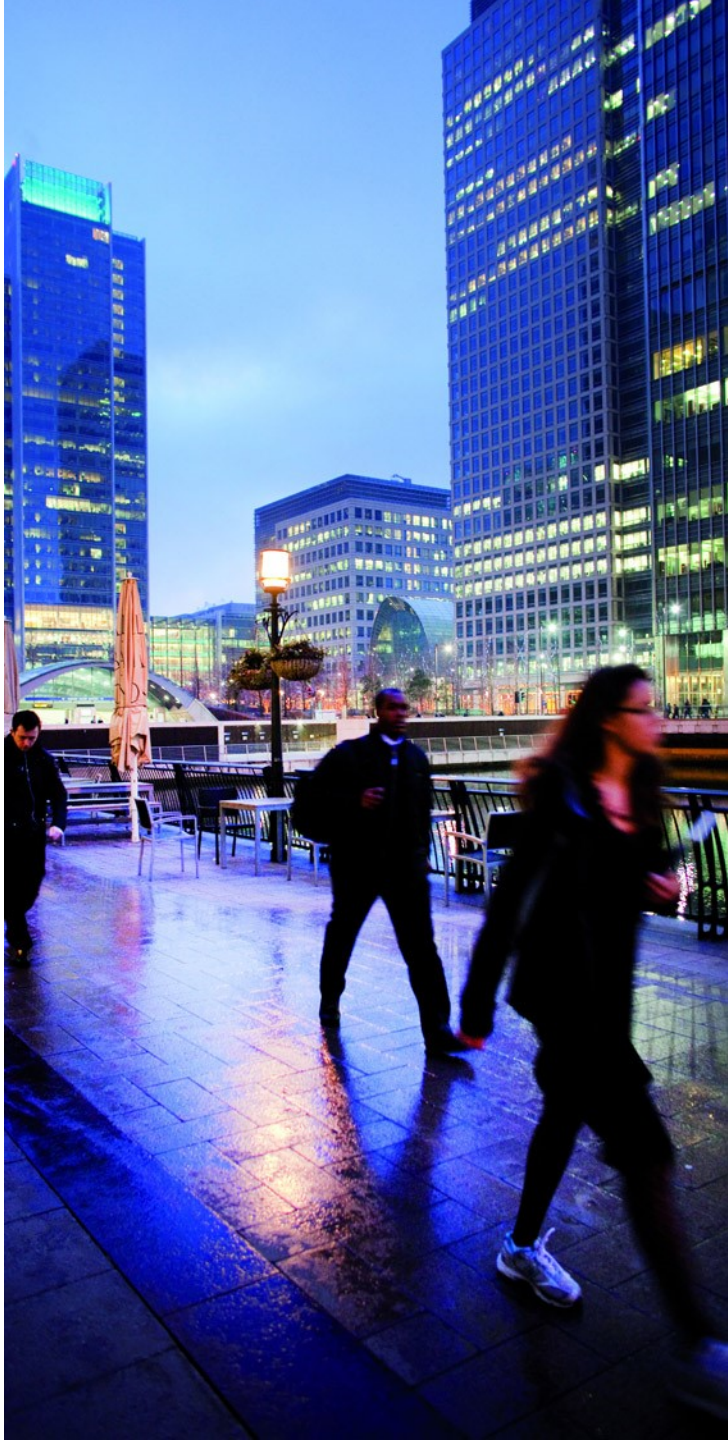
Landing Gear System (*LGS*) Spécification Formelle

GT Ingénierie des Exigences
Toulouse

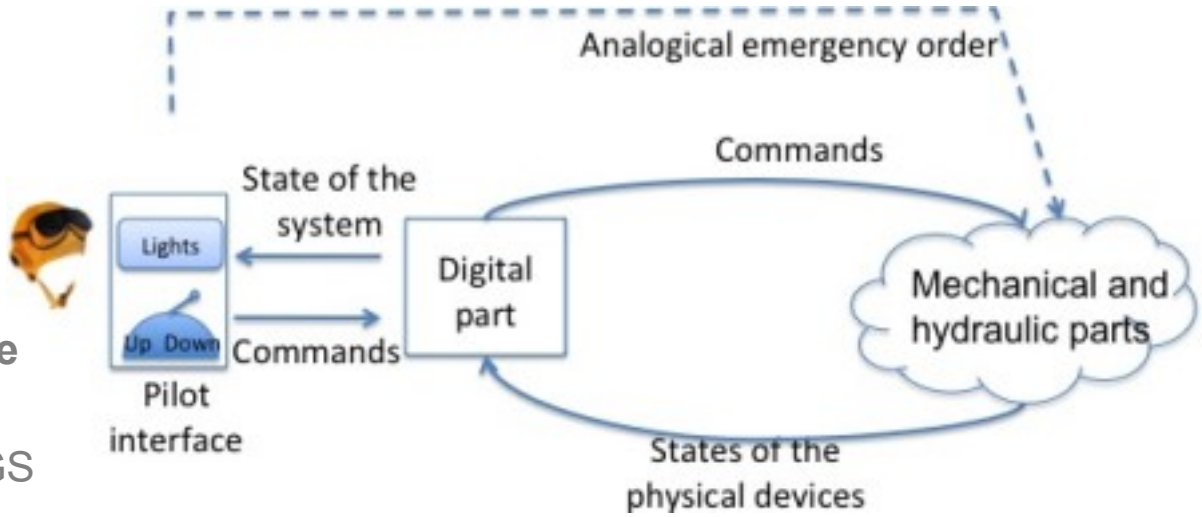
N. Thuy - EDF R&D
1er Octobre 2018



Sommaire

Introduction

- **Article de Frédéric Boniol et Virginie Wiels**
  - ONERA (Toulouse)
- **Présentation d'un système de trains d'atterrissage**
  - Landing Gear System, ou LGS



- **Spécification détaillée de son logiciel de contrôle en langage naturel**
- **Défi : modélisation formelle de cette spécification**
- **Mon objectif : comment valider la spécification et sa modélisation ?**
  - La spécification en langage naturel est-elle vraiment exempte de défauts ?
    - La modélisation formelle serait alors un vain exercice
  - La modélisation formelle représente-t-elle vraiment ce qu'on veut ?
  - Comment éviter de s'engager dans un processus de développement et de V&V du logiciel long, complexe et coûteux mais vicié à la base ?

