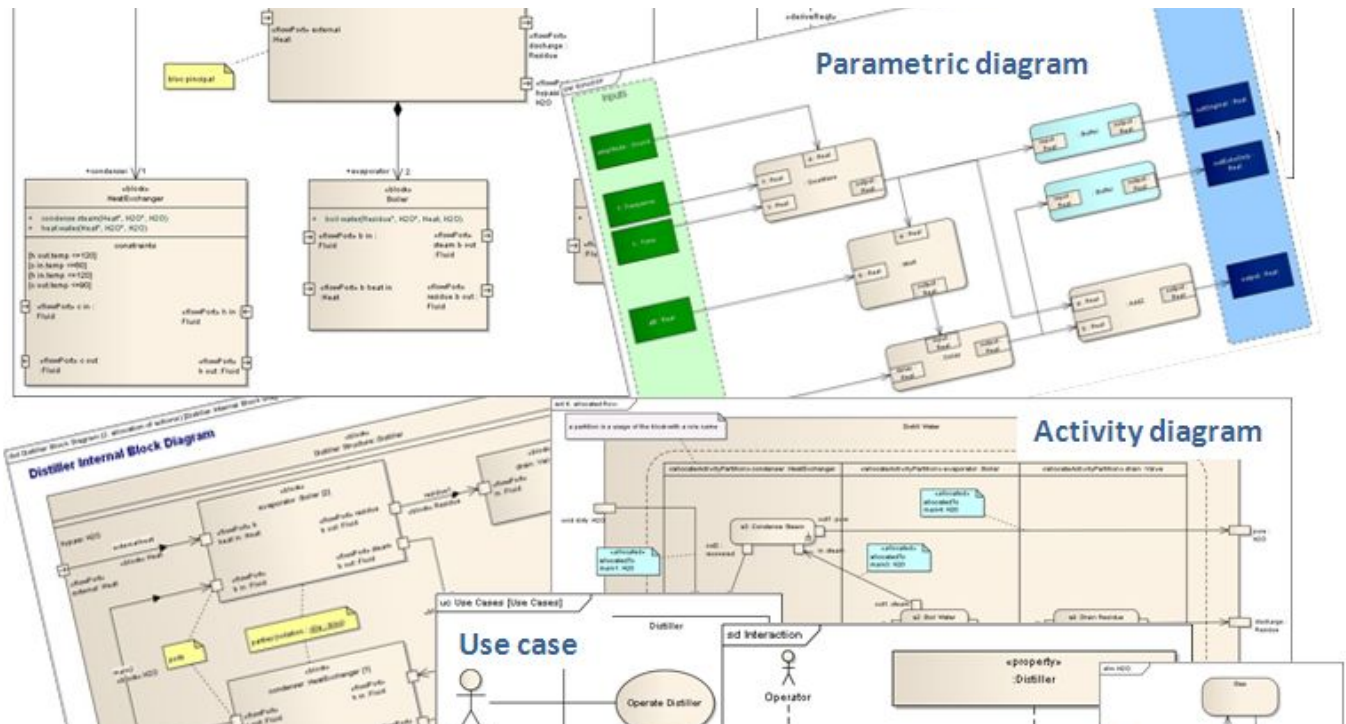


INGÉNIERIE SYSTÈME



Compétences visées:

- A1-01** Décrire le besoin et les exigences.
- A1-02** Traduire un besoin fonctionnel en exigences.
- A1-03** Définir les domaines d'application et les critères technico-économiques et environnementaux.
- A1-04** Qualifier et quantifier les exigences.
- A2-02** Définir les éléments influents du milieu extérieur.
- A2-03** Identifier la nature des flux échangés traversant la frontière d'étude.
- A3-01** Associer les fonctions aux constituants.
- A3-04** Identifier et décrire les liens entre les chaînes fonctionnelles.
- A3-03** Identifier et décrire les chaînes fonctionnelles du système.
- A3-05** Caractériser un constituant de la chaîne de puissance.
- A3-06** Caractériser un constituant de la chaîne d'information.
- E1-02** Distinguer les différents types de documents et de données en fonction de leurs usages.
- E1-05** Lire et décoder un document technique.



Table des matières

1	Système	3
1.1	Présentation générale	3
1.2	Cycle de vie d'un produit	4
2	Démarche d'ingénierie système	5
2.1	Analyse du besoin	6
2.1.1	Définir l'objectif du système et ses frontières	7
2.1.2	Expliciter et analyser les besoins de parties prenantes	7
2.1.3	Élaborer les exigences des parties prenantes	8
3	Définition d'une architecture fonctionnelle	9
3.1	Chaîne d'information et chaîne de puissance	9
3.2	Fonctions	10
3.2.1	Définir la décomposition fonctionnelle	10
3.2.2	Définir les interfaces fonctionnelles	10
4	Définition d'une architecture organique	11
4.1	Définition	11
4.2	Allocations des fonctions aux composants	11
4.3	Définir la décomposition organique	11
4.4	Vérification du cahier des charges	12
5	Synthèse sur le formalisme SysML	13
5.1	Les connecteurs	13
5.2	Le diagramme d'exigences	14
5.3	Le diagramme des cas d'utilisation	15
5.4	Le diagramme de séquence (hors programme)	15
5.5	Le diagramme de définition de blocs	16
5.6	Le diagramme de blocs internes	16



Objectif

Analyser grâce à l'outil SysML l'utilisation d'un drone de loisir.

1 Système

1.1 Présentation générale

Définition *Système*

Un système est un ensemble d'éléments en interaction de manière à former une entité pour atteindre un/des objectifs définis et quantifiables en termes de performance.

Deux grandes catégories de systèmes sont distinguées :

- les systèmes naturels ;
- les systèmes artificiels, créés par l'Homme.

Exemple : *Systèmes "généraux"*

- une machine / système transformant l'énergie
- un mécanisme
- un moteur / machine fournissant de l'énergie mécanique à partir d'une autre énergie en entrée

Exemple : *Systèmes spécifiques*

- la cordeuse à raquette de tennis (laboratoire)
- le pilote automatique de bateau (laboratoire)
- un smartphone

L'étude d'un système se fait en considérant son **milieu environnant**.

Définition *Milieu environnant*

Le milieu environnant correspond aux éléments en relation avec le système, l'ensemble étant défini par une frontière (fictive) d'étude.

Un système agit sur une **matière d'oeuvre**.

Définition Matière d'oeuvre

La matière d'oeuvre est l'élément modifié par l'intermédiaire du système.

Exemple : Machine à café

- Matière d'oeuvre « entrante » : eau froide + café moulu
- Matière d'oeuvre « sortante » : café chaud

On parle alors de **valeur ajoutée**, ici, dans l'exemple, chauffer l'eau et infuser le café.

Définition Valeur ajoutée

La valeur ajoutée est la modification, par le système, de la matière d'oeuvre. Cette valeur ajoutée, quantifiable, est un indicateur des performances du système.

