

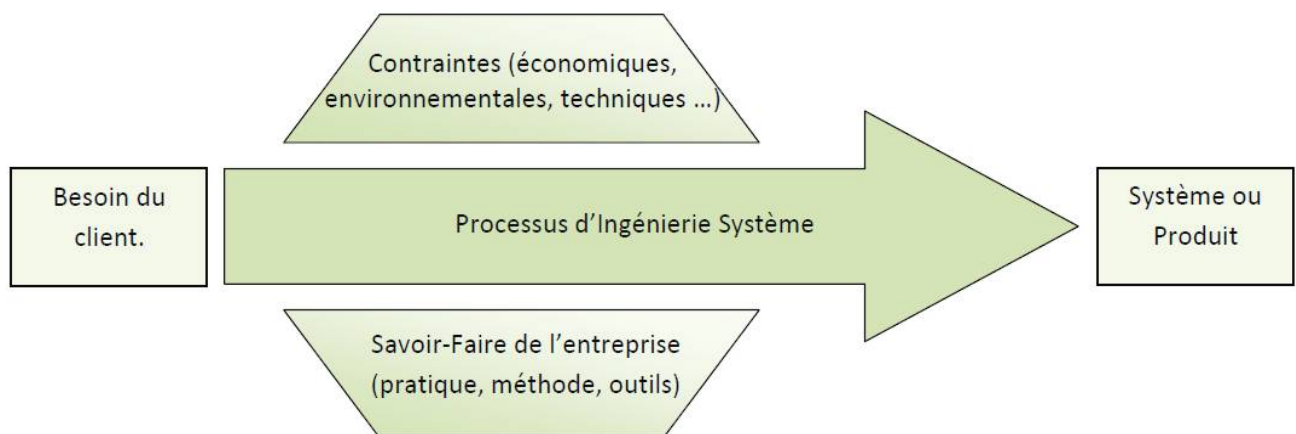
Analyse des systèmes : Langage SysML

1. Introduction / définitions.

1.1. Ingénierie Système

L'Ingénierie Système (IS) est une démarche méthodologique générale qui permet de concevoir, faire évoluer et vérifier un système.

L'ensemble de ces activités permet d'apporter une solution économique et performante aux besoins d'un client et à la vie d'une entreprise.

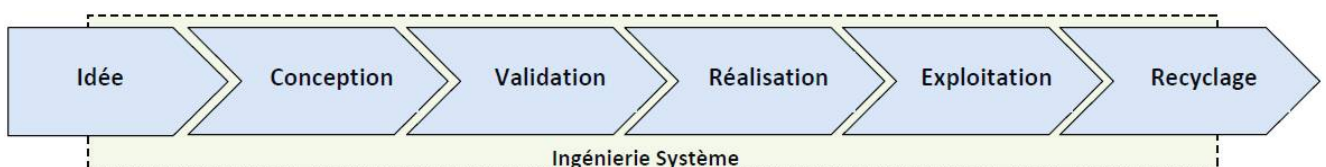


La démarche de l'Ingénierie Système fait appel à trois visions d'un produit ou système :

- ✓ Fonctionnelle : permet de décrire l'expression du besoin et la réponse en terme fonction et de cahier des charges.
- ✓ Structurale : permet de décrire la structure du système.
- ✓ Comportementale : permet de décrire le comportement du système.

Ces trois visions utilisent de nombreux outils de représentation : dessin 2-3D, schéma de principe, représentation symbolique...

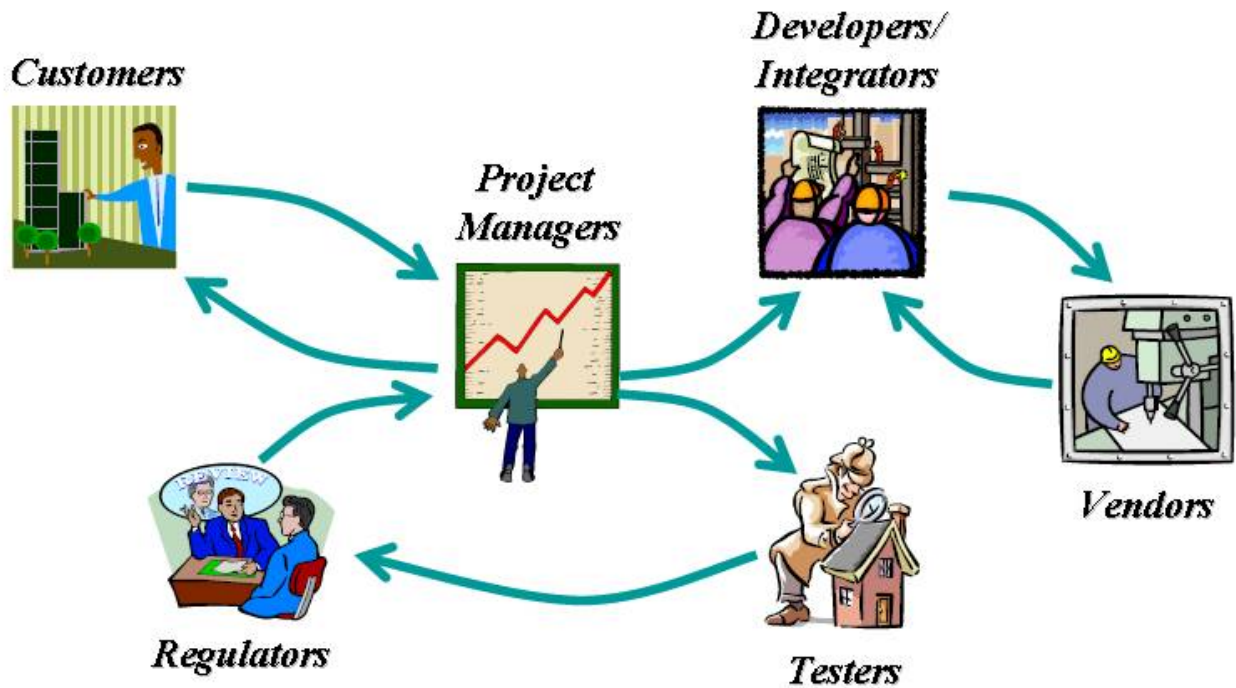
Il s'est avéré nécessaire d'avoir un outil numérique commun d'un bout à l'autre de la chaîne pour des raisons de performance et de compétitivité.



Le langage SysML répond à ce besoin.

Le langage SysML est un moyen de regrouper dans un modèle commun à tous les corps de métiers, les spécifications, les contraintes, et les paramètres de l'ensemble du système.

Il permet ainsi d'éviter tout problème de communication dans l'élaboration du produit.



1.2. Différentes étapes du cycle de vie d'un système :

La notion de « cycle de vie » est indissociable d'un système.

Elle exprime les différentes étapes qui vont de l'analyse du besoin jusqu'à l'élimination et/ou le recyclage de ses constituants.



1.3. Analyse du besoin :

« Un besoin est une nécessité ou un désir éprouvé par un utilisateur » (NF X 50-150).

Le client est sensible à l'évolution du contexte économique, social et environnemental ainsi qu'au degré d'innovation, le besoin évolue donc constamment.

Exemple : le vélo à assistance électrique.



1.4. Cahier des Charges Fonctionnel (C.d.C.F.) :

Le cahier des charges fonctionnel est un document qui permet de formaliser avec précision le besoin du demandeur.

Il énumère les contraintes, qui peuvent être économiques, environnementales, humaines, industrielles, matérielles...

2. Le langage SysML

SysML est un langage graphique, composé de diagrammes qui permettent d'aborder plus facilement les systèmes pluri techniques.

SysML « Systems Modeling Language », veut dire Langage de Modélisation de systèmes.

