



# LES DIAGRAMMES SysML

Mémo

Professeur: YASSINE FARTOUH

v1.1

CPGE - MARRAKECH

## 1 Introduction

Le langage **SysML (Systems Modeling Language)** est un langage de modélisation permettant de décrire tout ou partie d'un système technique, d'un point de vue transversal, comportemental ou structurel. C'est un outil de communication utilisé dans toutes les phases de vie du produit afin que les différents acteurs puissent communiquer et mesurer les écarts entre le produit en cours de réalisation et celui souhaité.

Le **SysML** a été adopté par l'**OMG (Object Management Group)** en Juillet 2006, et sa version 1.0 est rendue officielle en Septembre 2007.

### 1.1 Les différents diagrammes

Il s'articule autour de 9 types de diagrammes, dont seulement 7 sont au programme des CPGE. Aucun ordre d'analyse des 7 diagrammes n'est préconisé, un travail simultané sur l'ensemble est souhaité.

Toute modification de l'un des diagrammes engendre généralement des modifications dans les autres représentations.

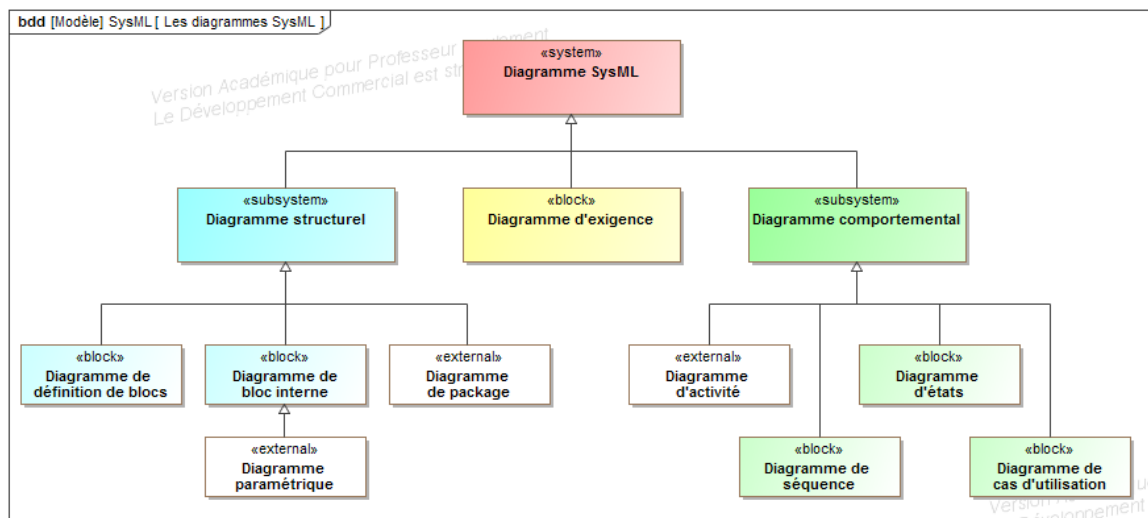


FIGURE 1 – Récapitulatif des diagrammes SysML

### 1.2 Éléments graphiques communs

Tous les diagrammes SysML comportent en haut à gauche un **cartouche**, qui permet d'identifier le diagramme en indiquant au minimum le type de diagramme et le nom de l'élément concerné.

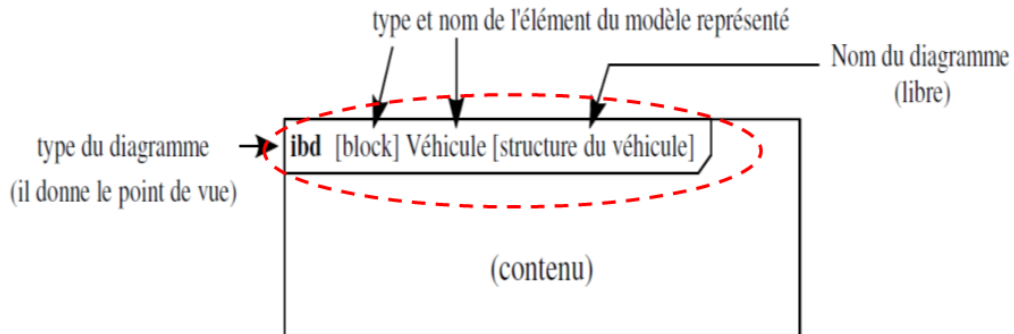


FIGURE 2 – Exemple de cartouche

## 2 Diagramme de contexte

Le **diagramme de contexte** permet de **définir les frontières de l'étude** et en particulier de préciser la phase du cycle de vie dans laquelle on situe l'étude (généralement la phase d'utilisation). Il répond à la question : « *Quels sont les acteurs et éléments environnants du système ?* ». Il n'y a pas de syntaxe spécifique à utiliser.