7 Péchés Capitaux de la Spécification des Exigences

- **Inadéquation**
  - Elément inapproprié dans certaines, voire toutes circonstances
- **Silence**
  - Caractéristique nécessaire à laquelle ne correspond aucune exigence
- **Ambiguité**
  - Elément pouvant être compris différemment par les lecteurs visés

Défauts d'expression
vs.
Défauts de compréhension

- **Surspécification**
  - Elément appartenant au domaine de la solution et non du problème
- **Flou**
  - Elément exprimé de façon telle qu'il n'y a pas de critère objectif de satisfaction
- **Contradiction**
  - Eléments ne pouvant pas être satisfaits ensemble
- **Lettre au Père Noël**
  - Elément ou ensemble d'éléments désirés mais pas forcément réalisables ou réalistes

Retour d'Expérience

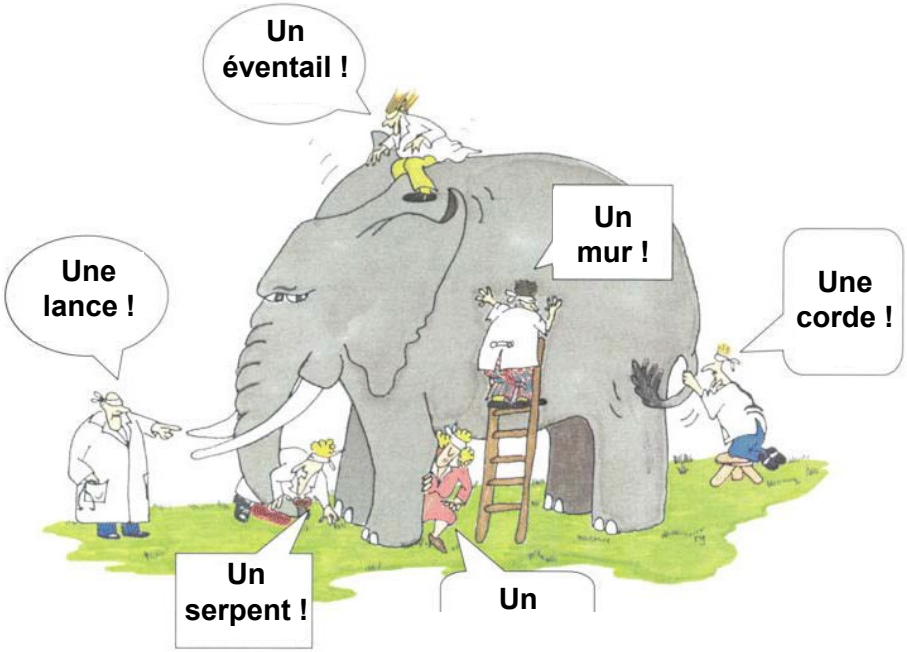
- **De très gros efforts pour le développement et la V&V du logiciel**
  - Gros volume de normes internationales
  - La spécification des exigences est le point de départ et est souvent tenue pour acquise
- **Peu de normes, guides et méthodes pour la qualité de la spécification des exigences comportementales**
  - Un certain nombre pour le logiciel et les systèmes embarqués
  - Pas grand chose pour les systèmes physiques
- **L'expérience montre qu'elle laisse souvent à désirer**
  - Aussi bien pour le logiciel et les systèmes embarqués que pour les systèmes physiques
  - Dans tous les secteurs industriels, même pour des systèmes soumis à de très fortes exigences de sûreté de fonctionnement
    - EPRI TR 1016731 Operating Experience Insights on Common-Cause Failures in Digital Instrumentation and Control Systems : Les événements causés par des spécifications inadéquates sont nettement plus fréquents que ceux causés par des erreurs de programmation
    - Certains accidents aériens

Sommaire

- Introduction
- **L'ingénierie des systèmes techniques complexes**
- La modélisation du *LGS*
- Conclusions

Ingénierie des Systèmes Techniques Complexes

- ... ne pouvant être compris dans tous les aspects requis par un seul individu, une seule équipe, une seule discipline



études prospectives avant-projets

construction, rénovation, remplacement

cahier des charges
spécification
architecture
conception détaillée
test unitaire
intégration
validation
mise en service

exploitation

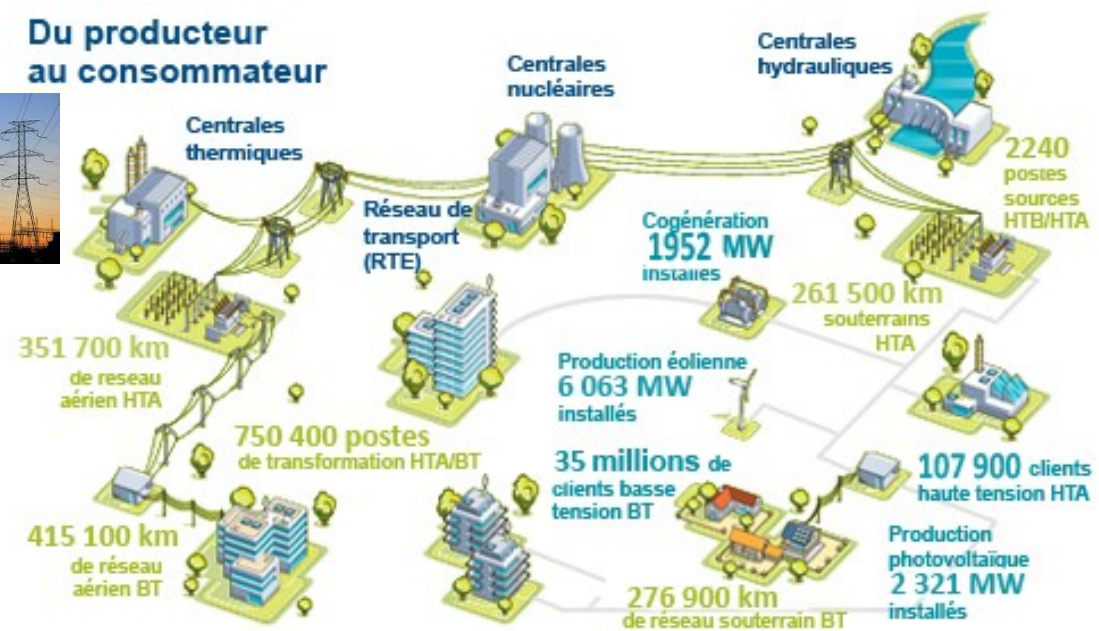
optimisation | *diagnostics, prognostics*

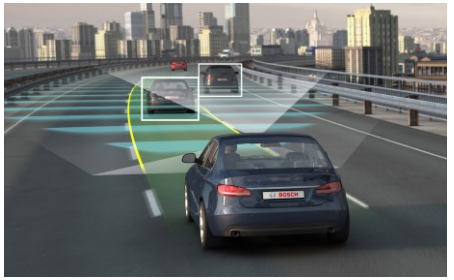
arrêts pour révision déconstruction

Systèmes Cyber-Physiques (Socio-Cyber-Physiques)
Systèmes de Systèmes

- **Beaucoup de systèmes complexes sont des systèmes socio-cyber-physiques (SSCP)**
  - Aspects humains et sociétaux
  - Informatique & communication
  - Physique, géométrie et topologie, connexions, ...

