

# SysML : les diagrammes

DIDIER FAGNON, STÉPHANE GASTON <sup>[1]</sup>

**L'outil SysML est un langage constitué de nombreux diagrammes. Nous vous proposons une ressource sous la forme de fiches-outils qui trouveront une bonne place dans vos aide-mémoire de TP.**

Dans les pages « Décryptage » de la revue, nous vous avons présenté l'utilisation raisonnée de SysML (*Systems Modeling Language*) dans l'analyse fonctionnelle d'un système existant. Ce langage, commun à tous les champs disciplinaires, est composé de diagrammes qui permettent d'aborder plus facilement les systèmes pluritechniques, que ce soit en phase de

[1] Respectivement professeur agrégé de génie mécanique au lycée Chaptal de Paris (75008) et professeur de construction mécanique au lycée Denis-Papin de La Courneuve (93).

## mois-clés

lycée technologique, modélisation, outil & méthode



A → B	<b>Association</b> : relation d'égal à égal entre deux éléments <b>A utilise B</b> Est utilisé dans 2 diagrammes : cas d'utilisation, définition de blocs
.....→	<b>Dépendance</b> : 2 items distincts mais dont l'un dépend de l'autre <b>A dépend de B</b> Est utilisé dans 3 diagrammes : exigences, cas d'utilisation, définition de blocs
—◇	<b>Agrégation</b> : un élément est une composante facultative de l'autre <b>A entre dans la composition de B sans être indispensable à son fonctionnement</b> Est utilisé dans 2 diagrammes : exigences, définition de blocs
—◆	<b>Composition</b> : un élément est une composante obligatoire de l'autre <b>A entre dans la composition de B et lui est indispensable</b> Est utilisé dans 2 diagrammes : exigences, définition de blocs
—▷	<b>Généralisation</b> : dépendance de type « filiation » entre 2 items <b>A est une sorte de B</b> Est utilisé dans 2 diagrammes : cas d'utilisation, définition de blocs
—⊕	<b>Conteneur</b> : relation d'inclusion entre 2 items <b>B contient A</b> Est utilisé dans 3 diagrammes : exigences, cas d'utilisation, définition de blocs

## ■ Les principales relations entre les blocs

### OUVRAGE DE RÉFÉRENCE

*SysML par l'exemple*  
*Un langage de modélisation pour systèmes complexes*

Auteur : Pascal Roques

Éditeur : Eyrolles

Ce livre de 235 pages est exclusivement disponible en version PDF, payante, à télécharger sur le site de l'éditeur :

<http://izibook.eyrolles.com>

De la naissance du SysML aux exemples concrets, il consacre un chapitre entier à chacun des diagrammes. Il est très illustré et prend pour fil conducteur une étude de cas sur le radio-réveil. C'est l'ouvrage de référence en français. Indispensable.

conception ou en phase d'analyse d'un existant. Il peut également décrire le cheminement de la matière, de l'énergie et de l'information (les trois domaines essentiels du référentiel des STI2D), ses diagrammes offrant la possibilité de représenter à la fois les composants et les flux de toute nature. Nous allons donc nous attacher ici à décoder le vocabulaire et la grammaire de cet outil d'aide à la modélisation.

Les diagrammes SysML, le plus souvent, sont liés entre eux (interconnectés) et ont leur description propre. Ils peuvent remplacer la plupart des autres outils de description auparavant utilisés (Grafcet, Fast, SADT, etc). Voici, parmi les neuf types de diagrammes SysML, les six qui ont été retenus par le référentiel :

diagramme des exigences (*requirement diagram*)  
diagramme des cas d'utilisation (*use case diagram*)  
diagramme de séquence (*sequence diagram*)  
diagramme d'état (*state diagram*)  
diagramme de définition de blocs (*definition block diagram*)  
diagramme de blocs internes (*internal block diagram*)

Ils décrivent différents aspects :

#### ● Aspects comportementaux :

– diagrammes fonctionnels (que doit faire le système?) :

diagramme des exigences  
diagramme des cas d'utilisation

– diagrammes dynamiques (comment le système doit-il se comporter?) :

diagramme de séquence  
diagramme d'état (état et transitions)

#### ● Aspects structurels :

– diagrammes statiques (comment le système est-il construit?) :

diagramme de définition de blocs  
diagramme de blocs internes



Dans les diagrammes, on trouve des boîtes et des relations entre ces boîtes. Ces relations peuvent être de natures différentes, elles s'expriment donc avec des symboles différents (voir la figure page précédente).

