



LANGAGE SYSML

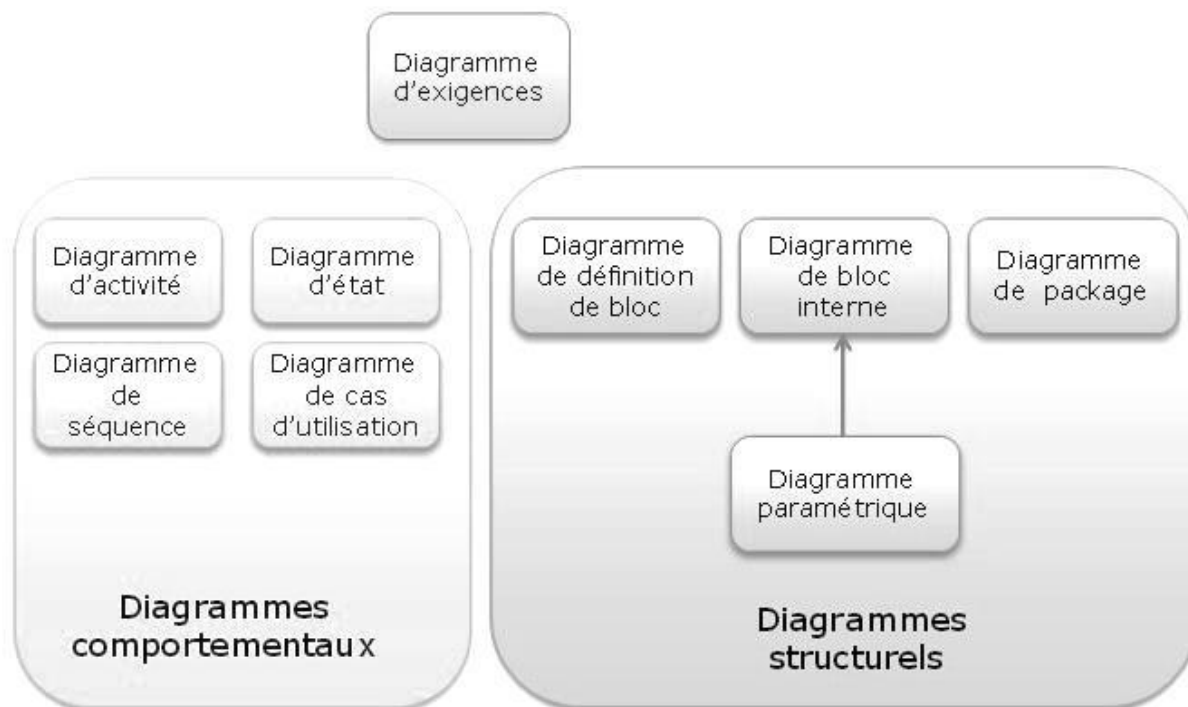


Contenu

1	DIAGRAMME TRANSVERSAL	2
1.1	DIAGRAMME D'EXIGENCES (SYSML REQUIREMENTS DIAGRAM)	2
2	DIAGRAMMES COMPORTEMENTAUX	3
2.1	DIAGRAMME DE CONTEXTE.....	3
2.2	DIAGRAMME DE CAS D'UTILISATION (SYSML USE CASE DIAGRAM)	3
2.3	DIAGRAMME DE SEQUENCE (SYSML SEQUENCE DIAGRAM)	4
2.4	DIAGRAMME D'ETATS / TRANSITIONS (SYSML STATE MACHINE DIAGRAM)	5
3	DIAGRAMMES STRUCTURELS	5
3.1	DIAGRAMME DE DEFINITION DE BLOCS (SYSML BLOCK DEFINITION DIAGRAM)	5
3.2	DIAGRAMME DE BLOCS INTERNES (SYSML INTERNAL BLOCK DIAGRAM)	6
4	DEFINITIONS DES DIFFERENTS TYPES D'ASSOCIATION.....	6
5	DEFINITIONS DES RELATIONS DANS LES DIAGRAMMES.....	6
6	EXEMPLE : SECATEUR ELECTRIQUE PELLENC.....	7
6.1	ANALYSER LE BESOIN	7
6.2	ETABLIR LE CAHIER DES CHARGES FONCTIONNEL (CdCF).....	7
6.3	CONCEVOIR LE SECATEUR	9

Le langage **SysML** - **S**ystems **M**odeling **L**anguage - est un langage de modélisation permettant de décrire tout ou partie d'un système technique, d'un point de vue transversal, comportemental ou structurel.

