

Comprendre un cahier des charges

De nos jours, les systèmes sont de plus en plus complexes, le SysML (Systems Modeling Language) est un langage de modélisation permettant une étude plus facile de ces systèmes en structurant la description et l'analyse de ces derniers.

Le SysML est composé de **9** diagrammes : **5** diagrammes comportementaux et **4** diagrammes structurels. Dans ce chapitre, nous allons voir les plus utilisés dans le programme de première de sciences de l'ingénieur à l'aide de la voiture radiocommandée Tamiya.



● Systems Modeling Language (SysML)

Les diagrammes SysML sont toujours composés d'une cartouche permettant de définir le nom du diagramme ainsi que le nom du système modélisé.

req - Exigences de propulsion

Contexte général

Le diagramme de contexte (**context diagram : cd**) définit la frontière de l'objet et spécifie les acteurs et éléments environnants.

Ce diagramme ne fait pas partie des diagrammes normalisés dans le langage SysML mais on le retrouve généralement pour contextualiser le système.

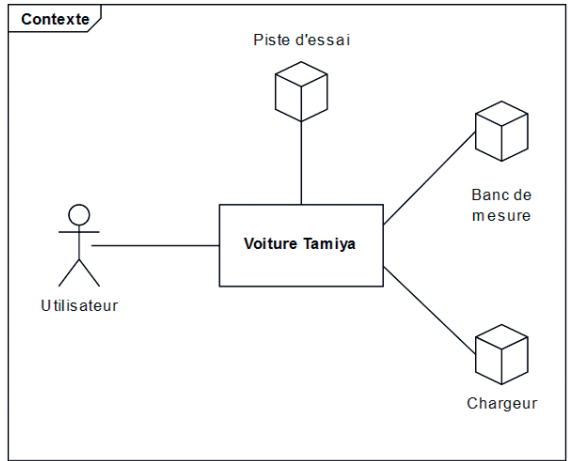
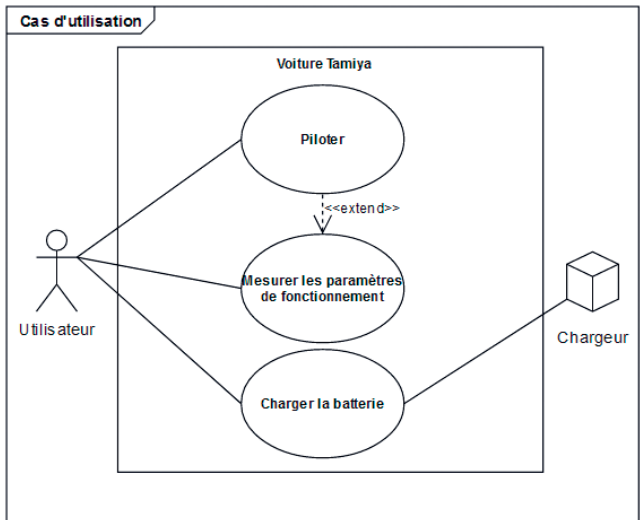


Diagramme cas d'utilisation

Le diagramme de cas d'utilisation (**use case diagram : uc**) permet de mettre en relation les fonctionnalités avec les acteurs à qui il rend service.



Dans le diagramme de cas d'utilisation, nous retrouvons les acteurs principaux à gauche et les acteurs secondaires (ou les éléments en interaction) avec le système à droite.

On rencontre dans ce diagramme deux types de flèches : les flèches « include » et les flèches « extend ».

