I – La langage SysML

Afin de répondre aux exigences et aux besoins de la société, de concrétiser les innovations pour
améliorer l'existant ou développer de nouveaux produits, les systèmes intègrent des approches
pluritechniques.

Par ailleurs, la disparité des outils existants aujourd'hui, souvent propres à chaque domaine rend
difficile une spécification cohérente ainsi que la communication et la compréhension au sein d'une
équipe regroupant des spécialistes de plusieurs disciplines.

Le langage de modélisation objet SysML (System Modelling Language) s'appuie sur une description graphique des systèmes en utilisant un certain nombre de diagrammes et permet de représenter les composants et les flux de toutes natures dont notamment :

- Les constituants du système
- Les programmes informatiques
- Les flux d'information
- Les flux d''énergie.

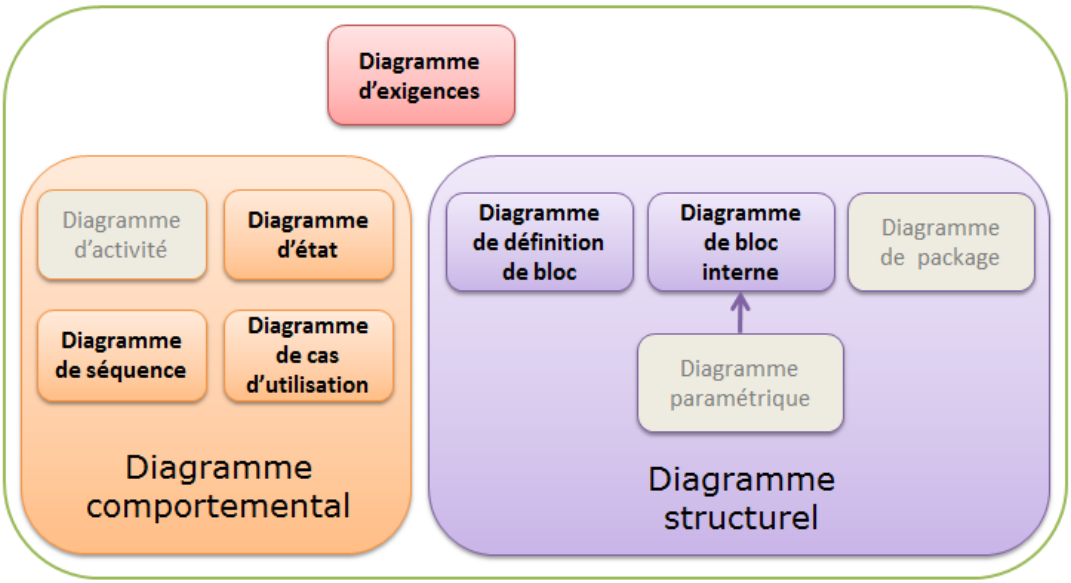
Le langage SysML permet de décrire de façon abstraite à travers différents points de vue
cohérents les systèmes afin d'en permettre la compréhension et l'analyse.

II - Diagrammes et syntaxe retenus en STI2D

II-1 Diagrammes utilisés

SysML est un langage puissant, mais on se limitera à la lecture et à l'interprétation des diagrammes suivants :

- Diagramme des cas d'utilisation (Use Case Diagram) ;
- Diagramme des exigences (Requirement Diagram) ;
- Diagramme de séquences (Sequence Diagram) ;
- Diagramme de définitions de blocs (Definition Block Diagram) ;
- Diagramme de blocs interne (Internal Block Diagram) ;
- Diagramme d'états (State Diagram).

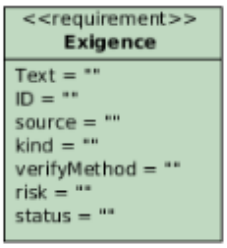


Ces diagrammes ne sont pas indépendants et permettent d'associer les éléments de diagrammes
différents. C'est l'un des points forts de ce type de langage.
Il est ainsi possible de conserver la traçabilité des éléments dans les différents diagrammes, par exemple :

- Lier une exigence avec des blocs : cela permet d'avoir le lien fonctions – solutions ;
- Lier des états avec les blocs : cela permet d'avoir le lien entre les actions et les composants qui les réalisent ;
- Lier les cas d'utilisation avec les scénarii des diagrammes de séquences.

exigences

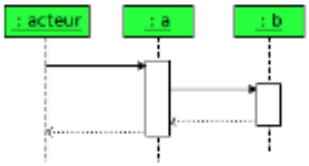
Représenter les contraintes techniques ou non du système.



cas d'utilisation

Représenter les fonctionnalités attendues du système dans leur contexte.