Le langage **SysML** s'articule autour de neuf types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système.



1 DIAGRAMME TRANSVERSAL

1.1 Diagramme d'exigences (SysML Requirements Diagram)

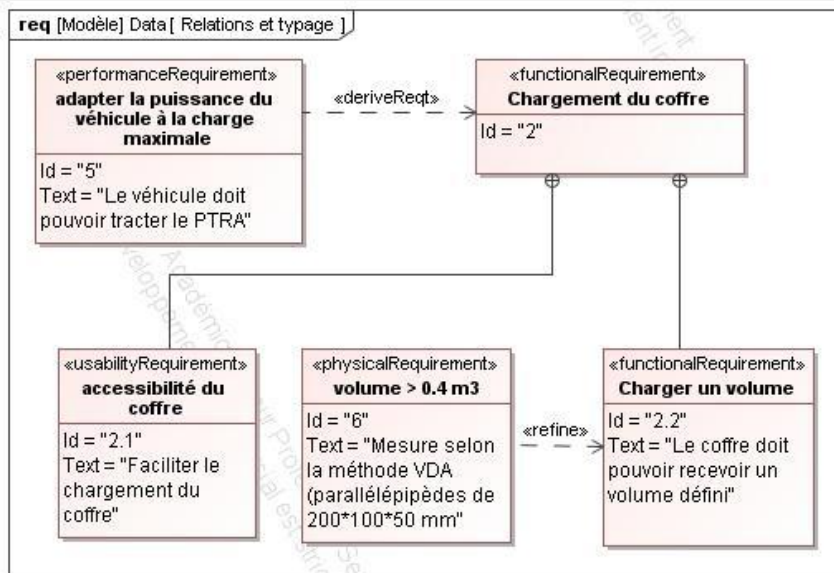
Rôle :

Représente toutes les exigences du système :

- exigences environnementales,
- exigences économiques,
- exigences fonctionnelles,
- exigences techniques,
- etc.

Limites et préconisation :

- Ne pas chercher à poser toutes les exigences sinon illisible.
- Réaliser plusieurs diagrammes d'exigences si nécessaire.
- Regrouper les exigences techniques sur un seul diagramme par exemple, puis les autres groupes d'exigences sur d'autres diagrammes.

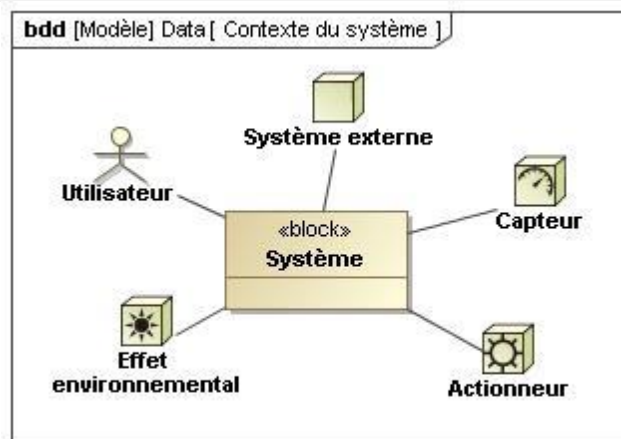


2 DIAGRAMMES COMPORTEMENTAUX

2.1 Diagramme de contexte

Rôle :

- Il répond à la question : "Quels sont les acteurs et éléments environnants du système ?".
- Il permet de définir les frontières de l'étude, et en particulier de préciser la phase du cycle de vie dans laquelle on situe l'étude (généralement la phase d'utilisation).



Limites et préconisation :

Ce diagramme devra bien sûr faire apparaître tous les acteurs intervenants dans le diagramme de cas d'utilisation, mais il fera aussi apparaître les différents acteurs ou éléments intervenant dans une exigence.

Il n'y a aucune recommandation spécifique sur la manière dont il sera établi. Il pourra se faire par :

- une carte mentale,
- un bdd (diagramme de définitions de blocs) SysML (le plus souvent),
- un ibd (diagramme de blocs internes) SysML.

