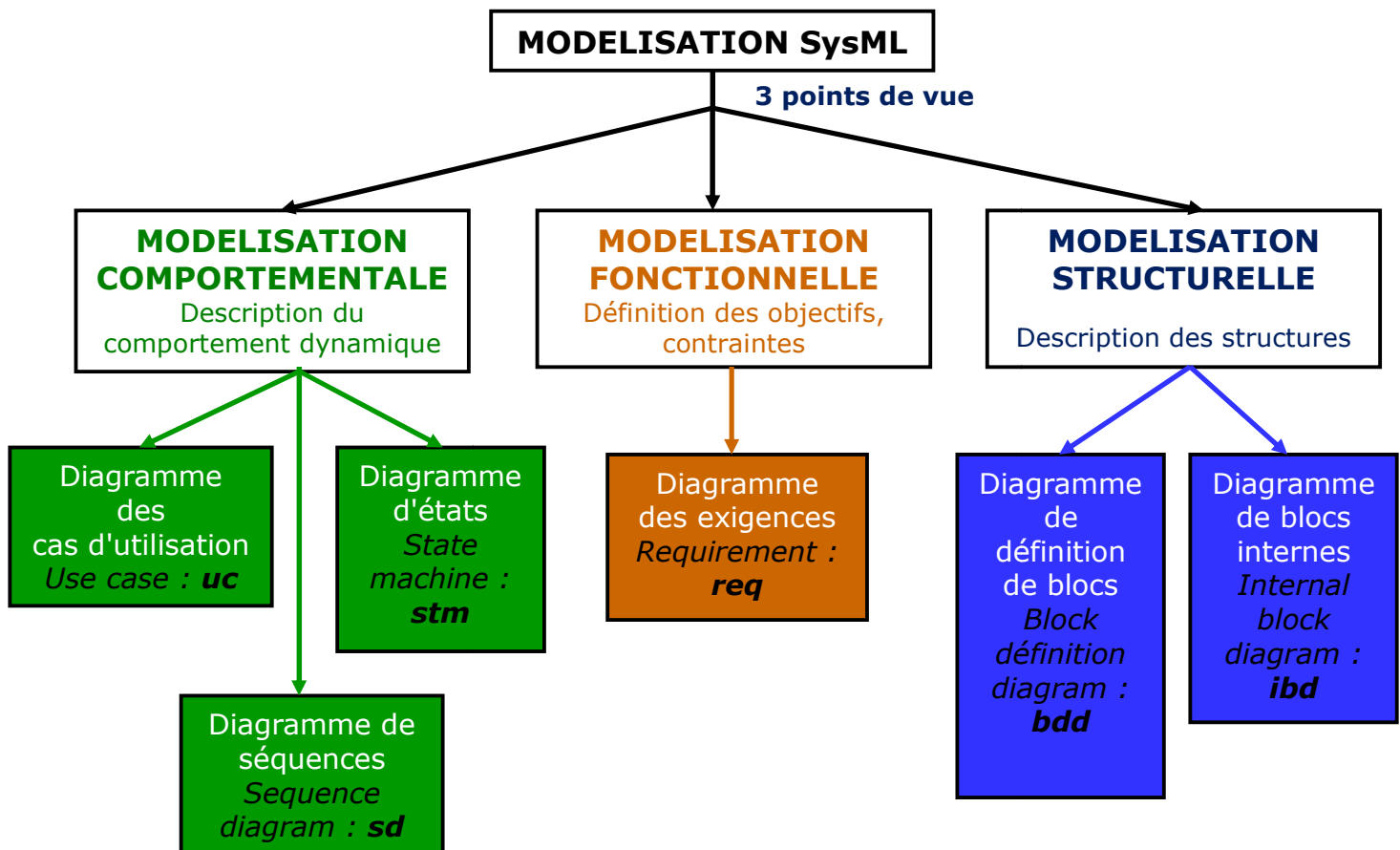


1 – PRESENTATION

Le langage SysML (Systems Modeling Language) est un outil de représentations abstraites (modèles) utilisé pour développer ou décrire des systèmes complexes.

Il permet d'utiliser un langage commun à tous les concepteurs et utilisateurs d'un système (ingénieur, technicien, service marketing, client etc....)

Il comporte 9 diagrammes mais seulement 6 diagrammes seront utilisés dans le cadre STI2D. Ils permettront de structurer la description selon 3 points de vu.



