



ANALYSE PAR METHODE SYSML

Objectifs et Compétences

- Analyser le besoin et appréhender les problématiques
- Définir les frontières de l'analyse
- Appréhender les analyses fonctionnelle, structurelle et comportementale

Savoirs

Je connais:

- Analyse fonctionnelle, structurelle et comportementale

Savoir Faire

Je sais faire:

- Décrire le besoin et présenter la fonction globale
- Identifier les architectures fonctionnelles et structurelles
- Identifier la nature des flux échangés et leurs caractéristiques
- Identifier les fonctions techniques et les constituants associés
- Définir les limites et les contraintes choisies ou imposées

Sommaire

I. DEFINITIONS	2
I.1. ORGANISATION.....	2
I.2. DEFINITIONS.....	2
II. DIAGRAMME DES CAS D'UTILISATION (UC).....	4
III. DIAGRAMME DES EXIGENCES (REQ).....	6
IV. DIAGRAMME DE SEQUENCE (SD).....	8
V. DIAGRAMME D'ETATS (STM).....	9
VI. DIAGRAMMES DE DEFINITIONS DE BLOCS (BDD).....	11
VII. DIAGRAMMES DE BLOCS INTERNES (IBD).....	12
VIII. EXEMPLE D'UN RADIO REVEIL.....	14

I. DEFINITIONS

I.1. Organisation

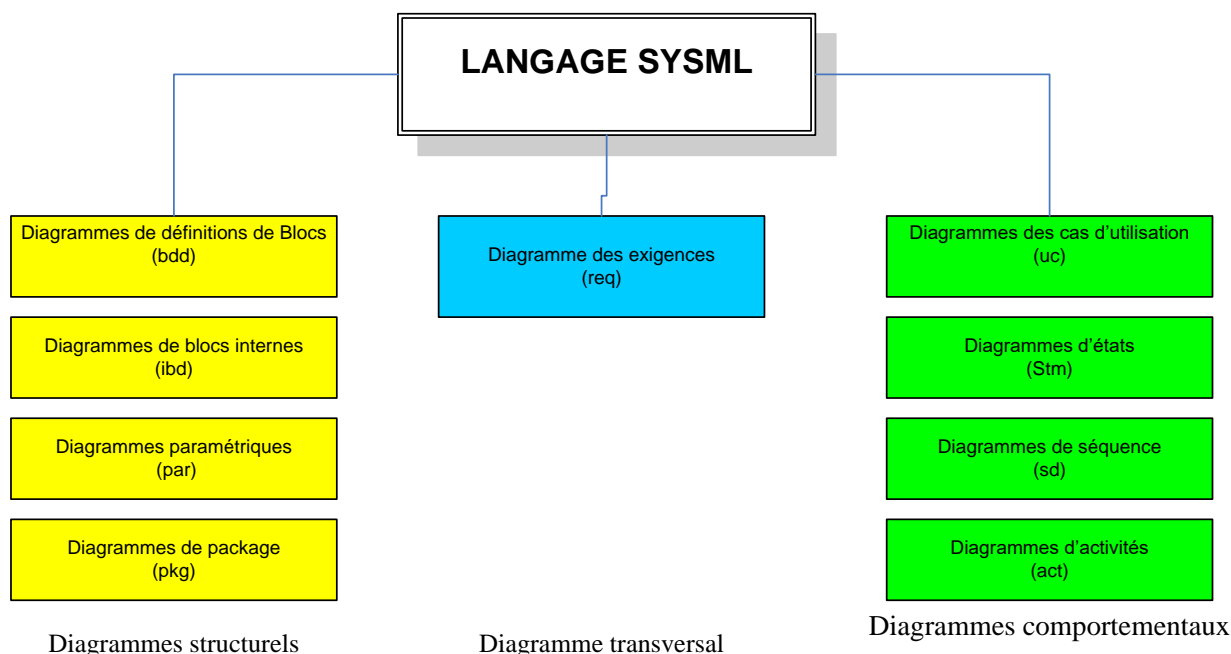
L'ingénierie des systèmes est une approche scientifique interdisciplinaire de formation récente, dont le but est de formaliser et d'appréhender la conception de systèmes complexes avec succès.

SYSML est l'acronyme de **Systems Modeling Language** (langage de modélisation des systèmes) ; c'est un langage dont le but est de décrire tout ou partie d'un système technique, d'un point de vue transversal, comportemental ou structurel.



Système : Ensemble de constituants inter-reliés qui interagissent les uns avec les autres d'une manière organisée pour accomplir une finalité commune.

Le langage **SysML** s'articule autour de **9 types de diagrammes**, divisés en diagrammes de structure (**approche structurelle**), diagramme transversal (**approche fonctionnelle**) et diagrammes comportementaux (**approche comportementale**).



I.2. Définitions

Le diagramme transversal (approche fonctionnelle) est le **diagramme d'exigences (req)** qui montre les exigences du système et les relations entre elles. On y retrouve des fonctions et des valeurs disponibles dans un cahier des charges fonctionnel.

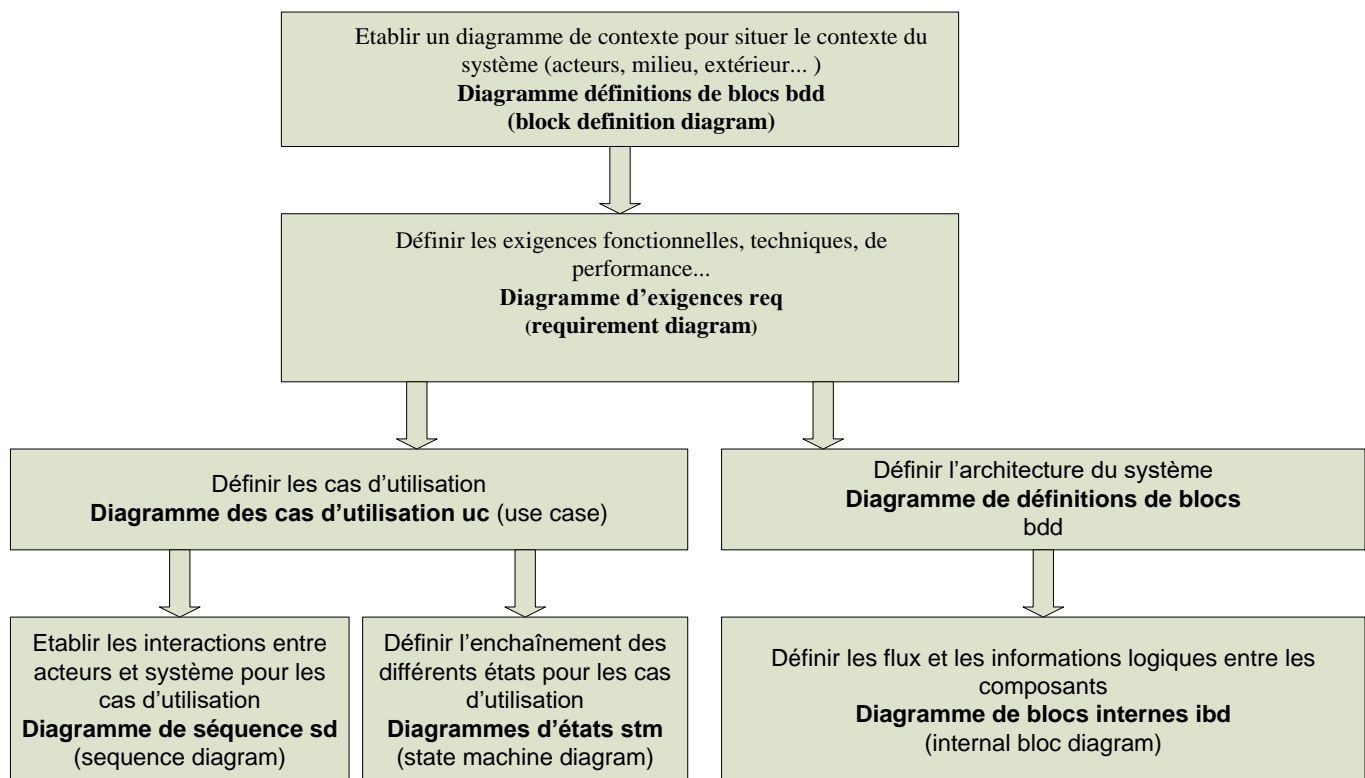
Les diagrammes de structure sont :

- le **diagramme de définition de blocs (bdd)** qui décrit le système en le découpant en sous-parties, sous-sous-parties... Il définit aussi les grandeurs associées.
- le **diagramme de bloc interne (ibd)** qui permet de décrire les liaisons entre les différents blocs (électriques, mécaniques,...)
- le **diagramme paramétrique (par)** qui permet de faire le lien "mathématique" entre les grandeurs introduites dans le bdd.
- le **diagramme de package (pkg)** qui fait le lien entre tout mais non étudié en ATS.