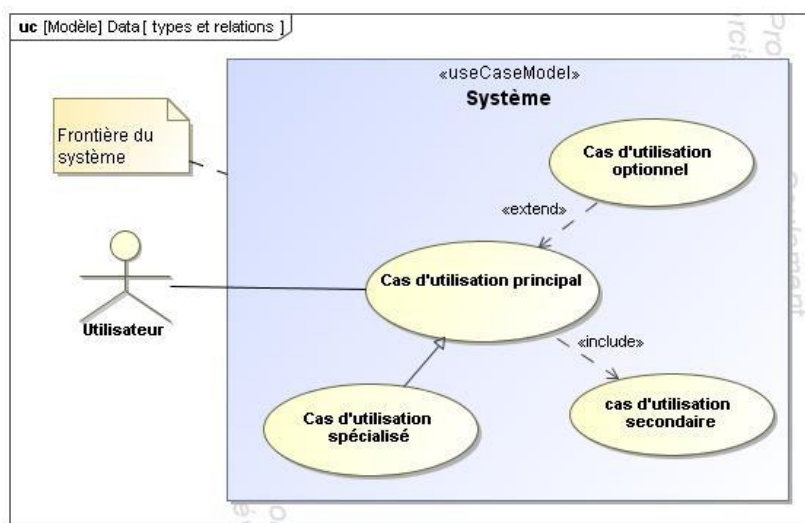
2.2 Diagramme de cas d'utilisation (SysML Use Case Diagram)

Rôle :

- Il répond à la question : "quels services rend le système ?".
- Montre les fonctionnalités offertes par le système.
Fonctionnalité = cas d'utilisation = service rendu en autonomie d'un bout à l'autre par le système.
- Le résultat est visible par l'acteur (entité extérieure en interaction avec le système).

Limites et préconisation :

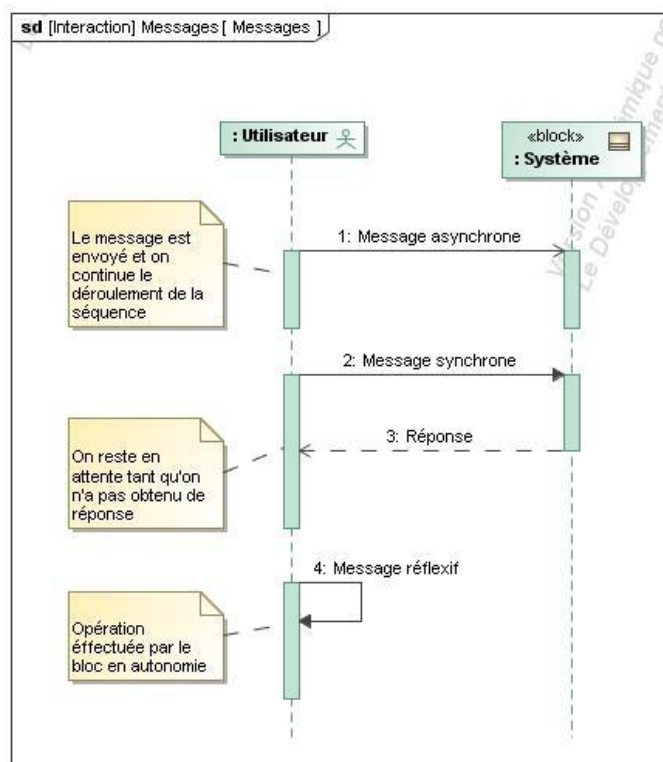
Ce diagramme devra bien sûr faire apparaître tous les acteurs intervenants dans le diagramme de cas d'utilisation, mais il fera aussi apparaître les différents acteurs ou éléments intervenant dans une exigence.



2.3 Diagramme de séquence (SysML Sequence Diagram)

Rôle :

