1ère Enseignement transversal		
.Lycée Déodat de Séverac STI 2D	**Description fonctionnelle des systèmes**	**CHAPITRE 1**
	Langage de modélisation graphique de système : SysML	**Cours C11**
Objectif de la séance : L'élève doit être capable de lire et interpréter un diagramme du langage SysML, décrivant l'aspect comportemental ou structurel d'un système. **Durée :** 3h **Connaissances visées :** C.2.1 – Approche fonctionnelle des systèmes. C 2.2.2 – Représentations symboliques.		SysML™

1. Pourquoi SysML ?

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de le réaliser.
Un système est un ensemble de constituants inter-reliés qui interagissent les uns avec les autres d'une manière organisée pour accomplir une finalité commune (NASA 1995).
SysML est l'acronyme de Systems Modeling Language, soit Langage de Modélisation de Systèmes.
SysML apporte une standardisation du vocabulaire. Ce nouveau langage, ajoute aussi la possibilité de représenter les exigences du système comme elles sont définies dans un cahier des charges, les éléments non-logiciels (mécanique, hydraulique, capteur…), les équations physiques, les flux continus (matière, énergie, etc.) et les allocations.
En résumé, l'intérêt de SysML est :

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Générer des simulations de comportement d'un système.
- Documenter un projet.

2. Qui utilise SysML aujourd'hui ?

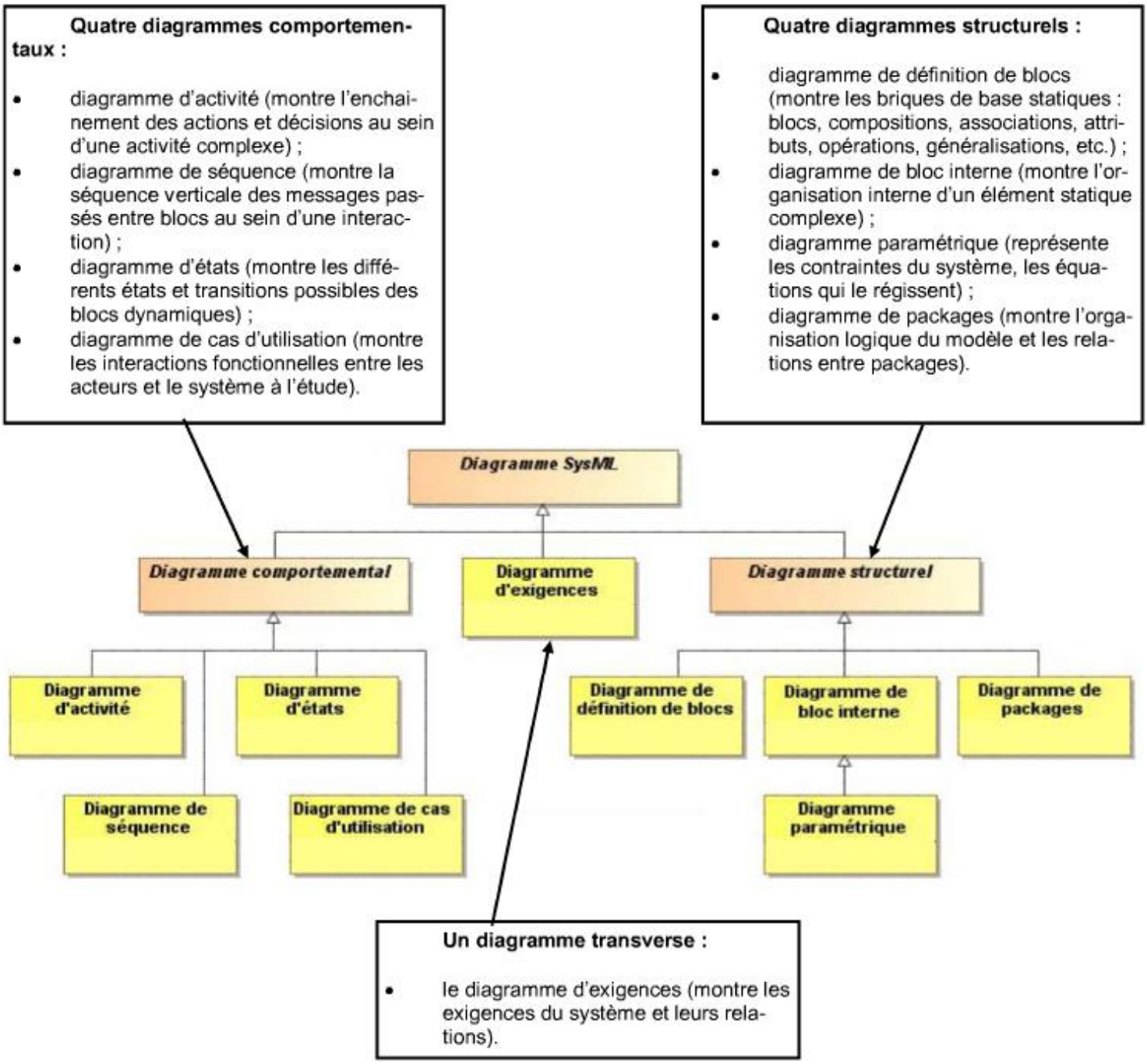
Airbus, CNES, NASA, Renault, BMW, VEGA Space GmbH, MIT Lincoln Laboratory, Lockheed Martin, US Army, ESO (European Organisation for Astronomical Research), Boeing, Raytheon, Thales, ESA (European Space Agency), Sopra Group, Rockwell Collins Inc., JPL (coentreprise avec la NASA), GE Aviation, NEWTEC LLC, BAE

