Industrialisation du logiciel Temps Réel Critique

Sommaire

- Projets opérationnels
- Les outils du marché utilisés et les contraintes associées
- CS et les méthodes
- CS et la R&D
- Conclusion

COMMUNICATION & SYSTEMES

Projets opérationnels

- **Plusieurs projets industriels**

1. **Alarmes** (AIRBUS)
2. Filière **Myriade** (CNES)
3. Projet **CSS D2X6Z9** (PSA)
   - Fiches en 4 volets, dans le dossier fourni au participant
   - Cartographie

} *sous ensembles produits par CS*

COMMUNICATION & SYSTEMES

Domaines d'applications et Caractéristiques des projets opérationnels

- **Domaines :**
  1) **Applications spécifiques et lourdes** : spatiales, aéronautiques, militaires…
  2) **Grand public** : assistance à la conduite automobile

- **Caractéristiques de ces STRC :**
  - Part croissante du logiciel embarqué
  - Gestion de nombreux systèmes
  - Contrôle de fonctionnement de façon automatique et **sûre**…

COMMUNICATION & SYSTEMES

Cartographie de la complexité des projets opérationnels

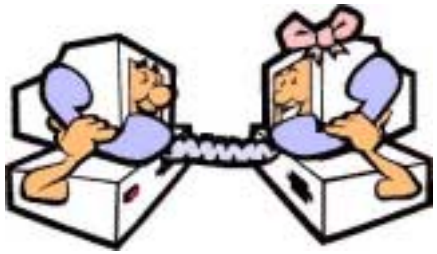
Complexité ↑

Coût de déploiement ↓
Contraintes de fiabilité ↑

1- Composant

2- Quelques Composants

3-Multi Composants



4-Multi- Composants **«banalisés»** et **«bon marché»** *architecture répartie et hétérogène*

Besoins/T

Émergents, à venir…

COMMUNICATION & SYSTEMES

Fonctionnalités principales des projets opérationnels

- **Dualité Contrôle/Commande :**
  - Commande ↔ Lois en temps échantillonné
  - Contrôle ↔ Evénements discrets
- **Temps réel critique :**
  - Contraintes temporelles
  - Défaillance ⇒ conséquences… INADMISSIBLES
- **Ressources limitées :**
  - Mémoire, puissance de calcul **limitées**
- **Exigences croissantes :**
  - Sûreté et certification (normes DO178B, EN50128, OSEK…)
  - Réutilisation
- **Besoin émergent ou à venir :**
  - Multi-composants **banalisés** et à **faibles coûts**
  - Architecture répartie et hétérogène (Proximité «capteurs et actionneurs»/calcul)

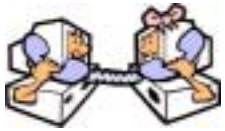
COMMUNICATION & SYSTEMES

Méthodes mises en œuvre dans les projets opérationnels

1- Composant

2- Quelques Composants

3- Multi Composants



4- Multi- Composants «banalisés» et «bon marché»

Programmation classique :

Documentation agrémentée de pseudo-code, diagrammes explicatifs…

Structuration Et Rationalisation:

Programmation Structurée

Composants Ada…

Spécification et conception :

Spécifications en «notations formelles»

Méthodes objets : HOOD, *UML*…

«Génie logiciel» vs. «cycle de vie» :

Méthodes à objets et formelles

Outils ad hoc le long du «Cycle», dont génération de code

CSS D2X6Z9 (PSA)

Alarmes (AIRBUS)

ATC A380 (AIRBUS)

Myriade (CNES)

COMMUNICATION & SYSTEMES

Outils utilisés dans les projets opérationnels

1- Composant

Langages assembleur

2- Quelques Composants

Langages classiques (C, Pascal, Ada…)

Procédures Intensives de test et de validation (RTRT…)

Développement sur Station de travail

OS spécifiques à génériques embarqués

©Photo MBDA France

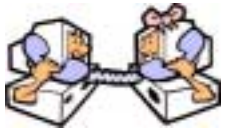
3- Multi Composants

Langages rigoureux (impératifs…)

Langages synchrones (Lustre…)

Outils d'aide semi-auto

Ateliers : STOOD, RHAPSODY, ROSE RT…



4- Multi- Composants «banalisés» et «bon marché»

Vers des ateliers dédiés

Prototypage rapide des lois de commande (Scicos/Scilab, Matlab/Simulink…)

Générateur automatique «certifié» de code (SCADE…)

CSS D2X6Z9 (PSA)

