

# Modélisation et analyse des systèmes hétérogènes (DIS-MASTIC-05)

Bases

J. Christian Attiogbé

Février 2023 - Collège doctoral Pays de la Loire

# Plan de la suite (adaptation selon contraintes)

- 1 Introduction
- 2 Caractérisation de systèmes hétérogènes
- 3 Approche Ingénierie Système
- 4 Introduction aux sciences de l'ingénieur et au génie logiciel
- 5 Analyse des besoins
- 6 Analyse des exigences/besoins (Requirement analysis)
- 7 Introduction à la modélisation

# Présentation du cours

- Hétérogénéité des systèmes : la question, des éléments de caractérisation
- Abstractions nécessaires et leur formalisation avec des objets manipulables
- Ouverture sur l'interdisciplinarité, pendant les études/constructions/travaux

# Chapitre 1 : **Hétérogénéité** des systèmes

👉 Caractériser et décrire les **systèmes**

# Systèmes : essai de définition

## Système

Un système est un ensemble de **composants** (ou **sous-systèmes**) en **interaction**.

Le comportement du système dépend non seulement des comportements individuels des composants, mais surtout de l'**interaction entre les composants**.

# Exemple de système



Figure: Une horloge mécanique

# Exemple de système

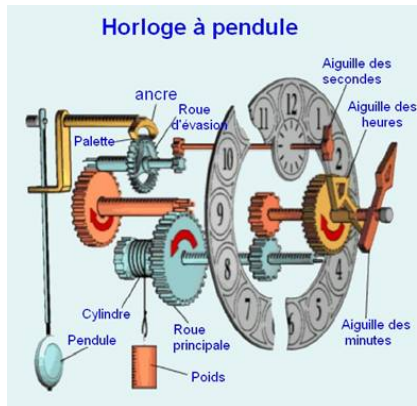


Figure: Une horloge mécanique ([lamesuredutemps.weebly.com](http://lamesuredutemps.weebly.com))

# Exemples de système

## Exemple de système

Système d'**acheminement d'un message entre acteurs distants** (classique/numérique).

**Composants** : acteurs (ordinateurs ou humains), serveurs (ordinateurs), réseaux (infrastructures variées), équipements du réseau.

## Exemple de système

**Système de production dans une industrie** (alimentaire, automobile, électroménager, ...)

**Composants** : des postes-métiers, des postes alimentations en matières, des stocks, du convoyage,



# Système complexe

## Système complexe

Un système dans lequel les **relations** et les **interactions** entre les composants (éventuellement nombreux) sont en **grand nombre**, de **différentes sortes**, non triviales/élémentaires, ....

## Système complexe

et en interaction avec **son ou l' environnement**, changeant (évolutif, incertain)

# Exemple de système complexe

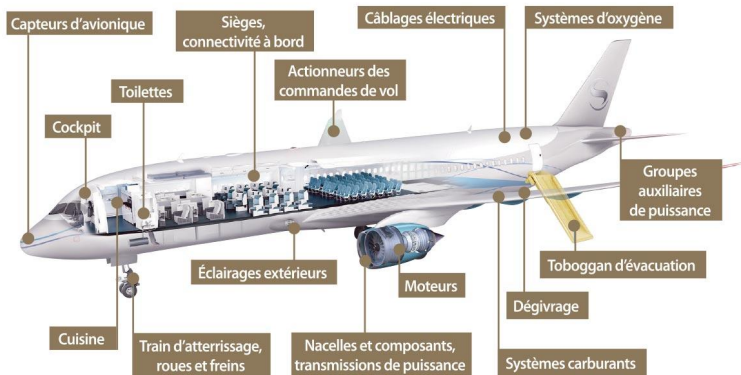


Figure: Composants d'un avion ()

# Exemples de systèmes complexes

## Exemple de système complexe

Pilotage automatique d'un avion

**Composants d'un avion** : nombre élevé, interactions nombreuses en temps réel

## Exemple de système complexe

- (contrôle) Centrale nucléaire
- (système) Transaction bancaire
- Contrôle du transport (aérien, ferroviaire, etc)

# Systemes hétérogènes

## Systeme hétérogène

Un système dont les **composants** sont de natures différentes, variées, en grand nombre ou pas.