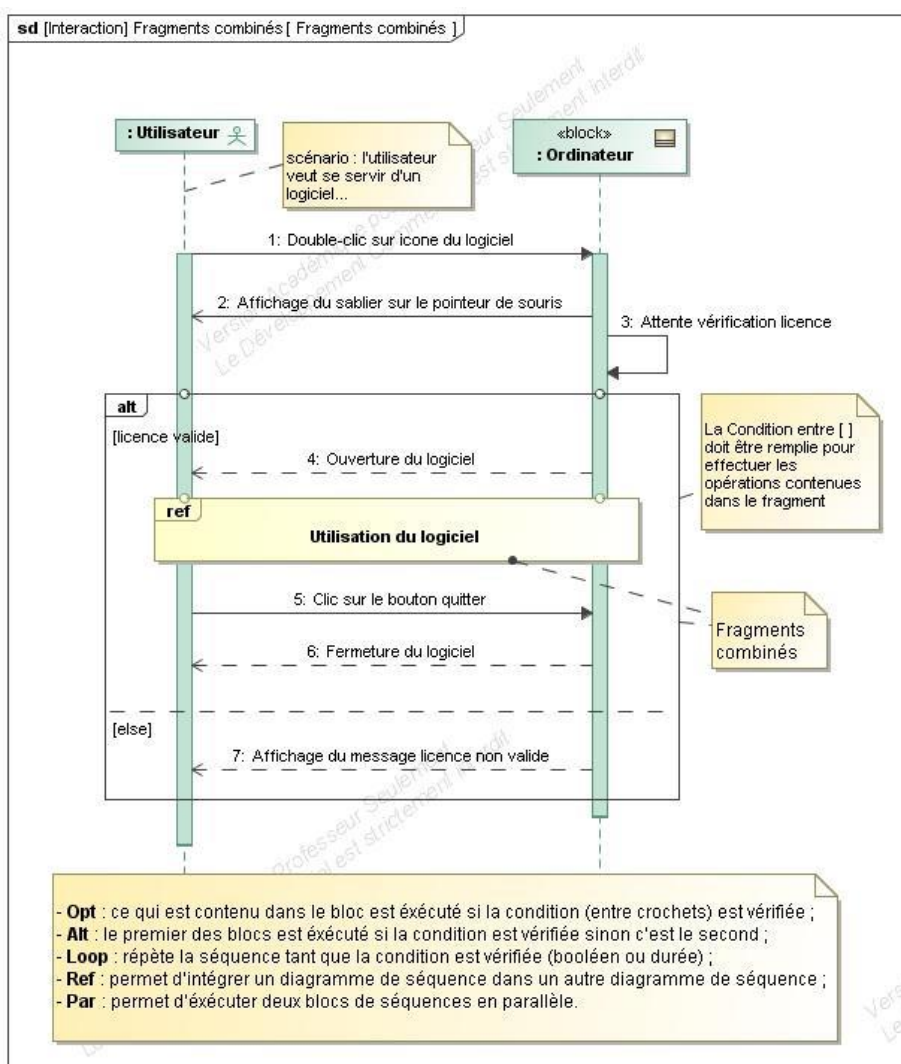
- Il répond à la question : "Comment est réalisé ce cas d'utilisation ?".
- Décrit les scénarios correspondant aux cas d'utilisation, un cas d'utilisation est décrit par au moins un diagramme de séquence.
- Montre les interactions entre différents éléments d'un point de vue séquentiel, enchaînement et nature des échanges.
- Dans un premier temps il est préférable de faire un diagramme de séquence « système », celui-ci étant vu comme une boîte noire.
- Pour des détails vous pouvez ensuite montrer les interactions au sein du système (décomposé en ses différents éléments).



Limites et préconisation :

Il existe les **fragments combinés** pour montrer des variantes dans un scénario. Cela doit être utilisé avec parcimonie car les diagrammes de séquence ne sont pas des algorithmes.

Un scénario se décrit dans un cadre bien précis. Vous aurez donc des scénarios de réussite et des scénarios d'échecs (gestion des problèmes).



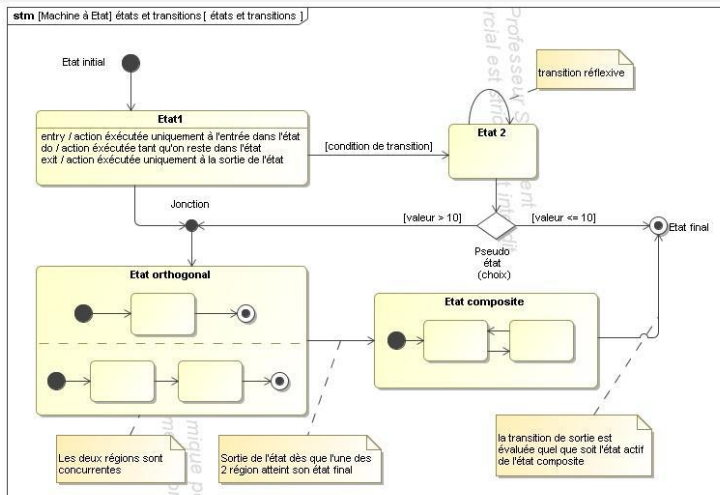
2.4 Diagramme d'états / transitions (SysML State Machine Diagram)

Rôle :

- Il répond à la question : "Comment représenter les différents états du système ?"
- Décrit le comportement d'un programme sous forme de machine d'états.
- Montre les différents états pris par le système (ou un sous-système) en fonction des interactions