Les diagrammes de comportement sont :

- le **diagramme des cas d'utilisation (uc)** qui permet d'énumérer l'ensemble des fonctions et des contraintes que doit respecter le système.
- le **diagramme d'état (stm)** qui montre les étapes par lesquelles le système va passer.
- Le **diagramme de séquences (sd)** qui décrit le scénario des cas d'utilisation de manière chronologique.
- Le **diagramme d'activités (act)** qui montre comment évoluent et comment sont traités les flux au travers du système.

Quand on doit décrire un système, on peut par exemple suivre la méthode suivante, mais toute approche reste valable comme par exemple de commencer par les cas d'utilisation pour bien définir les frontières de l'étude...

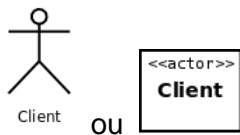
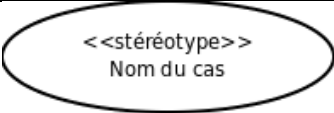

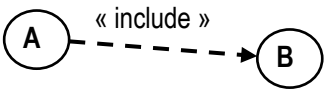
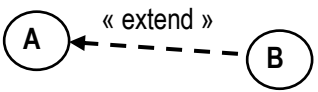
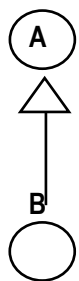


En ATS, vous devez être capable de lire des schémas SYSML, éventuellement de les compléter de manière simple.

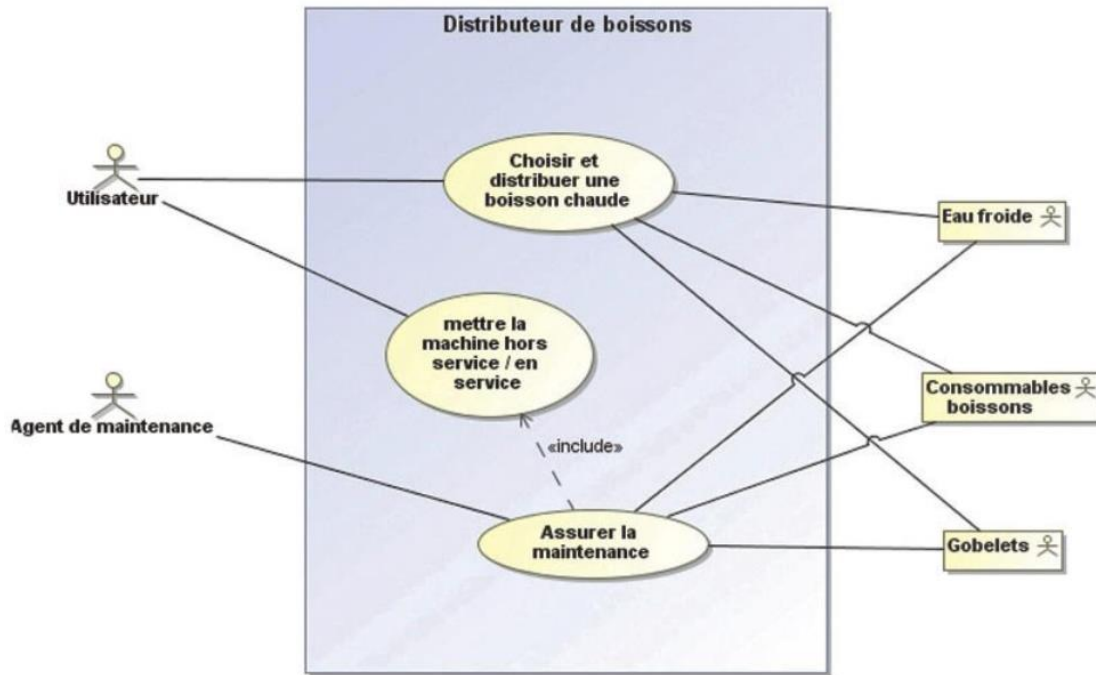
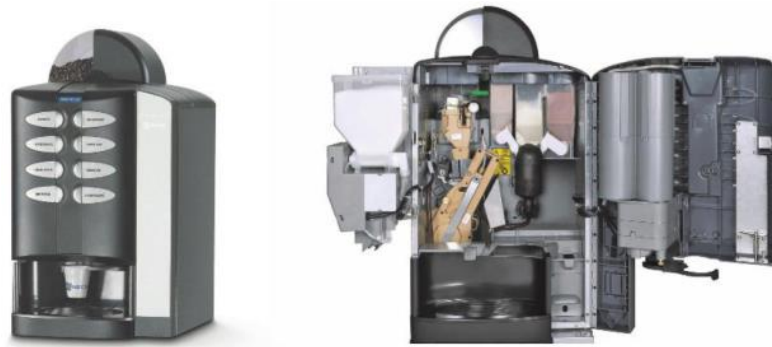
II. DIAGRAMME DES CAS D'UTILISATION (uc)

C'est la première étape d'analyse d'un système. Il permet d'exprimer le besoin d'un utilisateur. Le diagramme des cas d'utilisation a pour but de délimiter le système et d'identifier les possibilités d'interactions entre le système et les acteurs (intervenants extérieurs au système) c'est-à-dire toutes les fonctionnalités que doit fournir le système.

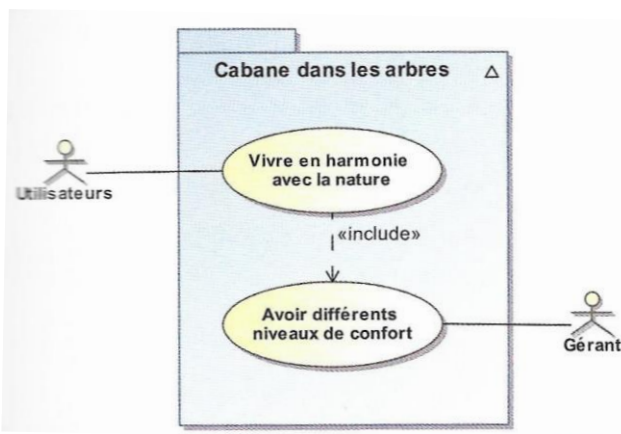
Explication des différents pictogrammes employés sur ce diagramme :

Désignation	Représentation	Définition
Acteur principal		<p>Un acteur est <i>principal</i> pour un cas d'utilisation lorsque ce cas rend service à cet acteur.</p> <p>Les autres acteurs sont alors <i>secondaires</i>.</p>
Acteur secondaire	Bonhomme ou classeur.	Un cas d'utilisation a au plus un acteur principal.
Cas d'utilisation	 <p>Ellipse contenant le nom du cas (verbe à l'infinitif), et optionnellement un stéréotype.</p>	<p>Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.</p>
Association		Une relation d'association est un chemin de communication entre un acteur et un cas d'utilisation.
Relation d'inclusion		Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend de B.
Relation d'extension		Un cas d'utilisation B étend un cas d'utilisation A lorsque le cas d'utilisation B peut être appelé au cours de l'exécution du cas d'utilisation A.
Relation de généralisation	 <p>Flèche dont la pointe est un triangle fermé désignant le cas le plus général.</p>	Un cas A est une généralisation d'un cas B si B est un cas particulier de A.
Stéréotype	« extend »	Un stéréotype est une annotation s'appliquant sur un élément.

Exemple Cas d'utilisation du distributeur de boisson chaude COLIBRI



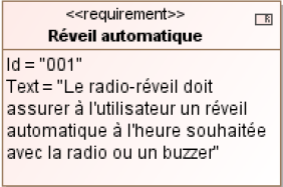


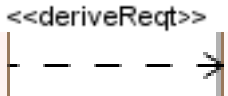
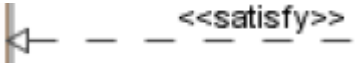
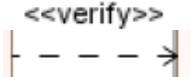
Exemple 2 : Cas d'utilisation pour une cabane dans les arbres:



III. DIAGRAMME DES EXIGENCES (req)

Le diagramme des exigences (« Requirement », **req** sur les diagrammes) permet de répertorier et d’analyser les contraintes et les performances du système. Il permet de structurer les besoins.

Ce diagramme est un outil de **représentation des fonctionnalités du système**. C’est un moyen de communication entre les concepteurs et les clients du système.

