

C 1.10

C 1.11

Acquérir une méthode de recherche exploratoire d'informations et de leur exploitation par l'utilisation avancée des moteurs de recherche.

Adopter progressivement une démarche raisonnée dans la recherche d'informations.

Il est possible de trouver beaucoup d'informations sur les différents sites constituant l'Internet. Des milliers de sites, des millions d'informations, Internet serait une illusion s'il n'existait le moyen de retrouver une aiguille dans une botte de Web. Des outils de recherche ont été conçus à cet effet.



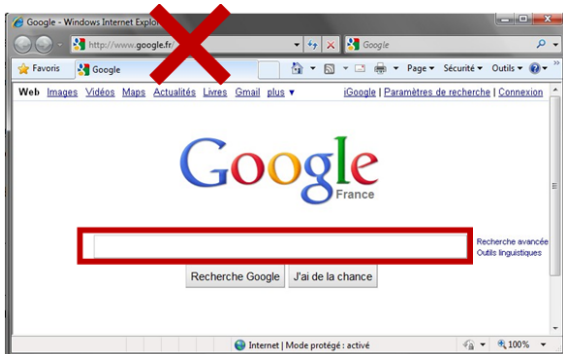
Les moteurs de recherche

L'outil le plus utilisé est **le moteur de recherche**. Il en existe plusieurs [1]. Ils sont alimentés par des robots qui enregistrent nuit et jour le Web.

Quelque soit le moteur de recherche utilisé, le principe d'utilisation reste le même : Il suffit de saisir à l'endroit prévu à cet effet [2] les mots les plus significatifs concernant le sujet recherché : on les appelle les **mots-clés**.

Pour plus d'informations sur le fonctionnement des moteurs de recherche, va sur <http://youtu.be/BNHR6IQJGZs>

Pour affiner la recherche, il est aussi possible d'utiliser des opérateurs logiques [3].



tous les mots suivants :	Saisissez les mots importants : <code>terrier tricolore</code>
ce mot ou cette expression exact(e) :	Ajoutez des guillemets autour des mots exacts : <code>"terrier"</code>
l'un des mots suivants :	Saisissez OR entre tous les mots à inclure : <code>miniature OR standard</code>
aucun des mots suivants :	Placez un signe - (moins) devant les mots à exclure : <code>-ronqueur, -"Jack Russell"</code>

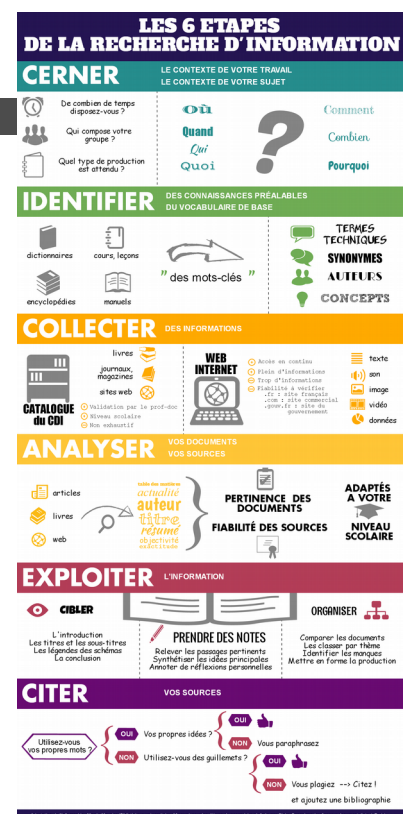
[3]

Méthode de recherche



1. CERNER le contexte du travail et du sujet
2. IDENTIFIER des connaissances, du vocabulaire
3. COLLECTER des informations
4. ANALYSER les documents et les sources
5. EXPLOITER l'information
6. CITER vos sources

source : <https://fenetresur.wordpress.com/2014/10/19/infographie-les-etapes-de-la-recherche-dinformation/>



Les informations publiées sur Internet ne sont pas toutes fiables car chacun est libre d'ajouter des contenus à des sites (notamment sur les sites personnels, forums, blogs, ...).

Pour évaluer la fiabilité d'un site, pose-toi 6 questions :

1. Qui est l'éditeur ? *Est-ce une encyclopédie, un journal, une revue ?*
2. Qui est l'auteur ? *Le site propose-t-il son nom et/ou son adresse mail ?*
3. La ressource est-elle neutre ? *Est-ce que je trouve des faits et des explications ou l'opinion de l'auteur ?*
4. La ressource est-elle détaillée ? *Est-ce que des statistiques, des références, ... sont données ?*
5. La ressource est-elle fiable ? *Est-ce que je peux trouver la même information sur un autre site ?*
6. La ressource est-elle à jour ? *À quelle date l'information a été écrite ? Le lien est-il toujours actif ?*



Source : <https://fr.brainpop.com/technoingenierie/technologie/informationetdelacommunication/ressourcesenligne/>

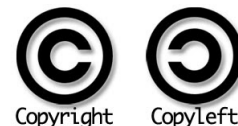


C 4.3

S'engager dans un projet de création et publication sur papier ou en ligne utile à une communauté d'utilisateurs dans ou hors de l'établissement qui respecte droit et éthique de l'information.

Tout ce qui se trouve sur Internet a été déposé par quelqu'un, qui en est le **propriétaire**. Ce dernier peut autoriser ou non son utilisation selon deux principes qui régissent le **droit d'auteur** :

- Le copyright permet une utilisation payante ou autorisée
- Le copyleft est libre de droit



Il est donc indispensable de vérifier les droits lorsque l'on copie un document numérique (texte, image, photos, vidéo, musique, ...).

Entre ces positions extrêmes que représentent le Copyright et le Copyleft, et face au besoin des créateurs désireux de communiquer aux internautes ce qui est possible (ou non) de faire avec leur œuvre, un outil simple a été créé :

le **Creative Commons**



Licence Creative Commons	Bouton	Usage commercial permis?	Nouvelle version permise?
Attribution			
Partage à l'identique de la licence originale (Share Alike)			 Mais sous licence originale
Aucune d'œuvre dérivée (No Derivatives)			
Non-commerciale (Non-Commercial)			 La nouvelle œuvre et ses versions subséquentes devront être non-commerciale
Non-commerciale (Non-Commercial) + Partage à l'identique (Share alike)			 La nouvelle œuvre et ses versions subséquentes devront être non-commerciale
Non-commerciale (Non-Commercial) + Pas d'œuvre dérivée (No Derivatives)			

Pour plus d'information : <https://framablog.org/2008/03/11/education-b2i-creative-commons/>



- En cas de reprise d'un contenu, il est essentiel de **vérifier les termes exacts de la licence** accordée par l'auteur de l'œuvre.
- Le recours à des licences connues et très répandues comme les Creative Commons évite la répétition de cette lecture parfois complexe et fastidieuse.
- Lorsque l'auteur n'a pas exprimé la façon dont il souhaitait être cité, il est recommandé de **citer son nom**, et éventuellement **la source de l'œuvre** (nom du site, URL), et lorsque le nom n'est pas communiqué, d'indiquer au moins la source.

<http://eduscol.education.fr/internet-responsable/ressources/legamedia/contenus-ouverts-ou-libres.html>



Le besoin, c'est quoi ?



Le besoin est un sentiment de manque, une nécessité ou un désir ressenti par une personne.

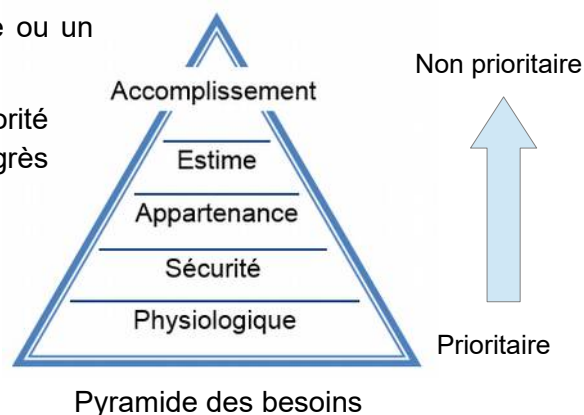
Le besoin peuvent être classé selon un ordre de priorité (voir pyramide ci-contre). Il peut aussi évoluer en fonction du progrès technique, des inventions et des innovations.

Si un objet technique ou un système ne répond pas à un besoin alors il n'est d'aucune utilité !

Parfois cependant, un système peut être conçu et faire naître de nouveaux besoins.



Exemple :



Définir le besoin auquel répond un système



Pour qu'un système réponde au besoin de l'utilisateur, le concepteur doit définir avec précision : La **mission** à remplir par le système, l'**environnement** de celui-ci et les **utilisations** qui en seront faites.



1 – On définit d'abord la mission du système c'est-à-dire son exigence ou sa fonction principale.

Système doit permettre de *Verbe à l'infinitif*

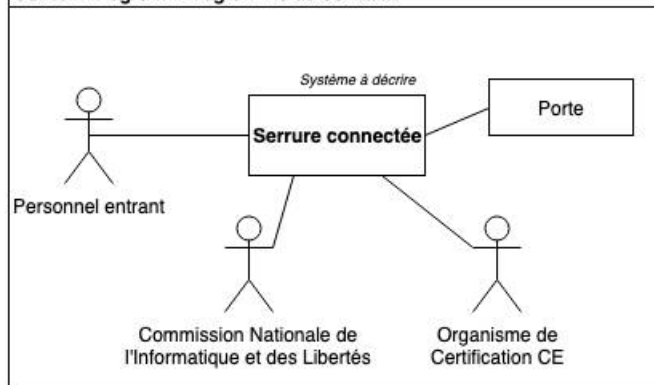
Mission de la serrure connectée

La serrure connectée doit permettre d'entrer et sortir d'un local ou d'une maison sans clé.

2 – L'analyse de l'environnement ou contexte identifient :

- les **acteurs** qui jouent un rôle ou interagissent avec le système
- les **éléments** et **contraintes de l'environnement** du système

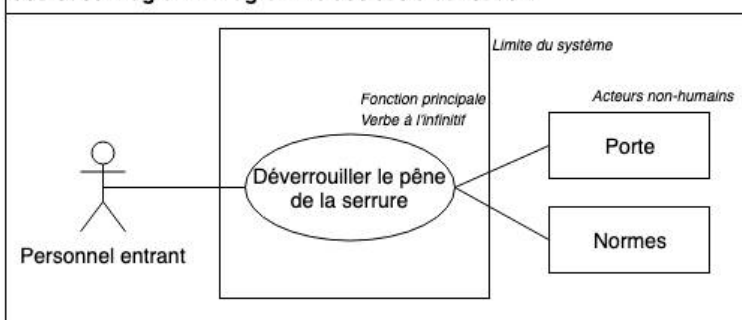
Context Diagram : Diagramme de contexte



3 – Les systèmes sont souvent multi-fonctions et multi-usages. L'étude des cas d'utilisation va permettre enfin de recenser :

- les **acteurs** humains à l'origine d'une interaction
- les **acteurs non-humains**
- la limite du système
- les relations entre les acteurs et les cas d'utilisation du système

Use Case Diagram : Diagramme des cas d'utilisation



Le SysML est un langage graphique qui permet de répondre à ces questions.

Exemple avec une serrure connectée (source : Ac-Dijon)



TECHNOLOGIE

Ce que je dois retenir

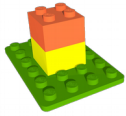
CONTRAINTES, PERFORMANCES D'UN OBJET TECHNIQUE

CYCLE
4

CT 2.3
DIC 1.2

Identifier les conditions, contraintes (normes et règlements) et ressources correspondantes, qualifier et quantifier simplement les performances d'un objet technique existant ou à créer.

La conception d'un objet



Pour répondre aux **besoins** de l'utilisateur, le **concepteur** doit lister les **exigences à satisfaire** : les **performances** à atteindre, les **normes** et **contraintes** à respecter pour ensuite choisir les **solutions** adaptées.

