LYCÉE de l'ÉLORN
le choix de votre réussite

Fiche de cours
Ingénierie système
SysML

Documents disponibles sur sciencesdelingenieur.fr ▸ menu ressources pédagogiques ▸ Spécialité Sciences de l'Ingénieur

1- Ingénierie des systèmes

1.1 Pourquoi une ingénierie système ?

- **Pour répondre au mieux au besoin.**
  Éviter le cas typique d'une situation où la solution envisagée ne répond pas au besoin initial, ou n'est pas adaptée à celui-ci. Il s'agit souvent de solutions dite "de facilité" choisies pour des raisons budgétaires et/ou temporelles. Elles vont souvent engendrer des surcoûts ou des délais supplémentaires.

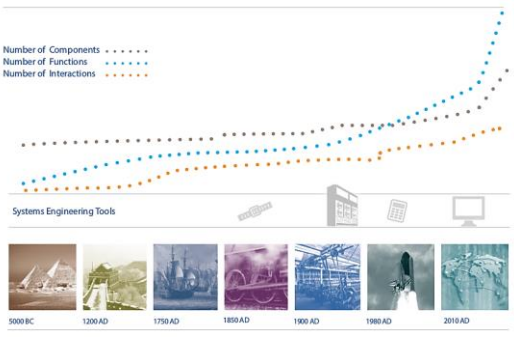
Facteurs d'échecs	Causes racines	Facteurs de succès
37 %	Besoins, exigences	40 %
9 %	Projet, ressources	23 %
8 %	Gestion des données techniques	14 %
11 %	Technique, technologie	9 %

Étude du Standish Group (cabinet international de conseil de projets) sur les causes du succès ou d'échecs

Exemple : En 2014, les nouveaux trains régionaux sont conçus plus larges que leurs prédécesseurs, afin de répondre au mieux aux attentes du public (en termes de nombre de places, de confort d'utilisation, ...). Les ingénieurs de la SNCF de l'époque, qui ont définis le cahier des charges et notamment les dimensions des nouvelles rames, avaient omis de vérifier que celles-ci étaient conformes aux infrastructures existantes. Aucune norme n'existant alors quant à l'écartement entre le quai et la voie, celui-ci pouvait légèrement varier d'une gare à l'autre. Conséquence : 1300 quais (sur les 9000 que comptent les gares françaises) nécessitèrent un rabotage de 1 à 2 cm !

- **Pour organiser au mieux le projet.**
  On peut classer les projets en trois catégories :
  - **Succès** : le projet est réalisé en temps et en heure, dans les budgets alloués, avec toutes les fonctionnalités et caractéristiques spécifiées à l'origine.
  - **Challengé** : le projet est terminé et opérationnel, mais avec un dépassement du budget et/ou des délais, et moins de fonctionnalités que prévues initialement.
  - **Echec** : le projet est abandonné avant la fin prévue ou n'est jamais mis en œuvre.

- **Pour s'adapter à la complexité croissante des systèmes.**
  La complexité croissante des systèmes, des projets, induit des facteurs de risques de plus en plus nombreux et fréquents qu'il faut savoir aujourd'hui gérer et anticiper.



1.2 Le produit, le système

Il a été imaginé et réalisé pour satisfaire un besoin. Un produit n'est pas obligatoirement un objet technique mais peut aussi bien être un service ou un processus. 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.

La description des systèmes nécessite d'utiliser un langage qui soit compréhensible par toutes les parties qui vont soit l'utiliser, le vendre, le maintenir, le fabriquer ou le concevoir. Nous allons dans ce premier cours de l'année commencer par définir cette notion de système et mettre en place un outil de description.

1.3 Le besoin

La première question que l'on doit se poser lors de l'étude d'un système est : À quoi sert-il ? Il est donc nécessaire de définir le besoin :

« Un besoin est une nécessité, un désir éprouvé par un utilisateur »

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. Les besoins exprimés pour un même objectif évoluent au cours du temps :

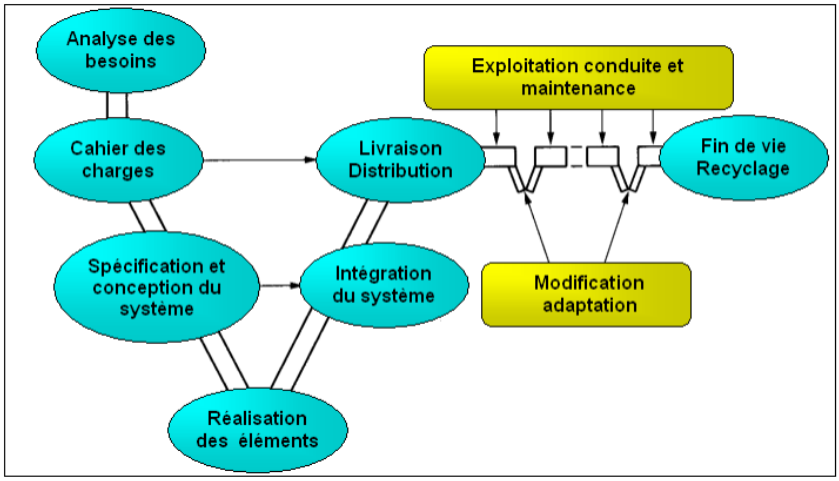
- D'une simple description d'un usage dans les premiers temps ;
- En précisant ensuite des contraintes ;
  - de coûts,
  - de facilité d'utilisation,
  - d'affichage de l'innovation ;
- En précisant plus récemment :
  - l'impact environnemental,
  - l'impact social,
  - l'impact sur le développement,
  - …

