

Objectifs

- Transformer un besoin émergeant en la définition d'un système
- S'appuyer sur des représentations du problème et de ses solutions possibles pour communiquer, appréhender, conceptualiser, concevoir, estimer, simuler, valider et justifier des choix.

5 SysML - UML

1 SysML - UML

a Modélisation avec SysML et UML

UML est l'acronyme de *Unified Modeling Language*, ce qui peut se traduire en français par **Langage de Modélisation Unifié**.

SysML est l'acronyme de *Systems Modeling Language*, soit en français par **Langage de Modélisation de Systèmes**.

SysML et UML sont en effet des **langages graphiques** qui permettent de représenter et de communiquer les divers aspects d'un système. En théorie, n'importe qui peut les lire. Aux graphiques sont bien sûr associés des textes qui en expliquent le contenu.

Il est quasiment impossible de donner une représentation graphique complète d'un système. Mais ces langages offrent un moyen de donner une vue globale d'un tel système, sous forme de **plusieurs vues partielles**, suffisamment précise pour en décrire les constituants et le fonctionnement.

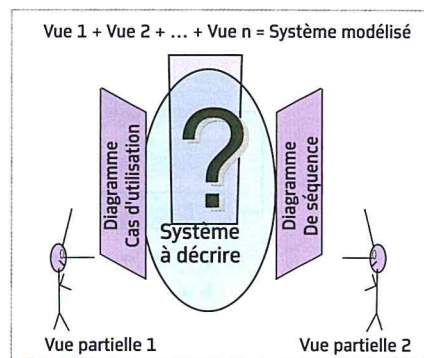


Fig.1 Modélisation avec UML et SysML

EXEMPLE

C'est comme si on souhaitait modéliser une statue avec des photographies, la somme des photos donnerait une idée utilisable en pratique sans risque d'erreur grave, mais ce ne sera pas la statue...

b De UML à SysML

Un **système** est un ensemble de constituants inter-reliés qui interagissent les uns avec les autres d'une manière organisée pour accomplir une finalité commune (NASA 1995).

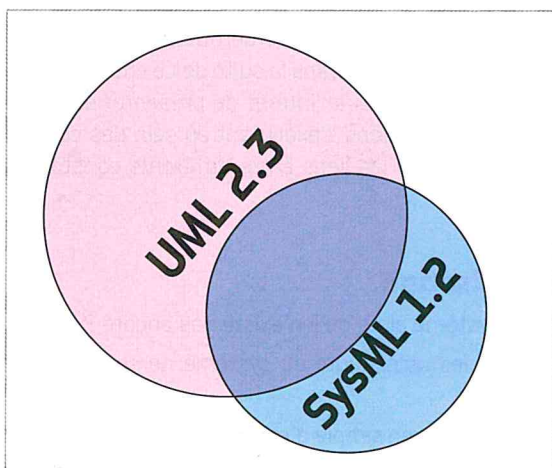


Fig.2 UML et SysML

UML est né du domaine du logiciel, son utilisation en ingénierie système ne s'est pas fait sans une évolution de ce langage. UML est fortement guidée par la programmation objet (POO).

SysML apporte une simplification et une standardisation du vocabulaire, plus question ici de classes, d'objets, ou d'héritage. Ce nouveau langage, ajoute aussi la possibilité de représenter les exigences du système comme elle sont définies dans un cahier des charges, les éléments non-logiciels (mécanique, hydraulique, capteur...), les équations physiques, les flux continus (matière, énergie, etc.) et les allocations.

2 SysML

a Les diagrammes SysML

SysML s'articule autour de neuf diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système. Ces diagrammes peuvent être classés en trois catégories :

- ▶ ceux qui donnent une **vue statique** du système, c'est à dire sans que le temps ne rentre en compte, il s'agit des **diagrammes de structure** ;
- ▶ ceux qui donnent une **vue dynamique** du système, on les appelle aussi les **diagrammes de comportement** ;
- ▶ et celui qui est l'**expression du besoin**, le seul diagramme transverse, le diagramme d'exigences.

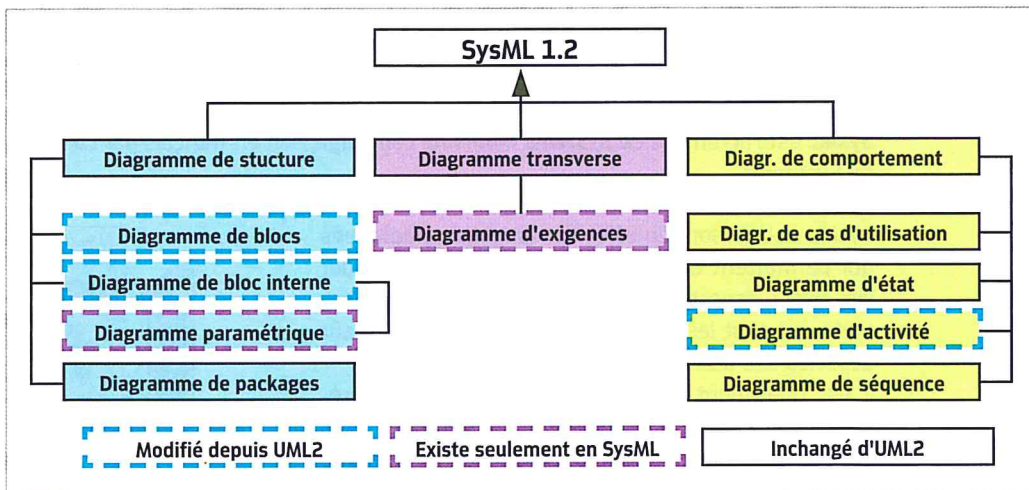


Fig. 3 SysML, 3 catégories, 9 diagrammes

Tous les diagrammes SysML ne sont pas toujours tous utiles pour modéliser un système donné.