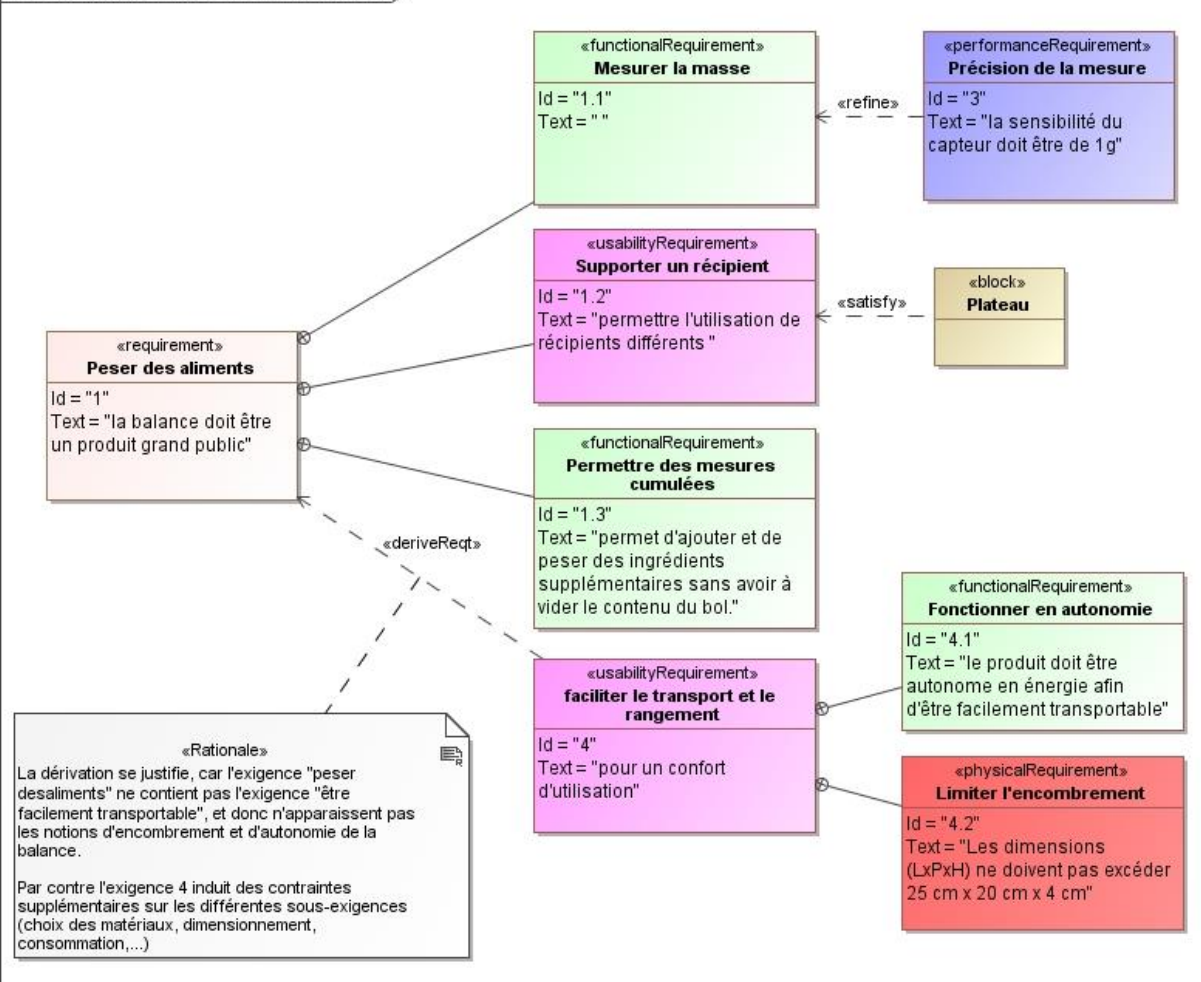
Représentation graphique	Nom / Rôle
	Exigence : Un numéro Un texte explicatif
	la contenance permet de décomposer une exigence composite en plusieurs exigences unitaires
Entre 2 exigences : 	le raffinement (« refine ») consiste en l’ajout de précisions, par exemple de données quantitatives. Entre exigence – élément comportemental (cas d’utilisation, diagramme d’états, etc.) : permet de mettre en œuvre l’exigence.
	la dérivation (« deriveReq ») consiste à relier des exigences de niveaux différents
	<ul style="list-style-type: none"> exigence – bloc d’architecture : « satisfy » ; satisfaire l’exigence.
	<ul style="list-style-type: none"> exigence – cas de test : « verify » permet de vérifier une exigence

Exemple : La balance HALO (Terrillon)

Caractéristiques :

- Précision de 1 g ;
- Charge maximale = 3 kg ;
- Dimensions : 24 x 19,5 x 3,5 cm
- Interface Homme/Machine :
 - Bouton Marche/Arrêt/Tare
 - Bouton Conversion g/ml
 - Afficheur LCD 4 digits
- Design épuré aux formes circulaires ;
- Gamme de 5 coloris tendances ;
- Conforme aux directives DEEE et RoHS.

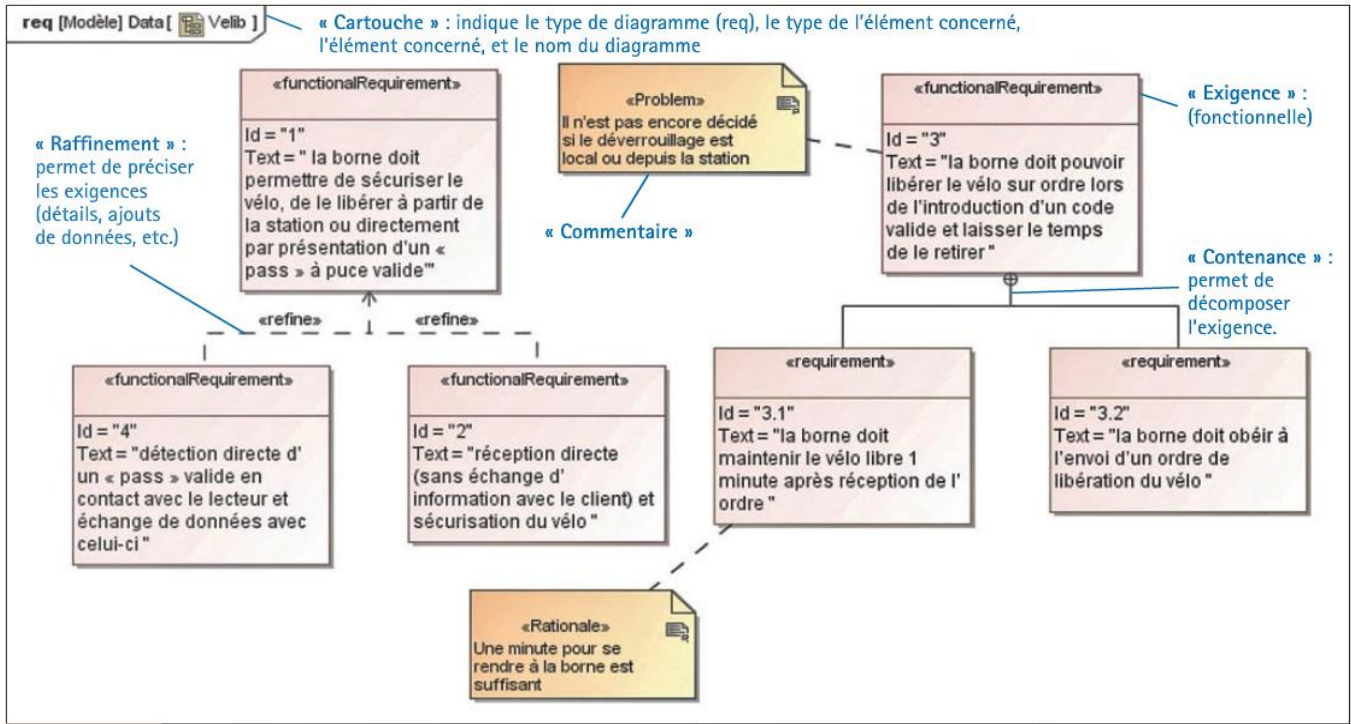




Exemple : le Vélib



Pour pouvoir utiliser un vélo, il faut dans un premier temps s'abonner (minimum 24h) : pour cela, il faut, par exemple, se rendre à une station qui permet de régler le montant et une caution (carte bancaire). On a ensuite le choix de retirer le vélo grâce au code porté sur la fiche de reçu, ou par l'usage d'un « pass » de transport en commun. Ensuite, le vélo peut être retiré à une borne. Pour restituer le vélo, il suffit de le présenter dans le verrou de n'importe quelle borne disponible dans la commune.