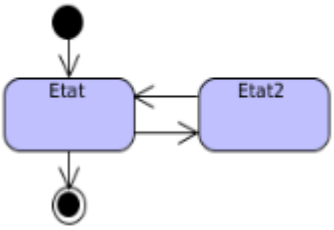
séquence



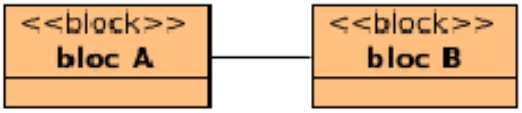
Décrire les échanges au sein d'un cas ou plusieurs cas d'utilisation.

états-transitions

Illustrer les changements d'états d'un système ou sous-système.



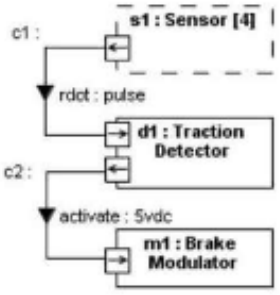
blocs



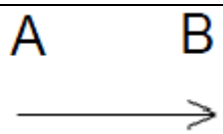
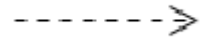
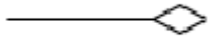
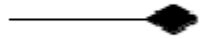
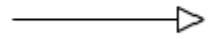
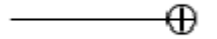
Représenter la structure globale d'un système

bloc interne

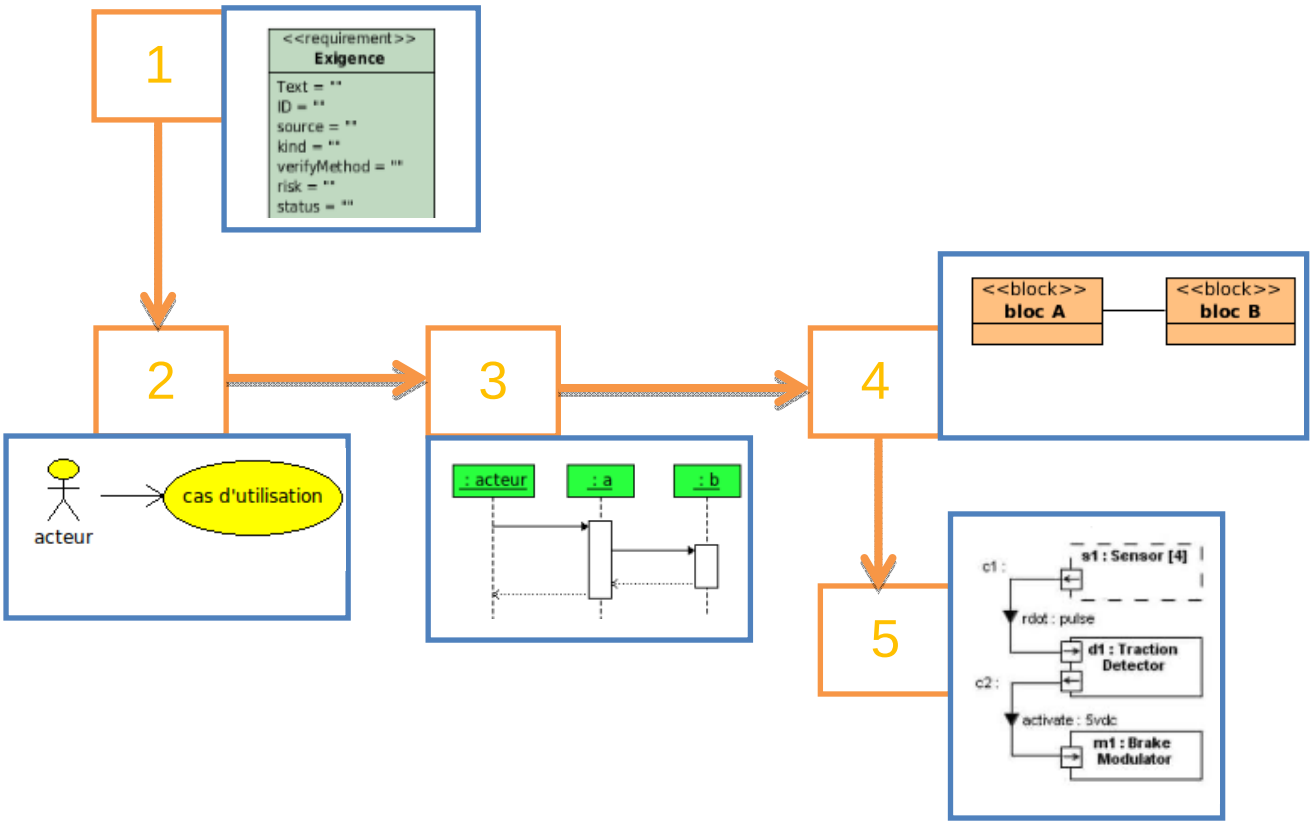
Illustrer les liens et flux entre les blocs



