



Modélisation Sysml

Table des matières

I - Modélisation Sysml	3
1. Introduction.....	3
2. Présentation de l'exemple d'application.....	3
3. Modélisation structurelle CONTEXTE.....	4
4. Modélisation fonctionnelle EXIGENCE.....	5
5. Modélisation comportementale CAS D'UTILISATION	7
6. Modélisation structurelle DEFINITION DE BLOC	9
7. Modélisation structurelle DEFINITION DE BLOC INTERNE.....	12
8. Modélisation comportementale SEQUENCE.....	13
9. Modélisation comportementale ETAT	17

I Modélisation Sysml

1. Introduction

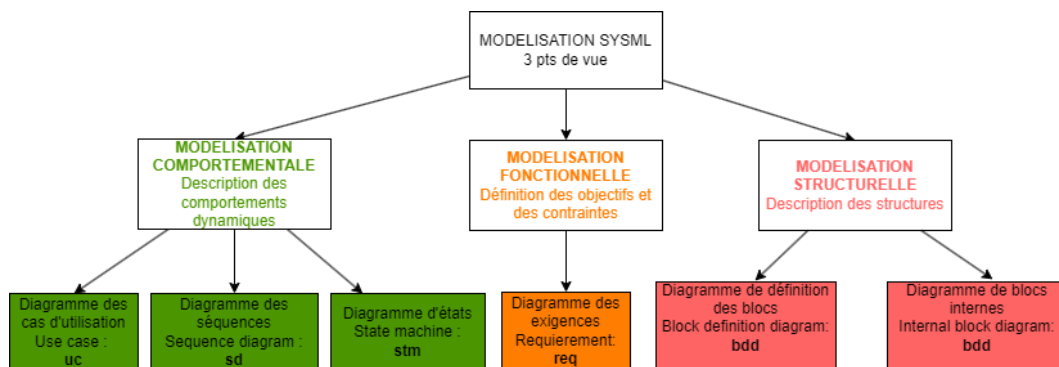
Le SysML (Systems Modeling Language) est un langage de modélisation graphique spécifique au domaine de l'ingénierie système. Il permet :

- l'analyse
- la conception
- les tests
- et la validation

de nombreux systèmes et systèmes-de-systèmes. Il est donc utilisé par tous les acteurs intervenant lors de la vie du système.

Mais ce n'est pas un outil exhaustif car il ne représente que les informations nécessaires permettant de saisir les intentions au travers de 3 points de vue et 9 diagrammes.

Les 6 diagrammes étudiés



Le diagramme de contexte

⊕ Complément

Ce diagramme non normalisé permet de présenter le système étudié dans son environnement utilise un diagramme de block (bdd) comme modèle.

2. Présentation de l'exemple d'application

Pour illustrer chaque diagramme Sysml, l'exemple utilisé sera une balance de cuisine dont les caractéristiques sont les suivantes :

- Précision de 1 g ;
- Charge maximale = 4kg ;
- Dimensions : 250 x 200x 40mm
- Interface Homme/Machine :
- Bouton Marche/Arrêt/Tare
- Bouton Conversion g/ml
- Afficheur LCD 4 digits
- Fonctionne avec une pile Cr2032 fournie

- Design épuré aux formes circulaires ;
- Gamme de 5 coloris tendances ;
- Conforme aux directives DEEE et RoHS

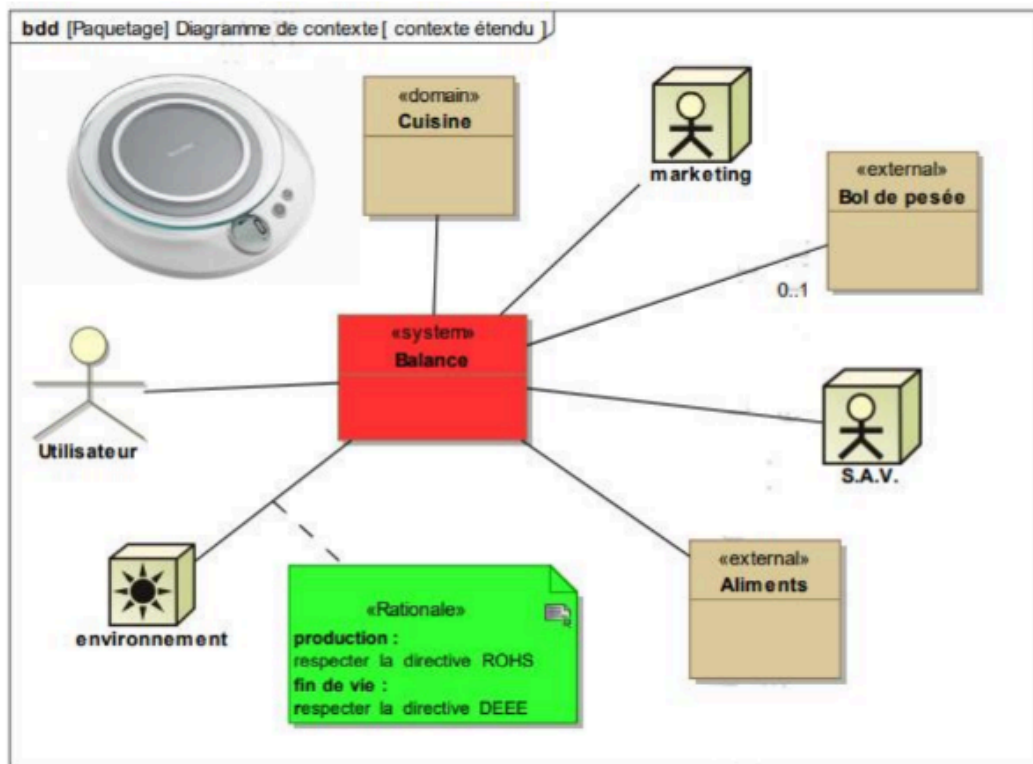


Balance Halo

3. Modélisation structurelle CONTEXTE

Diagramme de contexte

Il présente tous les éléments externes qui influencent le système étudié et le système lui-même. Il utilise la représentation des diagrammes de blocks BDD ou de blocks internes IBD.



Balance Halo

- Utilise un diagramme des cas d'utilisation ; voir chapitres suivants.
- Utilise un diagramme des exigences ; voir chapitres suivants.

