

- activités exécutées en entrant dans l'état (repérées par le terme « entry ») ;
- celles exécutées tant que l'on reste dans l'état (terme « do ») ;
- celles exécutées en sortant de l'état (terme « exit »).

L'état est représenté par un rectangle aux coins arrondis dans le SMD.

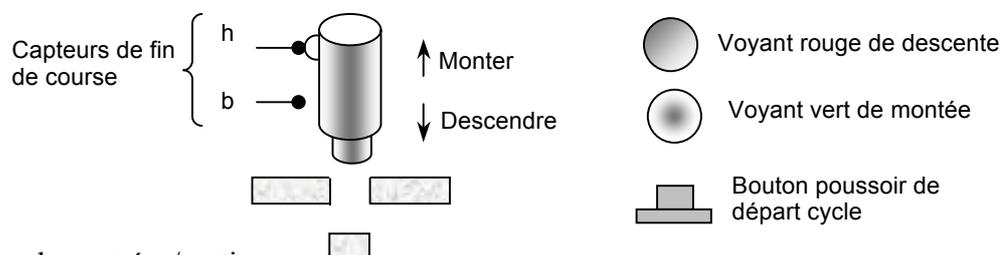
On lui attribue un substantif d'action appelé **activité** de l'état (*Eclairage voyant, Rotation moteur, Attente, Comptage, Déplacement, etc.*).

Remarque : le système étudié (ou le sous système) est appelé **bloc**. Un **bloc** passe par une **succession d'états** au cours de sa vie.

Important, dans un SMD donné : un seul état peut être actif à un instant donné... **SAUF** dans des états orthogonaux dont le découpage en plusieurs régions permet la simultanéité d'états différents. Voir la fin du cours.

3.2. Un premier diagramme d'état : vocabulaire et les notions de base

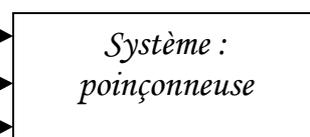
Exemple d'une poinçonneuse, cahier des charges de la séquence de poinçonnage
 Quand le départ cycle est demandé par l'opérateur, un signal sonore de prévention retenti. Le poinçon descend et un voyant rouge s'allume. Quand le poinçon atteint sa position basse, un voyant vert s'allume, et il remonte. Arrivé en position haute il attend un nouvel ordre de l'opérateur. Si on arrête la machine, le poinçonnage ne peut plus avoir lieu.



Bilan des entrées/sorties

Entrées :