Dans le cadre de la CPGE, l'accent est porté sur trois aspects du système :

- le système souhaité, défini par un ensemble de documents dont le cahier des charges ;
- le système simulé, défini numériquement par un ou des fichiers exécutables par un ou des logiciels de simulation ;
- le système réel, dont une version peut-être disponible physiquement ou virtuellement dans le laboratoire.

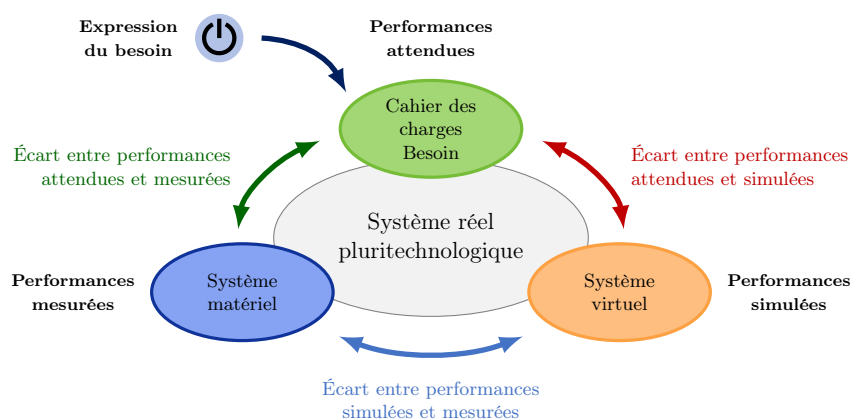


FIGURE 1 – Définition des écarts

Il est attendu de comparer continuellement ces trois aspects : quantifier les écarts entre eux et les attentes initiales du cahier des charges.

1.2 Cycle de vie d'un produit

La modélisation du cycle de vie est essentielle lors de la conception d'un système. En effet, beaucoup d'études ne se contentent que d'analyser la phase de vie d'utilisation, néanmoins cela peut provoquer des impacts importants sur les autres phases.

Définition Cycle de vie

Le cycle de vie correspond aux phases consécutives et liées d'un système de produits, de l'acquisition des matières premières ou de

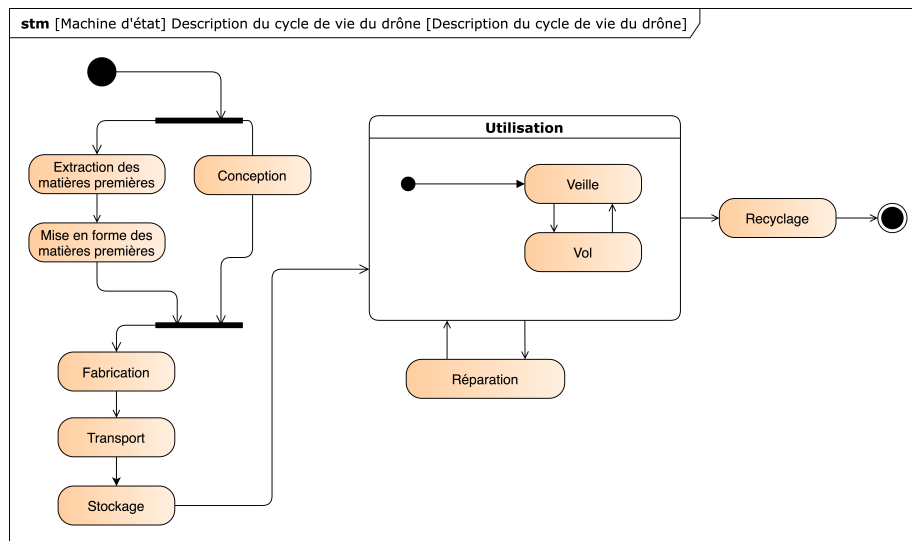


FIGURE 2 – Cycle de vie d'un produit



la génération des ressources naturelles à l'élimination finale.

Exemple : avec le langage SysML



2 Démarche d'ingénierie système

Définition Démarche d'ingénierie système

L'ingénierie système est une démarche méthodologique générale qui englobe des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes.

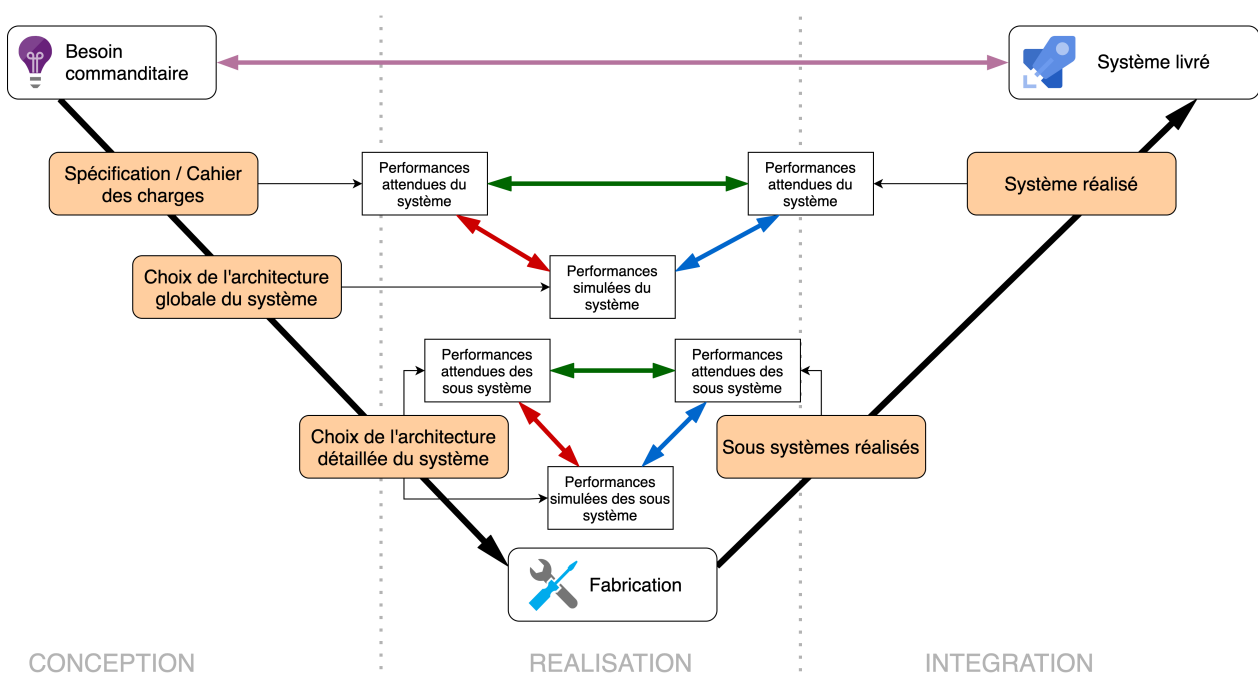


FIGURE 3 – Démarche ingénieur

2.1 Analyse du besoin

Il est important de ne pas négliger cette partie afin que l'ensemble des parties prenantes aient la même compréhension du système qui va être conçu. L'enjeu est d'éviter que le système conçu soit différent de celui souhaité.