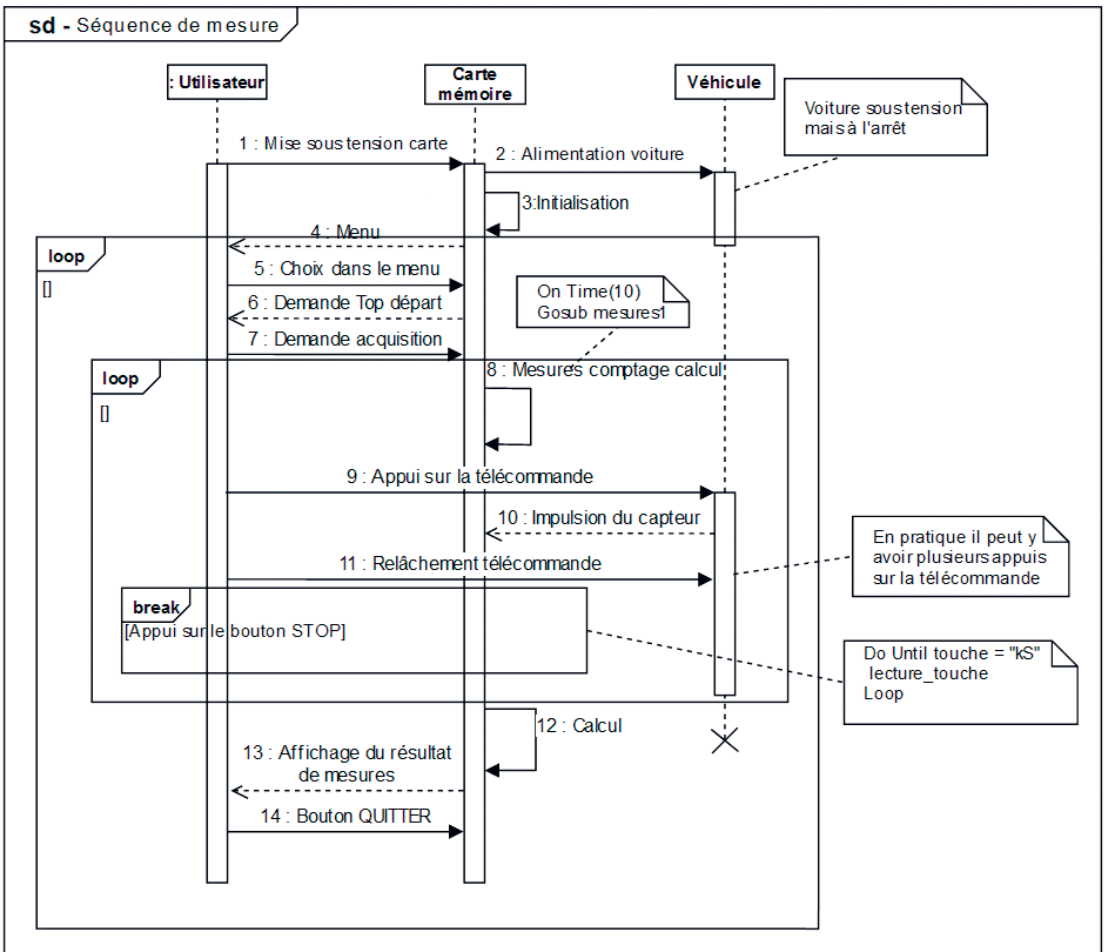


La flèche « **include** » va indiquer que la fonction pointée est indispensable au bon fonctionnement.

La flèche « **extend** » va indiquer que la fonction peut être réalisée mais qu'elle n'est pas obligatoire.

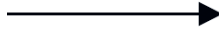
Diagramme de séquence

Le diagramme de séquence (**s**equen**d**iagram : **sd**) décrit la chronologie des interactions entre les éléments du système ou entre le système et l'extérieur.

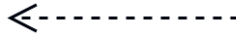


Dans un diagramme de séquence, on retrouve quatre types de messages :

- **Le message synchrone** : Il met en relation différentes lignes de vie (lignes en pointillé, surmontées d'un rectangle). C'est un message qui attend une réponse de la part du destinataire. Tant qu'il n'y a pas de réponse, l'émetteur est bloqué. Ce message est représenté par une flèche à tête pleine.



- **Le message de réponse** : Il répond à un message synchrone et libère le blocage de son destinataire (émetteur du message synchrone). Ce message est asynchrone car il n'attend pas de réponse, il est représenté par une flèche en pointillé.



- **Le message asynchrone** : L'émetteur envoie un message sans attendre de réponse. Il est représenté par une flèche en trait plein.

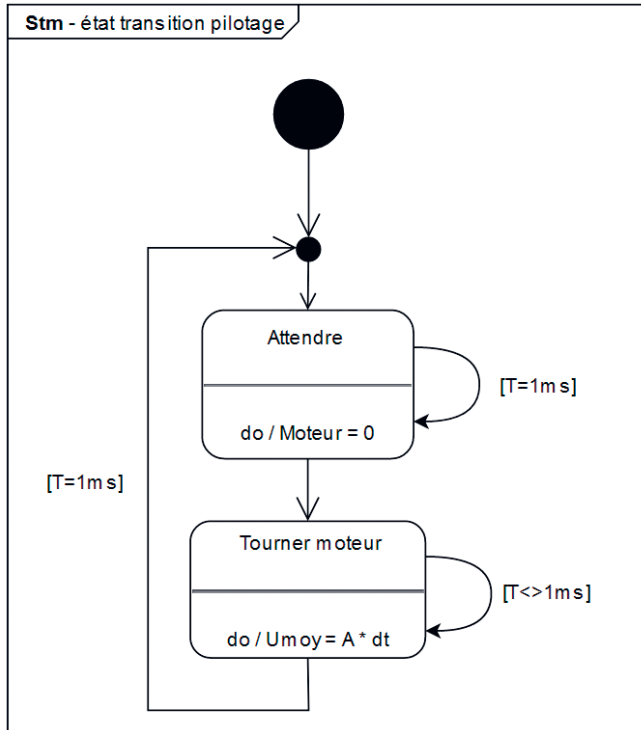


- **Le message réflexif** : L'émetteur et le récepteur sont identiques, cela permet de représenter une opération interne. Il est représenté par une flèche bouclée avec une tête pleine.



