

# 1- Ingénierie Système et langage SysML<sup>1</sup>

---

## Contenu :

<b>A. POURQUOI UNE INGÉNIERIE SYSTÈME ?.....</b>	<b>3</b>
A.1.    Pour répondre au mieux au besoin .....	3
A.2.    Pour organiser au mieux le projet .....	4
A.3.    Pour s'adapter à la complexité croissante des systèmes.....	5
<b>B. APPROCHE SYSTÈME .....</b>	<b>7</b>
B.1.    Typologie des systèmes.....	7
B.1.1.    Définition.....	7
B.1.2.    « Système à faire », « Système pour faire » .....	7
B.1.3.    « Sous-système », « Sur-système » .....	7
B.2.    Frontières, contexte.....	9
B.3.    Points de vue, visions .....	9
B.4.    Cycle de vie.....	11
B.4.1.    Étapes du cycle de vie d'un produit.....	11
B.4.2.    Les impacts environnementaux.....	11
B.4.3.    Analyse du cycle de vie .....	12
<b>C. L'APPROCHE « INGÉNIERIE SYSTÈME » .....</b>	<b>13</b>
C.1.    Définition .....	13
C.2.    L'approche « processus » .....	14
C.3.    L'approche temporelle des processus techniques .....	17
C.4.    Problème vs Solution.....	17
<b>D. LE LANGAGE SysML .....</b>	<b>18</b>
D.1.    Origines.....	18
D.2.    Les diagrammes .....	19
D.3.    Les objets et les relations .....	20

---

<sup>1</sup> D'après le cours de Y. Le Gallou lycée Baggio Lille

<b>E. L'I.S. : UNE DÉMARCHE .....</b>	<b>22</b>
<b>E.1. Définition des besoins des parties prenantes (Analyse du besoin) .....</b>	<b>22</b>
E.1.1. Définir la mission principale du système .....	23
E.1.2. Définir le(s) contexte(s) du système.....	24
E.1.3. Définir les utilisations du système.....	24
E.1.4. Décrire les scénarios d'utilisation.....	25
E.1.5. Définir les besoins des parties prenantes .....	26
E.1.6. Vérifier, valider et documenter la Définition des Besoins des Parties Prenantes.	28
<b>E.2. Analyse des exigences.....</b>	<b>29</b>
E.2.1. Analyser le périmètre du système .....	29
E.2.2. Définir les concepts système .....	29
E.2.3. Décrire les missions du système .....	30
E.2.4. Définir les exigences système .....	32
E.2.5. Assurer la traçabilité des exigences système.....	33
E.2.6. Vérifier, valider et documenter les exigences système .....	34
<b>E.3. Conception de l'architecture .....</b>	<b>35</b>
E.3.2. Définir la vue logique interne.....	37
E.3.3. Associer les opérations aux états.....	37
E.3.4. Vérifier l'architecture logique .....	38
E.3.5. Analyser les architectures candidates.....	38
E.3.6. Allouer les opérations aux sous-systèmes .....	38
E.3.7. Définir les échanges avec les sous-systèmes .....	40
E.3.8. Définir la vue interne du système .....	40
E.3.9. Vérifier l'architecture physique.....	41
E.3.10. Valider et documenter la conception de l'architecture.....	41
<b>E.4. Synthèse et conclusion.....</b>	<b>42</b>

## A. POURQUOI UNE INGÉNIERIE SYSTÈME ?

La nécessité de l'I.S. peut se justifier au travers de trois principaux constats.

### A.1. Pour répondre au mieux au besoin

Les photos suivantes illustrent le cas typique d'une situation où la solution envisagée ne répond pas au besoin initial, ou n'est pas adaptée à celui-ci.



Un camion-grue repêche une voiture...



... mais celle-ci entraîne le camion-grue avec elle.



Finalement un second camion-grue à stabilisateurs repêche la voiture...



... ainsi que le premier camion-grue.

C'est le cas typique d'une solution dite "de facilité", où pour des raisons budgétaires et/ou temporelles on préfère user de solutions sous-optimales (en termes d'efficacité, de fonctionnalités offertes), et qui bien souvent engendrent des surcoûts ou des délais supplémentaires (soit donc l'effet inverse de celui escompté).

Autre exemple : en 2014, les nouveaux trains régionaux sont conçus plus larges que leurs prédecesseurs, 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 !

De manière générale, le Standish Group [1] a établi que les principales causes de succès ou d'échecs d'un projet sont liées aux exigences (un besoin est une exigence de besoin) :

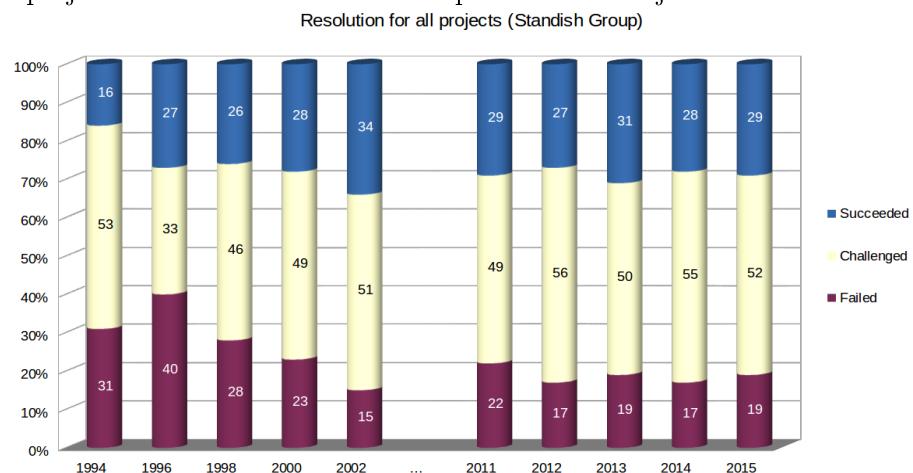
Facteurs d'échecs	Causes racines	Facteurs de succès
37 %	Besoins, exigences	40 %
9 %	Projet, ressources	23 %
8 %	Gestion des données techniques	14 %
11 %	Technique, technologie	9 %

### A.2. Pour organiser au mieux le projet

Le Standish Group [1] analyse depuis 1994 plusieurs milliers de projets informatiques dans le monde entier. Cette étude peut être extrapolée pour les systèmes complexes.

Elle a classé 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.
- Challenger : 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.

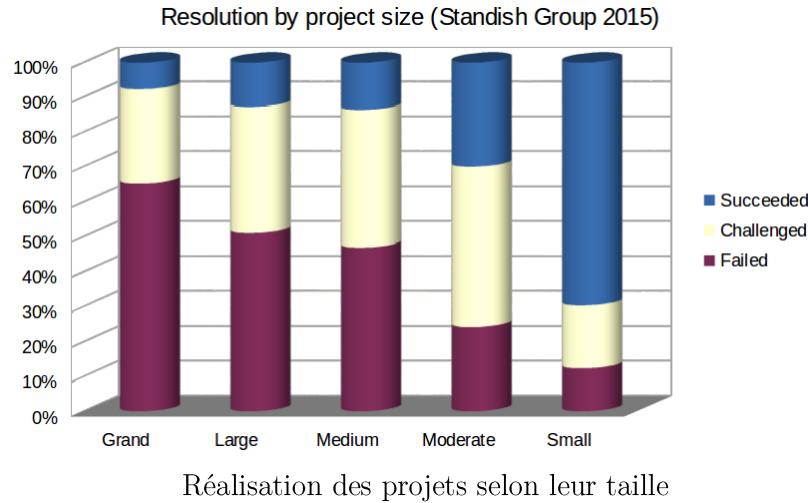


Historique du Standish Group pour la réalisation des projets

Ces résultats mettent en évidence que, suite à une période transitoire correspondant à l'émergence des méthodes dites Agiles [2], les taux de réussite et d'échecs demeurent quasiment constants depuis le début des années 2000, pour seulement 30 % de projets respectant intégralement le cahier des charges !

Le Standish Group a répertorié en 2015 ces mêmes résultats pour l'ensemble des projets selon leur taille.

Il apparaît clairement une forte corrélation où plus la taille du projet augmente (et donc avec elle les besoins en termes de gestion et d'organisation) plus la réussite est compromise.

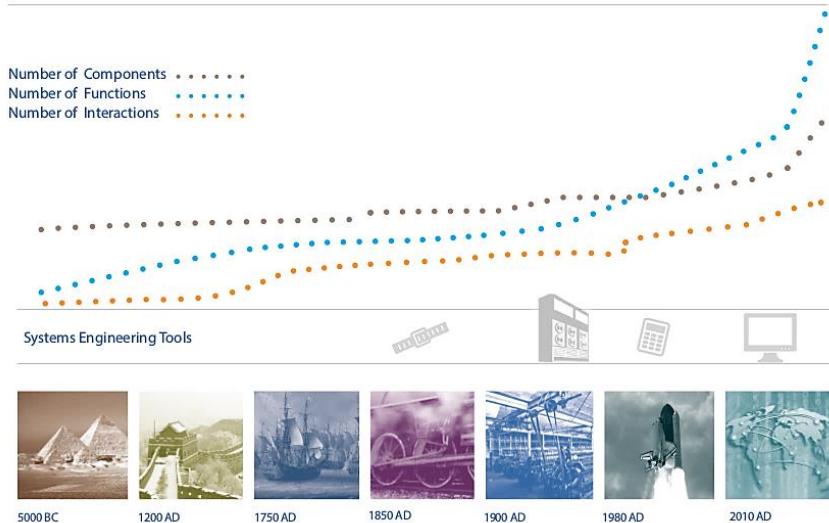


Réalisation des projets selon leur taille

### A.3. Pour s'adapter à la complexité croissante des systèmes

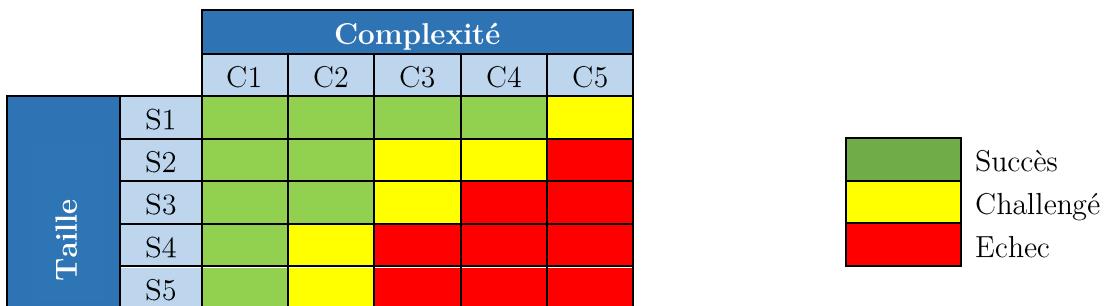
La complexité est un terme souvent utilisé pour désigner les caractéristiques (facteurs de complexité) qui rendent difficiles l'ingénierie des systèmes [3] telles que :

- Complexité structurelle : nombre, variété, hétérogénéité des composants ;
- Complexité dynamique : nombre et variété des interactions, combinatoire des parcours dans l'espace des états du système ;
- Importance des facteurs humains : méconnaissance de l'environnement, incertitudes diverses concernant le projet et/ou le système.



Complexité croissante des systèmes [4]

Cette 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.

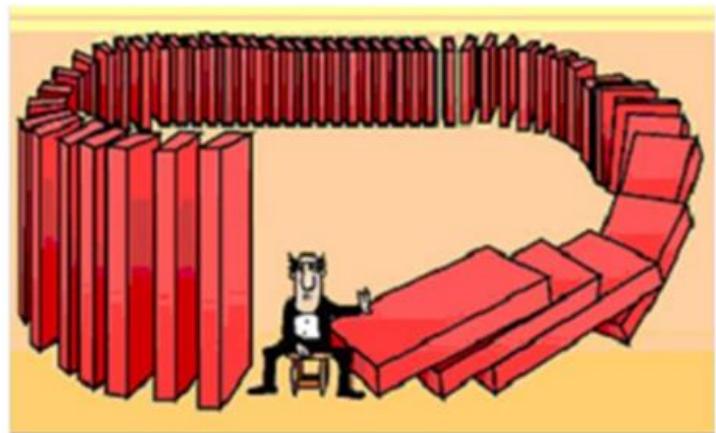


Risque vs taille/complexité des projets [1]

D'après [5], il n'y a pas aujourd'hui de mesure reconnue de la complexité, c'est encore un sujet de recherche.

Le comportement des systèmes complexes ne peut pas être déterminé analytiquement. Cela signifie que l'application de méthodes de calcul ne conduit pas à la fourniture de l'action optimale.

Des décisions doivent être prises en fonction d'une situation particulière, sans preuve d'avoir atteint un optimum.



Dans les systèmes complexes, les causes et les effets sont souvent éloignés dans le temps et l'espace [5]

Les attentes concernent :

- La simplification des processus de développement : les normes actuelles affichent parfois une complexité par trop dissuasive ;
- La simplification perçue des modes d'utilisation : les IHM des Mac et autres Ipad et Iphone sont un bon exemple ;
- La simplification des architectures : les études de la « Lean Advancement Initiative<sup>2</sup> » montrent que la majorité des erreurs sont dans les interfaces.

---

<sup>2</sup> Voir <http://lean.mit.edu>

## B. APPROCHE SYSTÈME

Avant d'aller plus loin, il est primordial de définir certaines notions et concepts propres aux systèmes.

### B.1. Typologie des systèmes

#### B.1.1. Définition

**Système** [6] : « Constitution d'un ensemble d'éléments dont l'association est organisée pour répondre à une finalité dans un environnement donné ».

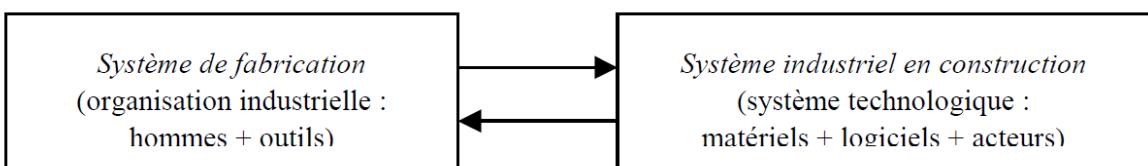
#### B.1.2. « Système à faire », « Système pour faire »

Selon Daniel Krob [7] : « Il est notamment important de comprendre que l'ingénierie système joue sur deux registres principaux fortement couplés, à savoir :

- Une dimension *technique*, centrée sur le *système industriel* (en tant que tel) que l'on veut construire ;
- Une dimension *organisationnelle et managériale* qui est centrée sur le *système de fabrication* du système industriel que l'on cherche à réaliser. »

Deux sortes de systèmes sont ainsi impliquées dans l'I.S. :

- Le **système à faire** : le système industriel à réaliser en tant que tel, répondant à une finalité à forte valeur ajoutée (amélioration d'un service, réduction d'un impact, ...) ; système mettant en œuvre l'I.S. (on parle alors de *système d'intérêt*) ;
- Le **système pour faire** : le système de fabrication (le projet), répondant à une finalité de réaliser le système le plus optimal possible selon des contraintes propres (délais, budget, RH, ...) ; système mis en œuvre pour réaliser l'I.S..



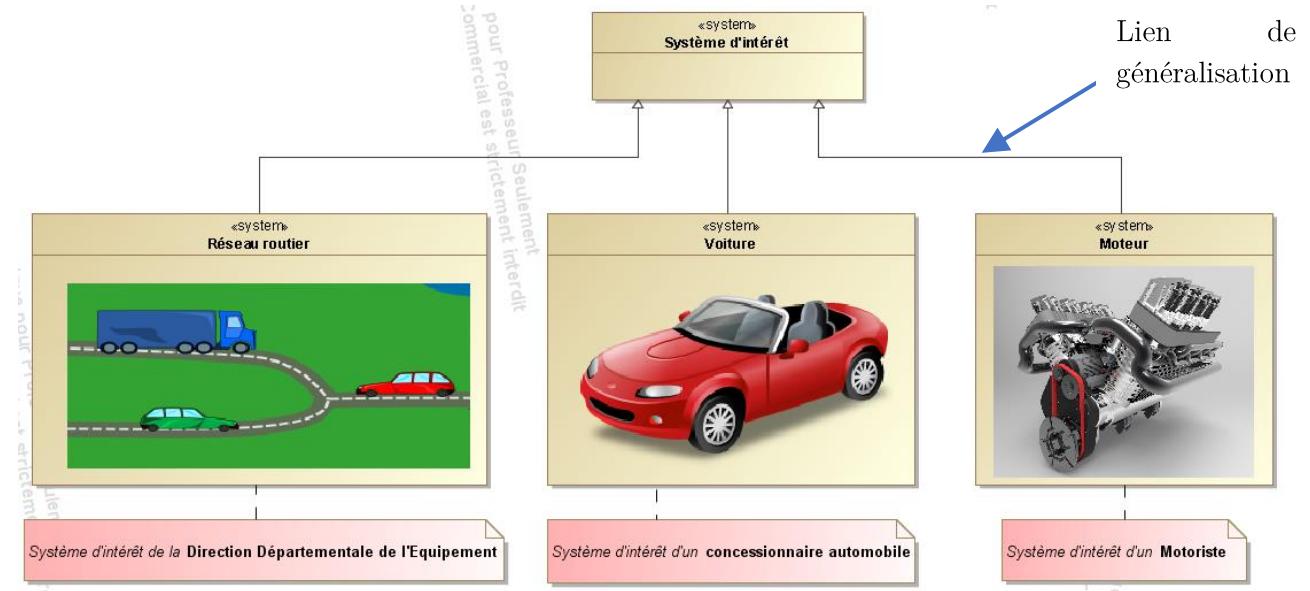
Les deux systèmes structurant la démarche d'ingénierie [7]

#### B.1.3. « Sous-système », « Sur-système »

- Le système d'intérêt est constitué lui-même de systèmes de plus bas niveau : par rapport au système étudié, ils constituent des *sous-systèmes* du système d'intérêt.
- Le système d'intérêt fait partie lui-même d'un système de plus haut niveau : par rapport au système étudié, il constitue le *sur-système* du système d'intérêt.

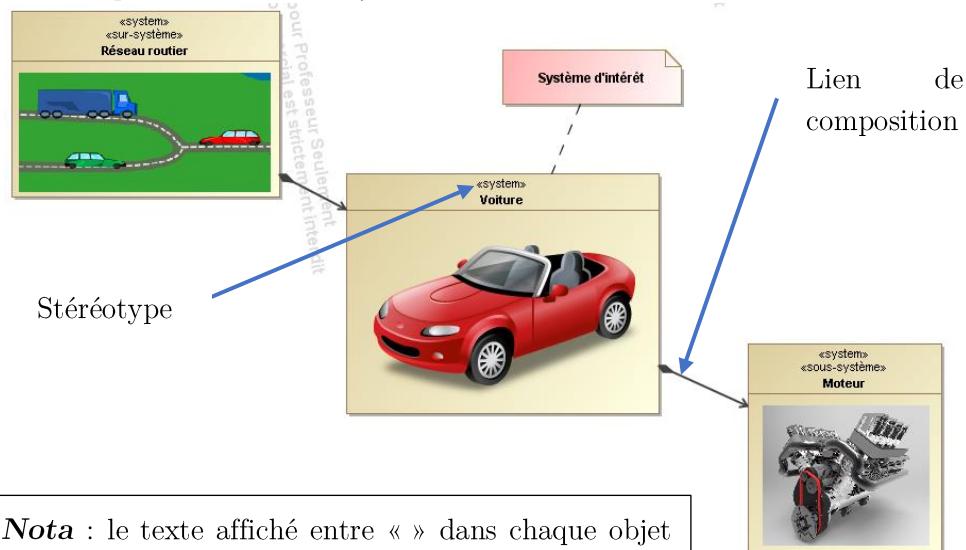
**Remarque** : un système (sous-entendu « complexe ») à au moins deux sous-systèmes, et n'a au plus qu'un seul et unique sur-système.

Prenons l'exemple suivant : un réseau routier, une voiture, un moteur. Trois systèmes par définition, mais différents selon le centre d'intérêt.



**Nota :** le lien reliant les blocs « system » (terminé par une flèche fermée) est un **lien de généralisation** en langage SysML : il signifie, dans le sens de la flèche, que l'objet de départ est un cas particulier de l'objet d'arrivée, objet plus générique. Dans cet exemple, le réseau routier est un cas particulier d'un système d'intérêt, aux yeux de la DDE<sup>3</sup>.

Maintenant, si parmi ces trois systèmes, le système d'intérêt est la voiture, alors le moteur constitue un sous-système de la voiture, qui elle-même constitue un sous-système du réseau routier, qui est donc le sur-système de la voiture.



**Nota :** le texte affiché entre « » dans chaque objet s'appelle un **stéréotype** : il spécifie le(s) type(s) de l'objet en question, et confère à l'objet certaines propriétés et attributs, comme les relations possibles avec d'autres objets.

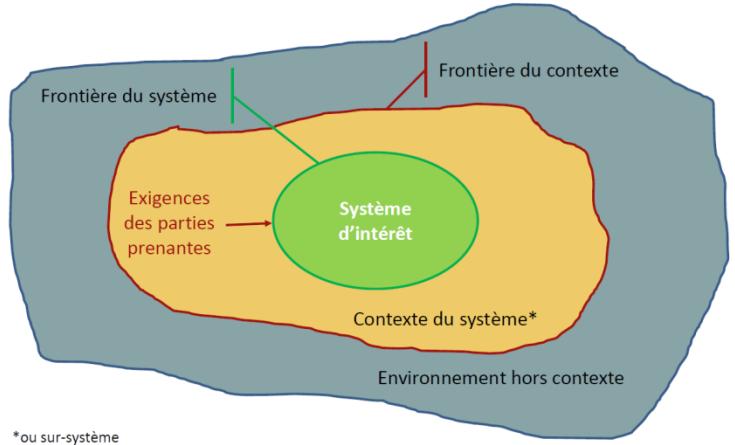
**Nota :** le lien reliant les blocs (avec un losange noir au départ et terminé par une flèche ouverte) est un **lien de composition** en langage SysML. Il signifie, dans le sens de la flèche, que l'objet de départ est composé de l'objet d'arrivée : ce dernier fait donc partie intégrante de l'objet de départ considéré.