Nous allons détailler les éléments graphiques relatifs à chaque diagramme pour en faciliter la lecture, sous forme de fiches-outils que vous pourrez utiliser ou fournir au fur et à mesure aux élèves. Ces fiches

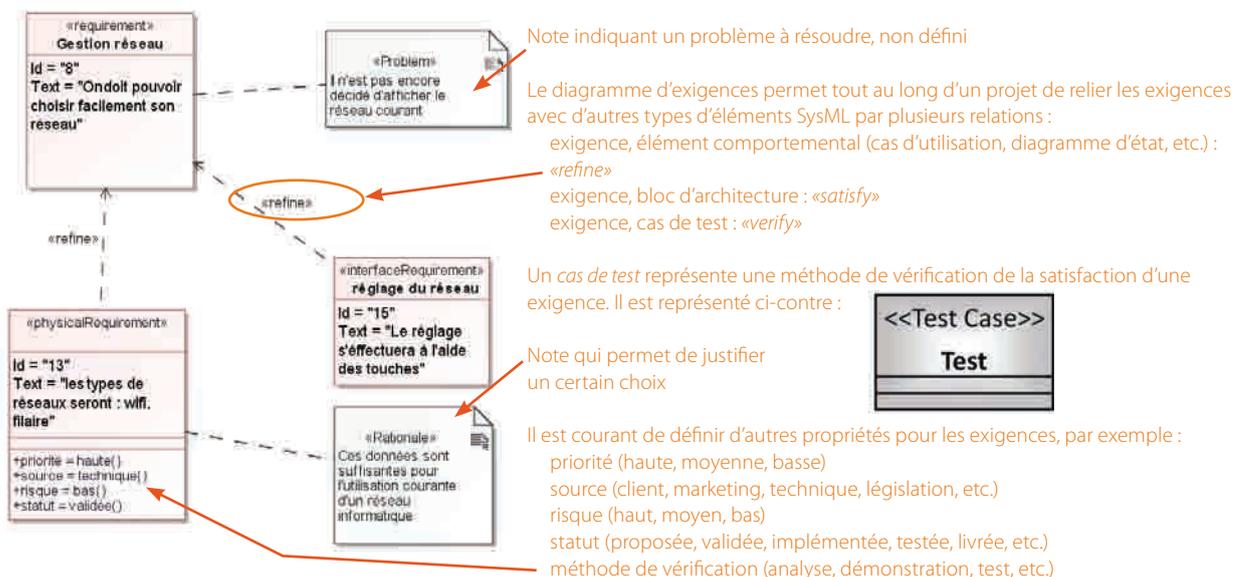
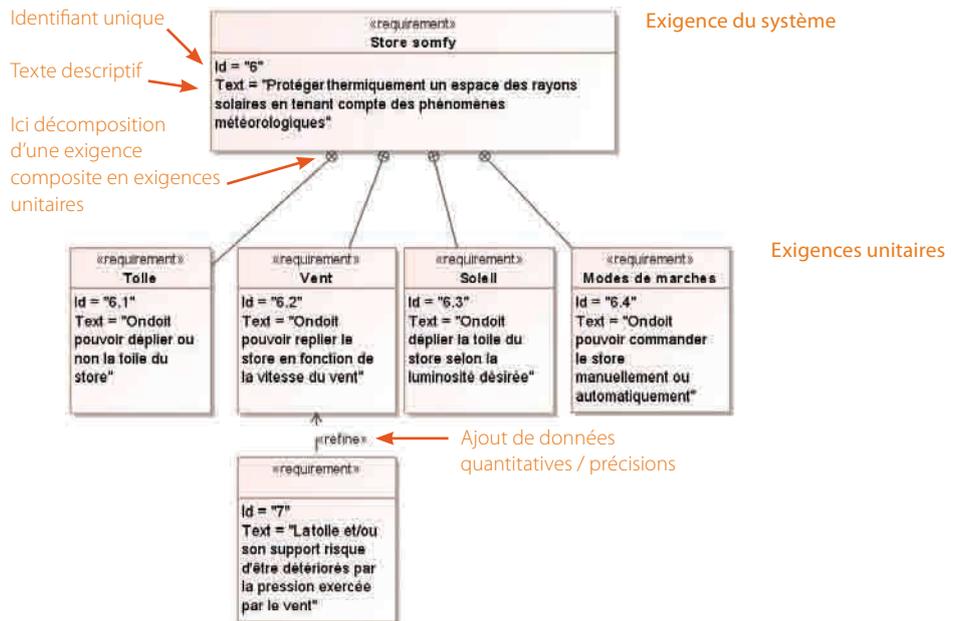
peuvent aussi constituer un document ressource dans un dossier de TP. Il n'est surtout pas question de les utiliser comme support de cours pour apprendre le langage SysML. En revanche, elles permettent, au fur et à mesure que l'enseignant aborde dans ses pratiques l'analyse des trois points de vue (comportemental, structurel et fonctionnel), d'introduire les diagrammes adaptés aux besoins pédagogiques. ■

