

Supélec

Modélisation des systèmes

Modélisation « Multi-Paradigme(s) »

Cécile Hardebolle

cecile.hardebolle@supelec.fr

Thématiques abordées

- ▶ Pourquoi utiliser plusieurs langages de modélisation ?
- ▶ Langages et outils fondamentalement hétérogènes
 - ▶ Cas des systèmes hybrides
 - ▶ Cas des outils tels que Matlab/Simulink
 - ▶ Cas d'UML
- ▶ Approches liées à l'Ingénierie Dirigée par les Modèles
- ▶ Problématiques liées à la composition de modèles



Pourquoi plusieurs langages ?

Pourquoi plusieurs langages ?

1. Contexte et rappels
2. Décomposition en sous-systèmes
3. Raffinement de modèles
4. Activités du cycle de conception
5. Vues non fonctionnelles
6. En résumé

Exemple de système complexe : STARMAC/PARROT A.R. Drone

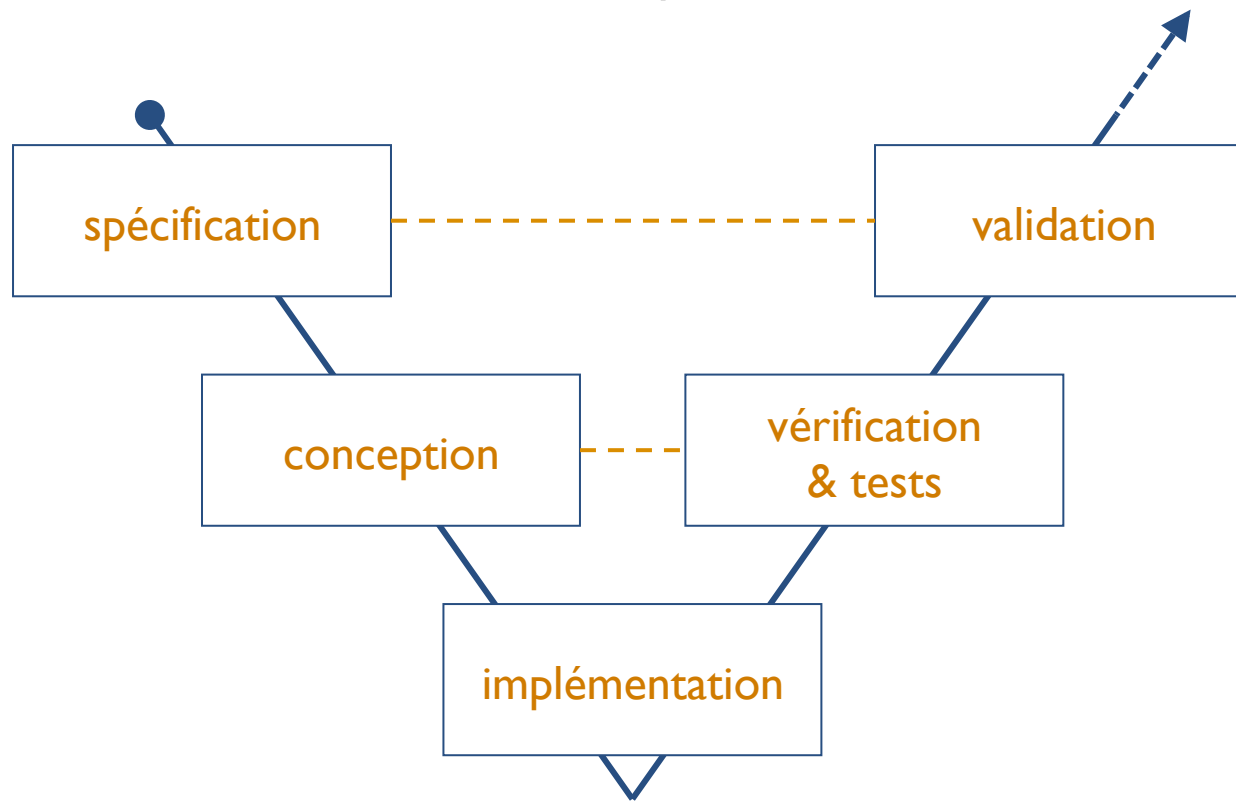


Exemple de système complexe : STARMAC/PARROT A.R. Drone



Ingénierie système

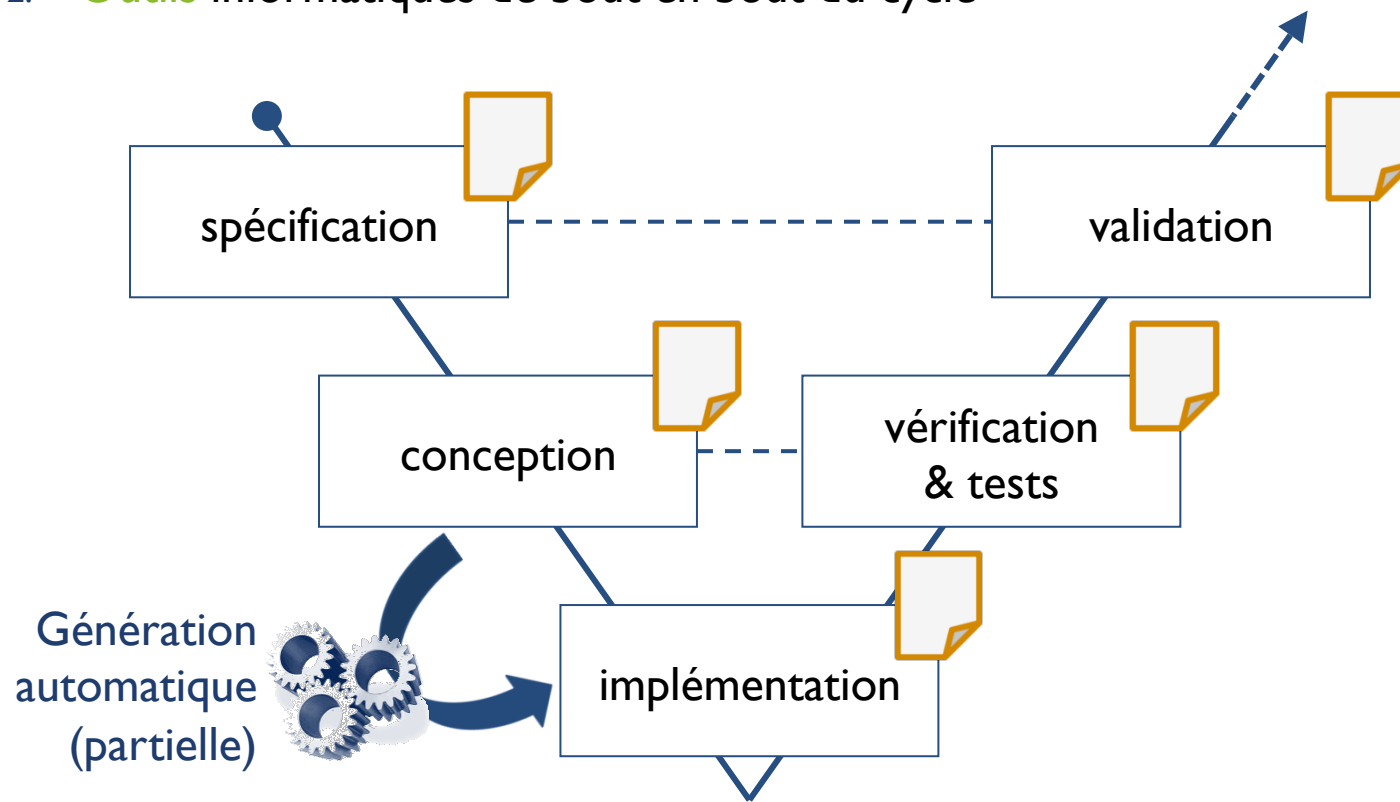
- « L'ingénierie désigne l'ensemble des fonctions s'étendant de la conception et des études à la responsabilité de la construction et au contrôle des équipements d'une installation technique ou industrielle. » [Wikipédia]



Ingénierie Dirigée par les Modèles

► Ingénierie Dirigée par les Modèles (IDM) =

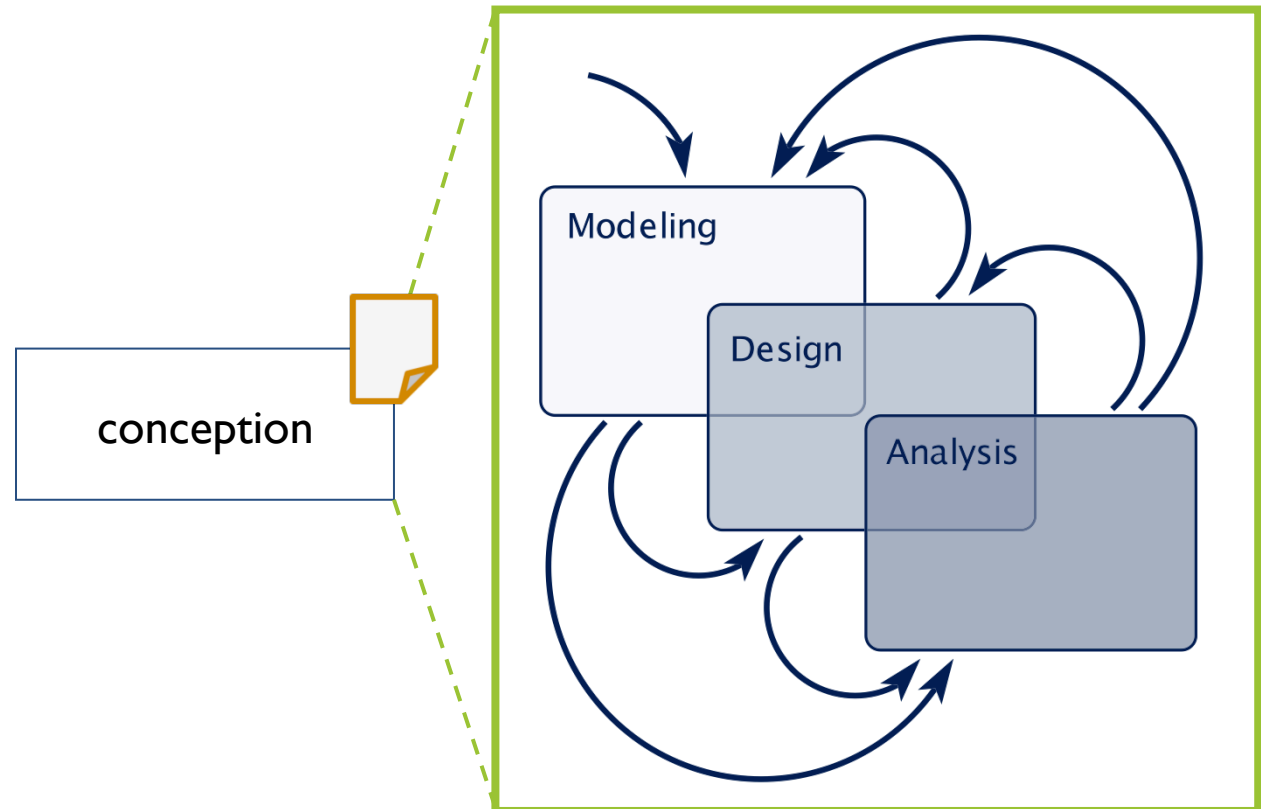
1. **Modèles** informatiques = support prépondérant du cycle de développement
2. **Outils** informatiques de bout en bout du cycle



Ingénierie Dirigée par les Modèles

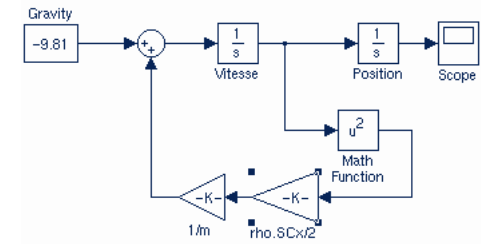
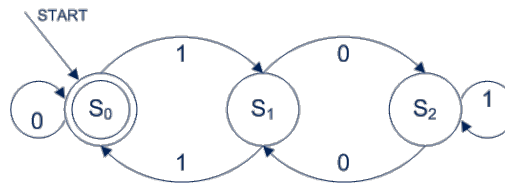
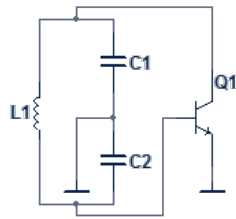
► Ingénierie Dirigée par les Modèles (IDM) =

1. **Modèles** informatiques = support prépondérant du cycle de développement
2. **Outils** informatiques de bout en bout du cycle



Langage de modélisation

- ▶ **Langage de modélisation** : langage informatique permettant de décrire des modèles
 - ▶ Syntaxe abstraite : **concepts**
 - ▶ Sémantique : **signification**
 - ▶ Syntaxe concrète : **notation**



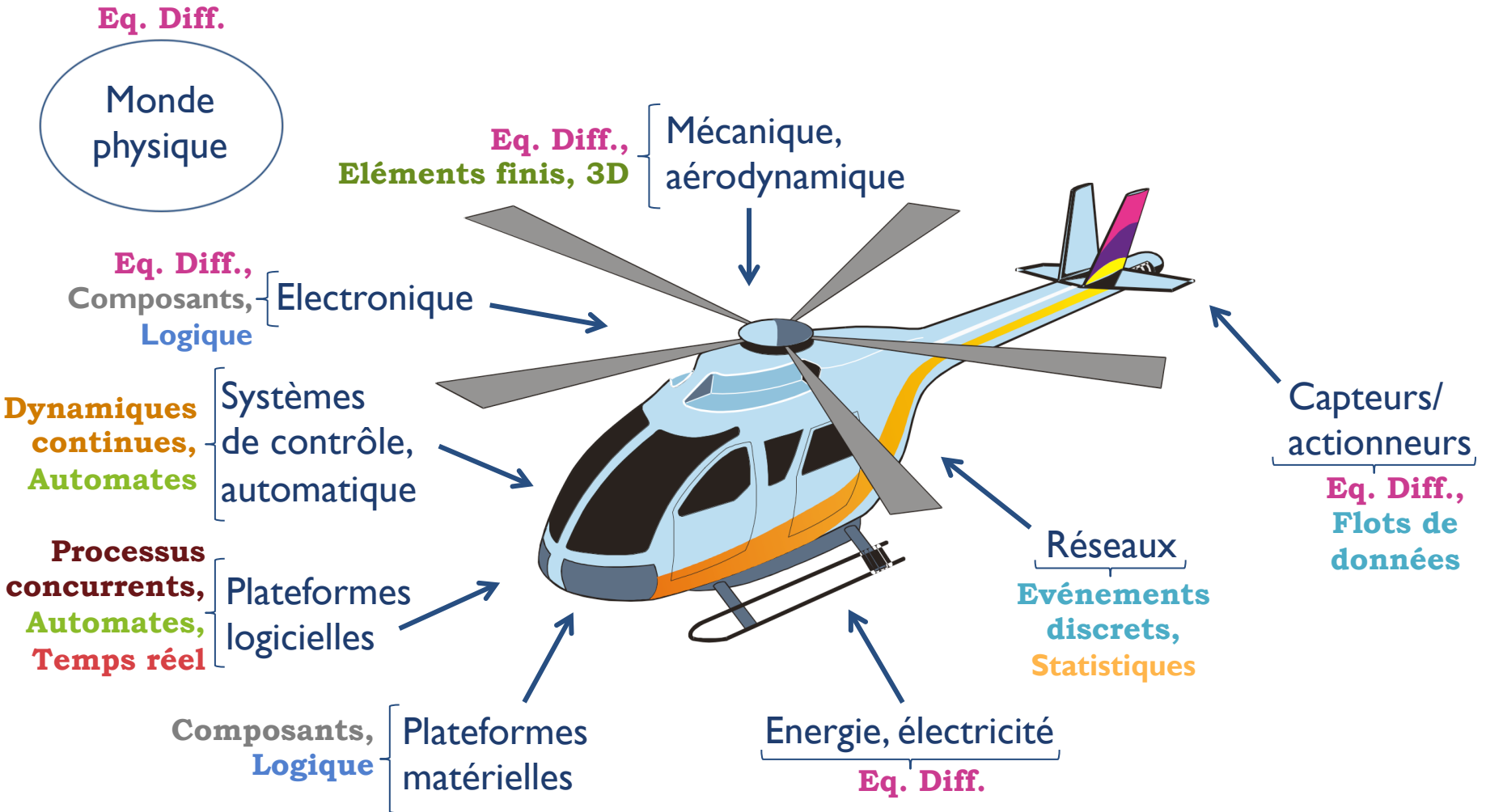
- ▶ **Paradigme de modélisation** : « façon de modéliser »
 - ▶ $\approx$ **Famille de langages** de modélisation qui partagent des éléments de syntaxe et de sémantique

▶ **Modélisation « multi-paradigme »** = modélisation avec **plusieurs paradigmes** !