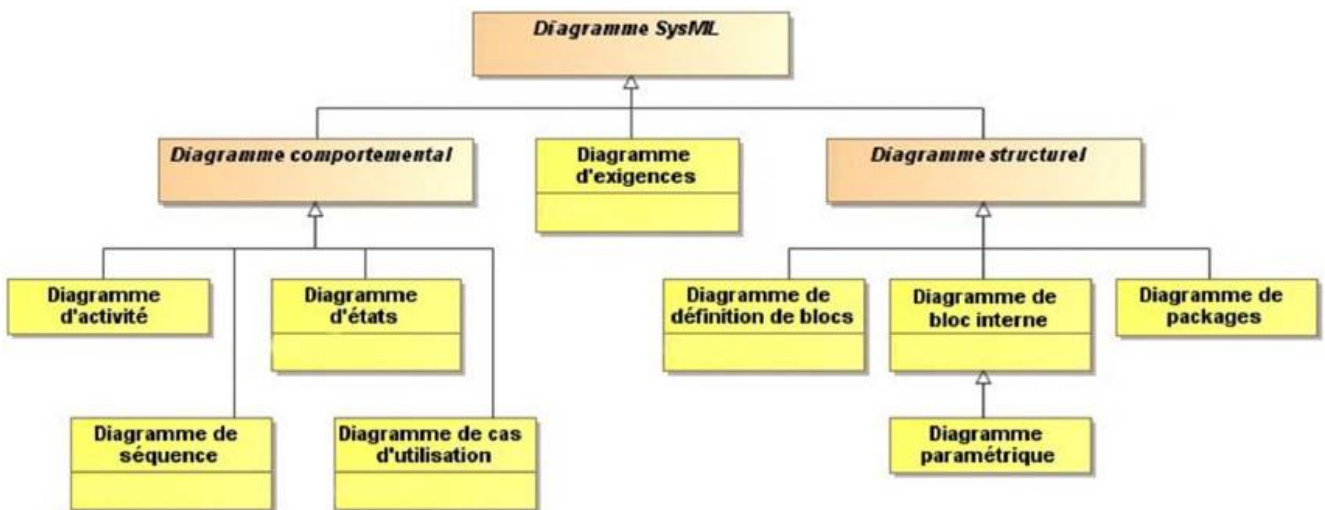
Les diagrammes SysML permettent de représenter :

- ✓ Les exigences du système.
- ✓ Les composants du système.
- ✓ Les flux de toute nature (matière, énergie et information).
- ✓ Le fonctionnement du système.

Les diagrammes SysML :

- ✓ Sont utilisés tout au long du cycle de vie du système (conception, analyse...).
- ✓ Sont communs à tous les champs disciplinaires.
- ✓ Ont leur description propre et sont le plus souvent liés entre eux.
- ✓ Remplacent la plupart des autres outils de description auparavant utilisés.

Il y a 9 diagrammes SysML (tous ne sont pas au programme) :



3. Diagramme des exigences (requirement diagram, Notation SysML: req)

But : Il décrit les exigences du cahier des charges fonctionnel.

Une exigence exprime une capacité ou une contrainte à satisfaire par un système.

Les exigences servent à établir un contrat entre le client et les réalisateurs du futur système

Règles :

- ✓ On indique l'exigence du système dans le premier rectangle, avec un texte descriptif et un identifiant unique.
- ✓ On décompose cette exigence en exigences unitaires.
- ✓ On peut ajouter des données quantitatives et des précisions.

Limites et préconisation :

- ✓ Ne pas chercher à poser toutes les exigences pour rester lisible.
- ✓ Réaliser plusieurs diagrammes d'exigences si nécessaire.

Exemple 1 : Store automatisé.

Ces dernières années, une demande du marché grandissante s'est développée concernant les stores de protection solaire (terrasses de bars, vitrines de magasins, pavillons de particuliers, etc...).

Pour une plus grande simplicité d'utilisation, notamment afin d'éviter une commande manuelle fastidieuse, des mécanismes de commande motorisés et des systèmes automatiques de contrôle de stores se sont développés.

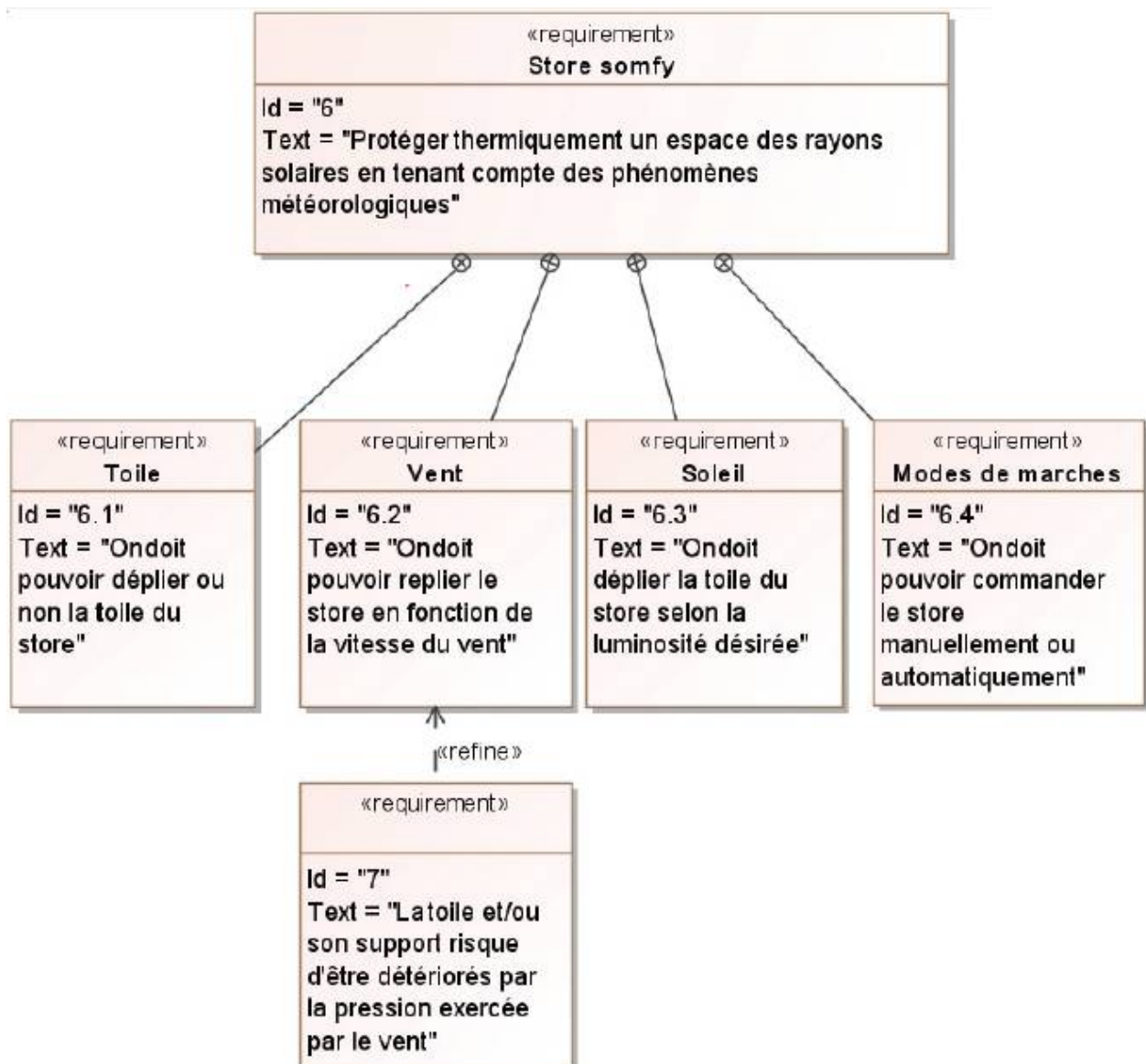
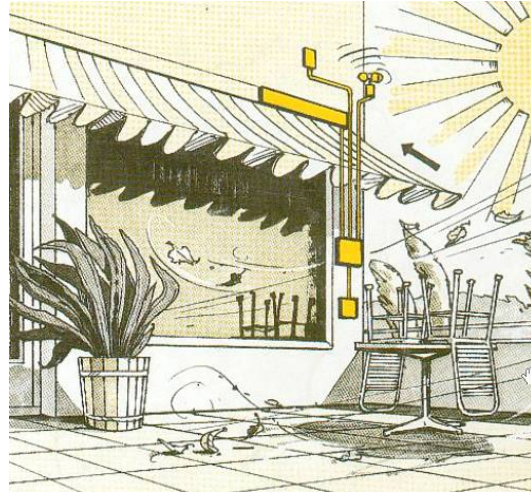
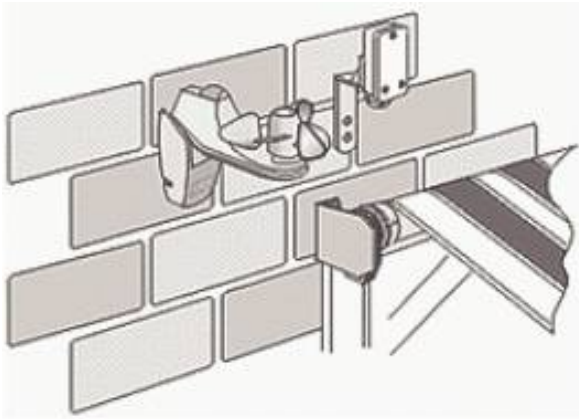
Le système SOMFY :