Exigences liées à l'usage

Je voudrais pouvoir loger ma famille dans un logement confortable...

Exigences liées à l'estime

J'aimerais une maison en bois, je trouve ça tellement joli!

Les utilisateurs

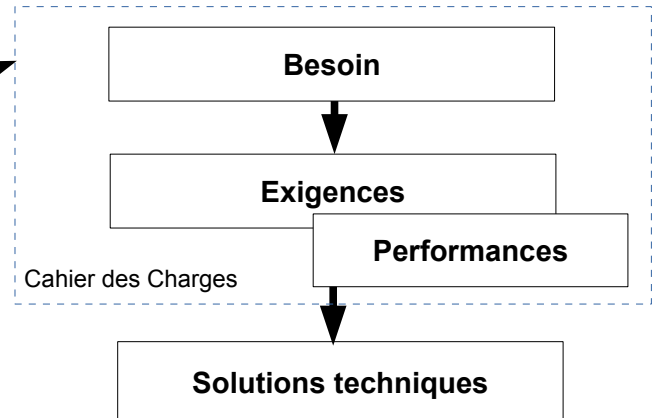
Besoin

Nous venons vous voir pour un projet de construction

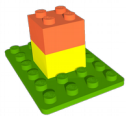


Le concepteur

(Ce mot vient du verbe concevoir)



Les exigences à satisfaire

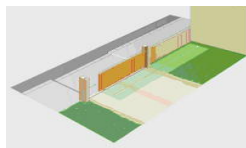
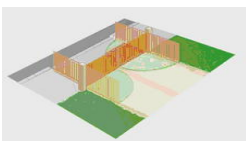


Une **exigence** est une fonction à remplir ou une contrainte à satisfaire par un système. Le concepteur devra donc en tenir compte lors de la recherche de solution. Les choix définitifs d'une solution seront donc des **compromis** qui dépendront du niveau de performance attendu.

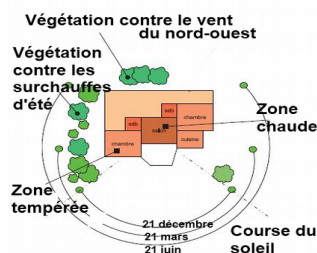
Les exigences peuvent être de « types » ...

fonctionnement : Liées à l'environnement d'utilisation

Ex : Espace pour la solution > ouverture du portail à double battant ou coulissant



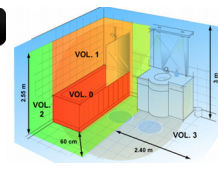
développement durable : Liées au respect de l'environnement



esthétique : Liées aux goûts de l'utilisateur

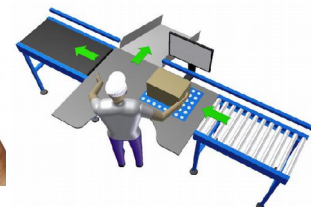


normes : Liées à la protection, à la simplification ou à la sécurisation de l'utilisation du système



Vol.	appareils électriques autorisés
0	aucun
1	Norme IP X 4 (très basse tension 12V)
2	Norme IP X 3 (protection contre la pluie)
3	Norme IP X 1 (protection contre les gouttes)

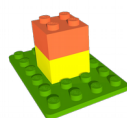
ergonomie : Liées à la relation avec l'utilisateur



budget : Liées au prix de revient et de vente de l'objet

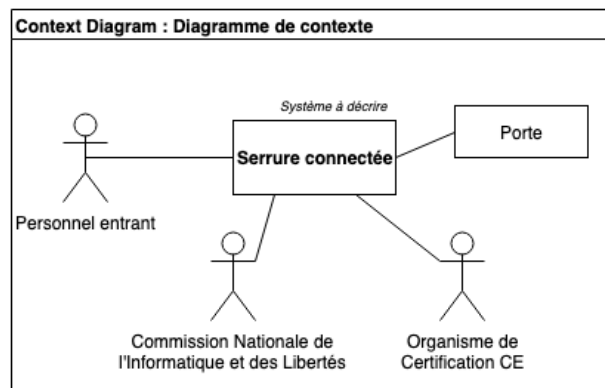
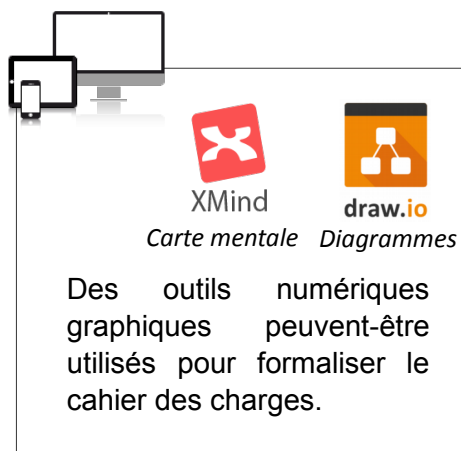


Identifier le contexte



Le **concepteur** rédige un document appelé **Cahier des Charges** qui identifie le besoin auquel le système doit répondre, les utilisations qui en seront faites. Dès la mission du système formulée, il est indispensable d'identifier le contexte d'utilisation du système en listant les éléments de l'environnement qui interagissent avec lui.

Langage de modélisation SysML – Contexte du système
Exemple ici avec une serrure connectée (source : Ac. Dijon)

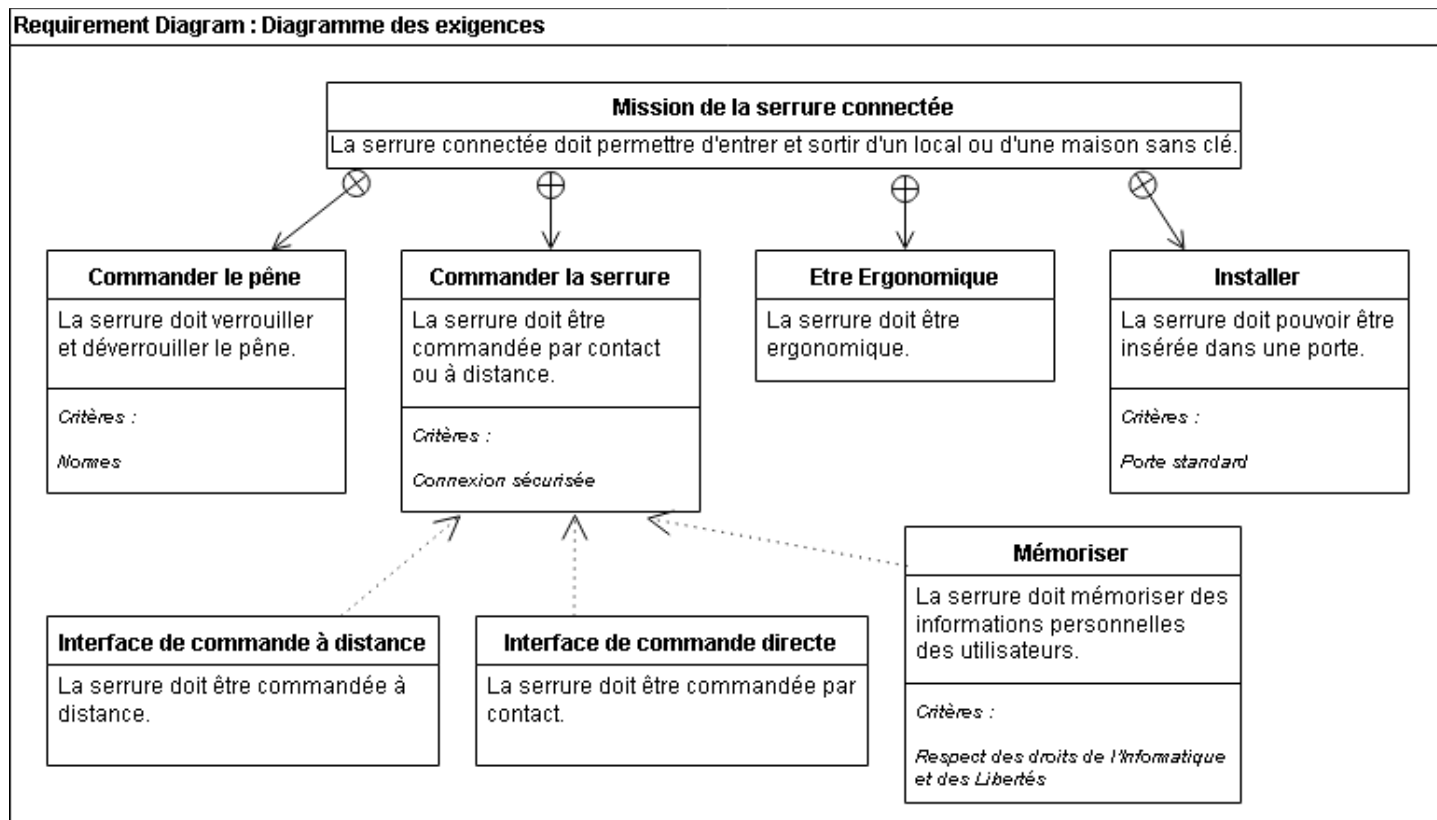


Qualifier et quantifier les performances du système



Pour chaque exigence, il est nécessaire de préciser les critères et niveaux de performances demandés. Le niveau de performance demandé a un impact direct sur le choix des solutions et sur le coût du système.

Langage de modélisation SysML – Exigences et performances
Exemple ici avec une serrure connectée (source : Ac. Dijon)



Le SysML, c'est quoi ?

SysML = Systems Modeling Language

Langage de Modélisation de Systèmes



C'est un outil qui permet de représenter des objets techniques, sous forme de schémas appelés diagrammes.



