

Analyse Systémique et Fonctionnelle d'un système

Le langage SYSML

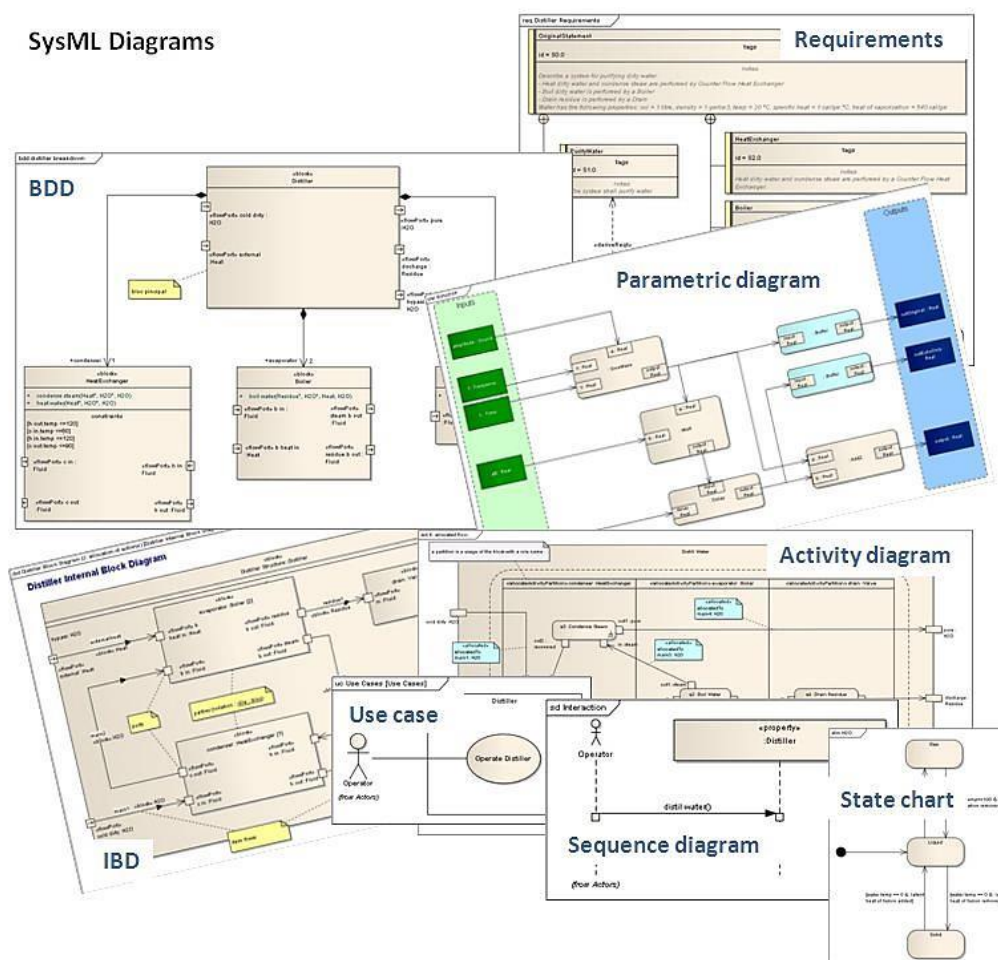


Table des matières

1. Les différents diagrammes du langage SysML.....	2
2. Le diagramme d'exigence (Requirement diagram, notation SysML : req).....	2
2.1. Son rôle :.....	2
2.2. Représentation graphique.....	3
2.3. Relations au sein du diagramme des exigences.....	3
2.4. Conseils.....	4
3. Diagramme de cas d'utilisation (use case diagram, notation SysML : uc).....	4
3.1. Son rôle.....	4
3.2. Représentation graphique.....	4
3.3. Relations au sein du diagramme.....	5
3.4. Conseils.....	5
4. Diagrammes de définition de bloc (Notation SysML : bdd).....	5
4.1. Rôles.....	5
4.2. Représentation graphique.....	5
4.3. Relation au sein du diagramme.....	6
4.4. Conseils.....	6
5. Diagramme de bloc interne (internal block diagram, notation SysML : ibd).....	6
5.1. Rôle.....	6
5.2. Représentation graphique.....	7
5.3. Relation au sein du diagramme.....	7
5.4. Conseils.....	7
6. Le diagramme de séquence (sequence diagram, notation SysML : sd).....	8
6.1. Rôle.....	8
6.2. Représentation graphique.....	8
6.3. Les liens et conditions dans le diagramme.....	9
7. Le diagramme d'état (state machine diagram, notation SysML : stm).....	9
7.1. Rôle.....	9
7.2. Représentation graphique.....	9
7.3. Symbolique du diagramme.....	10
8. Synthèse.....	10

Source : http://www4.ac-nancy-metz.fr/cpge-pmf-epinal/Cours_TD_SII/Modelisation_fonctionnelle_systeme.pdf



Le langage SysML (Systems Modeling Language) est fait pour :