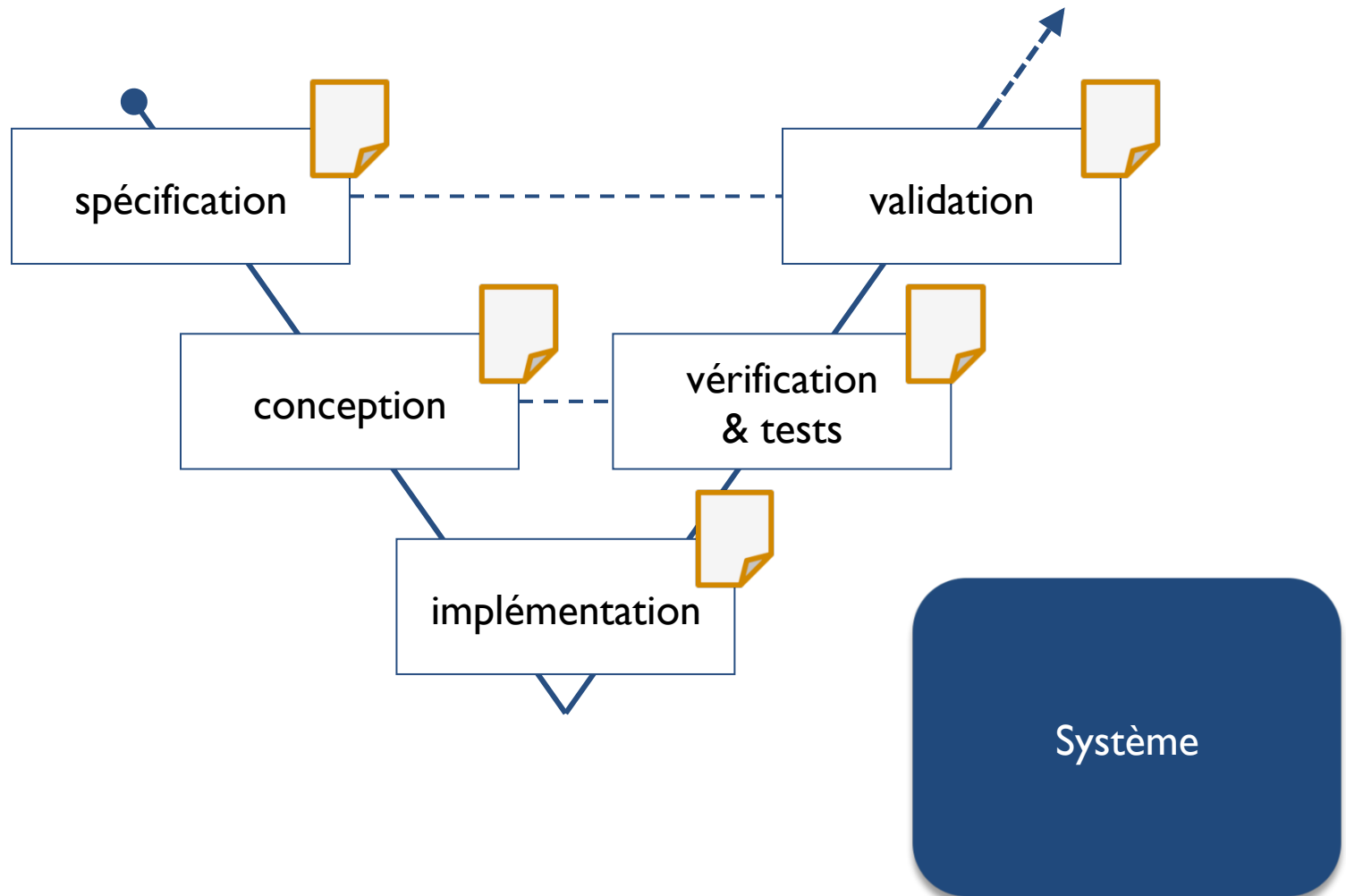
Pourquoi plusieurs langages ?

1. Contexte et rappels
2. Décomposition en sous-systèmes
3. Raffinement de modèles
4. Activités du cycle de conception
5. Vues non fonctionnelles
6. En résumé

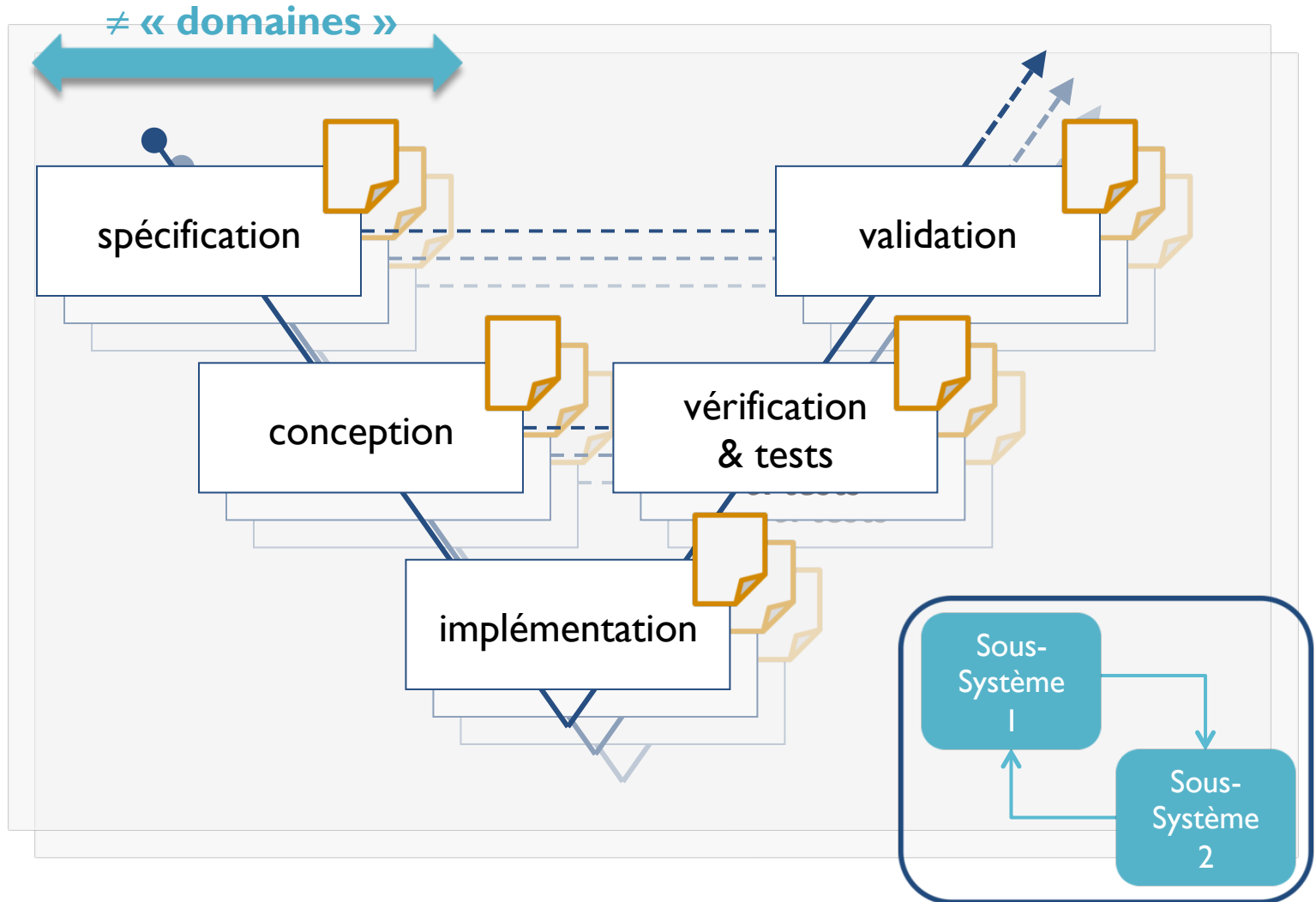
Pourquoi plusieurs langages ?