Diagramme d'état-transition

Le diagramme d'état-transition (**state machine diagram : stm**) représente les différents états que peuvent prendre le système et les transitions lors des changements d'états. Chaque état correspond à une séquence. C'est le diagramme d'état qui peut générer tous les scénarios possibles.



Dans un diagramme d'état-transition, les états sont représentés par des rectangles et les transitions par des flèches comme ci-dessus. Pour annoncer le début et la fin du diagramme, on retrouve les deux éléments suivants :

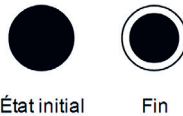
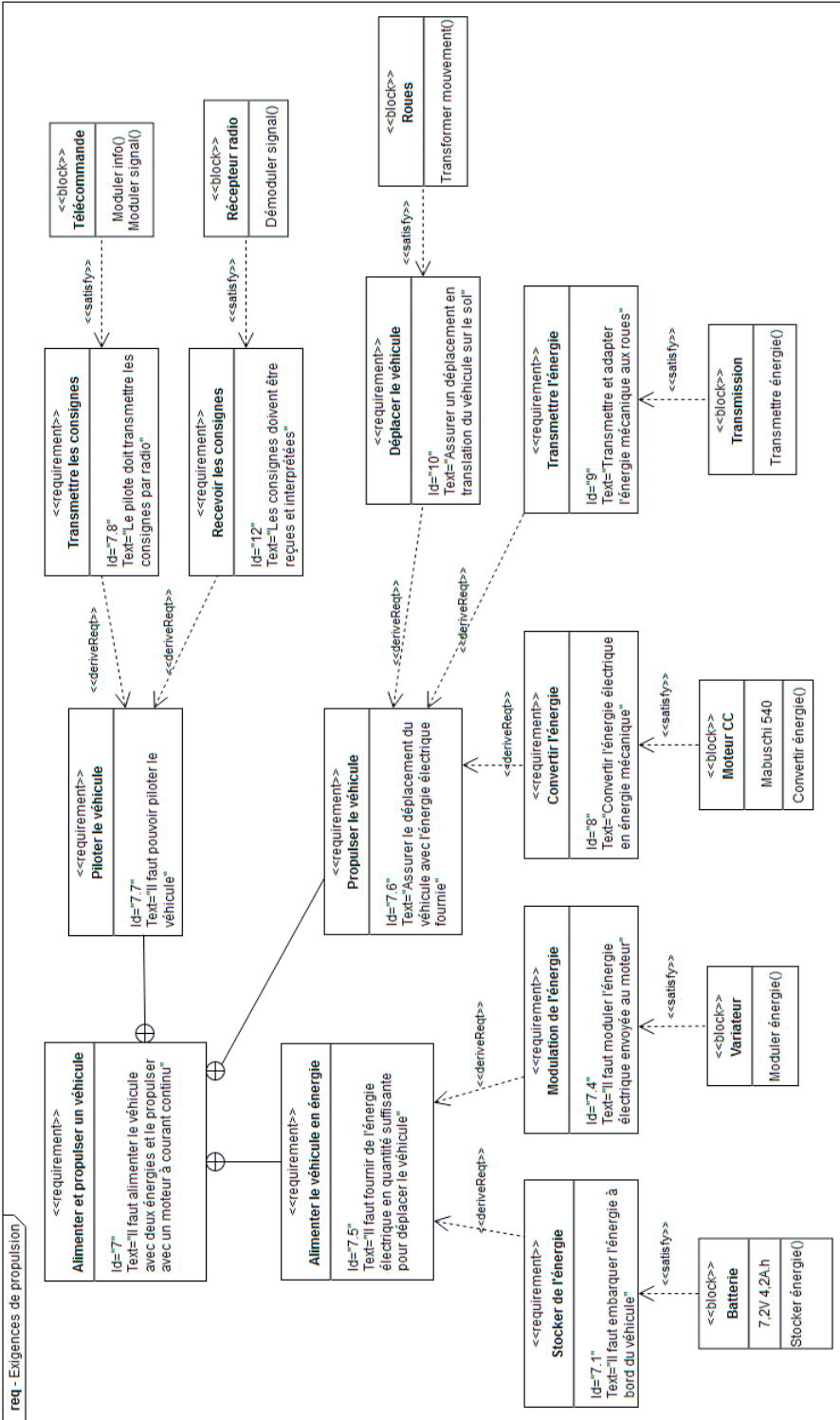


Diagramme des exigences

Le diagramme des exigences (**req**uirement diagram : **req**) permet de décrire les différentes fonctions d'un système dans lequel figureront les besoins et les services attendus.

