









Séquence 1 : Expérimenter et analyser les systèmes  
pluritechnologiques-Annexe 1

# Éléments de syntaxe en SysML

Acronymes et symboles utilisés dans cette annexe :

Nom français	English name	Acronyme	Symbole
Diagramme d'exigences	Requirement Diagram	RD	
Diagramme de contexte	Context Diagram	CD	
Diagramme de cas d'utilisation	Use Case Diagram	UCD	
Diagramme de séquence	Sequence Diagram	SD	
Diagramme d'états/transitions	State Machine Diagram	SMD	
Diagramme d'activités	Activity Diagram	AD	
Diagramme de définition de blocs	Block Definition Diagram	BDD	
Diagramme de blocs internes	Internal Block Diagram	IBD	

# 1. LE DIAGRAMME D'EXIGENCES

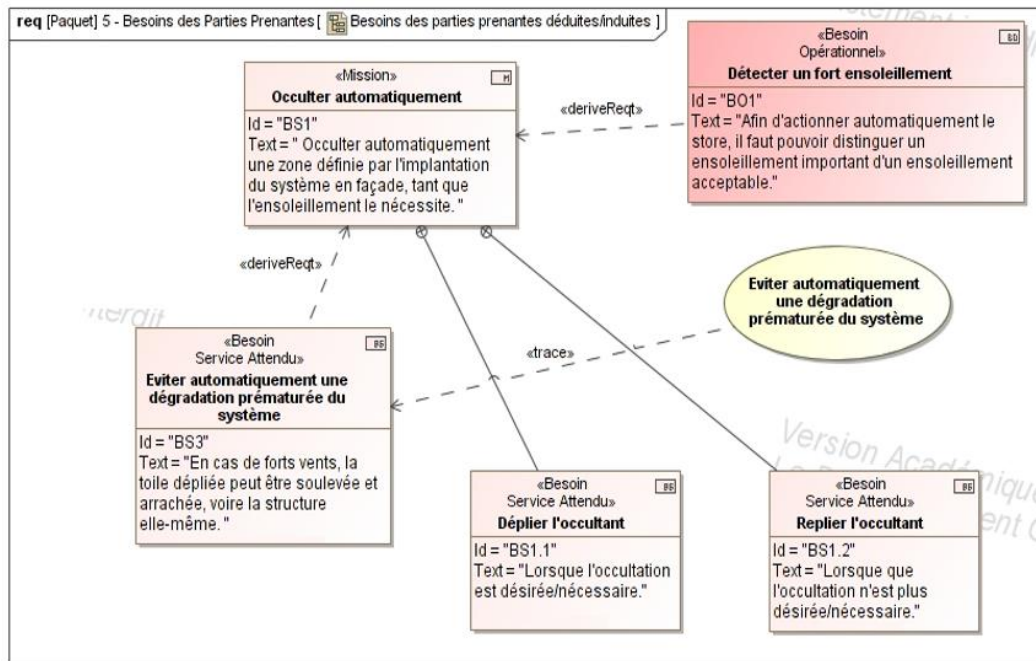


## 1.1. Finalité

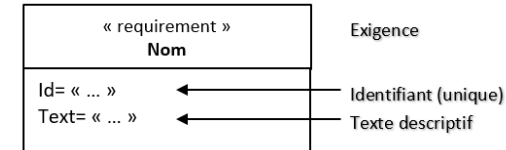
- Répertoire des besoins/exigences garantissant la bonne conception du système.
- Diagramme transversal pour les besoins comme pour les exigences systèmes, évoluant au fil de l'IS