Ce diagramme s'appuie sur la notion de bloc (Voir **diagramme bdd**).

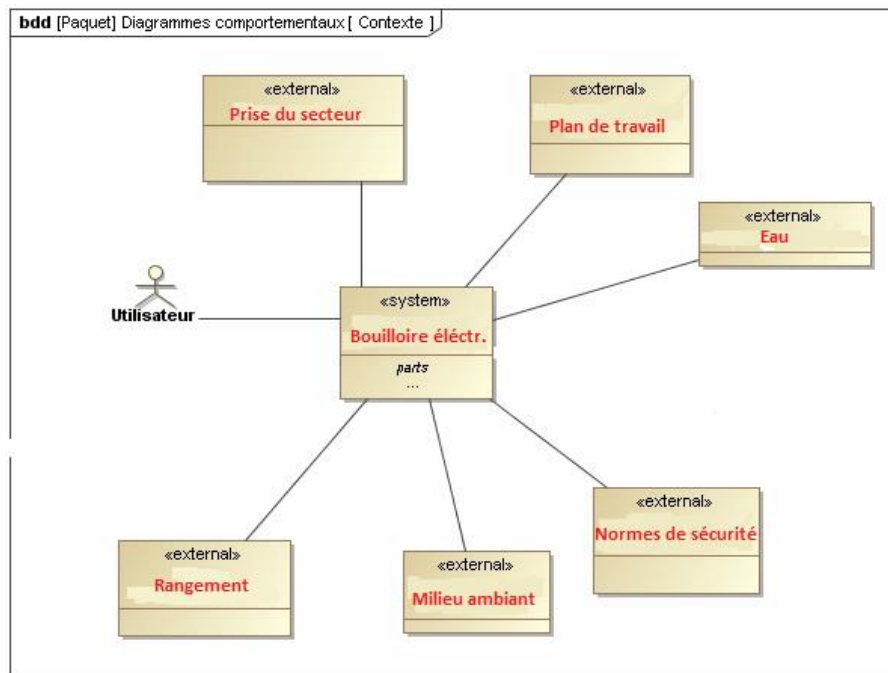


FIGURE 3 – diagramme de contexte : Bouilloire électrique

### 3 Diagramme des cas d'utilisation (uc : use case diagram)

Le **diagramme des cas d'utilisation** est un diagramme fonctionnel qui montre les fonctionnalités offertes par le système.

Il répond à la question : « *Quels services rend le système aux différents acteurs ?* ».

Un **acteur** est un utilisateur (stickman) ou bien un élément extérieur (bloc).

Le diagramme des cas d'utilisation exprime alors les services offerts par le système aux utilisateurs (acteurs).

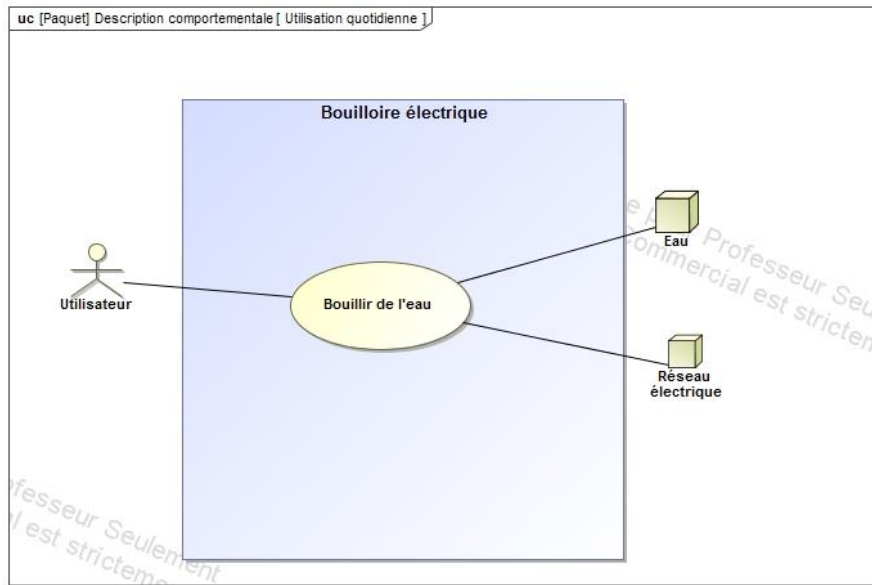


FIGURE 4 – diagramme de cas d'utilisation : Bouilloire électrique

La description de la syntaxe est fournie ci-dessous :