<sup>3</sup> DDE : Direction Départementale de l'Équipement

**Remarque** : il est courant en ingénierie système de parler de « Système de systèmes » (SoS pour « System of Systems »), définit comme : « l’assemblage de constituants réalisant une tâche qu’aucun autre système ne peut accomplir lui-même. Chaque constituant possède son propre management, objectifs et ressources, de manière coordonnée au sein du SoS et adaptée aux objectifs de celui-ci » [6].

## B.2. Frontières, contexte

Le système d’intérêt est défini par tous les éléments qui participent à le rendre fonctionnel. En phase d’utilisation, il évolue dans un environnement (un contexte), en interaction avec d’autres éléments et/ou acteurs.



Ces derniers, que ce soient des personnes physiques ou morales, sont appelées « **parties prenantes** » du système, ayant un intérêt direct vis-à-vis de celui-ci (exigences des parties prenantes).

Tout ce qui ne figure pas dans ce contexte est considéré « hors-contexte ».

**Remarque** : le contexte du système constitue le sur-système quand il existe. Ainsi, des éléments hors-contexte pour un système pourront être dans le contexte du sur-système : il est parfois utile, pour bien définir la frontière du système, de bien définir le sur-système en lui-même.

## B.3. Points de vue, visions

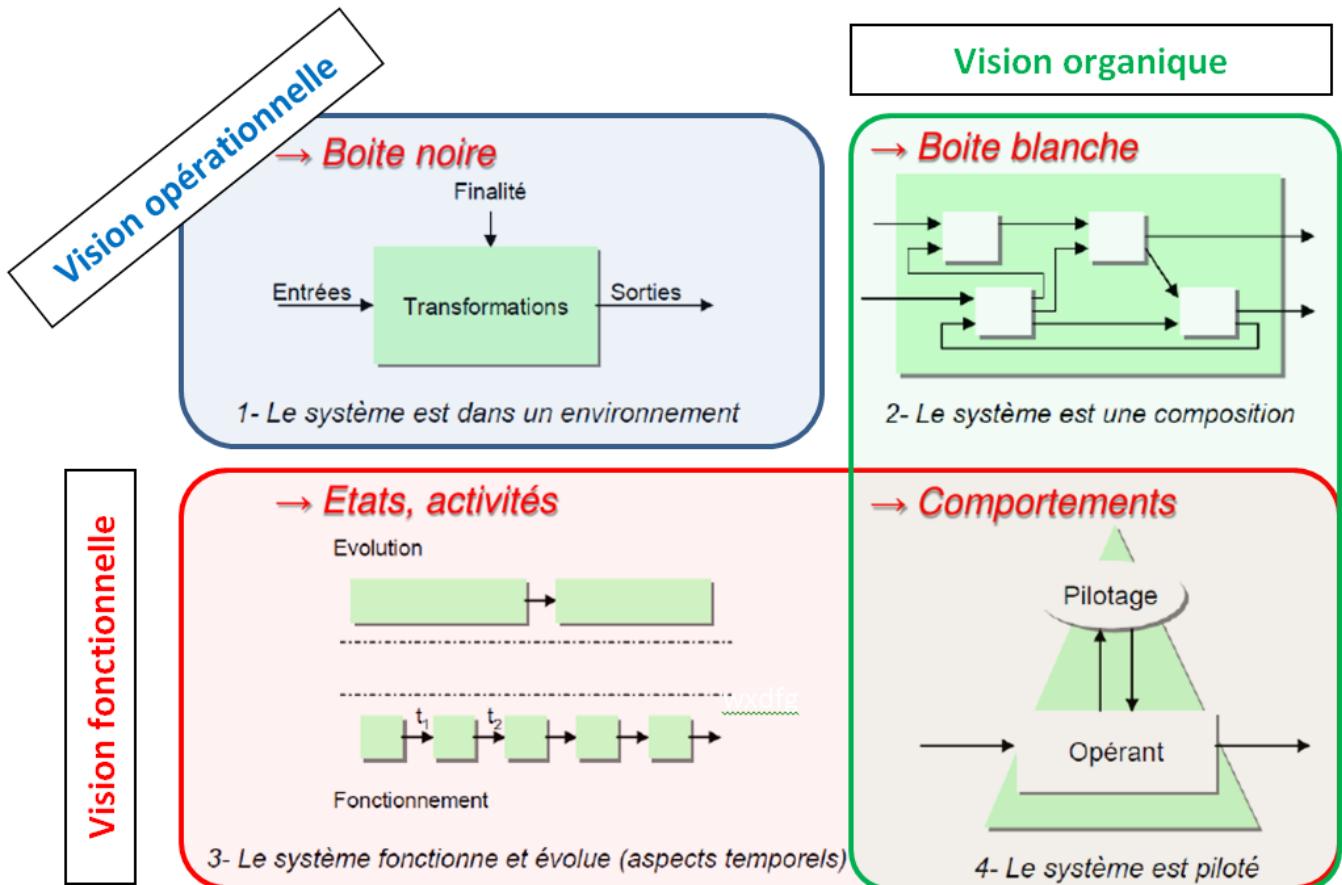
D’après [7] : « Pour analyser un système réel donné, l’architecte système utilise classiquement trois angles d’analyse – ou visions – qui structurent notamment le référentiel de l’ingénieur, au sens des exigences et des spécifications, à savoir :

- La **vision opérationnelle** : qui a pour but de définir le pourquoi du système, autrement dit de préciser la mission du système et ce à quoi sert le système ;
- La **vision fonctionnelle** : qui a pour objectif d’expliquer le *fonctionnement logique* du système, i.e. ce qu’il fait indépendamment de la façon dont on le réalisera ;
- La **vision organique** : qui définit la façon dont le système est *concrètement réalisé*, i.e. l’organisation et la dynamique de ses composants matériels, logiciels et humains. »

Ces différentes visions permettent de répondre au fameux tryptique du « Why ? What ? How ? » :

Visions	Répond à la question	Exemple de points de vue	Instances de points de vue (Brosse à dents électronique)
Opérationnelle	Pourquoi ? (Why)	Mission, contexte opérationnel, contexte stratégique	Dents propres et saines, gain de temps, salle de bain « High tech »
Fonctionnelle	Quoi ? (What)	Fonction, fonctionnement et mode de fonctionnement	Brossage, régulation de vitesse, programmation de la force de brossage
Organique	Comment ? (How)	Composant, organe et configuration technique	Tête, base, corps régulateur de vitesse

A ces différentes visions, on associe fréquemment les notions de boîte noire, boîte blanche, d'états d'un système, d'activités d'un système, de pilotage d'un système [8] :



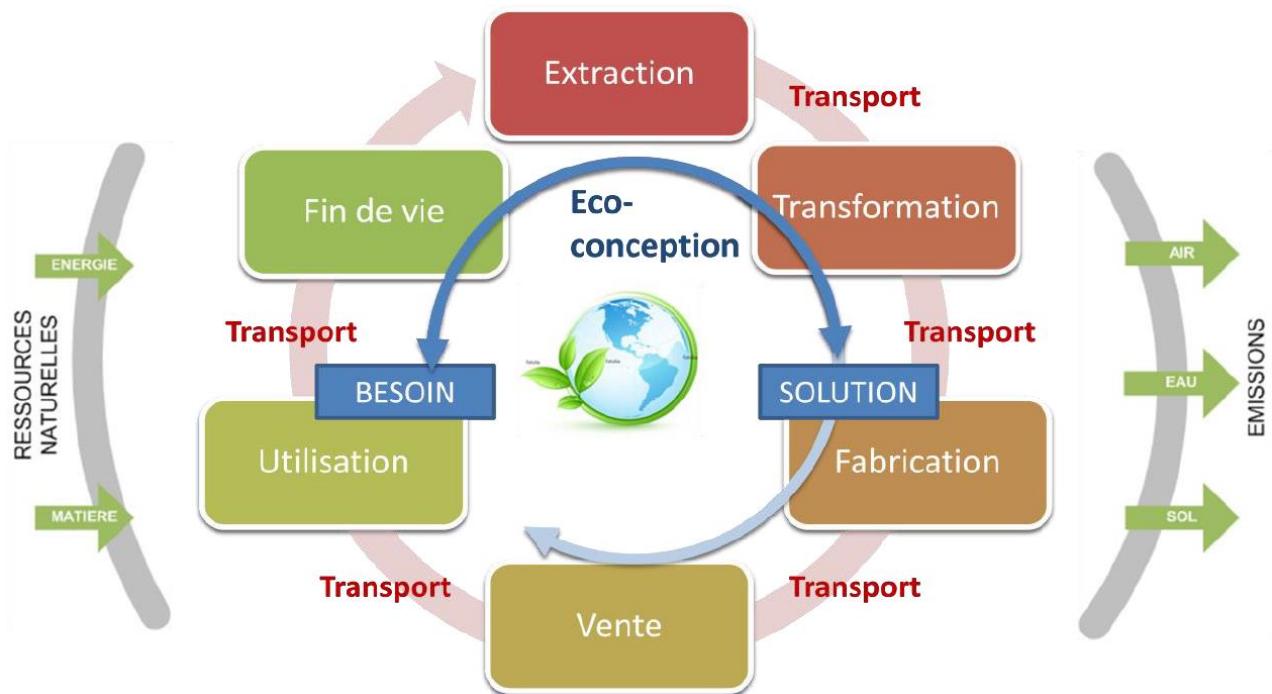
## B.4. Cycle de vie

Nous distinguerons ici deux visions du cycle de vie d'un système, très supplémentaire et dont la finalité diffère selon l'usage :

- Le cycle de vie d'un produit, défini par la norme I.S.O 14040, et utile à des fins d'analyse du cycle de vie et de limitation des impacts environnementaux tout au long du cycle de vie ;
- Le cycle de vie d'un système, défini par la norme I.S.O 15288, et utile à mettre en œuvre l'I.S. dans une démarche de conception.

### B.4.1. Étapes du cycle de vie d'un produit

Le cycle de vie commence à l'extraction des matières premières jusqu'à son élimination (fin de vie ou recyclage). La prise en compte de chacune de ces étapes pour l'analyse des différents impacts est primordiale, afin d'éviter un déplacement des problèmes (transfert d'impact). La démarche d'éco-conception ou d'éco-construction s'inscrit totalement dans ce cycle.



A chaque étape :

- Un produit consomme de l'énergie et des matières premières non renouvelables ;
- Un produit crée des impacts sur l'air, l'eau, le sol.

### B.4.2. Les impacts environnementaux

Les impacts environnementaux sont classés en 3 familles d'impact :

- Les ressources naturelles non renouvelables,
- Les pollutions
- Les nuisances.

A chaque impact correspond alors une unité de référence.

Familles d'impacts	Impacts	Substance ou unité de référence
Ressources naturelles non renouvelables	Consommation d'énergie non renouvelable	MégaJoule ( <i>MJ</i> )
	Consommation de ressources naturelles non renouvelables	Antimoine ( <i>Sb</i> )
Pollutions	Gaz à Effet de Serre (GES) ou Global Warming Potential (GWP)	Dioxyde de carbone ( <i>CO<sub>2</sub></i> )
	Acidification liée aux pluies acides	Dioxyde de soufre ( <i>SO<sub>2</sub></i> )
	Eutrophisation <sup>4</sup> liée à l'enrichissement des milieux aquatiques en sels nutritifs	Composé phosphaté ( <i>PO 3–4</i> )
	Dégénération de la couche d'ozone	Fréon 11 (CFC-11)
	Toxicité d'une substance sur les organismes (écotoxicité)	1,4 Dichlorobenzène (1,4 DCB)
Nuisances	Acoustiques	Décibel (dB)
	Visuelles	
	Olfactives	

**Nota :** Pour comparer les différents impacts environnementaux, on utilise généralement le taux de CO<sub>2</sub> équivalent émis.

Matériau	Émissions équivalentes de CO <sub>2</sub> en kg par tonne produite
Verre bouteille	120
Ciment	250
Acier	300 à 850 selon le pourcentage de ferrailles
Verre plat	400
Papier-carton	300 à 500
Plastiques (polyéthylène, polystyrène, PCV, PET...)	500 à 1600
Aluminium	600 à 3000 selon le pourcentage de déchets d'aluminium

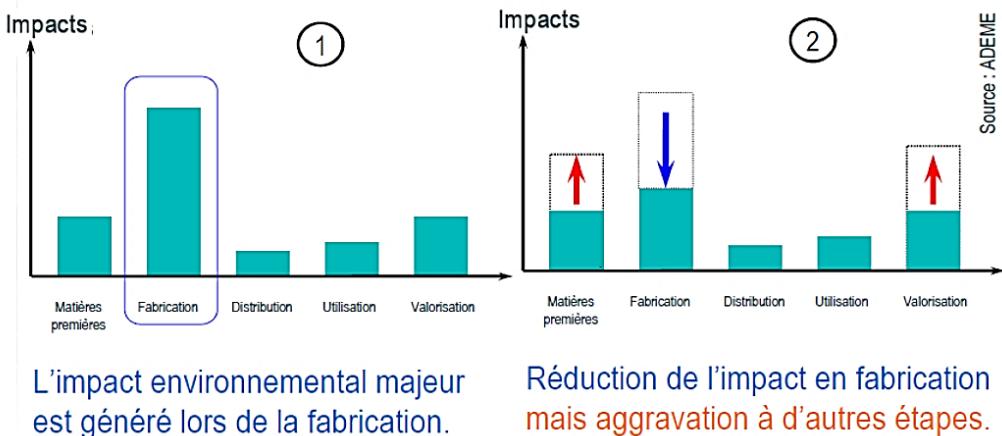
#### B.4.3. Analyse du cycle de vie

L'approche du cycle de vie peut être :

- **Multi-étapes** : prise en compte d'un critère sur toutes les étapes du cycle de vie du produit (depuis l'extraction des matières premières jusqu'à la fin de vie du produit, en passant par la fabrication, la distribution, l'utilisation) ;
- **Multicritères** : prise en compte pour une étape de l'ensemble des impacts environnementaux générés (matières premières, énergie, préservation de la biodiversité, pollution de l'eau, de l'air, des sols, déchets, bruit...).

<sup>4</sup> Eutrophisation : l'eutrophisation des milieux aquatiques est un déséquilibre du milieu provoqué par l'augmentation de la concentration d'azote et de phosphore dans le milieu. Elle est caractérisée par une croissance excessive des plantes et des algues due à la forte disponibilité des nutriments (Wikipédia)

L'analyse multicritères et multi-étapes évite tout transfert d'un impact à un autre. Il est donc nécessaire d'inventorier un maximum de paramètres (quantité de matière et d'énergie utilisés, procédés de fabrication, type de transport, ...) afin de mener l'analyse la plus exhaustive possible.



L'Analyse de Cycle de Vie (ACV) est contenue dans la série de norme I.S.O 14040 [9]. Plusieurs objectifs peuvent justifier la mise en œuvre de ces analyses :

- Publier une information sur le produit ;
- Concevoir ou reconcevoir ;
- Respecter un cadre réglementaire.

## C. L'APPROCHE « INGÉNIERIE SYSTÈME »

### C.1. Définition

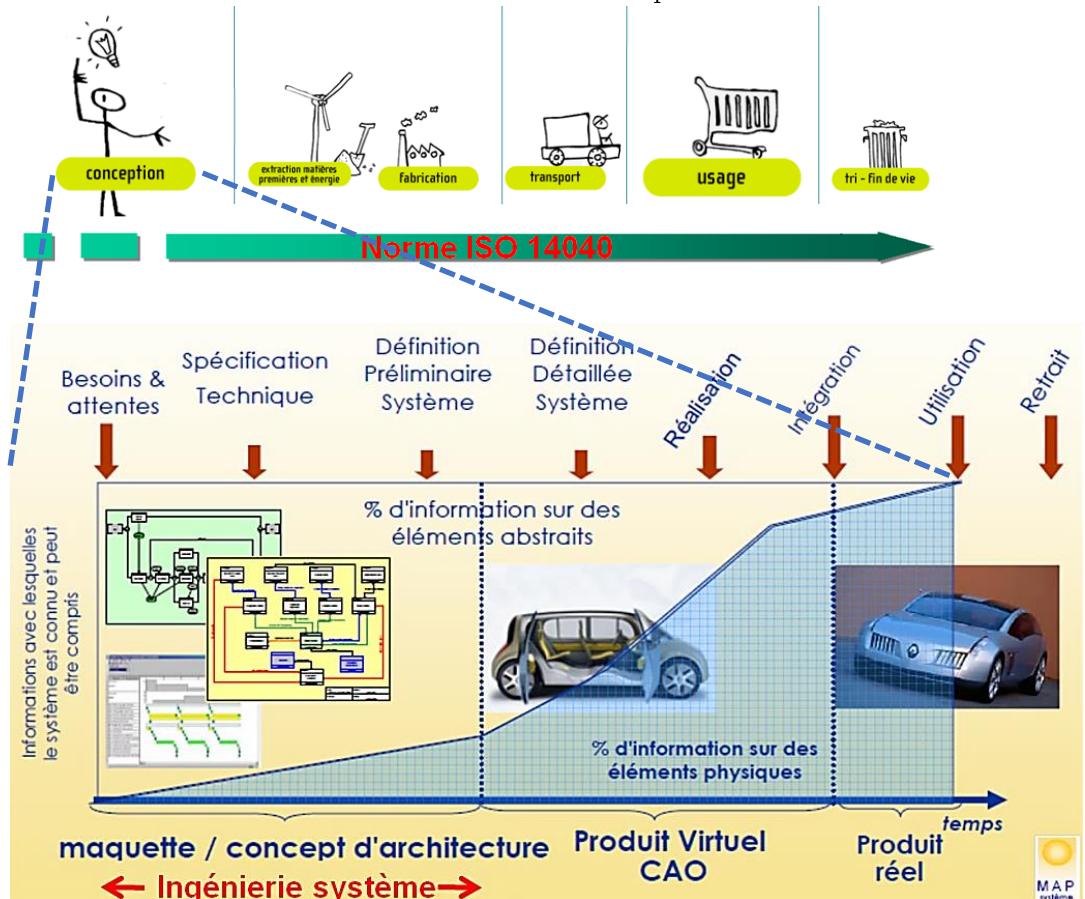
**Ingénierie système** [6] : « Démarche méthodologique coopérative et interdisciplinaire qui englobe l'ensemble des activités adéquates pour concevoir, développer, faire évoluer et vérifier un ensemble de produits, processus et compétences humaines apportant une solution économique et performante aux besoins des parties prenantes et acceptable par tous (inspiré de IEEE 1220). »

Cet ensemble est intégré en un système, dans un contexte de recherche d'équilibre et d'optimisation sur tout son cycle de vie.

Selon l'AFI.S. [8] : « L'I.S. est un mode de pensée et une façon d'appréhender les affaires par une approche structurée pour passer du besoin à la solution. ». Cela inclut plusieurs aspects :

- La vision système des produits (voir chapitre B) ;
- Une démarche structurée pour le développement des systèmes, à savoir :
  - Un ensemble de processus, de méthodes et de techniques ;
  - Des techniques de développement via des modèles (SysML, ...) ;
  - Un couplage avec le management de projet (« système pour faire »/« système à faire »).

Par rapport au cycle de vie d'un système, l'ingénierie système prend pleinement position en phase de conception, tout en restant dans le domaine du conceptuel ; la conception détaillée et la réalisation incombant elles à une maîtrise d'œuvre plus orientée « métier ».



## C.2. L'approche « processus »

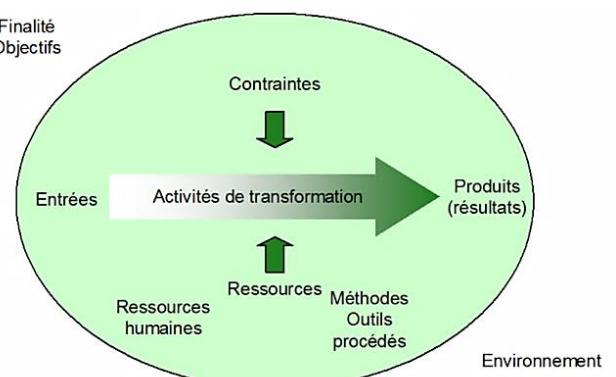
« Pourquoi une approche processus ? Parce que toutes les démarches d'évaluation et d'amélioration de la qualité (I.S.O 9000, Malcolm Baldrige, Modèles de Maturité, ...) partent du principe qu'un "bon" processus correctement mis en oeuvre engendre de "bons" produits » [10].

**Processus** : ensemble d'activités corrélées ou interactives qui transforment des éléments d'entrée en éléments de sortie (produits, résultats).