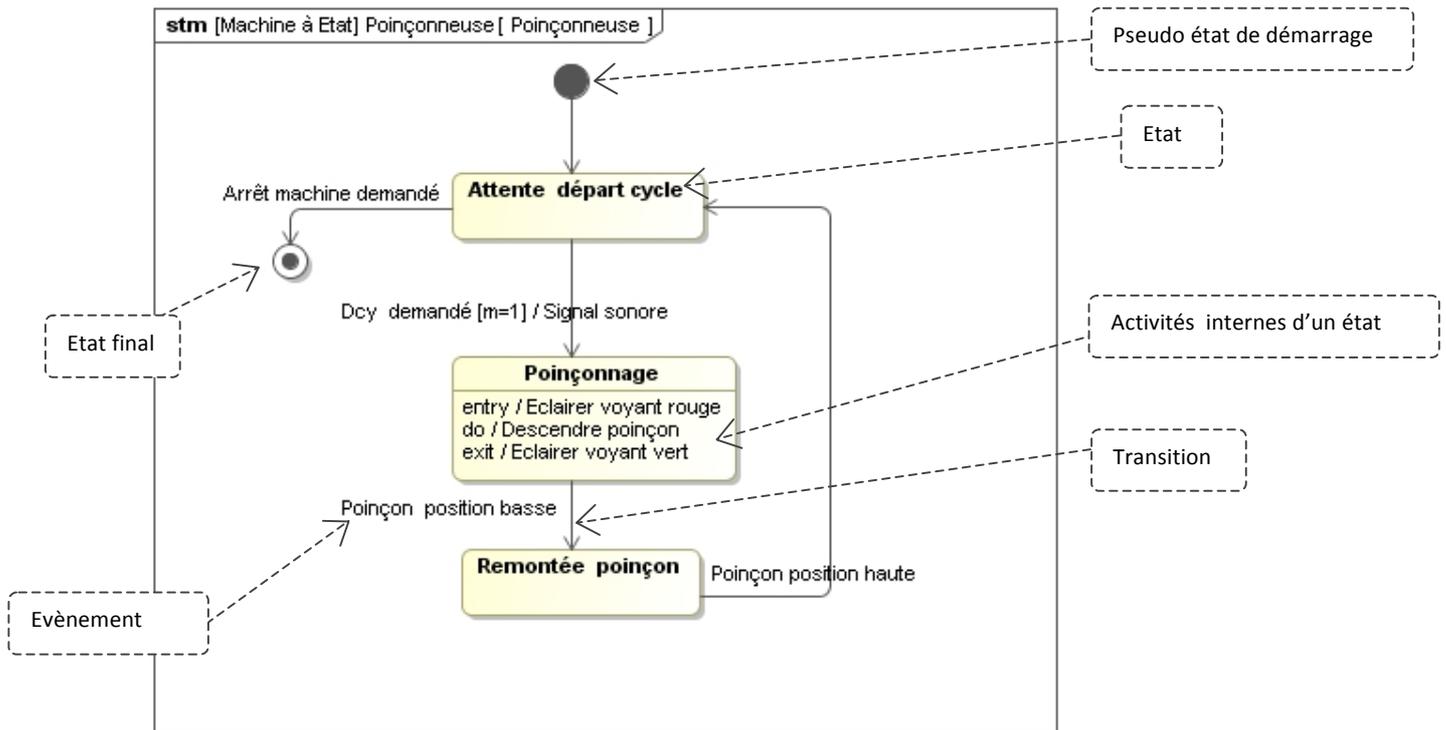
Départ cycle
 Poinçon haut
 Poinçon bas



Sorties :

Monter poinçon
 Descendre poinçon
 Voyants de descente / montée





Pseudo état de démarrage : précède la première **transition** à franchir au lancement du diagramme d'état. Ne peut être suivi que d'une seule transition. Ne peut pas être la cible d'une transition. Pas d'activité associée. Il ne peut y avoir qu'un seul pseudo état de démarrage dans un SMD. (Sauf dans des **états orthogonaux**).

Pseudo état de fin : termine une séquence décrite dans le SMD. Il peut y avoir plusieurs pseudo états finaux dans un SMD. Quand un état final est atteint toute la séquence décrite dans le SMD s'arrête.

Plusieurs **transitions** peuvent aboutir à un même pseudo état final.

Transition : une transition lie deux états 1 et 2. Elle représente la possibilité de sortie d'un état 1 **d'entrer** dans un état 2. L'état précédent est appelé la **source** de la transition, l'état suivant est la **cible**.

Bien entendu, une transition n'a la possibilité d'être **déclenchée** que si le **bloc** se trouve dans l'**état source** de la transition.

Le déclenchement d'une transition a pour conséquence : la **sortie de l'état source et l'entrée dans l'état cible**. Le déclenchement d'une transition implique un changement d'état du **bloc** : il était dans l'état 1, il se trouve maintenant dans l'état 2.

Une transition ne possède pas toujours un évènement associé. On l'appelle alors **transition automatique**.

On parle de **déclenchement/déclencher** une transition (On peut aussi rencontrer le vocabulaire « franchissement/franchir » une transition : à éviter).

Evènement : occurrence associée à une **transition** et qui peut la **déclencher**.

Un évènement se produit à un instant déterminé dans le temps et possède une durée nulle.

On peut adjoindre à la transition une **condition de garde** et/ou un **effet** (voir la syntaxe complète en exemple ci-dessus : *Dcy demandé [m=1] /Signal sonore*).

La condition de garde (booléenne) est évaluée si l'évènement se produit : quand la condition de garde est vraie on dit alors que la transition est **validée**.