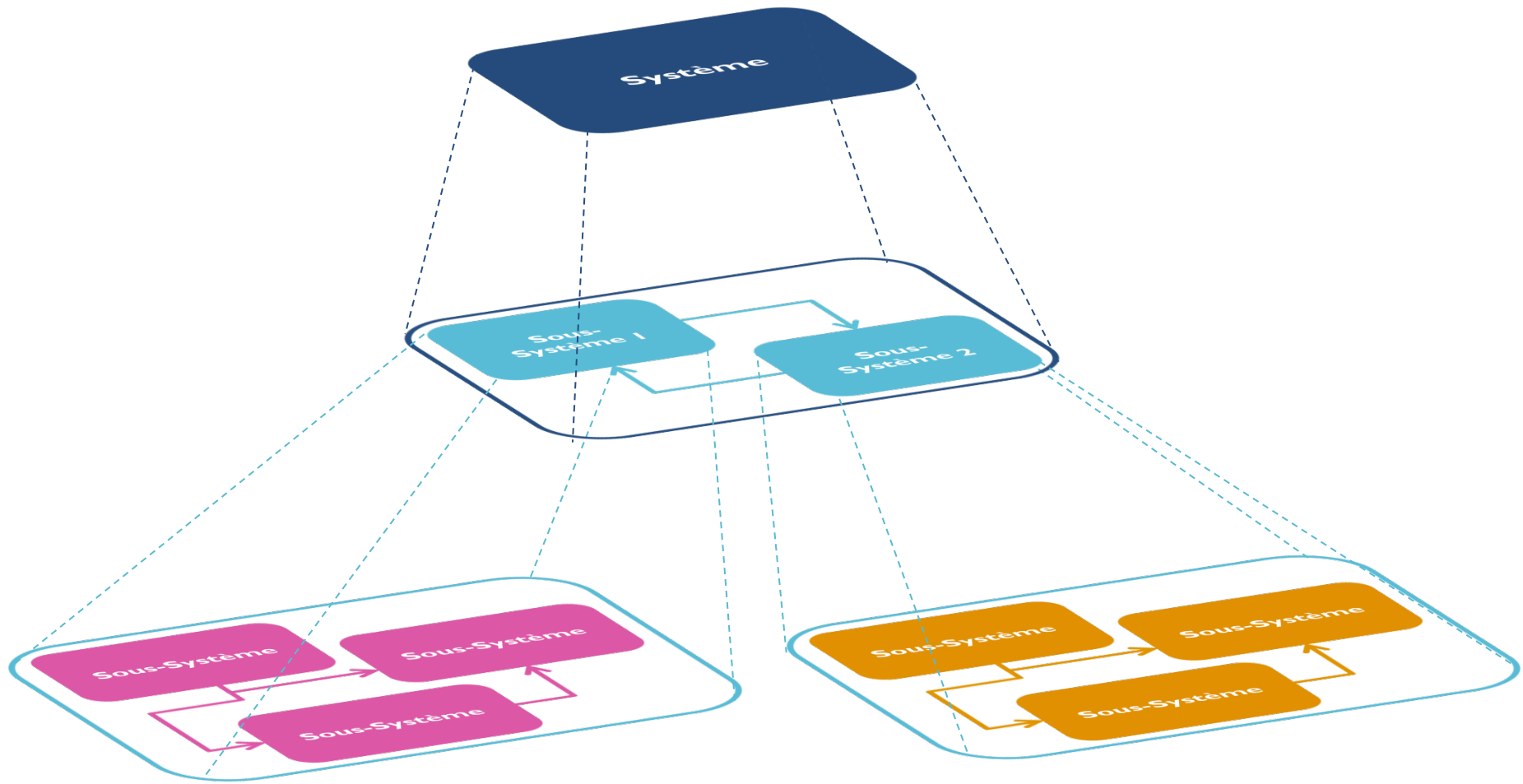
Décomposition en sous-systèmes



Décomposition en sous-systèmes



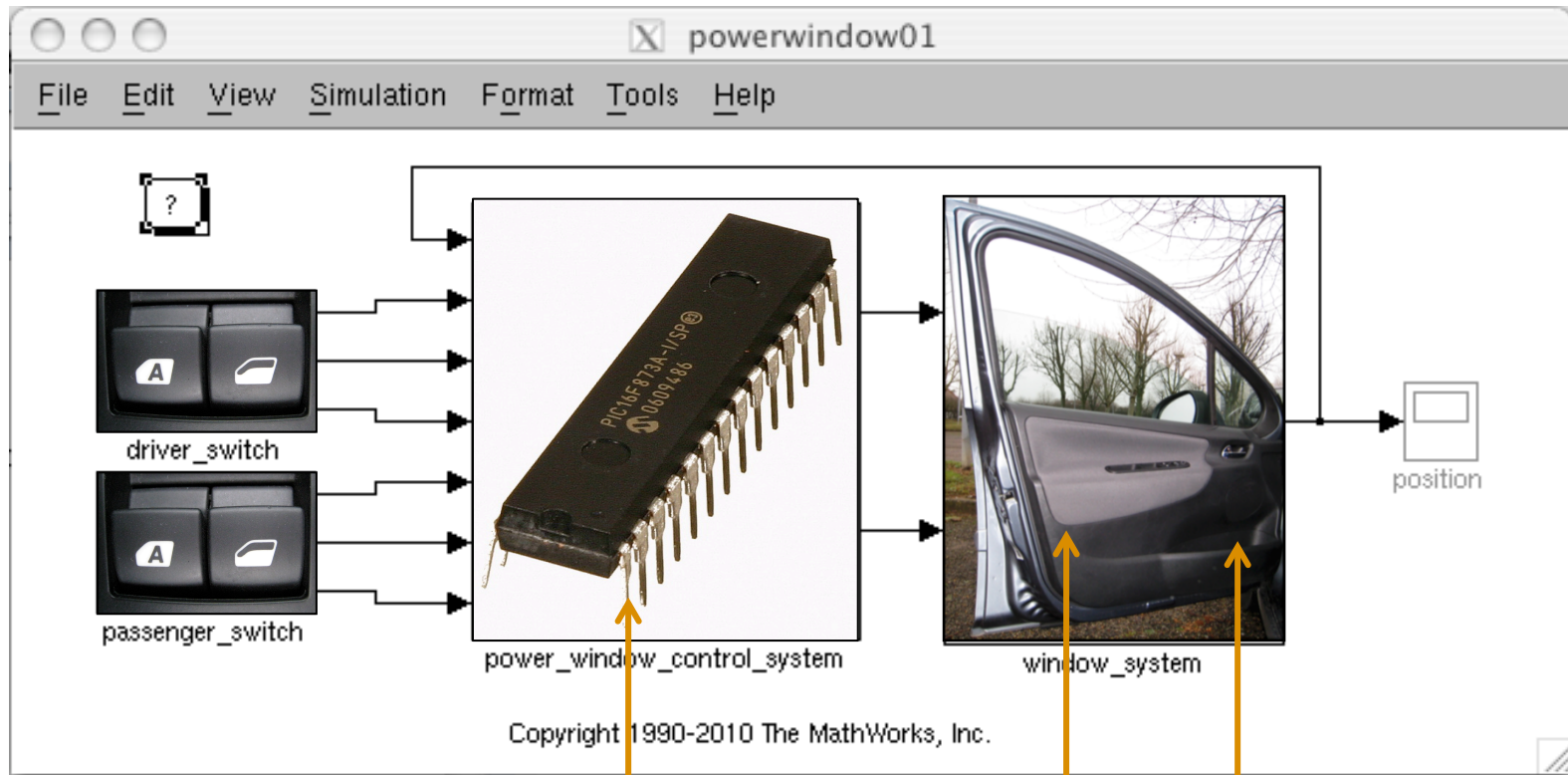
Modélisation hiérarchique



Exemple : Lève-vitre



Exemple : Lève-vitre



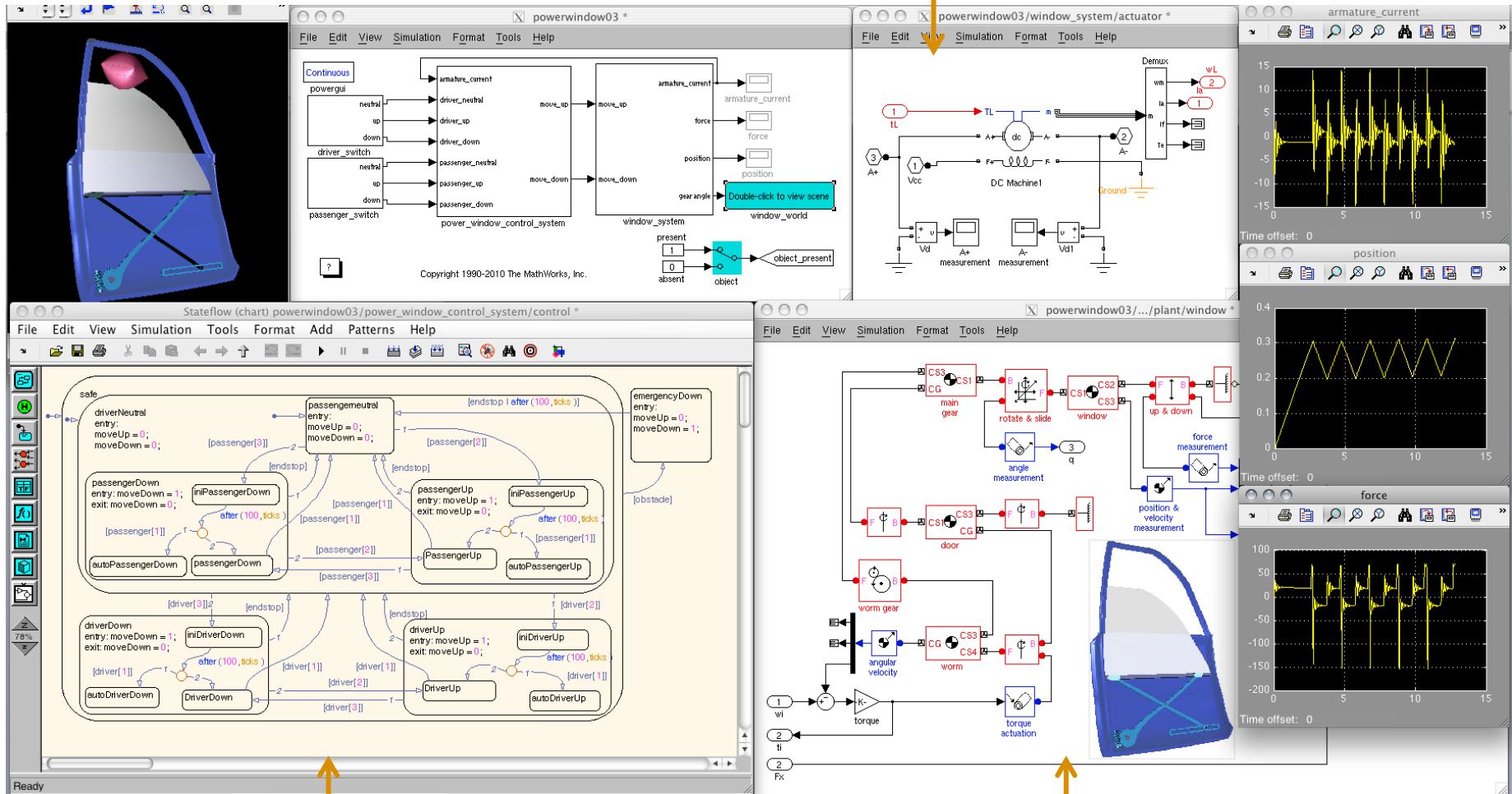
Système de
contrôle

Mécanique

Electricité

Exemple : Lève-vitre

Electricité



18

Système de contrôle

Modélisation des systèmes

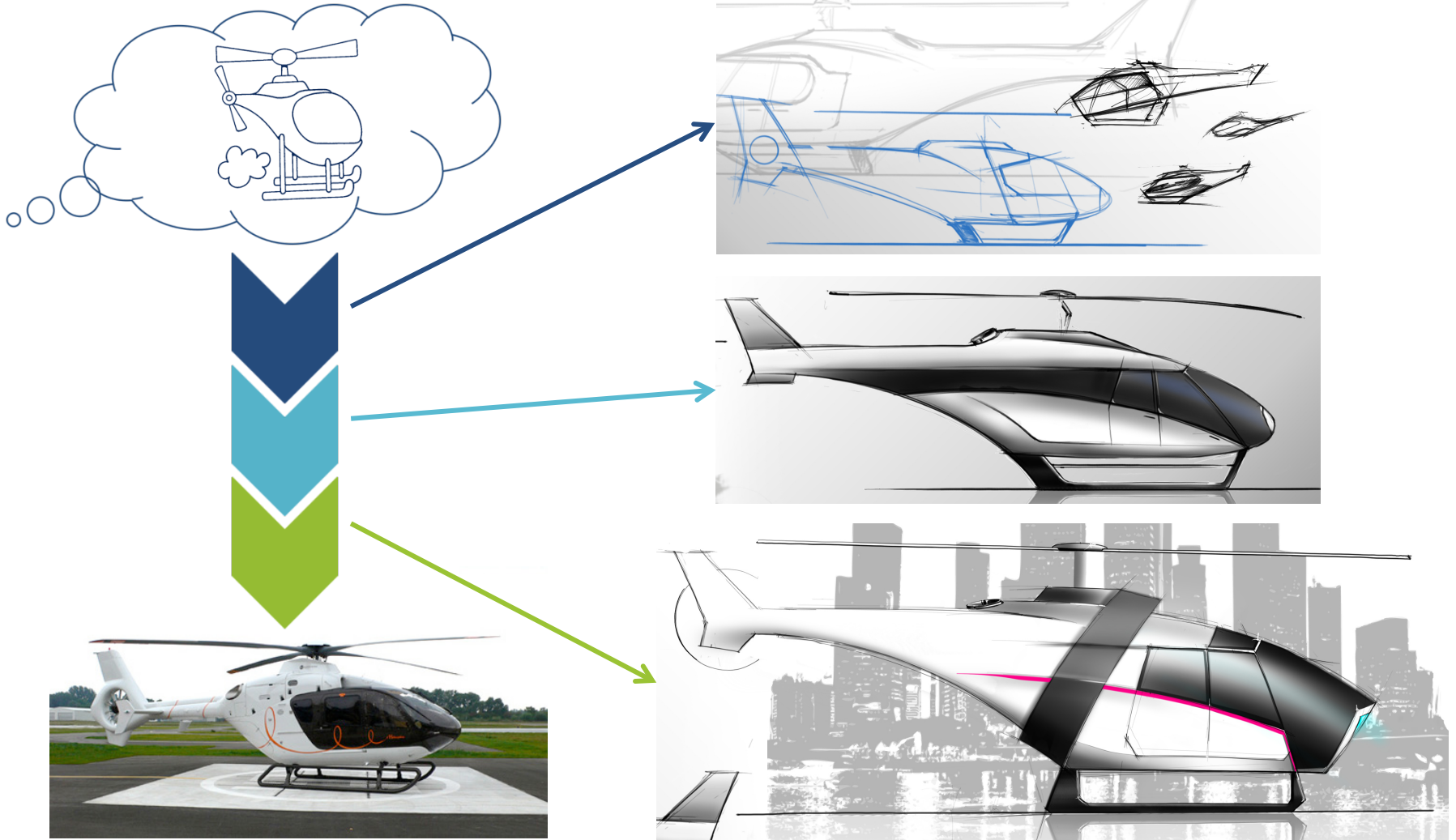
Mécanique

16/02/11

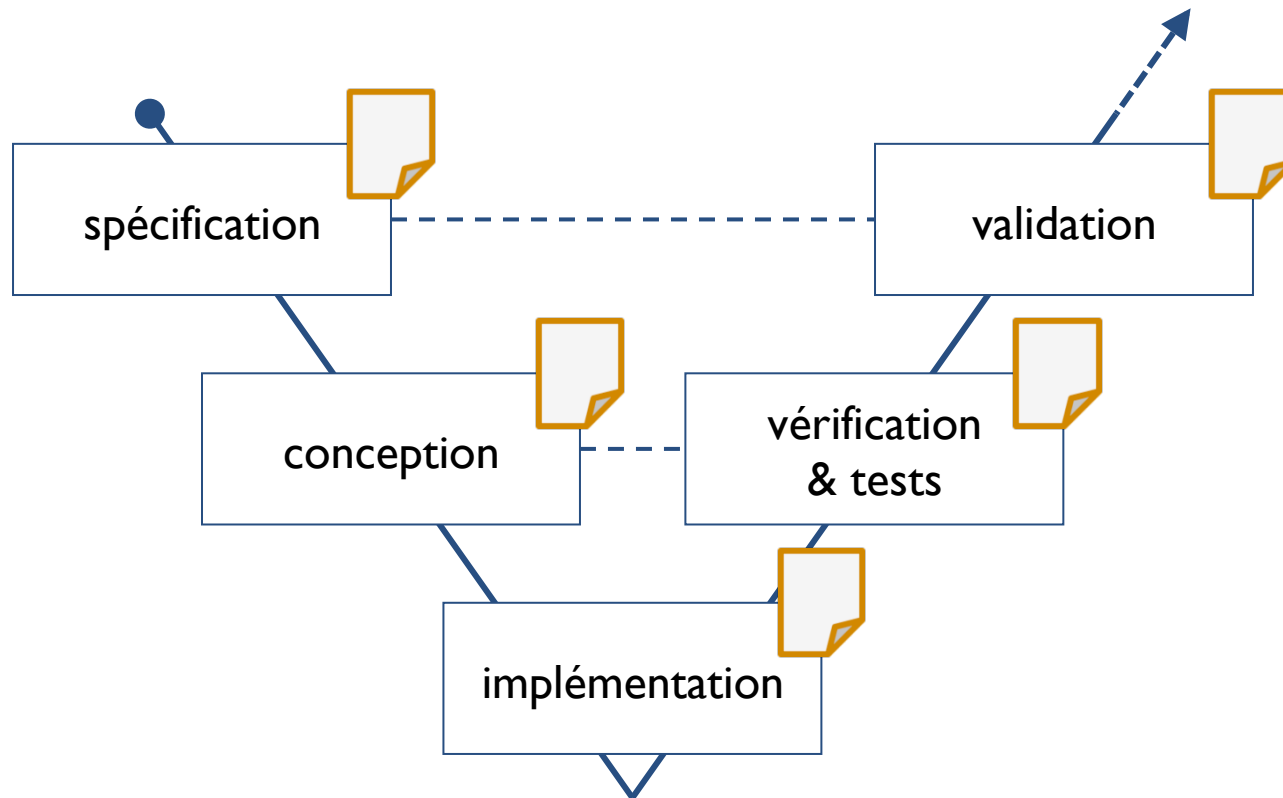
Pourquoi plusieurs langages ?

1. Contexte et rappels
2. Décomposition en sous-systèmes
3. Raffinement de modèles
4. Activités du cycle de conception
5. Vues non fonctionnelles
6. En résumé

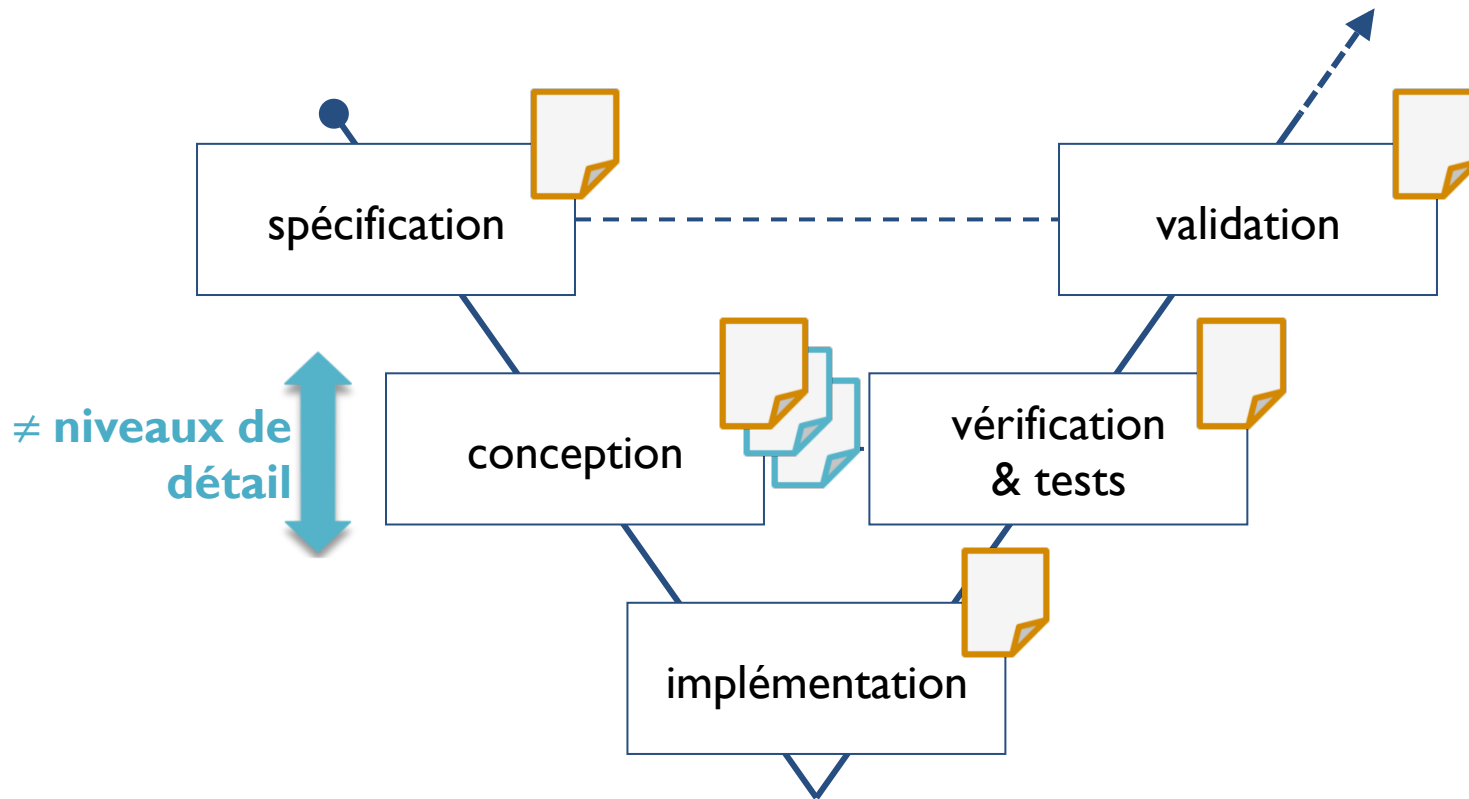
Pourquoi plusieurs langages ?