II-2 Les principales relations

	**Association** : relation d'égal à égal entre 2 éléments ➢ A utilise B ➢ 2 diagrammes : cas d'utilisation, blocs
	**Dépendance** : 2 items indépendants mais dont l'un dépend de l'autre. ➢ A dépend de B ➢ 3 diagrammes : exigences, cas d'utilisation, blocs,
	**Agrégation** : un élément est une composante facultative d'un autre. ➢ A entre dans la composition de B (sans être indispensable) ➢ 2 diagrammes : exigences, blocs
	**Composition** : un élément est une composante obligatoire d'un autre. ➢ A entre dans la composition de B et lui est indispensable ➢ 2 diagrammes : exigences, blocs
	**Généralisation** : dépendance de type filiation entre 2 items ➢ A est une sorte de B ➢ 2 diagrammes : cas d'utilisation, blocs
	**Conteneur** : relation d'inclusion entre 2 items ➢ B contient A ➢ 3 diagrammes : exigences, cas d'utilisation, blocs,

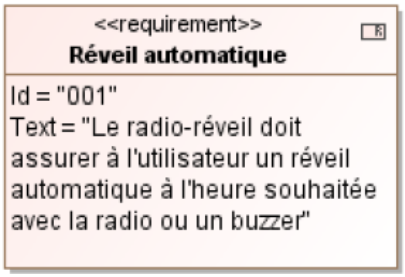
II – 3 Exemple de démarche possible



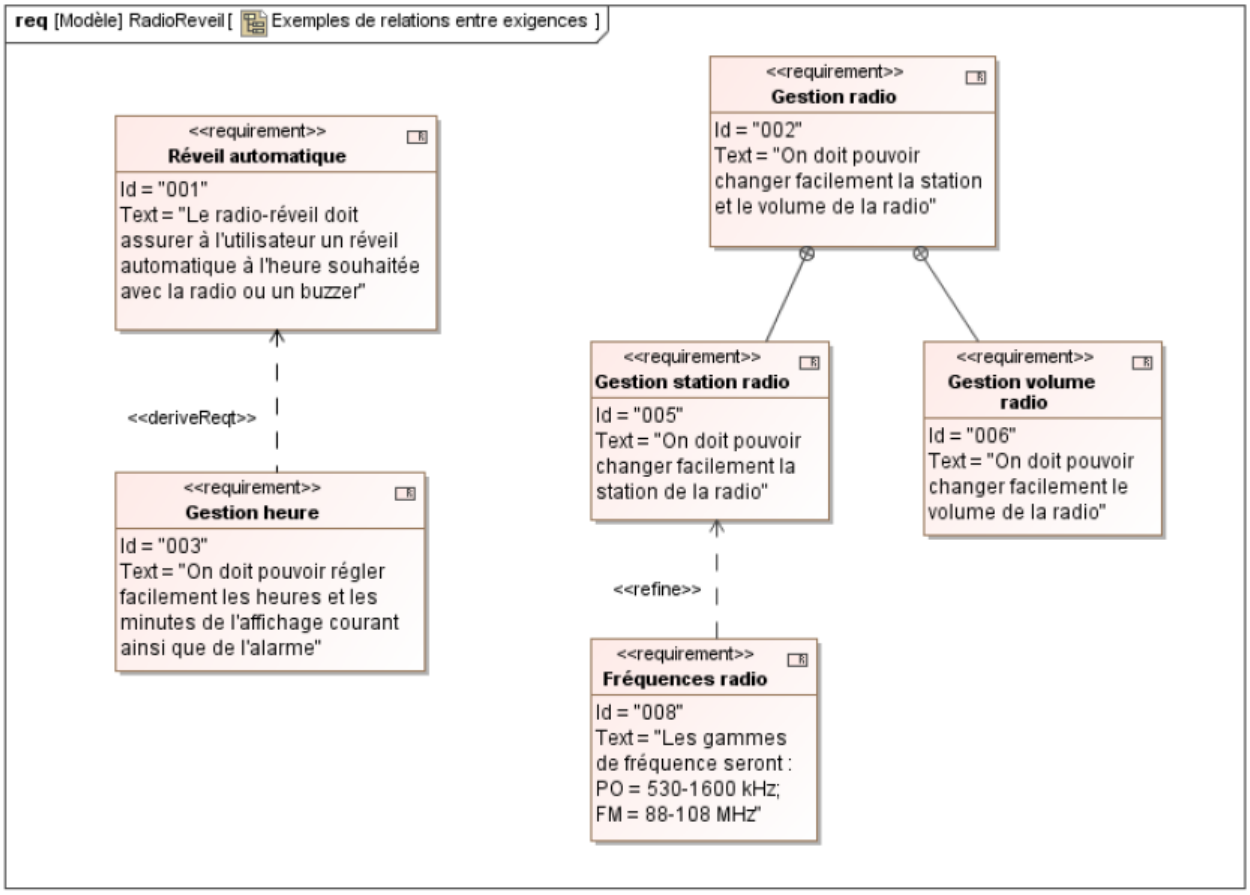
III – Exemple : Le radio réveil

III.1 Diagramme d'exigences

La première exigence fondamentale concerne la capacité à assurer à l'utilisateur un réveil automatique à l'heure souhaitée avec la radio ou un buzzer.



