

AEE

Une Architecture Electronique Embarquée pour les systèmes embarqués dans l'automobile

* *Françoise Simonot-Lion* (simonot@loria.fr)

INPL - LORIA (UMR CNRS 7503) – Nancy

* *Jean-Pierre Elloy* (elloy@irccyn.ec-nantes.fr)

* *Yvon Trinquet* (Yvon.Trinquet@irccyn.ec-nantes.fr)

ECN - IRCCyN (UMR CNRS 6597) – Nantes

AEE
Architecture Electronique Embarquée

<http://aee.inria.fr>

Constructeurs :

PSA PEUGEOT CITROËN



Laboratoires de recherche :



INRIA

IRCYN



Equipementiers :



SIEMENS

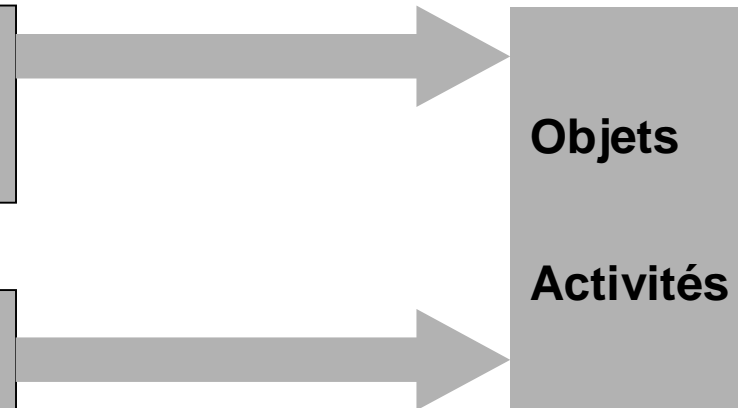
Valeo

+ EADS Launch Vehicles

Financé par le Ministère de l'Industrie (septembre 1998 - décembre 2001)

Plan

- 1 - Le contexte
- 2 - La problématique
- 3 - L'atelier de développement
- 4 - La construction d'un système
 - ◆ *Couche « Projet »*
 - ◆ *Couche « Fonctionnelle »*
 - ◆ *Les composants architecturaux*
 - ◆ *Couche « Logicielle »*
 - ◆ *Couche « Matérielle »*
 - ◆ *Couche « Opérationnelle »*
- 5 - Conclusions, points durs et ... la suite



1 - Le contexte

1.1 - Le contexte du marché

Quelques chiffres

- Production de véhicules dans le monde :
40 millions (1998) → 60 millions (2010)

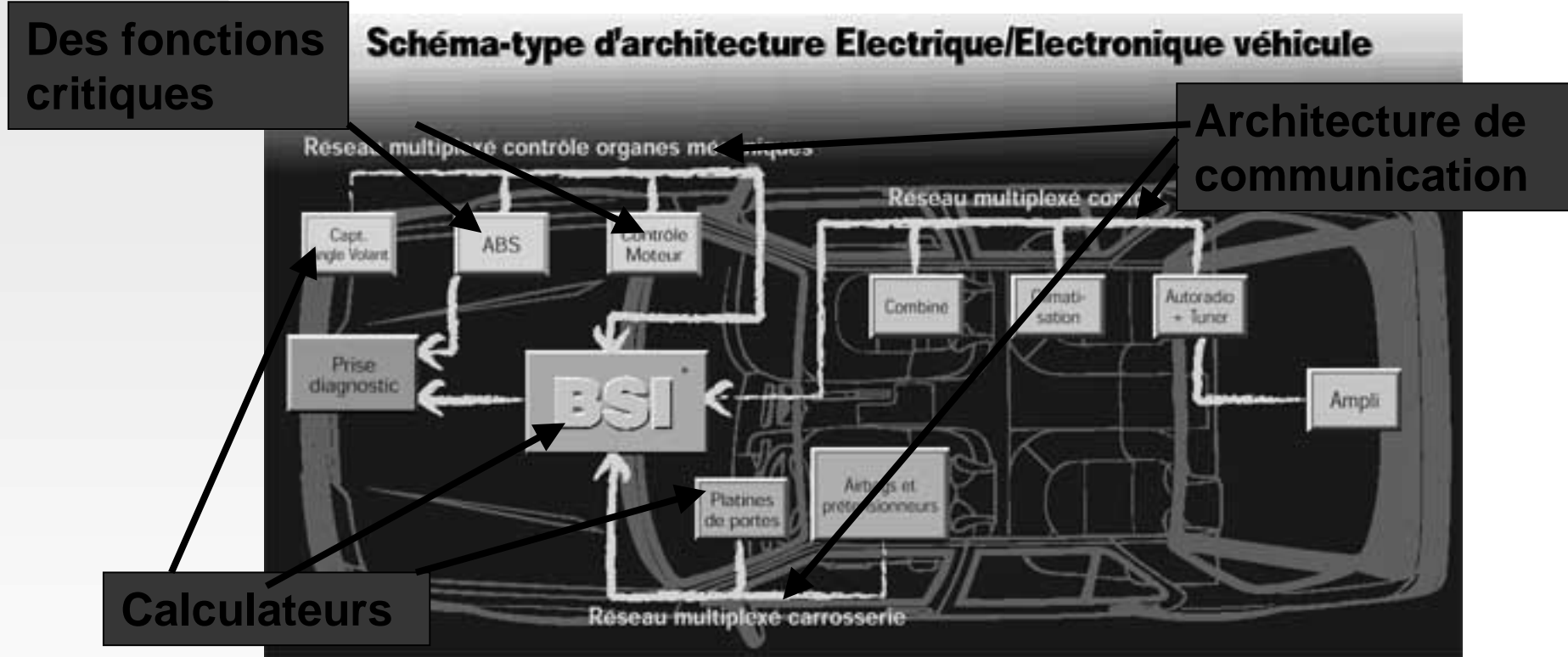
- Coût de l'électronique embarquée
37 000 M\$ (1995) → 60 000 M\$ (2000)

+10% by year

- $\frac{\text{Coût de l'électronique}}{\text{Coût du véhicule}} = 20\%$

- Augmentation du confort et de la sécurité
- Les nouvelles normes de pollution
- Prestations nouvelles (impossibles en technologie classique)
- Fonctions X-by-wire

1.2 - Le contexte technologique



Complexité fonctionnelle

- **Lois de commande multivariable**
- **Nombre de grandeurs d'influence**
- **Nombre de modes de fonctionnement**

Complexité architecturale

- **Nombre et hétérogénéité des calculateurs et réseaux**
- **Diversité des contraintes comportementales et temporelles**

Complexité inter-systèmes

Accroissement de la criticité

1.3 - Le contexte de développement

Acteurs du développement

- **Cycle de développement partagé entre**
 - **équipementiers**
 - **constructeurs**
- **Echanges constructeurs / équipementiers**
 - **échange de documents ou de produits**
 - **boîte noire / boîte grise / boîte blanche**
 - **cahier des charges (exigences) / test**
- **Relations constructeurs / équipementiers**
 - **propriétés intellectuelles**

2 - La problématique

2.1 - La demande

... du client

... du constructeur

- Intégration rapide de services
- Réduction des coûts

- Personnalisation des véhicules
 - Diversification de l'offre
 - Sortie fréquente de modèles

... du produit

- Qualité
- Sûreté

**Flexibilité
des
architectures**

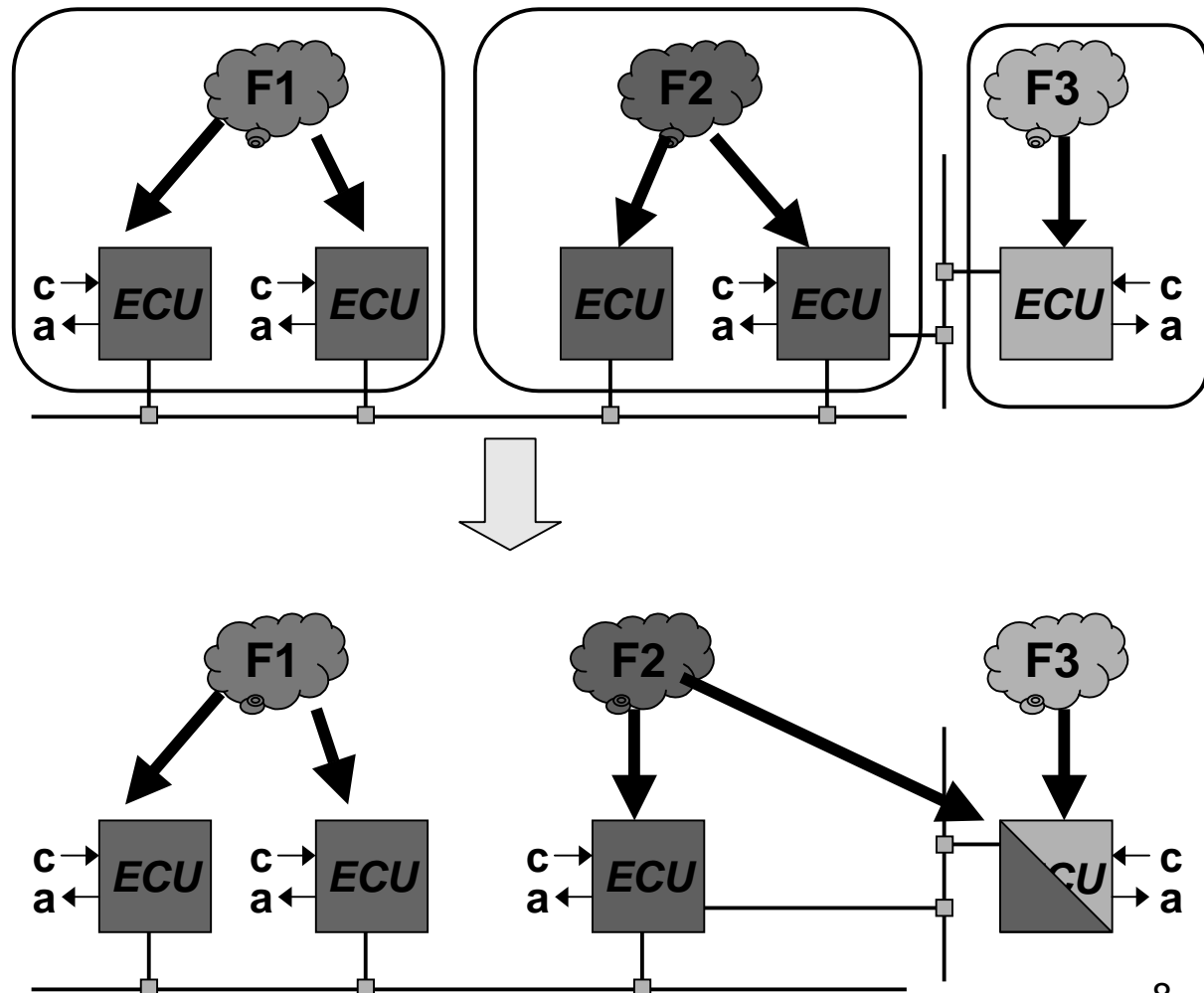
```
graph TD; subgraph Client; C[... du client]; end; subgraph Constructor; C1[... du constructeur]; end; subgraph Product; P[... du produit]; end; C --> Flex[Flexibilité des architectures]; C1 --> Flex; P --> Flex;
```