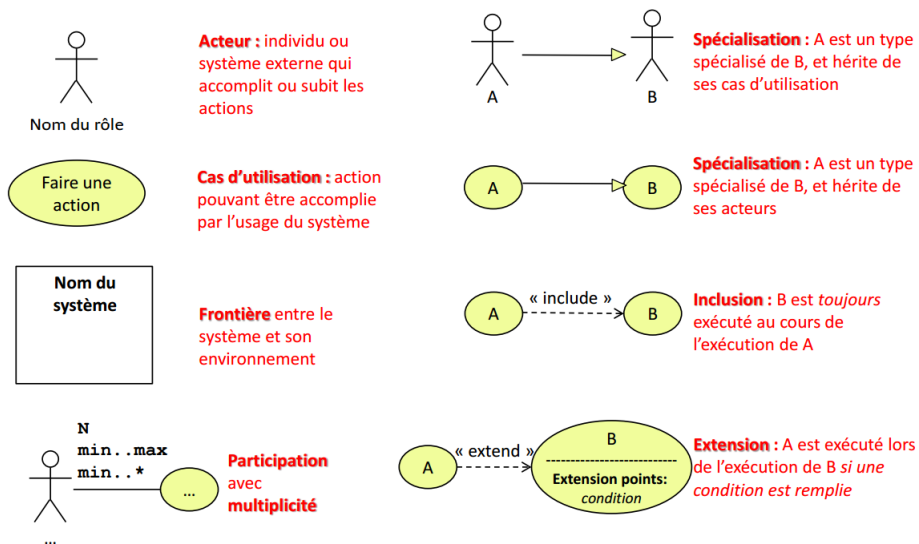


FIGURE 5 – Syntaxe du diagramme des cas d'utilisation

## 4 Diagramme des exigences (req : requirement diagram)

### 4.1 Notion d'Exigence

Une exigence permet de spécifier une capacité ou une contrainte qui doit être satisfaite par un système.

Elle peut spécifier une fonction que le système devra réaliser ou une condition de performance technique, physique, de fiabilité, de sécurité, etc.

Les exigences servent à établir un contrat entre le client et les réalisateurs du système.

### 4.2 diagramme des exigences

C'est un diagramme fonctionnel qui décrit les exigences du cahier des charges fonctionnel. Il répond à la question : « *Quels sont les contraintes et fonctionnalités du système ?* ».



#### Remarque

Les exigences sont exprimées par des verbes à l'infinitif.

