

# CHAPITRE 1

---

## ÉTUDE DES SYSTÈMES

## 1.1 Besoin, produit, système

### 1.1.1 Le besoin

Tout produit, tout système naît de la volonté de satisfaire un besoin. Ce besoin peut être un simple rêve, une envie ou la réponse à un problème.

La norme NF X50 - 150 précise : « Un besoin est une nécessité, un désir éprouvé par un utilisateur ».

Au cours du temps, les besoins exprimés ont évolués, d'une simple description d'un usage :

- tondre la pelouse,
- se déplacer rapidement,
- communiquer à distance,
- mémoriser l'information,
- ...,

en imposant ensuite une contrainte de coût ou une exigence d'innovation :

- tondre la pelouse automatiquement,
- communiquer à distance la voix et l'image avec un objet portable,
- enregistrer toute ma discothèque et l'écouter n'importe où,
- ...,

À ces descriptions se rajoute maintenant une exigence d'environnement :

- la réalisation du besoin doit se faire sans déchets,
- être recyclable,
- être équitable, ....

### 1.1.2 Le produit, système

C'est une réalisation humaine, il a été imaginé et réalisé pour satisfaire le besoin. Le produit est la réponse au besoin.

Un produit n'est pas obligatoirement un objet technique, cela peut tout aussi bien être un service ou un processus.

Nous intéresserons ici uniquement aux produits techniques (industriels).

Le terme « produit » est souvent remplacé par le terme « système » qui permet une signification plus riche et permet d'élargir à d'autres champs que les produits industriels.

Avec l'évolution des besoins, les produits ont suivi la même évolution vers plus de technologie et d'automatisation intégrée.






Nettoyer le sol	Nettoyer le sol et ramasser la poussière	Nettoyer le sol, ramasser une grande quantité de poussière	Nettoyer le sol, ramasser la poussière sans sac	Nettoyer le sol automatiquement
				

FIGURE 1.1 – Évolution des produits

Le terme « produit » est souvent remplacé par le terme « système » qui permet une signification plus riche et permet d'élargir à d'autres champs que les produits industriels.

#### a) Quelques définitions de système

Nous commençons par définir ce qu'est un « système ». Nous pouvons trouver diverses définitions du système, nous en citons les suivantes :

- A « system » is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents ; that is, all things required to produce system - level results. The results include system-level qualities, properties, characteristics, functions, behavior,

and performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationship among the parts; that is, how they are interconnected. (en. System, NASA (2007)) Un système est un ensemble de composants inter reliés qui interagissent les uns avec les autres d'une manière organisée pour accomplir une finalité commune.

- (en. System, INCOSE (2010)) The systems are man-made, created and utilized to provide services in defined environments for the benefit of users and other stakeholders. These systems may be configured with one or more of the following : hardware, software, humans, processes (e.g., review process), procedures (e.g., operator instructions), facilities, and naturally occurring entities (e.g., water, organisms, minerals). In practice, they are thought of as products or services. The perception and definition of a particular system, its architecture and its system elements depend on an observer's interests and responsibilities. One person's system - of - interest can be viewed as a system element in another person's system - of -interest. Conversely, it can be viewed as being part of the environment of operation for another person's system - of - interest. Un système est un ensemble intégré d'éléments qui accomplissent un objectif défini. [INCOSE2004]

Nous nous restreindrons à caractériser les systèmes conçus par l'homme et nous les définissons par :

Un système est un ensemble de composants qui collaborent à la réalisation d'un ensemble de tâches en vue de fournir un ensemble de services, cet ensemble est soumis à un environnement donné et interagit ainsi avec un sous-ensemble des éléments de cet environnement .

## 1.2 Analyse fonctionnelle et le cahier des charges

D'après la norme AFNOR NF X 50-151, l'analyse fonctionnelle est une démarche qui consiste à rechercher, ordonner, caractériser, hiérarchiser et / ou valoriser les fonctions du produit attendu par l'utilisateur.

L'analyse fonctionnelle consiste à rechercher et à caractériser les fonctions offertes par un produit placé dans un système pour satisfaire les besoins de son utilisateur.

L'objectif de l'analyse fonctionnelle est d'établir le cahier des charges du produit

Pour établir le cahier des charges et analyser le produit nous allons utiliser un ensemble de diagrammes du langage de description SysML (Système Modeling Language).

## 1.3 SysML - première approche

### 1.3.1 Les diagrammes

SysML est un langage d'étude et de description des systèmes.

*Le langage est la capacité d'exprimer une pensée et de communiquer au moyen d'un système de signes (vocaux, gestuel, graphiques, tactiles, olfactifs, etc.) doté d'une sémantique, et le plus souvent d'une syntaxe (mais ce n'est pas systématique<sup>1</sup>). Plus couramment, le langage est un moyen de communication [wikipédia].*

Le langage SysML (<http://fr.wikipedia.org/wiki/SysML>), correspond à cette définition, c'est un langage **graphique** qui comporte 9 diagrammes.

C'est un langage basé sur **UML**<sup>1</sup>, un langage de description des projets informatiques, il en reprend la symbolique et la syntaxe en ajoutant des diagrammes spécifiques à la description et l'analyse des système et en supprimant ceux dédiés aux projets informatiques.

Le langage de description UML est utilisé par les plus grandes les sociétés informatiques, plusieurs logiciels dédié à l'UML avec des plugins SysML existent<sup>2</sup>.

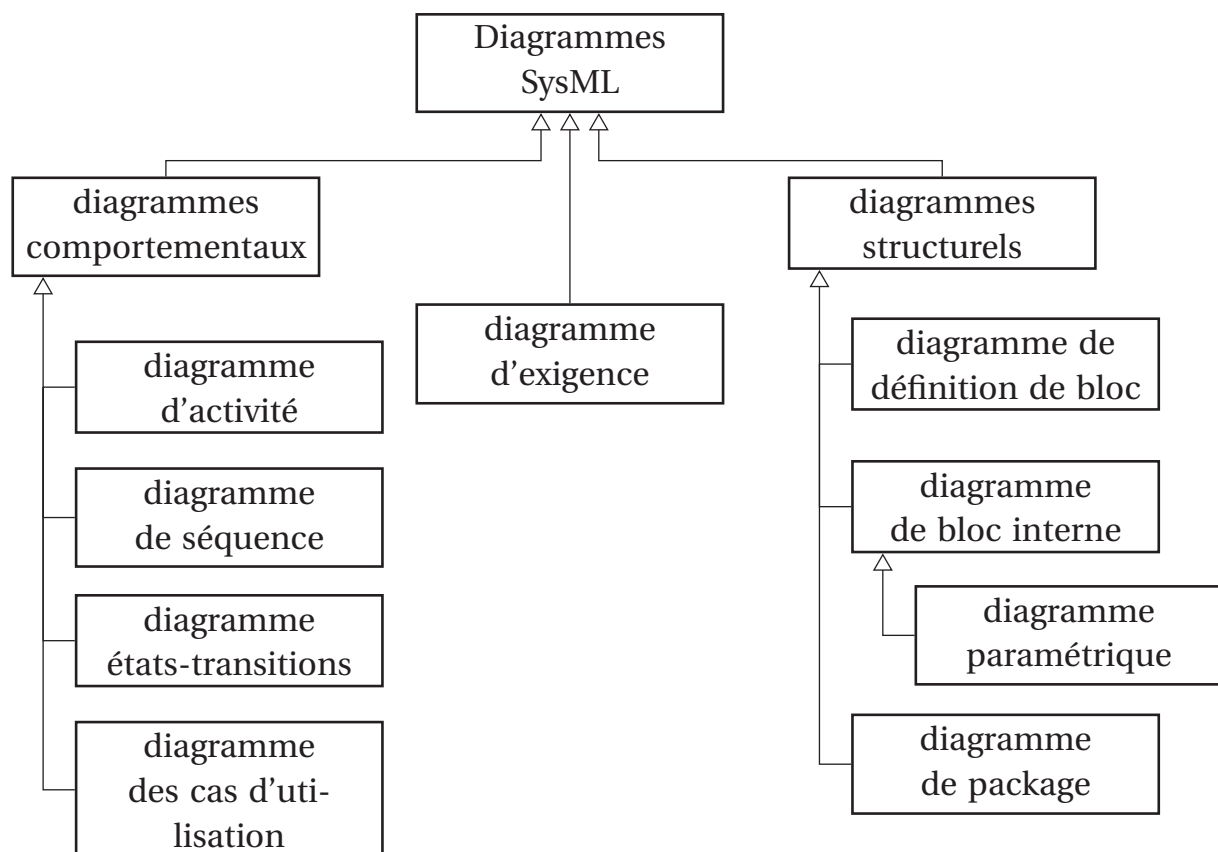


FIGURE 1.2 – Les diagrammes SysML

Les diagrammes SysML regroupent les différents diagrammes qui vont tra-

1. [http://fr.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://fr.wikipedia.org/wiki/Unified_Modeling_Language)

2. [http://fr.wikipedia.org/wiki/Comparaison\\_des\\_logiciels\\_d'UML](http://fr.wikipedia.org/wiki/Comparaison_des_logiciels_d'UML)

duire les fonctionnalités, la structure, le comportement et les exigences du système.

diagramme	description
req requirements exigences	Le diagramme des exigences permet de représenter les contraintes techniques ou non du système(voir 1.4.3 page 11)

En SysML, le comportement d'un système peut être modélisé à l'aide de quatre diagrammes :

diagramme	description
uc use case cas d'utilisation	le diagramme de cas d'utilisation permet de représenter les fonctionnalités du système dans leur contexte (voir 1.4.2 page 7)
act activity activité	le diagramme d'activité décrit l'enchaînement des actions lié à un cas d'utilisation. Le diagramme d'activité permet de représenter le flux de données et de contrôle entre différentes actions. En effet, ce diagramme est le plus adapté pour représenter une architecture fonctionnelle du point de vue structurel et comportemental.
sd sequence séquence	Le diagramme de séquence permet de représenter les échanges entre différentes parties d'un système, et avec des systèmes ou acteurs externes au système. Il décrit, en particulier, les cas d'utilisation.
stm state machine état-transition	le diagramme d'états illustre les les changements d'états d'un système ou d'un sous système. Il décrit les transitions entre états et les actions qu'un système réalise en réponse à des événements.

Les diagrammes structurels décrivent la constitution du système du plus général au plus précis :

type abrégé diagramme	description
bdd : block definition définition de bloc	le diagramme de définition de blocs représente la structure globale du système. il montre les constituants de base statiques (voir 1.5.5 page 18) : blocs, compositions, associations, attributs, opérations, généralisations,...
ibd : internal block bloc interne	le diagramme de bloc interne montre l'organisation interne d'un bloc et les flux qui relient les blocs le constituants (voir 1.5.6 page 20)
par : parametric paramétrique	le diagramme paramétrique modélise le comportement d'un bloc ou plusieurs dans le temps à partir des équations qui le régissent
pkg : package package	le diagramme de packages montre l'organisation logique du modèle et les relations entre packages

### 1.3.2 Symbolisation

#### a) Cartouche

Tous les diagrammes sont représentés dans un cadre avec un cartouche précisant la nature du diagramme

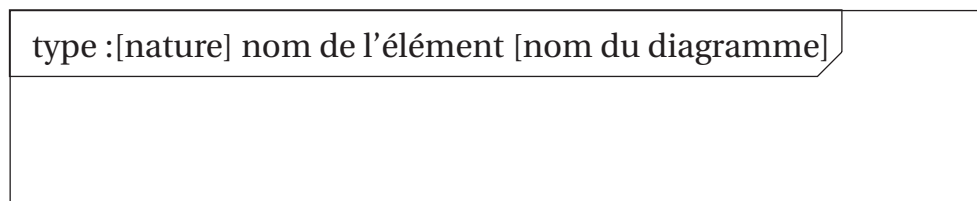


FIGURE 1.3 – Cartouche des diagrammes SysML

L'abréviation (req, uc, ibd, bdd, ...) indique le type de diagramme, on complète la désignation du diagramme par le nom de l'élément étudié. Le nom du diagramme permet de préciser le niveau d'approfondissement.

## 1.4 Premier exemple- Étude sommaire d'un HSUV

On s'intéresse à la description d'un véhicule tout terrain "vert".

### 1.4.1 Définir le contexte

La première étape que se soit en phase de conception d'un nouveau produit ou de l'analyse de celui-ci consiste à capturer (recueillir) les besoins et le contexte.