4. Modélisation fonctionnelle EXIGENCE

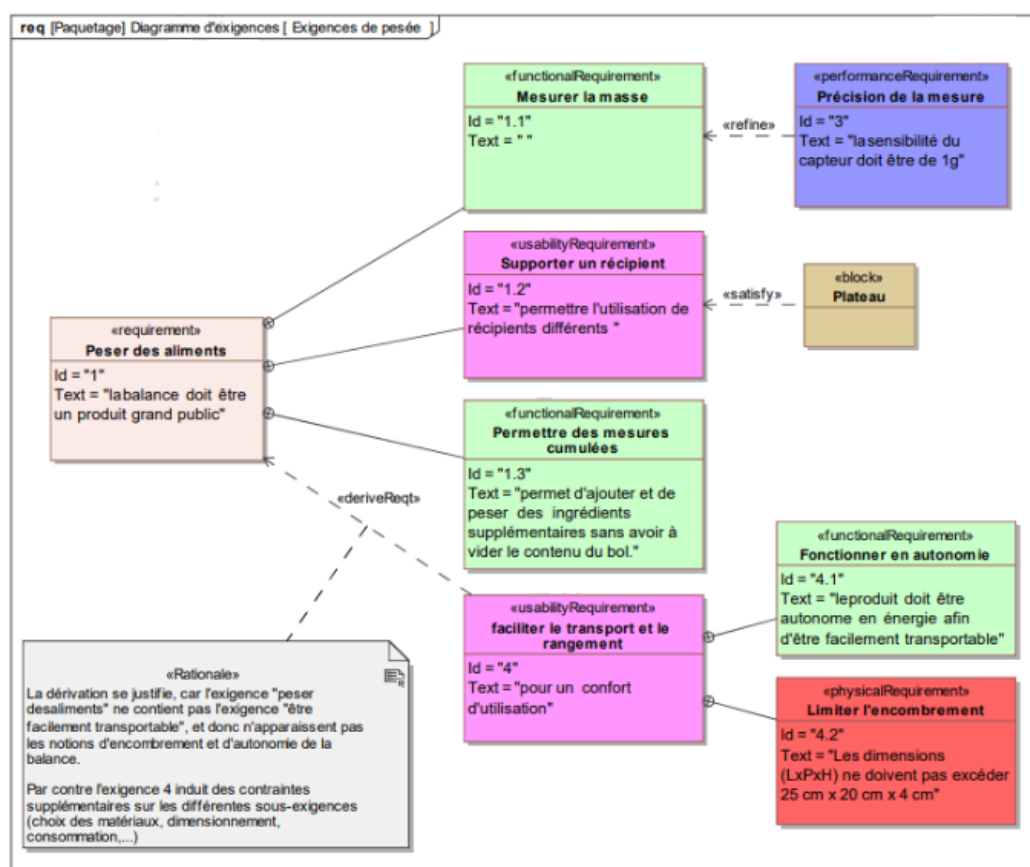
Diagramme des exigences (REQ)

Le diagramme d'exigence est une représentation du cahier des charges. Il doit recenser les fonctions et les contraintes du système qu'on appelle "Exigence" (Requirement). Les exigences servent à établir un contrat (au sens légal) entre les différents partenaires.

- Les fonctions et les contraintes sont exprimées de manière préférentiel par un verbe du premier groupe à l'infinitif

Le système devra ainsi réaliser des niveaux de contraintes ou les conditions de performances qui devront être satisfaites par le système.

Elles sont représentées par des cadres hiérarchisés contenant différentes informations (noms, identifiant, description, niveau de performance,...) reliés entre eux par des liens de contenance.



Cartouche du diagramme

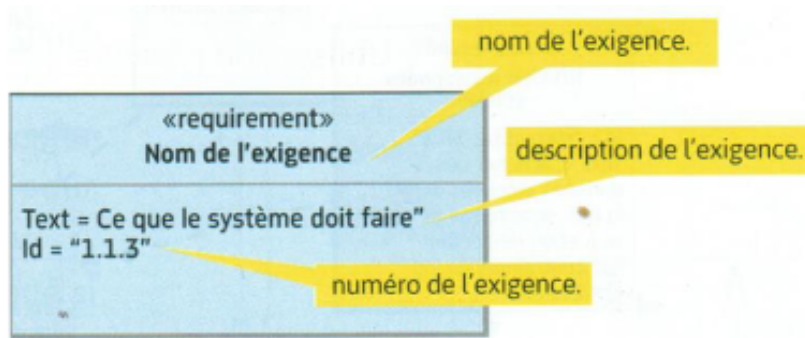
Ce cartouche situé en haut à gauche spécifie le type de diagramme et le nom du système concerné

req [Paquetage] Diagramme d'exigences [Exigences de pesée]

Propriétés de base d'un bloc

Un bloc d'exigence contient au minimum :

- Un libellé de fonction
- Un identifiant unique :id. Cet identifiant suivant rang 1. ou 1.1 ou encore 1.1.1 indique le niveau hiérarchique du bloc considéré
- Et presque systématiquement un texte descriptif



Les liaisons

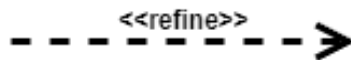
- Contenance

décompose une exigence composite en plusieurs exigences unitaire



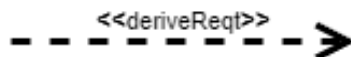
- Raffinement

<<refine>> permet l'ajout de précisions, des données quantitatives par exemple.



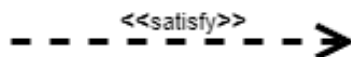
- Dérivation

<<deriveReq>> relie des exigences de niveaux différents, des exigences système à des exigences de niveau sous-système



- Satisfy

définit un composant issu du diagramme de bloc (bdd) qui satisfait l'exigence



Notes graphiques

Deux mots-clés particuliers ont été définis afin de les représenter :

- des problèmes à résoudre <<problem >>
- des justificatifs <<rationale >>

La traçabilité

Elle permet de s'assurer de la cohérence entre ce que fait réellement le projet et ce qu'il doit faire afin de faciliter l'analyse d'impact en cas de changement.

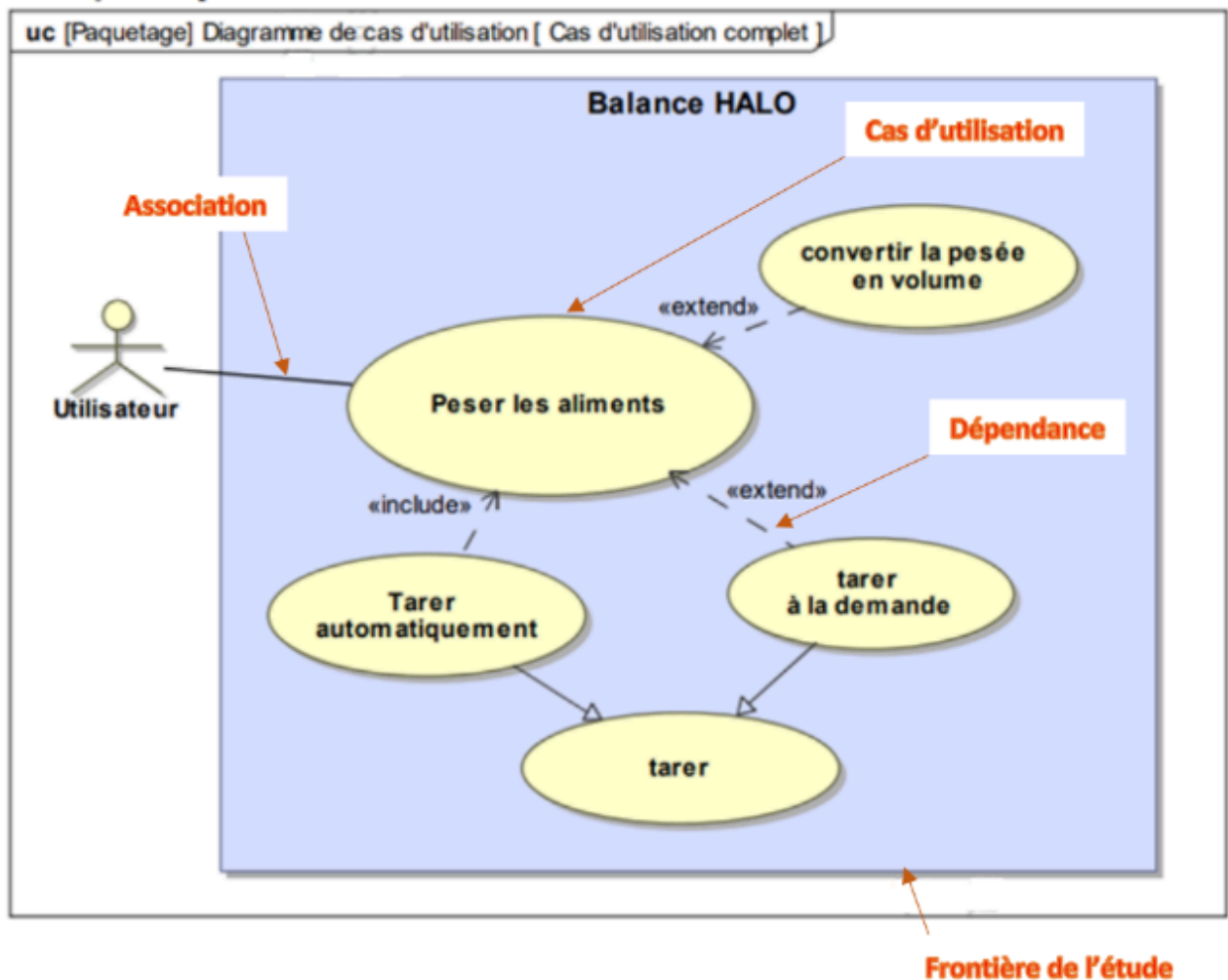
Le diagramme d'exigences permet ainsi tout au long d'un projet de **relier les exigences avec d'autres types d'élément SysML** par plusieurs relations :