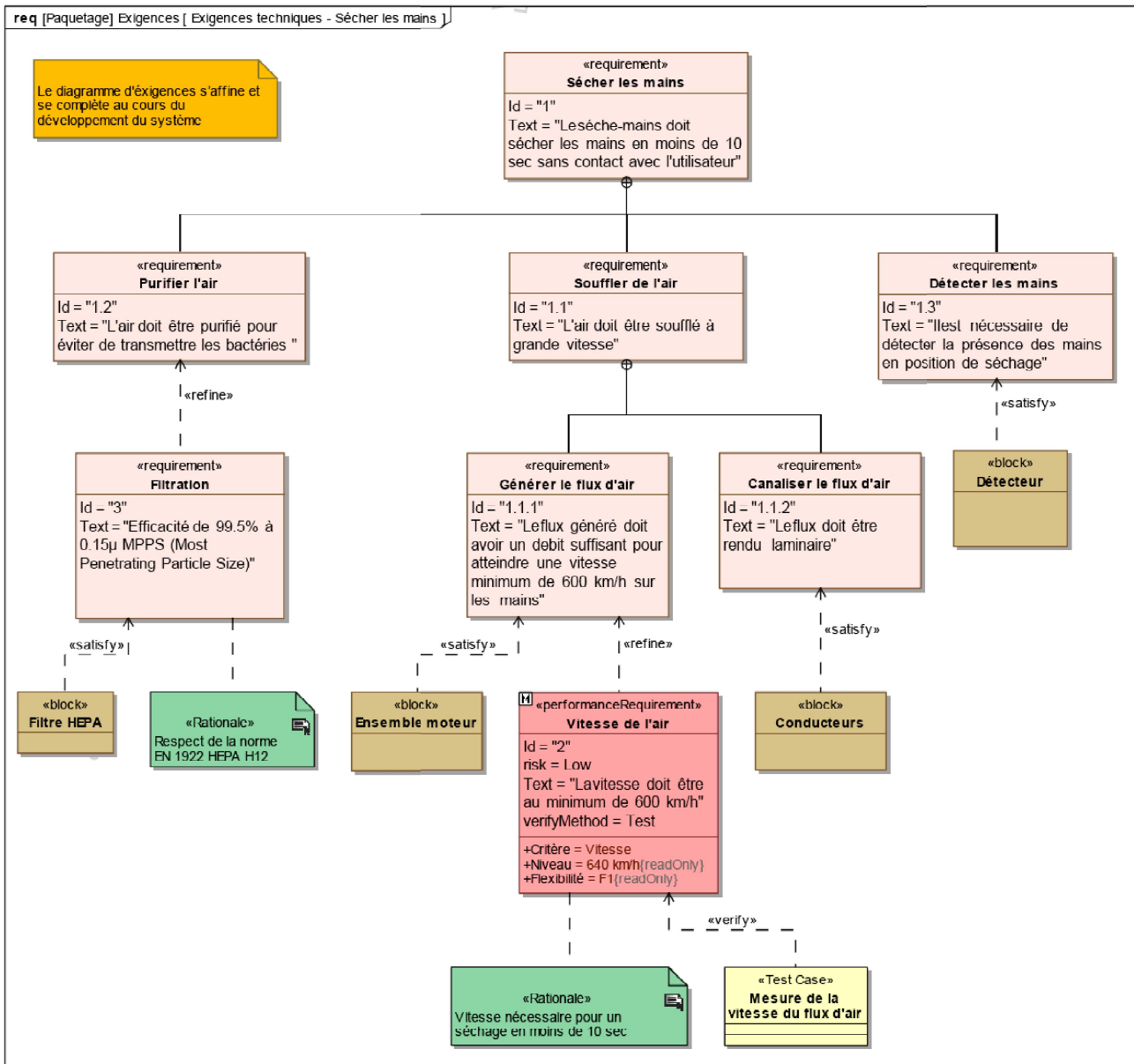
Pour découvrir ces différents diagrammes, nous prendrons pour support le sèche-mains Dyson « airblade ».

Il a pour particularité d'être plus rapide, plus hygiénique et beaucoup moins énergivore que les autres sèche-mains.