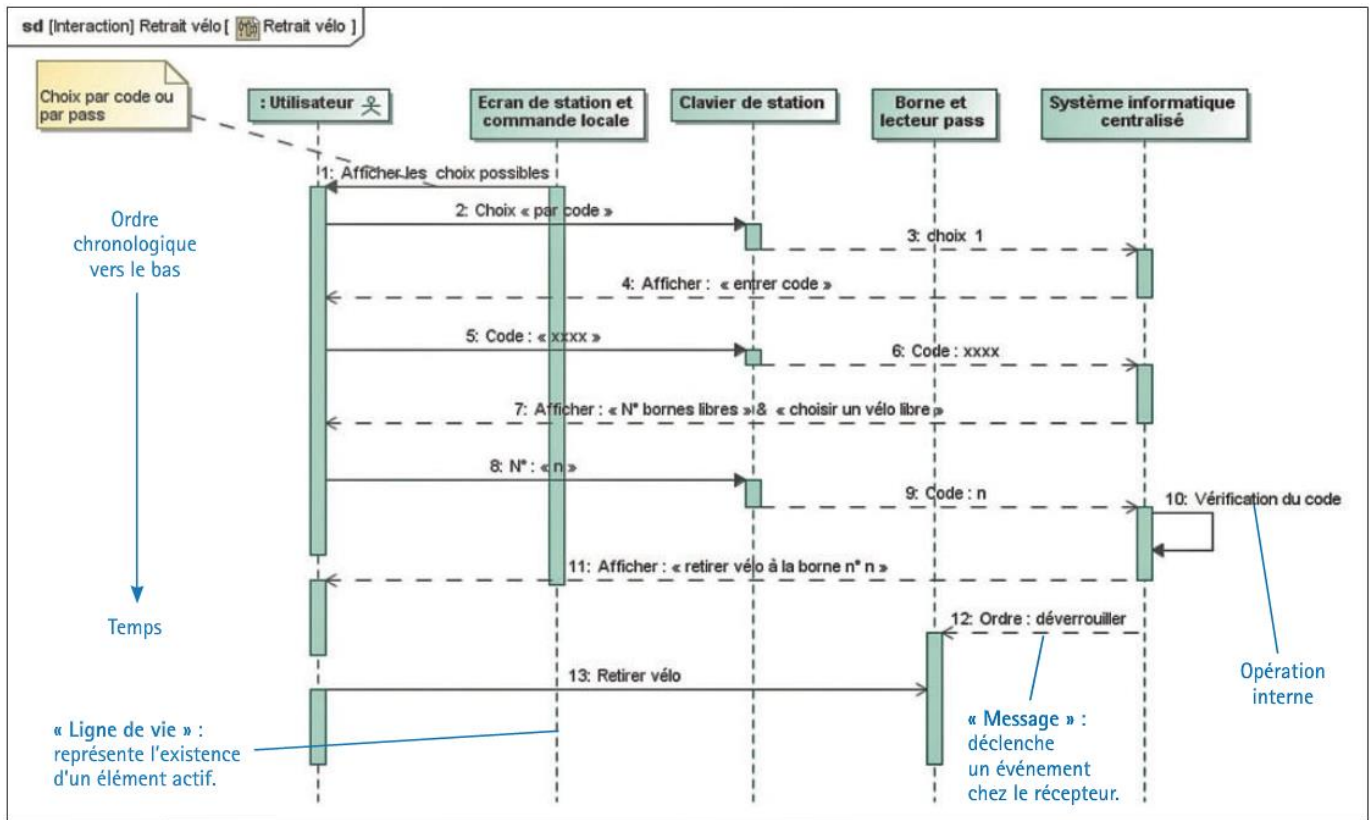


IV. DIAGRAMME DE SEQUENCE (sd)

C'est un diagramme comportemental. Il décrit le scénario des interactions dans le temps entre les acteurs et les objets. Il montre la chronologie des échanges issus d'un cas d'utilisation.

Représentation graphique		Nom / Rôle
		<p>Ligne de vie système ou acteur</p> <p>Ligne de vie d'un bloc</p> <p>Message synchrone + réponse</p> <p>Message asynchrone</p> <p>Message réflexif</p> <p>Activation : travail du système du bloc.</p>
<p>La ligne de vie : Elle se représente par un rectangle, auquel est accrochée une ligne verticale en trait interrompu.</p> <p>Début d'exécution</p> <p>Fin d'exécution</p>		
Message	→	Message envoyé
	← - - - -	Message de réponse

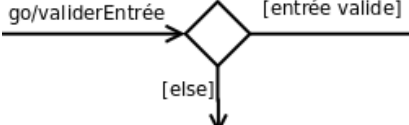


Exemple : « Vélib » (cas d'utilisation « Retirer un vélo par code »)



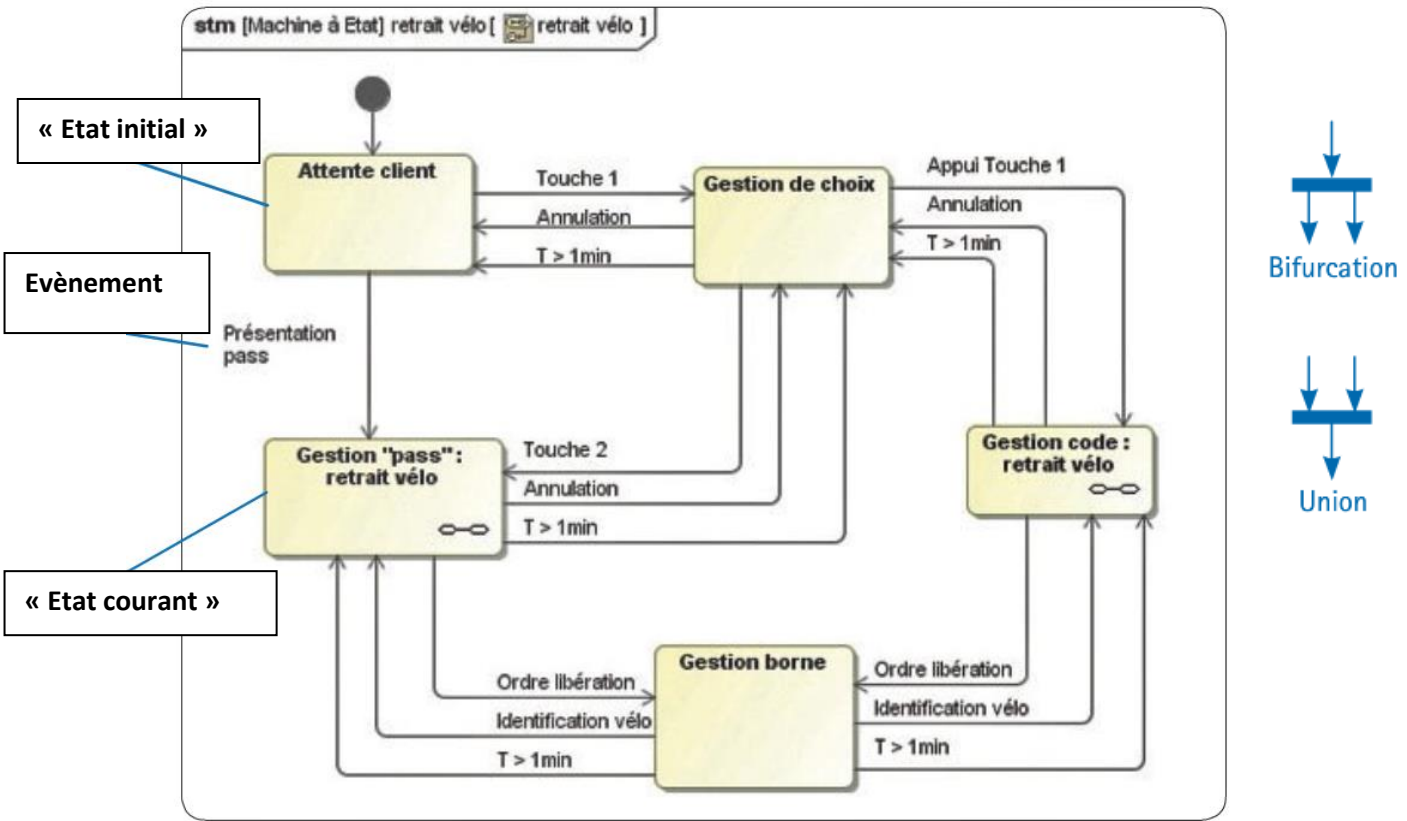
V. DIAGRAMME D'ETATS (stm)

Le diagramme d'états décrit les états successifs d'un « objet » (système, sous-système, composants, etc...) en réaction à des « évènements », les transitions. **Il permet de montrer les évènements qui provoquent un changement.**

Représentation graphique	Nom	Définition
	Etat initial	Désigne le point de départ de la séquence, il peut correspondre à la mise en énergie.
	Etat final.	Représente la fin du fonctionnement du système.
	Etat	représente une situation durant la vie d'un bloc pendant laquelle : • il satisfait une certaine condition ; il exécute une certaine activité ;
	Transitions externes :	l'évènement arrive pour que la transition soit déclenchée. Evènement : Spécification d'une occurrence qui peut déclencher une réaction sur un élément : call, change, after et when.
	Transition interne :	_ entry : permet de spécifier une activité qui s'accomplit quand on entre dans l'état. _ exit : permet de spécifier une activité qui s'accomplit quand on sort de l'état. _ do : Une activité do commence dès que l'activité entry est terminée _ include : permet d'invoquer un sous-diagramme d'états-transitions

	<p>Point de décision</p>	<p>Quand il y a un choix</p>
	<p>Un état composite</p>	<p>(aussi appelé super-état) permet d'englober plusieurs sous-états exclusifs</p>
		<p>L'activation du pseudo-état History permet à un super-état de se souvenir du dernier sous-état séquentiel qui était actif avant une transition sortante</p>