# Exemples de Systèmes hétérogènes

## Exemple de Système hétérogène

- Un avion (de nombreux composants, de différentes natures)

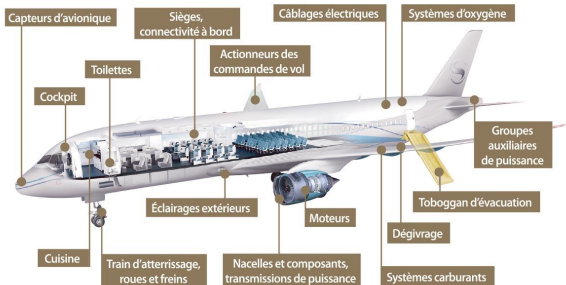


Figure: Composants d'un avion ()

# Un composant du système : train d'atterrissage



Figure: Train d'atterrissage (prd-sc101-cdn.rtx.com)

# Composant du train d'atterrissage : partie hydraulique

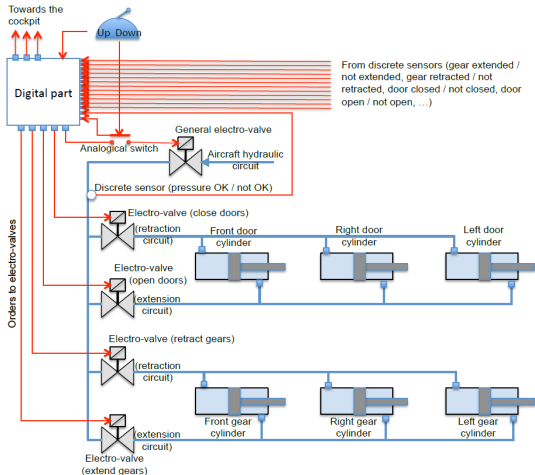


Figure: Partie hydraulique du train d'atterrissage (ABZ case study)

# Autres exemples de systèmes hétérogènes


## Applications basiques - informatiques

- Serveur/clients de données ou de services variés
- Les clients sont divers et variés
- Interface standardisée (socket - middleware)
- Systèmes industriels automatisés
- Systèmes industriels+humains/environnement



# Autres exemples de systèmes hétérogènes (suite)

## Applications informatiques, basiques ou non

- Système de tri postal - Chine  
vidéo  cliquez ici pour la Video Youtube.
- Système de production ( par exemple de véhicule : postes de travail soudure/peinture, humain... , tapis roulant, ),
- Centrale nucléaire
- Voiture (mécanique, électronique, ...)
- Mécatronique
- Métier à tisser - Filature
- Systèmes naturels
- ...

# Exemples de systèmes hétérogènes



Figure: El campesino, Nantes 2011

# Exemples de systèmes hétérogènes (suite)



Figure: Eléphant des machines de l'Île - Nantes (i-pinimg.com)

# Exemples de systèmes hétérogènes (suite)

Les systèmes naturels, biologiques, vivants



Figure: Eléphant à Mae Sa - Thaïlande (mackoo.com)

# Systèmes naturels versus systèmes industriels

Interaction entre les végétaux

Interaction entre les animaux

Communauté humaine : interaction entre de petits animaux

Interaction entre des systèmes de production, de communication, de contrôle, financier, etc

# Que conclure à partir de ces exemples ?

Qu'est-ce qui caractérise les systèmes (complexes) hétérogènes ?

Observer




Caractériser ↔ modéliser ↔ analyser



Prédire - développer - maintenir

# Pistes de travail pour nos préoccupations

En vue de la modélisation et l'analyse de systèmes hétérogènes :

- Caractérisation des systèmes et de leurs composants
- Modélisation des composants et du système hétérogène
-  Approche globale d'analyse en amont de la modélisation → ingénierie système

# Caractérisation des systèmes hétérogènes : essai

## Quelques caractéristiques

- Forte interaction  
composition, communication, échange, ...
- Type  
diversité des comportements/fonctionnalités,  
diversité des données échangées,  
diversité des échelles
- Hiérarchie entre les composants d'un système  
(priorité, prédation, préemption, ...)
- Impact de l'environnement

**modéliser** : des interactions, des types de composants, de données, de protocoles, d'échelles de valeur, d'échelle de temps, des hiérarchies et des priorités, des incertitudes/des paramètres



# Caractér. des systèmes hétérogènes : essai (suite)

## Quelques caractéristiques

- Invariant ? des systèmes  
quelles propriétés invariantes ?
- Langages de communication ?  
Y a-t-il nécessité d'un langage de communication pour  
l'interaction dans le système ?  
nécessité d'un langage/de langages pour décrire le système  
hétérogène ?