La définition du contexte commence par l'établissement de la frontière du produit étudié. Le produit est au centre et on trouve autour tous les intervenants, les acteurs du systèmes, C'est à dire :

- Définir le contexte général dans lequel le produit va être utilisé.
- Définir les acteurs humains ou non qui vont interagir avec le système.
- Définir les fonctionnalités attendues du système.
- Définir le contexte technique du produit.
- Définir le fonctionnement dynamique de chaque fonctionnalité.
- Définir les besoins en interface homme machine (IHM).
- Rédiger un cahier des charges fonctionnel et technique qui permettra à la fois de réaliser le produit et de le valider au fur et à mesure.

Le terme d'acteur(qui agit sur le système) est a prendre au sens large et pas uniquement au sens humain.

Pour mieux comprendre, on peut s'intéresser au diagramme de contexte de la figure 1.4.

On distingue sur le diagramme de contexte :

- le produit (la voiture)
- les acteurs, qui se décomposent en :
  - le conducteur (acteur obligatoire)
  - les passagers
  - les bagages
  - le service de maintenance
  - le milieu extérieur (route, temps, ...)

Les liens entre le produits et les acteurs sont indiqués sur le graphe.

Tous les acteurs n'ont pas le même poids

- le conducteur est obligatoire,
- les passagers et les bagages ne sont pas nécessairement présents,
- le temps et la route ont toujours une influence,
- le service de maintenance n'intervient que si nécessaire ou suivant un cycle pré-établi.

### 1.4.2 Les acteurs et les cas d'utilisation

Le système étudié est toujours analysé au travers de différents point de vue ou en utilisant le vocabulaire des diagrammes SysML, au travers de différents cas d'utilisation (use case en anglais<sup>3</sup>).

Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation (représenté par des ovales) reliés par des associations (lignes) à leurs acteurs (icône d'un stick man). Chaque association signifie simplement « participe à »

Les cas d'utilisation sont décrit par un verbe à l'infinitif.

---

3. vous avez noté que les différents diagrammes et éléments sont notés en anglais, la norme actuellement n'est définie qu'en anglais!!!

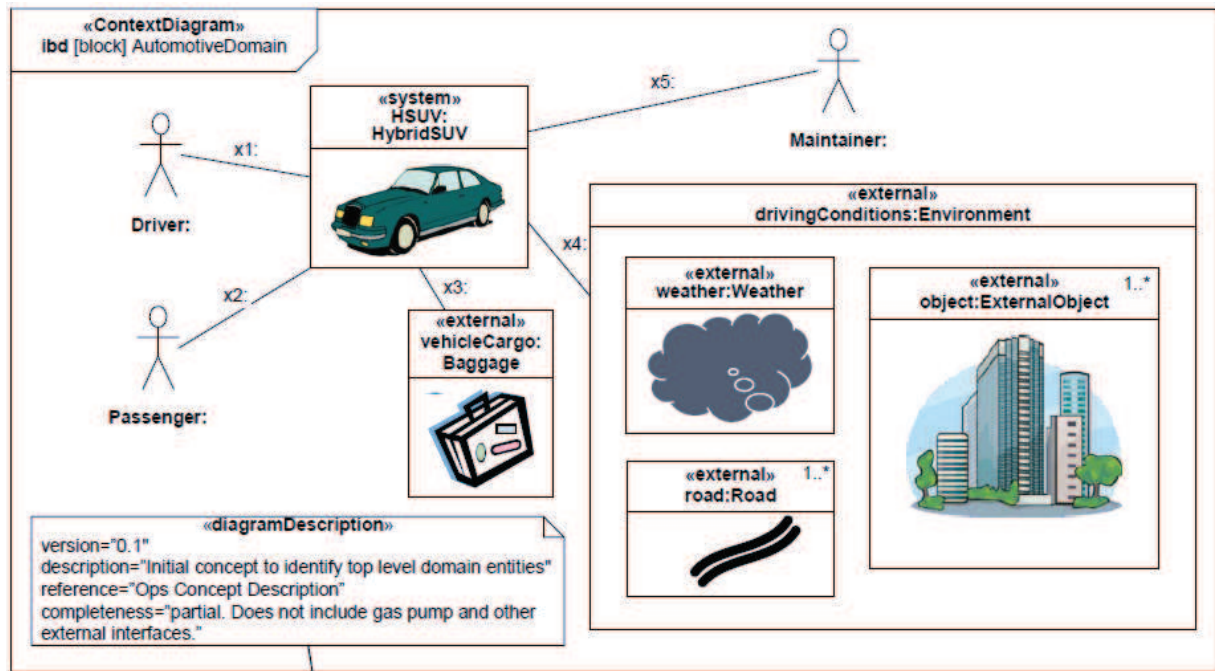


FIGURE 1.4 – Diagramme de contexte d'une voiture

On peut commencer par définir les différents acteurs du système et pour chacun définir les fonctions.

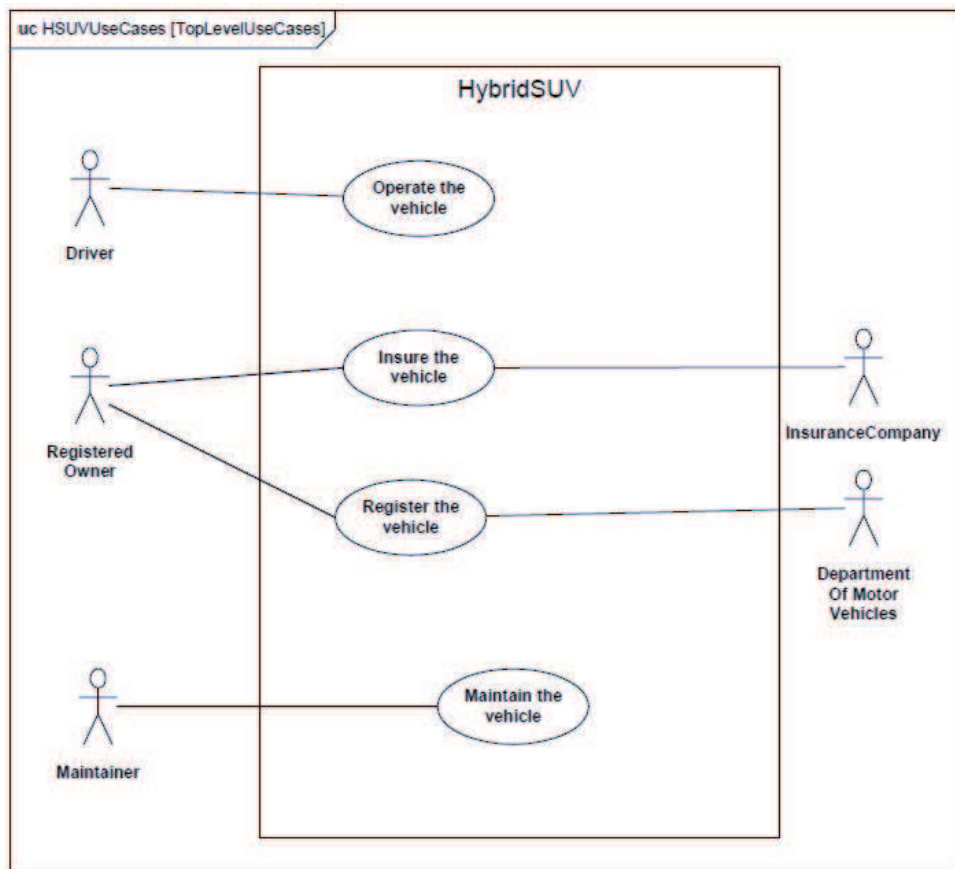


FIGURE 1.5 – Diagramme général des cas d'utilisation



Si on réalise une transposition en français, on retrouve :

- les fonctions nécessaires<sup>4</sup> à l'utilisation d'un véhicule
  - Conduire le véhicule
  - Assurer le véhicule
  - Déclarer le véhicule
  - Maintenir le véhicule
- les acteurs
  - le conducteur
  - le propriétaire (pas forcément le propriétaire)
  - la compagnie d'assurance
  - la préfecture
  - le garage

On constate que certains acteurs sont humains alors que les autres sont des personnes morales ou des systèmes extérieurs. Les liens précisent les relations entre les différents acteurs et les cas d'utilisation.

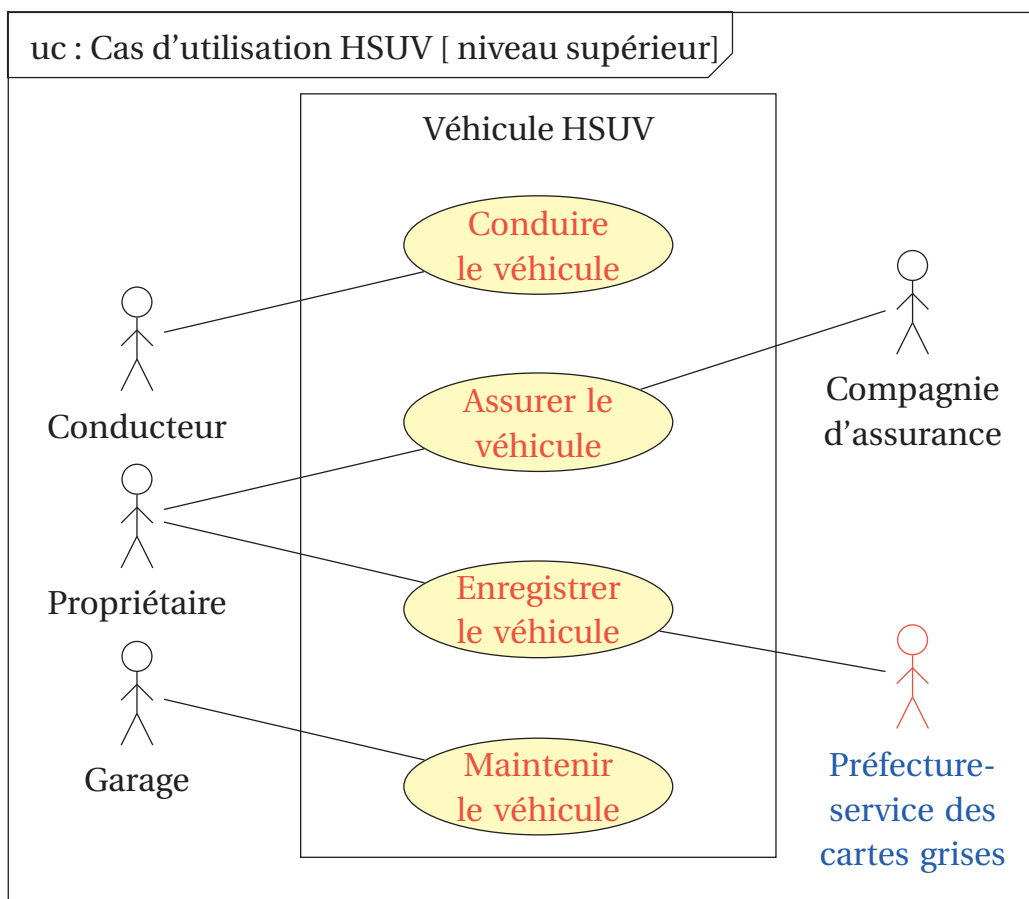


FIGURE 1.6 – Diagramme général des cas d'utilisation du HSUV

Ce qui nous donne le diagramme de la figure 1.6, qui est un diagramme général (de niveau supérieur) du véhicule. Il est nécessaire d'affiner cette étude en

4. cette liste n'est pas exhaustive

détaillant les différents cas d'utilisation. On peut par exemple détailler le cas d'utilisation du conducteur.

Pour cela on renouvelle l'étude en précisant les différentes fonctions qui constituent la fonction générique « Conduire le véhicule », c'est à dire :

- démarrer le véhicule,
- accélérer,
- diriger,
- freiner,
- stationner.

Les liens entre les fonctions ne sont pas tous de même nature :

- Le lien entre « Conduire le véhicule » et « Accélérer », « Diriger » et « Freiner » est un lien d'inclusion (*include*). Le cas d'utilisation « Conduire le véhicule » inclus nécessairement les cas d'utilisation « Accélérer », « Diriger » et « Freiner ».
- Le lien entre « Démarrer le véhicule » et « Conduire le véhicule » est un lien d'extension (*extend*).