L'effet est une action consécutive du franchissement de la transition (ici le *signal sonore* est **l'effet** qui se produit sa transition est déclenchée).

Activités internes, entry, do, exit, explication de l'exemple :

La poinçonneuse est dans l'état « Attente de départ de cycle ». Quand l'évènement « Départ cycle » se produit et que le bouton marche est enfoncé ($m=1$), un signal sonore est émis. La transition vers l'état suivant « poinçonnage » est alors déclenchée.

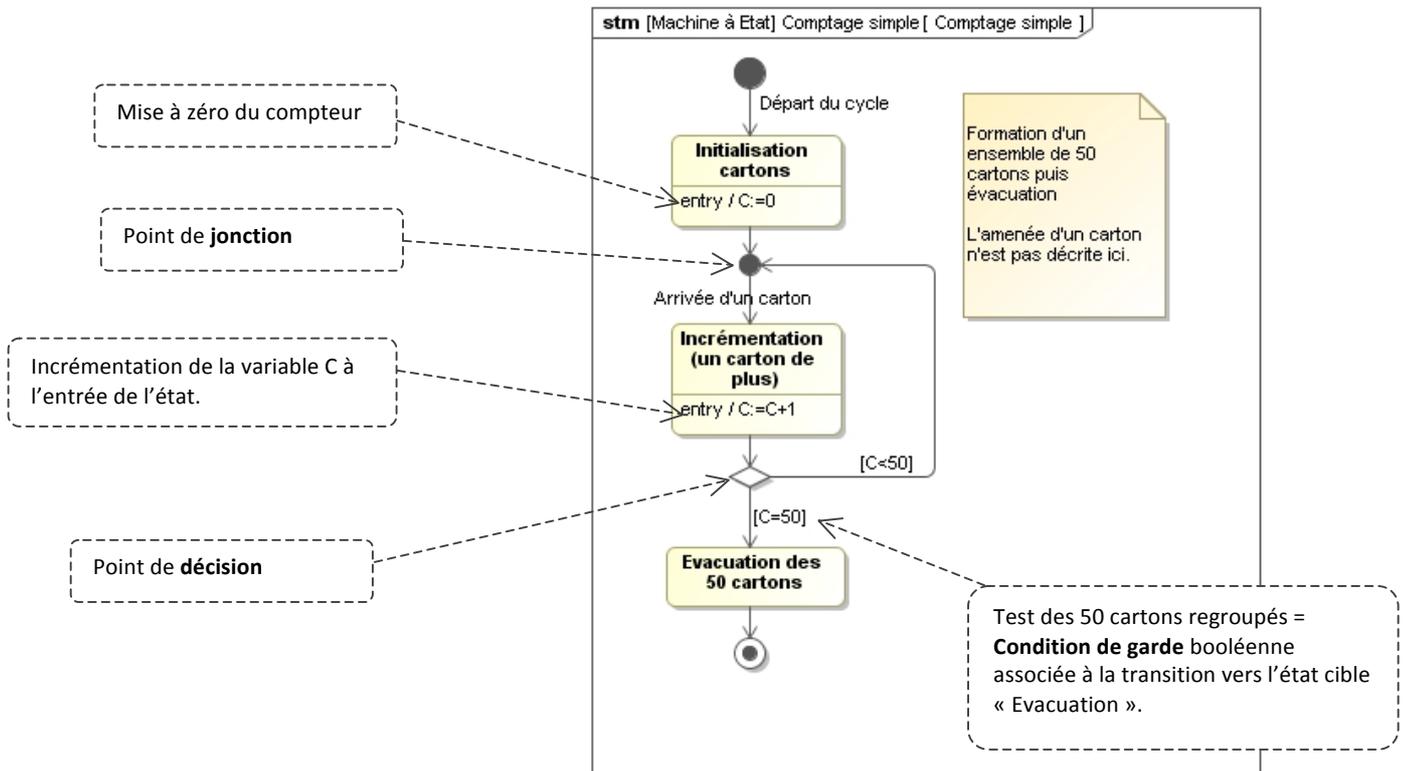
La poinçonneuse quitte l'état « d'attente », et entre dans l'état de « poinçonnage ». Cet état possède des activités internes : un voyant rouge s'allume à l'entrée de l'état, la descente du poinçon a lieu.