Nous nous intéresserons pour cette première approche, seulement à quatre diagrammes :

- ▶ le **diagramme de cas d'utilisation**, ce diagramme assure la relation entre l'utilisateur et le système mis en œuvre ;
- ▶ le **diagramme d'états-transitions** qui représente la façon dont évolue notre système ou une sous entité de celui-ci ;
- ▶ le **diagramme de séquence** qui modélise la succession chronologique des opérations réalisées par un acteur ;
- ▶ et le **diagramme de blocs** relatif à la modélisation structurelle. C'est un assemblage de blocs. Le concept universel de « bloc » proposé par SysML permet de modéliser tout élément structurel.

Remarque : Pour la modélisation structurelle, SysML est déroutant. Il utilise les diagrammes de classes d'UML pour faire une description matérielle. Dans la suite de ce chapitre vous aurez un aperçu du **diagramme de déploiement** au sens d'UML qui a l'intérêt de présenter simplement les liens numériques entre les constituants et les applications s'exécutant au sein des constituants et de son équivalent en SysML qui permet de présenter les liens entre différents constituants physiques : le **diagramme de blocs**.

b Diagramme de cas d'utilisation

Comment faire pour représenter un système alors qu'il n'existe pas encore ?

Bien souvent la maîtrise d'ouvrage et les utilisateurs du système ne sont pas des spécialistes de SysML, encore moins des techniciens.

Le diagramme de cas d'utilisation est un moyen simple d'exprimer leurs besoins. Il permet de les analyser et de les organiser. Il permet également de recenser les grandes fonctionnalités d'un système. Il s'agit d'une des premières étapes dans une modélisation SysML.

Un diagramme de cas d'utilisation capture le comportement d'un système tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Ils permettent d'exprimer le besoin des utilisateurs d'un système.

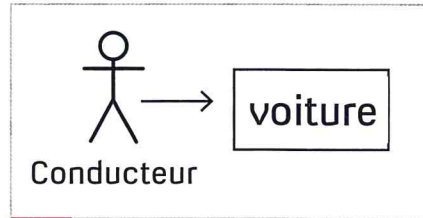


Fig. 4 Le conducteur utilise sa voiture

EXEMPLE

Quels sont les utilisateurs d'une automobile ?

Commençons par les acteurs faciles à identifier : le **conducteur**, il conduit le système ; les **passagers** utilisent le système, mais le **passager avant** à des droits que ceux de l'arrière n'ont pas (réglage des stations de radio, de la température, ...).

Et les autres, le garagiste, le piéton, le pompiste, ... sont ils des « acteurs » de notre système ?



ANALYSES DES SYSTÈMES ET REPRÉSENTATIONS

ACTIVITÉ 1

Si le véhicule devient automatique et électrique comme le Cycab quels sont les changements ?

Pour mémoire le Cycab est un véhicule automatique électrique, il n'y a pas de pilote mais une personne qui choisi une destination.

Les acteurs

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Il se représente par un petit bonhomme avec son nom (son rôle) inscrit dessous.

Il est également possible de représenter un acteur sous la forme d'un rectangle, c'est en général la forme employée pour les acteurs « non-humain ».

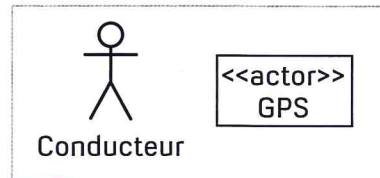


Fig. 5 Acteur humain, acteur matériel

Relations entre acteurs

Il existe une relation possible entre deux acteurs, c'est la **généralisation** : un acteur A est une généralisation d'un acteur B si l'acteur A peut être remplacé par l'acteur B.

Dans ce cas, tous les cas d'utilisation accessibles à A le sont aussi à B, mais l'inverse n'est pas vrai.

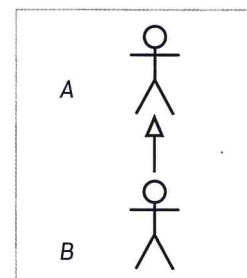


Fig. 6

EXEMPLE

La figure montre que l'opérateur de maintenance est un opérateur avec un pouvoir supplémentaire : en plus d'être apte à prélever un échantillon et suivre une trajectoire, il peut piloter le robot en mode manuel. Par contre, l'opérateur ne peut pas piloter le robot en mode manuel.

On peut donc dire que l'« Opérateur de maintenance » est une sorte de « d'Opérateur ».

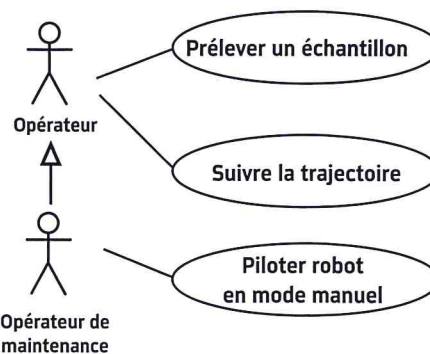


Diagramme de cas d'utilisation du ROBOVOLC

ACTIVITÉ 2 Rechercher et organiser les acteurs

Une voiture est conduite par un conducteur, un passager voyage à ses côtés. Ce voyageur est parfois même un copilote. Trois passagers se trouvent aux places arrières. Le passager avant a accès au GPS, à la radio et aux commandes de chauffage climatisation. Le garagiste entretient ce véhicule.

Les cas d'utilisation

Un cas d'utilisation est **une fonctionnalité visible de l'extérieur**. Il réalise **un service de bout en bout**, rendu par le système, sans imposer le mode de réalisation de ce service.

Un cas d'utilisation se représente par une ellipse contenant l'intitulé du cas, sous forme d'un verbe à l'infinitif et d'un éventuel complément d'objet direct.

La frontière du système est représentée par un cadre. Le nom du système figure à l'intérieur du cadre, en haut. Les acteurs sont à l'extérieur et les cas d'utilisation à l'intérieur.

Une relation **d'association** est un chemin de communication entre un acteur et un cas d'utilisation et est représentée par un trait continu.