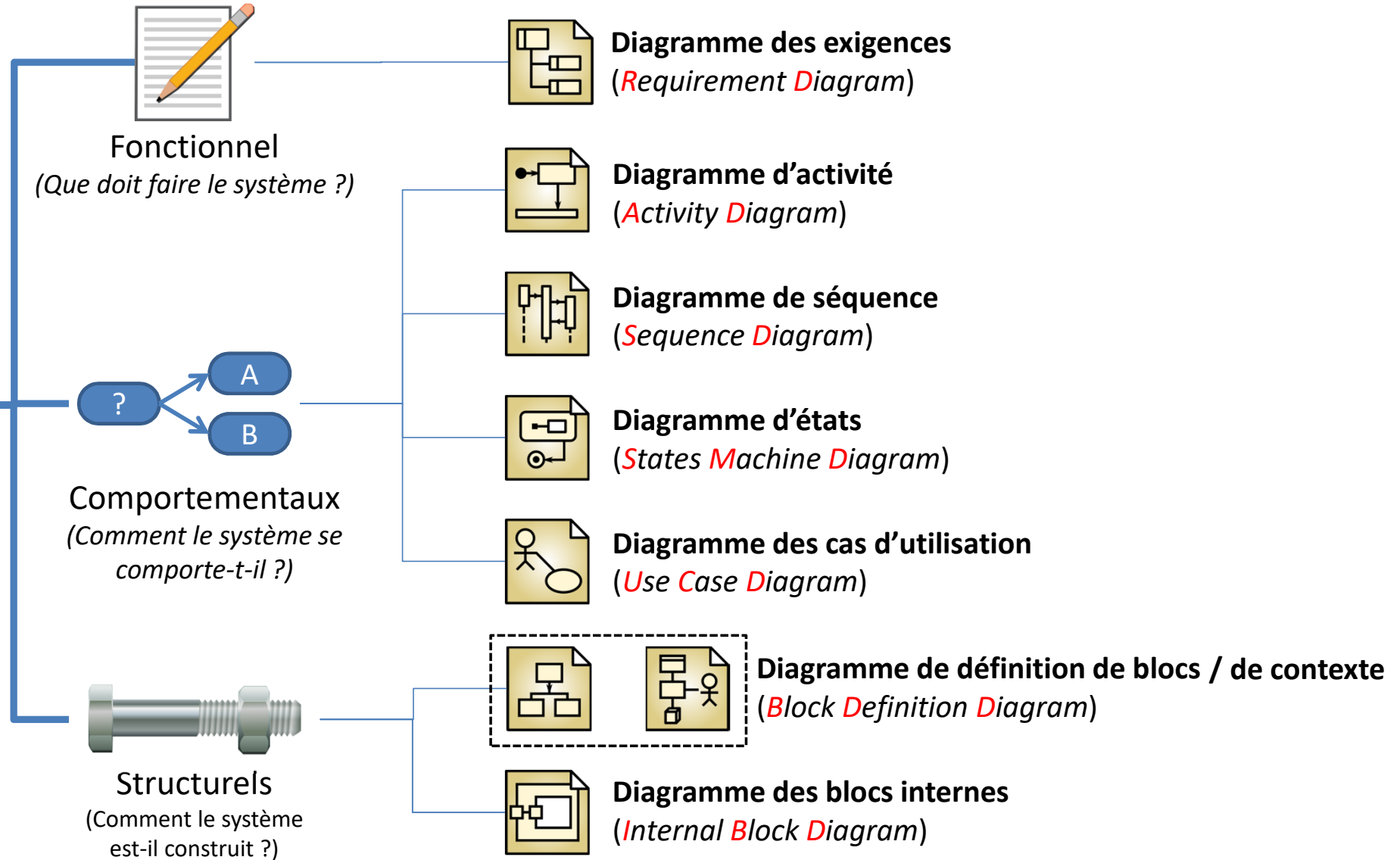
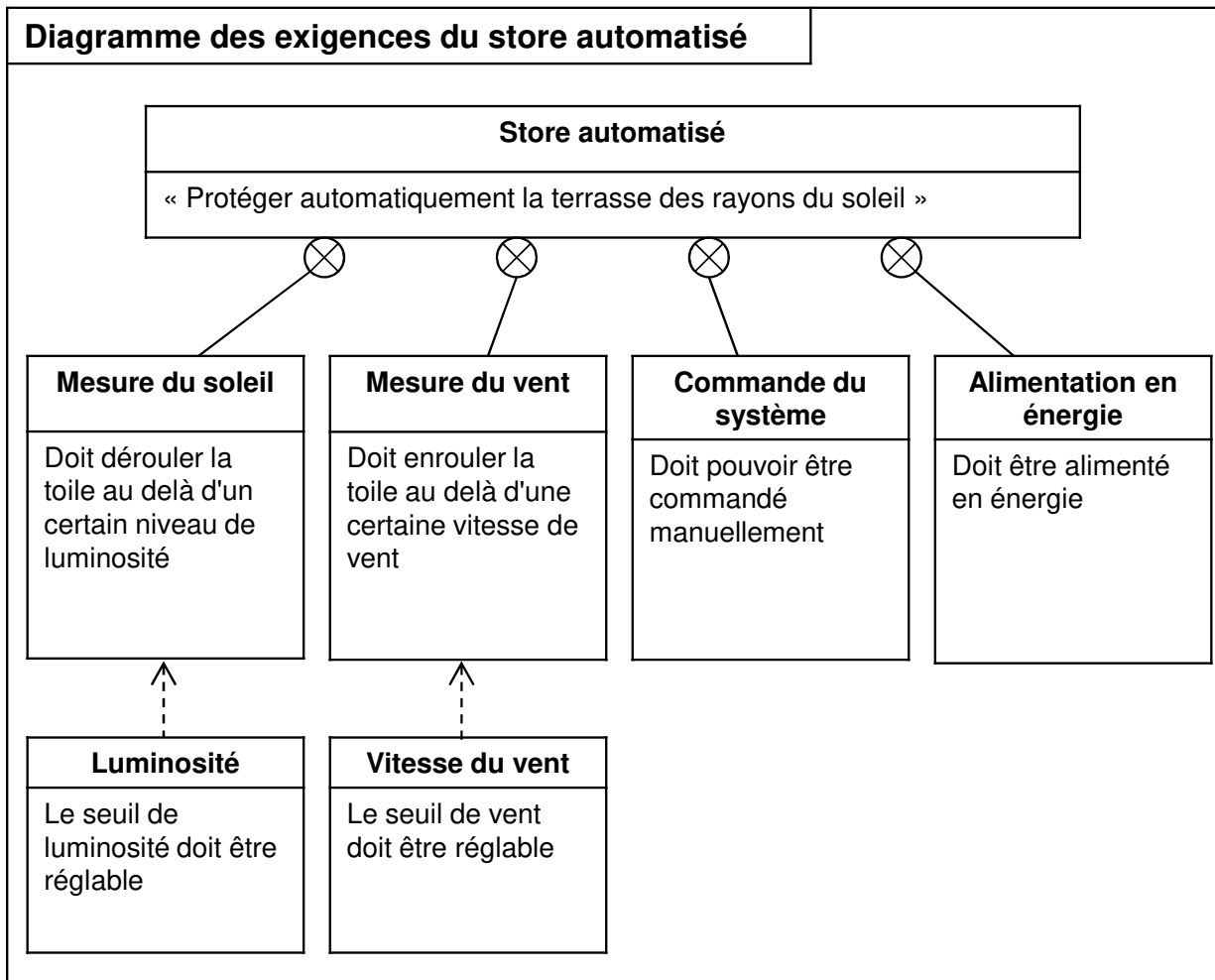




Diagramme des exigences (*Requirement Diagram*)



C'est le **cahier des charges fonctionnel** du système
(Le système doit...). Ce sont les **exigences** du CdCF.



1- Dans le premier rectangle, on indique le **nom** et l'exigence du système (**mission principale**).

2- Ensuite, on décompose la mission principale en **exigences unitaires**. On précise au dessus l'élément concerné.

3- Pour chaque exigences unitaires, on peut ajouter des précisions, des valeurs, des données...

→ Ce diagramme présente les fonctions ou les contraintes que l'objet doit satisfaire. **Nous allons réutiliser ces éléments pour le diagramme suivant, le diagramme des cas d'utilisation.**



Exigences



Cas d'utilisation



Séquence



Définition de blocs



Contexte



Blocs internes



États



Activité

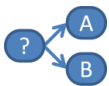
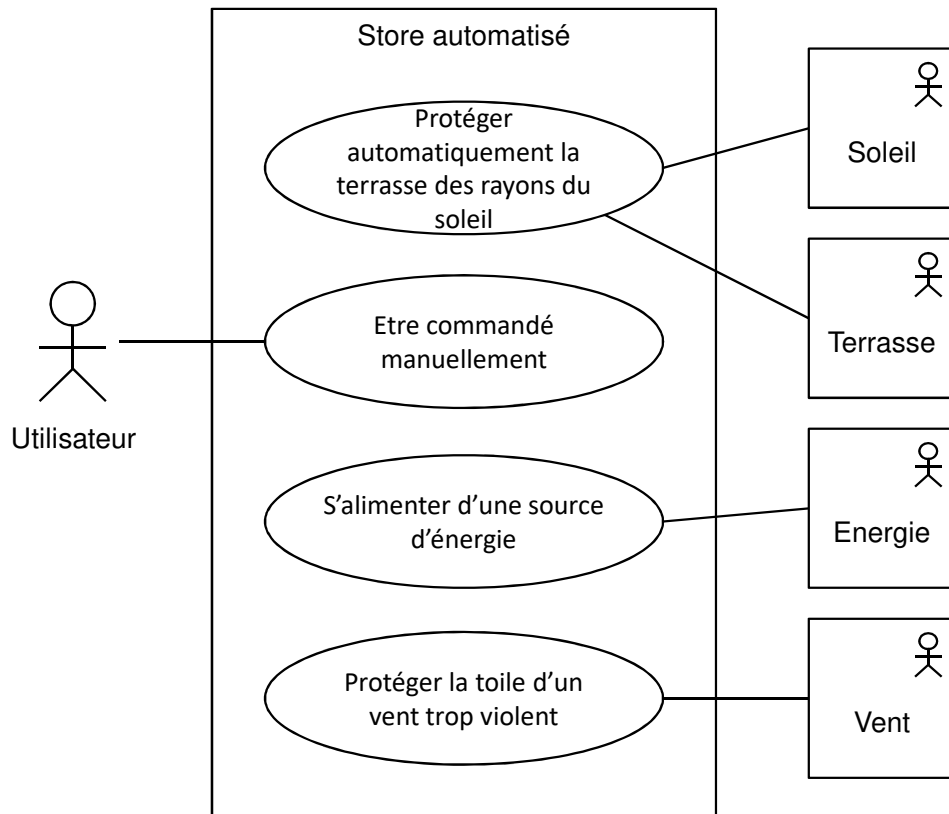


Diagramme des cas d'utilisation (*Use Case Diagram*)



Il exprime les **services offerts par l'objet aux acteurs**.
→ Décrit ce que fait l'objet (et non ce que fait l'utilisateur) mais sans dire comment il le fait.

Diagramme des cas d'utilisation du store automatisé



1- Un grand rectangle indique la **frontière de l'objet**

2- À gauche, on place des **acteurs humains**.

À droite, les **acteurs non humains**.

(→ Ces éléments sont les mêmes que ceux du diagramme de contexte).

3- Dans des ovals, on décrit les actions réalisables par le système (les services rendus par le système aux acteurs, sous forme de verbe à l'infinitif plus compléments)

Et on va les relier aux acteurs concernés.

i Il arrive que des actions (ovales) ou des acteurs soient reliés entre eux si il agissent l'un sur l'autre.

On y retrouve généralement la mission principale et les exigences déjà identifiées dans le diagramme de contexte (mais on précise comment sont assurées les missions)



Exigences



Cas d'utilisation



Séquence



Définition de blocs



Contexte



Blocs internes



États



Activité

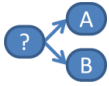
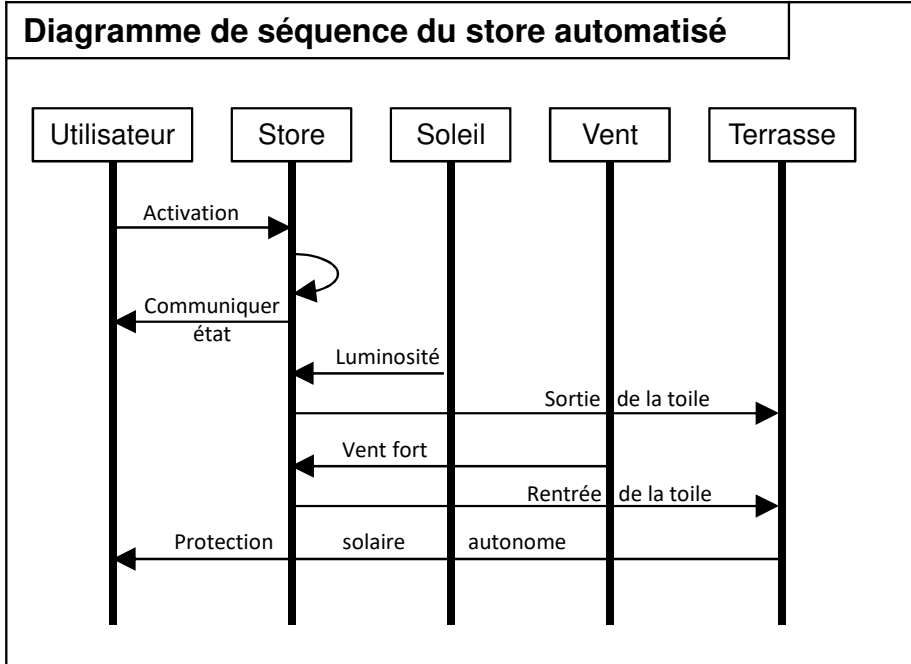


Diagramme de séquence (*Sequence Diagram*)



Il présente les **interactions entre les acteurs et l'objet** selon un **ordre chronologique**.