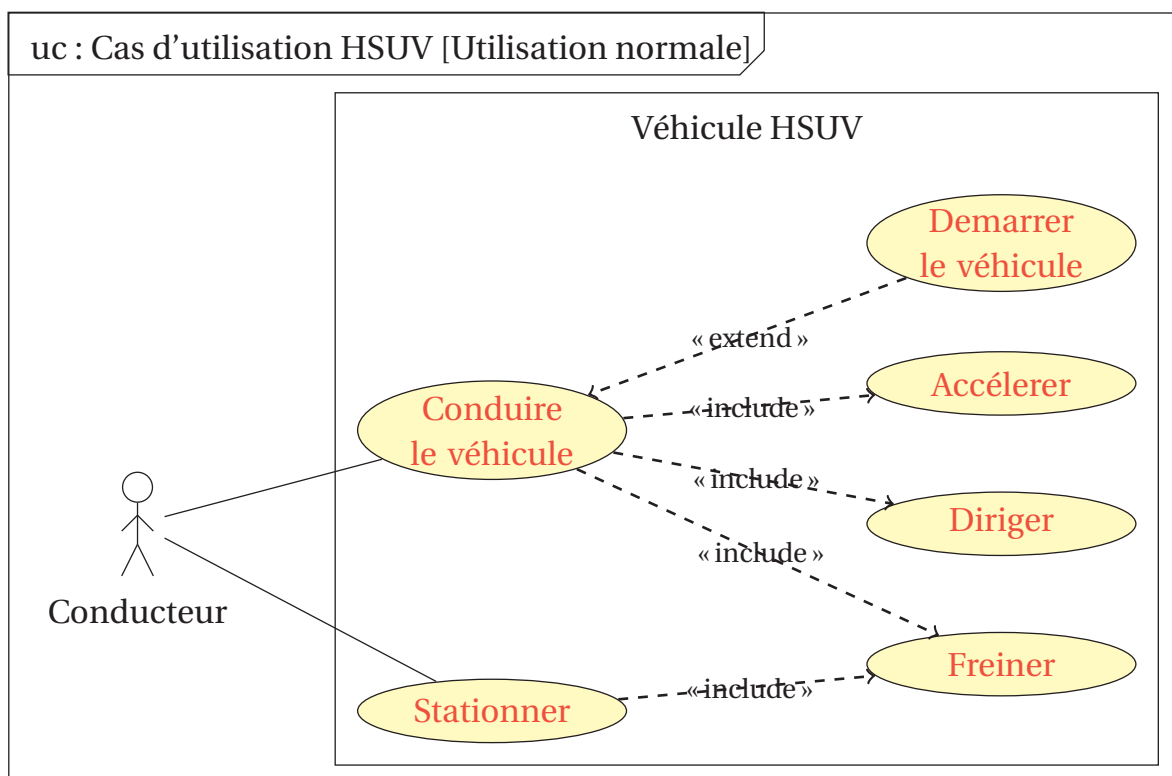


FIGURE 1.7 – Cas d'utilisation : du point de vue du conducteur

Il est possible de continuer à affiner cette étude, mais on se limite en général à deux ou trois niveaux.

### 1.4.3 La modélisation des exigences

En parallèle à la définition des cas d'utilisation, il faut se poser la question des capacités que doit posséder le système mais aussi les contraintes qu'il doit respecter. Dans la symbolique du langage de description SysML, ces contraintes et ces capacités sont les *exigences* que doit remplir le système.

Le diagramme d'exigence est le schéma associé à cette description.

L'objectif du *diagramme d'exigences* est de traduire graphiquement et hiérarchiquement les capacités et les contraintes que doit satisfaire le système.

On peut commencer par définir les exigences générale du système.

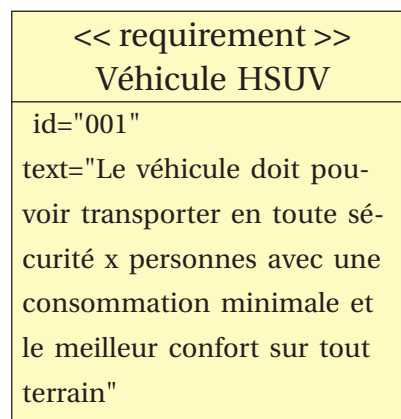


FIGURE 1.8 – Exigence principale du véhicule

Dans la symbolique de SysML, une exigence est représentée dans un cadre avec le mot clef « requirement » (exigence en anglais). On trouve dans ce cadre :

- le titre du cadre, précisant le domaine de l'exigence,
- puis au moins deux éléments :
  - un identifiant unique,
  - un texte descriptif de l'exigence.

Il est ensuite nécessaire de préciser ces exigences pour chacun des constituants du système en prenant en compte les différents cas d'utilisation.

On peut ainsi décliner l'exigence principale du système en autant d'exigence que devront remplir les sous systèmes afin de respecter l'exigence principale. le diagramme de la figure 1.9 montre une décomposition partielle des exigences du véhicule étudié au travers de 5 des constituants.

La liaison entre les exigences est ici une liaison de contenance (ligne terminée par un cercle et un croix du côté du conteneur), elle montre la décomposition d'une exigence composite en plusieurs exigences unitaires.

Pour aller plus loin, nous allons continuer l'étude en nous appuyant sur un autre exemple moins (beaucoup moins) complexe. Cet exemple va nous permettre de compléter notre étude du langage de description SysML.

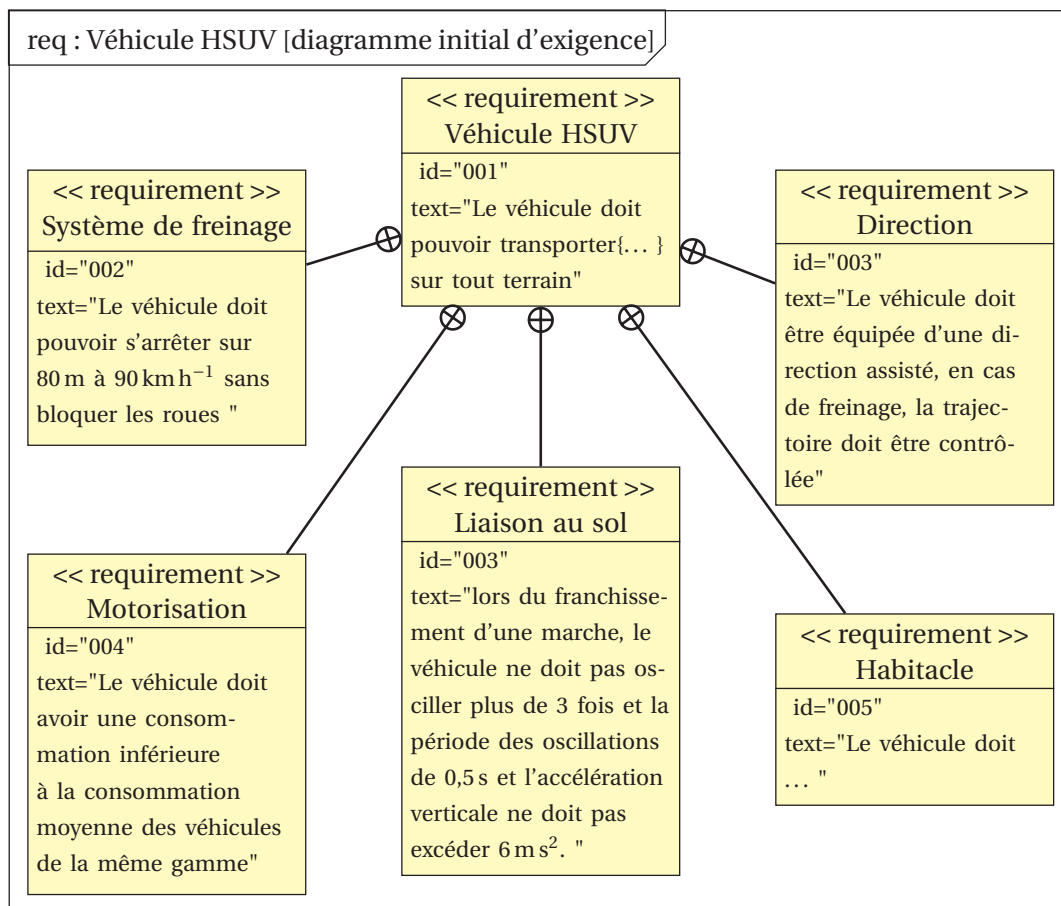


FIGURE 1.9 – Diagramme initial d'exigence du véhicule (incomplet!)

## 1.5 Exemple guide : balance de ménage

### 1.5.1 Description

Cette balance de cuisine permet de mesurer aussi bien des liquides que des produits solides jusqu'à 4 kg grâce à son large plateau en verre qui accepte tous les récipients, la précision annoncée est de 1 g. Elle est proposée dans un grand choix de couleurs des plus classiques aux plus tendances afin de s'intégrer à toutes les cuisines, son faible volume (25 cm × 20 cm × 4 cm) assure un rangement et une manipulation aisés.



FIGURE 1.10 – Balance de ménage Halo®

Un mode de tarage automatique permet à la fois de prendre en compte la masse du récipient mais aussi de réaliser un pesage cumulatif des différents in-

grédients d'une recette en les ajoutant successivement dans le même bol.

Un mode de conversion kg / l permet d'obtenir l'équivalent en volume d'eau de la quantité pesée.

Fonctionne avec 1 pile Cr2032 incluse.

### 1.5.2 Décrire l'environnement : diagramme de contexte

Le diagramme de contexte va nous permettre de décrire l'environnement du système

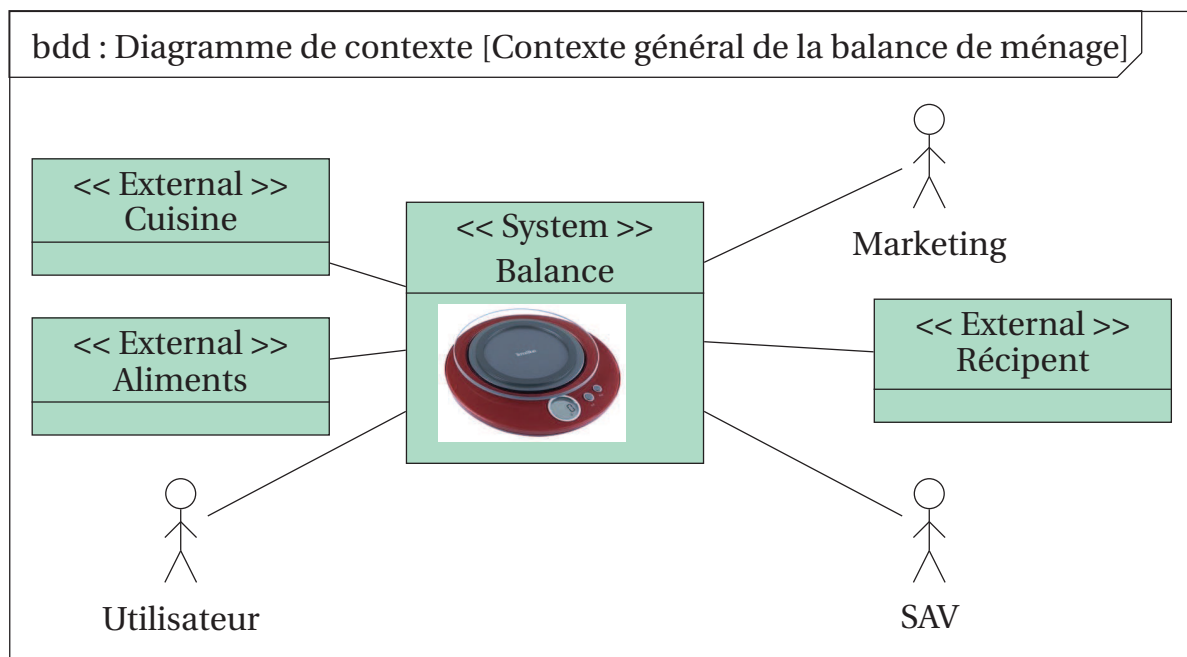


FIGURE 1.11 – Diagramme de contexte général

Dans ce diagramme, on retrouve les différents acteurs susceptibles d'avoir un rôle sur le système « Balance ».

On le limite ensuite à celui de l'utilisation courante avec l'acteur principal.

**Remarque :** Le diagramme de contexte n'est pas un diagramme défini dans la norme SysML, il est construit à partir d'un diagramme de définition de bloc (bdd) ou d'un diagramme de blocs internes (ibd).

### 1.5.3 Identifier les cas d'utilisation : diagramme des cas d'utilisation

Du point de vue d'utilisateur, on distingue 3 cas d'utilisation

- peser,
- tarer,
- convertir kg / l.