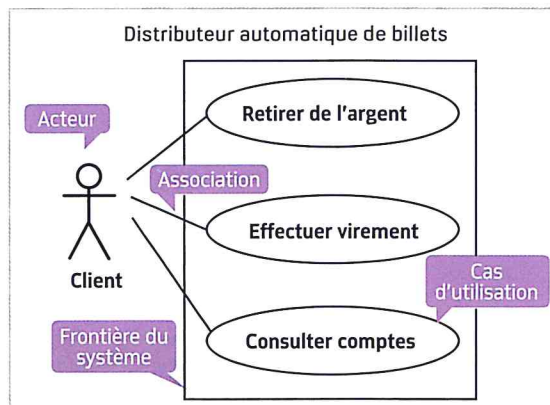


Fig. 7 Diagramme de cas d'utilisation

ACTIVITÉ 3 Le téléphone portable

Le cahier des charges est très simple, il s'agit de modéliser un téléphone portable. Ce téléphone ne fait que téléphoner et envoyer des SMS.

1. Rechercher et organiser le/les acteur(s)
2. Rechercher et organiser les cas d'utilisation

Les inclusions permettent également de décomposer un cas complexe en sous-cas plus simples.

Attention !

- ne pas en abuser
- les cas d'utilisation ne s'enchaînent pas, puisqu'il n'y a aucune représentation temporelle dans un diagramme de cas d'utilisation.

Relations

Il existe principalement trois types de relations

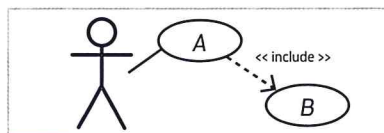


Fig. 8 Relation d'inclusion

Un cas **A inclut** le cas **B**. Lorsque A est sollicité, B l'est obligatoirement, comme une partie de A. Cette dépendance est symbolisée par le stéréotype << include >>.

EXEMPLE

L'enregistrement du point de passage est commun à suivre une trajectoire et piloter le robot en mode manuel. C'est une factorisation (voir fig. 11).

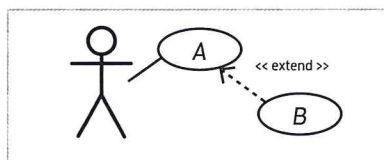


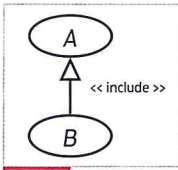
Fig. 9 Relation d'extension

On dit qu'un cas d'utilisation **B étend** un cas d'utilisation **A** lorsque le cas d'utilisation **B** peut être appelé au cours de l'exécution du cas d'utilisation **A**. L'extension est optionnelle. Cette dépendance est symbolisée par << extend >>. L'extension peut intervenir à un point précis du cas étendu.

Ce point s'appelle le **point d'extension**.

EXEMPLE

Le robot s'arrête s'il rencontre un obstacle à moins de 50 centimètres (voir Fig. 11).



Un cas A est une généralisation d'un cas B si B est un cas particulier de A.

Fig. 10 Relation de généralisation

EXEMPLE

Le prélèvement d'échantillon peut être de type gazeux ou minéral (voir Fig. 11).

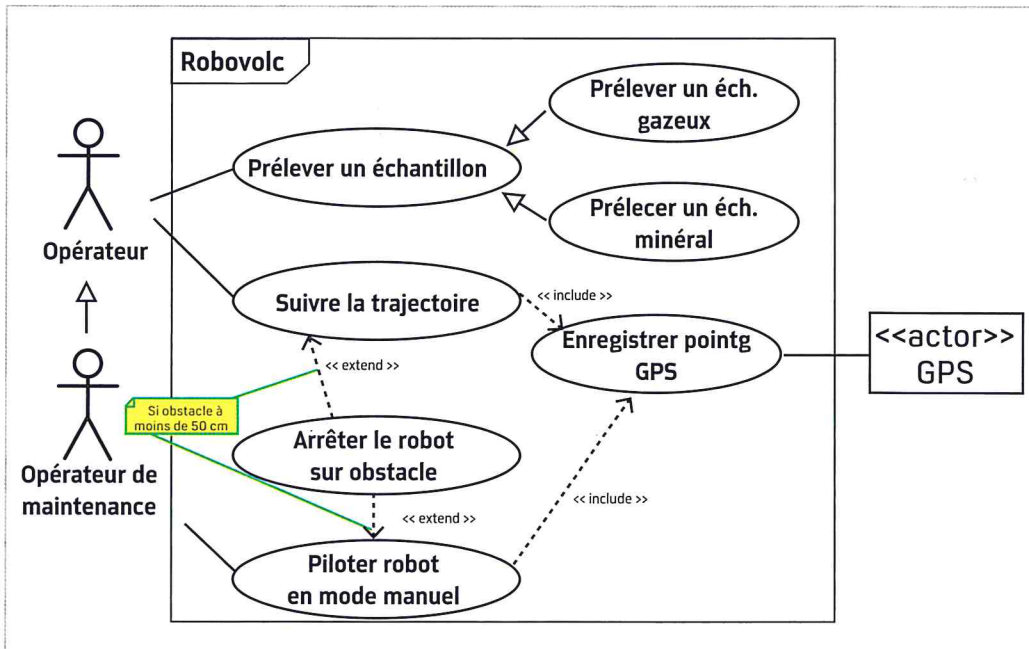


Fig. 11 Diagramme de cas d'utilisation du ROBOVOLC

C Diagramme de séquences

► Principe et représentation

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique.

Une ligne de vie se représente par un rectangle, auquel est accrochée une ligne verticale pointillée.

Un message définit une communication particulière entre des lignes de vie.