Alarmes (AIRBUS)

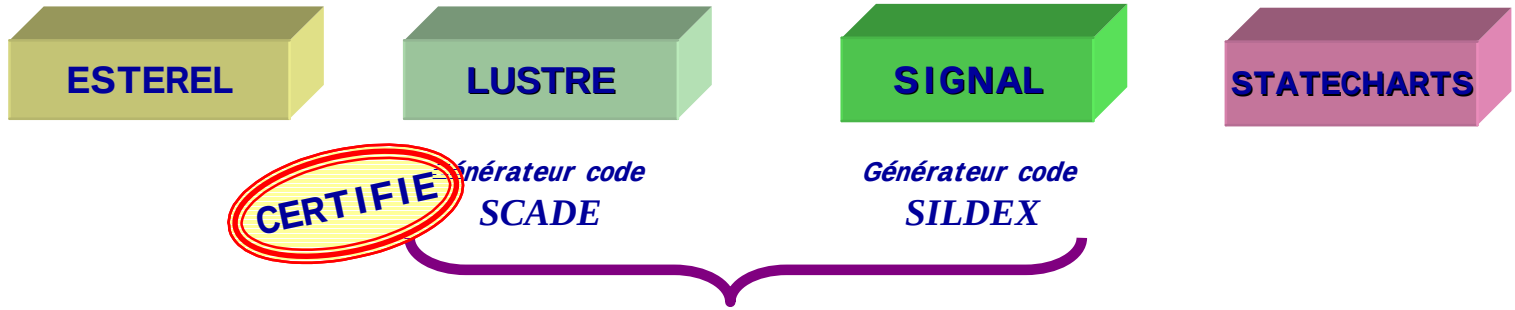
ATC A380 (AIRBUS)

Myriade (CNES)

COMMUNICATION & SYSTEMES

Les outils du marché et les contraintes associées - Communauté de l'informatique :

Langages synchrones



Formalisme
«Automates»

Formalisme
«Flots de données»

«Graphique»