- ✓ Protège le store contre le vent, selon un seuil réglable.
- ✓ Actionne le store automatiquement en fonction du soleil, selon un seuil réglable.

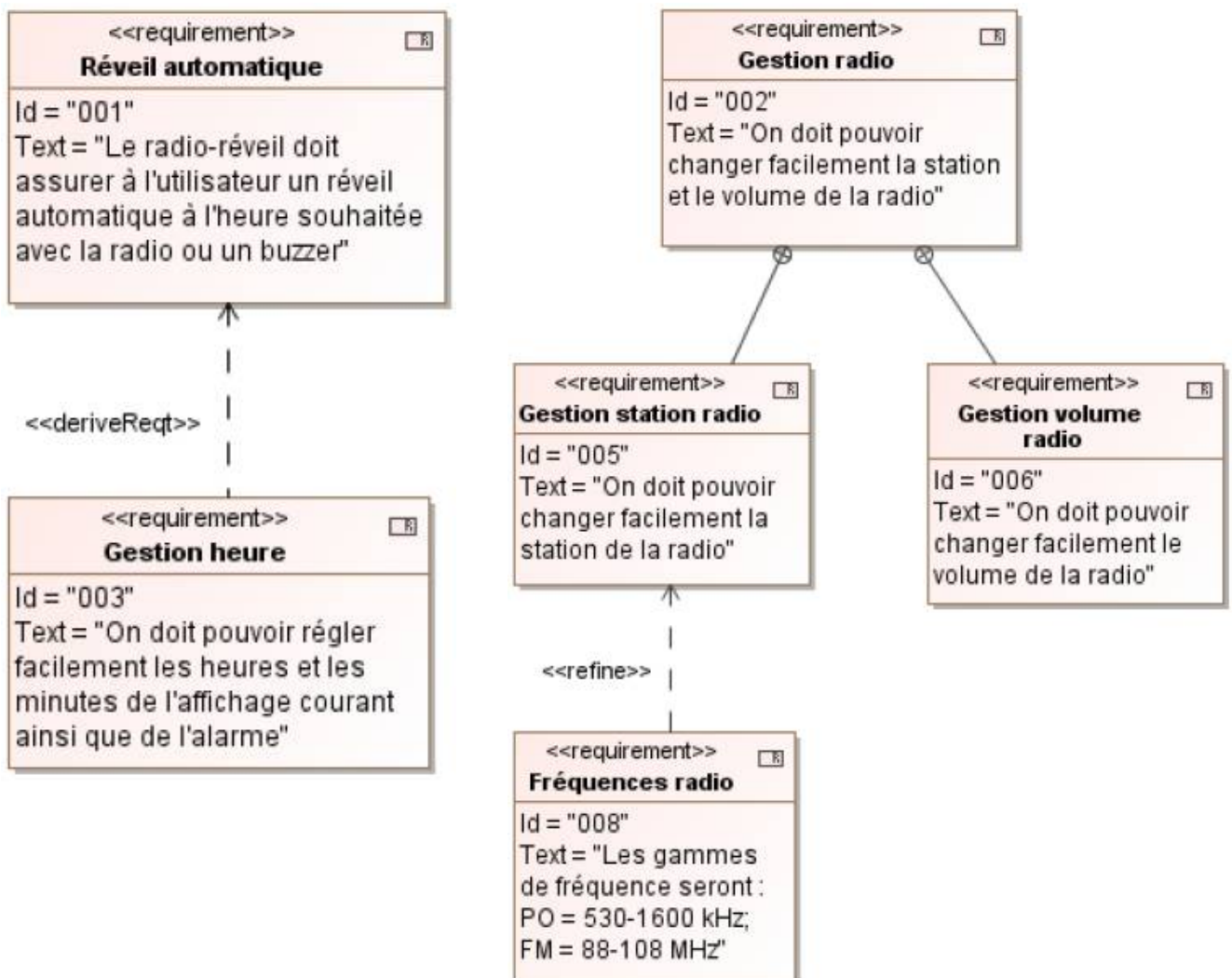
- ✓ Permet si besoin une commande manuelle du store.
- ✓ Visualise l'état de l'automatisme, par des LED.

Une fois les consignes fixées, l'automatisme gère complètement la montée et la descente du store, sans l'intervention humaine, en gardant toujours comme priorité la vitesse du vent.

Le système est peu bruyant, son installation est discrète et son prix de revient est très raisonnable.



Exemple 2 : Radio réveil à projecteur.



Symboles de relation utilisés :