FICHE-OUTIL 1

Diagramme des exigences

En anglais : *requirement diagram*  
Notation SysML: *req*

C'est un diagramme fonctionnel. Il décrit les exigences du cahier des charges fonctionnel. Une exigence exprime une capacité ou une contrainte à satisfaire par un système. Elle peut exprimer une fonction que devra réaliser le système ou une condition de performance technique, physique, de sécurité, de fiabilité, d'ergonomie, d'esthétisme...



FICHE-OUTIL 2

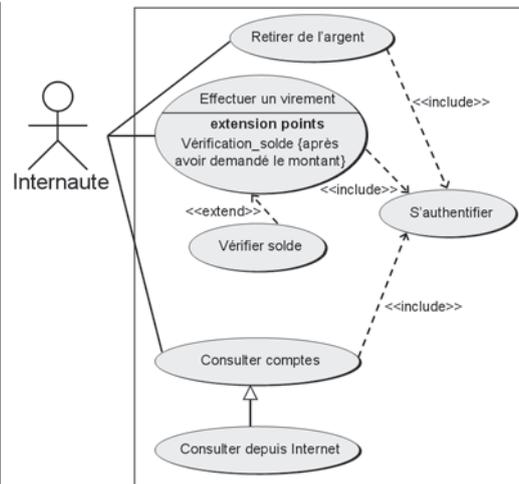
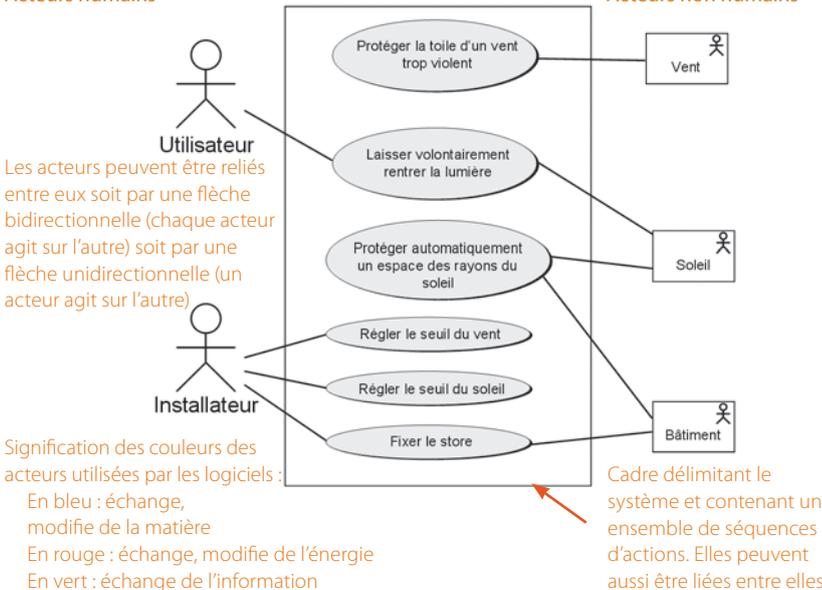
Diagramme des cas d'utilisation

En anglais : *use case diagram*  
 Notation SysML: *uc*

C'est un diagramme fonctionnel. Il montre les interactions fonctionnelles des acteurs et du système d'étude. Il délimite précisément le système, décrit ce que fera le système sans spécifier comment (et non ce que fera l'utilisateur). Il exprime les services (*use cases*) offerts par le système aux utilisateurs (*actors*).