- <<refine>> en liaison avec les diagrammes :
 - cas d'utilisation
 - diagramme d'états
 - , etc.
- <<satisfy>> bloc d'architecture
 - lien avec l'élément qui satisfait l'exigence

5. Modélisation comportementale CAS D'UTILISATION

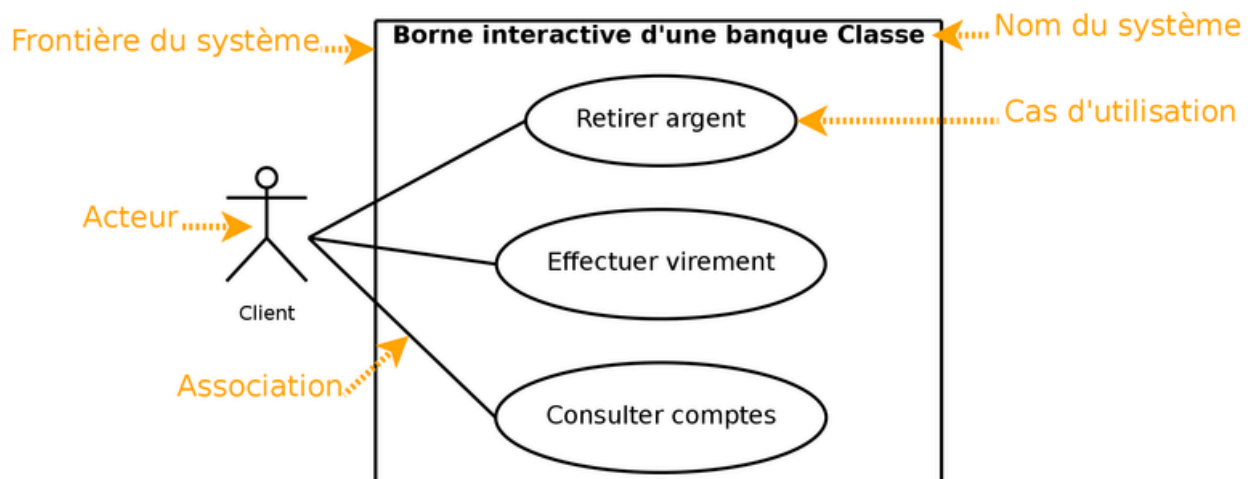
Diagramme des cas d'utilisation (UC)

- Il représente les services effectués par le système du point de vue de l'acteur sans spécifier comment il le fera.
- Il permet également de déterminer les frontières du système en le plaçant dans son contexte
- A ce niveau les fonctions internes ne sont pas décrites.



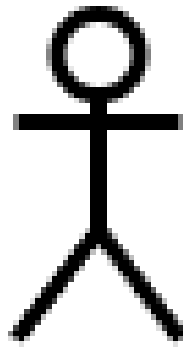
Nom du système

On rappelle le nom du système en haut de la frontière de celui-ci.



Les acteurs

Un acteur représente un rôle qui peut être tenu par un humain ou n'importe quel autre système. Il indique avec quoi le système sera en interaction.



Actor

L'acteur principal humain est celui pour qui le cas d'utilisation produit un résultat concret. Il est possible d'avoir plusieurs acteurs principaux.

- Le symbole est à gauche du diagramme

Les acteurs secondaires (non sont souvent sollicités pour des informations complémentaires ; ils peuvent uniquement consulter ou informer le système lors de l'exécution du cas d'utilisation.

- Leur symbole est à droite du diagramme

Cas d'utilisation

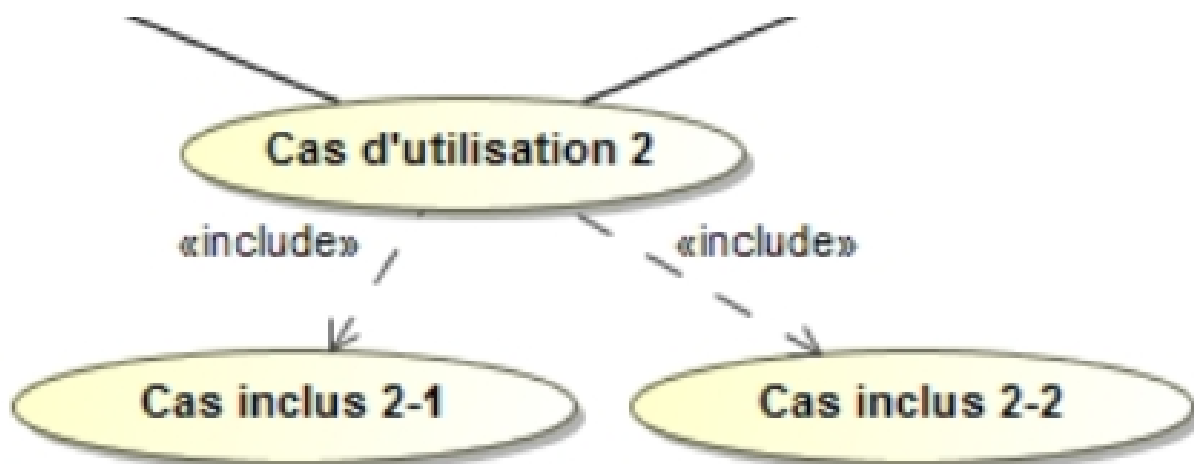
La recherche des cas d'utilisation consiste à trouver des cas dans lesquels le système est utilisé pour mener à bien une tâche spécifique d'un acteur.