Le tarage possède deux modes, un mode obligatoire, la mise à zéro initiale puis un mode à la demande permettant des pesées cumulées ou au changement de récipient.

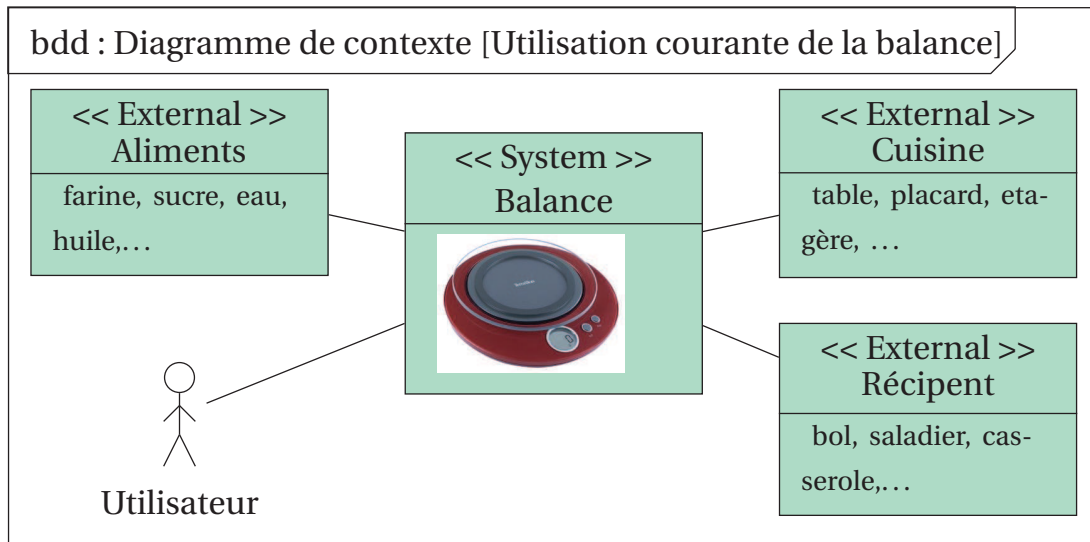


FIGURE 1.12 – Diagramme de contexte limité à l'usage principal

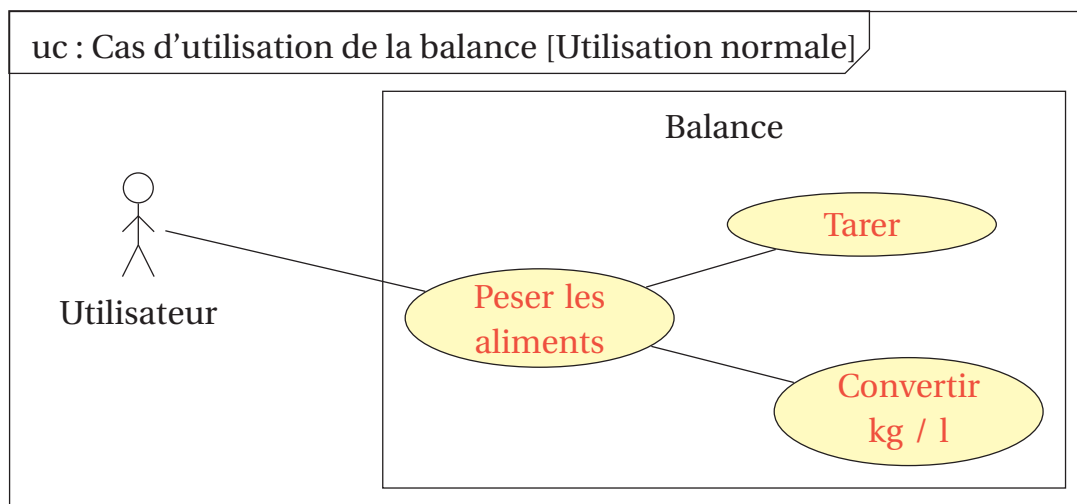


FIGURE 1.13 – Diagramme général des cas d'utilisation de la balance

Il est possible sur ce diagramme de préciser aussi la nature des liens entre les différents cas d'utilisation :

- un lien de type « include » entre la pesée et le tarage (le tarage est obligatoire)
- un lien de type « extend » entre la pesée et la conversion (non nécessaire à la pesée).

#### 1.5.4 Décrire les exigences du système : diagramme d'exigence

L'exigence principale de la balance est bien sûr **Peser les aliments**. Le système doit pour cela :

- mesurer la masse,
- supporter un récipient,
- permettre des mesures cumulées.

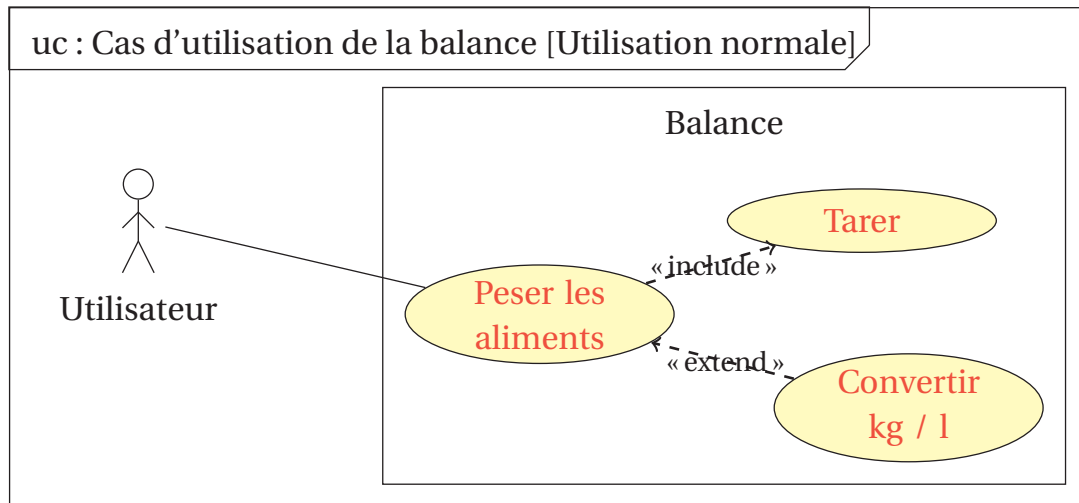


FIGURE 1.14 – Diagramme général des cas d'utilisation de la balance

Une dernière exigence, plus ergonomique que fonctionnelle peut être ajoutée  
 – faciliter le transport et le rangement.

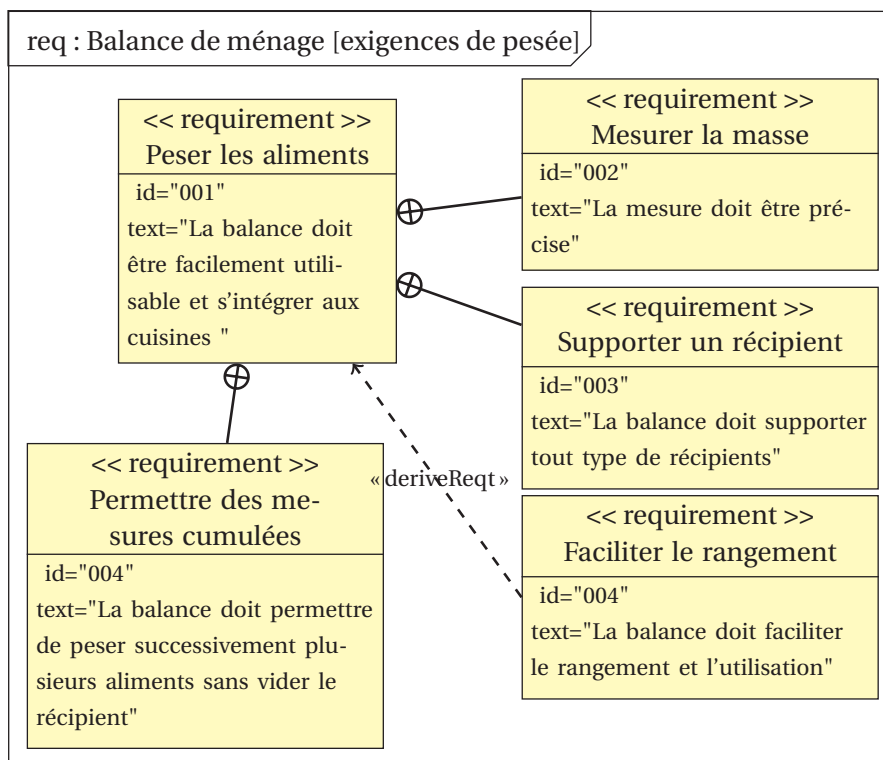


FIGURE 1.15 – Diagramme d'exigence de la balance de ménage

On retrouve sur le diagramme les 3 exigences principales de la balance, elles sont toutes les trois liées à l'exigence principale par un lien de contenance, en effet, ces trois exigences sont nécessaires pour réaliser la pesée. Par contre la dernière exigence, elle participe à l'amélioration de l'utilisation mais n'est pas nécessaire à la pesée, cette exigence est précisée par un lien de dérivation d'exigence.

Cette analyse peut encore être affinée. Sur le diagramme d'exigences suivant (figure 1.16), un nouveau niveau d'exigence a été ajouté sous l'exigence **Faciliter le rangement** qui apporte des précisions sur les notions d'autonomie et de rangement.

Sur les figures 1.15 et 1.16 on distingue différents liens entre les exigences ou les autres éléments, la signification de ces liens est précisée dans le tableau 1.1.

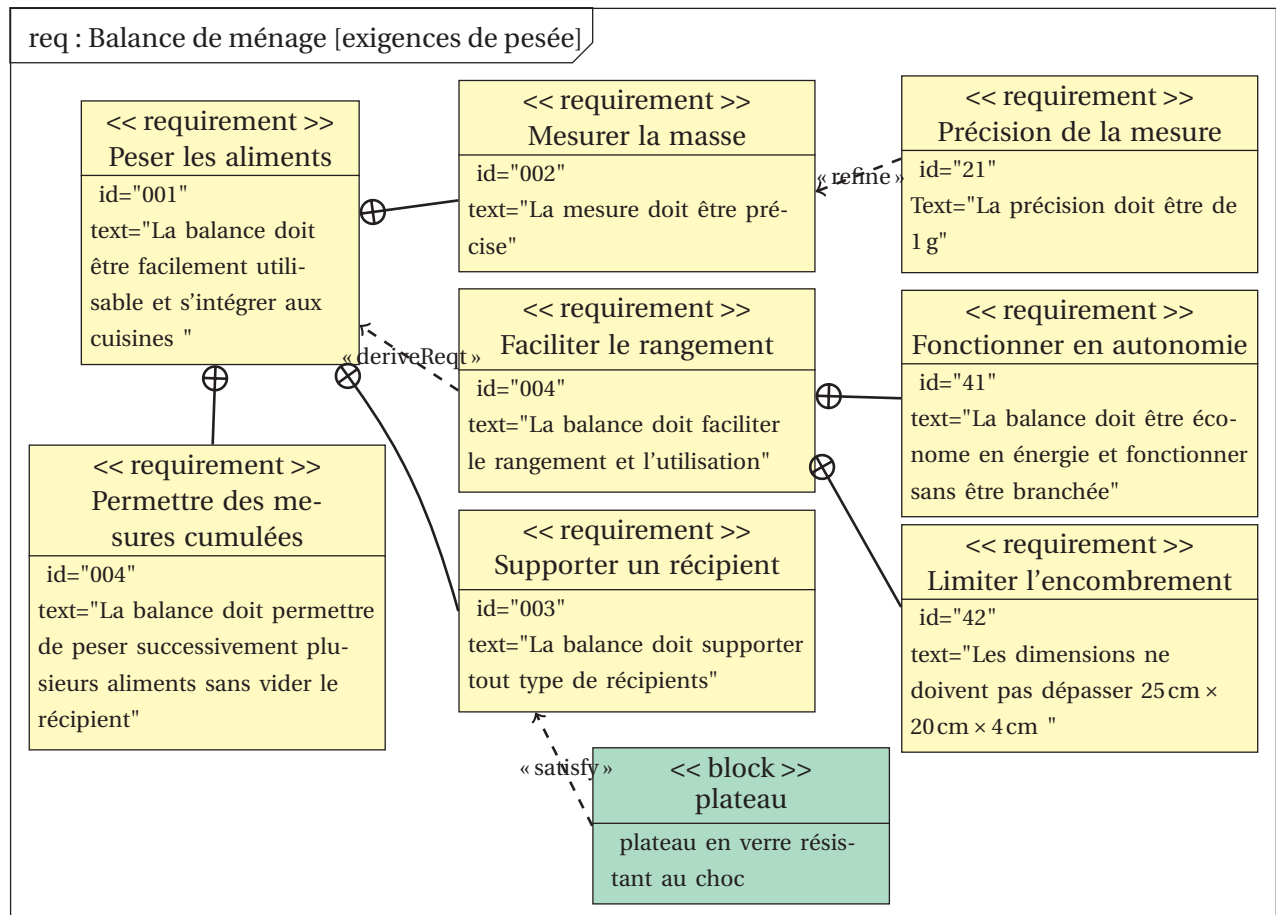


FIGURE 1.16 – Diagramme d'exigence affiné de la balance de ménage

En complément de l'exigence **Mesurer la masse**, l'exigence de précision du cahier des charges a été ajoutée. Le lien entre les deux exigences est un lien qui affine la description, il est noté avec le mot clef « refine », .