1- En haut, dans des cadres, on place **tous les acteurs d'un cas d'utilisation***. On commence par l'objet étudié. On replace à gauche de l'objet étudié (ici, le store) l'acteur principal et à droite les acteurs secondaires.

2- On trace des flèches qui vont retracer, dans l'ordre chronologique, chaque échange entre les acteurs et l'objet. On lit ce diagramme de haut en bas.

* : l'acteur énergie n'est pas représenté. En effet, il y a en principe, un diagramme de séquence par cas d'utilisation mais on essaye d'en faire un seul qui contient toutes les interactions possibles. Ici, l'auteur a choisi de ne pas représenter l'alimentation en énergie.

→ Ce diagramme permet de représenter, étape par étape, les échanges (information, énergie, matière) entre chaque acteurs et l'objet lui-même dans un ordre chronologique, représenté par les lignes de vie verticales.



Exigences



Cas d'utilisation



Séquence



Définition de blocs



Contexte



Blocs internes



États



Activité

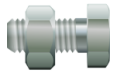


Diagramme de définition de blocs (*Block Definition Diagram*)



Il permet simplement de **lister tous les blocs** et les sous-blocs **qui constituent l'objet**.

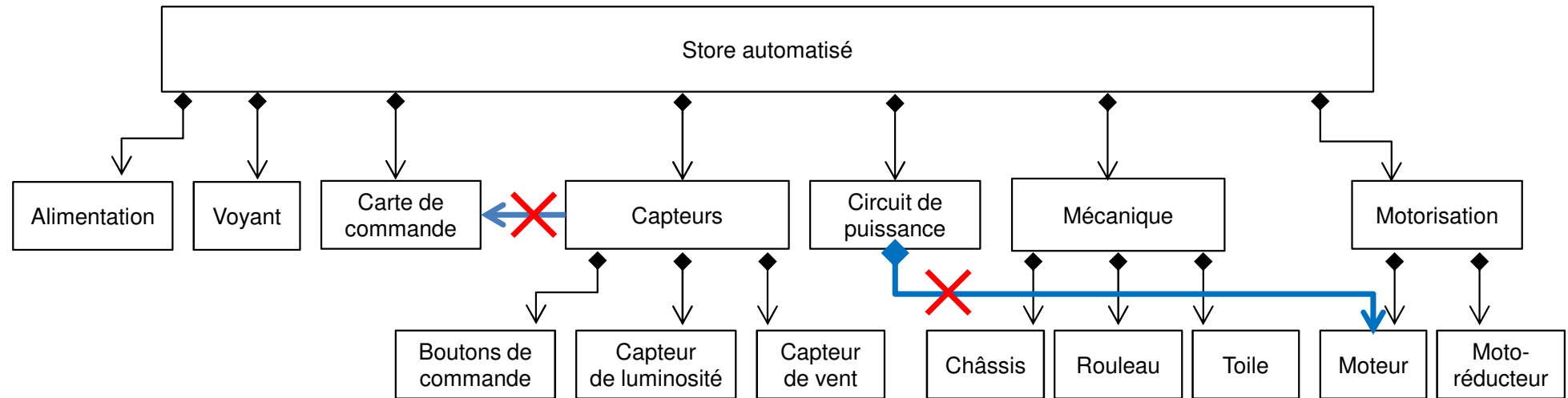
1- Au centre, en haut, on place l'objet

2- En dessous, on place chaque bloc et éventuellement des « sous-blocs » que l'on trouve dans l'objet

i Dans ce diagramme, il n'est pas nécessaire d'expliquer comment les blocs interagissent entre eux. Par exemple, pas besoin d'indiquer que les capteurs sont reliés à la carte de commande.

→ Ce diagramme permet de faire l'inventaire de tous les blocs qui composent l'objet.

Diagramme de définition de blocs du store automatisé





Exigences



Cas d'utilisation



Séquence



Définition de blocs



Contexte



Blocs internes



États



Activité

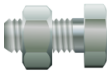
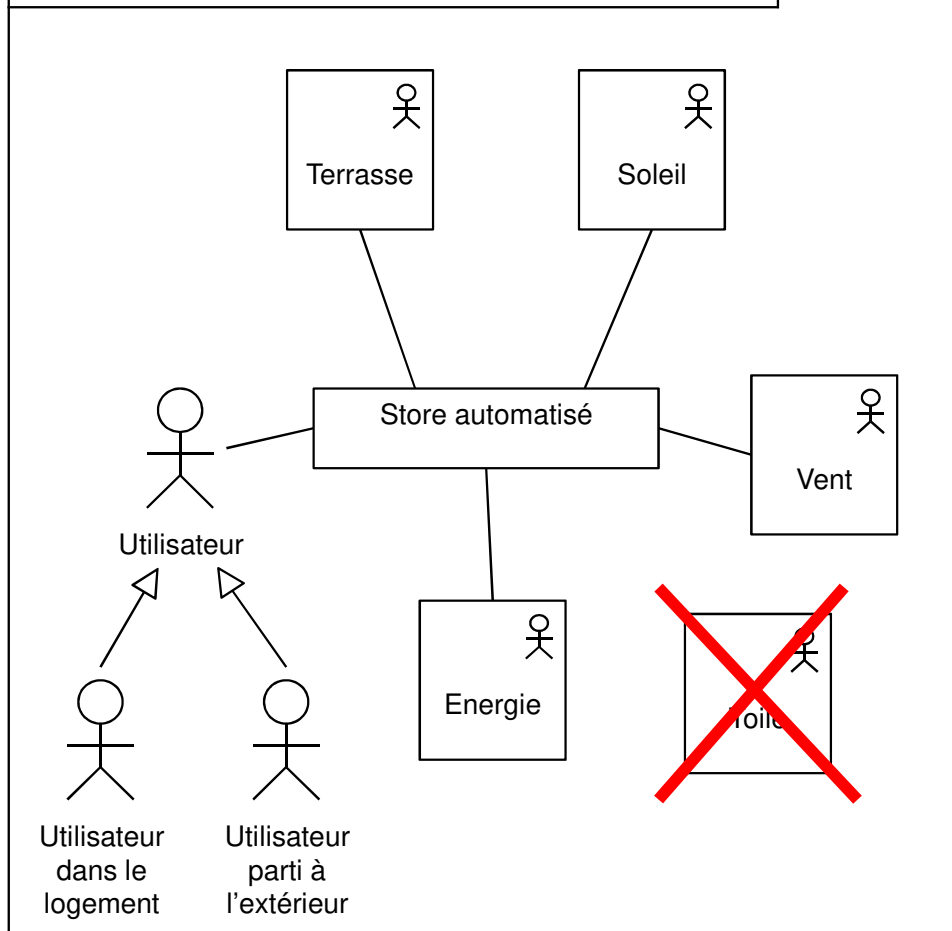


Diagramme de contexte (Context Diagram)



Il recense les éléments extérieurs (**acteurs**)
qui interagissent avec l'objet.

Diagramme de contexte du store automatisé



1- Au centre, on place l'objet

2- On place tout autour, les acteurs qui interagissent avec l'objet (**ce sont les mêmes que ceux du diagramme des cas d'utilisation**)

Si besoin, un acteur peut être décomposé en deux sous acteurs .

i Il ne faut pas placer comme acteur des éléments qui appartiennent au système.

Exemple ici : toile, capteurs... ne sont pas à indiquer.

→ Ce diagramme permet de lister tous les éléments extérieurs en relation avec l'objet.

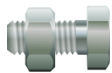
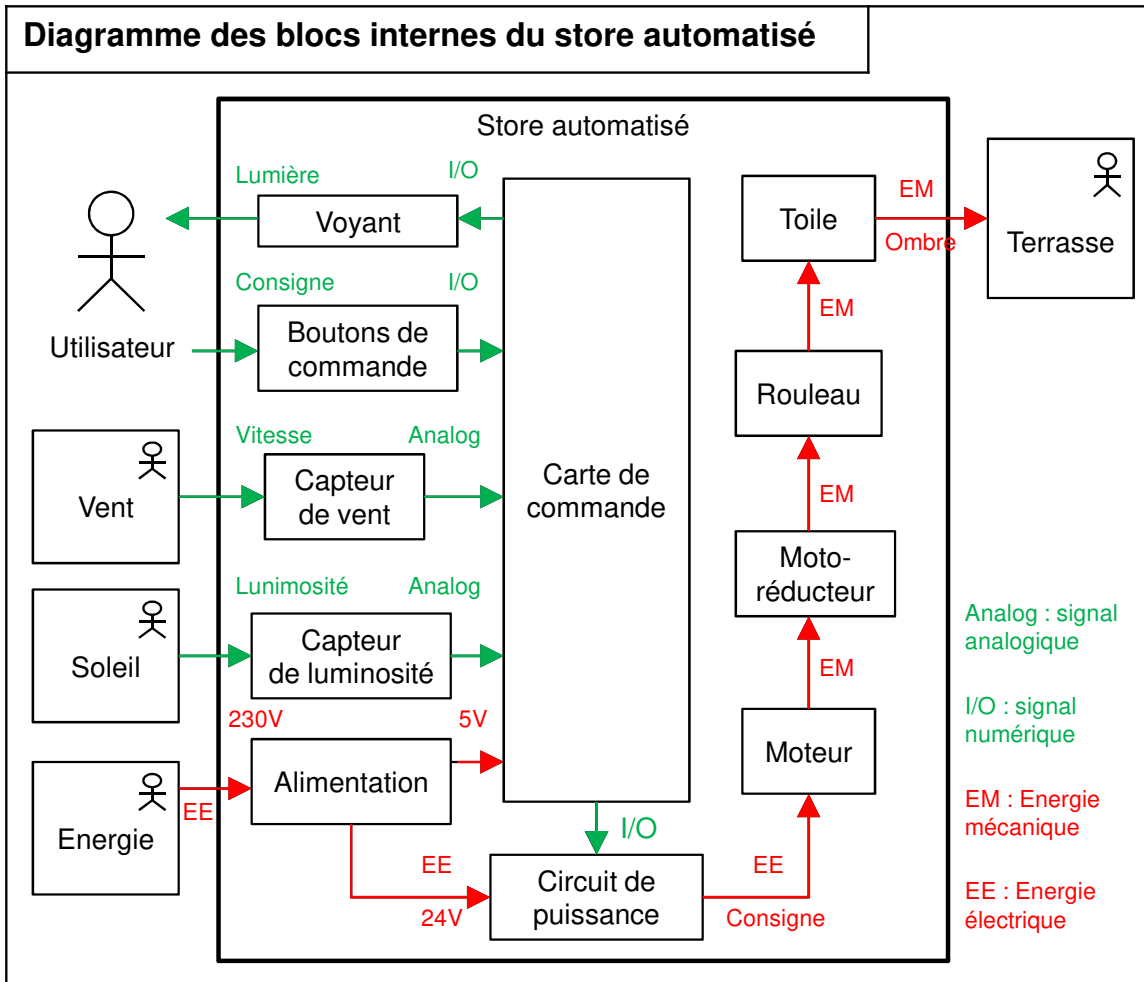


Diagramme des blocs internes (Internal Block Diagram)



Il permet de définir comment l'**information**,
l'**énergie** et la **matière** circulent à travers l'objet.



1- Dans un grand rectangle, on délimite l'objet.