**Remarque :** un besoin est une exigence au sens «requirement» de SysML.

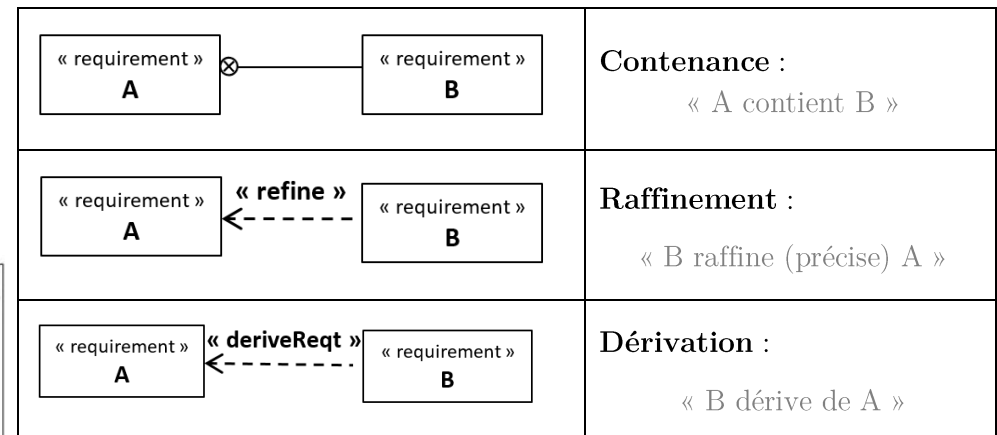
## 1.2. Vue d'ensemble



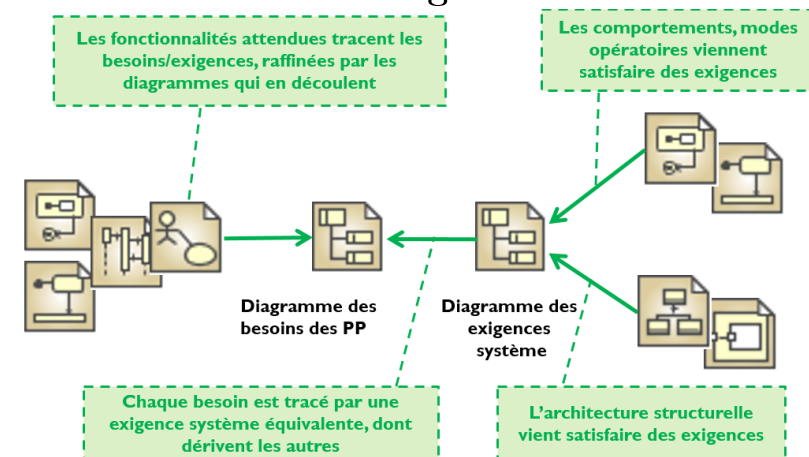
## 1.3. Exigence



## 1.4. Liens entre exigences



## 1.5. Liens avec les autres diagrammes



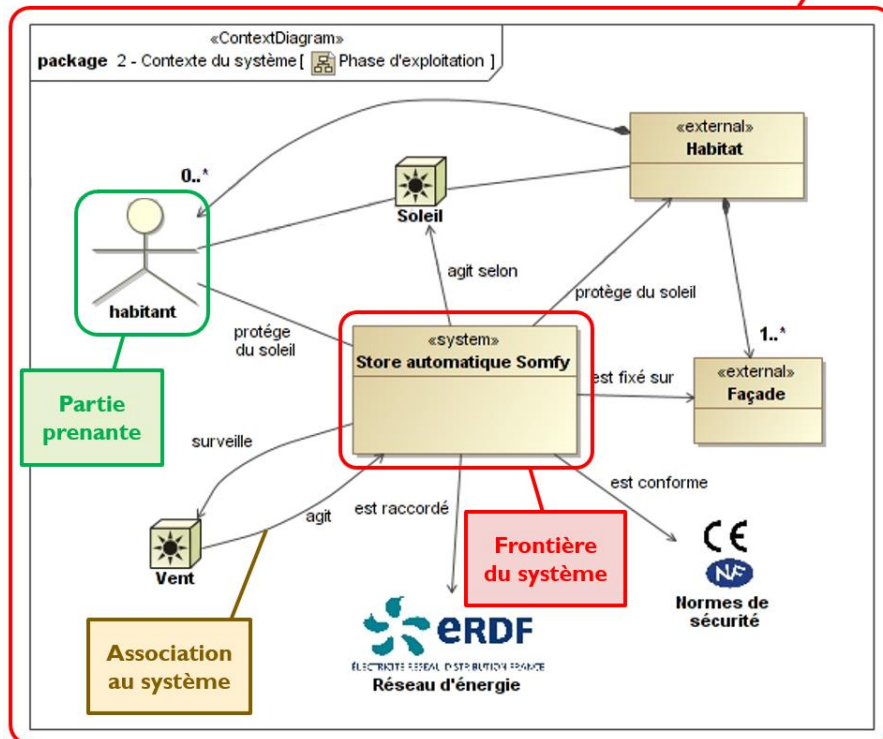
## 2. LE DIAGRAMME DE CONTEXTE



### 2.1. Finalité

- Mettre en évidence les éléments du contexte :
  - Parties prenantes
  - Eléments extérieurs
- Montrer leurs interactions avec le système
- Définir les frontières du système, du contexte