- Chaque cas représente un ensemble de séquence d'actions réalisées par le système.
- Les cas d'utilisation sont exprimées de manière préférentiel par un verbe du premier groupe à l'infinitif

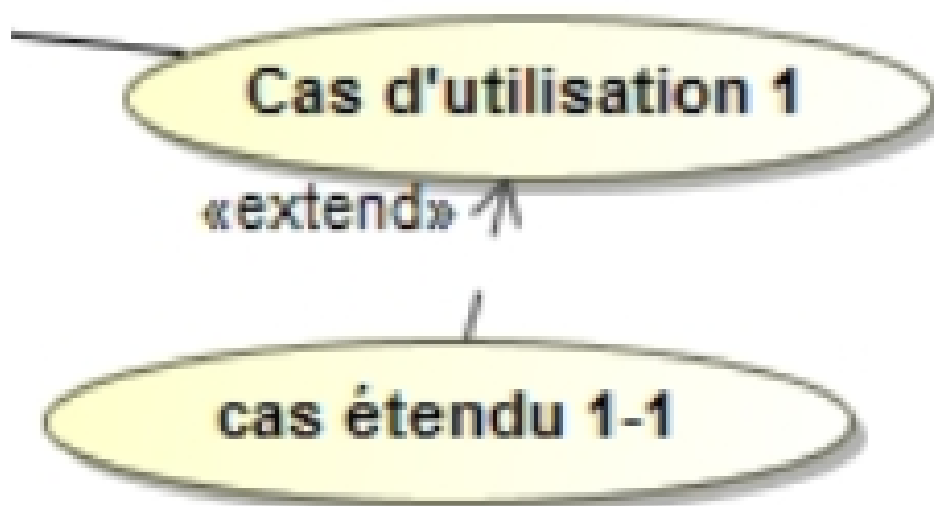


Relation d'inclusion :<<include>>

Il permet de faire ressortir une sous-fonctionnalité. Le cas d'utilisation de base inclut obligatoirement le service décrit par cette relation.

**Relation d'extension :<<extend>>**

Le cas d'utilisation de base incorpore un autre cas d'utilisation mais optionnel.

**6. Modélisation structurelle DEFINITION DE BLOC****Diagramme de définition de bloc (BDD)**

Il décrit la structure du système en montrant :

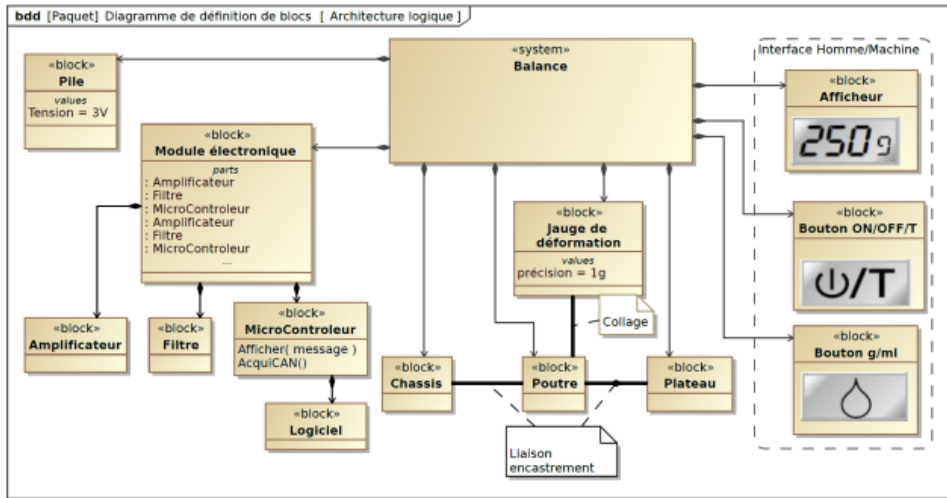
- les sous-ensembles (nommés blocs) qui le constituent
- les relations qui les lient par des liens de composition.
- en donnant une représentation statique

Il existe deux types de diagrammes de block :

- Le diagramme de bloc du système
- Le diagramme de bloc de contexte ? Ce diagramme non normalisé a été évoqué précédemment.

Le diagramme de bloc du système

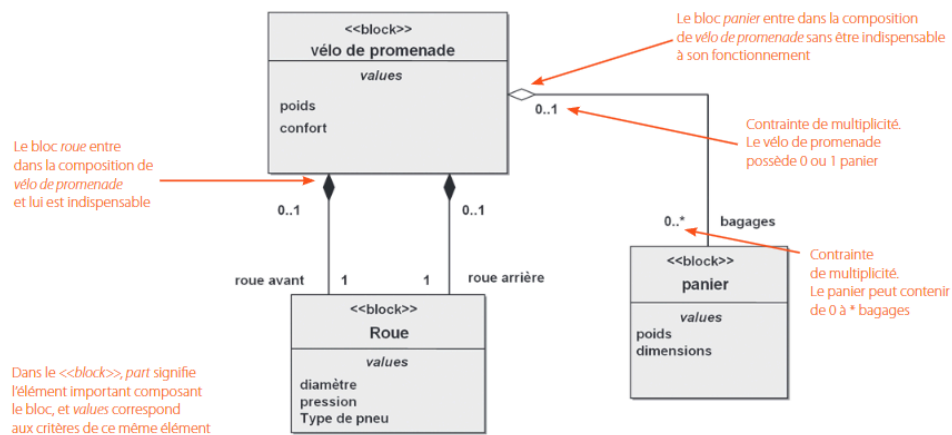
Le BDD du système définit l'architecture matérielle et logicielle de l'ensemble du système ou d'un sous-ensemble.



Le bloc

Le bloc contient différentes propriétés ,qui peuvent être :

- Propriétés de type « Valeurs » : caractéristiques quantifiables (couleur, marque, année de mise en service, etc.).



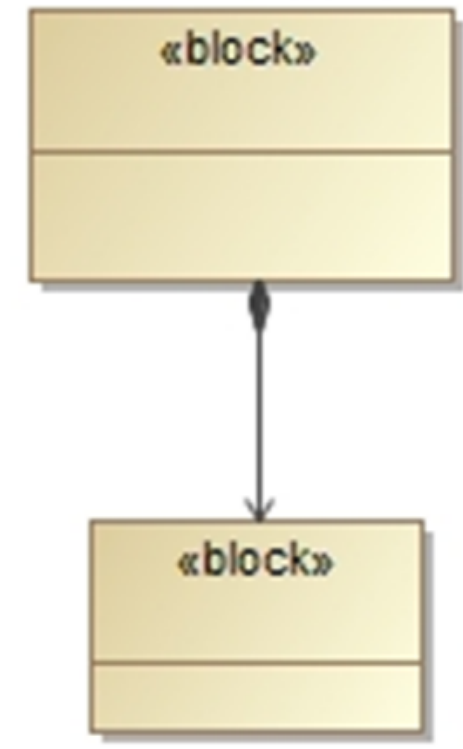
- Propriétés de type « Parties » : elles décrivent la décomposition du bloc en termes d'autres blocs.