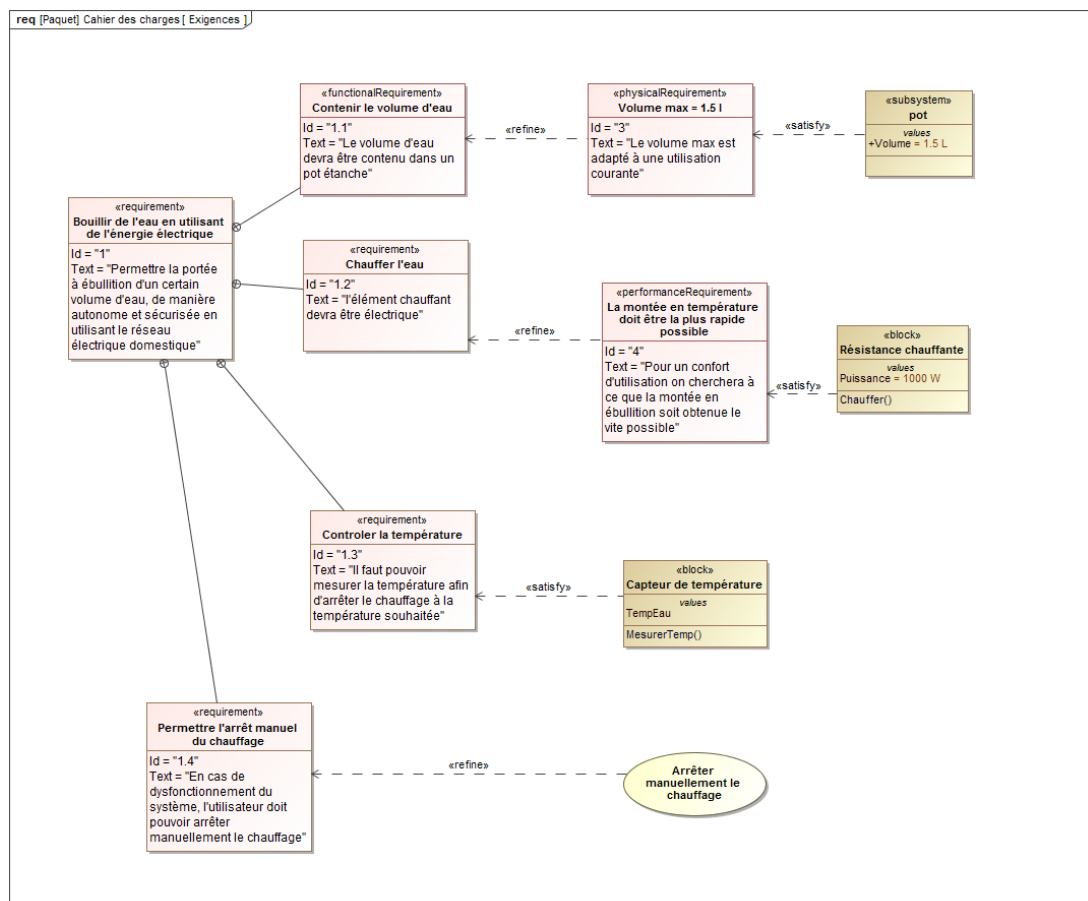


FIGURE 6 – diagramme d'exigence : Bouilloire électrique



## Remarque

un CdCF peut être explicité par les diagrammes uc et req en leur adjoignant une liste de fonctions de service et contrainte du système tout en précisant les critères d'appréciation, les niveaux et les flexibilité

La description de la syntaxe est fournie ci-dessous :

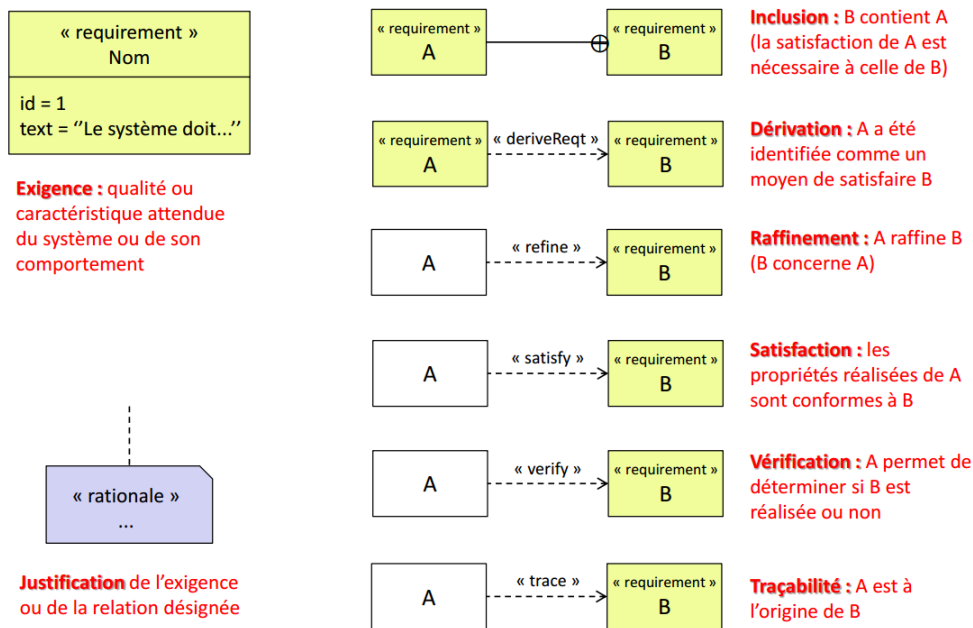


FIGURE 7 – Syntaxe du diagramme des exigences

## 5 Le diagramme de définition de blocs (*bdd* : *block definition diagram*)

Le **diagramme de définition des blocs** décrit la hiérarchie du système et les classifications système/composant. Il répond à la question « *qui contient quoi ?* », ce qui permet de voir rapidement la structure globale du système.