On note aussi que le respect de l'exigence **Supporter un récipient** est réalisée par un plateau en verre. Cette solution est précisé dans un « block » et un lien noté « satisfy » lie l'exigence au bloc.

Dans ce diagramme, apparaît un autre élément des diagrammes SysML, le « block ». Ici, le block plateau caractérise le constituant matériel plateau qui satisfait (lien satisfy) l'exigence « supporter un récipient ».

Il est possible d'aller plus loin dans la description des exigences en précisant la nature de celles-ci.

- Exigences fonctionnelles - **functionalrequirement** - elles précisent les exi-



Contenance	précise la décomposition d'une exigence en exigences unitaires (l'exigence principale est coté cercle avec la croix)	
Raffinement refine	l'exigence pointée par la flèche est précisée par celle à l'autre extrémité	
Derivation deriveReq	indique que l'exigence pointée par la flèche induit l'exigence en queue de flèche	
Satisfaction satisfy	l'exigence pointée par la flèche est satisfaite par l'élément (en général un bloc) en queue de flèche	

TABLE 1.1 – Les différents liens du diagramme d'exigence

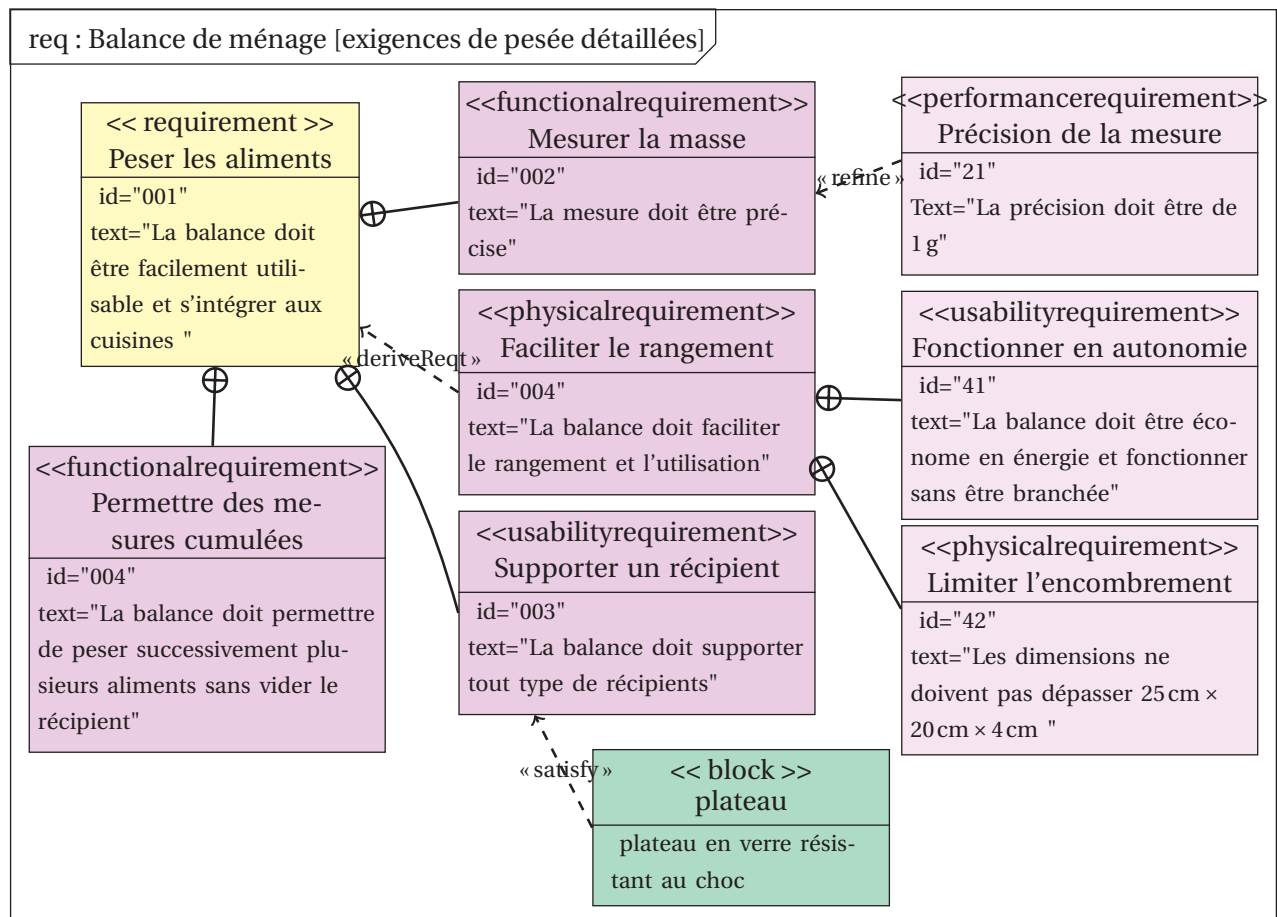


FIGURE 1.17 – Diagramme d'exigence avec nature des exigences

gences qui découlent d'une caractéristique fonctionnelle que doit posséder le produit.

- Exigences physiques - **physicalrequirement** - elles décrivent les exigences qui découlent d'une propriété physique du produit.
- Exigences de performance - **performancerequirement** - décrivent le niveau

de qualité d'une exigence.

- Exigences d'usage -**usabilityrequirement** - décrivent les critères d'utilisation (d'usage) que doit posséder le produit.

Le diagramme des exigences de la figure 1.17 précise pour la balance de ménage la décomposition des ces exigences.

Il est aussi possible des décrire des exigences qui ne sont directement fonctionnelles comme des exigences commerciales, de marketing ou du design.

Le diagramme des exigences ne sert pas seulement à placer des contraintes dans le modèle. Il permet aussi de ramener des éléments venant d'autres diagrammes du modèle et ainsi de vérifier qu'une exigence a bien été prise en compte. Le « block » plateau est un éléments qui vient du diagramme de définition de bloc qui satisfait une des exigences du système.

### 1.5.5 Préciser la constitution du système : diagramme de définition de blocs

L'étape suivante de l'analyse du système doit décrire les éléments constituant le système. Le diagramme de définition de blocs (bdd) est utilisé pour préciser les éléments constituant le système.

Ce diagramme est construit en respectant la structure physique et logicielle du système. Les liens (voir tableau 1.2) entre les différents constituants sont indiqués par des liens précisant la dépendance de chacun des constituants.

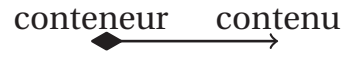


Composition	le bloc conteneur a nécessairement besoin du bloc contenu pour fonctionner	
Agrégation	le bloc conteneur n'a pas nécessairement besoin du bloc contenu	
Association	les blocs ne sont hiérarchiquement liés	

TABLE 1.2 – Les différents liens du diagramme de définition de bloc

Sur le diagramme de la figure 1.18 on note que la balance comporte 5 constituants principaux (le châssis, le plateau, le capteur, la pile et l'interface utilisateur), tous ces éléments sont nécessaires au fonctionnement de la balance, ce qui est précisé par les liens de composition. L'interface utilisateur est aussi décomposée.

Chaque bloc peut être plus ou moins détaillé en fonction de la finesse de l'étude. Un bloc peut ainsi comporter plusieurs éléments complémentaires :

- *values* : précisent quantitativement les valeurs caractéristiques du bloc.
- *params*
- *parts* : précisent les parties (sous-parties) constituant le bloc

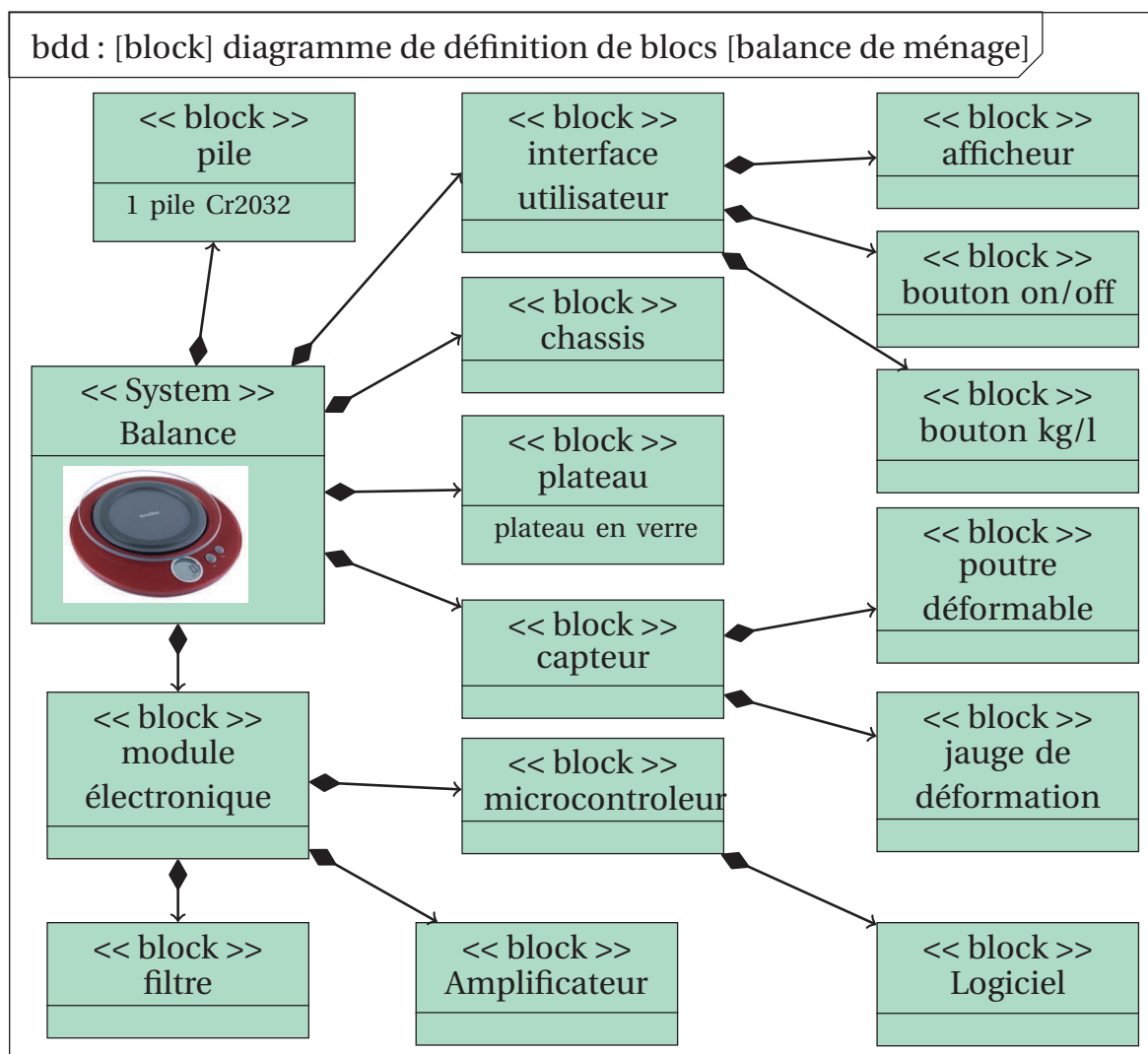


FIGURE 1.18 – diagramme de définition de blocs de la balance de ménage

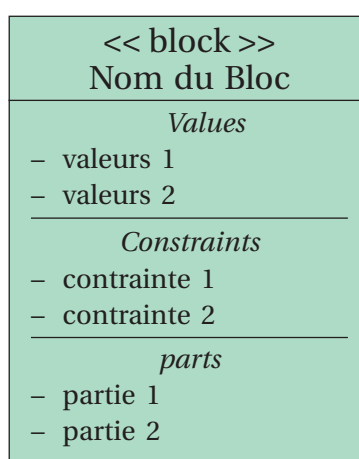


FIGURE 1.19 – détails d'un bloc

- *constraints* : les contraintes précisent des relations, équations entre les paramètres du bloc

– références

### 1.5.6 Préciser les liens et les flux : diagramme de blocs interne

Le diagramme de blocs interne (ibd) permet lui de préciser les flux qui transitent entre les différents blocs du système.