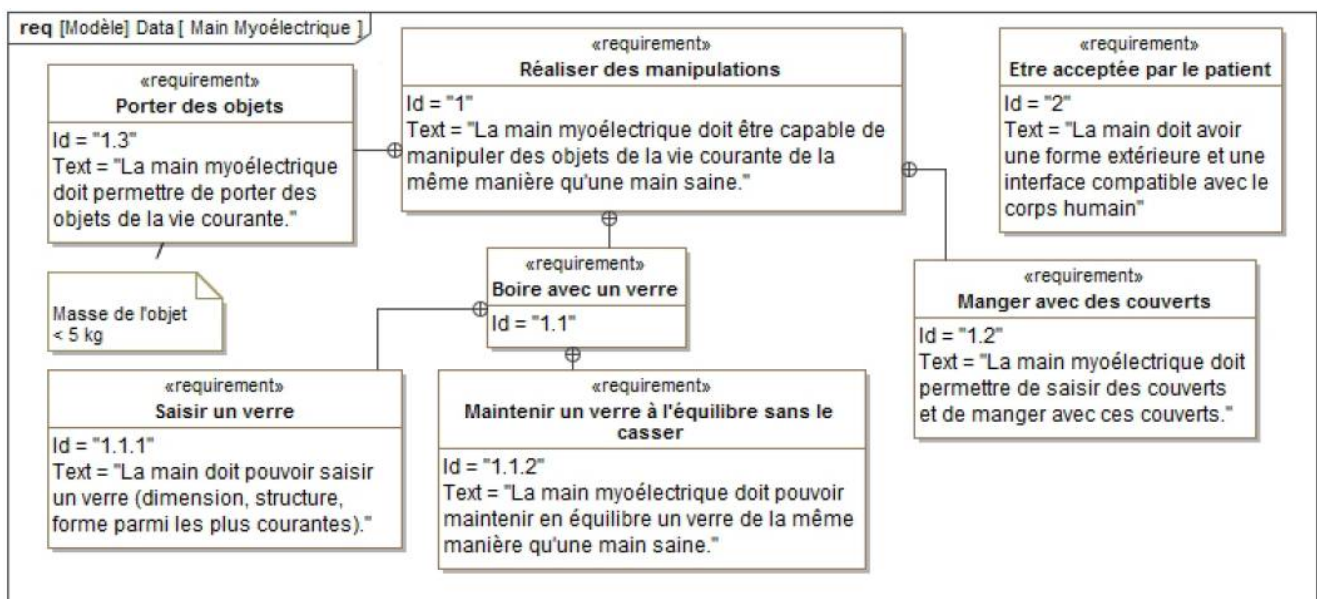
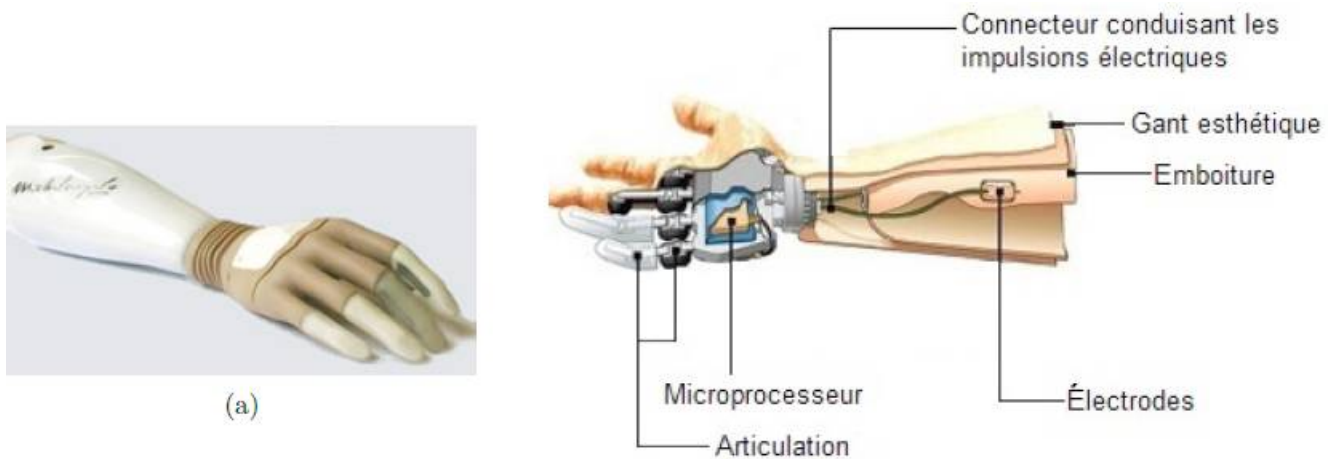
Contenance : Décompose une exigence composite en plusieurs exigences unitaires.



Raffinement (refine) : Consiste en l'ajout de précisions.

Dérivation (deriveReq) : Consiste à relier des exigences de niveaux différents, par exemple des exigences système à des exigences de niveau sous-système, etc...

Exemple 3 : Prothèse de main



Gestion des exigences

Le diagramme d'exigences permet tout au long d'un projet de relier les exigences avec d'autres types d'élément SysML.

Précision sur les exigences

Il est courant de définir des propriétés pour les exigences, par exemple :

- ✓ Priorité : haute, moyenne, basse.
- ✓ Source : client, marketing, technique, législation
- ✓ Risque : haut, moyen, bas.
- ✓ Statut : proposée, validée, implémentée, testée, livrée

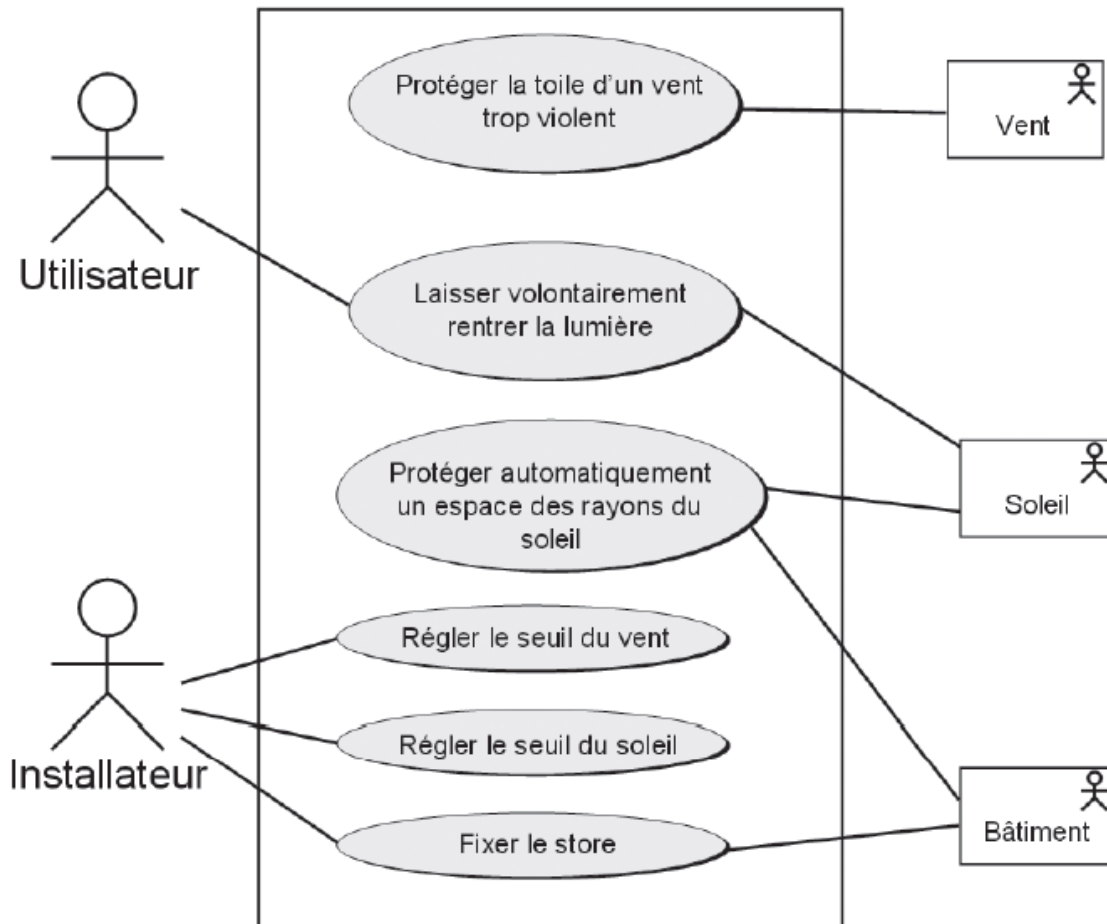
4. Diagramme des cas d'utilisation (use case diagram, Notation SysML: uc)

But : Il montre les interactions fonctionnelles des acteurs et du système d'étude.

Il délimite précisément le système, décrit ce que fera le système sans spécifier comment (et non ce que fera l'utilisateur).

Il exprime les services (use cases) offerts par le système aux utilisateurs (actors).

