

Ingénierie système

L' **Ingénierie Système** (ou ingénierie de systèmes) est une démarche méthodologique générale qui englobe l'ensemble des activités adéquates pour concevoir, faire évoluer et vérifier un système apportant une solution économique et performante aux besoins d'un client tout en satisfaisant l'ensemble des parties prenantes.

1. Système

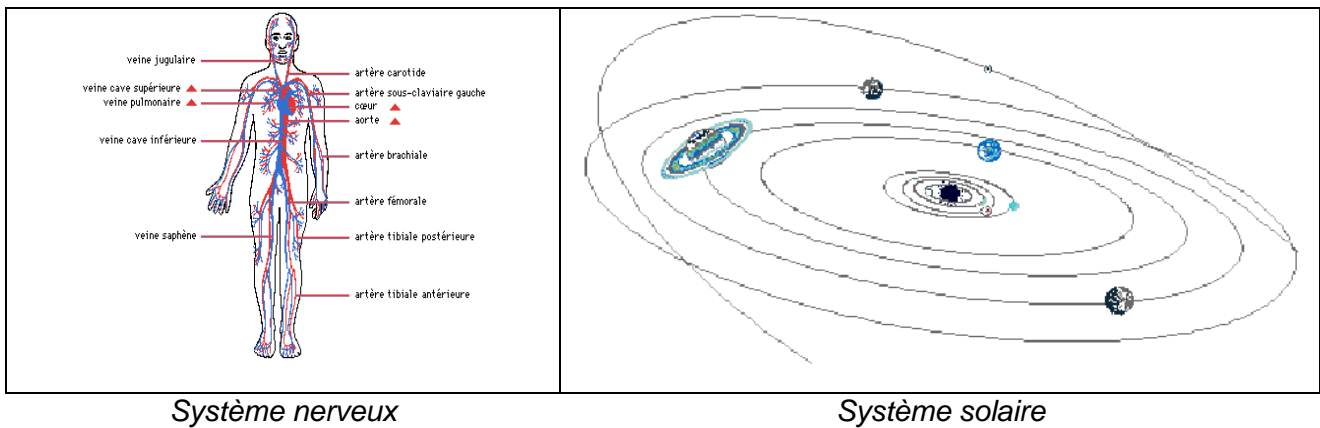
1.1. Définitions

Un **système** est un arrangement d'éléments en **interaction** organisée pour atteindre un ou plusieurs **objectifs** définis.

Source : ISO/IEC 15288:2002(E)

Les systèmes peuvent être classés en deux grandes familles :

- **les systèmes naturels**, non créés par l'homme :



- **les systèmes artificiels (ou industriels)**, créés par l'homme, pour remplir une ou plusieurs fonctions précises.



vélo

Remarques :

- un système n'est pas seulement un ensemble d'éléments, il ne suffit pas de connaître chaque composant individuellement pour connaître le comportement global du système.
- Un système complexe s'oppose à un système simple où une information d'entrée permet de trouver le comportement de sortie par une analyse déductive.

Un système est complexe si son comportement n'est pas prédictible de manière déductive, mais est le résultat des liaisons entre ses composants. Un **système industriel** est par nature **complexe** puisque son comportement résulte de l'ensemble des interactions entre les différents composants.

Les systèmes simples sont traités en sciences fondamentales et font appel à l'esprit de logique. Les problèmes sont bien posés et il y a souvent une solution unique. Leur comportement se place dans le cadre des modèles de connaissances (le comportement de chaque composant est géré par un phénomène physique élémentaire, et le comportement du système entier est la somme des comportements élémentaires).

L'analyse des systèmes complexes est réalisée en Sciences Industrielles pour l'Ingénieur. Elle nécessite de modéliser le comportement des composants du système et de leurs interactions. Cette approche sollicite l'esprit de synthèse et d'initiative.

1.2. Classification

1.2.1. Contexte ou domaine d'application

Le domaine d'application est le milieu physico-économique dans lequel évolue le système et pour lequel il a été conçu. On parle alors de **système industriel** par opposition au système d'étude utilisé dans les laboratoires dans un but pédagogique (maquette, prototype).

On distingue par exemple les domaines d'application suivants :

- **de diffusion limitée :**

On rencontre ces systèmes dans le domaine de la production industrielle principalement. Ces systèmes, souvent unitaires, font l'objet d'étude réduite et sont constitués d'éléments standards assemblés.

exemple : machines de production

- **de grande diffusion :**

Ils sont produits en grande série et font l'objet d'études poussées pour optimiser les coûts. On rencontre ces systèmes dans de nombreux domaines :

- ✓ électroménager : *machine à laver, etc.*
- ✓ domotique : *store, ouvre-portail, etc.*
- ✓ distribution service : *distributeur de billets, station de lavage, péage d'autoroute, etc.*
- ✓ médical : *respirateurs, pompes, etc.*
- ✓ automobile : *ABS, direction assistée, suspension active, etc.*
- ✓ ...

1.2.2. Critères Technico-économiques

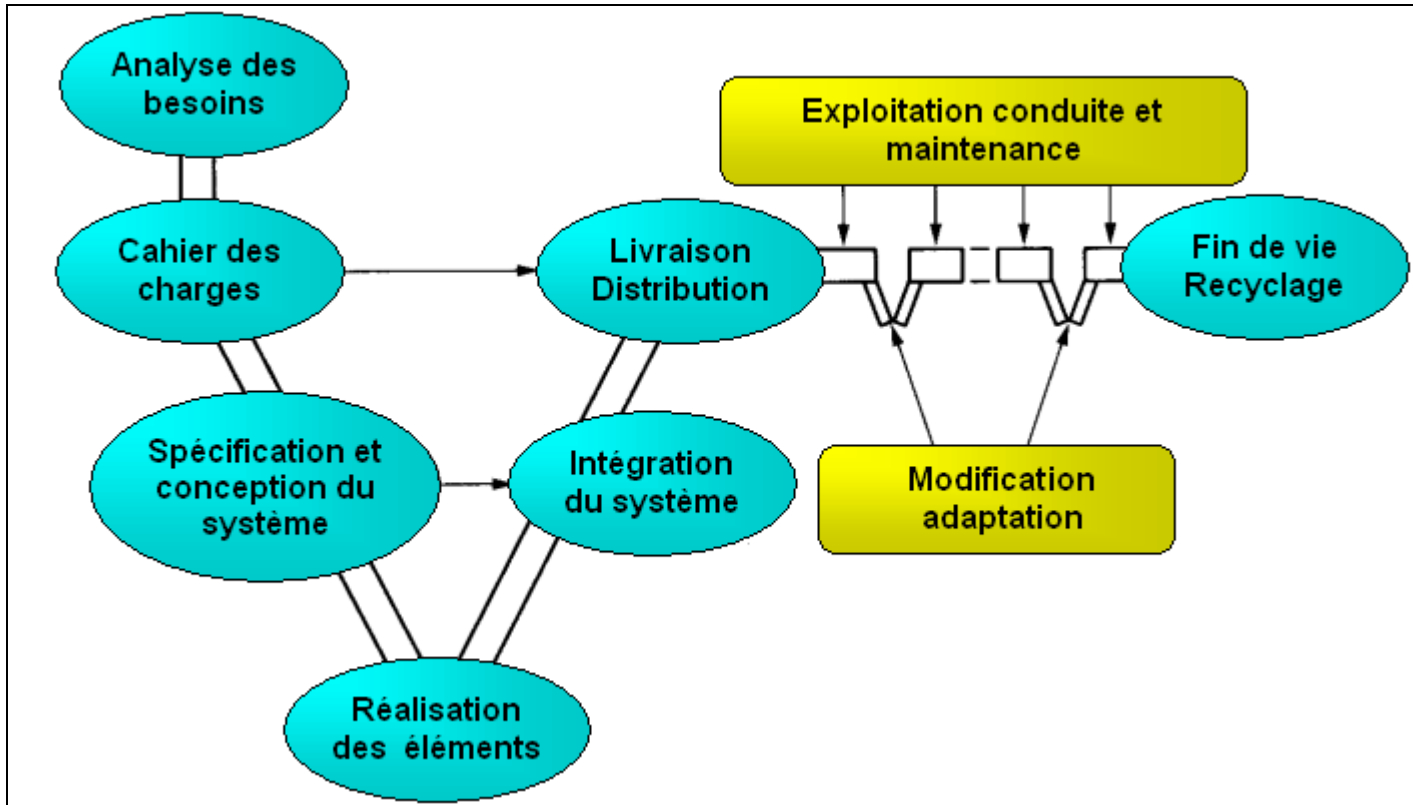
Ces critères sont pris en compte lors de conception du système industriel : ainsi, nous pouvons citer :

