

LES DIAGRAMMES SysML

1 Introduction

1.1 Les différents diagrammes

Pour rappel, SysML est un langage de modélisation des systèmes complexes. C'est un ensemble de diagrammes que l'on peut catégoriser de cette façon :

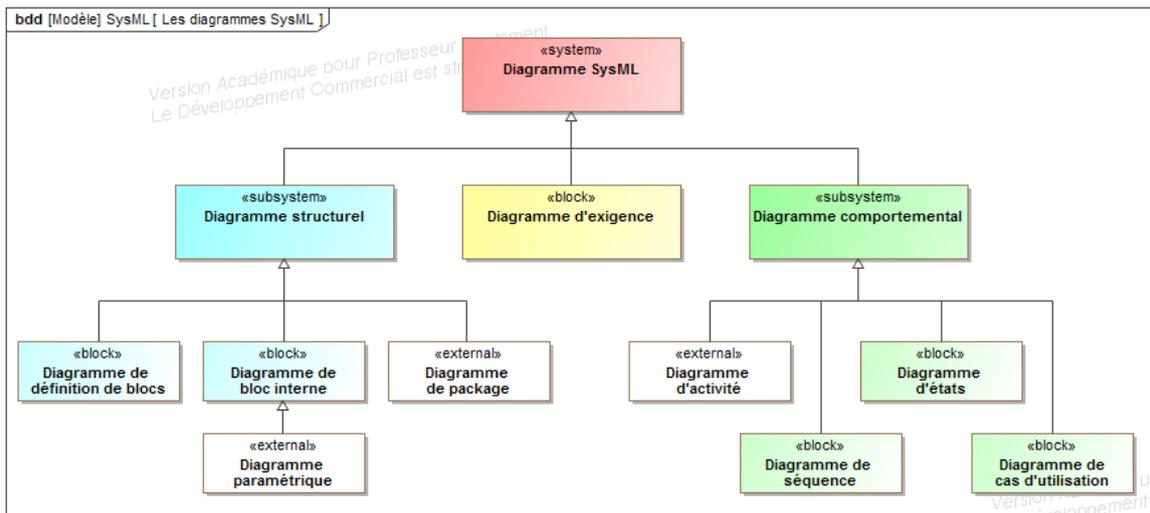


FIGURE 1 – Récapitulatif des diagrammes SysML

1.2 Éléments graphiques communs

Tous les diagrammes SysML comportent en haut à gauche un **cartouche**, qui permet d'identifier le diagramme en indiquant au minimum le type de diagramme et le nom de l'élément concerné.

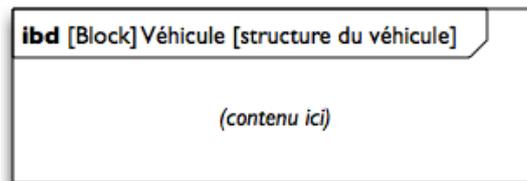


FIGURE 2 – Exemple de cartouche

1.3 SysML en PT

D'après les programmes officiels, les diagrammes d'activité (*act*), les diagrammes paramétriques (*par*) et les diagrammes de packages (*pkg*) ne sont pas à connaître en PT.

Par ailleurs, il ne peut vous être demandé d'écrire des diagrammes SysML, et la connaissance de la syntaxe ne peut être exigible. En revanche, la modification de diagrammes est au programme.

2 Les diagrammes de structure

2.1 Le diagramme de définition de blocs - *bdd*

Le diagramme de définition de blocs permet de définir une arborescence de blocs. Ces blocs pouvant modéliser un élément du contexte, une fonction ou un composant, le diagramme *bdd* aura une signification différente.

