

SYSTÉMIQUE, INGÉNIERIE SYSTÈME ET MBSE

Date : 19 décembre 2022

Auteur(s) : Benoît Eynard, Matthieu Bricogne

Copyright : B. Eynard, M. Bricogne, université de technologie de Compiègne

Licence : CC 4.0 BY-NC-SA [<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.fr>] + licence commerciale ET-LIOS [<https://et-lios.s-mart.fr/licencecommerciale/>]

Table des matières

1. Introduction : concevoir un produit aujourd'hui	4
1.1. Introduction	4
1.2. Les grandes tendances liées à la conception de produits	4
1.2.1. Un éclairage macroscopique sur l'évolution des produits	4
1.2.2. L'intégration physique	5
1.2.3. L'intégration fonctionnelle	6
1.2.4. La dématérialisation	6
1.2.5. L'intégration de la connectivité	7
1.3. Les produits multidisciplinaires	8
1.3.1. La mécatronique	8
1.3.2. Les produits intelligents	8
1.3.3. Les systèmes cyber-physiques	9
1.3.4. En synthèse : parler de systèmes multidisciplinaires	9
1.3.5. Besoin d'expertises, pas de silos !	10
1.4. Et ET-LIOS dans tout ça ?	11
1.4.1. <ET-LIOS> Module E	11
2. Systémique et théorie des systèmes	15
2.1. Introduction	15
2.2. Définition d'un système	15
2.3. L'approche systémique	16
2.4. La notion d'interaction	17
2.5. La notion de point de vues	19
2.6. Les concepts liés à la systémique	20
3. L'Ingénierie Système - IS	23
3.1. Introduction	23
3.2. L'ingénierie système en bref...	23
3.3. Historique de l'Ingénierie Système	24
3.4. Standards et normes de l'Ingénierie Système	25
3.5. Les principaux concepts au cœur de l'IS	27
3.6. APMO - un cadre d'analyse pour les techniques et méthodes	30
3.7. Les différents processus de l'Ingénierie Système et le focus de ce cours	32
4. Model-Based Systems Engineering (MBSE)	34
4.1. Introduction	34
4.2. Pourquoi le MBSE ?	34
4.3. Model-Based Systems Engineering (MBSE)	35
4.4. Quelques exemples de langage modélisation	35
4.5. Systems Modeling Language	39
4.6. La notation SysML	43
4.6.1. Introduction	43
4.6.2. Les blocs	43
4.6.3. Diagramme de définition de blocs	44
4.6.3.1. BDD	45
4.6.3.2. Exercice	46

4.6.4. Diagramme de blocs internes	47
4.6.4.1. IBD	47
4.6.4.2. Exercice	49
4.6.4.3. Exercice	50
4.6.5. Diagramme de packages	50
4.6.5.1. PKG	50
4.6.5.2. Exercice	52
4.6.5.3. Exercice	52
4.6.5.4. Exercice	53
4.6.6. Diagramme d'exigences	53
4.6.6.1. REQ	53
4.6.6.2. Exercice	58
4.6.7. Diagramme de cas d'utilisation	58
4.6.7.1. UC	58
4.6.7.2. Exercice	59
4.6.7.3. Exercice	60
4.6.8. Diagramme de séquence	60
4.6.8.1. SD	60
4.6.8.2. Exercice	62
4.6.9. Diagramme d'activité	62
4.6.9.1. ACT	62
4.6.9.2. Exercice	63
4.6.9.3. Diagramme de séquence vs. d'activité	64
4.7. Synthèse	64
Synthèse	65
Glossaire	66
Bibliographie	70

1. Introduction : concevoir un produit aujourd'hui

1.1. Introduction

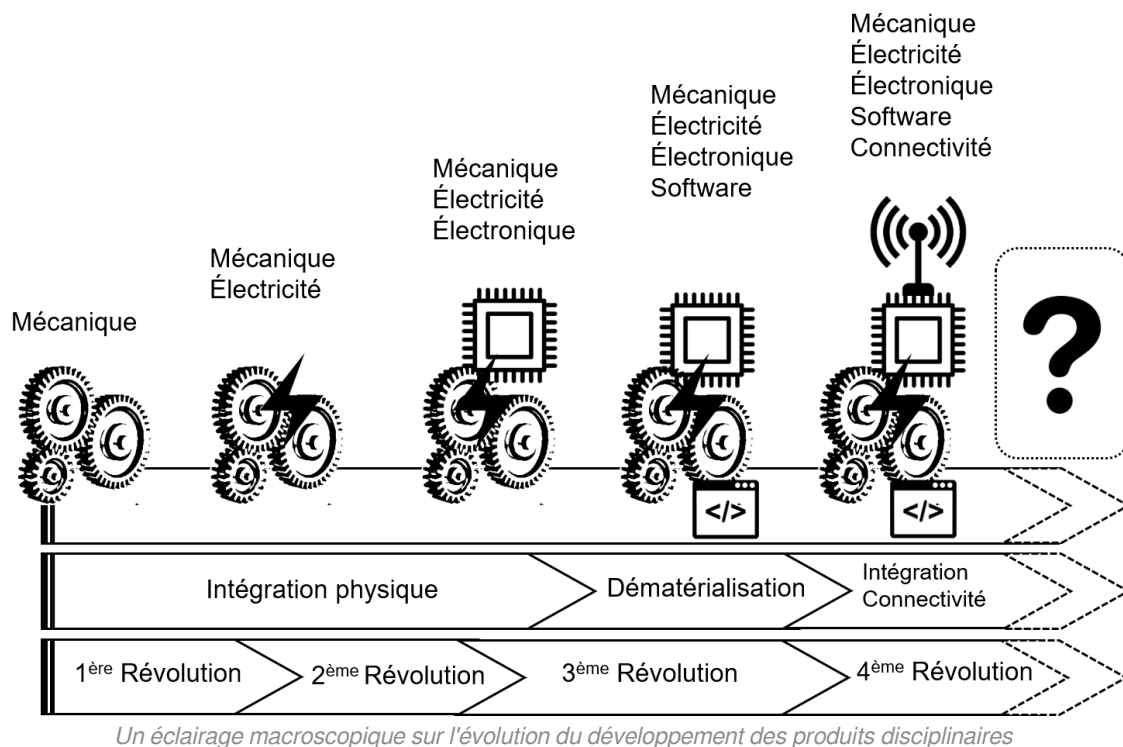
Objectifs pédagogiques : Comprendre les grands enjeux de la conception de système

Comprendre les grandes tendances liées à l'évolution des produits

Appréhender les difficultés engendrées par la conception de systèmes multidisciplinaires

1.2. Les grandes tendances liées à la conception de produits

1.2.1. Un éclairage macroscopique sur l'évolution des produits



Cette figure présente de manière synthétique un éclairage de l'évolution technologique des produits en parallèle de l'apparition des technologies en elles-mêmes ayant mené aux révolutions industrielles et aux ères subséquentes (Thèse Julia Guérineau^[p.71]). Ainsi, nous distinguons majoritairement trois « ères ». La première est l'**ère matérielle** et traduit l'idée de produits se basant sur la mécanique, l'électricité ou encore l'électronique.

L'apparition du logiciel et de l'informatique a offert de nouvelles possibilités pour le développement de produits innovants et la réalisation des fonctions menant à l'**ère du logiciel** (dématérialisation).

Enfin, le développement des infrastructures réseaux et d'Internet, sa propagation aux objets de la vie courante et aux moyens industriels, le développement de nouvelles technologies de connectivité ont démocratisé cette dernière.

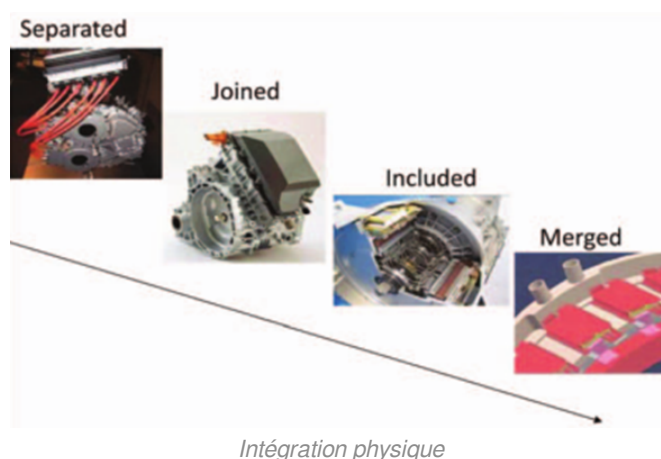
La conjonction de ces technologies avec le numérique, qui s'inscrit dans la continuité de l'ère précédente, nous permet de statuer que nous sommes actuellement au cœur de l'**ère du numérique et de la connectivité**.

Pour aller plus loin

Jusqu'à peu, le développement de systèmes se résumait à rendre le produit le plus attractif possible, souvent en y intégrant le maximum de technologies, en se fixant un cadre en termes de coût, qualité et délais de mise sur le marché.

Aujourd'hui, des travaux de recherche tentent de combler le fossé entre les méthodes de conception des produits « traditionnelles » et l'acceptation par les consommateurs de produits fonctionnels, mais plus simples et plus durables, afin de rendre ces produits plus compatibles avec les objectifs écologiques liés aux limites planétaires. Ces travaux^[p.70], qui s'apparentent au *Design for Sustainability* [https://en.wikipedia.org/wiki/Design_for_the_Environment], cherchent à prendre en considération les impacts liés à la façon de concevoir, produire, consommer et jeter les produits. Ils ne sont pas abordés explicitement dans ce module, mais restent au cœur des préoccupations des auteurs de ce module et des mentions relatives à ces travaux sont faits dans certains contenus.

1.2.2. L'intégration physique



Un type d'intégration relative au produit conçu est l'intégration spatiale. Également appelée intégration physique, cette intégration vise généralement à réduire l'encombrement et la masse du système. La figure ci-dessus illustre grâce à un alternateur-démarrreur les différents niveaux d'intégration possibles, allant jusqu'à la fusion des différents éléments matériels (*hardware*) du système.

Ce type d'intégration touche les disciplines mécanique, électrique et électronique.

Les composants mécaniques et électriques sont fortement imbriqués et deviennent même parfois le « packaging » de l'électronique. Ce type d'intégration induit de nouvelles difficultés avec des phénomènes de couplages, faibles et forts.

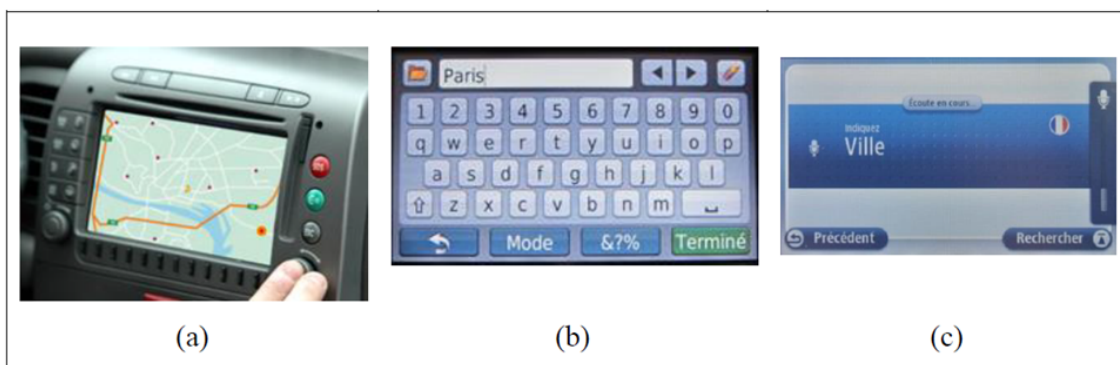
1.2.3. L'intégration fonctionnelle



Intégration fonctionnelle

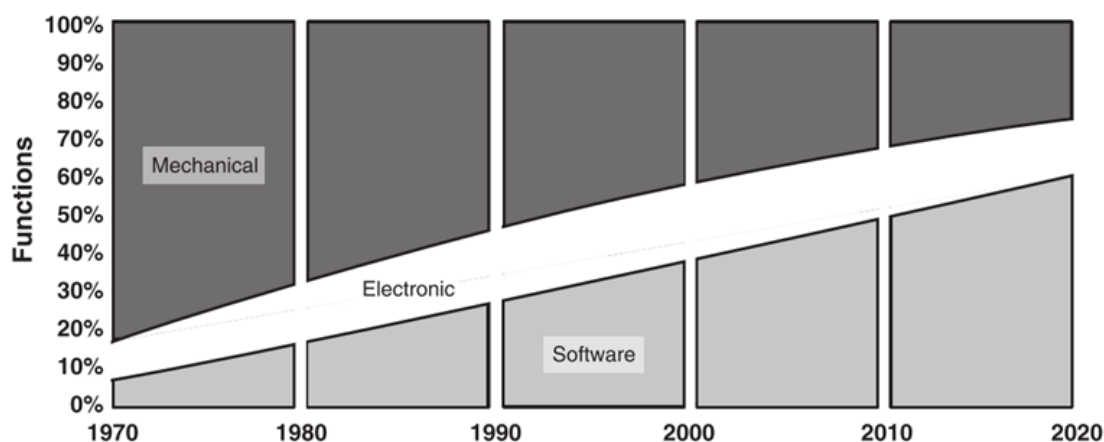
L'intégration fonctionnelle correspond à l'agrégation de fonctions au sein d'un même produit et est illustrée par la figure ci-dessus. Elle présente l'évolution d'un téléphone portable auquel de nombreuses fonctionnalités ont été ajoutées, telles que la gestion de l'agenda, des contacts et des courriels, la prise de photo, la navigation par GPS, le paiement sans contact, etc.

1.2.4. La dématérialisation



Dématérialisation

La dématérialisation est une forme d'intégration fonctionnelle. Elle correspond au remplacement de fonctionnalités jusqu'alors assurées par des composants physiques par de nouvelles fonctions logicielles. Nous illustrons ici cette dématérialisation par le choix de différentes solutions pour remplir la fonction de saisie d'adresse sur un GPS automobile : sur la figure ci-dessus, l'interface purement mécanique (a) est remplacée par un clavier tactile (b) puis par une dictée vocale (c).



Évolution du concept de la mécatronique repris par (Bricogne et al., 2016)

Cette dématérialisation est une tendance qui ne fait que croître depuis les années 70. La figure ci-dessus illustre cette part grandissante des composants logiciels au détriment des composants matériels. Cette tendance s'est développée sous l'ère du logiciel.

1.2.5. L'intégration de la connectivité

Il existe une multitude de technologies permettant d'assurer la connectivité d'un système.

Sans être exhaustif, voici quelques normes ou technologies ayant chacune leurs spécificités en termes de portée, débit ou consommation énergétique :

- Bluetooth 4.2 & 5.0 (BLE)
- WiFi
- WiMax
- Lora
- Sigfox
- ZigBee
- RFID
- NFC
- Weightless (Weightless-W, Weightless-N, Weightless-P)
- DASH7 Alliance Protocol (D7A)
- Réseau mobile : 2G / 2.5G / 2.75G / 3G / 4G (4G+) et 5G



Exemple de normes ou technologies apparues à l'ère de la connectivité (mis à jour en 2020)

Cette connectivité, parfois qualifiée d'ubiquitaire, permet l'amélioration des performances du produit, notamment car cela rend possible des traitements de l'information déportés. Bien au-delà de l'aspect « gadget » qui est prêté à certains produits connectés de grande consommation, ce sont les apports croisés du numérique et de la connectivité qui transforment les produits et leur environnement, leur développement, les modèles économiques associés, mais aussi leur production ou leur recyclage.

1.3. Les produits multidisciplinaires

1.3.1. La mécatronique

L'introduction des produits mécatroniques résulte d'une demande croissante de plus de performance, de flexibilité et de fiabilité dans les produits électromécaniques. Les gains ont été réalisés grâce à la mise en œuvre et l'intégration de nouvelles disciplines (principalement électronique, automatique et informatique).

Ainsi, le concept initial de mécatronique a évolué à partir des années 1970 pour prendre en considération de façon de plus en plus large l'automatique et l'informatique. La mécatronique traduit la recherche d'une synergie entre la mécanique, l'électronique/électricité, l'automatique et l'informatique.

1.3.2. Les produits intelligents

Les « produits intelligents » ou « **smart products** » sont probablement les appellations les plus connues du grand public par leur usage répandu pour caractériser divers produits de grande consommation actuellement commercialisés. Ils sont des objets de la vie courante ayant été « augmentés » par des composants électroniques, électriques, mais également du logiciel et de la connectivité. Grâce à leurs capteurs et actionneurs, ils sont en mesure de percevoir et interagir avec leur environnement ou avec d'autres objets intelligents.

L'intelligence mentionnée peut également posséder deux facettes. L'intelligence peut être centrée sur le produit, avec l'objectif de rendre ce dernier autonome dans ses actions, parfois sans aucune intervention humaine ; ou à l'inverse, être centrée sur l'utilisateur en le rendant plus « intelligent » à travers sa

responsabilisation et sa prise de décisions, pouvant être exécutée par le produit par la suite.

[Extraits du chapitre 2 de la thèse de Thèse Julia Guérineau^[p.71]]

1.3.3. Les systèmes cyber-physiques

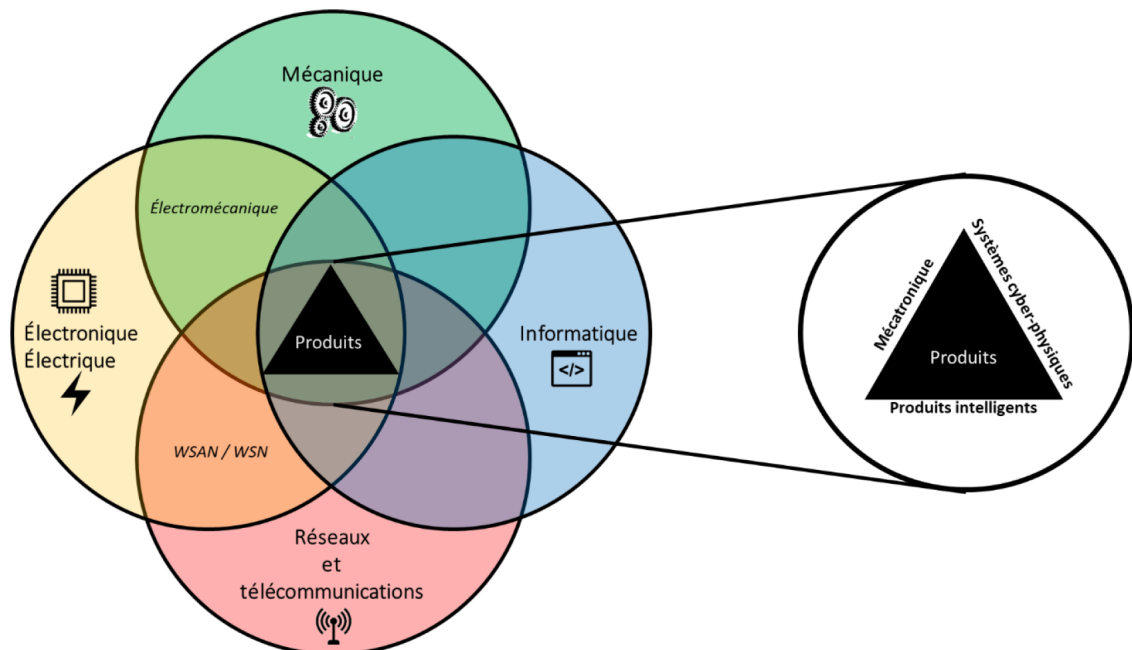
Les systèmes cyber-physiques (SCP), ou **Cyber Physical Products** (CPS), fait référence à l'intégration de capacités en traitement de l'information dans des systèmes physiques. En effet, le traitement de l'information (**computation**), la communication et le contrôle sont trois capacités fondamentales des CPS. Ils sont donc conceptualisés à travers l'idée d'un système virtuel, « **cyber** » composé des 3C, en lien étroit avec le monde physique et le contrôle de ses aspects dynamiques.

Les CPS sont voulus plus performants, résilients, flexibles, autonomes, extensibles – **scalable** – ou encore distribués et sont au final plus complexes à conceptualiser puis à concevoir.

[Extraits du chapitre 2 de la thèse de Thèse Julia Guérineau^[p.71]]

1.3.4. En synthèse : parler de systèmes multidisciplinaires

Que le système à concevoir soit qualifié de système mécatronique, de **Cyber Physical System** ou de système intelligent, il devra intégrer des capacités de captation d'informations liées à son environnement, d'interaction avec son environnement, de traitement des informations, de prise de décisions, de connexion à des infrastructures, de communication, de collaboration avec d'autres systèmes environnants, etc.



Systèmes multidisciplinaires - Synthèse

1.3.5. Besoin d'expertises, pas de silos !

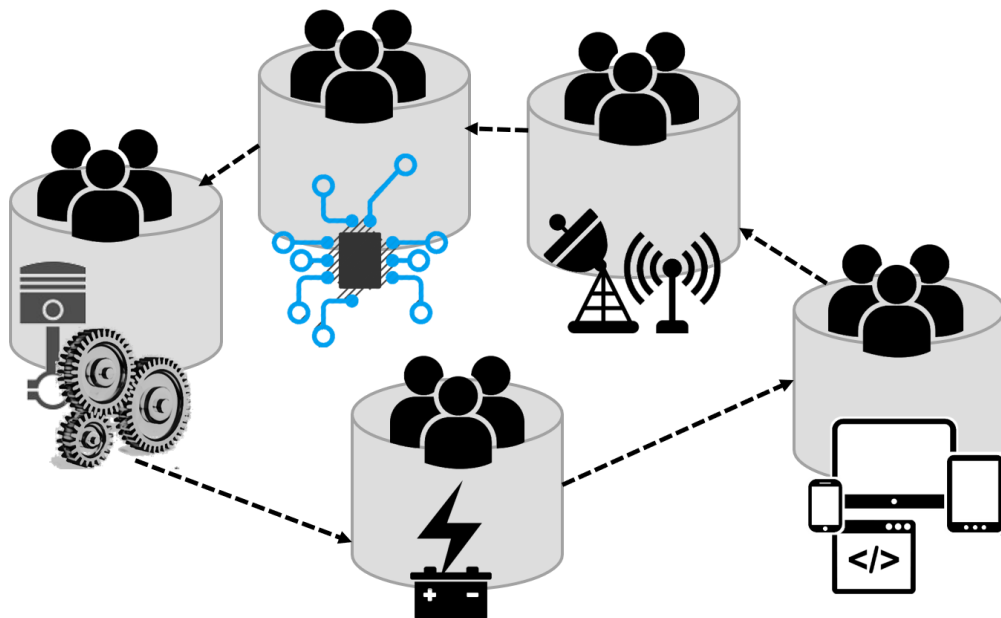
De nombreuses expertises nécessaires

Pour réaliser les nombreuses fonctions d'un système pluritechnologique, de nombreuses expertises sont requises. Ces expertises sont généralement à la croisée d'un rôle dans l'entreprise (conception, industrialisation, production, qualité, etc.) et d'une discipline d'origine (mécanique, électrique, électronique, logiciel, etc.).