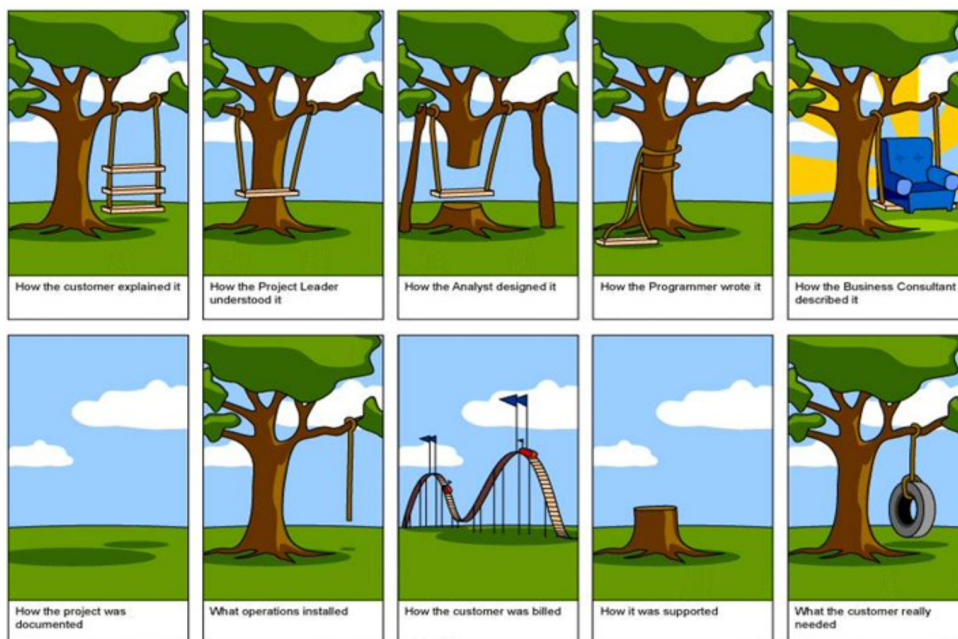


FIGURE 4 – Analyse du besoin

Définition *Exigence*

Une exigence prescrit ce qu'un produit doit faire, avec quelles performances et sous quelles conditions pour atteindre un but donné.

Définition *Cahier des charges fonctionnel (CDCF)*

Le CDCF est un document formalisant l'expression des besoins et facilitant la communication entre les parties prenantes.

Définition *Besoin*

Le besoin est une nécessité ou un désir ressenti par un utilisateur

Définition *Service*

Un service est un résultat généré par des fonctions à l'interface entre le système et l'environnement d'utilisation, et par des fonctions internes au système, pour répondre aux besoins de l'utilisateur.

Définition *Partie prenante*

Partie ayant un droit, une part ou une prérogative qui fait que le système ou certaines de ses propriétés doivent satisfaire les besoin ou les attentes de cette partie.

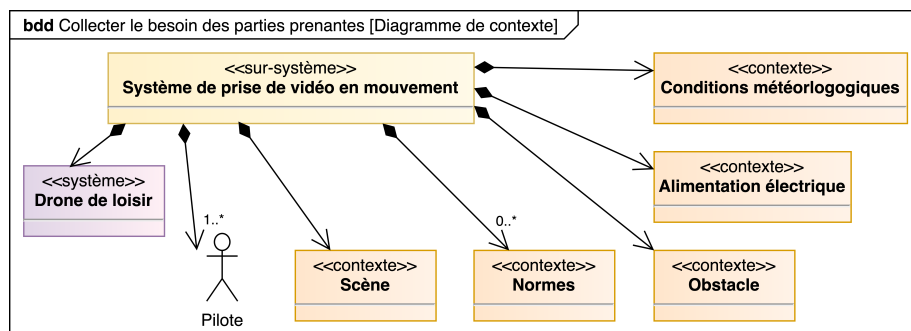
2.1.1 Définir l'objectif du système et ses frontières

Afin de fournir un cahier des charges fonctionnel sur lequel tout le monde est d'accord, il faut définir le contexte opérationnel du système. Tout d'abord, la mission du système doit être définie. Cela peut se résumer en une phrase définissant l'intérêt global du système.

Ensuite, il est possible de dresser le cycle de vie attendu du système en n'oubliant pas les phases de maintenance et de retrait à l'aide d'un diagramme d'état comme cela a été mentionné précédemment.

Enfin, pour chaque phase du cycle de vie, il est possible de déterminer tous les interacteurs dans un diagramme de contexte. Cela se fait par un diagramme de blocs en SysML. Celui-ci présente les différents éléments composant le sursystème du système (contexte d'utilisation) pour une phase de vie.

Exemple : *Diagramme de contexte du système de prise de vue d'un drone de loisir*



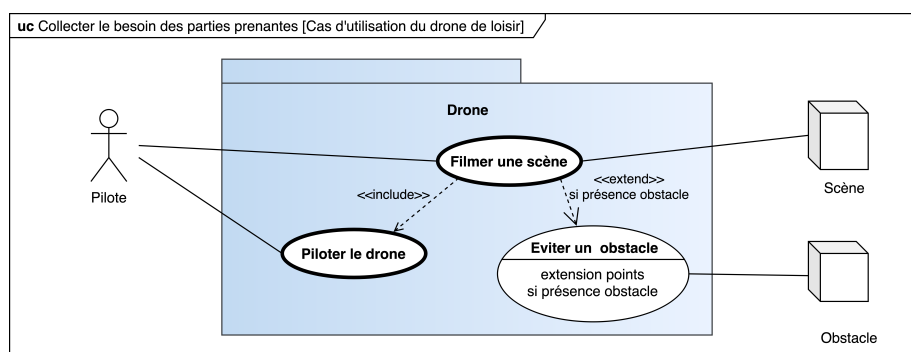
Ainsi, pour le drone de loisir dans le contexte de prise de vidéos de loisir en mouvement, plusieurs interacteurs sont présents : au moins un pilote, une scène, des normes, possiblement un ou plusieurs obstacles, une alimentation électrique et des conditions météorologiques.

2.1.2 Expliciter et analyser les besoins de parties prenantes

Il est donc maintenant possible d'éliciter (ie : comprendre et de modéliser) les besoins des parties prenantes et de les analyser. Pour cela, il faut déterminer les services attendus en se basant sur les différents acteurs pour chaque phase du cycle de vie. Cela va ainsi permettre que les différentes parties prenantes soient d'accord sur le système à concevoir.