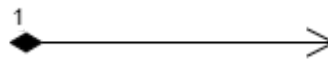
Relations

Relation de type **Composition** :

- Le bloc contenu fait partie du bloc principal. Un bloc représente le tout, et les autres ses parties

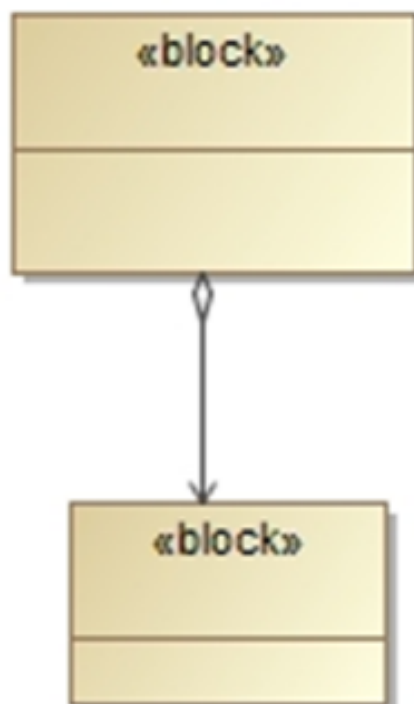


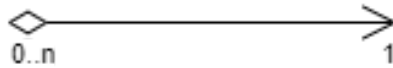
Lien d'agrégation forte précisant la quantité de blocs contenus.



Relation de type **Agrégation** :

- Le bloc ne fait pas partie du bloc principal. Pour fonctionner le bloc principal n'a pas forcément besoin du bloc contenu mais il peut en avoir besoin à un moment donné.

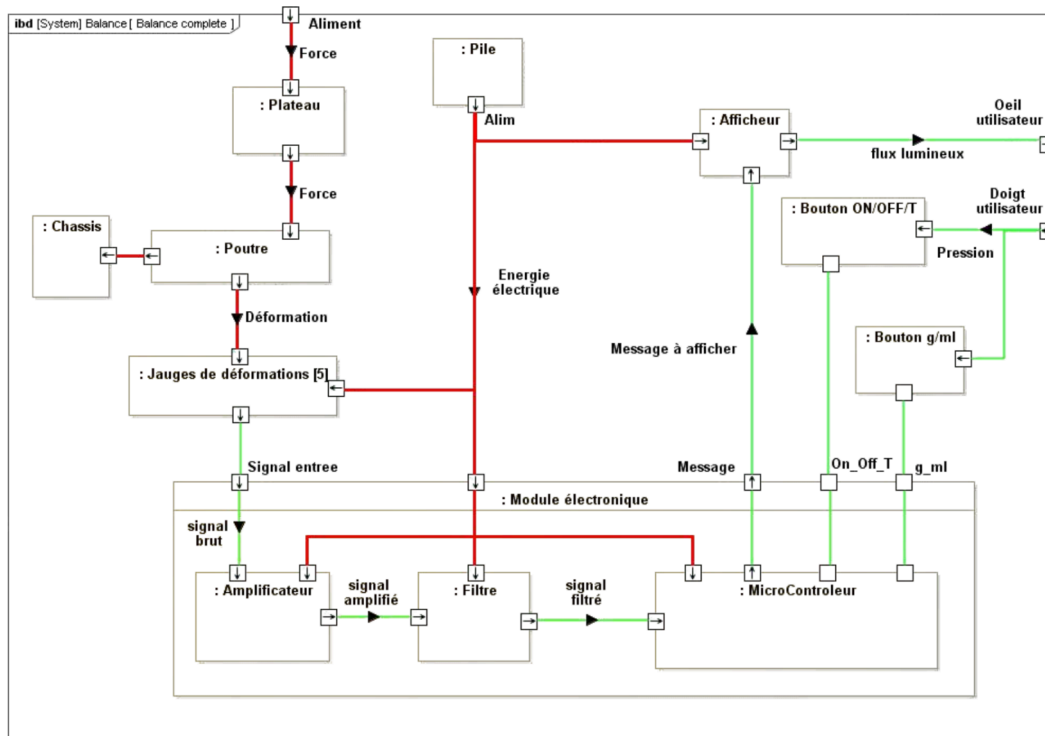




7. Modélisation structurelle DEFINITION DE BLOC INTERNE

Diagramme de définition de bloc interne (IBD)

- Ce diagramme (internal block diagram ou ibd) permet de décrire la structure interne d'un bloc extrait du BDD.
- Ce bloc extrait peut être le système, un sous-système, ou un bloc élémentaire.
- Il décrit les échanges de flux d'énergie, de matière ou d'information. Il est possible d'y adjoindre leur nature ; électrique, mécanique, pneumatique...



Ports des blocs internes

Chaque bloc interne possède un ou plusieurs ports, par lesquels vont transiter :

- les flux
- les consignes

Port de **flux** :

- Il permet de faire entrer ou sortir un flux du bloc.
- Il peut être unidirectionnel ou bidirectionnel.
- Les flux sont ici des flux physiques dont on peut nommer la nature



Port **standard** :

Un port standard permet de connecter le bloc à une consigne, un ordre de commande ou un service.



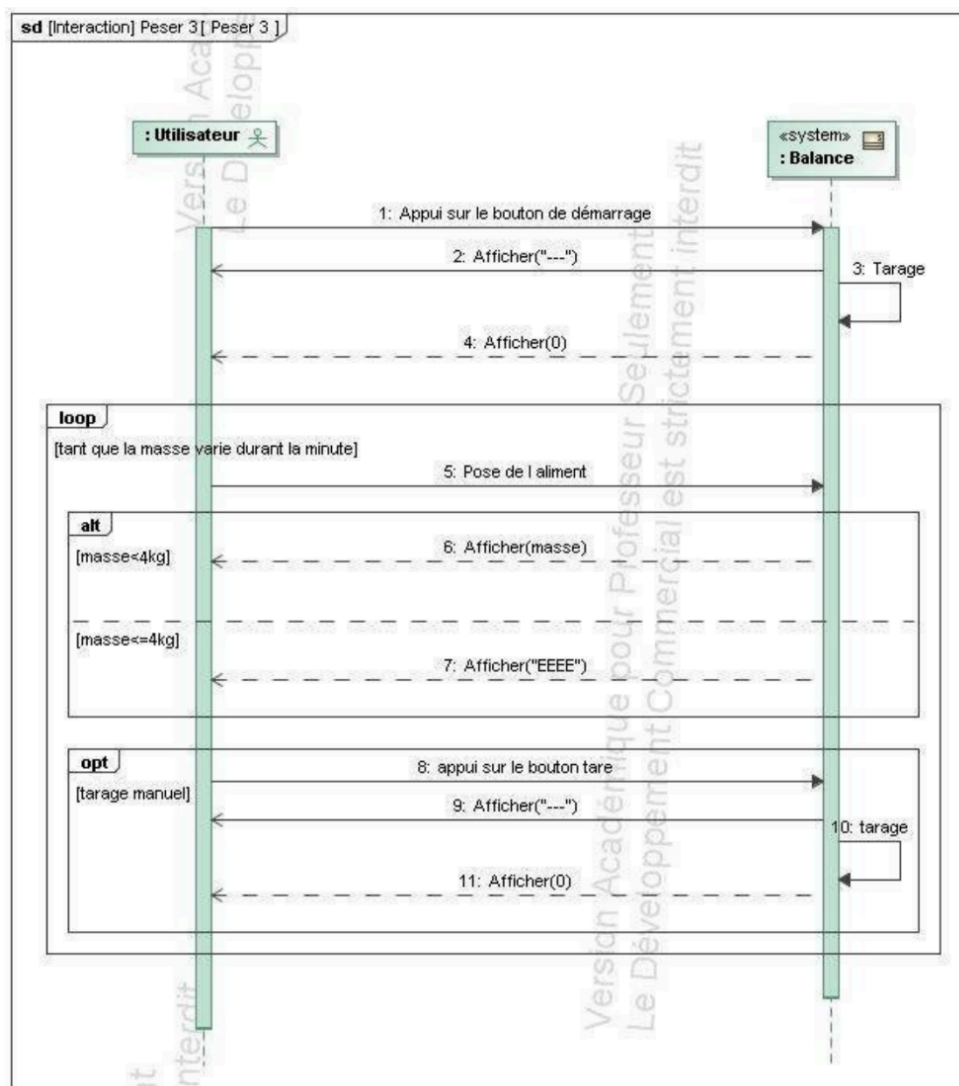
8. Modélisation comportementale SEQUENCE