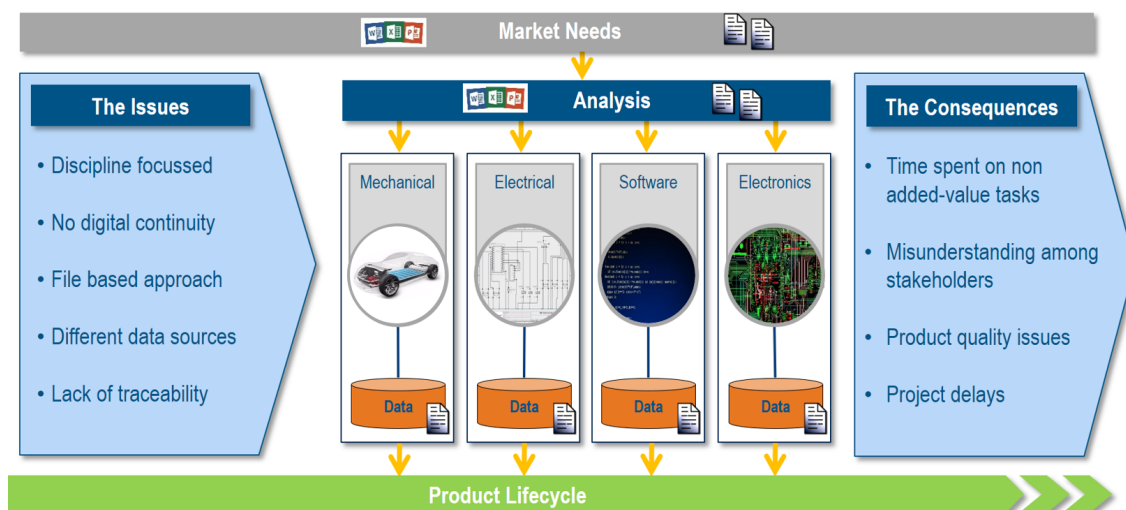
Act. d'ingénierie / Disciplines	Architecture système	Conception / Développement	Calculs / Tests	Méthodes / Industrialisation	Production	...
Electrique						
Electronique		X	← chaque case correspond à une expertise			
Logicielle						
Mécanique						
...						

La collaboration entre ces expertises est rendue difficile par le défaut de compréhension des autres disciplines par un expert donné et par le manque de continuité numérique, les outils spécifiques utilisés qui possèdent leurs propres formats de données et qui sont peu interopérables, la logique de structuration de l'information qui est spécifique, etc.

On parle alors d'un fonctionnement en silos.



Les silos disciplinaires



Le fonctionnement en silos des expertises : problématiques et conséquences

1.4. Et ET-LIOS dans tout ça ?

1.4.1. <ET-LIOS> Module E

Génèse et objectifs du projet

Le projet <ET-LIOS> est porté par le groupement d'intérêt scientifique S.mart (ex AIP-PRIMECA – www.s.mart.fr). Le GIS S.mart fédère plus de 80 universités, établissements internes et composantes universitaires ainsi que des écoles d'ingénieurs publiques ou privées avec une ambition de porter au plan académique les thématiques de l'Industrie du Futur en formation, recherche et innovation.

Dans le cadre du projet proposé, l'ambition est de mutualiser les expériences acquises dans les différents pôles et établissements durant la période du Covid19 en matière de pédagogie à distance et de continuité des formations dans les domaines technologique et scientifique, en particulier pour les sujets de l'Industrie du Futur comme notamment : Conception-simulation-prototypage 3D ; Fabrication avancée et métrologie ; Système cyber-physique de production et eMaintenance ; Jumeau numérique et virtual commissioning des systèmes de production automatisée ; Systèmes intelligents et modélisation multiphysique ; Ingénierie soutenable et responsable.

Le projet ET-LIOS vise à développer et mettre au service des formations à caractère scientifique et technologique une infrastructure numérique en réseau de virtualisation de solutions logicielles et d'hébergement de contenus pédagogiques réalisés. Ces contenus sont structurés en modules thématiques répondant aux besoins de formation en matière d'Industrie du Futur avec un objectif de compréhension et maîtrise des technologies de manière compétitive et soutenable. Enfin, le projet s'appuie sur la dynamique et le réseau du GIS S.mart fédérant un très grand nombre des acteurs universitaires français proposant des diplômes dédiés à l'Industrie du Futur.

Objectifs du module E

Dans ce contexte de développement d'objets connectés, adaptatifs et intelligents, le concepteur intègre

des fonctions, des structures et des comportements de nature sensorielle et motrice liées par un système de contrôle. La conception et la mise en oeuvre de ces composants complexes, légers et résistants, doués d'une « intelligence », capables d'effectuer des mesures et de réaliser des mouvements ou des adaptations de leur comportement, nécessitent de développer et mobiliser des compétences nouvelles. Ces compétences, liées à la conception multidisciplinaire visent notamment à :

- Analyser les exigences et/ou concevoir une architecture fonctionnelle grâce à l'ingénierie système
- Concevoir, dans une perspective d'intégration finale, les fonctions mécanique, électronique, informatique
- Modéliser et simuler lesdites fonctions dans un environnement permettant de forts couplages entre les physiques
- Réaliser des prototypes de composants ou de systèmes intelligents et/ou adaptatifs intégrant leurs fonctions, leurs structures et leurs comportements de nature sensorielle et motrice

Modalités pédagogiques

Ainsi, le module E abordera des sujets comme l'ingénierie système avec un focus sur le **Model Based Systems Engineering**, la simulation système (également appelée 0D/1D), mais également les phases de conception détaillée en électronique, électrique, contrôle/commande et en mécanique.

La logique principale des modalités pédagogiques est de privilégier les classes inversées et les exercices de mise en application à distance pour privilégier le présentiel pour le projet final de conceptions, simulations, réalisations.

Cas d'étude : le drone araignée

Afin de mieux appréhender les connaissances transmises dans le cadre de ce module et de pouvoir appliquer les compétences acquises, un cas fil rouge a été imaginé. Il s'agit d'un drone araignée hexapode.

Pourquoi le drone araignée ?

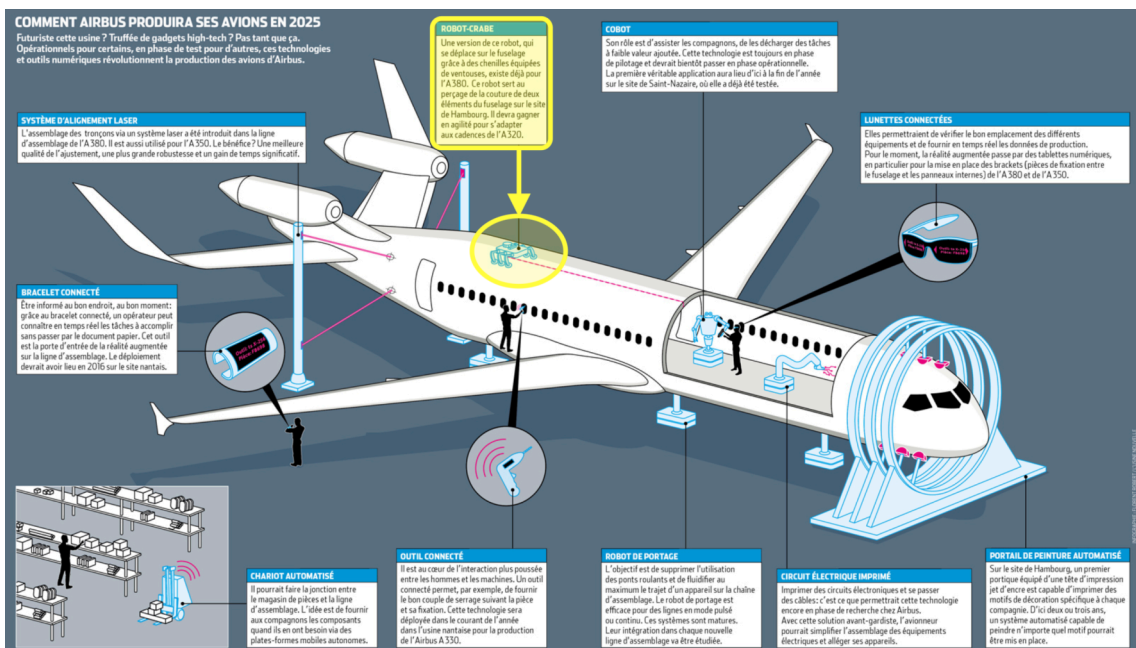
La problématique de l'inspection ou de la maintenance de grandes structures ou de structures 3D complexes réside parfois dans la difficulté d'accès à certains espaces ou encore dans la dangerosité de certaines opérations. Pour faciliter ces inspections, une solution robotisée a été imaginée. Il s'agit d'un drone terrestre grim pant, baptisé « Light Expert Examination Automated Arthropod » - LEEAA. Pour ce faire, une analyse fonctionnelle « traditionnelle » a été réalisée et un premier prototype a été conçu. Il s'agit d'un robot araignée hexapode.



Structure de base du drone araignée

L'objectif de ce drone est de faciliter les inspections et la maintenance de structures telles que des carlingues d'avion ou buildings vitrés.

Une potentielle utilisation du drone araignée est illustrée sur la figure ci-dessous.



Projet d'usine du futur : comment Airbus produira ses avions en 2025

Réglementaire

L'ensemble des travaux s'inspirent, reprennent ou s'appuient sur des réalisations d'étudiants impliqués dans différentes formations :

- aux Arts et Métiers, campus d'Aix en Provence, Mastère Spécialisé® « Créateur de solutions drones : usages innovants et technologies »
- à l'INSA de Lyon, département Génie Mécanique

- à l'IUT de Mantes
- à l'Université de Technologie de Compiègne, département Ingénierie Mécanique
 - module TN29 - « Outils de définition et de développement de systèmes »
 - module TX - « Travaux Expérimentaux »

Un grand merci pour leur implication !

2. Systémique et théorie des systèmes

2.1. Introduction

Objectifs pédagogiques

Appréhender les concepts de base liés à la systémique
Identifier quelques fondateurs de ce courant de pensée
Comprendre la contribution de la théorie des systèmes dans le développement d'une démarche d'ingénierie système
Acquérir le vocabulaire lié aux concepts clé de ce courant de pensée

2.2. Définition d'un système

Dans les fondamentaux de la systémique, un système est considéré comme un système ouvert qui présente des interactions dynamiques avec d'autres systèmes de son environnement. Ce paradigme d'interaction dynamique entre systèmes va permettre de théoriser la définition d'un système et de ses constituants ou d'un système de systèmes et de ses sous-systèmes.

La norme internationale ISO 15288 donne la définition suivante d'un système en tant que : ***combinaison d'éléments en interaction, et qui sont organisés de manière à atteindre un ou plusieurs buts préétablies.***

Cette définition illustre bien les préoccupations principales : un système est un ensemble d'éléments [**préoccupation structurelle**] en interaction [**préoccupation dynamique**], organisés pour atteindre un objectif [**préoccupation liée à l'objectif**].

Voir la norme internationale ISO 15288^[p.70].

La systémique trouve des applications nombreuses et variées en automatique, biologie, cybernétique, économie, informatique, génie industriel, management, sociologie...

◆ Remarque

Parmi les pères fondateurs et contributeurs de la systémique mentionnons Karl Ludwig von Bertalanffy (1901-1972), biologiste autrichien, auteur de General System Theory^[p.70] en 1968 ou Jean-Louis Le Moigne (né en 1931), auteur de la Théorie du Système Général - théorie de la modélisation^[p.70] en 1977.

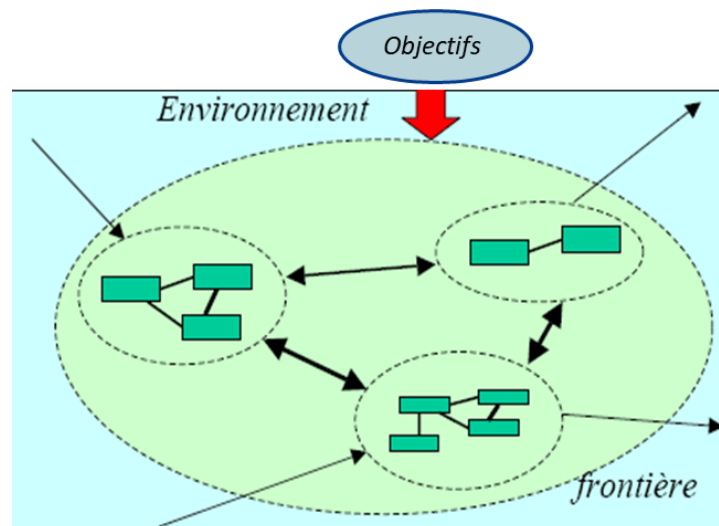


von Bertalanffy

2.3. L'approche systémique

Dans l'approche systémique, un système :

- est doté d'objectifs ;
- possède une frontière, qui détermine ses limites. Cela permet l'identification des éléments qui n'appartiennent pas au système mais qui interagissent avec lui ;
- est constitué de composants : description des éléments qui composent le système et des relations (interactions) entre ces éléments ;
- possède un environnement : ensemble des éléments n'appartenant pas au système mais susceptibles d'affecter ses propriétés ou son aptitude à atteindre ses objectifs.



Caractérisation d'un système

◆ Remarque

Un composant ou encore élément d'un système peut être de nature très diverse : matériel ou physique, compétence humaine, logiciel, données, procédure, processus.

L'intérêt de l'approche systémique est de mettre explicitement en évidence les différentes relations - ou interactions - entre les éléments de l'ensemble considéré, c'est-à-dire entre les constituants du système mais également entre le système et son environnement. Ainsi, l'AFIS rappelle dans son guide « introduction au penser système^[p.70] » que le « penser système » permet d'appréhender des situations complexes et facilite l'identification du système à analyser par :

- la compréhension de la situation ou du système dans son contexte,
- l'anticipation d'évènements engendrés à terme par la situation actuelle,
- la prise en compte de plusieurs points de vue,
- la prise de décision vis-à-vis de risques à éviter ou d'opportunités à saisir.

◆ Remarque

On distingue les systèmes naturels (système solaire ou organismes vivants), non considérés dans ce module, et les systèmes créés par l'homme.

Ces derniers peuvent être des systèmes techniques (voiture, avion, réseau électrique...) ou sociaux-techniques (système politique, système économique, etc.). Ici, nous nous focalisons sur les systèmes techniques, même si de nombreux concepts restent valables pour les systèmes sociaux-techniques.

2.4. La notion d'interaction

✎ Notion d'interaction

La notion d'interaction entre les composants met en lumière l'influence que les éléments peuvent avoir les uns sur les autres.

L'interaction entre éléments peut être de l'échange de matière, d'énergie, d'information ou encore de données, conformément à une procédure (séquence en enchaînement de tâches pouvant préciser l'ordre des échanges réalisés) et selon un vecteur : contact physique, propagation d'ondes (acoustique, électromagnétique, lumineuse, vibratoire...), transfert énergétique ou thermique, transmission d'efforts mécaniques ou fluidiques, transmission de données ou de signaux...

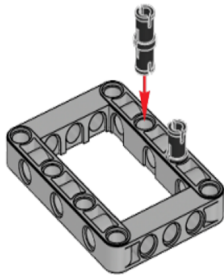
📦 Complément La notion d'interaction est rapidement liée à la notion d'interface

Une interface est un élément à la périphérie d'un système / sous système par laquelle quelque chose est échangé. Dans les travaux de thèse de Chen Zheng^[p.70], 4 types d'interface ont été proposés.

Type: which types of transfers occur through one interface?

1. Geometric interface

Geometric interface indicates how one element is physically connected to another.



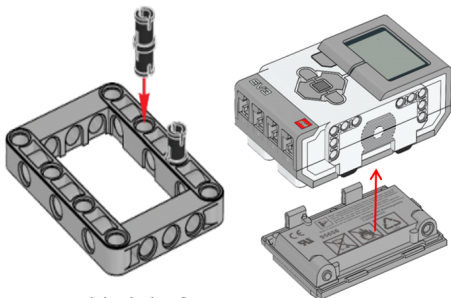
Geometric interface

Type: which types of transfers occur through one interface?

1. Geometric interface

2. Energy interface

Energy interface indicates how energy (electrical energy, mechanical energy...) is transferred between elements



Geometric interface Energy interface

▪ Interface classification

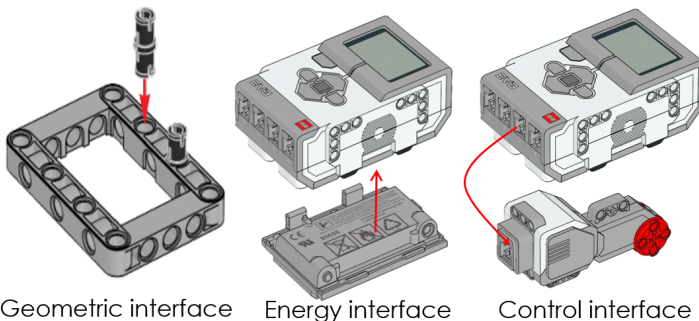
Type: which types of transfers occur through one interface?

1. Geometric interface

2. Energy interface

3. Control interface

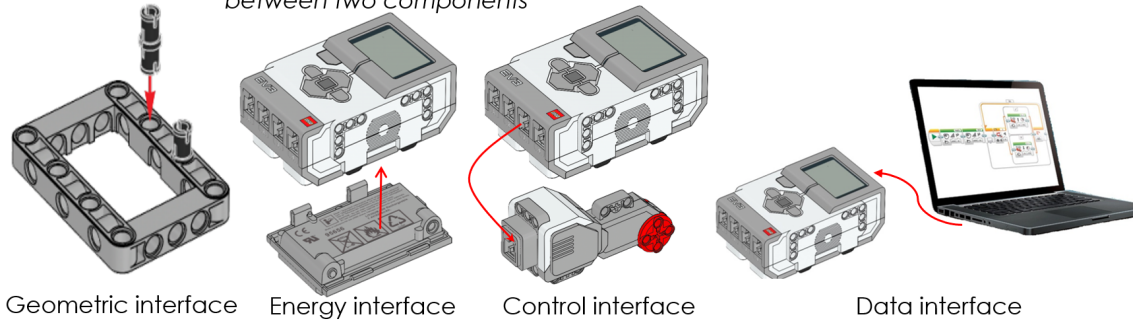
Control interface indicates how one element will be controlled by another.



Geometric interface Energy interface Control interface

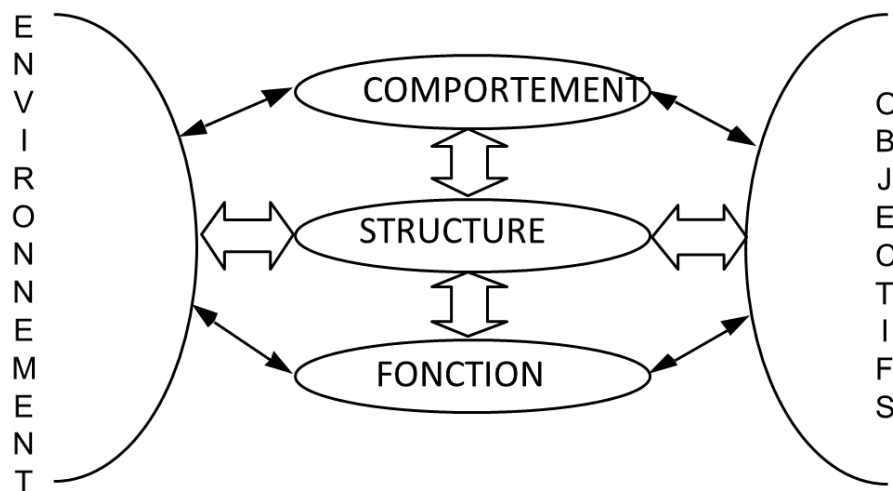
- Interface classification
 - Type: which types of transfers occur through one interface?
 - 1. Geometric interface
 - 2. Energy interface
 - 3. Control interface
 - 4. Data interface

Data interface indicates how communication information is transferred between two components



2.5. La notion de point de vues

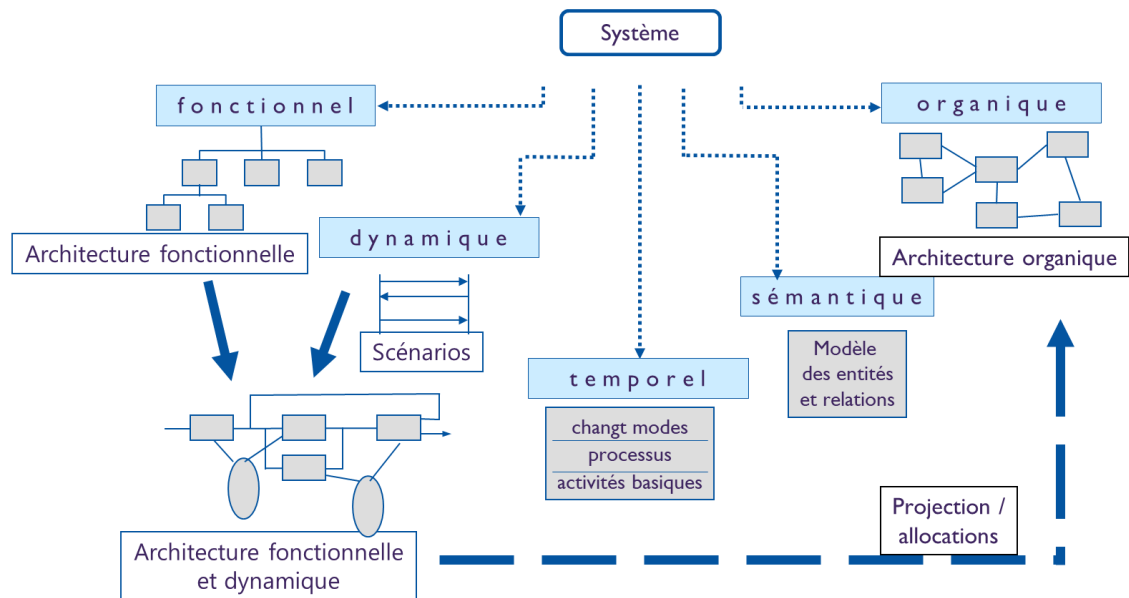
Un apport important de la systémique est également la notion de point de vues qu'il est possible d'adopter pour considérer un système. Classiquement les vues fonctionnelle, comportementale et structurelle (issues du modèle FBS : Function-Behavior-Structure^[p.70]) sont a minima prises en compte.



Concepts du modèle FBS

Mais d'autres points de vues peuvent être adoptés. On parle ainsi des vues :

- Fonctionnelle : transformations des entrées en sorties
- Architecturale (organique) : les composants du système
- Sémantique : définition des concepts manipulés
- Statique : les relations entre éléments
- Dynamique : les évolutions dans le temps du système (comportement)
- etc.



(Adapté d'après Alain Faisandier)

2.6. Les concepts liés à la systémique

📖 Réglementaire

La plupart de ces définitions sont issues de Introduction au Penser Système^[p.70].

📖 Système

Un système est un tout, composé d'un ensemble d'éléments qui interagissent de façon organisée pour atteindre un ou plusieurs objectifs définis.

- Définition INCOSE : "An integrated set of elements, sub-systems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services and other support elements"
- Définition ISO 15288 : "A combinaison of interacting elements organized to achieve one or more stated purposes"

📖 Finalité

La finalité est le caractère de ce qui tend vers un but.

La finalité du système correspond au « pourquoi », au sens de ce dernier. C'est sa raison d'être.

📖 Parties prenantes

Une partie prenante est une entité concernée par le système, sa conception, son utilisation, ses impacts dans un contexte donné. De ce fait, la partie prenante est susceptible d'émettre (ou d'avoir

sans nécessairement savoir les exprimer !) des besoins ou des contraintes vis-à-vis du système.

Structure

La structure est la manière dont les parties d'un tout sont arrangées entre elles.

Dans le cadre du système considéré, il représente l'organisation des parties du système qui lui donne sa cohésion et constitue une de ses caractéristiques permanentes.

Architecture

L'architecture est l'organisation fondamentale d'un système représenté par ses constituants, leurs interrelations, leurs relations avec l'environnement.