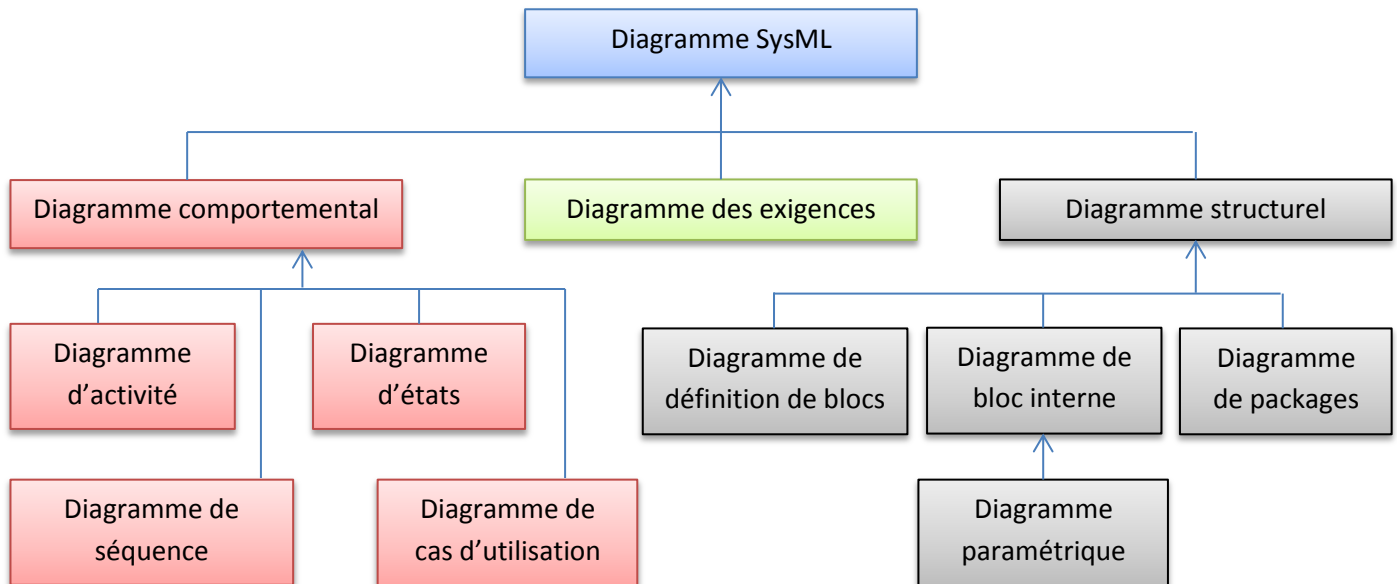
- Analyser la structure et le fonctionnement des systèmes.
- Décrire et concevoir les systèmes.
- Vérifier et valider la faisabilité d'un système avant sa réalisation.

Le langage SysML permet de «représenter» et «communiquer» sous forme de blocs ou diagrammes les divers aspects d'un système :

- Diagrammes structurels : représentation statique du système (définition des matériels mis en œuvre).
- Diagrammes comportementaux : représentation dynamique du système (cycle de fonctionnement de la machine).
- Diagrammes d'exigences : mettent en évidence l'expression des besoins liés au système et à ses fonctionnalités.

Ces diagrammes ne sont pas indépendants les uns des autres. Les données qui les constituent sont des objets informatiques munis de propriétés cohérentes, aussi ce langage ne prend toute sa signification qu'avec un support informatique qui assure la cohérence entre les différentes données introduites dans les diagrammes. On peut ainsi obtenir un système virtuel testable

1. Les différents diagrammes du langage SysML



SysML est aussi une méthode de conception, les diagrammes sont donc ordonnés. La première action consiste à traduire le besoin fonctionnel en exigences en s'appuyant sur le cahier des charges. En sciences de l'ingénieur on effectuera des analyses de systèmes, on comparera donc le système modélisé par le langage SysML avec le système réel.

2. Le diagramme d'exigence (Requirement diagram, notation SysML : req)

2.1. Son rôle :

Il décrit les exigences du cahier des charges fonctionnel. 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...