**modéliser** : des propriétés locales aux composants, globales au système, invariantes, des exigences/engagements, ...

**choisir/définir** : le/les langage/s

# Ingénierie système (*Systems Engineering*)

Approche IS convient à la modélisation des systèmes hétérogènes

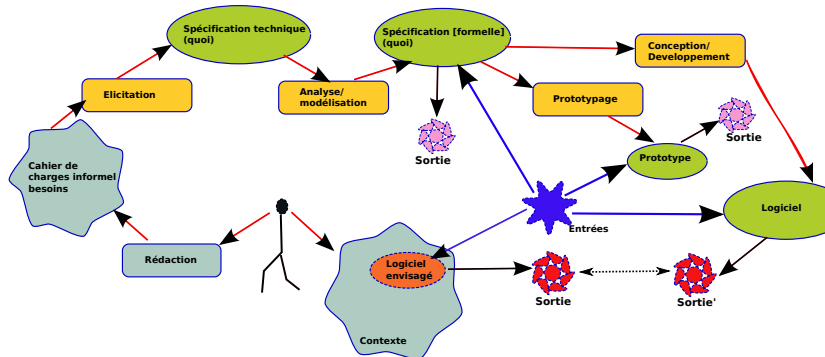


Figure: Construction du logiciel

# Ingénierie système (*Systems Engineering*)

## Système rappel

Un système est un ensemble de **composants** en interaction. Le comportement du système dépend non seulement des comportements individuels des composants, mais surtout de **l'interaction entre les composants**.

- la notion de système se décline en plusieurs niveaux : un composant peut être vu à son tour comme un système et ainsi être décomposé en d'autres **sous-composants (sous-systèmes)**.
- Un composant peut être physique (matériel) ou logiciel.
- L'ingénierie des systèmes offre un cadre adapté à l'étude des produits pluritechnologiques.
- Des langages comme **SysML**, **AADL** permettent de décrire un système avec ses composants.

# Ingénierie système

## Systems Engineering: the Process (*System Engineering*, Boeing)

A process that transforms an operational need or market opportunity into a system description to support detail design.

- Requirements Analysis
- Functional Analysis
- Synthesis
- Systems Analysis / Management

# Ingénierie système

## Systems Architecture: a Product (*System Engineering*, Boeing)

The design teams interpretation / implementation of customer requirements communicated through:

- System usage scenarios (i.e., Use Cases)
- Functional components & interrelationships
- Physical subsystems & interfaces
- Etc

# Ingénierie système

## Benefits of Architecting (*System Engineering*, Boeing)

- Identifies all system elements **Earlier vs. Later**
- Matches function to requirements
- Capture & communicate key concepts
- Results in **one design**
- **Manages increasing complexity**
- Allows **modular design**

# Ingénierie système

## Summary

- Increasingly complex systems drive **a need for better, clearer design descriptions**
- **Architectures** convey the system designers interpretation of the requirements
- Architectures may be presented by **a variety of views which collectively describe the system**
- As part of the systems engineering process, systems architecting defines and manages development of a system

# Fondamentaux en ingénierie

Définitions : Exigences, besoins, spécifications, cahier de charges, ...

Besoins, Needs, *Requirement* - côté client

*Requirements are the users' description of what the finished product should do.*

Spécification des besoins - côté technique

*A specification is an **explicit set of requirements** to be satisfied by the software.*

*A specification is the technical description of the software, covering the requirements and much more (cost, technicalities, problems, ...).*



# Fondamentaux en ingénierie

## Notions fondamentales

Cahier de charges, Spécifications fonctionnelles,  
Architecture des applications logicielles, Cycles de vie

## Analyse des besoins

### Exigences fonctionnelles

Elicitation, analyse,

⇒ spécification, vérification,  
validation des besoins d'un  
(projet) logiciel/système.

## Ingénierie système

### Décomposition fonctionnelle,

Décomposition modulaire,

⇒ Construction des composants,  
Assemblage des composants  
par composition !

# Conception et réalisation de produits industriels

En sciences de l'ingénieur la **démarche scientifique pour réaliser un produit industriel** passe par deux principales phases : la **conception** puis la **fabrication** du produit.

Les **étapes du processus de réalisation** sont :

- l'élaboration du cahier des charges fonctionnel,
- l'analyse,
- la conception,
- la réalisation de prototypes,
- la mise au point produit-process,
- l'industrialisation.