Ci-dessus, l'évolution des produits et des besoins liés au nettoyage d'un sol chez les particuliers.

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 fil électrique et sans sac	Nettoyer le sol automatiquement
				

1.4 Cycle de vie d'un système

Afin de mieux cerner les liens qui existent entre un système, son environnement et les différents intervenants sur ce système, il est classique de représenter la vie d'un système sous la forme d'un V (ou plus précisément d'une racine carrée).



1.4.1 Analyse du besoin :

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

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

☞ Quels besoins pourraient satisfaire nos futurs smartphones ?

1.4.2 Cahier des Charges Fonctionnel (C.d.C.F.) :

Le cahier des charges fonctionnel est un document qui permet de formaliser avec précision le besoin du demandeur. C'est un tableau de bord qui définit le projet et détaille les conditions dans lesquelles il doit être réalisé. C'est le lien de compréhension entre l'entreprise et le client.

1.4.3 Spécification et conception du système :

C'est la phase qui permet de définir précisément ce que doit être le système, d'en choisir tous les composants et la manière dont ils seront reliés.

1.4.4 Réalisation des éléments :

Au cours de cette phase de passage à la réalité, l'ensemble des constituants matériels : pièces mécaniques, moteurs, cartes électroniques, écrans de visualisation, etc. et immatériels : programmes informatiques, logiciels, etc. sont réunis en vue de l'assemblage.

1.4.5 Intégration du système :

C'est la phase de montage, d'assemblage de composants matériels et immatériels

1.4.6 Livraison et Distribution :

Cette phase de vie peut apporter des contraintes supplémentaires au système.

1.4.7 Exploitation, conduite et maintenance :

La dernière phase correspond en fait à la vie réelle du système en fonctionnement depuis la livraison jusqu'à la mise au rebut finale. Pendant cette phase, le système est existant. C'est la phase de rentabilité du système, elle peut être entrecoupée de phases de modification, adaptation et amélioration du système.

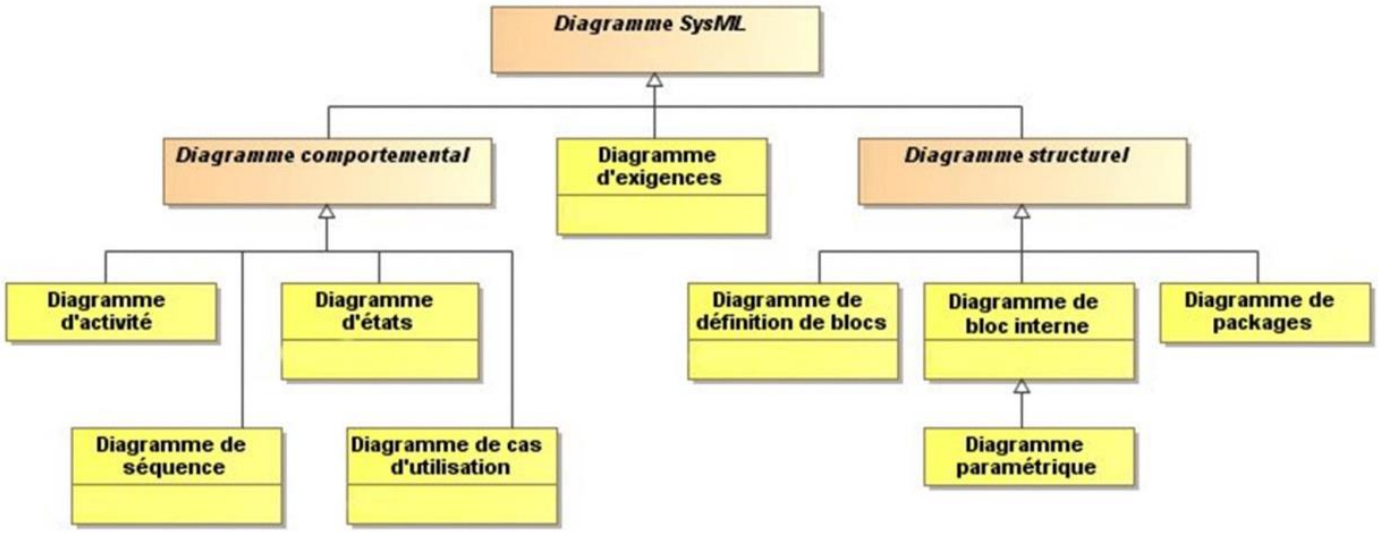
2- Le langage SysML

La modélisation avec le langage SysML est un outil relativement récent (début des années 2000) qui fait partie de l'UML (Unified Modeling Language), une standardisation utilisée dans l'industrie logicielle.