2- A l'extérieur de ce rectangle, on place les acteurs (définis dans le diagramme de contexte)

3- A l'intérieur du rectangle, on place les blocs de l'objet (définis dans le diagramme de définition des blocs)

i Ici, le **Châassis** n'est pas représenté car il n'a pas d'échange de flux dans l'objet

4- On représente par des flèches les flux d'information, d'énergie et de matière.

i On peut utiliser des codes couleurs et ajouter des informations sur les flux

i Diag. séquence / ports SysML

→ Dans ce diagramme, on voit les échanges et interactions entre les différents blocs.

Il remplace la chaîne d'énergie et d'information.



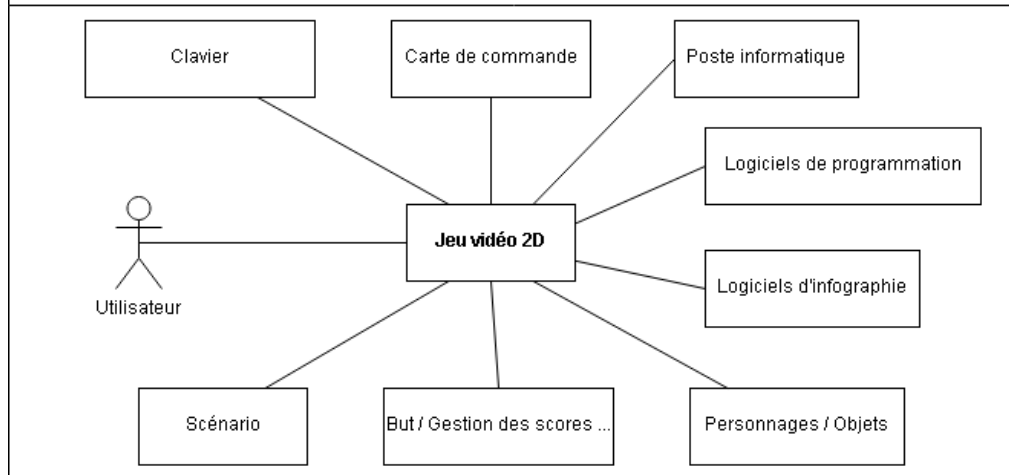
Analyse SysML Jeu vidéo 2D en 4eme



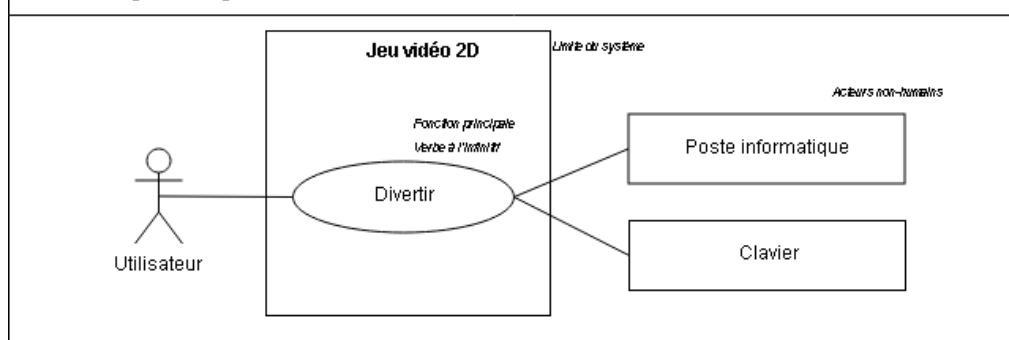
Mission du système

Le jeu vidéo 2D doit permettre de divertir son utilisateur

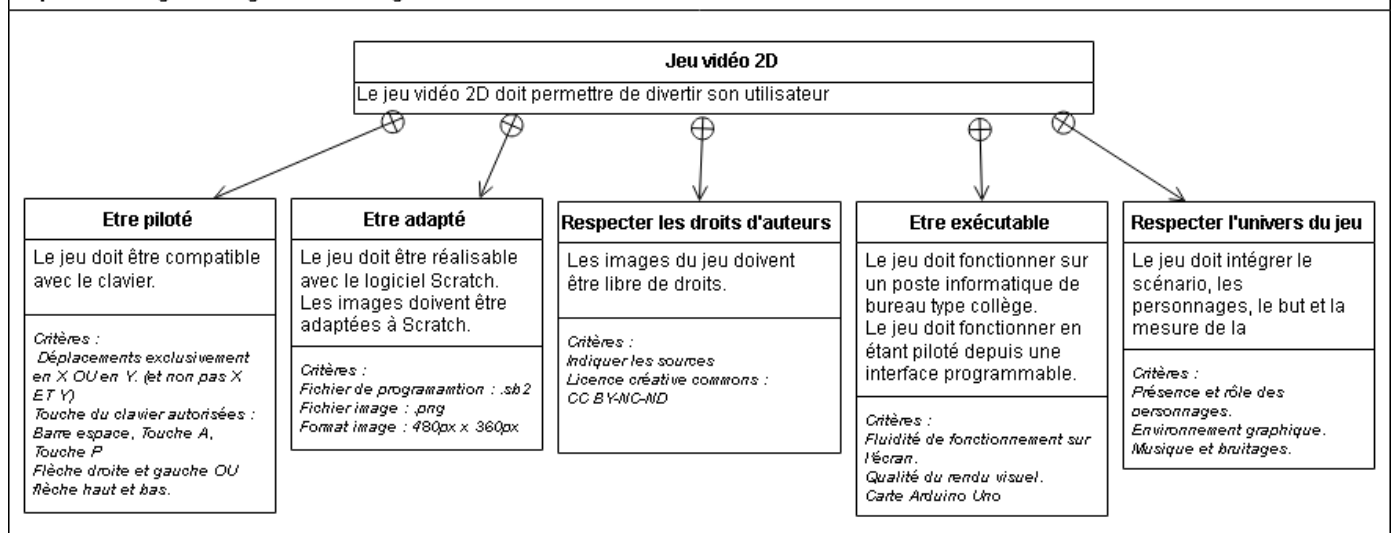
Context Diagram : Diagramme de contexte



Use Case Diagram : Diagramme des cas d'utilisation



Requirement Diagram : Diagramme des exigences



Un système embarqué est un système **électronique** et **informatique autonome**, qui est capable de **réagir** souvent **en temps réel** et de **réaliser des tâches précises**.

- Exemples de systèmes embarqués :

Aspirateur robot



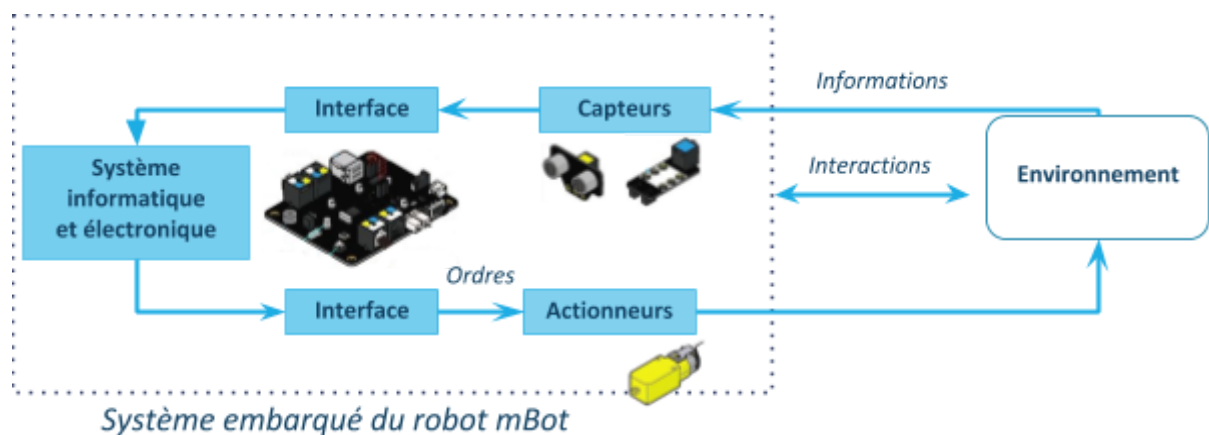
Robot mBot



Voiture autonome



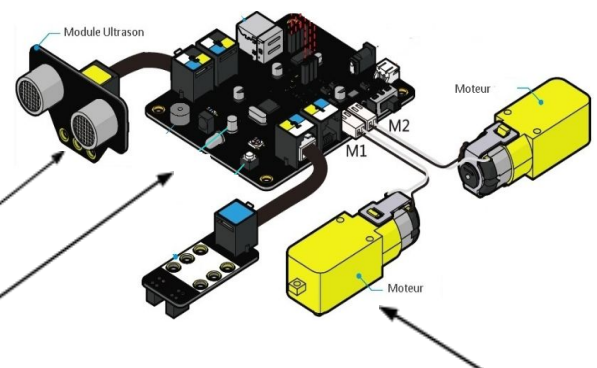
- Le **système embarqué** du **mBot** comprend des **capteurs**, des **actionneurs**, une **interface** et un **programme stocké** dans sa **mémoire**. Il assure un **fonctionnement autonome** qui **réagit avec son environnement**.



- Pour que le robot mBot soit **autonome** dans ses déplacements en **temps réel** et donc éviter les obstacles, on va utiliser ses **capteurs ultrasons** grâce à son **programme téléchargé**.

Exemple de fonctionnement du système embarqué du robot mBot :

- L'utilisateur **télécharge** un **programme** dans la mémoire de la carte électronique avec le câble USB.
- Pour éviter l'obstacle, le robot utilise les **informations** venant de son **capteur à ultrasons**.
- La **carte électronique** (avec son **programme**) **traite** ces **informations**.
- Le programme de la **carte électronique** **envoie** des **ordres** pour **commander** ses **moteurs** afin de tourner avant la collision.

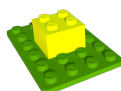


Un **système embarqué** est un système **électronique** et **informatique autonome** capable de **réagir en temps réel** et de **réaliser des tâches précises** (déplacements, préhension...).

Il est intégré dans un objet et permet, à partir de son **ordinateur** (microprocesseur, mémoires, carte mère, alimentation électrique autonome...), de ses **capteurs**, de ses **actionneurs** et d'un **programme stocké** dans sa **mémoire**, d'**assurer un fonctionnement autonome**.

	TECHNOLOGIE <i>Ce que je dois retenir</i>	L'ALGORITHME ALGORIGRAMME OU LOGIGRAMME	CYCLE 4
CT 1.3 – CT 2.5 – CT 2.7 DIC 1.5	Imaginer des solutions pour produire des objets et des éléments de programmes informatiques en réponse au besoin.		
CT 3.1 OTSCIS 2.1	Exprimer sa pensée à l'aide d'outils de description adaptés : croquis, schémas, graphes, diagrammes, tableaux.		
CT 4.2 – CT 5.5 IP 2.3	Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.		

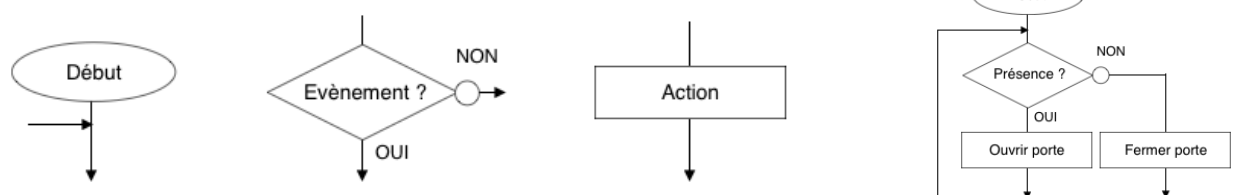
Symboles de base



Un algorithme est une suite d'instructions précises et structurées qui décrit la manière dont on résout un problème.

Cette description peut être textuelle (si, alors, sinon, tant que ...) ou graphique (appelé également organigramme ou logigramme).

