

FICHE 00

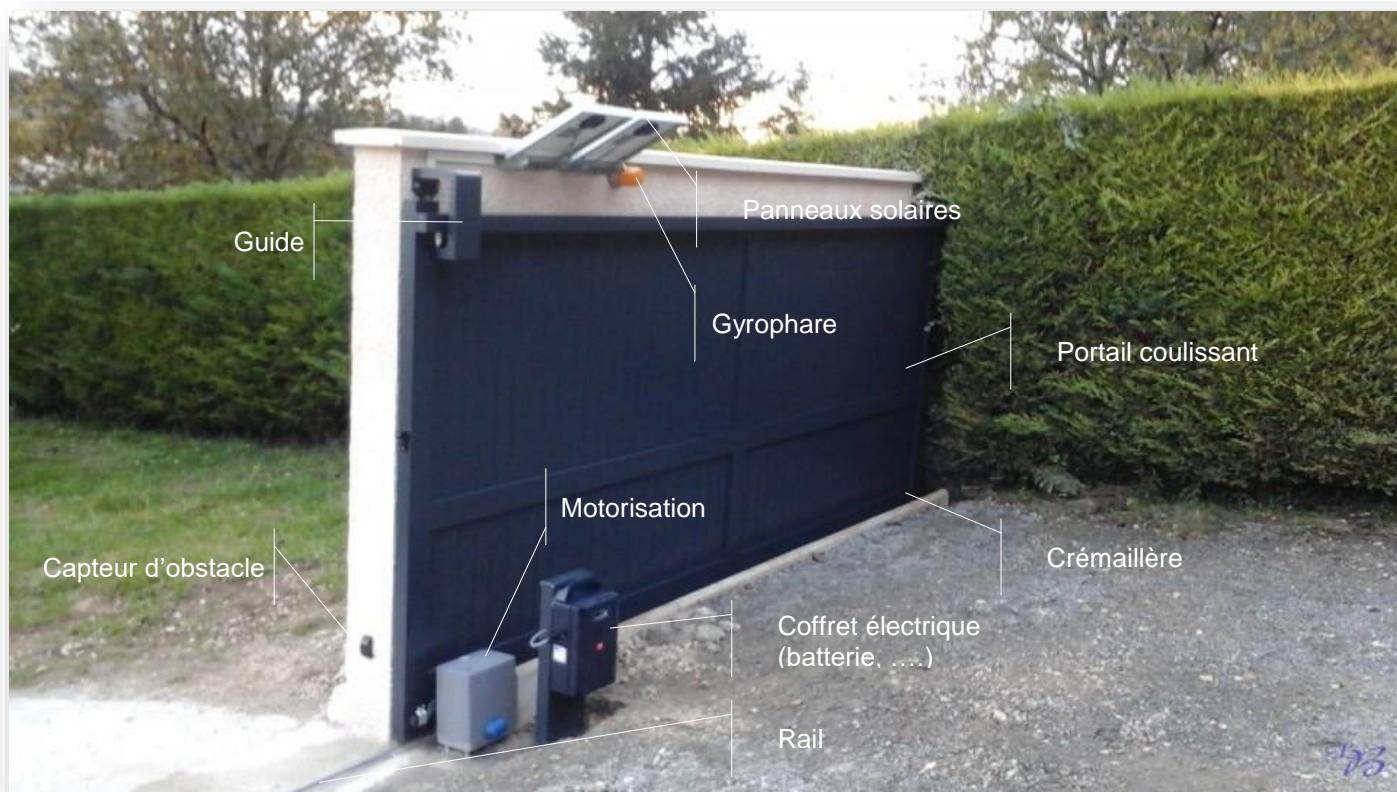
ANALYSE FONCTIONNELLE

site : jeanpaul.bousquet.free.fr

1) Généralités sur les systèmes

Toujours conçu dans le but de répondre à un besoin, un système est une **association structurée de constituants** ayant des relations entre eux. Ces constituants et ces éléments peuvent être nombreux et les interactions de formes différentes (échanges de matières, énergies ou informations) : on parle alors de **système complexe pluri-technologiques**.

Exemple :



2) Introduction

EXPRESSION ET CARACTÉRISATION DU BESOIN, LA BÊTE À CORNES.

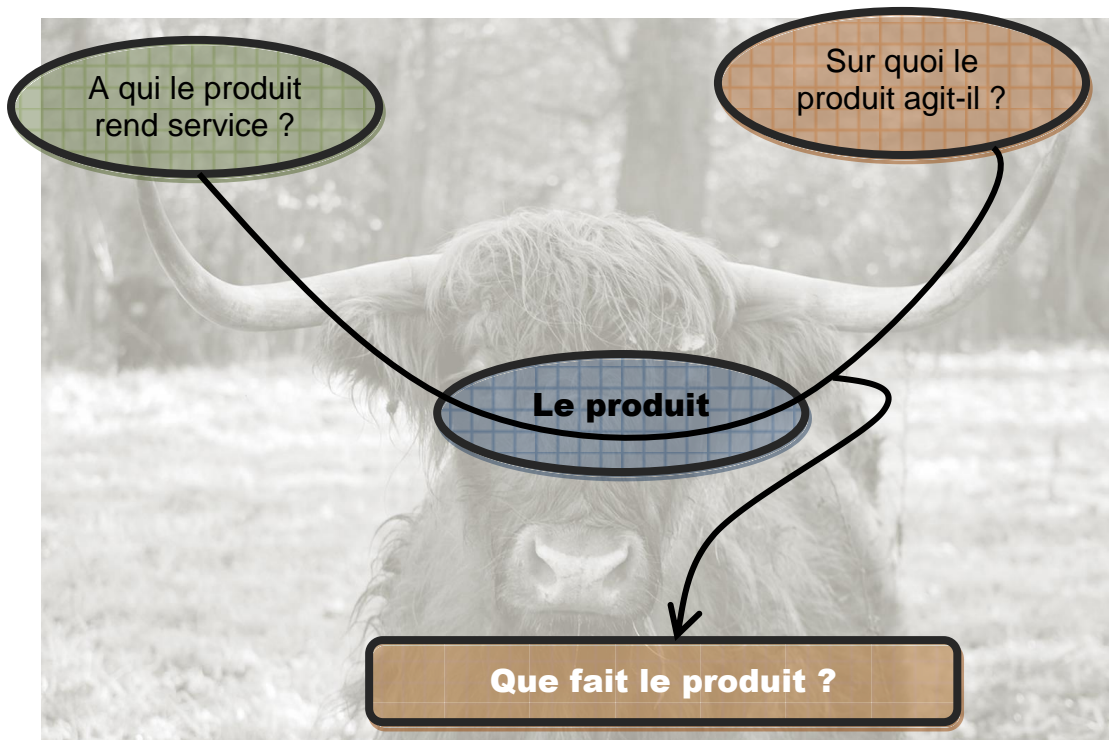
L'expression et la caractérisation du besoin se fait en utilisant la méthode APTE (Application des Techniques d'Entreprise), sous la forme d'un diagramme, appelé Bête à Cornes et présenté en page suivante. Le besoin est une nécessité, un désir exprimé par un utilisateur, par exemple dans le cas qui nous est présenté il aurait pu s'exprimer de la manière suivante :

« Je souhaiterais pouvoir ouvrir et fermer mon portail sans effort à partir de mon véhicule sans avoir à descendre de celui-ci. »

Il exprime tout produit ou service en termes de **services attendus** : **pas en termes de solutions techniques !** puisque à ce stade on ne sait pas encore comment le système sera conçu.