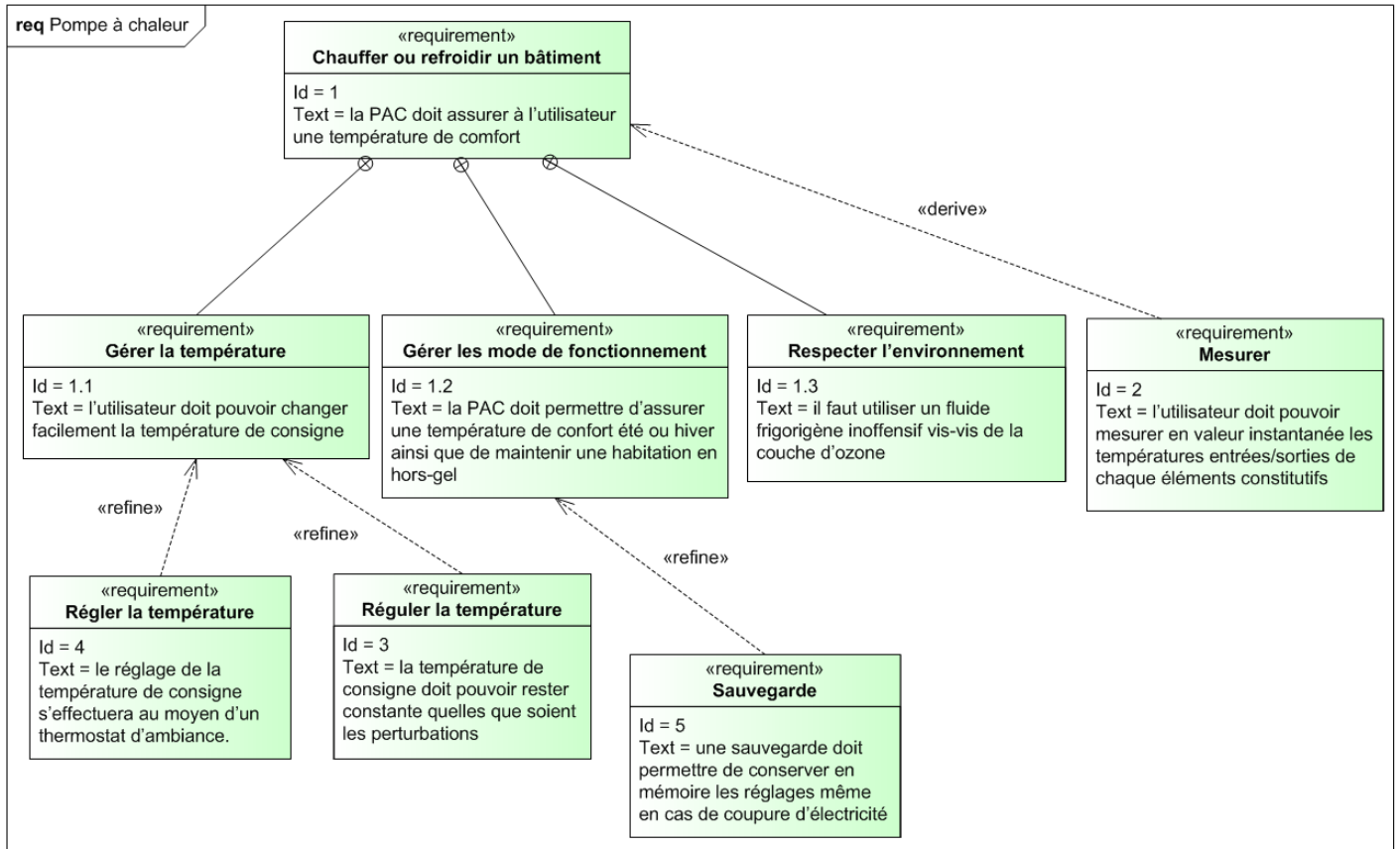
Les exigences servent à établir un contrat entre le client et les réalisateurs du futur système.



2.2. Représentation graphique

- 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 ajouter des données quantitatives et des précisions.

Exemple sur une Pompe à Chaleur



2.3. Relations au sein du diagramme des exigences

Lien de contenance	Lien de dérivation "derive" ou "deriveReqt"	Lien de raffinement "refine"	Lien de solution "satisfy"
L'exigence située du côté du cercle avec la croix, contient l'exigence (plus détaillée) placée à l'autre extrémité.	Il exprime un lien entre 2 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.	Il précise, souvent par des données quantitatives, l'exigence pointée par la flèche.	Il précise le composant satisfaisant l'exigence pointée par la flèche.



2.4. Conseils

- Ne pas chercher à poser toutes les exigences pour rester lisible.
- Réaliser plusieurs diagrammes d'exigences si nécessaire, regrouper les exigences techniques sur un seul diagramme par exemple, puis les autres groupes d'exigences sur d'autres diagrammes.