- ✓ La durée de vie
- ✓ Le coût
- ✓ La fiabilité
- ✓ La quantité

Certains appareils photographiques ou rasoirs sont dit jetables. Leur coût est faible mais ils ont une durée de vie très limitée.

1.3. Cycle de vie d'un système

Afin de mieux cerner les liens qui existent entre un système, son environnement et les différents intervenants sur ce système, il est classique de représenter la vie d'un système sous la forme d'un V (vé) ou plus précisément d'une racine carrée.



Cycle de vie

Les cinq premières phases correspondent à la création du système, elles se terminent par la livraison du système terminé prêt à fonctionner.

1.3.1. Analyse du besoin

« Un besoin est une nécessité ou un désir éprouvé par un utilisateur » (NF X 50-150).

Pour valider le besoin, il faut se poser les trois questions :

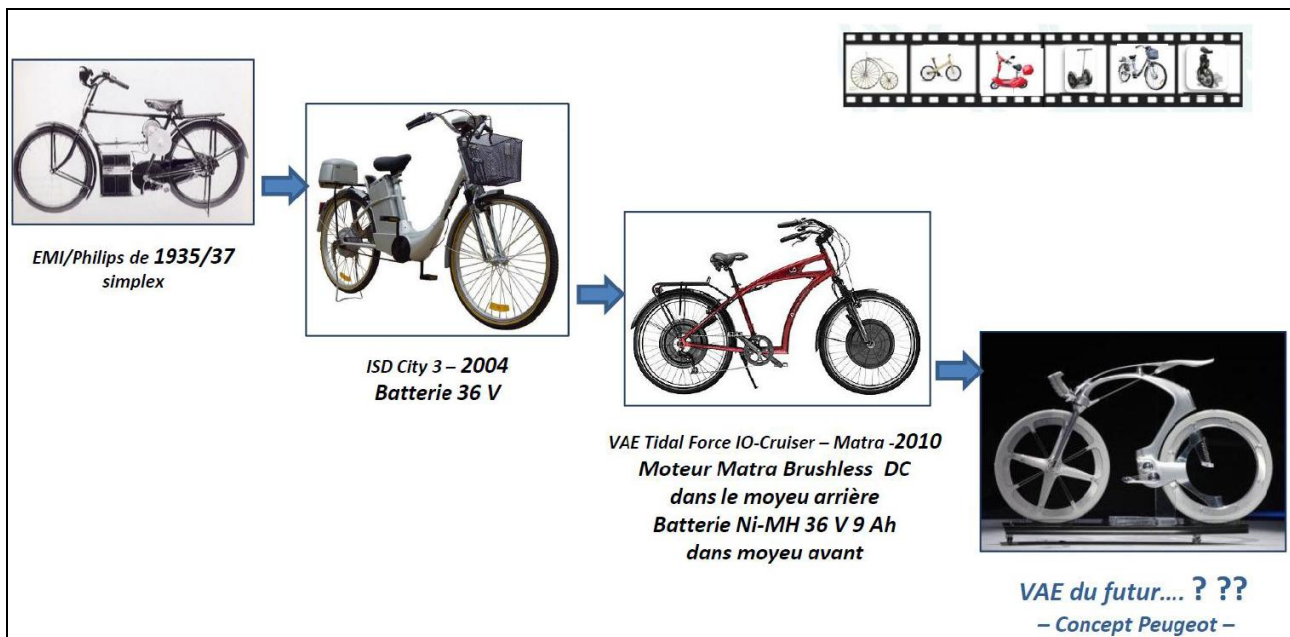
- ✓ Pourquoi le produit existe-t-il ?
- ✓ Qu'est-ce qui pourrait faire évoluer le besoin ?
- ✓ Qu'est-ce qui pourrait le faire disparaître ?

Exemple 1 : La cabine téléphonique

si une cabine téléphonique permet à l'utilisateur d'avoir accès au réseau téléphonique, le besoin d'une cabine téléphonique est notablement réduit par l'apparition des téléphones portables.

Le client est sensible à l'évolution du contexte économique, social et environnemental ainsi qu'au degré d'innovation, le besoin évolue donc constamment.

Exemple 2 : Le vélo à assistance électrique



1.3.2. Etablir le Cahier des Charges Fonctionnel (C.d.C.F.) :

Le cahier des charges fonctionnel est un document qui permet de formaliser avec précision le besoin du demandeur. C'est un tableau de bord qui définit le projet et détaille les conditions dans lesquelles il doit être réalisé. C'est le lien de compréhension entre l'entreprise et le client.

Les contraintes de base sont économiques (les contraintes monétaires comme le budget de fonctionnement), environnementales (le caractère recyclable du produit, etc.), humaines (par exemple, dans le cas d'un jouet pour enfant, il doit être léger, ne pas contenir de petites pièces, etc.), industrielles (par exemple, il doit être fabriqué au Canada) et matérielles (par exemple, il doit spécifier les morceaux qui peuvent être remplacés, comme des piles, il doit préciser le recours à tel ou tel serveur d'applications).

1.3.3. Spécifications et conception du système :

C'est la phase qui permet de définir précisément ce que doit être le système, d'en choisir tous les composants et la manière dont ils seront reliés.

1.3.4. Réalisation des éléments :

Au cours de cette phase de passage à la réalité, l'ensemble des constituants matériels : pièces mécaniques, moteurs, cartes électroniques, écrans de visualisation, etc. et immatériels : programmes informatiques, logiciels, etc. sont réunis en vue de l'assemblage.

1.3.5. Intégration du système :

C'est la phase de montage, d'assemblage de composants matériels et immatériels, le terme intégration montre à quel point les interrelations sont étroites et multiples.

1.3.6. Livraison / Distribution :

Cette phase de vie peut apporter des contraintes supplémentaires au système.