A titre d'exemple :

	coût moyen annuel pour une utilisation type (en €)	Empreinte carbone en g CO ²
Dyson	33	0,94
Sèche-main air chaud traditionnel	161	----
Sèche-mains papier	1460	12,5

2 - DIAGRAMME DES EXIGENCES (REQ)



STI2D	Le langage SysML	
Option EE et SIN Terminale		

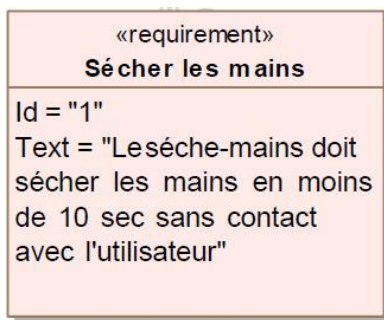
2.1 Ce diagramme permet de :

- répertorier et analyser les contraintes et les performances du système
- structurer les besoins
- faciliter la communication entre concepteurs et clients

Une exigence (Requirement, req en abrégé) 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, de fiabilité, de sécurité, etc.

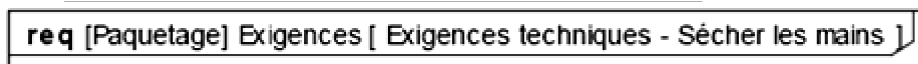
Les exigences servent à établir un contrat entre le client et les réalisateurs du futur système.



Les deux propriétés de base d'une exigence sont :

- **un identifiant unique** (permettant ensuite de gérer la traçabilité avec l'architecture, etc.) ;
- **un texte descriptif.**


2.2 Le cartouche de diagramme :



Un cartouche positionné en haut à gauche du diagramme dans un pentagone permet de spécifier le type de diagramme SysML, le type de l'élément concerné, l'élément concerné, et le nom du diagramme.

2.3 Les liaisons


Les exigences peuvent être reliées entre elles par des relations de :

- **contenance**  qui permet de décomposer une exigence composite en plusieurs exigences unitaires.
- **raffinement** « **refine** » qui consiste en l'ajout de précisions, par exemple de données quantitatives ;
- **dérivation** « **deriveReq** » qui consiste à relier des exigences de niveaux différents par exemple des exigences système à des exigences de niveau sous-système, etc.

2.4 Les notes graphiques (commentaires)

Deux mots-clés particuliers ont été définis afin de les représenter :

- des problèmes à résoudre « **problem** » ;
- des justificatifs « **rationale** ».

STI2D	Le langage SysML	
Option EE et SIN Terminale		

2.5 La traçabilité

Elle permet de s'assurer de la cohérence entre ce que fait réellement le projet et ce qu'il doit faire et de faciliter l'analyse d'impact en cas de changement.

Le diagramme d'exigences permet ainsi tout au long d'un projet de relier les exigences avec d'autres types d'élément SysML par plusieurs relations :

- **exigence élément comportemental « refine »** (cas d'utilisation, diagramme d'états, etc.);
- **exigence bloc d'architecture « satisfy »** (lien avec l'élément qui satisfait l'exigence)
- **exigence cas de test « verify »**. Un cas de test représente une méthode de vérification de la satisfaction d'une exigence. Il est représenté en SysML par un rectangle avec le mot-clé « **Test Case** ».

2.6 Application

Quelles sont les trois exigences contenues dans l'exigence 1 "Sécher les mains" ?

Quel est l'identifiant de l'exigence "Canaliser le flux d'air" ?

Quelle est l'exigence qui précise (refine) l'exigence "Générer le flux d'air" ?

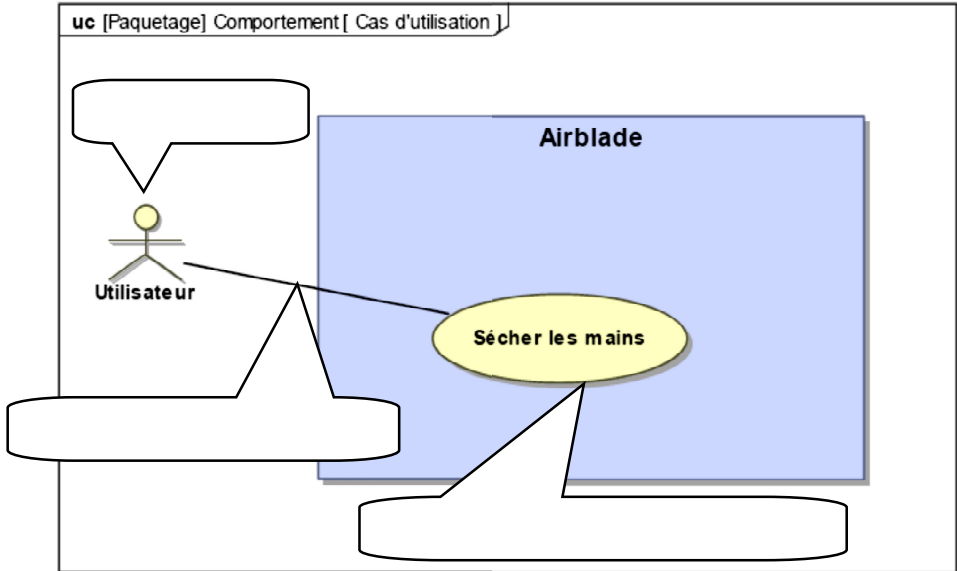
Qu'est-ce qui justifie l'exigence de filtration ?