Flexibilité des architectures

- Réutilisation
- Intégration
- Allocation

ECU: calculateur
c : capteur
a : actionneur

Flexibilité de l'allocation du logiciel



2.2 - Le processus

... d'intégration

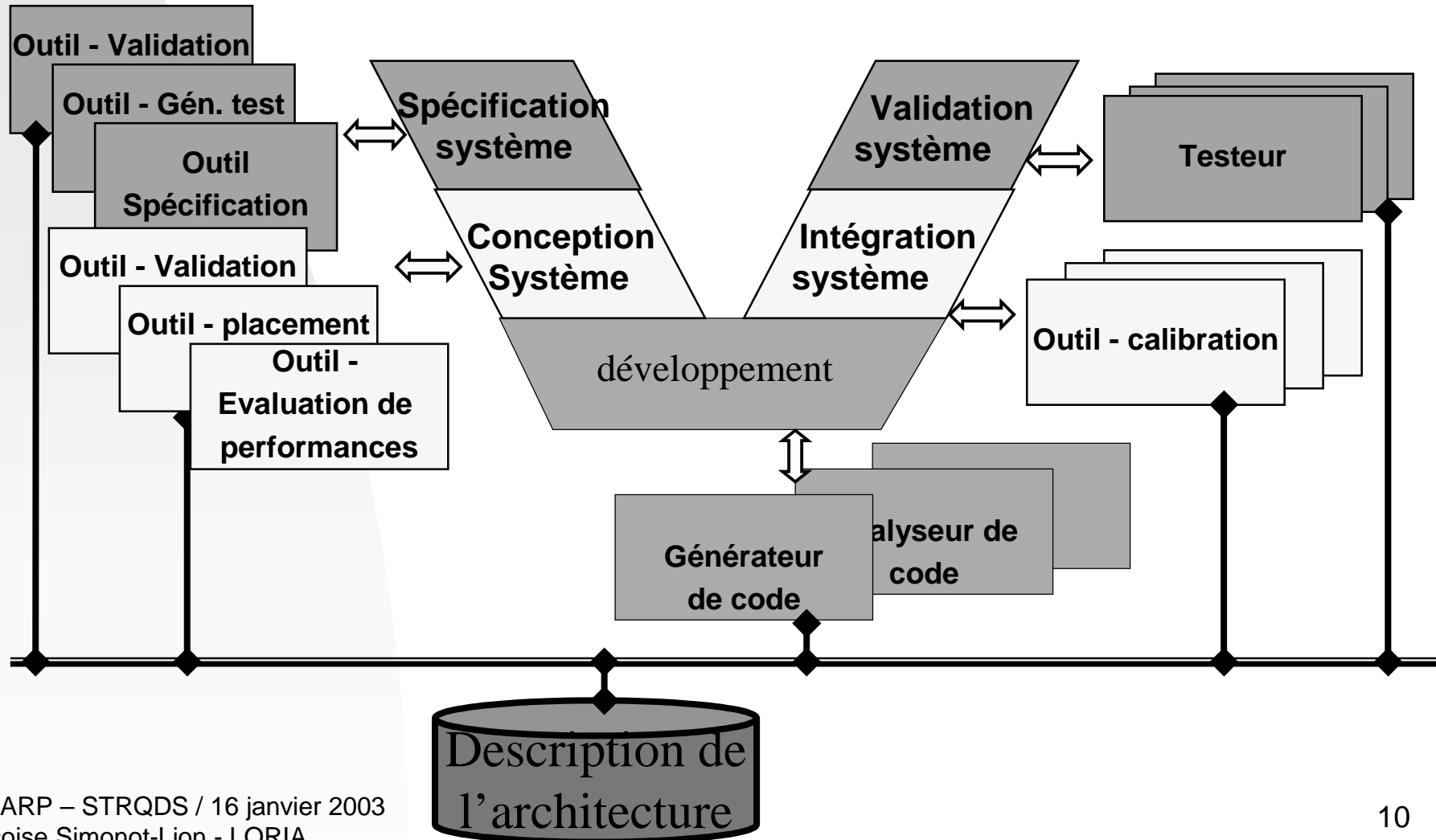
- **Système = assemblage de composants**
- **Cahier des charges -> suivi des exigences**

... d'échange

- **Délimitation de sous-architectures**
- **Négociation de compatibilité d'exigences**
- **Modélisation externe de prestations**
- **Fourniture de boîtes blanches, grises ou noires**

3 - L'atelier

3.1 - L'ossature



3.2 - Le dénominateur commun



AIL_Transport :
Architecture Implementation Language
pour les applications dans le domaine du transport

- Définit un référentiel commun pour modéliser tout ou partie d'une architecture électronique embarquée
- Facilite les échanges entre acteurs
- Est utilisé à toutes les étapes du processus de développement

Diagrammes de classes UML

4 - La construction d'un système

... s'appuie sur des niveaux d'abstraction qui ...

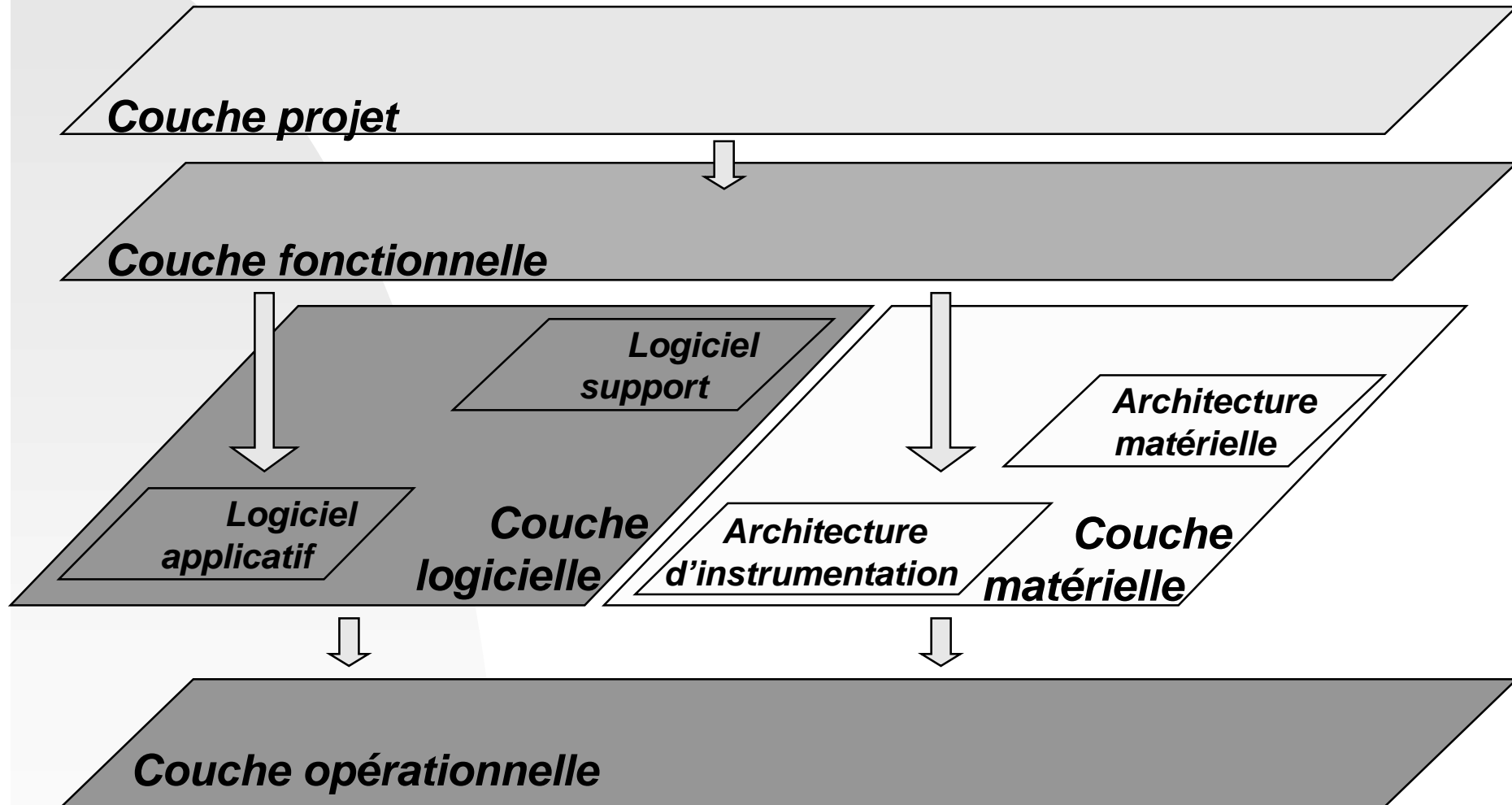
... définissent des points de vue sur tout ou partie d'une architecture électronique embarquée

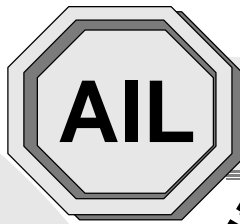
... sont des descriptions différentes du même système, et utilisées pour des fins différentes :

- **une vue de haut niveau**
(spécification du produit - spécification fonctionnelle)
- **une vue intermédiaire décrivant les aspects logiciel et matériel**
- **une vue de bas niveau décrivant le système tel qu'il est**

Descriptions de l'implantation

... se décline en couches :





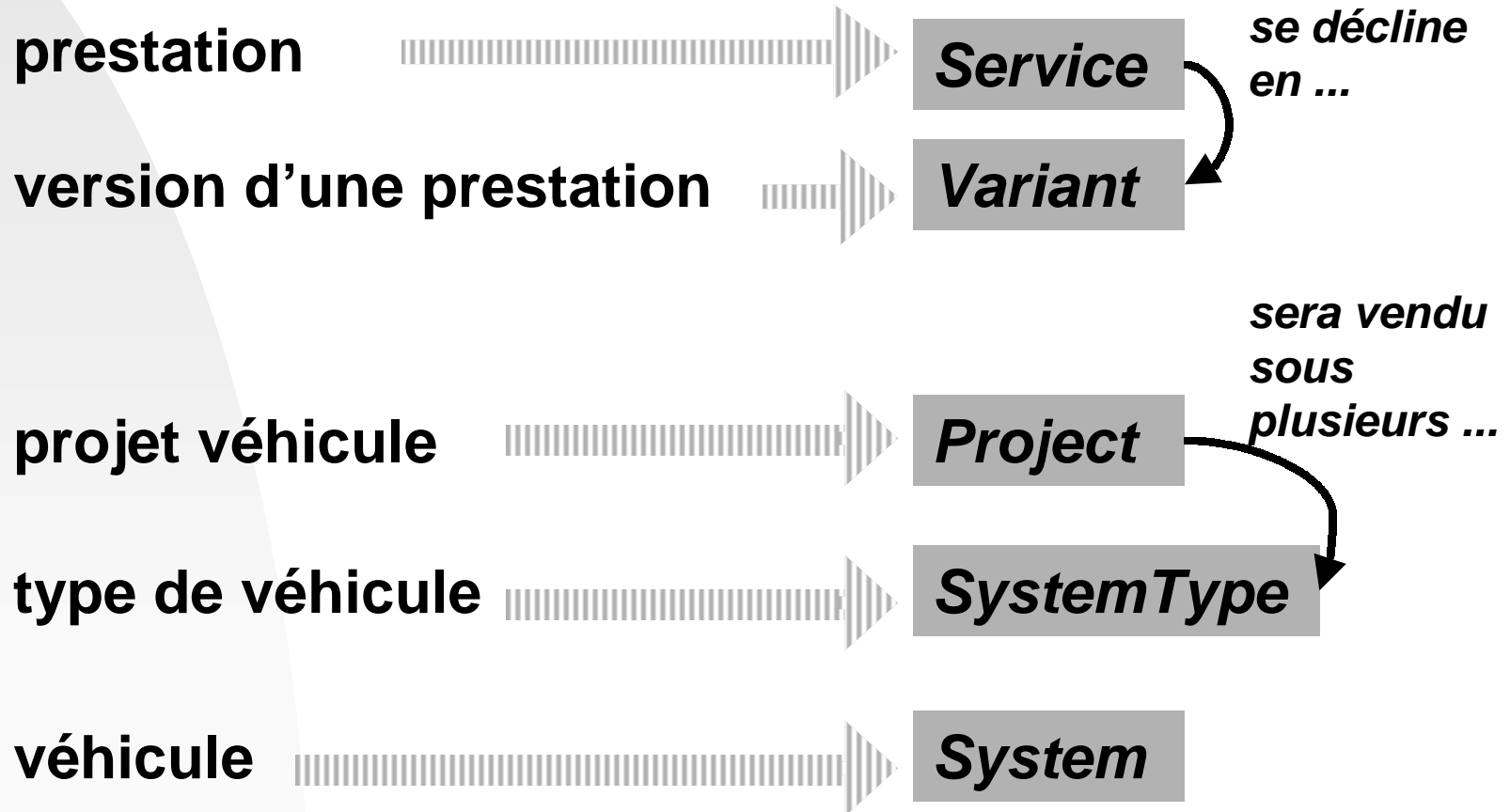
Classes de la couche « Projet »

... modélisent la variété de la gamme des produits (véhicules) du projet, en fonction :

- du niveau de confort et de prestation de chaque produit
- des alternatives des mêmes services à chaque niveau
- des options offertes au client

- Un **véhicule** est la réalisation d'un ensemble de **prestations**
- Suivant le **type** du véhicule, des **prestations** sont fournies ou en options
- Une **prestation** peut se décliner en plusieurs **versions**
- Suivant le **type** du véhicule, une **prestation** possible est fournie dans une **version** imposée, ou choisie par le client parmi des **versions** proposées

AIL Classes de la couche « **Projet** »