L'architecture d'un système est une représentation, à un niveau d'abstraction et de granularité donné, d'un système sous forme d'une structure identifiant les éléments constitutifs du système et leurs interactions.

Comportement

Le comportement du système est l'ensemble des phénomènes organisés par lesquels le système, agit et réagit par rapport à son environnement et aux parties prenantes.

Fonctionnement

Le fonctionnement d'un système est sa manière de remplir sa fonction, ses objectifs et d'atteindre sa finalité.

Etat

Un état correspond à un comportement caractéristique et temporaire du système à un instant d'observation donné.

Cycle de vie d'un système

Le cycle de vie d'un système correspond à son évolution dans le temps. Il peut-être décrit par des états successifs. Cette notion est à rapprocher de la notion de cycle de vie du « Product Lifecycle Management [[https://fr.wikipedia.org/wiki/Gestion_du_cycle_de_vie_\(produit\)](https://fr.wikipedia.org/wiki/Gestion_du_cycle_de_vie_(produit))] », mais ne possède pas nécessairement la même couverture.

Relation

Une relation est un lien d'influence ou d'interaction.

Il existe des relations cause-effet : une situation, un événement, génère la mise en oeuvre d'une réponse dynamique de la part du système pour réguler, temporiser, démultiplier, etc.

Il existe des relations cause-effet-cause ou boucle de rétroaction : elles permettent d'appréhender l'impact de la variation d'un événement sur son environnement et la façon dont cet environnement réagit vis-à-vis du système, en créant un phénomène de régulation ou d'amplification.

**Influence**

L'influence est l'action qu'exerce quelque chose sur une autre chose ou sur quelqu'un. Elle est monodirectionnelle.

**Interaction**

L'interaction est la réaction réciproque de deux phénomènes l'un sur l'autre. Elle est bidirectionnelle

3. L'Ingénierie Système - IS

3.1. Introduction

Objectifs pédagogiques : Appréhender l'approche système

Comprendre en quoi l'Ingénierie Système est une façon de mettre en œuvre une « approche systémique »

Connaître les grandes étapes d'une démarche d'Ingénierie Système

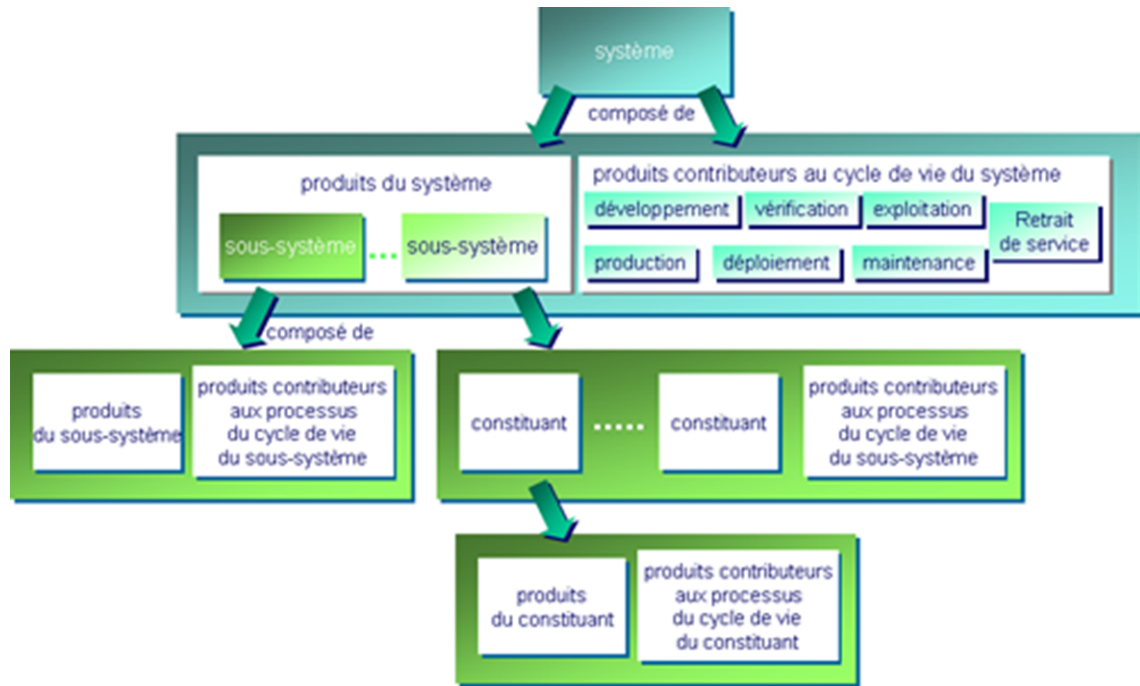
3.2. L'ingénierie système en bref...

L'Ingénierie Système (IS) est une démarche méthodologique coopérative et interdisciplinaire, fondée sur la science et l'expérience, qui englobe l'ensemble des activités adéquates pour concevoir, développer, faire évoluer et vérifier un système, apportant une solution économique et performante aux besoins des parties prenantes et acceptable par tous. Dans cette perspective, un système se définit par un ensemble intégré de produits, processus et compétences humaines.

L'IS doit s'entendre comme une approche globale et systématique apte à mettre en cohérence les méthodes et démarches existantes, telles que l'analyse fonctionnelle, l'analyse de la valeur, la sûreté de fonctionnement, la conception à coût objectif ou des méthodes de conception disciplinaire associées à une spécialité, un domaine ou un métier particulier et bénéficiant de leurs retours d'expérience.

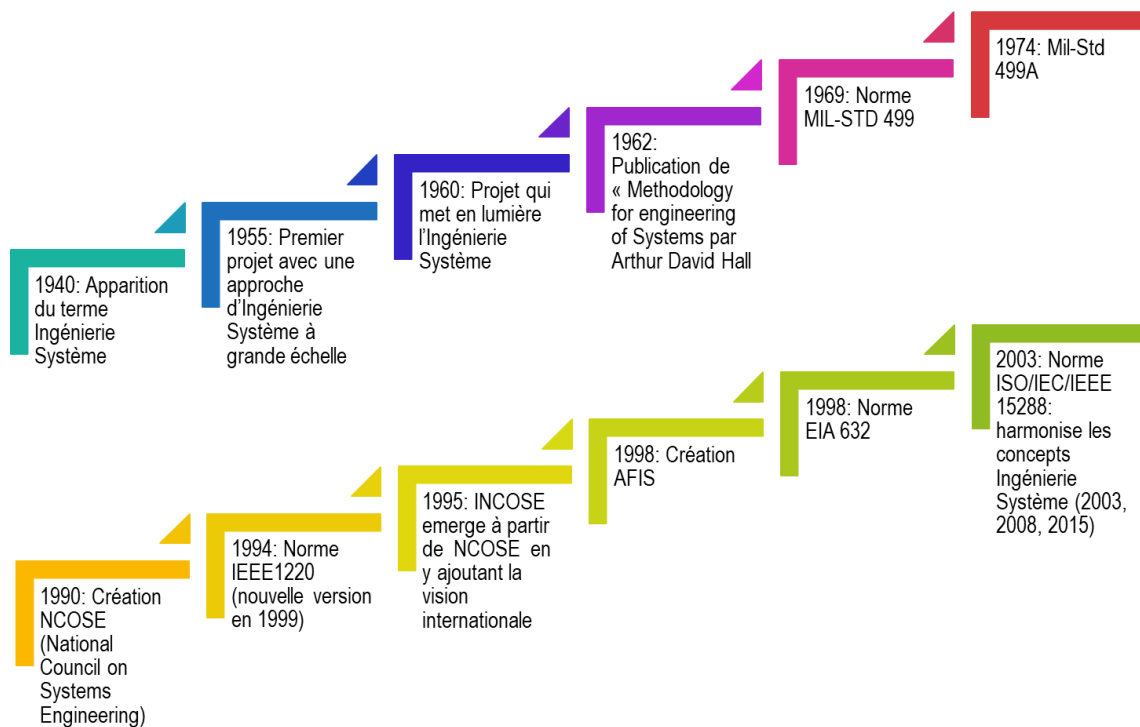
En IS, la définition du système comporte :

- celle de ses sous-systèmes et constituants (matériels, logiciels, organisations et compétences humaines) et de leurs interfaces, sièges des interactions recherchées ;
- celles des processus de leurs cycles de vie permettant de les concevoir, produire, vérifier, distribuer, déployer, exploiter, maintenir en condition opérationnelle et retirer du service, et donc des produits contributeurs nécessaires à ces processus.

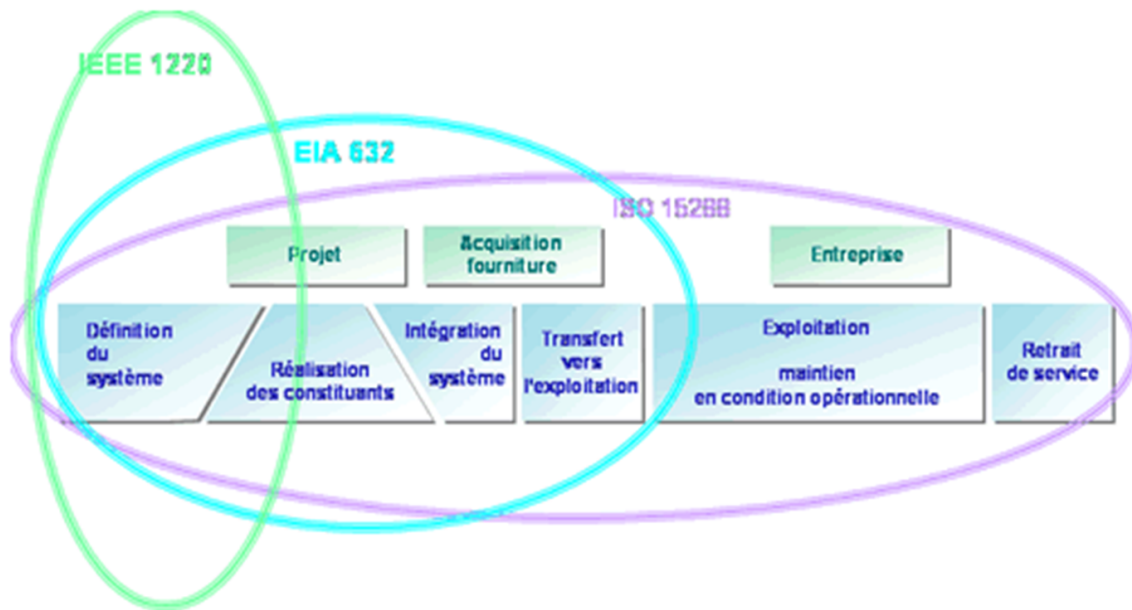


Composition d'un système en sous-systèmes et contributeurs au cycle de vie

3.3. Historique de l'Ingénierie Système

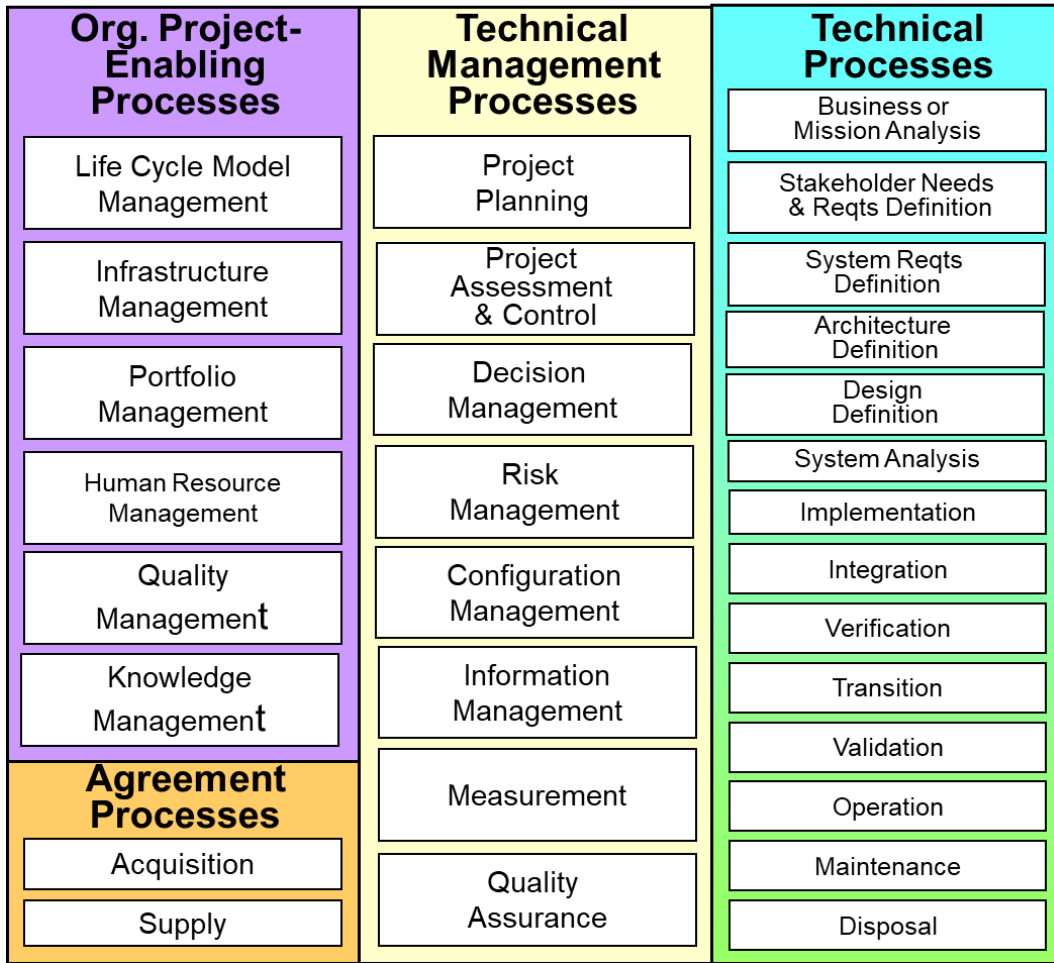


3.4. Standards et normes de l'Ingénierie Système

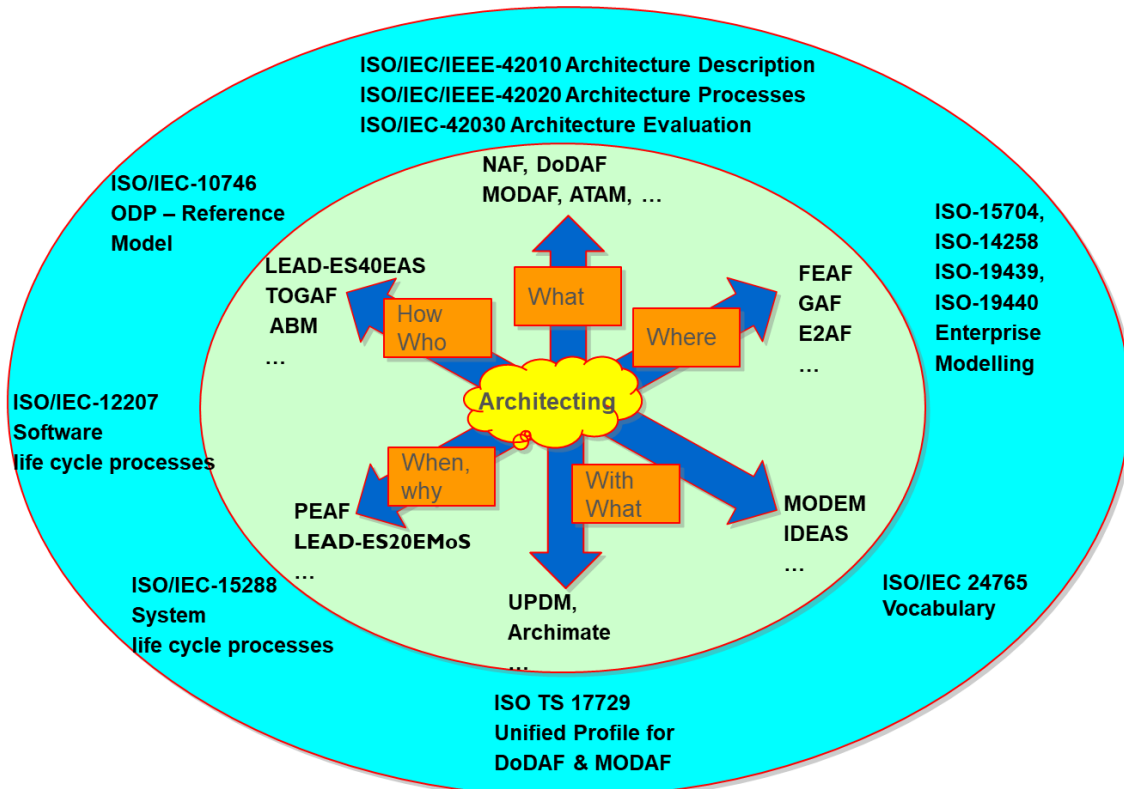


De l'IEEE 1220 à l'ISO 15288 : couverture respective des trois normes par rapport aux activités techniques sur le cycle de vie du système, aux activités de management du projet, aux activités de gestion contractuelle aux activités de management de l'IS dans l'entreprise.

15288:2015 Processes



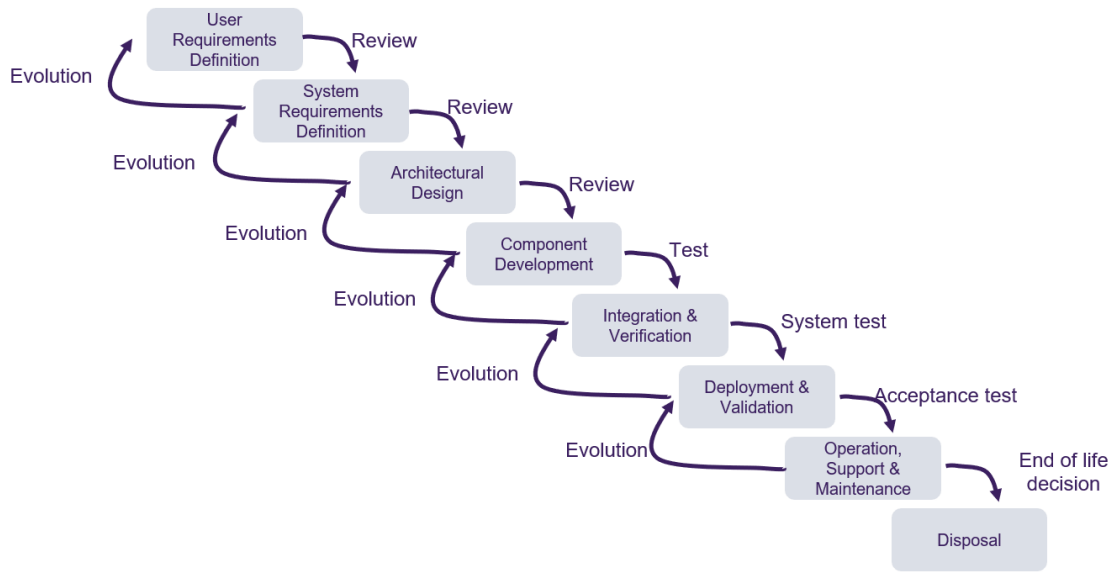
Norme ISO 15288 : 2015



Toujours plus de standards et normes en Ingénierie Système

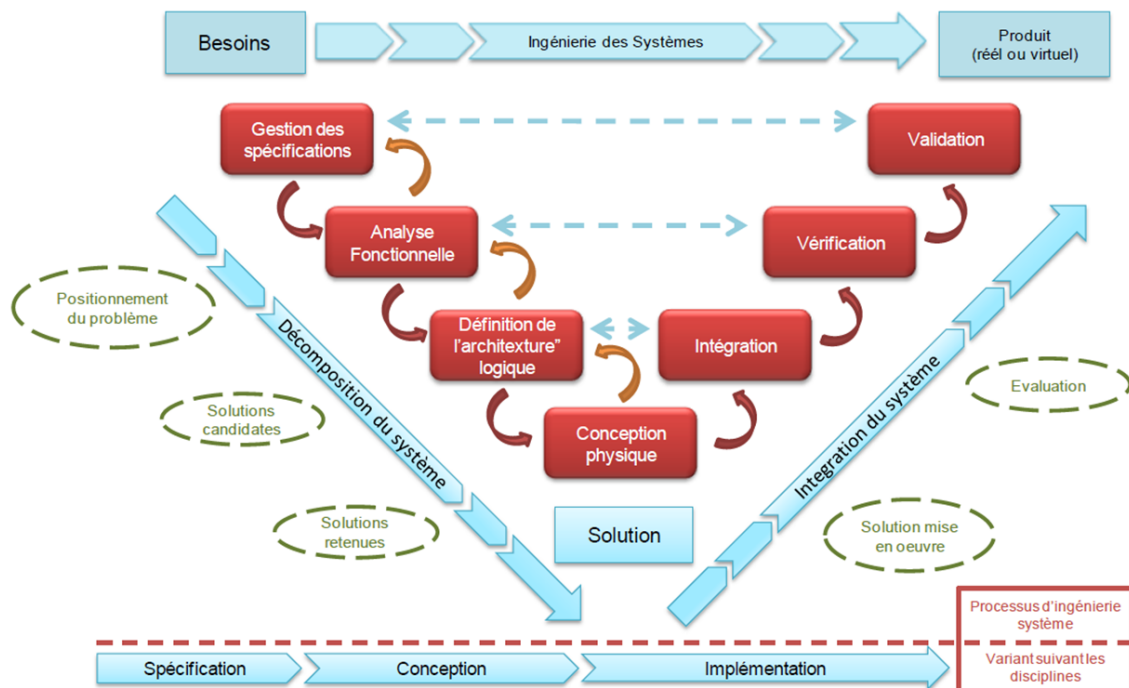
3.5. Les principaux concepts au cœur de l'IS

Entre l'initialisation d'un projet et la production d'un système, il est nécessaire de passer par de nombreuses étapes pour concevoir un système. Il existe de nombreux modèles de processus. Le plus basique est le cycle en cascade (waterfall). Très linéaire, il ne permet pas les itérations et ne permet pas les anticipations.



W. Royce 1970
 "Managing the development of large software systems"

Par exemple, le plus traditionnel à ce jour, est le cycle en 'V', présenté ci-après.