### 2.2. Vue d'ensemble



### 2.3. Représentation des interactions

Association, orientée ou non, dont la nature peut être explicitée sur le lien. Le sens de la flèche donne le sens de lecture.

### 2.4. Représentations communes des interacteurs

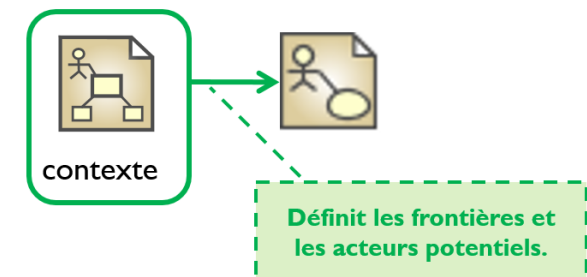
- Les interacteurs peuvent être des parties prenantes (personne physique ou morale) ou des éléments extérieurs.

	« Sticky boy » ou Stickman Personne physique
	Effet environnemental : l'environnement lui-même, ou seulement un élément physique
	Personne morale
	«external» Façade
	Elément quelconque (icône d'acteur générique ou bloc)

- Ils peuvent être parfois directement représentés en image.

### 2.5. Liens avec les autres diagrammes

- Tous les acteurs de l'UCD doivent être repris du diagramme de contexte (la réciproque n'est pas vraie)



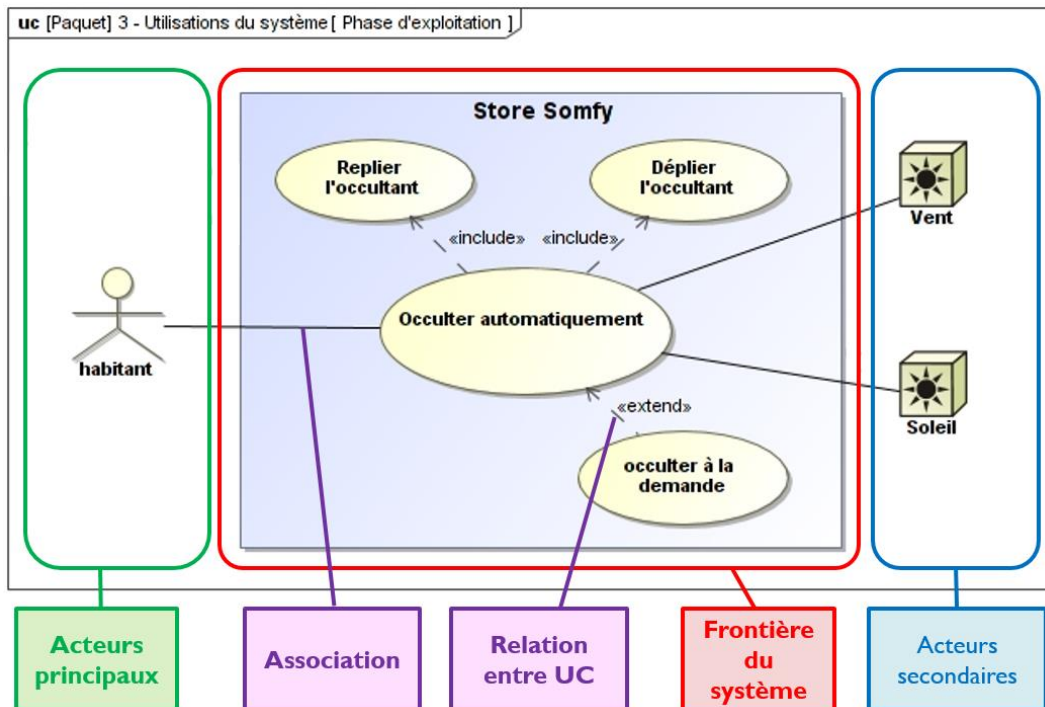
### 3. LE DIAGRAMME DE CAS D'UTILISATION



#### 3.1. Finalité

- Rendre compte des services rendus (fonctionnalités offertes) par le système ;
- Identifier les acteurs à qui ces services sont rendus (acteurs principaux, placés à gauche) ;
- Identifier les acteurs/éléments participant aux cas d'utilisation (acteurs secondaires, placés à droite).