Quel est le bloc qui satisfait (réalise) la fonction " détecter les mains" ?

Quelle est la vitesse minimum du flux d'air sur les mains ?

Quel est le temps de séchage maximum des mains ?

3 – DIAGRAMME DES CAS D'UTILISATION (UC)



C'est une représentation des fonctionnalités du système.

Un **cas d'utilisation (use case, ou uc)**

représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

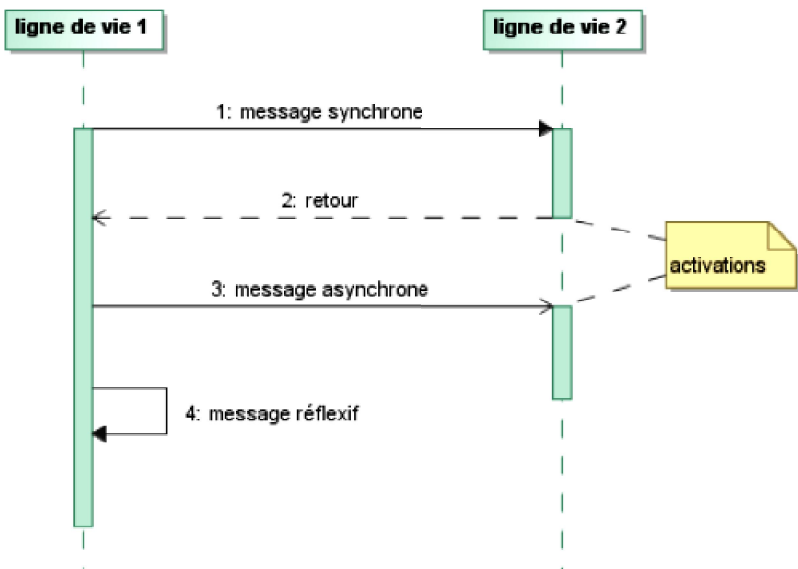
Dans notre exemple l'acteur est un être humain et il est relié par une association au cas d'utilisation "Sécher les mains". D'autres acteurs peuvent être des systèmes techniques qui interagissent avec le système étudié.

Nous étudierons dans un TD les autres possibilités offertes par ce diagramme.

4 – DIAGRAMME DE SEQUENCE (SD)

Le diagramme de séquence décrit la chronologie des échanges issus d'un cas d'utilisation entre les acteurs et les objets.

4.1 Notation de bases : Il se lit de haut en bas.



Ligne de vie : ligne verticale en pointillée représentant un élément participant au diagramme de séquence. Elle possède un nom et un type.

Message : élément de communication unidirectionnel entre lignes de vie qui déclenche une activité dans le destinataire. La réception d'un message provoque un événement chez le récepteur.

La flèche pointillée représente un retour.