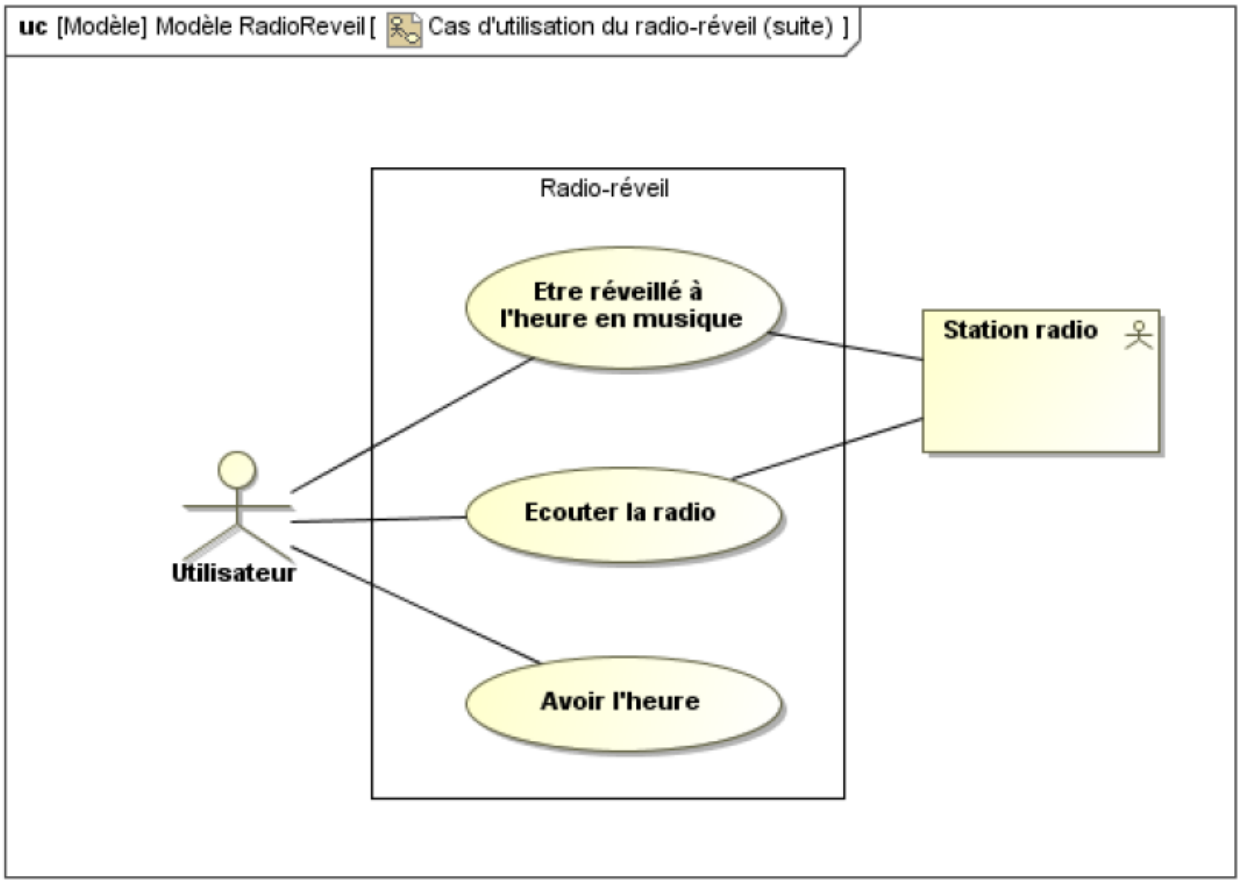
On peut également lister des exigences sur le réglage de la radio, de l'horloge et de l'alarme, ainsi que sur la nécessité d'un mécanisme de sauvegarde.