Acteurs humains

Acteurs non humains



Relations standardisées entre cas d'utilisation

`<<include>>` : le cas d'utilisation de base « ne peut se faire sans » ou « impose que »  
`<<extend>>` : le cas d'utilisation de base « peut éventuellement se faire avec »  
 Ci-dessus : *Vérifier solde* est une extension d'*Effectuer un virement*

FICHE-OUTIL 3

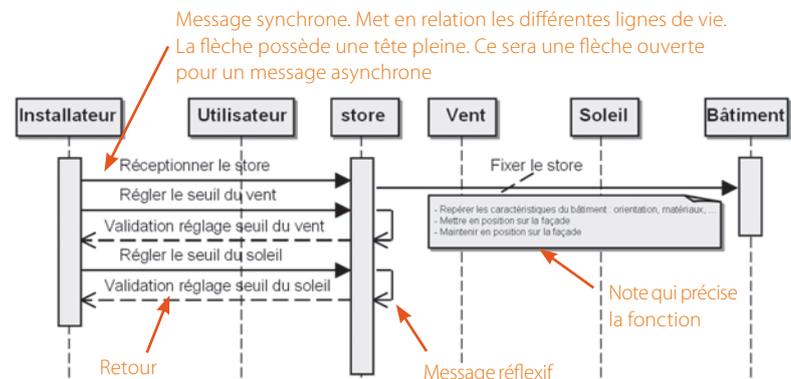
Diagramme de séquence

En anglais : *sequence diagram*  
 Notation SysML: *sd*

C'est un diagramme dynamique. Il représente les échanges de messages entre les acteurs et le système ou entre des parties durant une séquence temporelle d'actions appelée *scénario*. Il permet de représenter la dynamique d'un *use case* ou la collaboration d'un ensemble d'objets internes au système. Il montre la séquence, représentation verticale chronologique, des messages passés entre blocs au sein d'une interaction. Nota : Il y a autant de diagrammes de séquence que de scénarios possibles.

Lignes de vie

Trait pointillé vertical surmonté d'un rectangle de description



Un cas d'utilisation peut dépendre d'autres cas d'utilisation. Il s'appelle alors fragment. Chaque fragment possède un opérateur et peut être divisé en opérands. Les principaux opérateurs sont :  
*loop* : boucle. Le fragment ou autre cas d'utilisation peut s'exécuter plusieurs fois, et la condition de garde (voir « Diagramme d'état ») explicite l'itération  
*opt* : optionnel. Le fragment ou autre cas d'utilisation ne s'exécute que si la condition fournie est vraie  
*alt* : fragments alternatifs. Seul le fragment possédant la condition vraie s'exécutera  
*par* : parallèle. Il est utilisé pour représenter des interactions ayant lieu en parallèle