2.1. Présentation

Le langage SysML - Systems Modeling Language - est un langage de modélisation spécifique au domaine de l'ingénierie système. Il permet la spécification, l'analyse, la conception, la vérification et la validation de nombreux systèmes.

Le langage SysML s'articule autour de neuf types de diagrammes. Chacun d'eux étant dédié à la représentation des concepts particuliers d'un système.



Le diagramme des exigences : diagramme transversal, puisque chaque élément modélisé est en lien avec au moins une exigence, qui montre les besoins/exigences du système et leurs relations

Les diagrammes comportementaux :

- **diagramme de cas d'utilisation** : montre les services rendus par le système, en interaction avec les acteurs concernés ;
- **diagramme de séquence** : montre le déroulement temporel des interactions entre le système et son environnement (boîte noire), voire au sein du système lui-même (boîte blanche) ;
- **diagramme d'états** : montre les différents états du système et leurs modes de changement ;
- **diagramme d'activité** : montre l'enchainement séquentiel des activités réalisées dans les états.

Les diagrammes structurels :

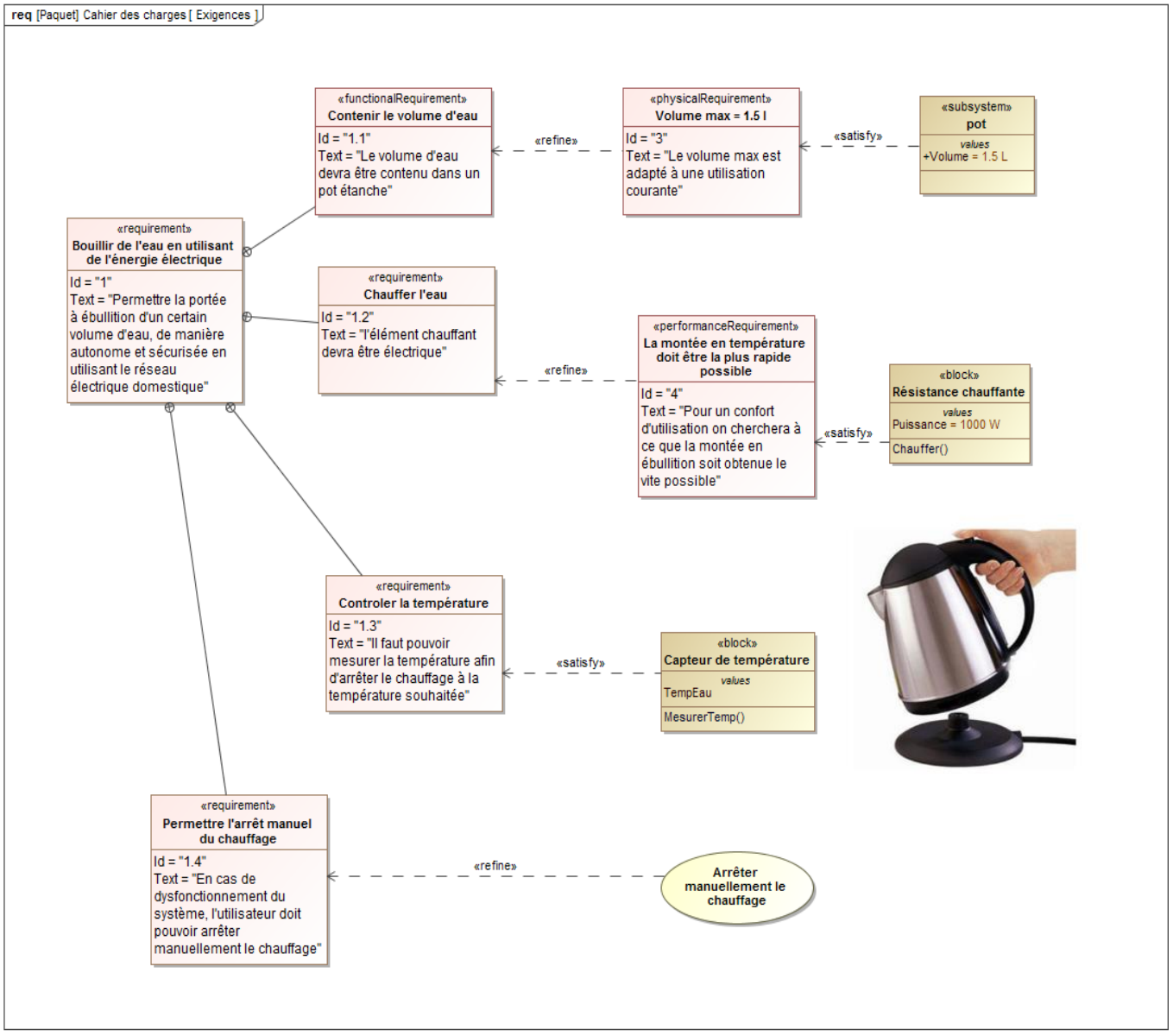
- **diagramme de définition de blocs** : montre la structure du système sous forme de décomposition en éléments plus simples (sous-systèmes, blocs) ;
- **diagramme de bloc interne** : montre l'organisation interne des constituants et leurs interactions en termes d'échanges de flux MEI (Matière – Energie – Information) ;
- **diagramme paramétrique** : représente les contraintes des constituants, les équations qui les régissent ;
- **diagramme de packages** : montre l'organisation logique du modèle et les relations entre packages.