AIL Classes de la couche « **Projet** »

prestation



Service

version d'une prestation



Variant

projet véhicule



Project

type de véhicule



SystemType

véhicule



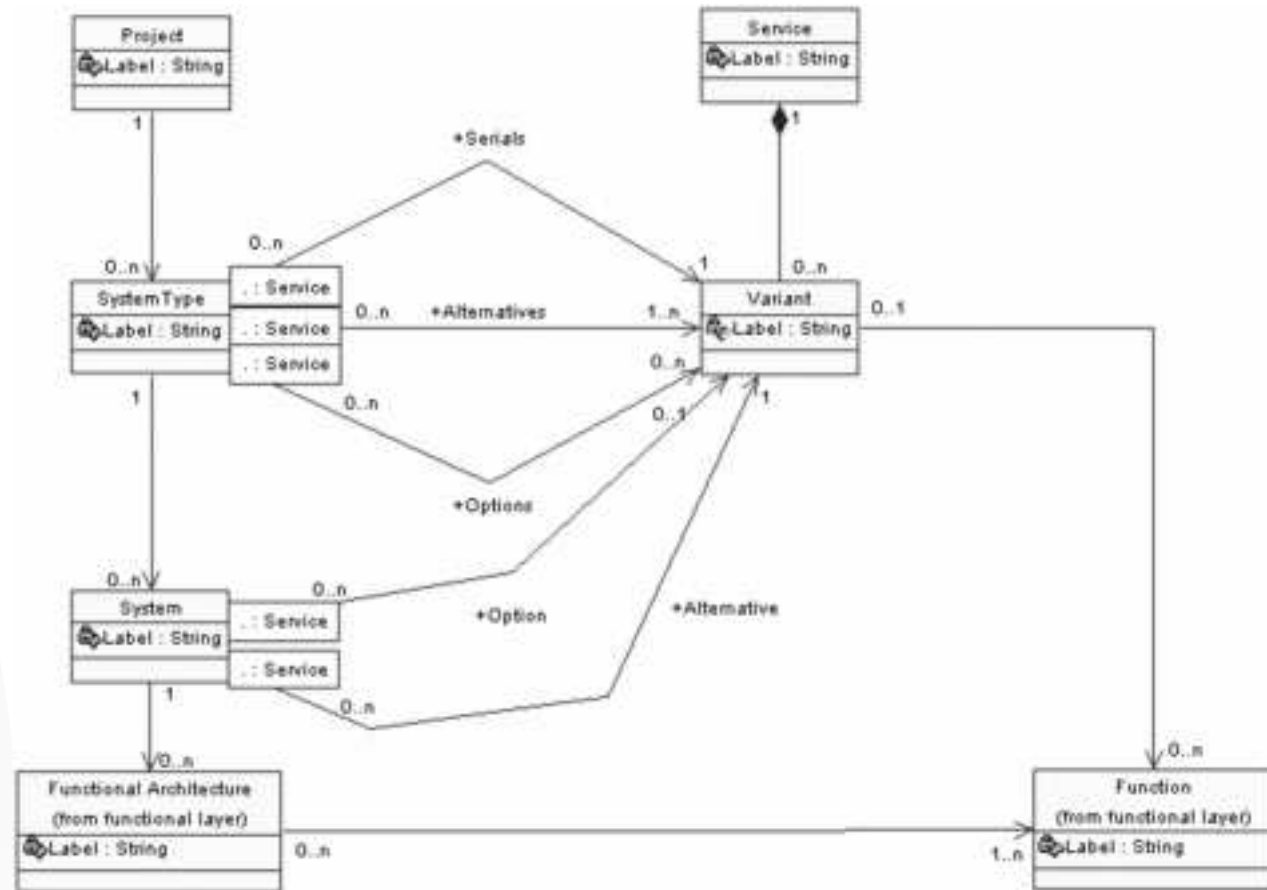
System

ce que peut acheter un client

est un ensemble possible de ...



Classes de la couche « Projet »



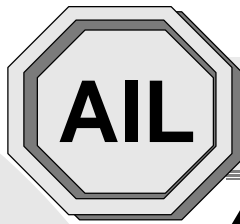
Activités sur la couche « Projet »

- Vérification de cohérence

*exemple: tout système électronique possible (objet **System**) embarqué dans un véhicule d'une gamme donnée (objet **Project**) réfère des prestations (objet **Service**) existantes dans une version (objet **Variant**) existante et possible pour le type du véhicule (objet **SystemType**) et la prestation choisie*

- Documentation

-- > **Gestion de configuration, historisation**

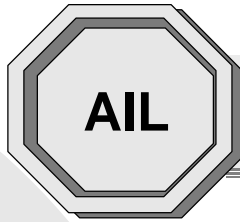


Classes de la couche « Fonctionnelle »

... modélisent la **spécification fonctionnelle** des prestations attendues de l'architecture embarquée, **indépendamment** de leur réalisation et de leur localisation dans l'architecture réelle

- chaque **version** d'une **prestation** est réalisée par une **fonction**
- il peut y avoir (lors des études) plusieurs solutions alternatives
- à chaque **véhicule** correspond plusieurs **organisations** d'un ensemble de fonctions

- toute fonction peut consommer des **données** fonctionnelles en **entrée** et en produire en **sortie**
- les **capteurs** et **actionneurs** sont des sources et des puits de données
- la réalisation d'une prestation dans un **mode de fonctionnement** est décrite par des fonctions et les **flots de données** qu'elles s'échangent



Classes de la couche « Fonctionnelle »

fonction



Function

*se décompose
en ...*

consomme

données consommées



Functional Input

produit

données produites

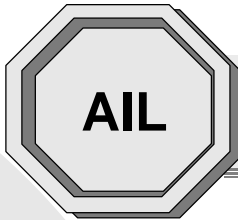


Functional Output

capteur / actionneur



Device



Classes de la couche « Fonctionnelle »

Organisation des fonctions d'un véhicule



Functional Architecture

contient des ...

Function

*est reliée à d'autres fonctions
ou à l'instrumentation par ses ...*

Functional Input

Functional Output

via des ...

**Connexion de fonctions,
de fonctions avec l'instrumentation
par flots de données**



Functional Flow



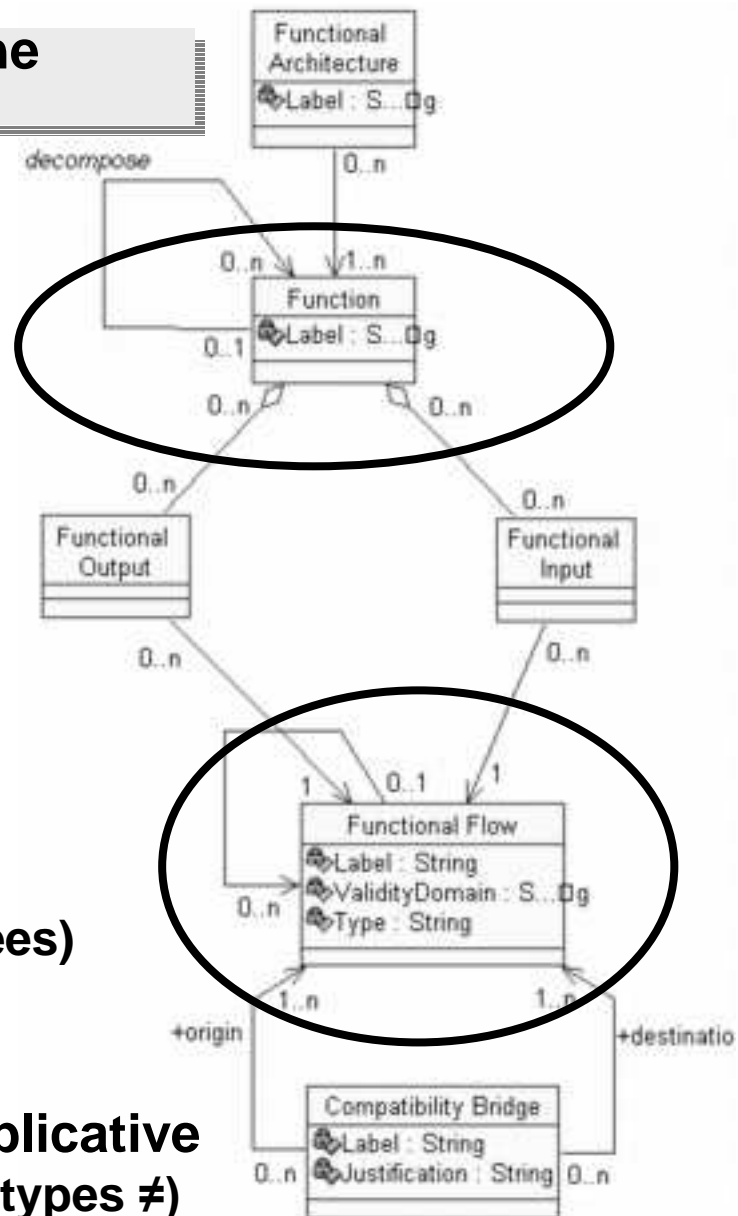
**Classes de la couche
« Fonctionnelle »**

Organisation des fonctions

Ports d'entrées-sorties

**Flots de données
(types des données échangées)**

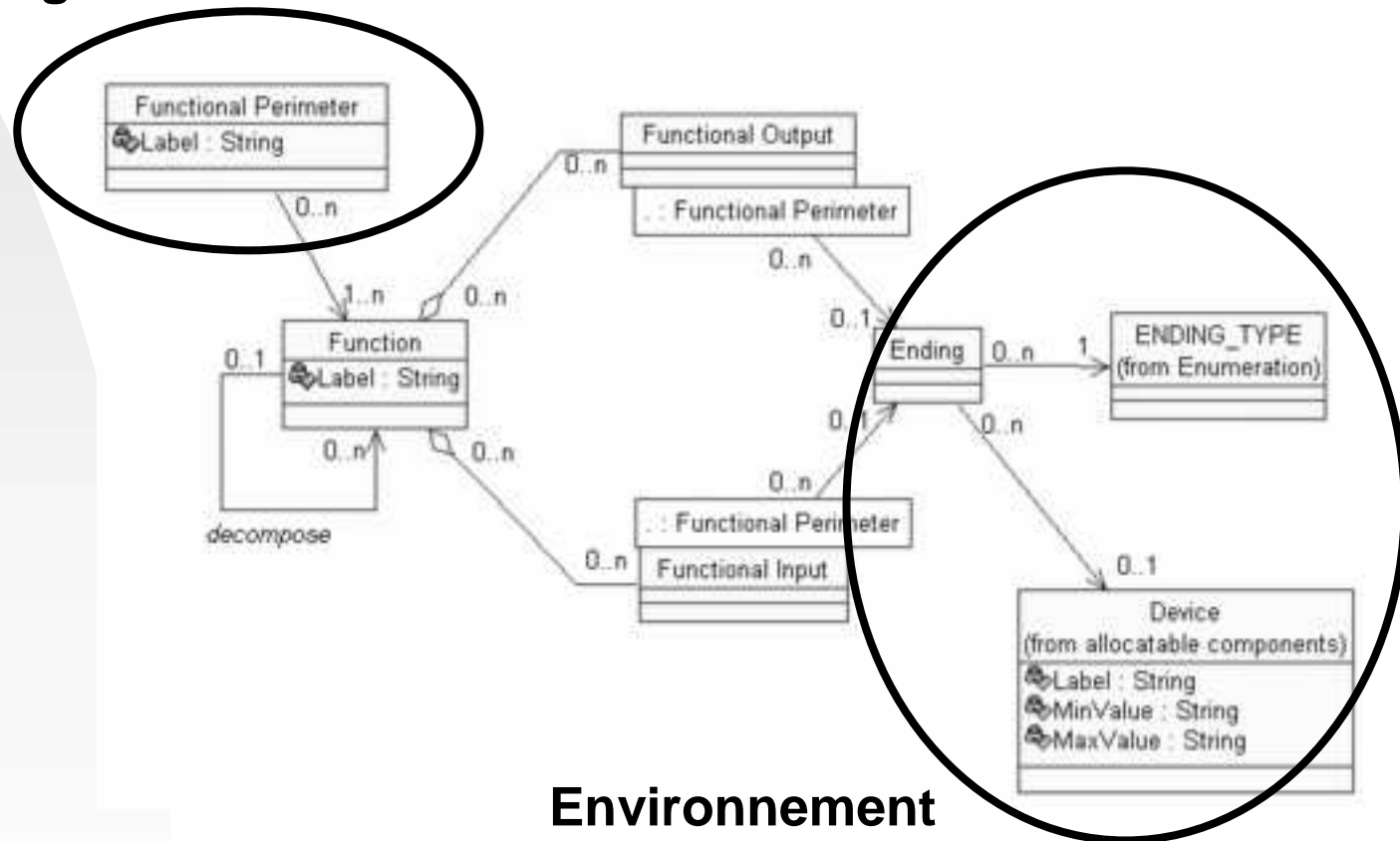
**Compatibilité applicative
(entre données de types ≠)**