Ce faisant, un processus :

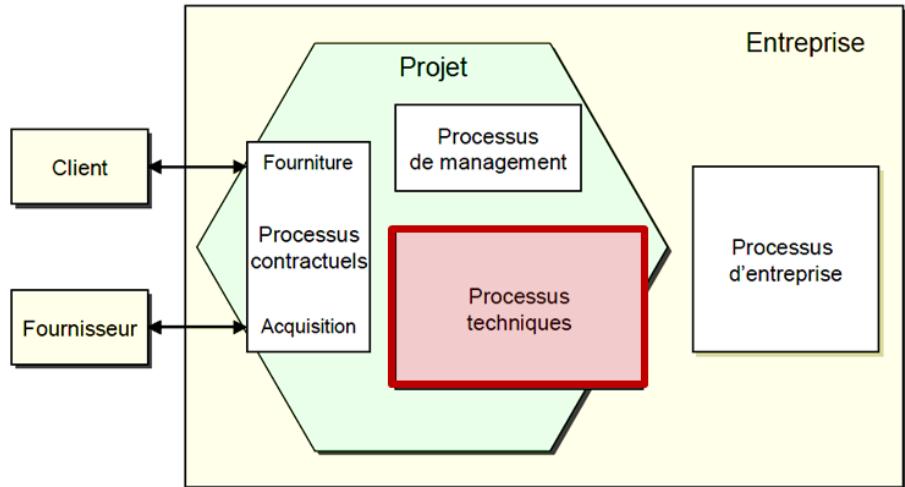
- Répond à une finalité (déclinée en objectifs de résultats ou produits) ;
- Est immergé dans un environnement donné ;
- Est soumis à des contraintes et consomme des ressources.

L'AFLS. [8] en donne un acronyme : PRECT (Produits, Ressources, Entrées, Contraintes, Transformations).



La norme I.S.O 15288 [6] définit 25 processus tout au long du cycle de vie d'un système, regroupés en 4 groupes :

- Les *processus contractuels* : aux interfaces de l'entreprise, pour fournir au client un produit répondant à ses besoins, nécessitant l'acquisition de constituants auprès de fournisseurs/sous-traitants ;
- Les *processus d'entreprise* : principalement liés à la logistique interne, le budget, la gestion des ressources humaines, la qualité ;
- Les *processus de management* : ou processus de projet (planification, évaluation et contrôle) ;
- Les *processus techniques* : comporte tous les processus propres à la conception, le maintien en conditions opérationnelles et le retrait d'un système.



Le cœur de l'I.S. constitue les 3 premiers processus techniques, qui sont :

- **La définition des besoins des parties prenantes** : processus aboutissant sur la spécification des besoins (on parle aussi d'« analyse du besoin »), qui est du ressort de la maîtrise d'ouvrage<sup>5</sup> (MOA) ;
- **L'analyse des exigences** : processus aboutissant sur les spécifications techniques (on parle aussi d'« analyse fonctionnelle externe »), qui est du ressort de la maîtrise d'œuvre<sup>6</sup> (MOE) ;
- **La conception de l'architecture** : processus aboutissant à la spécification des architectures fonctionnelle puis physique, qui est du ressort de la MOE, et en particulier au dossier de conception qui sera transmis à la maîtrise d'œuvre « métier » pour la réalisation (on parle aussi d'« analyse fonctionnelle interne »).

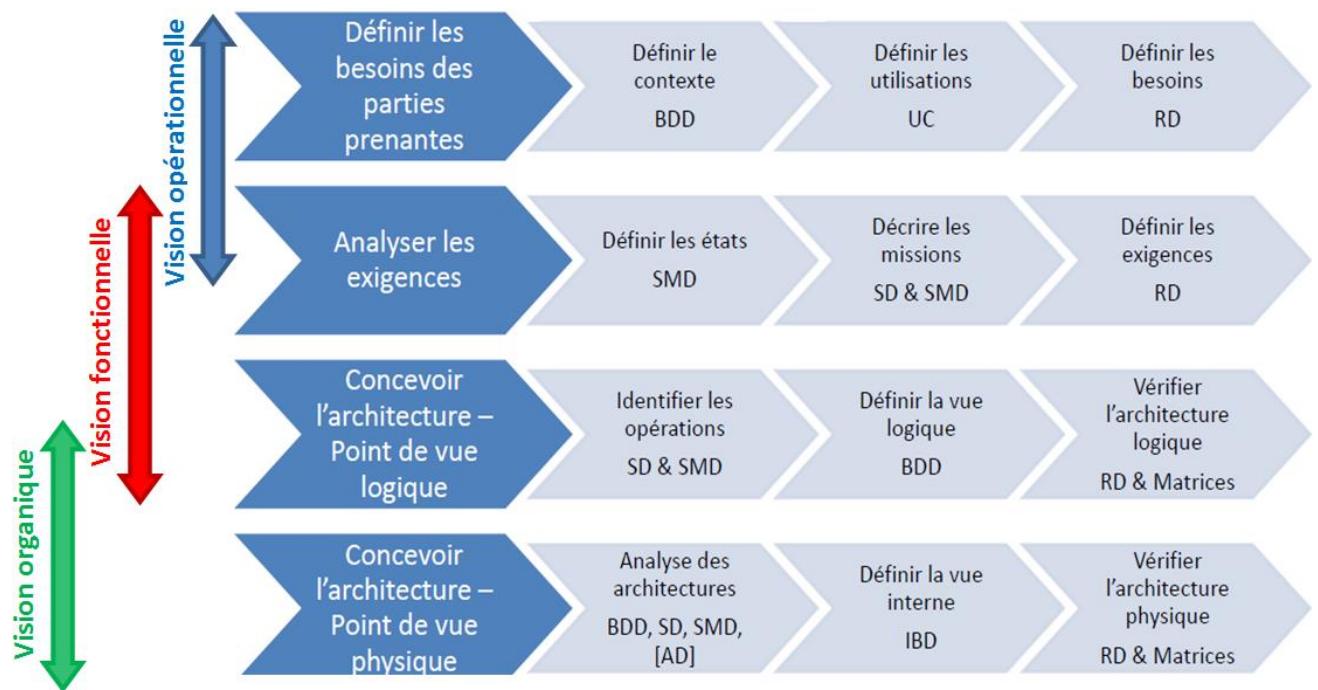
---

<sup>5</sup> Le maître d'ouvrage est responsable du besoin (de l'intégration des expressions de besoin des parties prenantes intéressées). Son rôle est d'obtenir un système répondant à ce besoin et de le mettre à disposition des exploitants et utilisateurs. Il reste responsable de l'intégration du système dans l'environnement d'exploitation.

<sup>6</sup> Le maître d'œuvre est responsable de la solution. Son rôle est de fournir un système répondant à ce besoin. Il est à ce titre l'intégrateur des parties prenantes concernées par la réalisation (intégration de leurs exigences, du projet de réalisation), ainsi qu'intégrateur, donc architecte, de la solution.

Chacun de ces processus mobilise différentes activités, produisant/exploitant un modèle de description du système (ici en langage SysML), permettant une ingénierie système « orientée modèle ».

MBSE = Model Based System Engineering



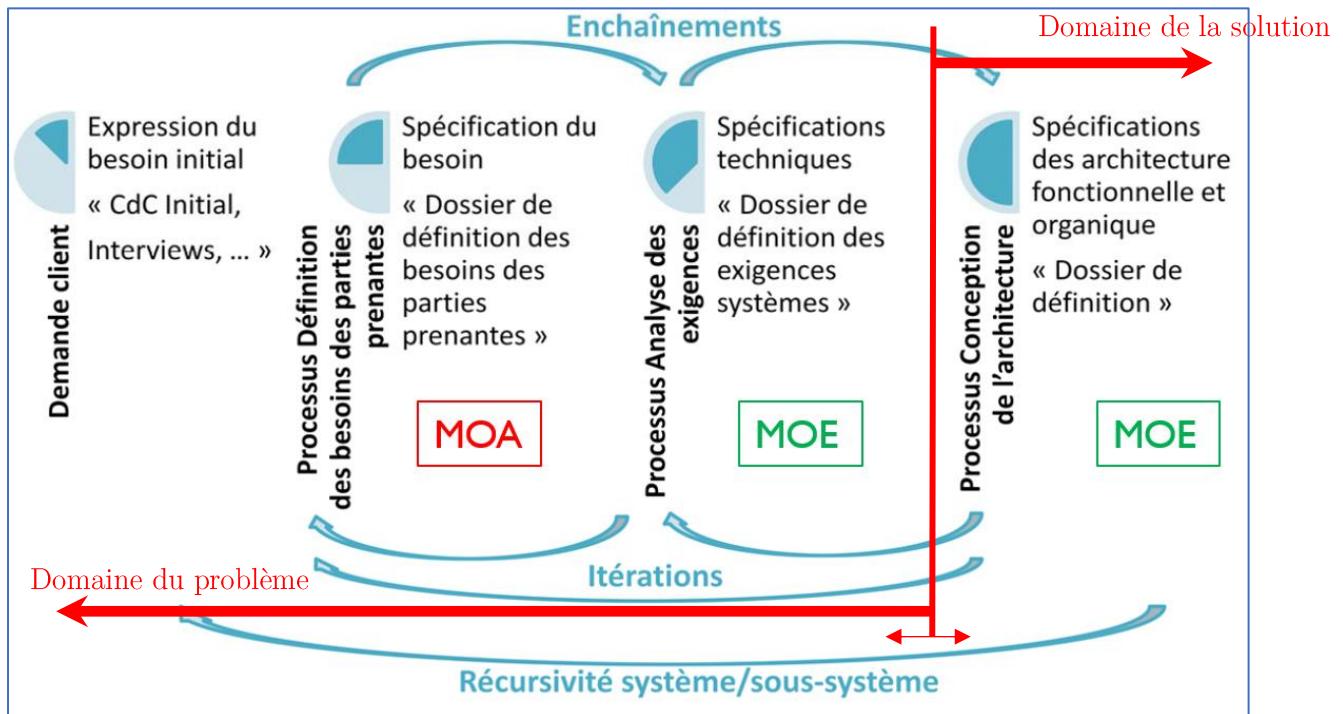
*Vision synthétique des activités réalisées durant les 3 processus techniques, et leurs modèles SysML associés.*

**Remarque :** Toute confusion entre les rôles de maître d'ouvrage et de maître d'œuvre est préjudiciable à la mise en œuvre de bonnes pratiques d'I.S. et à la bonne organisation du projet. En voici des exemples : confusion entre cahier des charges et spécification, validation insuffisante de la spécification par le maître d'ouvrage, ingérence du maître d'ouvrage dans les choix de solution, dont sous-traitance directe de parties de solution sans passer par la validation du maître d'œuvre, ...

### C.3. L'approche temporelle des processus techniques

L'ingénierie système est une démarche :

- *Séquentielle* – au sens relationnel d'antériorité/postériorité des uns par rapport aux autres (et non de « linéarité » de la démarche !), dans l'enchaînement des processus et de leurs activités : on commence par ceci, pour ensuite faire cela ...
- *Itérative* – dans les retours que cela implique : ayant fait cela, je dois refaire cela ...
- *Récursive* – au fil de la conception, chaque raffinement du modèle implique la même démarche (ayant décomposé ceci en ceci-cela, je recommence avec ceci-cela ...).



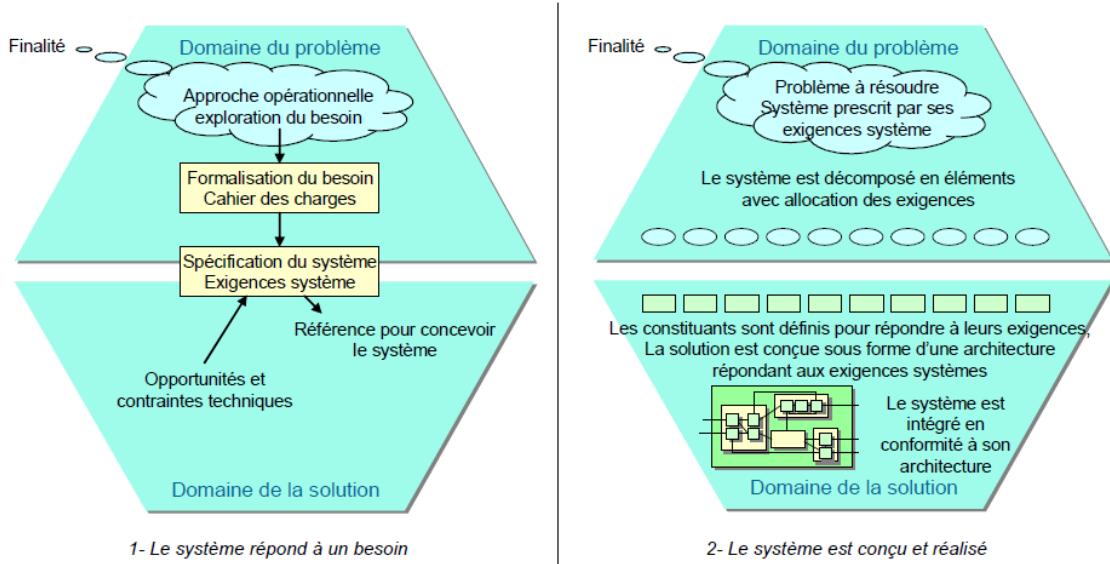
### C.4. Problème vs Solution

D'après [11] et [6], la spécification du besoin est un ensemble d'activités d'ingénierie système dans lesquelles les besoins de parties prenantes sont examinés **dans l'espace du problème** pour définir le système d'intérêt.

En réponse aux besoins exprimés par la MOA, le maître d'œuvre établit une spécification qui explicite sa compréhension du cahier des charges, la transcrivant selon les concepts système retenus. Cette spécification, une fois validée par le maître d'ouvrage, formalise le problème à

résoudre en **prescrivant** ce que le « système à faire » doit faire ... Elle sert alors de référentiel aux activités de conception.

**A ce titre, elle reste, en principe, dans le strict domaine du problème**, sans préjuger des choix techniques de solution, tout en cherchant à garantir la faisabilité de la solution par la prise en compte des contraintes des parties prenantes impliquées [8].



La dualité problème/solution selon [8]

## D. LE LANGAGE SysML

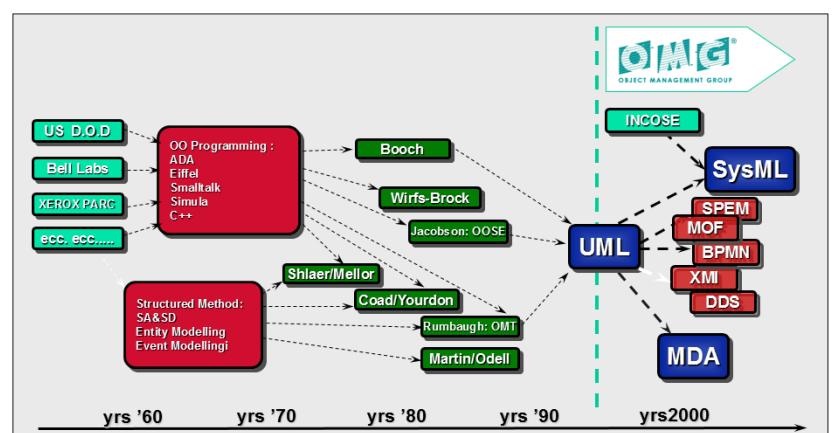
### D.1. Origines



La modélisation avec le langage UML, né d'une standardisation des différents langages de programmation orientés objets, est une pratique bien établie dans l'industrie logicielle. Bien que le langage UML permette par son caractère à usage général d'adresser de nombreux besoins pour l'I.S., il a été adapté pour définir un langage de modélisation des systèmes : SysML ou **Systems Modeling Language**.

Le besoin de définir un langage basé sur UML pour l'I.S. a été initié en 2001 par l'organisation internationale de l'ingénierie système INCOSE, qui s'est mise en relation avec l'OMG<sup>7</sup>, organisme responsable d'UML. SysML en quelques dates :

- 07/2006 : OMG annonce l'adoption de SysML
- 09/2007 : SysML v1.0 (première spécification)
- 05/2017 : SysML v1.5 [12] (version actuelle)

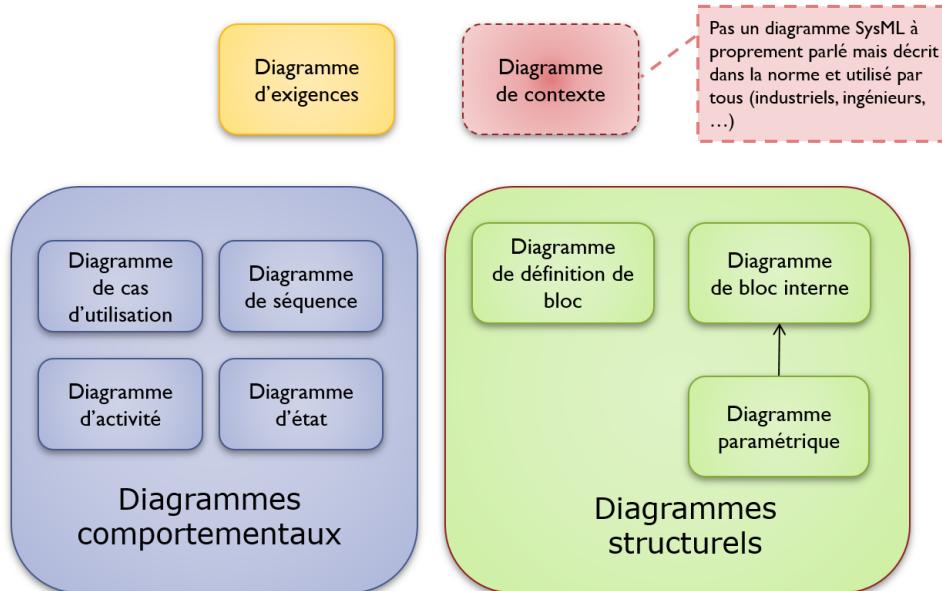


<sup>7</sup> Pour Object Management Group

SysML est donc un outil de modélisation relativement « jeune », mais qui aujourd’hui est à un stade de maturité avancé, et fait de plus en plus consensus chez les industriels pour la mise en œuvre de l’I.S..

L’objet de ce cours ne se veut pas un cours à proprement parlé sur le langage SysML, mais il est nécessaire d’introduire certaines notions et éléments de langage propre à celui-ci pour la bonne compréhension de la suite.

## D.2. Les diagrammes



*Les différents diagrammes SysML regroupés par catégories*

- 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'état** : montre les différents états du système et leurs modes de changement ;
  - **Diagramme d'activité** : montre l'enchaînement 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.

A ces trois grands groupes il convient d'y adjoindre le **diagramme de contexte**, non référencé en tant que tel dans la norme (il peut être fait avec un diagramme de définition de blocs ou de bloc interne), mais évoqué dans celle-ci et largement utilisé en pratique.

Une pratique largement répandue est d'utiliser des acronymes en lieu et place des noms des diagrammes (forme réduite bien pratique à l'usage). Les acronymes et symboles employés dans la suite de ce cours sont les suivants :

Nom français	English Name	Acronyme	Autre acronymes <sup>8</sup>	Symbole <sup>9</sup>
Diagramme d'exigences	Requirement Diagram	RD	Req	
Diagramme de cas d'utilisation	Use Case Diagram	UCD	Uc	
Diagramme de séquence	Sequence Diagram	SD	Seq	
Diagramme d'états/transitions	State Machine Diagram	SMD	Stm	
Diagramme d'activités	Activity Diagram	AD	Act	
Diagramme de définition de blocs	Block Diagram	BDD	-	
Diagramme de blocs internes	Internal Block Diagram	IBD	-	

### D.3. Les objets et les relations

SysML est un langage orienté objet, possédant chacun des attributs spécifiques. La plupart des objets sont représentés par un rectangle contenant :

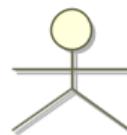
- Un entête : elle contient le stéréotype précisant la nature de l'objet (entre « ») ainsi que son nom ;
- Un (ou plusieurs) compartiment(s) : précisant certaines propriétés spécifiques (un numéro d'identifiant et un texte pour une exigence ; les éléments (parts), les valeurs, les propriétés, les méthodes (opérations) pour un bloc, ...)

Certains objets particuliers sont représentés par un symbole : c'est le cas des acteurs comme l'utilisateur (représenté par un bonhomme bâton, ou « sticky boy » ou stickman).

---

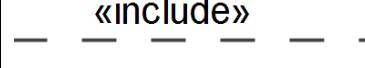
<sup>8</sup> Acronymes que l'on rencontre très fréquemment, qui dérivent des appellations largement utilisées par la plupart des solutions logicielles SysML (comme MagicDraw, voir note suivante). Néanmoins, ceux-ci constituent bien souvent un abus de langage.

<sup>9</sup> Ne servent qu'à titre d'identification rapide, et proviennent du logiciel MagicDraw [14] développé par la société NoMagic.

Exemples d'objets en langage SysML		
<pre>&lt;&lt;requirement&gt;&gt; Réveil automatique Id = "001" Text = "Le radio-réveil doit assurer à l'utilisateur un réveil automatique à l'heure souhaitée avec la radio ou un buzzer"</pre>	<pre>«block» Nom du bloc/composant propriété1 ... opération1() opération2()</pre>	 <p>Utilisateur</p>  <p>vent</p>
Une exigence	Un bloc	Des acteurs

Ces objets peuvent être en relation les uns les autres, par le biais d'un lien exprimant le type de relation entre les deux objets. Très souvent, il est basé sur une flèche (indiquant le sens de lecture de la relation), les relations se différenciant les unes des autres par :

- Le type de trait : plein ou en pointillé ;
- Le type de flèche : sans, avec flèche pleine ou ouverte ;
- Le symbole attaché à l'objet « parent », l'objet relié étant considéré comme l' « enfant » ;
- Un stéréotype éventuel : pour les liens en pointillé, précisant la nature de celui-ci.

Principaux liens entre objets en langage SysML <sup>10</sup>			
Symbol	Nom	Description	Remarque
	Contenance	L'objet de gauche contient l'objet de droite.	Concerne uniquement les exigences
	Association	Les deux objets sont associés entre eux	La nature de l'association peut être précisée sur le lien
	Composition	L'objet de gauche est composé de l'objet de droite.	Concerne principalement les blocs
	Inclusion	L'objet de gauche inclut l'objet de droite.	Concerne uniquement les cas d'utilisations
	Spécialisation/ Généralisation	L'objet de gauche est une spécialisation de l'objet de droite	Concerne tous les types d'objets

<sup>10</sup> Non valables pour les liens entre états et les liens entre activités, qui ont leur syntaxe et leur sémantique propre.