Systems, Siemens AG, Philips, Bombardier Transportation, SKF, Peugeot SA, Cyclocity (Vélib',Velô Toulouse…), Alstom, Michelin, Segway, Valeo, Dassault Systemes et bien d'autres …

3. SysML et les différents diagrammes

SysML est un langage de modélisation permettant de décrire tout ou partie d'un système technique, d'un point de vue structurel, d'interactions ou comportemental.
SysML s'articule autour de 9 diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système. Dans ce chapitre, nous en étudierons 3 :

-Diagramme de cas d'utilisation
-Diagramme d'exigences
-Diagramme de séquence

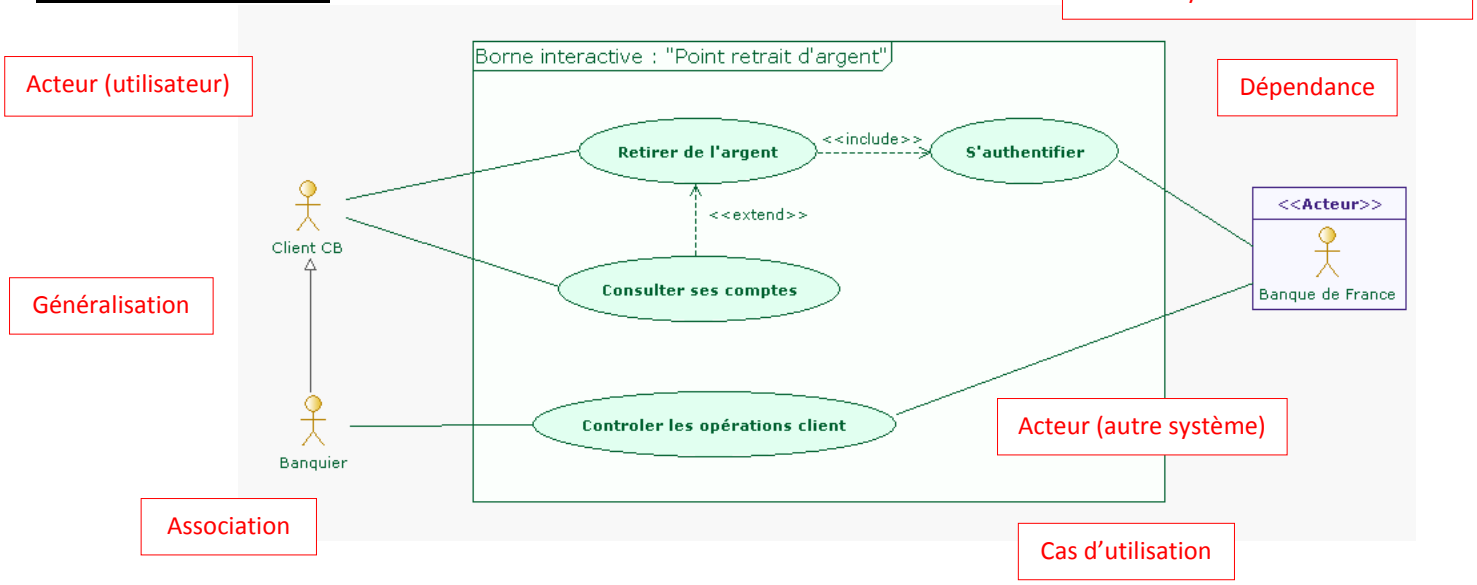


3.1 - Diagramme des Cas d'Utilisation (Use Case).

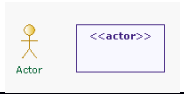
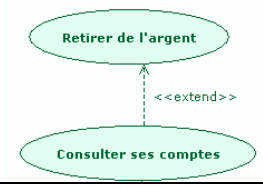
Il s'agit souvent du point de départ d'une conception. Le diagramme de cas d'utilisation permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
Ce type de diagrammes comporte :

- Des cas d'utilisation ;
- Des acteurs ;
- Des relations de dépendance, de généralisation et d'association.