- **Certains sont aussi des systèmes de systèmes (SdS)**
  - Plusieurs SSCP en interaction ayant chacun leur propre cycle de vie

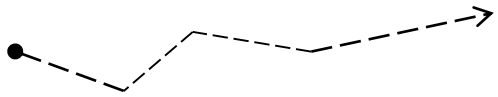




Modélisation des Phénomènes Dynamiques

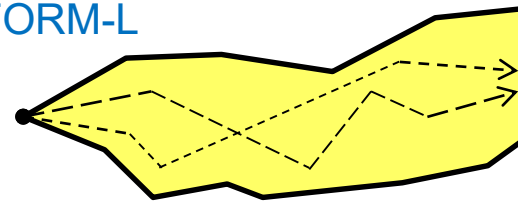
- **La modélisation informelle ou semi-formelle**
  - Ambigüe et souvent incomplète, peu d'outils facilitant la maîtrise de la complexité
  - Exemples de langages semi-formels : SADT, SysML

- **La modélisation formelle classique et déterministe**
  - A partir des conditions initiales et des conditions aux limites, un seul comportement possible
  - Exemples de langages impératifs : Modelica, langages à blocs fonctionnels, ...
  - Précise et détaillée ➔ seulement dans les phases aval de l'ingénierie système



Modèle Dynamique Classique

- **La modélisation formelle par contraintes**
  - Enveloppes des phénomènes attendus ou supposés : n'imposent pas de solution a priori
    - Physique, cyber, défaillances, finance, actions humaines, ...
  - Exemples de langages de contraintes : PSL, StimuLus, FORM-L
  - Bien adaptés pour spécifier les exigences, et pour certains, les hypothèses et les solutions préliminaires

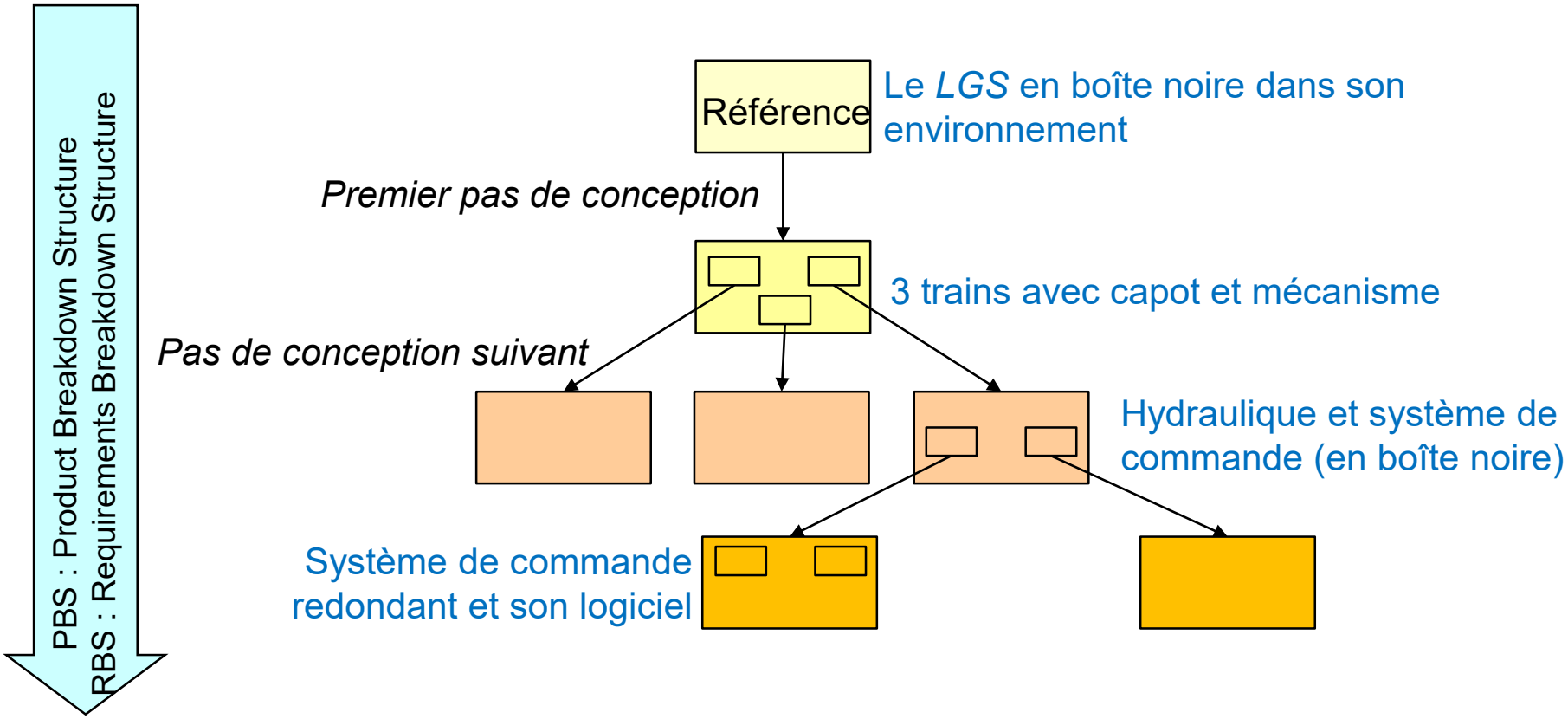


Modèle Dynamique par Contraintes

Principes Généraux

- **Replacer le système et ses aspects dynamiques dans leur contexte**
  - Expression des exigences sur le système et des hypothèses sur son environnement
  - L'environnement est composé d'autres systèmes, d'humains (agissant parfois selon des procédures), ou de l'environnement physique
- **Expression "naturelle" des exigences et hypothèses, puis modélisation formelle**
- **Modélisation des solutions au fur et à mesure de l'avancement de la conception**
- **Utilisation (massive) de la simulation pour vérifier chaque étape**
  - Que la modélisation reflète bien les intentions et la réalité
  - Que les solutions sont conformes aux exigences en amont, dans les conditions définies par les hypothèses
  - Vérification formelle quand elle est possible
- **Cette approche permet également**
  - Des AMDEC (Analyses des Modes de Défaillance, et de leurs Effets et Conséquences)
  - STPA (System Theoretical Process Analysis)
  - Des tests et des validations de la réalisation (Software, Hardware-in-the-Loop)

Raffinement Progressif des Solutions et des Modèles



Modèle de Référence

- **Base pour la vérification des solutions et de leurs modèles**
  - Le système et les entités de son environnement sont vus comme des boîtes noires

A. Identifier les entités de l'environnement interagissant avec le système

- Autres systèmes, acteurs humains, l'environment physique

B. Identifier les situations

- Etats du système, états des entités de l'environnement, buts opérationnels, transitions
- Sans oublier les situations anormales

C. Identifier et caractériser les flux

- Fluides, informations, événements
- Peuvent dépendre des situations (aggressions et invasions en situations anormales)