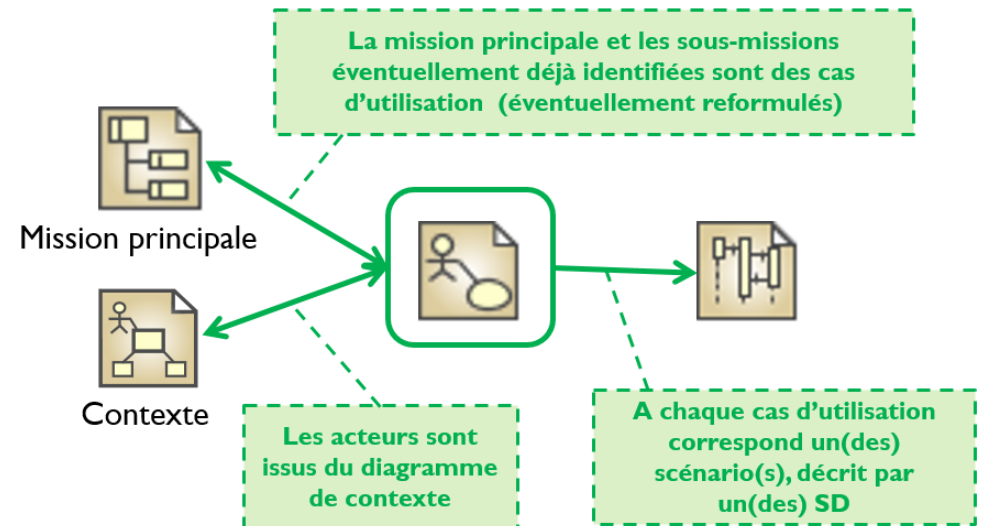
#### 3.2. Vue d'ensemble



#### 3.3. Liens entre cas d'utilisation

	<b>Inclusion</b> : le service A utilise le sous-service B « A inclus B »
	<b>Extension</b> : B est une variante de A selon une modalité différente « B étend A »
	<b>Spécialisation/Généralisation</b> : « B est une spécialisation de A » « A est une généralisation de B »