- **Architectures matérielles**

«AAA»
Adéquation Algorithme Architecture

COMMUNICATION & SYSTEMES

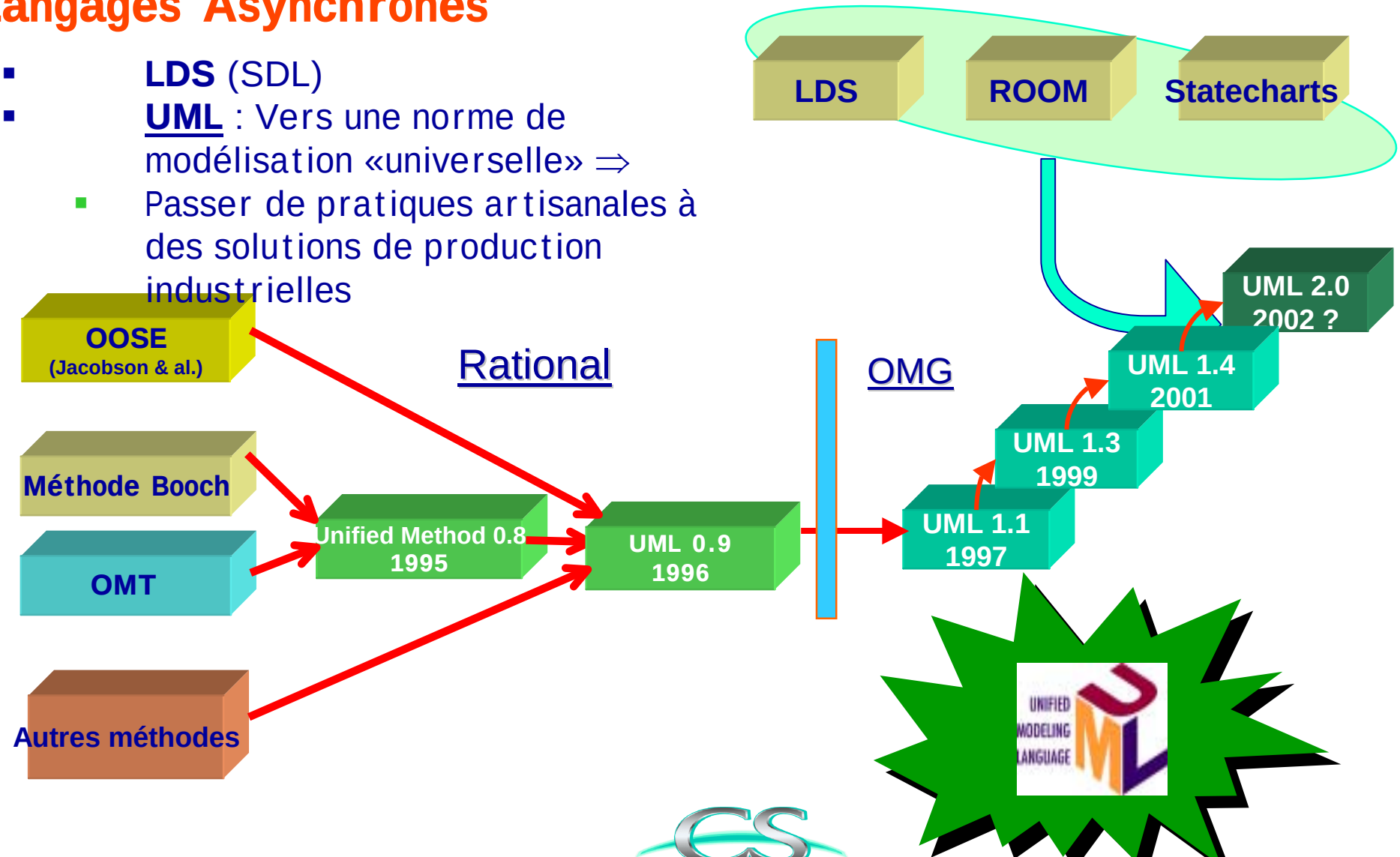
Langages synchrones

- **Forces :**
  - Langages parallèles (au sens de l'expression)
  - Compilateurs → «code centralisé»
  - Mise au point de programmes :
    - Valeurs numériques
    - Evénements en E/S
    - Optimisation
  - Conformité de la mise en œuvre de la spécification au niveau Modèle ⇔ SURETE