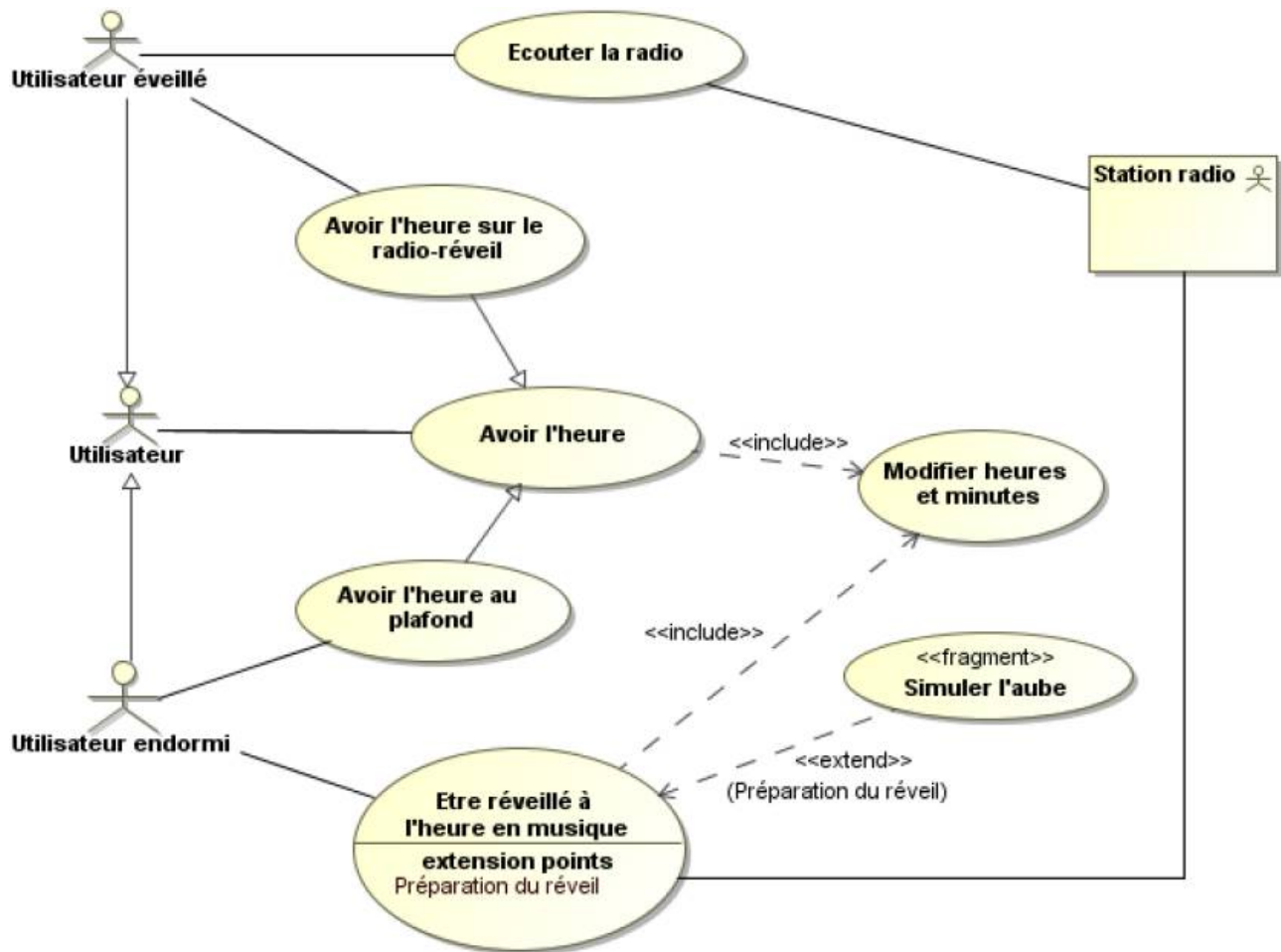
Exemple 1 : Store automatisé.



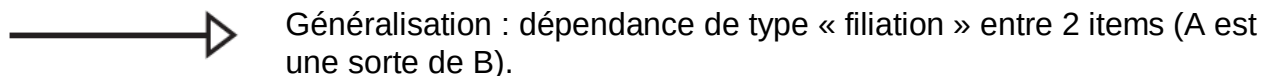
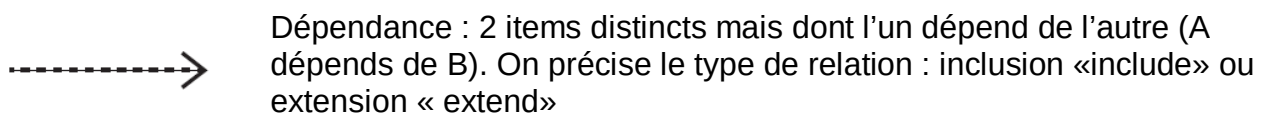
Règles :

- ✓ On trace un cadre délimitant le système et contenant un ensemble de séquences d'actions (Elles peuvent aussi être liées entre elles).
- ✓ A gauche on place les acteurs humains à droite les acteurs non humains (un acteur non humain est représenté par un rectangle).
- ✓ On décrit les actions réalisables par le système (les services rendus par le système aux acteurs, sous forme de verbe à l'infinitif plus compléments)
- ✓ Les acteurs peuvent être reliés entre eux soit par une flèche bidirectionnelle (chaque acteur agit sur l'autre) soit par une flèche unidirectionnelle (un acteur agit sur l'autre)

Exemple 2 : Radio réveil à projecteur.



Symboles de relation utilisés dans l'exemple :



5. Diagramme de définition de block (Notation SysML : bdd).

Rôle : Montre le système du point de vue composant.
Il répond à la question « qui contient quoi ? ».

Le premier bloc représente le système que l'on veut décrire. Ce bloc peut représenter un système complet, un sous-système ou un composant élémentaire.

Les blocs sont décomposés afin de décrire la hiérarchie du système.

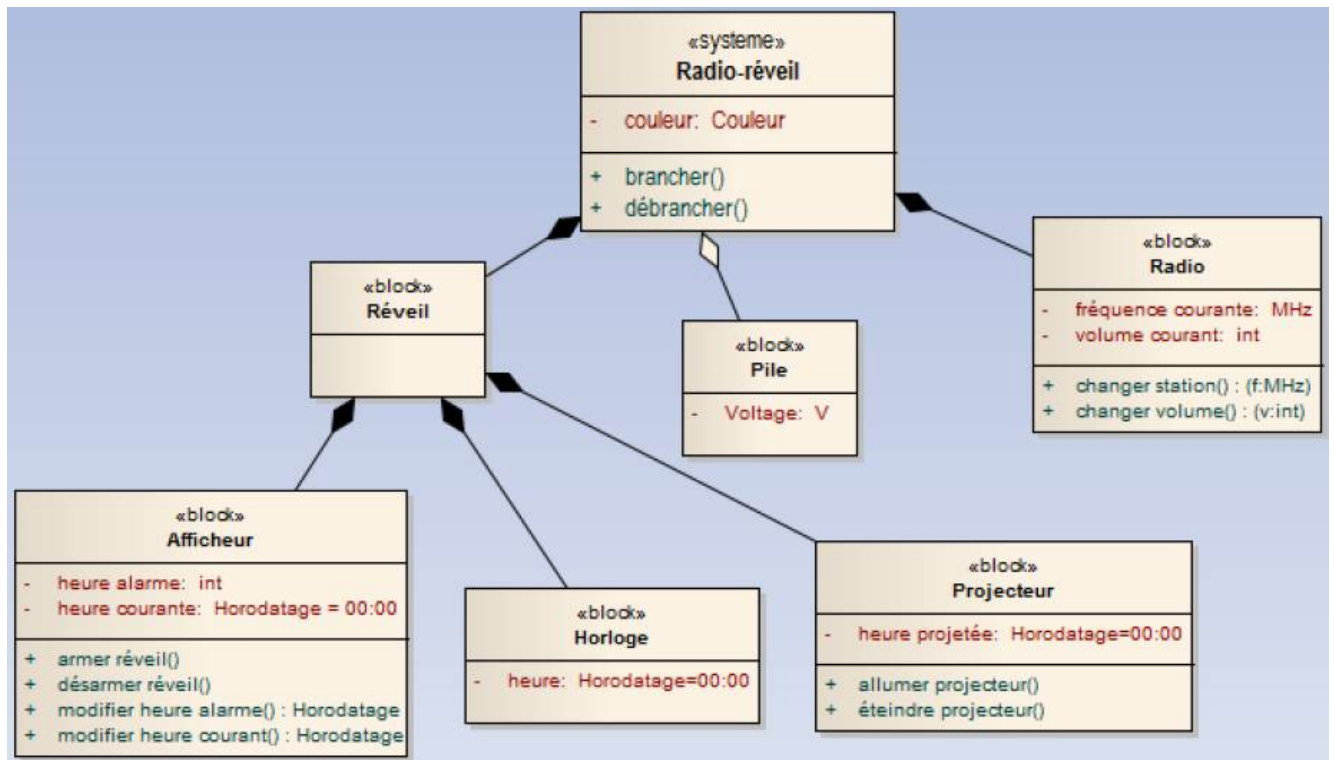
On a différentes zones :