III.2 Diagramme de cas d'utilisation

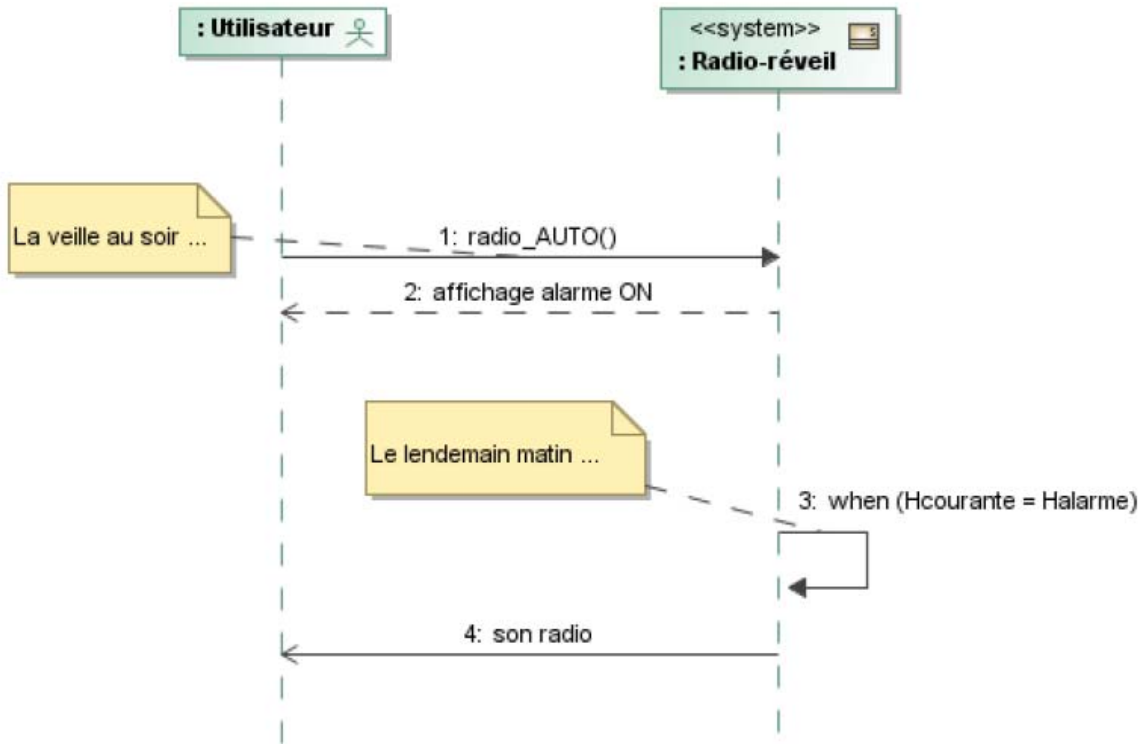
Pour notre étude de cas, une première version du diagramme de cas d'utilisation consiste à considérer un seul acteur (l'utilisateur) connecté à un unique cas d'utilisation (être réveillé à l'heure en musique).

Ensuite, on peut se dire que l'utilisateur, alors qu'il est réveillé, est susceptible d'utiliser le radio-réveil en tant que simple radio ou horloge.



III.3 Diagramme de séquence

Un premier exemple de diagramme de séquence du cas d'utilisation *Être réveillé à l'heure en musique* est donné ci-dessous :