1.3.7. Exploitation, conduite et maintenance :

La dernière phase correspond en fait à la vie réelle du système en fonctionnement depuis la livraison jusqu'à la mise au rebut finale. Pendant cette phase, le système est existant.

C'est la phase de rentabilité du système, elle peut être entrecoupée de phases de modification, adaptation et amélioration du système que l'on présente sous forme de petits V (vé).

Sur cette représentation de la vie d'un système, on peut marquer assez clairement les différents programmes de Sciences Industrielles en classes préparatoires aux grandes écoles :

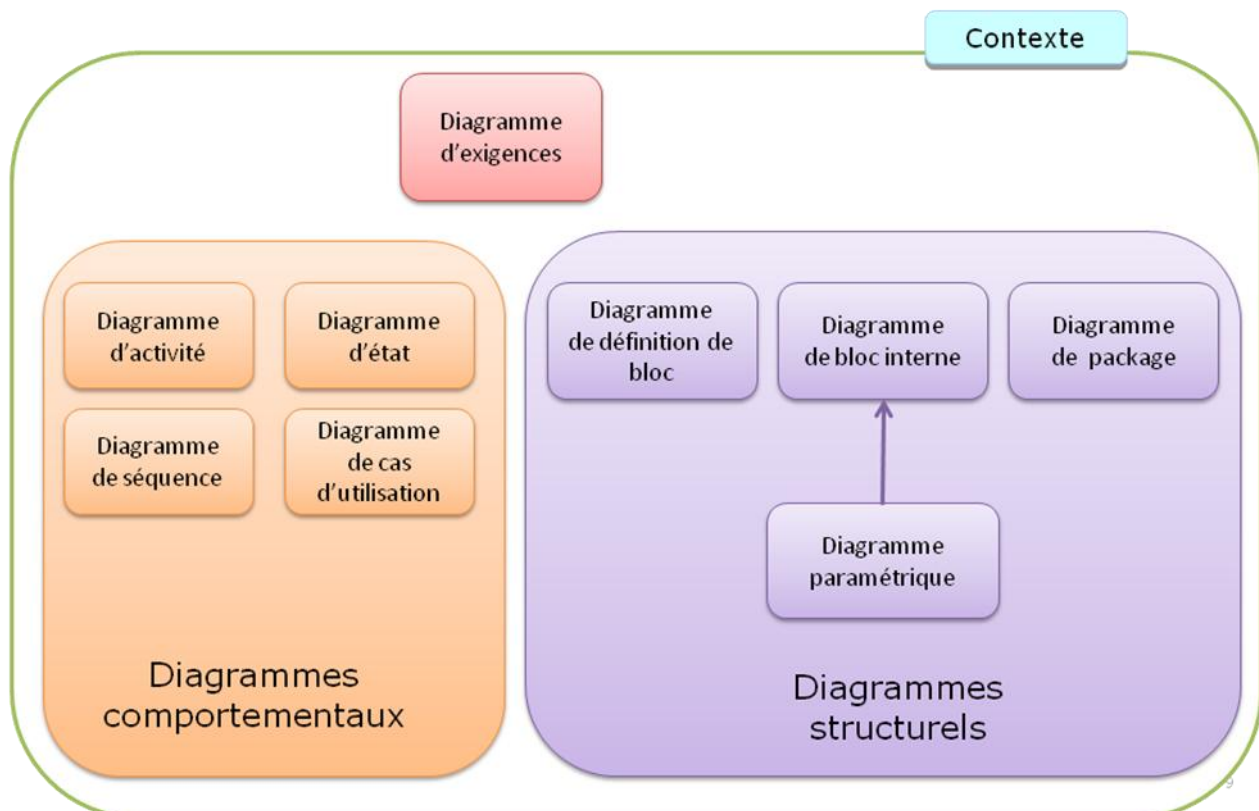
- ◆ **MPSI** : le programme est centré sur la phase 5 et exclut les petits V, l'objet est donc l'analyse de systèmes existants et vérification des performances ;
- ◆ **PCSI** : le programme est toujours sur la phase 5 y compris les petits V, l'objet est donc l'analyse et la vérification des performances d'un système existant, mais aussi la proposition de modifications ;
- ◆ **PTSI** : le programme prend en compte l'ensemble de la vie du système, les cinq phases sont abordées et notamment la phase 2 de conception des systèmes. Dans tous les cas, les systèmes sont à dominante mécanique.

2. Le langage SysML.

2.1. Présentation

Le langage **SysML** - **S**ystems **M**odeling **L**anguage - est un langage de modélisation spécifique au domaine de l'ingénierie système. Il permet la spécification, l'analyse, la conception, la vérification et la validation de nombreux systèmes et systèmes-de-systèmes

Le langage SysML s'articule autour de neuf types de diagrammes. Chacun d'eux étant dédié à la représentation des concepts particuliers d'un système.



SysML est un outil de modélisation et non une méthode d'analyse.

Si tous les diagrammes sont liés à un même système, ils peuvent être abordés dans n'importe quel ordre.

Chaque utilisateur est libre de créer sa propre méthodologie.

2.2. Le contexte

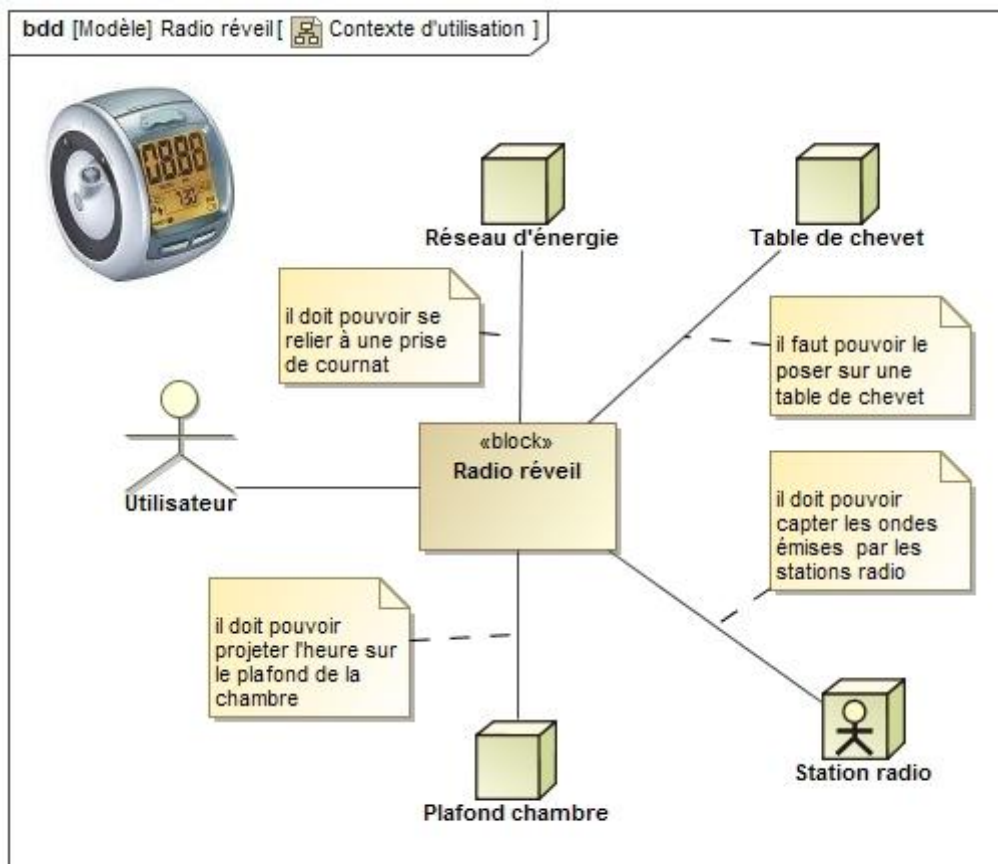
Il permet de définir les frontières et l'environnement du système étudié. Il permet en particulier de préciser la phase du cycle de vie dans laquelle on situe l'étude (généralement la phase d'utilisation).