Les flux peuvent être de toute nature : matière, énergie, information. Sur chaque bloc, les ports précisent la nature des flux échangés.

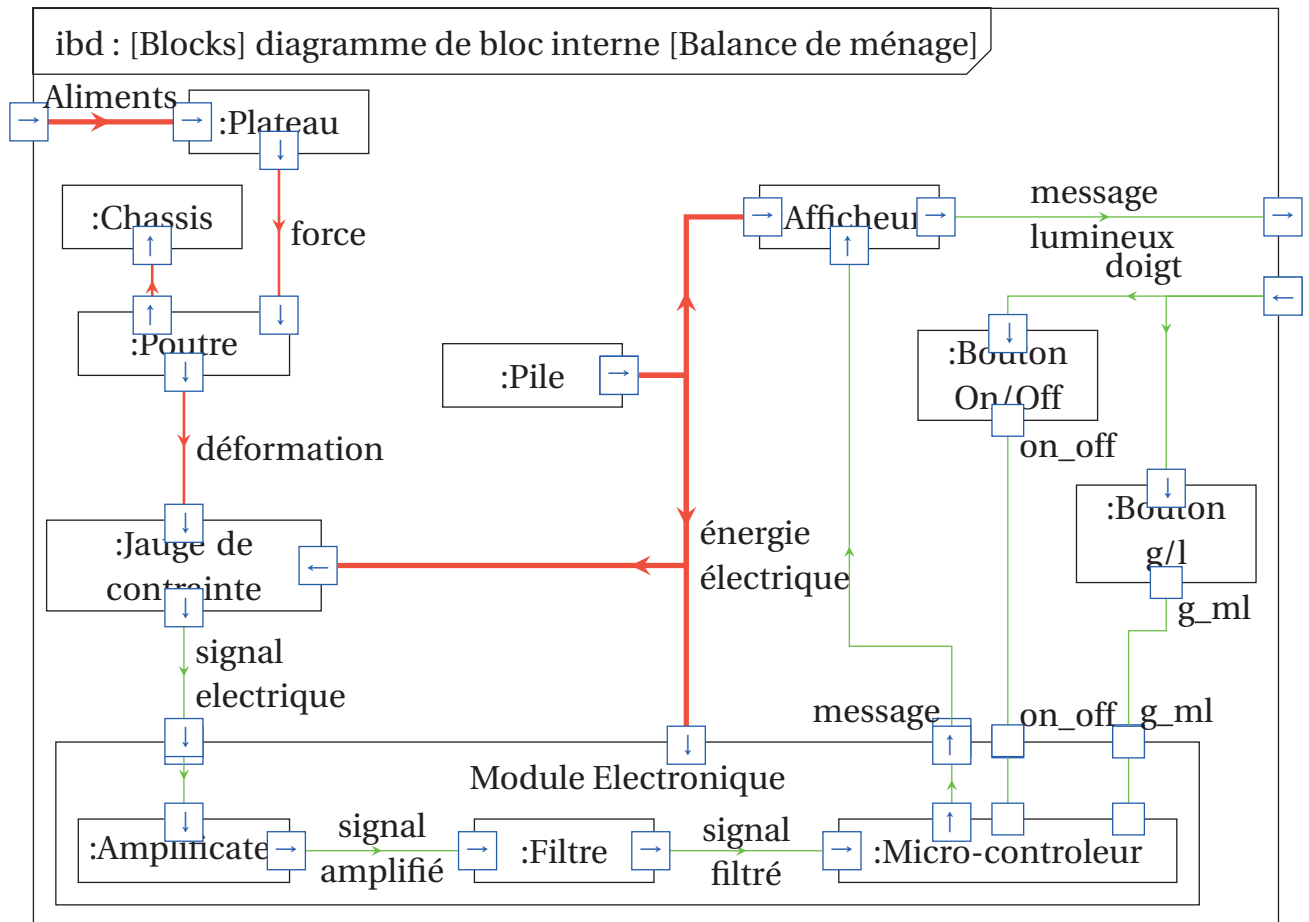


FIGURE 1.20 – diagramme de bloc interne

On retrouve ainsi sur la figure 1.20 le diagramme de bloc interne de la balance de ménage, on y distingue :

- le flux d'énergie électrique qui va de la pile aux différents constituants
- le flux de la pesée, de la dépose des aliments sur le plateau, le poids déforme la poutre qui étire (comprime) les jauges de déformation.
- cette déformation génère un signal électrique de faible amplitude (qq mV), qu'il est nécessaire d'amplifier puis de filtrer afin d'avoir un signal électrique « propre » à l'entrée du microcontrôleur.
- le microcontrôleur à partir des deux informations issues des boutons et du signal filtré, génère un message affiché sur l'afficheur.

Le logiciel n'apparaît pas, il est inclut dans le bloc du microcontrôleur.

Nous utiliserons ce diagramme pour décrire les chaînes d'énergie et d'information d'un système en l'associant avec d'autres représentations.

### 1.5.7 Décrire l'organisation matérielle et logicielle : diagramme de package

Le diagramme de package n'est pas explicitement au programme, nous ne le développerons qu'au travers des exercices.

### 1.5.8 Décrire la succession des états : diagramme d'états transition

Le diagramme d'états-transition (State machine diagram) permet de décrire la suite des états d'un (un système, un sous-système). Chaque état décrit une situation dans laquelle se trouve le bloc, les liens entre les différents états représentent les différentes évolutions possibles d'un état en un autre. Le passage d'un état à un autre se fait en franchissant la transition liée au lien.

On peut ainsi décrire la succession des états de la balance par le diagramme suivant.

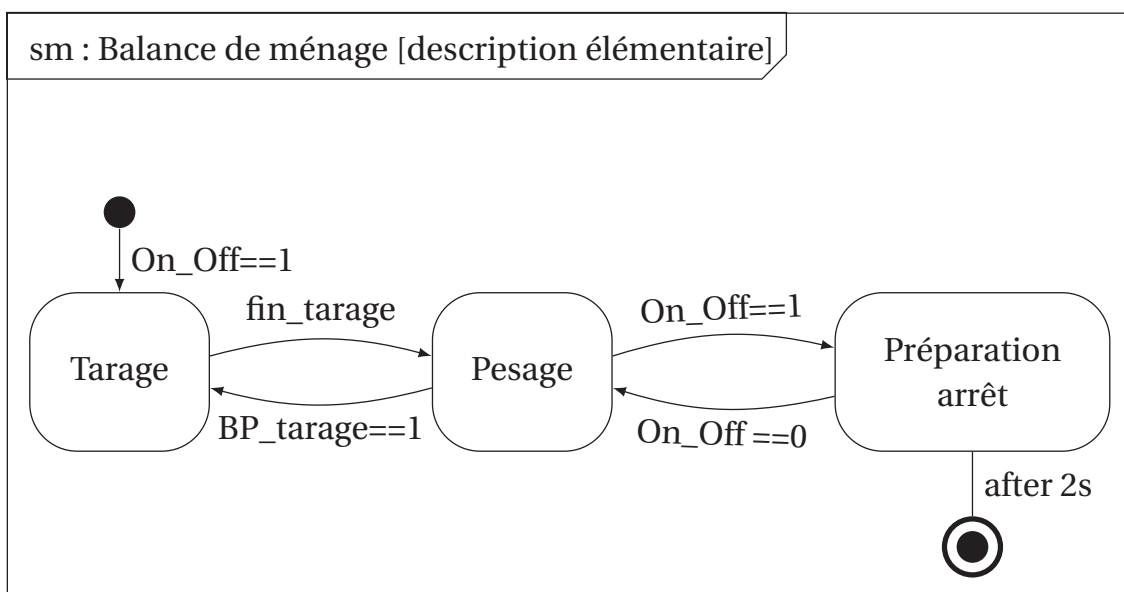


FIGURE 1.21 – Diagramme états-transitions de la balance de ménage - [description élémentaire]

Ce diagramme correspond à une description élémentaire et incomplète du fonctionnement. Il décrit le cycle suivant :

- l'utilisateur appuie sur le bouton marche ( $On\_Off==1$ ) cet *événement* active l'état du Tarage ;
- dès que le tarage est terminé, le pesage le pesage peut-être réalisé ;
- un nouvel appui sur le bouton tarage ( $BP\_tarage==1$ ) replace le système dans l'état Tarage ;

- un appui sur On\_Off place le système dans un mode d'arrêt à condition que le bouton ne soit pas relâché ( $\text{On\_Off}==0$ ) dans un délai de 2s, au bout du délai, la balance s'éteint.

Le diagramme suivant (figure 1.22) présente une évolution prenant en compte d'autres possibilités d'évolution :

- la possibilité de réaliser une conversion poids / volume équivalent à la demande
- l'arrêt en cas d'inactivité au bout d'une minute.

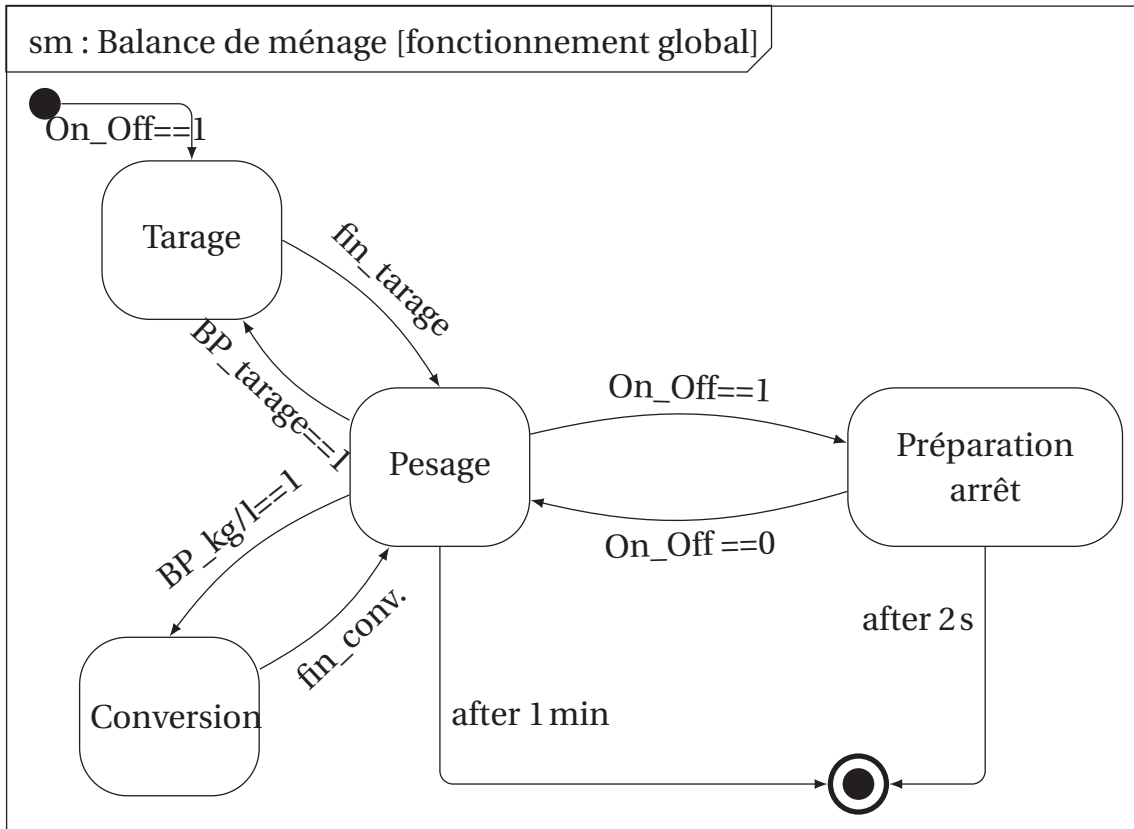


FIGURE 1.22 – Diagramme états-transitions de la balance de ménage

Les actions associées aux états peuvent être réalisées à trois instants particuliers

- à l'entrée dans l'état, cet instant est identifié par le mot clef : *entry*, la tâche associée est effectuée à chaque fois que l'état est activé.
- à la sortie de l'état, cet instant est identifié par le mot clef : *exit*, dès qu'une transition peut être franchie, l'action associée à la sortie de l'état est réalisée.
- pendant la durée de l'état : le mot clef *do* précise les actions qui seront réalisées pendant la durée d'activité de l'état.