2.2 Le diagramme d'exigences (REQUIREMENT DIAGRAM, notation sysml : REQ)

Ce diagramme décrit les exigences du cahier des charges fonctionnel qui servent à établir un contrat entre le client et les concepteurs du futur système.
Une exigence exprime une capacité ou une contrainte à satisfaire par un système. Elle peut exprimer une fonction que devra réaliser le système ou une condition de performance technique, physique, de sécurité, de fiabilité, d'ergonomie, d'esthétisme...

Exemple d'une bouilloire électrique :

On indique l'exigence du système dans le premier rectangle, avec un texte descriptif et un identifiant unique. On décompose cette exigence en exigences unitaires. On peut y ajouter des données quantitatives et des précisions.



Les relations au sein du diagramme d'exigences

Lien de contenance : 

Permet de décomposer une exigence composite en plusieurs exigences unitaires.

Lien de dérivation « derive » ou deriveReqt » : 

Il exprime un lien entre deux exigences de niveaux différents. Celle située à l'origine de la flèche découle (sans y être contenue) de l'exigence pointée pour exprimer une cohérence du système. Il implique généralement un choix d'architecture.

Lien de raffinement « refine » : 

Il précise une exigence pointée par la flèche, souvent par des données quantitatives.

Lien de solution « satisfy » : 

Il précise le composant ou la solution technique, satisfaisant l'exigence pointée par la flèche.

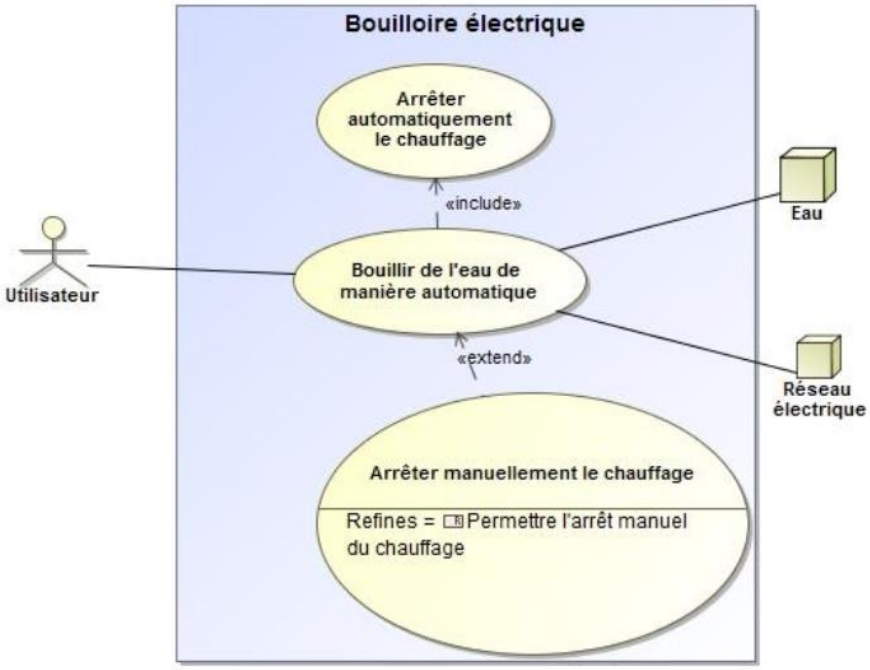
2.3 Diagramme des cas d'utilisation (USE CASE DIAGRAM, notation sysml : UC)

Il montre les interactions fonctionnelles entre les acteurs (utilisateurs, agents de maintenance, techniciens, vendeurs, ...) et le système étudié. Il délimite précisément la frontière du système et décrit ce qu'il fera sans spécifier comment.
Il exprime les services (**use cases**) offert par le système étudié aux utilisateurs (**actors**). Ce diagramme ne doit indiquer ni la manière dont il va assurer les services, ni les solutions technologiques envisagées.