FICHE-OUTIL 4

Diagramme d'état

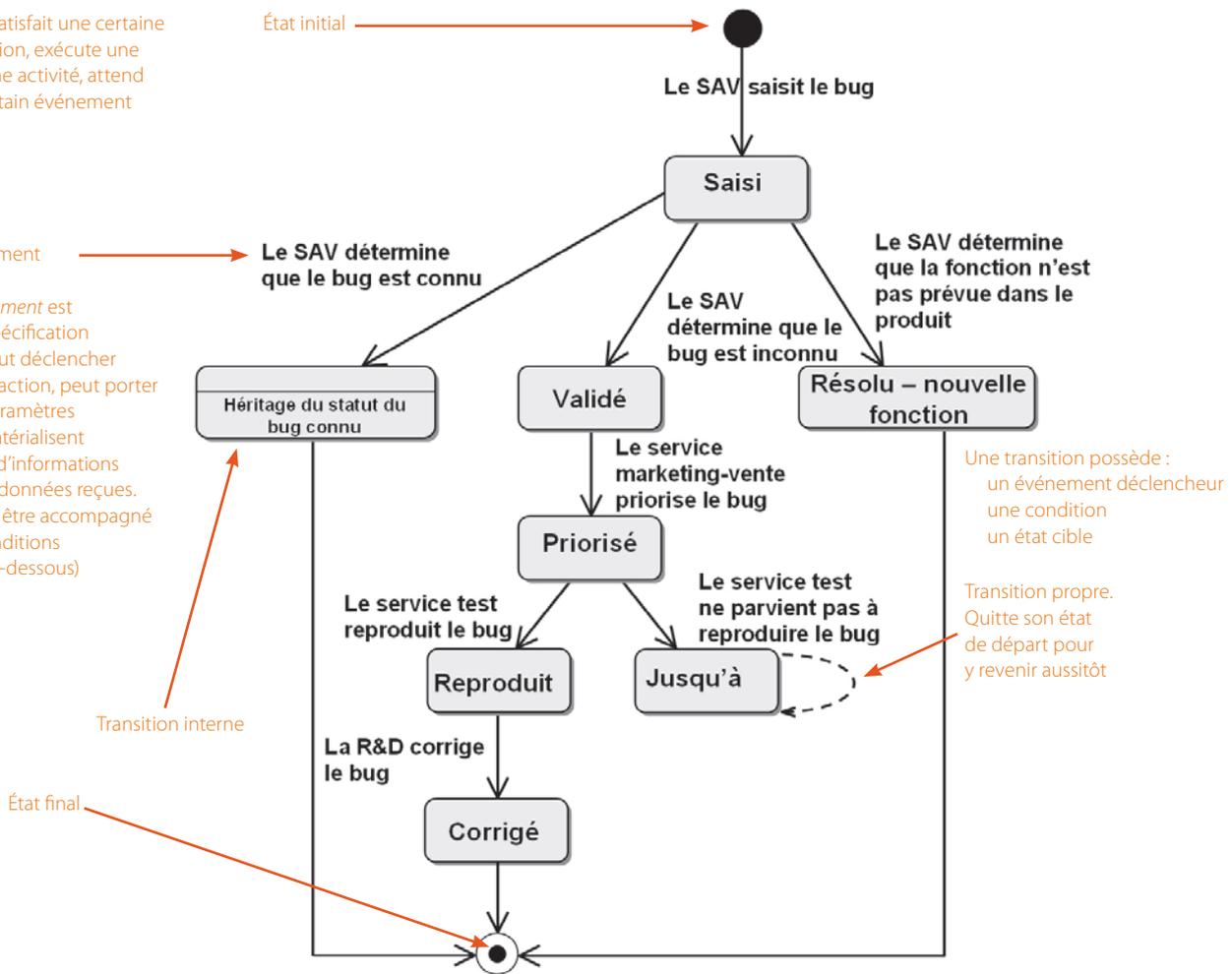
En anglais : *state diagram*  
Notation SysML: *stm*

C'est un diagramme dynamique.  
Il montre les différents états successifs et les transitions possibles des blocs dynamiques.  
Il représente la succession des états d'un système ou d'un sous-système.  
La dynamique des changements est entretenue par des événements.

L'état satisfait une certaine condition, exécute une certaine activité, attend un certain événement

Événement

L'événement est une spécification qui peut déclencher une réaction, peut porter des paramètres qui matérialisent le flot d'informations ou de données reçues. Il peut être accompagné de conditions (voir ci-dessous)



Une transition possède :  
un événement déclencheur  
une condition  
un état cible

Transition propre.  
Quitte son état de départ pour y revenir aussitôt

Condition ou condition de garde

C'est une expression booléenne qui doit être vraie lorsque l'événement arrive pour que la transition soit déclenchée. Elle est notée entre crochets. S'il y a plusieurs transitions avec un même événement, elles doivent avoir des conditions différentes.

Lorsqu'il y a plus de deux conditions, il est recommandé d'utiliser le mot-clé *else* pour garantir l'exhaustivité.