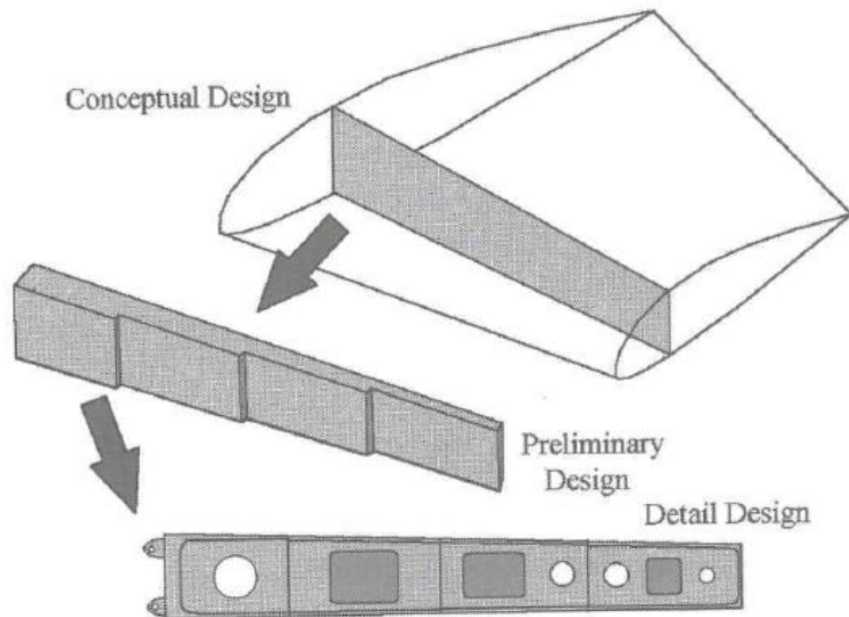


Un exemple de cycle en V

❖ Pour aller plus loin D'autres exemples de processus

Des macros étapes que l'on retrouve « partout »

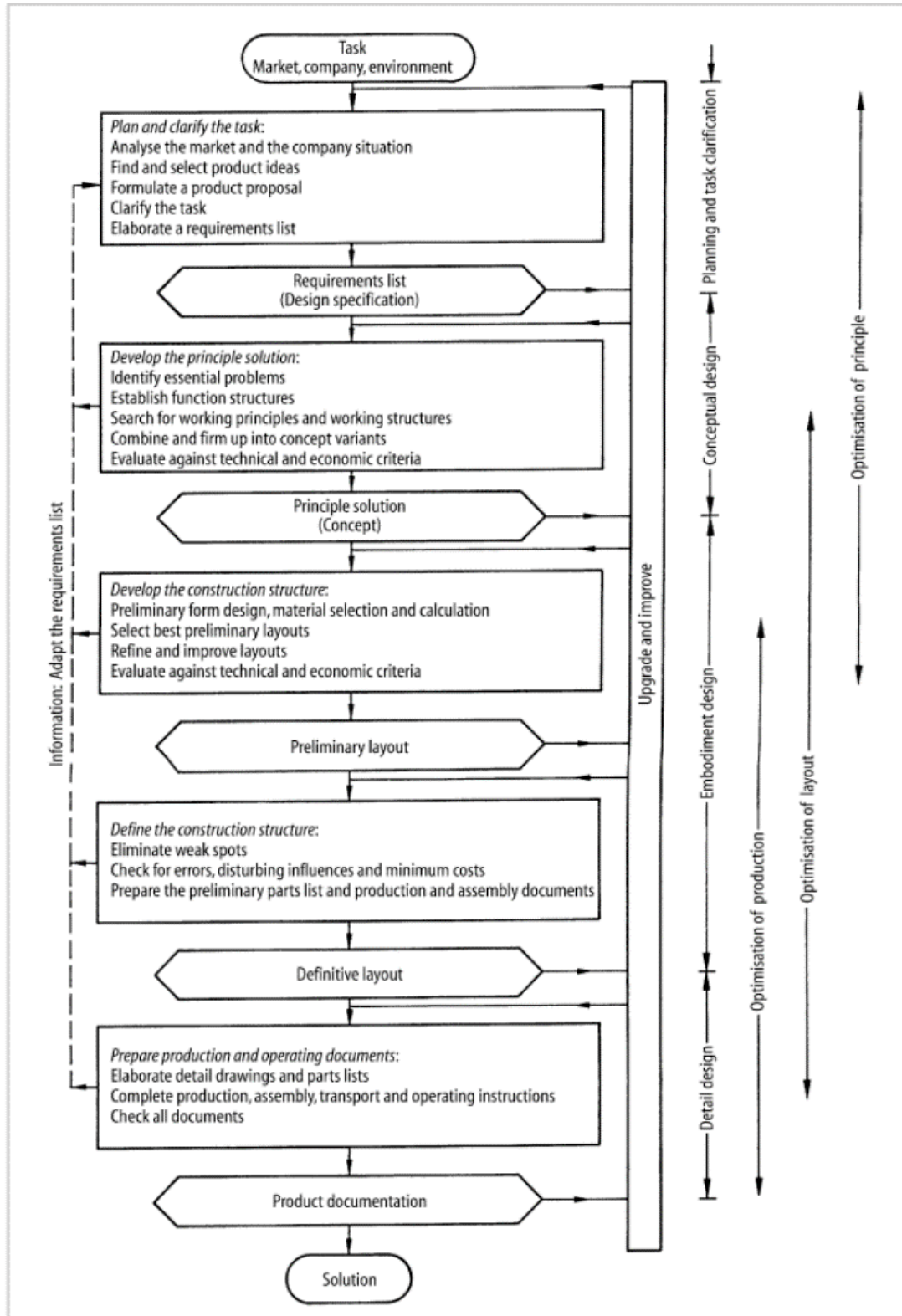
- Planning et clarification des tâches
- Élaboration du concept
- Conception préliminaire
- Conception détaillée



Design phases: front wing spar

(Raymer, 1999^[p.70])

LA référence : un processus **systematique**



(Pahl & Beitz, 2003^[p.70])

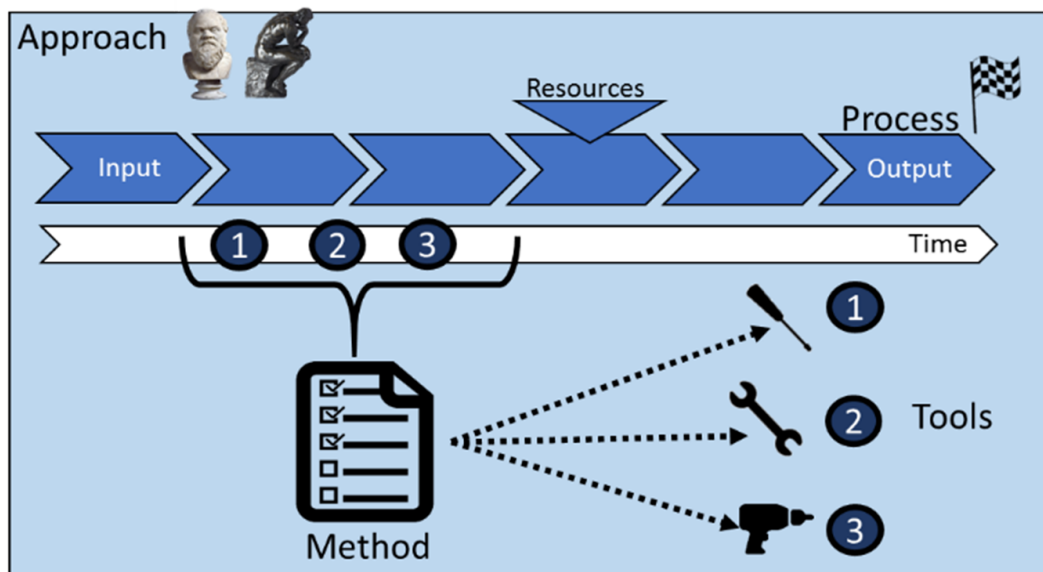
Un sujet très étudié...

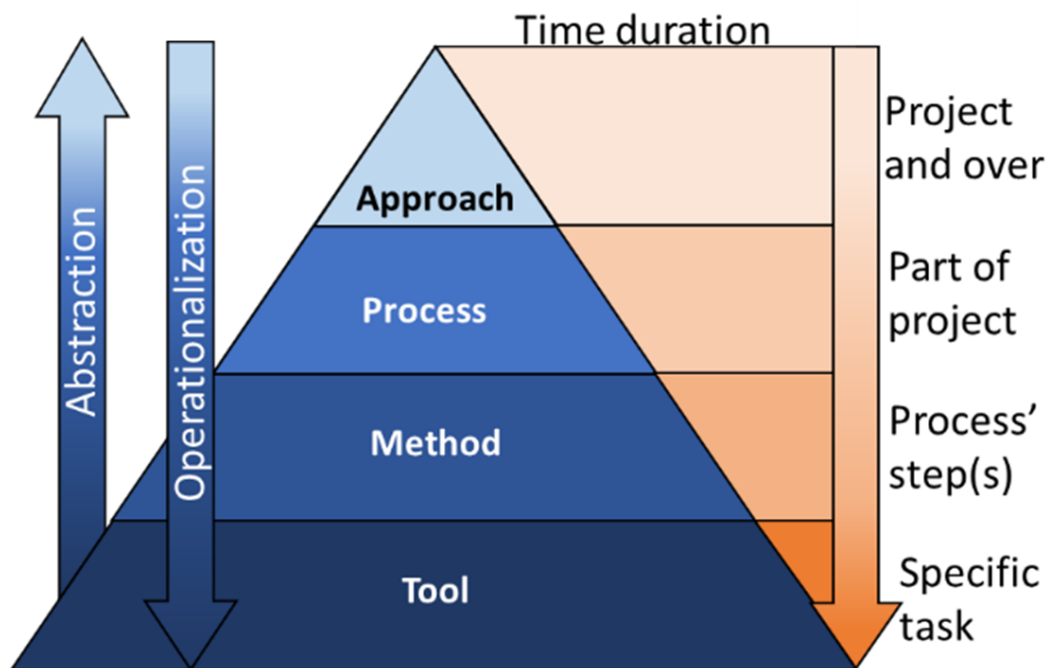
Models	Establishing a need phase	Analysis of task phase	Conceptual design phase		Embodiment design phase		Detailed design phase		Implementation phase	
Booz et al. (1967)	X	New product strategy development	Idea generation	Screening & evaluation	Business analysis	Development	Testing	Commercialisation		
Archer (1968)	X	Programming : data collection	Analysis	Synthesis	Development		Communication	X		
Svensson (1974)	Need	X	Concepts	Verification	Decisions	X		Manufacture		
Wilson (1980)	Societal need	Recognize & formalize	FR's & constraints	Ideate and create	Analyze and/or test		Product, prototype, process	X		
Urban and Hauser (1980)	Opportunity identification	Design			Testing			Introduction (launch)	Life cycle management	
VDI-2222 (1982)	X	Planning	Conceptual design	Embodiment design	Detail design		X			
Hubka and Eder (1982)	X	X	Conceptual design	Lay-out design	Detail design		X			
Crawford (1984)	X	Strategic planning	Concept generation	Pre-technical evaluation	Technical development		Commercialisation			
Pahl and Beitz (1984)	Task	Clarification of task	Conceptual design	Embodiment design	Detailed design		X			
French (1985)	Need	Analysis of problem	Conceptual design	Embodiment of schemes	Detailing		X			
Ray (1985)	Recognise problem	Exploration of problem	Define problem	Search for alternative proposals	Predict outcome	Test for feasible alternatives	Judge feasible alternatives	Specify solution	Implement	
Cooper (1986)	Ideation	Preliminary investigation	Detailed investigation	Development	Testing & Validation	X		Full production & market launch		
Andreasen and Hein (1987)	Recognition of need	Investigation of need	Product principle	Product design		Production preparation		Execution		
Pugh (1991)	Market	Specification	Concept design			Detail design	Manufacture	Sell		
Hales (1993)	Idea, need, proposal, brief	Task clarification	Conceptual design	Embodiment design	Detail design		X			
Baxter (1995)	Assess innovation opportunity	Possible products	Possible concepts	Possible embodiments	Possible details		New product			
Ulrich and Eppinger (1995)	X	Strategic planning	Concept development	System-level design	Detail design		Testing & refinement	Production ramp-up		
Ullman (1997)	Identify needs	Plan for the design process	Develop engineering specifications	Develop concept	Develop product			X		
BS7000 (1997)	Concept	Feasibility	Implementation (or realisation)					Termination		
Black (1999)	Brief/concept	Review of 'state of the art'	Synthesis	Inspiration	Experimentation	Analysis / reflect	Synthesis	Decisions to constraints	Output	X
Cross (2000)	X	Exploration	Generation	Evaluation	Communication		X			
Design Council (2006)	Discover	Define	Develop	Deliver			X			
Industrial Innovation Process 2006	Mission statement	Market research	Ideas phase	Concept phase	Feasibility Phase	Pre production				

(Howard, 2008^[p.70])

3.6. APMO - un cadre d'analyse pour les techniques et méthodes

Afin de mieux comprendre ce qui distingue les différentes techniques et méthodes, les hiérarchiser, positionner les liens entre elles, Julia Guérineau^[p.71] a proposé un cadre d'analyse dénommé APMO pour Approche - Processus - Méthode - Outil.





Approche

Une approche peut être vue comme une **philosophie**. C'est un **regroupement de principes** permettant d'aborder de manière macroscopique un développement produit. La liste de principes peut être transposée sous la forme de processus, méthodes et outils, afin de l'opérationnaliser. L'approche se positionne au niveau stratégique de l'entreprise.

Processus

Un processus regroupe une **série d'étapes** mésoscopiques **organisées de manière temporelle** pour répondre à une finalité et disposant d'éléments d'entrée (besoin client, cahier des charges), de ressources (financières, humaines, informatiques, etc.) en vue d'obtenir un résultat pouvant prendre la forme d'un produit. Le processus peut être itératif et comporter des jalons. Un processus organise les étapes du développement produit et s'appuie sur des méthodes et des outils pour l'opérationnalisation des étapes. Le processus est relatif au niveau tactique de l'entreprise.

Méthode (dans le cadre de la conception de produit)

Mise en application d'un **regroupement de règles et de pratiques d'ingénierie**, s'inscrivant dans un processus, permettant de réaliser une démarche technique en vue d'atteindre un résultat. Les tâches de la méthode peuvent être à leur tour réalisées à l'aide d'outils. La méthode se positionne au niveau opérationnel de l'entreprise.

Outil

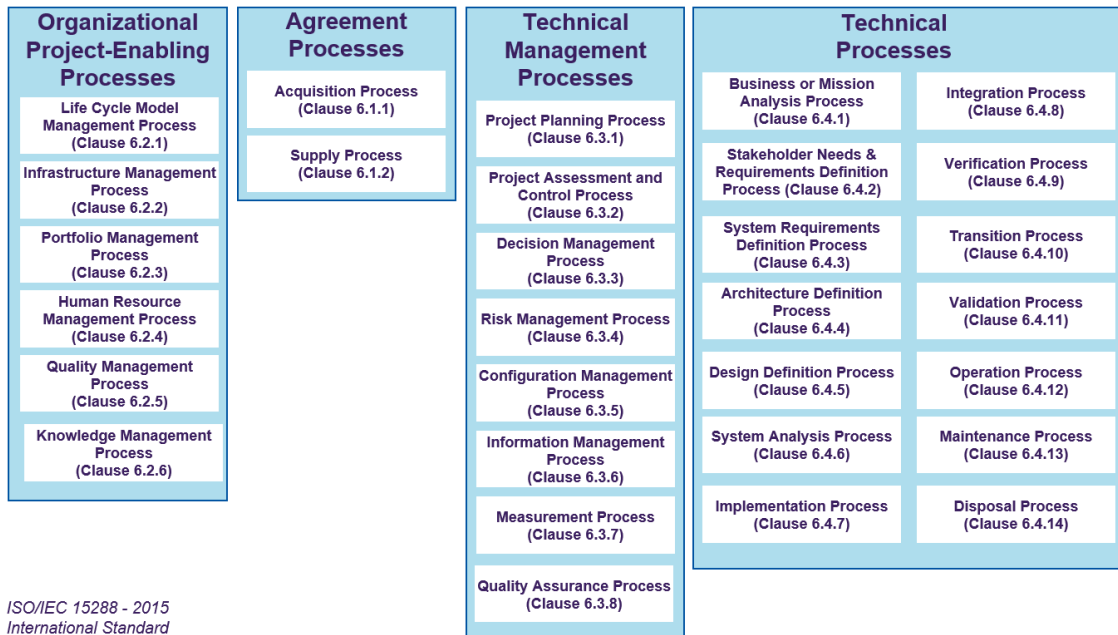
Un outil est un **moyen d'aide à la réalisation d'une tâche circonscrite d'une méthode en vue d'obtenir et/ou d'améliorer un résultat**, d'agir sur un élément en particulier. Un outil peut ainsi aider à réaliser ou supporter une tâche. L'outil est en lien avec le niveau opérationnel de l'entreprise. Un outil intervient dans un but et à un moment défini du développement produit. On peut distinguer les **outils d'édition** qui ont pour but de créer/produire/améliorer un résultat ; et les **outils de**

| **gestion** qui visent à partager et maintenir un résultat.

3.7. Les différents processus de l'Ingénierie Système et le focus de ce cours

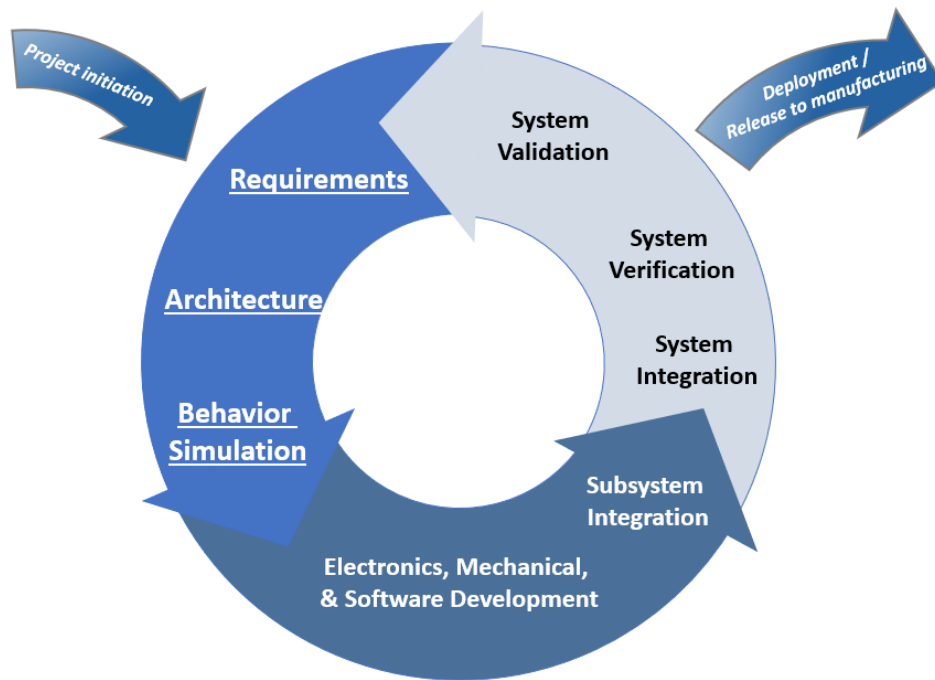
Les processus de la norme ISO 15288

Il existe de très nombreux processus dans la norme ISO 15288.



- Les processus « agreement » permettent d'établir une relation fiable entre les parties prenantes d'un projet : il s'appuie sur une reconnaissance mutuelle des termes et conditions dans lesquels une relation de travail est menée entre 2 organisations. Cela peut concerner le client comme les fournisseurs, etc.
- Les processus liés à la gestion du projet : ils permettent de décrire et contrôler l'exécution du projet afin d'en suivre l'avancement.
- Les processus organisationnels : ils décrivent comment une entreprise doit être organisée pour garantir sa capacité à acquérir/fournir un produit ou un service, en s'appuyant notamment sur les activités transversales (projets, infrastructures, ressources, qualité...)
- Les processus techniques, qui correspondent à ce qui est principalement décrit dans ce module.

Ces processus techniques sont repris sous une forme circulaire ci-dessous :



Ainsi, on retrouve sur la figure :

- l'élicitation et la formalisation des exigences liées au besoin
- l'élaboration de l'architecture du système
- l'usage de simulation très tôt dans le processus pour orienter la conception voire entériner certains choix
- réaliser le développement détaillé des différents sous systèmes et composants
- intégrer progressivement des sous-systèmes au système
- vérifier et valider que les exigences initiales sont bien respectées

❖ Pour aller plus loin IS vs. autres approches dans la pratique ?

3 des caractéristiques principales de l'IS sont, en comparaison de ce que revendiquent d'autres approches :

- la capture, l'analyse et le suivi des exigences du client tout au long du processus
- la conception basée sur des modèles pour une conception holistique de l'architecture du système
- la simulation du comportement du système pour une validation précoce des performances du produit

4. Model-Based Systems Engineering (MBSE)

4.1. Introduction

Objectifs pédagogiques : MBSE

Comprendre l'intérêt de la modélisation pour la conception de systèmes multidisciplinaires
Comprendre le rôle du MBSE dans une démarche d'Ingénierie Système

Objectifs pédagogiques : SysML

Découvrir le langage SysML pour mettre en œuvre le MBSE
Savoir modéliser un système en choisissant les diagrammes SysML les plus appropriés

4.2. Pourquoi le MBSE ?

« L'ingénierie système 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.

La transformation d'un besoin émergeant en la définition d'un système lui apportant une solution met en œuvre de multiples activités intellectuelles faisant passer progressivement de concepts abstraits à la définition rigoureuse de produits. Il est nécessaire de s'appuyer sur des représentations tant du problème que de ses solutions possibles à différents niveaux d'abstraction pour appréhender, conceptualiser, concevoir, estimer, simuler, valider, justifier des choix, communiquer. C'est le rôle de la modélisation ! » SysML par l'exemple - Un langage de modélisation pour systèmes complexes^[p.70]

La modélisation est l'activité principale d'ingénierie. Qu'elle soit mathématique, géométrique, conceptuelle, etc. nous avons besoin de modèles pour réaliser une abstraction de la réalité, afin d'être en mesure de comprendre puis de réaliser une action donnée.

Si nous voulons que l'ensemble des acteurs d'un projet de conception acquiert une vision commune concernant le système à concevoir, nous ne pouvons pas nous appuyer sur un simple descriptif textuel. En effet, les textes rédigés, qu'ils soient descriptifs ou prescriptifs, sont soumis à un risque de mauvaise interprétation, ou d'une divergence importante dans la compréhension des parties prenantes. De plus, l'évolution de ces descriptifs engendre des problèmes de mise à jour, de cohérence ou de suivi des différentes versions.

La modélisation permet de résoudre en partie ces difficultés. Le Model-Based Systems Engineering (MBSE) permet de supporter une démarche d'Ingénierie Système en s'appuyant sur des modèles.

4.3. Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering (MBSE)

≈ Ingénierie Système Basée sur les Modèles (ISBM)

Le MBSE permet de conduire une démarche d'Ingénierie Système en s'appuyant sur des modèles, ce qui permet de partager une vision commune entre les acteurs du projet.

Le MBSE, également parfois appelé Model-Based Systems Development (MBSD), est un paradigme de processus d'ingénierie des systèmes qui met l'accent sur l'application de principes rigoureux de modélisation de l'architecture et des meilleures pratiques aux activités d'ingénierie des systèmes tout au long du cycle de vie du développement des systèmes. Ces activités comprennent, entre autres, l'analyse des besoins, l'analyse fonctionnelle, l'analyse des performances, certains aspects de la conception du système, la spécification de l'architecture du système, ainsi que la vérification et la validation (V&V) du système.

Les modèles réalisés s'appuient sur une syntaxe et une sémantique formalisées.