Classes de la couche « Fonctionnelle »

Diagramme de contexte



Environnement du diagramme

Activités sur la couche « Fonctionnelle »

- **Vérification de cohérence**

*exemple: dans une système embarqué (objet **FunctionalArchitecture**)
toute donnée consommée (objet **Functional Input**) par une
fonction (objet **Function**) est produite (objet **Functional
Output**) par une fonction ou par un capteur (objet **Device**)*

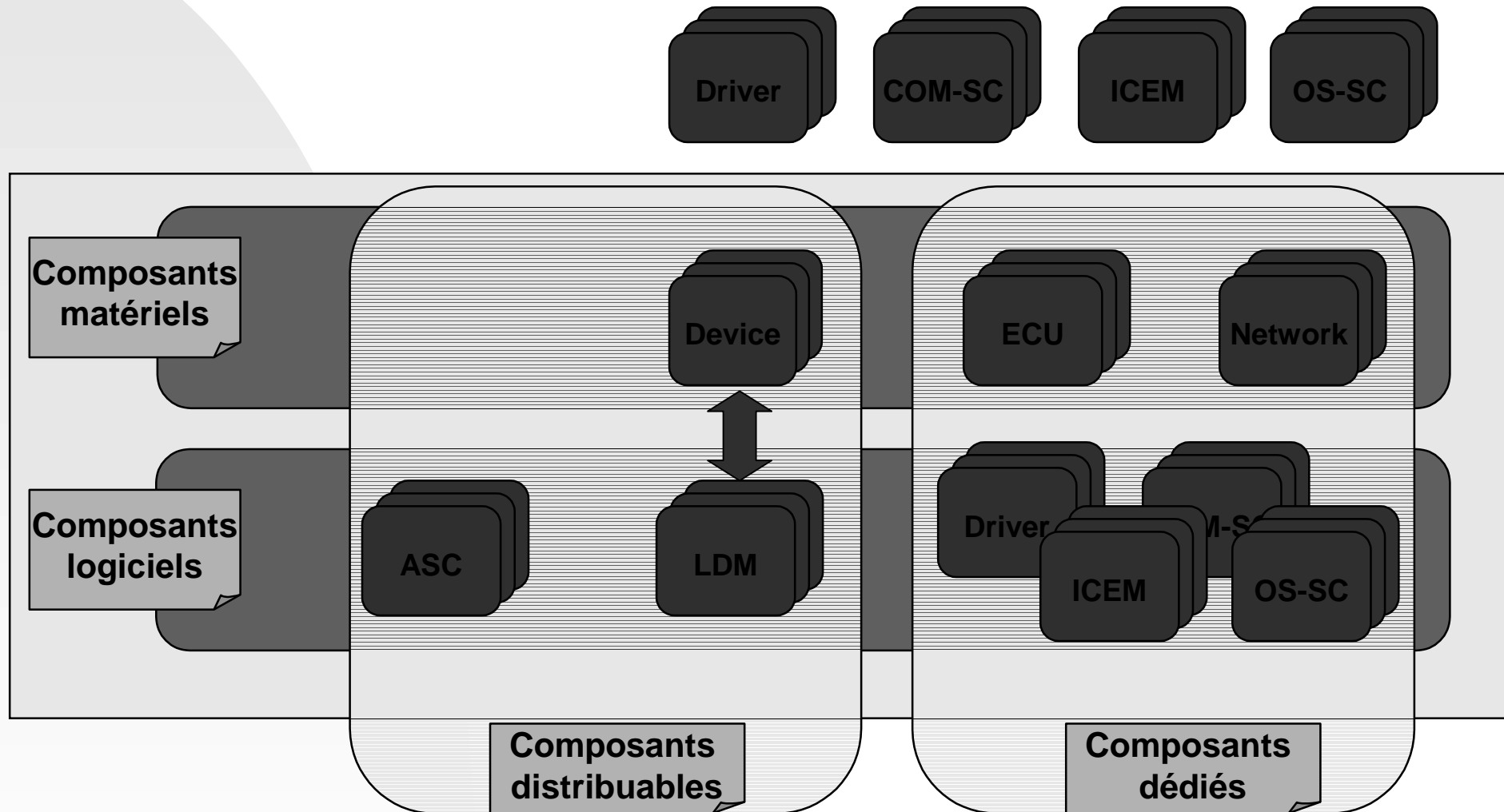
- **Documentation**

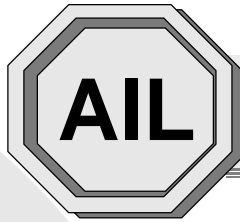
-- > **Modélisation comportementale**

-- > **Vérification des spécifications fonctionnelles**

connexion à un outil de preuve de comportement / temps

Les composants architecturaux



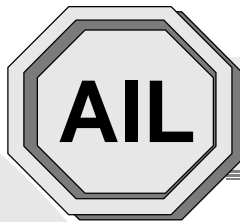


Classes de la couche « Logicielle »



... modélisent la partie du système qui est réalisée sous forme logicielle

- le modèle distingue entre logiciel *allouable* et logiciel *résident*
- la plus petite entité de la couche logicielle logiciel est une *entité de programmation* non sécable entre plusieurs sites
- les *échanges de données* entre les fonctions de la couche fonctionnelle sont dérivés dans la couche logicielle



Classes de la couche « Logicielle »

composant logiciel



SoftwareComponent

données consommées



SoftwareInput

données produites



SoftwareOutput

**organisation des
composants logiciels**

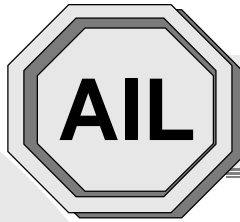


***Allocatable Components
Architecture***

**connexion entre
composants
logiciels**



SoftwareFlow



Classes de la couche « Logicielle »

Un composant logiciel

Software component

... peut être un **composant distribuable**



Free Software component

c'est-à-dire un :

• **composant applicatif**



ASC

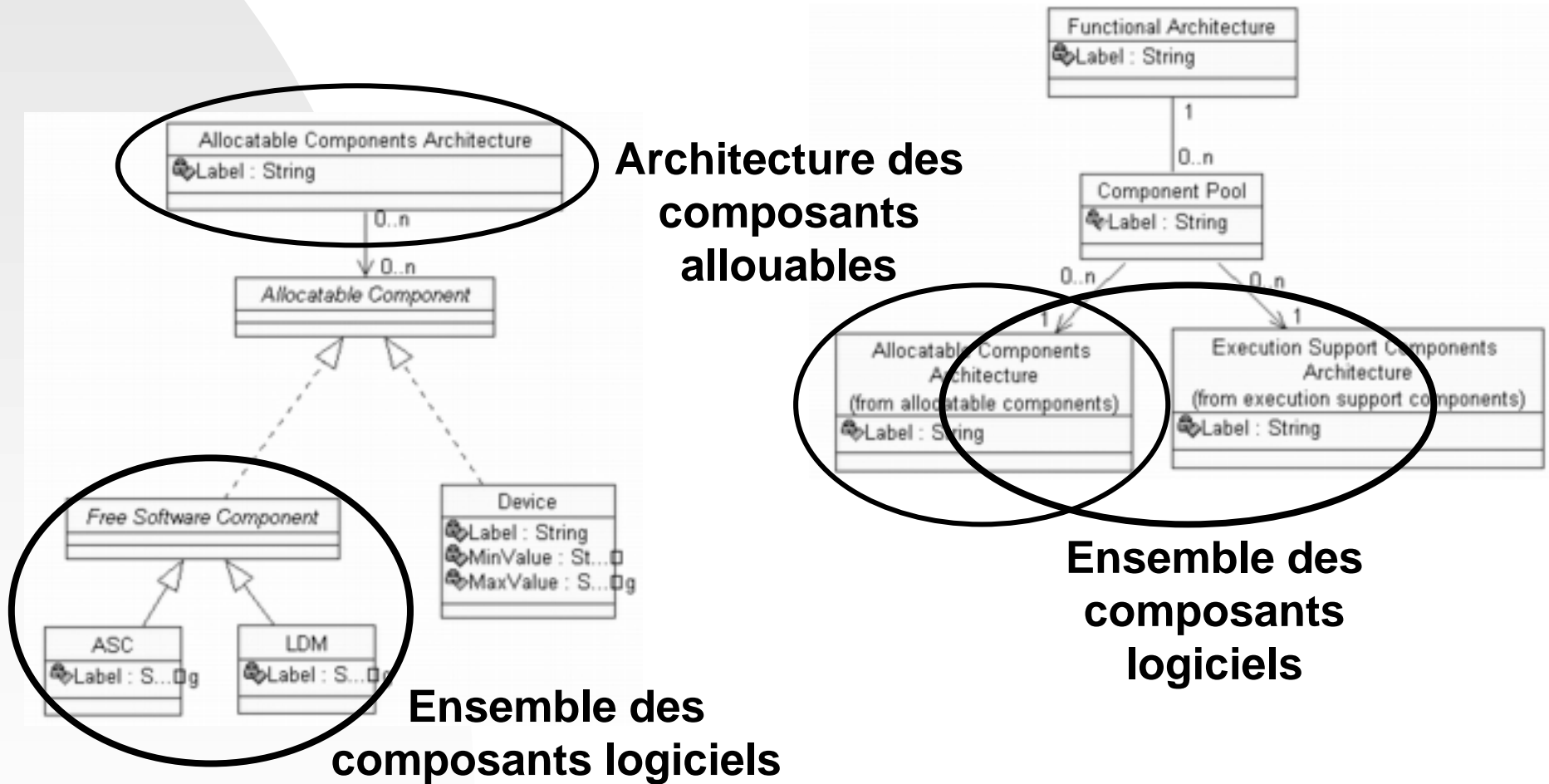
• **gestionnaire local de capteur ou actionneur**

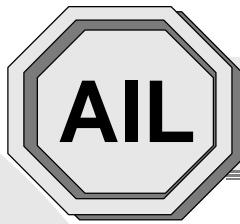


LDM



Classes de la couche « Logicielle »





Classes de la couche « Logicielle »

Un composant logiciel

Software component

... peut être aussi un composant dédié à un calculateur, c'est-à-dire un :