Il est utilisé pour décrire l'architecture matérielle du système, et est similaire à la première page d'une notice de montage d'un meuble, indiquant la liste des éléments et des pièces à assembler avec leurs quantités respectives.

Un bloc peut modéliser tout le système (voir diagramme de contexte), un élément matériel ou logiciel.

On peut également préciser le nombre d'entités près des extrémités des traits. Ce nombre est 1 par défaut. Lorsque une relation de généralisation est utilisée, les blocs spécialisés :

- héritent de toutes les propriétés du bloc général ;
- peuvent éventuellement en posséder d'autres qui leurs sont spécifiques.

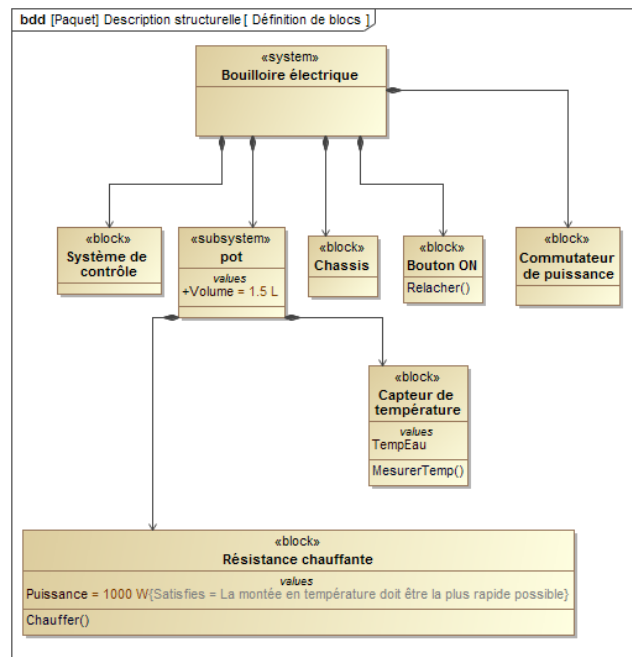


FIGURE 8 – diagramme de définition des blocs : Bouilloire électrique

La description de la syntaxe est fournie ci-dessous :

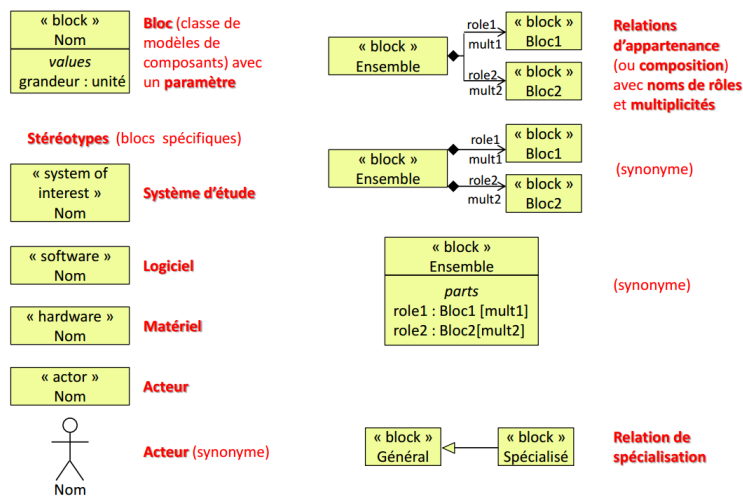


FIGURE 9 – Syntaxe du diagramme *bdd*



### Remarque

Il existe également la relation d'agrégation simple. Par rapport à la relation de composition (ou agrégation forte), elle est modélisée par un **losange vide**. La nuance principale concerne le lien entre les deux blocs. Pour une composition, la destruction de l'élément supérieur entraîne la destruction de l'élément inférieur, ce qui n'est pas le cas pour une agrégation simple.

## 6 Le diagramme de bloc interne (*ibd* : *internal block diagram*)

Le diagramme **ibd** permet de représenter les **échanges de matière, information et énergie** entre les éléments (parts) d'un même bloc.

Chaque bloc d'un diagramme **bdd** contenant d'autres blocs peut être représenté par un diagramme **ibd**.