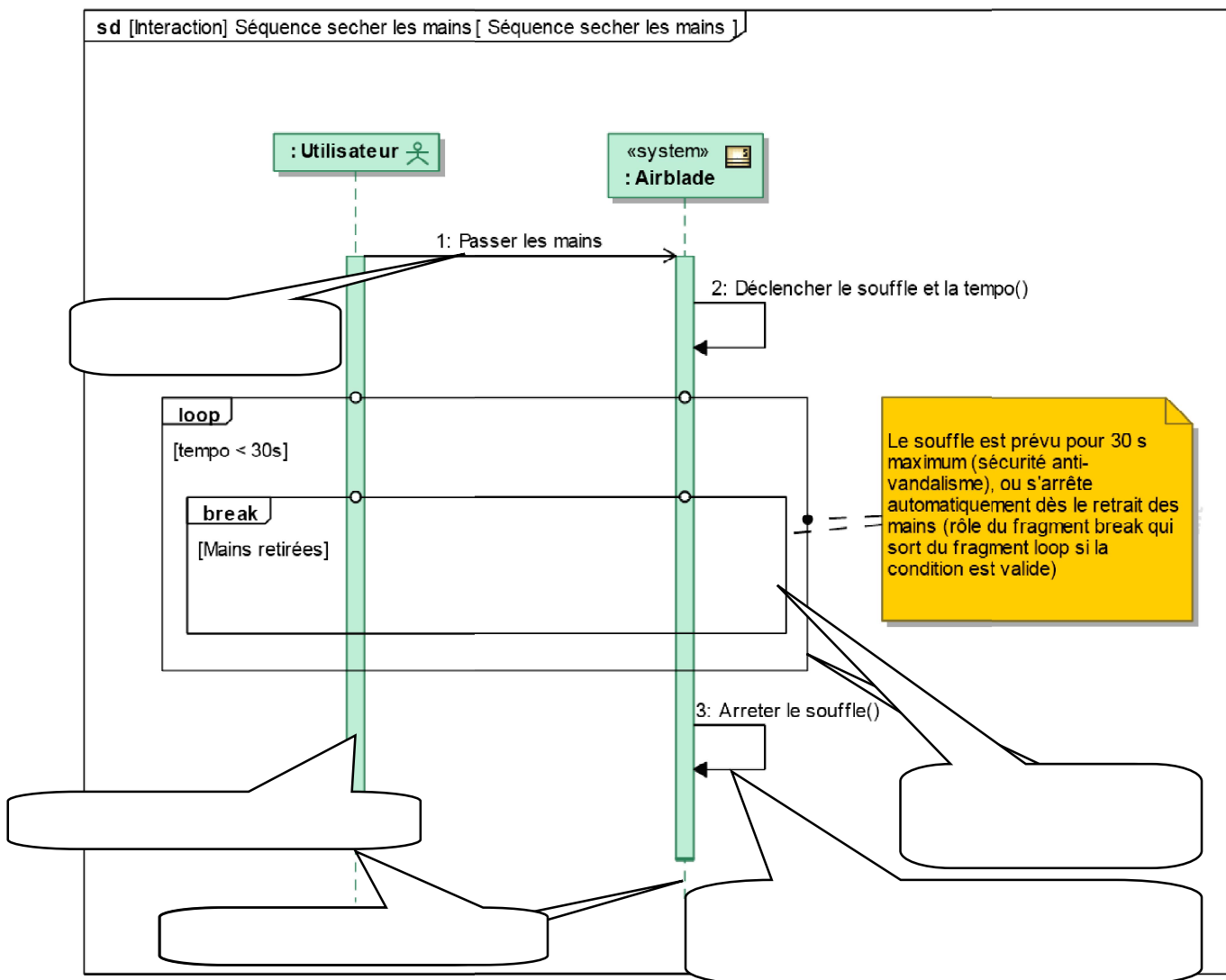
Cela signifie que le message en question est le résultat direct du message précédent. Un message synchrone (émetteur bloqué en attente de réponse) est représenté par une flèche pleine →, alors qu'un message asynchrone est représenté par une flèche évidée →.

La flèche qui boucle (**message réflexif**) permet de représenter un **comportement interne**.

Activation

Les bandes verticales le long d'une ligne de vie représentent des périodes d'activation. Elles sont optionnelles, mais permettent de mieux comprendre la flèche pointillée du message de retour.

4.2 Cas du sèche-mains



Les échanges de messages sont ici extrêmement simples puisqu'il n'y a qu'un message "**Passer les mains**"

Par contre ce diagramme fait apparaître les notions de **fragments combinés**.

Chaque fragment possède un opérateur et peut être divisé en opérandes.

Ici, un **fragment "Loop"** qui signifie que l'**action** du souffle se **déroule en boucle** tant que la **condition entre crochets (tempo<30s)** est vraie.

Si pendant cette période, les mains sont retirées [**Mains retirées**], un **fragment "break"** sort de la boucle pour aller vers l'opération interne "Arrêter le souffle".

5- DIAGRAMME D'ETATS (STM)