**composant du
gestionnaire d'échange**



ICEM

pilote d'E/S et réseau



Driver

COM SC

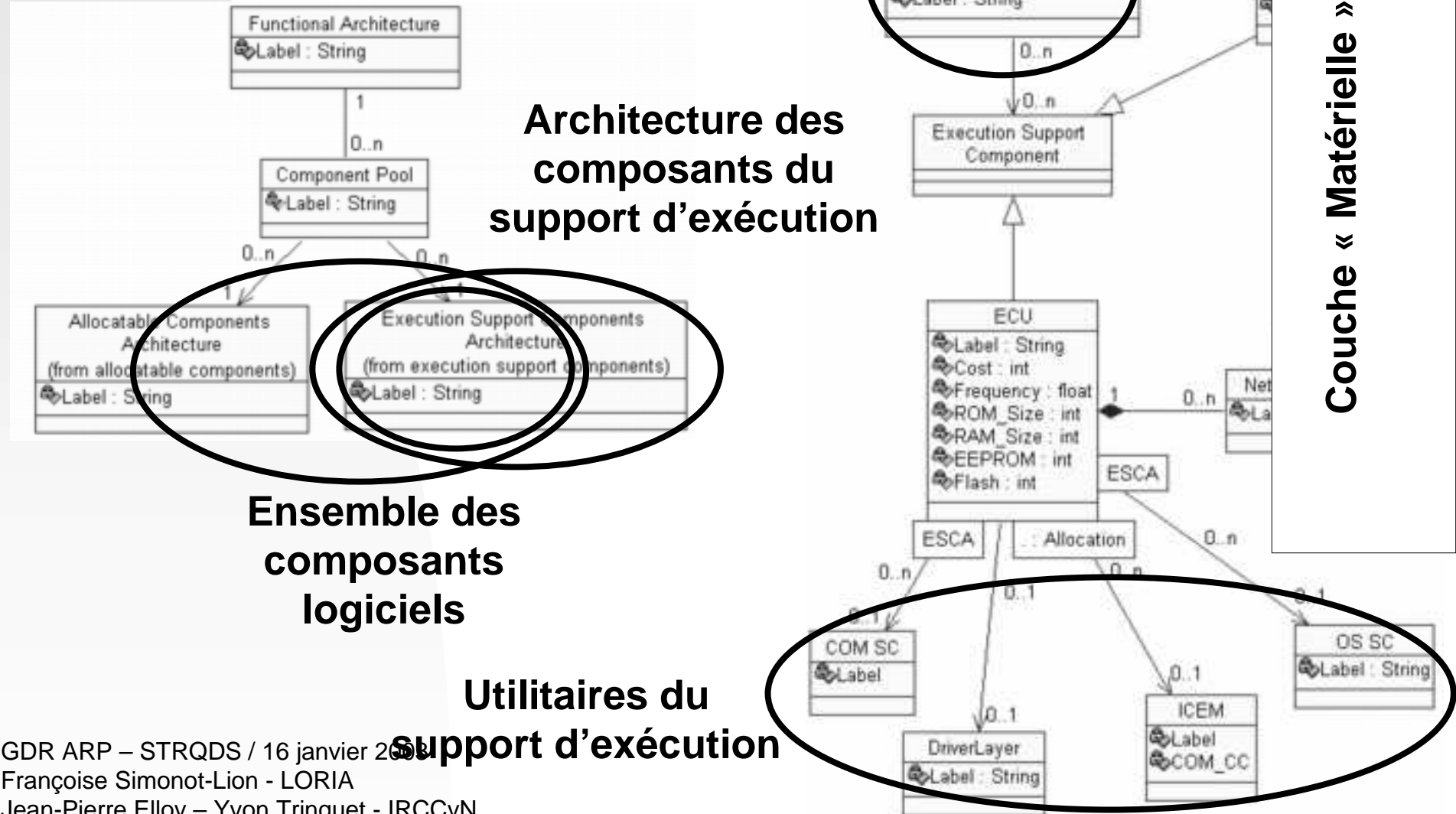
**composant du système
d'exploitation**

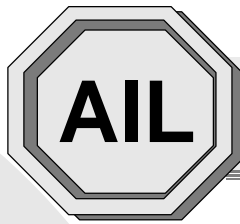


OS SC



Classes de la couche « Logicielle »





Classes de la couche « Logicielle »

Un composant logiciel

Software component

est constitué de:

**entités de
programmation**

...caractérisées par :



Logical Task

mode d'activation

WCET/type de processeur

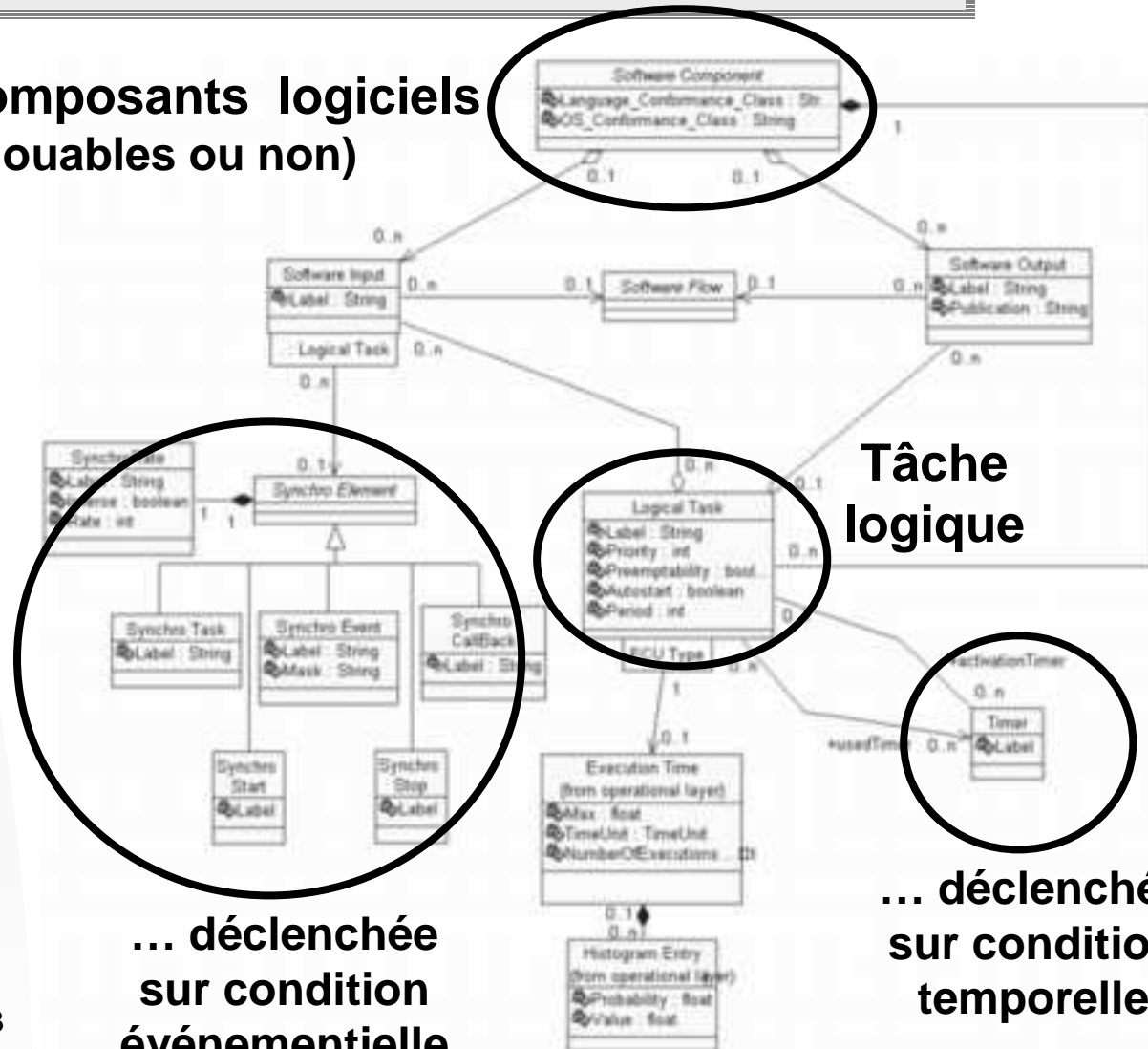
paramètres d'ordonnancement

...



Classes de la couche « Logicielle »

Tous composants logiciels
(allouables ou non)



... déclenchée
sur condition
événementielle

Tâche
logique

... déclenchée
sur condition
temporelle

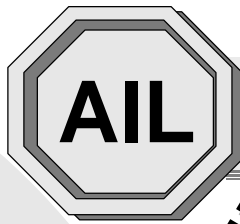
Activités sur la couche « Logicielle »

- **Vérification de cohérence**

*exemple: chaque donnée produite par une fonction (objet **Function**) de la couche fonctionnelle est produite par une tâche logique (objet **Logical Task**) dérivée de cette fonction.*

- **Documentation**

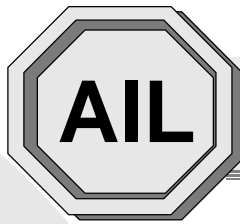
-- > **Génération du code source des composants logiciels**



Classes de la couche « Matérielle »

... modélisent les éléments matériels du support d'exécution de l'architecture, les capteurs et les actionneurs

- Les **calculateurs** (micro-contrôleurs) et les **réseaux**
- Les **connexions** des calculateurs aux réseaux
- Les dépendances technologiques entre **capteurs** et entre **actionneurs**



Classes de la couche « Matérielle »

calculateur



ECU

réseau



Network

**connexion de
calculateur sur
réseau**



Network Connection

**capteur et
actionneur**

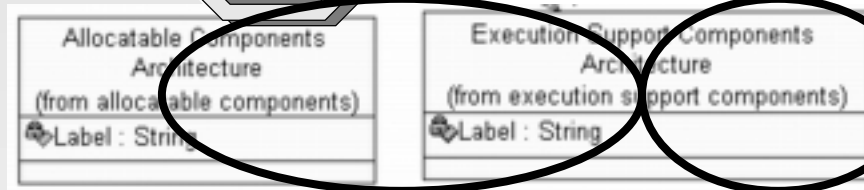


Device

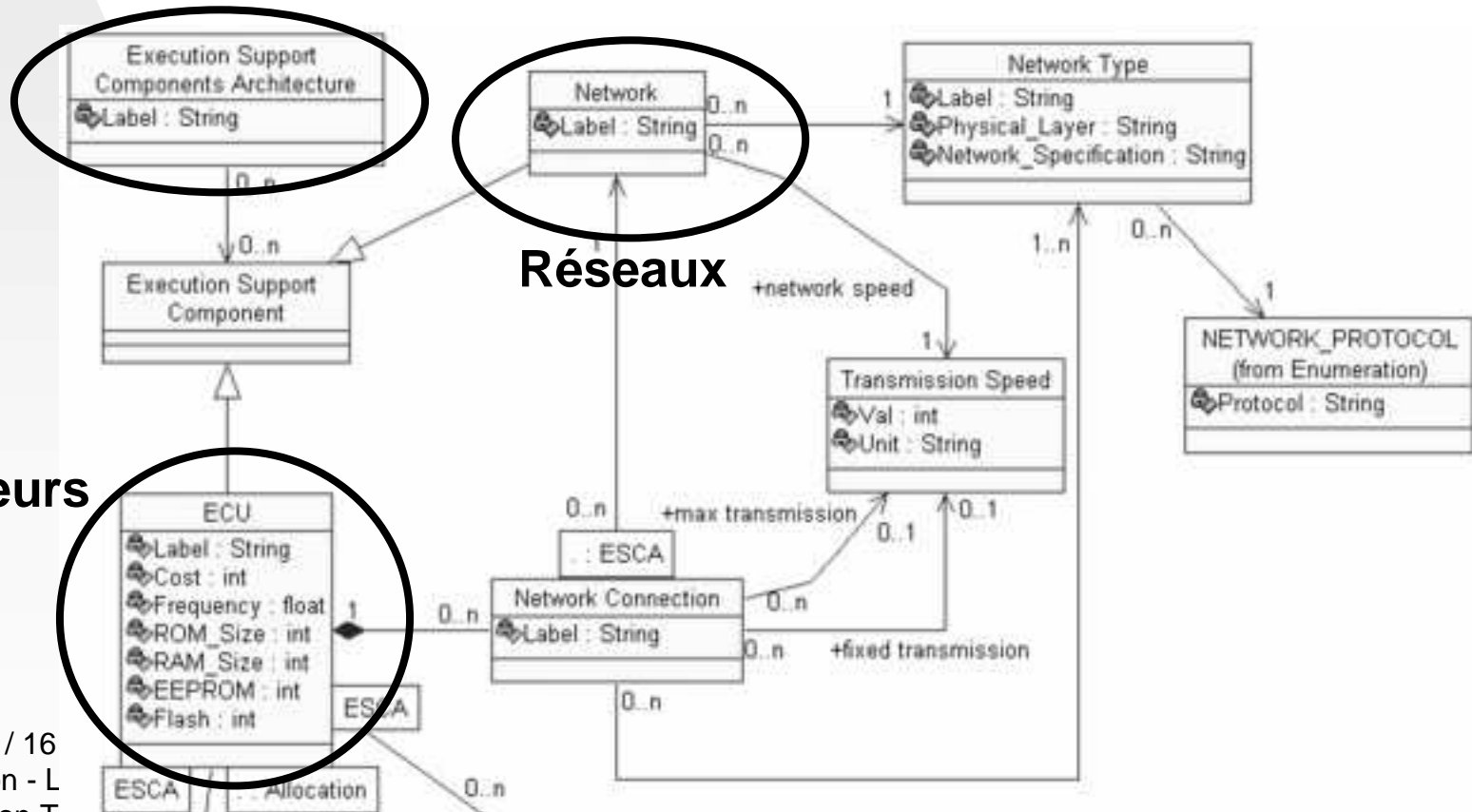
Equipment



Classes de la couche « Matérielle »



Composants matériels du support d'exécution

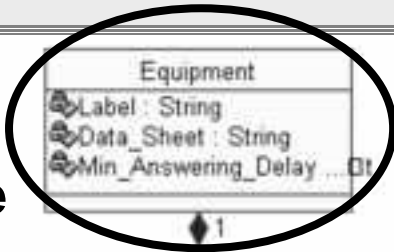


Calculateurs

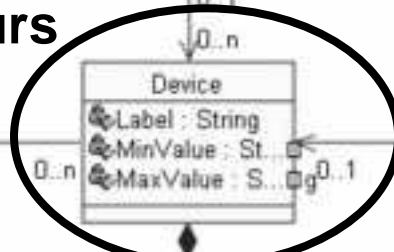
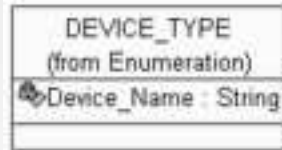