Pour valider le besoin, il faut se poser les questions :

- Pourquoi le produit existe-t-il ?
- Qu'est-ce qui pourrait le faire évoluer ?



Validation du besoin

Nous venons de voir qu'un système est une association de constituants dont le rôle est de réaliser tout ou partie des actions du processus pour réaliser les tâches à accomplir.

Ces actions internes, qui permettent donc d'assurer la réalisation des fonctions de service, sont appelées **fonctions techniques** :

- Détecter la présence d'un obstacle -> Capteur d'obstacle
- Avertir l'utilisateur du fonctionnement -> Gyrophare
- Recharger les batteries -> Panneaux solaires
-

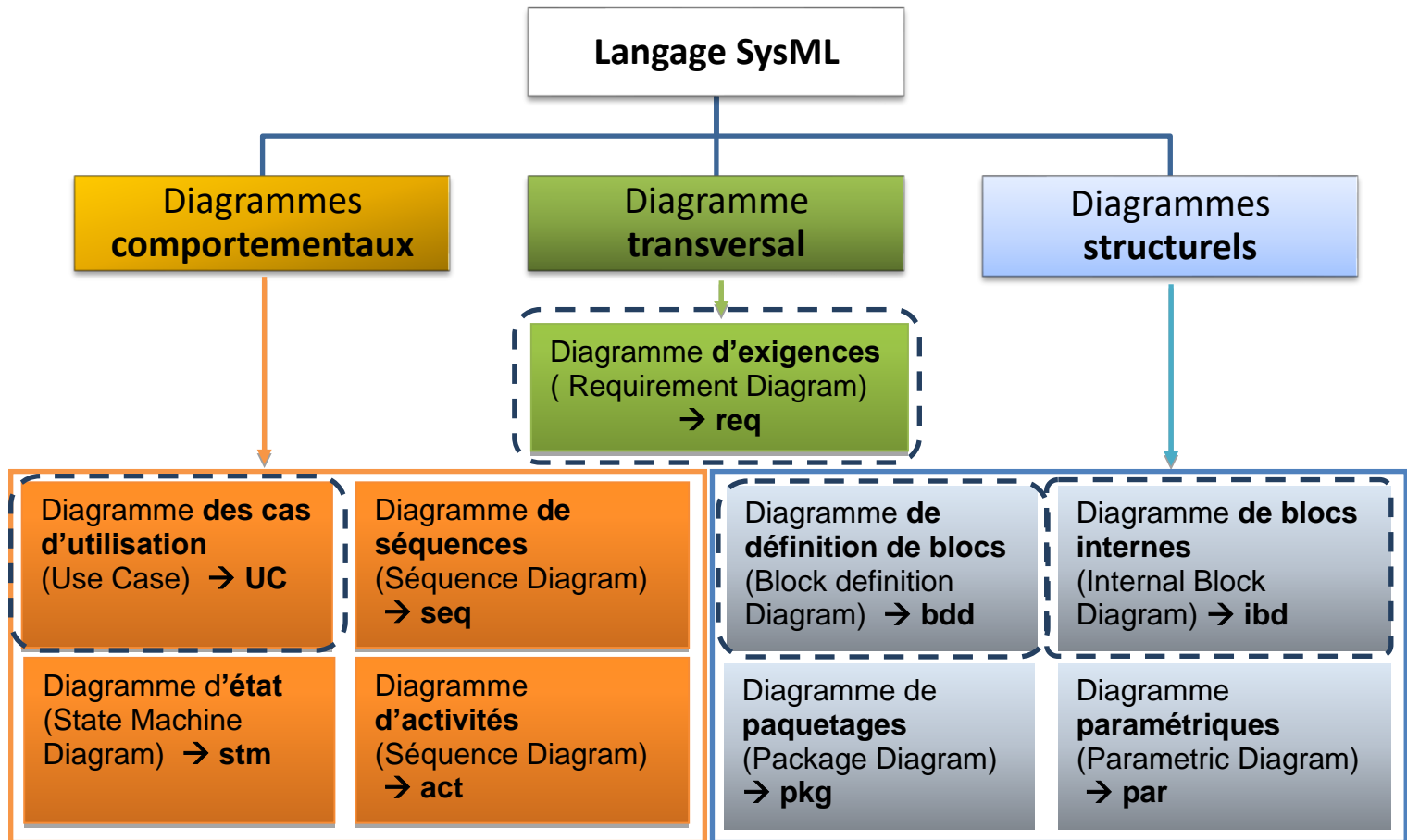
3) Le langage SysML

Sans être de véritables experts vous vous doutez bien que les systèmes techniques actuels sont d'une très grande complexité, tant au niveau de sa structure qu'au niveau fonctionnel, mêlant le plus souvent plusieurs domaines : la mécanique, l'électronique, l'optique et bien d'autres. Ils ne peuvent donc être conçus que par un groupe d'experts de spécialités différentes, parmi lesquels des ingénieurs (en acoustique, vibration, électronique, informatique, mécanique, etc.). Par le passé et encore aujourd'hui, beaucoup d'outils différents sont utilisés en fonction d'un domaine ou d'un autre. L'inconvénient majeur est qu'ils ne sont compréhensibles que par les initiés. Il était donc nécessaire de disposer d'un langage commun à l'ensemble des acteurs.

Cet outil a dû répondre à deux contraintes :

- Être suffisamment simple à comprendre pour que tout le monde puisse l'utiliser sans formation initiale particulière.
- Être suffisamment développé pour ne pas être un frein à la créativité et donc être utilisable dans une phase de développement par des spécialistes.

Ce langage est constitué de 9 blocs :



Point de vue comportemental :

Comment le système évolue ?

- Le diagramme des **cas d'utilisation** (*Use Case Diagram*, indicateur **uc** ou **ucd**) décrit principalement les objectifs poursuivis par l'acteur primaire à travers le système et donc les interactions de l'acteur primaire (typiquement l'utilisateur) avec le système étudié. Des acteurs secondaires peuvent également être indiqués sur ce diagramme.
- Le diagramme de **séquence** (*Sequence Diagram*, indicateur **seq**) décrit l'enchaînement des messages passés entre des instances de blocs en interaction.
- Le diagramme d'**états** (*State Machine Diagram*, indicateur **stm**) décrit les différents états du système ainsi que les transitions possibles entre ces différents états.
- Le diagramme d'**activités** (*Activity Diagram*, indicateur **act**) décrit l'enchaînement des tâches (appelées « flux de contrôles ») ainsi que l'utilisation des données (appelées « flux de données ») dans le cadre d'un processus.