- ✓ La définition du bloc qui présente son nom et son type.
- ✓ Les attributs qui représentent des propriétés qui caractérisent ce bloc.
- ✓ Les opérations qui représentent ce que l'on peut demander au bloc

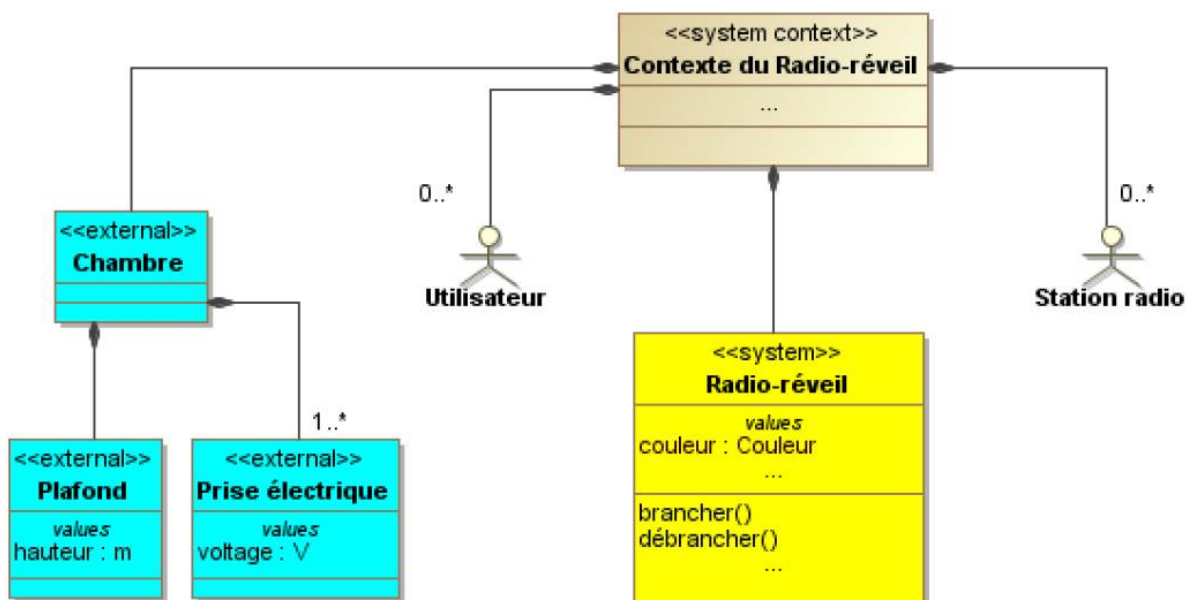
Limites et préconisation :

- ✓ La question du zoom est importante, même si on peut descendre assez bas dans les détails, il n'est en général pas pertinent de le faire. Ce diagramme est utile pour montrer les grosses briques du système.
- ✓ Il n'est pas obligatoire de faire apparaître les propriétés et les opérations dans chaque bloc. Dans ce cas le diagramme est relativement pauvre en informations, mais il offre d'un coup d'œil la structure du système.

Exemple : Diagramme de définition de bloc « bbd » du Radio réveil à projecteur.

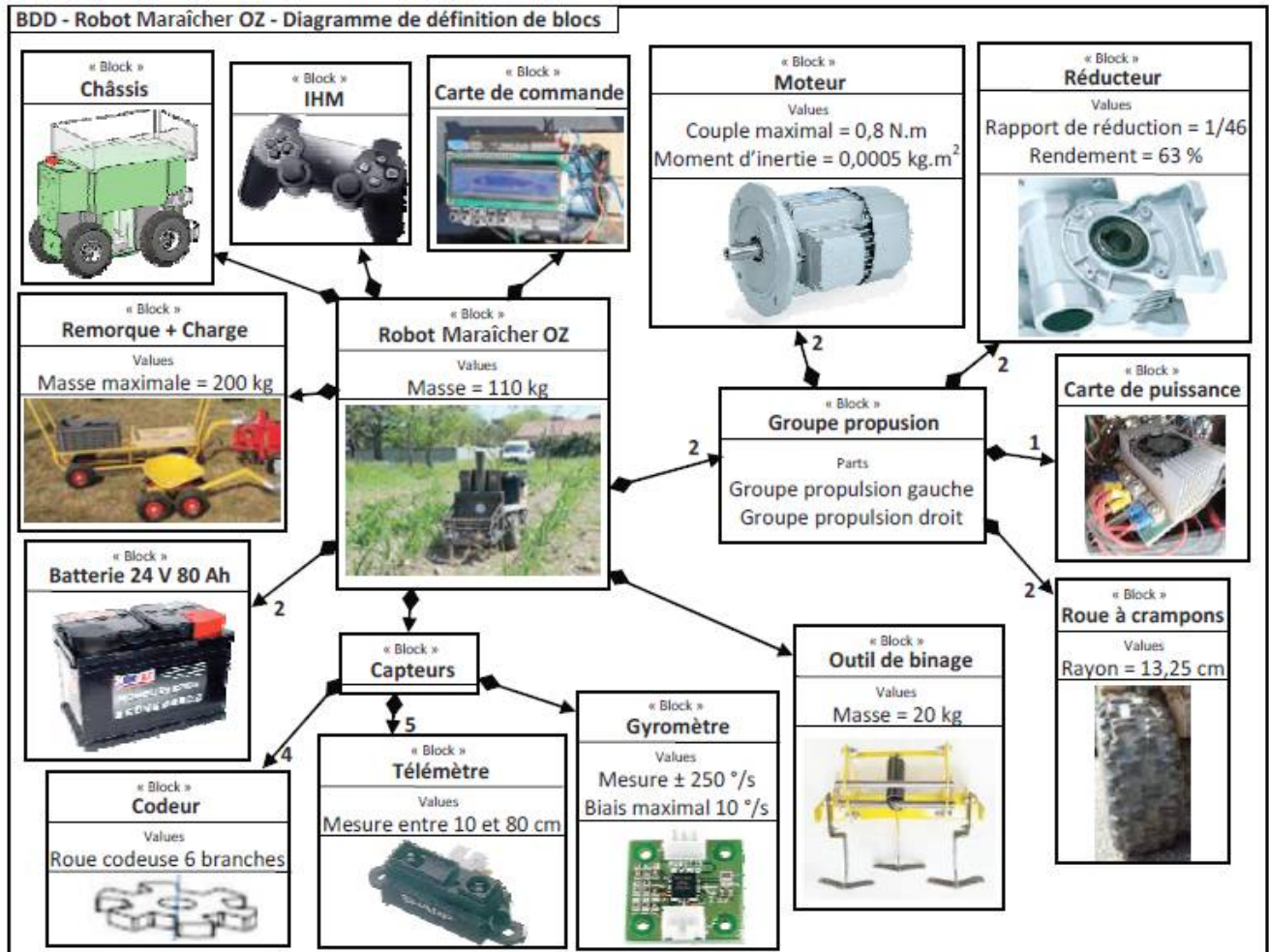


D'un point de vue méthodologique, il est souvent intéressant de remonter d'un cran et de modéliser le contexte du bloc principal (celui qui porte le mot-clé « system context »).