Quand il est en position basse, la transition suivante est franchie, la poinçonneuse sort de l'état de poinçonnage, et l'effet de sortie « éteindre le voyant » a lieu.

3.3. Quelques fonctions plus avancées

3.3.1. Comptage / Reprise de séquence

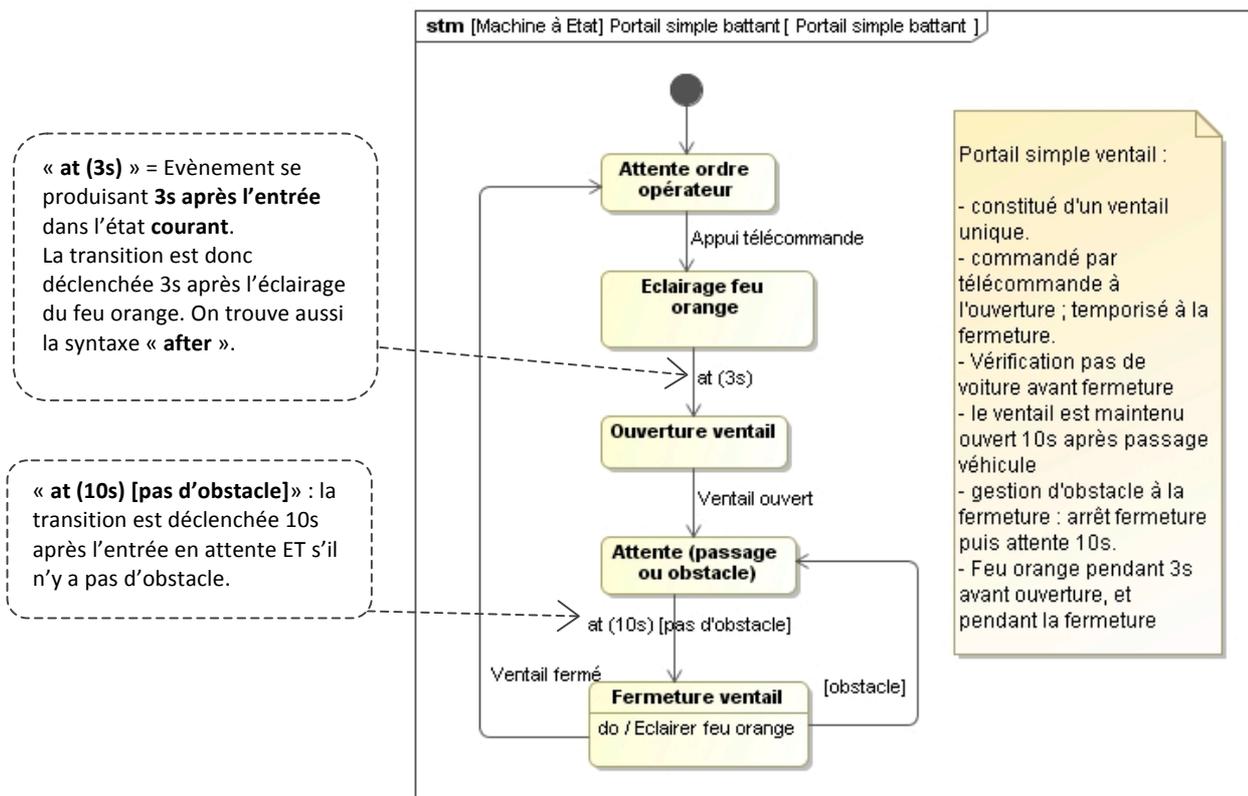
Cahier des charges : à la demande du départ cycle par l'opérateur, on doit compter 50 cartons, puis les évacuer. Le système d'approvisionnement de chaque carton n'est pas l'objet de l'étude ici. Seul le système de comptage doit être décrit. Seul l'évènement d'arrivée d'un carton déclenche donc le comptage.