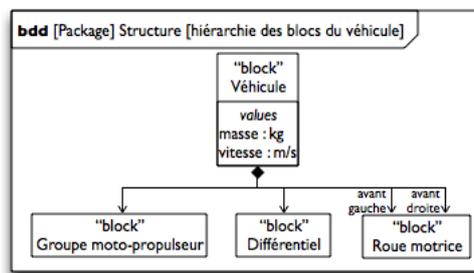


FIGURE 3 – Diagramme *bdd* simple

La description de la syntaxe est fournie ci-dessous :

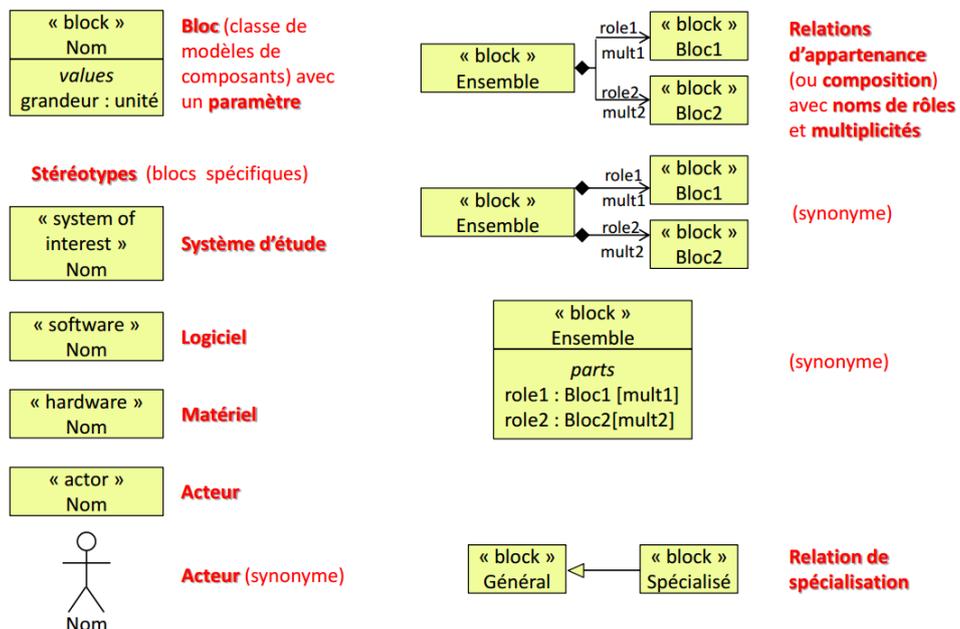


FIGURE 4 – Syntaxe du diagramme *bdd*

Il existe également la relation d'agrégation simple. Par rapport à la relation de composition (ou

agrégation forte), elle est modélisée par un losange vide. La nuance principale concerne le lien entre les deux blocs. Pour une composition, la destruction de l'élément supérieur entraîne la destruction de l'élément inférieur, ce qui n'est pas le cas pour une agrégation simple.

2.2 Le diagramme de bloc interne - *ibd*

Le diagramme *ibd* permet de montrer entre les différents sous-blocs d'un bloc et de spécifier les interfaces à l'aide de ports. Afin de distinguer l'utilisation de plusieurs sous-blocs d'une même catégorie, on utilise l'instance d'un bloc modélisée à l'aide des deux points :

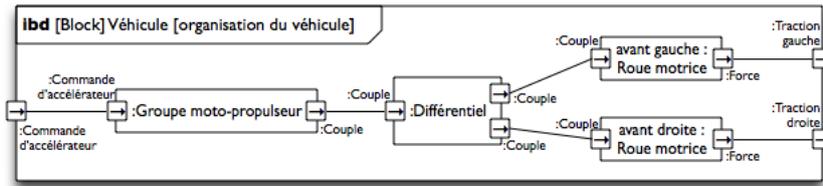


FIGURE 5 – Diagramme *ibd* simple

La description de la syntaxe est fournie ci-dessous :