Diagramme de séquence (SEQ)

Le diagramme de séquence est un diagramme comportemental qui décrit la chronologie des échanges issus d'un cas d'utilisation entre les acteurs et le système et/ou ses sous-systèmes.

Il complète le diagramme de cas d'utilisation en décrivant graphiquement les interactions à l'aide de messages.

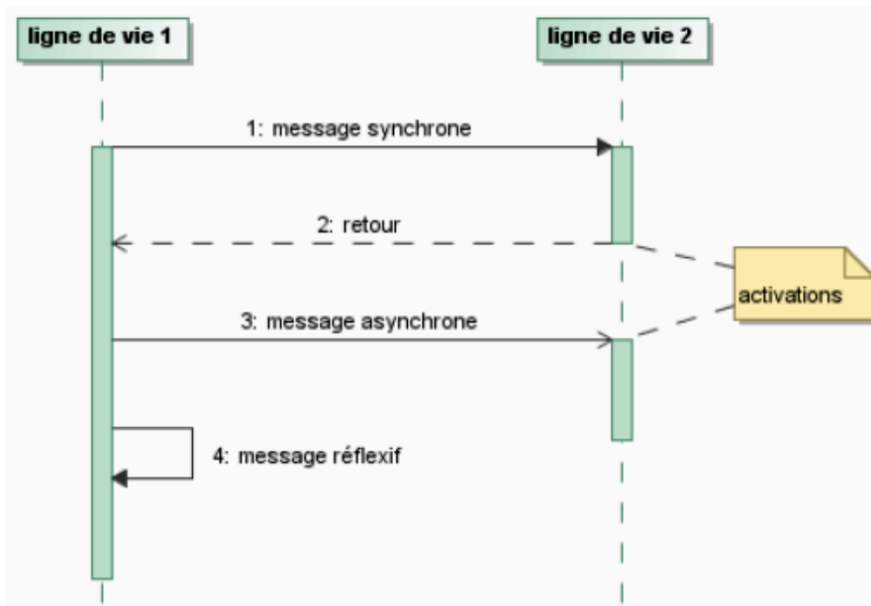
- Dans un premier temps, un diagramme de séquence «système» permet de décrire le comportement du système vu de l'extérieur.
- Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.
- On a un diagramme de séquence par cas d'utilisation



Lecture

Le diagramme de séquence se lit de haut en bas :

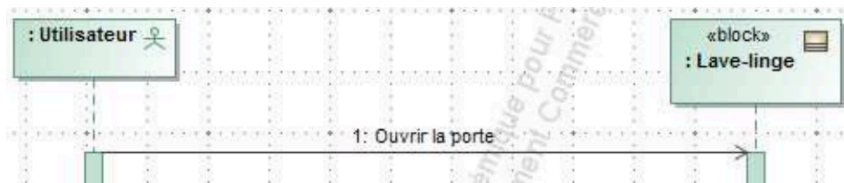
- Ligne de vie : ligne verticale en pointillée représentant un élément participant au diagramme de séquence. Elle possède un nom et un type.
- Message : é l é m e n t de communication unidirectionnel entre lignes de vie qui déclenche une activité dans le destinataire. La réception d'un message provoque un événement chez le récepteur.



Message asynchrone

L'émetteur envoie son message sans attendre de réponse en retour (:

- Symbole : flèche pleine terminée par deux traits.



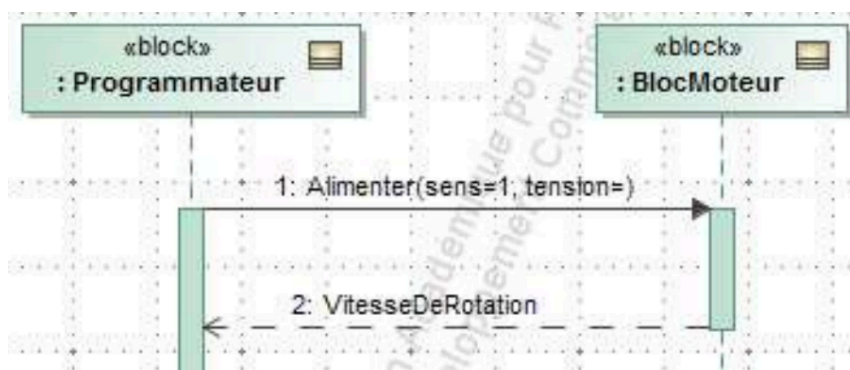
Message synchrone

L'émetteur du message est bloqué en attente de la réponse ; il ne fait rien entre l'émission et la réception.

- Symbole : flèche pleine et triangle noir plein

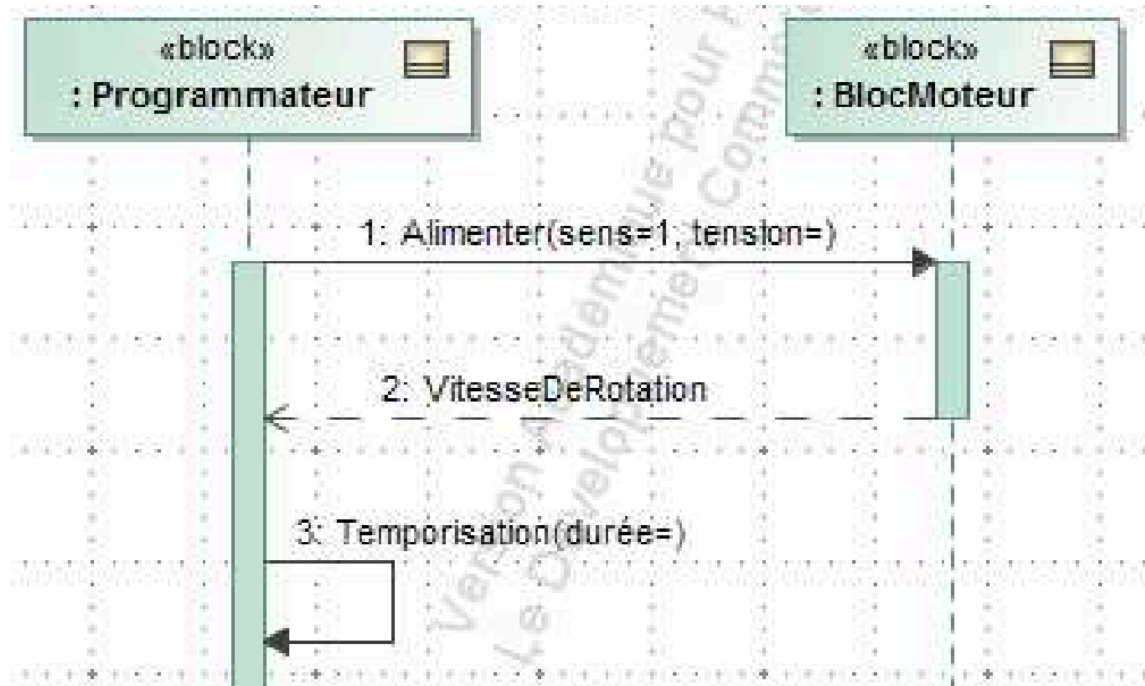
Il y a forcément un retour ; le message retour est le résultat direct du message précédent