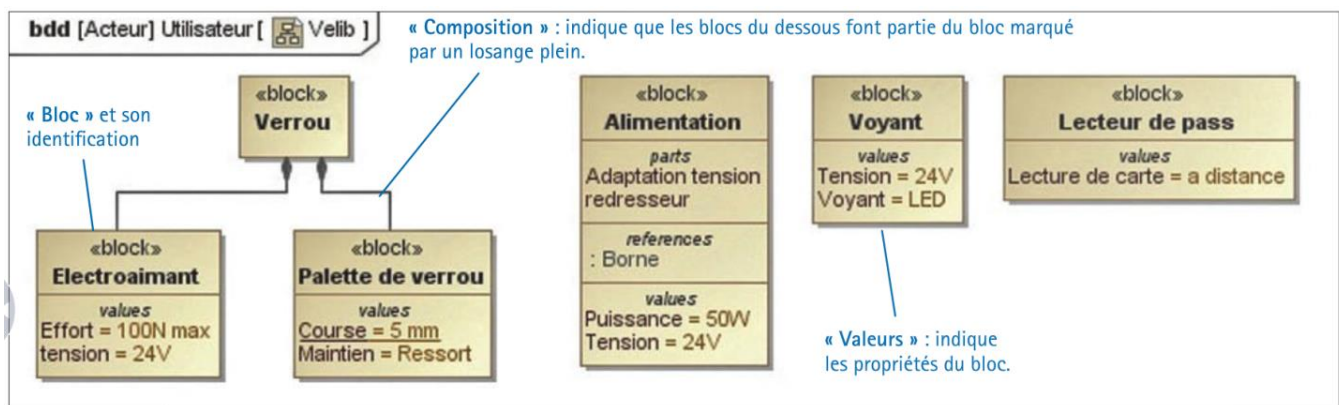
exemple : Diagramme d'états du système « vélib » (état composite « retrait d'un vélo »)



VI. DIAGRAMMES DE DEFINITIONS DE BLOCS (bdd)

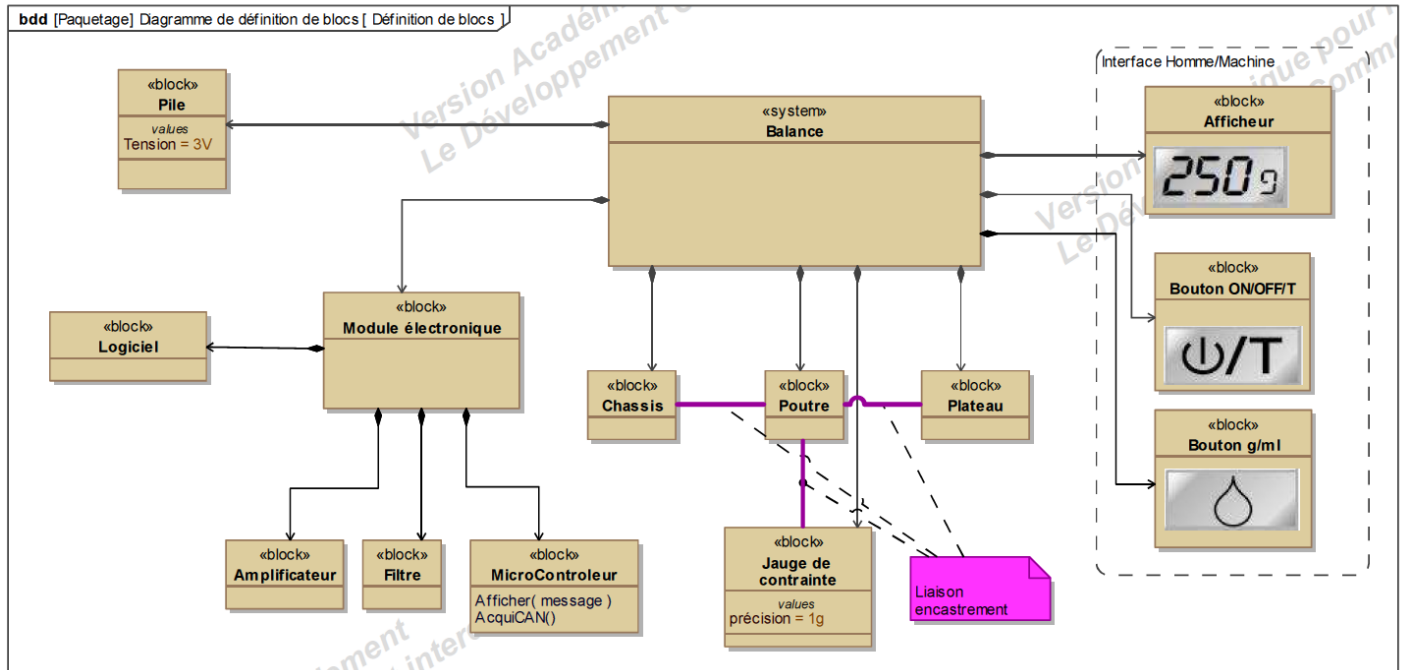
C'est un diagramme **structurel**. Il définit les éléments de structure d'un système, d'un sous-système ou d'un constituant et leurs propriétés. Le diagramme de blocs nous permet de prendre connaissance de la structure interne du système. Il permet de répondre à la question « qui contient quoi ? ». Le bdd est similaire à la première page d'une notice de montage d'un meuble, indiquant la liste des éléments et des pièces à assembler avec leurs quantités respectives.

Représentation graphique	Nom / Rôle
	<p>Blocs : Il constitue la brique de base pour la modélisation de la structure</p>
	<p>Relation entre 2 blocs</p>
	<p>Relation de composition (losange plein)</p>
	<p>Relation d'agrégation (losange vide)</p>
	<p>Relation d'héritage (généralisation)</p>



Exemple : Modélisation structurelle par « blocs » de la borne « Vélib ».

Description SysML de la balance HALO

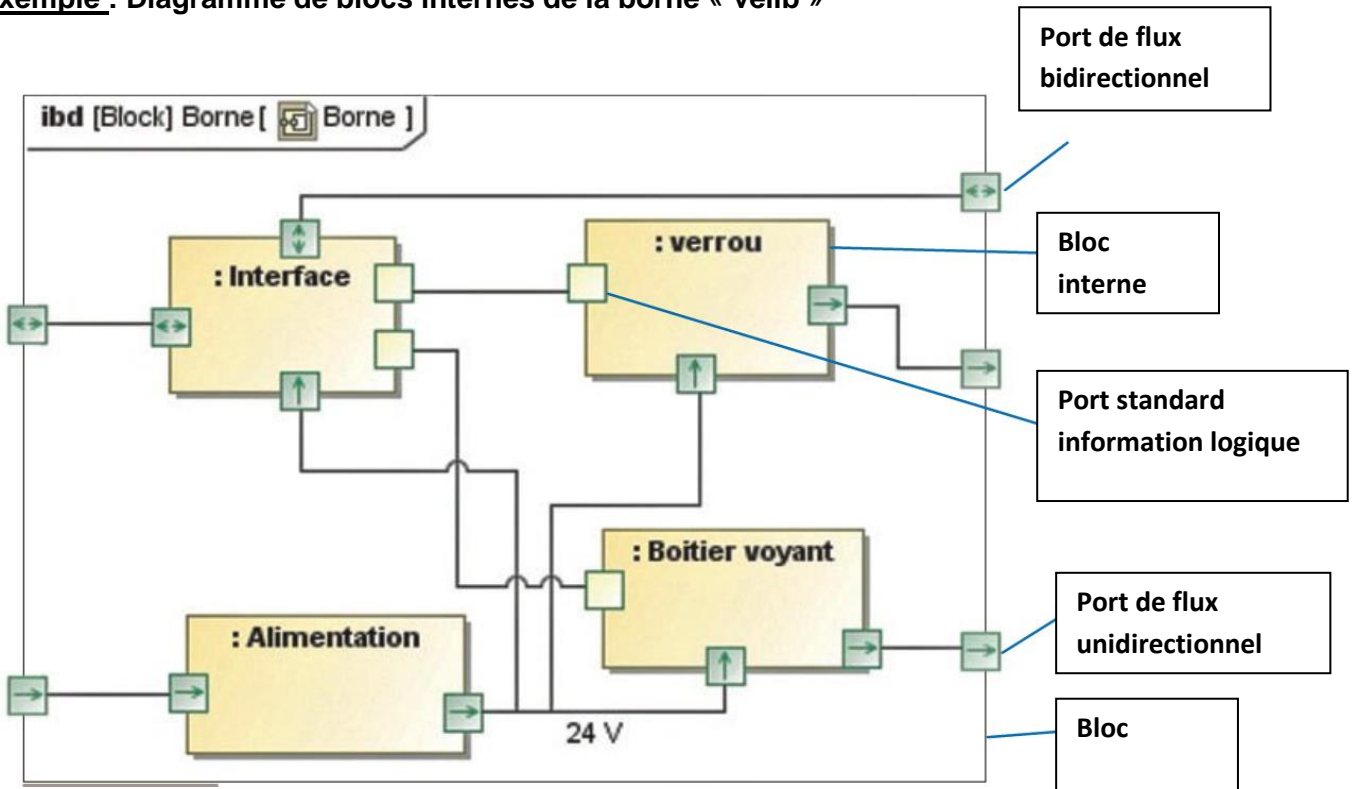


VII. DIAGRAMMES DE BLOCS INTERNES (ibd)

C'est un diagramme **structurel**. Il décrit l'architecture matérielle d'un système, les échanges internes entre ses éléments ou avec l'extérieur. Les blocs peuvent décrire un système complet, un sous-système ou un composant élémentaire. Le diagramme ibd permet de représenter les connexions entre blocs mais aussi (et surtout !) les échanges de matière, d'énergie et d'information avec la notion de ports.