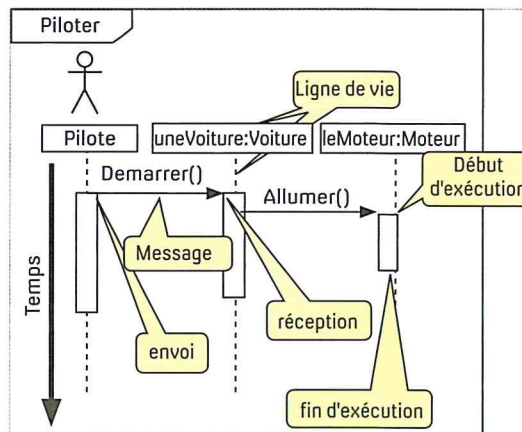


Fig. 12 Démarrage d'une voiture

EXEMPLE

L'exemple de la Fig. 12 montre l'acteur *Pilote* qui envoie le message *démarrer()* à *uneVoiture* qui est une instance de block *Voiture*. Cette voiture, envoie le message *allumer()* à l'objet (l'instance) *leMoteur* qui est de classe (block) *Moteur*. *uneVoiture* aurait pu être une *peugeot 207* et *leMoteur* aurait pu être *1.4 HDI n° 103495*. Ce sont des instances des deux blocks voiture et moteur, un peu comme « un gâteau » et un « moule à gâteau ».

SysML prévoit également la possibilité de mettre des contraintes temporelle.

► Fragments combinés

Les fragments combinés SysML proposent une notation facilitatrice. Chaque fragment de diagramme de séquence possède un « opérateur » et peut être divisé en sous-diagrammes.

Les principaux opérateurs sont :

- **loop**, boucle. Le fragment peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération ;
- **opt**, optionnel. Le fragment ne s'exécute que si la condition fournie est vraie ;
- **alt**, fragment alternatif. Le fragment est exécuter seulement SI la condition est vraie ; une alternative SINON peut être précisée pour le cas où la condition est fausse.

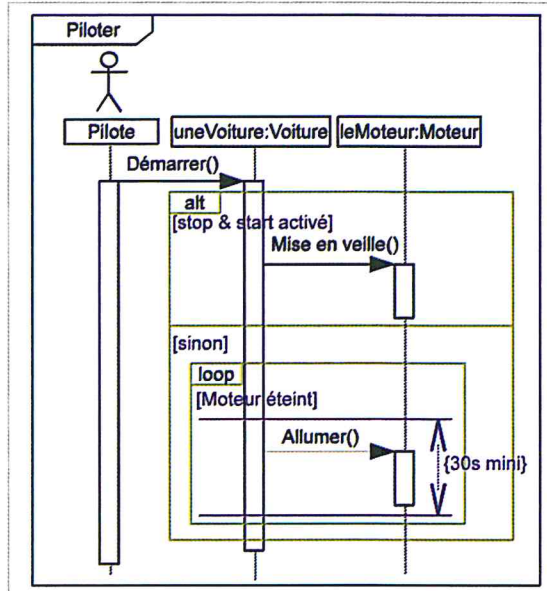


Fig. 13 Démarrage d'une voiture équipée du "stop & start"

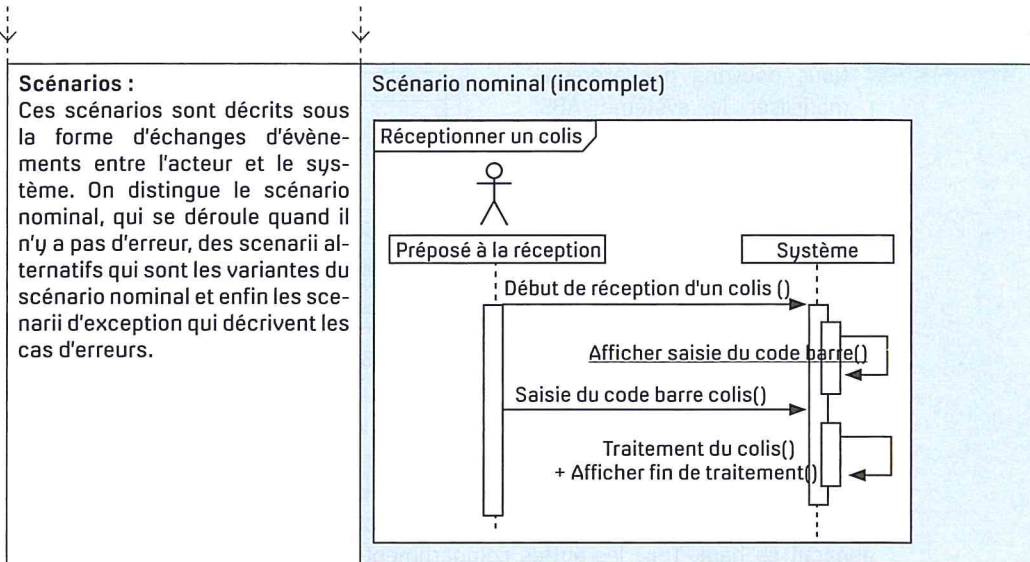
► Présentation des cas d'utilisation

Lorsque les cas d'utilisation sont trouvés, il faut les définir plus précisément. Chaque cas d'utilisation donne lieu à une fiche textuelle (réalisée à l'aide d'un plan-type).

Nom : Utiliser une tournure à l'infinif.	Réceptionner un colis.
Objectif : Une description résumée permettant de comprendre l'intention principale du cas d'utilisation. Cette partie est souvent renseignée au début du projet dans la phase de découverte des cas d'utilisation.	Le cas d'utilisation colis permet à l'acteur « préposé à la réception de colis » de réceptionner un colis et de l'enregistrer dans le système via son code barre. Si son code barre est illisible, il le saisie, si le colis est détérioré, il l'enregistre et le refuse auprès du transporteur.
Acteurs principaux : Ceux qui vont réaliser le cas d'utilisation (la relation avec le cas d'utilisation est illustrée par le trait liant le cas d'utilisation et l'acteur dans un diagramme de cas d'utilisation).	« préposé à la réception de colis »
Acteurs secondaires : Ceux qui ne font que recevoir des informations à l'issue de la réalisation du cas d'utilisation.	
Dates : Les dates de créations et de mise à jour de la description courante.	05/10/2010
Responsable : Le nom du ou des responsables.	Marie Tournelle
Version : Le numéro de version.	V0.1
Les préconditions : Elles décrivent dans quel état doit être le système (l'application) avant que ce cas d'utilisation puisse être déclenché. Les post conditions : Elles décrivent l'état du système à l'issue des différents scenarios.	Précondition : Système initialisé ; préposé à la réception authentifié sur le système PostCondition : néant

SysML Il est possible d'envoyer un message à un système. Mais il n'est pas souhaitable d'envoyer un message à un acteur. Sauf dans des cas où l'acteur est un périphérique externe (ex : imprimante)

La rédaction des scénarios est relativement simple et peu coûteuse. Il est possible de la faire à l'aide de logiciels et de séquences.