Exemple

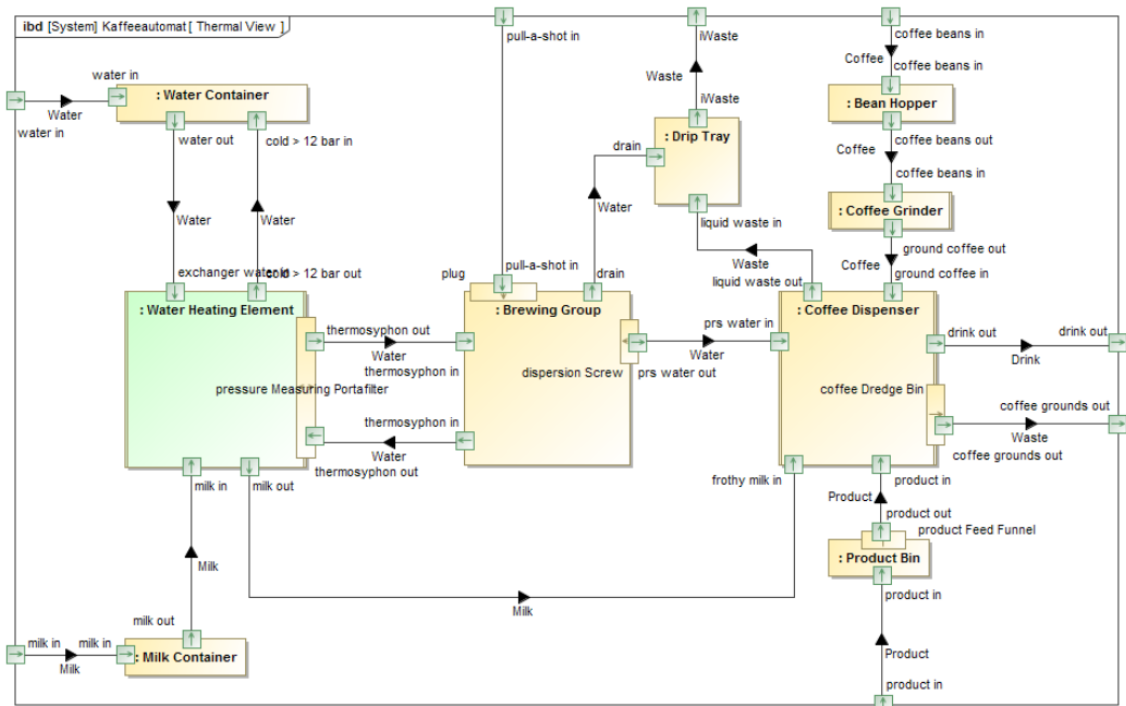


Diagramme de blocs internes d'une machine à café

4.4. Quelques exemples de langage modélisation

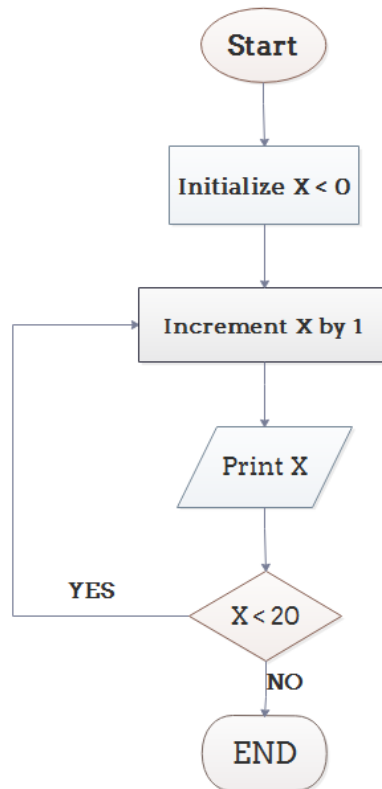
Il existe différentes techniques de modélisation. Elles sont à la base de la collaboration de nombreuses

disciplines d'ingénierie. Parmi ces langages de modélisation, on retrouve par exemple :

Logigrammes

≈ *flow diagrams*

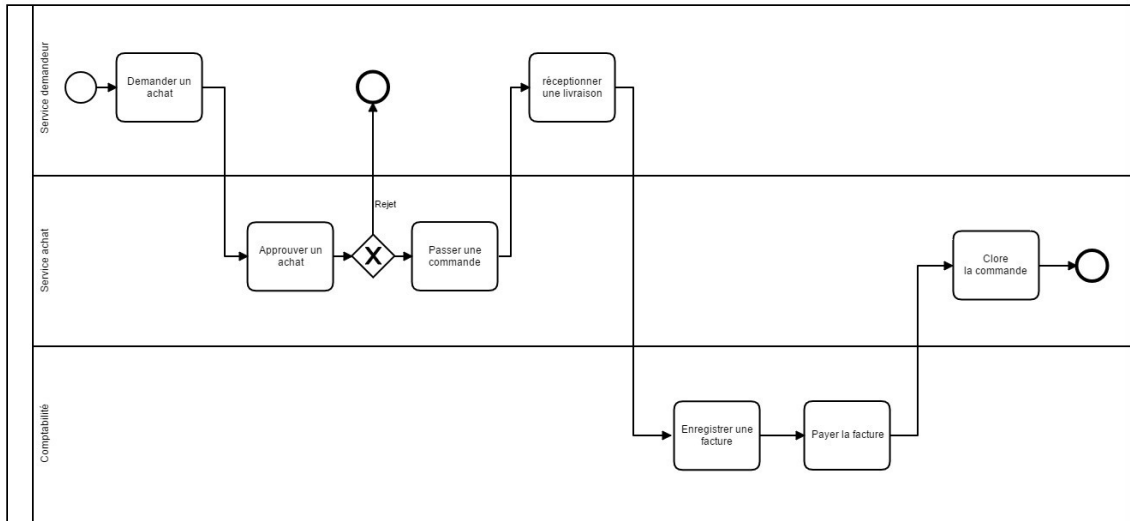
Ils permettent de visualiser les actions à mener, étapes et les décisions à prendre pour atteindre un objectif défini. Lorsqu'ils sont utilisés pour présenter un algorithme, on parle de Flowchart [https://fr.wikipedia.org/wiki/Organigramme_de_programmation].



BPMN

≈ *Business process model and notation*

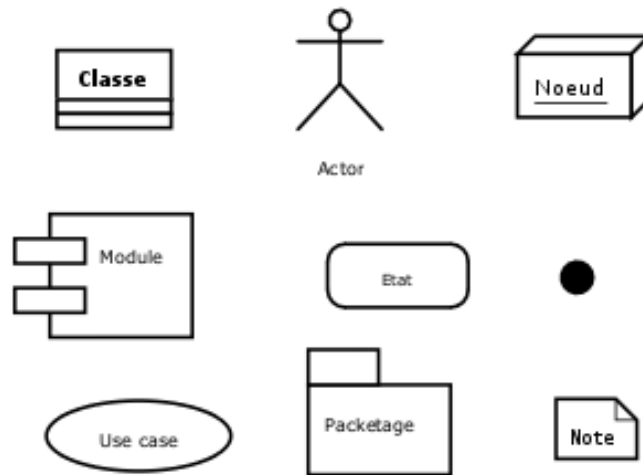
Plus qu'un langage normalisé, le BPMN [https://fr.wikipedia.org/wiki/Business_process_model_and_notation] est une méthode de modélisation des processus d'affaires pour décrire les chaînes de valeur et les activités métier d'une organisation sous forme d'une représentation graphique.



UML

≈ Unified Modeling Language

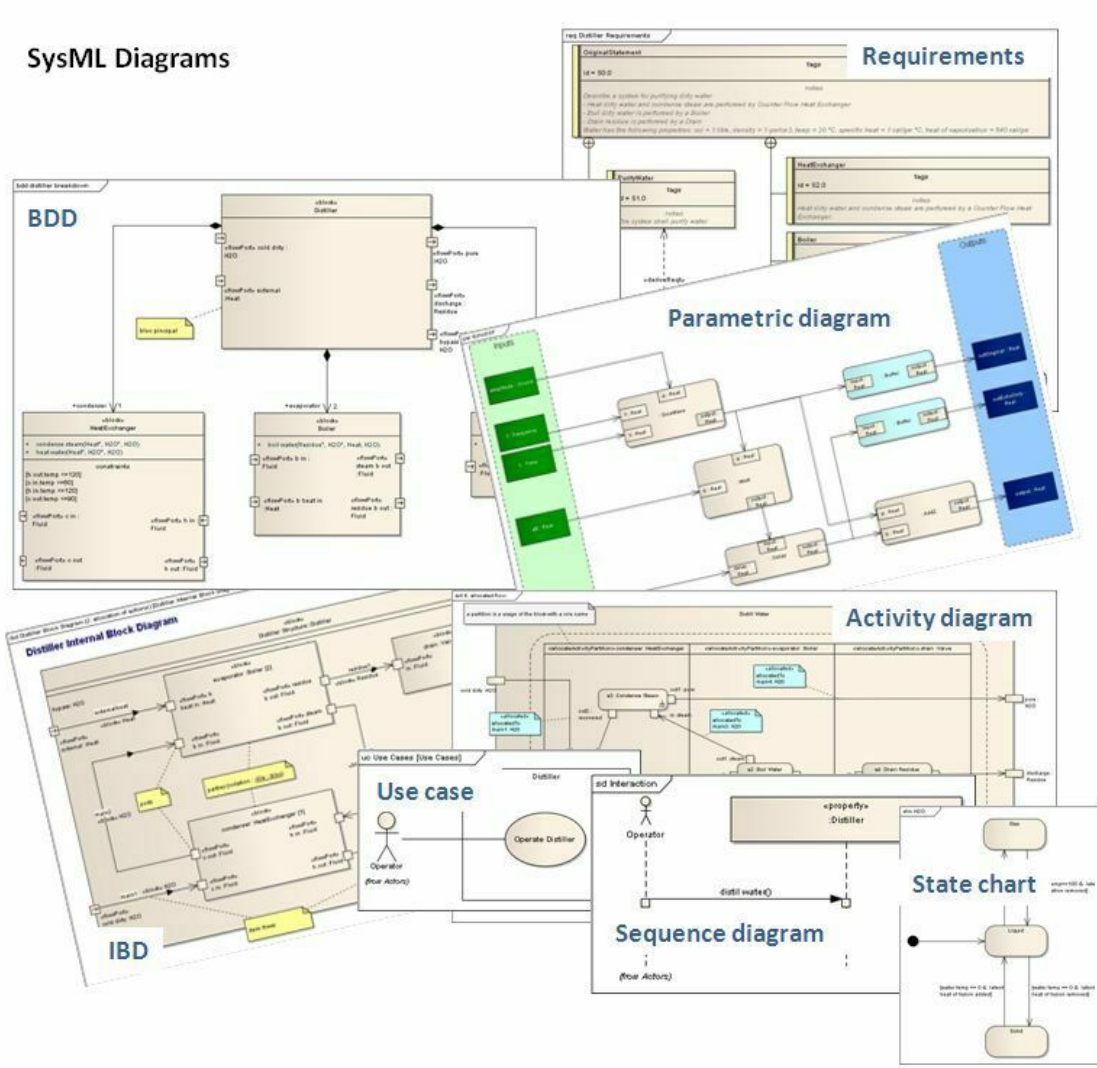
L'UML est un langage de modélisation graphique supportant une méthode normalisée utilisée dans les domaines du développement logiciel et en conception orientée objet.



SysML

≈ Systems Modeling Language

Le SysML 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. Issu de l'UML, il est plus adapté à la conception de systèmes multidisciplinaires et est également normalisé par l'Object Management Group (OMG) [<https://www.omg.org/>].



Attention Caractéristiques des modèles

Lorsque l'on veut parler d'un modèle, on doit décrire ses caractéristiques. Voici quelques unes des caractéristiques couramment utilisées :

- Statique
- Dynamique
- Formel
- Informel
- Abstrait
- Concret
- Graphique

Modèle statique

Permet de décrire la tructure d'un système

**Modèle dynamique**

Permet de décrire l'évolution du système au cours du temps

**Modèle formel**

Un modèle est dit formel si la syntaxe et la sémantique sont définies mathématiquement

**Modèle informel**

Un modèle est dit informel si la syntaxe ou la sémantique sont peu ou mal définies

**Modèle abstrait**

Un modèle abstrait permet la représentation des fonctionnalités du système

**Modèle concret**

Un modèle concret représente des éléments et composants du système

**Modèle graphique**

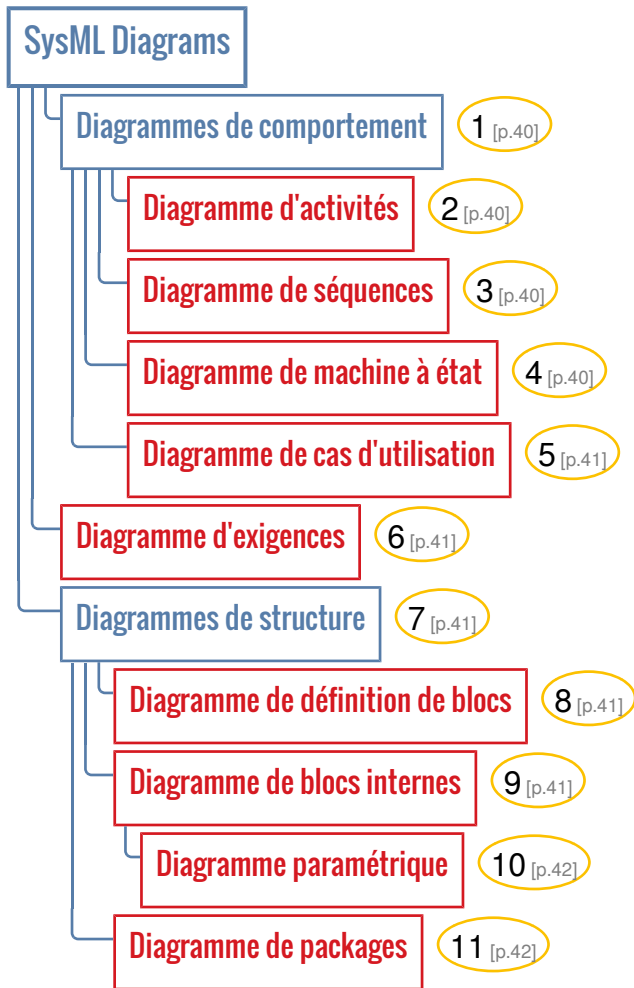
Un modèle graphique utilise des éléments graphiques (boîtes et flèches !) pour faire passer le message souhaité. Il s'oppose souvent au modèle textuel.

4.5. Systems Modeling Language

Les informations présentes sur cette page sont issues du Object Management Group - Systems Modeling Language (SysML)^[p.70], de la page SysML.org^[p.70]

Le SysML est un langage graphique dédié au Model-Based System Engineering. Il a été pensé pour les ingénieurs systèmes et est constitué de 3 grandes familles de diagrammes :

- les diagrammes de structure
- les diagrammes de comportement
- les diagrammes d'exigence



1 Diagrammes de comportement

≈ *behavior diagrams*

Les diagrammes de comportement comprennent le diagramme de cas d'utilisation, le diagramme d'activité, le diagramme de séquence et le diagramme de machine à états.

2 Diagramme d'activités

≈ *activity diagram*

Le diagramme d'activités représente le flux de données et de contrôle entre les activités.

3 Diagramme de séquences

≈ *sequence diagram*

Le diagramme de séquences représente l'interaction entre les parties collaboratrices d'un système.

4 Diagramme de machine à état

≈ *state machine diagram*

Le diagramme de machine à état décrit les transitions d'état et les actions qu'un système ou ses parties effectuent en réponse à des événements.

5 Diagramme de cas d'utilisation

≈ UC

≈ use case

Un diagramme de cas d'utilisation fournit une description de haut niveau de la fonctionnalité qui est réalisée par l'interaction entre les systèmes ou les parties du système.

Les diagrammes de cas d'utilisation intègrent des relations standardisées :

- <<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 »
- <<spécialisation>> : il permet de définir la spécialisation d'un cas d'utilisation

6 Diagramme d'exigences

≈ req

≈ requirement diagram

Les diagrammes d'exigences servent à préciser ce que l'on attend du système, généralement dérivées du cahier des charges.

7 Diagrammes de structure

≈ structure diagrams

Les diagrammes de structure servent à définir l'architecture du système.

Le bloc est l'unité de base de la structure en SysML et peut être utilisé pour représenter le matériel, le logiciel, les installations, le personnel ou tout autre élément du système. La structure du système est représentée par des diagrammes de définition de blocs et des diagrammes de blocs internes. Les packages servent à organiser le modèle.

8 Diagramme de définition de blocs

≈ bdd

≈ block definition diagram

Un diagramme de définition de blocs est un diagramme structurel statique.

L'objectif des diagrammes de définition de blocs est de spécifier les composants du système, leur contenu (propriétés, comportements, contraintes), leurs interfaces et leurs relations.

9 Diagramme de blocs internes

≈ ibd

≈ internal block diagram

Un diagramme de blocs internes est un diagramme structurel statique.

L'objectif des diagrammes de blocs internes (IBD) est de montrer le contenu structurel encapsulé (composants, propriétés, connecteurs, ports, interfaces) des blocs afin qu'ils puissent être décomposés de manière récursive et "reliés" à l'aide d'interfaces. Autrement dit, un IBD est une vue "boîte blanche" (je comprends et je décris ce qu'il y a à l'intérieur du système considéré) d'un bloc qui a jusqu'alors été décrit en "boîte noire" (je ne sais pas ce qui se passe à l'intérieur, je ne peux que constater son comportement).

10 Diagramme paramétrique

≈ *parametric diagram*

Le diagramme paramétrique représente les contraintes sur les valeurs des propriétés du système, telles que les performances, la fiabilité, les propriétés de masse, etc.

11 Diagramme de packages

≈ *pkg*

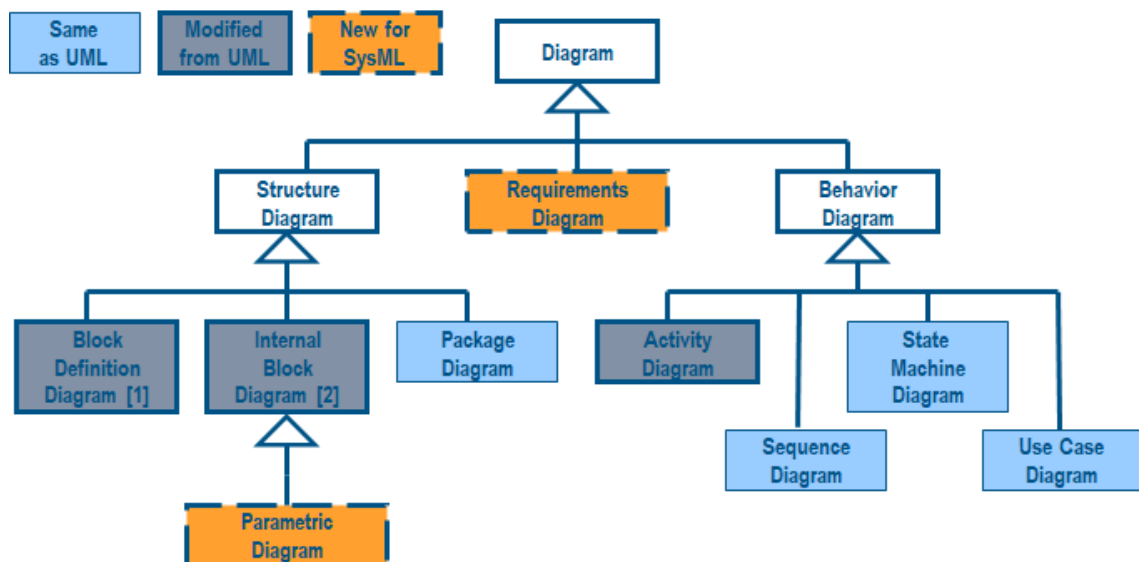
≈ *package diagram*

Un diagramme de packages est un diagramme structurel statique.

L'objectif d'un diagramme de packages est d'organiser les modèles complexes et de grande taille.

❖ Pour aller plus loin UML vs. SysML : quels points communs, quelles différences ?

Bien qu'il puisse être intéressant de connaître les concepts principaux de la modélisation UML (classe, attribut, méthode, association, cardinalité, héritage, composition et agrégation), le SysML a été pensé pour les ingénieurs systèmes, avec notamment une sémantique plus riche et flexible, et un nombre restreint de diagrammes.



UML vs. SysML

SysML réutilise plusieurs des principaux diagrammes UML. Dans certains cas, les diagrammes UML sont strictement réutilisés, tels que les diagrammes de cas d'utilisation, de séquence, de machine à états et de package, tandis que dans d'autres cas, ils sont modifiés afin de les mettre en cohérence avec les extensions SysML. Par exemple, le diagramme de définition de blocs et le diagramme de blocs internes sont respectivement similaires au diagramme de classe et au diagramme de structure composite UML, mais comportent des extensions ou des omissions. Les diagrammes d'activité ont également été modifiés.

SysML n'utilise pas tous les types de diagrammes UML tels que le diagramme d'objet, le diagramme de communication, le diagramme de synthèse des interactions, le diagramme de synchronisation et le diagramme de déploiement. Ceci est cohérent avec la logique selon laquelle SysML représente un sous-ensemble de UML. Dans le cas des diagrammes de déploiement, le déploiement du logiciel sur le matériel peut être représenté dans le diagramme de bloc interne de SysML. Dans le cas des

diagrammes d'interaction et de communication, on estime que les diagrammes de comportement SysML fournissent une couverture adéquate pour représenter le comportement sans qu'il soit nécessaire d'inclure ces types de diagrammes. Deux nouveaux types de diagrammes ont été ajoutés à SysML : le diagramme d'exigences et le diagramme paramétrique.

[SysML v1.0]

4.6. La notation SysML

4.6.1. Introduction

Objectifs pédagogiques

Connaître les rudiments de la notation SysML [p.68]

4.6.2. Les blocs



Bloc

≈ *block*

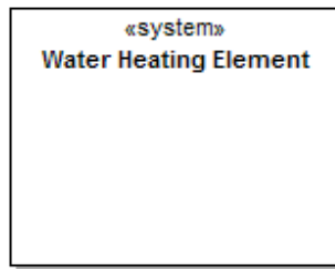
Un bloc est une unité modulaire pour décrire un système ou un élément dans un système. Il est représenté schématiquement par un rectangle avec un compartiment de nom stéréotypé par « block » ou parfois « system ».

Blocs SysML

Les blocs SysML peuvent être utilisés dans toutes les phases de spécification et de conception d'un système, et peuvent être appliqués à de nombreux types de systèmes différents. Ils permettent notamment de modéliser la décomposition logique ou physique d'un système, et de spécifier les éléments logiciels, matériels ou humains.

Stéréotype et type

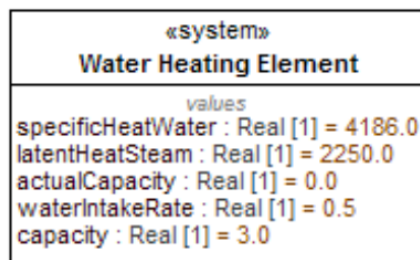
Un bloc définit un « type » réutilisable dans d'autres diagrammes. Ici, le type est « Water Heating Element ».



Un bloc est utilisé pour représenter une variété de types de systèmes/éléments, avec, par exemple :

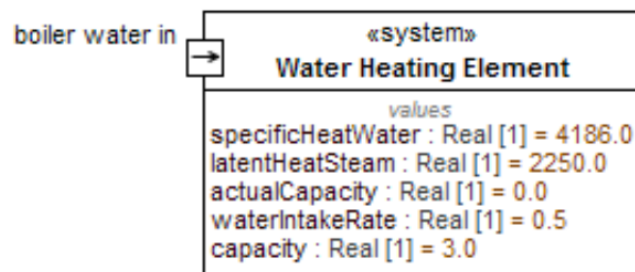
- Matériel
- Logiciel
- Données
- Procédure
- Installation
- Organisation

Ici, dans le compartiment inférieur, les caractéristiques attendues sont précisées.