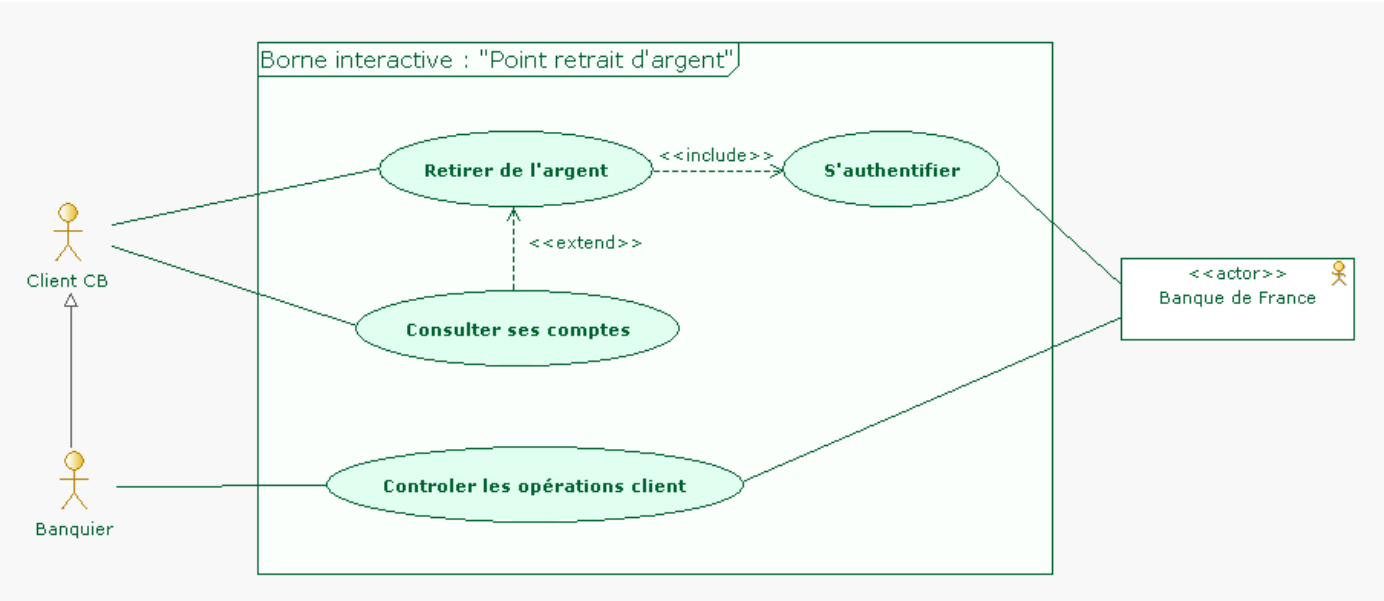
Eléments graphiques :



Interprétation des éléments graphiques :

Les acteurs 	Rôle joué par **un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.**
Les cas d'utilisation 	« Cas d'utilisation » est **une fonction ou interaction entre acteur et système, dans le but de répondre à un besoin. Un cas d'utilisation doit être relié à au moins un acteur. Il est exprimé par un verbe à l'infinitif.**
Les relations : **de dépendance,** 	**Dépendance** : il existe deux types de dépendance : - **Inclusion** (include): **un cas intègre le comportement d'un autre** (le cas d'utilisation « s'authentifier » est inclus dans « retirer de l'argent »). - **Extension** (extend): **comportement optionnel du système** (à partir de « retirer de l'argent », fonction du distributeur, on peut « consulter ses comptes »). *Les dépendances permettent de décomposer un cas complexe, mais l'abus de dépendance rend la lecture complexe.*
de généralisation 	**Généralisation**: **relation hiérarchique entre deux acteurs ou cas d'utilisation**. Le banquier est le spécialiste de la banque par rapport au client. La fonction « retirer de l'argent » est le cas général du système, « Consulter ses comptes » est un cas particulier.
et d'association 	**Association : chemin de communication entre un acteur et un cas d'utilisation.**

Interprétation de l'exemple :



À partir de l'exemple donné, répondre aux questions suivantes :

- ***Quel est le système étudié ?***

  Borne interactive : « Point retrait d'argent»

- ***Quel est le cas d'utilisation principal du système ?***

  Retirer de l'argent

- ***Quels sont les autres cas d'utilisation ?***

  Consulter ses comptes

  Contrôler les opérations client

- ***Quel est l'utilisateur (acteur) principal ? (cocher la bonne réponse)***

  ☐ Banquier ☒ Client CB ☐ Banque de France

- ***Quel est le rôle du banquier par rapport au client ?***

  Le banquier est le spécialiste de la banque par rapport au client

- ***Que représente la Banque de France ? (cocher la (ou les) bonne(s) réponse(s))***

  ☐ Acteur principal ☒ Acteur secondaire ☒ Un autre système

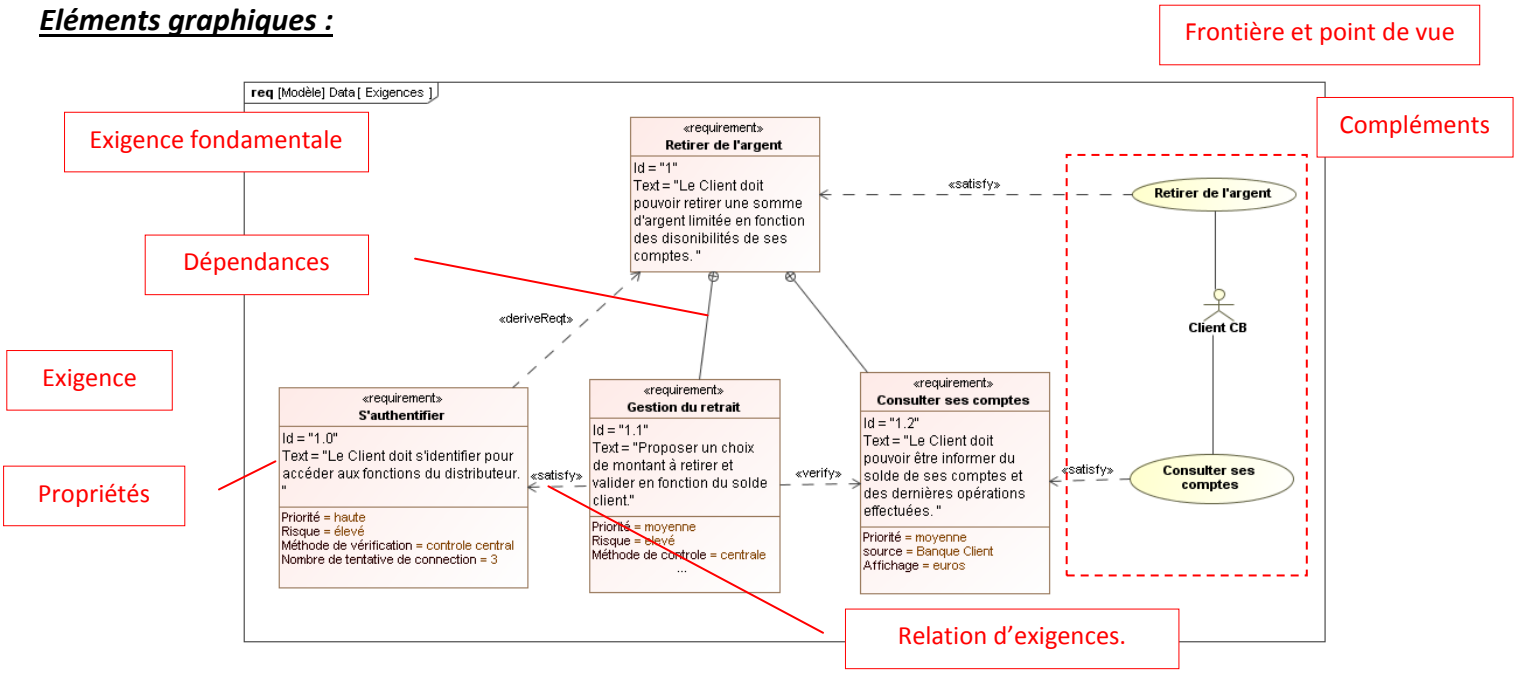
- ***Quelle est la différence graphique entre un acteur « humain » et un acteur « système » ?***