Classes de la couche « Matérielle »

Dépendance technologique

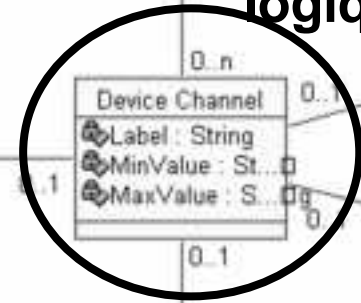
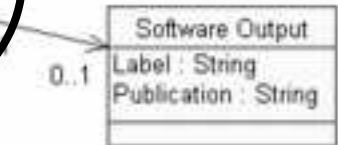
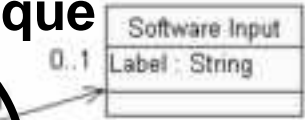
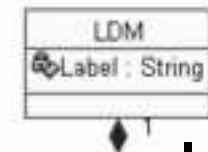


Capteurs et actionneurs



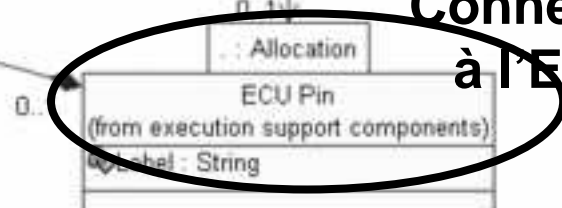
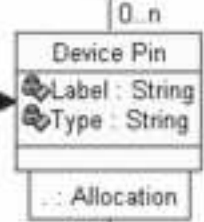
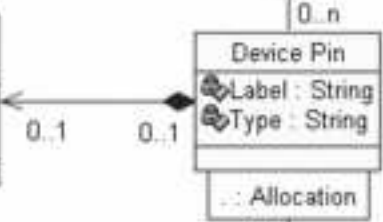
Composants logiciels

Nom logique

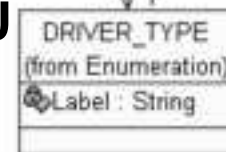
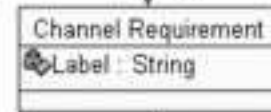


établi après placement

Matériel du support d'exécution

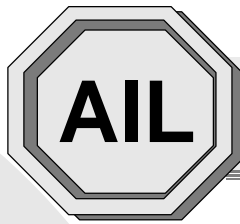


Connexion à l'ECU



Activités sur la couche « Matérielle »

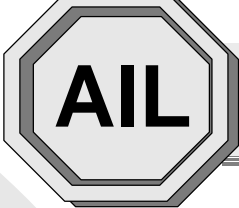
- Vérification de cohérence
- Documentation
 - > Calcul de coûts
 - > Dérivation des plans de câblage
 - > Schéma énergétique et compatibilité CEM



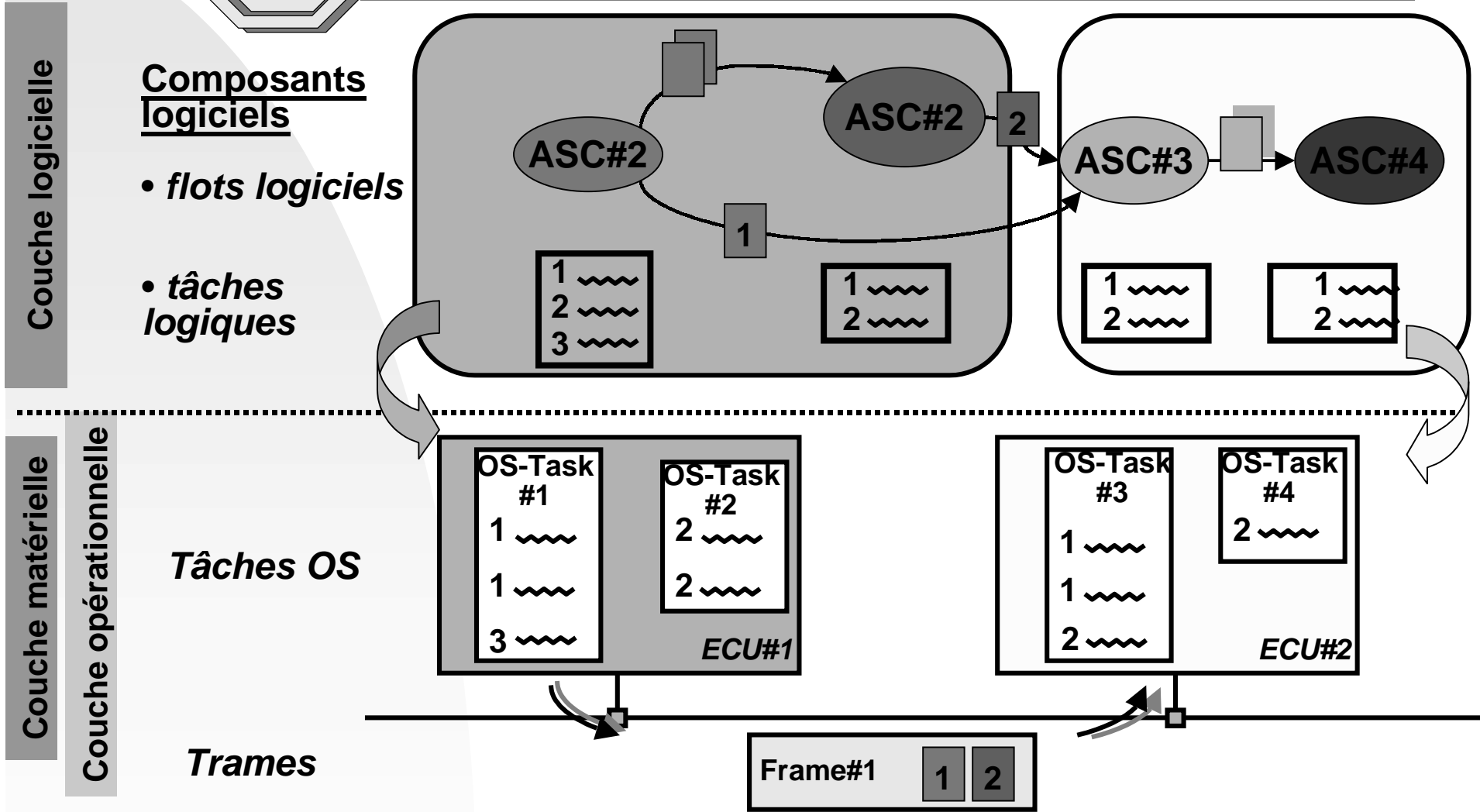
Classes de la couche « Opérationnelle »

... modélisent les entités exécutables après codage, placement et optimisation --> tâches OS

... modélisent la localisation des données et leurs inclusions en trames dans les échanges répartis

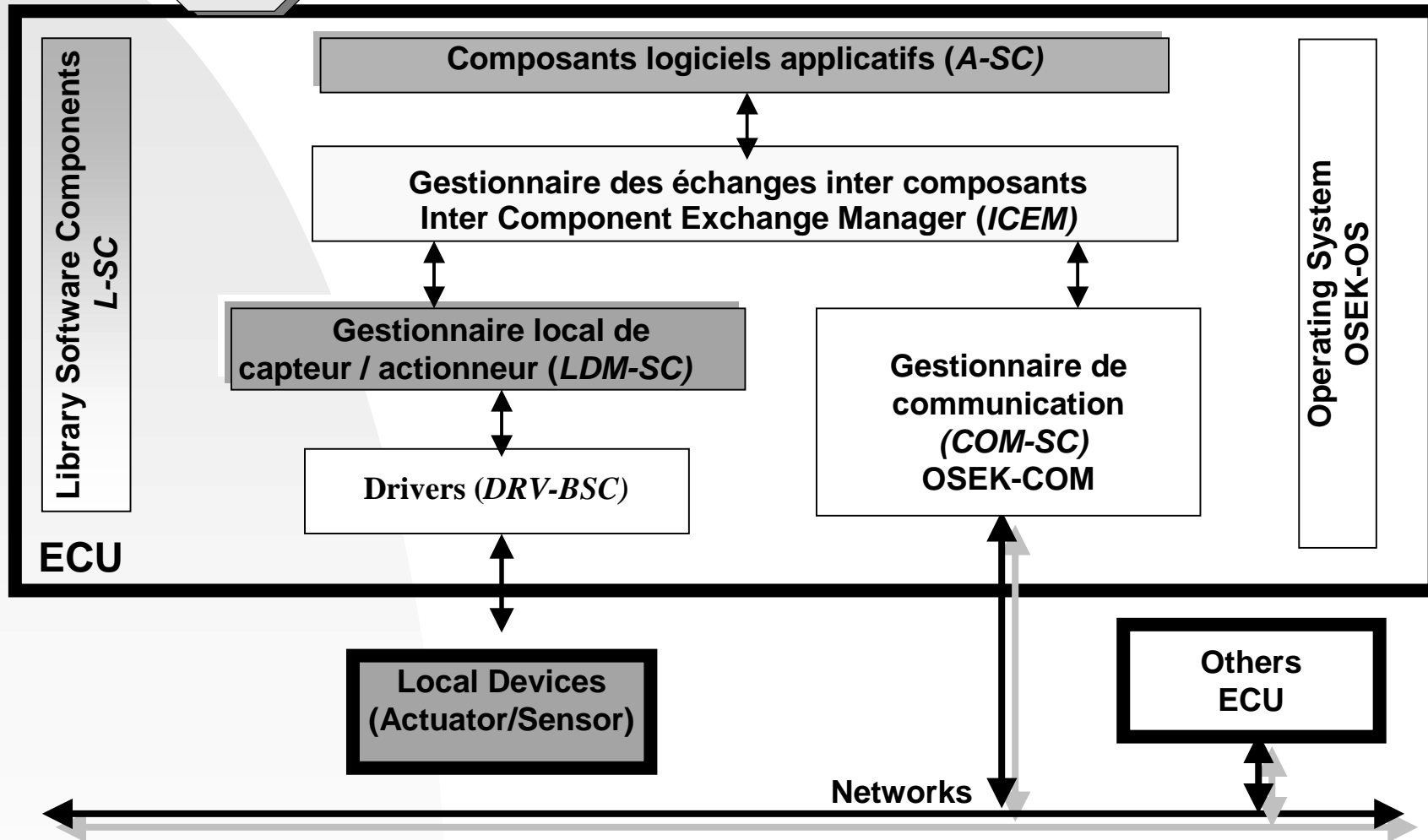


Éléments de la couche « Opérationnelle »