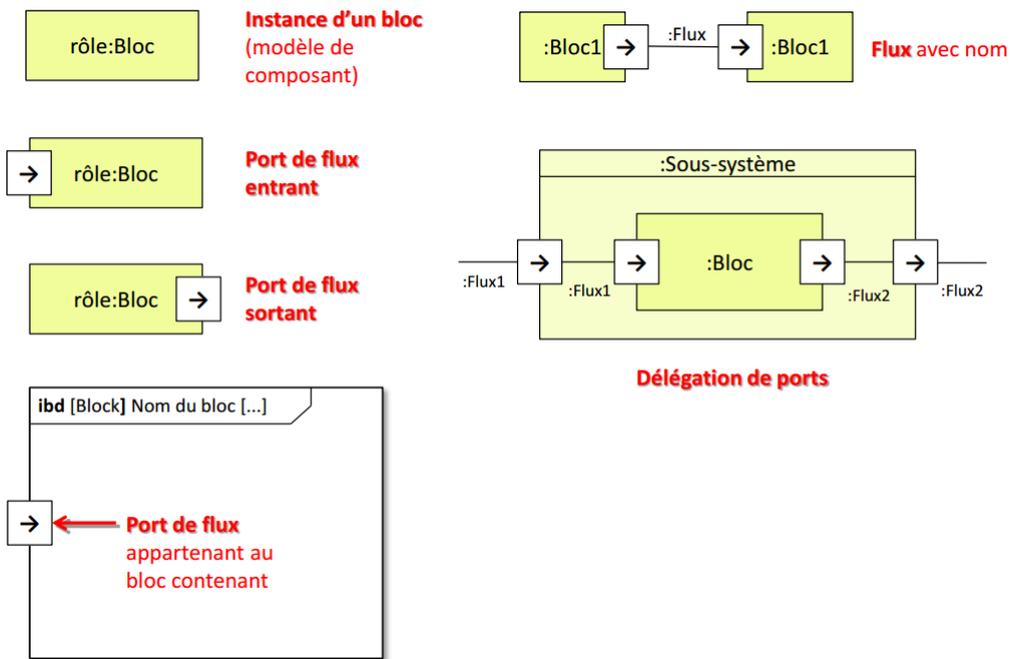


FIGURE 6 – Syntaxe du diagramme *ibd*

2.3 Le diagramme paramétrique - *par*

Le diagramme paramétrique permet de spécifier le comportement attendu d'un bloc sous la forme d'équations reliant les sorties aux entrées.

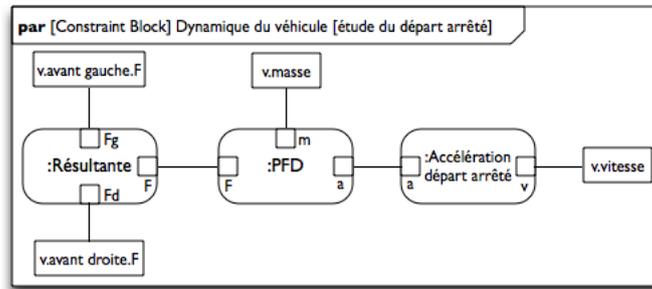


FIGURE 7 – Diagramme paramétrique simple

La description de la syntaxe est fournie ci-dessous :