  Pour différencier les deux, on encadre l'acteur système

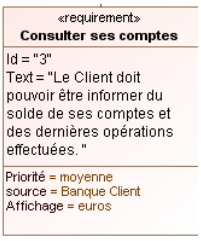
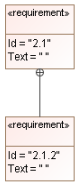
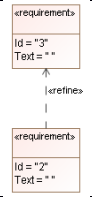
3.2 - Diagramme d'exigences.

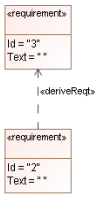
Le diagramme d'exigences permet **de représenter graphiquement les exigences du système en matière de contrainte ou de capacité afin de remplir une fonction et de satisfaire un besoin.**

Eléments graphiques :

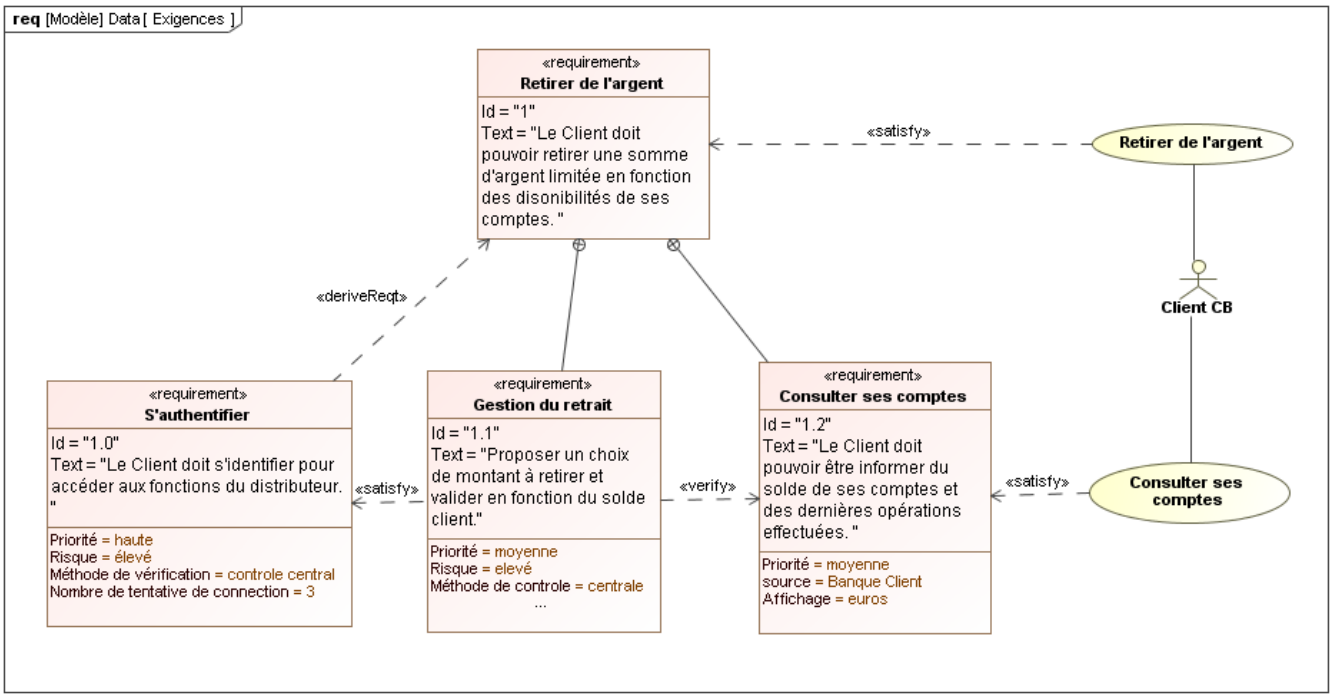


Interprétation des éléments graphiques :