Un point de vue structurel :

Comment est « architecturé » le système ?

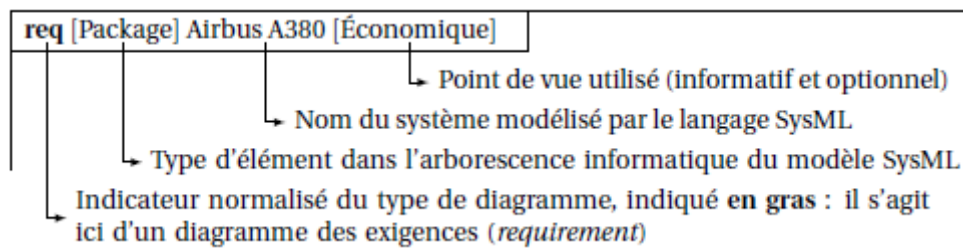
- Le diagramme de **définition de blocs** (*Block Definition Diagram*, indicateur **bdd**) décrit l'architecture matérielle, logicielle, etc. du système.
- Le diagramme de **bloc interne** (*Internal Block Diagram*, indicateur **ibd**) décrit l'organisation interne d'un élément et les interactions entre ses composants selon des flux d'énergie, de matière ou d'information.
- Le diagramme **paramétrique** (*Parametric Diagram*, indicateur **par**) est un cas particulier du diagramme de blocs internes qui décrit les contraintes internes du système à l'aide d'équations liant des propriétés de blocs.
- Le diagramme de **paquetages** (*Package Diagram*, indicateur **pkg**) décrit l'organisation logique du modèle et les relations entre paquetages en fonction des besoins (facteur d'échelle) afin de clarifier la lecture du modèle via des macro-éléments.

Un point de vue transversal,

A quoi doit-il répondre ?

– Le diagramme d'**exigences** (*Requirement Diagram*, indicateur **req**) est associé à ce point de vue et il décrit ce que doit réaliser le système ainsi que les contraintes auxquelles il doit se soumettre.

Chaque diagramme SysML représente un élément particulier du modèle selon un certain point de vue : afin de le repérer, chaque diagramme comporte un « cartouche » présenté sur la figure ci-dessous, positionné sur la partie supérieure gauche du cadre.



Source : Science Industrielle de l'ingénieur MPSI-PCSI-PTSI Edition Vuibert Prépas

3.1) Diagramme Comportemental

3.1.1) Diagramme des Cas d'utilisation

3.1.1.1) Son rôle

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).

Il répond à la question : "quels services rend le système ?".

3.1.1.2) Représentation graphique

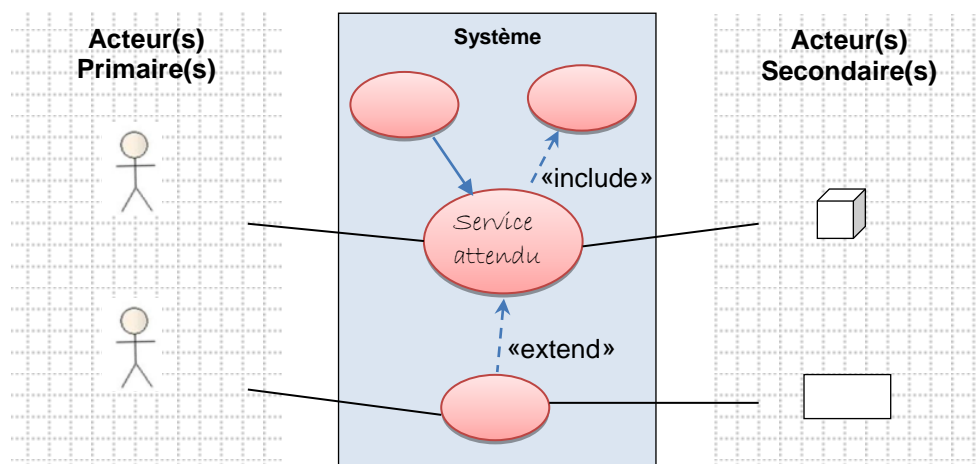
On trace un cadre délimitant le **système**.

Celui-ci contient un ensemble de **cas d'utilisation** inscrits à l'intérieur de « bulles/ovales » (Elles peuvent aussi être liées entre elles). On y décrit les actions réalisables par le système (les services rendus par le système aux acteurs, sous forme de verbe à l'infinitif plus compléments).

Des deux côtés gravitent les **acteurs**, on entend par ce terme toute entité externe au système étudié qui interagit (réalise/accomplit ou subit) de façon autonome avec le système.

- À **gauche** on place les *acteurs principaux dits primaires* (Par exemple : un humain, un serveur, un autre système, un animal) **à qui le système rend service**.
- À **droite** les *acteurs secondaires éventuels* qui subissent / réagissent avec le système (**Entités aidant le système à remplir sa mission**).

Un acteur non humain est représenté par un rectangle ou un cube.



Les acteurs peuvent être reliés entre eux soit par un trait soit par une flèche unidirectionnelle (un acteur agit sur l'autre).

La relation d'association est un chemin de communication entre un acteur et un cas d'utilisation	Le cas d'utilisation peut incorporer (option) celui placé à l'origine de la flèche.	Le cas d'utilisation incorpore systématiquement celui placé à l'extrémité de la flèche.	Ils relient des cas d'utilisation descendants qui héritent de la description d'un cas d'utilisation supérieur (parent commun)
Lien d'association	Lien d'extension	Lien d'inclusion	Lien de spécialisation

Nota :