- Symbole : flèche en pointillés



Message réflexif

Ce message représente un comportement interne

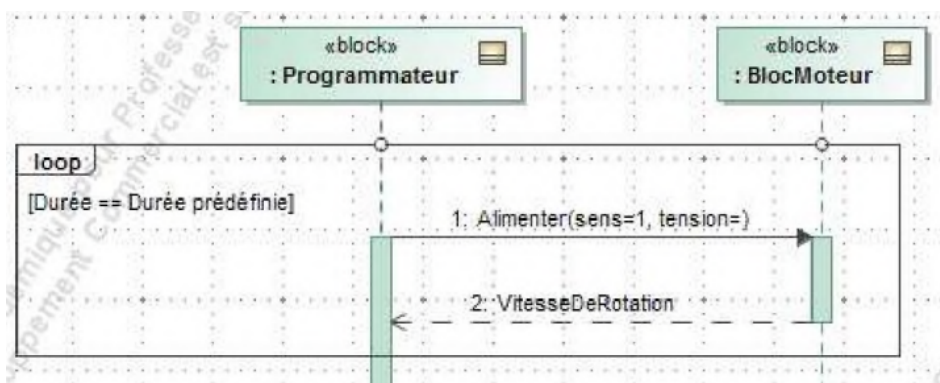


Fragments combinés

Un fragment combiné permet de faire apparaître un comportement différent en fonction de certaines conditions.

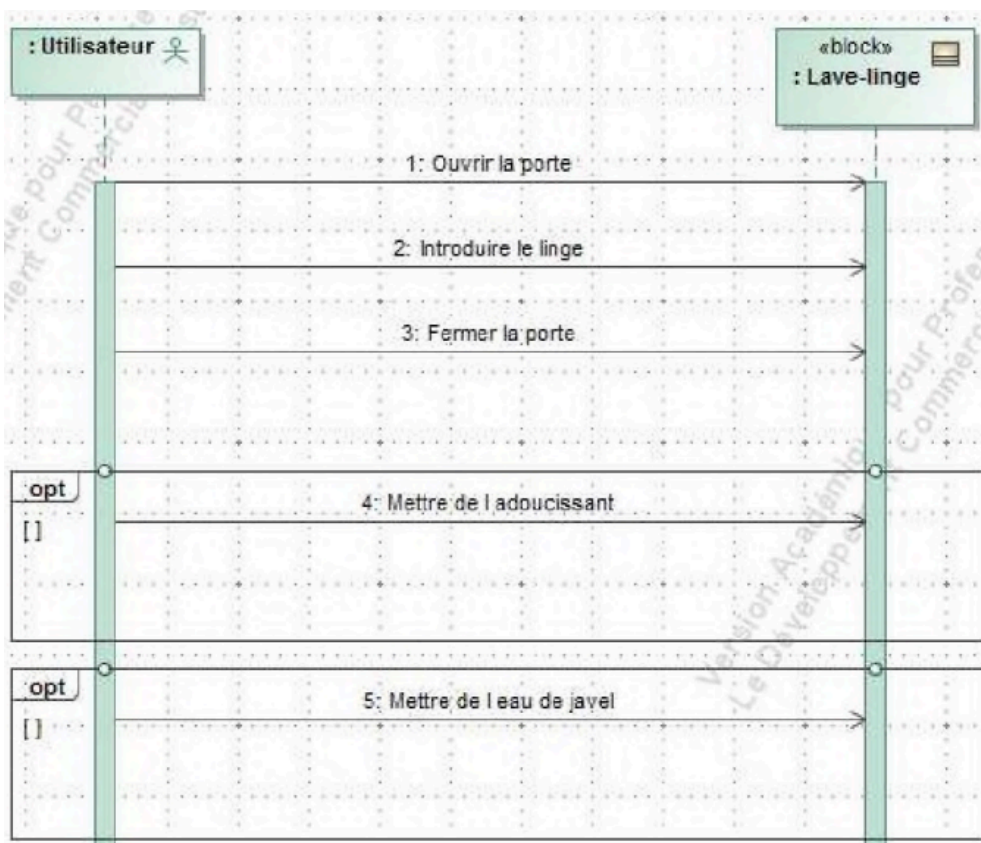
Boucle <<Loop>>

Les échanges contenus dans un fragment combiné de type «Boucle» sont réalisés tant qu'une condition appelée condition de garde est vraie.