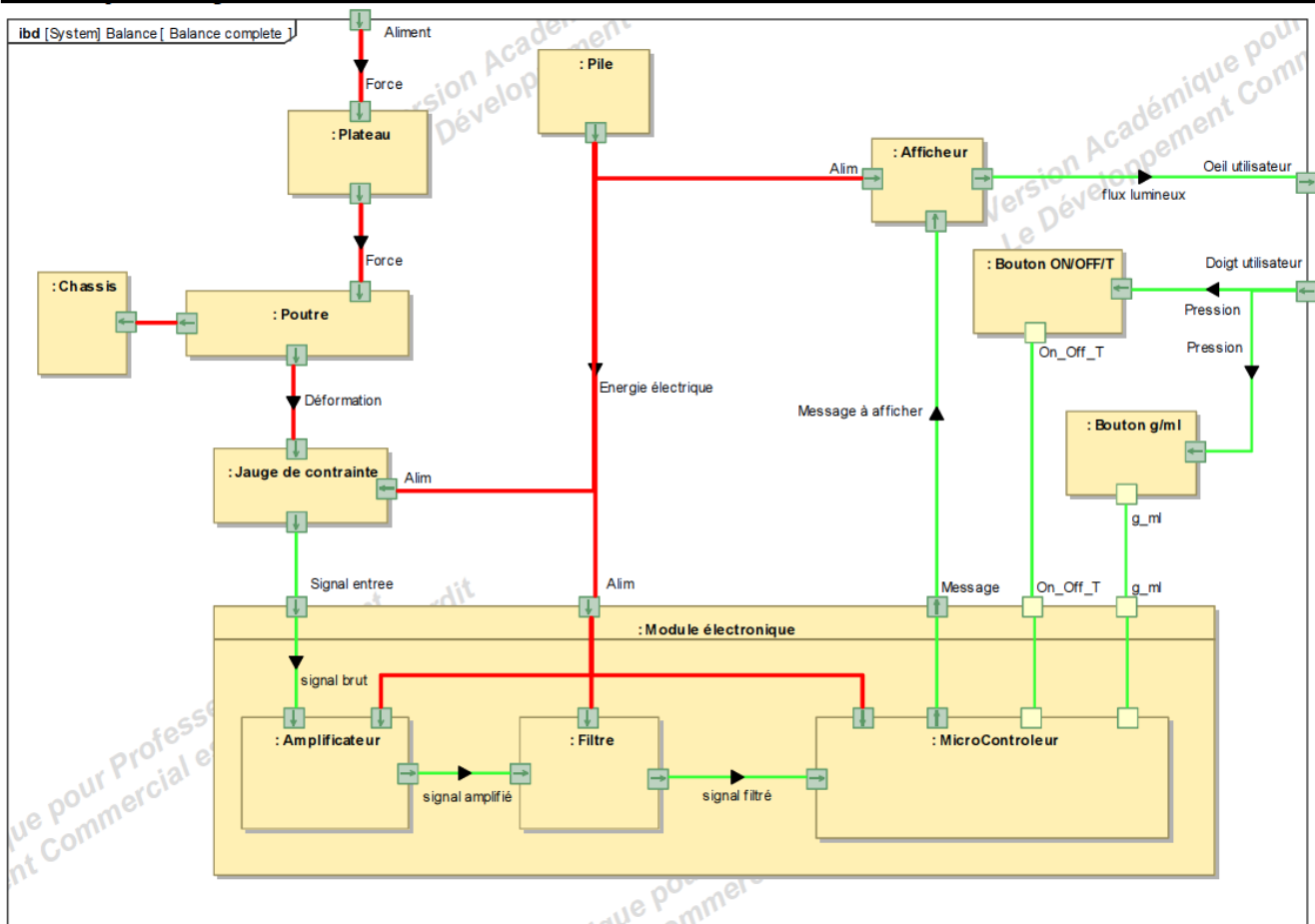
Représentation graphique	Nom / Rôle
	Bloc : c'est le constituant
	Relation entre bloc
	Port standard : Ils expriment les échanges logiques entre blocs. Ils ne peuvent pas être connectés aux ports de flux.
	Port de flux (flow port) : Ils expriment la circulation de flux physiques entre blocs (énergies, fluides, données....) en entrée ou en sortie.

Exemple : Diagramme de blocs internes de la borne « vélib »



Un diagramme de bloc interne représente l'intérieur d'un bloc issu du diagramme de blocs (pas interne celui-là). On voit ainsi apparaître les échanges entre blocs de même niveau (blocs qui ne se contiennent pas entre eux) comme ci-dessous :

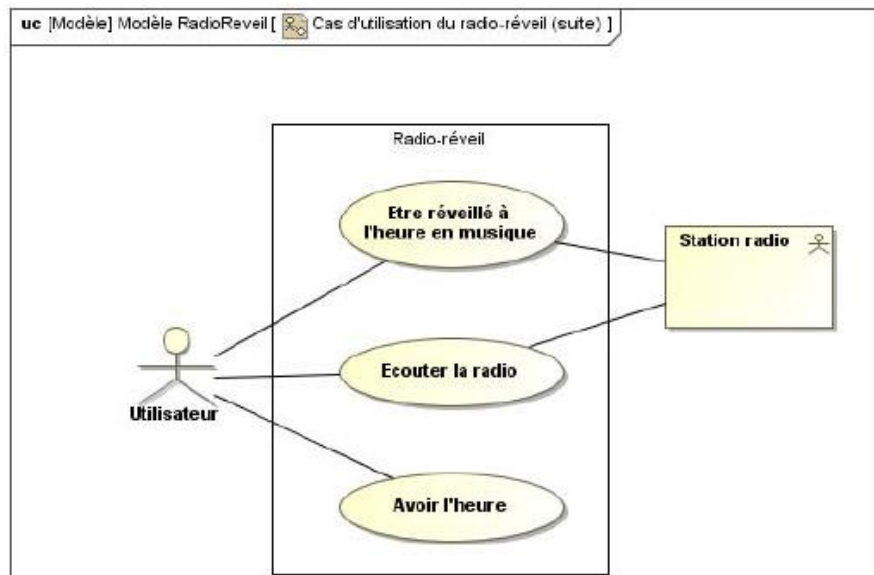
Description SysML de la balance HALO



VIII. EXEMPLE D'UN RADIO REVEIL

Dans l'exemple du radio-réveil, la première exigence fondamentale concerne la capacité à assurer à l'utilisateur un réveil automatique à l'heure souhaitée avec la radio ou un buzzer.

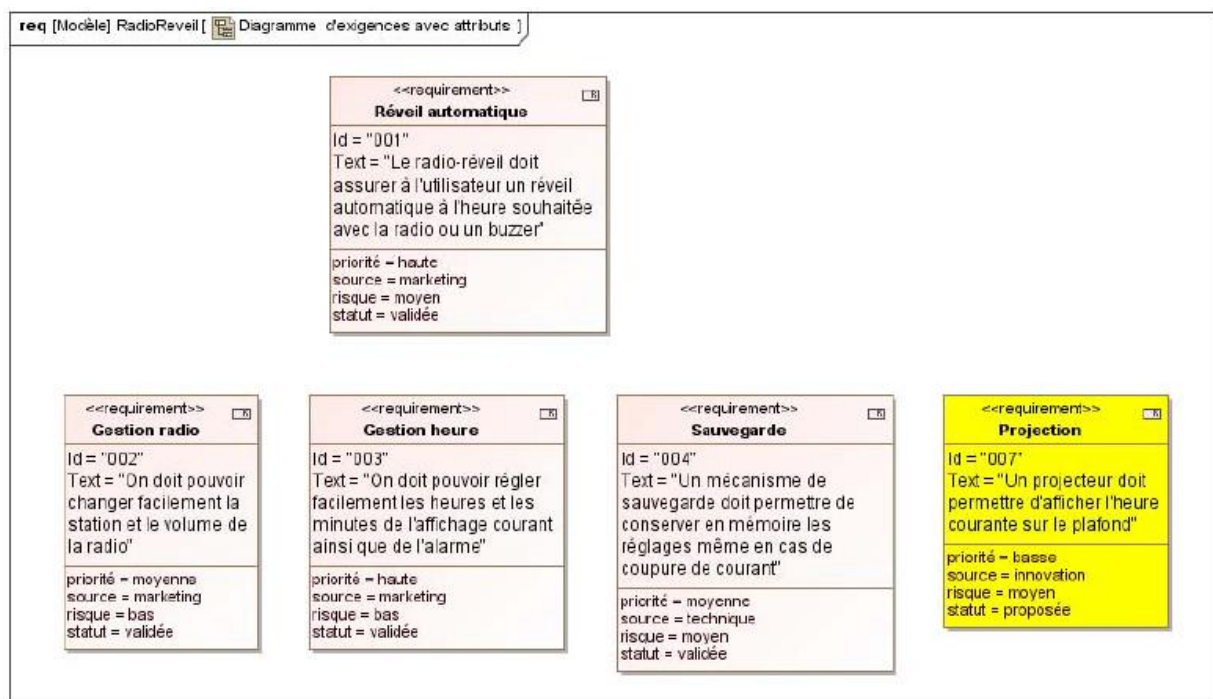
On peut aussi vouloir s'en servir pour écouter la radio ou tout simplement avoir l'heure !