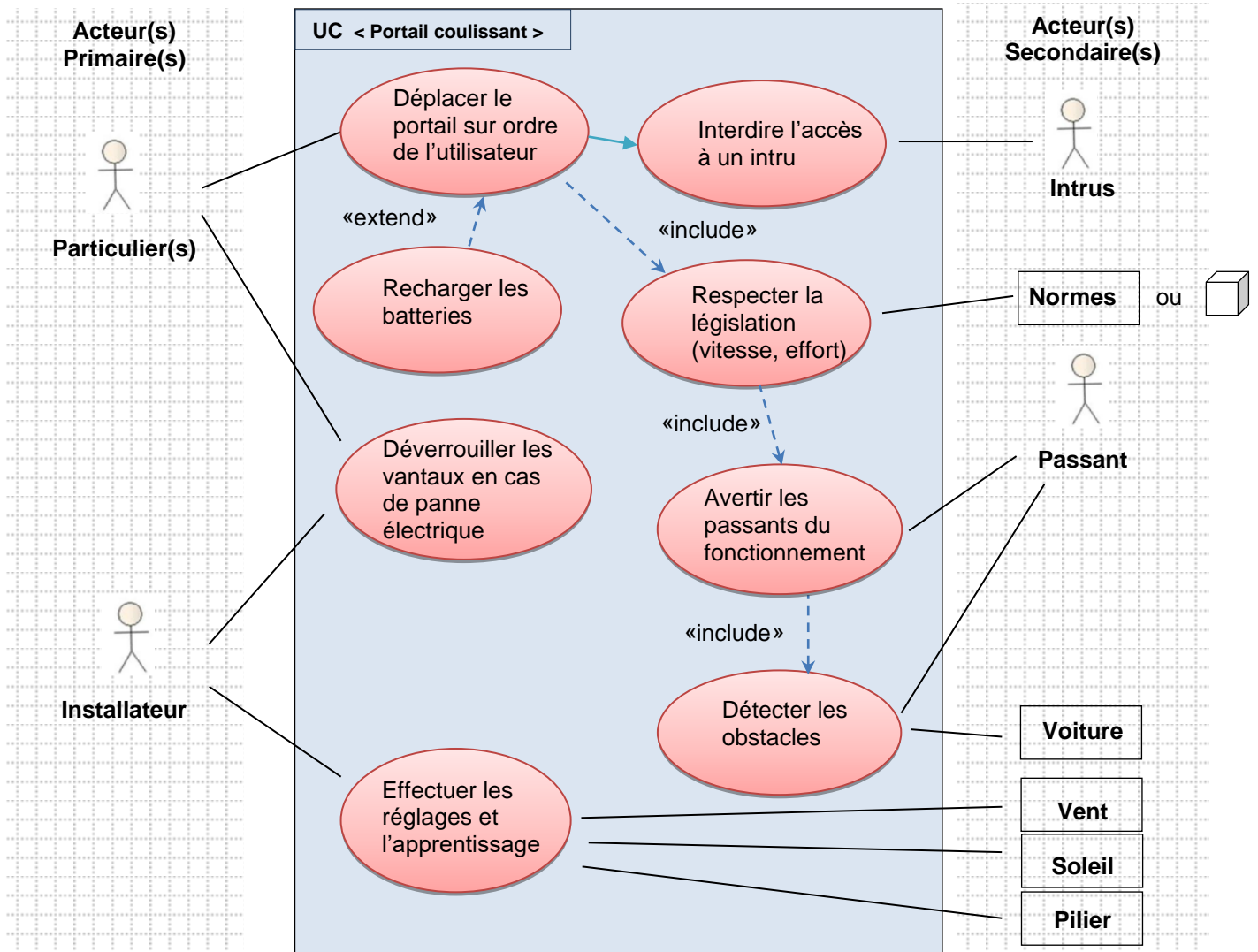
Ce diagramme ne doit indiquer ni la manière dont il va assurer les services, ni les solutions technologiques envisagées.

Plusieurs diagrammes d'utilisations peuvent être établis pour un système afin d'en améliorer la compréhension.

3.1.1.3) Exemple

Nota : le client n'a pas les compétences pour réaliser l'installation et la maintenance du système.

Dans notre cas, l'autonomie énergétique est une option du système.



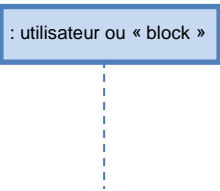

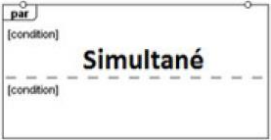
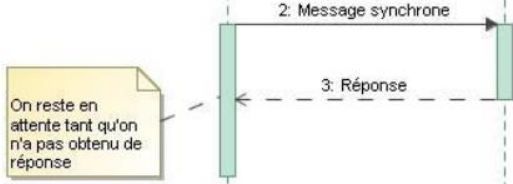

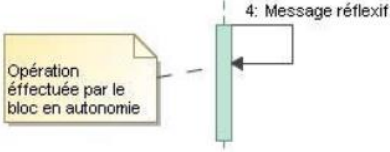












3.2.1) Diagramme de Séquence

3.2.1.1) Son rôle

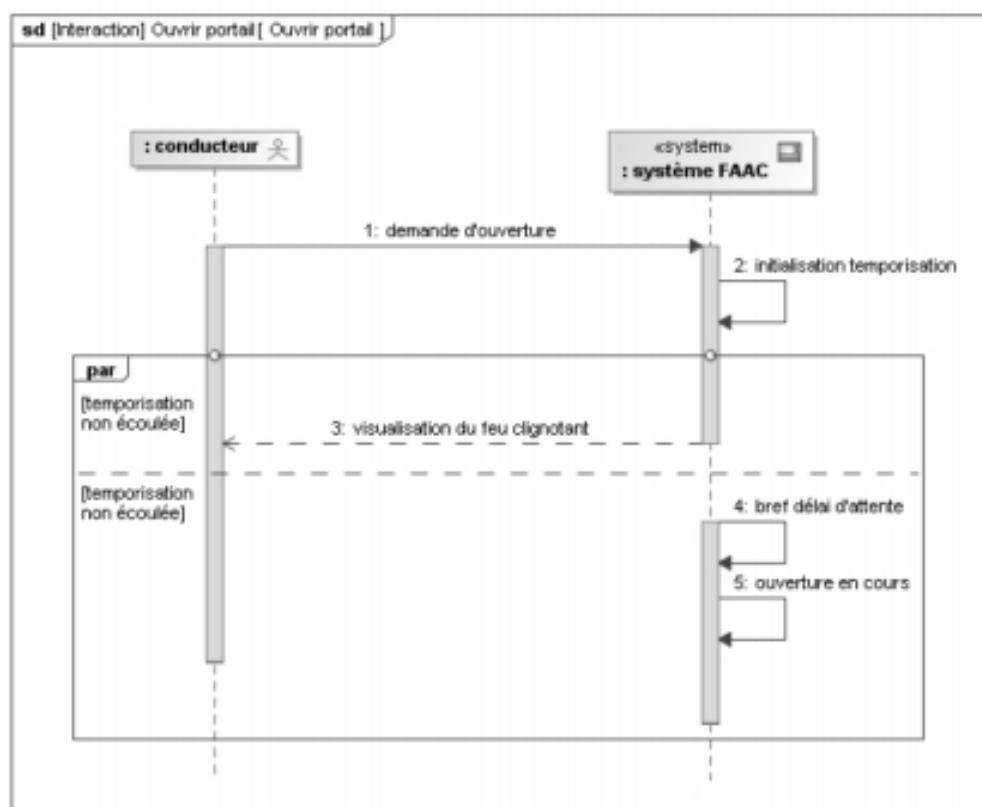
Il répond à la question : "Comment est réalisé ce cas d'utilisation ?", il décrit l'enchaînement séquentiel des interactions. Vous pouvez y observer le comportement du système. Il peut en exister plusieurs.