Dans ce cas des normes d'écritures sont à respecter :

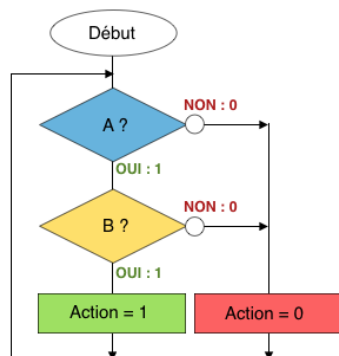
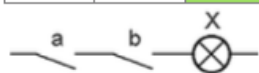


Fonctions ET et OU

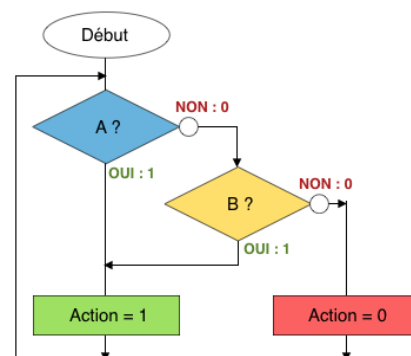
L'utilisation des fonctions ET et OU sont essentielles pour présenter correctement une solution.



Fonction ET		
A ?	B ?	Sortie
0	0	0
0	1	0
1	0	0
1	1	1



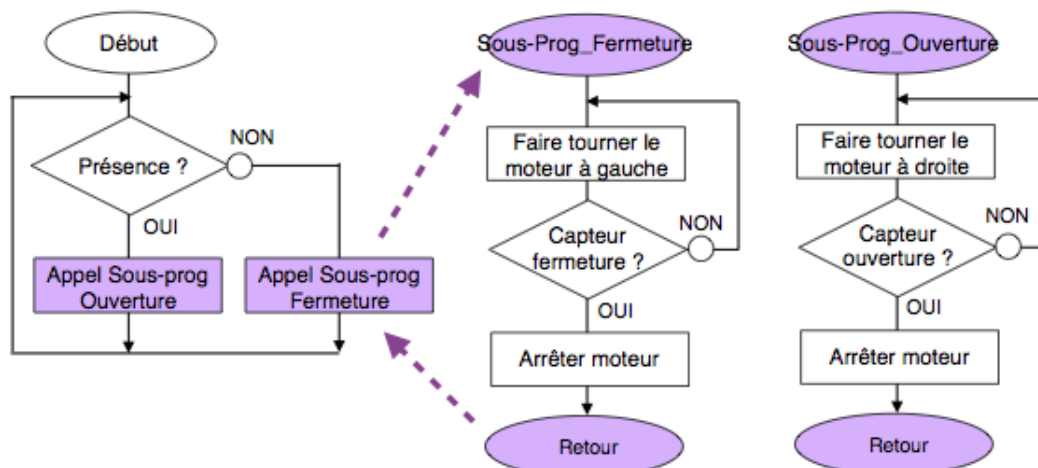
Fonction OU		
A ?	B ?	Sortie
0	0	0
0	1	1
1	0	1
1	1	1

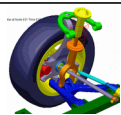


Algorithme et gestion des sous-problèmes



L'utilisation des sous-problèmes est idéale pour une meilleure lisibilité, pour alléger l'algorithme lors de succession d'actions identiques, pour faciliter le travail en collaboration, pour faciliter une recherche d'erreur (test individuel des sous-problèmes).



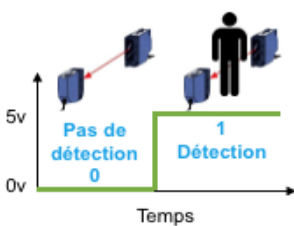
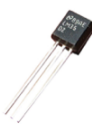
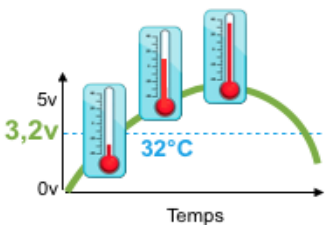
CT 1.2
MSOST 1.6

Mesurer des grandeurs de manière directe ou indirecte

Nature d'une information : logique ou analogique

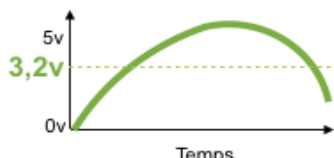
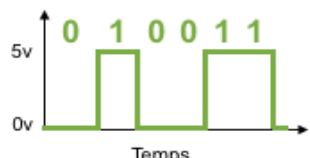
Une **information** peut être **logique** ou **analogique**.

Le choix du capteur sera déterminant pour interpréter l'information souhaitée.

Exemple de capteur	Signal fournie par le capteur	Information interprétée
 Barrière infrarouge		Détection ou pas de passage Information type LOGIQUE 2 valeurs possibles (tout ou rien)
 Capteur de température		Température en degrés Information type ANALOGIQUE Plusieurs valeurs possibles

Nature d'un signal : Analogique ou numérique

Un capteur fournit un **signal** de type **Analogique** ou **numérique**.

Signal Analogique	Signal numérique
	
Souvent un signal analogique évolue en tension (volt) <i>Exemple : 3,2 volts – Capteur de température</i>	Un signal numérique est une suite de 0 et de 1 <i>Exemple : 010011 - « Capteur » Ultrason</i>

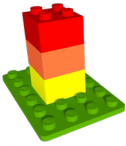
Un signal analogique doit être convertie en numérique pour pouvoir être traité par le microcontrôleur.
C'est la numérisation du signal.

Principe de fonctionnement d'un détecteur, capteur, codeur



Type de capteur	Exemple	Information	Exemple	Signal
Détecteur	1 ou 0	Logique	Détection ou pas (tout ou rien)	Numérique
Capteur	3,2 volts	Analogique	Degrés, Lux, ... : 32°C	Analogique
Codeur	010011	Analogique	Position, ... : 45°	Numérique

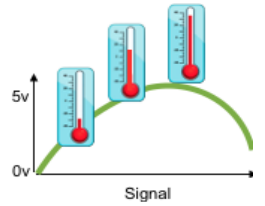
Principe de fonctionnement d'un capteur : numérisation



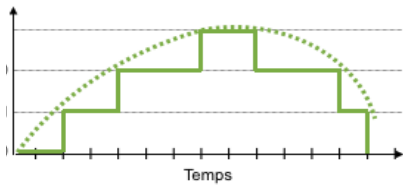
Un signal analogique doit être converti en numérique pour pouvoir être traité par le microcontrôleur (interface programmable) : C'est la numérisation du signal.

Plus la numérisation utilise de bits, meilleure est la précision.

Exemple avec un capteur de température :



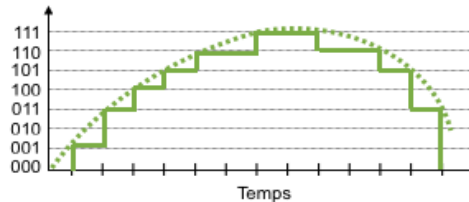
Numérisation sur 2 bits



Soit 4 valeurs possibles : de 0 à 3

Puissance de 2	2 ¹	2 ⁰
Décimal	2	1
0	0	0
1	0	1
2	1	0
3	1	1

Numérisation sur 3 bits

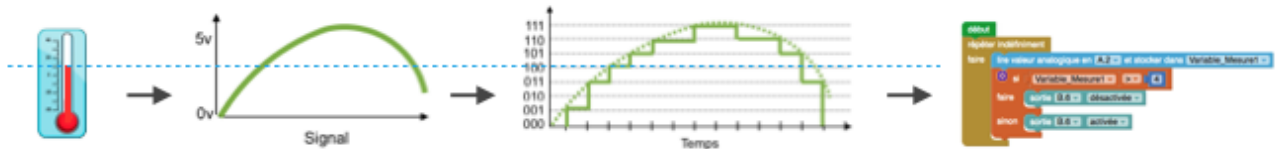


Soit 8 valeurs possibles : de 0 à 7

Exemple :

100 en binaire correspond à 4 en décimal.

Puissance de 2	2 ²	2 ¹	2 ⁰
Décimal	4	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



Température extérieure

32°C

Acquisition en analogique

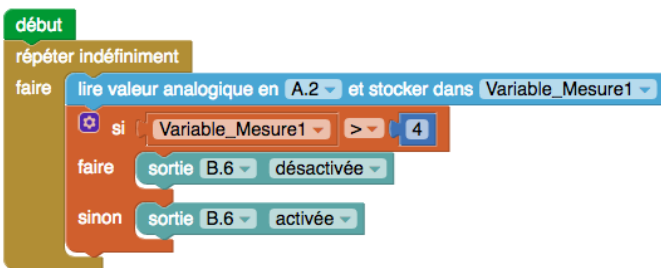
3,2 Volts

Numérisation

Binaire : 100

Traitement

Décimal : 4



Exemple avec le capteur de température qui communique sur l'entrée A2 du microcontrôleur.

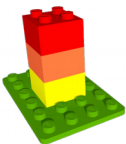
La valeur analogique est enregistrée dans la variable : Variable_Mesure1.

Si la variable > 4 (soit ici par ex 100 en binaire).

La sortie B6 se désactive (arrêt du chauffage)

Sinon la sortie B6 s'active (chauffage)

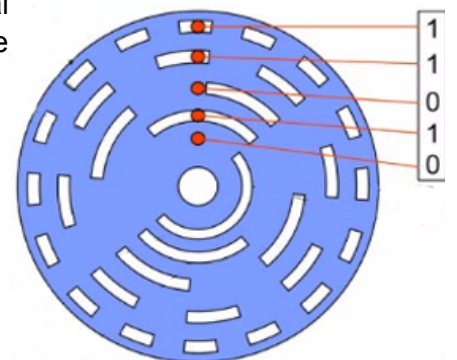
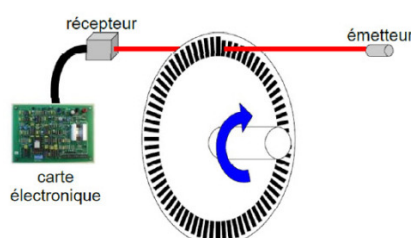
Principe de fonctionnement d'un codeur



L'avantage d'utiliser un codeur, est qu'il fournit un signal directement numérique, il peut donc être directement traité par le microcontrôleur.

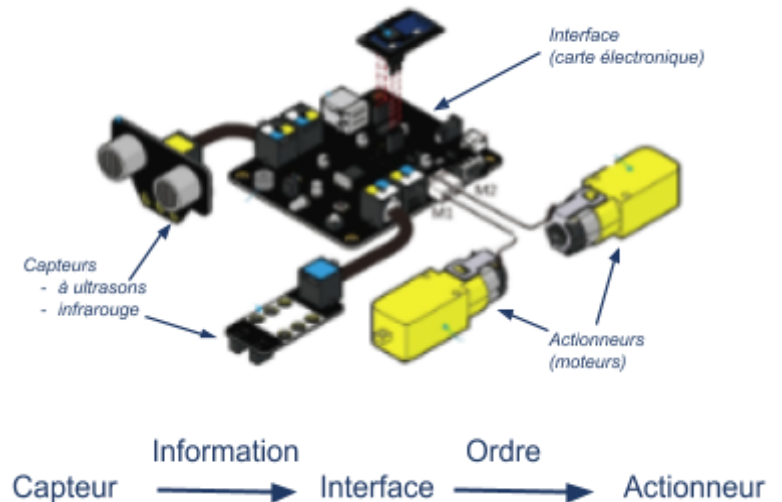
Exemple ici avec un codeur angulaire de position :

32 positions possibles soit une précision de $360^\circ / 32 = 11,25^\circ$ position codée sur 5 bits.



Pour qu'un **système automatisé** réalise une action, il faut une **interface** qui fait le lien entre les **capteurs** et les **actionneurs**.