Raffinement de modèles



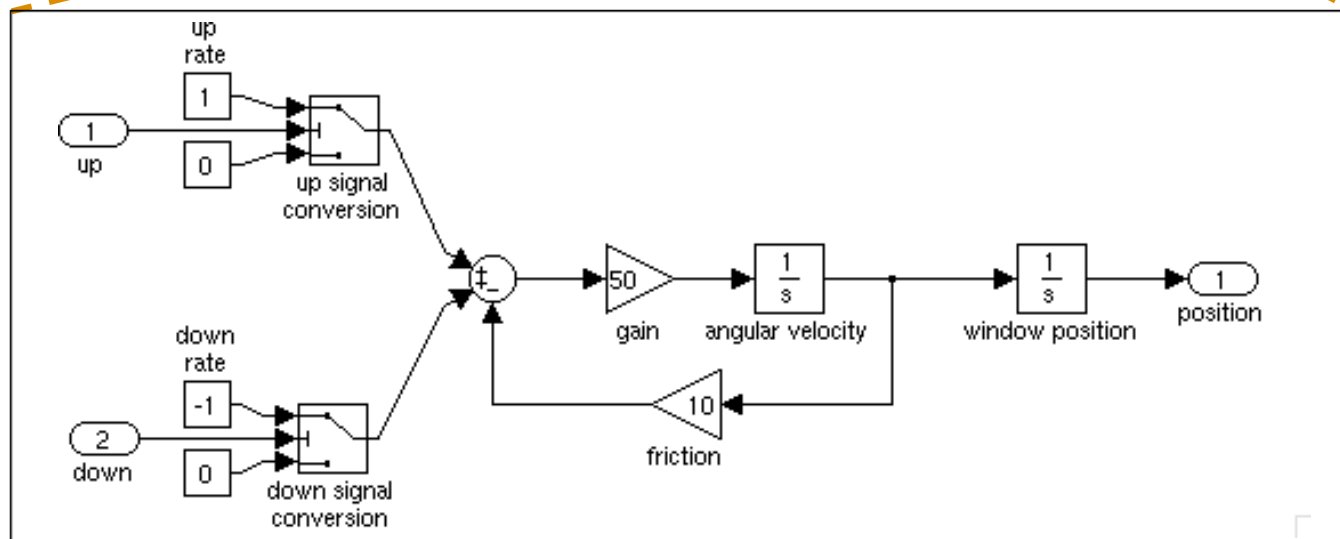
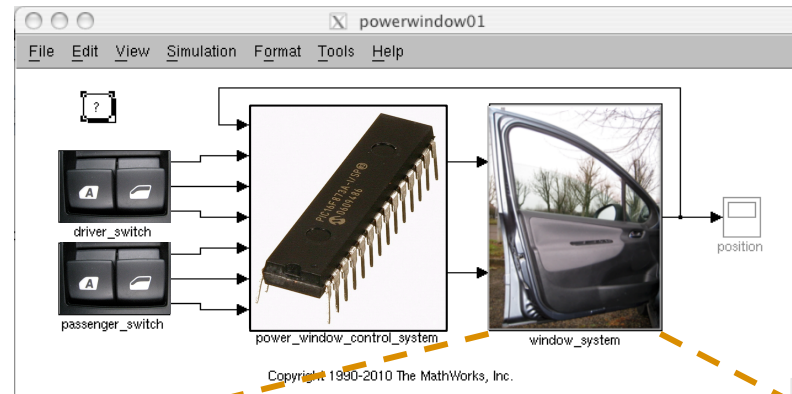
Raffinement de modèles



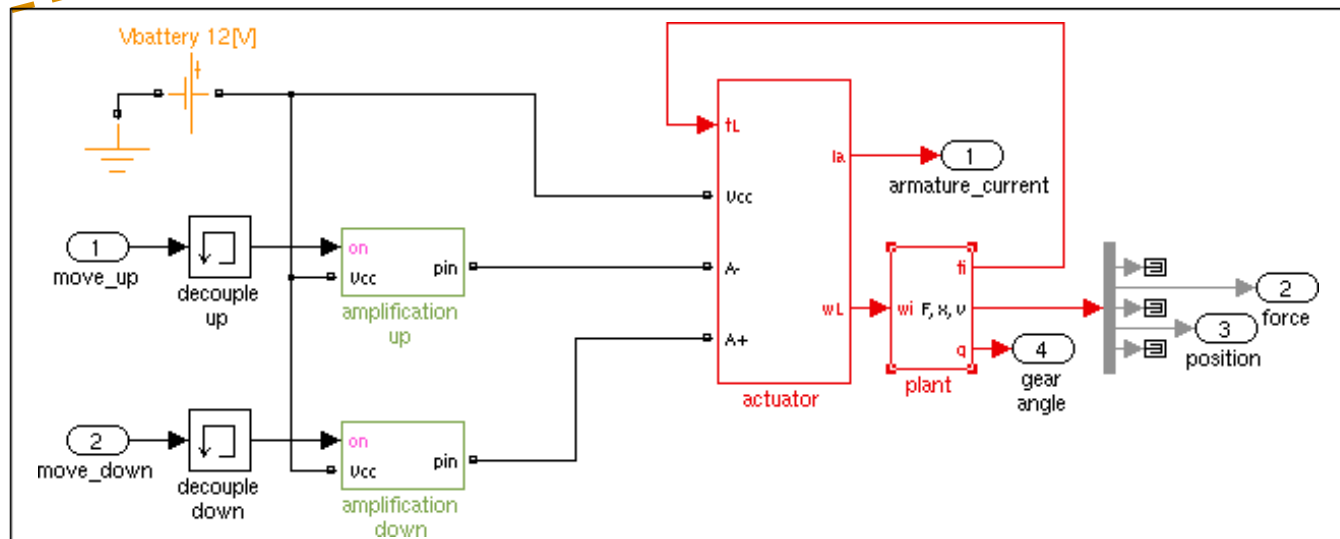
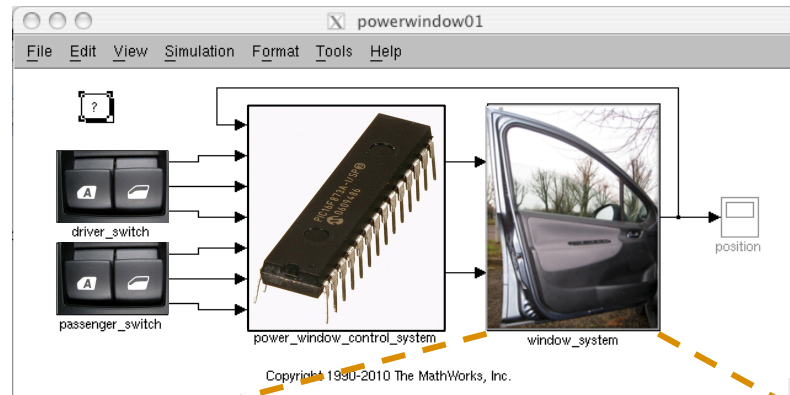
Exemple : Lève-vitre



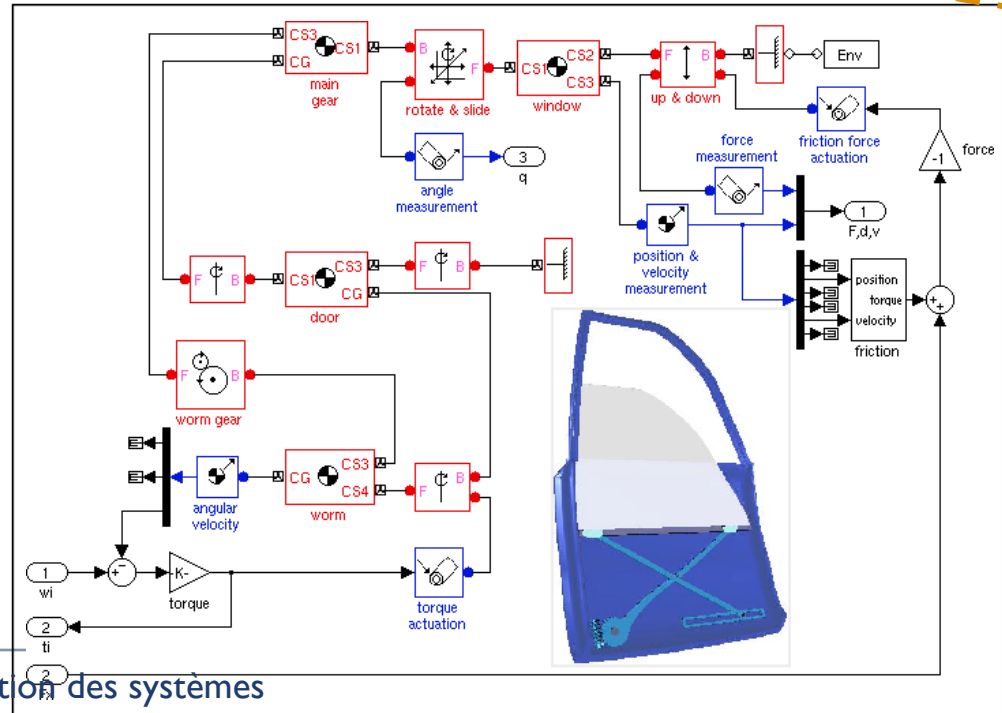
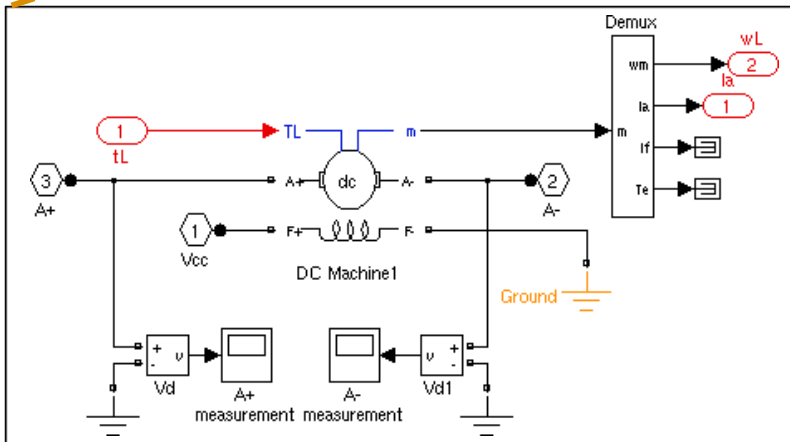
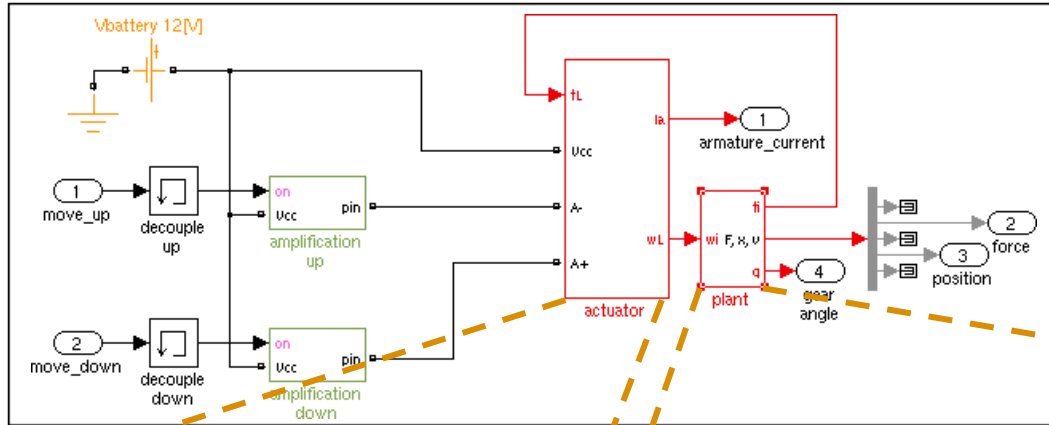
Exemple : Lève-vitre, modèle simplifié



Exemple : Lève-vitre, modèle détaillé



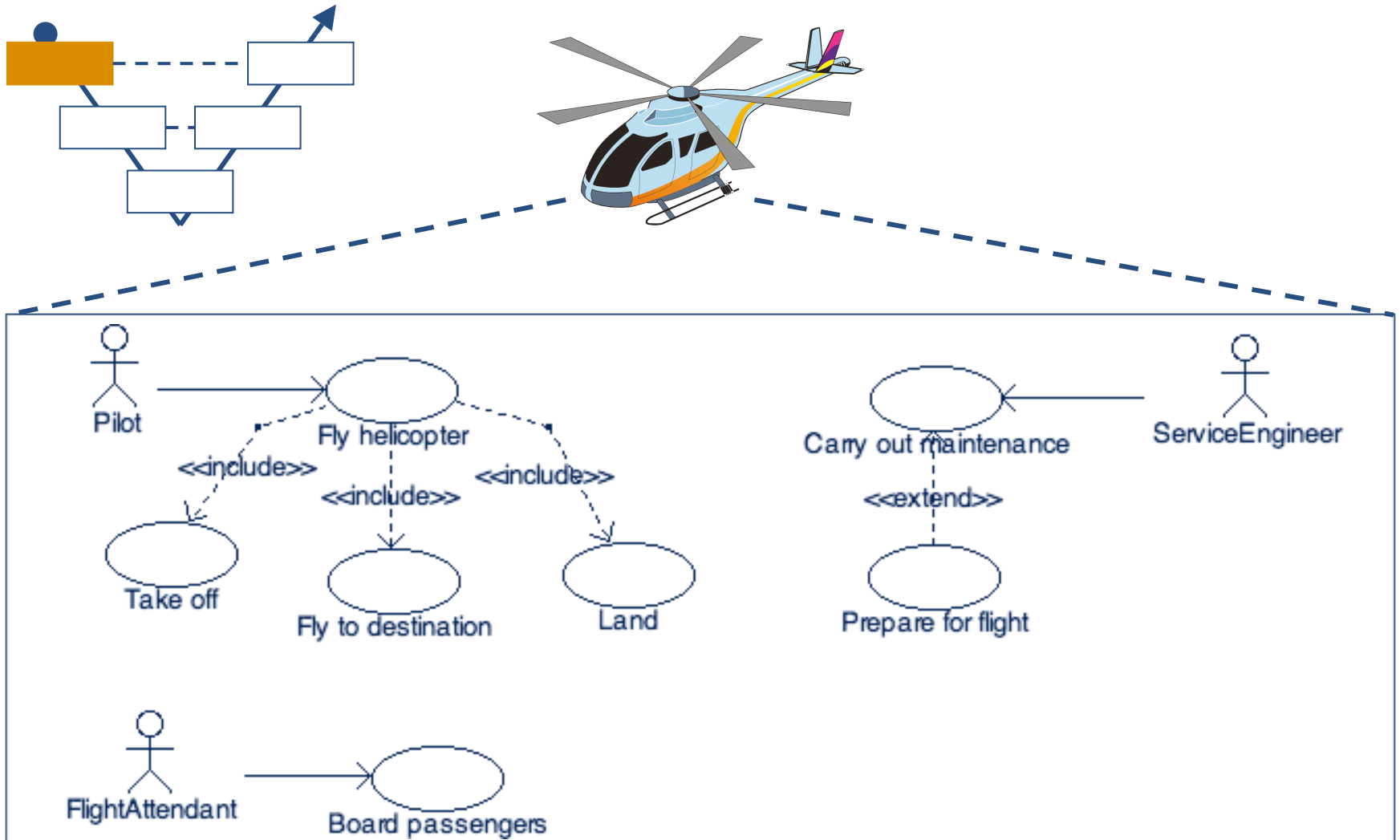
Exemple : Lève-vitre, modèle détaillé (suite)