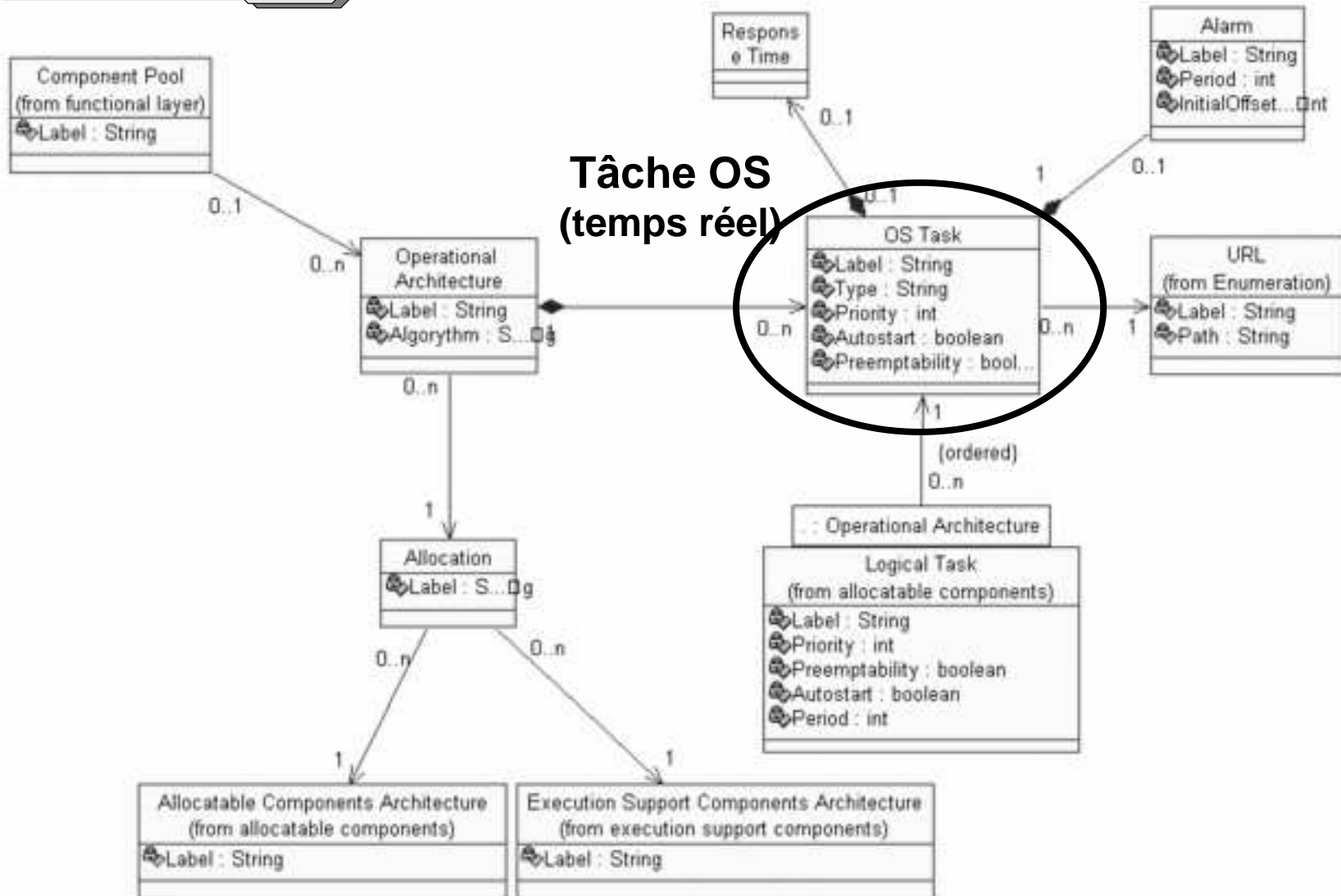


Architecture « Opérationnelle » d'un site



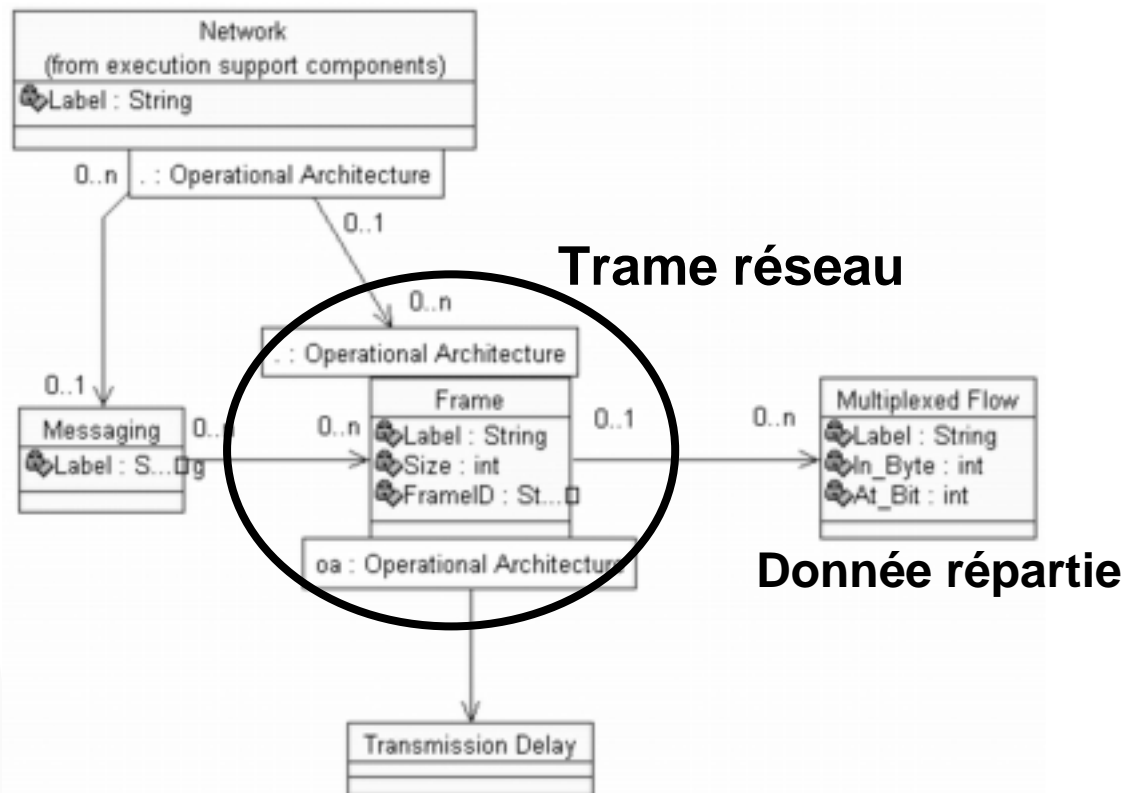


Classes de la couche « Opérationnelle »





Classes de la couche « Opérationnelle »

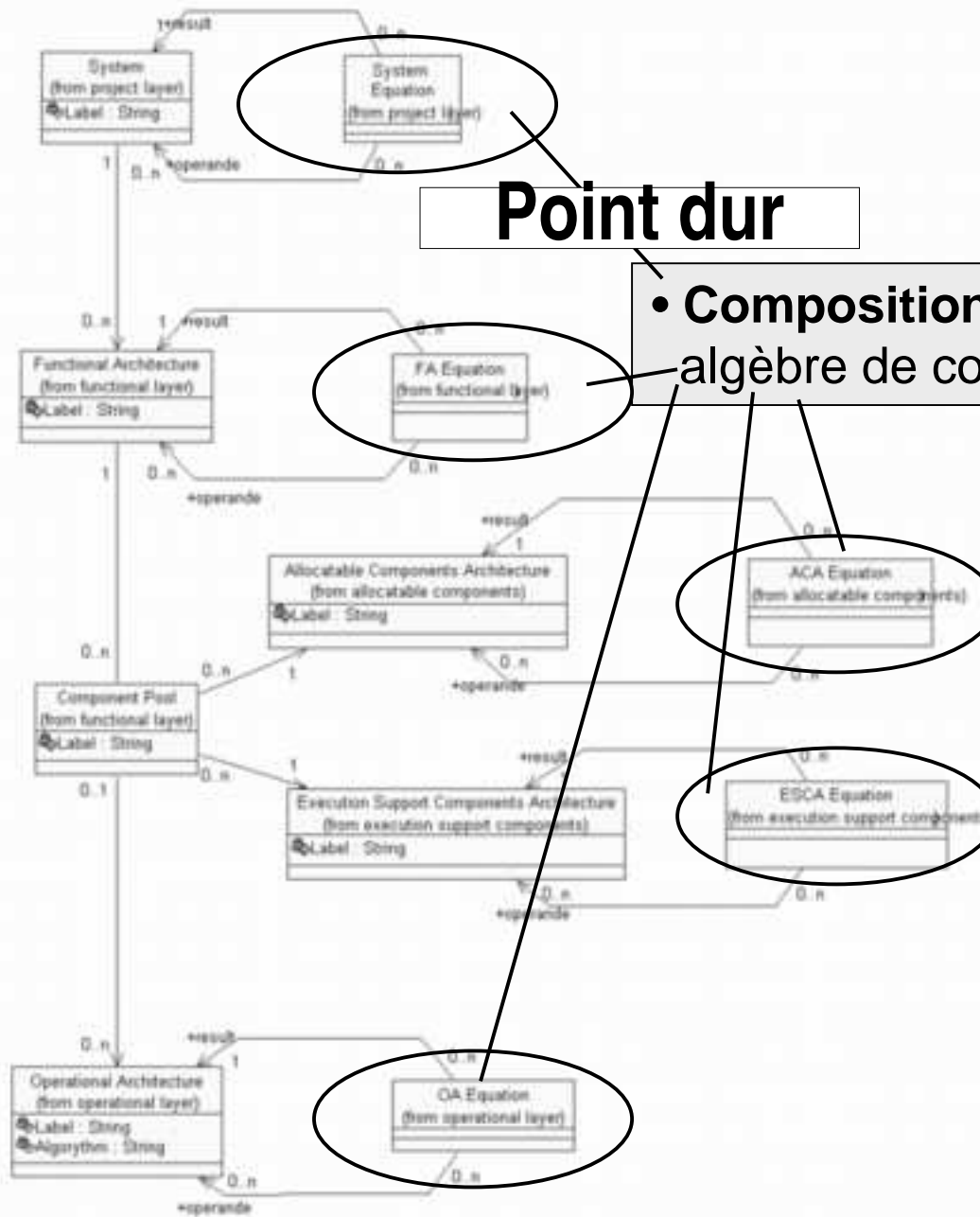


Activités sur la couche « Opérationnelle »

- Vérification de cohérence
 - Placement automatique sous contraintes
 - Validation de propriétés temps réel
(évaluation de performances)
 - Documentation
- > **Génération de vecteur de tests**
-- > **Configuration, déploiement**

Conclusions

- **AIL_Transport : un langage de référence pour modéliser les systèmes électroniques embarqués dans l'automobile**
 - qui assure la maîtrise de la diversité des systèmes
 - qui facilite les échanges constructeurs / équipementiers
- **Validation sur des prototypes (outils de placement, d'évaluation de performances, de documentation, démonstrateurs)**