3. Diagramme de cas d'utilisation (use case diagram, notation SysML : uc)

3.1. Son rôle

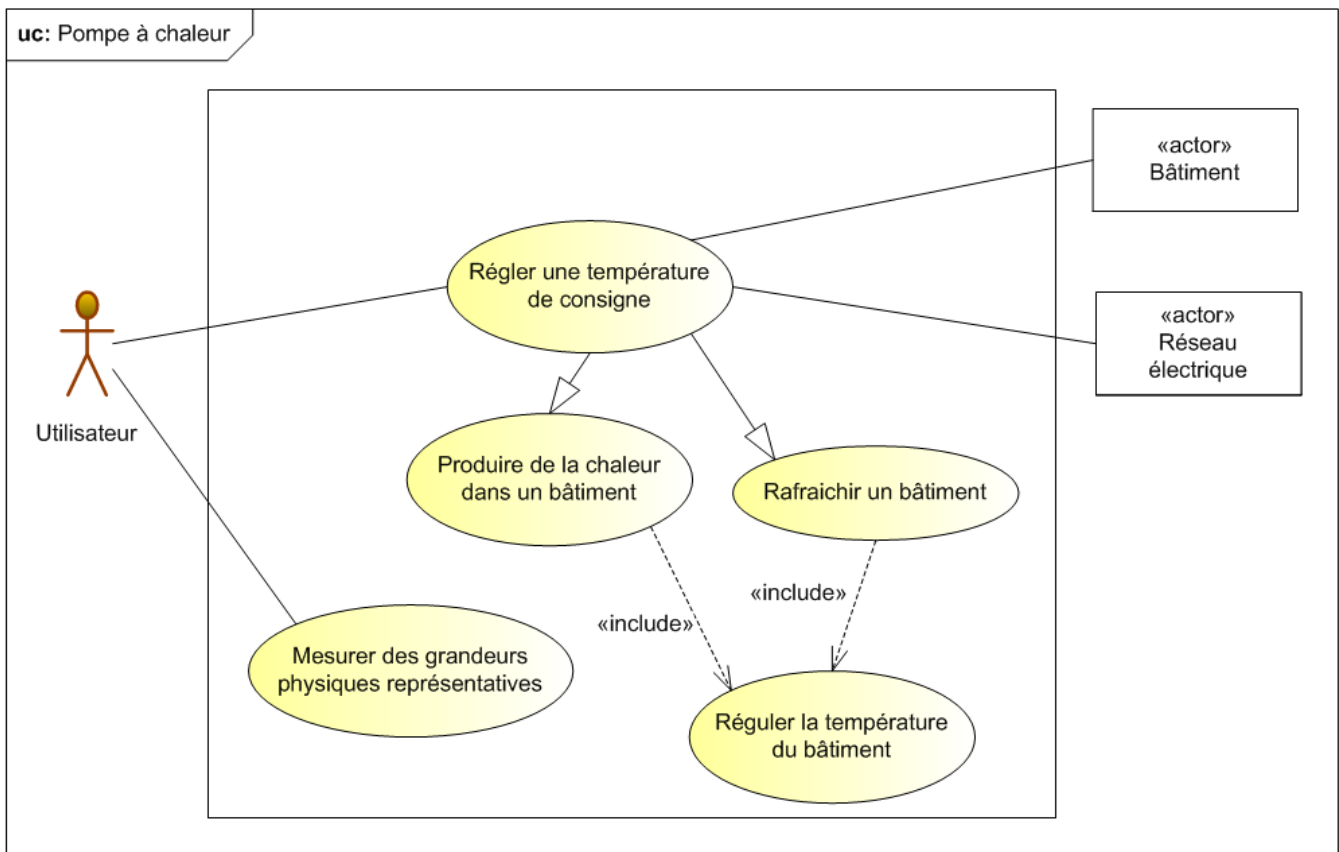
Il montre les interactions fonctionnelles des acteurs et du système d'étude. Il délimite précisément le système, décrit ce que fera le système sans spécifier comment (et non ce que fera l'utilisateur).

Il exprime les services (use cases) offerts par le système aux utilisateurs (actors).

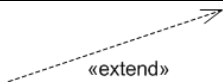
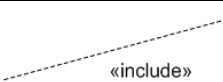

3.2. Représentation graphique

- On trace un cadre délimitant le système et contenant un ensemble de séquences d'actions (Elles peuvent aussi être liées entre elles).
- À gauche on place les acteurs humains, à droite les acteurs non humains (un acteur non humain est représenté par un rectangle ou cube).
- 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).
- 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 sur une pompe à chaleur



3.3. Relations au sein du diagramme

Lien d'extension "extend"	Lien d'inclusion "include"	Lien de spécialisation/généralisation
		
Le cas d'utilisation de base peut (option) incorporer celui placé à l'origine de la flèche.	Le cas d'utilisation de base incorpore systématiquement celui placé à l'extrémité de la flèche.	Ils relient des cas d'utilisation descendants qui héritent de la description d'un cas d'utilisation supérieur (parent commun).

3.4. Conseils

- Ce diagramme ne doit indiquer ni la manière dont il va assurer les services, ni les solutions technologiques envisagées.
- Plusieurs diagrammes d'utilisations peuvent être établis pour un système afin d'en améliorer la compréhension.