## E. L'I.S. : UNE DÉMARCHE

Dans toute la suite de ce paragraphe, nous illustrerons les différentes activités d'I.S. par la réalisation (au sens de la modélisation et non de la conception complète) d'un déclencheur de sécurité pour parachute, dont le besoin initial (issu du client) est le suivant :

**Expression du besoin initial** : Lors de sauts en chute libre, il arrive que le parachutiste ne puisse ouvrir par lui-même son parachute (dysfonctionnement du mécanisme d'ouverture) ou ne soit pas en mesure de le faire (évanouissement). La chute devient dès lors mortelle.

Un déclencheur de sécurité permettrait de palier à ces problèmes, et permettrait ainsi aux parachutistes de sauter en toute sécurité : il ne se déclencherait que si le parachutiste continue de chuter à une vitesse élevée à une altitude minimale jugée critique (hauteur nécessaire au temps d'ouverture du parachute et au freinage correct de la chute).

### E.1. Définition des besoins des parties prenantes (Analyse du besoin)

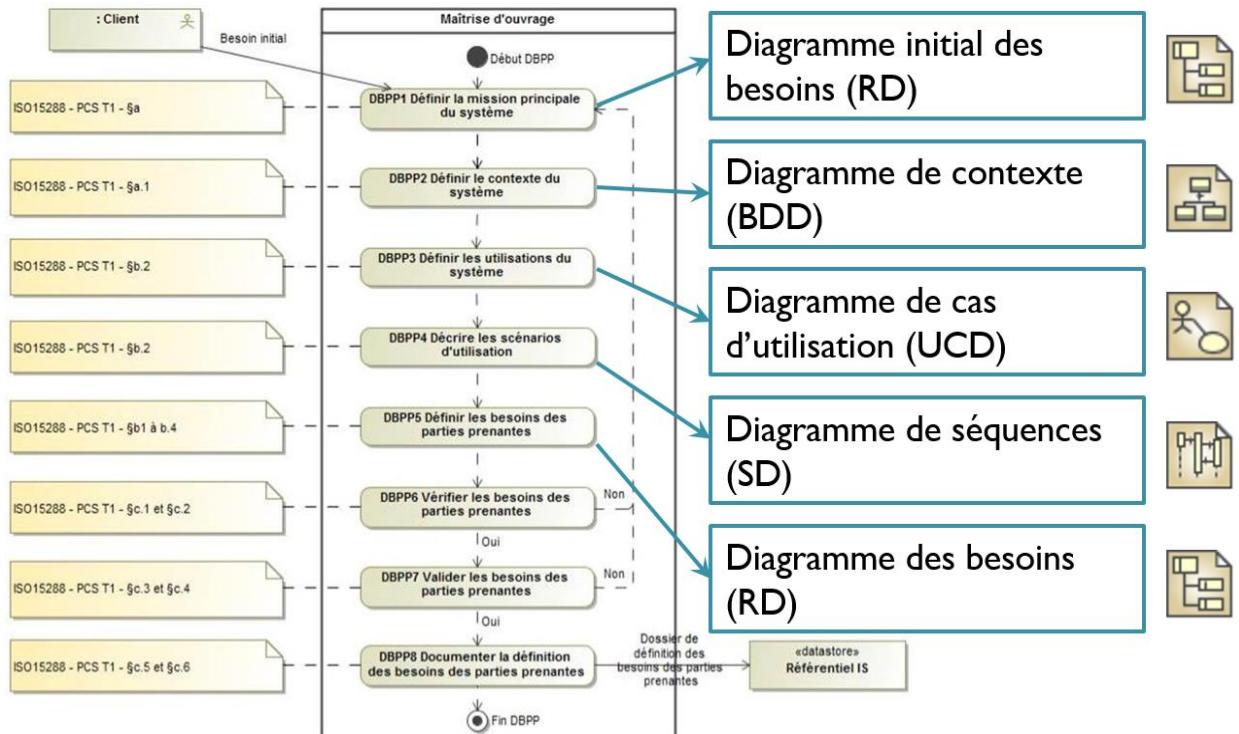
**Objet du processus** : Définir les besoins applicables à un système pour fournir, dans un environnement donné, les services dont les utilisateurs et les autres parties prenantes ont besoin.

Ce processus identifie les parties prenantes<sup>11</sup>, ou éventuellement des classes de parties, qui seront engagées vis-à-vis du système, durant son cycle de vie. Il identifie aussi leurs besoins et leurs souhaits.

Dans le respect de l'I.S.O 15288 [6], **UNE** démarche générale pour définir les besoins des parties prenantes peut s'établir de la façon suivante :

---

<sup>11</sup> Acteur, individuel ou collectif, activement ou passivement concerné par une décision ou un projet



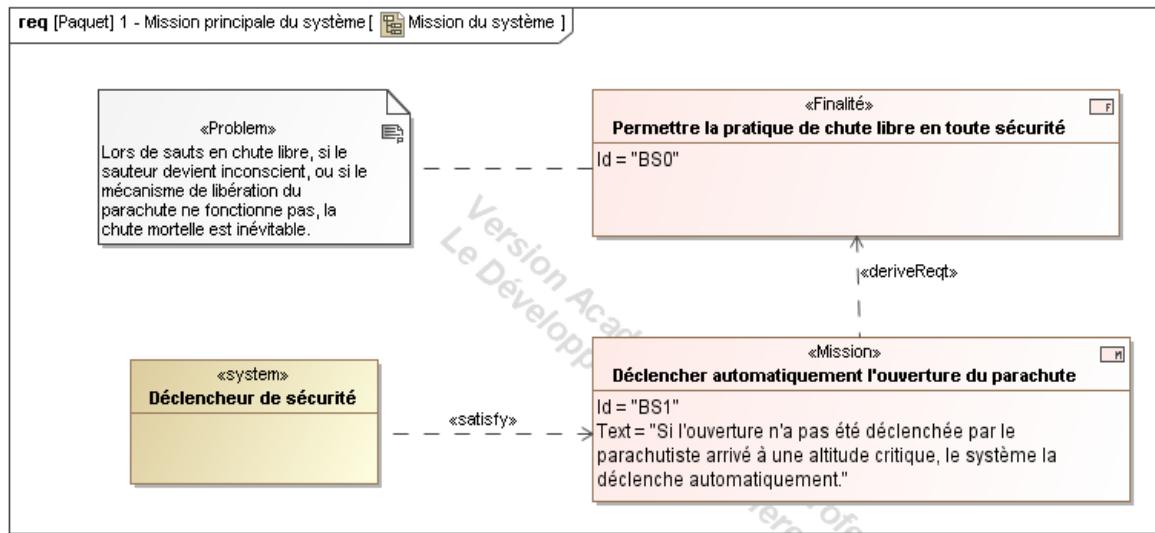
**Remarque :** Les diagrammes SysML proposés à titre d'exemple peuvent être et sont le plus souvent partiels (incomplets, non exhaustifs, ...).

#### E.1.1. Définir la mission principale du système

Pour débuter la définition des besoins des parties prenantes, une première analyse du besoin doit être menée pour définir la mission principale du système. Cette première analyse cadre globalement le système à faire.

Pour formaliser le résultat de cette activité de définition de la mission du système, on réalise un diagramme d'exigences (RD) :

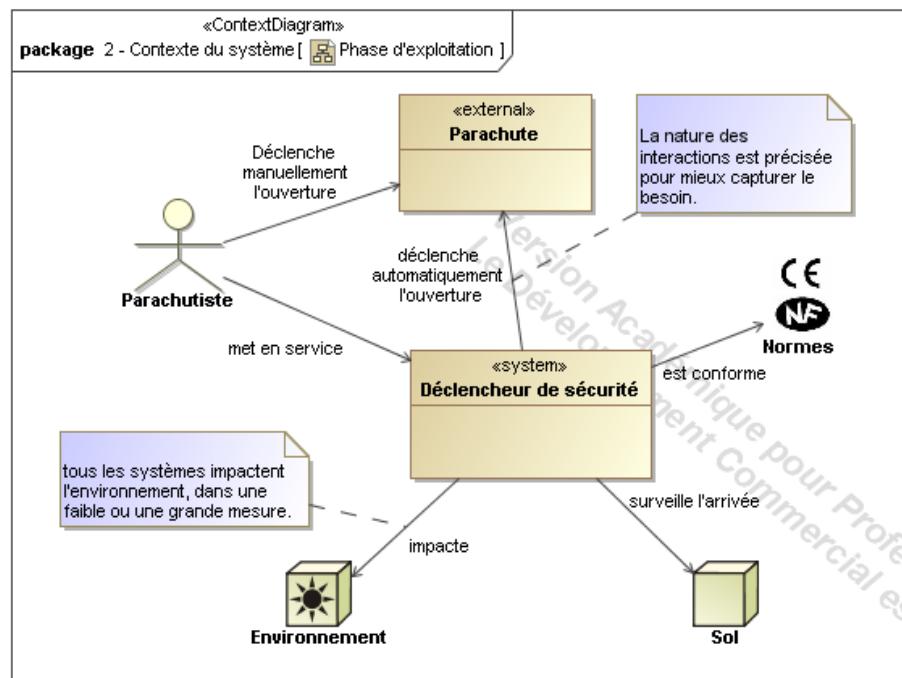
- Un système a une mission principale ou un service principal à remplir, on peut en déduire ses missions ou services (quoi) à remplir pour les parties prenantes ;
- La mission provient (dérive) d'une finalité ou d'une raison d'être (pourquoi) qui justifie la décision de réaliser un système relativement à la résolution d'un problème ;
- La mission peut être affinée si nécessaire (décomposée en sous-missions, objectifs ; si déjà identifiés dans le besoin initial).



### E.1.2. Définir le(s) contexte(s) du système

On identifie les parties prenantes du système étudié dans les contextes relatifs à ces différentes phases de vie (exploitation ou utilisation, conception, réalisation ou production, soutien, retrait ou fin de vie, ...).

Pour chaque phase du cycle de vie, les parties prenantes ayant un intérêt avec le système (quelles sont les parties prenantes pouvant exprimer des contraintes ou des besoins quand le système est dans cette phase du cycle de vie ?) sont identifiées dans des diagrammes de définition de bloc (BDD SysML).

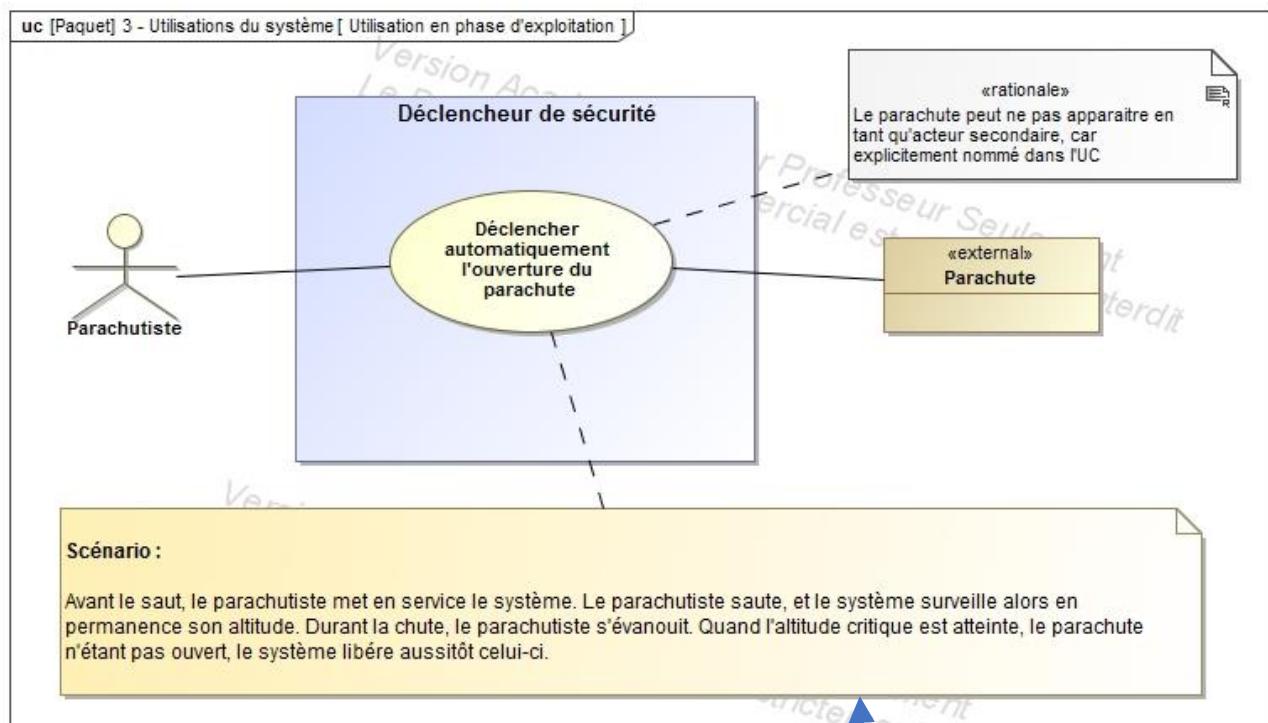


### E.1.3. Définir les utilisations du système

Le contexte étant défini, on s'attache à définir les fonctionnalités à réaliser pour chaque phase de vie où le système doit répondre à des besoins de type service. Certaines phases de vie ne donnent pas toujours lieu à la mise en place de service pour le système (conception, maintenance, ...).

La mission principale du système se retrouve souvent dans le cas d'utilisation principal de la phase exploitation.

Pour formaliser les différentes utilisations du système, on réalise, en fonction de la complexité, au moins un diagramme de cas d'utilisation (UCD) pour chacune des phases de vie où des services sont attendus du système.



**Remarque :** Les acteurs présents dans ce diagramme sont repris du diagramme de contexte, ou en d'autres termes tous les acteurs présents dans un UCD doivent l'être dans le contexte (et non réciproquement).

#### E.1.4. Décrire les scénarios d'utilisation

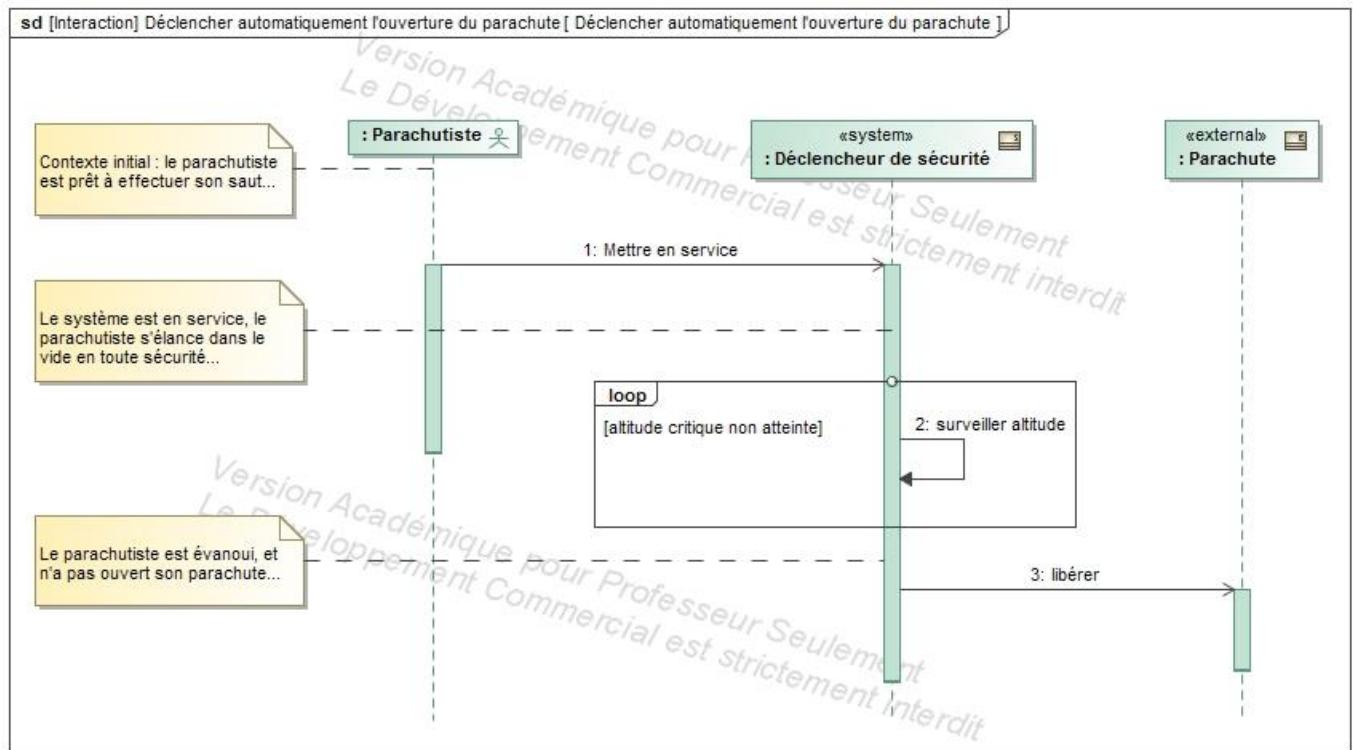
Pour préciser les différentes utilisations du système (représentées par les cas d'utilisation précédents), on réalise pour chaque cas d'utilisation une description textuelle sous la forme d'un scénario.

La description textuelle précise en général :

- Le contexte (opérationnel) ;
- Les actions et interactions ;
- Leurs enchaînements et conditions éventuelles ;
- Les acteurs qui font l'action ;
- Les données en entrée et en sortie ;
- Les options éventuelles.

Le scénario ainsi décrit peut ensuite être traduit en diagramme de séquence (SD). Un scénario nominal est suffisant (scénario de réussite) mais, selon les cas, on pourra être amené à effectuer différents scénarios (comme un (ou des) scénario(s) d'échec), le but n'étant pas d'être le plus exhaustif possible.

Ainsi, le scénario ci-dessus donnerait sous forme de diagramme de séquence :



**Remarque :** la description textuelle du scénario est importante à plus d'un égard car :

- Elle permet de s'affranchir dans un premier temps de l'outil SysML ;
- Elle permet de vérifier très vite que le cas d'utilisation en est bien un (si on est incapable d'écrire au moins un scénario, c'est que le cas d'utilisation n'en est pas un) ;
- Même avec une pratique accrue, elle reste un moyen sûr de bien décrire un cas d'utilisation tout en restant dans l'espace du problème (remarquez les termes employés ici...), avant d'envisager sa traduction en séquence.

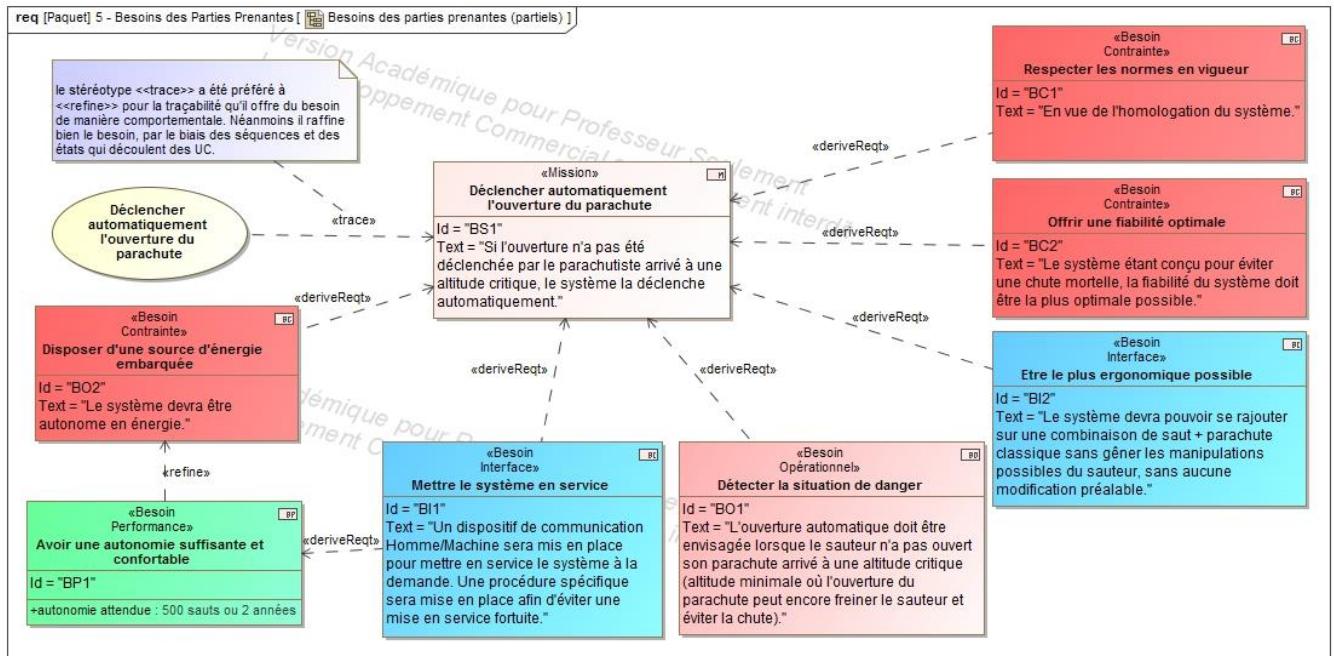
#### E.1.5. Définir les besoins des parties prenantes

On peut alors définir pour chacune des phases de vie du système les besoins des parties prenantes, à partir des éléments du besoin initial : mission, finalité, besoins, contraintes ; complétés sur la base des analyses précédentes : étude des services attendus (diagramme de cas d'utilisation), étude du contexte (diagramme de définition de bloc).