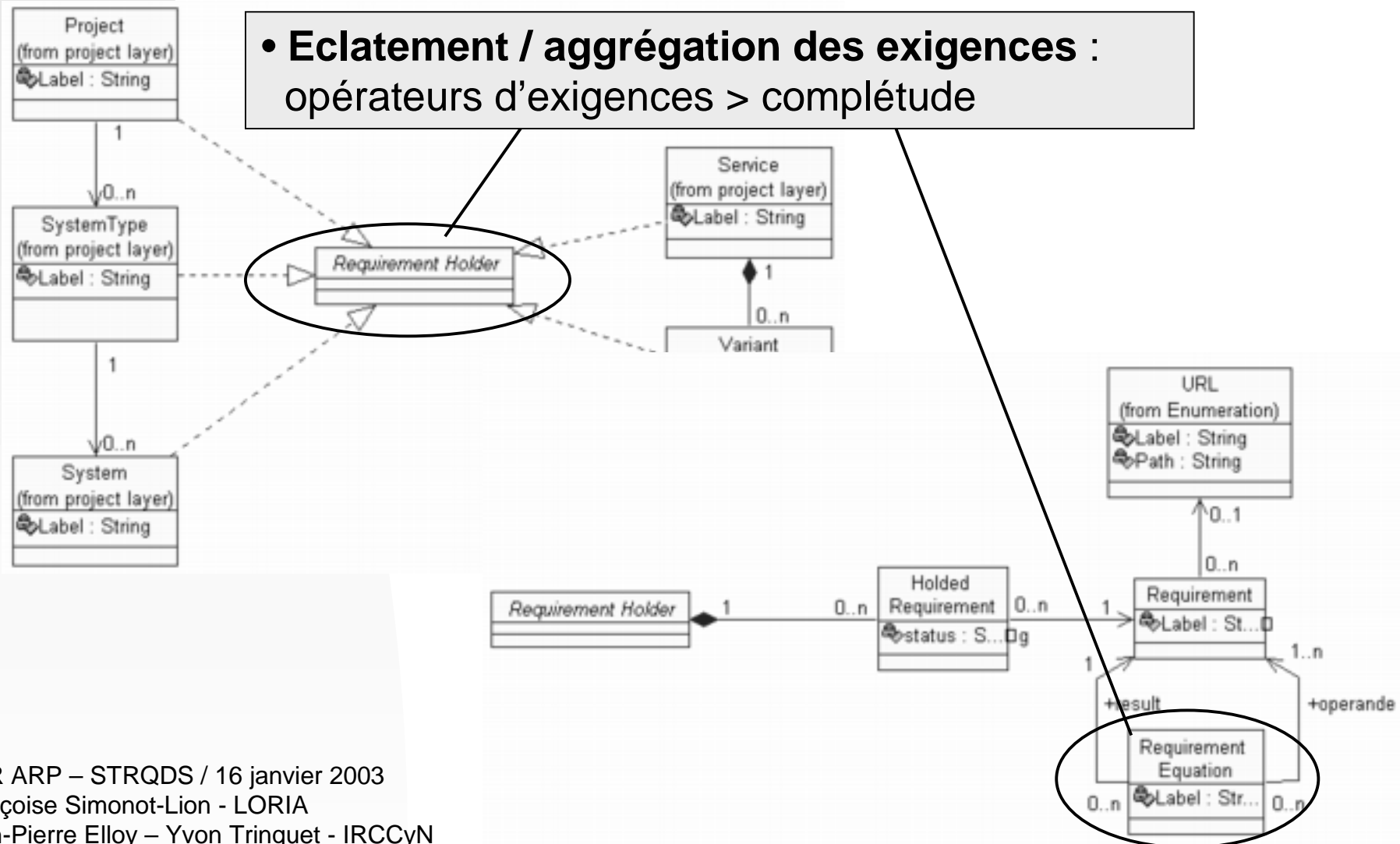


Point dur

- **Composition d'architectures :**
algèbre de composition > cohérence

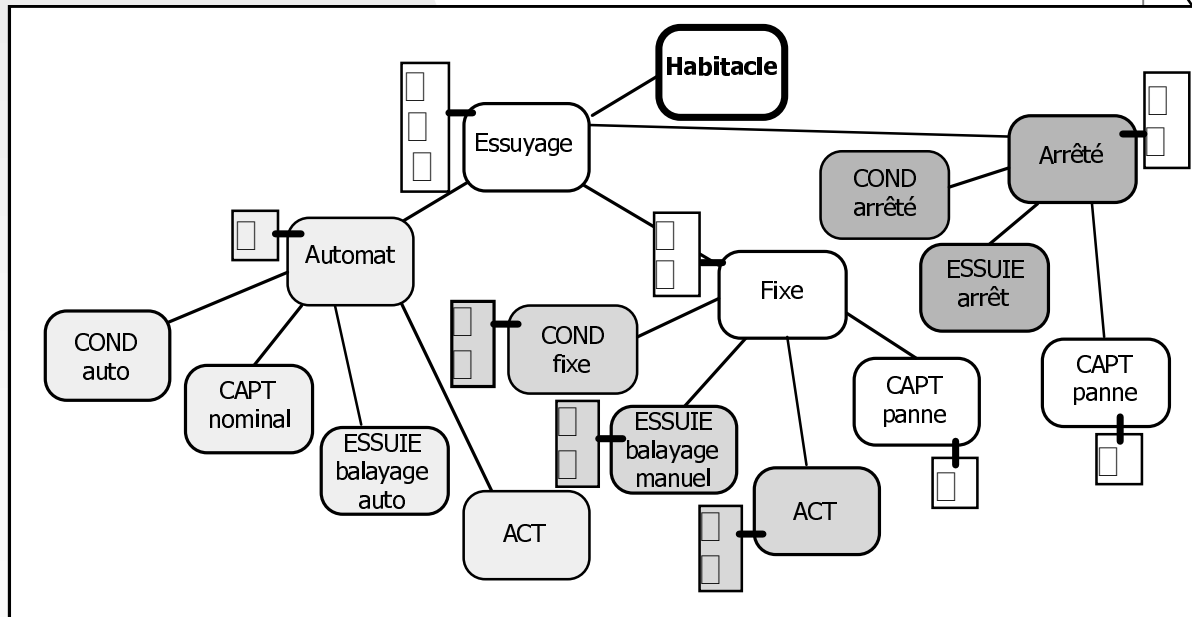
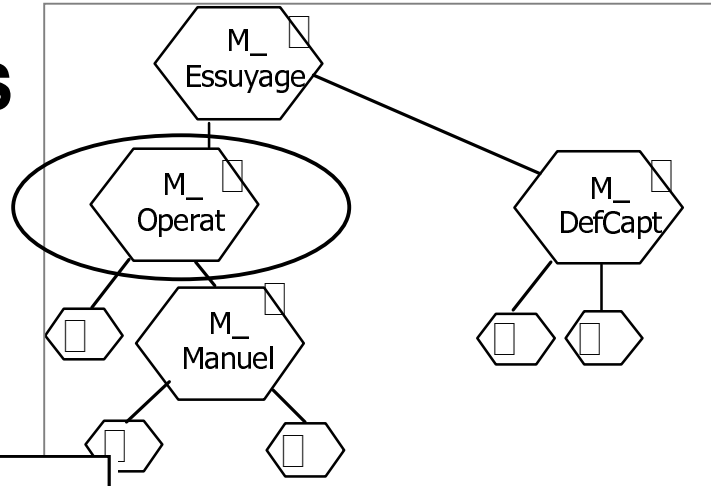
Point dur

- **Eclatement / agrégation des exigences :**
opérateurs d'exigences > complétude

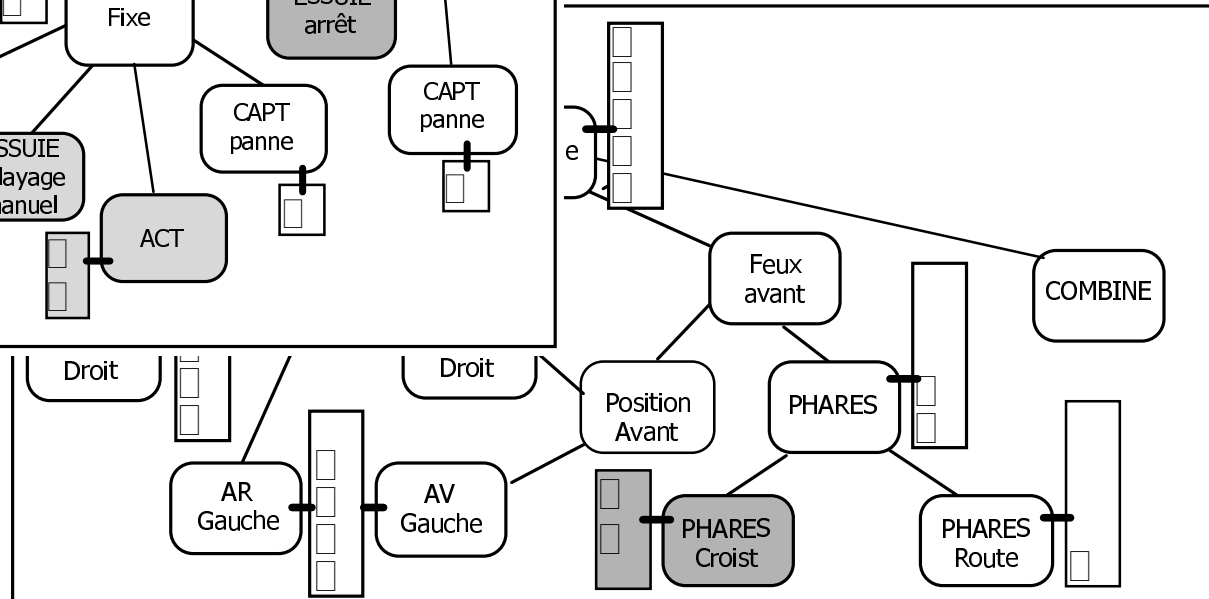
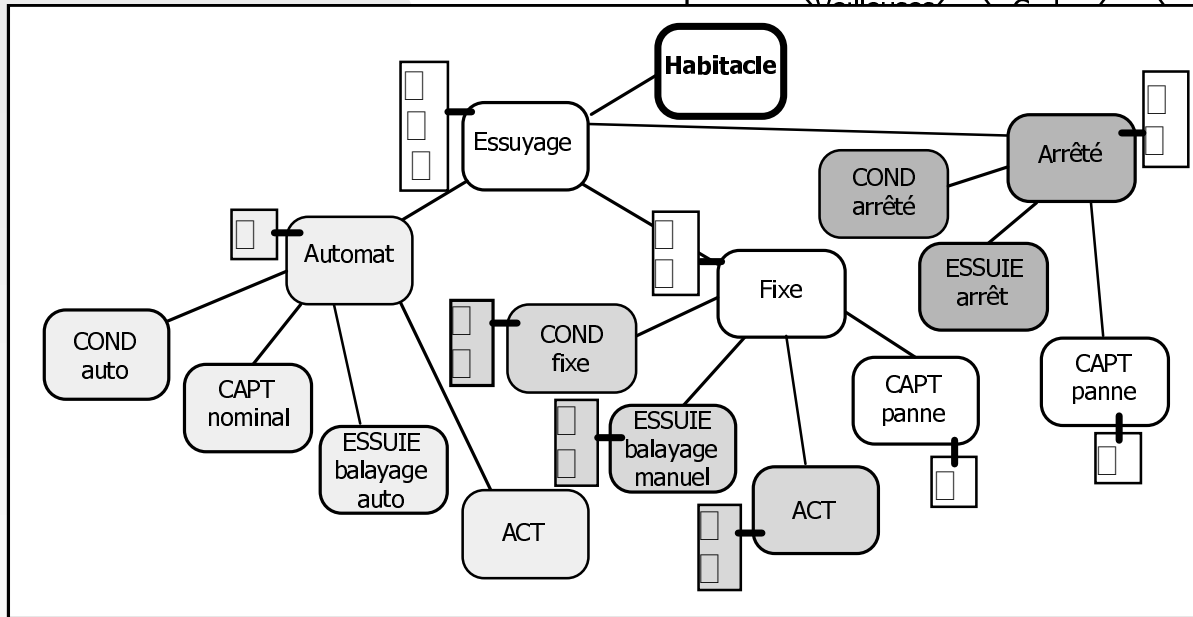
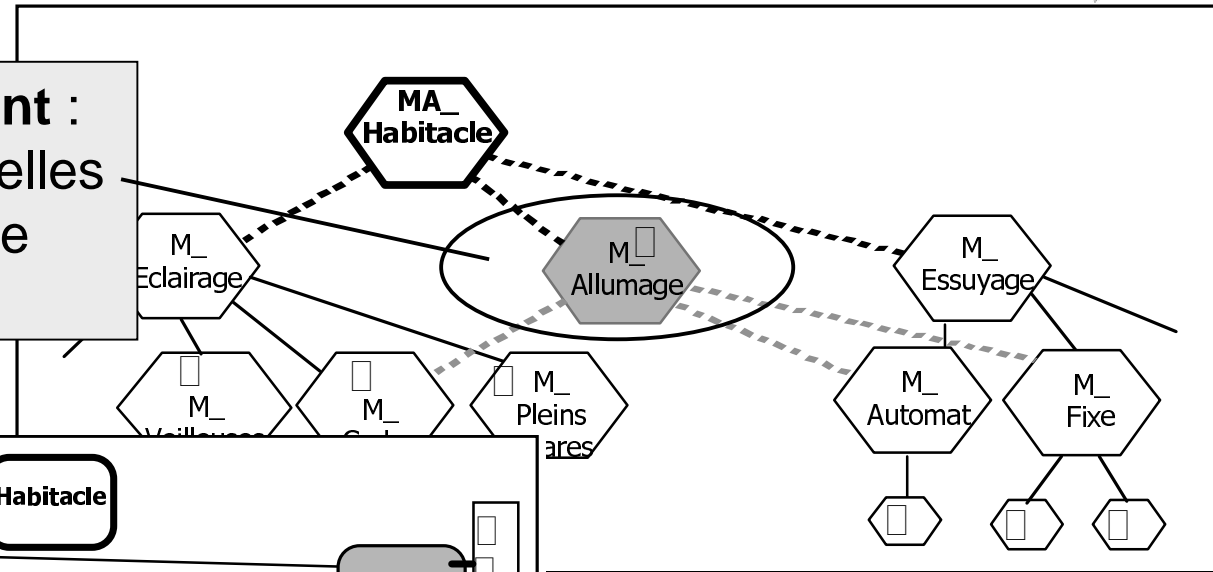


Points durs

- **Modes de fonctionnement :**
synthèse d'inter-gestionnaires



• **Modes de fonctionnement :**
synthèse de modes passerelles
contraints par des modes de
recouvrement

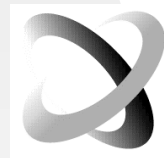


Points durs ...

- Cohérence fonctionnelle, comportementale et temporelle entre la **spécification** qui est décrite dans la couche **fonctionnelle** et la **conception** décrite dans la couche **opérationnelle**.
- Mécanismes de construction associant sous-architectures en **boîtes blanches, grises et noires** et garantissant les propriétés des sous-architectures
- Processus de conception bottom-up avec garantie des propriétés
- Spécification et dérivation des paramètres de **sûreté de fonctionnement** de la spécification à la conception (ex : sécurité niveau 3 -- > temps borné entre deux événements)
- ...

... la suite

- **Standardisation : UML Profile, XML Application**
- **Projet européen ITEA - EAST EEA**



I T E A
INFORMATION TECHNOLOGY
FOR EUROPEAN ADVANCEMENT