La représentation graphique du diagramme des cas d'utilisation :

- On trace un cadre délimitant le système étudié et contenant un ensemble de séquences d'actions (elles peuvent aussi être liées entre elles).
- A gauche du cadre, on place les acteurs humains, à droite du cadre on place les acteurs non humains (un acteur non humain est représenté par un rectangle).
- On décrit les actions réalisables par le système étudié dans des bulles placées à l'intérieur du cadre (les services rendus par le système aux acteurs sont généralement décrits par un verbe à l'infinitif suivi de compléments).
- Les acteurs peuvent être reliés entre eux soit par une flèche bidirectionnelle (chaque acteur agit sur l'autre) soit par une flèche unidirectionnelle (un acteur agit sur l'autre).

Exemple de la bouilloire électrique :



Les relations au sein du diagramme des cas d'utilisation

Lien d'extension « extend » : «extend»

Le cas d'utilisation de base peut incorporer celui placé à l'origine de la flèche. Il revêt un caractère optionnel.

Lien d'inclusion « include » : «include»

Le cas d'utilisation de base incorpore systématiquement celui placé à l'extrémité de la flèche.

Lien de spécialisation / généralisation :

Ils relient des cas d'utilisation descendants qui héritent de la description d'un cas d'utilisation supérieur (parent commun).

Plusieurs diagrammes des cas d'utilisation peuvent être établis pour un système, en fonction des circonstances d'utilisation (circonstance normale, circonstance en défaillance, circonstance en maintenance, ...) ceci afin d'en améliorer la compréhension et la lecture.

2.4 Diagrammes de définition de blocs (notation sysml : BDD)

Le bloc SysML (block) constitue la brique de base pour la modélisation de la structure d'un système.

On distinguera :

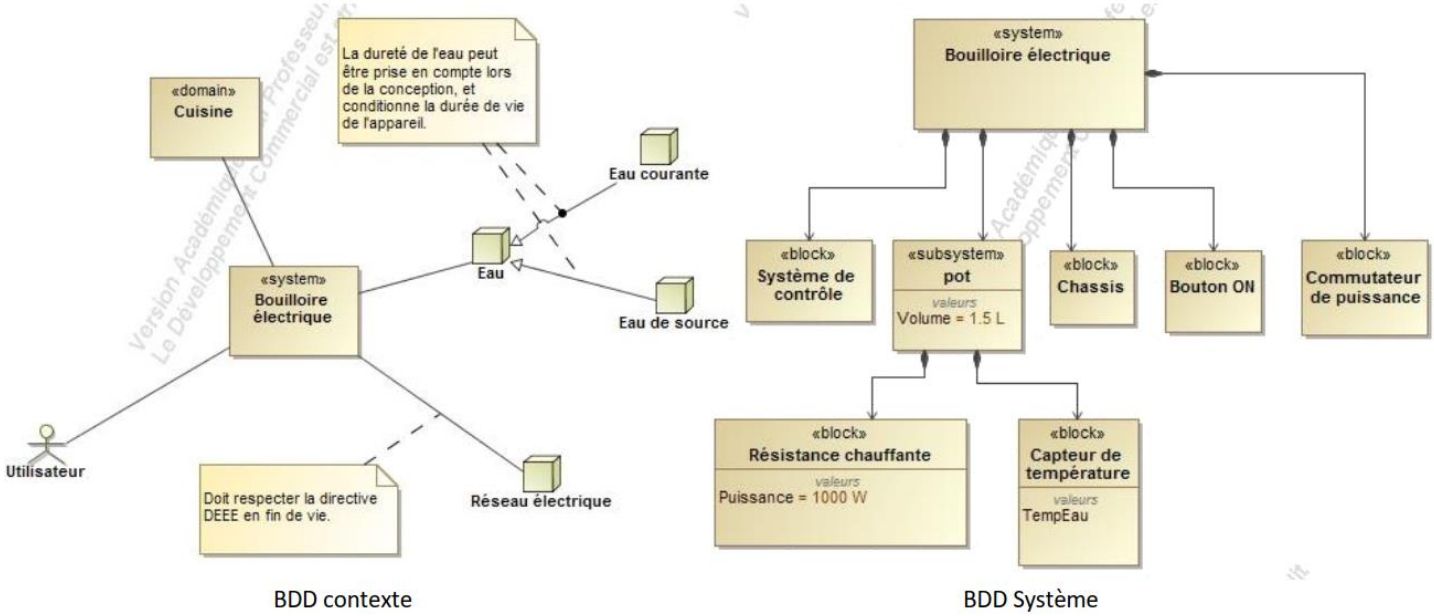
Le diagramme de définition de Blocs du contexte :

Il définit le système dans son environnement (éléments humains et matériel qui interagissent avec lui).

Le diagramme de définition de Blocs du système :

Il définit l'architecture matérielle et logicielle globale du système sous-représentation arborescente de blocs. Chacun d'eux se limite à la définition d'une famille (ou classe) de composants principaux.

Exemple de la bouilloire électrique :



BDD contexte

BDD Système

Les relations au sein des diagrammes de définition de Blocs

Lien de spécification / généralisation : 

Ils relient des cas d'utilisation descendants qui héritent de la description d'un cas d'utilisation supérieur (parent commun).