Il répond à la question : "Quels sont les acteurs et éléments environnants du système ?".

Il n'y a aucune recommandation spécifique sur la manière de le représenter. Il pourra se faire par :

- ✓ Une carte mentale,
- ✓ Un bdd (diagramme de définitions de blocs) SysML (le plus souvent),
- ✓ Un ibd (diagramme de blocs internes) SysML.

• Exemple du radio réveil à projecteur Philips :



2.3. Les différents diagrammes

2.3.1. Le diagramme des exigences (Requirement Diagram)

« Une exigence permet de spécifier une capacité ou une contrainte qui doit être satisfaite par un système. Elle peut spécifier une fonction que le système devra réaliser ou une condition de performance, de fiabilité, de sécurité, etc. Les exigences servent à établir un contrat entre le client et les réalisateurs du futur système. » (définition extraite du livre de Pascal ROQUES)

• Rôle :

Représenter toutes les exigences du système.

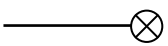
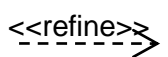
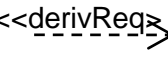
- ✓ Exigences environnementales,
- ✓ Exigences économiques,

- ✓ Exigences fonctionnelles,
- ✓ Exigences techniques,
- ✓ etc.

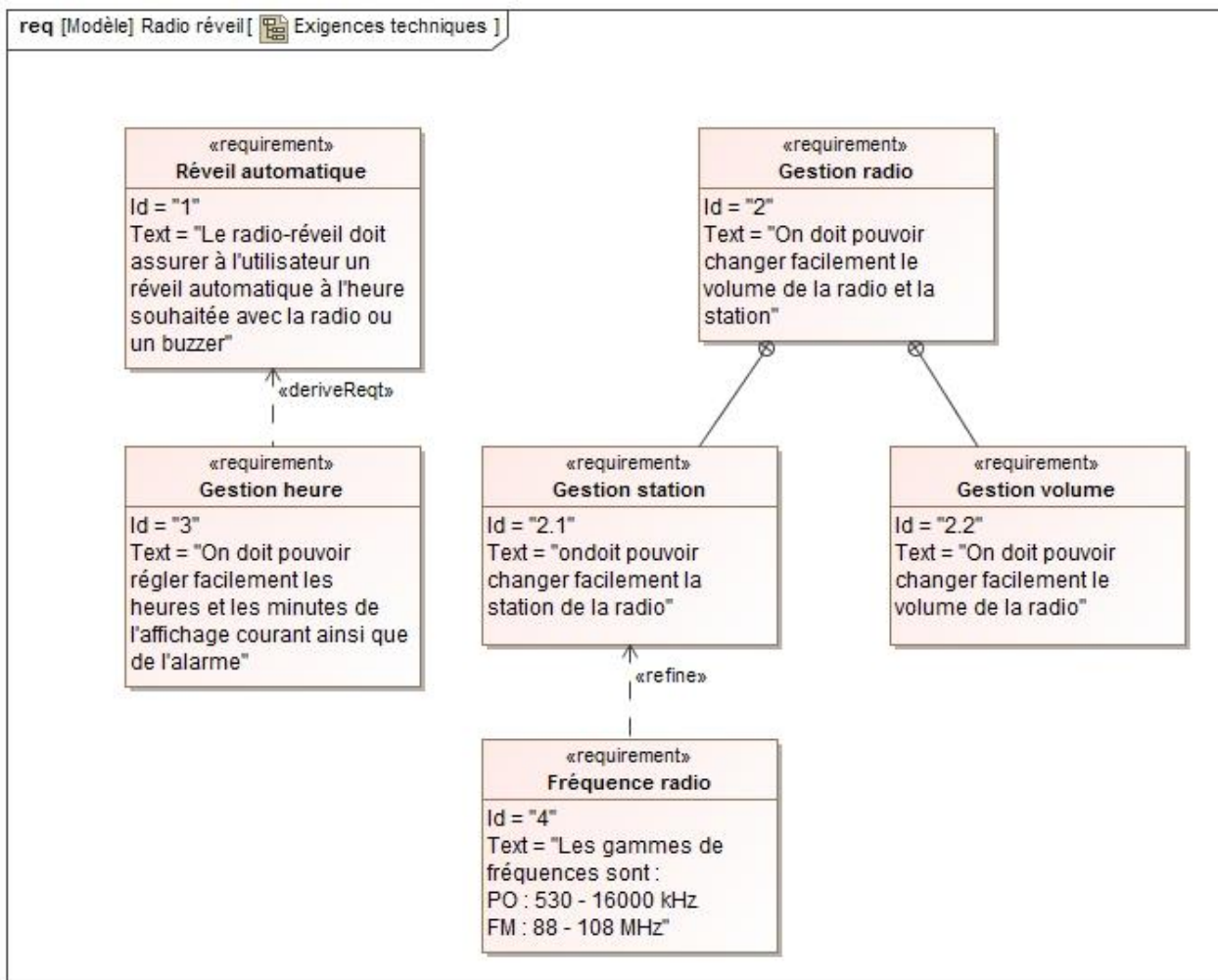
Remarque : La définition des exigences permet d'élaborer le cahier des charges.

• **Représentation**

- ✓ Un cartouche positionné en haut à gauche permet de spécifier le type de diagramme SysML, le type de l'élément concerné, l'élément concerné et le nom du diagramme.
- ✓ Chaque exigence est décrite par un texte à l'intérieur d'un rectangle marqué du stéréotype <<requirement>> identifié de façon unique (cela permet entre autre de gérer une traçabilité avec l'architecture).
- ✓ Les exigences peuvent être reliées entre elles par des relations de contenance, de raffinement ou de dérivation.