La syntaxe complète des transitions s'écrit : *événement[garde]/action* et comporte trois éléments.

- L'*événement* est l'élément déclenchant la transition, dès qu'il est vrai (état logique vrai) le franchissement peut avoir lieu.

- La *condition de garde*, est une condition logique optionnelle qui, lorsqu'elle est présente doit être vraie pour que la transition soit franchie.
- L'*action*, est réalisée, si elle est présente, dès que la transition est franchie.

Un état peut inclure lui-même un diagramme état-transition.

Nous verrons plus loin des exemples et exercices avec la syntaxe complète des diagrammes état-transition.

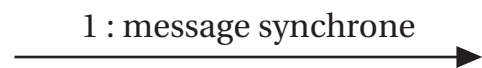
### 1.5.9 Décrire le comportement : diagramme de séquence

Le diagramme de séquence permet de représenter les informations échangées entre les acteurs et le système tout en tenant compte du temps.

On retrouve sur le diagramme de séquence de la balance (figure 1.23) les principaux éléments :

- les acteurs et le système : chaque élément est représenté dans un cadre, une ligne de vie en pointillé verticale, montre le déroulement du temps.
- les 3 types de messages échangés :

#### Message synchrone :



ce message est émis par un des éléments vers un autre. Le message émis est précisé sur la flèche. Ce type de message attend une réponse.

#### Réponse à un message synchrone :



La réponse est précisée sur la flèche.

#### Message asynchrone :



Ce message est émis sans attendre de réponse.

- quelques structures de description :

**loop** le contenu de la structure *loop*, est exécutée en boucle tant que la condition n'est pas vraie (structure Tant Que cond. Faire ...);

**alt** la structure *alt* précise les évolutions alternatives (structure Si cond Alors cas vrai Sinon ...);

**opt** la structure *opt* précise les évolutions optionnelles (structure Si cond Alors Faire ...);

nous verrons d'autres structures dans les exemples (structure parallèle *par*).

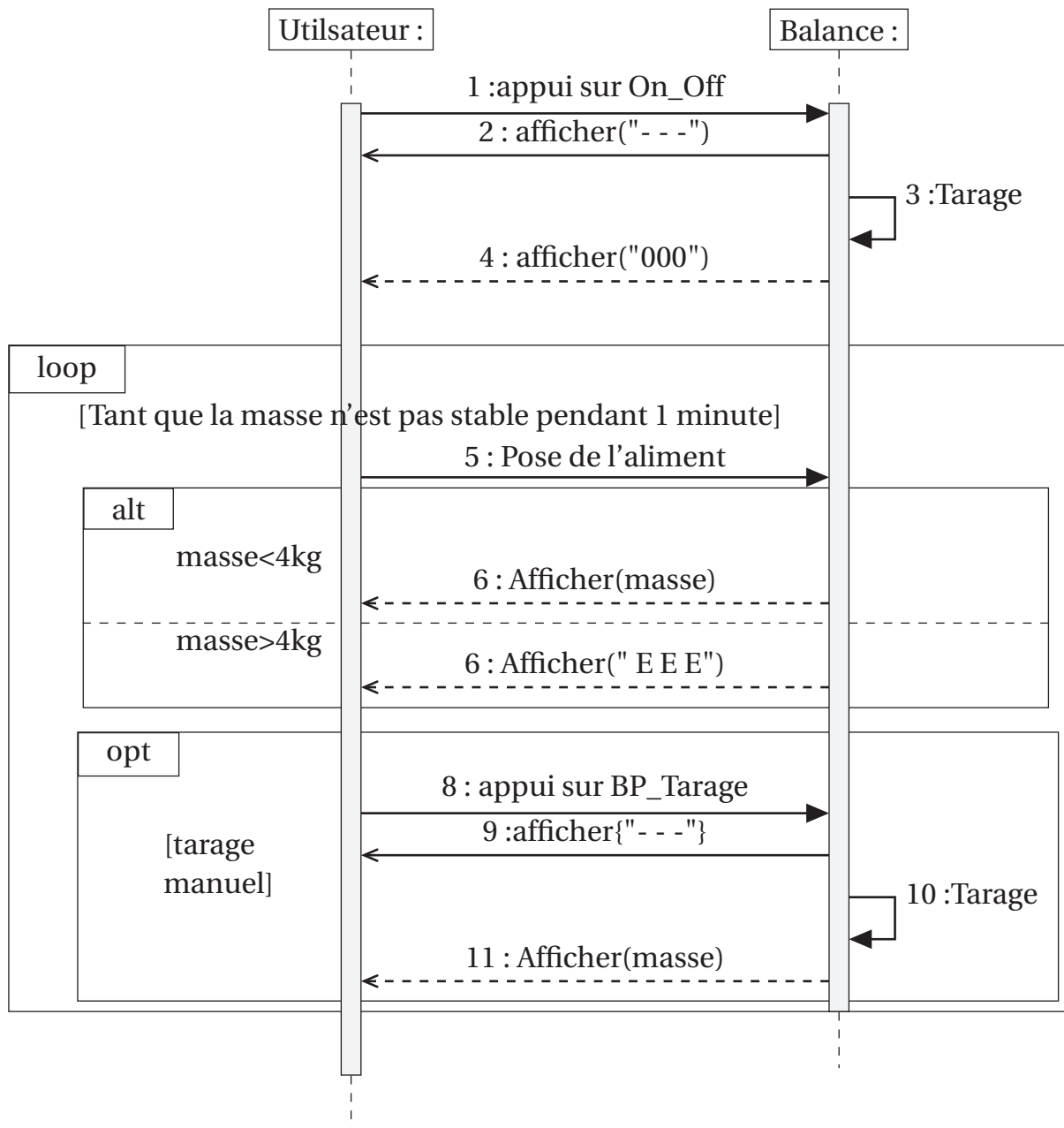


FIGURE 1.23 – Diagramme de séquence de la balance de ménage

### 1.5.10 Décrire le comportement : diagramme d'activité

Le diagramme d'activité n'est pas explicitement au programme, nous ne le développerons qu'au travers des exercices

### 1.5.11 Exercices

#### Exercice 1- Caractérisation d'une guitare

*Corrigé page 33*

Soit la guitare de la figure 1.24

**Q1.** Tracer le diagramme de définition de bloc de la guitare



Le guitariste utilise un capodastre pour changer de gamme

**Q2.** Installer dans ce diagramme le capodastre



(a) Bloc



(b) guitare

FIGURE 1.24 – diagramme de définition de bloc d'une guitare

## Exercice 2- Tourniquet

Corrigé page 33

Un tourniquet, utilisé pour contrôler l'accès aux métros et des manèges de parc d'attractions, est une porte avec trois bras rotatifs à hauteur de la taille. Initialement, les bras sont verrouillés, empêchant les clients de passer à travers. Le dépôt d'une pièce de monnaie ou de jetons dans une fente sur le tourniquet déverrouille les bras, ce qui permet un seul client de passer. Après que le client soit passé, les bras sont verrouillés à nouveau jusqu'à ce qu'une autre pièce de monnaie soit insérée.

**Q1.** Décrire avec le formalisme SysML les exigences du système.

**Q2.** Décrire l'utilisation principale du système.

**Q3.** Traduire le fonctionnement par une machine d'états.



FIGURE 1.25 – Tourniquet

## 1.6 Classification des systèmes

### 1.6.1 Selon la nature de la commande

Les systèmes peuvent être classés dans trois grandes catégories selon la nature de la commande :

**Systèmes non mécanisé :** ces systèmes regroupent tous les systèmes pour lesquels l'utilisateur contribue à la fois à la commande et à l'énergie. Un vélo est un bon exemple de cette famille de système.

**Systèmes mécanisés :** l'utilisateur ne fournit plus l'énergie nécessaire au fonctionnement, son rôle est de commander le système. L'utilisateur est nécessaire, le système n'étant pas capable de décider. Une moto, une voiture, un aspirateur, sont des exemples de système mécanisé.

**Systèmes automatisés :** l'algorithme de décision est implémenté dans la partie commande du système, à partir des consignes de fonctionnement indiquées par l'utilisateur, le système peut évoluer seul. Un robot aspirateur, est un exemple de l'évolution de l'aspirateur. La voiture sans conducteur<sup>5</sup> est un modèle du futur de la voiture automatisée.

### 1.6.2 Selon la la nature des informations

Un autre possibilité de classification est de s'intéresser à la nature des informations échangées

**Systèmes tout ou rien :** les systèmes tout ou rien, sont les systèmes pour lesquels l'évolution du comportement ne dépend que de l'évolution de variables binaires (marche/arrêt, ouvert/fermé, présent/absent, 5V/ 0V, ...). Dans un système tout ou rien, les grandeurs physiques ne sont prises en compte que lors du franchissement des seuils ( $\text{température} > T_{\text{seuil}}$ ).

Le fonctionnement d'un four piloté par une commande tout ou rien peut être décrit par l'algorithme 1

**Systèmes continus :** dans un système continu, on ne s'intéresse pas uniquement au franchissement de seuils des grandeurs physiques, mais à l'évolution temporelle des signaux analogiques que manipulent le système. L'algorithme 2 montre un exemple de régulation d'un four à partir de la mesure continue de la température du four et une commande « proportionnelle » du chauffage.

**Systèmes échantillonnés :** L'informatisation des partie commandes (automate, microprocesseur, microcontrôleur, . . .) qui même si elle a une capacité de calcul et une fréquence de fonctionnement de plus en plus grande ne peut prendre

5. [http://fr.wikipedia.org/wiki/Voiture\\_sans\\_conducteur\\_de\\_Google](http://fr.wikipedia.org/wiki/Voiture_sans_conducteur_de_Google)

en compte l'évolution d'une grandeur physique qu'à des instants particuliers. Il est donc nécessaire d'échantillonner le signal physique, c'est à dire le prélever à des instants  $T_e$  (voir l'algorithme 3).

**Données :**  $T(t)$  : température du four  
**Données :**  $T_{min}$  : seuil mini ;  $T_{max}$  : seuil max  
**Résultat :** *Chauffage* : état binaire vrai/faux  
**début**

```

tant que le four fonctionne faire
  Lire  $T(t)$ 
  si  $T(t) < T_{min}$  alors
    | Chauffage = vrai
  fin
  si  $T(t) > T_{max}$  alors
    | Chauffage = faux
  fin
fin
fin

```

**Algorithme 1 :** Fonctionnement tout ou rien d'un four

**Données :**  $T(t)$  : température du four  
**Données :**  $T_{cons}$  : température de consigne  
**Résultat :** *Chauffage(t)* : réel  
**début**

```

tant que le four fonctionne faire
  Lire  $T(t)$ 
  si  $T < T_{cons}$  alors
    |  $Chauffage(t) = K \cdot (T_{cons} - T(t))$ 
  fin
  sinon
    |  $Chauffage(t) = 0$ 
  fin
fin
fin

```

**Algorithme 2 :** Fonctionnement avec régulation continue d'un four

**Données :**  $T(t)$  : température du four  
**Données :**  $T_{cons}$  : température de consigne  
**Données :**  $i$  : entier numéro de l'échantillon  
**Résultat :**  $Chauffage(t)$  : réel  
**début**

```

tant que le four fonctionne faire
  toutes les  $xx$  millisecondes
  début
    Lire  $T(n \cdot xx)$ 
    si  $T < T_{cons}$  alors
      |  $Chauffage(n \cdot xx) = K \cdot (T_{cons} - T(n \cdot xx))$ 
    fin
    sinon
      |  $Chauffage(n \cdot xx) = 0$ 
    fin
  fin
fin

```

**Algorithme 3 :** Fonctionnement avec régulation échantillonnée d'un four

### Exercice 3- Représentation de la réponse temporelle d'un four *Corrigé page 33*

**Q1.** Tracer l'allure de l'évolution de la température du four et de la commande pour chacun des trois algorithmes

## 1.7 Description générique d'un système automatisé

On peut représenter la structure par le schéma de la figure 1.26.