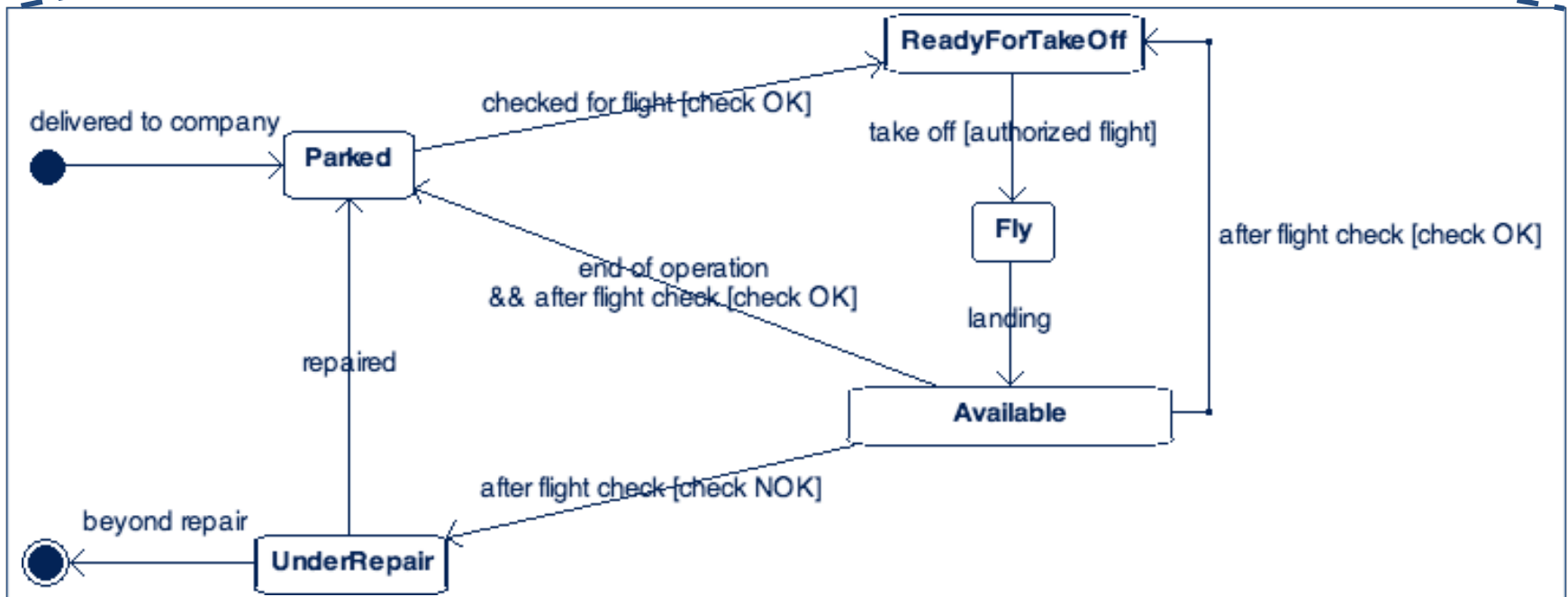
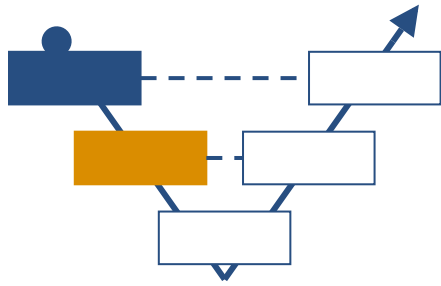
Pourquoi plusieurs langages ?

1. Contexte et rappels
2. Décomposition en sous-systèmes
3. Raffinement de modèles
4. Activités du cycle de conception
5. Vues non fonctionnelles
6. En résumé

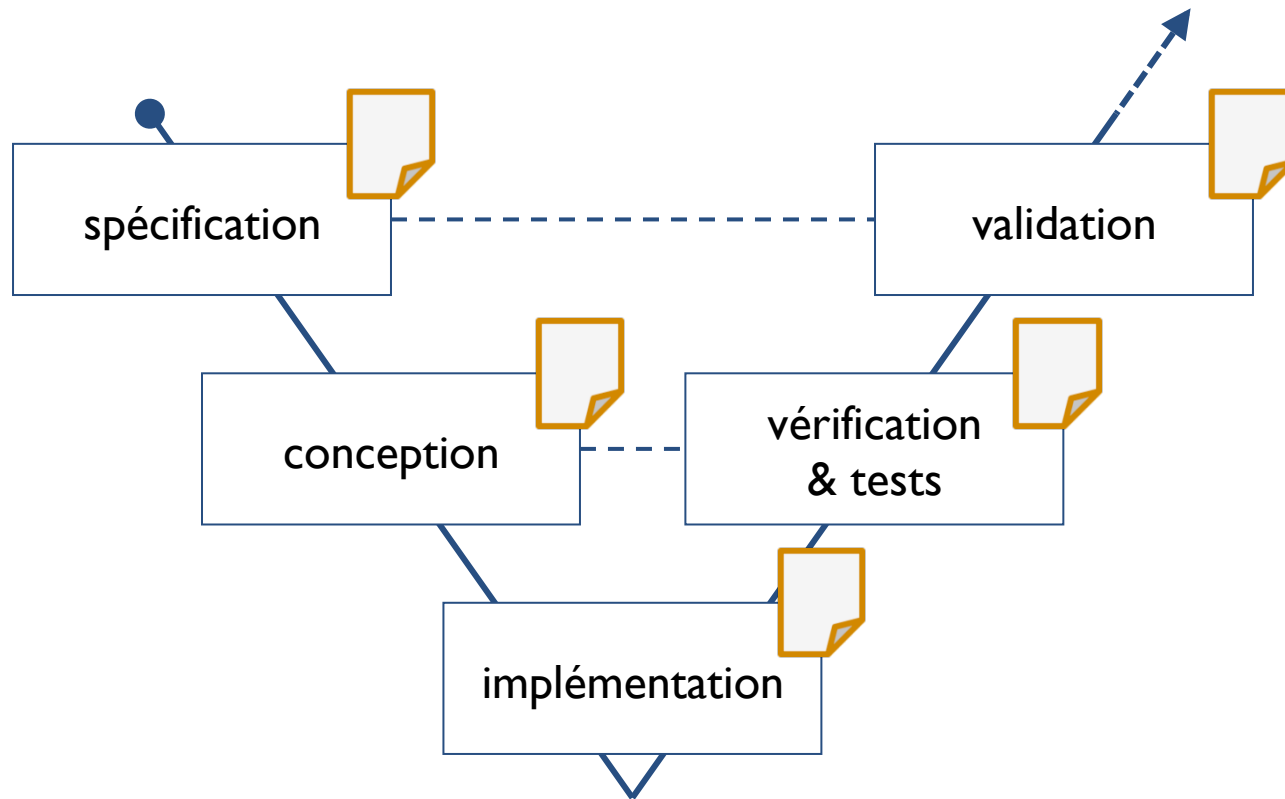
Pourquoi plusieurs langages ?



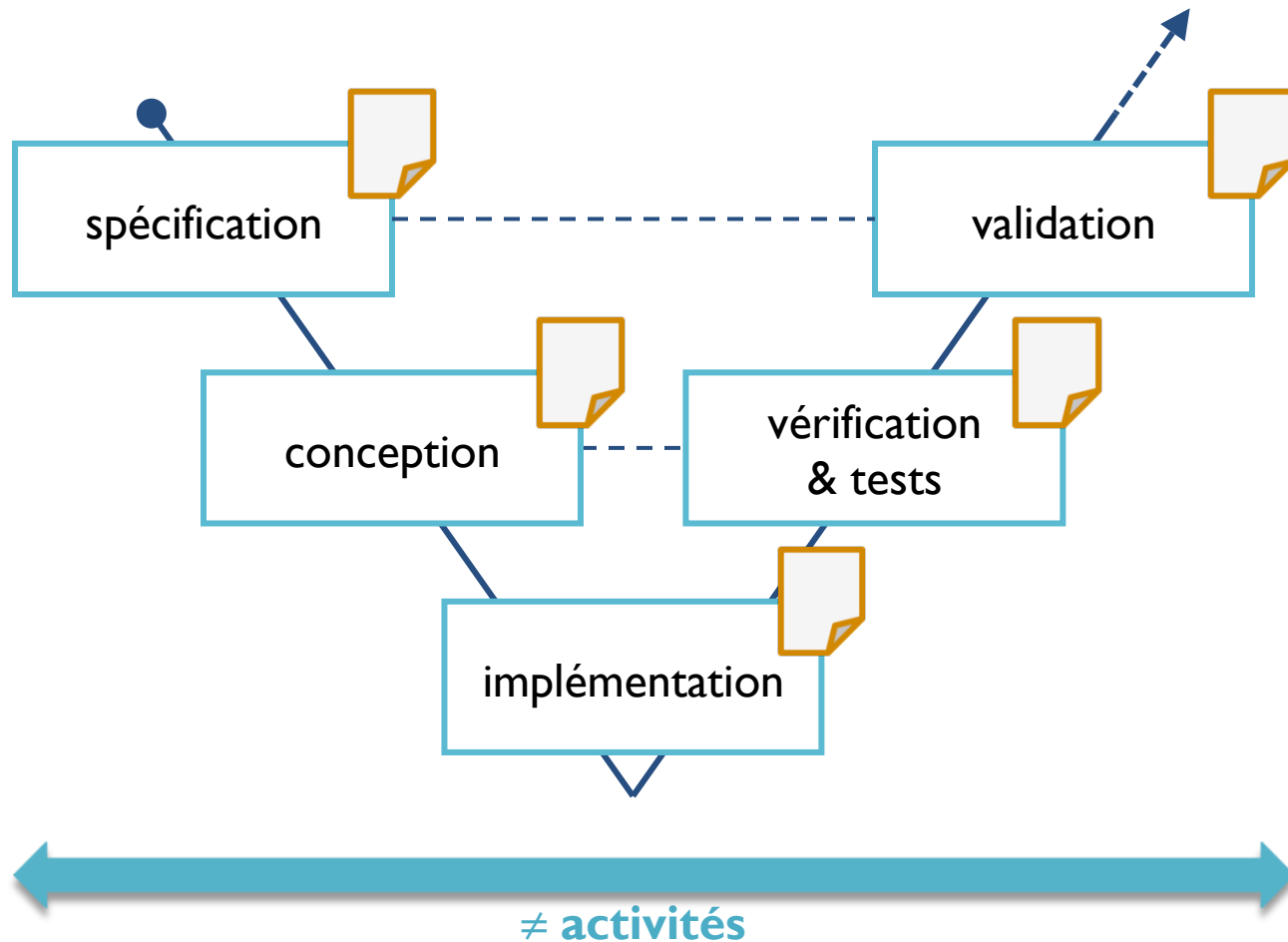
Pourquoi plusieurs langages ?



Activités du cycle



Activités du cycle

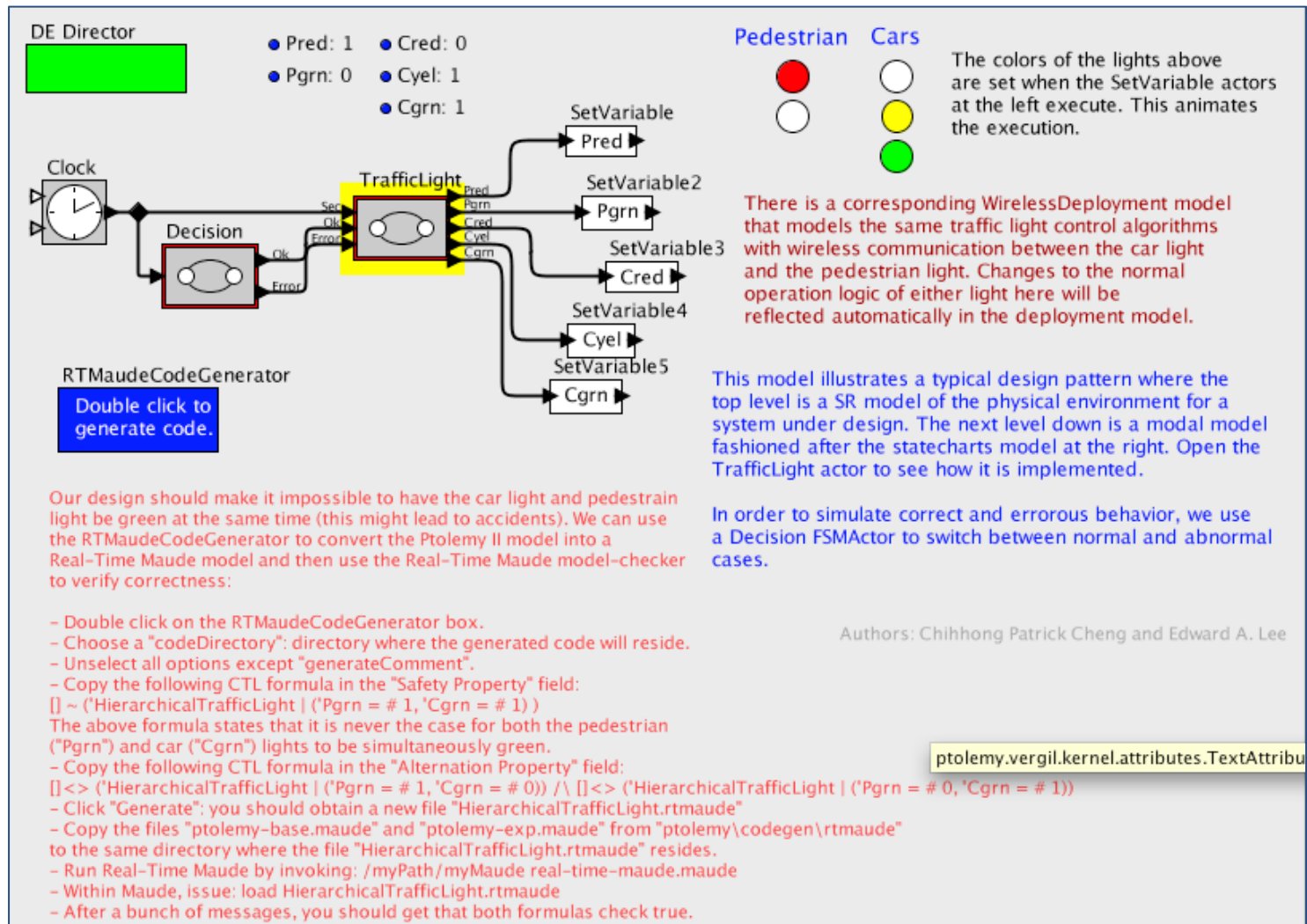


Exemple : Feux tricolores



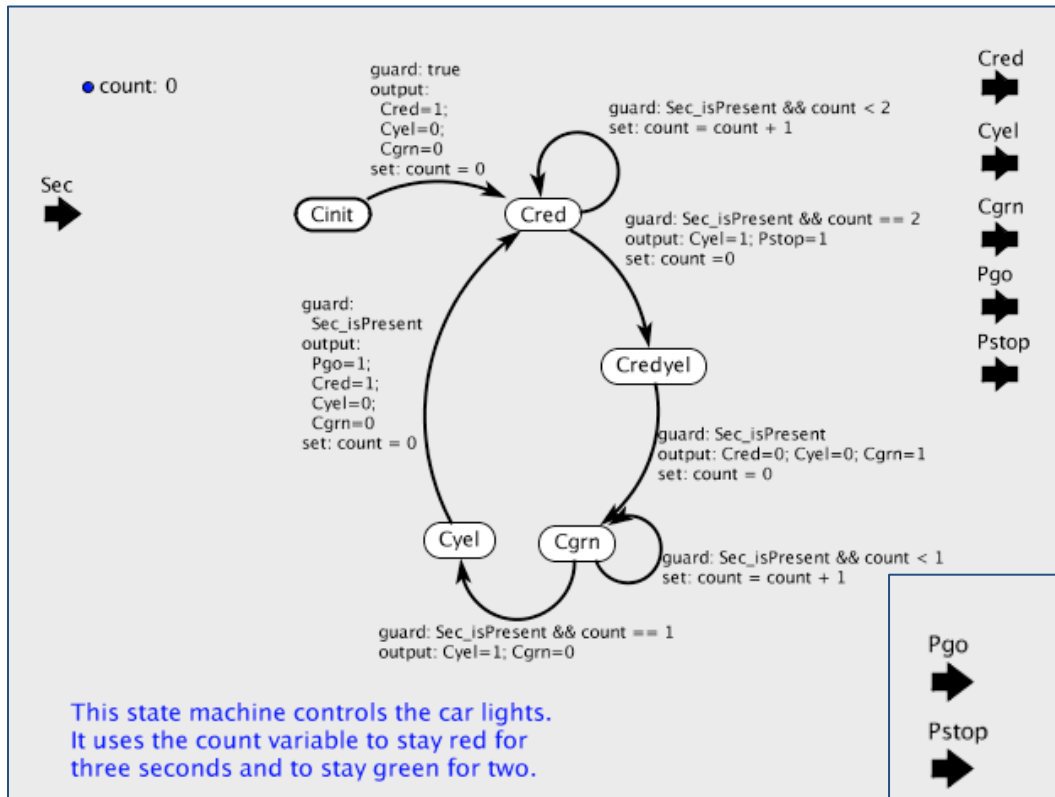
Exemple :