#### 3.4. Liens avec les autres diagrammes



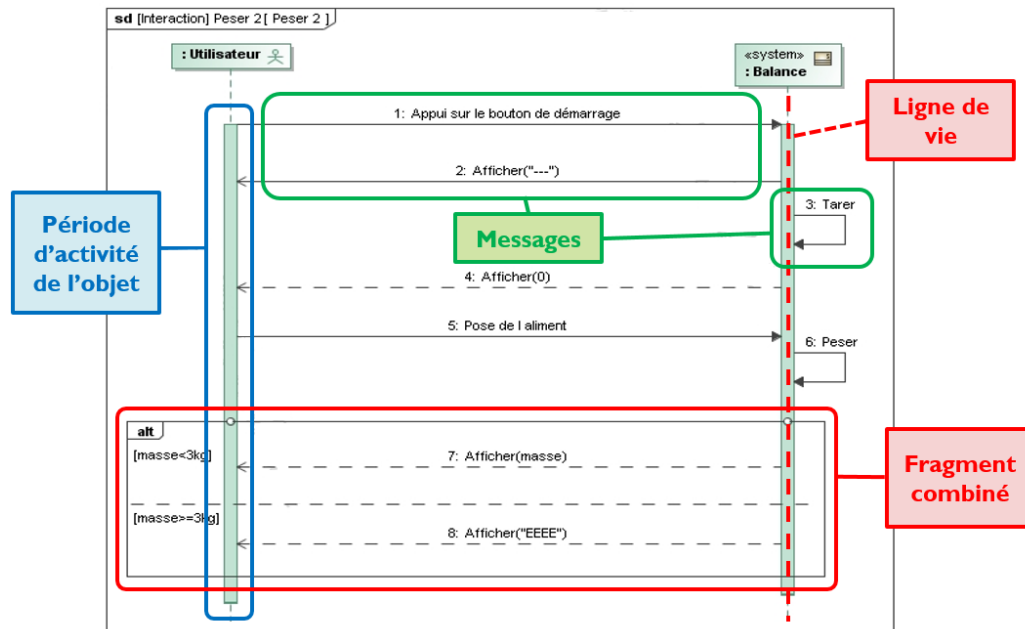
## 4. LE DIAGRAMME DE SÉQUENCE



### 4.1. Finalité

- Décrire temporellement le scénario d'un cas d'utilisation

### 4.2. Vue d'ensemble



	Message asynchrone : message n'attendant aucune réponse
	Message réflexif : Interaction interne

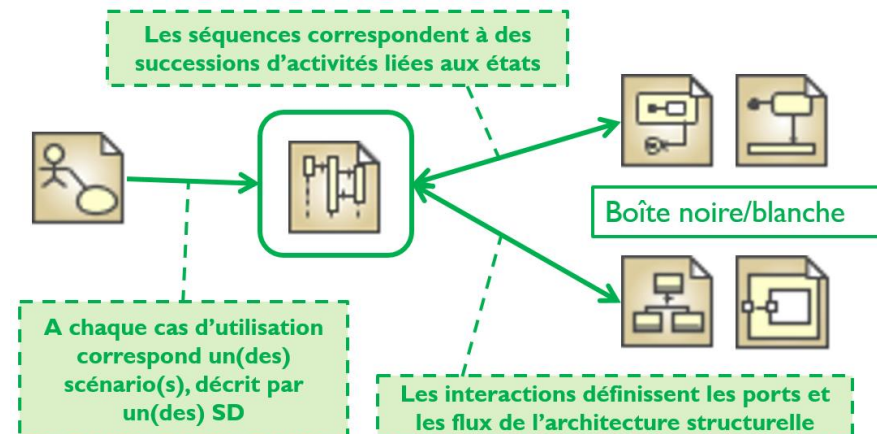
### 4.4. Fragments combinés

opt [condition]	Le fragment ne s'exécute que si la condition est vraie (if...then)
alt [condition]	opt avec un second fragment exécuté si la condition est fausse (if...then...else)
par	Les deux fragments s'exécutent simultanément (en parallèle). Peut contenir une condition.
loop [condition]	Boucle tant que condition vraie
ref	Référence à un autre diagramme de séquence.

### 4.3. Types de message

	Message synchrone : message qui demande une réponse
	Réponse (d'un message synchrone)