ACTIVITÉ 4 Proposer un scénario présentant le cas du traitement d'un colis abimé.

Pour mémoire : si le colis est détérioré, il l'enregistre et le refuse auprès du transporteur, ce scénario sera un **scénario d'exception**.

e Diagramme de déploiement - Diagramme de blocs

Diagramme de déploiement (UML)

Comment représenter une architecture de système, un système contient du matériel mais aussi des modules logiciels déployés !

Un **diagramme de déploiement** décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des **composants** logiciels sur ces matériels. Chaque ressource étant matérialisée par un **nœud**, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les **connexions** entre les composants ou les nœuds.

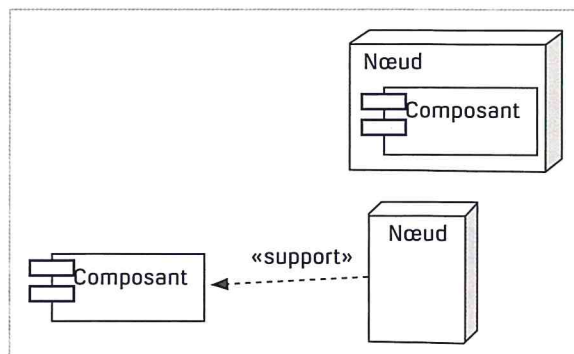


Fig. 14 Représentation des nœuds et des composants

Il existe deux possibilités pour représenter l'affectation d'un composant à un nœud.

Représentation des nœuds et des composants

Chaque ressource est matérialisée par un nœud représenté par un cube comportant un nom. Un nœud peut posséder des attributs (quantité de mémoire, vitesse du processeur, ...).

Pour montrer qu'un composant est affecté à un nœud, il faut soit placer le composant dans le nœud, soit les relier par une relation de dépendance stéréotypée «support» orientée du composant vers le nœud.

Dans un diagramme de déploiement, les associations entre nœuds sont des chemins de communication qui permettent l'échange d'informations. Il est possible d'ajouter des informations comme le type de support et/ou le nombre liens.

Diagramme de définition de bloc (SysML)

Le diagramme de définition de bloc (BDD, ou *Block Definition Diagram*) représente la vue boîte noire d'un bloc. Ainsi le **bloc principal** et la hiérarchie des blocs qui le composent, qu'ils soient logiciels ou matériels, peuvent être spécifiés dans ce diagramme.

C'est ce qui est vrai sur la P00 et les diagrammes de blocs est vrai aussi sur les diagrammes de blocs.

EXEMPLE

Nous pouvons par exemple modéliser le système ABS ci-dessous qui contient un bloc de contrôle « Système anti-blocage des roues » lui-même composé d'un bloc de détection d'adhérence et d'un bloc de régulation de freinage

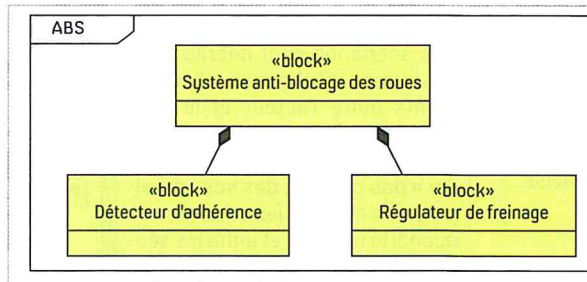


Fig. 15 ABS vu à travers SysML

Bloc

Le bloc SysML (*block*) constitue la brique de base pour la modélisation de la structure d'un système. Il peut représenter un système complet, un sous-ensemble ou un composant élémentaire.

Un bloc est représenté graphiquement par un rectangle découpé en compartiments. Le nom du bloc apparaît en haut. Tous les autres compartiments ont des labels indiquant ce qu'ils contiennent : valeurs, parties, etc.

Le mot-clé « block » apparaît par défaut, c'est un stéréotype. Il est possible d'en définir de nouveaux : « system », « sub_system », etc.

EXEMPLE

Le bloc représente notre système complet. Il sera nécessaire par la suite de le raffiner. On peut noter qu'il contient déjà des sous-parties ou sous-systèmes, roues et moteur.

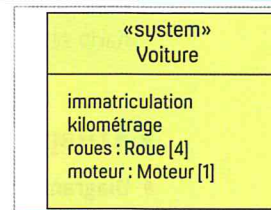


Fig. 16 Système "Voiture"

Les ports de flux

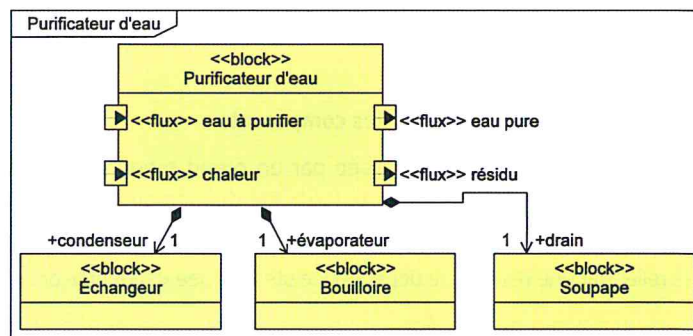
Le bloc permet de décrire les flots qui circulent à travers un système. Les ports de flux (*flow ports*) sont une nouveauté SysML ; ils représentent ce qui peut circuler en entrée et/ou en sortie d'un bloc, que ce soit des données, de la matière ou de l'énergie. On peut les définir pour tous les blocs, et même faire des assemblages.