Feux tricolores, simulation

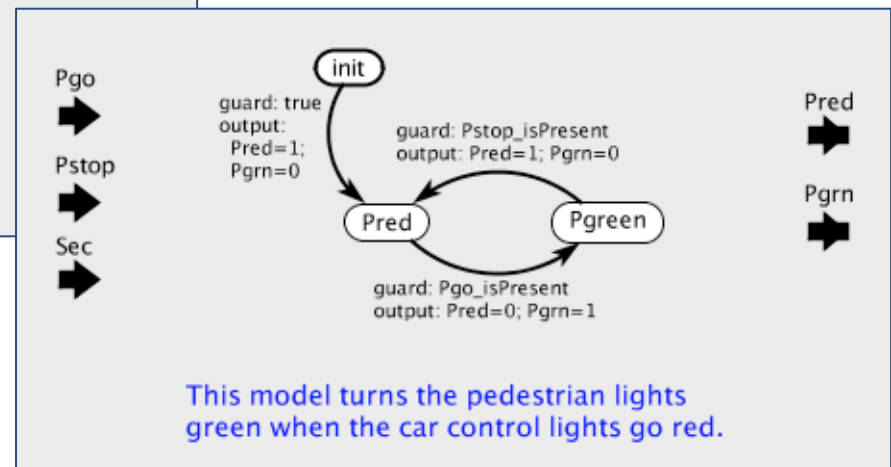


Exemple :

Feux tricolores, automates



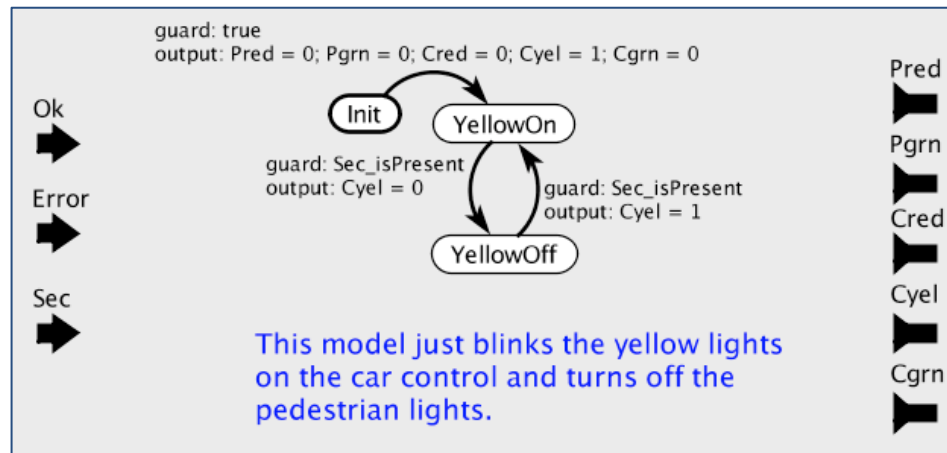
Fonctionnement
normal



Exemple :

Feux tricolores, automates (suite)

Fonctionnement
dégradé



Exemple :

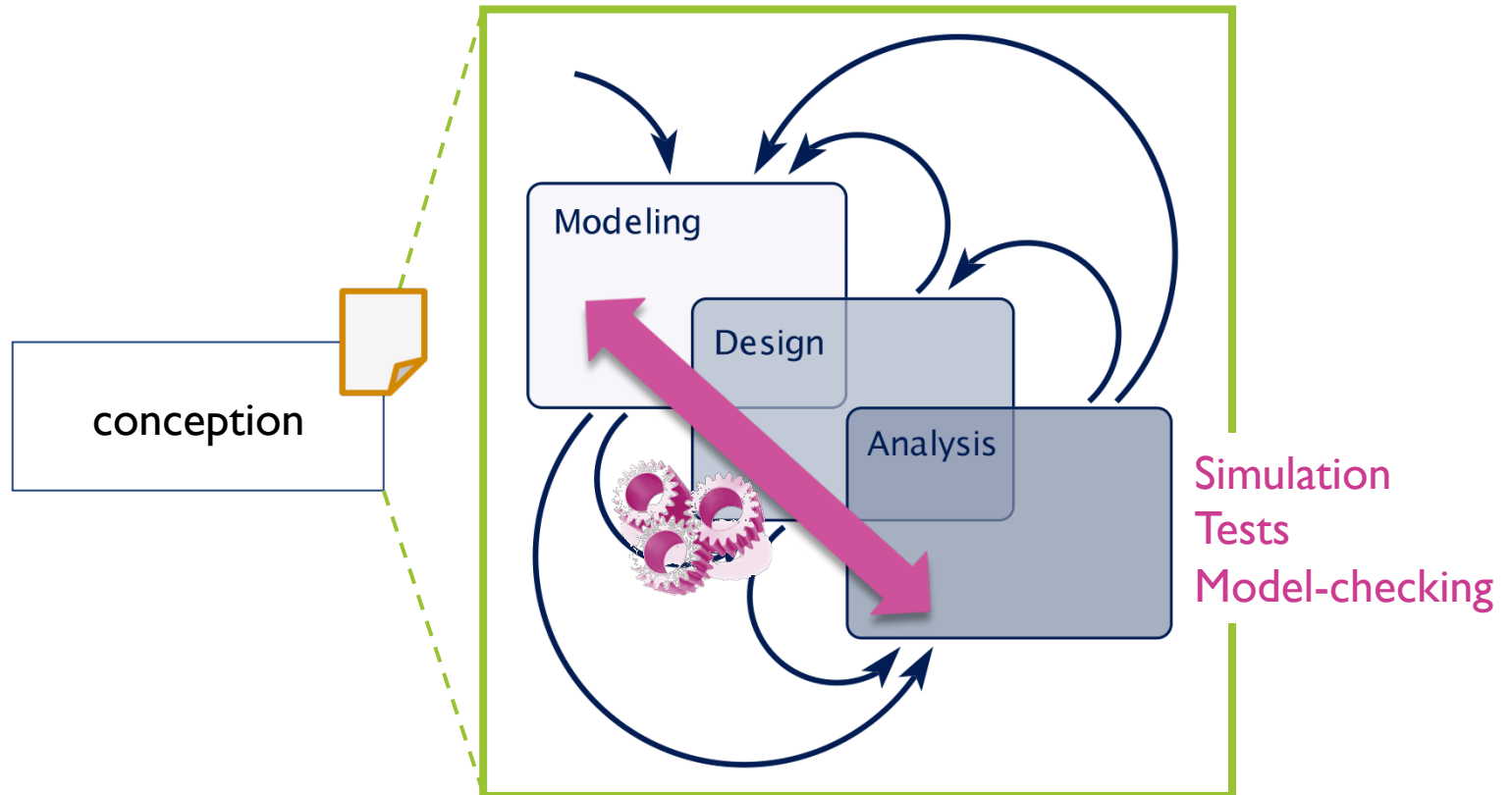
Feux tricolores, model checking

- ▶ **Vérification de modèle** avec Maude et la Logique Temporelle
- ▶ Propriété de **sûreté**
 - ▶ « Les feux pour les voitures et pour les piétons ne sont jamais simultanément au vert. »
 - ▶ $\Box \sim ('HierarchicalTrafficLight \mid ('Pgrn = \# 1, 'Cgrn = \# 1))$
- ▶ Propriété de **vivacité**
 - ▶ « Il existera toujours un moment où le feu pour les piétons sera vert et le feu pour les voitures ne sera pas vert.
Et il existera toujours un moment où le feu pour les voitures sera vert et le feu pour les piétons ne sera pas vert. »
 - ▶ $'HierarchicalTrafficLight : ($
 $\Box \langle \rangle (this \mid ('Pgrn = \# 1, 'Cgrn = \# 0))$
 $\wedge$
 $\Box \langle \rangle (this \mid ('Pgrn = \# 0, 'Cgrn = \# 1)))$



Transformations automatiques

- Objectif = généralisation de « WYPIWYE » [G. Berry]

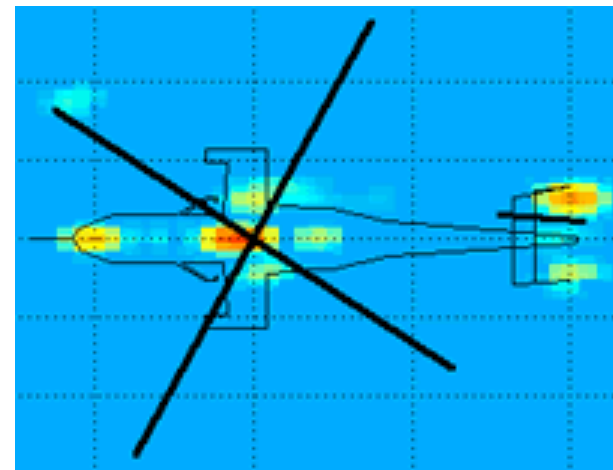
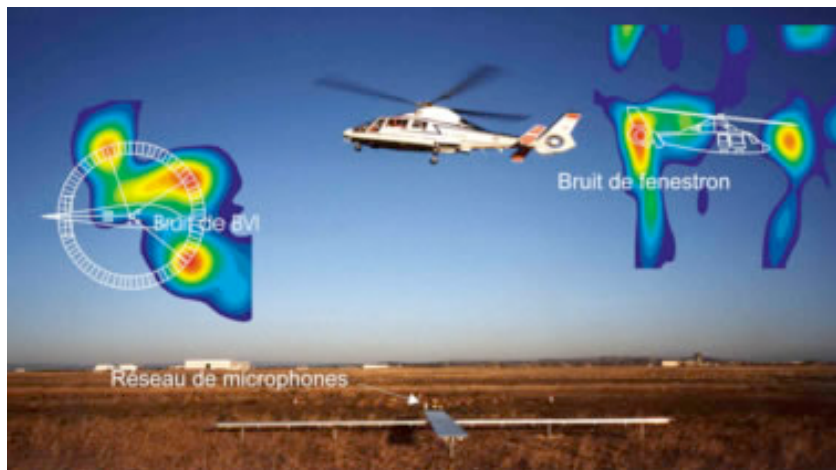


Pourquoi plusieurs langages ?

1. Contexte et rappels
2. Décomposition en sous-systèmes
3. Raffinement de modèles
4. Activités du cycle de conception
5. Vues non fonctionnelles
6. En résumé

Pourquoi plusieurs langages ?

- Spécification d'un hélicoptère furtif =
il vole + il est silencieux + il est indétectable au radar

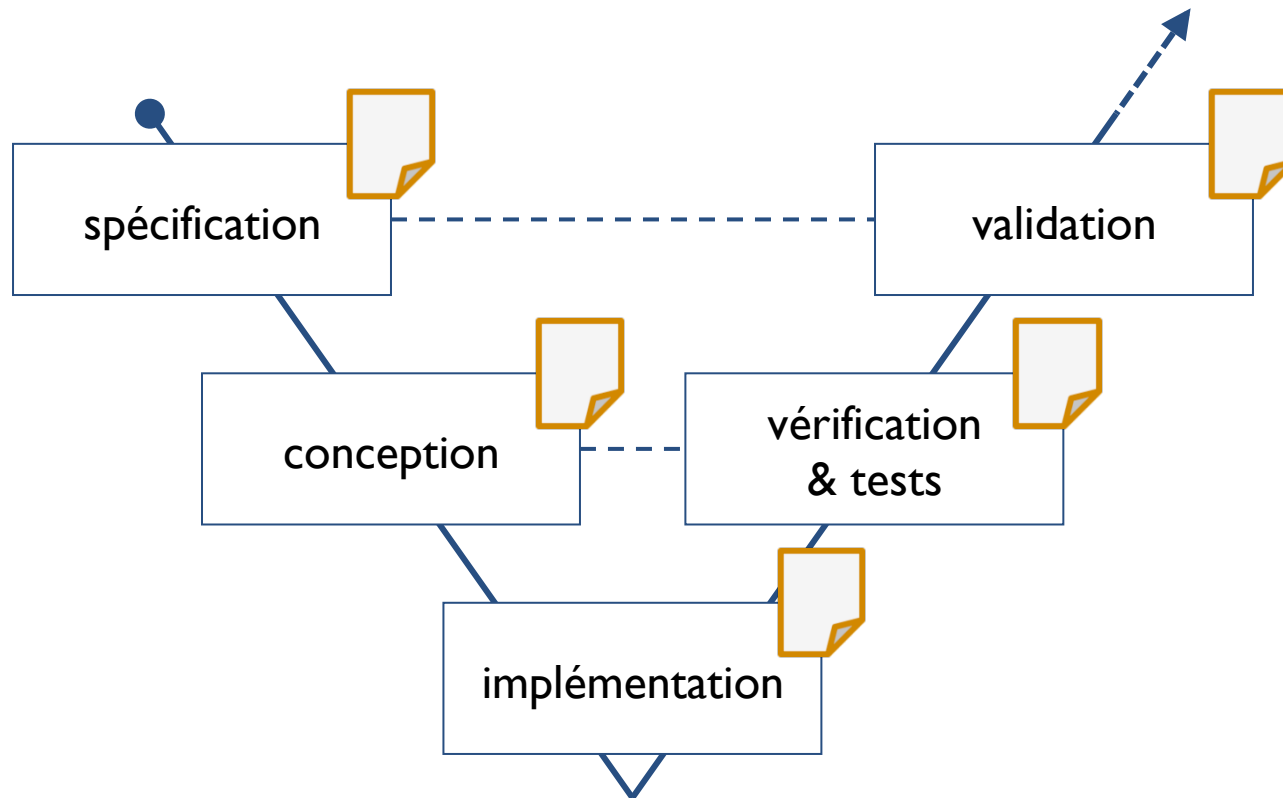


Pourquoi plusieurs langages ?