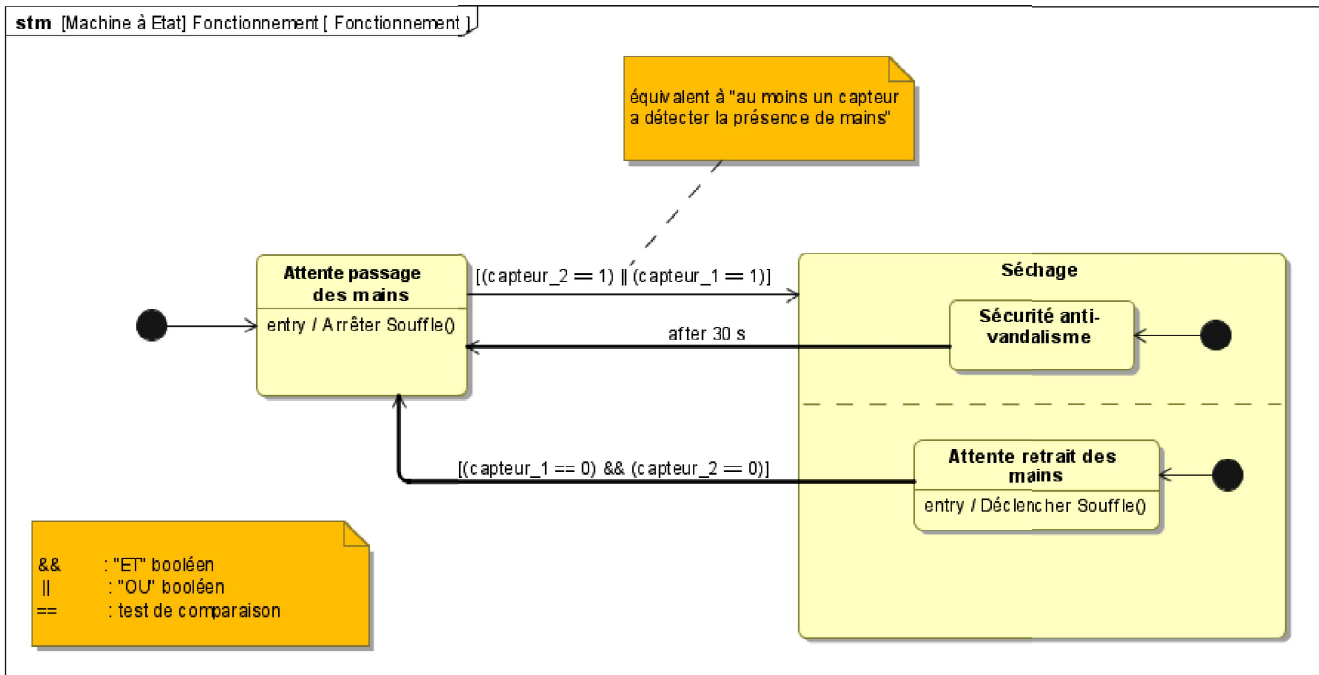
Il décrit les états successifs d'un système ou sous-système qui occupent des états bien définis au cours de leur fonctionnement (machine d'état).

C'est le cas notamment de fonctionnement séquentiel.

Les principaux outils de description sont les suivants :

	Etat initial : il désigne le point de départ de la séquence qui peut correspondre à la mise en énergie
	Etat final : il représente la fin du fonctionnement du système qui peut correspondre à la mise hors énergie. Plus aucun état n'est actif
	Etat : un état représente une situation pendant laquelle: <ul style="list-style-type: none"> • il satisfait une certaine condition, • il exécute une certaine activité, • il attend un certain événement.
	Etat composite : aussi appelé super-état; il permet d'englober plusieurs sous-états. Il comporte un état initial.
	Transition : elle représente l'évolution du système d'un état à un autre selon une condition de déclenchement.
	Fourche (Fork) : active simultanément 2 états à partir d'un état antérieur. Il existe également l'union qui active 1 état à partir de 2 états précédents.

5.1 Diagramme d'état du sèche-main



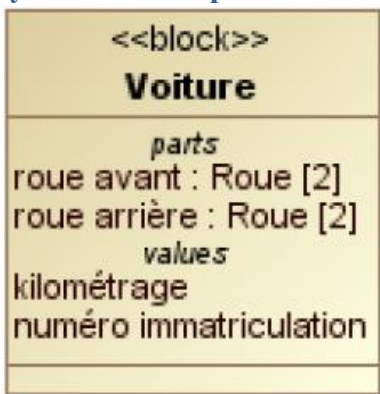
5.2 Application

Quelles sont les conditions pour activer l'état de séchage ?

Quelle est l'équation logique de la transition entre "Attente retrait des mains" et "Attente passage des mains" ?

6 – DIAGRAMME DE DEFINITION DE BLOC (BDD)

Ce diagramme décrit la hiérarchie du système et les classifications système/composant.



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

Il peut représenter un système complet, un sous-système ou un composant élémentaire.

Les blocs sont décomposables et peuvent posséder un comportement.