Exigence 	Une exigence permet **de spécifier une capacité ou une contrainte qui doit être satisfaite par un système**. Elle peut spécifier une fonction à réaliser ou une condition de performance, de fiabilité, de sécurité… Les exigences représentent **le contrat (C.d.C.F) entre le client et les concepteurs du système.** Il est courant de définir d'autres propriétés quantifiées ou pas afin d'être précis sur les fonctions à remplir par le système.
Les dépendances	Les exigences peuvent être reliées entre elles par plusieurs types de relation, nous en utiliserons trois principalement : relation de contenance, de raffinement ou de dérivation :
	**la contenance** (ligne terminée par un cercle contenant une croix du côté du conteneur) (« confinement ») : **permet de décomposer une exigence composite en plusieurs exigences unitaires, plus faciles ensuite à tracer vis-à-vis de l'architecture ou des tests ;**
	**le raffinement** (« refine »): **consiste en l'ajout de précisions, par exemple de données quantitatives ;**

	**la dérivation** (« deriveReqt ») : **consiste à relier des exigences de niveaux différents**, par exemple des exigences système à des exigences de niveau sous-système, etc. Elle implique généralement des choix d'architecture.
Compléments **(facultatif)**	Le diagramme d'exigences permet tout au long d'un projet de relier les exigences entre elles et avec d'autres types d'élément SysML (cas d'utilisation, bloc, état) par plusieurs types de relation. **Relations d'exigence** : « verify » : **consiste à exiger une vérification de test entre deux éléments.** « satisfy » : **consiste à satisfaire un élément pour obtenir l'élément associé.**

Interprétation de l'exemple :



À partir de l'exemple donné, répondre aux questions suivantes :

Pour l'exigence « s'authentifier », quelle est la valeur définissant la propriété « nombre de tentatives de connexions » ?

nombre de tentative de connections : 3

Quel sont les exigences les plus sensibles au risque de sécurité ?

S'authentifier

Gérer le retrait

Parmi les exigences précédentes, laquelle a la priorité la plus élevée ?

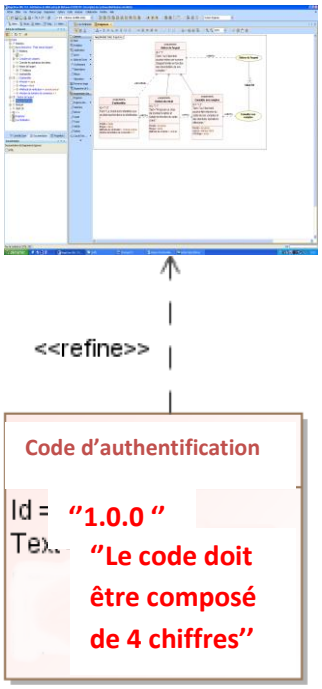
S'authentifier

Pour l'exigence « Gestion du retrait », quelles relations d'exigence avons-nous ?

« verify »

« satisfy »

Ajouter, sur le schéma ci-dessous, une dépendance de type « refine » à l'exigence « S'authentifier » :