Relation d'association :

Ce simple trait exprime un lien d'égal à égal qui permet souvent de relier le contexte au système étudié.

Relation de composition :

Le bloc situé du côté du losange plein a besoin d'un sous-bloc côté flèche.

Relation d'agrégation : 

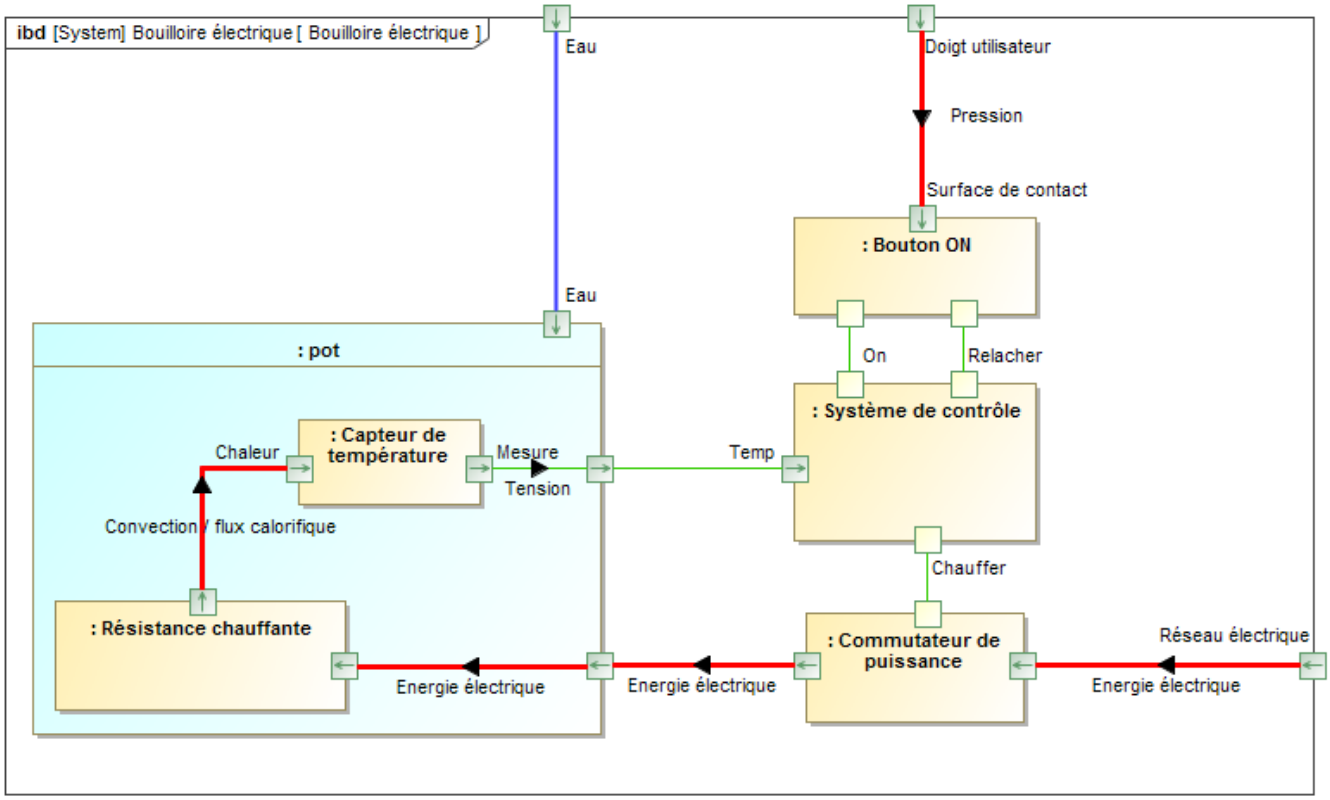
Relation similaire au lien de composition mais qui revêt un caractère optionnel.

Sa lecture : « …. peut avoir un.... »

- Un BDD ne décrit ni les fonctions, ni le comportement du système.
- Ce diagramme est utile pour montrer les « grosses briques » du système.

2.5 Diagramme de bloc interne (notation : INTERNAL BLOCK DIAGRAM « IBD »)

Il décrit la structure interne d'un bloc issu du BDD, c'est à dire ses composants, ou blocs internes et les échanges, flux de matière, d'énergie ou d'information, entre ses composants. Le diagramme de bloc interne fait apparaître l'ensemble des composants d'un bloc issu du BDD ainsi que les flux et leurs orientations.



Les ports et les relations au sein du diagramme de bloc interne

Le port de flux : 

Cette interface autorise l'entrée et/ou la sortie du flux (matière, énergie, information) vis à vis d'un bloc. Ce port possède au moins un sens, son nom est facultatif.

Le port standard : 

Il représente une interface qui n'est pas liée à un flux mais à un service, une opération, une consigne ou un ordre de commande.