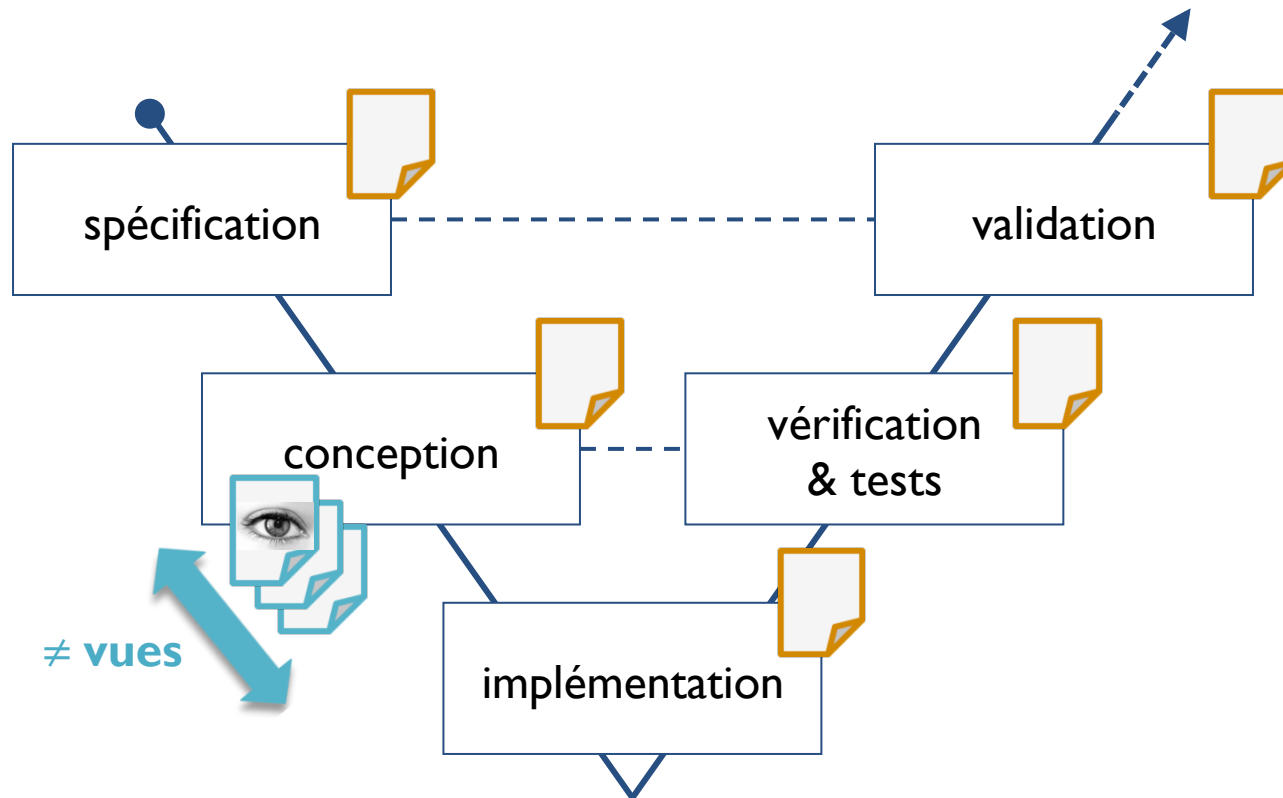
- ▶ Spécification d'une **voiture** = **elle roule** +
 - ▶ Emission de CO2
 - ▶ Consommation carburant
 - ▶ Confort acoustique
 - ▶ ...
- ▶ Spécification d'un **site web** = **il rend un service** +
 - ▶ Temps de réponse
 - ▶ Disponibilité
 - ▶ Intégrité
 - ▶



Propriétés non fonctionnelles

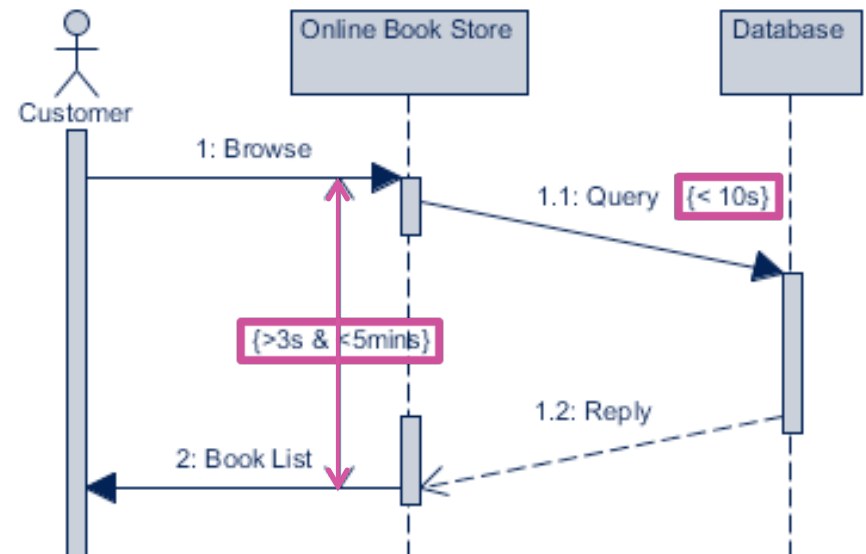
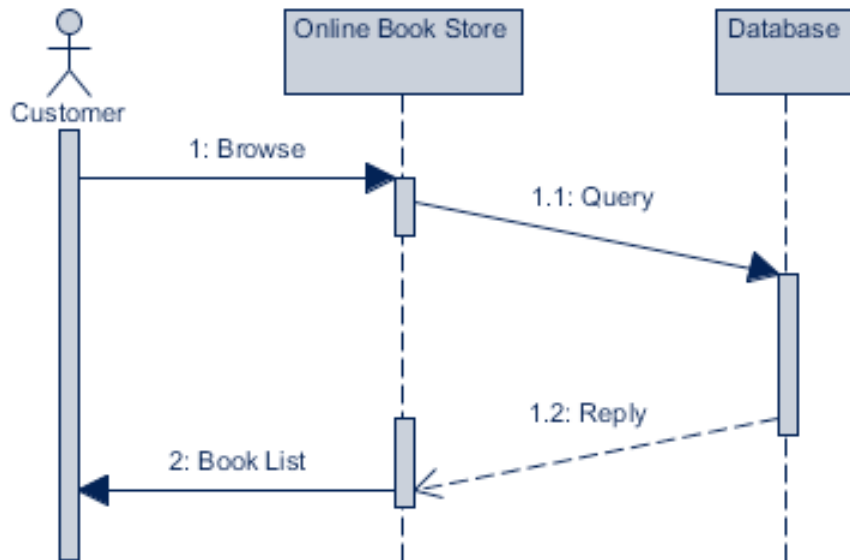


Propriétés non fonctionnelles



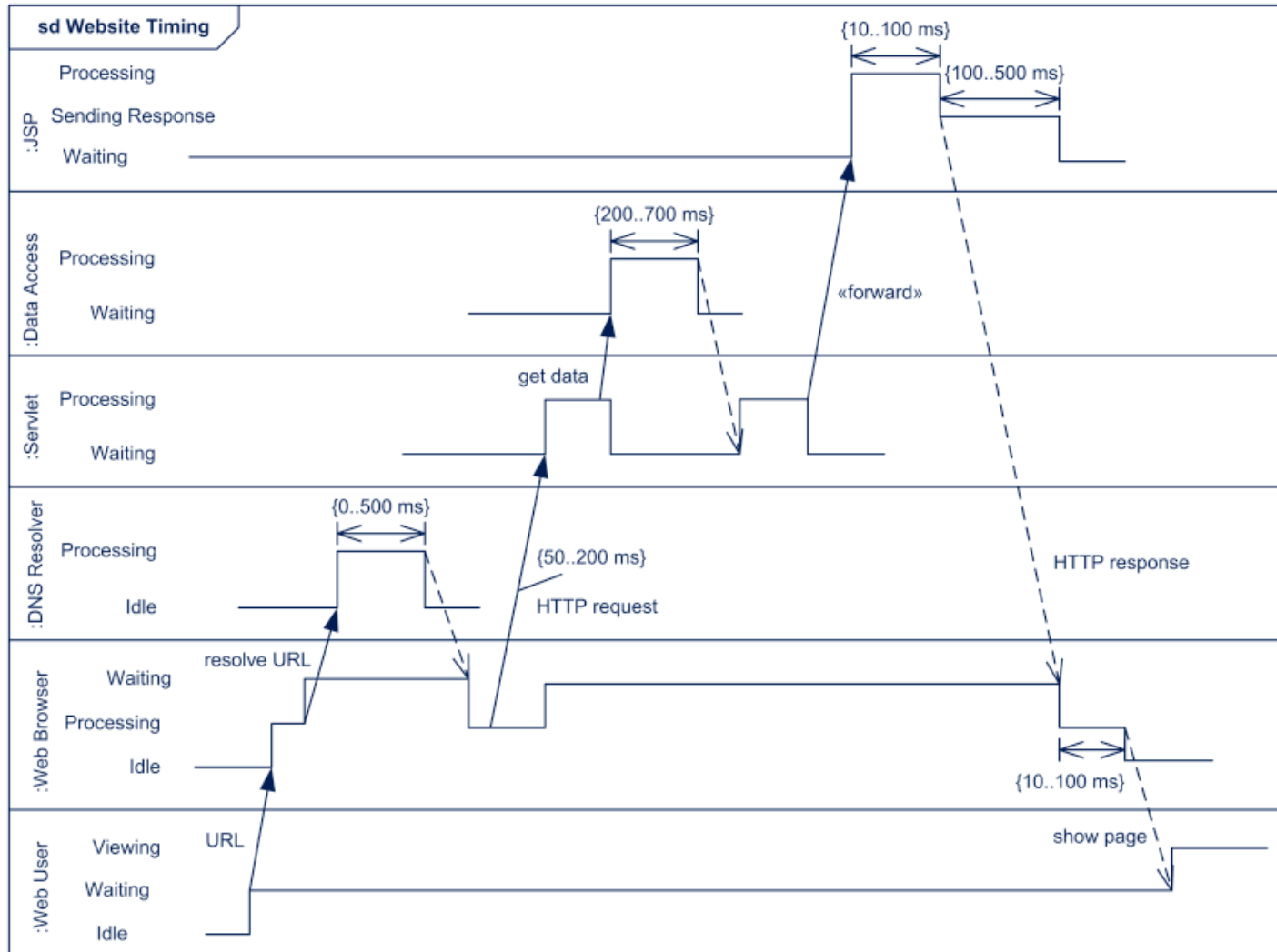
Exemple :

Temps de réponse d'un site web



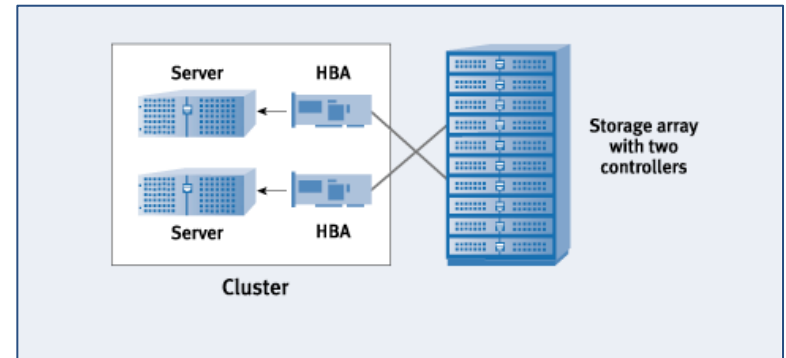
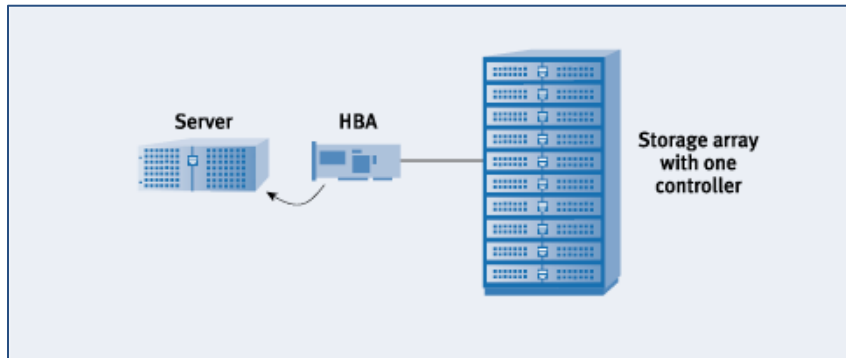
Exemple :

Temps de réponse d'un site web (2)

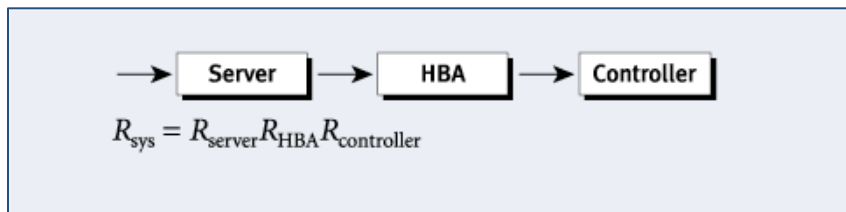


Exemple : Fiabilité d'un système de stockage

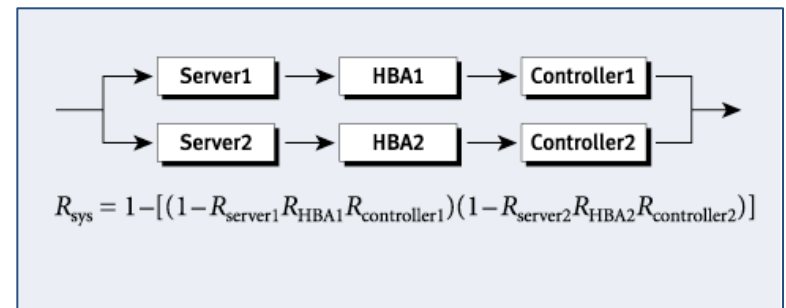
- Architecture en connexion directe
- Architecture en cluster