4. Diagrammes de définition de bloc (Notation SysML : bdd)

4.1. Rôles

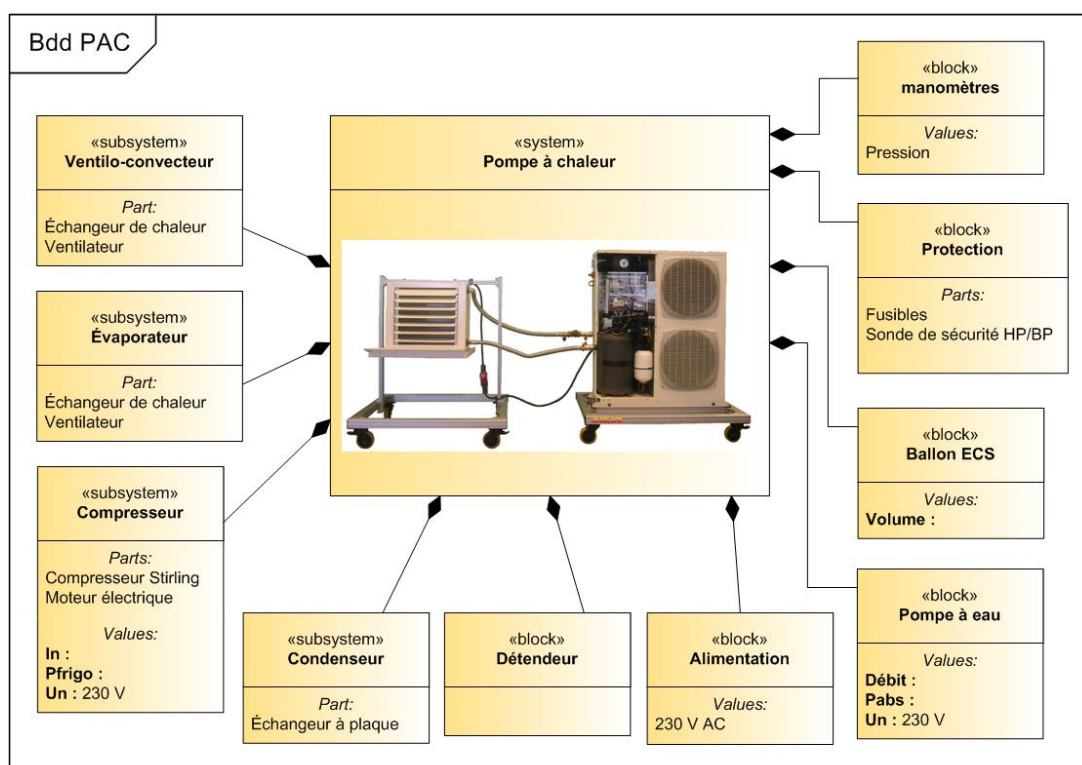
Bdd du contexte : Il définit le système dans son environnement (éléments humains et matériels qui interagissent avec lui).

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

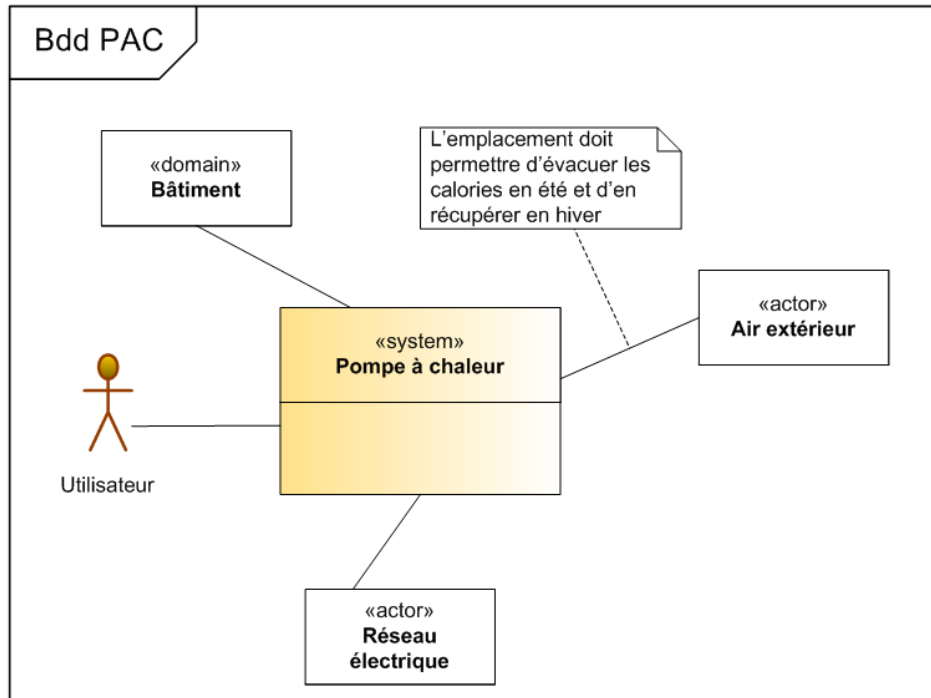
4.2. Représentation graphique

Le bloc SysML (« block ») constitue la brique de base pour la modélisation de la structure d'un système. Ce bloc peut représenter un système complet, un sous-système ou un composant élémentaire.

Exemple d'un bdd système sur une pompe à chaleur



Exemple d'un bdd contexte sur une pompe à chaleur



4.3. Relation au sein du diagramme

Lien de spécialisation/généralisation	Relation d'association	Relation de composition	Relation d'agrégation
Ils relient des cas d'utilisation descendants qui héritent de la description d'un cas d'utilisation supérieur (parent commun).	Ce simple trait exprime un lien d'égal à égal qui permet souvent de relier le contexte au système étudié.	Le bloc situé du côté du losange plein a besoin d'un sous-bloc coté flèche. Cela veut dire que le sous-système se décompose de nouveau en blocs. Sa lecture : " ... possède un ".	Relation similaire au lien de composition mais qui revêt un caractère optionnel. Sa lecture : "... peut avoir un"

4.4. Conseils

- Un BDD ne décrit ni les fonctions, ni le comportement du système.
- Ce diagramme est utile pour montrer les grosses briques du système.
- Il n'est pas obligatoire de faire apparaître les propriétés et les opérations dans chaque bloc, ainsi il offre d'un coup d'œil la structure du système.