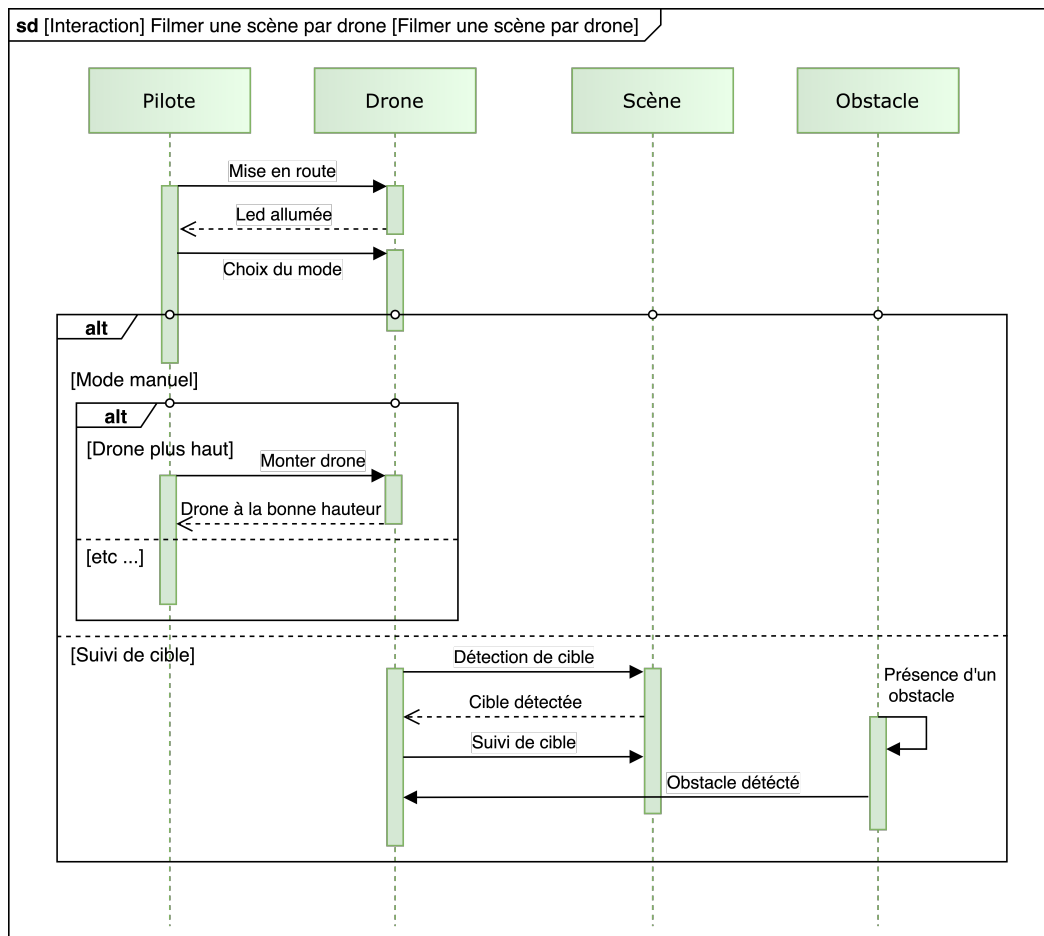
Cela se fait à l'aide d'un diagramme SysML nommé UseCase Diagram (Diagramme des Cas d'utilisation). Il est possible d'en dresser un global ou de les raffiner en en réalisant un pour chaque phase du cycle de vie du système.

Exemple : *Diagramme des cas d'utilisation du système de prise de vue d'un drone de loisir*



Ensuite pour chaque cas d'utilisation il est possible de déterminer un diagramme de séquence, permettant d'établir les relations entre le système et les inter-acteurs.

Exemple : *Diagramme de séquence concernant l'action de filmer par un drone*

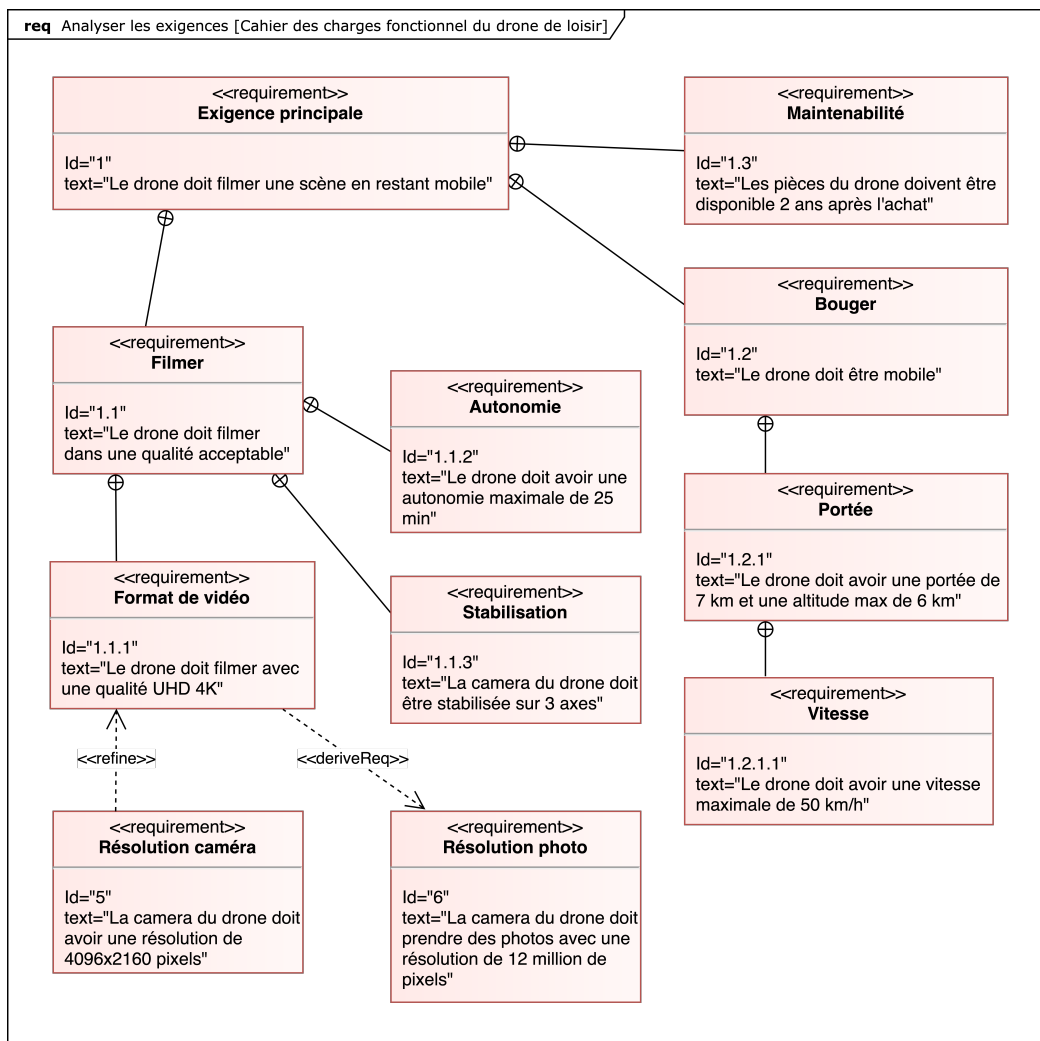


2.1.3 Élaborer les exigences des parties prenantes

Maintenant que le contexte opérationnel et les services attendus ont été définis, il est possible de rédiger l'ensemble des exigences des parties prenantes dans le cahier des charges fonctionnel.

Pour synthétiser ce travail, on peut utiliser un diagramme des exigences (req) en SysML.