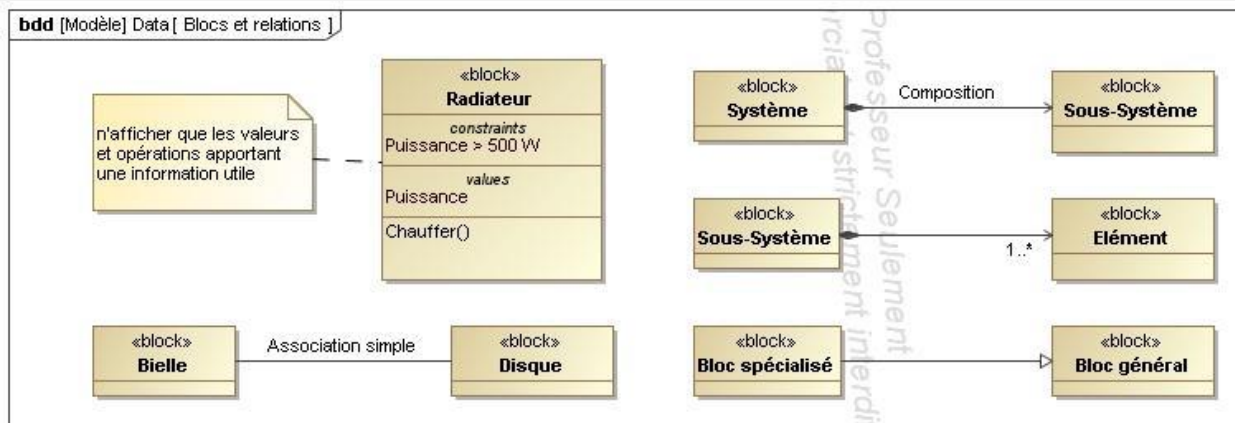
Limites et préconisation :

- Ce diagramme servira avant tout à décrire le fonctionnement d'un programme.
- Ce diagramme trouve toute sa place en lien avec les logiciels de simulation comportementale (Matlab, LabView, Modelica,...) .



3 DIAGRAMMES STRUCTURELS

3.1 Diagramme de définition de blocs (SysML Block Definition Diagram)



Rôle :

- Il répond à la question "qui contient quoi?".
- Montre le système d'un point de vue composé/composant
- Il peut aussi montrer les caractéristiques principales de chaque bloc en faisant apparaître les opérations (rôles) et les propriétés (caractéristiques).
- Permet de représenter les liens entre les blocs de même niveau par une association (simple trait entre 2 blocs).

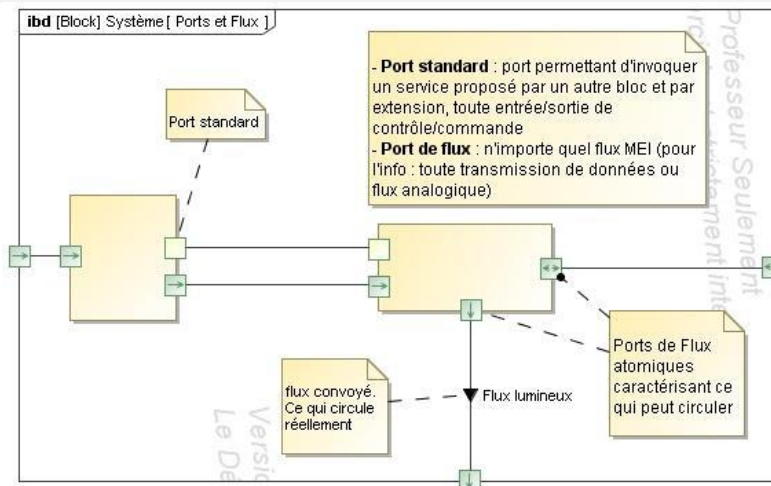
Limites et préconisation :

- La question du zoom est importante. Même si on peut descendre assez bas dans les détails, il ne sera pas pertinent en général 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.

3.2 Diagramme de blocs internes (SysML Internal Block Diagram)

Rôle :

- Permet de représenter les échanges de matière/information/énergie entre blocs de même niveau grâce aux ports de flux (petit carré avec une flèche).
- Permet de représenter les services invoqués par un autre bloc grâce aux ports standards (petit carré sans flèche), et par extension toute entrée/sortie de contrôle/commande.
- Permet de représenter les liens entre les blocs de même niveau.



Limites et préconisation :

Il faut bien retenir que les liens se représentent entre blocs de même niveau, ils ne se contiennent pas.