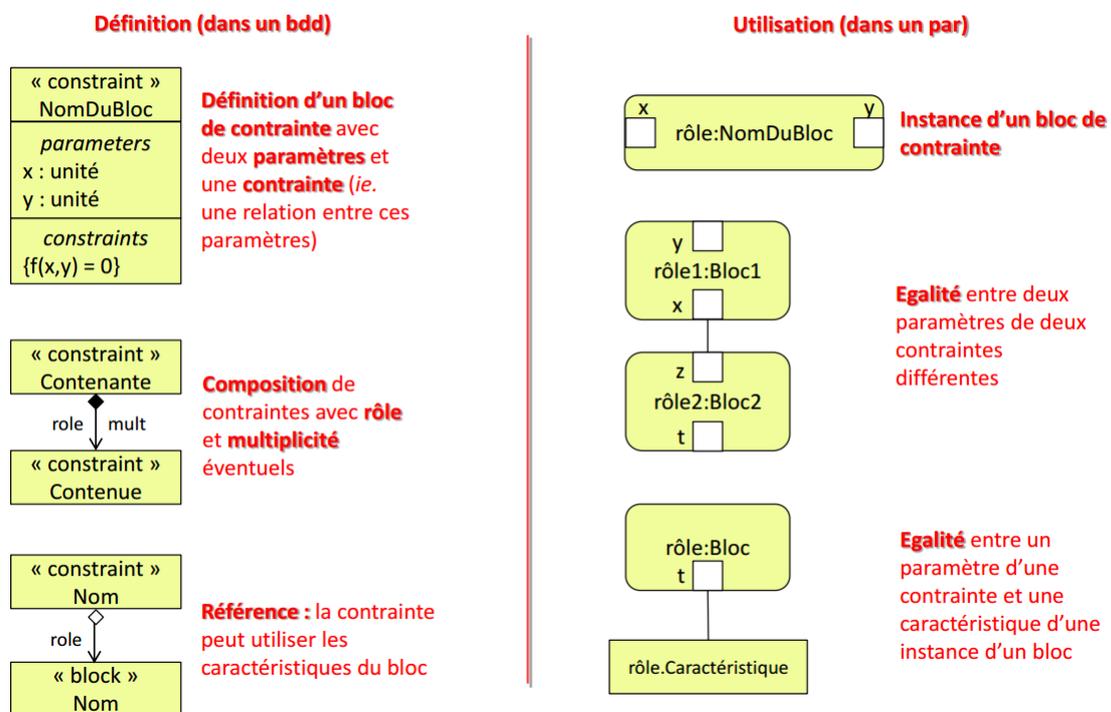


FIGURE 8 – Syntaxe du diagramme paramétrique

2.4 Le diagramme de package - *pkg*

Celui-ci est hors-programme et il est très peu utilisé dans l'industrie.

3 Le diagramme d'exigences - req

Le diagramme des exigences permet de montrer les liens entre les différentes exigences et les autres éléments du modèles.

La description de la syntaxe est fournie ci-dessous :

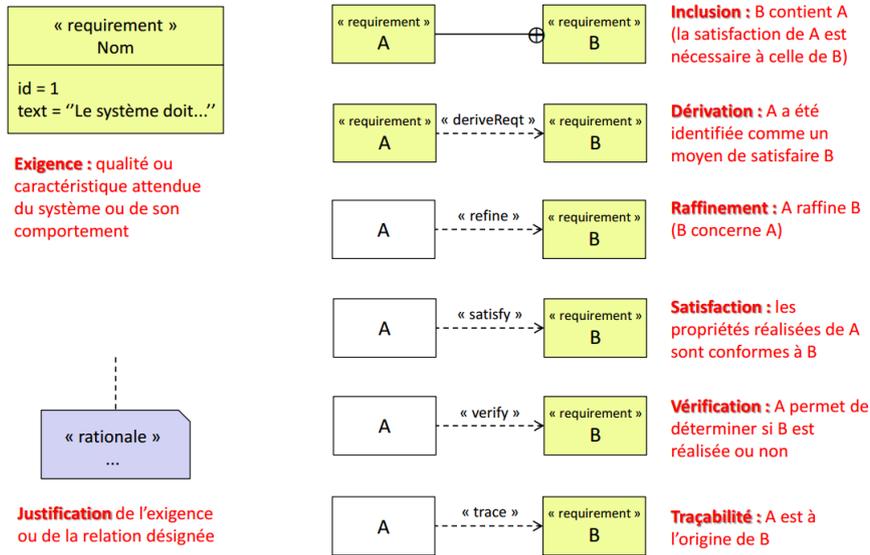


FIGURE 9 – Syntaxe du diagramme des exigences

4 Les diagrammes comportementaux

4.1 Diagramme d'activités - act

Le diagramme des activités permet de montrer l'enchaînement d'activités. Les deux types de liens permet de montrer les flux d'objets (traits pleins) et les flux de commande (traits pointillés).

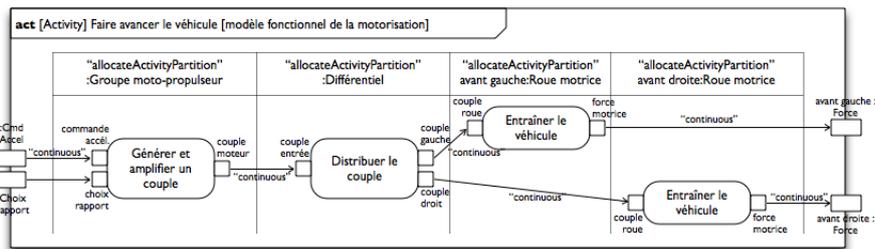


FIGURE 10 – Diagramme d'activités simple

La description de la syntaxe est fournie ci-dessous :