Exemple :

Robot de
maraichage



6. Diagramme de block interne (Internal Block Diagram « ibd »).

Le diagramme de bloc se base sur le « bdd ».

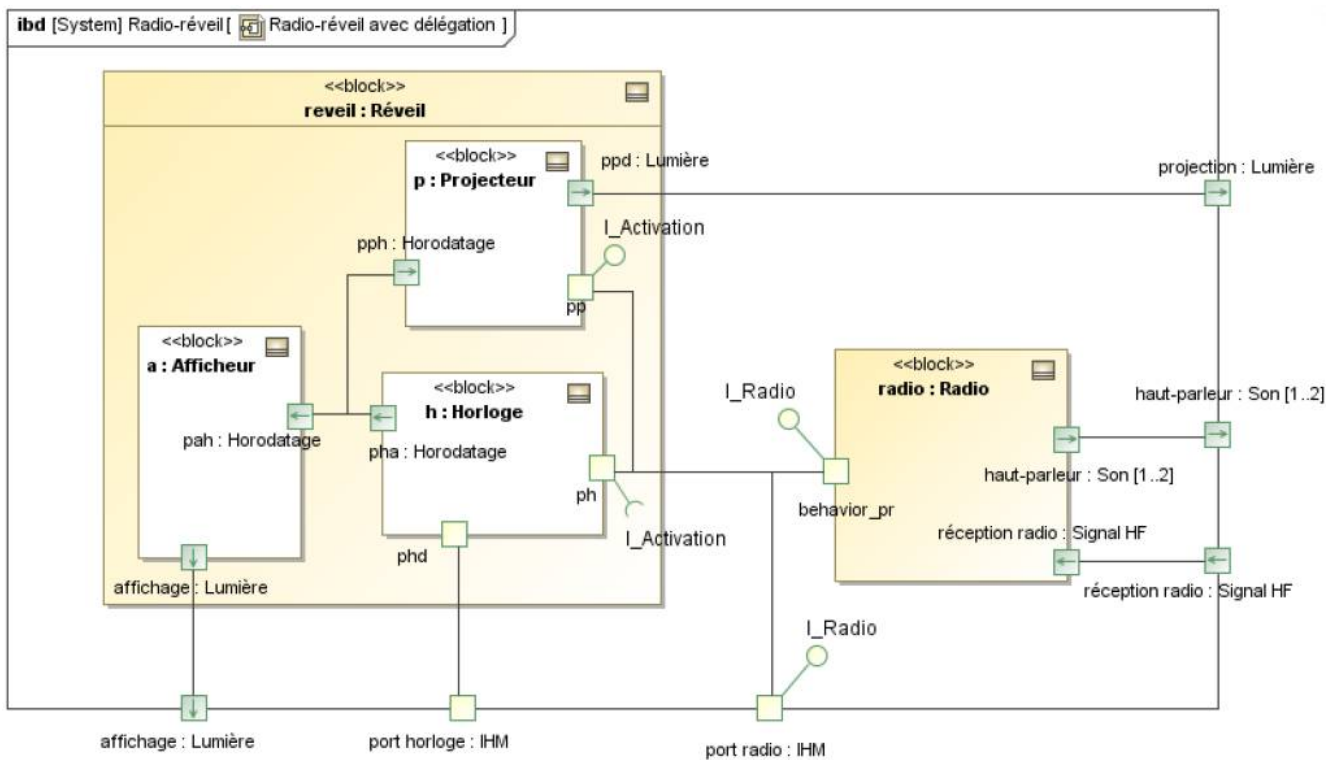
Il décrit les flux entre les blocs.

Les « ports » définissent les points d'entrée ou de sortie entre les blocs (un bloc peut avoir plusieurs ports).

Les ports peuvent être de deux natures :

- ✓  **flux (flow port)** : ce type de port autorise la circulation de flux physiques entre les blocs (données, fluides, énergie).
- ✓  **standard** : ce type de port autorise la description de services logiques entre les blocs, au moyen d'interfaces regroupant des opérations.

Exemple : Radio réveil (basé sur le « bdd »)



7. Exemple : Sécateur électrique.

La période de la taille de la vigne dure environ 2 mois. Les viticulteurs coupent 8 à 10 heures par jour.

Pour réduire la fatigue de la main et du bras, la société PELLENC commercialise un sécateur électrique à commande électronique.

Ce système se compose d'une valise contenant la partie commande PC (portée sur le dos de l'utilisateur) et alimentant un sécateur (tenu par la main de l'utilisateur) par un câble.