Afin d'aider à la définition des besoins, ceux-ci sont classés de la façon suivante :

- Service attendu (la mission en fait partie, et tous cas d'utilisation) ;
- Opérationnel (mode de fonctionnement, modes de marche, condition d'évolution, ...) ;
- Performance;
- Interface (physique, ergonomie, interopérabilité, ...) ;
- Contrainte (liée à une phase de vie, environnement du système, réglementation, coût, ...).

Pour formaliser les besoins des parties prenantes, on réalise, en fonction de la complexité, au moins un diagramme d'exigences (RD) pour chacune des phases de vie.



Les besoins ainsi définis, constituent ce qu'on appelle le cahier des charges fonctionnel :

**Cahier des charges fonctionnel (CDCF)** : document ayant une structure normalisée et formalisant ce dont le client a besoin ainsi que l'ensemble de ses requêtes, le tout sans spécifier la solution technique.

## E.1.6. Vérifier, valider et documenter la Définition des Besoins des Parties Prenantes

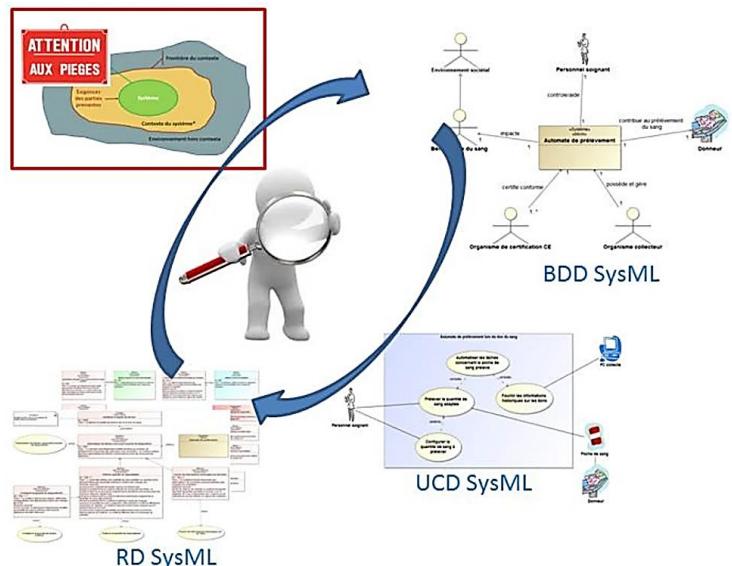
### ➤ Vérification au fil de l'eau et à la fin du processus

Le résultat du travail doit être vérifié tout au long de son avancement, ainsi que de façon plus globale à la fin dans une étape spécifique de vérification.

#### Pièges à éviter :

Concernant la définition des frontières du système, il faut éviter certains pièges classiques :

- Mal cibler le périmètre, élément ajoutés ou oubliés dans le système (exemple des opérateurs) ;
- Se tromper de système, en se trompant de niveau (sur-système, constituants).



### ➤ Valider les besoins des parties prenantes

Les résultats des travaux réalisés doivent être transmis au client pour qu'il puisse statuer sur la bonne compréhension et formalisation de son problème. Si nécessaire le travail d'analyse doit être repris jusqu'à obtenir un consensus avec le client.



### ➤ Documenter la définition des besoins des parties prenantes

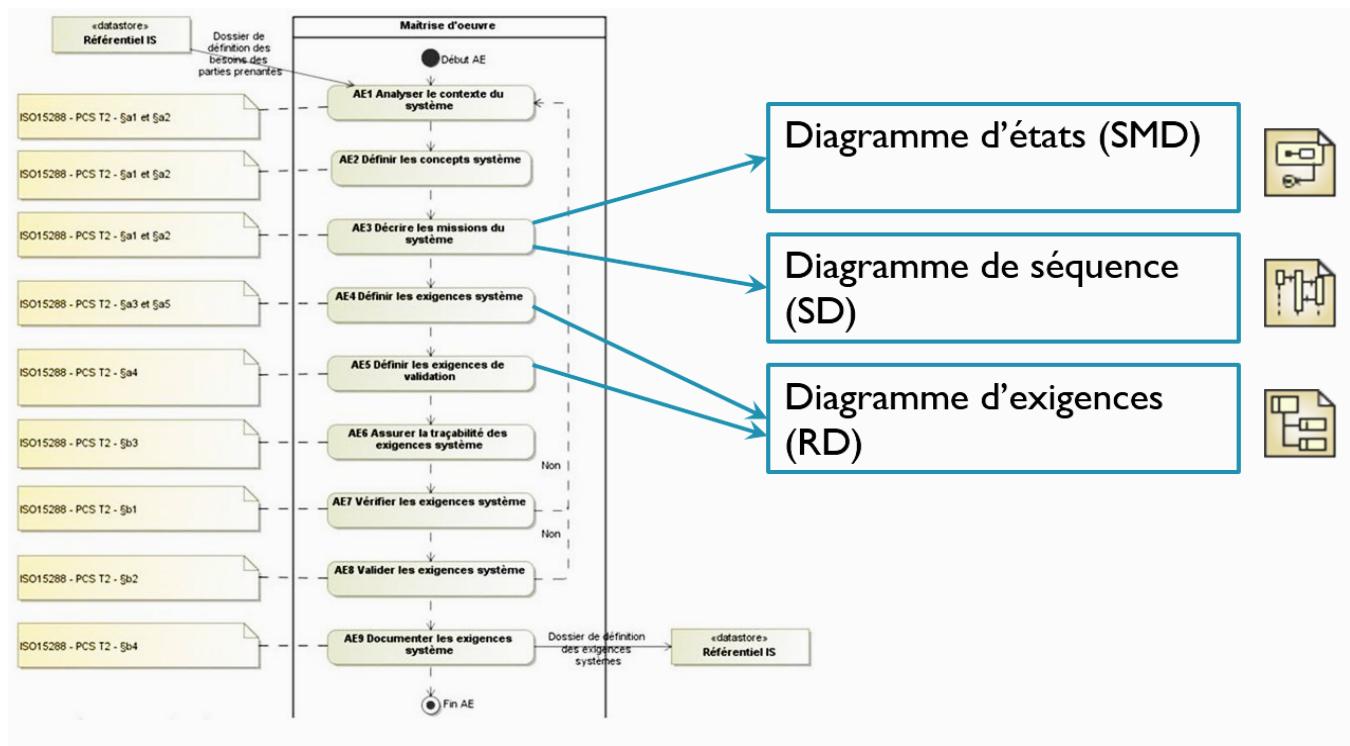
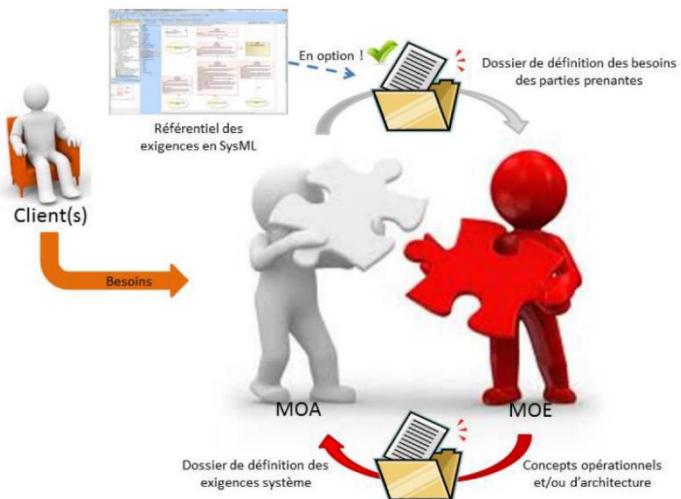
En guise de synthèse du processus de définition des besoins des parties prenantes et après validation par le client, il est nécessaire d'enregistrer convenablement les besoins obtenus. Un document de définition des besoins des parties prenantes est alors élaboré (formalisation du CDCF, référentiel I.S.).

#	Id	Nom	Text
1	BC1	Respecter les normes en vigueur	En vue de l'homologation du système.
2	BC2	Offrir une fiabilité optimale	Le système étant conçu pour éviter une chute mortelle, la fiabilité du système doit être la plus optimale possible.
3	BI1	Mettre le système en service	Un dispositif de communication Homme/Machine sera mis en place pour mettre en service le système à la demande. Une procédure spécifique sera mise en place afin d'éviter une mise en service fortuite.
4	BI2	Etre le plus ergonomique possible	Le système devra pouvoir se rapprocher sur une combinaison de saut + parachute classique sans gêner les manipulations possibles du sauteur, sans aucune modification préalable.
5	BO1	Déetecter la situation de danger	L'ouverture automatique doit être envisagée lorsque le sauteur n'a pas ouvert son parachute arrivé à une altitude critique (altitude minimale où l'ouverture du parachute peut encore freiner le sauteur et éviter la chute).
6	BO2	Disposer d'une source d'énergie embarquée	Le système devra être autonome en énergie.
7	BP1	Avoir une autonomie suffisante et confortable	
8	BS0	Permettre la pratique de chute libre en toute sécurité	
9	BS1	Déclencher automatiquement l'ouverture du parachute	Si l'ouverture n'a pas été déclenchée par le parachutiste arrivé à une altitude critique, le système la déclenche automatiquement.
10	BS2	Ouvrir le parachute à la demande	Afin de laisser la possibilité au parachutiste d'ouvrir son parachute quand il le souhaite.

## E.2. Analyse des exigences

**Objet du processus :** L'analyse des exigences vise à spécifier ce que le système doit faire pour répondre aux besoins des parties prenantes.

Démarche générale :



### E.2.1. Analyser le contexte du système

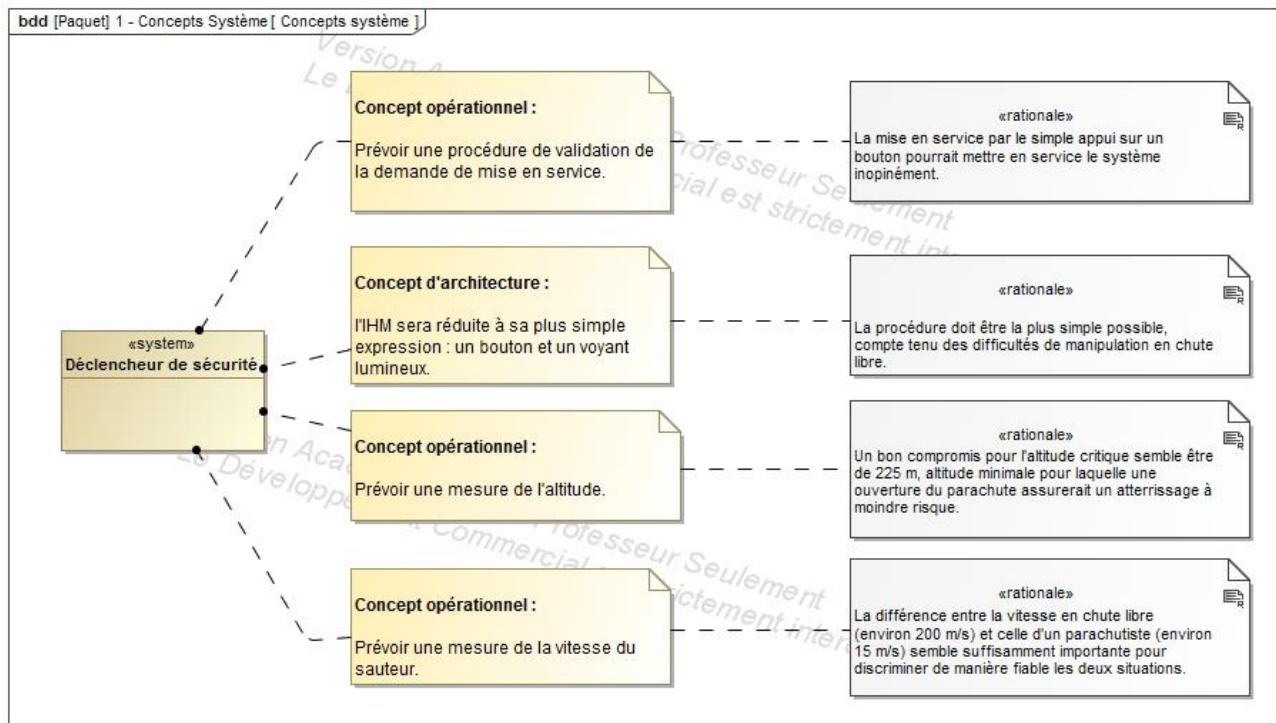
La maîtrise d'œuvre en charge de l'analyse reprend les documents fournis par la MOA, pour éventuellement les compléter avec le point de vue de la MOE.

### E.2.2. Définir les concepts système

La MOE peut sur la base de son expertise et de sa compréhension du problème introduire d'éventuels concepts système.

Les concepts système peuvent être :

- Des **concepts opérationnels**, comme par exemple « Réaliser la mission en ne réalisant qu'un seul déplacement » ou « Réaliser la mission en totale autonomie énergétique » ;
- Des **concepts d'architecture** logique ou physique, comme par exemple « Mettre en œuvre une architecture logique conforme à la norme X.Y » ou « Utiliser un robot mobile ».



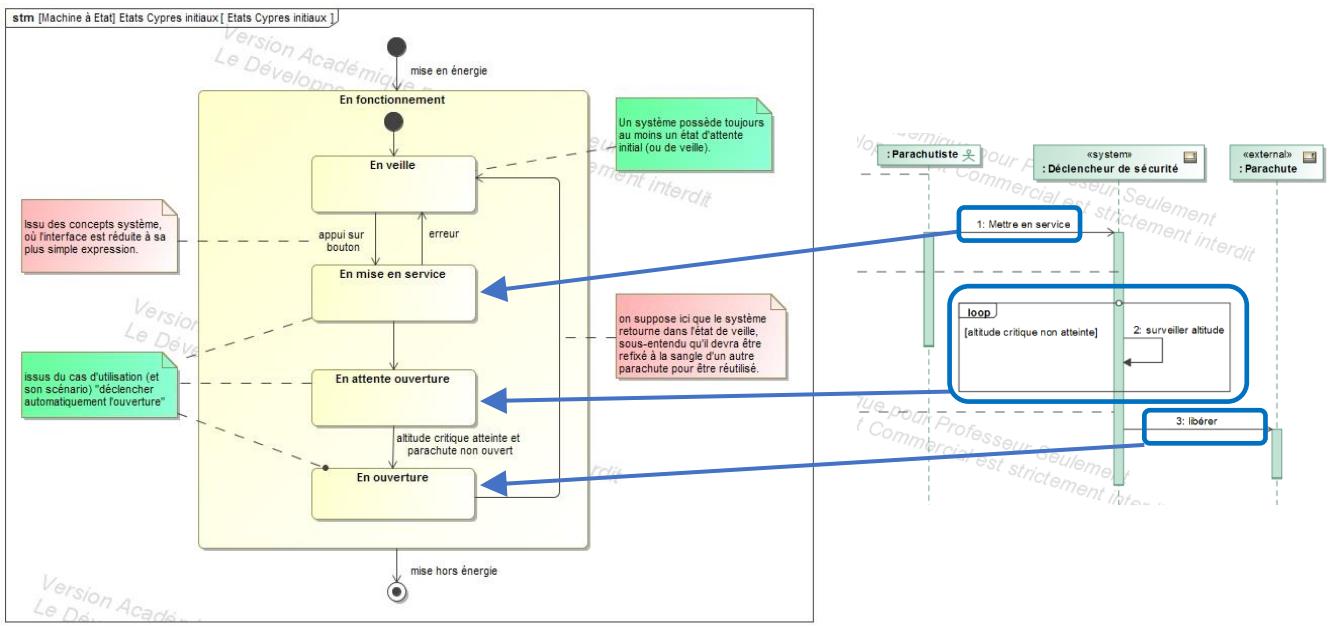
**Remarque :** attention à rester le plus possible dans l'espace du problème. Les exigences système prescrivent ce que le système doit faire et non comment il doit le faire !

#### E.2.3. Décrire les missions du système

Pour débuter l'analyse des exigences système, on doit décrire les missions du système. Le système est vu comme une boîte noire.

L'étape consiste à identifier les états du système. Il est possible sur la base des cas d'utilisations et des modes opératoires associés de définir les états du système et les conditions de passage de l'un à l'autre.

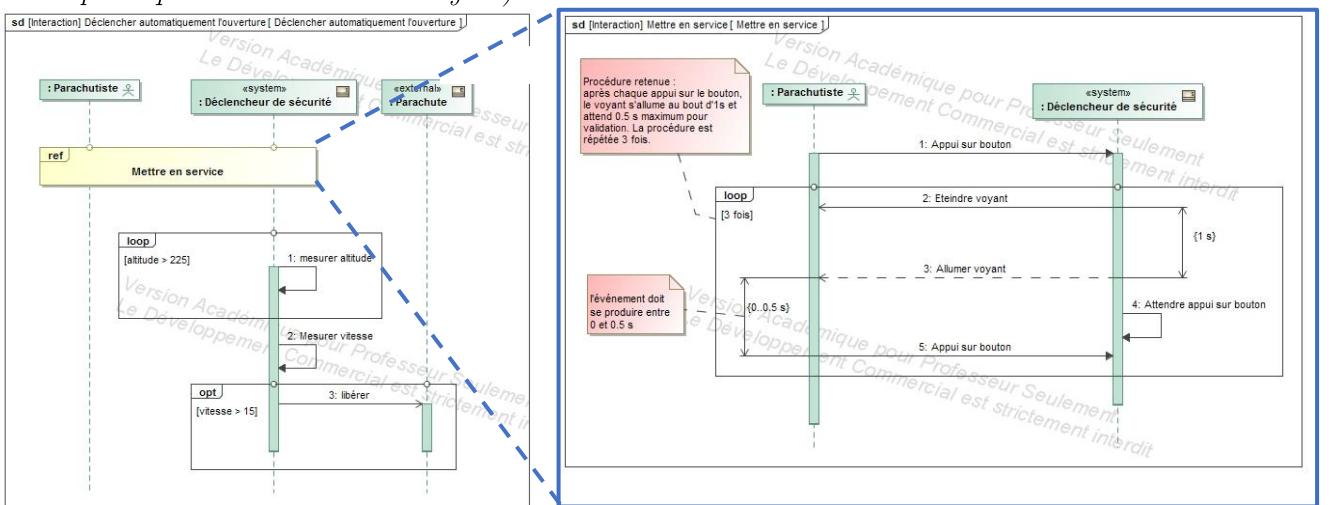
- **Identifier les états initiaux :** dans un premier temps, et afin de réaliser un diagramme d'état « système », chaque cas d'utilisation déjà identifié peut-être formalisé.



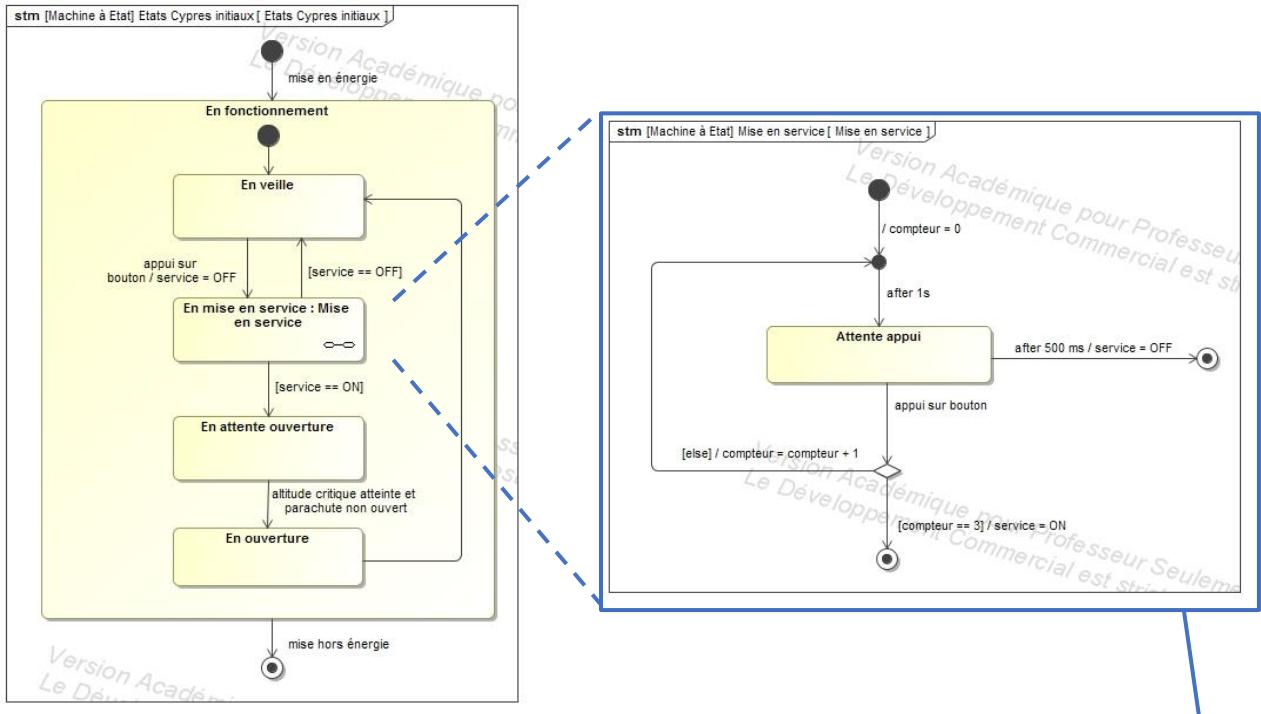
**Remarque :** Les états supplémentaires comme « En veille » sont à définir. Ils dépendent largement du savoir-faire de la MOE. Il ne faut pas négliger l'aspect itératif de la démarche qui amène à remettre en cause autant que nécessaire l'ensemble des modèles.