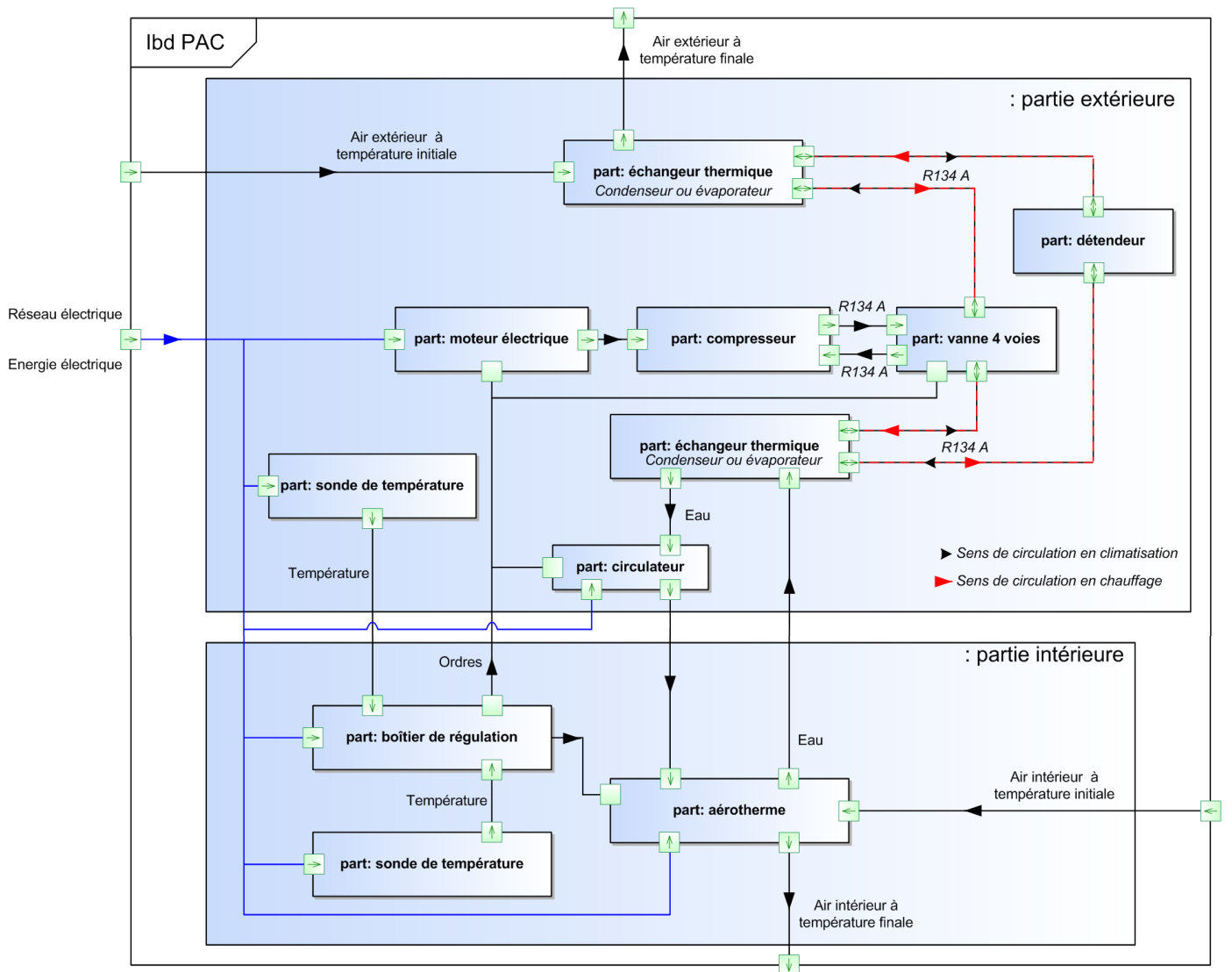
5. Diagramme de bloc interne (internal block diagram, notation SysML : ibd)

5.1. Rôle

Il décrit la structure interne d'un bloc issu du BDD, c'est-à-dire ses composants et les échanges (flux de matière, d'énergie ou d'information) entre les blocs internes.



5.2.Représentation graphique



5.3.Relation au sein du diagramme

Le port de flux	Le port standard	Le connecteur
Cette interface autorise l'entrée et/ou la sortie du flux (matière, énergie, information) vis-à-vis d'un block. Elle possède au moins un sens, son nom est facultatif.	Représente une interface qui n'est pas liée à un flux mais à un service, une opération, une consigne ou un ordre de commande.	Ce lien relie deux ports. En cas de flux, sa nature peut être précisée.