- Diagramme de fiabilité



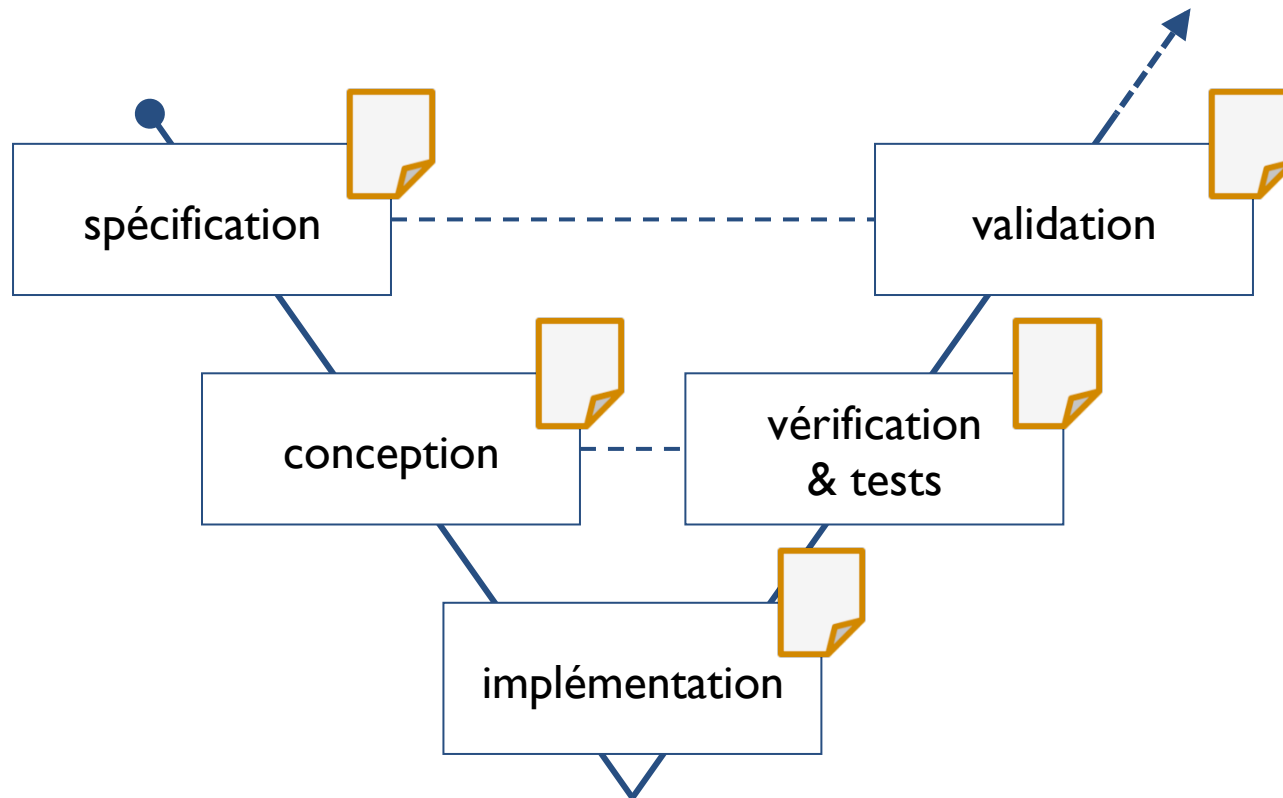
- Diagramme de fiabilité



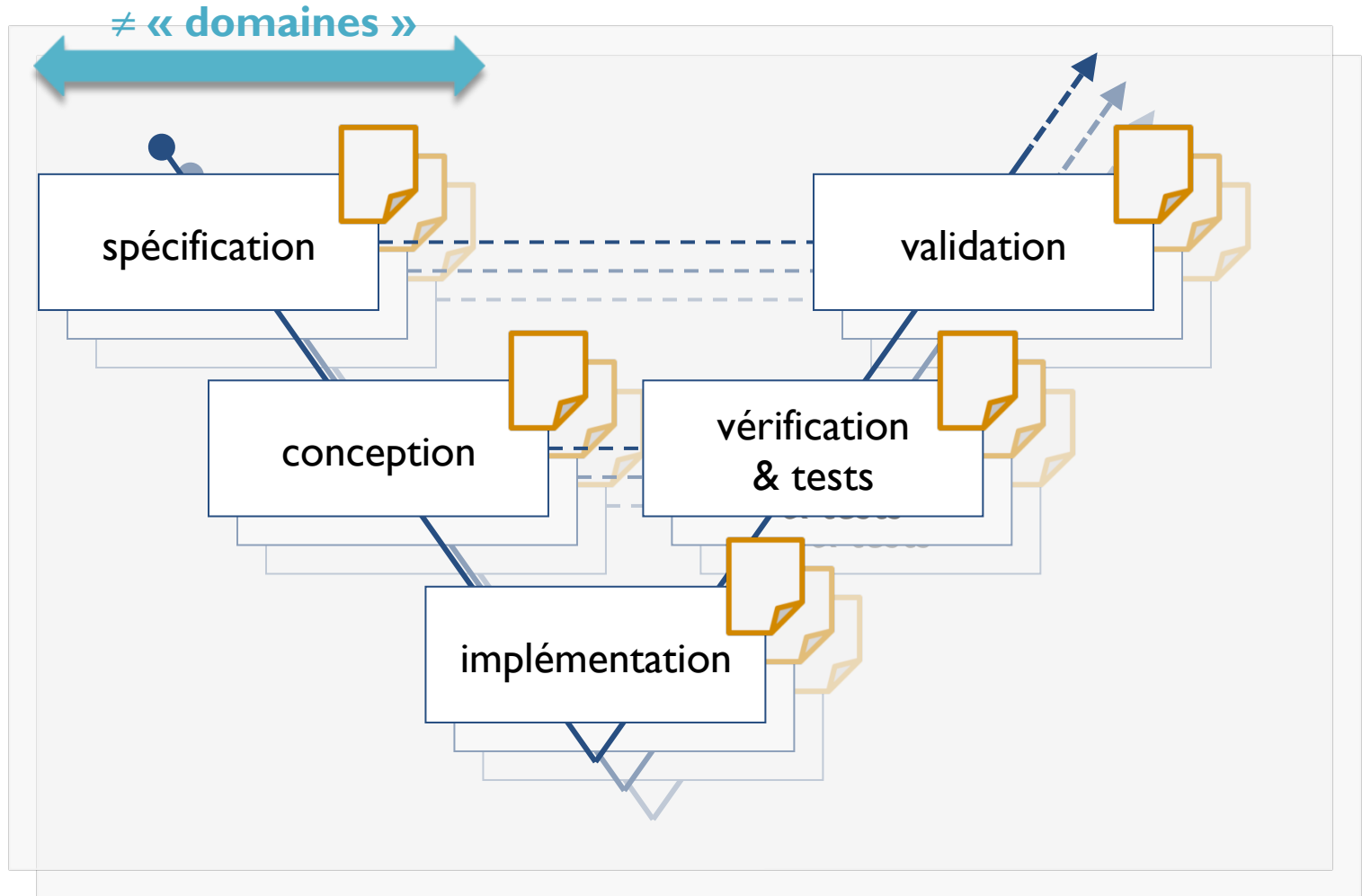
Pourquoi plusieurs langages ?

1. Contexte et rappels
2. Décomposition en sous-systèmes
3. Raffinement de modèles
4. Activités du cycle de conception
5. Vues non fonctionnelles
6. En résumé

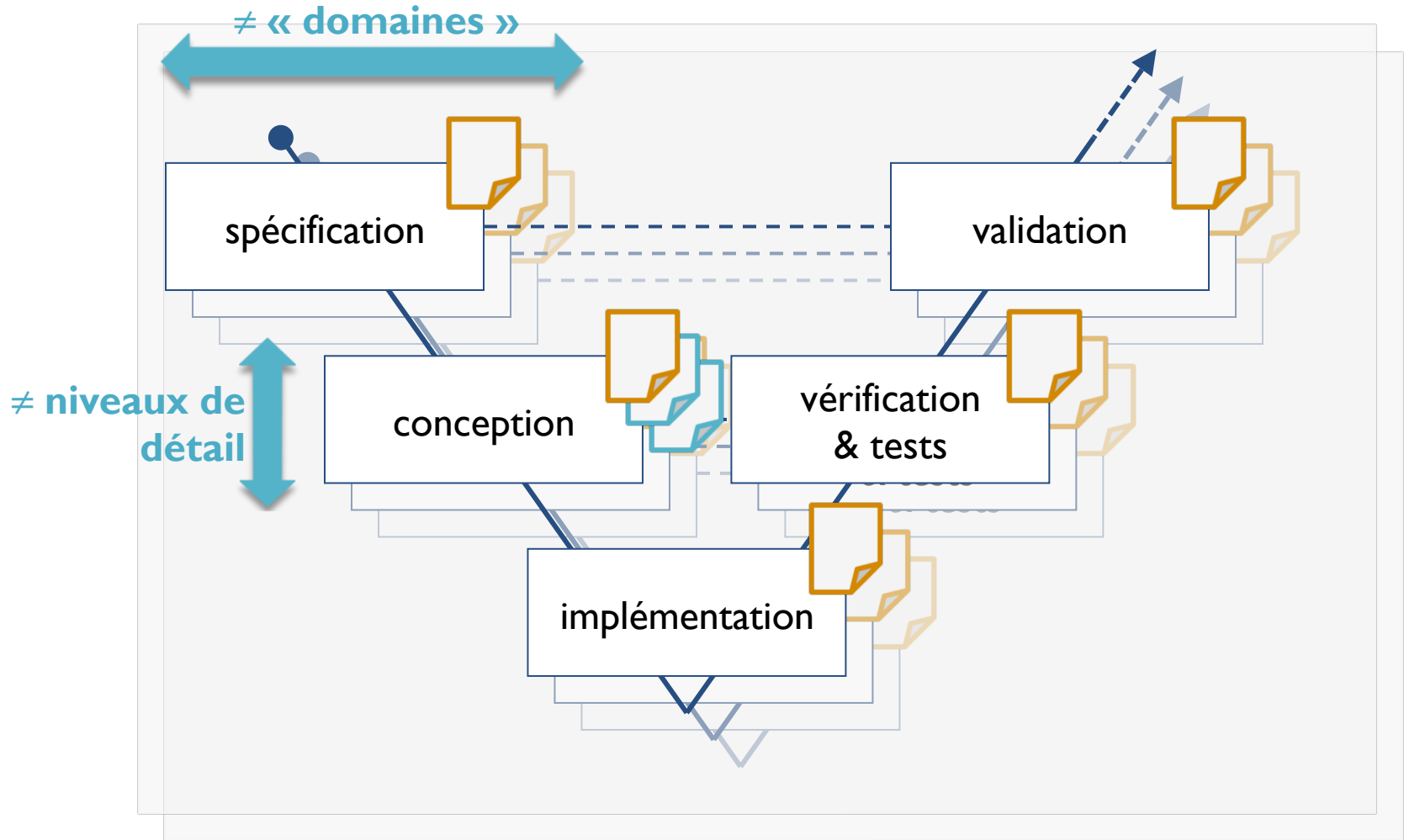
En résumé



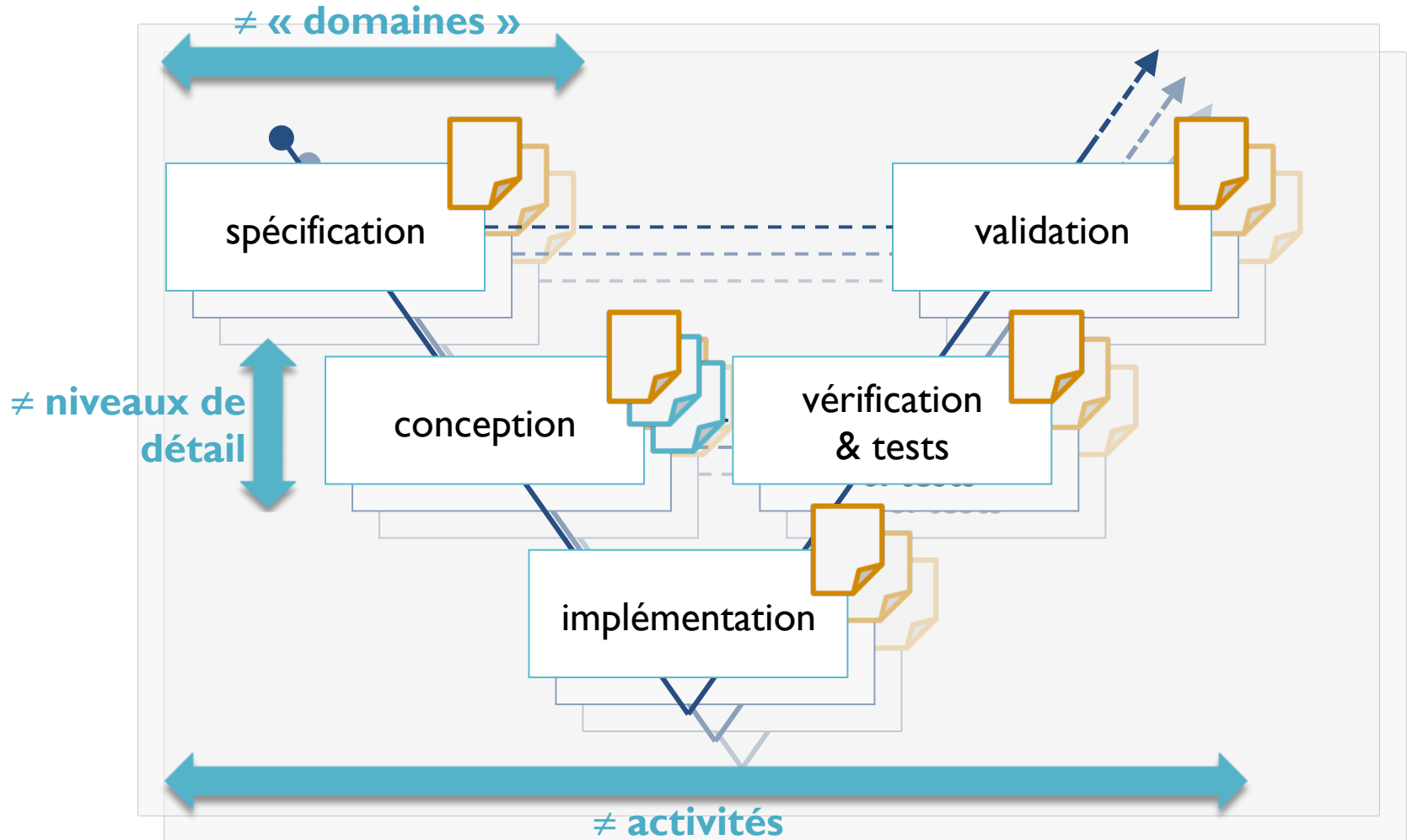
En résumé



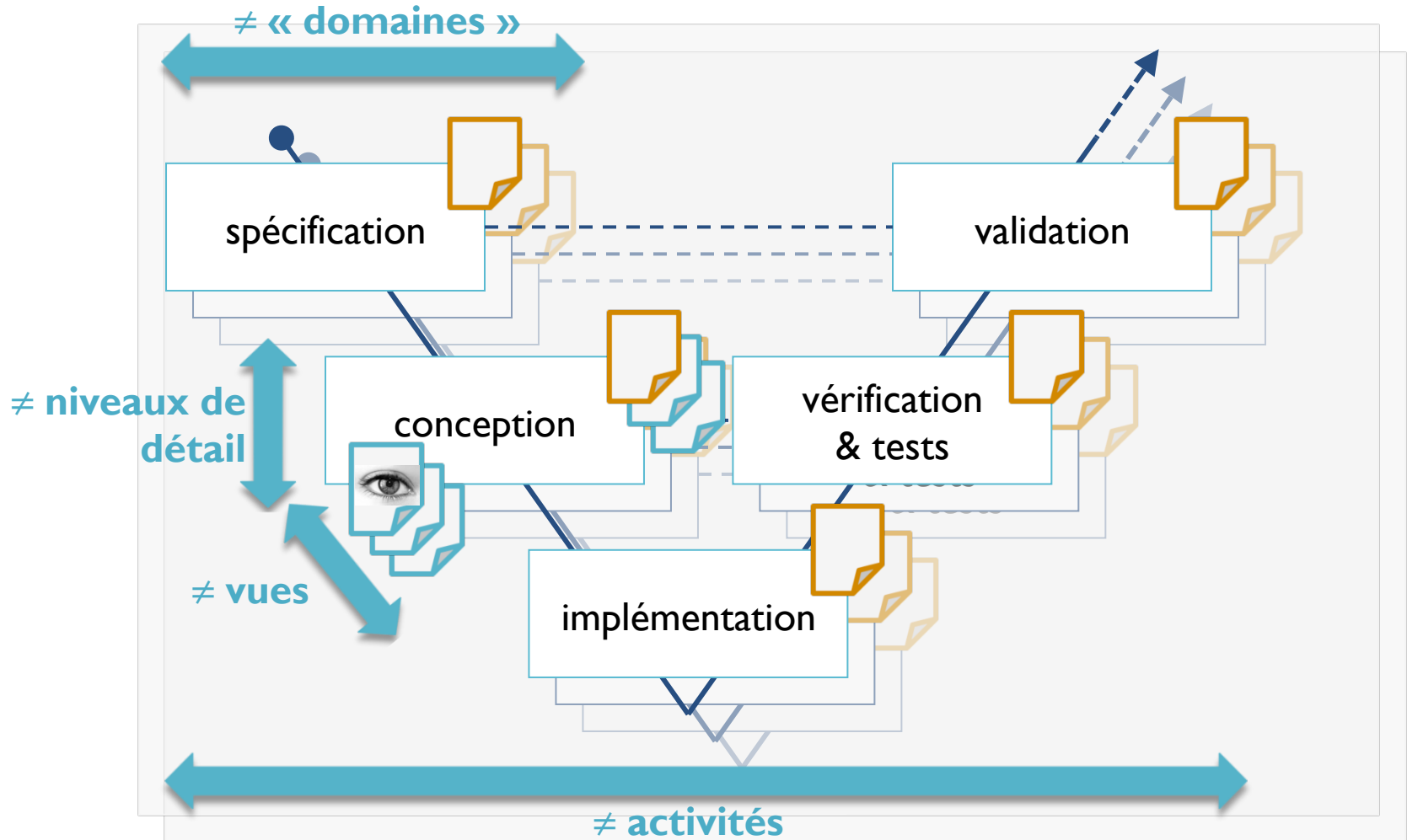
En résumé



En résumé



En résumé



Sources & bibliographie

- ▶ TheMathWorks – Matlab / Simulink – modèles de démonstration
- ▶ U. Berkeley, EECS, CHES – Ptolemy II – modèles de démonstration
- ▶ Edward A. Lee and Sanjit A. Seshia,
Introduction to Embedded Systems, A Cyber-Physical Systems Approach,
<http://LeeSeshia.org>, ISBN 978-0-557-70857-4, 2011
- ▶ Berkeley course EECS 149 « Introduction to Embedded Systems »,
Edward A. Lee and Sanjit A. Seshia
- ▶ « Modélisation des systèmes », Cours de troisième année, option SI,
Supélec, Frédéric Boulanger