Ce sont les mêmes étapes qui sont suivies pour la réalisation de logiciels ou de systèmes contenant du logiciel.

# Etapes de conception et de réalisation de produits

- La durée des étapes est plus ou moins longue, selon les contextes, les méthodes, les outils.
- L'**analyse** des systèmes **consiste à décomposer le/s produits en composants (sous-systèmes)**
- L'étape de mise au point se fait généralement par **simulation** et **essais**.  
Certaines étapes/phases, comme la simulation, sont elles même assistées par ordinateur.
- Lorsqu'on adopte une démarche rigoureuse, bâtie sur des méthodes formelles de modélisation et d'analyse, on allonge le temps de l'analyse conception et on réduit considérablement le temps des mises au point. On idéalise la **construction correcte immédiate**.

# Conception de produit (ou analyse fonctionnelle)

## Objectif

L'objectif de la conception est de prévoir ce que sera le comportement du produit lors de son utilisation en conditions réelles.

## Réponse d'un produit

Le **produit interagit avec son environnement**, ce qui entraîne une modification de certaines grandeurs physiques pouvant caractériser soit l'environnement, soit le produit lui-même. Une telle modification est appelée réponse du produit.

## Conception

La tâche du concepteur est de **déterminer les réponses, ou du moins celles qui font partie des objectifs de l'étude**, afin de s'assurer que les performances réalisées seront bien conformes aux performances attendues.

# Etude d'un système

Dans les sciences de l'ingénieur, deux approches scientifiques permettent de déterminer et prévoir les réponses et les performances réalisées par un produit, à partir des informations dont on dispose sur l'environnement.

- l'**approche par simulation (on travaille dans le virtuel)** : elle est basée sur une **modélisation mathématique**
- l'**approche par essai ou mesure (on travaille dans le réel)** : consiste à réaliser un **prototype/une maquette** du produit, le mettre dans les conditions réelles ou identiques, puis de mesurer (en s'aidant d'instruments) les réponses et les modifications des grandeurs de l'environnement.

# Diagnostic en conception

La **conception est validée** lorsque l'**écart entre les résultats/comportements/performances attendus et ceux observés est le plus faible possible**.

De fortes hypothèses :

- le modèle pour la simulation est fidèle au produit.
- l'essai peut venir confirmer les résultats de la simulation
- les conditions de l'essai sont représentatives des situations réelles

Lorsque l'**écart est grand**, on doit effectuer un diagnostic pour **trouver les causes et les corriger**.

# Analyse des exigences (*Software requirements analysis*)

## Analyse des exigences - du logiciel au système ?

Les **exigences** du logiciel (**Software requirements**) expriment les besoins et les contraintes liés à un logiciel qui (peut) contribue(r) à la solution d'un problème réel.

On a des exigences sur le produit/logiciel ou des exigences sur la démarche de réalisation.

Une propriété essentielle de toute exigence, est qu'elle soit vérifiable en tant que **caractéristique élémentaire**, comme **exigence fonctionnelle** ou comme **exigence non-fonctionnelle** au niveau global d'un système.

Elicitation, **analyse**, **modélisation**, **vérification**, validation des besoins d'un (projet) logiciel / système.



## Fondamentaux - logiciels/systèmes

Les **exigences fonctionnelles** décrivent les fonctions que le logiciel doit fournir.

Les **exigences non-fonctionnelles** sont celles qui contraignent la solution fournie par le logiciel.

Les propriétés non-fonctionnelles sont souvent vues comme des contraintes ou des exigences de qualité.

Les propriétés non-fonctionnelles peuvent être classées en plusieurs sous-catégories :

- exigences de performance, exigences de maintenabilité, exigences de sûreté,
- exigences de fiabilité, exigences de sécurité, exigences d'interopérabilité,
- etc

# Sources des exigences et élicitation des exigences

Les sources des exigences d'un logiciel peuvent être nombreuses et variées ;

il faut les identifier pour en faire un recueil approprié (élicitation des besoins).

Il y a différentes techniques d'élicitation des besoins :

- Interviews, scenarios (tels que les *use case de UML*),
- Réunions de groupe, observation, *User stories*, ...
- Prototypes/maquettes

Réaliser ensuite un cahier de charges (spécification informelle des besoins).

Extension aux systèmes hétérogènes ?