Ceci peut donc conduire à un digramme des cas d'utilisation tel que ci-contre...

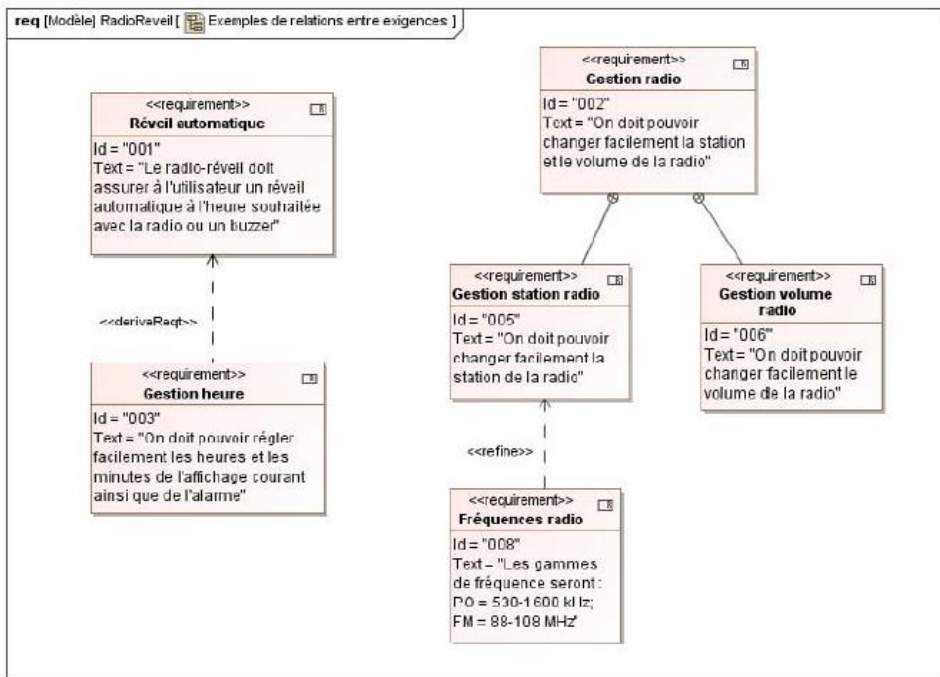
On peut également lister des exigences sur le réglage de la radio, de l'horloge et de l'alarme, ainsi que sur la nécessité d'un mécanisme de sauvegarde et définir des priorités... ainsi que la possibilité d'afficher l'heure courante au plafond grâce à un projecteur intégré.

On peut par exemple obtenir un diagramme d'exigences assez simple où on retrouve les exigences principales (voir ci-dessous) :

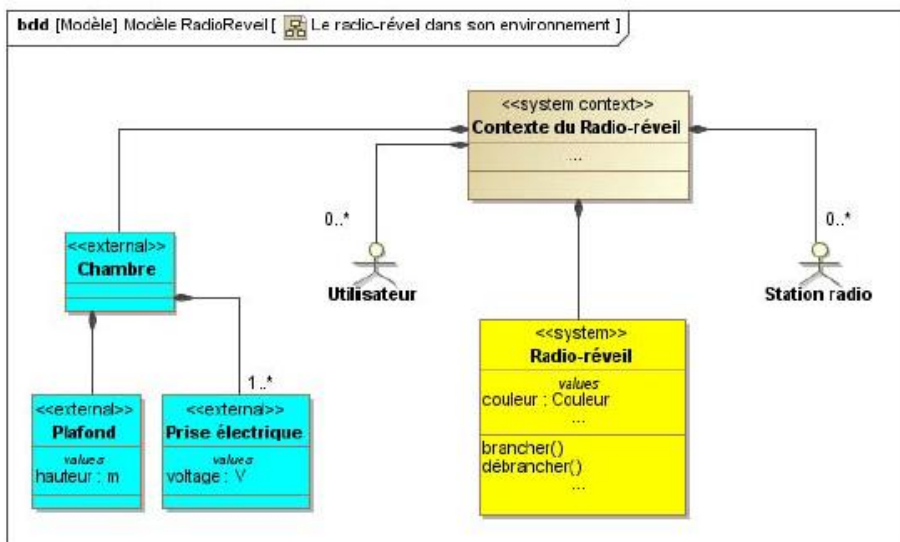


On peut ensuite préciser les relations entre les exigences...

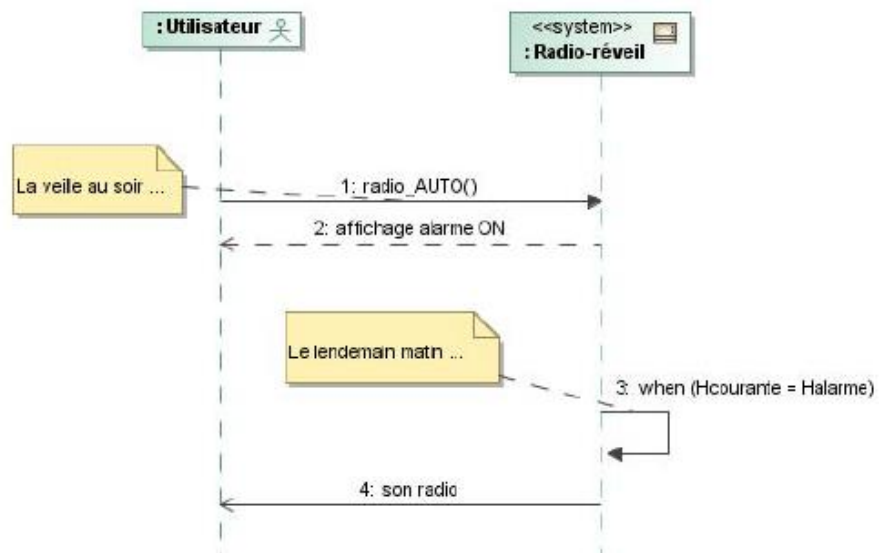
On retrouve alors des valeurs numériques qui pourraient être présentes dans le Cahier des Charges Fonctionnel.



Il est souvent utile de replacer le système dans son environnement sous forme d'un diagramme de contexte, qui reste un cas particulier des Diagrammes de définitions de blocs (bdd) :

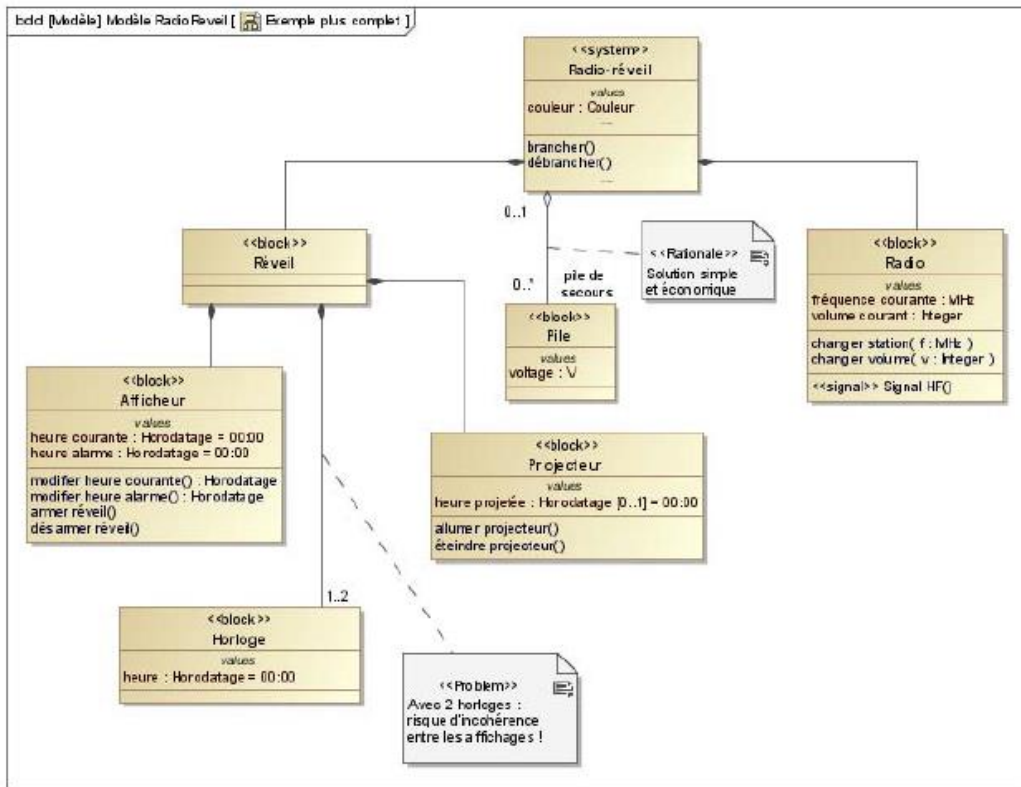
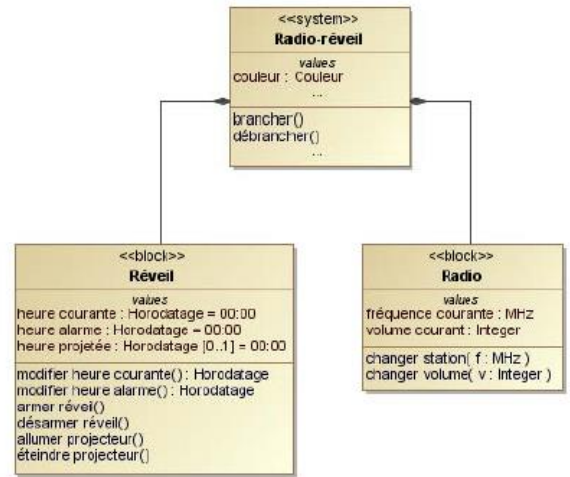