Exemple : *Diagramme des exigences partiel d'un drone de loisir*



3 Définition d’une architecture fonctionnelle

Une fois que le cahier des charges fonctionnel est bien défini, il est important de définir quelles seront les fonctions qui vont y répondre.

Définition *Fonction*

Une fonction est une tâche ou activité exécutée pour atteindre un résultat attendu.

Définition *Interface*

Une interface est une frontière entre 2 fonctions, définie par plusieurs caractéristiques concernant les fonctions et les flux.

3.1 Chaîne d’information et chaîne de puissance

Afin de faciliter la conception des systèmes, plusieurs patrons de conception existent dont celui de la chaîne d’information et de la chaîne de puissance (instantanée).



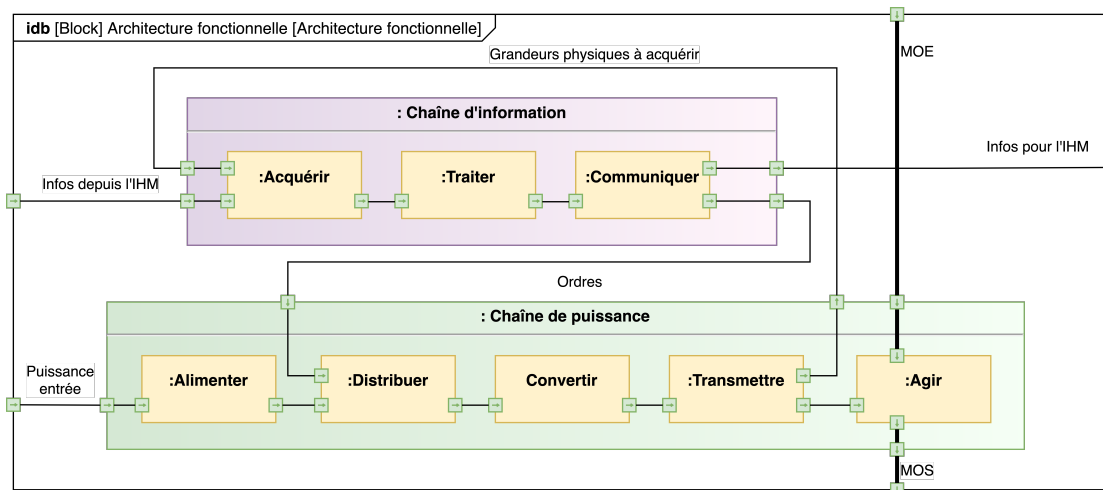


FIGURE 5 – Chaîne d’information et chaîne de puissance sous forme d’IBD

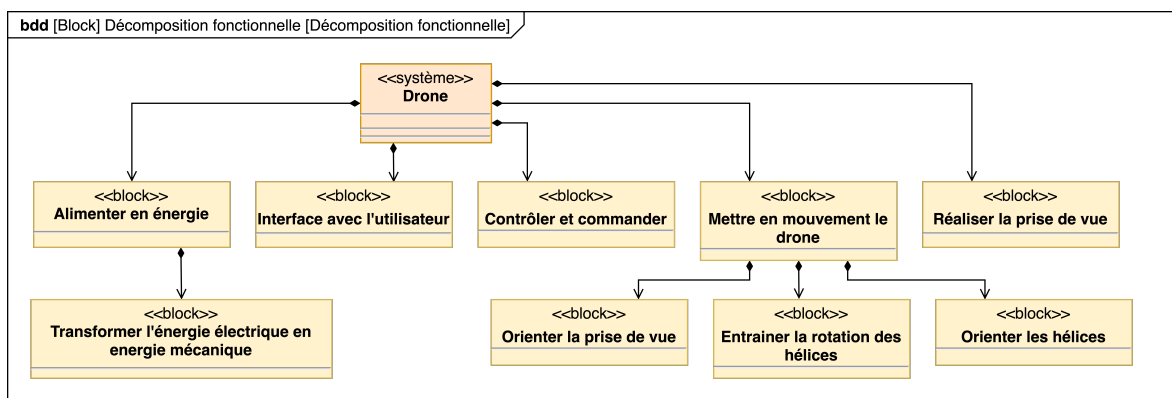
Ce patron de conception permet de déterminer les principales fonctions que peut contenir un système. Bien entendu, selon la typologie du système, certaines fonctions peuvent ne pas exister et d’autres peuvent être démultipliées.

3.2 Fonctions

3.2.1 Définir la décomposition fonctionnelle

A l’aide du cahier des charges et du patron de conception, il est possible de déterminer une décomposition fonctionnelle du système souhaité.

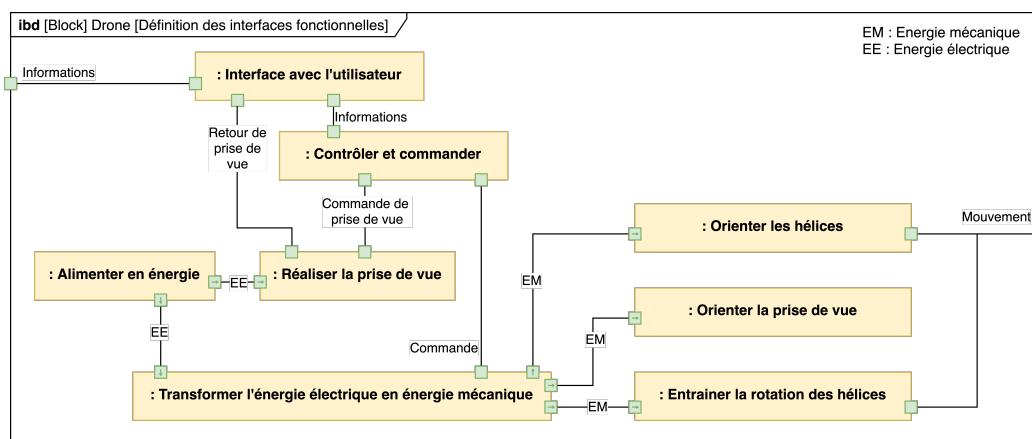
Exemple : *Décomposition fonctionnelle d’un drone de loisir*



3.2.2 Définir les interfaces fonctionnelles

Après avoir déterminé les différentes fonctions, il est possible de définir comment les flux transitent d’une fonction à l’autre. Ces flux peuvent être de matière, d’énergie ou d’information.

Exemple : Interfaces fonctionnelles d'un drone de loisir



Cette architecture fonctionnelle va permettre de chercher des composants existants permettant de réaliser cette fonction ou d'en créer des nouveaux, ce qui sera l'intérêt de l'activité suivante. De plus, par les interfaces, il est possible d'avoir des critères de fonctionnement.

4 Définition d'une architecture organique

Maintenant que l'architecture fonctionnelle est définie, il est possible d'allouer les fonctions à des composants.

4.1 Définition

 **Définition** *Composant*

Un composant est défini comme étant une entité à structure discrète, telle qu'un module d'assemblage ou de logiciel, au sein d'un système considéré à un niveau d'analyse particulier.