Le bloc permet de définir un ensemble de caractéristiques qui décrivent le système / l'élément, avec, par exemple :

- les composants : structurels - ce dont il est composé
- les opérations : comportementales - ce qu'il peut faire
- les contraintes : sur tout aspect - ce qu'il doit respecter
- les exigences auxquelles le bloc répond - ce qu'il « satisfait »

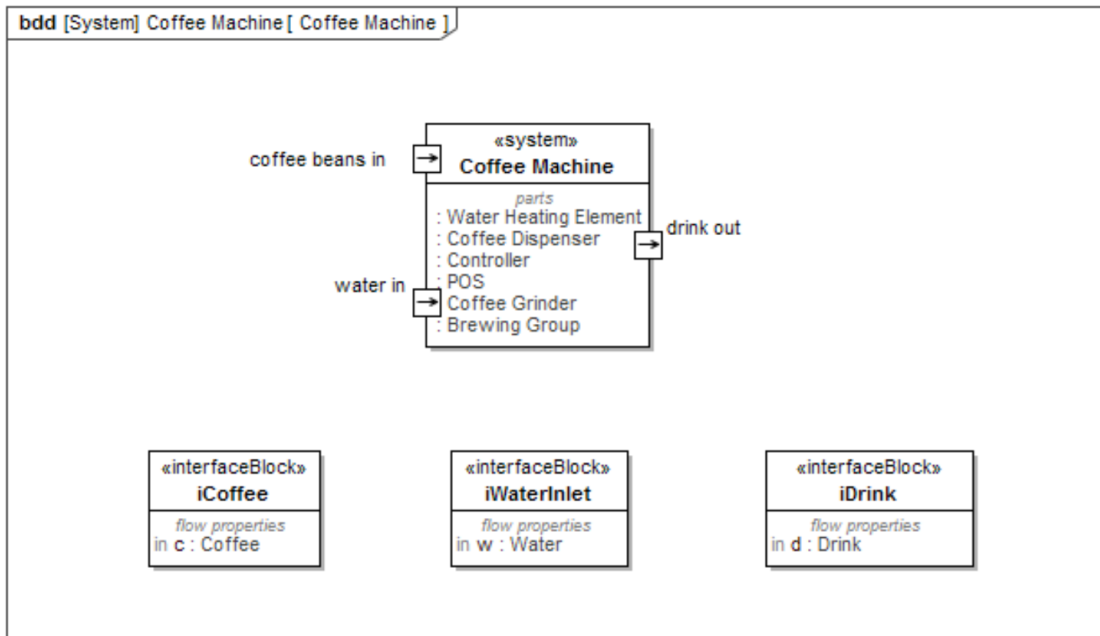


4.6.3. Diagramme de définition de blocs


4.6.3.1. BDD

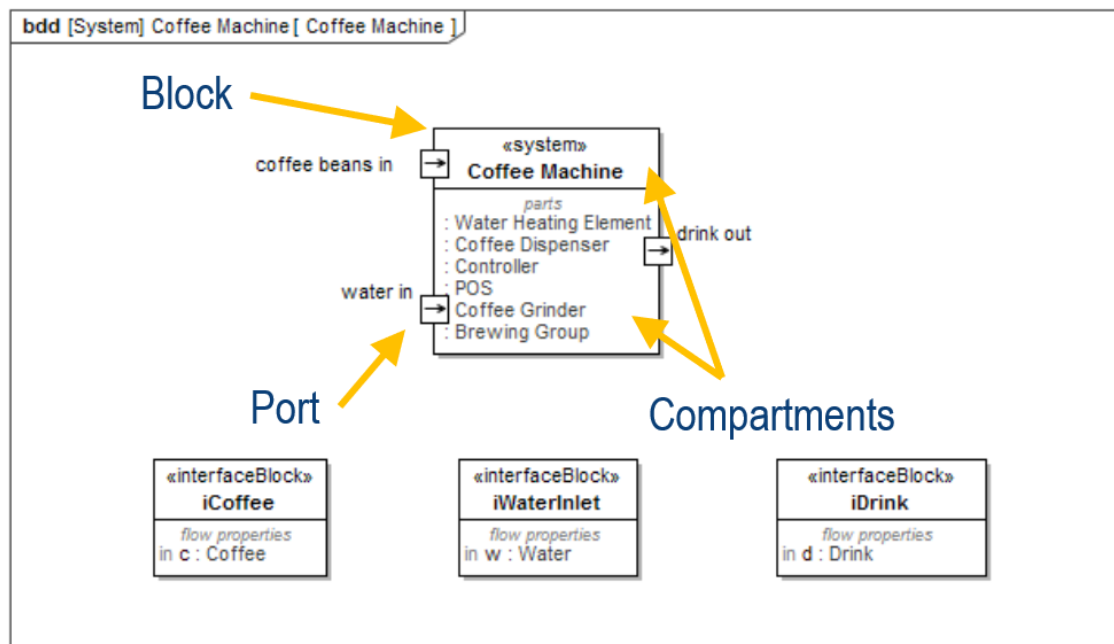
Le diagramme de définition de blocs s'appuie sur la notion de  bloc [p.43].

L'image ci-dessus présente un exemple de diagramme de définition de blocs, relativement simple, et ne faisant pas encore apparaître les relations entre les blocs.



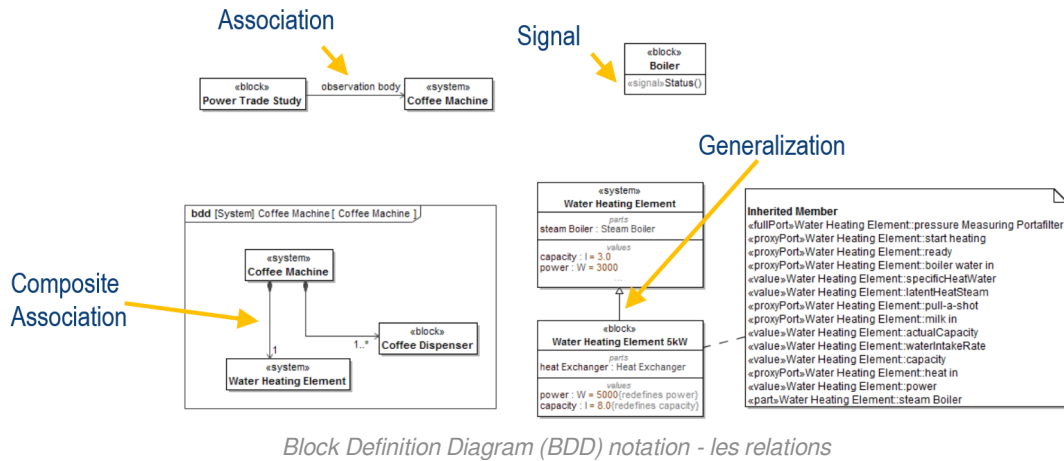
Block Definition Diagram (BDD) notation - exemple

Plus précisément, ces blocs sont composés de compartiments, qui présentent des propriétés de type composants, référence et valeur. L'ajout de port est possible, mais cet élément est présenté avec les  diagrammes de blocs internes.



Block Definition Diagram (BDD) notation - les compartiments

Les diagrammes de blocs comportent également des informations concernant les relations qui existent entre ces blocs. La figure ci-après illustre quelques types de relations réalisables.



Block Definition Diagram (BDD) notation - les relations

La relation de généralisation (flèche triangulaire creuse $\longrightarrow \triangleright$) indique que le bloc **Water Heating Element 5kW** hérite des caractéristiques du bloc **Water Heating Element**. Il s'agit donc d'un élément très proche de l'héritage UML.

La relation de composition (losange plein $\longrightarrow \blacklozenge$) permet de décomposer un élément en sous-éléments.

La relation d'agrégation (losange creux $\longrightarrow \diamond$), non présentée sur cette image, est un lien plus faible : il indique que le sous-élément n'est pas nécessairement présent.

La relation d'association, matérialisée par un simple trait (relation bidirectionnel) ou par un trait se terminant par une flèche (relation unidirectionnelle \longrightarrow), montre que deux éléments communiquent. Il n'y a alors pas de relation hiérarchique entre ces blocs.

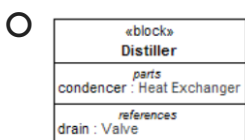
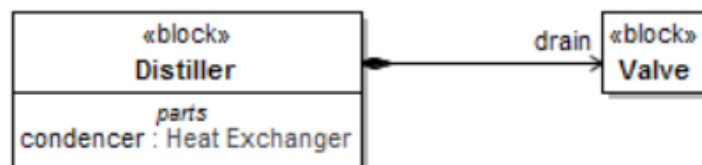
Les cardinalités (par exemple : $\blacklozenge_{0..1}$) indiquent le nombre de pièces.

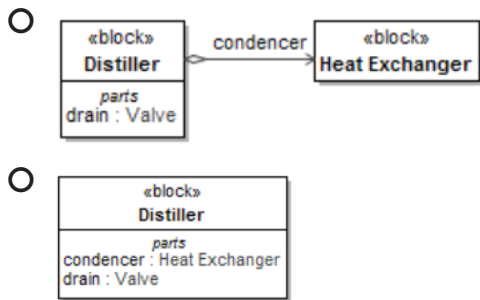
Notez qu'un bloc peut avoir une association avec un acteur, indiquant une certaine forme d'interaction entre les deux.

4.6.3.2. Exercice

Temps estimé : 3 min

Parmi les images suivantes, laquelle représente la même structure ?

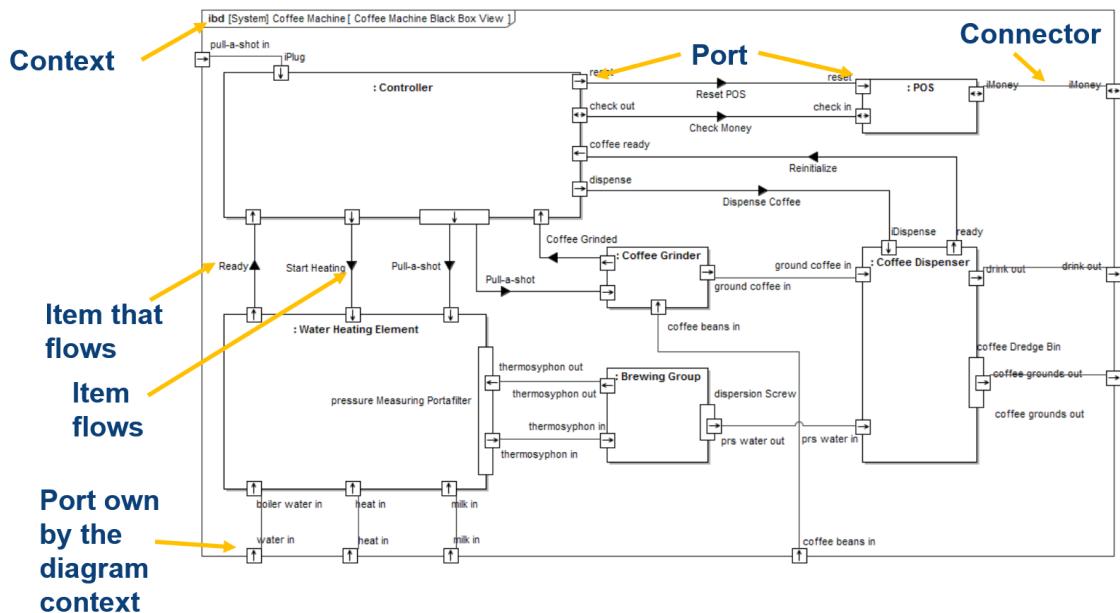




4.6.4. Diagramme de blocs internes

4.6.4.1. IBD

Le diagramme de blocs internes s'appuie sur la notion de bloc [p.43].



Internal Block Diagram (IBD) notation

Notez la cohérence entre ce diagramme et le diagramme BDD [p.45] (noms des parties, types et cardinalités).

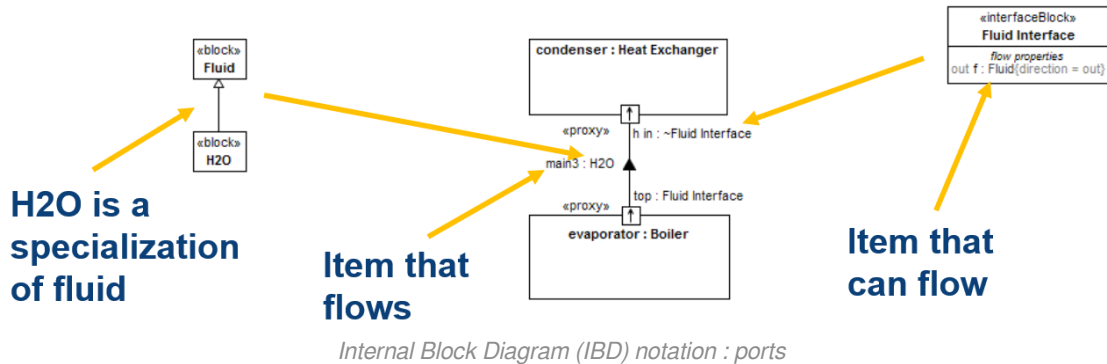
Ce diagramme permet de décrire le fonctionnement interne du système, d'un point de vue qui reste statique. Il introduit la notion de port (avec le nom et le type), de flux (nature du flux et direction) et la notion de connecteur.

Les noms de port peuvent être affichés n'importe où à côté du port (donc à l'intérieur ou à l'extérieur de l'élément qui possède le port) et peuvent optionnellement montrer le nom ou le type ou les deux.

Le rôle des ports

Les ports sont des points d'interaction entre les blocs.

Ils spécifient ce qui **peut** entrer ou sortir d'un élément, à travers le type du flux. Les ports peuvent laisser passer des flux qui peuvent être des données, des matériaux, des fluides ou de l'énergie. Les ports sont souvent utilisés pour modéliser des connexions physiques.



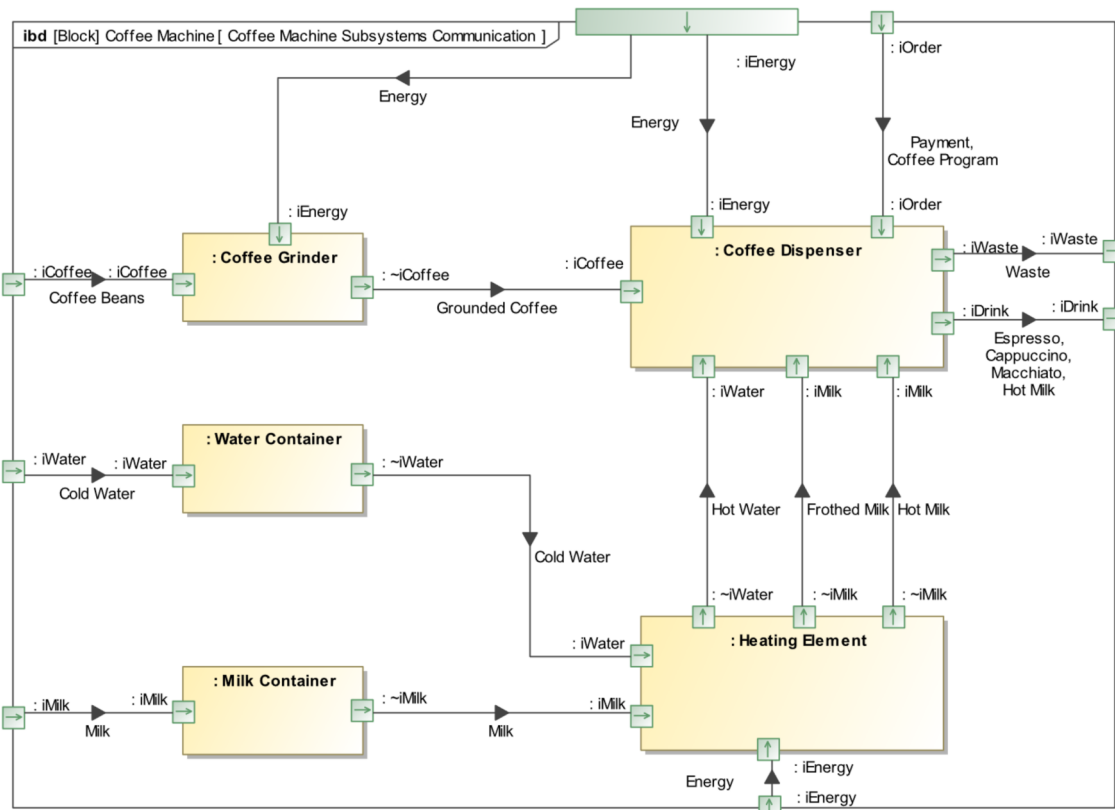
Exemple et exploitations des ports

L'intérêt d'un modèle logique IBD est de définir les flux conceptuels entre les différents blocs internes d'un bloc plus général.

L'intention première est de documenter ce qui **doit circuler** entre les parties, plutôt que de documenter **comment** cela circule.

Les types de flux, lorsqu'ils sont indiqués, sont généralement à un niveau d'abstraction qui permet d'identifier simplement les grandes catégories de types de flux.

L'analyse des flux entre les blocs permet parfois d'aboutir à la création d'opérations de "traitement des flux" pour ces blocs. Ces opérations peuvent ensuite être utilisées dans la modélisation du comportement du système.



Internal Block Diagram (IBD) example

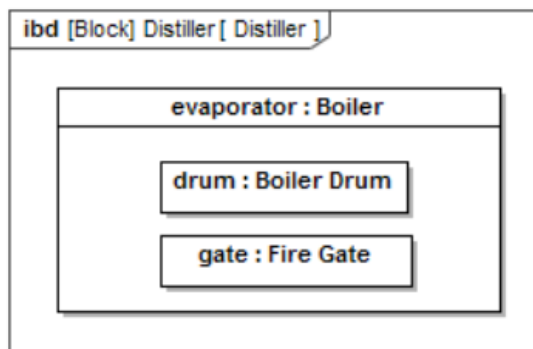
Par la suite, selon les capacités du logiciel de modélisation utilisé, il est possible d'extraire l'ensemble des ports d'un modèle sous la forme d'un tableau.

Criteria							
Element Type: Connector		Context: Coffee Machine		Filter: ▼			
#	Part A	Port A	Port A Features	Item Flow	Port B	Port B Features	Part B
1	Coffee Grinder	out p2: ~Coffee	out c: Coffee	Grounded Coffee	in p2: iCoffee	in c: Coffee	Coffee Dispenser
2	Heating Element	out p4: ~Water	out w: Water	Hot Water	in p4: iWater	in w: Water	Coffee Dispenser
3	Heating Element	out p7: ~Milk	out m: Milk	Frothed Milk	in p7: iMilk	in m: Milk	Coffee Dispenser
4	Heating Element	out p9: ~Milk	out m: Milk	Hot Milk	in p9: iMilk	in m: Milk	Coffee Dispenser
5	Milk Container	out p6: ~Milk	out m: Milk	Milk	in p6: iMilk	in m: Milk	Heating Element
6	Water Container	out p3: ~Water	out w: Water	Cold Water	in p3: iWater	in w: Water	Heating Element
7	Coffee Machine	in p1: iCoffee	in c: Coffee	Coffee Beans	in p1: iCoffee	in c: Coffee	Coffee Grinder
8	Coffee Machine	in p2: iWater	in w: Water	Cold Water	in p2: iWater	in w: Water	Water Container
9	Coffee Machine	out p3: iWaste	out w: Waste	Waste	out p3: iWaste	out w: Waste	Coffee Dispenser
10	Coffee Machine	out p4: iDrink	out d: Drink	Espresso Cappuccino Macchiato Hot Milk	out p4: iDrink	out d: Drink	Coffee Dispenser
11	Coffee Machine	in p5: iMilk	in m: Milk	Milk	in p5: iMilk	in m: Milk	Milk Container
12	Coffee Machine	in p6: iEnergy	in e: Energy	Energy	in p6: iEnergy	in e: Energy	Coffee Dispenser
13	Coffee Machine	in p6: iEnergy	in e: Energy	Energy	in p6: iEnergy	in e: Energy	Coffee Grinder
14	Coffee Machine	in p7: iOrder	in cp: Coffee Program in p: Payment	Payment Coffee Program	in p7: iOrder	in cp: Coffee Program in p: Payment	Coffee Dispenser
15	Coffee Machine	in p8: iEnergy	in e: Energy	Energy	in p8: iEnergy	in e: Energy	Heating Element

Internal Block Diagram (IBD) notation : synthèse des ports

4.6.4.2. Exercice

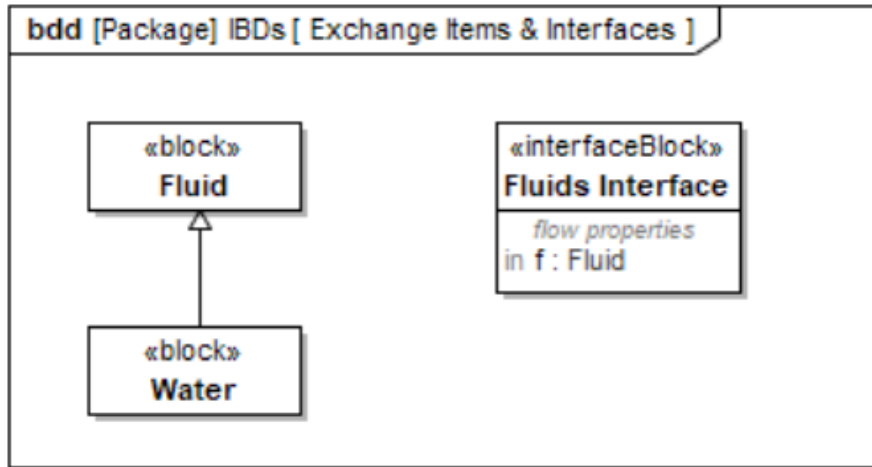
Lequel des éléments suivants est le bloc de contexte pour le diagramme fourni ?



- «block» Boiler
- «block» Boiler Drum
- «block» Distiller
- «block» Fire Gate

4.6.4.3. Exercice

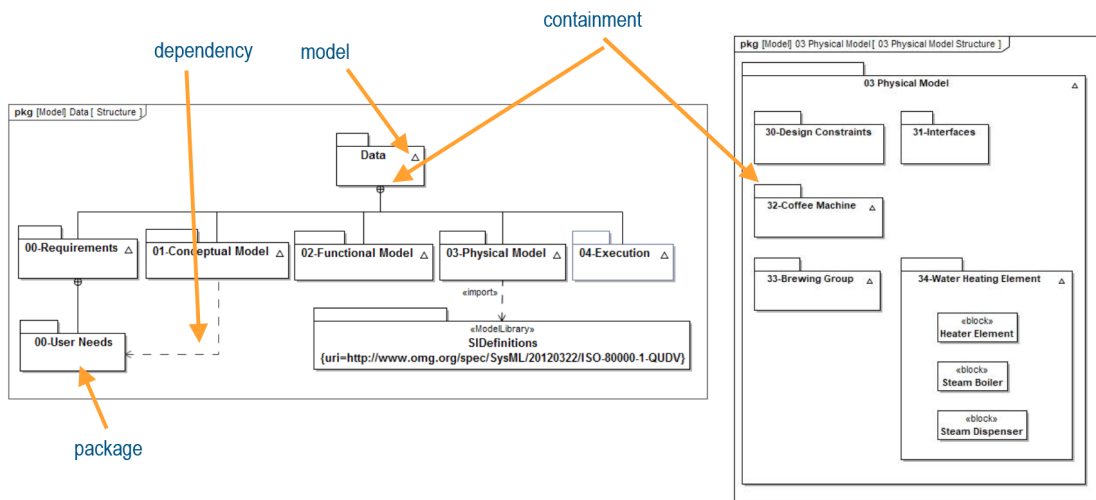
Lesquels des IBD suivants sont corrects ?



- ibd [Block] System X [System X]**
- ibd [Block] System X [System X]**
- ibd [Block] System X [System X]**
- ibd [Block] System X [System X]**

4.6.5. Diagramme de packages

4.6.5.1. PKG



Package Diagram (PKG) notation

Cet exemple montre les notations alternatives disponibles pour montrer la structure des packages, ainsi que pour montrer un exemple de dépendance.