- **Faiblesses :**
  - Ils ne permettent pas la
    - Répartition
    - Tolérance aux pannes
    - Conformité de la mise en œuvre de la spécification au niveau Système

Langages Asynchrones

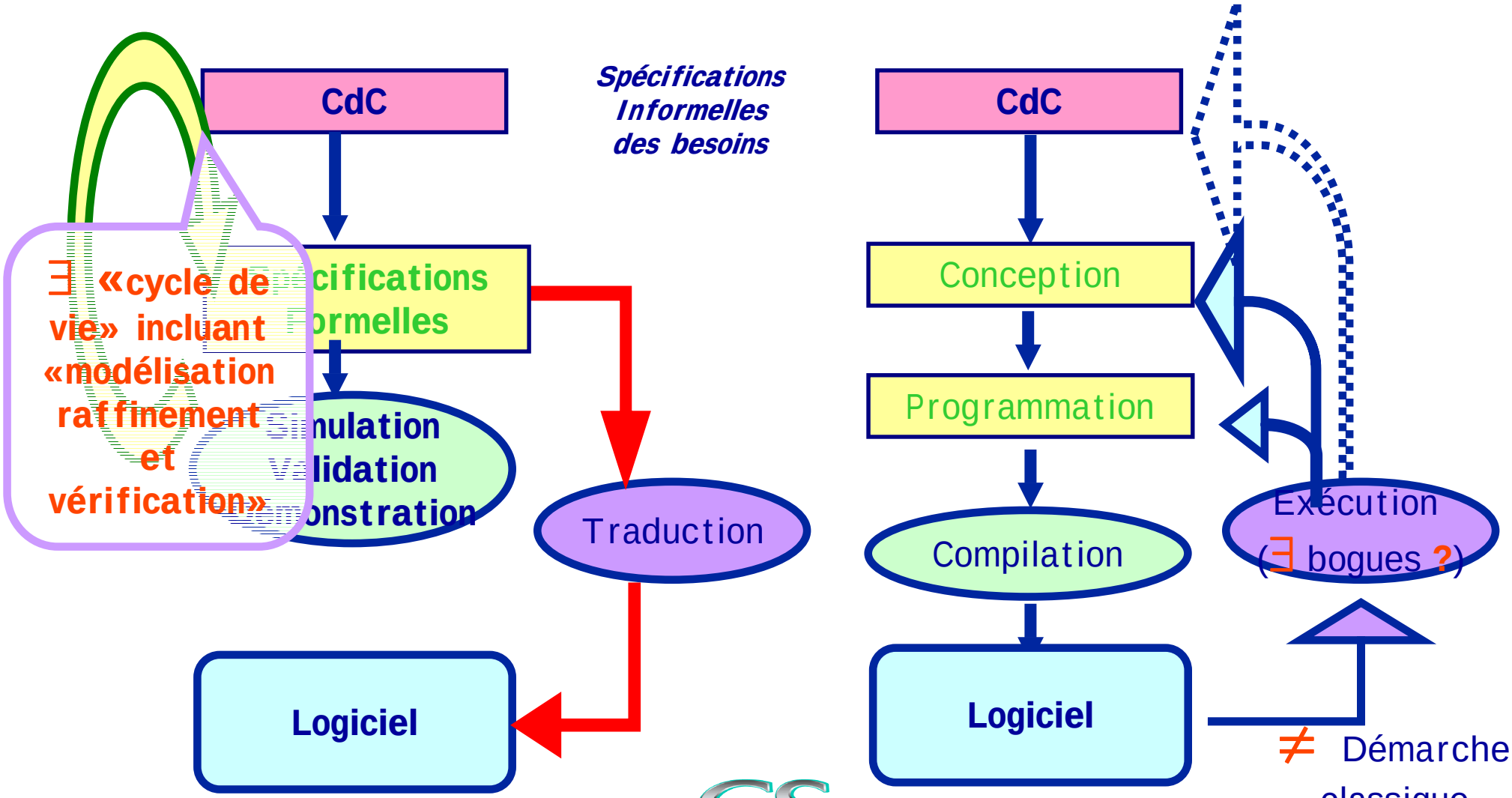
- **LDS** (SDL)
- **UML** : Vers une norme de modélisation «universelle» ⇒
  - Passer de pratiques artisanales à des solutions de production industrielles