3.2.1.2) Représentation graphique

Chaque élément en haut (block ou objet) est un élément qui participe à la séquence. Les lignes verticales en pointillés sont les lignes de vie et représentent sans échelle le temps qui s'écoule. Les flèches horizontales représentent l'activité, elles sont toujours accompagnées d'un message numéroté (texte).

	<p>Élément structurel</p> <p>Ligne de vie</p>	 <p>Message Aller Pas de message Retour</p>						
	<p>Exécution en parallèle</p>	 <p>Message Aller Message Retour obligatoire</p>						
	<p>Boucle, pouvant s'exécuter plusieurs fois tant que la condition est vraie</p>	 <p>Comportement interne (test ...) Temps s'écoulant de haut en bas</p>						
	<p>Le 1^{er} bloc est exécuté si la condition est vraie sinon c'est la condition 2 qui est exécuté.</p>	<p>Fragment : Longueur de la séquence</p>  <p>Période d'activité du composant</p>						
	<p>Optionnel : Le fragment ne s'exécute que si la condition fournie est vraie</p>	<table border="1"> <tr> <td> <p>1 : xxx</p>  </td><td> <p>2 : xxx</p>  </td><td> <p>3 : xxx</p>  </td></tr> <tr> <td> <p>La flèche est pleine → attend d'une réponse</p> </td><td> <p>La flèche est vide → n'attend de réponse</p> </td><td> <p>Message de réponse du récepteur suite à un message synchrone</p> </td></tr> </table>	<p>1 : xxx</p> 	<p>2 : xxx</p> 	<p>3 : xxx</p> 	<p>La flèche est pleine → attend d'une réponse</p>	<p>La flèche est vide → n'attend de réponse</p>	<p>Message de réponse du récepteur suite à un message synchrone</p>
<p>1 : xxx</p> 	<p>2 : xxx</p> 	<p>3 : xxx</p> 						
<p>La flèche est pleine → attend d'une réponse</p>	<p>La flèche est vide → n'attend de réponse</p>	<p>Message de réponse du récepteur suite à un message synchrone</p>						

3.1.1.3) Exemple



3.2) Diagramme Structurel

3.2.1) Diagramme de définition de blocks (bdd)

3.2.1.1) Son rôle

Ils constituent la base de la description structurelle, il peut désigner le système « **system of interest** », un de ses sous-ensembles « **subsystem** », un composant élémentaire ne pouvant pas être décomposé « **block** » ou encore un logiciel « **software** » ou un matériel « **hardware** » ou un acteur « **actor** » ou une de ses fonctionnalités.

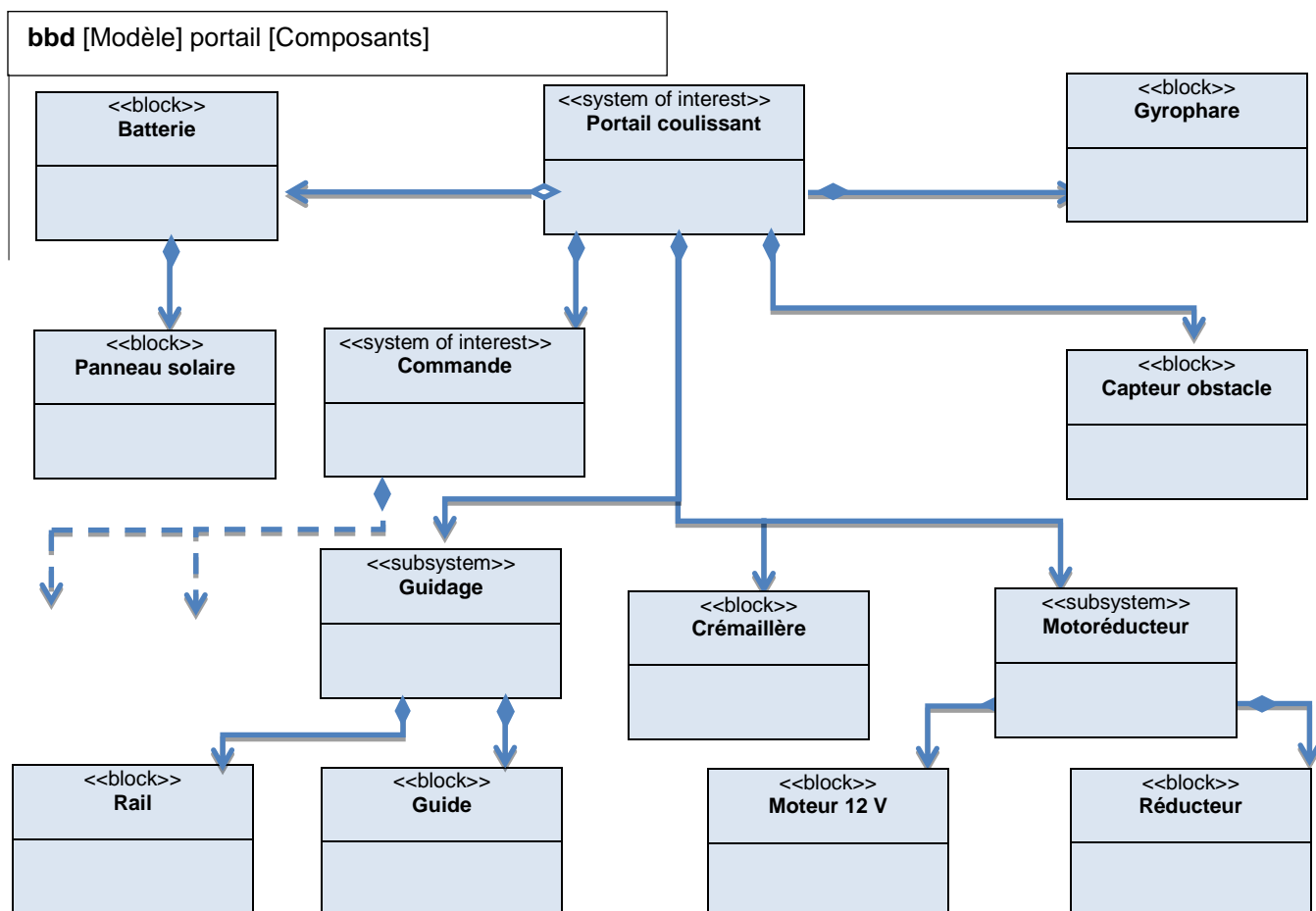
3.2.1.2) Représentation graphique

On trace un cadre « étagé » dans lequel on définit des propriétés particulières (infos,...) concernant chaque composant. De celui-ci part ou arrive des relations. Ces relations sont de deux types :

La relation indique qu'un élément est structurellement indispensable à l'autre. Le losange plein est du côté du système principal, l'autre du côté du composant	Idem qu'à côté mais avec un sens moins fort, elle indique que le composant est présent de manière optionnelle. Exemple : les bagages dans une voiture.
La relation pourrait se traduire par : <i>block1</i> a un/une <i>block2</i>	
Relation de composition	Relation d'agrégation