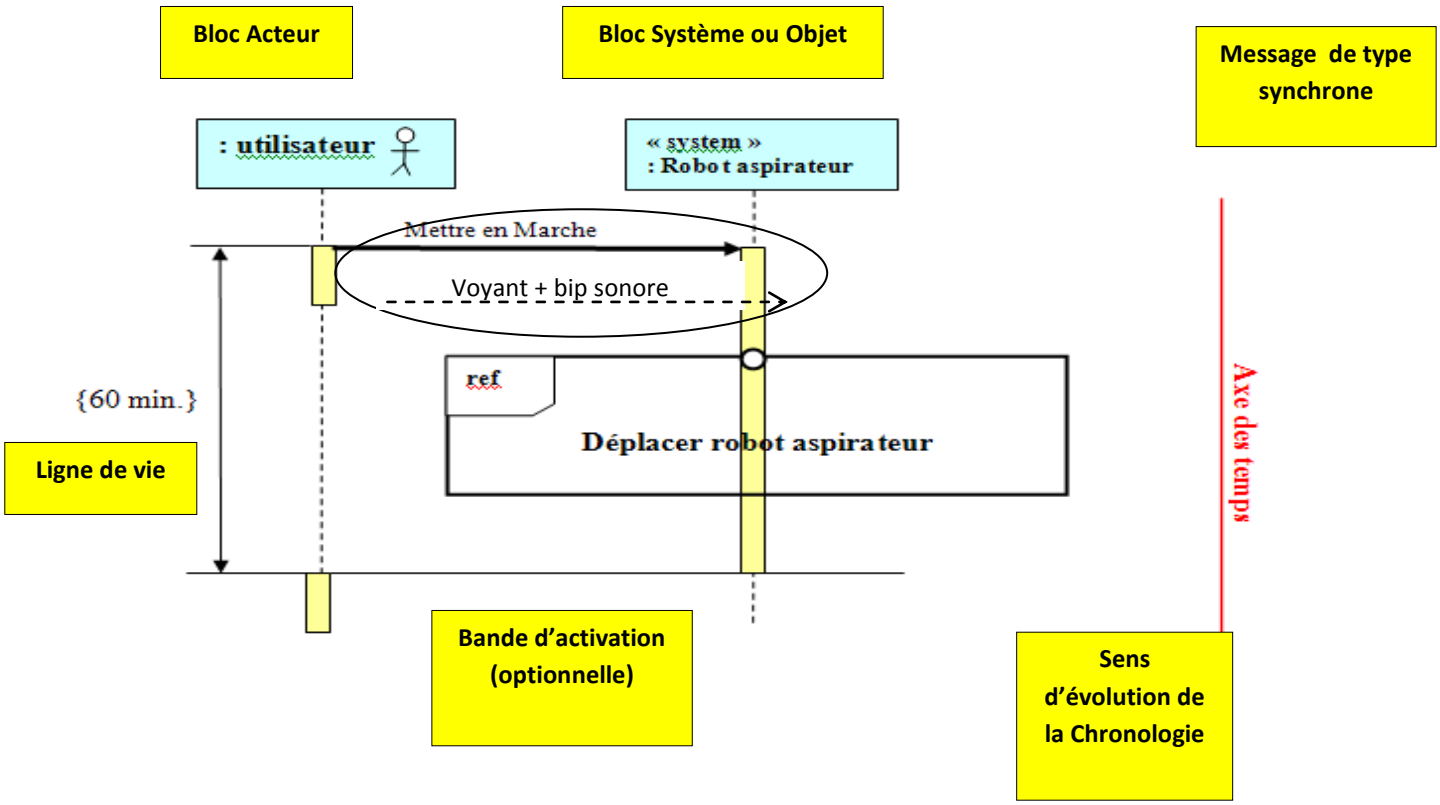


3.3 - Diagramme de séquence

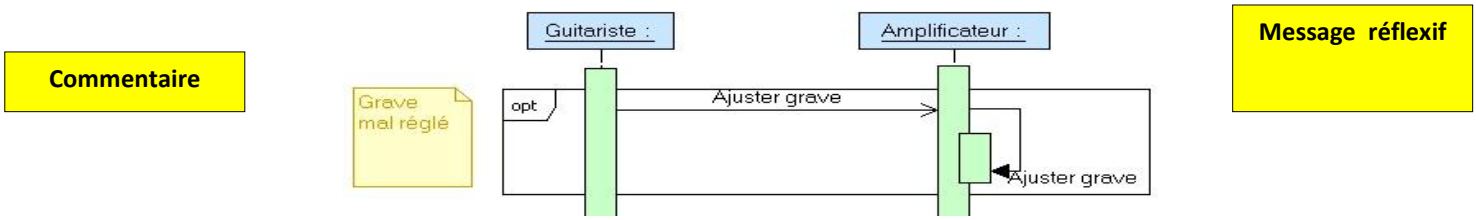
Un cas d'utilisation représente les interactions entre un acteur et le système d'un point de vue « boite noire », et comprend l'ensemble des scénarios identifiés. **Ces scénarios** peuvent être détaillés par un diagramme de séquence.
Le diagramme de séquence représente les éléments intervenant dans le scénario ainsi que les messages échangés **dans un ordre chronologique.**

Eléments graphiques :

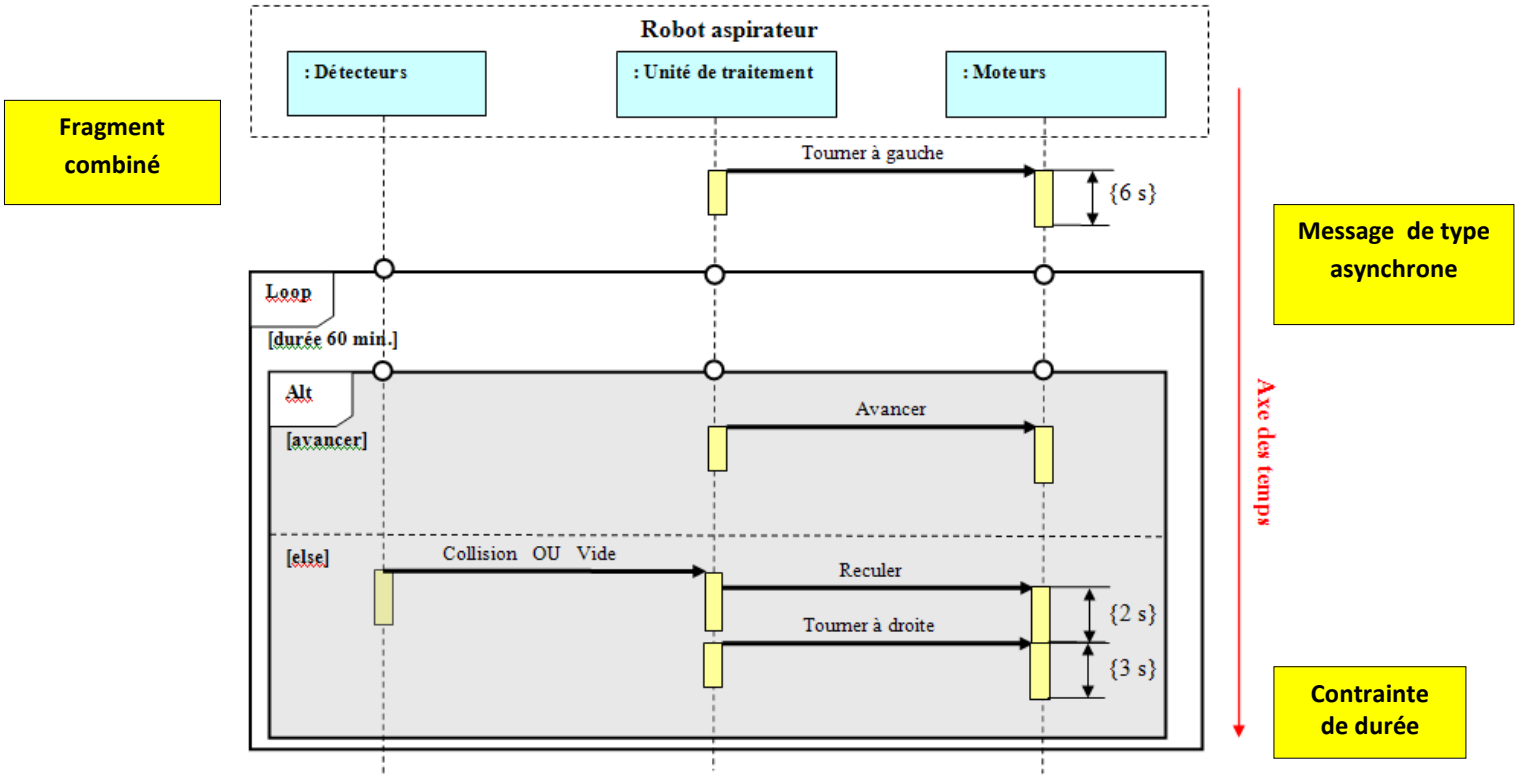
Dans un premier temps, on peut choisir de représenter les interactions entre l'acteur et le système (vue boîte noire).