D. Modéliser les hypothèses que fait le système sur son environnement

- Peuvent aussi dépendre des situations

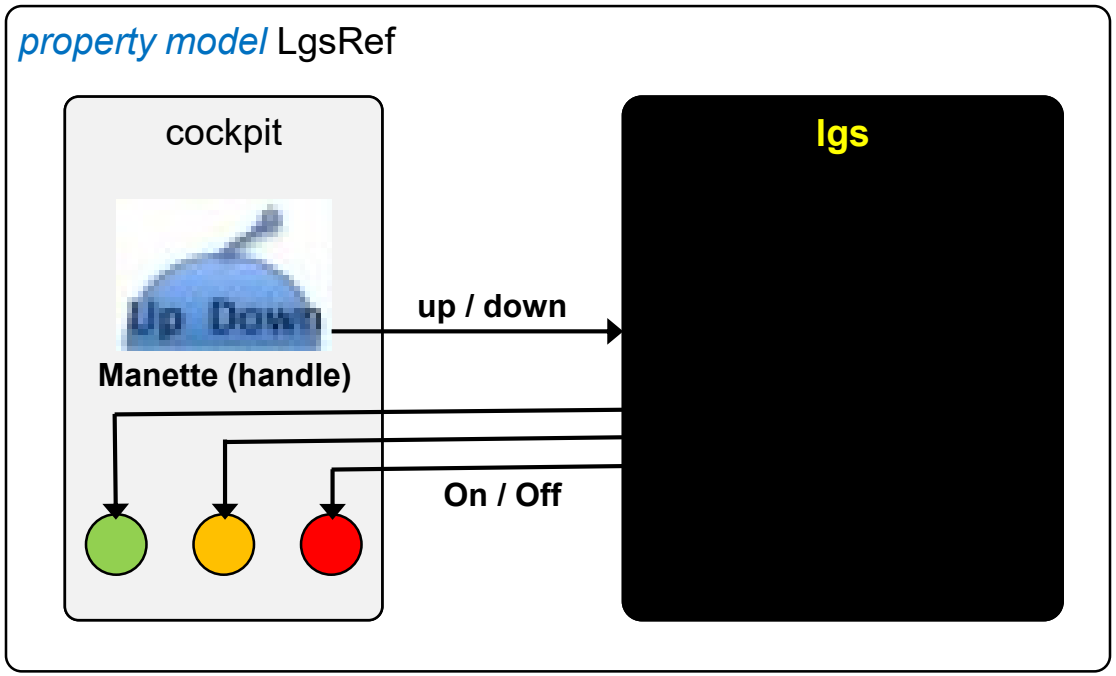
E. Modéliser les exigences essentielles requises du système

- Peuvent aussi dépendre des situations

Sommaire

- Introduction
- L'ingénierie des systèmes techniques complexes
- La modélisation du *LGS*
- Conclusions

LgsRef : Modèle de Référence du *LGS*



LgsRef : Propriétés et Exigences

Exigences Opérationnelles du *LGS*

- **Propriété désirée *pB1*: quand la manette passe en position *up*, si l'avion n'est pas au sol, le *LGS* doit être en position repliée dans les 15 s, sauf s'il y a eu contre ordre** (r12)

- **Propriétés similaires**
  - *pA1* pour le déploiement du *LGS*, (r11)
  - *pB2* pour le non déploiement intempestif (r22)
  - *pA2* pour le non repli intempestif (r21)

- **Exigence *pfd1* : la probabilité de ne pas satisfaire *pA1* ou *pB1* doit être inférieure à 10^{-4} à la sollicitation**
  - L'aspect probabiliste vise à éliminer les solutions "paresseuses", où on se contente d'allumer la lampe rouge

- **Exigence *reliability1* : moins de 10^{-3} déploiements ou replis intempestifs par an**

Exigences d'Information au Pilote

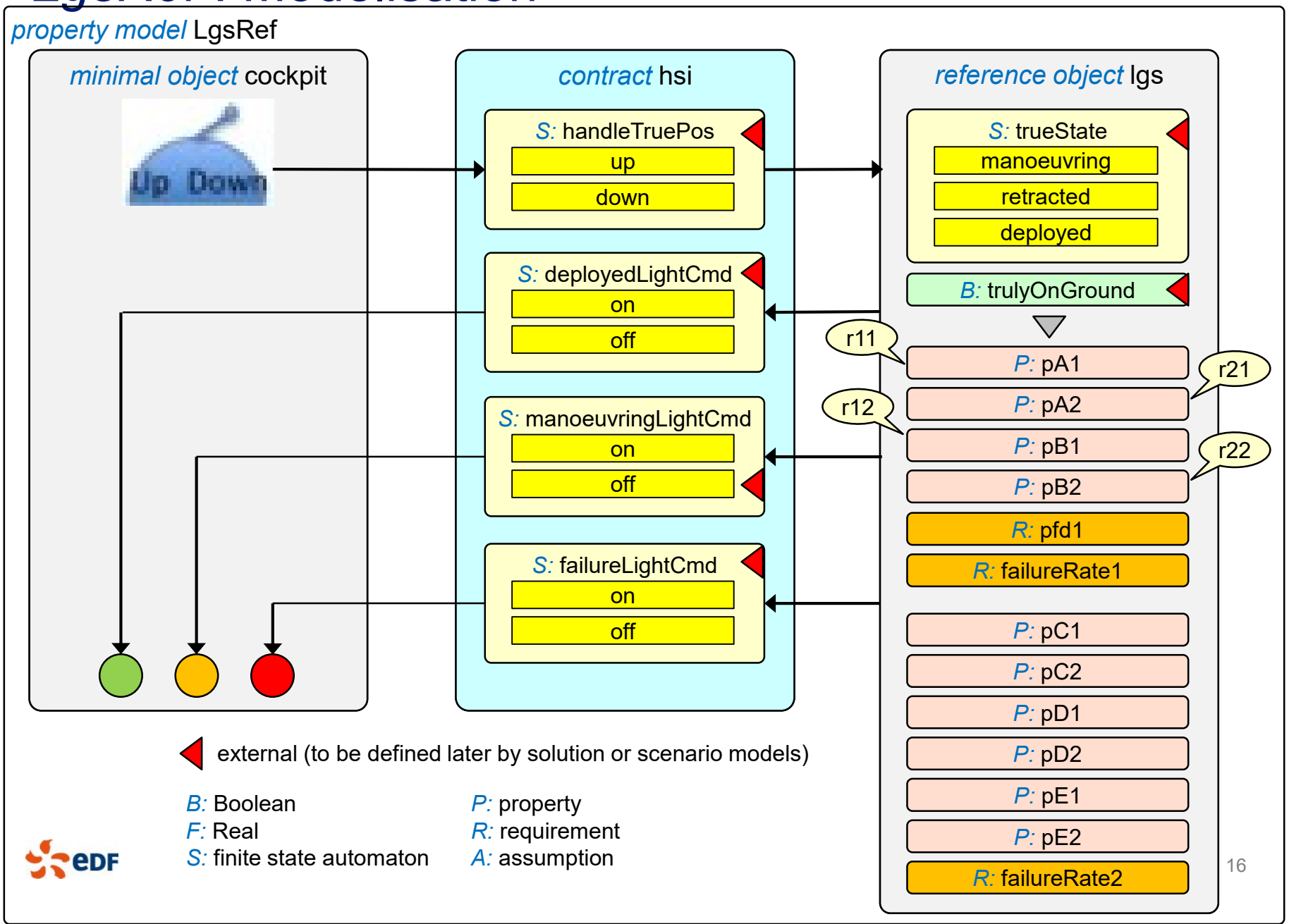
- **Propriété désirée *pE1* : en cas de violation des propriétés opérationnelles désirées, la lampe rouge doit être allumée dans un délai maximum de 100 ms**