Langages Asynchrones

- **Offrent :**
  - Modularité des spécifications
  - Séparation des problèmes
  - Facilité de la réutilisation de composants (héritage…)
  - Support au raffinement et à la lisibilité de spécifications
  - Facilité de la maintenance
  - Description cohérente de divers aspects du système dont :
    - Représentation des données
    - Concurrence
    - Expression des responsabilités dans le système
- Les ateliers support à UML **implémentent mal le méta-modèle** et sont **permissifs** :
  - Absence d'une définition précise des opérations (méthodes)
  - Spécifications non vérifiées
  - Problème d'expression des contraintes temps réel et des performances assignées
- LDS (voir exposé ATC A380)

CS et les méthodes - Le formel «revisité»



CS et les méthodes - Le formel revisité : classification

1. Méthodes **orientées propriétés** ⇔ descriptions dans un langage permettant d'énoncer les **propriétés attendues du système**

2. Méthodes **orientées modèles** ⇔ **construction** d'un système à partir **d'objets fondamentaux pré-établis**

3. Méthodes **Hybrides :**
   - VDM, Z et B
   - LOTOS
   - ***Méthodes à objets et formelles :***
     - CO-OPN
     - ***UML 2.0 + formel***

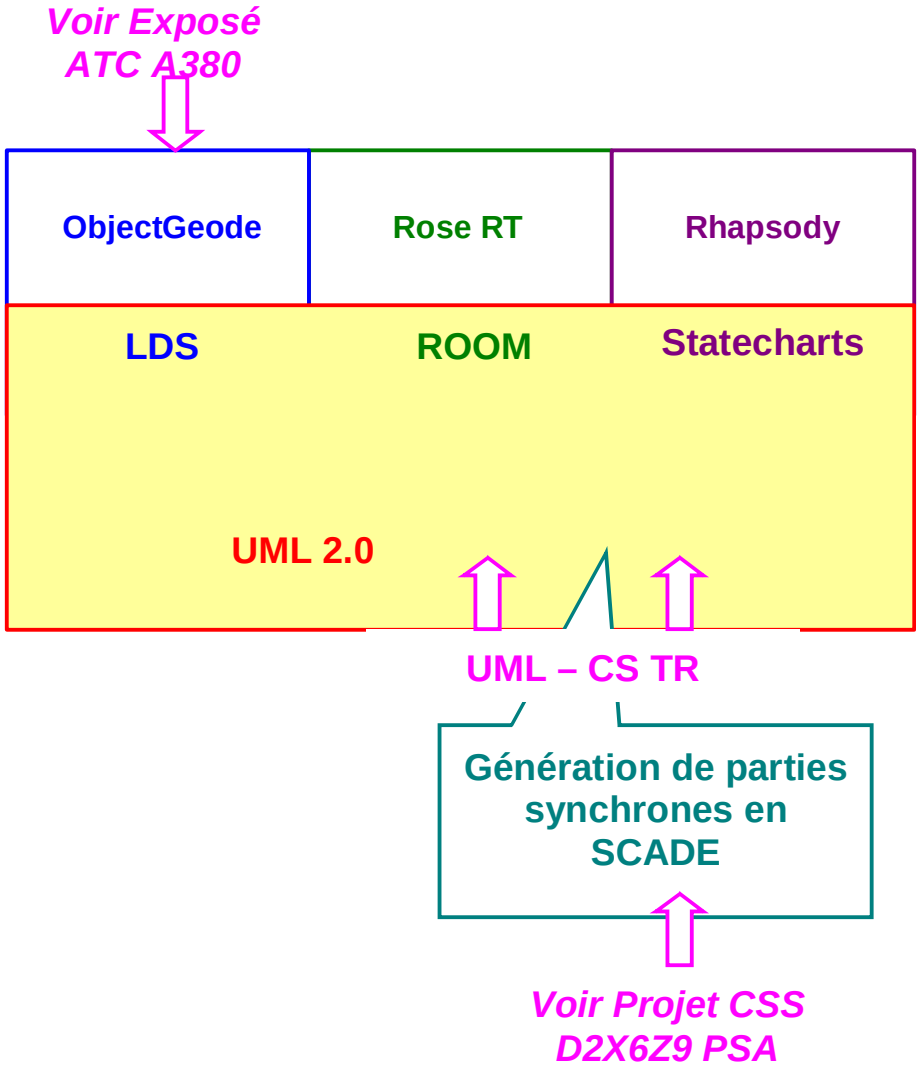
COMMUNICATION & SYSTEMES

CS et les méthodes – «UML – CS»

- **UML - CS**
  - Utilisée sur tous nos projets à objets
  - Cycle intensif de formation : 200 ingénieurs, sur 30 mois.

- **UML - CS TR :**
  - Indépendante de la notation
  - Décrit de façon factuelle les différentes phases de la modélisation, sans se soucier de leur implantation (UML, LDS, STATECHARTS, HOOD)
  - **Etude interne**, d'abord avec confrontation Rose RT
  - **Etude AIRBUS**, ensuite
    - Participation à l'étude ayant permis d'effectuer les choix des méthodes et outils pour l'A380 «approche full UML pour améliorer le processus existant»
    - Extension de la confrontation des outils à RHAPSODY et OBJECTGEODE

CS et les méthodes



CS et la R&D

- **Démarche :** approche **disciplinée** et à **faibles coûts** du développement des STRC

- Projets actifs :
  - **ACOTRIS**
    - Partenaires : CS Chef de file, CEA LIST, MBDA France, INRIA et SITIA
    - Interopérabilité entre les outils «Asynchrone (UML/ACCORD) + Synchrone(SIGNAL) + AAA (SynDEx)»
  - **NEPTUNE**
    - 5ème PCRD
    - Outil de vérification statique / semi-dynamique et de documentation de modèles UML

- Projet «labellisé» (AAP2002, RNTL)
  - **ECLIPSE**
    - Interopérabilité entre les outils «Prototypage rapide de lois de commande (Scicos/Scilab) + AAA (SynDEx)»
- Intention de projet : EOI NEPTUNE (6ème PCRD)
    - En entrée : résultats des projets ACOTRIS + NEPTUNE
    - Extension de Neptune à tout le cycle et au TR

COMMUNICATION & SYSTEMES

Conclusion

- Enjeux du temps réel critique :
  - Fiabilisation du logiciel
  - Maîtrise des coûts.

NOTRE DEMARCHE

- Fruit de notre **REx** et de l'approche conjointe de notre **R&D**
  - Répond aux besoins de nos «donneurs d'ordre»
  - Concourt à améliorer le processus à chaque étape du cycle de vie**.**

- Optimise et raccourcit ce processus (sémantique formelle $\leftrightarrow$ **modèle) :**
  - Formalisation par modèles $\Rightarrow$ déterminisme, prédictibilité du comportement…
  - Méthodologie amont de validation et de test de l'ensemble des comportements possibles $\Rightarrow$ résultats fiables…
  - Production par voie automatique de code $\Rightarrow$ certification.

- Prochains efforts : **intégrer encore plus de**
  - Validation amont et/ou vérifications statiques
  - Preuve amont et aval en vue de la certification.

COMMUNICATION & SYSTEMES