- **Décrire précisément les scénarios** : les scénarios des cas d'utilisation sont formalisés en diagrammes de séquence (si tel n'était pas le cas), enrichis des concepts systèmes apportés par la MOE.

*Raffinement du scénario : la procédure de validation retenue pour la mise en service sera une succession de 3 appuis réguliers, à l'allumage du voyant (l'appui sur le bouton provoquant l'extinction du voyant).*



- **Raffiner les états sur la base des scénarios** : L'étape suivante consiste à décrire des états plus élémentaires du système. A partir des diagrammes de séquence, il est possible de définir des états et des conditions d'évolution associées (messages du diagramme de séquence, ...) en identifiant les périodes d'activité ou d'inactivité du système.



**A noter :** La flèche entre deux états s'appelle une transition, dont la syntaxe associée précise le franchissement. Elle s'exprime sous la forme :

*événement [condition de garde] / effet*

La transition est validée quand l'**événement** se produit ET que la **condition de garde** est vérifiée (évaluation d'une expression booléenne). L'**effet** est produit durant le franchissement.

**Remarque :** les diagrammes d'états seront étudiés plus en détail dans le chapitre sur les « systèmes à événements discrets ».

#### E.2.4. Définir les exigences système

**Exigence (selon[8]) :** Une exigence prescrit une propriété jugée nécessaire. Son énoncé peut être :

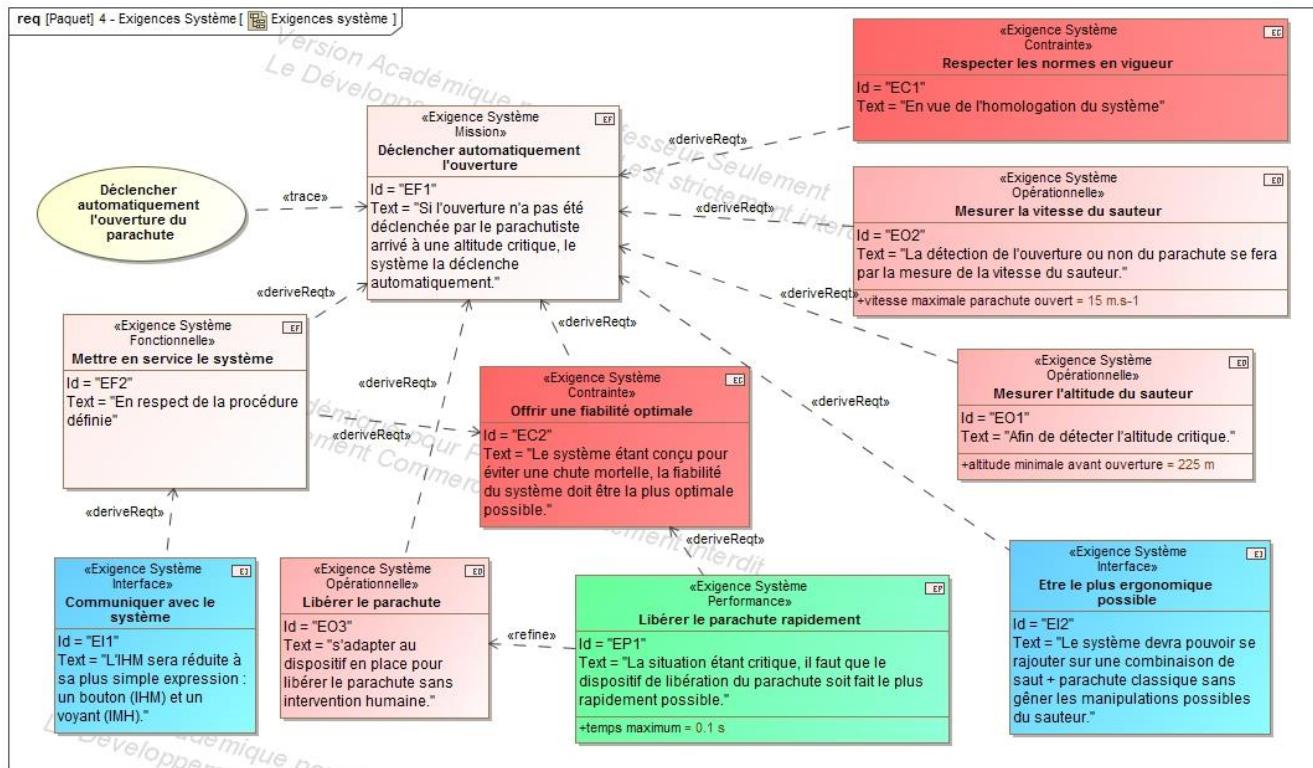
- un service ou une fonction (Transporter ...);
- une caractéristique (Tenir dans un volume de ...);
- une aptitude (Supporter une température de ...);
- ou une limitation (Utiliser uniquement ...);

À laquelle doit satisfaire un système, un produit ou un processus.

Les exigences sont décrites par différents attributs : un nom, un identifiant, une description textuelle.

Afin d'aider à la définition des exigences système et de faciliter leur lecture, on peut préciser la nature de l'exigence (fonctionnelle, opérationnelle, d'interface, de performance, de contrainte, de validation).

La définition des exigences systèmes est basée sur l'ensemble des documents issus de la MOA et produits par la MOE, et leur analyse.



**Remarque :** très souvent, les diagrammes d'exigence des sujets de concours seront réduits à leur plus simple expression, sans stéréotype particulier ni même parfois texte descriptif, et présenté bien souvent sous forme d'arborescence (horizontale ou verticale).

#### E.2.5. Assurer la traçabilité des exigences système

Chaque besoin exprimé dans le document des besoins des parties prenantes doit se retrouver exprimé (tracé), voire précisé, par au moins une exigence système.

La plupart des solutions logicielles permettent d'établir de telles relations sous forme matricielle.

	BC1	BC2	BC3	BC4	BC5	BC6	BC7	BC8	BC9	BC10	BC11	BC12	BC13	BC14	BC15	BC16	BC17	BC18	BC19	BC20	BC21	BC22	BC23	BC24	BC25	BC26	BC27	BC28	BC29	BC30	BC31	BC32	BC33	BC34	BC35	BC36	BC37	BC38	BC39	BC40	BC41	BC42	BC43	BC44	BC45	BC46	BC47	BC48	BC49	BC50	BC51	BC52	BC53	BC54	BC55	BC56	BC57	BC58	BC59	BC60	BC61	BC62	BC63	BC64	BC65	BC66	BC67	BC68	BC69	BC70	BC71	BC72	BC73	BC74	BC75	BC76	BC77	BC78	BC79	BC80	BC81	BC82	BC83	BC84	BC85	BC86	BC87	BC88	BC89	BC90	BC91	BC92	BC93	BC94	BC95	BC96	BC97	BC98	BC99	BC100	BC101	BC102	BC103	BC104	BC105	BC106	BC107	BC108	BC109	BC110	BC111	BC112	BC113	BC114	BC115	BC116	BC117	BC118	BC119	BC120	BC121	BC122	BC123	BC124	BC125	BC126	BC127	BC128	BC129	BC130	BC131	BC132	BC133	BC134	BC135	BC136	BC137	BC138	BC139	BC140	BC141	BC142	BC143	BC144	BC145	BC146	BC147	BC148	BC149	BC150	BC151	BC152	BC153	BC154	BC155	BC156	BC157	BC158	BC159	BC160	BC161	BC162	BC163	BC164	BC165	BC166	BC167	BC168	BC169	BC170	BC171	BC172	BC173	BC174	BC175	BC176	BC177	BC178	BC179	BC180	BC181	BC182	BC183	BC184	BC185	BC186	BC187	BC188	BC189	BC190	BC191	BC192	BC193	BC194	BC195	BC196	BC197	BC198	BC199	BC200	BC201	BC202	BC203	BC204	BC205	BC206	BC207	BC208	BC209	BC210	BC211	BC212	BC213	BC214	BC215	BC216	BC217	BC218	BC219	BC220	BC221	BC222	BC223	BC224	BC225	BC226	BC227	BC228	BC229	BC230	BC231	BC232	BC233	BC234	BC235	BC236	BC237	BC238	BC239	BC240	BC241	BC242	BC243	BC244	BC245	BC246	BC247	BC248	BC249	BC250	BC251	BC252	BC253	BC254	BC255	BC256	BC257	BC258	BC259	BC260	BC261	BC262	BC263	BC264	BC265	BC266	BC267	BC268	BC269	BC270	BC271	BC272	BC273	BC274	BC275	BC276	BC277	BC278	BC279	BC280	BC281	BC282	BC283	BC284	BC285	BC286	BC287	BC288	BC289	BC290	BC291	BC292	BC293	BC294	BC295	BC296	BC297	BC298	BC299	BC300	BC301	BC302	BC303	BC304	BC305	BC306	BC307	BC308	BC309	BC310	BC311	BC312	BC313	BC314	BC315	BC316	BC317	BC318	BC319	BC320	BC321	BC322	BC323	BC324	BC325	BC326	BC327	BC328	BC329	BC330	BC331	BC332	BC333	BC334	BC335	BC336	BC337	BC338	BC339	BC340	BC341	BC342	BC343	BC344	BC345	BC346	BC347	BC348	BC349	BC350	BC351	BC352	BC353	BC354	BC355	BC356	BC357	BC358	BC359	BC360	BC361	BC362	BC363	BC364	BC365	BC366	BC367	BC368	BC369	BC370	BC371	BC372	BC373	BC374	BC375	BC376	BC377	BC378	BC379	BC380	BC381	BC382	BC383	BC384	BC385	BC386	BC387	BC388	BC389	BC390	BC391	BC392	BC393	BC394	BC395	BC396	BC397	BC398	BC399	BC400	BC401	BC402	BC403	BC404	BC405	BC406	BC407	BC408	BC409	BC410	BC411	BC412	BC413	BC414	BC415	BC416	BC417	BC418	BC419	BC420	BC421	BC422	BC423	BC424	BC425	BC426	BC427	BC428	BC429	BC430	BC431	BC432	BC433	BC434	BC435	BC436	BC437	BC438	BC439	BC440	BC441	BC442	BC443	BC444	BC445	BC446	BC447	BC448	BC449	BC450	BC451	BC452	BC453	BC454	BC455	BC456	BC457	BC458	BC459	BC460	BC461	BC462	BC463	BC464	BC465	BC466	BC467	BC468	BC469	BC470	BC471	BC472	BC473	BC474	BC475	BC476	BC477	BC478	BC479	BC480	BC481	BC482	BC483	BC484	BC485	BC486	BC487	BC488	BC489	BC490	BC491	BC492	BC493	BC494	BC495	BC496	BC497	BC498	BC499	BC500	BC501	BC502	BC503	BC504	BC505	BC506	BC507	BC508	BC509	BC510	BC511	BC512	BC513	BC514	BC515	BC516	BC517	BC518	BC519	BC520	BC521	BC522	BC523	BC524	BC525	BC526	BC527	BC528	BC529	BC530	BC531	BC532	BC533	BC534	BC535	BC536	BC537	BC538	BC539	BC540	BC541	BC542	BC543	BC544	BC545	BC546	BC547	BC548	BC549	BC550	BC551	BC552	BC553	BC554	BC555	BC556	BC557	BC558	BC559	BC560	BC561	BC562	BC563	BC564	BC565	BC566	BC567	BC568	BC569	BC570	BC571	BC572	BC573	BC574	BC575	BC576	BC577	BC578	BC579	BC580	BC581	BC582	BC583	BC584	BC585	BC586	BC587	BC588	BC589	BC590	BC591	BC592	BC593	BC594	BC595	BC596	BC597	BC598	BC599	BC600	BC601	BC602	BC603	BC604	BC605	BC606	BC607	BC608	BC609	BC610	BC611	BC612	BC613	BC614	BC615	BC616	BC617	BC618	BC619	BC620	BC621	BC622	BC623	BC624	BC625	BC626	BC627	BC628	BC629	BC630	BC631	BC632	BC633	BC634	BC635	BC636	BC637	BC638	BC639	BC640	BC641	BC642	BC643	BC644	BC645	BC646	BC647	BC648	BC649	BC650	BC651	BC652	BC653	BC654	BC655	BC656	BC657	BC658	BC659	BC660	BC661	BC662	BC663	BC664	BC665	BC666	BC667	BC668	BC669	BC670	BC671	BC672	BC673	BC674	BC675	BC676	BC677	BC678	BC679	BC680	BC681	BC682	BC683	BC684	BC685	BC686	BC687	BC688	BC689	BC690	BC691	BC692	BC693	BC694	BC695	BC696	BC697	BC698	BC699	BC700	BC701	BC702	BC703	BC704	BC705	BC706	BC707	BC708	BC709	BC710	BC711	BC712	BC713	BC714	BC715	BC716	BC717	BC718	BC719	BC720	BC721	BC722	BC723	BC724	BC725	BC726	BC727	BC728	BC729	BC730	BC731	BC732	BC733	BC734	BC735	BC736	BC737	BC738	BC739	BC740	BC741	BC742	BC743	BC744	BC745	BC746	BC747	BC748	BC749	BC750	BC751	BC752	BC753	BC754	BC755	BC756	BC757	BC758	BC759	BC760	BC761	BC762	BC763	BC764	BC765	BC766	BC767	BC768	BC769	BC770	BC771	BC772	BC773	BC774	BC775	BC776	BC777	BC778	BC779	BC780	BC781	BC782	BC783	BC784	BC785	BC786	BC787	BC788	BC789	BC790	BC791	BC792	BC793	BC794	BC795	BC796	BC797	BC798	BC799	BC800	BC801	BC802	BC803	BC804	BC805	BC806	BC807	BC808	BC809	BC810	BC811	BC812	BC813	BC814	BC815	BC816	BC817	BC818	BC819	BC820	BC821	BC822	BC823	BC824	BC825	BC826	BC827	BC828	BC829	BC830	BC831	BC832	BC833	BC834	BC835	BC836	BC837	BC838	BC839	BC840	BC841	BC842	BC843	BC844	BC845	BC846	BC847	BC848	BC849	BC850	BC851	BC852	BC853	BC854	BC855	BC856	BC857	BC858	BC859	BC860	BC861	BC862	BC863	BC864	BC865	BC866	BC867	BC868	BC869	BC870	BC871	BC872	BC873	BC874	BC875	BC876	BC877	BC878	BC879	BC880	BC881	BC882	BC883	BC884	BC885	BC886	BC887	BC888	BC889	BC890	BC891	BC892	BC893	BC894	BC895	BC896	BC897	BC898	BC899	BC900	BC901	BC902	BC903	BC904	BC905	BC906	BC907	BC908	BC909	BC910	BC911	BC912	BC913	BC914	BC915	BC916	BC917	BC918	BC919	BC920	BC921	BC922	BC923	BC924	BC925	BC926	BC927	BC928	BC929	BC930	BC931	BC932	BC933	BC934	BC935	BC936	BC937	BC938	BC939	BC940	BC941	BC942	BC943	BC944	BC945	BC946	BC947	BC948	BC949	BC950	BC951	BC952	BC953	BC954	BC955	BC956	BC957	BC958	BC959	BC960	BC961	BC962	BC963	BC964	BC965	BC966	BC967	BC968	BC969	BC970	BC971	BC972	BC973	BC974	BC975	BC976	BC977	BC978	BC979	BC980	BC981	BC982	BC983	BC984	BC985	BC986	BC987	BC988	BC989	BC990	BC991	BC992	BC993	BC994	BC995	BC996	BC997	BC998	BC999	BC9999

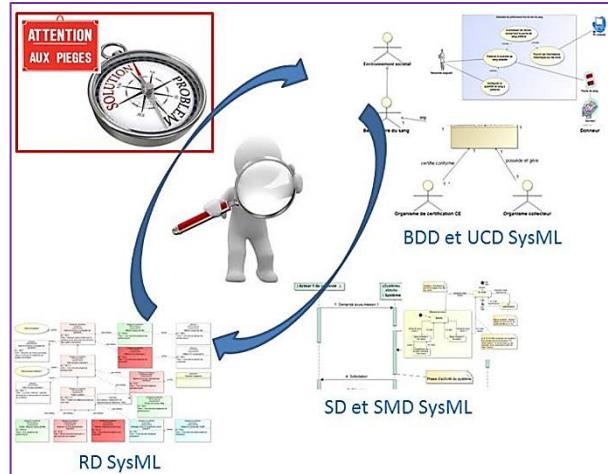
## E.2.6. Vérifier, valider et documenter les exigences système

### ➤ Vérification au fil de l'eau et à la fin du processus

Le résultat du travail doit être vérifié tout au long de son avancement, ainsi que de façon plus globale à la fin dans une étape spécifique de vérification.

#### Pièges à éviter :

Le piège le plus classique à ce niveau est de trop rapidement définir une solution au problème posé par les parties prenantes. Il est essentiel de rester au niveau de la prescription du résultat attendu (exprimé par les besoins).



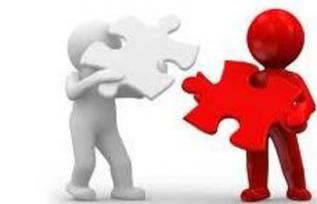
### ➤ Valider les exigences système

Comme lors de la phase de définition des besoins des parties prenantes, les résultats des travaux réalisés doivent être transmis à la maîtrise d'ouvrage (ou au client) pour qu'elle puisse statuer sur la bonne compréhension et formalisation de ses besoins et que les compromis éventuels soient acceptables.



### ➤ Documenter la définition des exigences système

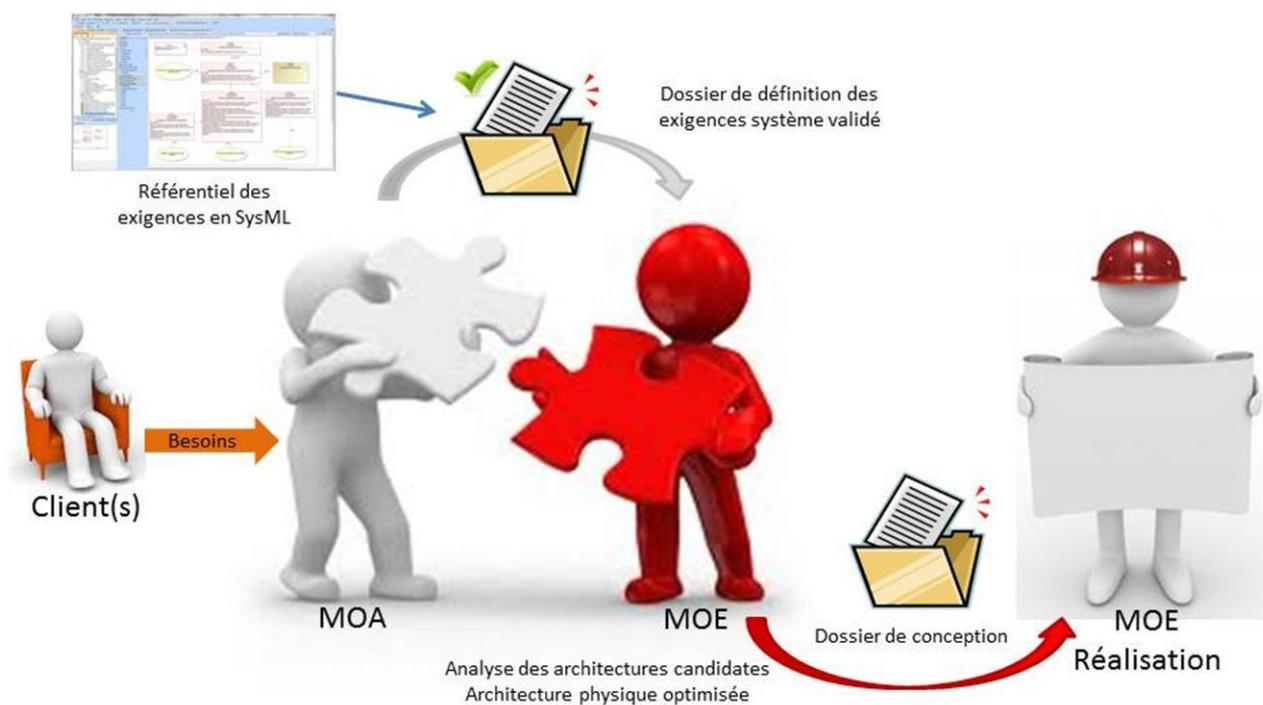
En guise de synthèse du processus de d'analyse des exigences et après validation par le client, il est nécessaire d'enregistrer convenablement les exigences système obtenues. Un document de définition des exigences système est alors élaboré (spécification fonctionnelle du système, référentiel I.S.).