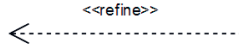


Le diagramme des exigences est composé de plusieurs liaisons :

- **Conteneur** : Indique une inclusion entre deux items.



- **Raffinement** : Ajoute des précisions sur l'élément pointé comme l'ajout de données quantitatives.



- **Dérivation** : Permet de relier des exigences de niveaux différents.



- **Satisfaction** : Répond à la demande formulée par l'exigence pointée.

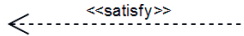
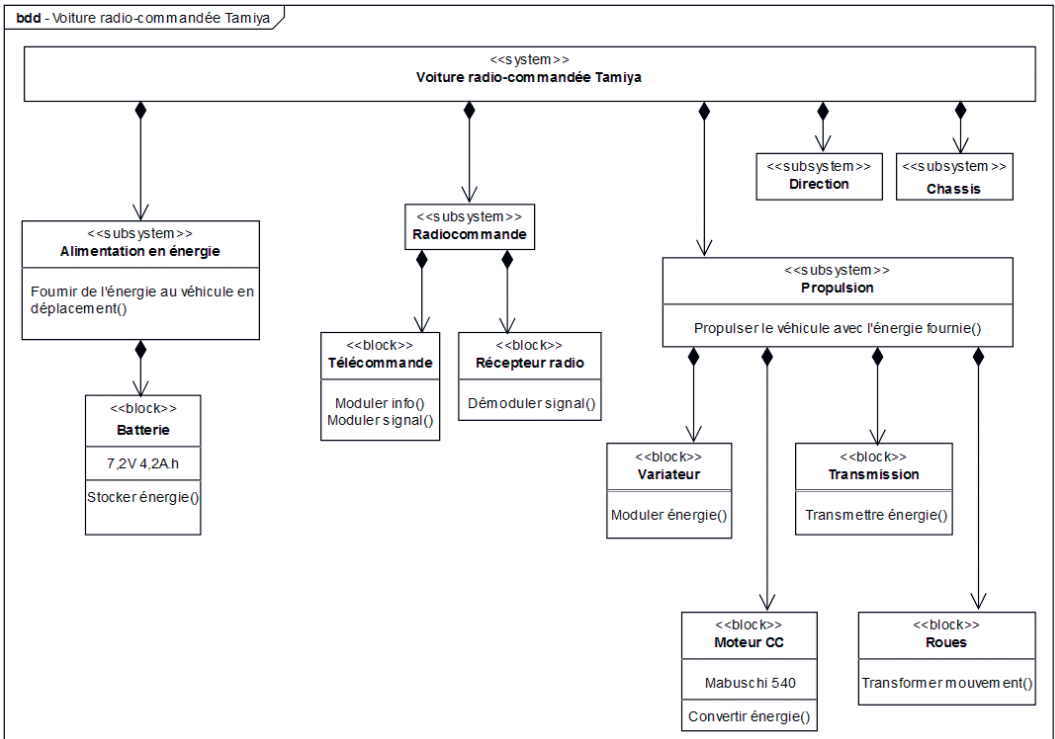


Diagramme de définition de blocs

Le diagramme de définition de blocs (**block definition diagram** : **bdd**) permet de définir l'architecture globale du système par un organigramme de blocs. Il est possible de faire apparaître les propriétés de chaque bloc.



Dans le diagramme de définition de blocs, nous retrouvons deux liaisons, l'**agrégation** et la **composition**.

■ **Agrégation** : Indique que le composant est facultatif.



■ **Composition** : Indique que le composant est obligatoire.



Diagramme de bloc interne

Le diagramme de bloc interne (**i**nternal **b**lock **d**iagram : **ibd**) décrit l'architecture matérielle de la structure interne d'un bloc issu du **bdd** en faisant apparaître ses composants et les échanges de flux d'énergie ou d'information. Dans le diagramme **ibd** suivant, nous retrouvons la chaîne d'information et d'énergie en version SysML de la voiture Tamiya.