On retrouve sur ce schéma, les principaux constituants d'un système :

- une partie commande : elle communique avec l'opérateur mais aussi les autres système, elle émet vers la partie opérative les signaux de commande des actionneurs, elle reçoit des capteurs les informations sur le système.
- une partie opérative : elle comporte les différents constituants qui agissent sur la **matière d'œuvre** pour la modifier, les actionneurs nécessaires à ces actions.
- une interface avec l'opérateur : le pupitre ou IHM comporte les constituants qui facilitent le dialogue avec l'utilisateur.
- les capteurs prélèvent sur la partie opérative l'état des différents constituants.
- les pré-actionneurs ou modulateurs, autorisent le passage de l'énergie vers les actionneurs.

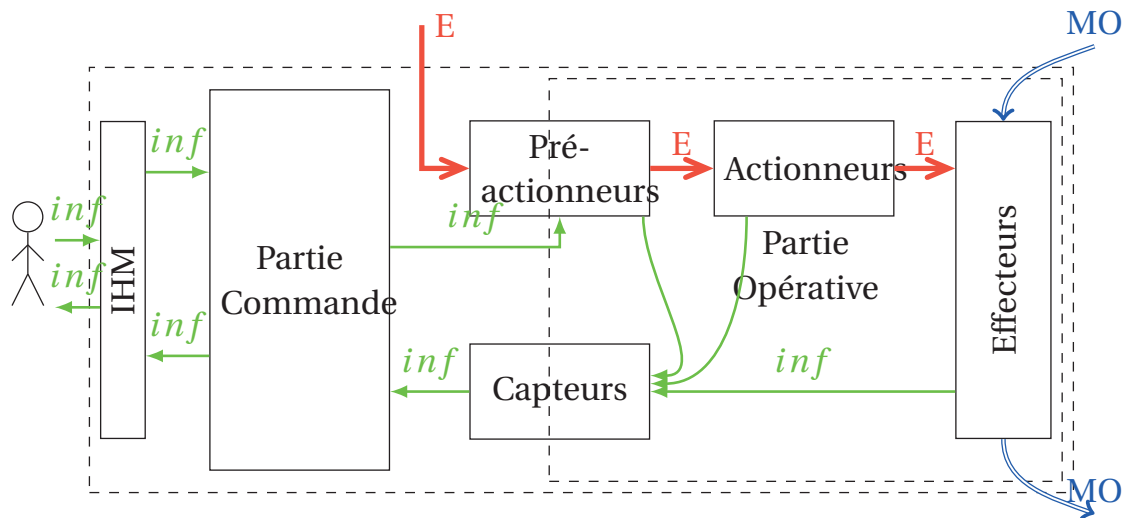


FIGURE 1.26 – décomposition structurelle d'un système automatisé

De manière générale, un système automatisé, peut être décrit par le diagramme SysML de la figure 1.28. Il comporte une partie commande ( régulateur, contrôleur, ...) et une partie opérative (système régulé, système asservis,...).

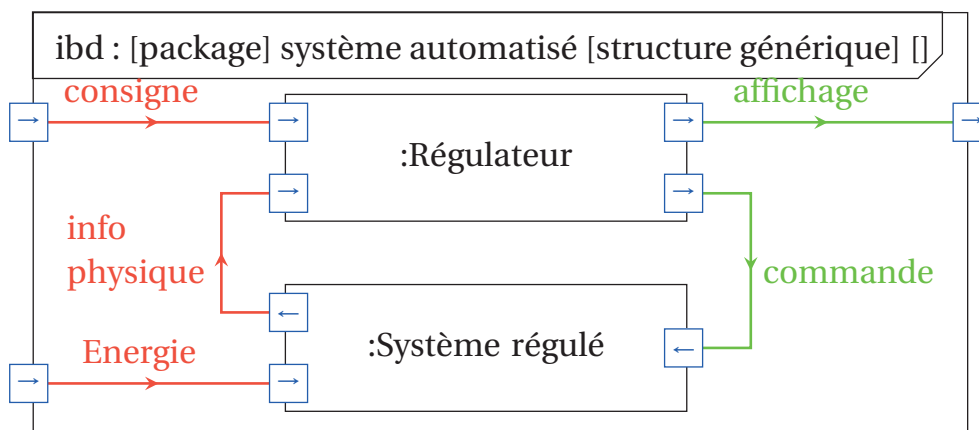


FIGURE 1.27 – Structure générique d'un système automatisé

Cette décomposition fait apparaître deux chaînes :

### a) Chaîne d'information

Cette chaîne est constituée des éléments qui manipulent les informations sous ses diverses formes. On retrouve ainsi

- un constituant chargé de gérer le système : en général ce constituant est construit autour d'un microprocesseur ou d'un microcontrôleur (PC, automate programmable, carte micro programmable, ...) mais peut aussi être réalisée à partir de constituants élémentaires à base d'amplificateurs opérationnels et de composants logiques ou séquentiels ;
- une interface avec l'utilisateur (IHM : interface Homme Machine) chargé de

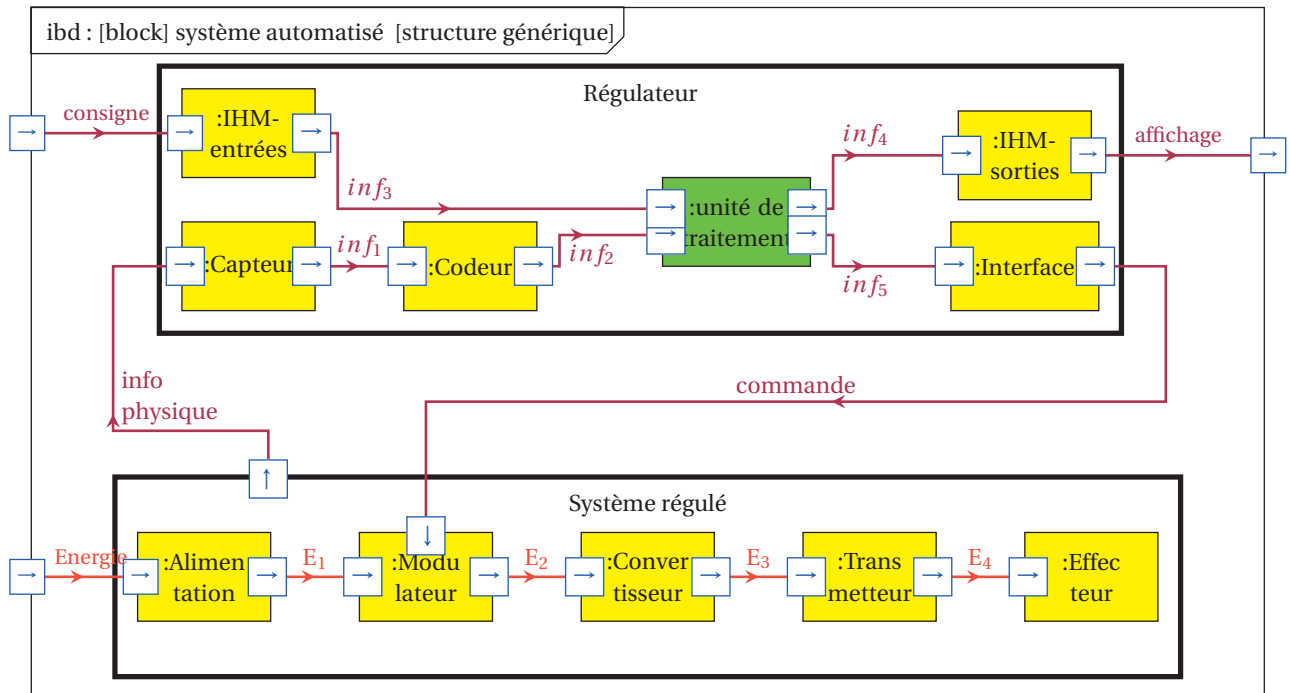


FIGURE 1.28 – Structure générique d'un système automatisé

- communiquer avec l'opérateur aussi bien en entrée qu'en sortie (clavier, écran, HP, ...);
- de capteurs prélevant sur le système des informations physiques (pression, température, vitesse, position, ...);
- de codeurs/convertisseurs chargés de traiter les signaux issus des capteurs (Convertisseur Analogique Numérique, Convertisseur Numérique Analogique, filtre, ...).
- d'interface et différents constituants de communication, dont ceux destinés à communiquer au système physique les ordres de commande.

## b) Chaîne d'énergie

Cette chaîne est constituée des éléments qui participent à la transformation de l'énergie, de l'énergie brute du réseau aux constituants agissant réellement sur le produit.

- l'alimentation regroupe les constituants participant à la conformation de l'énergie du réseau en une énergie utilisable par le système (transformateur, régulateur, ...).
- le modulateur autorise le passage de l'énergie vers les actionneurs et module celle-ci en fonction des ordres de commande envoyés par la chaîne d'information.
- les convertisseurs regroupent les constituants capables de transformer l'énergie en une autre énergie, les moteurs, les vérins, les radiateurs, ... sont des éléments de cette famille.

- transmetteurs : boîte de vitesse, engrenage, poulie courroie, ...
- effecteurs : les éléments agissants

Une autre représentation de ces chaînes fonctionnelles est de les décrire à partir des fonctions réalisées par les différents constituants (fig 1.29).

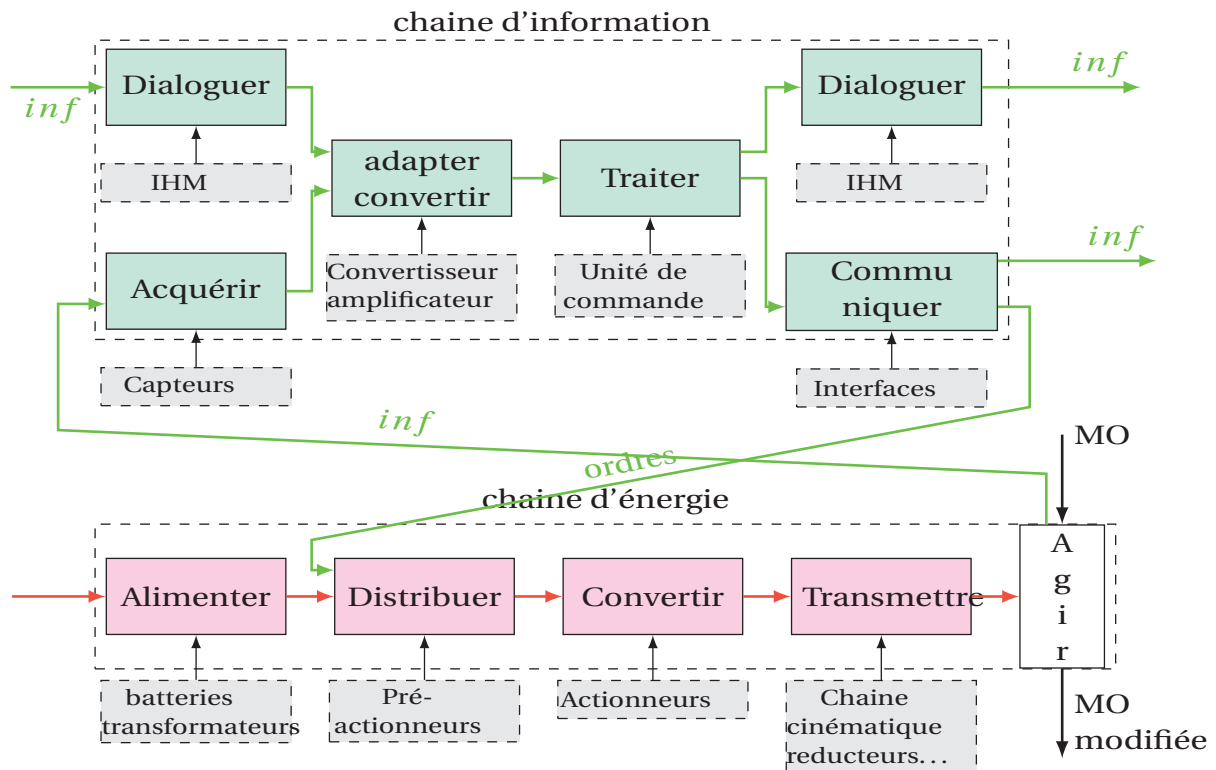


FIGURE 1.29 – Description fonctionnelles des chaînes d'information et d'énergie