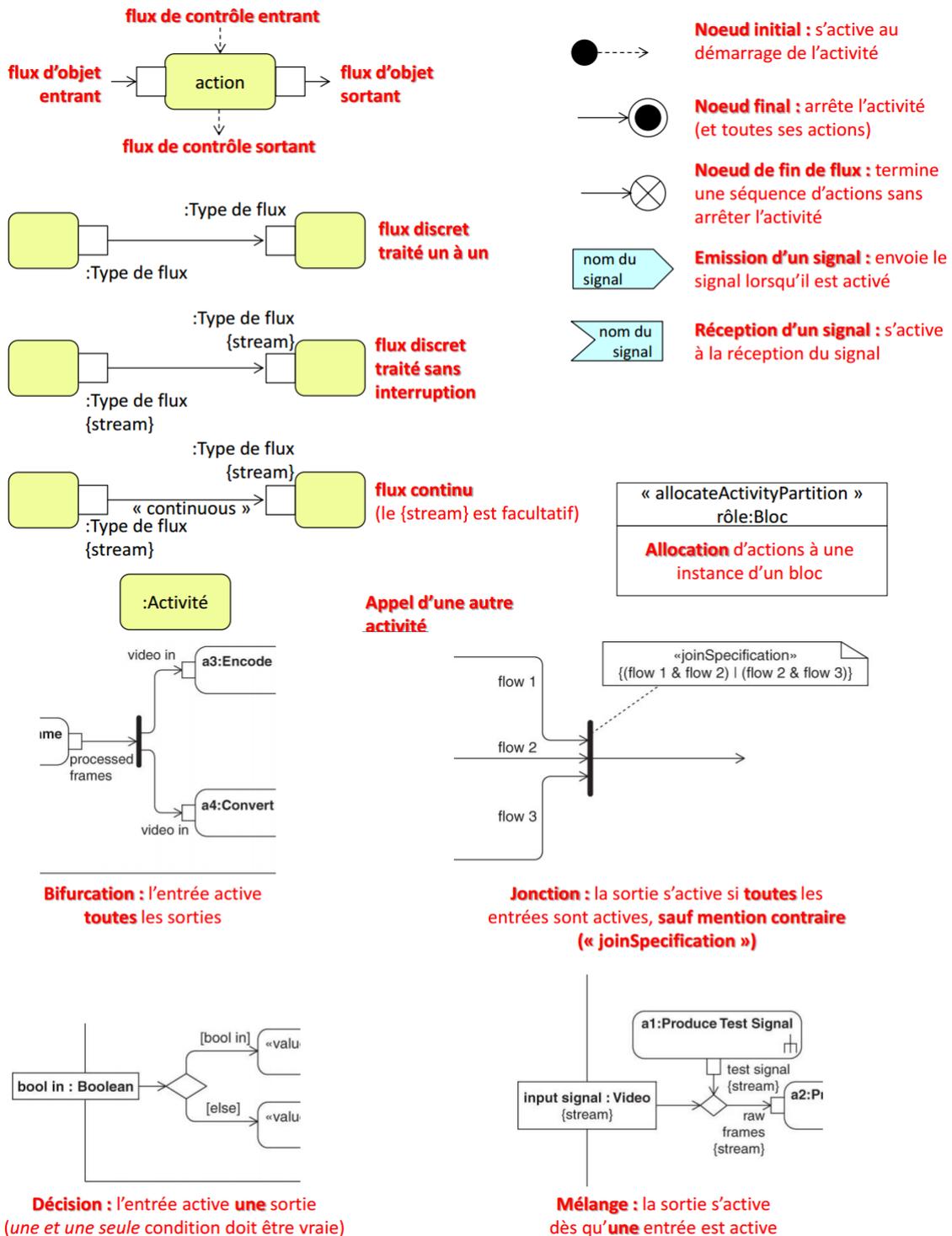


FIGURE 11 – Syntaxe du diagramme d'activités

4.2 Le diagramme de séquence - *sd*

Le diagramme de séquence est un autre diagramme comportemental proche du diagramme d'activités. Il permet de montrer les interactions entre différents objets (blocs, acteurs...).