Chaque bloc du BDD contenant d'autres blocs peut être représenté par un IBD.

Attention à bien faire la différence entre port standard et port de flux.

Port standard : désigne une interface permettant d'invoquer un service/une opération

Port de flux : canal d'Entrée/Sortie par lequel transite de la matière, de l'énergie ou de l'information (MEI).

4 DEFINITIONS DES DIFFERENTS TYPES D'ASSOCIATION

- Extend** : le cas d'utilisation source est une extension possible du cas d'utilisation destination.
- Include** : le cas d'utilisation source comprend obligatoirement le cas inclus.
- DeriveReq** : une ou plusieurs exigences sont dérivées d'une exigence.
- Satisfy** : un ou plusieurs éléments du modèle permettent de satisfaire une exigence.
- Verify** : un ou plusieurs éléments du modèle permettent de vérifier et valider une exigence.
- Refine** : un ou plusieurs éléments du modèle redéfinissent une exigence.

5 DEFINITIONS DES RELATIONS DANS LES DIAGRAMMES

- A** **B**
- **Association** : relation d'égal à égal entre 2 éléments. **A utilise B**
- 3 diagrammes : cas d'utilisation, définition des blocs, blocs internes.
 - - - - - → **Dépendance** : l'un des deux éléments dépend de l'autre. **A dépend de B**
- 3 diagrammes : cas d'utilisation, exigences, définition des blocs.
 - ◊ **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, définition des blocs.
 - ◆ **Composition** : Un élément est une composante obligatoire de l'autre. **A entre dans la composition de B et lui est indispensable à son fonctionnement.**
- 2 diagrammes : exigences, définition des blocs.
 - ▷ **Généralisation** : Dépendance de type 'filiation' entre 2 éléments. **A est une sorte de B.**
- 3 diagrammes : exigences, définition des blocs, blocs internes.
 - ⊕ **Conteneur** : relation d'inclusion entre 2 éléments. **B contient A**
- 2 diagrammes : exigences, définition des blocs.

6 EXEMPLE : SECATEUR ELECTRIQUE PELLENC

Pour appréhender le langage SysML, nous nous appuyerons sur un exemple concret : le sécateur électrique Lixion évolution (société PELLENC).

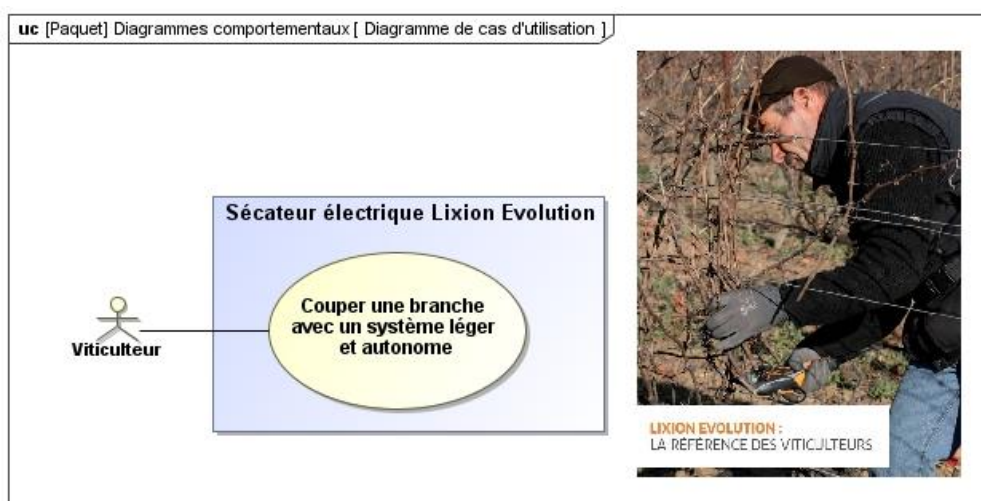


6.1 Analyser le besoin

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 cas d'utilisation :



6.2 Etablir le Cahier des Charges Fonctionnel (CdCF)

Il est extrait du tableau des exigences (SysML Requirements Table)