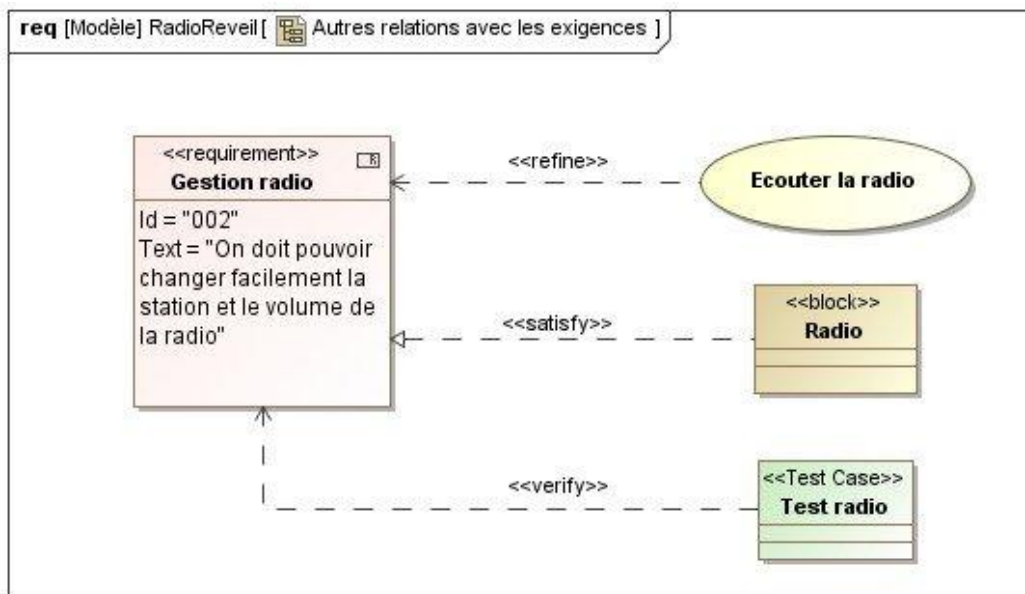
	la contenance permet de décomposer une exigence composite en plusieurs exigences unitaires
	le raffinement est un ajout de précision
	la dérivation relie des exigences de niveau différents

• **Exemple du radio réveil à projecteur Philips :**

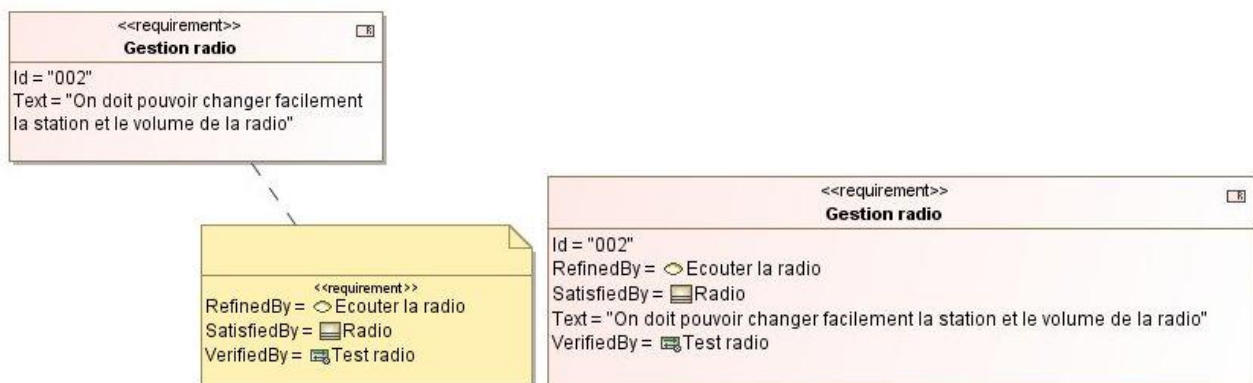


- **Traçabilité**

Le diagramme des exigences ne sert pas seulement à placer des contraintes dans le modèle. Il permet aussi de ramener des éléments venant d'autres diagrammes du modèle et ainsi de vérifier qu'une exigence a bien été prise en compte de quelque manière que ce soit, que ce soit par un cas d'utilisation <<refine>>, par un bloc d'architecture <<satisfy>> ou par un test <<vérify>>. Ainsi une exigence ne peut pas rester seule, ce qui indiquerait qu'il manque quelque chose dans une vue du modèle. Au final, elles doivent toutes être satisfaites.



Une autre représentation graphique proposée par SysML consiste à faire apparaître les relations de traçabilité par une note attachée à l'exigence ou encore par des attributs dans des compartiments supplémentaires.



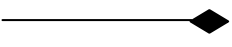
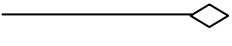
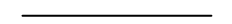
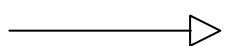
2.3.2. Le diagramme de définition de blocs (Blocks Définition Diagram)

- **Rôle :**

- ✓ Le diagramme de blocs nous permet de prendre connaissance de la **structure interne** du système. Il décrit la hiérarchie du système et les classifications système/composant.
- ✓ Il permet de répondre à la question « **qui contient quoi ?** ».
- ✓ C'est un diagramme qui se classe donc dans la catégorie des **diagrammes structurels**.
- ✓ Le bloc SysML constitue la brique de base pour la modélisation structurelle du système. Il peut représenter un sous-ensemble ou un composant (matériels ou logiciels) du système.

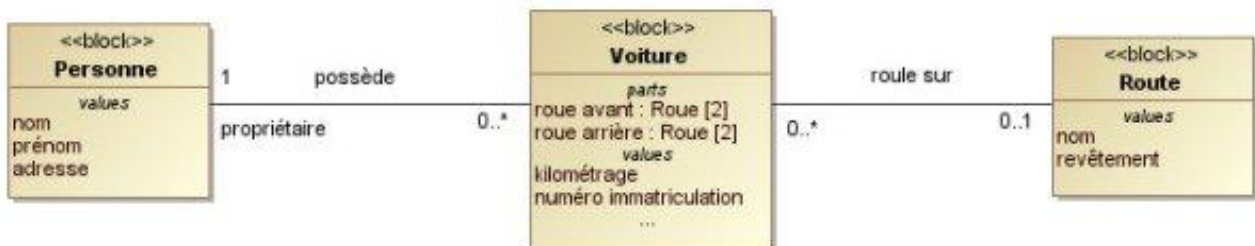
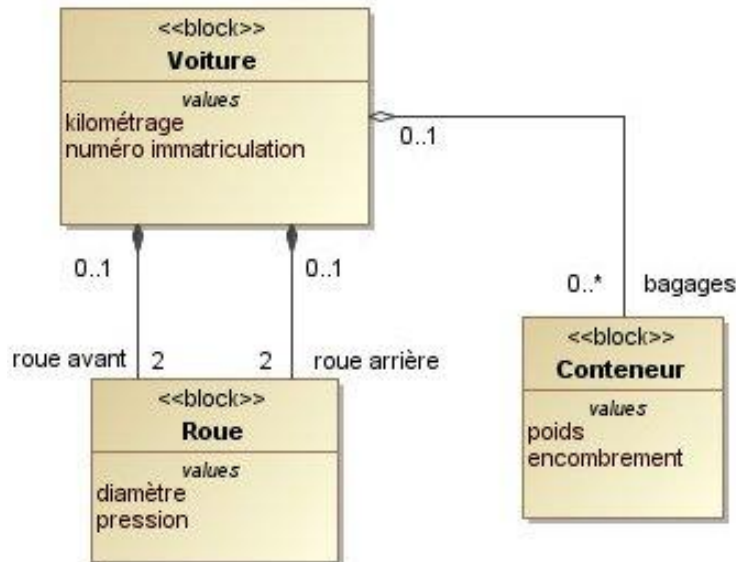
- **Représentation :**