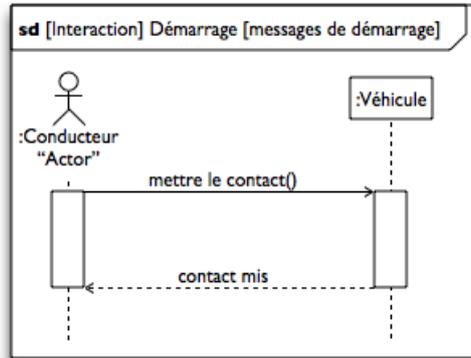


FIGURE 12 – Diagramme de séquence simple

La description de la syntaxe est fournie ci-dessous :

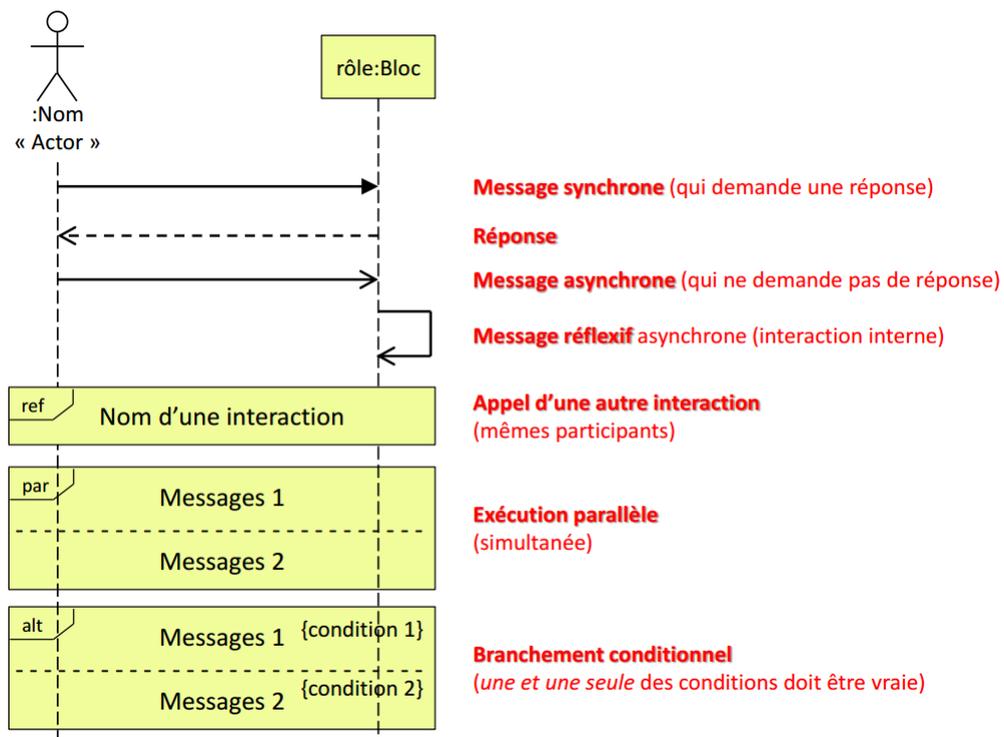


FIGURE 13 – Syntaxe du diagramme de séquence

4.3 Le diagramme d'états - *stm*

Comme cela sera vu dans un cours qui va suivre, un système peut être dans un état ou un mode particulier. C'est pour définir ces différents états et les transitions entre ces états que le diagramme d'états (ou statemachine, ou statechart) est utile.

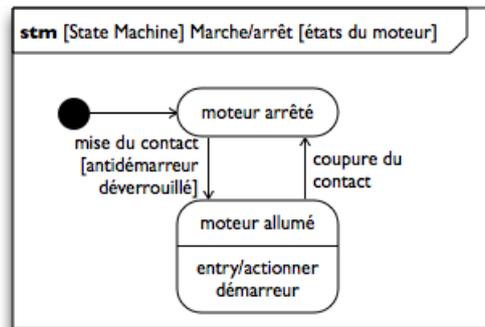


FIGURE 14 – Graphe d'états simple

La description de la syntaxe est fournie ci-dessous :