<<block>> Motoréducteur à courant continu
Constraints $C_m = k_t \cdot i$ $d\omega_m/dt = (C_m - f \cdot \omega_m) / J_{eq}$ $u = R \cdot i + L \cdot di/dt + k_e \cdot \omega_m$
Parts Moteur CC Vitesse nominale = 3000 rpm
properties Couple nominal = 1 N.m Vitesse nominale = 3000 rpm
Values $f_{eq} = 7e-5$ $J_{eq} = 6e-6$ $k_e = 0.02$ $k_t = 0.02$ $L = 7e-4$ $R = 5$

3.2.1.3) Exemple



3.2.2) Diagramme de définition de blocks interne (ibd)

3.2.2.1) Son rôle

- Il représente l'intérieur d'un bloc (block) et les échanges de matière /énergie/information (MEI) entre blocs de même niveau grâce aux ports de flux (petit carré avec une flèche).
- Il 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.
- Il permet de représenter les liens entre les blocs de même niveau.

3.2.2.2) Représentation graphique

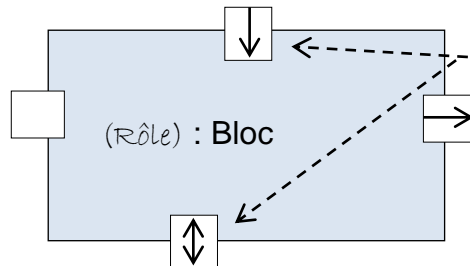
Les blocs sont reliés entre eux par des connecteurs.

La logique de connexion entre blocs est précisée par des ports.

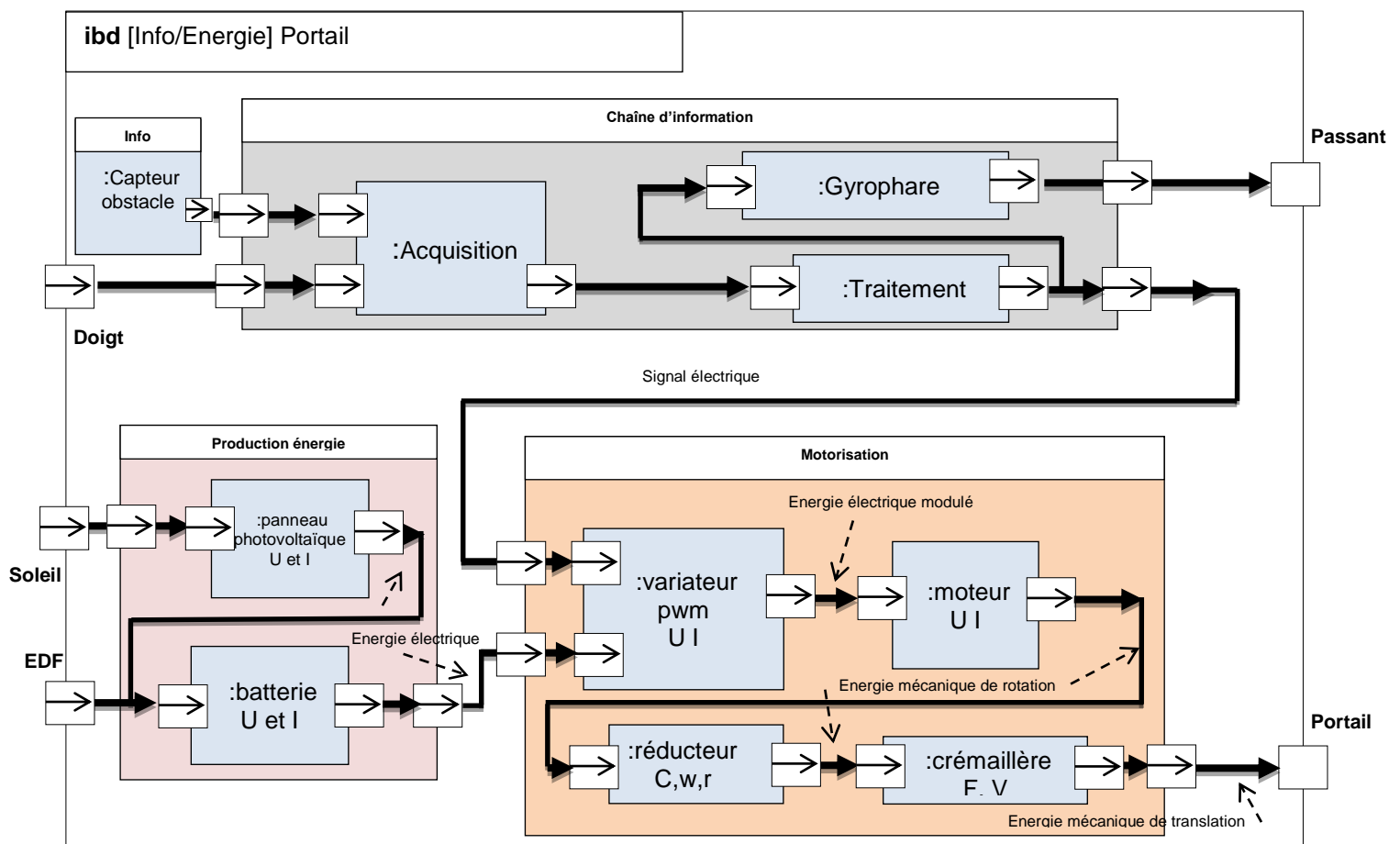
Attention à bien faire la différence entre port standard et port de flux.

Port de flux : canal d'Entrée/Sortie par lequel transite de la Matière, de l'Energie ou de l'Information (MEI).

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



3.2.2.3) Exemple



3.3) Diagramme Transversal

3.3.1) Diagramme d'existence

3.3.1.1) Son rôle

Le diagramme des exigences permet de répertorier et d'analyser les contraintes et les performances du système. Il permet de structurer les besoins.

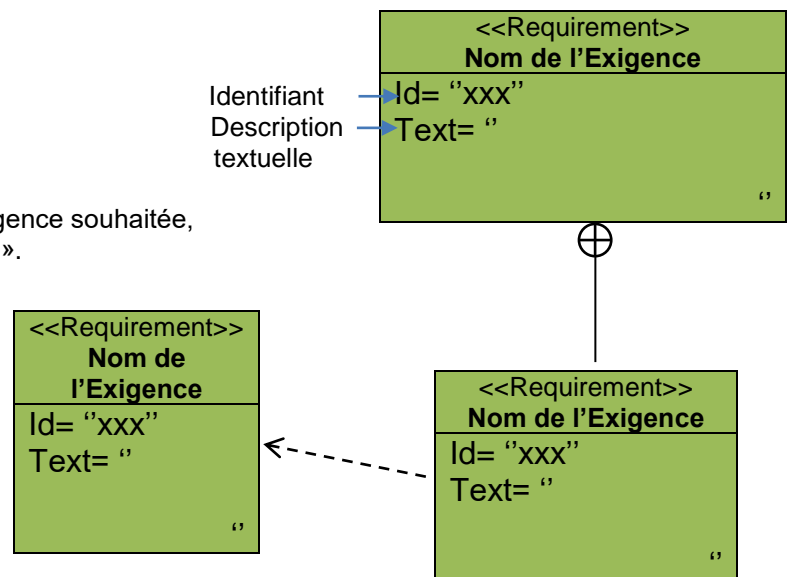
Ce diagramme est un outil de **représentation des fonctionnalités du système**. C'est un moyen de communication entre les concepteurs et les clients du système.