**Pour réaliser la conception de l'architecture du système**

### E.3. Conception de l'architecture

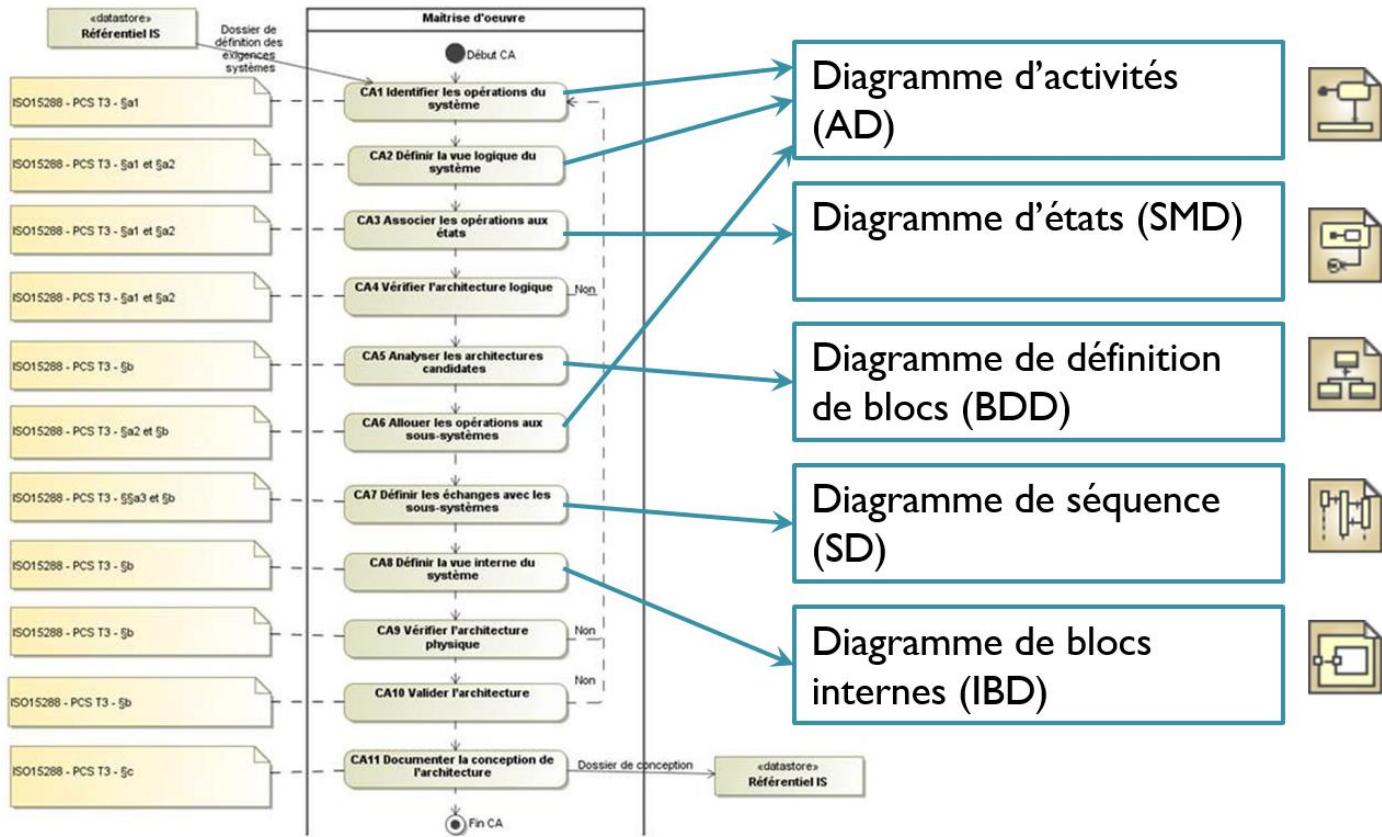
**Objet du processus :** Apporter une solution architecturale au(x) problème(s) étudié(s). La solution s'appuie sur des constituants qui pourront être faits, sous-traités, réutilisés avec ou sans modification, achetés « disponibles sur étagère ».



Ce processus est central dans la conception d'un système. Pour ce faire, il est nécessaire de définir une architecture logique puis une architecture physique en réponse aux exigences système exprimées pour satisfaire les besoins des parties prenantes.

En basculant dans l'espace de la solution, la conception de l'architecture décrit comment le système doit faire pour satisfaire les besoins des parties prenantes.

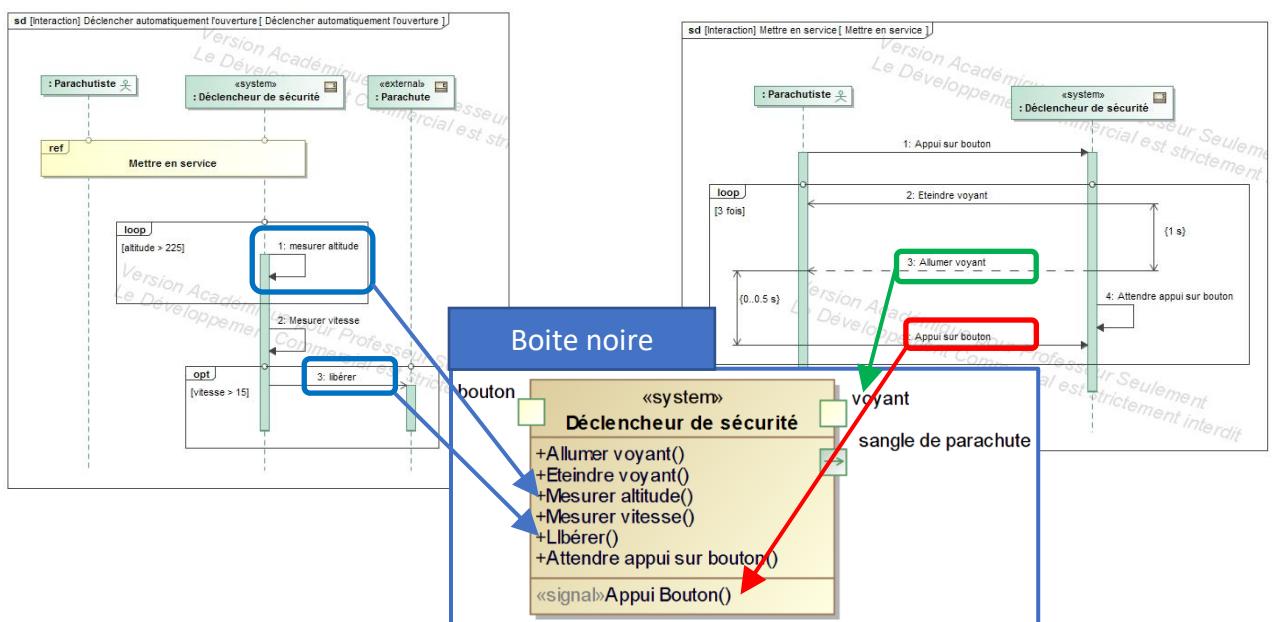
Démarche générale :



**Remarque :** dans la suite de ce document nous utilisons le terme « opération » et non « fonction » pour éviter toute confusion. La notion d'opération est ici prise au sens de SysML, elle permet donc d'identifier des comportements du système.

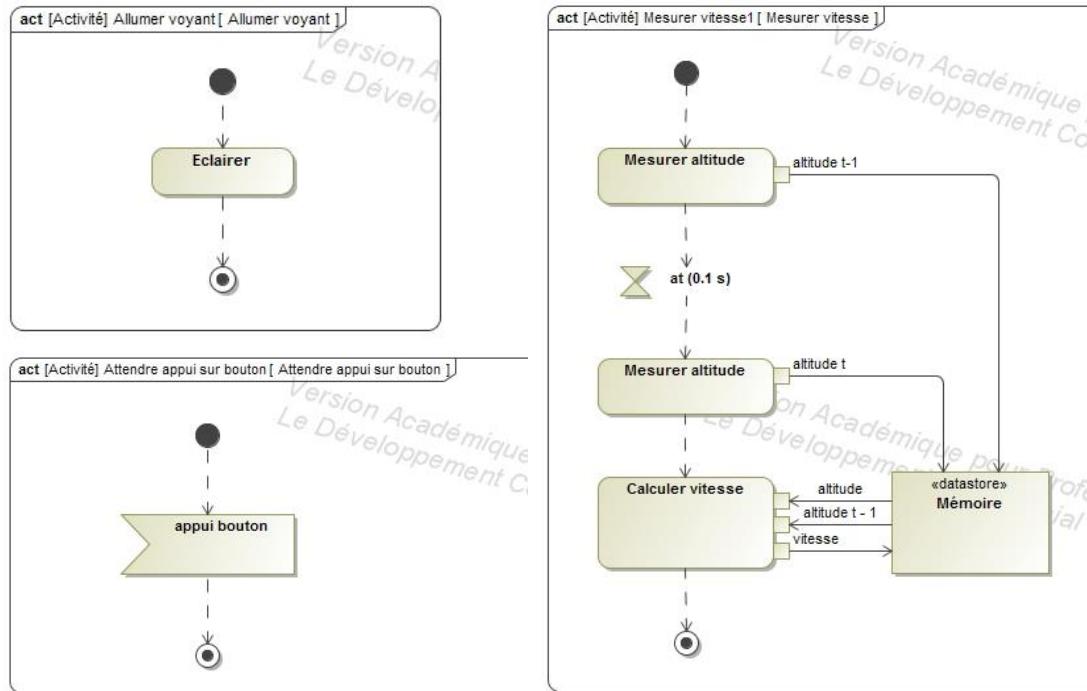
### E.3.1. Identifier les opérations du système

A partir des scénarios associés à chaque cas d'utilisation et décrits par des diagrammes de séquence, on fait apparaître les opérations, les signaux et les ports du système.



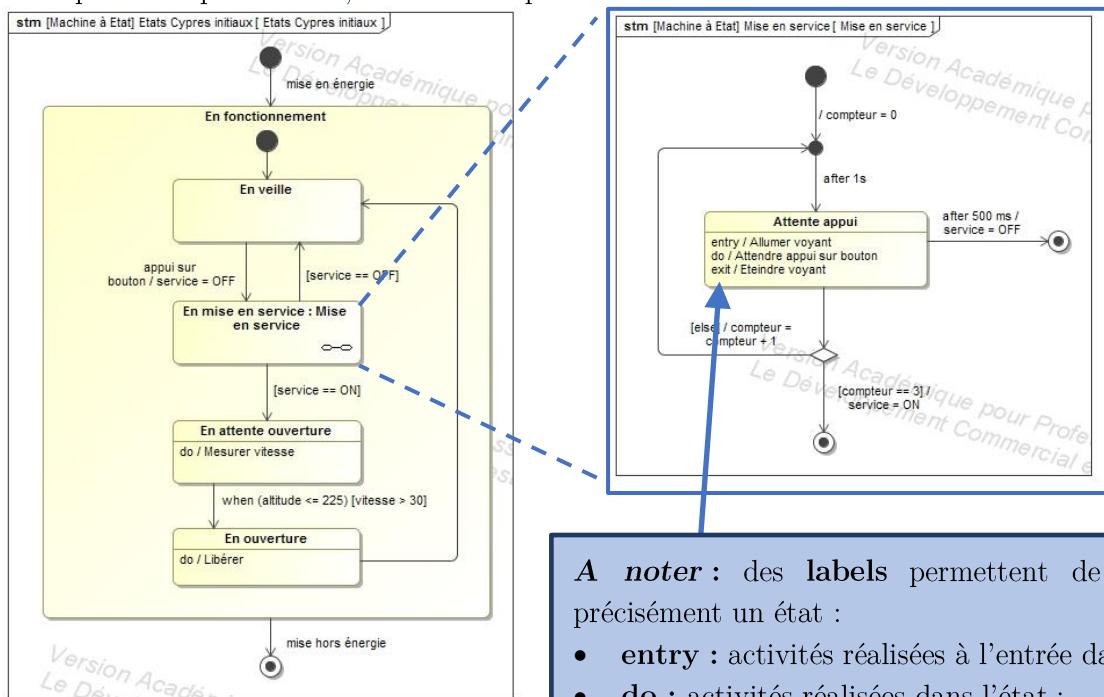
### E.3.2. Définir la vue logique interne

Pour chaque opération identifiée on réalise un diagramme d'activité (AD), en effectuant des choix fonctionnels (choix de l'algorithmique employée). L'I.S. étant une démarche itérative, plusieurs choix différents pourront être nécessaires avant de valider la solution choisie.



### E.3.3. Associer les opérations aux états

Sur la base des diagrammes d'états réalisés lors de l'analyse des exigences et de l'identification des opérations précédentes, associer les opérations aux états.



**A noter :** des labels permettent de caractériser précisément un état :

- **entry** : activités réalisées à l'entrée dans l'état ;
- **do** : activités réalisées dans l'état ;
- **exit** : activités réalisées à la sortie.

### E.3.4. Vérifier l'architecture logique

Les exigences système doivent être satisfaites par l'architecture logique. Cela se fait au moyen d'un diagramme de décomposition des activités, éventuellement complété par les matrices de relation exigences / activités :

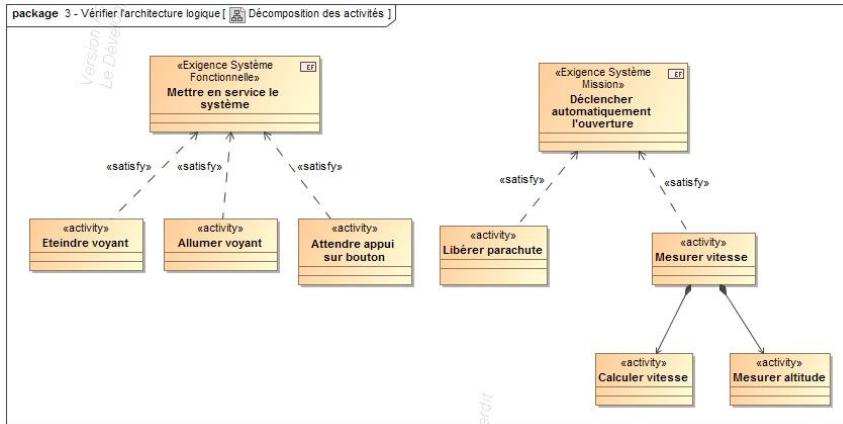


Diagramme hiérarchique des activités

Matrice exigences système / activités (Excel)

	2 - Etats système			3 - Exigences Système		
	En fonctionnement	En fonctionnement	En veille	EF1 Déclencher automatiquement l'ouverture	EF2 Mettre en service le système	
2	1	1	1	✓	✓	✓
3	2	2	3	✓	✓	✓

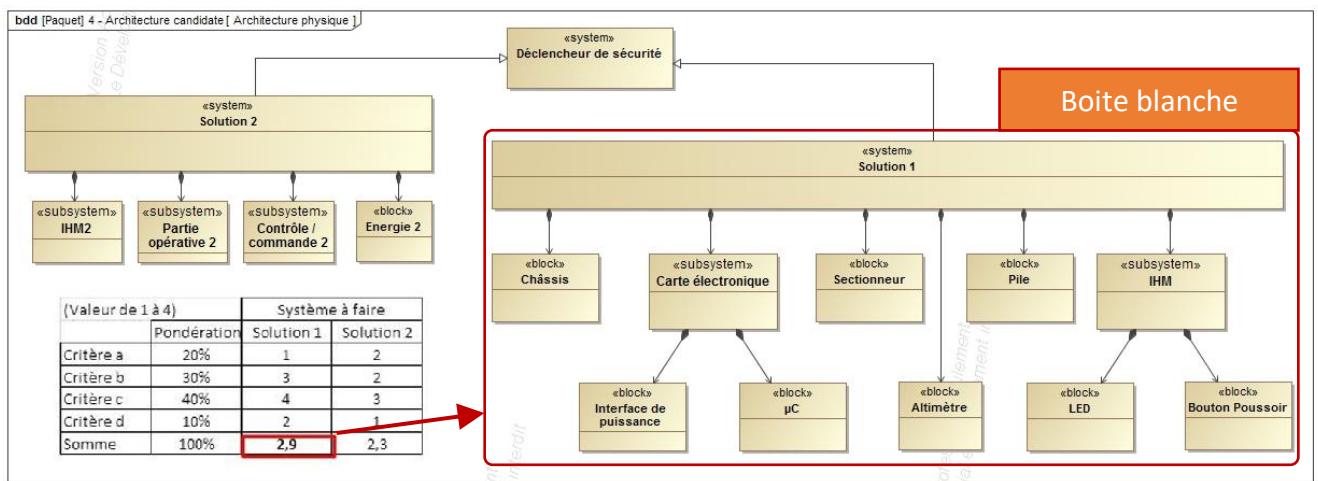
La matrice indique les états système (En fonctionnement, En fonctionnement, En veille) et les exigences système (EF1, EF2).

Matrice exigences système / activités

### E.3.5. Analyser les architectures candidates

Sur la base de l'architecture logique initiale précédente, il s'agit de réaliser un regroupement « logique » des opérations. Différentes propositions sont souhaitables pour pouvoir les comparer et choisir la meilleure relativement à des critères (savoir-faire, expérience, performance, coût, délais, ...).

On réalise ensuite une étude suivant des critères de choix pour dégager la solution qui semble la plus pertinente (on rencontre parfois l'expression d'« analyse de la valeur »).

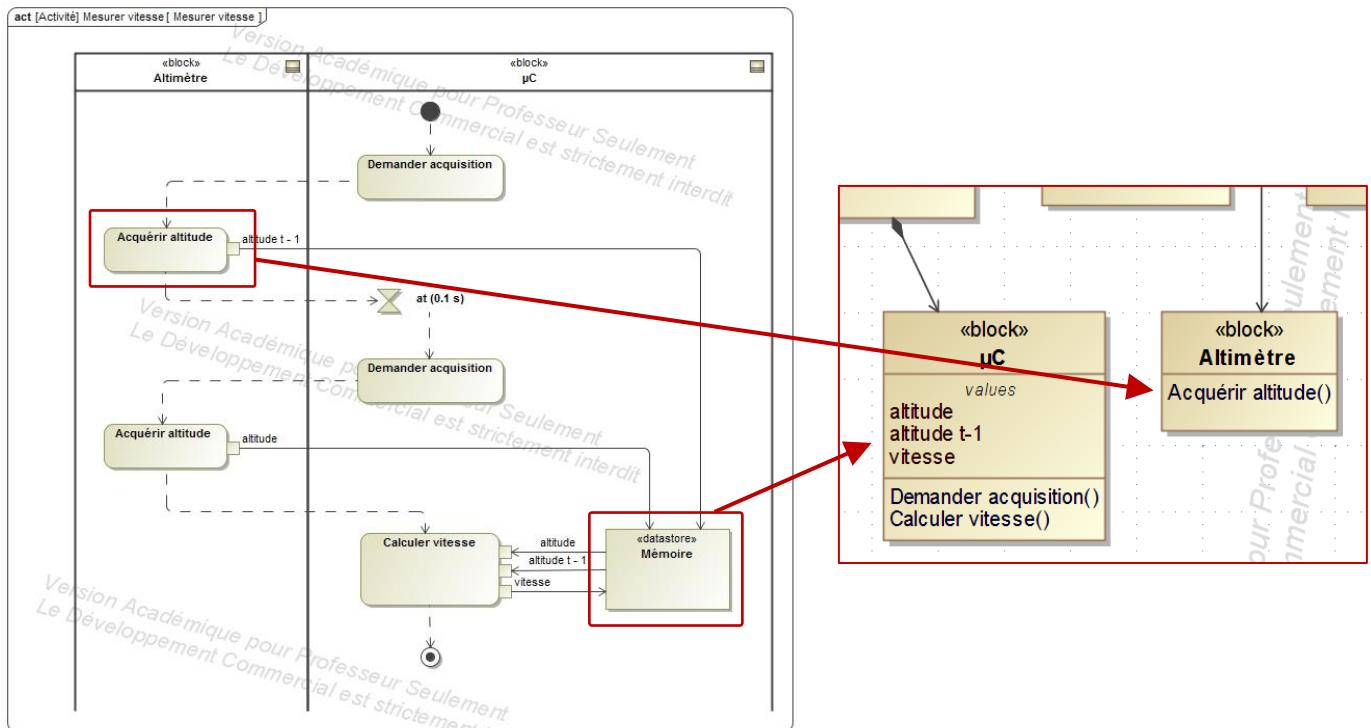


### E.3.6. Allouer les opérations aux sous-systèmes

Suite à l'activité précédente ou dans le même temps, respectivement pour l'architecture retenue ou pour toutes les architectures candidates, il est nécessaire d'allouer les opérations du système aux sous-systèmes. Il en est de même pour les signaux reçus et les attributs éventuels qui doivent être alloués aux sous-systèmes.

Dans cette phase de conception de la solution, de nouvelles opérations peuvent alors apparaître, en fonction de la décomposition.

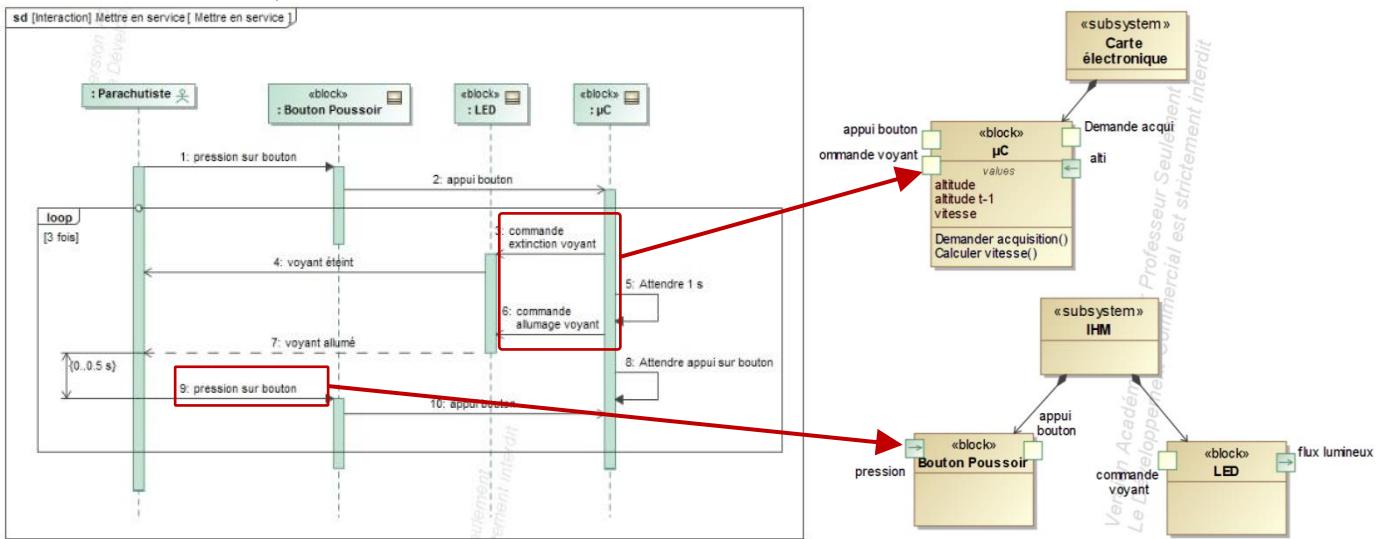
L'utilisation du diagramme d'activité permet d'aider à cette répartition, en utilisant le concept de partitionnement des activités selon des couloirs (ou lignes d'eau en référence au terme « swimlane »). Les opérations sont ainsi directement associées aux constituants.