EXEMPLE

Purificateur d'eau

Le bloc principal définit le purificateur d'eau, il est composé de 3 blocs :

- un échangeur de chaleur qui a un rôle de condenseur,
- une bouilloire qui a un rôle d'évaporateur,
- une soupape qui a un rôle de drain.



Les trois blocs sont physiquement liés au bloc principal (le purificateur d'eau), car les liens utilisés sur le diagramme sont des agrégations fortes ou compositions, représentées par un losange plein. Si un bloc n'en faisait pas physiquement partie, on parlerait alors d'une référence, et l'association utilisée serait une agrégation simple, représentée par un losange vide.

En termes de flux, le bloc purificateur d'eau utilise en entrée de l'eau froide et de la chaleur externe, et produit en sortie de l'eau purifiée, du résidu.

Propriétés

Les propriétés sont les caractéristiques structurelles de base des blocs. Elle peuvent être de quatre types : "parts" les parties d'autres blocs comme une composition ; "references" surtout utilisées dans les diagrammes de blocs internes ; "values" lorsque l'on exprime une caractéristique ; et "constraints" lorsque on souhaite ajouter une contrainte à un bloc ({X > Y}).

Les propriétés à retenir sont principalement :

- Les valeurs, qui sont utilisées pour modéliser les caractéristiques quantitatives des blocs. **Il est très important de bien définir les types de ces valeurs de manière réutilisable.**
- Les parties, qui décrivent la hiérarchie de décomposition du bloc en termes d'autres blocs.

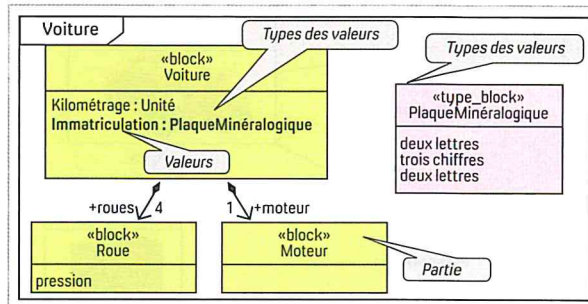


Fig. 17 BDD Voiture

Attention, les types peuvent être intégrés au bloc principal.

ANALYSES DES SYSTÈMES ET REPRÉSENTATIONS

EXEMPLE

Si l'on considère le bloc Voiture, on peut définir des valeurs : numéro d'immatriculation, kilométrage, vitesse courante, etc.

Le « kilométrage » sera typé par un type « Unité », lui-même défini par un type de base réel, et une information « km » ou « miles » dans les pays anglo-saxons.

Pour ce même bloc, on peut définir des parties telles que : un moteur ; quatre roues ; quatre portes ; un coffre ; etc.

Diagrammes de bloc interne

Le diagramme de bloc interne (*internal block diagram* ou ibd) décrit la structure interne du système. Ceci permet de décrire plus finement les différentes parties, ports et connecteurs. Chaque bloc peut se raffiner en un autre diagramme.

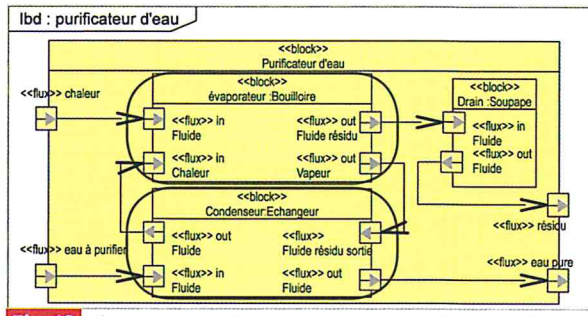


Fig. 18 Diagramme de blocs interne du purificateur d'eau

Les associations

Comme en POO il est possible de faire des associations. L'association est une relation n'impliquant pas de contenance, comme la composition ou l'agrégation, mais une relation d'égal à égal.

Attention, on peut vite mélanger des pommes, des poireaux et même des vaches avec ce type de diagrammes.

La voiture est la propriété d'une personne, qui peut en posséder plusieurs et joue le rôle de propriétaire. La Voiture roule sur une route... Ces deux relations sont représentées par des associations en SysML (lignes simples).

Ce diagramme est là pour montrer le contexte du système. Il est bien évident que Personne et Route ne font pas partie de notre système.

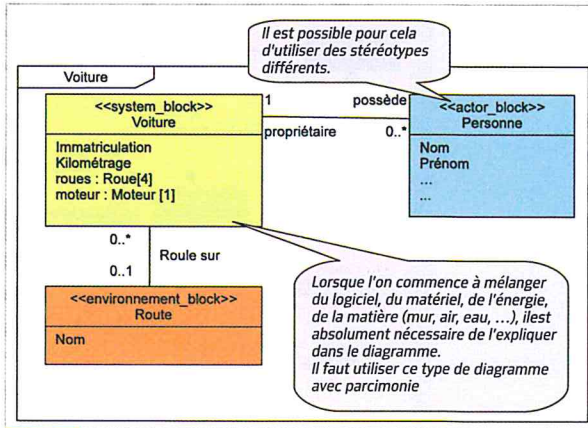


Fig. 19 Association BDD voiture - Personne

Notion de contexte

Avec un diagramme de bloc il est également possible de définir un contexte à notre système. Le contexte du système est utile pour fixer les idées au démarrage de notre analyse.