Le connecteur :

Ce lien relie deux ports. En cas de flux, sa nature peut être précisée.

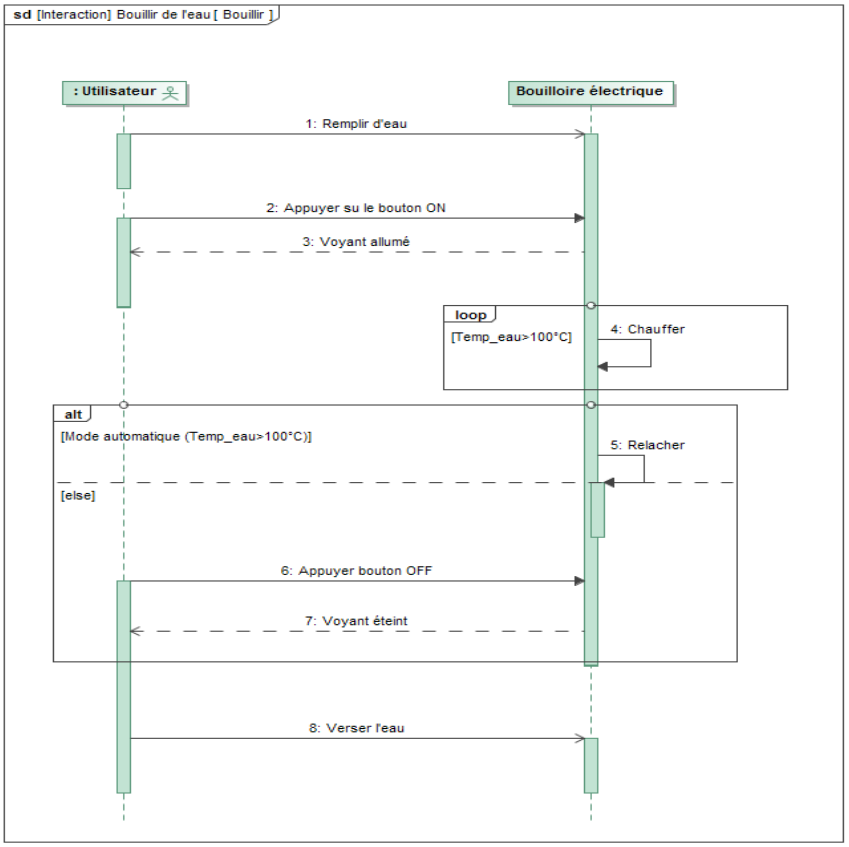
Conseils pour la rédaction du diagramme de bloc interne :

- Chaque bloc du BDD contenant d'autres blocs peut être représenté par un IBD.
- Attention à bien faire la différence entre port de commande et port de flux.

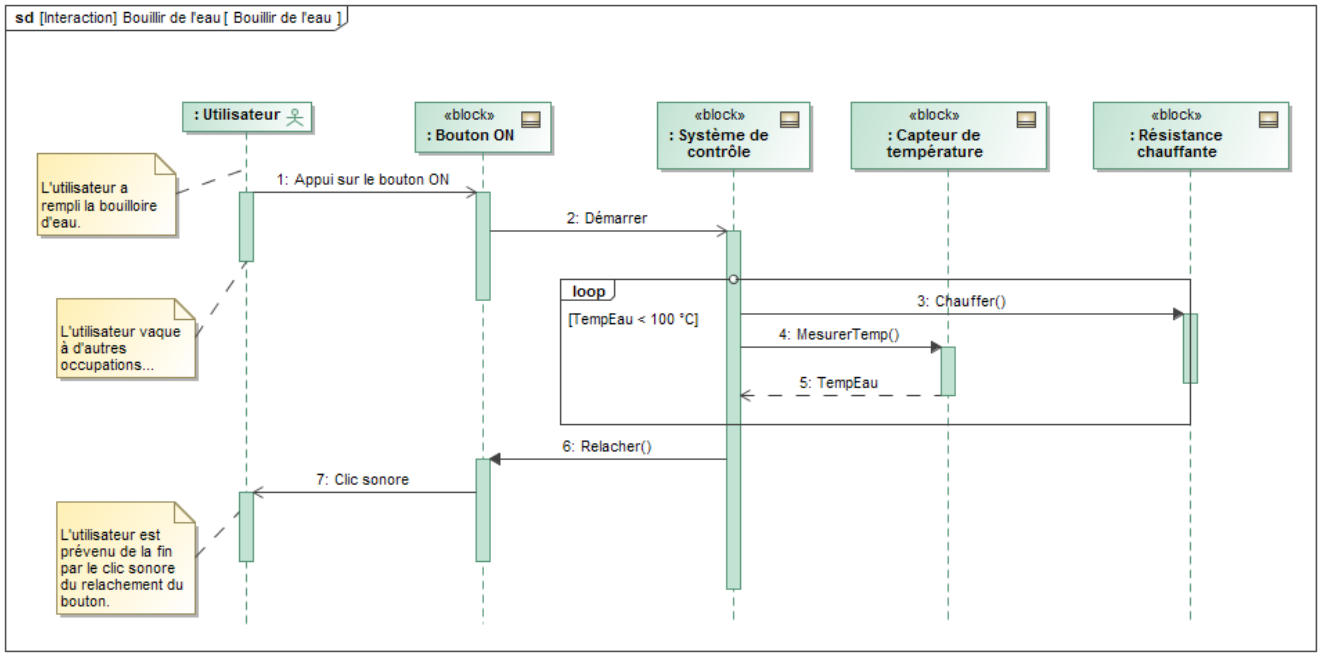
2.6 Le diagramme de séquence (notation sysml SEQUENCE DIAGRAM ou SD)