Le diagramme des blocs internes permet également de décrire la logique de connexion, de services et de flux entre blocs grâce au concept de « **port** ». Ces ports spécifient les points d'interaction **offerts** et **requis** entre les blocs.

Les ports peuvent être de deux natures :

- **port de flux (flow port)** : canal d'Entrée/Sortie par lequel transite de la matière, de l'énergie ou de l'information (MEI) ;
- **port standard** : désigne une interface permettant d'invoquer un service/une opération.

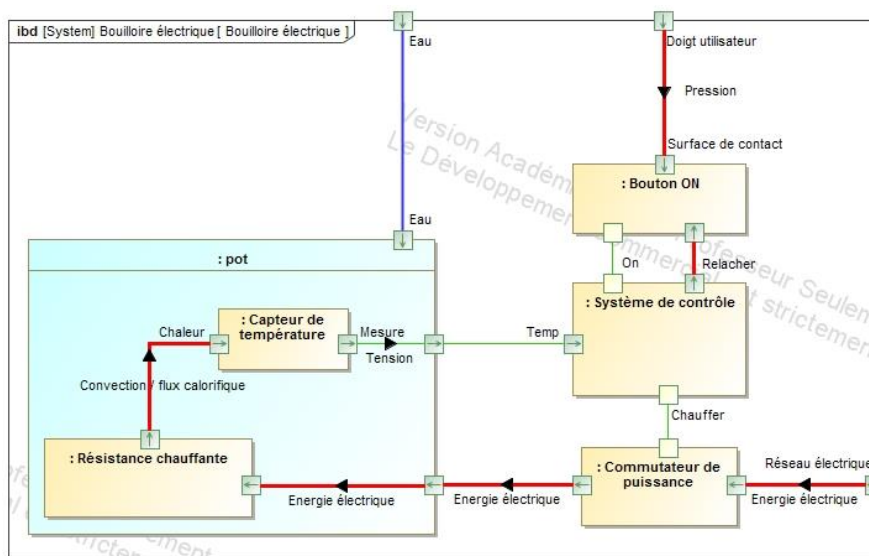


FIGURE 10 – diagramme de définition de bloc interne : Bouilloire électrique

La description de la syntaxe est fournie ci-dessous :

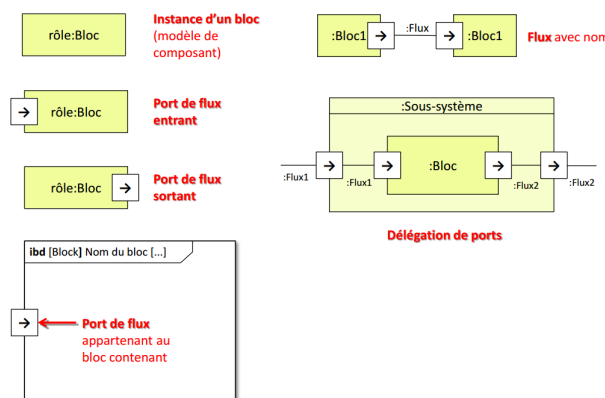


FIGURE 11 – Syntaxe du diagramme *ibd*

## 7 Le diagramme de séquence - (*sd* : *Sequence Diagram*)

C'est un diagramme dynamique qui représente les échanges de messages entre les acteurs et le système, ou entre des parties durant une séquence temporelle d'actions appelée scénario. Il permet de représenter la dynamique d'un cas d'utilisation ou la collaboration d'un ensemble d'objets internes au système (lignes de vie) sans préjuger de comment il la réalisera.



### Remarque

Un cas d'utilisation est décrit par au moins un diagramme de séquence.

Un cas d'utilisation peut dépendre d'autres cas d'utilisation. Il s'appellera alors : **fragment**. Chaque fragment possède un opérateur et peut être divisé en opérandes. Les principaux opérateurs sont :

- **loop** : boucle. Le fragment peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération ;
- **opt** : optionnel. Le fragment ne s'exécute que si la condition fournie est vraie ;
- **alt** : fragments alternatifs. Seul le fragment possédant la condition vraie s'exécutera.
- **par** : fragments parallèles. S'exécutent simultanément ;
- **break** : la séquence en cours s'interrompt si la condition précisée est vraie.