Ce diagramme est susceptible de changer tout au long du projet, au fur et à mesure que la structure évolue.

Un package peut regrouper n'importe quel nombre et type d'éléments de modèle, de n'importe quelle manière appropriée aux besoins du projet.

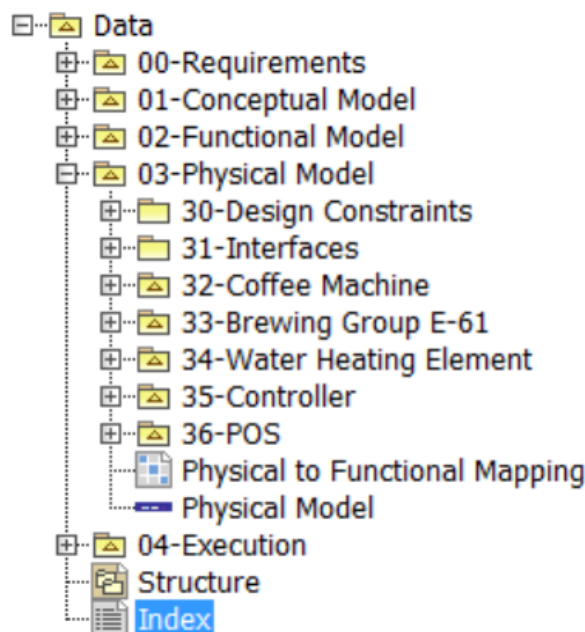
Il peut contenir d'autres paquets (imbriqués)

C'est un concept très similaire à celui des dossiers dans un système de fichiers.

Des dépendances implicites et explicites peuvent exister entre les packages.

Comme vu précédemment, la structure des packages et les dépendances explicites peuvent être représentées sur un diagramme de package.

La structure de ces packages a généralement un impact sur l'arborescence du modèle.



Un package peut regrouper n'importe quel nombre et type d'éléments de modèle, de n'importe quelle

manière appropriée aux besoins du projet.

Il peut contenir d'autres paquets (imbriqués).

C'est un concept très similaire à celui des dossiers dans un système de fichiers.

Des dépendances implicites et explicites peuvent exister entre les packages.

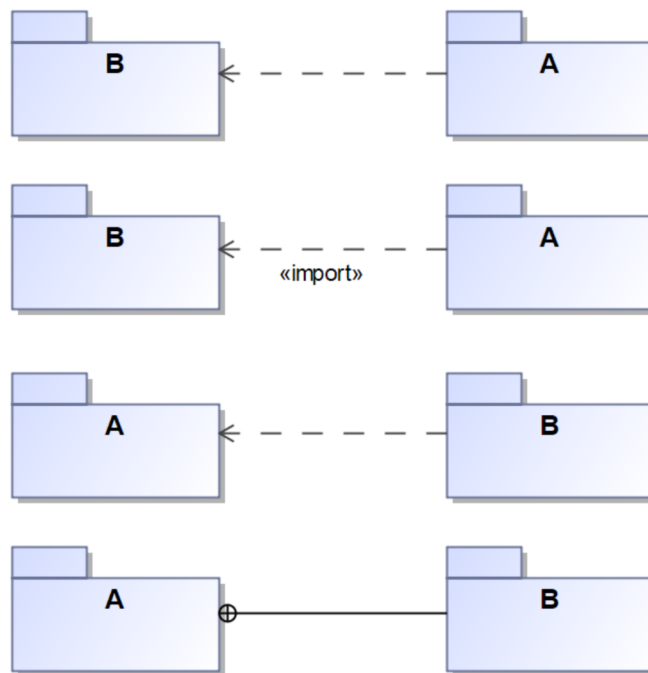
Comme vu précédemment, la structure des packages et les dépendances explicites peuvent être représentées sur un diagramme de package.

⚠ Attention

Le diagramme de package a un rôle fondamental pour la clarté du modèle et sa navigabilité.

4.6.5.2. Exercice

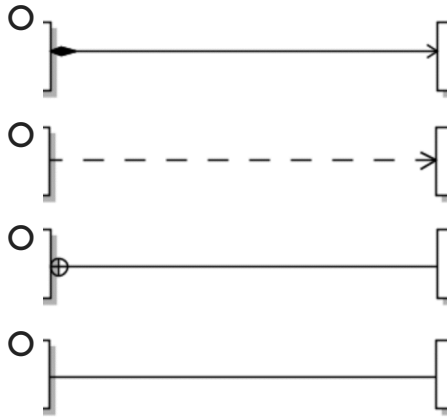
Lequel des énoncés suivants indique qu'un changement dans le paquet A peut avoir un impact sur les changements dans le paquet B ?



- La première ligne
- La seconde ligne
- La troisième ligne
- La quatrième ligne

4.6.5.3. Exercice

Parmi les types de liens suivants, lequel est utilisé pour établir la **structure** du modèle ?



4.6.5.4. Exercice

Lequel des éléments suivants affiche les mêmes informations ?



- A UML Use Case diagram showing Use Case A connected to Use Case B with an include relationship (solid line with open arrowhead and small circle at A). Use Case B is then connected to Block a with an include relationship (solid line with open arrowhead and small circle at B).
- A UML Use Case diagram showing Use Case A connected to Use Case B with an include relationship. Use Case B is connected to Block a with a realizes relationship (dashed line with open arrowhead).
- A UML Use Case diagram showing Use Case A connected to Use Case B with an include relationship. Use Case B contains Block a and is connected to Block b with an include relationship.
- A UML Use Case diagram showing Use Case A connected to Use Case B with an include relationship. Use Case B contains Block a and realizes it (dashed line with open arrowhead).

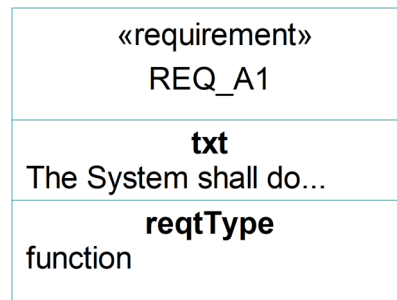
4.6.6. Diagramme d'exigences

4.6.6.1. REQ

Notation d'une exigence et informations associées

Une exigence "spécifie une capacité ou une condition qui doit (ou devrait) être satisfaite" [SysML].

Elle peut spécifier une fonction à réaliser ou des critères de performance à satisfaire.



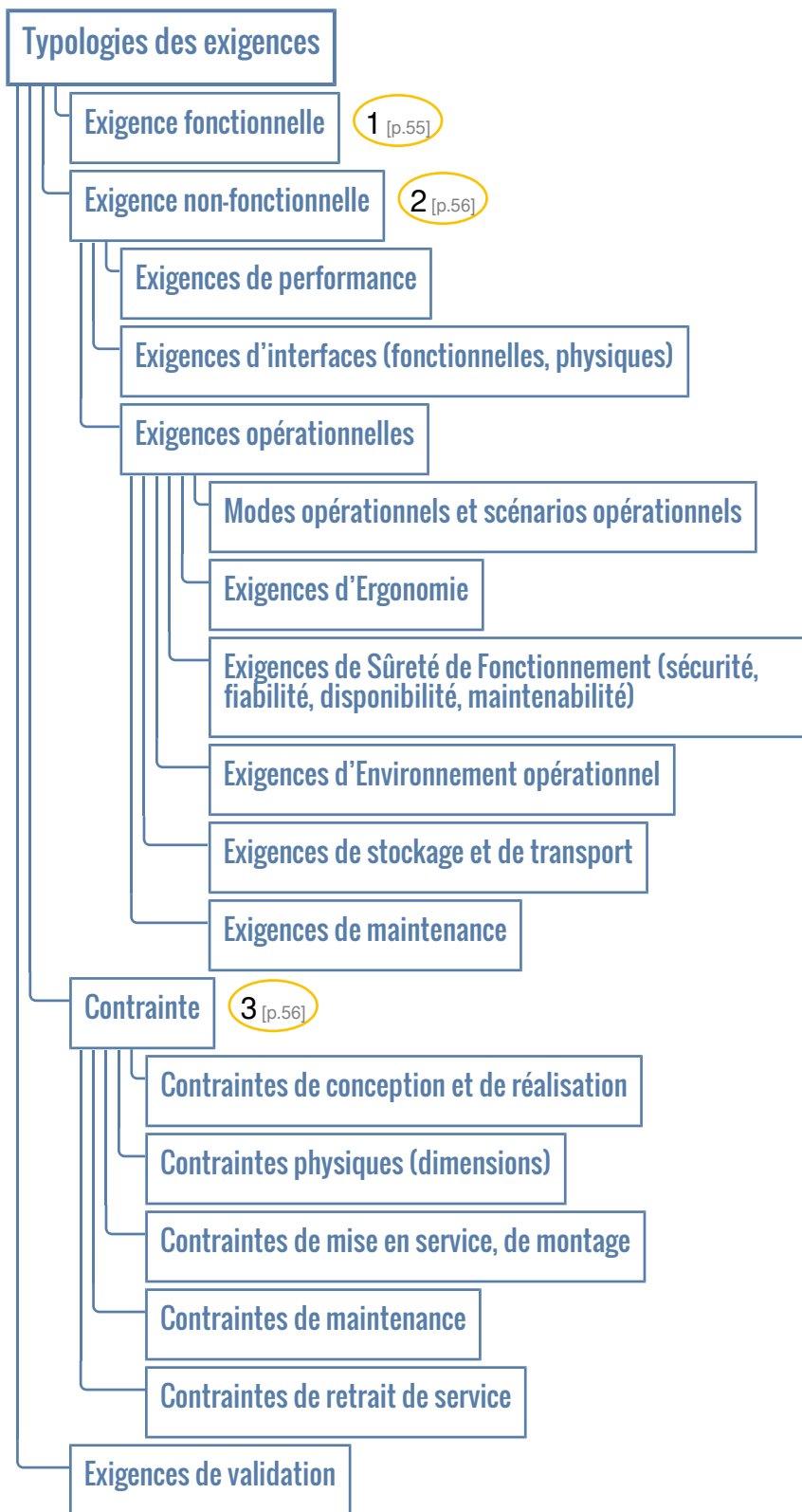
Requirement Diagram (REQ) notation

Cet exemple de notation illustre les attributs de base d'une exigence.

- ID : Un identifiant unique pour l'exigence
- Text : Une description textuelle de ce qui est requis.

Les utilisateurs peuvent créer des stéréotypes et des balises pour ajouter des propriétés spécifiques, par exemple :

- type d'exigence (performance, sécurité, fonctionnel, etc.)
- la priorité (par exemple : haute, moyenne, basse)
- la source ou l'origine (par exemple : client, marketing, technique, législation, etc.)
- le risque associé à cette exigence (par exemple : haut, moyen, bas) - concept souvent lié à la criticité
- le statut (par exemple : proposée, validée, implémentée, testée, livrée, etc.) - ce dernier est souvent précisé en cours de projet
- la méthode de vérification (par exemple : analyse, démonstration, test, etc.)



1 Exigence fonctionnelle

Une exigence fonctionnelle est une exigence définissant une fonction du système à développer. Ce que le système doit faire.

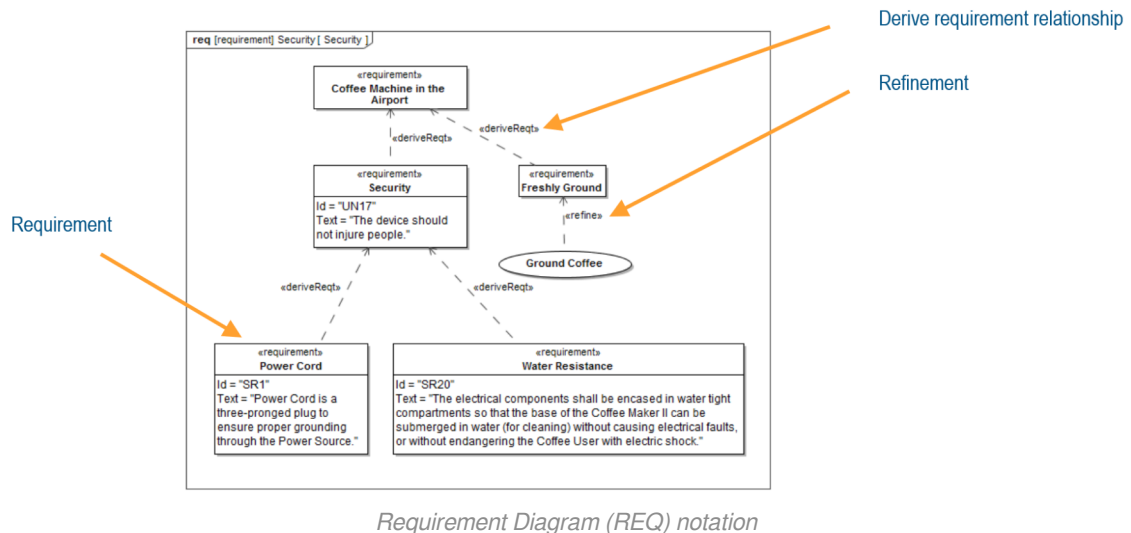
2 Exigence non-fonctionnelle

Une exigence non-fonctionnelle est une exigence qui caractérise une propriété (qualité) désirée du système telle que sa performance, sa robustesse, sa convivialité, sa maintenabilité, etc.

3 Contrainte

Une contrainte est une restriction sur une ou plusieurs valeurs d'une partie du système ou de tout le système.

Relations entre exigences



Dans l'exemple ci-dessus, deux types de relation importants sont à noter :

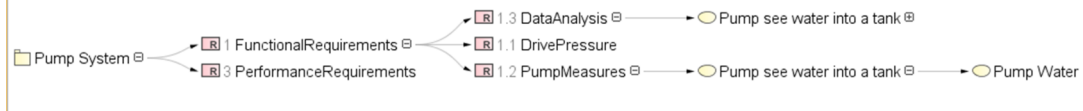
- Une relation « dérivée » (derive) est une dépendance entre deux exigences (une exigence dérivée et une exigence source), où l'exigence dérivée est générée ou déduite de l'exigence source.
- Une relation « de raffinement » (refine) est une dépendance qui décrit comment un élément de modèle ou un ensemble d'éléments raffine une exigence. Par exemple, ici, le cas d'utilisation « moulin à café » vient raffiner l'exigence fonctionnelle de café fraîchement moulu.

Un dernier type de lien entre exigences est également utilisé. Il s'agit du lien de contenance et est représentée par une ligne terminée par un cercle contenant une croix du côté du conteneur —⊕—. Ce lien permet de décomposer une exigence composite en plusieurs exigences unitaires.

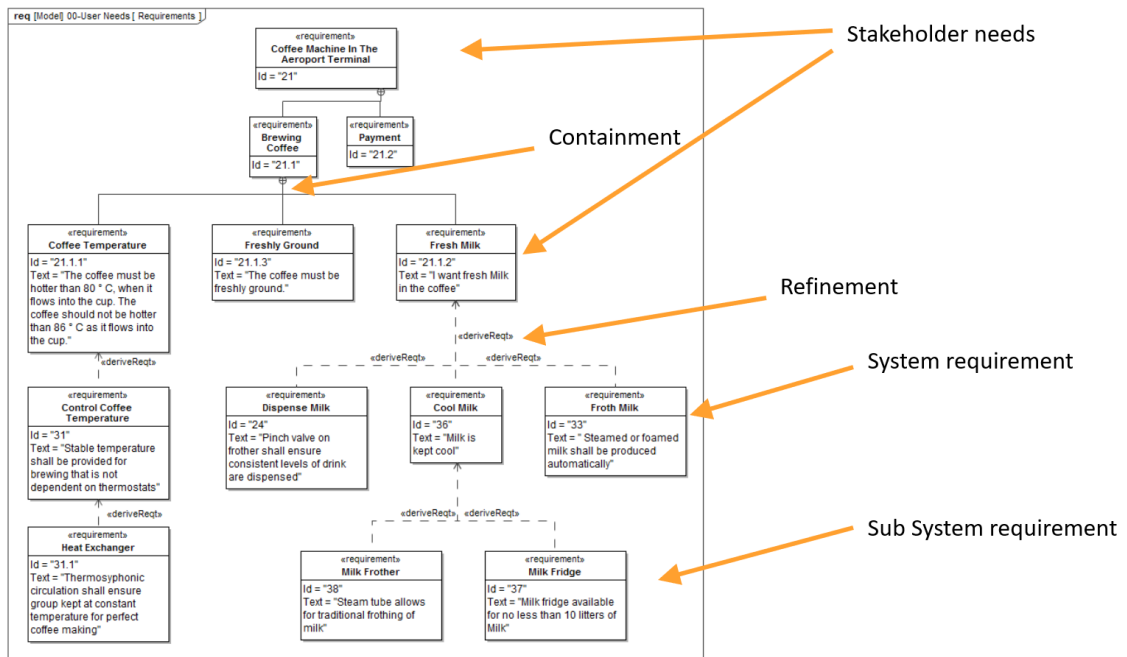
Exploitation des exigences

L'exploitation des exigences peut ensuite se faire sous une forme arborescente ou sous la forme d'un tableau.

#	Id	Name	Text
1	UN1	Coffee Machine in the A...	Mission Statement: Provide several types of coffee (with different ingredients) after
2	UN2	Fresh Milk	I want fresh milk.
3	UN3	Footprint	The footprint must be less than 1.5 square meters.
4	UN4	Hygiene Requirements	The device must meet the health requirements.
5	UN5	Efficient Work	The devices must work efficiently after 1 year.
6	UN6	Failure Rate	The failure rate can not be higher than 1 time a month for max. 1 hours.
7	UN7	Money	The money must be kept safe.
8	UN8	Debit Card	I would like to pay by debit card.
9	UN9	Credit Card	I would like to pay by credit card.
10	UN10	Bills	The device must take bills up to 10 Euros.
11	UN11	Usability	The operation must be for all travelers and employees quick and easy to use.
12	UN12	UHT Milk	I want to UHT milk.
13	UN13	Freshly Ground	The coffee must be freshly ground.
14	UN14	Front Design	The front design is in a coffee brown tone with the images.
15	UN15	Coffee Temperature	The coffee must be hotter than 80 ° C when it flows into the cup. The coffee should not be
16	UN16	Machine Maintenance	The machine should be filled and serviced only once a day. Maintenance should not take
17	UN17	Security	The device should not injure people.
18	UN18	Coffee Cup	I want to bring my own coffee cup (0.25 l).



Imbrication des exigences entre elles et niveau de détails attendus



Requirement Diagram (REQ) example

L'exemple ci-dessus illustre les différents niveaux de détails qu'il est parfois nécessaire de préciser pour aller du plus général (attentes des différentes parties prenantes) aux exigences liées aux sous-systèmes, en passant par les exigences liées au système à concevoir.

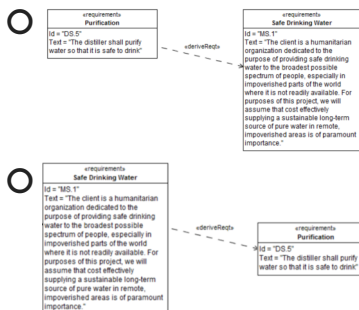
Il est courant de définir d'autres propriétés pour les exigences, par exemple :

- la priorité (par exemple : haute, moyenne, basse)
- la source ou l'origine (par exemple : client, marketing, technique, législation, etc.)
- le risque associé à cette exigence (par exemple : haut, moyen, bas)
- le statut (par exemple : proposée, validée, implémentée, testée, livrée, etc.) - ce dernier est souvent précisé en cours de projet

- la méthode de vérification (par exemple : analyse, démonstration, test, etc.)

4.6.6.2. Exercice

Lequel des diagrammes suivants représente une bonne utilisation de la dérivation ?



4.6.7. Diagramme de cas d'utilisation

4.6.7.1. UC

Un cas d'utilisation ou use case permet de décrire la capacité ou le service rendu par un système.

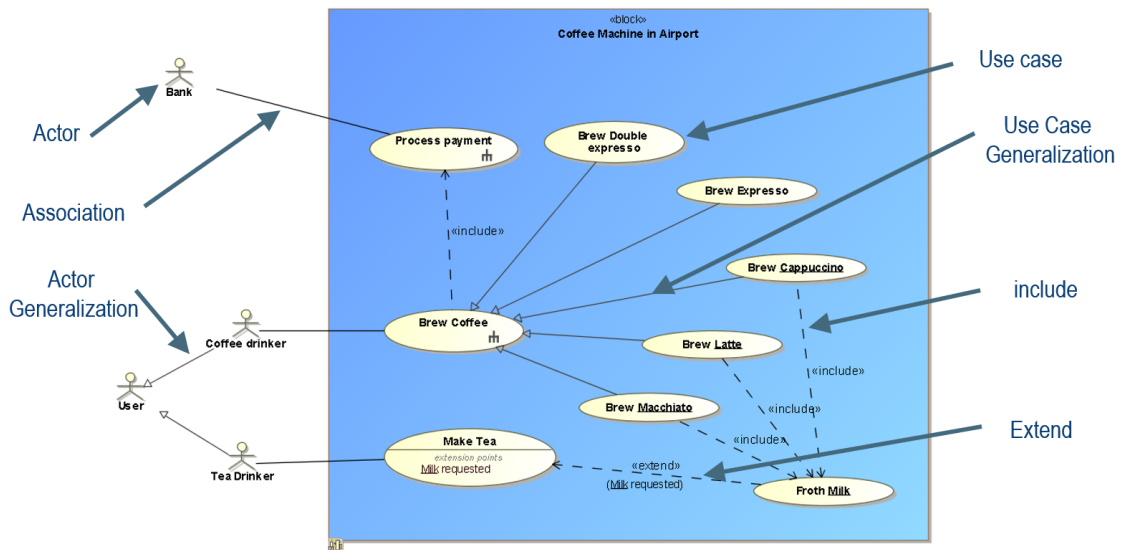
Le cas d'utilisation définit un but ou un objectif par rapport à un point de vue utilisateur externe (interacteur) du système.

Les cas d'utilisation peuvent aussi constituer la base d'un test réalisable au niveau du système.

Ainsi, de manière simpliste, un test d'acceptation du système peut correspondre au fait de tester tous les cas d'utilisation décrits.

Pour l'ingénieur système, la modélisation des cas d'utilisation constitue un moyen pratique de discuter des capacités avec les parties prenantes du système.

Les concepts et la modélisation des cas d'utilisation sont hérités d'UML.



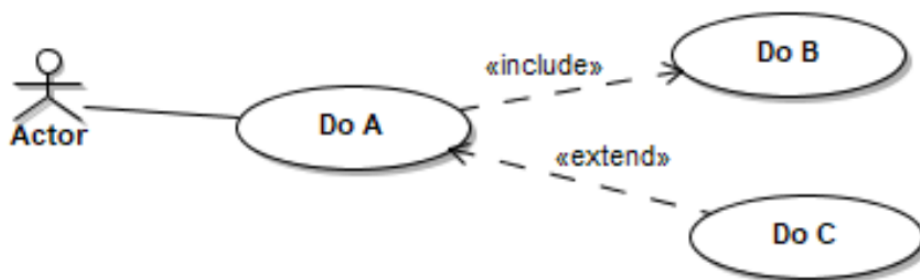
Dans l'exemple ci-dessus, on retrouve, on retrouve différents liens entre les cas d'utilisation :

- l'inclusion : il est nécessaire de faire mousser le lait pour faire des Macchiato, Cappuccino ou Latte
- l'extension fournit une fonctionnalité facultative (dans le sens où elle n'est pas nécessaire pour atteindre le but du use case). Ici, le fait de faire de la vapeur d'eau pour faire mousser le lait permet aussi de faire du thé.
- la généralisation : faire un café peut être précisé par des cas d'utilisation comme faire un expresso, faire un double expresso, etc. Toutes les étapes proposées dans le cas d'utilisation faire un café seront également valables pour les cas d'utilisation qui le précisent.