- Un **capteur** réalise l'**acquisition** d'une **grandeur physique** (température, luminosité, présence, distance, ...) qu'il **transforme** en un **signal électrique**.
- L'**interface** **reçoit** les **informations** des capteurs, les **traite** et **envoie** des **ordres** aux actionneurs.
- L'**actionneur** **transforme** l'énergie d'entrée pour **réaliser** une **action**.



- Pour **capter** des **informations**, on peut utiliser différents **capteurs** :



Capteur à ultrasons

Détecte un obstacle à distance



Capteur infrarouge

Détecte un marquage noir sur le sol



Microrupteur

Détecte une ouverture de porte



Détecteur de mouvement

Détecte une présence

- Pour **réaliser** différentes **actions**, on peut utiliser différents **actionneurs** :



Moteur

Transforme l'énergie électrique en mouvement



Buzzer

Transforme l'énergie électrique en son



DEL

Transforme l'énergie électrique en lumière

Un ou des **capteurs** permettent d'**acquérir des informations** qui sont ensuite **traitées par une interface programmable** pour piloter un ou des **actionneurs** qui réalisent l'action (à partir de l'énergie qu'il reçoit).

Structurer les connaissances

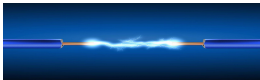
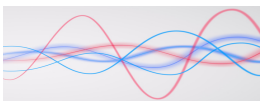
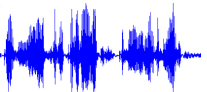
Forme et transmission du signal

CYCLE 4

>

» Début de cycle
 » Milieu de cycle
 » Fin de cycle

Pour assurer la **communication** et le **fonctionnement** des **objets connectés**, on utilise **différentes formes de signaux** transmis par **différents moyens de transmission**.

Un signal peut prendre différentes formes :		Un signal est transmis par différents moyens :	
Signal électrique		Transmission par fil	Transmission sans fil
Signal lumineux		 <i>Fil de cuivre</i>	 <i>Ondes infrarouges</i>
Signal sonore		 <i>Câble réseau</i>	<i>Ondes électromagnétiques</i>     <i>Bluetooth Wifi 3G/4G/5G</i> 
Signal radio		 <i>Fibre optique</i>  <i>Câble USB</i>	<i>Ondes sonores</i>   <i>Vibrations mécaniques</i>

- Pour **reconnaître la forme et la transmission du signal** :

1 - On repère l'**émetteur**, le **récepteur** et les **composants** utilisés.



2 - Entre la clé **WIFI** et la carte **WIFI** :

Forme du signal : **signal radio**

Transmission du signal : par **ondes électromagnétiques**

3 - Dans le **câble USB** :

Forme du signal : **signal électrique** (impulsions électriques)

Transmission du signal : par **câble**



Un **signal** peut prendre **différentes formes** : signal **électrique**, signal **lumineux**, signal **sonore** ou signal **radio**.

Pour transmettre un signal, nous avons deux possibilités :

- soit par fil** : fils de cuivre ou fibre optique,
- soit sans fil** : ondes infrarouges, ondes électromagnétiques ou vibrations mécaniques.

Structurer les connaissances

Ecrire, mettre au point (tester, corriger) et exécuter un programme

CYCLE 4

>

» Début de cycle
 » Milieu de cycle
 » Fin de cycle

Lors de la **conception** d'un **système programmable**, le concepteur va devoir **écrire un programme** pour commander le système **en fonction du comportement attendu**.

- Le concepteur écrit un **algorithme à partir du cahier des charges**...
Puis, il crée **des programmes** à l'aide d'un **logiciel de programmation par blocs** pour **commander le système réel**.



Exemple : un robot évitant un obstacle

Si la distance détectée par le robot est inférieure à 10 cm, alors le robot devra s'arrêter, sinon il continue d'avancer.



Logiciel : Scratch



Logiciel : mBlock

Ces **logiciels** possèdent des **menus**, où sont stockés des **instructions**, qui permettront de **concevoir un programme** et enfin **vérifier le comportement attendu** en l'exécutant.



- Comment écrire, mettre au point un programme pour vérifier le comportement attendu ?

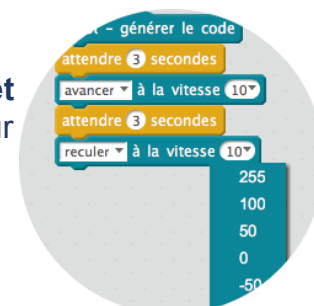
Pour écrire un programme, plusieurs étapes sont nécessaires :

- vérifier les attentes du cahier des charges** : tenir compte des **fonctions de service**.
- rédiger un algorithme** : des **phrases** qui permettent de respecter les **fonctions de service demandées**.
- repérer les capteurs** et les **actionneurs** à utiliser
- repérer les instructions** sur le **logiciel** de programmation
- mettre au point le programme** : **assembler les instructions**.
- exécuter le programme**
- vérifier le comportement attendu** : Si cela ne correspond pas au comportement attendu. Il faut corriger le programme.
- adapter le programme** pour obtenir le **comportement attendu**

Pour **écrire, mettre au point, exécuter un programme** commandant un **système réel** programmable et **vérifier le comportement attendu**, il faut donner des **instructions** à ce système, en utilisant des **langages** compréhensibles à la fois par le **concepteur** et par le **système programmé**. Plusieurs étapes sont nécessaires : on **rédige un algorithme**, que l'on met au point avec un **logiciel de programmation par blocs** et on vérifie le **comportement attendu** du système réel.

La programmation des **objets connectés** nécessite la gestion de **situations complexes** (déplacements, trajectoires, mesures des capteurs...). Pour résoudre ces problèmes, les programmeurs introduisent des **variables informatiques** dans leurs **algorithmes**.

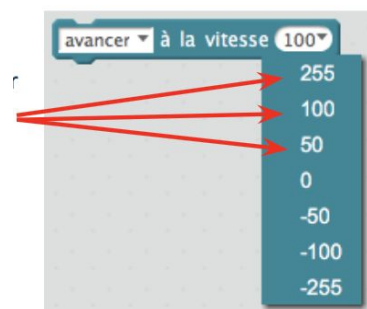
- Les **variables statiques** correspondent à des **valeurs constantes et fixes** que l'on utilise pour **fixer** des **valeurs numériques** comme pour définir des **vitesse**s, des **couleurs**, des **distances** ...



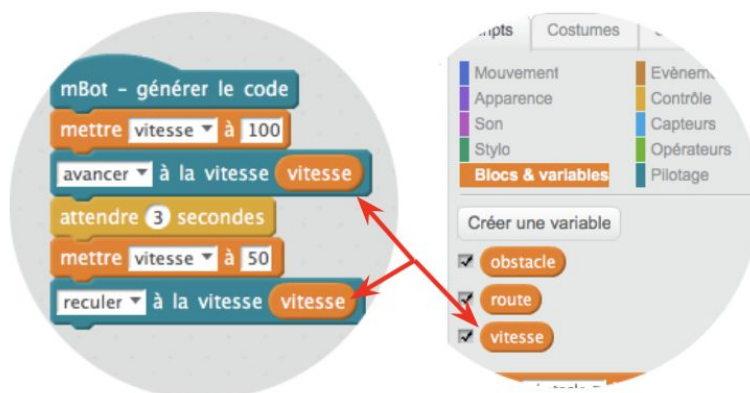
- Comment utiliser des **variables statiques** pour **régler** la vitesse d'un **robot** ?

Cas 1 - La plupart du temps, les **variables** sont **prédéfinies** dans le logiciel de programmation.

Le programmeur a le **choix entre plusieurs valeurs** pour **régler la vitesse** de son robot.



Cas 2 - Les **variables** peuvent aussi **être créées** par le programmeur. Elles **porteront un nom précis** en fonction du langage de programmation (exemple : var, B0, B1, vitesse...)



Les **variables** sont des éléments qui associent un **nom** (l'identifiant) à une **valeur** qui **sera implantée** dans la **mémoire** du système programmé. **Cette valeur** peut être une **vitesse**, une **distance**, un **temps** ... Une **variable** contient une **valeur** qui **peut varier au cours de l'exécution du programme**, comme la couleur des habits d'un personnage, le nombre d'activation d'un capteur, etc.

Structurer les
connaissances

Analyser le comportement d'un système réel et décomposer le problème posé en sous-problèmes

CYCLE 4

>

» Début de cycle

Pour analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande, il faut :

- Observer le système automatisé en fonctionnement.



Comment ouvrir et refermer automatiquement la porte du magasin ?

- Définir ses actions et rédiger l'algorithme correspondant à la résolution de son problème initial :

Exemple d'un algorithme pour le fonctionnement de la porte de supermarché avec 3 sous-problèmes :

- Si quelqu'un est détecté devant la porte automatisée, la porte doit d'ouvrir.
- S'il n'y a personne de détecté devant la porte automatisée, la porte doit se fermer.
- Ces 2 actions doivent être vérifiées en permanence indéfiniment.

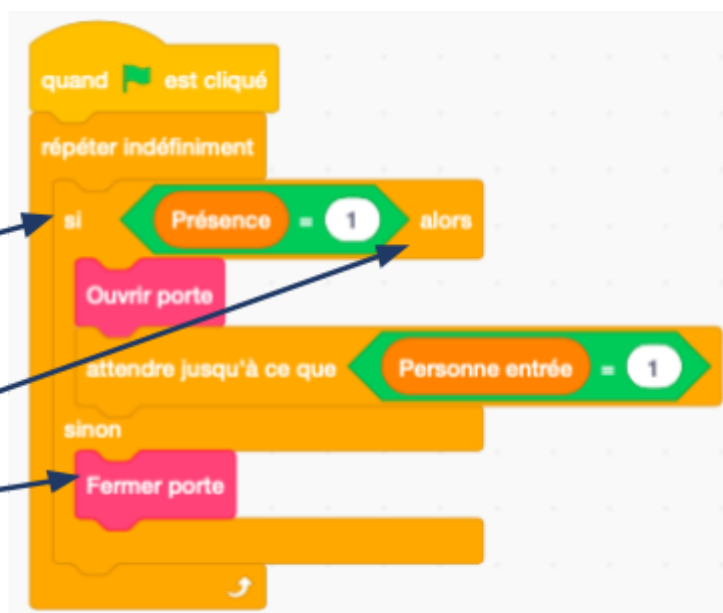
- Décomposer le problème en sous problèmes et traduire en blocs de programmation :

Algorithme :

Si une présence est détectée,

Alors la porte automatique s'ouvre le temps que la personne entre,

Sinon elle reste fermée



Le fonctionnement d'un système automatisé répond à un ou des problèmes posés.

L'observation de ce fonctionnement permet de définir et de décomposer le problème en sous-problèmes qui correspondent aux étapes de l'algorithme. Ces étapes sont ensuite traduites en blocs dans le logiciel de programmation.

Les **objets automatisés** qui nous entourent ont **besoin d'acquérir des informations sur leur environnement**. Cela implique de **mesurer** ou de **contrôler** des **grandeurs physiques** pour assurer leur fonction d'usage.

- Les **capteurs** sont donc capables de mesurer des **grandeurs physiques** et de les transmettre sous forme d'un **signal électrique analogique** ou **numérique** à un dispositif de commande.

Exemple avec le **capteur à ultrasons** du robot mBot :

Le capteur à ultrasons mesure des **distances** par rapport à l'obstacle : ce sont des **grandeurs physiques**.