5.4.Conseils

- Il faut bien retenir que les liens se représentent entre blocs de même niveau, ils ne se contiennent pas.
- 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



6. Le diagramme de séquence (sequence diagram, notation SysML : sd)

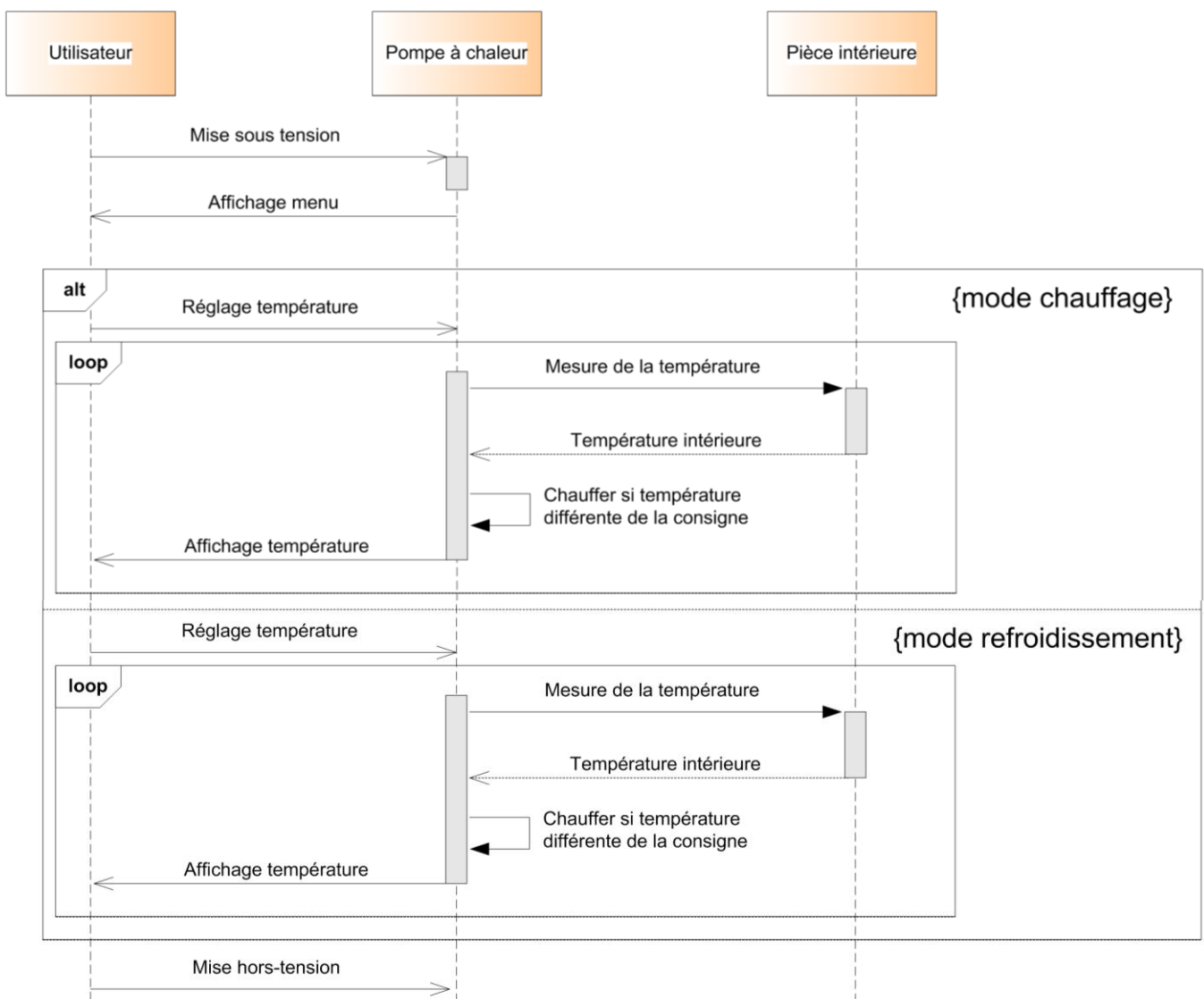
6.1.Rôle

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. En règle générale, on en écrira donc plusieurs. C'est un diagramme comportemental.