- **Propriété désirée *pE2* : la lampe rouge ne doit pas être allumée intempestivement**

- **Propriétés similaires pour les lampes verte (*pC1* et *pC2*) et orange (*pD1* et *pD2*)**

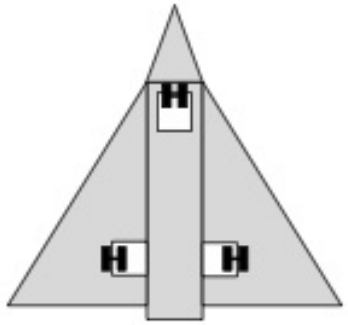
- **Exigence *reliability2* : la fréquence des défaillances dans la commande des lampes doit être inférieure à 10^{-3} par an**

LgsRef : Modélisation

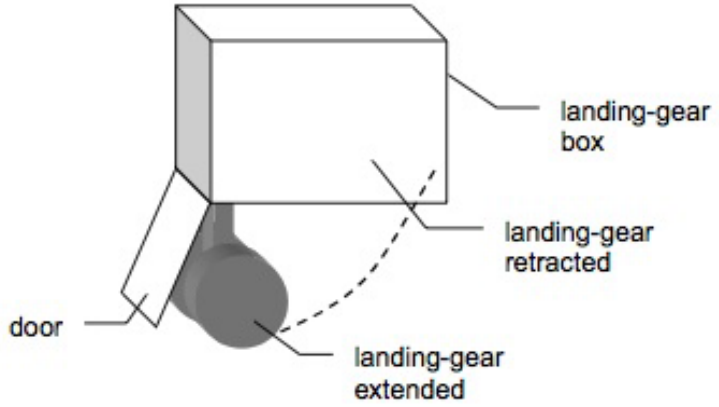


LgsDesign1 : Structure Générale et Manoeuvre

- **3 trains**
  - Un train Avant
  - Un train Droit
  - Un train Gauche

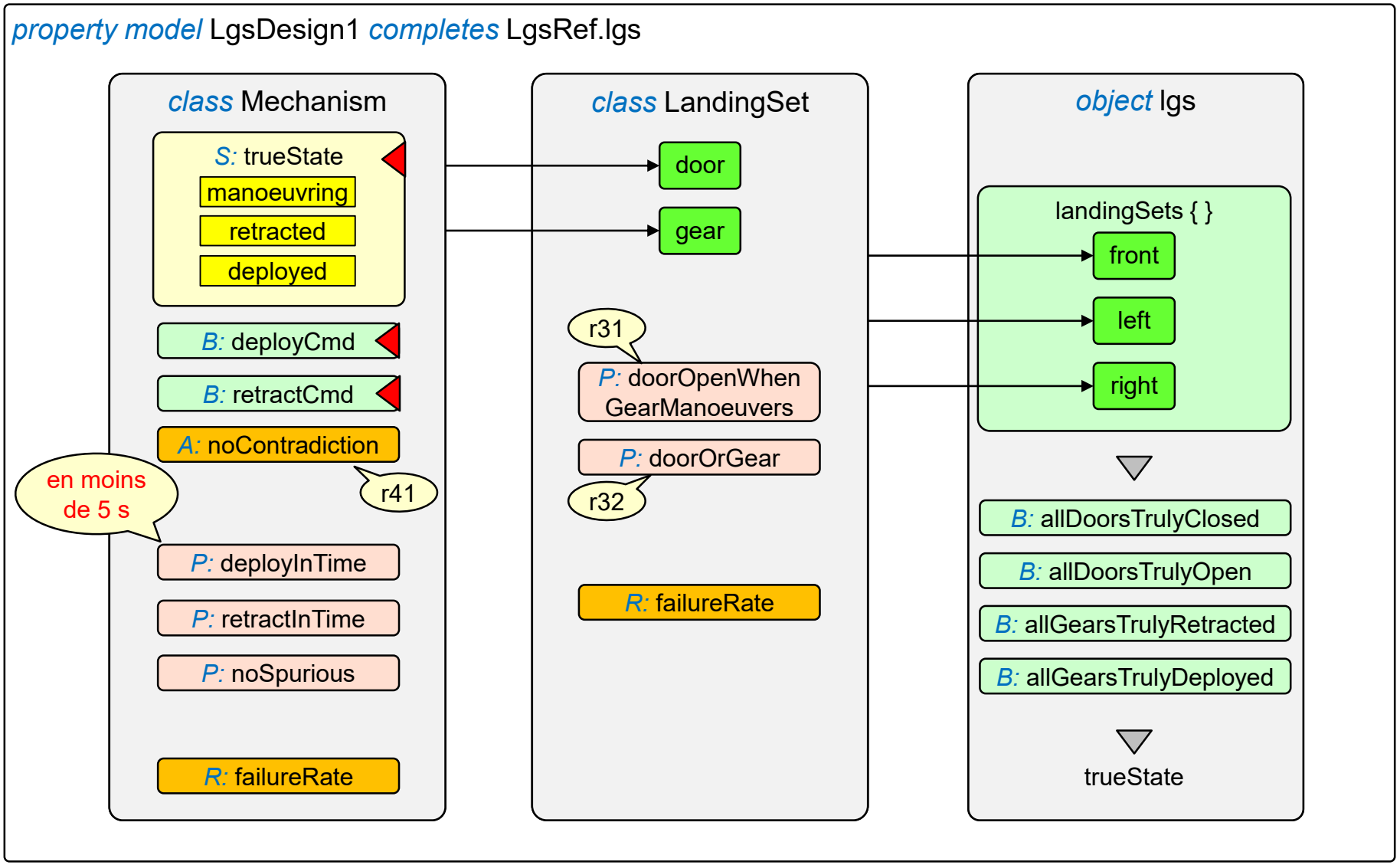


- **Le train Droit et le train Gauche sont identiques**
- **Chaque train est composé**
  - D'un capot (door)
  - D'un bras rétractable (gear) supportant les roues