### E.3.7. Définir les échanges avec les sous-systèmes

Dans cette étape, il convient de mettre en évidence les échanges des sous-systèmes entre eux et avec les éléments du contexte. Pour formaliser les interactions des sous-systèmes, on reprend les diagrammes de séquence où l'on fait apparaître les sous-systèmes, pour avoir une vision « boîte blanche » des échanges.

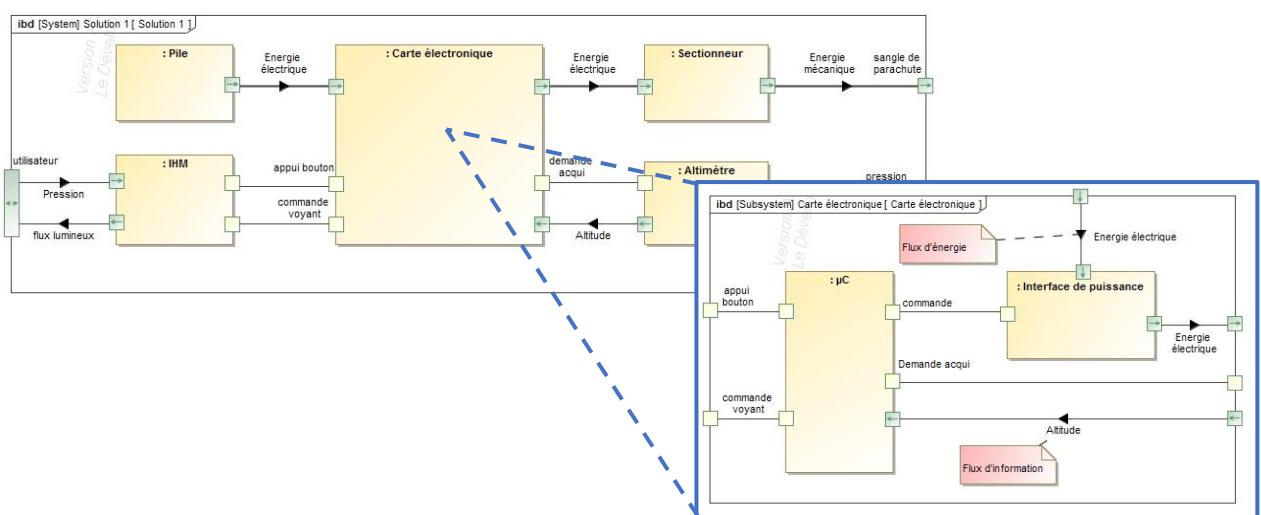
On peut alors compléter le diagramme de définition de bloc précédent pour ajouter les ports des blocs sous-systèmes :



### E.3.8. Définir la vue interne du système

L'ensemble des éléments précédents permet de construire la vue interne du système tel qu'il doit être réalisé. En particulier cela met en évidence les interfaces et flux entre les différents composants du système d'une part et avec l'environnement d'autre part.

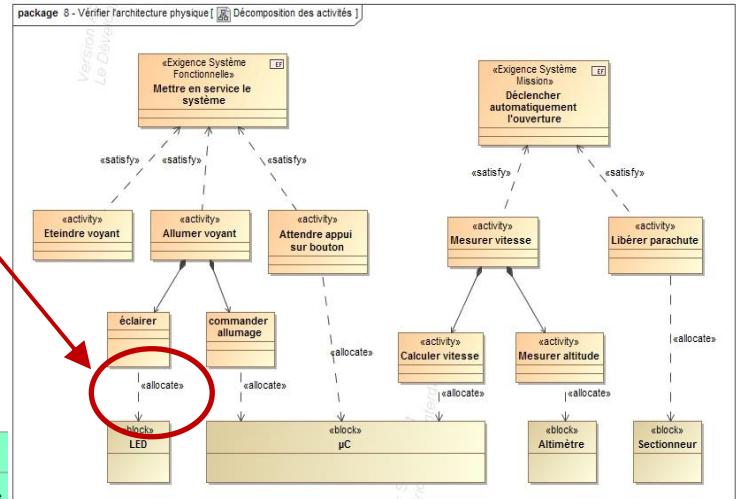
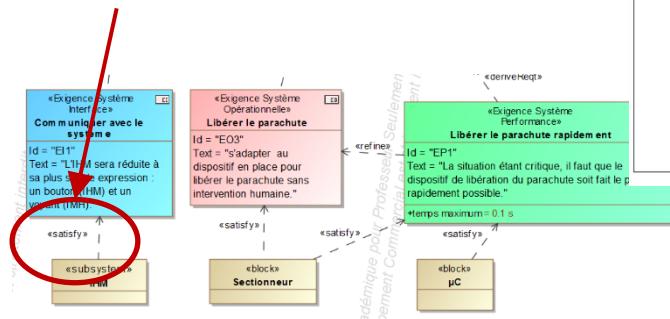
Au moins un diagramme de bloc interne est réalisé, mais selon l'architecture il sera utile de faire un IBD par sous-système.



### E.3.9. Vérifier l'architecture physique

Chaque constituant doit réaliser au moins une activité (sinon il n'est pas utile au système), et chaque activité doit être réalisée par un constituant (sans quoi elle a été oubliée) : cela correspond aux liens **d'allocation** entre activités et constituants.

De même, chaque constituant doit satisfaire au moins une exigence (sinon il n'est pas utile au système), et chaque exigence doit être satisfaite par au moins un constituant du système (sinon elle n'a pas été prise en compte). Cela se fait par un lien de « **satisfaction** » entre blocs et exigences.



**Remarque :** le diagramme hiérarchique des activités sous cette forme complète rappelle fortement le diagramme FAST<sup>12</sup>, sous forme verticale.

### E.3.10. Valider et documenter la conception de l'architecture

L'architecture peut être validée lorsque toutes les exigences systèmes sont satisfaites par le système, ses sous-systèmes et ses composants.

A l'issue de cette étape, pour chaque sous-système ou composant une décision peut être prise sur la suite à donner :

- Composant ou sous-système à acheter car disponible sur catalogue (composant « sur étagère ») ;
- Composant ou sous-système à réaliser (fabriquer) ;
- Composant ou sous-système à sous-traiter.

Dans ces deux derniers cas, les analyses menées précédemment servent d'éléments d'entrée et de spécifications aux processus de conception du composant ou du sous-système.

En guise de synthèse du processus de conception de l'architecture, il est nécessaire d'enregistrer convenablement l'architecture obtenue, par l'élaboration du dossier de conception.



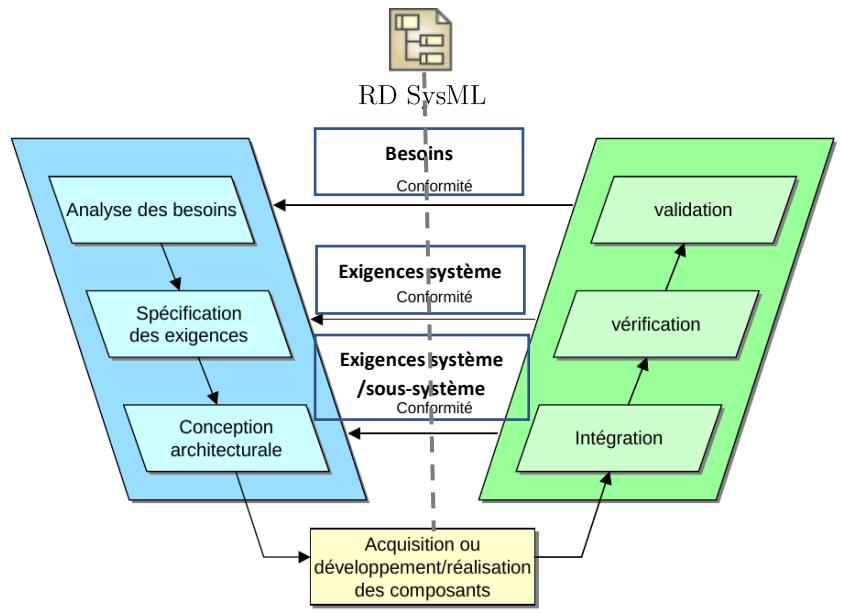
<sup>12</sup> Acronyme de Function Analysis System Technique, outils permettant dans une démarche logique d'aller du « pourquoi » (fonctions de service) au « comment » (solutions techniques) en passant par le « quoi » (fonctions techniques). Ce formalisme est décrit par la norme NF EN 12973.

#### E.4. Synthèse et conclusion

➤ **Démarche de conception :**

La démarche proposée s'inscrit totalement dans une représentation de type « cycle en V », de par la nature des activités et leurs enchainements, la traçabilité des exigences depuis le besoin initial jusqu'aux exigences système/sous-système.

Les phases d'IVVQ<sup>13</sup> sont grandement facilitées selon la qualité de l'ingénierie des exigences, et donc la bonne définition des besoins /exigences.



Démarche d'IS et cycle en V (source AFIS [8])

<sup>13</sup> Acronyme souvent employé et regroupant les phases :

d'**Intégration** : au sens d'intégrer en un système unique tous les constituants réalisés (réalisation du système de systèmes) ;

de **Vérification** : une fois le système réalisé, on vérifie les exigences (mesures entre le réel et l'attendu) ;

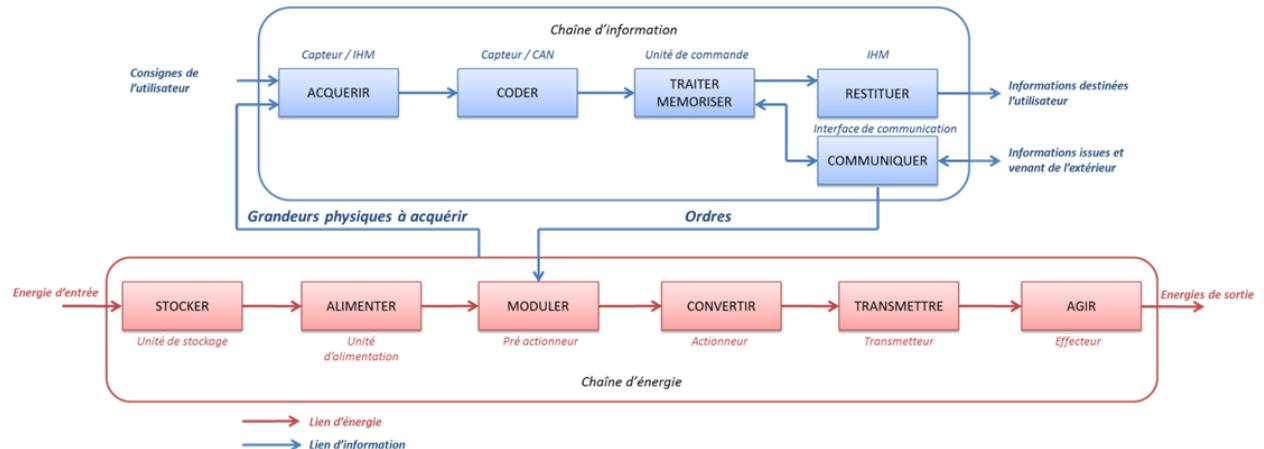
de **Validation** : une fois toutes les vérifications faites, on peut alors valider ;

et de **Qualification** : une fois le produit validé, on le qualifie par la caractérisation de ses performances (qui doivent donc être au moins égales à celles attendues).

## ➤ A propos de la chaîne fonctionnelle :

La chaîne fonctionnelle (ou chaîne d'information – chaîne de puissance) est un formalisme permettant de représenter de manière synthétique les architectures fonctionnelles et physiques. Elle représente les principales fonctionnalités préalablement identifiées et établies (auxquelles tout système peut se conformer), et les échanges au sein du système en termes de flux MEI.

Elle offre l'avantage d'offrir une architecture standard, et facilite ainsi grandement l'analyse fonctionnelle et structurelle (formalisme utilisé très fréquemment en SI).

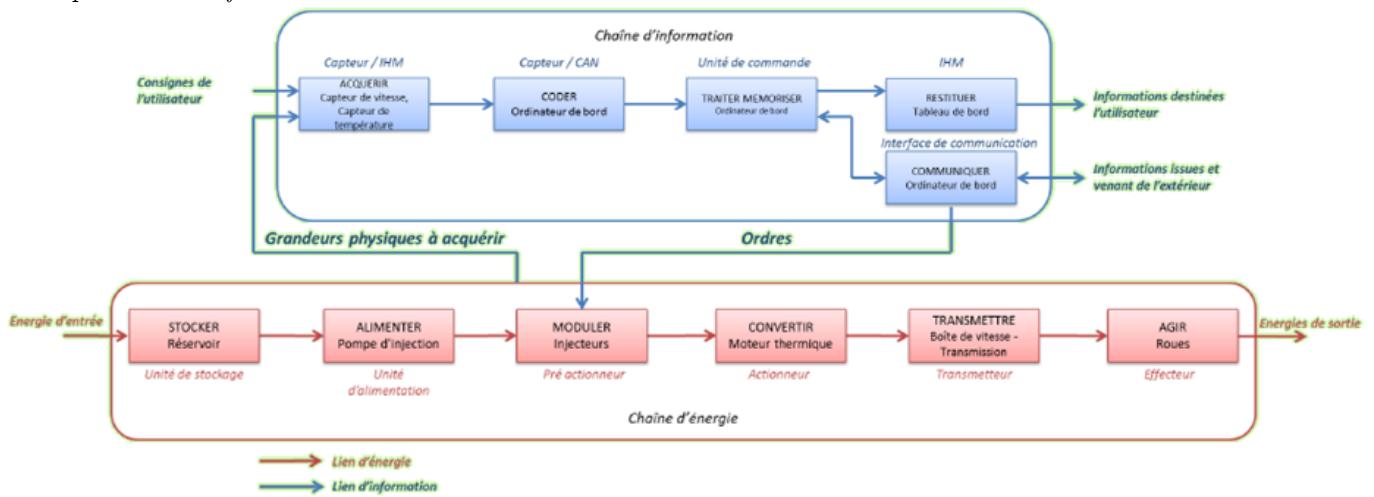


**Lien d'information :** Un lien d'information véhicule une information. Très souvent, la nature du signal est de type électrique (en  $V$ ).

**Lien de puissance (ou d'énergie) :** Un lien de puissance véhicule une grandeur de type **effort** et une grandeur de type **flux**. Le produit de ces deux grandeurs est une puissance.

Domaine	Effort	Flux
Électrotechnique	Tension ( $U$ en $V$ )	Courant ( $I$ en $A$ )
Mécanique de translation	Force ( $F$ en $N$ )	Vitesse ( $V$ en $m.s^{-1}$ )
Mécanique de rotation	Couple ( $C$ en $N.m$ )	Fréquence de rotation ( $\omega$ en $rad.s^{-1}$ )
Hydraulique – Pneumatique	Pression ( $P$ en $Pa$ )	Débit volumique ( $q$ en $m^3.s^{-1}$ )
Thermique	Température ( $T$ en $^\circ C$ )	Flux d'entropie ( $QS$ en $W.^\circ C^{-1}$ )

Exemple : chaîne fonctionnelle d'un véhicule.



➤ **Synthèse sur le MBSE<sup>14</sup> :**

Une modélisation SysML	
Spécifie :	Ne spécifie pas :
<ul style="list-style-type: none"> <li>• Ce que le système doit faire (aspects fonctionnels, comportementaux) ;</li> <li>• Ce qu'il doit être (aspects structurels) ;</li> <li>• En respect des besoins et des contraintes initiales (les exigences) ;</li> <li>• Le tout sous forme graphique ou tabulaire ;</li> <li>• Menée à bien grâce à l'I.S..</li> </ul>	<ul style="list-style-type: none"> <li>• Comment cela sera réalisé au final (on reste dans le conceptuel) : c'est le rôle du maître d'œuvre en charge de la réalisation, sur la base du modèle SysML ; qui utilise tout son savoir, savoir-faire et ses outils métiers pour réaliser le système/sous-système à sa charge ;</li> <li>• De modèle volumique, de schémas cinématiques, électriques, ...</li> </ul>

A partir de là, l'homme de l'art, l'homme du métier prend le relais pour :

- Analyser la documentation technique d'un produit sur catalogue et valider un choix ;
- Effectuer ses représentations symboliques nécessaires à une pré-étude de réalisation (mécanique, pneumatique, électronique, électrotechnique, ...) ;
- Utiliser des logiciels de CAO spécifiques pour modéliser la réalisation finale (Autodesk Inventor, Solid Works, Catia, Proteus, Revit, ...) ;
- Simuler avant réalisation pour valider le modèle ;
- Produire le prototype issu de cette modélisation, par les machines le permettant.

---

<sup>14</sup> MBSE : Model-Based Systems Engineering : l'ingénierie des systèmes basée sur le modèle

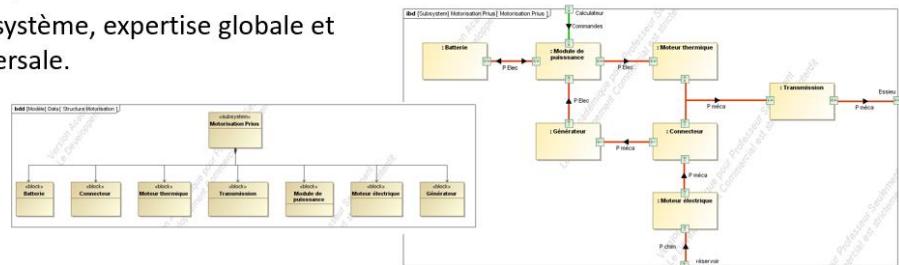
# La liaison IS / Métiers



Toyota Prius.

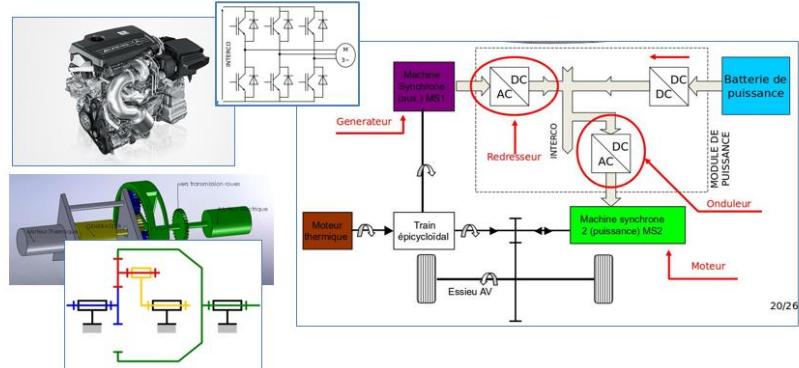
## Ingénieur Système :

Concepts système, expertise globale et transversale.



## Ingénieur métier :

Expertise métier,  
métier,  
outils métiers.



20/26

**Remarque :** les différents modèles établis durant l'ingénierie système peuvent servir de base à de la simulation comportementale (diagramme d'états, d'activités) et fonctionnelle (schémas-blocs, diagrammes de blocs internes, par le biais de leurs modèles équationnels, en lien avec le diagramme paramétrique) afin de valider les choix architecturaux.

## ➤ Pour conclure :

L'expertise globale et transversale d'un ingénieur système nécessite des connaissances et compétences dans chacun des processus métiers : en mécanique, électronique, électrotechnique, ... Mais doit-il pour autant être un bon technicien dans chacun de ces domaines ? ...

## Bibliographie

- [1] The Standish Group, «CHAOS Report,» [www.standishgroup.com](http://www.standishgroup.com), 2015.
- [2] Signataires Agile, «Manifeste pour le développement Agile de logiciels,» [www.agilemanifesto.org](http://www.agilemanifesto.org), 2001.
- [3] Collectif AFI.S., «Bonnes pratiques en ingénierie des exigences,» Cépaduès Editions, 2012.
- [4] INCOSE, «Systems Engineering Vision 2025,» 2014.
- [5] Collectif AFI.S., «Livre blanc AFI.S., Ingénierie Système : la vision pour les années 2020-2025,» Cépaduès Editions, 2012.
- [6] I.S.O/IEC JTC 1/SC 7, «I.S.O/IEC/IEEE 15288:2015,» International Organization for Standardisation, 2015.
- [7] D. Krob, «Eléments d'architecture des systèmes complexes,» 2007.
- [8] Collectif AFI.S., «Découvrir et comprendre l'Ingénierie Système,» Cépaduès Editions, 2012.
- [9] I.S.O/TC 207/SC 5, «I.S.O 14040:2006,» International Organization for Standardisation, 2006.
- [10] A. Faisandier, 2011. [En ligne]. Available: <http://mapsysteme.com>.
- [11] BKCASE Editorial Board, «The guide to Systems Engineering Body of Knowledge (SEBoK), v.1.9.,» 2017.
- [12] Object Management Group, «OMG System Modeling Language, v.1.5,» 2017.
- [13] A. D. K. a. K. J., «The impact of Systems Engineering on Complex System,» NASA, Published by SEE Stevens institute, 2004.
- [14] «MagicDraw,» Dassault Systèmes, 2018. [En ligne]. Available: <http://www.magicdraw.com>.