Les changements d'état interne se modélisent en utilisant le mot-clé *when* suivi d'une expression booléenne dont le passage de faux à vrai déclenche la transition. Le passage du temps se modélise en utilisant le mot-clé *after* suivi d'une expression représentant une durée, décomptée à partir de l'entrée dans l'état courant.



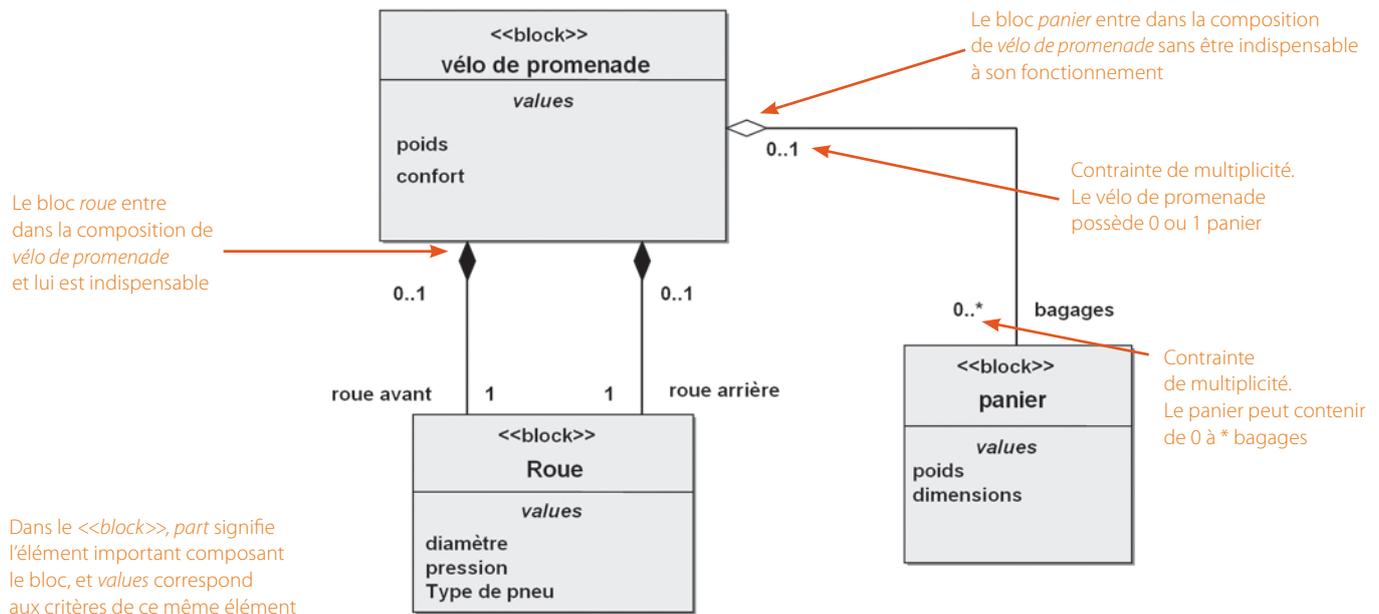
FICHE-OUTIL 5

Diagramme de définition de blocs

En anglais : *block definition diagram*  
 Notation SysML: *bdd*

C'est un diagramme statique. Il montre les briques statiques : blocs, composition, associations...

Il est utilisé pour décrire l'architecture matérielle du système. Un bloc est une entité bien délimitée qui encapsule principalement des attributs (variables d'état), des opérations (procédures comportementales), des contraintes, des ports (échange de flux avec l'extérieur) et des *parts* (sous-blocs internes). Un bloc peut modéliser tout le système, un élément matériel ou logiciel.



FICHE-OUTIL 6

Diagramme de blocs internes

En anglais : *internal block diagram*  
 Notation SysML: *ibd*

C'est un diagramme statique. Il est utilisé pour décrire l'architecture matérielle du système.

Il montre l'organisation interne d'un élément statique complexe. Il représente les instances des *parts* d'un bloc (objets). L'IBD est cadré à l'intérieur des frontières du bloc concerné. Les circulations de flux (MEI) entre les *parts* s'effectuent grâce aux connecteurs qui relient leurs ports. L'IBD d'un bloc est défini à partir du BDD correspondant. Un flux entre ou sort d'une *part* via un port.