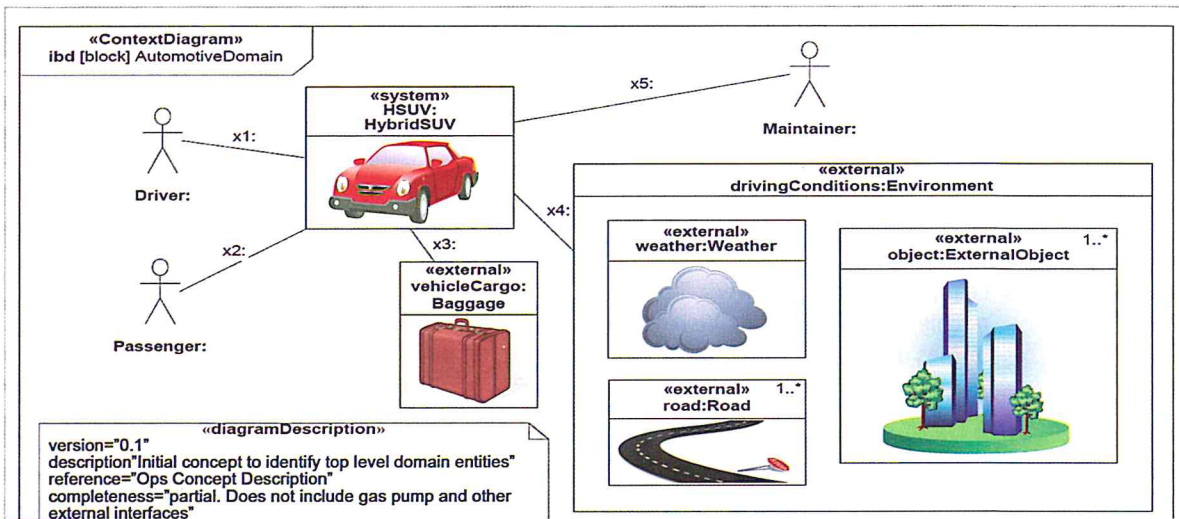


Fig. 20 Diagramme de blocs utilisé en tant que diagramme de contexte du système

ACTIVITÉ 5 Diagrammes de déploiement et diagramme de blocs d'un robot mobile

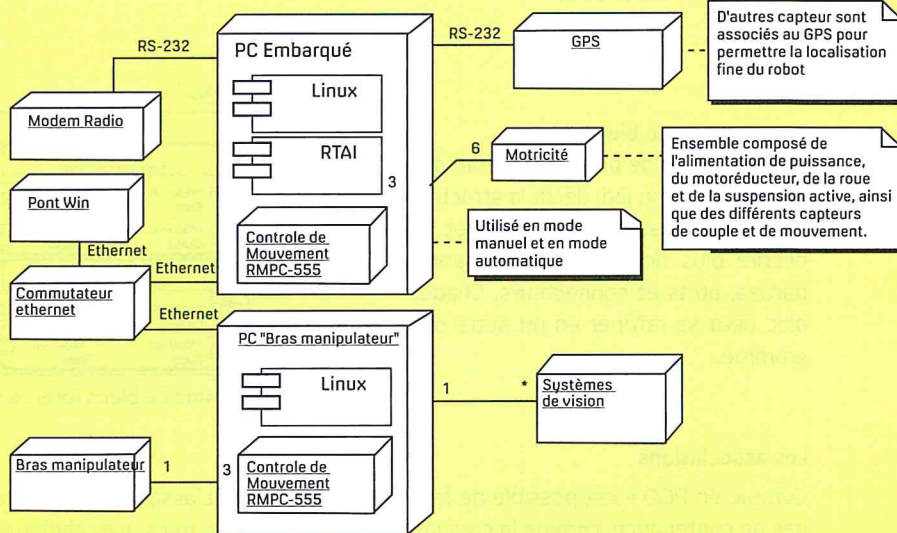
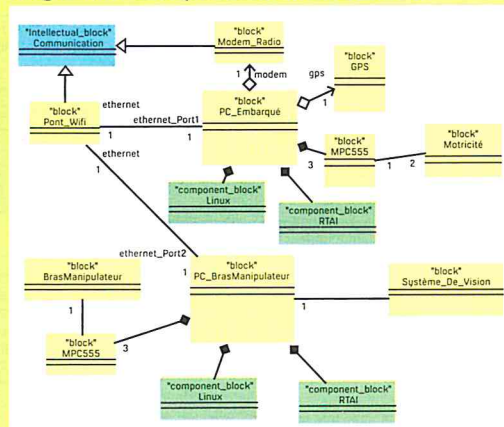


Diagramme de déploiement du ROBOVOLC



En vous appuyant sur les diagrammes de déploiement et de blocs, précisez :

Combien ce robot mobile possède-t-il de roues ?

Combien ce robot possède-t-il de bras manipulateurs ?

Quels sont ses moyens de communication ?

Combien ce robot mobile possède-t-il de moteurs de roues ?

Combien ce robot mobile possède-t-il de cartes contrôleurs ?

f Diagramme d'états-transitions

Comment symboliser les différents états d'un système ? Comment le système change-t-il d'états ?

Un **diagramme d'états-transitions** rassemble et organise les **états** et les **transitions** d'un système. Bien entendu, le modèle dynamique du système comprend plusieurs diagrammes d'états-transitions. Dans un même système il est possible que plusieurs automates s'exécutent en simultané.

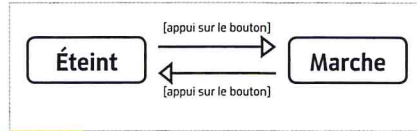
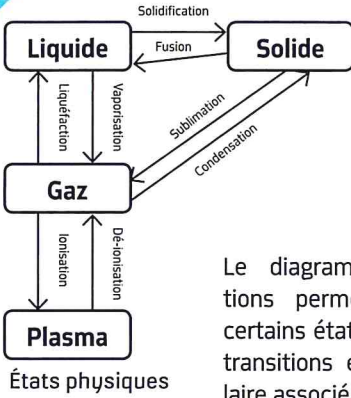


Fig. 21 Marche/Arrêt

ANALYSES DES SYSTÈMES ET REPRÉSENTATIONS

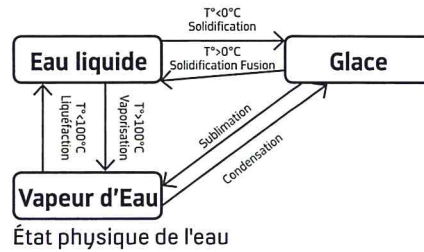
EXEMPLE



En physique, un état de la matière correspond à un certain degré de cohérence de la matière (densité, structure cristalline, indice de réfraction...) qui se traduit par des comportements définis par les lois de la physique (malléabilité, ductilité, viscosité, loi des gaz parfaits, etc.).