Il est similaire à la première page d'une notice de montage, indiquant la liste des éléments et des pièces à assembler.

3.3.1.2) Représentation graphique

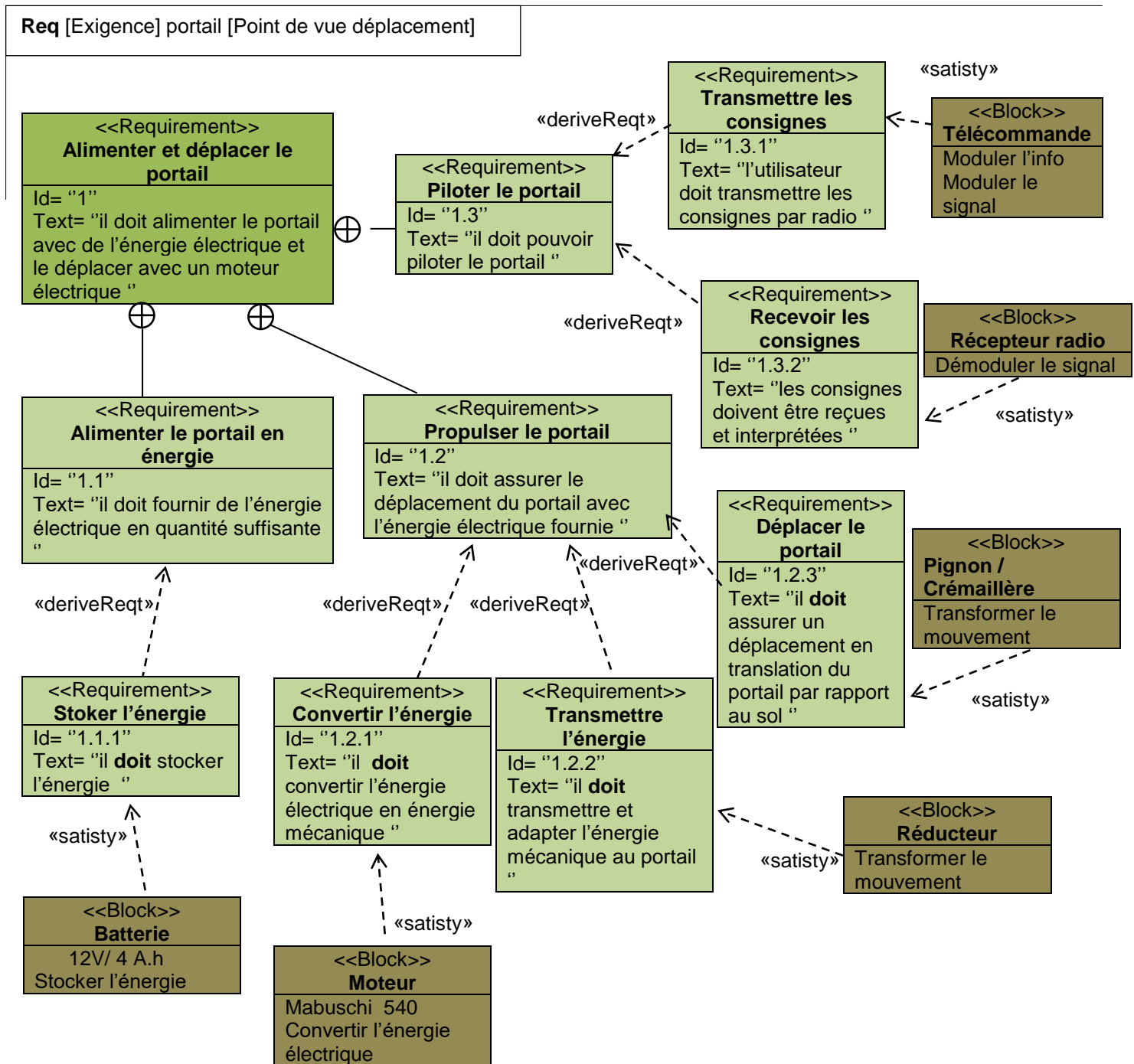
On trace un cadre dans lequel on définit :

- L'exigence :
- Un numéro Id : 1, 2, ...
Les blocs dépendant du bloc
parent Id : 1.2, 1.2, 1.3,
- Un texte explicatif qui explique l'exigence souhaitée, la phrase contient le verbe « devoir ».



	 «refine»	 «verify»	 «deriveReq»
Lien de contenance			
la contenance permet de décomposer une exigence composite en plusieurs exigences unitaires	Lien entre 2 exigences le raffinement consiste en l'ajout de précisions, par exemple de données quantitatives. Entre exigence – élément comportemental (cas d'utilisation, diagramme d'états, etc.) : permet de mettre en œuvre l'exigence.	exigence – cas de test : « verify » permet de vérifier une exigence	la dérivation consiste à relier des exigences de niveaux différents. L'exigence fille est déduite de l'exigence mère. Cette exigence n'est pas émise par le client en tant que tel, mais peut-être une exigence induite par plusieurs exigences.
		 «satisfy» exigence – bloc d'architecture : «satisfy» ; satisfaire l' exigence .	