Remarques :

- le **point de jonction** est un pseudo état. Il ne possède pas d'activité associée.
- le **point de décision** (petit losange), doit être suivi de deux transitions dont les conditions de garde sont **complémentaires**, donc incompatibles, donc qui ne peuvent se produire au même instant. Dans le cas contraire le SMD serait **non déterministe** ! Si ambiguïté, utiliser la condition *[else]*. Ici les conditions booléennes $[C=50]$ et $[C<50]$ ne peuvent avoir lieu simultanément.
- C est une variable **interne** au processus.

3.3.2. Evènements temporels



Remarque : il existe aussi l'évènement temporel **absolu** *when*. « *When heure courante = heure alarme* » par exemple. Dans ce cas on évalue un évènement par rapport au temps absolu de l'horloge, pas la durée écoulée depuis l'entrée dans un nouvel état.

3.3.3. Etat composite, état sous machine, états orthogonaux

Note : Nous profitons de ce paragraphe pour découvrir les **bifurcations (fork)** et **unions (join)**.

Prenons l'exemple du doseur Malaxeur dont la description est donnée en *annexe I* à la fin de ce cours.

Etat sous machine

Par choix, et par soucis de simplification de lecture du SMD, nous ne voulons pas détailler toute la séquence du cahier des charges ni chacun des états du bloc. Nous créons donc des **états sous machine**, eux même incluant une séquence éventuellement détaillées dans un autre SMD.

L'état sous machine est représenté par le petit haltère (en fait ce signe représente de manière symbolique deux états et une transition pour montrer qu'il y a un SMD à l'intérieur même de cet état).

⇒ Exemple du doseur/malaxeur : les états « Dosage des produits » et « Amenage des briquettes » sont des **états sous machine**. Chacun inclut une séquence qui peut être détaillée dans un autre SMD.

Etat composite

Un état composite possède une seule région où les états sont mutuellement exclusifs (= un seul état actif à la fois).

- Malaxage/évacuation = **état composite détaillé**.

Attention, règle de sortie de l'état composite : la **transition de sortie** d'un état composite est **évaluée** quelque soit l'état actif de l'état composite. En conséquence, la transition peut être déclenchée quelque soit l'état actif à l'intérieur de l'état composite. On peut donc sortir de l'état composite alors que son dernier état n'est pas actif.

- ⇒ *Exemple du doseur malaxeur : l'évènement « malaxeur en bas » est évalué dès qu'on est dans l'état composite malaxage/évacuation. Dès que l'évènement « malaxeur en bas » a lieu, sa transition est déclenchée et on quitte les états composant l'état composite « malaxage/évacuation ».*