Deux types de contraintes temporelles qui sont :

- **Contrainte de durée** : durée exacte, minimale ou maximale entre 2 événements ;
- **Contrainte de temps** : associées à des instants dans la description de la séquence d'échanges entre éléments.

La description de la syntaxe est fournie ci-dessous :

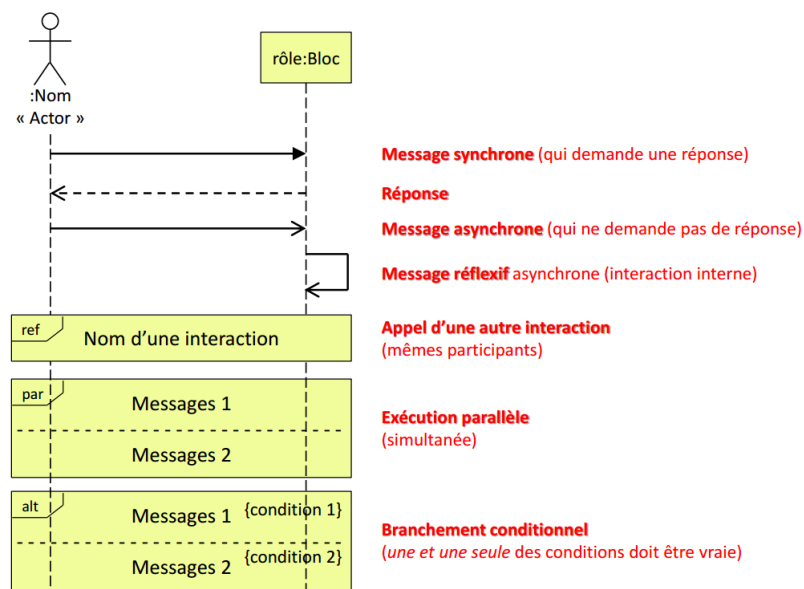


FIGURE 12 – Syntaxe du diagramme de séquence

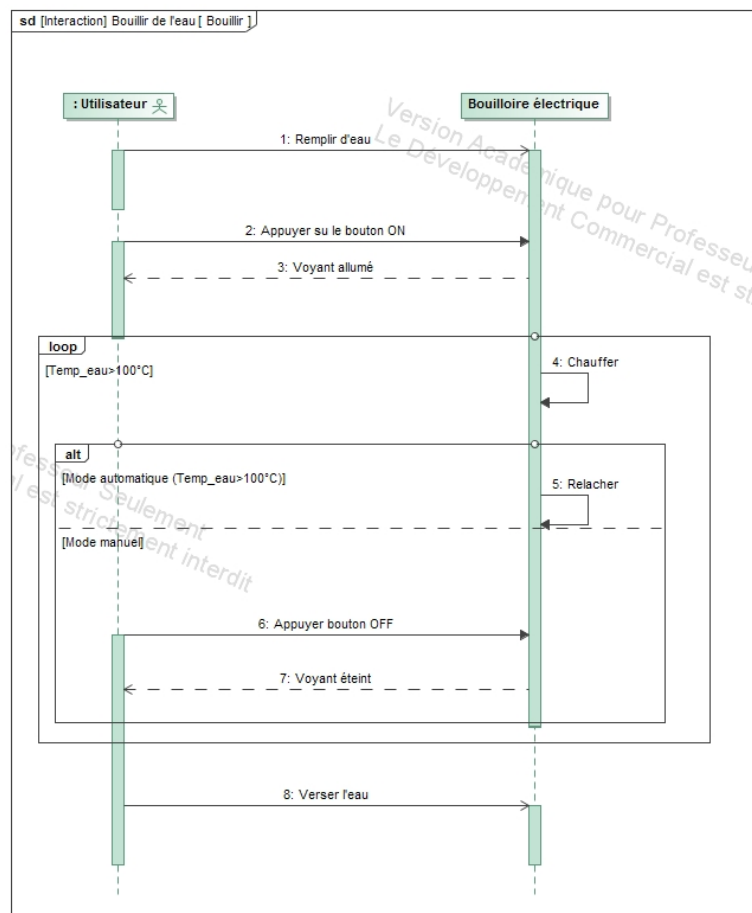


FIGURE 13 – diagramme de séquence : Bouilloire électrique

## 8 Le diagramme d'états - (*stm : State Machine Diagram*)

### 8.1 Notion d'état

Un état représente une situation durant la vie d'un bloc pendant laquelle :

- il satisfait une certaine condition ;
- il exécute une certaine activité ;
- il attend un certain événement.