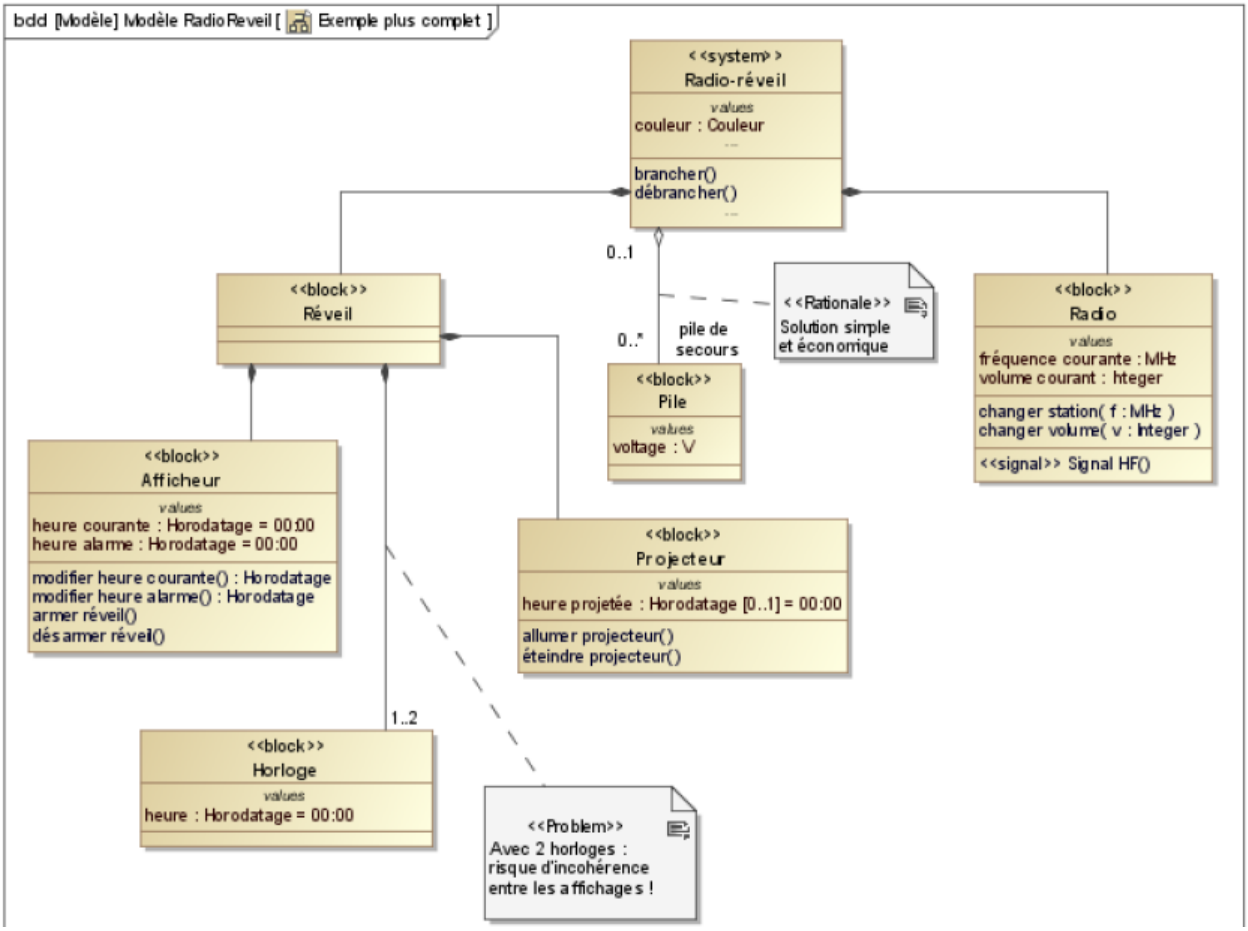
III.4 Diagramme de définition de bloc

Dans l'exemple suivant, nous avons modélisé le radio-réveil en tant que système à l'étude, avec une valeur : couleur, et deux parties : radio et réveil.

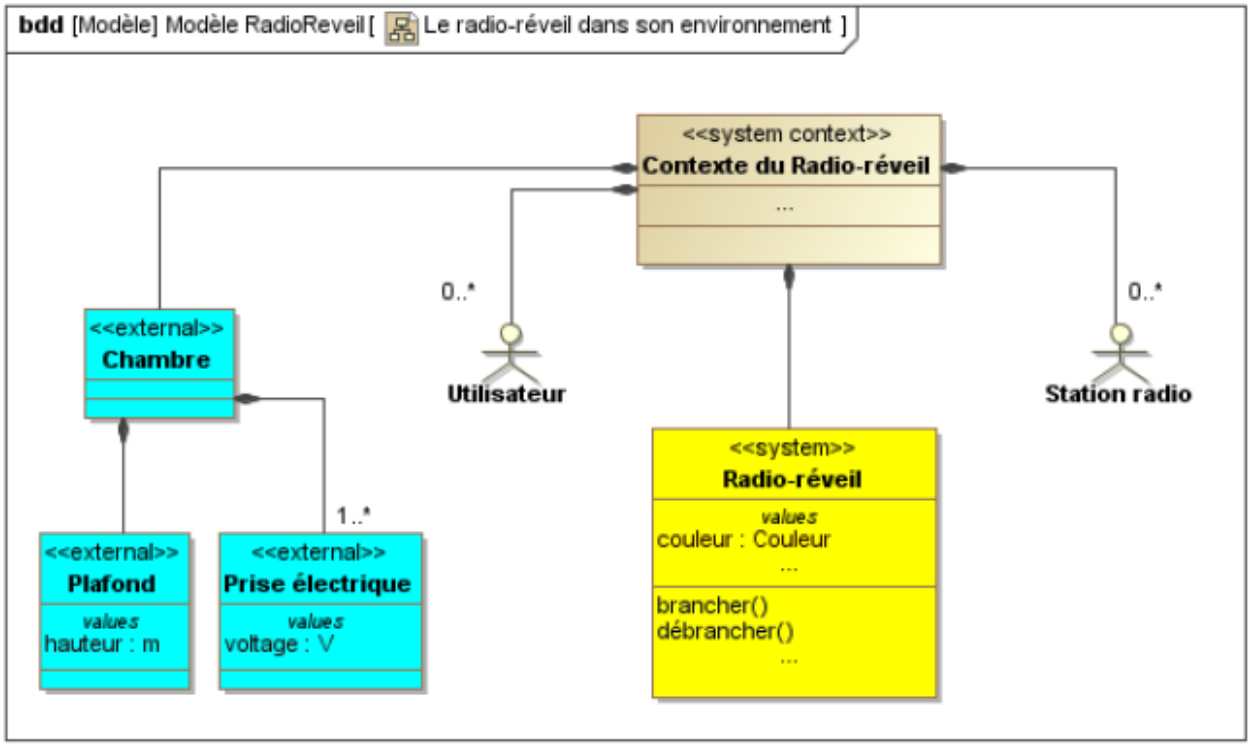
Nous pouvons préciser le type des parties et des valeurs :

- la partie radio est de type Radio (un nouveau bloc à créer) ;
- la partie réveil est de type Réveil (un nouveau bloc à créer) ;
- la valeur couleur est de type Couleur (une énumération à créer).