# Analyse des exigences (*Requirements Analysis*)

L'analyse des exigences permet

- de détecter et résoudre d'éventuels **conflits entre les exigences**.
- de découvrir les **limites du logiciel et comment il interagit son environnement**,
- **d'élaborer les exigences systèmes pour en dériver les exigences du logiciel**.

Les approches classiques d'analyses des exigences intègrent la modélisation (conceptuelle) en utilisant des méthodes telles que l'analyse structurée. On peut y ajouter la classification des exigences.

# Classification des exigences (*Requirements Classification*)

- Exigences **fonctionnelles** ou **non-fonctionnelles** (délais/temps, sécurité, qualité).
- Exigences sur le produit (logiciel) ou exigences sur la procédure (ITIL, SIL, ...)

Les exigences sur la procédure peuvent contraindre le choix des contractants, la démarche d'ingénierie logicielle à adopter, les normes ou méthodes à utiliser, etc.

# Méthodes/Outils d'analyse des exigences

- KAOS : *Knowledge Acquisition in automated specification* ou *Keep All Objectives Satisfied*
- Tables de Parnas : formalisme rigoureux pour expliciter les besoins
- SCR (Software Cost Reduction),
- RSML (*Requirements State Machine Language*)

## Modélisation conceptuelle (*Conceptual Modeling*)

L'élaboration de modèles pour un problème réel est fondamentale pour l'analyse des exigences du logiciel.

Les modèles aident à comprendre les situations dans lesquelles le problème survient, aussi bien que pour esquisser une solution.

Ainsi les modèles conceptuels comprennent les modèles des entités du domaine du problème, configurés pour refléter leurs inter-relations et dépendances.

# Modélisation conceptuelle (*Conceptual Modeling*)

Les modèles incluent par exemple :

## Exemples de modèles

des modèles à états : les **automates** - (diagrammes états-transitions) + variantes : **réseaux de Petri**, **algèbres de processus**,  
des modèles **stochastiques**, **probabilistes**  
des modèles de flots de données,  
des modèles à objets (statiques),  
des interactions entre usagers,  
etc

# Modélisation conceptuelle (*Conceptual Modeling*)

Les modèles incluent par exemple :

## Exemples de modèles

- des modèles à base d'équations différentielles
- Modèle à événements discrets : DEVS (Discrete Event Simulation), Event-B, ...
- Modèle à base d'agents
- Les automates cellulaires
- des "modèles" à base de l'algorithme de Boids (les *bird-oid objects*, Craig Reynolds)
- plus généralement, des modélisations pour systèmes naturels : équations différentielles, ...



# Modélisation conceptuelle (*Conceptual Modeling*)

Parmi les facteurs qui influent sur la modélisation :  
il y a notamment la **nature du problème** ou du **système**.

Les facteurs qui influent sur le choix du formalisme de modélisation :  
il y a notamment la **nature du problème**  
la **nature des composants**  
le **type de logiciel ou de système...**

# Modélisation formelle et analyse

L'expression formelle des exigences requiert l'emploi d'un langage avec une sémantique précise.

Le recours à l'analyse formelle des exigences permet :

- de les spécifier précisément et sans ambiguïtés évitant ainsi les mauvaises interprétations (modèles formels)
- de pouvoir raisonner sur les exigences, de prouver les propriétés du logiciel spécifié.

L'analyse formelle requiert des spécifications/modélisations formelles comme point de départ.

# Des considérations pratiques

Le traitement des exigences couvre tout le cycle de vie du logiciel ou du système.

- Changement des besoins initiaux
- Maintenance des exigences dans l'état qui reflètent le logiciel à construire, ou du logiciel qui est construit.
- Tracabilité des exigences.

# Références

- Parnas's Tables  
<https://cs.uwaterloo.ca/~jmatlee/talks/parnas01.pdf>
- Beck, A., Boeing, G., & Shannon, D. *Systems and Methods for Analyzing Requirements. US Patent 8650186*, 2014
- Chemuturi, M. (2013). *Requirements Engineering and Management for Software Development Projects*, ISBN 978-1-4614-5376-5, 2013
- Software Development Process activities and steps  
[http://www.uacg.bg/filebank/acadstaff/userfiles/publ\\_bg\\_397\\_SDP\\_activities\\_and\\_steps.pdf](http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf)
- B.P. Zeigler, *Theory of modelling and simulation*, Wiley, 1976