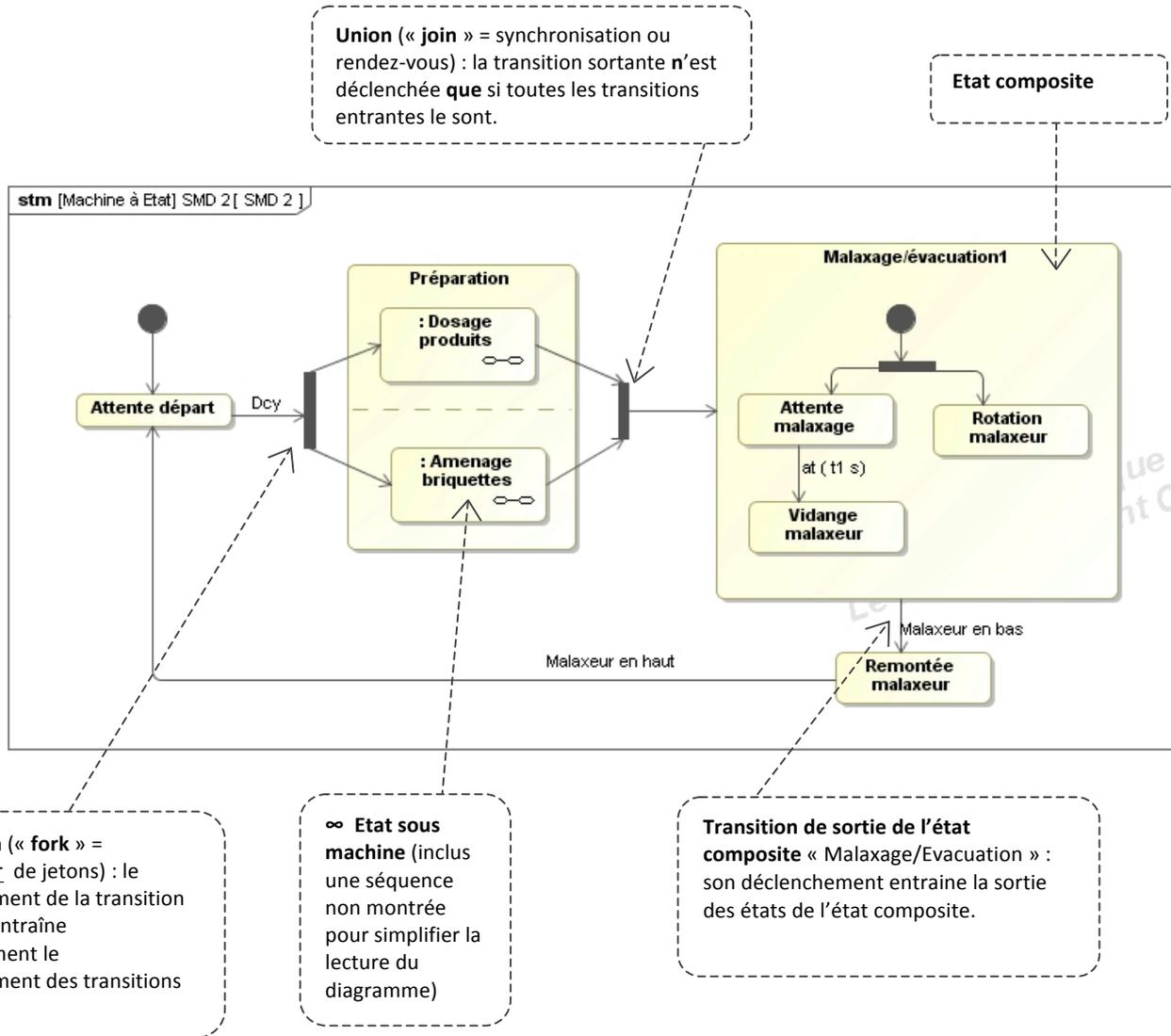
Etat orthogonal

Un état orthogonal est un état composite incluant **plusieurs régions**, donc, comportant autant d'états actifs que de régions (au plus !).