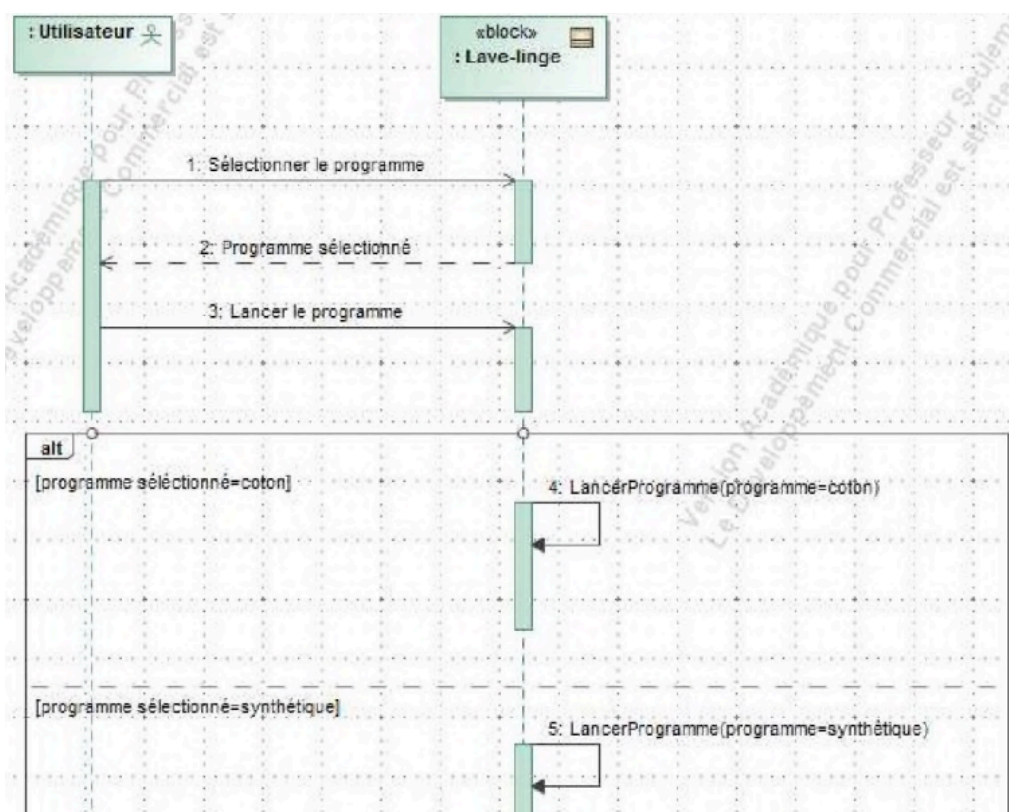
Boucle <<Option>>

Les échanges contenus dans un fragment combine de type «Option» ne sont réalisés que si la condition de garde est vraie.



Boucle <<Alt>>

En fonction de l'état de la condition de garde, l'une ou l'autre des possibilités contenues dans le fragment sera réalisée.



9. Modélisation comportementale ETAT

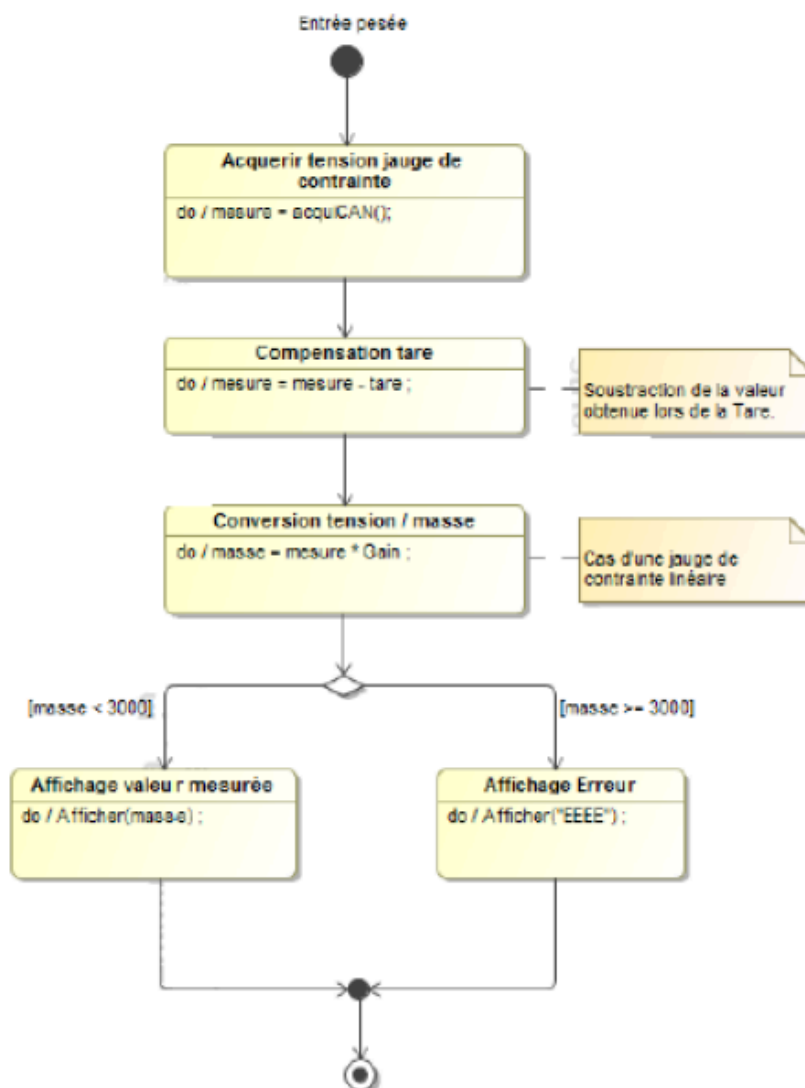
Diagramme d'état (STM)

On appellera «état» une situation d'une durée finie pendant laquelle le système :

- exécute une activité particulière (constitué de plusieurs actions)
- satisfait à une certaine condition ou est en attente d'un événement.

Le passage d'un état à un autre se fait en franchissant une transition.

Le diagramme d'état décrit les états successifs d'un système ou d'un sous-système qui occupent des états bien définis au cours de leur fonctionnement (machine d'état). C'est notamment le cas du fonctionnement séquentiel.



Les états

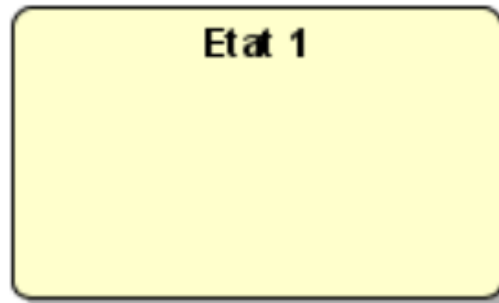
Etat initial : il désigne le point de départ de la séquence qui peut correspondre à la mise en énergie



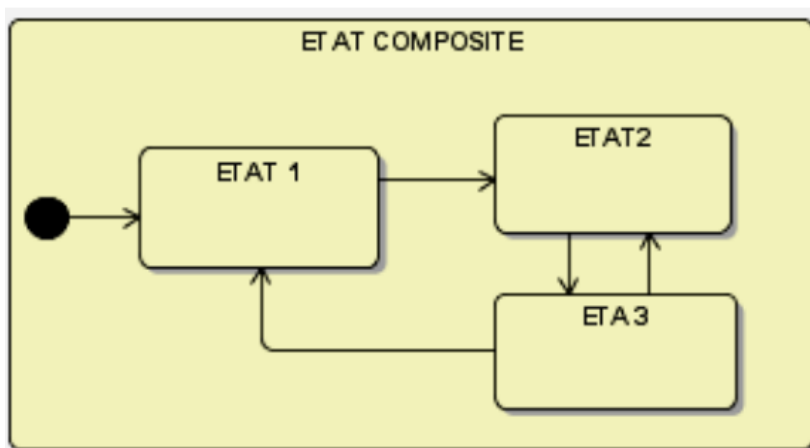
Etat final : il représente la fin du fonctionnement du système qui peut correspondre à la mise hors énergie. Plus aucun état n'est actif



Etat : un état représente une situation pendant laquelle: il satisfait une certaine condition, il exécute une certaine activité, il attend un certain événement.



Etat composite : aussi appelé super-état; il permet d'englober plusieurs sous-états. Il comporte un état initial.



Transition

Elle représente l'évolution du système d'un état à un autre selon une condition de déclenchement.

Condition

Fourche

Active simultanément 2 états à partir d'un état antérieur. Il existe également l'union qui active 1 état à partir de 2 états précédents.



La description d'un état peut inclure des événements internes, afin de montrer la réponse de l'état à l'occurrence de ces événements. Les événements entry, do et exit indiquent ce qu'il se passe :

- à l'entrée dans l'état (mot clé entry),
- pendant l'état (do),
- à la sortie de l'état (exit).