### 4.5. Liens avec les autres diagrammes



## 5. LE DIAGRAMME D'ÉTATS

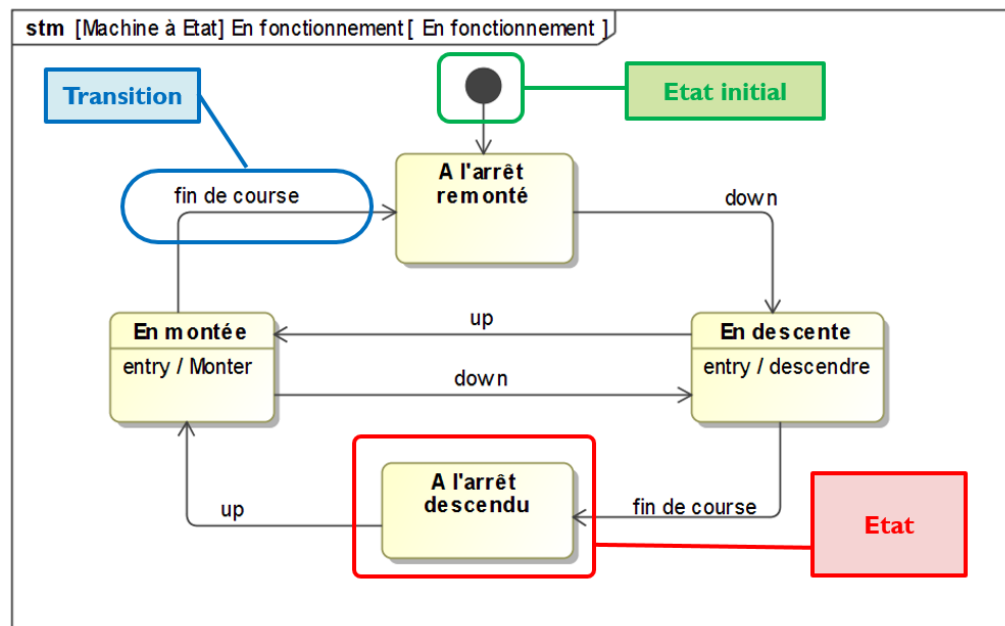


### 5.1. Finalité

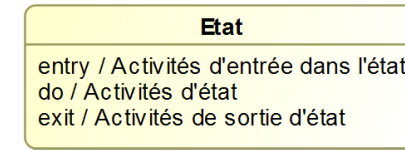
- Rendre compte des différents états possibles du système
- Rendre compte des modes opératoires régissant le passage de l'un à l'autre.

*Remarque* : les diagrammes d'états (comme les diagrammes d'activités) seront plus longuement abordés dans le cours sur les « Systèmes à événements discrets ».

### 5.2. Vue d'ensemble



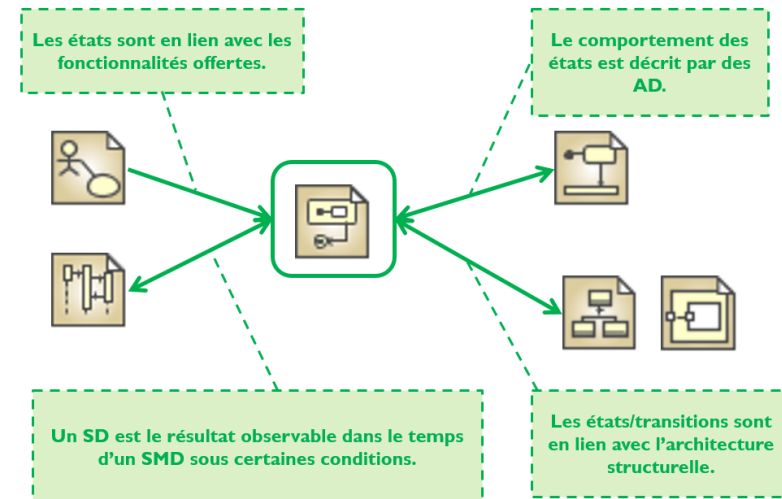
### 5.3. Activités d'état



### 5.4. Franchissements de transitions

$\xrightarrow{\text{event [condition]}}$	Quand l'évènement « event » se produit et que la « condition » est vraie.
$\xrightarrow{\text{after(durée)}}$	Après l'écoulement du temps « durée » une fois entré dans l'état.
$\xrightarrow{\hspace{2cm}}$	Franchissement inconditionnel une fois les activités d'état finies.