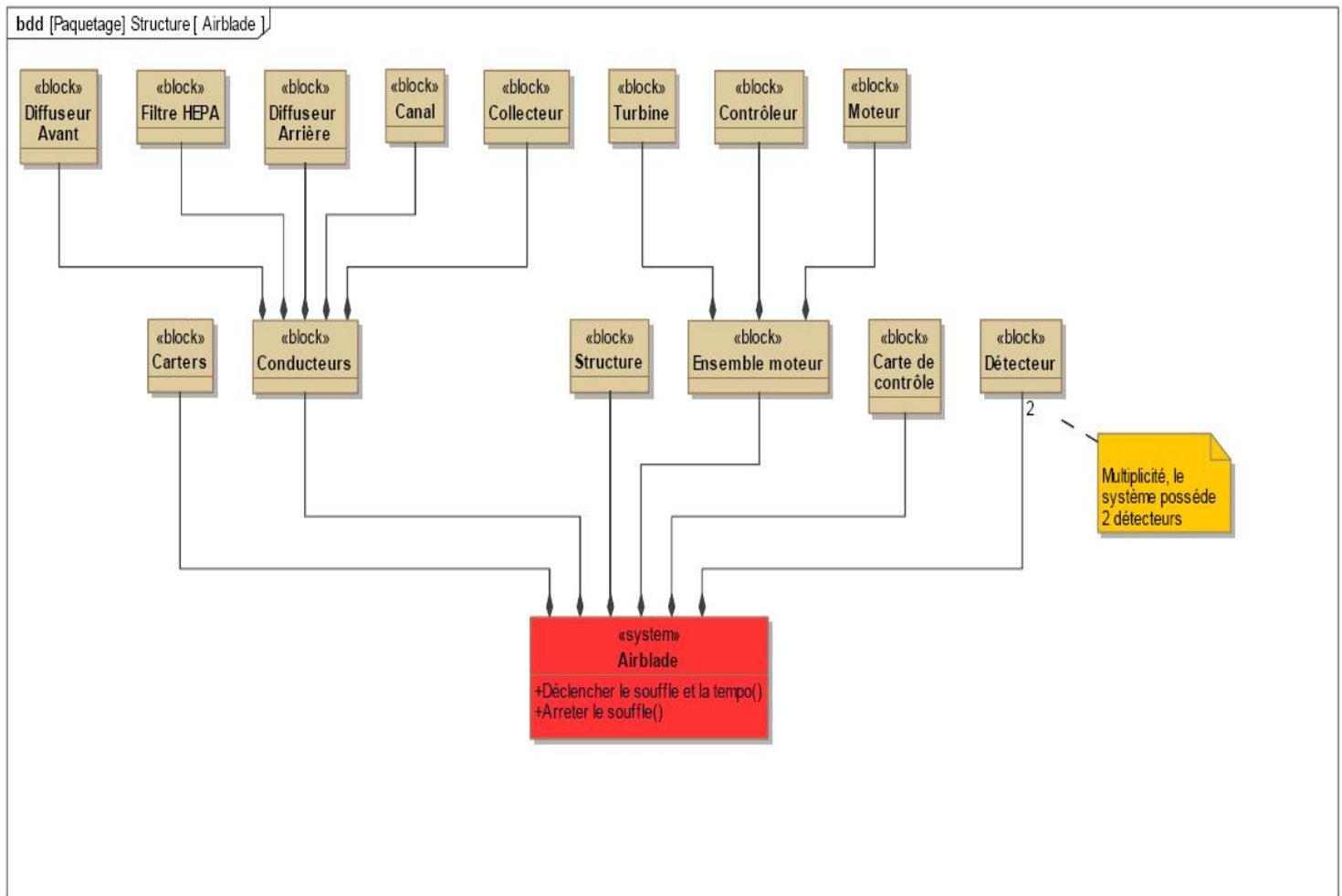
Les propriétés sont les caractéristiques structurelles de base des blocs.

Elles peuvent être de deux types principaux:

- **les valeurs (value properties)** décrivent des caractéristiques quantifiables en terme de valeur types (domaine de valeur, dimension et unité optionnelles)
- **les parties (part properties)** décrivent la hiérarchie de décomposition du bloc en termes d'autres blocs.

Un bloc peut être décomposé en plusieurs blocs avec des relations de composition \blacklozenge et d'agrégation \blacklozenge .