### 8.2 Notion d'événement

C'est une spécification pouvant déclencher une réaction, ou porter des paramètres qui matérialisent le flot d'informations ou de données reçues. Il peut être accompagné de conditions.

### 8.3 Diagramme d'état

Sert à décrire le fonctionnement d'un système (ou sous-système) en montrant les différents **états** successifs qu'il prend, en fonction des interactions. Il décrit les **transitions** entre états et les **actions** que le système ou ses parties réalisent en réponse aux **événements**.

Le diagramme d'état répond à la question : « *Comment représenter les différents états du système ?* ».

Une transition de passage d'un état à un autre possède :

- un événement déclencheur ;
- une condition ou condition de garde ;
- un état cible.

### 8.4 Condition ou condition de garde

C'est une expression booléenne qui doit être vraie lorsque l'événement arrive pour que la transition soit déclenchée. Elle est notée entre crochets.

- S'il y a plusieurs transitions avec un même événement, elles doivent avoir des conditions différentes.
- Lorsqu'il y a plus de deux conditions, il est recommandé d'utiliser le mot-clé **else** pour garantir l'exhaustivité.
- Les changements d'état interne se modélisent en utilisant le mot-clé **when** suivi d'une expression booléenne dont le passage de faux à vrai déclenche la transition.
- Le passage du temps se modélise en utilisant le mot-clé **after** suivi d'une expression représentant une durée, décomptée à partir de l'entrée dans l'état courant

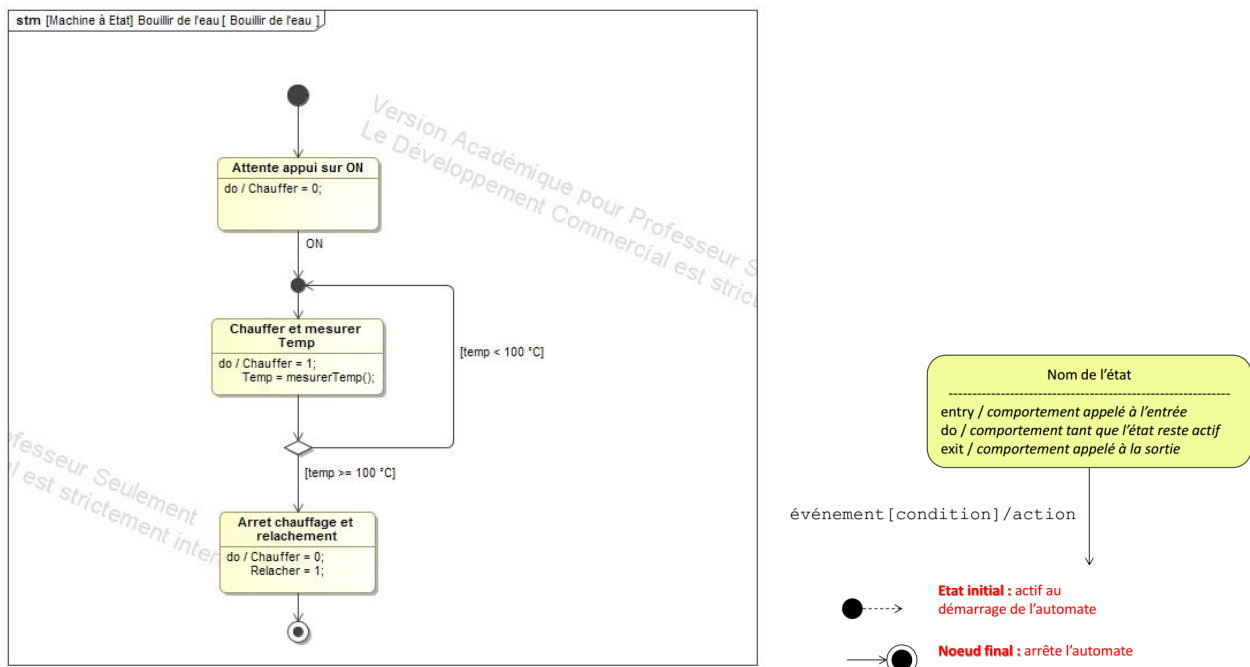


FIGURE 14 – diagramme d'état : Exemple de Bouilloire électrique et Syntaxe des Statecharts