### 5.5. Liens avec les autres diagrammes



## 6. LE DIAGRAMME DE DÉFINITION DE BLOCS

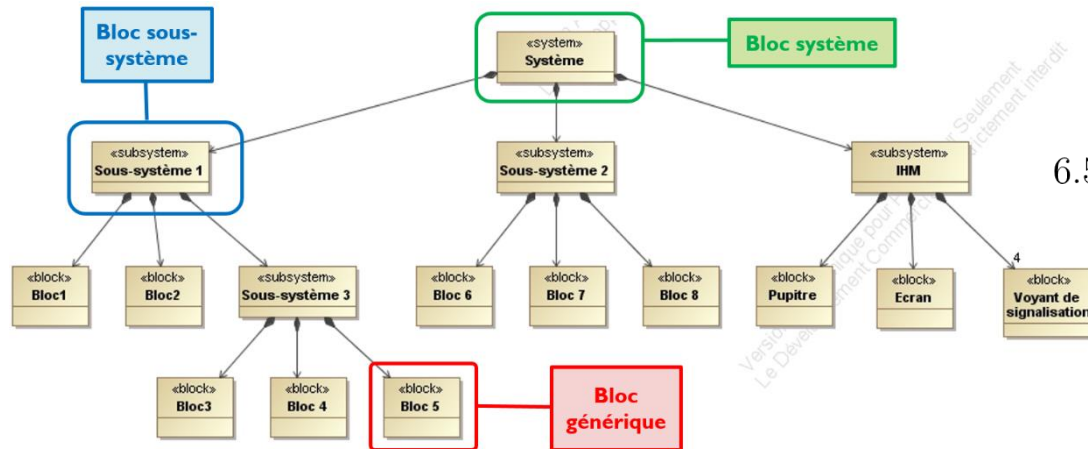


### 6.1. Finalité

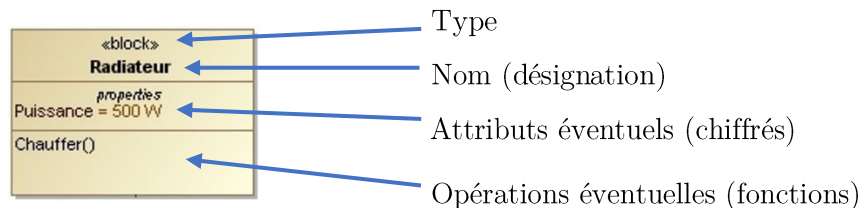
- Etablir la liste des différents constituants du système, de manière hiérarchique au besoin.

### 6.2. Vue d'ensemble

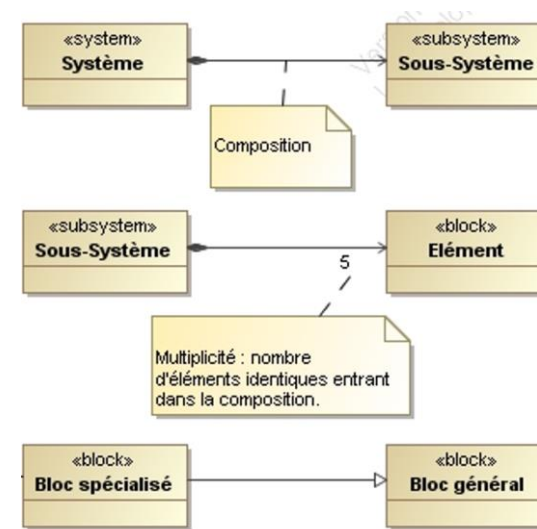
Architecture qui est souvent mise sous cette forme :