Le diagramme d'états-transitions permet d'en présenter certains états et d'en définir les transitions et donc le vocabulaire associé.

Si on avait limité notre réflexion à l'eau le diagramme aurait pu être celui-ci.



In SysML il est nécessaire de définir l'état initial et l'état final... dans le cas présent, qui est le premier de œuf ou de la poule ?

► Cas particulier : état initial et état final

L'état initial est un pseudo état qui indique l'état de départ, par défaut, lorsque le diagramme d'états-transitions, ou l'état enveloppant, est invoqué. Lorsqu'un objet est créé, il entre dans l'état initial.

L'état final est un pseudo état qui indique que le diagramme d'états-transitions est terminé.



Représentation graphique de l'état initial.



Représentation graphique de l'état final.

► Transition

Une transition définit la réponse d'un objet à l'occurrence d'un événement. Elle lie, généralement, deux états $E1$ et $E2$ et indique qu'un objet dans un état $E1$ peut entrer dans l'état $E2$ et exécuter certaines activités, si un événement déclencheur se produit.

Une transition externe est une transition qui modifie l'état actif. Il s'agit du type de transition le plus répandu. Elle est représentée par une flèche allant de l'état source vers l'état cible.



Fig. 22 La figure illustre la représentation graphique d'une transition externe entre deux états

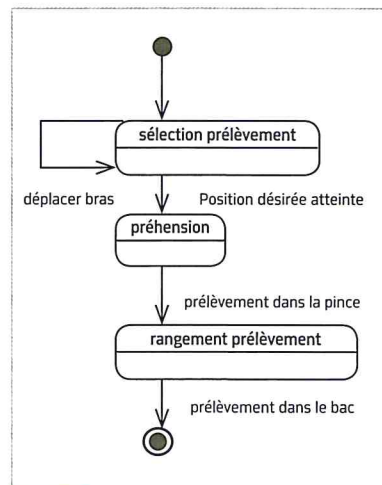
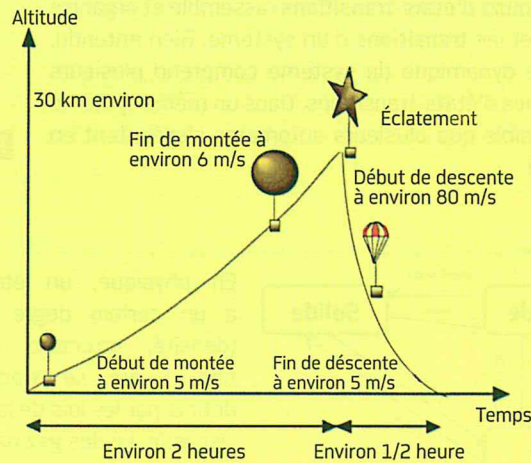


Fig. 23 Diagramme état-transition d'un prélèvement minéral

ACTIVITÉ 6 Le ballon-sonde météorologique

Un ballon est gonflé avec de l'hélium de façon à avoir une vitesse ascensionnelle de 5 à 6 m/s.



Une sonde effectue des mesures toutes les dix secondes, ce qui se traduit par un échantillonnage du profil de l'atmosphère tous les 50 mètres, du sol jusqu'à l'altitude d'éclatement du ballon.

Cette dernière est comprise entre 20 000 et 30 000 mètres.

Puis il s'en suit une descente en parachute, jusqu'à l'atterrissage.

Proposer un diagramme états-transitions permettant de présenter les différentes étapes de fonctionnement du ballon sonde.

SYNTHÈSE

On appelle **modèle** l'ensemble des symboles qui participent à la définition d'un système de modélisation. Ainsi, dans le cas de SysML, les symboles qui forment « l'alphabet » de ce langage se nomment le modèle SysML. Ce **modèle SysML** se partage à son tour en deux (sous-)modèles nommés, respectivement, **modèle structurel** et **modèle comportemental**.

Le **modèle structurel** permet ainsi de se représenter l'état statique du système modélisé.

Le **modèle comportemental** introduit une dimension de modélisation supplémentaire : le temps, permettant ainsi de décrire la dynamique du système modélisé.

En STI2D nous utilisons les **diagrammes de blocs** pour définir la structure du système, vous pouvez vous aider également des diagrammes de déploiements. Les **diagrammes de cas d'utilisation** et d'**états-transitions** pour modéliser le comportement d'un système et nous nous appuyons sur les **diagrammes de séquences** pour modéliser les interactions.

Lorsque je réalise une modélisation :

1. Je recense les exigences des cahiers des charges, éventuellement à travers un diagramme d'exigences (non présenté ici).
2. Je recense et ordonne les acteurs de mon système, humain et non-humain. Puis je réalise un diagramme de cas d'utilisation dit « système ». Dans ce diagramme le système est représenté par une bulle (un seul cas). Il est possible de remplacer ce diagramme de cas par un diagramme de blocs dit de « contexte ».
3. Je recherche toutes les fonctionnalités offertes par mon système aux utilisateurs. Je les recense et les ordonne dans un diagramme des cas d'utilisation.
4. Pour chaque cas d'utilisation je réalise à l'aide du plan type leur description. Pour chaque cas je recherche le scénario nominal et éventuellement les scénarios d'exceptions (les scénarios quand il y a des problèmes). Je les formalise à l'aide de diagrammes de séquences.
5. Pour les points clefs je réalise éventuellement un ou plusieurs diagrammes d'états-transitions.
6. Ensuite je peux concevoir l'architecture et/ou décrire mon système grâce à un diagramme de blocs et, si la complexité le justifie, avec des diagrammes de blocs internes. Je peux dans le cas de systèmes simples utiliser aussi un diagramme de déploiement.