3.3.1.3) Exemple



NOTA : Au niveau de la description « Text » il est possible aussi de chiffrer une exigence
 Au niveau du titre, soit sous forme d'une fonction (Top) soit un titre significatif (moins bien)

<<Requirement>> Déplacer le portail Exigence de vitesse
Id= "1.2.3" Text= "La vitesse de translation ne devra pas dépasser 10 cm/s"

<<Requirement>> Transmettre l'énergie Couple sortie réducteur
Id= "1.2.2" Text= " 50 Nm à 100 tr.min ⁻¹ "

4) Autre approche

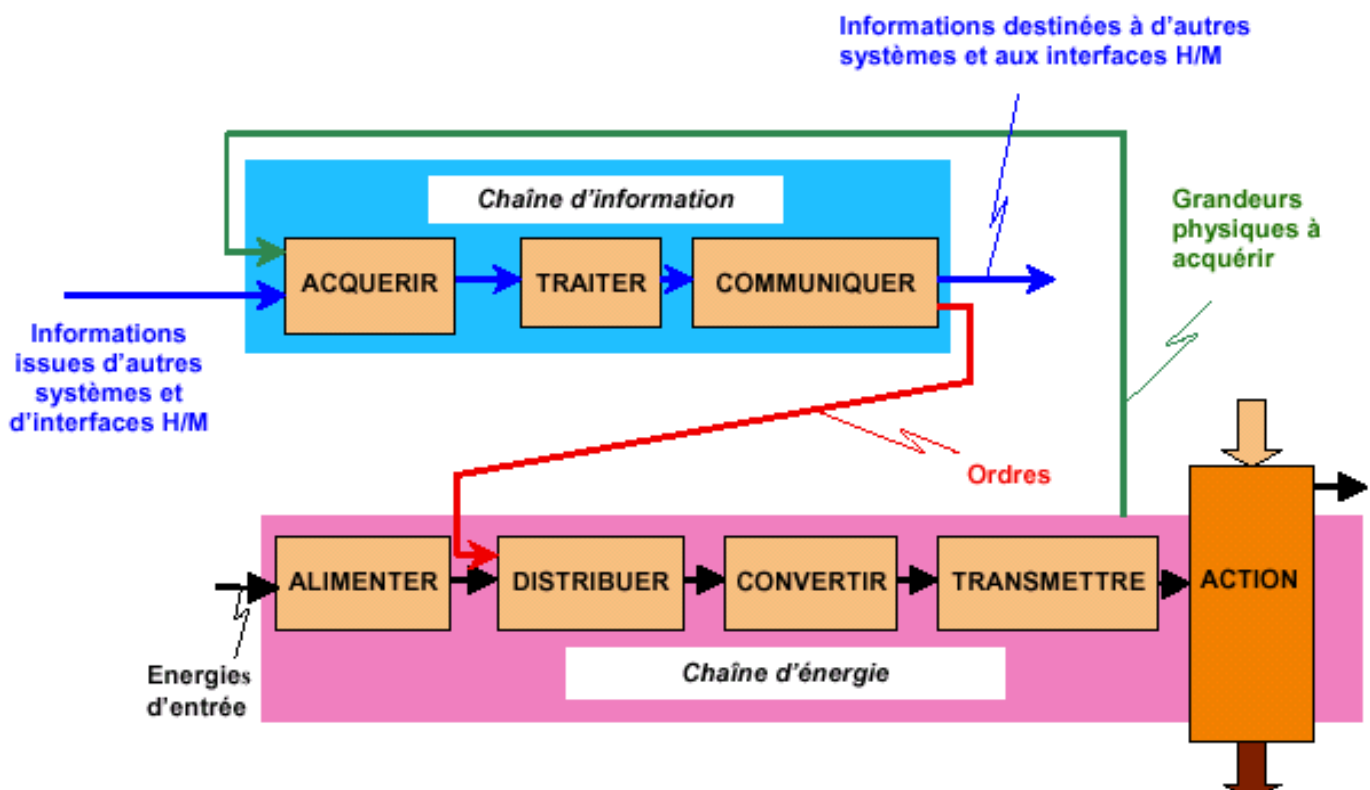
4.1- Chaîne d'énergie et chaîne d'information

Le processus de transformation de la matière d'œuvre par un système industriel complexe comporte plusieurs activités successives ou simultanées.

Ces activités correspondent souvent à une fonction technique ou une fonction de service à réaliser.

L'étude globale des systèmes conduit à distinguer 2 entités :

- la **chaîne d'information** (qui transfère, stocke, transforme l'information) ;
- la **chaîne d'énergie** (qui transforme l'énergie et permet d'agir sur le système physique)



Remarque :

Tous les systèmes n'utilisent pas la totalité des fonctions seulement les plus complexes.

Une analyse plus fine de ces 2 chaînes nous conduit à distinguer les différents constituants qui vont faire l'objet d'une étude concrète :

Chaîne d'ENERGIE

Ordre de la partie commande

Sources d'énergies

Énergies :
- électrique
- hydraulique
- pneumatique

Énergies :
- mécanique
- lumineuse
- thermique

Énergies disponibles pour les actions demandées par le cahier des charges

Alimenter et protéger

Distribuer

Convertir

Transmettre

-Prise réseau



-Conducteurs et Câbles
-Pile, accumulateur



-Transformateur



-Disjoncteur



-Fusible



-Sectionneur



Son rôle est de distribuer, sur ordre de la partie commande, l'énergie utile aux actionneurs.

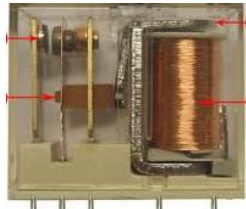
-Contacteur



-Distributeur
- pneumatique
- hydraulique



-Relais



Son rôle est de transformer l'énergie distribuée en énergie mécanique (de translation ou de rotation)

-Moteurs :
* à courant Alternatif



* à courant continu



-Vérins :
*pneumatiques



* hydrauliques



-Lampes d'éclairage
- ... Variateur
Démarreur électronique

Son rôle est d'adapter et de transmettre l'énergie mécanique délivrée par l'actionneur pour la rendre utilisable par l'effecteur*.

-Assemblages démontables



-Guidages en rotation et translation



-Accouplements, embrayages, limiteur de couple, frein

-Transmission par :
* poulies-courroie
* roues et chaîne
* engrenages



-Systèmes vis-écrou et à transformation de mouvement



Chaîne d'INFORMATION

Grandeurs
Physiques,
Consignes

Images
informationnelles
utilisables

Informations
traitées

Ordres,
messages

Acquérir

Traiter

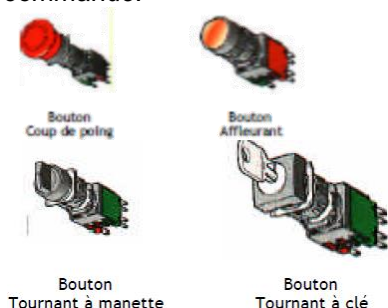
Communiquer

Son rôle est de prélever une grandeur physique et d'en produire une image exploitable par la partie commande.



Interface H/M

Son rôle est de traduire la consigne d'un utilisateur en une image exploitable par la partie commande.



Son rôle est, à l'aide du programme implanté, de traiter les informations en provenance des capteurs et de l'interface H/M afin d'émettre les ordres destinés aux pré-actionneurs des différentes chaînes d'énergie.

Elle envoie aussi des signalisations à l'interface M/H qui seront traduits en signaux lumineux et/ou sonores

-Automates programmables



-Ordinateurs
-Microcontrôleurs



-Modules logiques programmables

-Circuits de commande câblés



Son rôle est de permettre à l'opérateur d'être informé sur l'état du système.

-Commandes TOR



-Interfaces homme-machine



-Liaison utilisant le mode de transmission série

-Liaison utilisant le mode de transmission parallèle



-Réseau