Exemple de Diagramme de séquence (sd) du cas "être réveillé à l'heure" :



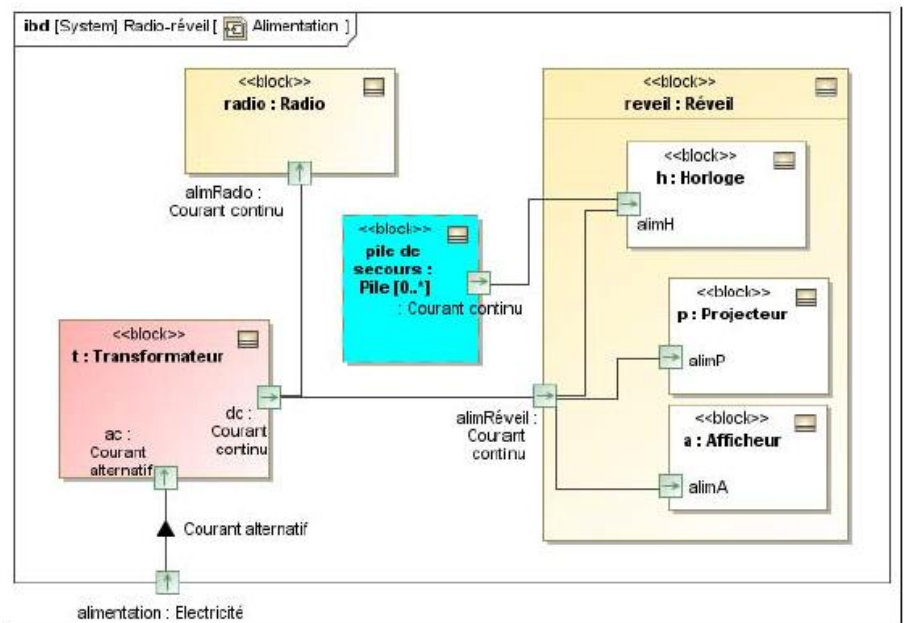
Ensuite on va commencer à décomposer en diagrammes de définitions de blocs (bdd):

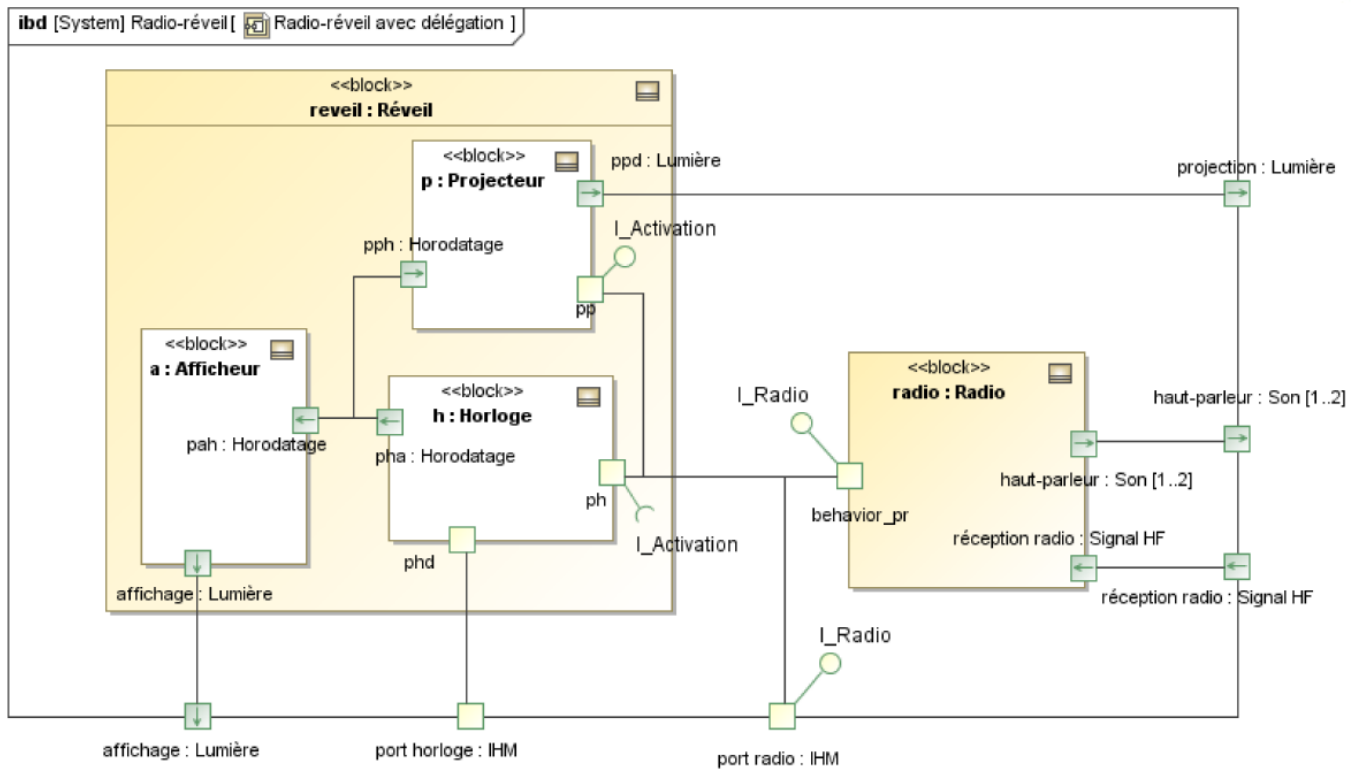
Cette décomposition d'abord simple peut ensuite être plus complète et représenter davantage de liens...



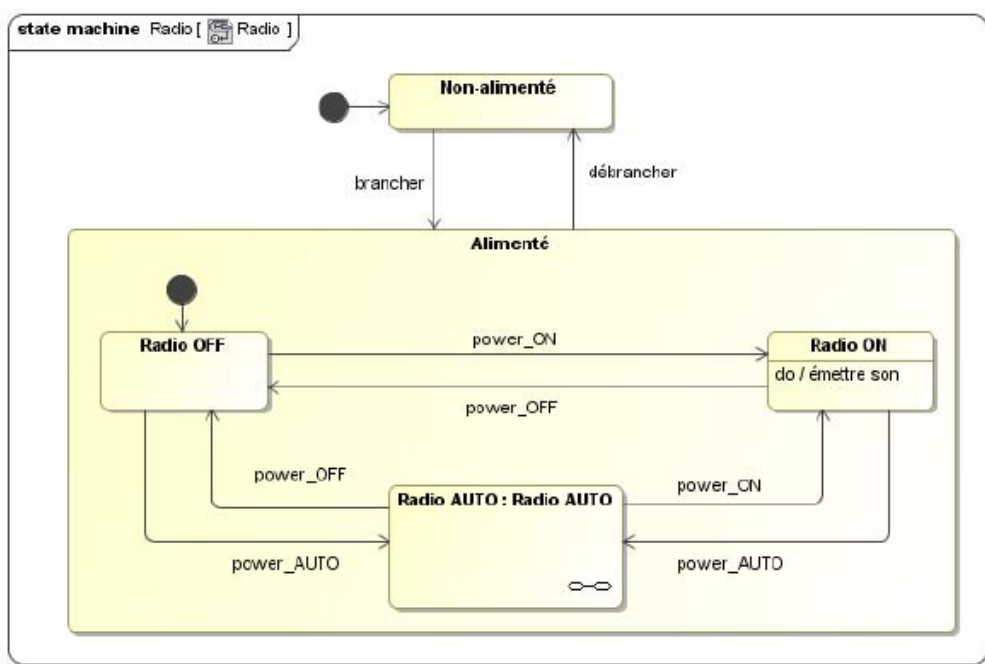
Ensuite chaque "block" sera décomposé en diagrammes de blocs internes (ibd) :

ex : ibd de l'alim électrique





Pour l'aspect comportemental, on peut représenter les diagrammes d'états (stm), comme celui-ci avec gestion de l'alimentation.



et ainsi de suite, on peut aller jusqu'au niveau souhaité.

Le langage SYSML a donc l'avantage d'être complet et de permettre la représentation de l'ensemble du système d'un point de vue structurel, fonctionnel ou comportemental.