Un objet est représenté par un bloc et une ligne verticale appelée « ligne de vie de l'objet ».
On peut représenter explicitement la période d'activité (ou bande d'activation) d'un objet, période pendant laquelle il exécute une action.
L'axe du temps est vertical, le flot de contrôles est horizontal.
Un message synchrone (émetteur bloqué en attente de réponse) est représenté par une flèche pleine, alors qu'un message asynchrone est représenté par une flèche évidée.
La flèche qui boucle (message réflexif) permet de représenter un comportement interne.



Afin de détailler les scénarios il est possible de rentrer en détails avec un diagramme de séquence qui représente les blocs internes du système intervenant (pour un message émis par l'acteur, le diagramme décrit l'enchaînement des messages échangés entre les blocs internes du système).
On parle ainsi de la vue boîte blanche (comportement du système).



<u>Les principaux fragments combinés</u> :

LOOP : boucle. Le fragment peut s'exécuter plusieurs fois et la condition explicite l'itération.

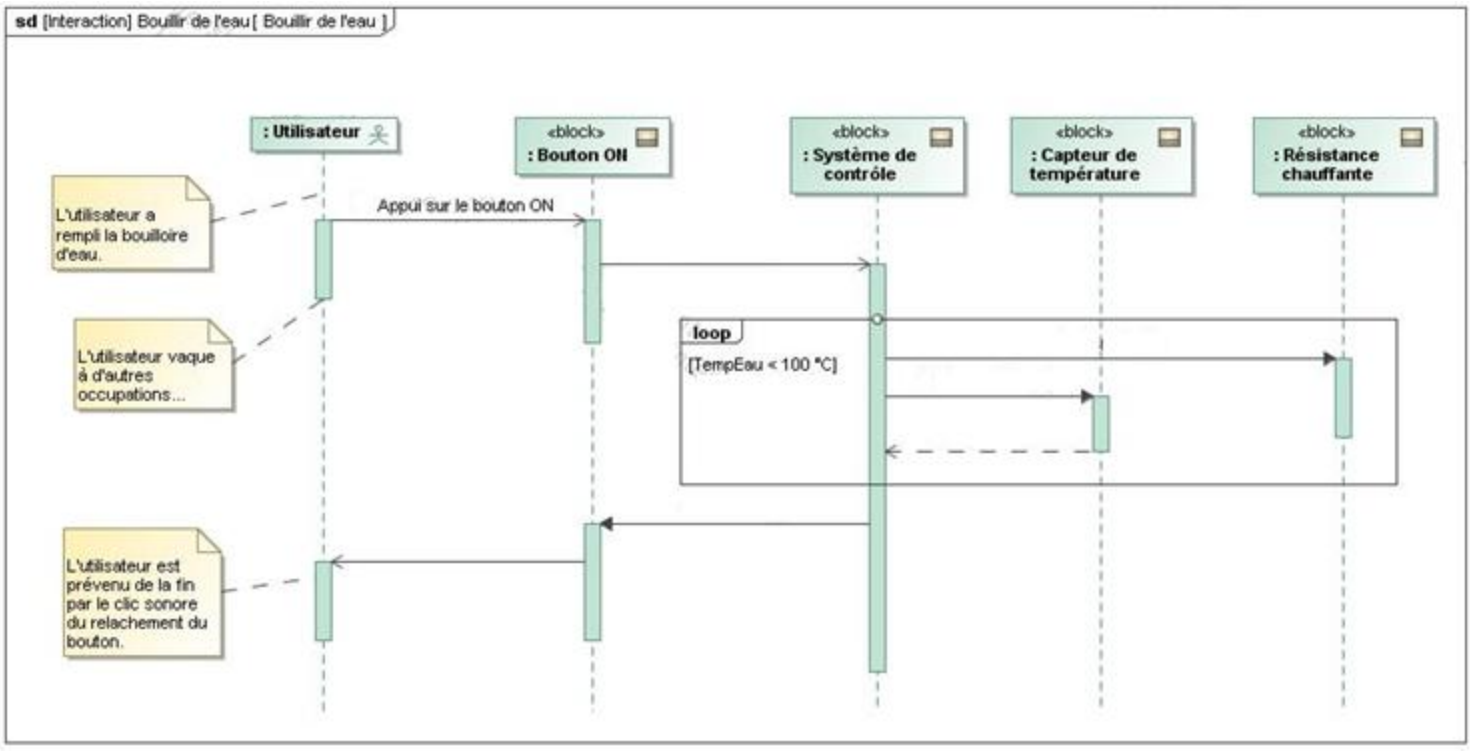
ALT : alternatif. Seul le fragment possédant la condition vraie s'exécute.