Il sert à décrire comment le système va remplir son contrat au niveau d'un cas d'utilisation. Il permet de décrire l'enchaînement séquentiel des interactions. Cela permet donc de montrer comment le système se comporte dans des scénarios de réussite comme dans des scénarios d'échec.
Chaque élément situé dans le haut du diagramme est un **objet, un acteur ou un constituant** pris comme une boîte noire. Les lignes verticales pointillées sont des **lignes de vie**. Elles représentent le temps sans qu'il y ait une échelle. Les rectangles superposés dessus désignent la **période d'activité de l'objet**.

Exemple avec deux lignes de vie :



Une autre version à plusieurs lignes de vie :



Les liens et conditions dans le diagramme de séquence

Les lignes horizontales entre des « messages ». Les messages étant des signaux, des événements ou des invocations d'opérations. Ils sont de deux types :

Les messages synchrones : ——————→

L'émetteur attend une réponse suite à son message. Il ne fait donc rien entre l'émission et la réception de la réponse. C'est donc bloquant pour l'émetteur. Il y a forcément un retour de ce message (puisqu'il est synchrone) et représenté par une flèche en pointillés. - - - - - - - - - - - - ->

Les messages asynchrones : ——————→

L'émetteur envoie son message sans attendre de réponse en retour.

Les types de conditions, dans un fragment combiné, sont :

Type « Alt » :

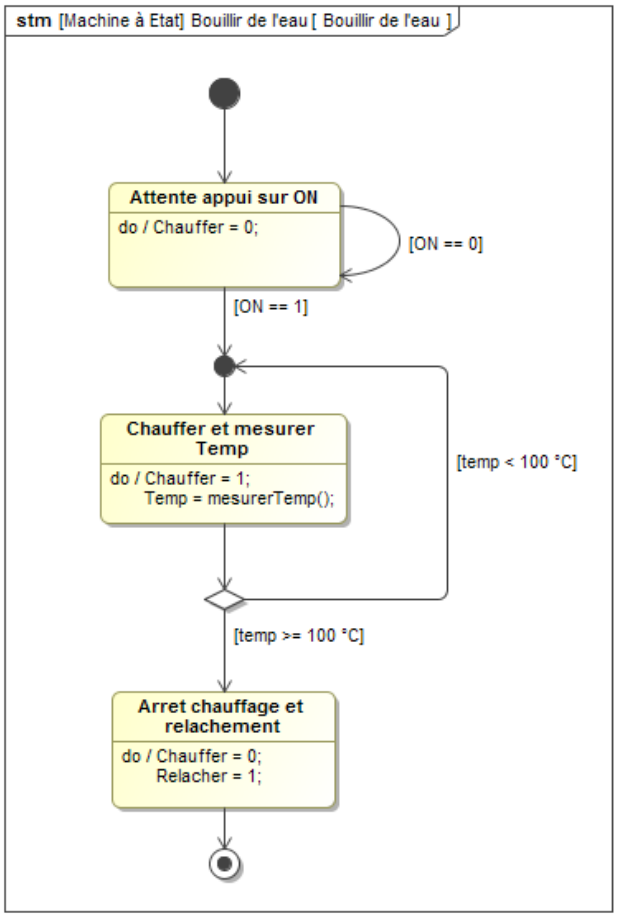
Indique une alternative. Suivant la condition, l'une ou l'autre alternative sera réalisée.

Type « Loop » :

Indique que ce qui est réalisé en boucle tant que la condition de garde est vraie.

2.7 Le diagramme d'états (notation sysml STATE MACHINE ou STM)

Il décrit les états que peut prendre le système et les transitions qui régissent les changements d'états.



La symbolique du diagramme d'états :

Etat initial : 

Représente le moment initial.

Etat final : 

Représente le moment final.

Etat : 

Représente un moment spécifique du comportement d'un objet qui correspond à une ***séquence (entrée, phase active ou attente, sortie)***.

Transition : 

Elle est représentée par une flèche, elle constitue un changement allant d'un état source à un état cible. Un état est actif lorsqu'une transition y mène et devient inactif lorsqu'une transition le quitte.
Un événement provoquant la transition peut être écrit à côté de la flèche de transition mais ce n'est pas obligatoire.