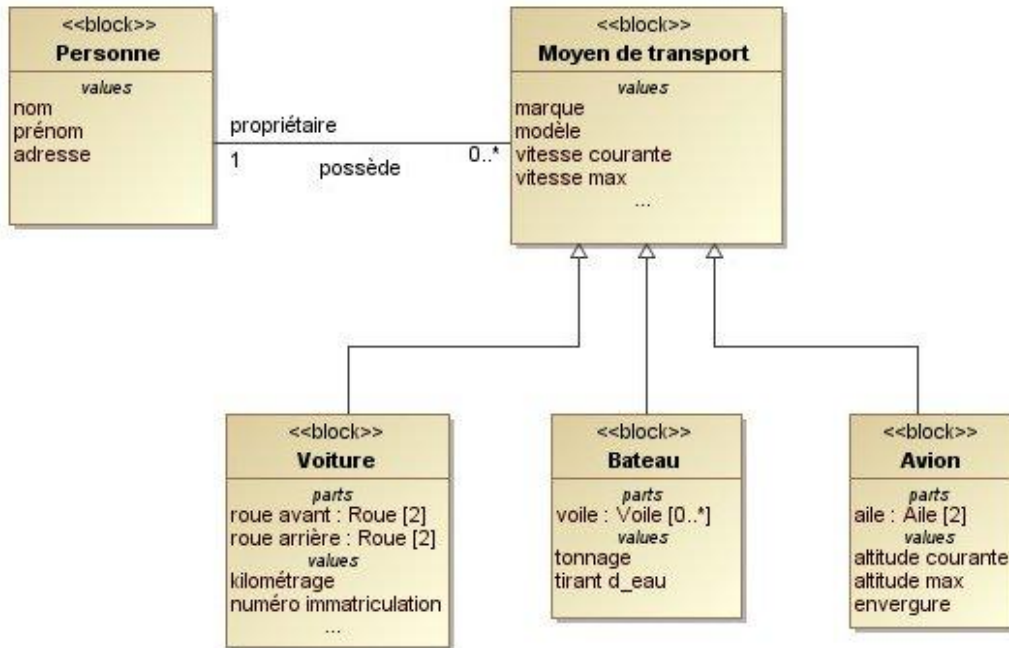
- ✓ chaque bloc est représenté graphiquement par un rectangle compartimenté. Le premier compartiment contenant le nom est les seul obligatoire. Les autres ont des "labels" indiquant ce qu'ils contiennent (valeurs, parties...)
- ✓ 4 liens permettent de relier les blocs entre eux :

Composition 	Le bloc conteneur est du côté du losange, le bloc contenu est du côté de la flèche (il n'est pas obligatoire de représenter l'extrémité). Elle indique que le bloc conteneur a nécessairement besoin du bloc contenu pour assurer les opérations qu'il offre.
Agrégation 	Pour fonctionner, le bloc conteneur n'a pas nécessairement besoin du bloc contenu. Par contre il est possible qu'à un moment donné ou à un autre, il en est besoin. Ce type de lien est moins fort que la composition.
Association 	Au contraire des autres, cette relation n'implique pas de contenance. La nature du lien peut dépendre du contexte. On peut ajouter des flèches aux extrémités dans le cas où la relation est unidirectionnelle.
Généralisation 	Le but est souvent de factoriser des propriétés communes (valeurs, parties...) à plusieurs blocs dans un bloc généralisé

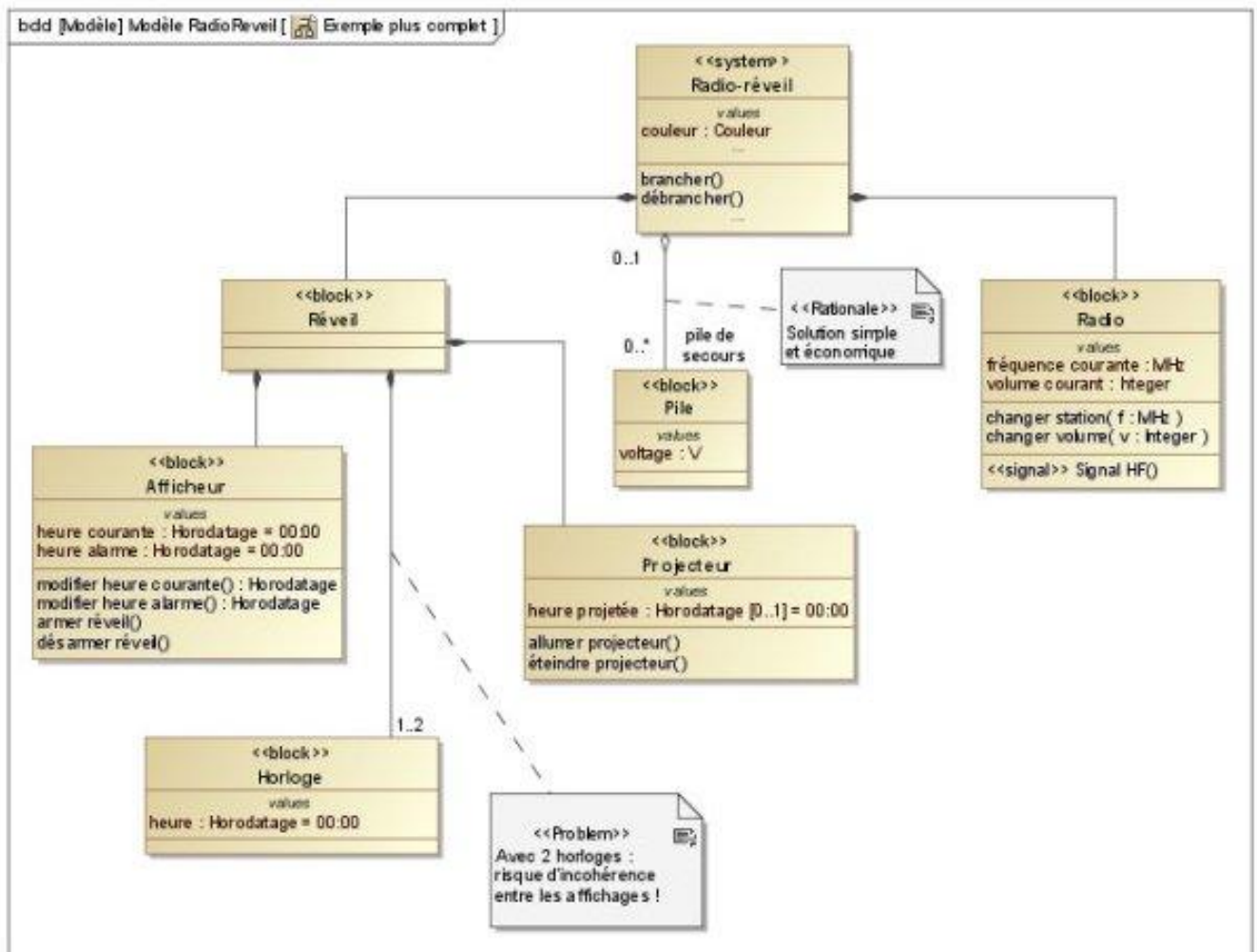
- ✓ Sur les liens d'un diagramme de blocs, il est possible de représenter la cardinalité ou multiplicité. Ce sont des nombres que l'on peut voir sur les liens entre blocs qui représentent le nombre minimum et maximum de possibilités. Quand il n'y en a pas, c'est qu'implicitement la cardinalité ou multiplicité vaut 1.