4.2 Allocations des fonctions aux composants

L'activité principale concernant l'architecture organique consiste en l'allocation des éléments fonctionnels aux différents composants de l'architecture organique qui peuvent être nouveaux, sur étagère d'un fournisseur ou repris d'un modèle existant.

Exemple : Table d'allocation partielle

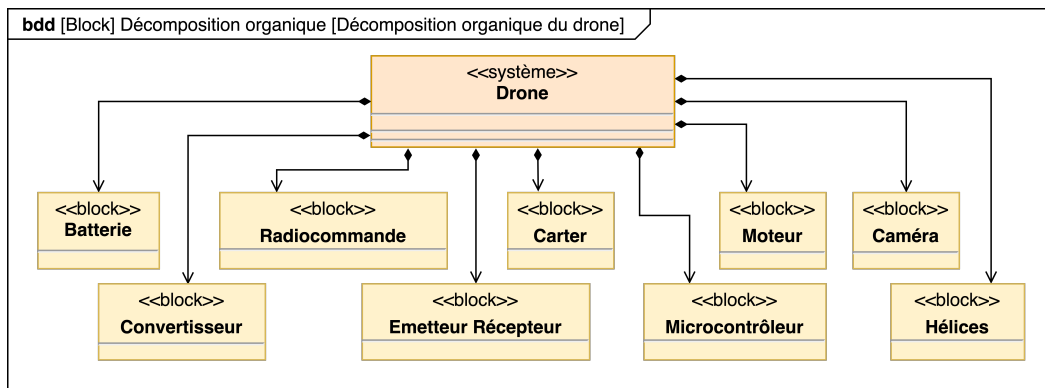
	Caméra	Moteur	Hélices
Mettre en mouvement le drone			
Réaliser la prise de vue			
...			

4.3 Définir la décomposition organique

Après s'être assuré de la présence d'un composant pour chaque fonction, il est possible de déterminer la décomposition organique du système

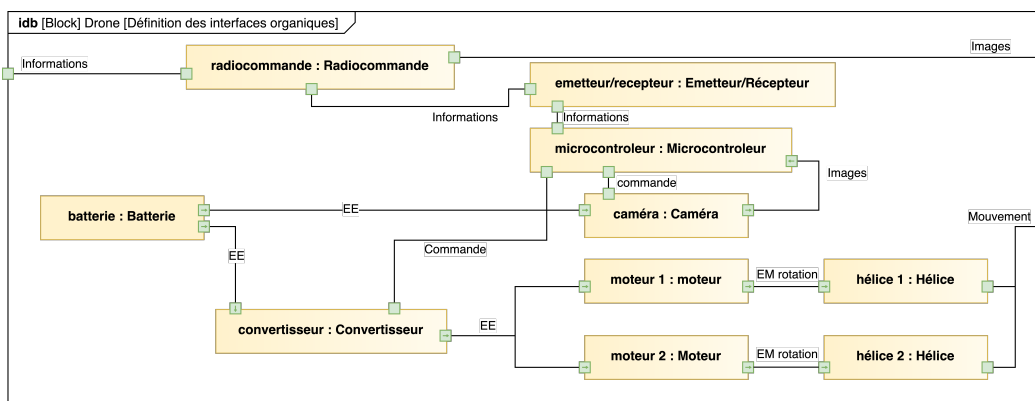


Exemple : *Décomposition organique d'un drone de loisir*



Similairement à celles fonctionnelles, il faut définir les interfaces organiques entre les différents composants

Exemple : *Interfaces organiques d'un drone de loisir*



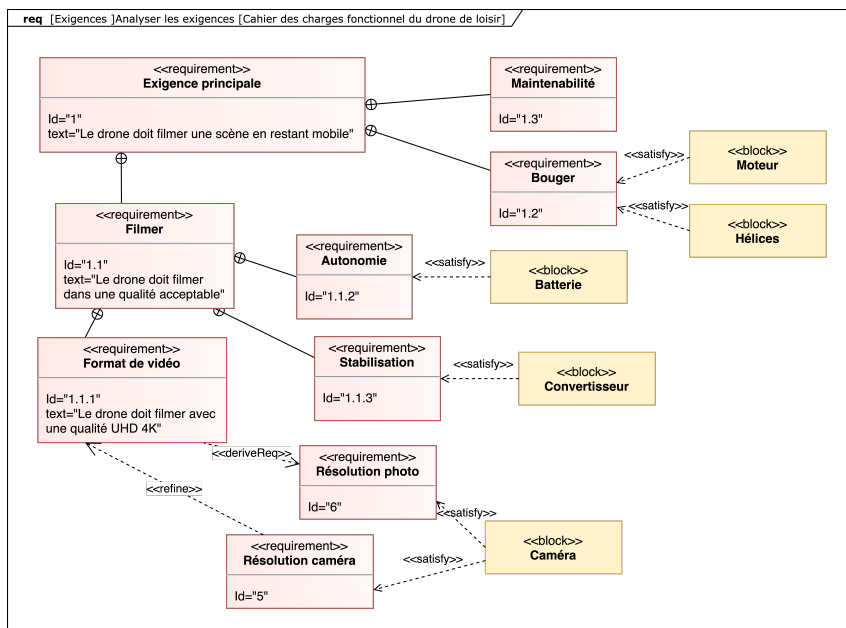
Cette activité peut paraître redondante à celle précédente mais n'avez-vous jamais eu du mal à trouver le bon connecteur pour votre smartphone pour le recharger ?

Les différents smartphones ont le même besoin fonctionnel (énergie électrique continue) mais selon les composants utilisés, l'interface organique entre le smartphone et le câble de recharge peut être différente.

4.4 Vérification du cahier des charges

Pour conclure, il est intéressant de vérifier que le produit est conforme aux exigences du cahier des charges fonctionnel.

Exemple : *Vérification partielle des exigences du drone*



5 Synthèse sur le formalisme SysML

SysML est un ensemble de diagrammes permettant de décrire les différentes activités. Il s’agit en quelque sorte d’une boîte à outils pour l’ingénierie Système.

Ensemble des diagrammes SysML

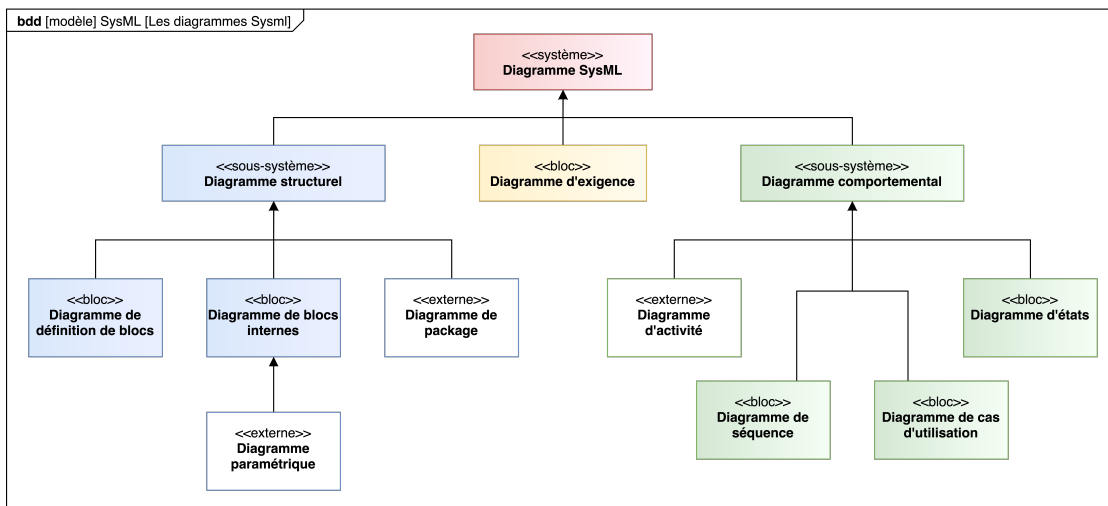


FIGURE 6 – Ensemble des diagrammes SysML

5.1 Les connecteurs

Les principales relations entre les blocs sont résumés ci-dessous, ils permettent de définir le lien entre 2 blocs.