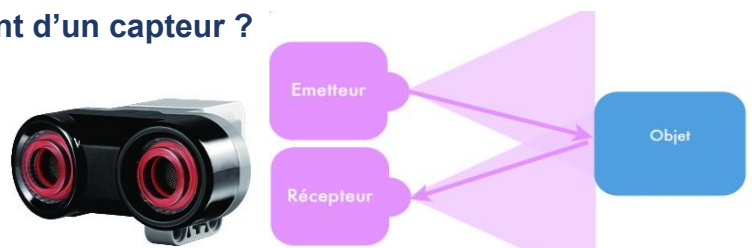
La **grandeur physique** est transmise à la **carte électronique** sous la forme d'un **signal électrique**.

Ce signal électrique est **analogique** puisqu'il **varie tout le temps** pendant le déplacement du robot.



- Le **capteur** est un **instrument de mesure** qui mesure une **grandeur physique**. Puis il émet un **signal électrique** proportionnel à la grandeur physique mesurée. Ce signal électrique peut prendre **différentes valeurs** qui sont **analogiques** ou **numériques**.
- Comment expliquer le fonctionnement d'un capteur ?**

Par exemple le **capteur à ultrasons** :



Le **capteur à ultrasons** utilise un **émetteur ultrason** qui envoie un son inaudible à l'humain. Lorsqu'un **objet est détecté**, le son « **rebondit** » sur l'objet, le **récepteur** reçoit le signal sonore dans un temps donné et le capteur détermine la **distance**.

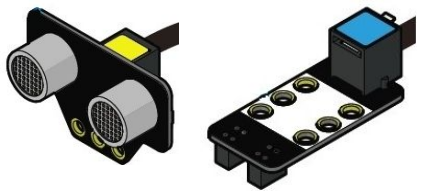
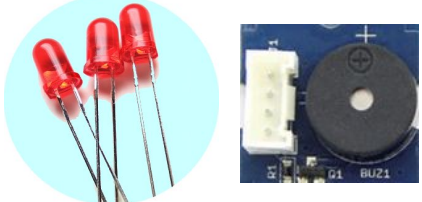
Les **capteurs** sont capables d'**acquérir et mesurer** des **grandeurs physiques** et de les **transmettre** sous forme d'un **signal électrique analogique ou numérique** à un dispositif de contrôle de commande.

Un objet technique a besoin de **capter des données**, les **traiter** et **communiquer** avec le reste du système **pour** effectuer une **action**.

- Toutes les fonctions qui participent à la détection des informations à leur traitement et leur communication constituent la **chaîne d'information** du système technique.

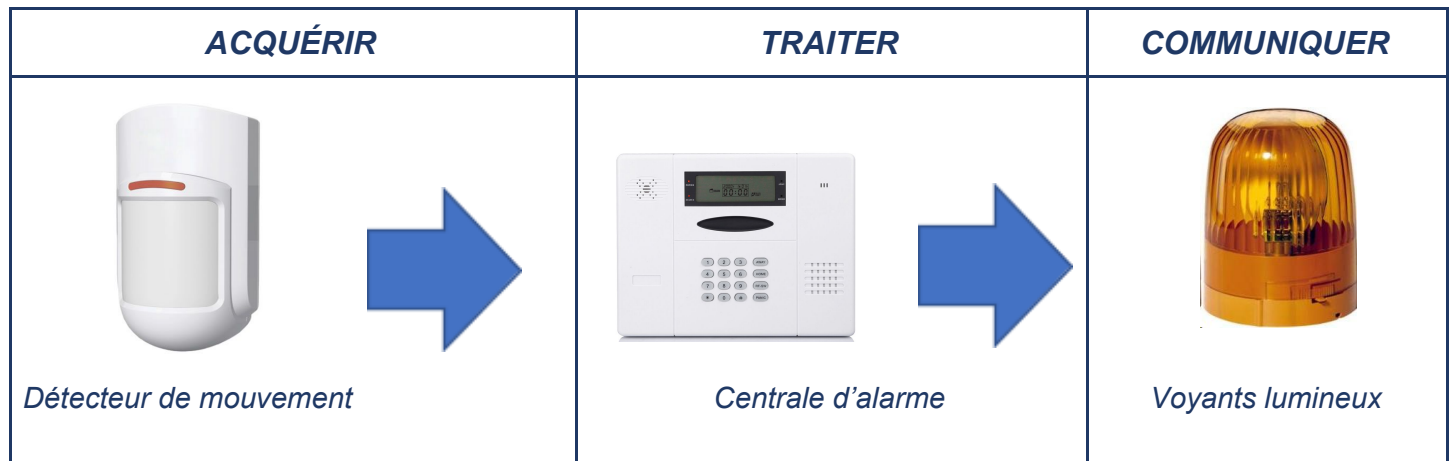


Exemple le robot Mbot

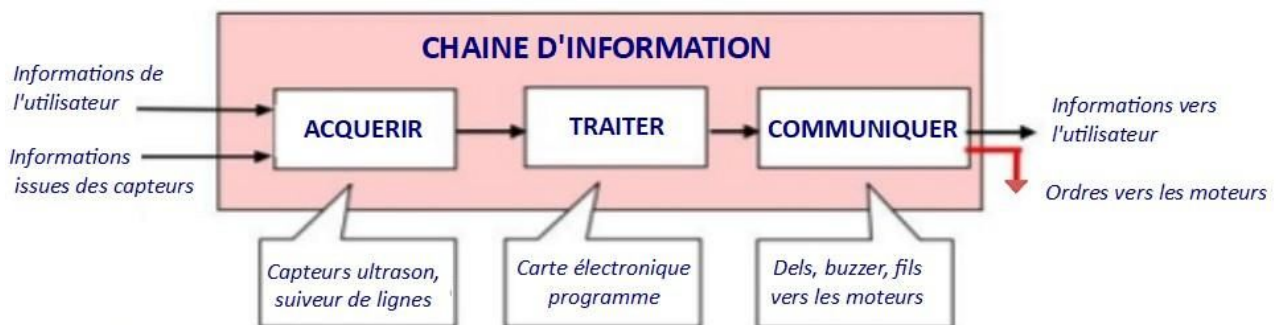
	Acquérir : cette fonction transforme une grandeur physique en un signal électrique support d'une information.	 Capteur (ultrasons) Capteur (suiveur de ligne)
	Traiter : cette fonction exécute les algorithmes (programmes) permettant de résoudre un problème et d'obtenir un résultat conforme à la fonction d'usage.	 Carte électronique (Microcontrôleur) Programme informatique (Scratch, Mblock)
	Communiquer : cette fonction assure le transfert des informations vers un utilisateur humain ou vers la chaîne d'énergie du système.	 Lampes de couleur (Dels) Signal sonore (Buzzer)

- Pour analyser le fonctionnement et la structure d'un système :
 - 1 - On repère les composants du système.
 - 2 - On associe les composants aux fonctions de la chaîne d'information.

Exemple d'une alarme anti-intrusion :



Voici des exemples de composants associés aux fonctions de la chaîne d'information :



La chaîne d'information est la partie du **système** qui **capte l'information** et la **traite** avant de la **communiquer**.

Elle est **composée** des trois fonctions : **Acquérir, Traiter et Communiquer**.

Pour fonctionner, les **objets techniques** ont besoin d'être **alimentés en énergie** et de **la transformer pour réaliser des actions**. Ceci est représenté par une **chaîne de plusieurs blocs fonctionnels** qui montrent le **parcours de l'énergie**.

- Dans la **chaîne d'énergie** on trouve **plusieurs blocs** qui **associent des fonctions** avec les **éléments de l'objet technique** :

Alimenter

Fournir l'énergie
nécessaire au système
pour réaliser l'action
désirée

Distribuer

Mettre à disposition
l'énergie

Convertir

Transformer l'énergie de départ
en une autre forme d'énergie
utilisable par le système

Transmettre

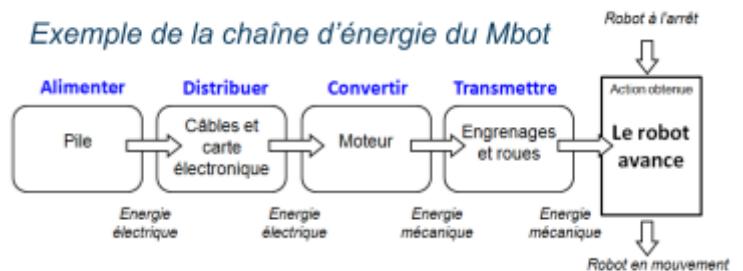
Transmettre l'énergie utile
jusqu'à l'endroit où elle est
utilisée

- Comment **comprendre** la **chaîne d'énergie** ?

1 - On **observe l'ordre** des
composants utilisés entre l'entrée de
l'énergie et l'action réalisée à la fin.

2 - On **associe** à chaque composant
une fonction.

Exemple de la chaîne d'énergie du Mbot



Le **fonctionnement d'un objet technique** peut être expliqué en **comprenant comment circule et se transforme l'énergie** dans cet objet. Pour représenter cela on utilise un **schéma appelé chaîne d'énergie**.

La **chaîne d'énergie** est généralement **composée** des fonctions techniques : **alimenter** et/ou **stocker**, **distribuer**, **convertir**, **transmettre**.

Pour pouvoir remplir leurs fonctions, **les objets techniques** sont construits à partir d'un ou **plusieurs matériaux**. Un matériau est fabriqué par **l'Homme** à partir d'une ou de plusieurs **matières premières**.

- Pour **identifier ou choisir des matériaux**, il faut :

1. **Connaître leur famille** : organiques, métaux, composites et céramiques



Organiques naturels

Les **matériaux organiques naturels** sont d'origine **animale** (laine, soie, cuir) ou **végétale** (bois, paille, coton).



Organiques artificiels

Les **matières plastiques** sont obtenues à partir du pétrole.



Métaux

Les **métaux** sont obtenus à partir de **minerai** extrait du sol. Quand on mélange **plusieurs métaux**, on obtient un **alliage**.



Composites

Les **matériaux composites** sont obtenus en assemblant des **matériaux différents** (qui ne se mélangent pas) afin d'obtenir un **nouveau matériau** avec des **performances** plus intéressantes.



*Céramiques
et minéraux*

Les matériaux **céramiques** sont obtenus à partir de la **terre** et du **sable cuit**.

Les matériaux **minéraux** sont issus de la roche.

Structurer les
connaissances

**Familles de matériaux avec leurs
principales caractéristiques**

CYCLE 4

>

» Début de cycle
» Milieu de cycle
» Fin de cycle

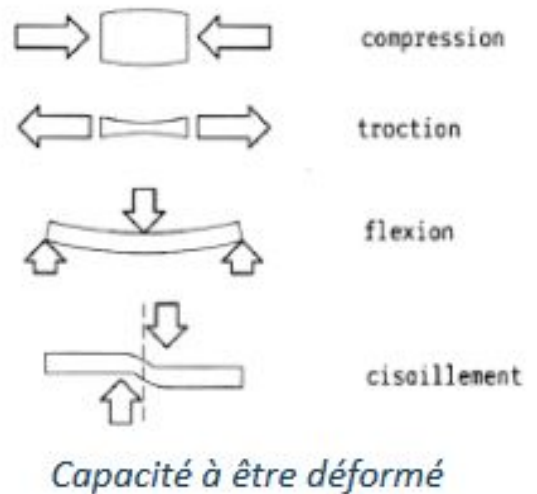
2. **Connaître quelques caractéristiques** : conductibilité thermique, conductibilité électrique, capacité à être déformé à chaud ou à froid, capacité à être recyclé.



Conductibilité
thermique



Conductibilité
électrique



3. **Effectuer des tests** et répondre aux questions : est-il composé de plusieurs matériaux ? est-il conducteur ? est-il attiré par un aimant ?



Multimètre



Aimant

On regroupe les matériaux par **familles** : **organiques**, **céramiques**, **métaux**, **composites**, en fonction de leurs **caractéristiques**.

On identifie les matériaux par l'**observation** et des **tests** : conductibilité électrique ou thermique, façonnage ...