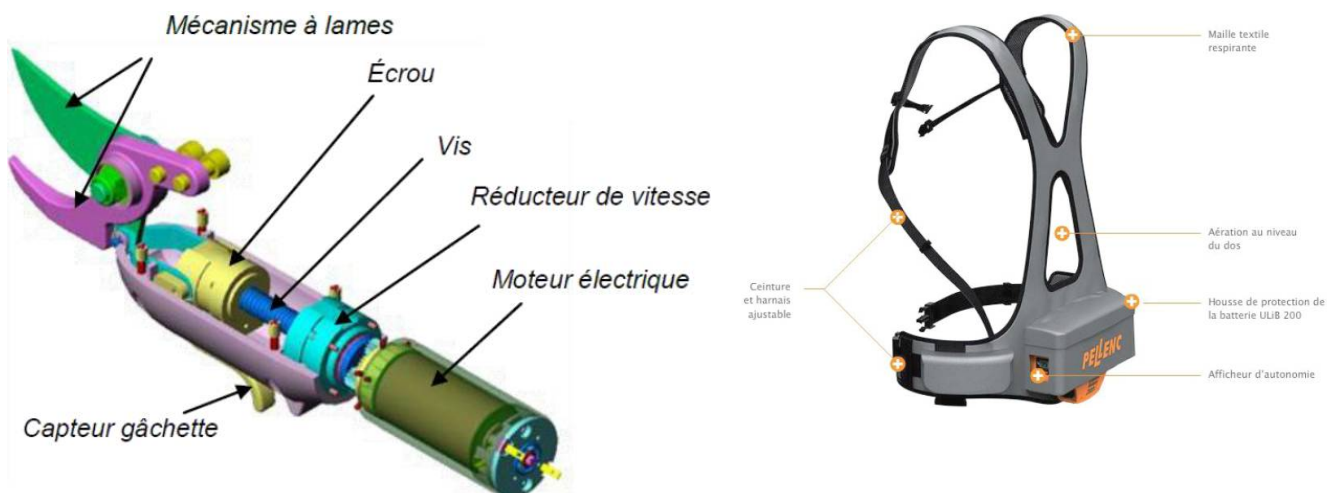


Diagramme de définition de block

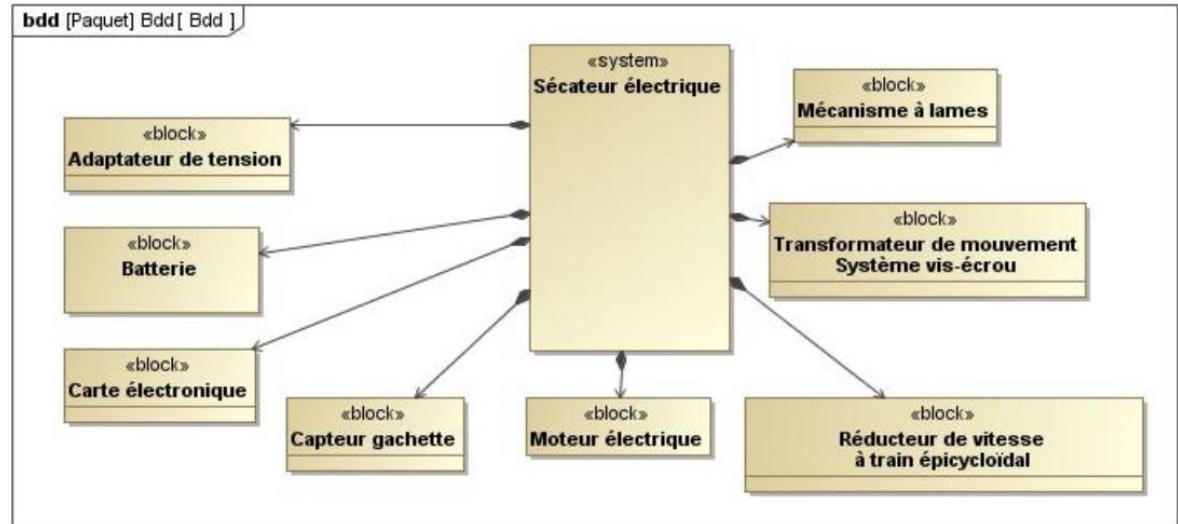
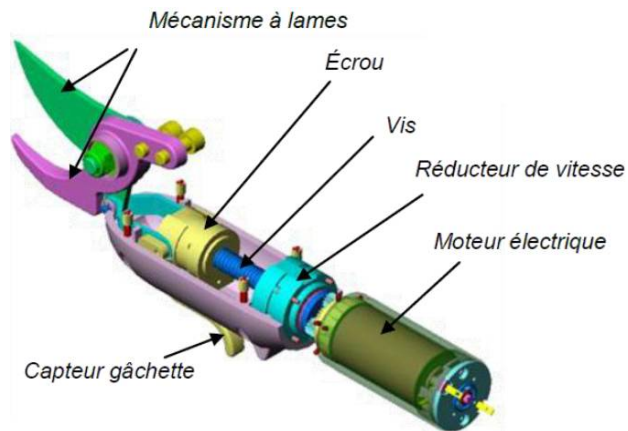


Diagramme de bloc interne

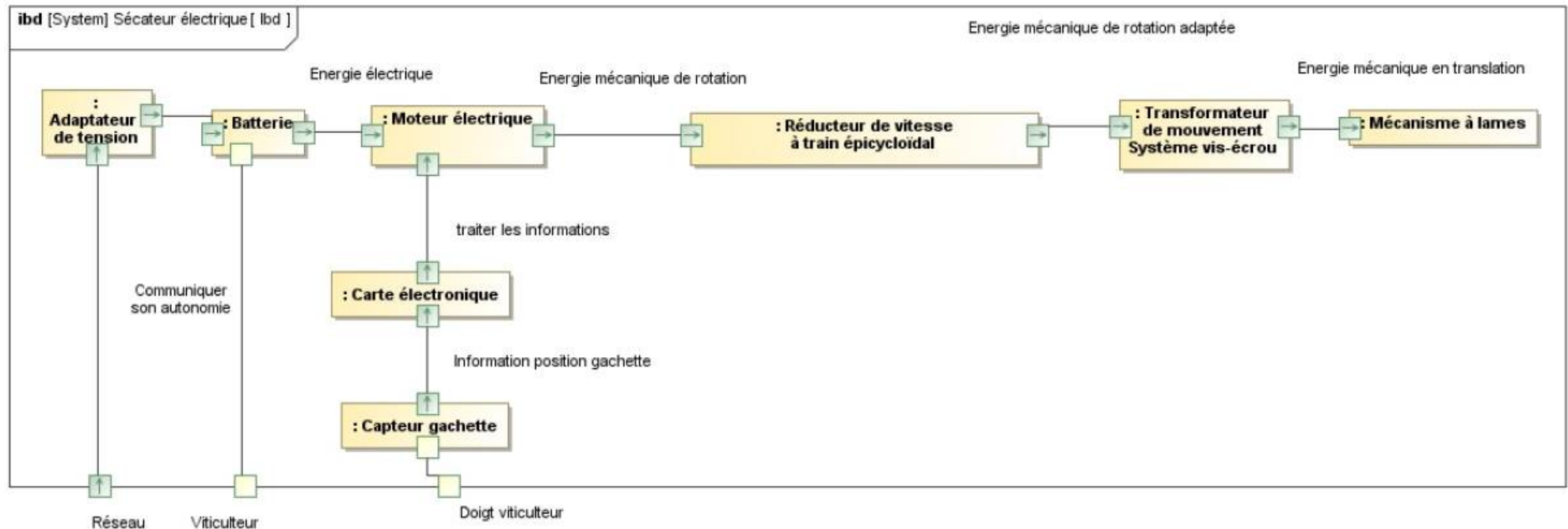


Diagramme de définition de bloc « bdd » du Radio réveil à projecteur.

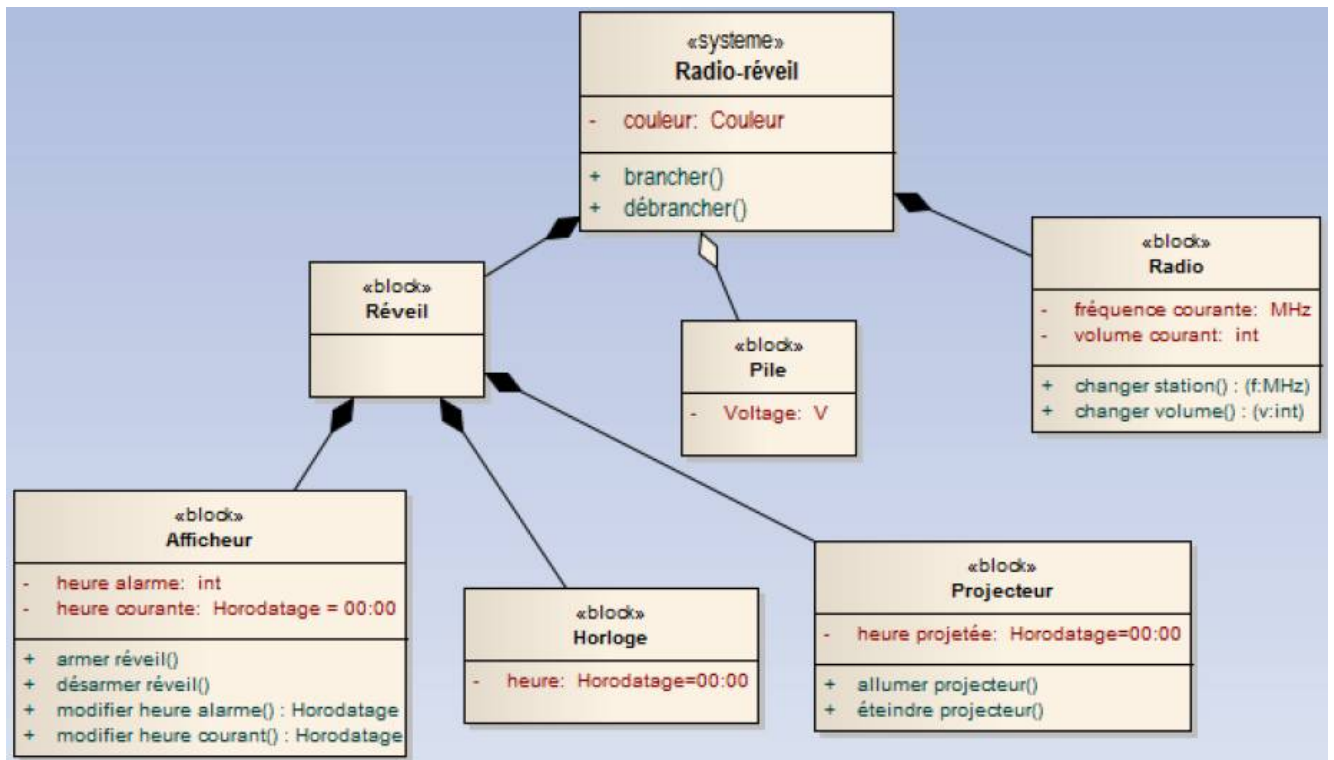


Diagramme de bloc interne du Radio réveil (basé sur le « bdd »)