- > **Association** : relation d'égal à égal entre deux éléments **A utilise B**
-> **Dépendance** : 2 items distincts mais dont l'un dépend de l'autre **A dépend de B**



—◇	Agrégation : un élément est une composante facultative de l'autre	A compose B sans être indispensable
—◆	Composition : un élément est une composante obligatoire de l'autre	A compose B et lui est indispensable
—▷	Généralisation : dépendance de type « filiation » entre 2 items	A est une sorte de B
—⊕	Conteneur : relation d'inclusion entre 2 items	B contient A

5.2 Le diagramme d'exigences

En anglais : **requirement diagram** Notation SysML : **req**

C'est un diagramme fonctionnel, qui décrit les exigences du cahier des charges fonctionnel. Une exigence exprime une capacité ou une contrainte à satisfaire par un système. Elle peut exprimer une fonction que devra réaliser le système ou une condition de performance technique, physique, de sécurité, de fiabilité, d'ergonomie, d'esthétisme...

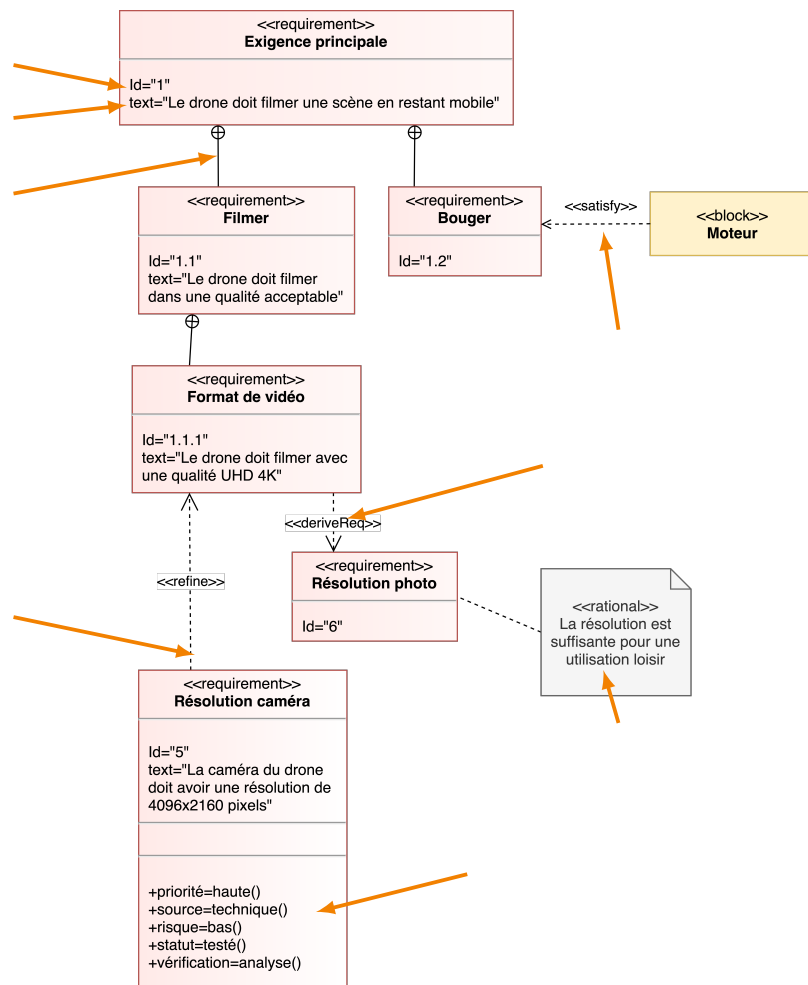


FIGURE 7 – Structure d'un diagramme d'exigence

Les propriétés peuvent être de différentes natures définies ci-dessous :

- + priorité (haute, moyenne, basse)
- + source (client, marketing, technique, législation, etc.)
- + risque (haut, moyen, bas)
- + statut (proposée, validée, implémentée, testée, livrée, etc.)



+ méthode de vérification (analyse, démonstration, test, etc.)

La gestion des exigences est l'ensemble des activités permettant de définir et de suivre les exigences d'un système au cours d'un projet. Elle permet :

- de s'assurer de la cohérence entre ce que fait réellement le projet et ce qu'il doit faire ;
- 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 :

- exigence – élément comportemental (cas d'utilisation, diagramme d'états, etc.) : « **refine** » ;
- exigence – bloc d'architecture : « **satisfy** » ;
- exigence – cas de test (méthode de vérification) : « **verify** ».

5.3 Le diagramme des cas d'utilisation

En anglais : **use case diagram** Notation SysML : **uc**

C'est un diagramme fonctionnel qui 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. Il exprime les services (use cases) offerts par le système aux utilisateurs (actors).

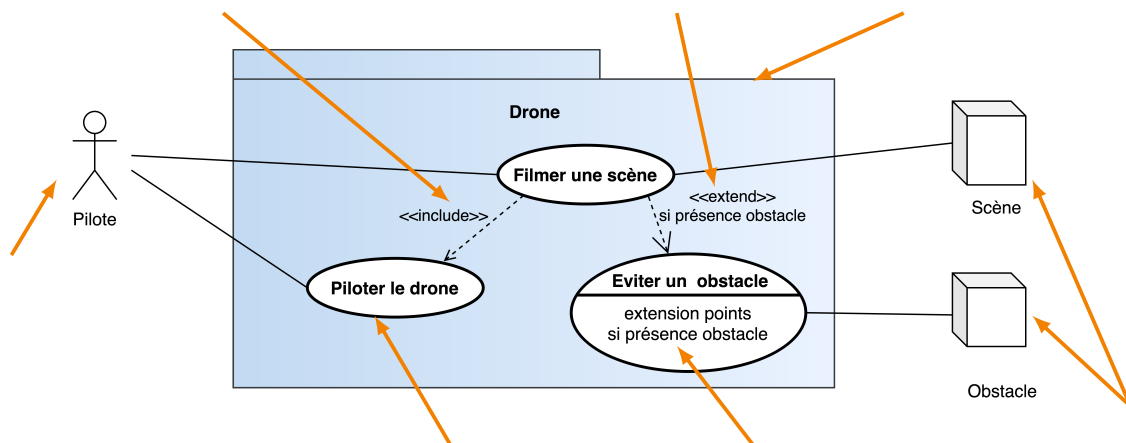


FIGURE 8 – Structure d'un diagramme des cas d'utilisation

5.4 Le diagramme de sequence (hors programme)

En anglais : **sequence diagram** Notation SysML : **sd**

C'est un diagramme comportemental qui représente les échanges de messages entre les acteurs et le système ou entre des parties durant une séquence temporelle d'actions appelée scénario.