• Exemple : la voiture





• Exemple du radio réveil



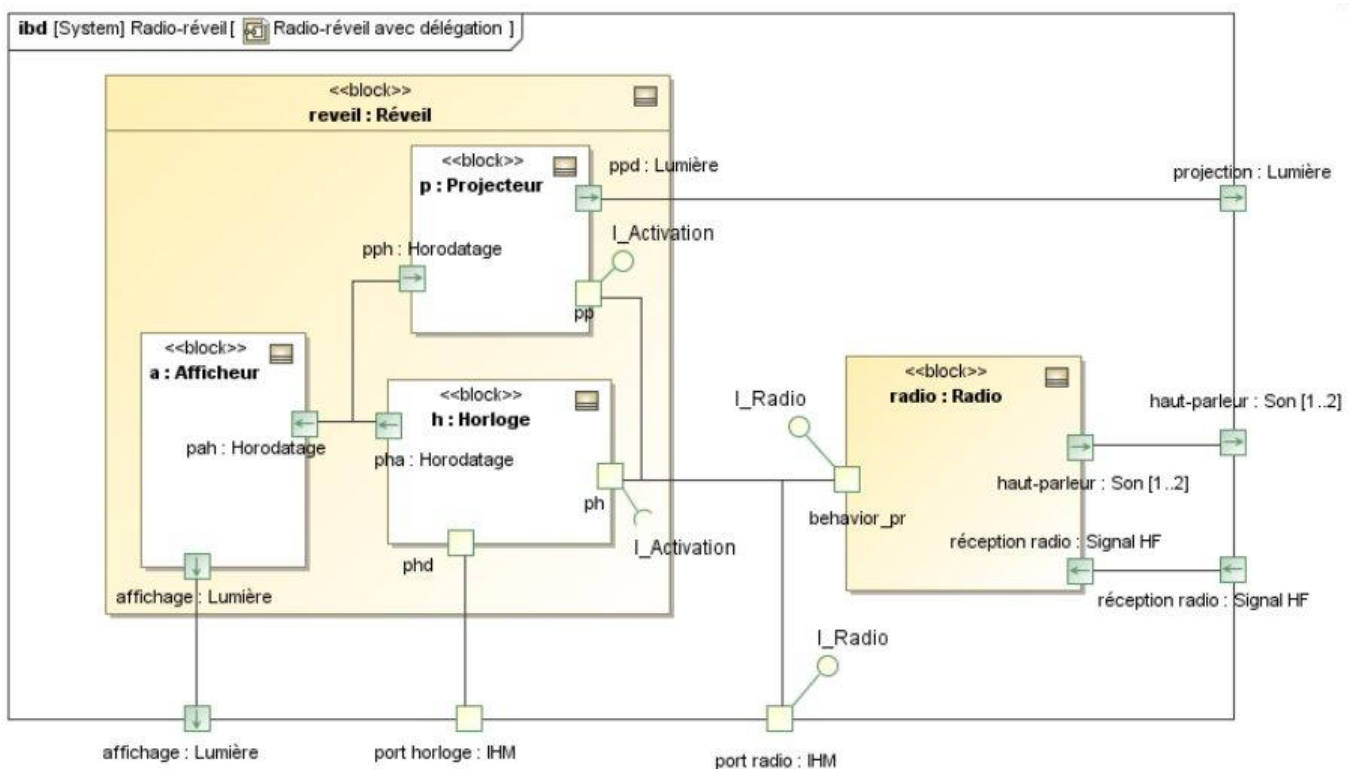
2.3.3. Le diagramme de blocs internes (Internal Blocks Diagram)

- **Rôle :**
 - ✓ Il permet de représenter les échanges de matière /énergie/information (MEI) entre blocs de même niveau grâce aux ports de flux (petit carré avec une flèche).
 - ✓ Il permet de représenter les services invoqués par un autre bloc grâce aux ports standards (petit carré sans flèche), et par extension toute entrée/sortie de contrôle/commande.
 - ✓ Il permet de représenter les liens entre les blocs de même niveau.
- **Représentation :**
 - ✓ Les blocs sont reliés entre eux par des connecteurs
 - ✓ la logique de connexion entre blocs est précisée par des ports. Attention à bien faire la différence entre port standard et port de flux.

Port de flux : canal d'Entrée/Sortie par lequel transite de la matière, de l'énergie ou de l'information (MEI).

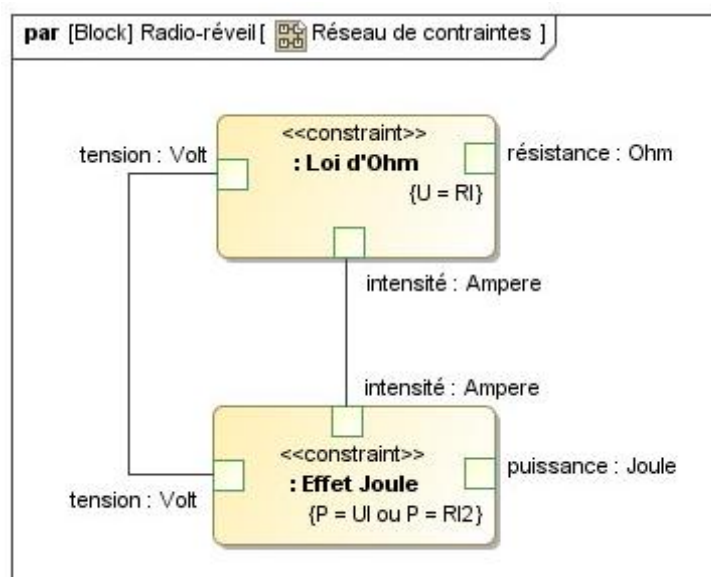
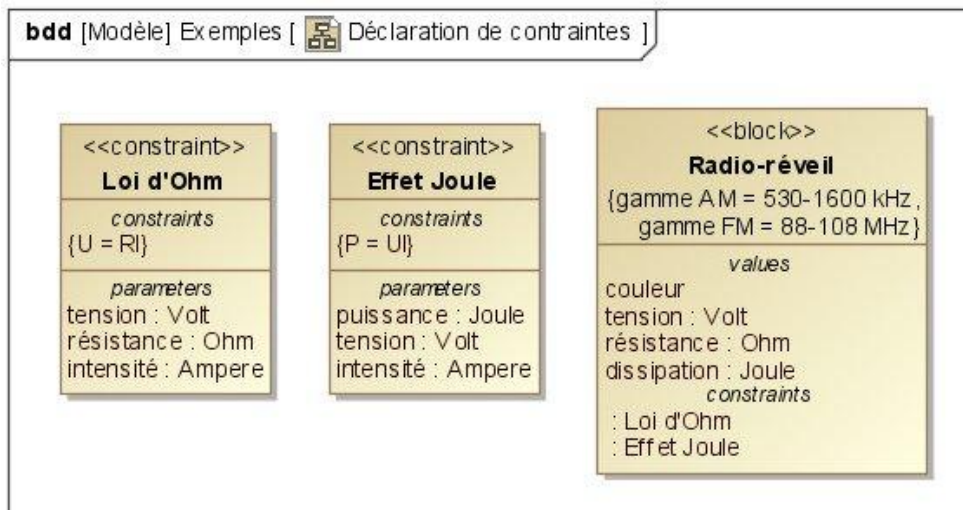
Port standard : désigne une interface permettant d'invoquer un service/une opération.