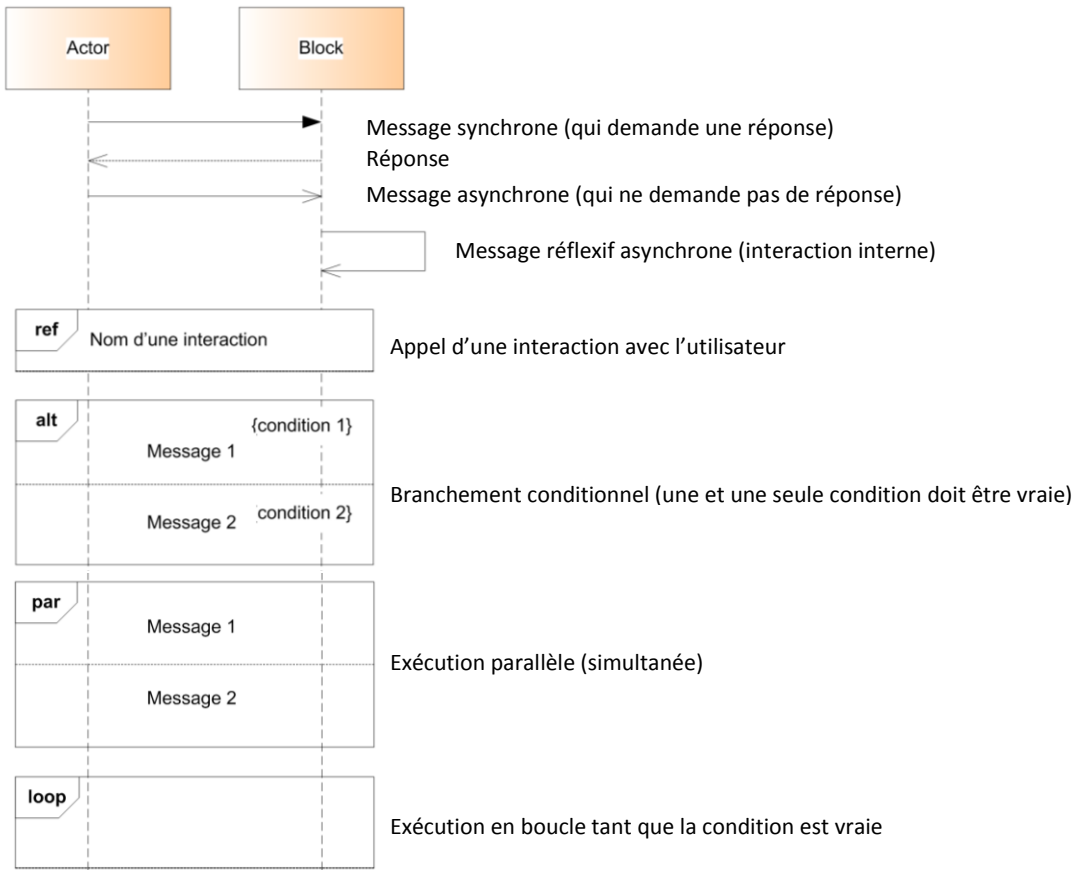
6.2.Représentation graphique

Chaque élément situé dans le haut du diagramme est un objet ou constituant pris comme boîte noire.

Les lignes verticales en pointillés sont des lignes de vie. Elles représentent le temps sans qu'il y ait une échelle. Les rectangles superposés dessus désignent l'activité de l'objet.



6.3. Les liens et conditions dans le diagramme

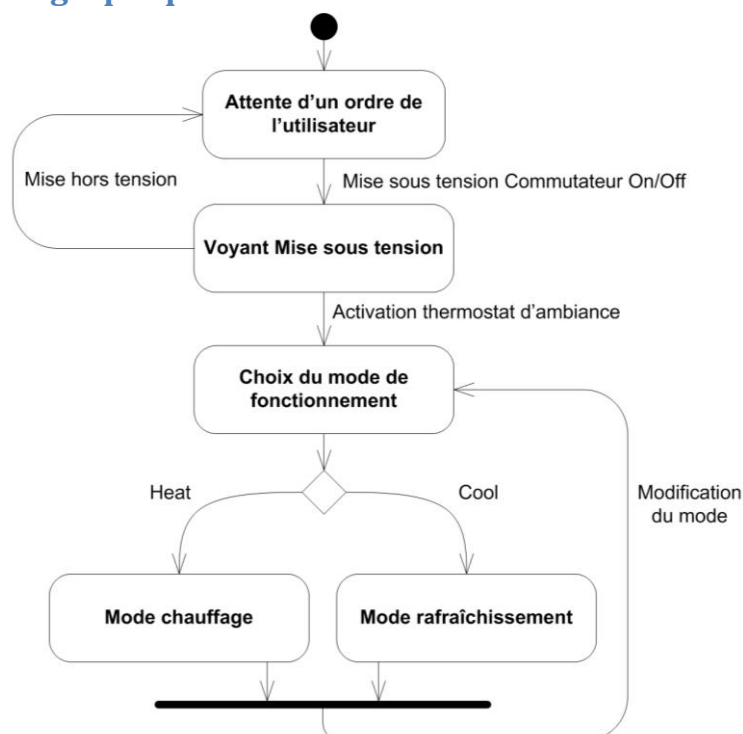


7. Le diagramme d'état (state machine diagram, notation SysML : stm)

7.1. Rôle

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

7.2. Représentation graphique



7.3.Symbolique du diagramme

● v	●	○	trigger[guard]/activity →
Représente le moment initial.	Représente le moment final.	Représente un moment spécifique du comportement d'un objet qui correspond à une séquence = entrée → phase active (ou attente) → sortie	Représentée par une flèche, elle constitue en changement d'état 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.

8. Synthèse