OPT : optionnel. Le fragment ne s'exécute que si la condition est vraie.

REF : référence. Un diagramme de séquence peut faire référence à un autre diagramme de séquence.

PAR : parallèle. Plusieurs opérations peuvent être exécutées en parallèle.

Interprétation d'un exemple :



De quel système le diagramme de séquence ci-dessus décrit-il le scénario ?

Une bouilloire d'eau électrique

Qui est l'acteur ?

L'utilisateur

Quelles sont les blocs internes du système intervenant dans ce diagramme de séquence ?

Bouton ON/Système de contrôle/Capteur de température/Résistance chauffante

Placer les séquences ci-dessous dans l'ordre chronologique.
(Chauffer/Relâcher/Température d'eau atteinte/Clic sonore/Mesurer la température/Démarrer)

1- **Appui sur le bouton ON**

2- **Démarrer**

3- **Chauffer**

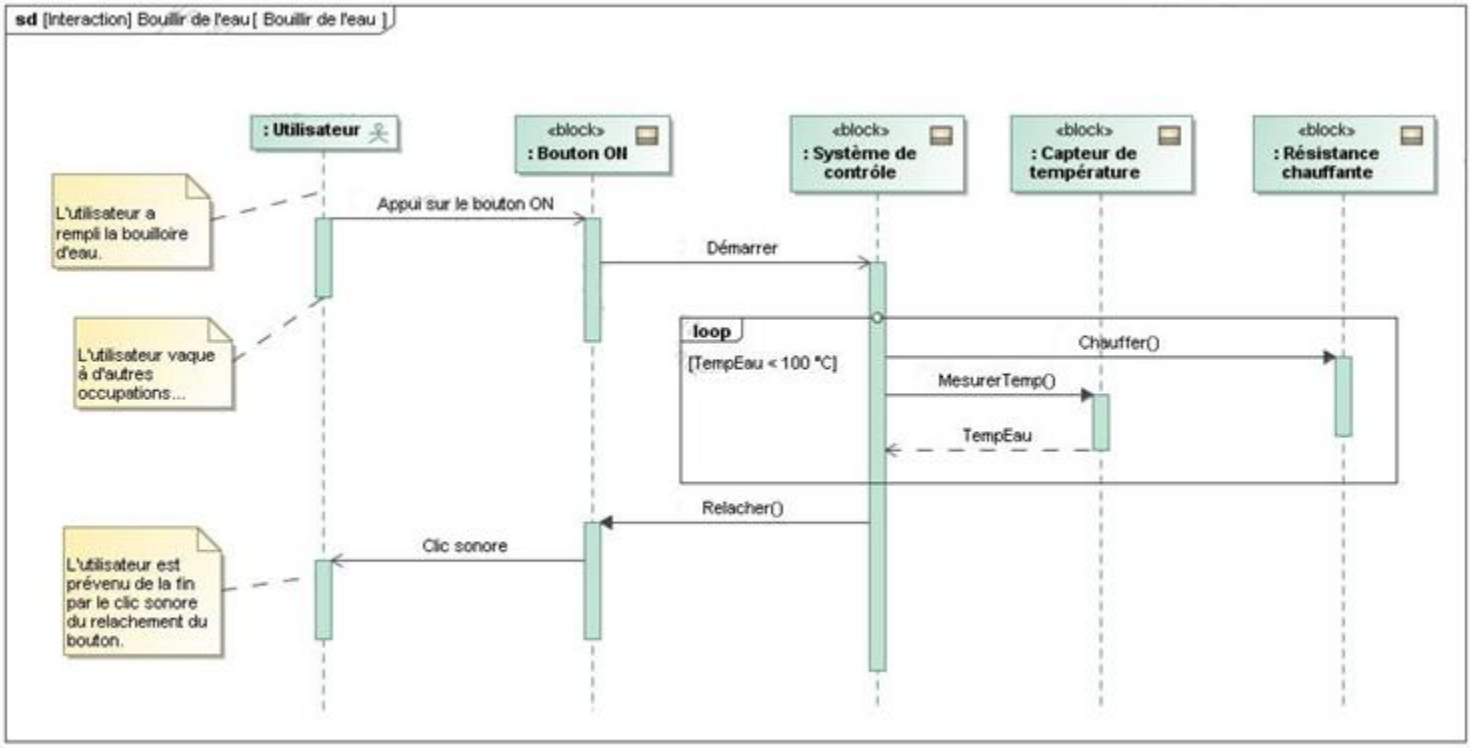
4- ***Mesurer la température***

5- ***Température d'eau atteinte***

6- ***Relâcher***

7- ***Clic sonore***

A partir de la chronologie définie précédemment, compléter le diagramme de séquence ci-dessus.



Décrire le fragment combiné : « loop »

Tant que la température est inférieure à 100°C, le bloc « Système de contrôle » envoie l'ordre de chauffer au bloc « résistance ».