6.1 Diagramme du sèche-mains



6.2 Application

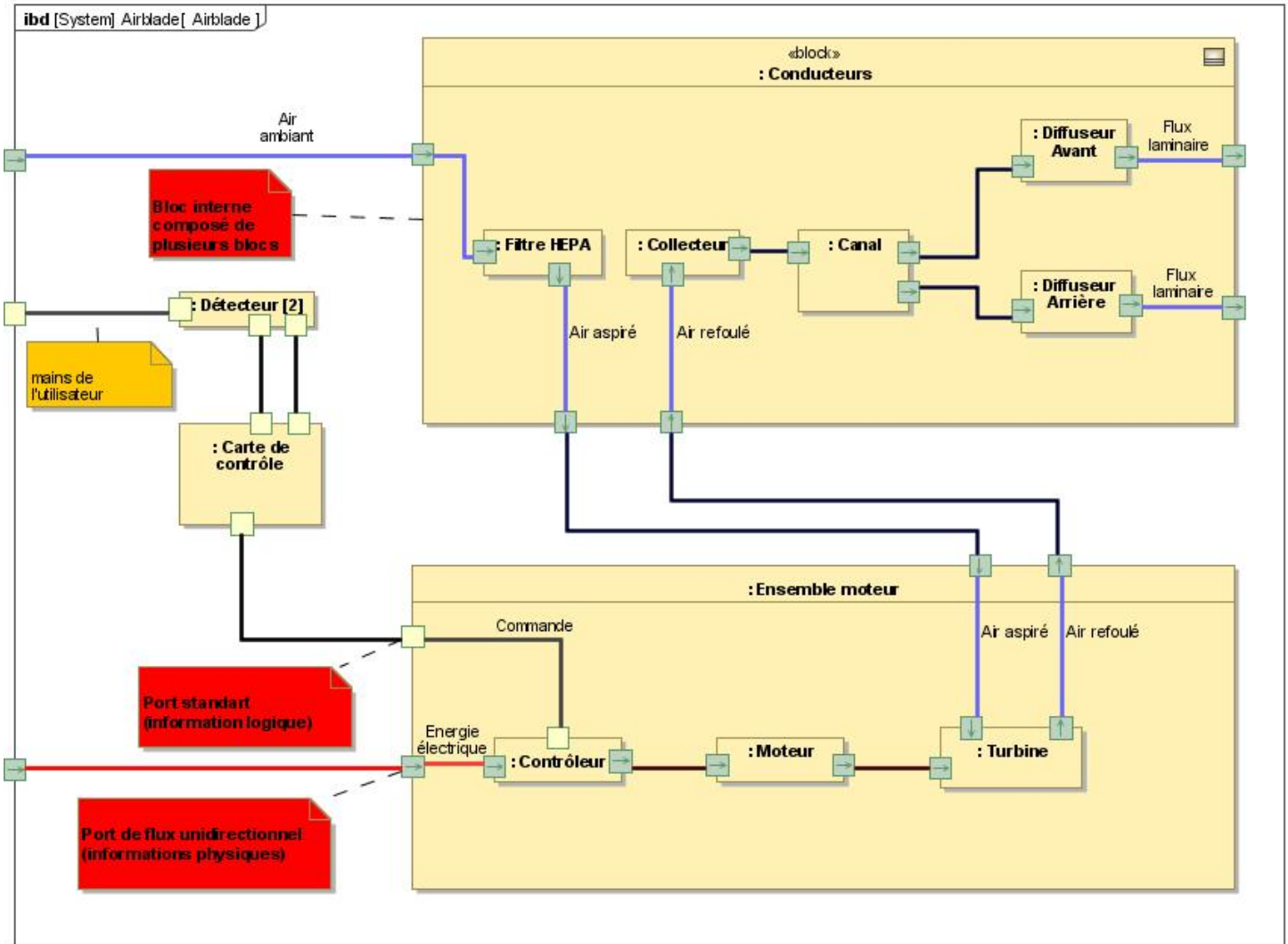
Quels sont les blocs qui composent l'ensemble moteur ?

Quel est le bloc qui assure une bonne hygiène lors du séchage ?

7 – DIAGRAMME DE BLOCS INTERNES (IBD)

Ce diagramme (internal block diagram ou ibd) décrit la structure interne du système et les échanges internes entre ses éléments ou avec l'extérieur.

7.1 Diagramme de blocs internes du sèche –mains



7.2 Exercices :

Quelle est le type d'énergie du flux entre le moteur et la turbine ?

Quelle est le type d'énergie du flux entre le contrôleur et le moteur ?

Pourquoi y-a-t-il 2 flux de liaison entre le bloc détecteur et le bloc carte de contrôle ?

Tracer sur le diagramme le trajet de l'air