Exemple plus complet du diagramme de définition de bloc du radio réveil :



Exemple du diagramme de définition de bloc du radio réveil dans son environnement :



III.5 Diagramme de bloc interne partiel du radio-réveil

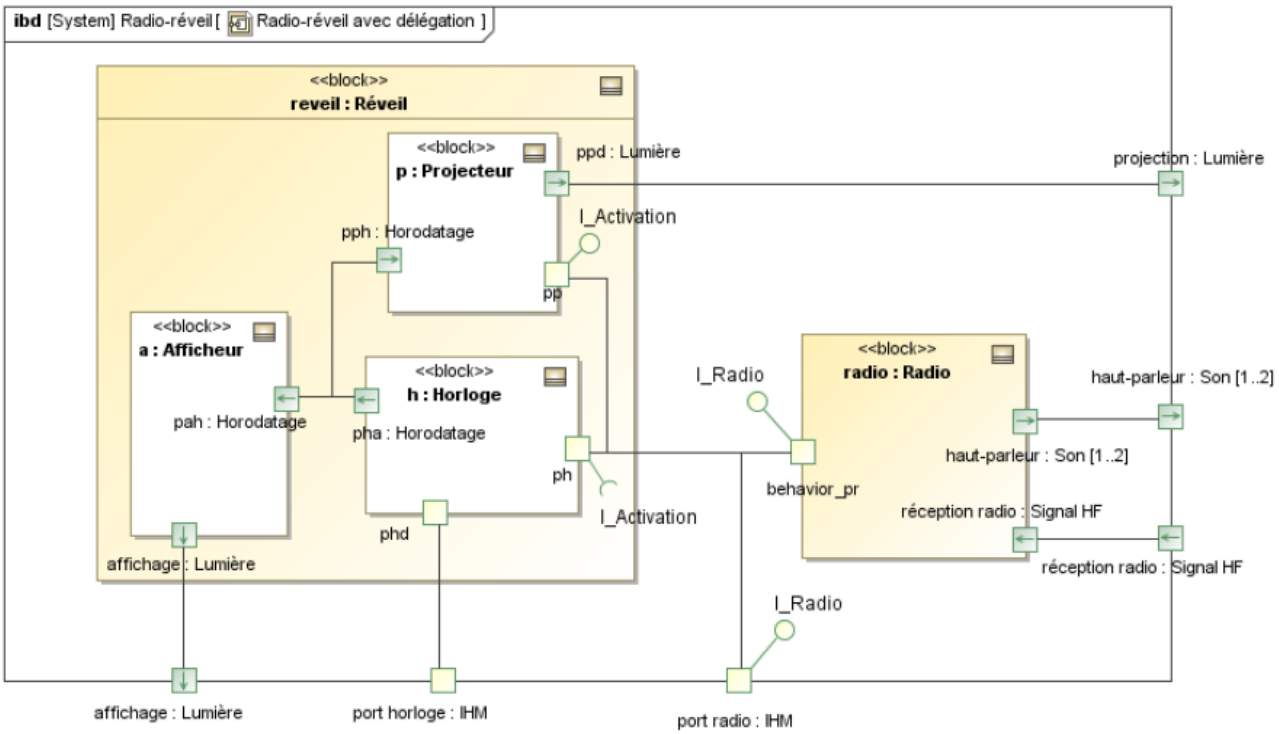
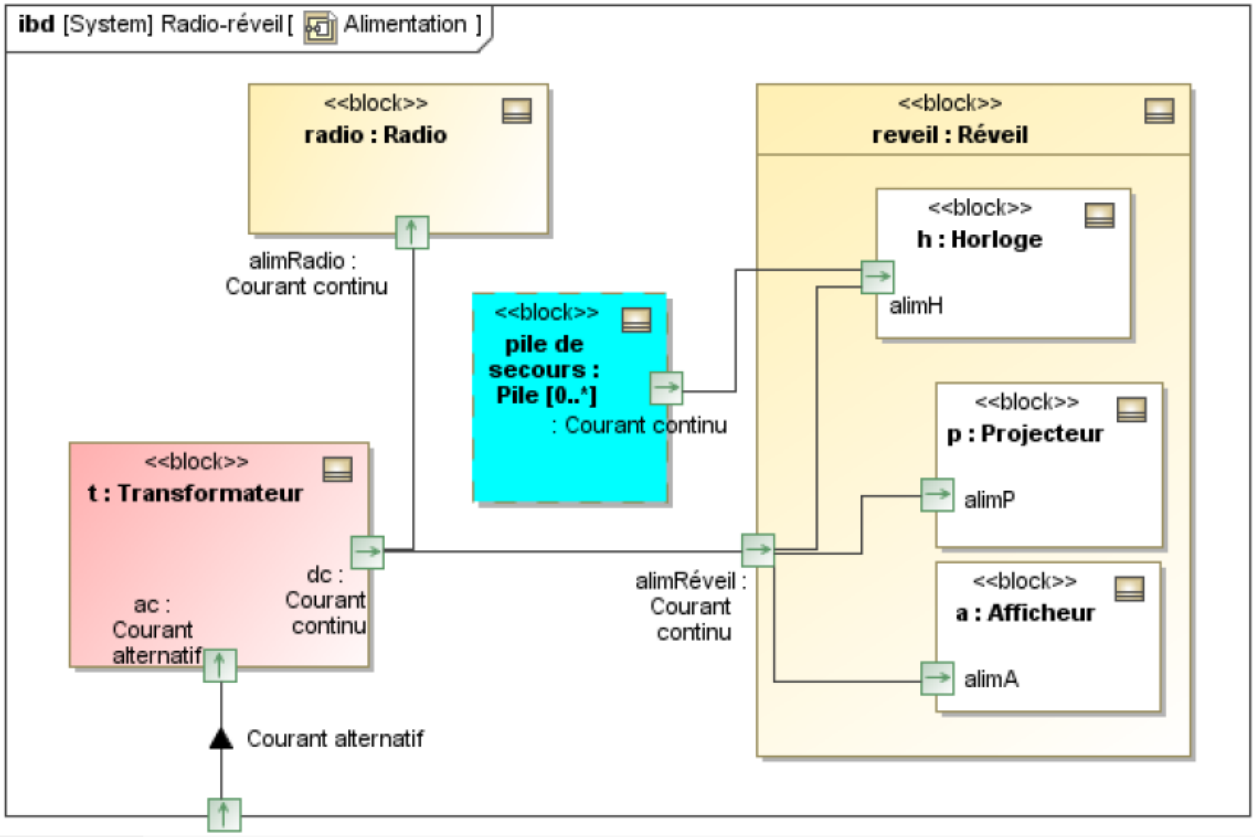
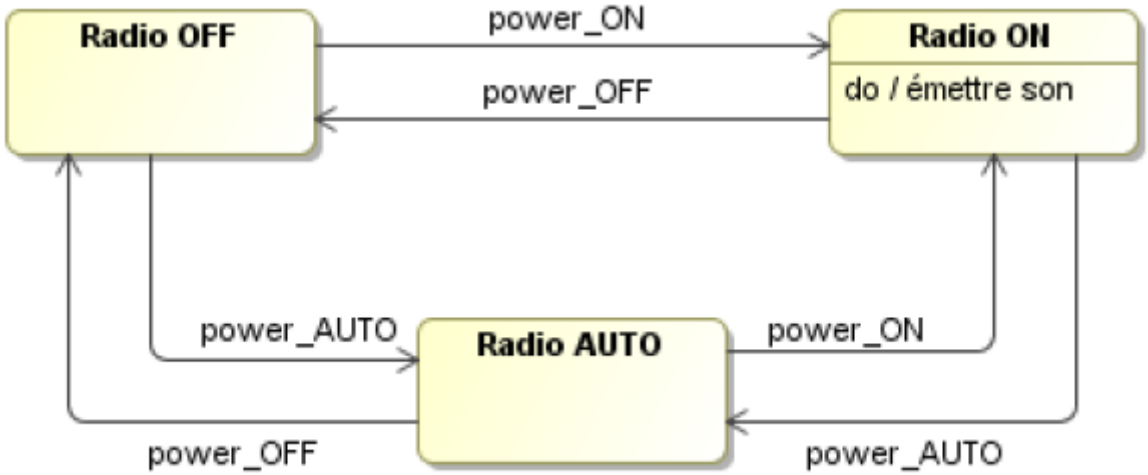


Diagramme de bloc interne du radio-réveil concernant l'alimentation :



III.6 Diagramme d'états

Commençons par déclarer trois états principaux correspondant aux trois positions du bouton physique permettant d'allumer la radio, de l'éteindre, ou d'armer l'alarme du réveil. Les événements de changement de position sont nommés power_ON, power_OFF,power_AUTO.



Soit ci-après le diagramme d'états avec gestion de l'alimentation :