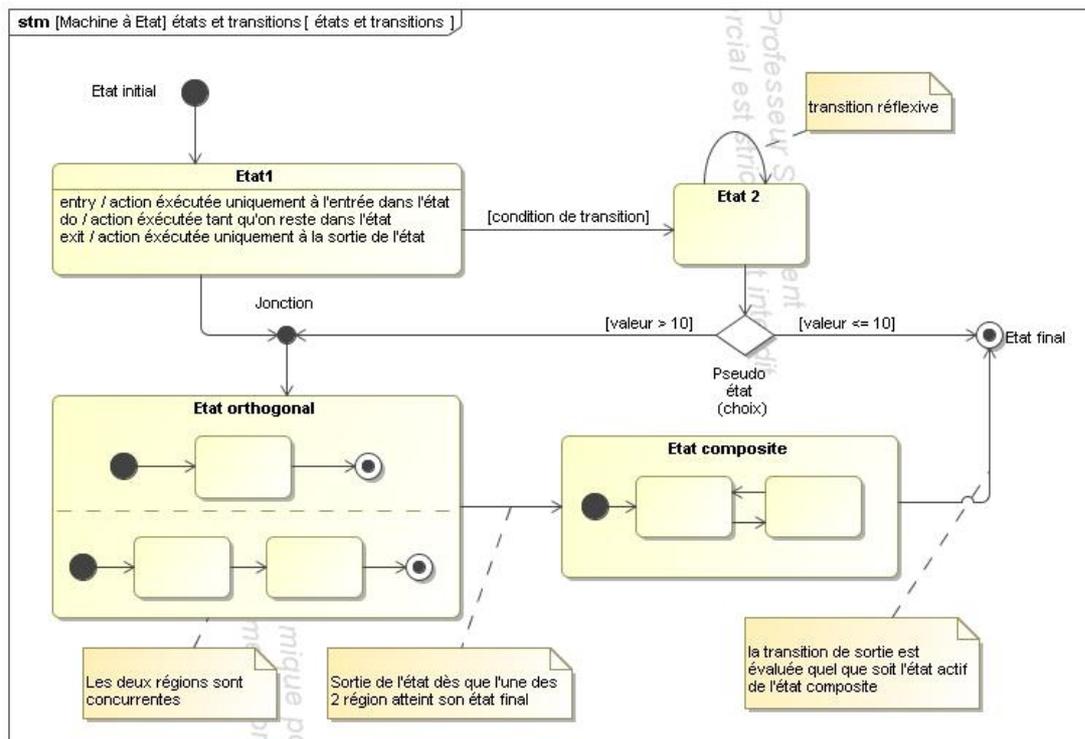
La sortie de l'état orthogonal se fait donc conformément à règle de sortie de l'état composite énoncée ci-dessus.

Mais cette règle de sortie peut-être gênante pour certaines séquences où on a besoin d'un **comportement synchrone** (= atteinte de tous les états finaux pour mettre fin à l'état global). Dans ce cas il faut adjoindre un pseudo état « join » (bifurcation). Voir l'exemple du doseur malaxeur.

- ⇒ *Exemple du doseur/malaxeur : l'état orthogonal « Préparation » est composé de deux régions concurrentes incluant les états sous machines « Dosage produit » et « Amenage briquettes ». Le pseudo état **join** garanti qu'on sortira de l'état orthogonal quand le dosage et l'amenage seront terminés (« rendez-vous »).*



3.4. Synthèse de la syntaxe des diagrammes d'états



3.5. Méthode de création d'un diagramme d'état

Pour créer un diagramme d'état, je vous propose la méthode simple suivante :

- a. Mettre en place les états
- b. Les relier par les transitions
- c. Indiquer les événements (et/ou condition de garde + effets éventuels)
- d. Vérifier la cohérence, simuler.

- FIN DU COURS SUR LES SYSTEMES SEQUENTIELS -

Pour la réalisation de ce cours, je me suis aidé :

- Du livre de Pascal Roques « SysML par l'exemple »
- De la synthèse de Philippe Duthu sur la modélisation des systèmes par SysML
- Du site internet Uml-SysML France : www.uml-sysml.org , renvoyant notamment aux explications de Laurent Audibert.

Je remercie en outre Yann Le Gallou et Baudouin Martin, formateurs SysML, pour l'aide qu'ils m'ont personnellement apporté.

ANNEXE I : EXEMPLE DU DOSEUR MALAXEUR AUTOMATISÉ

Source : exemple cité dans la norme Grafcet, CEI 60848-2 de 2002

Mise en situation/Cahier des charges

Un malaxeur N reçoit des produits A et B préalablement dosés par une bascule C, et des briquettes solubles amenées une par une par un tapis. L'automatisme décrit ci-dessous permet de réaliser un mélange comportant ces trois produits.

L'action sur le bouton «Départ Cycle» provoque simultanément le **pesage** des produits et **l'aménagement** des briquettes de la façon suivante:

- dosage du produit A jusqu'au repère «a» de la bascule, puis dosage du produit B jusqu'au repère «b» suivi de la vidange de la bascule C dans le malaxeur;
- aménagement de deux briquettes.

Le cycle se termine par la **rotation du malaxeur** et son pivotement final au bout d'un temps t1, la rotation du malaxeur étant maintenue pendant la vidange.