#	ID	Name	Text
1	3	Coût	< 1 400 € HT
2	1.2.1.1	Taille de la branche	Diamètre de La branche à tailler < 35 mm
3	1.2.3	Cadence de coupe	82 coupes par minute
4	1.2.2.1	Durée de coupe	0,5 s maximum
5	2.1.1	Temps d'utilisation	Autonomie jusqu'à 3 jours suivant l'utilisation
6	1.4.3.2	Diamètre	diamètre du manche < 40 mm
7	2.2.1.1	Masse maximale de la source énergie	2,5 kg
8	1.4.4.1	Masse maximale du sécateur seul	800 gr
9	5	Normes	le sécateur doit être conforme à l'article R.233-100 du Code du Travail
10	11.1	Température extérieure	Le sécateur doit résister à des températures extérieures comprises entre -5 et 50 °
11	11.4.1	Humidité, poussière	Le système doit être étanche

Diagramme d'exigences :

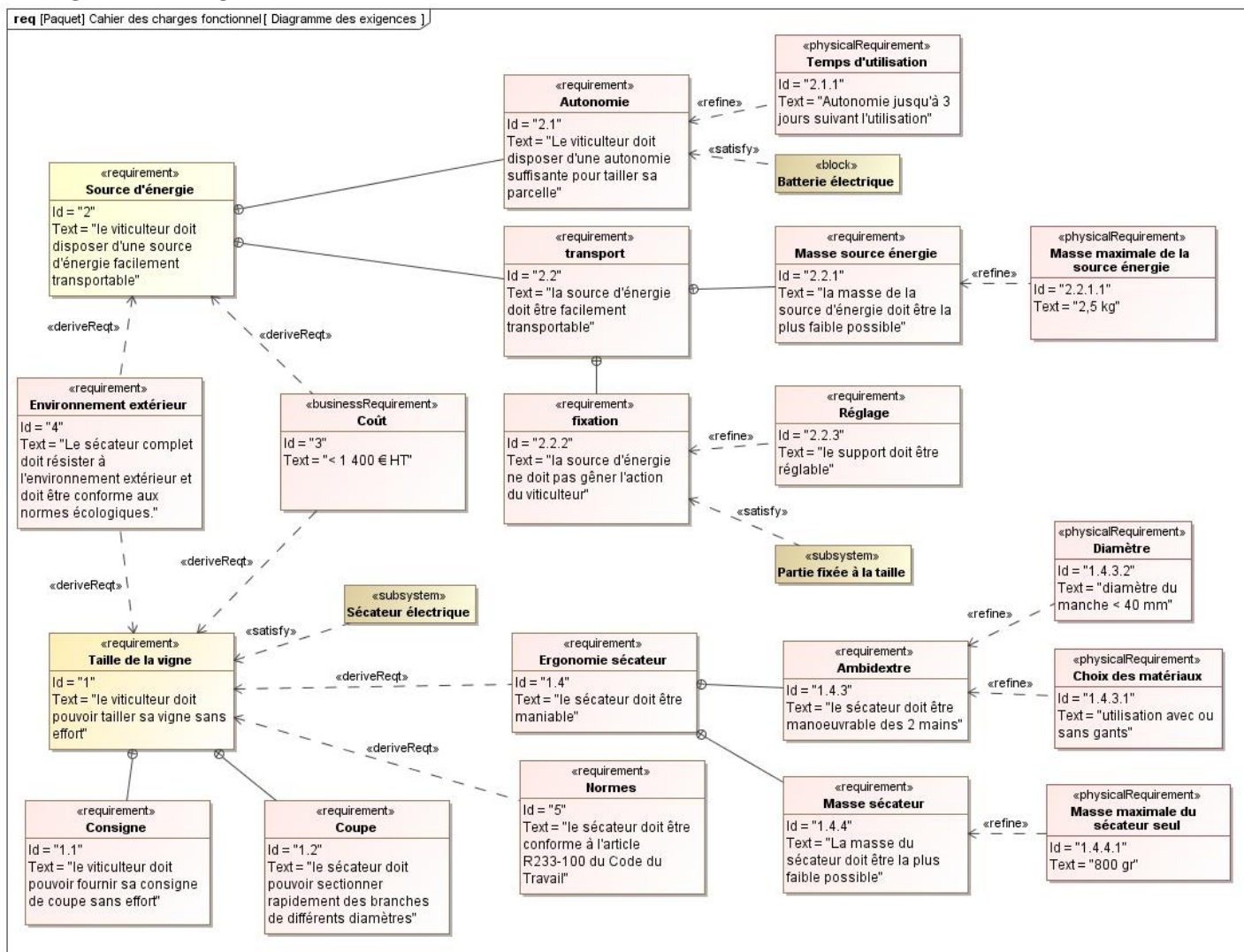
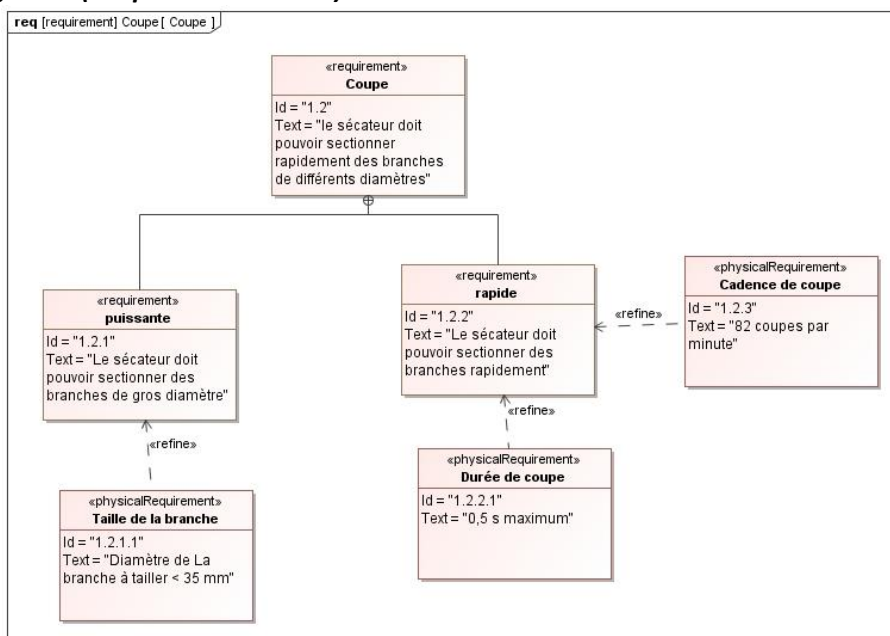