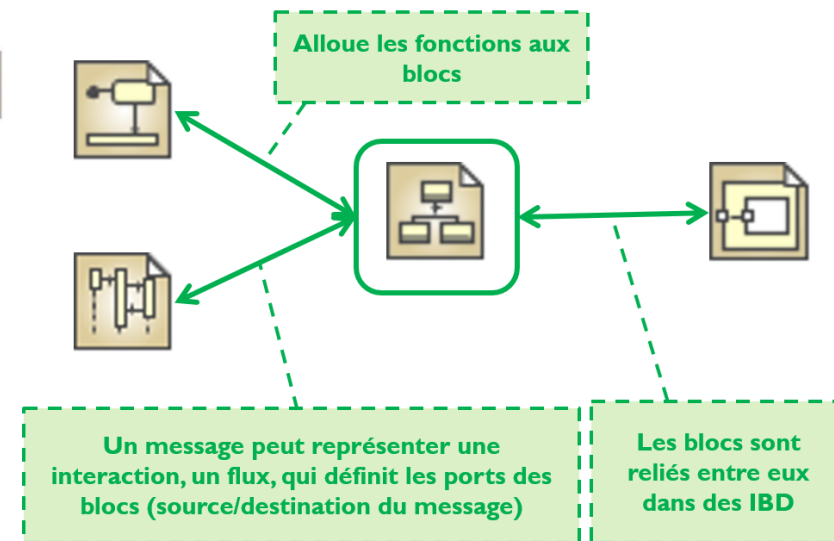
### 6.3. Paramètres d'un bloc



### 6.4. Liens entre blocs



### 6.5. Liens avec les autres diagrammes



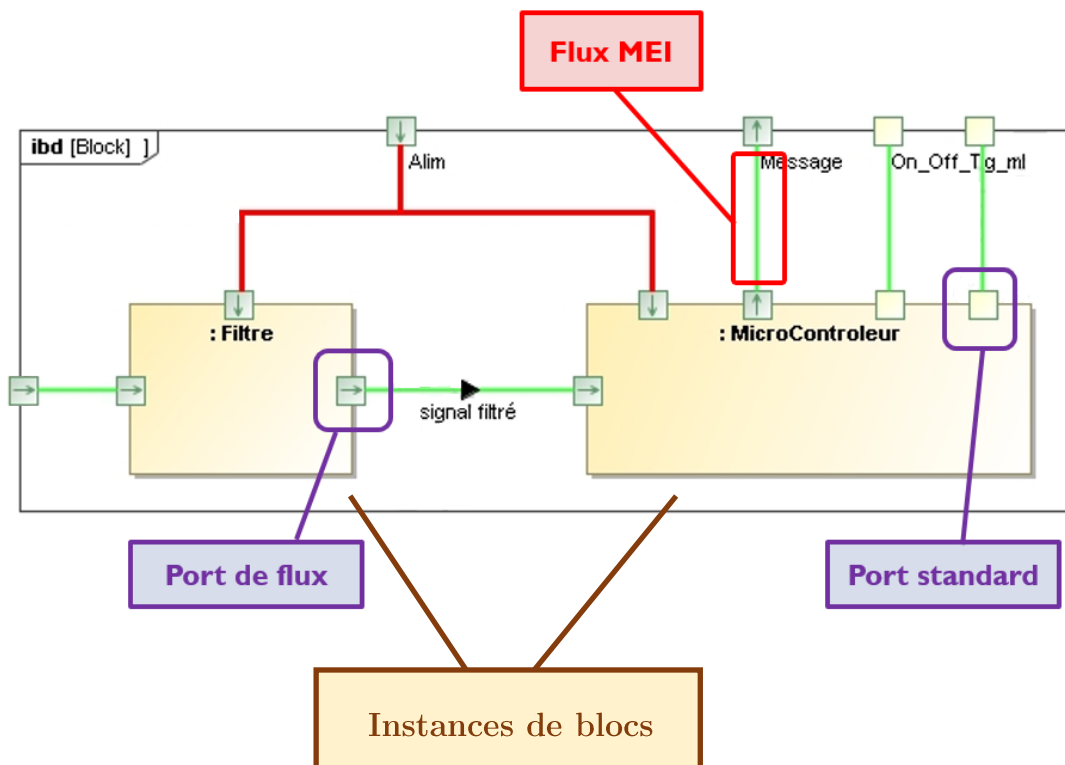
## 7. LE DIAGRAMME DE BLOCS INTERNES



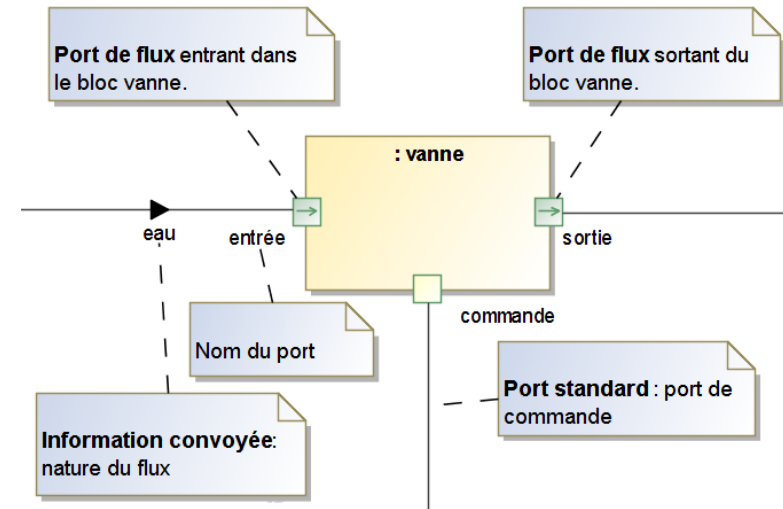
### 7.1.Finalité

- Rendre compte des flux MEI :
  - internes entre les constituants du système ;
  - externes entre le système et son environnement.

### 7.2.Vue d'ensemble



### 7.3.Flux / Ports



### 7.4.Liens avec les autres diagrammes