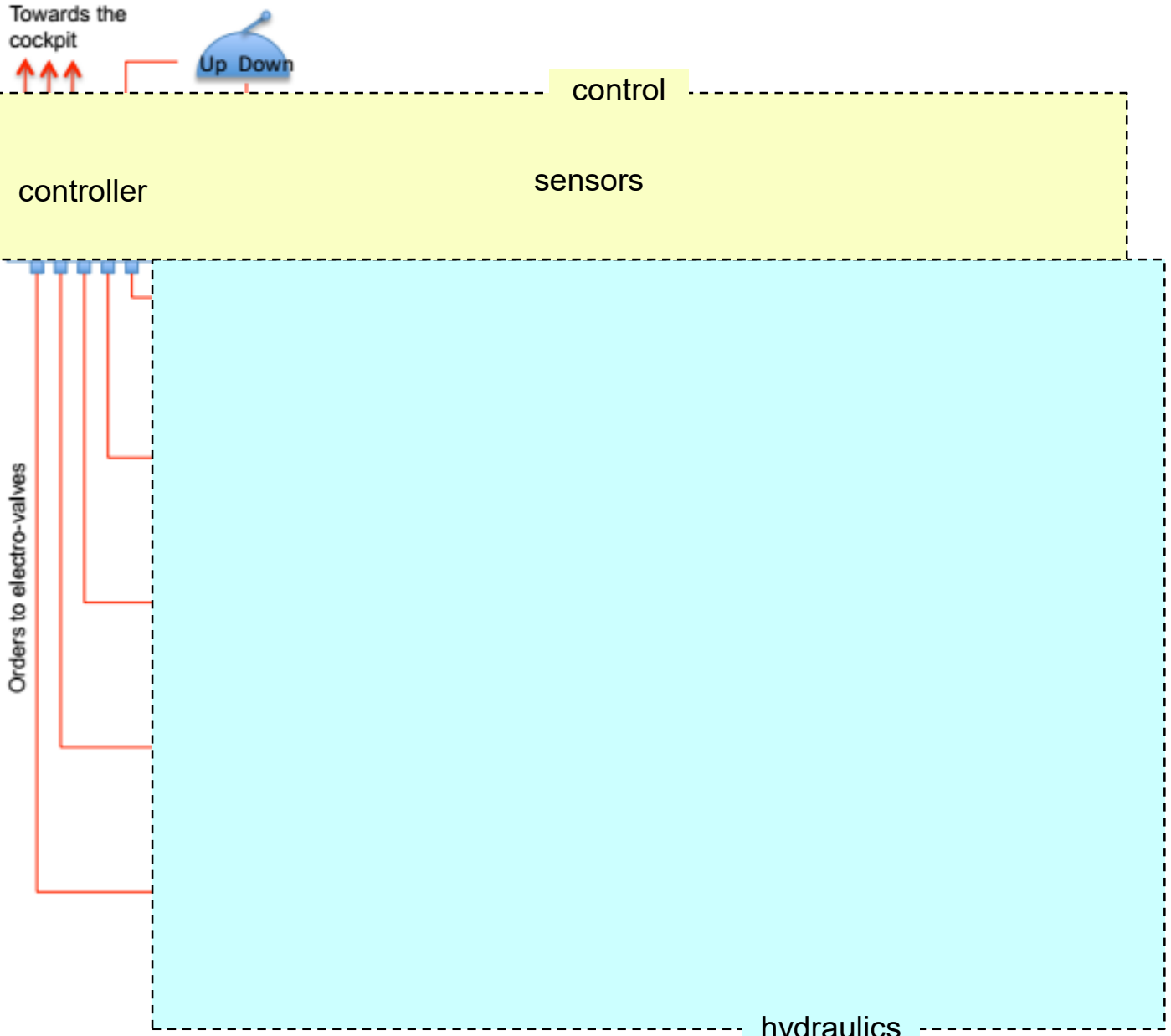


- **Pour sortir un train**
  - Ouvrir le capot
  - Déplier le bras
  - Fermer le capot
- **Pour rentrer un train**
  - Ouvrir le capot
  - Replier le bras
  - Fermer le capot