- **Exemple : radio réveil**



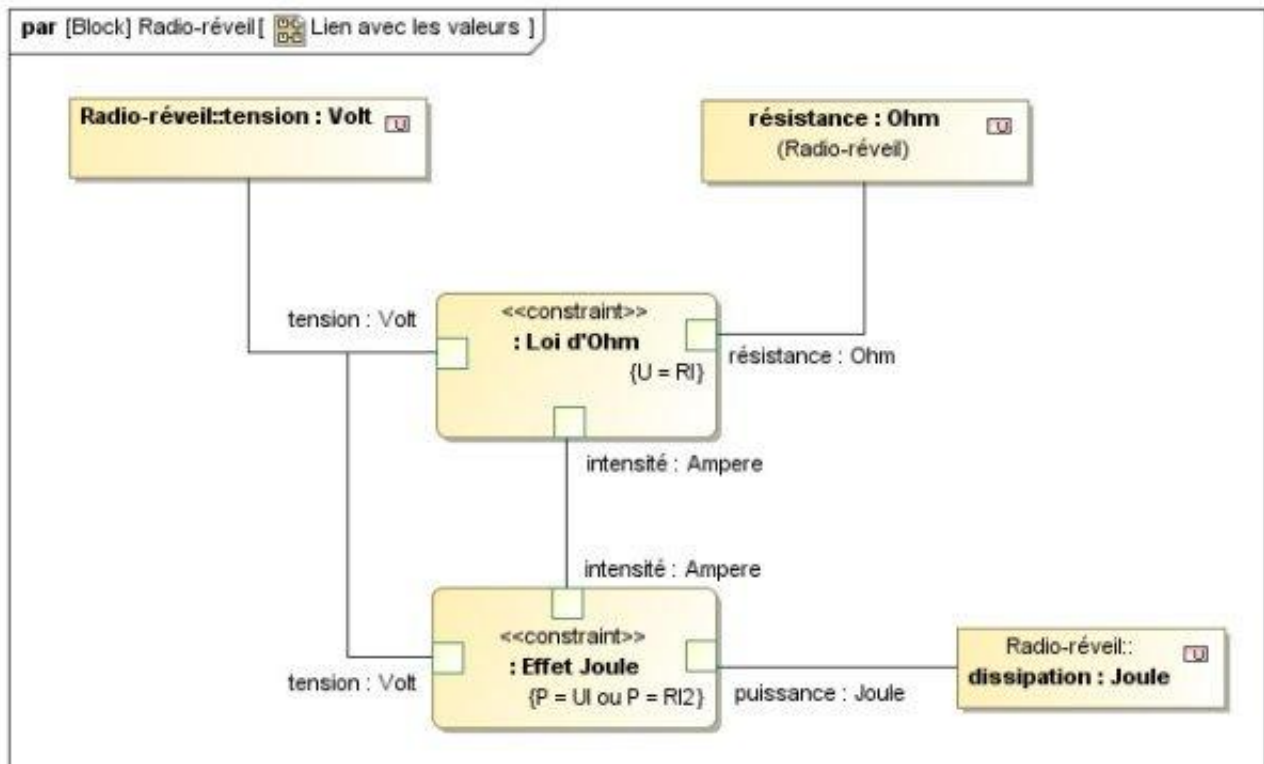
2.3.4. Le diagramme paramétrique (Parametric Diagram)

- **Rôle :**
 - ✓ Il permet de représenter des contraintes sur des valeurs de paramètres système, telles que performance, fiabilité, masse...
 - ✓ Il s'agit d'une spécialisation du diagramme de blocs internes où les seuls blocs utilisables sont des contraintes entre paramètres permettant de représenter graphiquement des équations et des relations mathématiques.
- **Représentation :**
 - ✓ Chaque contrainte doit être définie par des paramètres ainsi qu'une règle décrivant l'évolution de ces derniers.
 - ✓ Une contrainte est représentée par un bloc avec le stéréotype <<constraint>>
 - ✓ Les contraintes peuvent également apparaître dans le bloc de l'élément englobant.
 - ✓ au niveau du diagramme paramétrique, les contraintes sont représentées par des diagrammes aux coins arrondis.
- **Exemple : radio réveil**



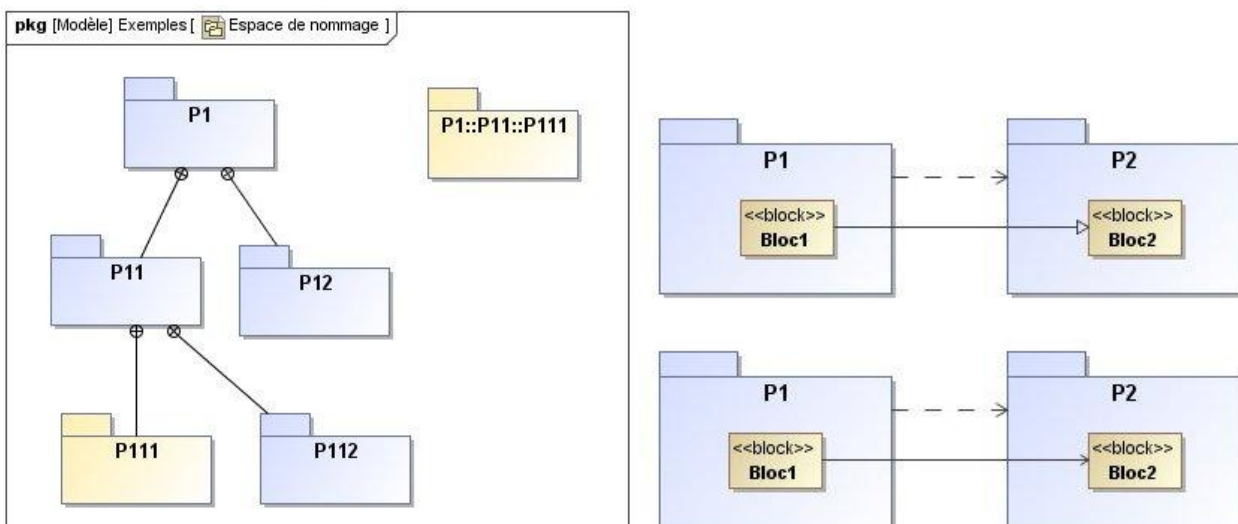
- **Complément :**

La contrainte s'instancie en reliant des valeurs aux paramètres formels



2.3.5. Le diagramme de package (Package Diagram)

- **Rôle :**
 - ✓ En SysML, tout élément du modèle est contenu que dans un seul conteneur. Si ce conteneur est détruit ou dupliqué, les éléments qu'il contient sont également détruits ou dupliqués. Le diagramme de package permet de regrouper les éléments d'un modèle par type, fonction....
- **Représentation :**
 - ✓ Chaque package est représenté sous la forme d'un dossier.
 - ✓ il existe 2 types de relations entre packages : la contenance et la dépendance



• Exemple : radio réveil