On retrouve également des liens avec les acteurs, extérieurs au système, qui peuvent être associés (simple trait) et qui peuvent eux également avoir des liens entre eux.

4.6.7.2. Exercice

Laquelle des propositions suivantes reflète ce cas d'utilisation ?



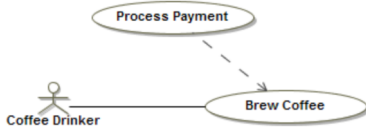
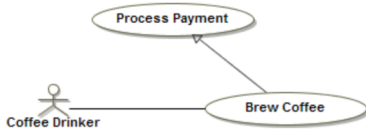
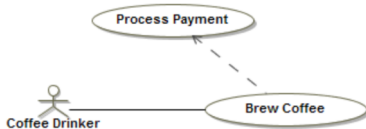
- Lorsque le cas d'utilisation A s'exécute, le cas d'utilisation B s'exécute et le cas d'utilisation C peut aussi s'exécuter.
- Lorsque le cas d'utilisation A est exécutée, le cas d'utilisation B et la tâche du cas d'utilisation C le sont également.
- Lorsque le cas d'utilisation A est exécutée, le cas d'utilisation B ou le cas d'utilisation C peut

également être exécutée.

- Lorsque le cas d'utilisation A est exécutée, le cas d'utilisation B peut être exécutée et le cas d'utilisation C le sera également.

4.6.7.3. Exercice

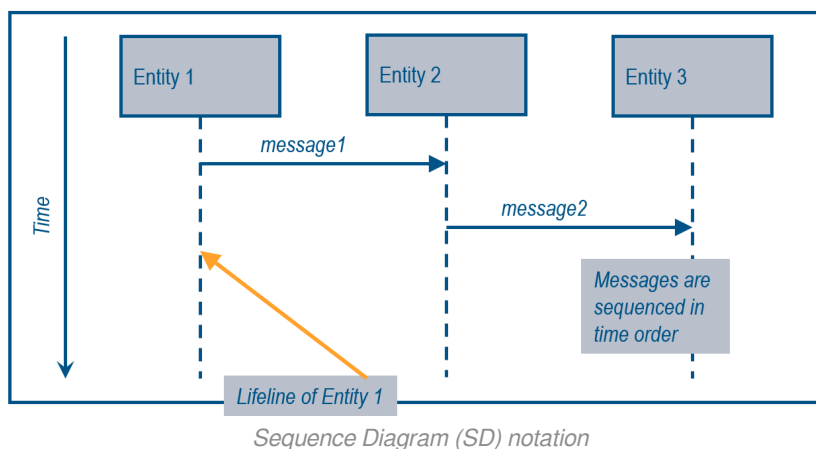
Parmi les 3 diagrammes suivants, lequel illustre le fait que l'inclusion du cas d'utilisation « processus de paiement » dépend de l'existence du cas d'utilisation de « faire un café » ?

- 
- 
- 

4.6.8. Diagramme de séquence

4.6.8.1. SD

Un diagramme de séquence décrit la séquence des échanges de messages (par exemple, les flux d'éléments) entre les lignes de vie des entités.



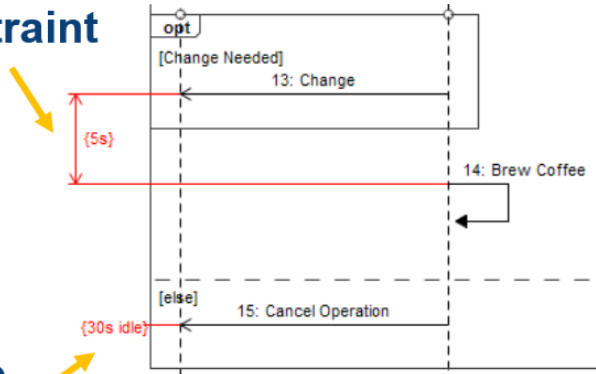
Les entités peuvent, par exemple, collaborer pour réaliser / préciser un cas d'utilisation.

Ce type de diagramme spécifie simplement l'ordre des messages, il ne modélise pas la durée relative

entre les messages.

Les messages envoyés entre les différentes entités peuvent être synchrones, asynchrones et des contraintes de temps peuvent être imposées.

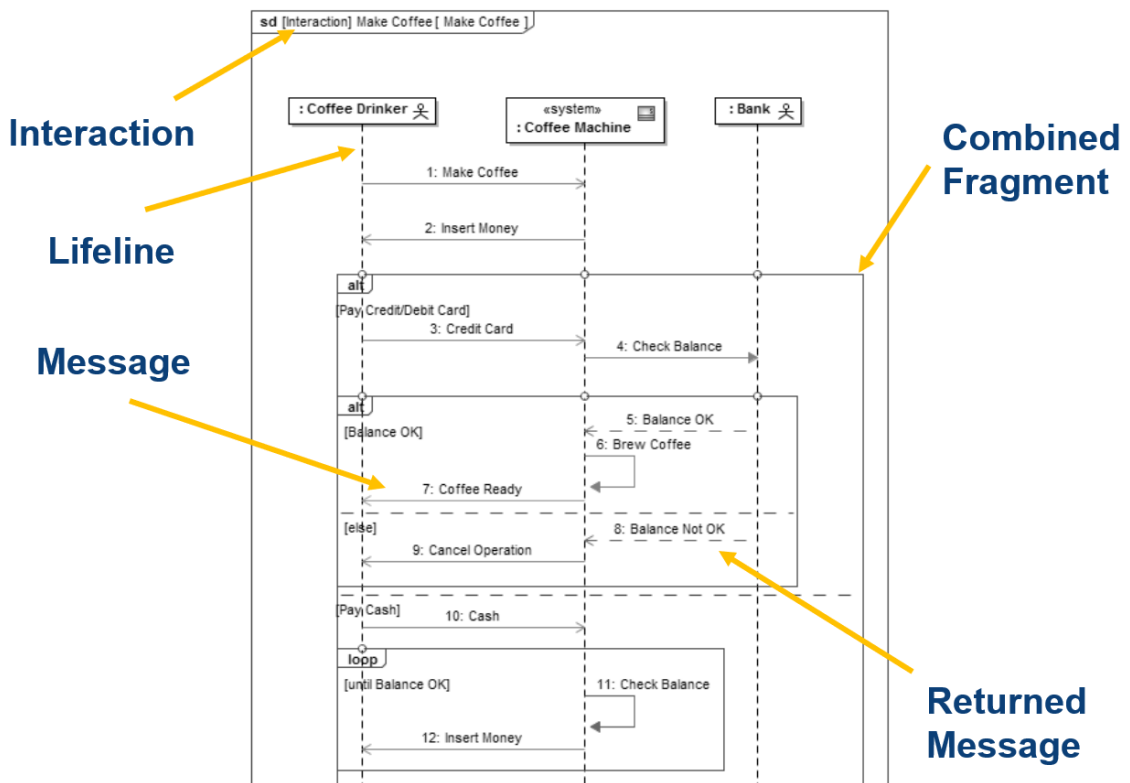
Duration Constraint



Time Constraint



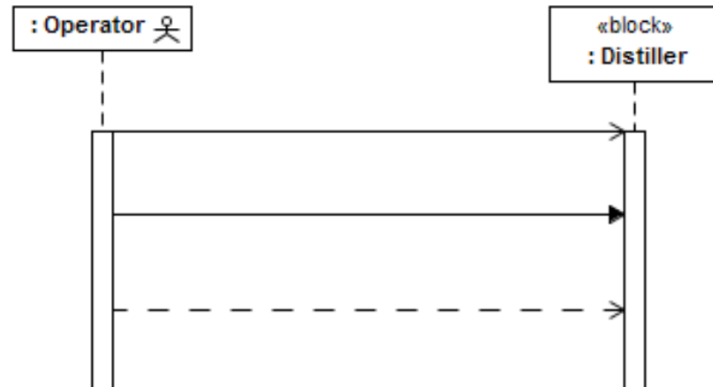
Sequence Diagram (SD) : message types



Sequence Diagram (SD) example

4.6.8.2. Exercice

Après l'envoi de quel message la ligne de vie attend-elle un message de réponse ?

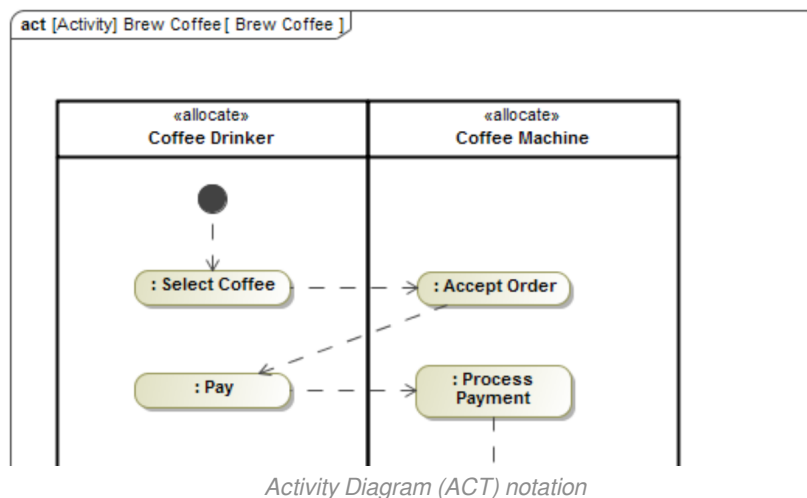


- 1 : trait plein avec une flèche à l'extrémité (asynchrone)
- 2 : trait plein avec une flèche pleine à l'extrémité (synchrone)
- 3 : trait pointillé avec flèche à l'extrémité (réponse)

4.6.9. Diagramme d'activité

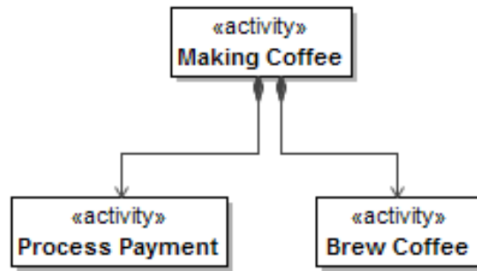
4.6.9.1. ACT

Un diagramme d'activité spécifie le comportement comme une séquence coordonnée d'actions. Le diagramme d'activité peut être utilisé à n'importe quelle phase du développement d'un système lorsque l'on souhaite obtenir une vue procédurale d'un processus.

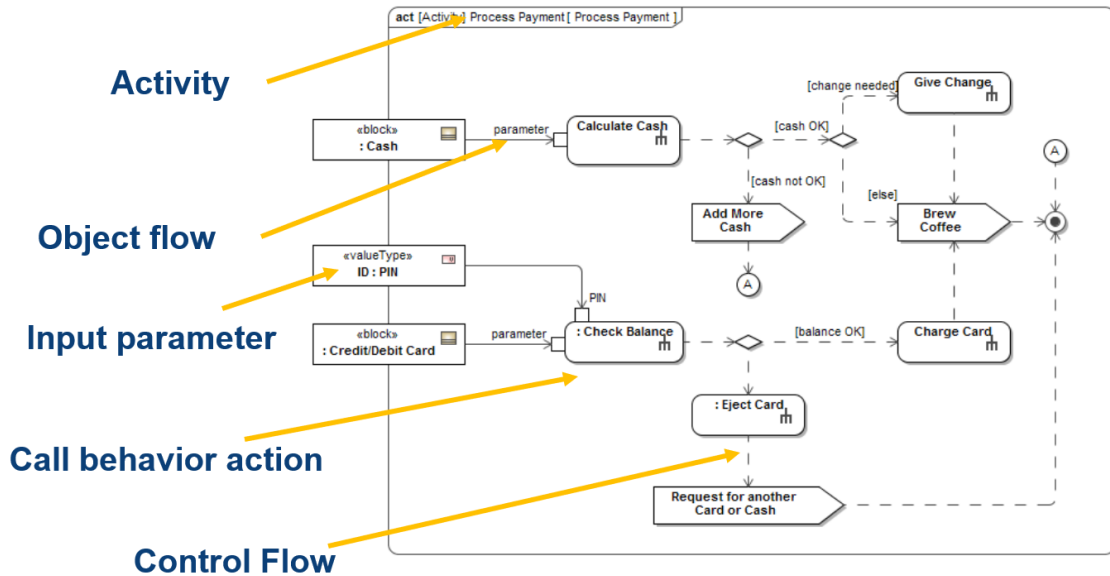


Dans cet exemple, l'activité préparer un café est composée de l'activité processus de

paiement et de l'activité ☑ faire le café.



On retrouve ainsi le point de départ (rond noir) et d'arrivée de l'activité (rond blanc et noir).



4.6.9.2. Exercice

Parmi les 4 diagrammes suivants, quelle est la bonne façon d'afficher les flux d'objets dans SysML ?

-
-
-
-

4.6.9.3. Diagramme de séquence vs. d'activité

Séquence	Activité
Permet de : <ul style="list-style-type: none"> ● voir comment les blocs collaborent dans un contexte particulier ● Modéliser la séquence d'événements dans un cas d'utilisation ● Modéliser la séquence d'événements dans un cas de test 	Permet de : <ul style="list-style-type: none"> ● Modéliser le flux d'action dans un cas d'utilisation ● Modéliser l'implémentation d'une méthode complexe, e.g. un algorithme ● Modéliser le comportement d'un bloc ; c'est une alternative au diagramme d'état ● Modéliser les effets de transition et les comportements d'entrée, de sortie et d'état

4.7. Synthèse

Dans le cadre de l'utilisation du MBSE, le modèle est le principal artefact ; il sert de référentiel pour stocker les éléments de définition du système à concevoir et les relations entre ces éléments.

Selon le logiciel utilisé, le principal avantage est le maintien de la cohérence lors des changements et la réduction des risques d'erreurs. Les protagonistes peuvent ainsi se concentrer sur l'architecture du système et non sur le maintien de la cohérence des documents décrivant le système.

Les différents diagrammes sont ensuite utilisés lors de étapes suivantes du processus de conception : les exigences sont un élément clé lié à la traçabilité, l'architecture du système peut servir à initier une nomenclature produit, etc.

Synthèse

Évaluation du module

Le module que vous avez étudié fait partie d'un projet en phase de test.

Nous vous invitons à compléter le formulaire d'évaluation dans une perspective d'amélioration continue.

Accès au formulaire :

- via l'URL du formulaire [\[https://forms.office.com/r/DPN2yFcKeH\]](https://forms.office.com/r/DPN2yFcKeH)

- via QR code :



Glossaire

Approche	Une approche peut être vue comme une philosophie . C'est un regroupement de principes permettant d'aborder de manière macroscopique un développement produit. La liste de principes peut être transposée sous la forme de processus, méthodes et outils, afin de l'opérationnaliser. L'approche se positionne au niveau stratégique de l'entreprise.	
Architecture	L'architecture est l'organisation fondamentale d'un système représenté par ses constituants, leurs interrelations, leurs relations avec l'environnement. L'architecture d'un système est une représentation, à un niveau d'abstraction et de granularité donné, d'un système sous forme d'une structure identifiant les éléments constitutifs du système et leurs interactions.	
Bloc ≈ <i>block</i>	Un bloc est une unité modulaire pour décrire un système ou un élément dans un système. Il est représenté schématiquement par un rectangle avec un compartiment de nom stéréotypé par « block » ou parfois « system ».	
BPMN ≈ <i>Business process model and notation</i>	Plus qu'un langage normalisé, le BPMN ^[https://fr.wikipedia.org/wiki/Business_process_model_and_notation] est une méthode de modélisation des processus d'affaires pour décrire les chaînes de valeur et les activités métier d'une organisation sous forme d'une représentation graphique.	
Comportement	Le comportement du système est l'ensemble des phénomènes organisés par lesquels le système, agit et réagit pr rapport à son environnement et aux parties prenantes.	
Cycle de vie d'un système	Le cycle de vie d'un système correspond à son évolution dans le temps. Il peut-être décrit par des états successifs. Cette notion est à rapprocher de la notion de cycle de vie du « Product Lifecycle Management ^{[<a 346="" 761="" 809"="" 89="" href="https://fr.wikipedia.org/wiki/Gestion_du_cycle_de_vie_(produit)]» », mais ne possède pas nécessairement la même couverture.</td> </tr> <tr> <td data-bbox=">Etat}	Un état correspond à un comportement caractéristique et temporaire du système à un instant d'observation donné.
Finalité	La finalité est le caractère de ce qui tend vers un but.	
Fonctionnement	Le fonctionnement d'un système est sa manière de remplir sa fonction, ses objectifs et d'atteindre sa finalité.	

Influence	L'influence est l'action qu'exerce quelque chose sur une autre chose ou sur quelqu'un. Elle est monodirectionnelle.
Interaction	L'interaction est la réaction réciproque de deux phénomènes l'un sur l'autre. Elle est bidirectionnelle
Logigrammes ≈ <i>flow diagrams</i>	Ils permettent de visualiser les actions à mener, étapes et les décisions à prendre pour atteindre un objectif défini. Lorsqu'ils sont utilisés pour présenter un algorithme, on parle de Flowchart [https://fr.wikipedia.org/wiki/Organigramme_de_programmation].
Méthode (dans le cadre de la conception de produit)	Mise en application d'un regroupement de règles et de pratiques d'ingénierie , s'inscrivant dans un processus, permettant de réaliser une démarche technique en vue d'atteindre un résultat. Les tâches de la méthode peuvent être à leur tour réalisées à l'aide d'outils. La méthode se positionne au niveau opérationnel de l'entreprise.
Model-Based Systems Engineering (MBSE) ≈ <i>Ingénierie Système Basée sur les Modèles (ISBM)</i>	Le MBSE permet de conduire une démarche d'Ingénierie Système en s'appuyant sur des modèles, ce qui permet de partager une vision commune entre les acteurs du projet.
Modèle abstrait	Un modèle abstrait permet la représentation des fonctionnalités du système
Modèle concret	Un modèle concret représente des éléments et composants du système
Modèle dynamique	Permet de décrire l'évolution du système au cours du temps
Modèle formel	Un modèle est dit formel si la syntaxe et la sémantique sont définies mathématiquement
Modèle graphique	Un modèle graphique utilise des éléments graphiques (boîtes et flèches !) pour faire passer le message souhaité. Il s'oppose souvent au modèle textuel.
Modèle informel	Un modèle est dit informel si la syntaxe ou la sémantique sont peu ou mal définies
Modèle statique	Permet de décrire la tructure d'un système
Notion d'interaction	La notion d'interaction entre les composants met en lumière l'influence que les éléments peuvent avoir les uns sur les autres.

Outil	Un outil est un moyen d'aide à la réalisation d'une tâche circonscrite d'une méthode en vue d'obtenir et/ou d'améliorer un résultat , d'agir sur un élément en particulier. Un outil peut ainsi aider à réaliser ou supporter une tâche. L'outil est en lien avec le niveau opérationnel de l'entreprise. Un outil intervient dans un but et à un moment défini du développement produit. On peut distinguer les outils d'édition qui ont pour but de créer/produire/améliorer un résultat ; et les outils de gestion qui visent à partager et maintenir un résultat.
Parties prenantes	Une partie prenante est une entité concernée par le système, sa conception, son utilisation, ses impacts dans un contexte donné. De ce fait, la partie prenante est susceptible d'émettre (ou d'avoir sans nécessairement savoir les exprimer !) des besoins ou des contraintes vis-à-vis du système.
Processus	Un processus regroupe une série d'étapes mésoscopiques organisées de manière temporelle pour répondre à une finalité et disposant d'éléments d'entrée (besoin client, cahier des charges), de ressources (financières, humaines, informatiques, etc.) en vue d'obtenir un résultat pouvant prendre la forme d'un produit. Le processus peut être itératif et comporter des jalons. Un processus organise les étapes du développement produit et s'appuie sur des méthodes et des outils pour l'opérationnalisation des étapes. Le processus est relatif au niveau tactique de l'entreprise.
Relation	Une relation est un lien d'influence ou d'interaction.
Structure	La structure est la manière dont les parties d'un tout sont arrangées entre elles. Dans le cadre du système considéré, il représente l'organisation des parties du système qui lui donne sa cohésion et constitue une de ses caractéristiques permanentes.
SysML <i>≈ Systems Modeling Language</i>	Le SysML 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. Issu de l'UML, il est plus adapté à la conception de systèmes multidisciplinaires et est également normalisé par l'Object Management Group (OMG)[https://www.omg.org/].

Système	<p>Un système est un tout, composé d'un ensemble d'éléments qui interagissent de façon organisée pour atteindre un ou plusieurs objectifs définis.</p> <ul style="list-style-type: none">● Définition INCOSE : “An integrated set of elements, sub-systems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services and other support elements”● Définition ISO 15288 : “A combinaison of interacting elements organized to achieve one or more stated purposes”
UML <i>≈ Unified Modeling Language</i>	L'UML est un langage de modélisation graphique supportant une méthode normalisée utilisée dans les domaines du développement logiciel et en conception orientée objet.

Bibliographie

FBS model	Gero J.S. (1990) "Design prototypes: a knowledge representation schema for design", AI Magazine, 11(4), pp. 26–36
General System Theory, von Bertalanffy	General System Theory, Karl Ludwig von Bertalanffy, 1968
Howard, T. J., Culley, S. J., & Dekoninck, E. (2008). Describing the creative design process by the integration of engineering design and cognitive psychology literature. Design Studies	Howard, T. J., Culley, S. J., & Dekoninck, E. (2008). Describing the creative design process by the integration of engineering design and cognitive psychology literature. Design Studies, 29(2), 160–180. https://doi.org/10.1016/j.destud.2008.01.001
https://doi.org/10.1016/j.destud.2008.01.001	
Introduction au Penser Système	Ouvrage AFIS - Edition Collectif AFIS, sous la direction de B.Daniel-Allegro, J.Tucoulou, A.Le Put
ISO15288	ISO/IEC/IEEE 15288:2015, Systems and software engineering - System life cycle processes, Geneva
Jean-Louis Le Moigne, La théorie du système général	Jean-Louis Le Moigne, La théorie du système général - théorie de la modélisation, 1977
OMG-SysML	Object Management Group - Systems Modeling Language (SysML)
https://www.omgsysml.org/	
Pahl, G., Beitz, W., & Wallace, K. (2003). Engineering design: a systematic approach	Pahl, G., Beitz, W., & Wallace, K. (2003). Engineering design: a systematic approach. Springer.
Projet D_TechnoSS	Projet D_TechnoSS - UTT ^[https://d-technoss.utt.fr/]
Raymer, D. P. (1999). Aircraft design: a conceptual approach (3rd edition)	Raymer, D. P. (1999). Aircraft design: a conceptual approach (3rd edition). American Institute of Aeronautics & Astronautics, Inc.
SysML.org	Site SysML.org, proposé par l'entreprise PivotPoint Technology Corp.
https://sysml.org	
SysML par l'exemple - Un langage de modélisation pour systèmes complexes	SysML par l'exemple - Un langage de modélisation pour systèmes complexes, Pascal Roques, 2011
https://www.eyrolles.com/Informatique/Livre/sysml-par-l-exemple-un-langage-de-modelisation-pour-systemes-complexes-9782212850062/	

Thèse Chen Zheng (2015)	Thèse Chen Zheng (2015)
Manuscrit thèse Chen Zheng	
Thèse Julia Guérineau	Soutenu le 07/12/2021 à l'Université de Technologie de Compiègne et à l'Ecole de Technologie Supérieure de Montréal