Il permet de représenter la dynamique d'un cas d'utilisation ou la collaboration d'un ensemble d'objets internes au système.

Les principaux opérateurs sont :

- **loop** : boucle. Le fragment ou autre cas d'utilisation peut s'exécuter plusieurs fois, et la condition de garde (voir « Diagramme d'état ») explicite l'itération
- **opt** : optionnel. Le fragment ou autre cas d'utilisation ne s'exécute que si la condition fournie est vraie
- **alt** : fragments alternatifs. Seul le fragment possédant la condition vraie s'exécutera par : parallèle. Il est utilisé pour représenter des interactions ayant lieu en parallèle

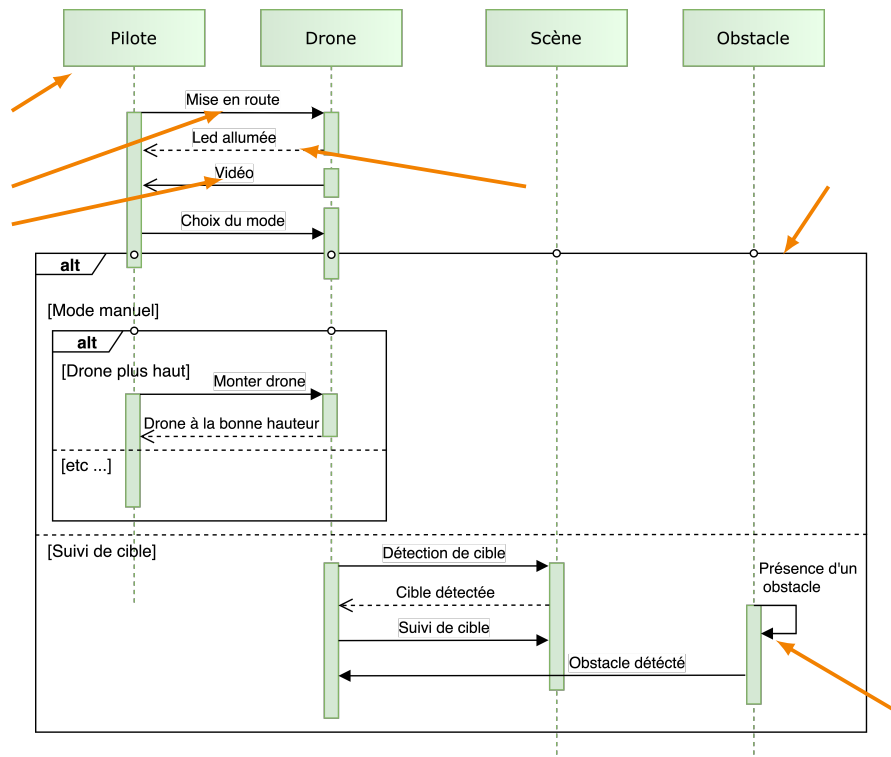


FIGURE 9 – Structure d’un diagramme des cas d’utilisation

5.5 Le diagramme de définition de blocs

En anglais : **block definition diagram** Notation SysML : **bdd**

C’est un diagramme structurel qui montre les briques statiques : blocs, composition, associations... Il est utilisé pour décrire l’architecture matérielle du système. Un bloc est une entité bien délimitée qui encapsule principalement des attributs (variables d’état), des opérations (procédures comportementales), des contraintes, des ports (échange de flux avec l’extérieur) et des parts (sous-blocs internes). Un bloc peut modéliser tout le système, un élément matériel ou logiciel.

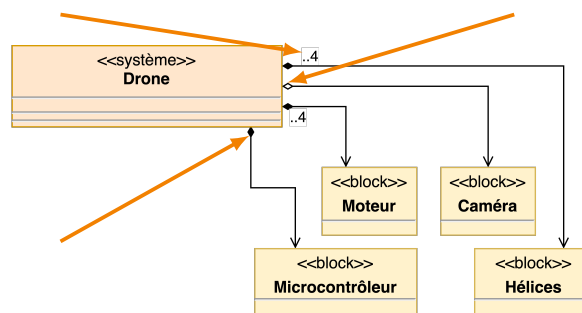


FIGURE 10 – Structure d’un diagramme des cas d’utilisation

5.6 Le diagramme de blocs internes

En anglais : **internal block diagram** Notation SysML : **ibd**

C’est un diagramme structurel, qui est utilisé pour décrire l’architecture matérielle du système. Il montre l’organisation interne d’un élément du système.

Il représente les instances des parts d'un bloc (objets). L'IBD est cadré à l'intérieur des frontières du bloc concerné.

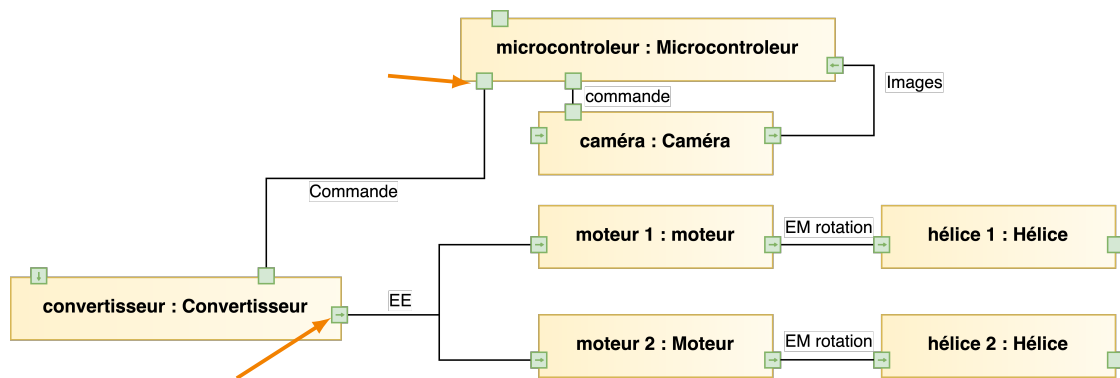


FIGURE 11 – Structure d'un diagramme des cas d'utilisation

Les circulations de flux se font entre les ports de différentes natures :

- **flow port** : flux de nature physique, à des données ou à de l'énergie. Il peut être atomique, c'est-à-dire à un seul sens, ou composite, à double sens
- **standard** : flux de nature logiciel ou logiques.

La nature des flux est bien souvent indiqué sur les lien entre les port. Il n'est pas permit de relier des ports de noture différentes.