Diagramme d'exigences (coupe de la branche) :



6.3 Concevoir le sécateur

Diagramme de définition de blocs du sécateur complet :

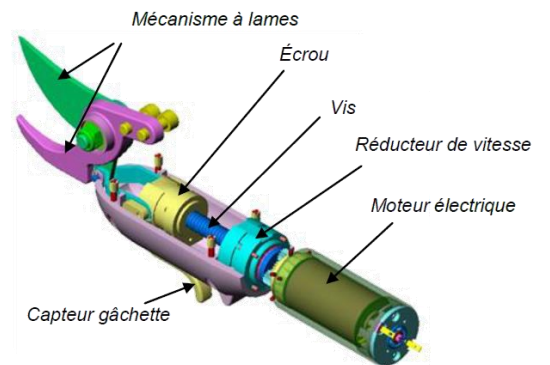
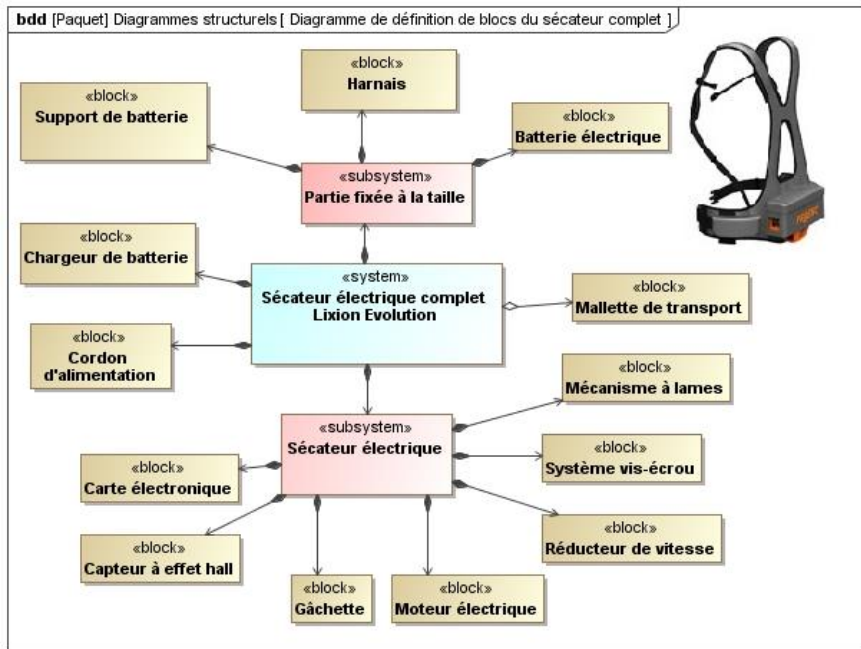


Diagramme de blocs internes du sécateur seul :