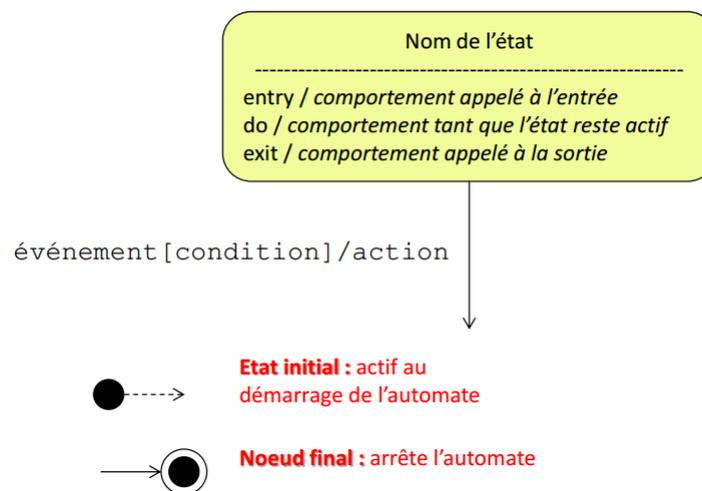


FIGURE 15 – Syntaxe des Statecharts

4.4 Le diagramme des cas d'utilisation - uc

Le diagramme des cas d'utilisation permet de montrer les cas d'utilisation d'un système par rapport à son environnement.

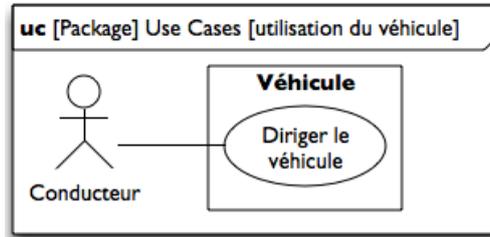


FIGURE 16 – Diagramme des cas d'utilisation simple

La description de la syntaxe est fournie ci-dessous :

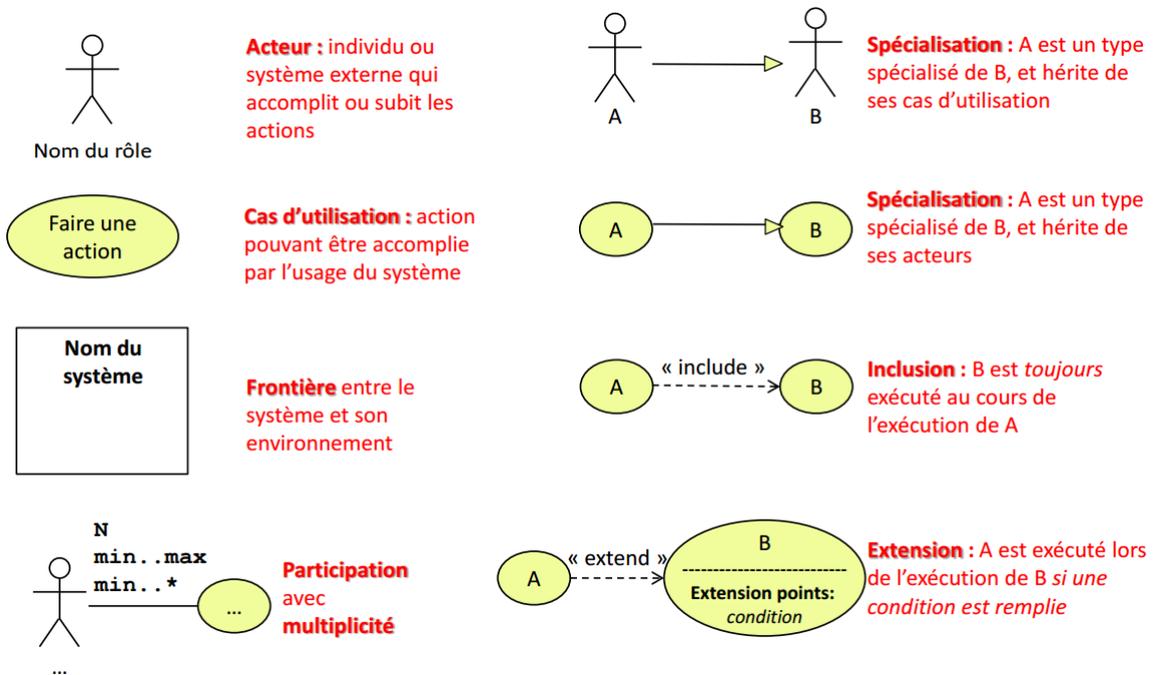


FIGURE 17 – Syntaxe du diagramme des cas d'utilisation