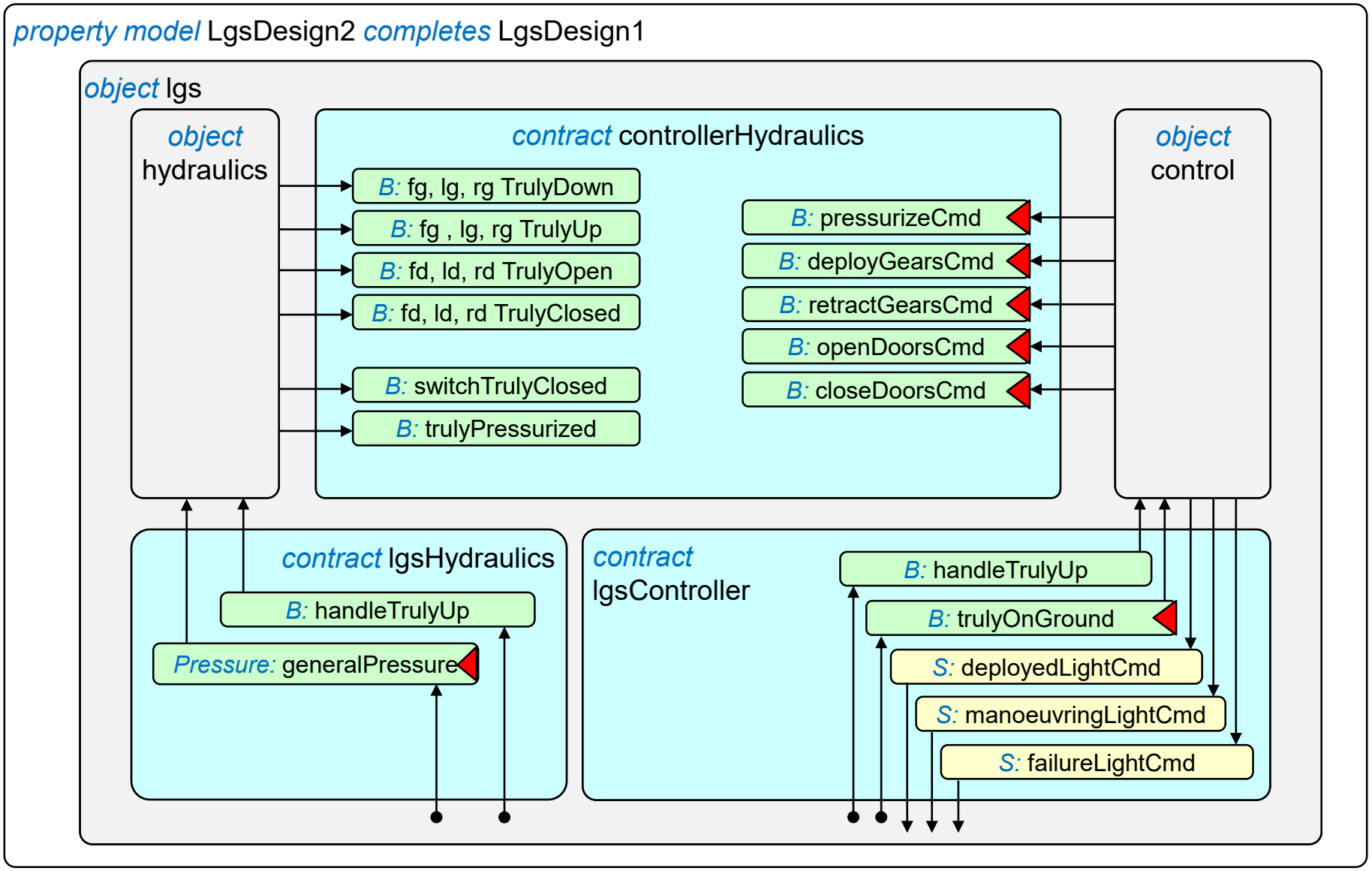
LgsDesign1 : Modélisation



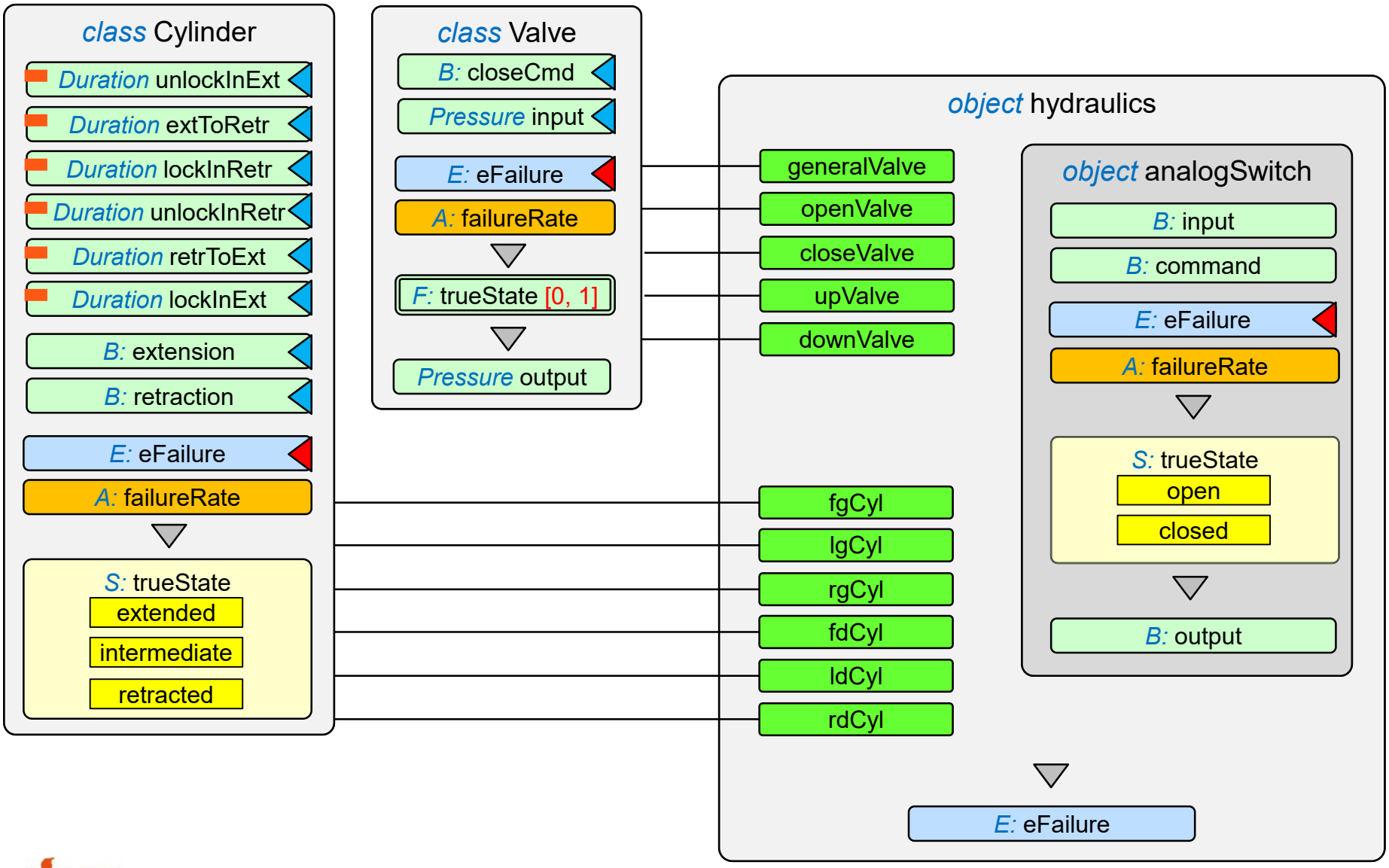
LgsDesign2 : Hydraulique et Commande



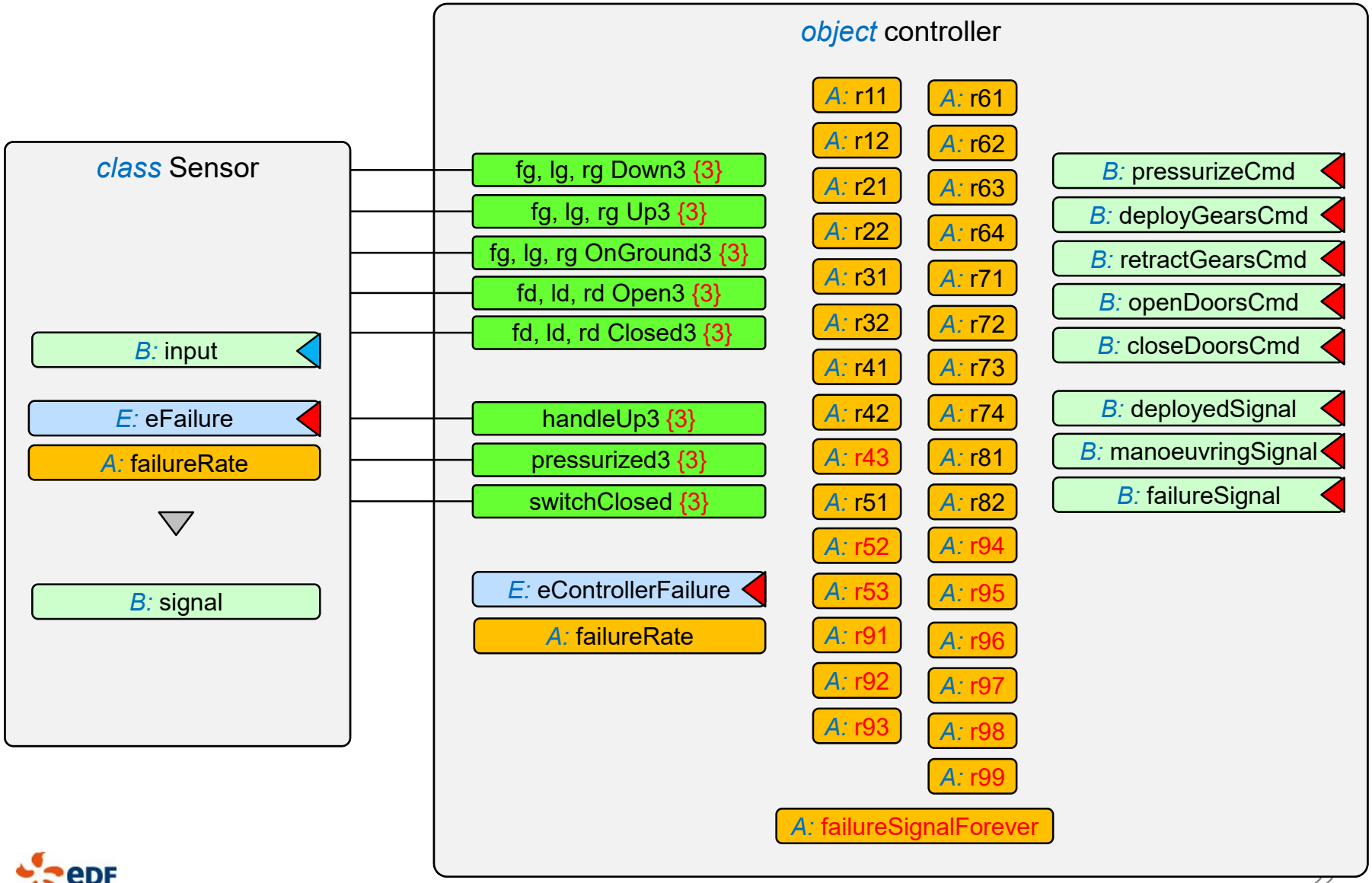
LgsDesign2 : Modélisation



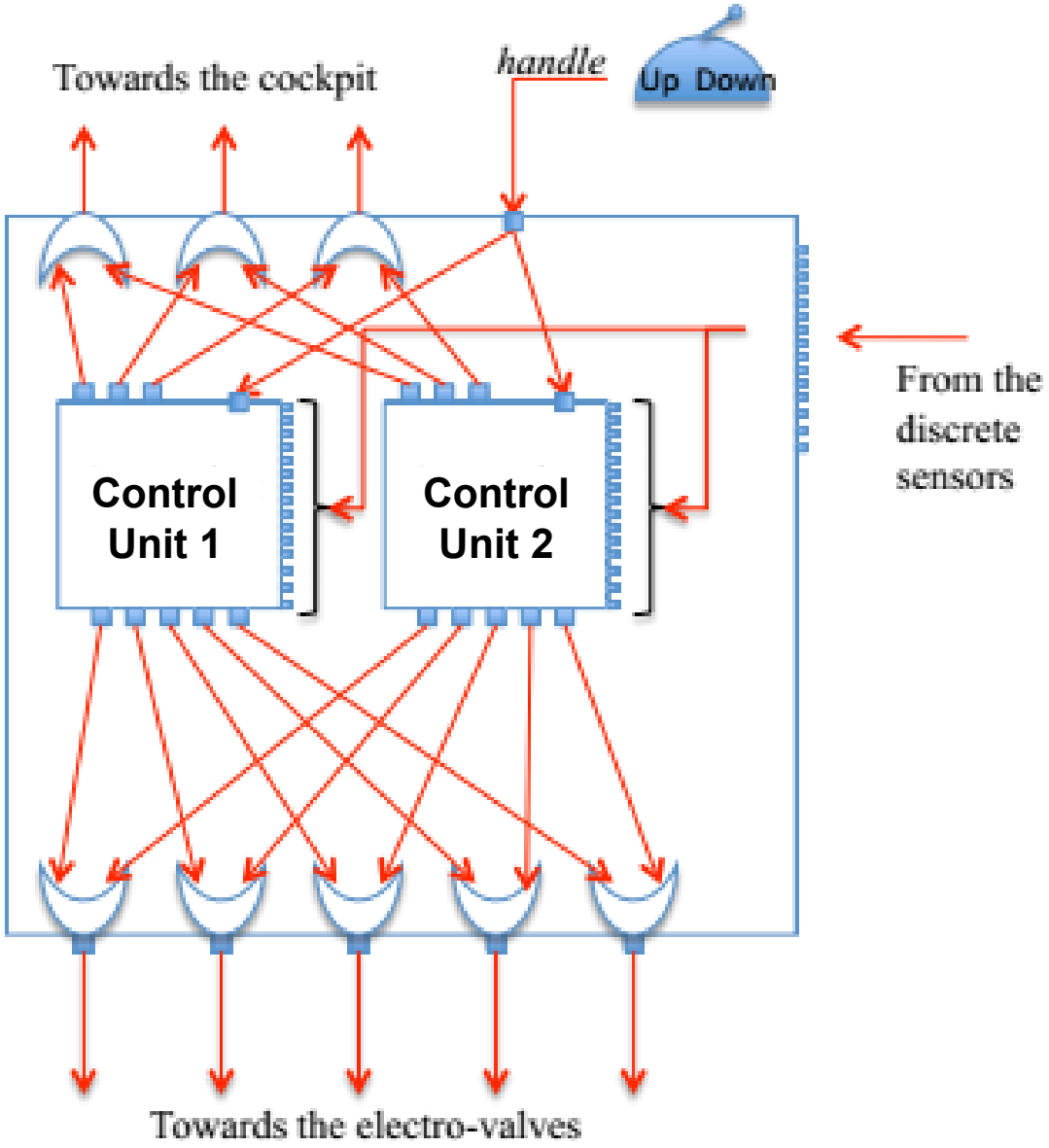
LgsDesign2 : Modélisation Pistons, Vannes, Switch



LgsDesign2 : Modélisation Contrôleur, Capteurs



LgsDesign3 : Deux Unités de Contrôle



LgsDesign3

- **Le contrôleur considère maintenant les hypothèses exprimées dans *LgsDesign2* comme des exigences**
- **Les unités de contrôle ont été modélisés comme des systèmes synchrones**
  - Mais elle ne sont pas synchronisées entre elles ➔ le contrôleur est un GALS (globalement asynchrone, localement synchrone)
- **Les deux unités de contrôle sont de conception identiques mais slles sont sujettes à des pannes et à des défauts de fabrication aléatoires**
- **On ne peut pas se contenter de faire un OU des sorties des deux unités de contrôle**
  - Si on veut satisfaire aux exigences de non contradiction des commandes
  - Si on veut satisfaire aux exigences de marge temporelle
- **La modélisation comportementale des unités de contrôle définit les 5 signaux de commande de vannes et les 3 signaux de commande des lampes de façon plus précise et plus impérative, mais garde encore une partie non déterministe**

Conclusions

- **L'approche proposée traite la spécification comportementale des logiciels et des systèmes embarqués comme un élément parmi les autres dans la conception des systèmes techniques**
- **La modélisation par contraintes se prête bien aux phases amont**
- **Une approche par petits pas facilite la vérification**
- **Les systèmes physiques sont des poupées russes : la spécification des exigences à un niveau est intimement liée à la conception du niveau du dessus**
- **La spécification des hypothèses est aussi importante que celle des exigences**
  - Ce qui est une hypothèse pour l'un est souvent une exigence pour l'autre
  - Ce qui est une hypothèse à une phase donnée peut être une exigence dans la phase suivante

Des questions ?