

Sébastien Gérard

Sebastien.Gerard@cea.fr

00 33 (0)1 69 08 90 57

Directeur de Thèse : François Terrier

profil UML pour les systèmes embarqués de l'automobile



PSA PEUGEOT CITROËN

Plan de la présentation

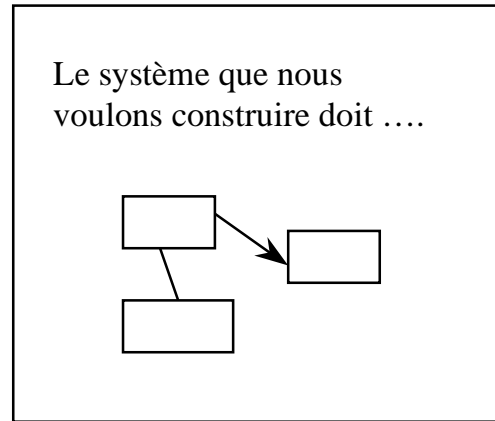
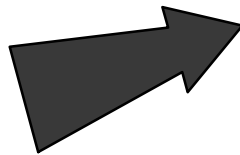
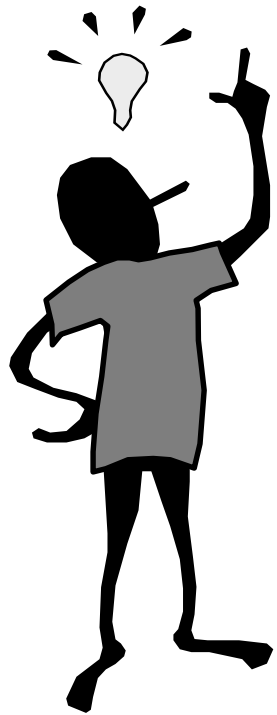
✓ Contexte et besoins

- ⊗ Contraintes imposées par le partenaire industriel
- ⊗ Choix définissant le cadre de travail

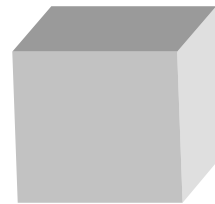
✓ Présentation de la notion de profil UML

✓ Le profil ACCORD/UML

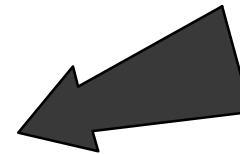
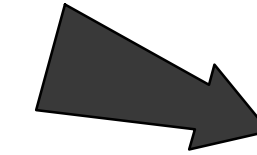
Contexte de travail avec PSA



Spécification



Produit



Fournisseurs

Donneur d'ordre

Automobile : Électricité et électronique

1993 : 12% 2005 : 25%

Quelques fonctions pilotées

Mobilité

Traction

Contrôle moteur
Transmission

Tenue de route

Suspension
Contrôle de stabilité

Direction

Direction assistée

Sécurité

Passive

Système d'airbags
Prétensionneurs

Active

Freinage
Anticollision

Intrusions

Condamnations

Confort

Commandes

Equipements
Véhicule

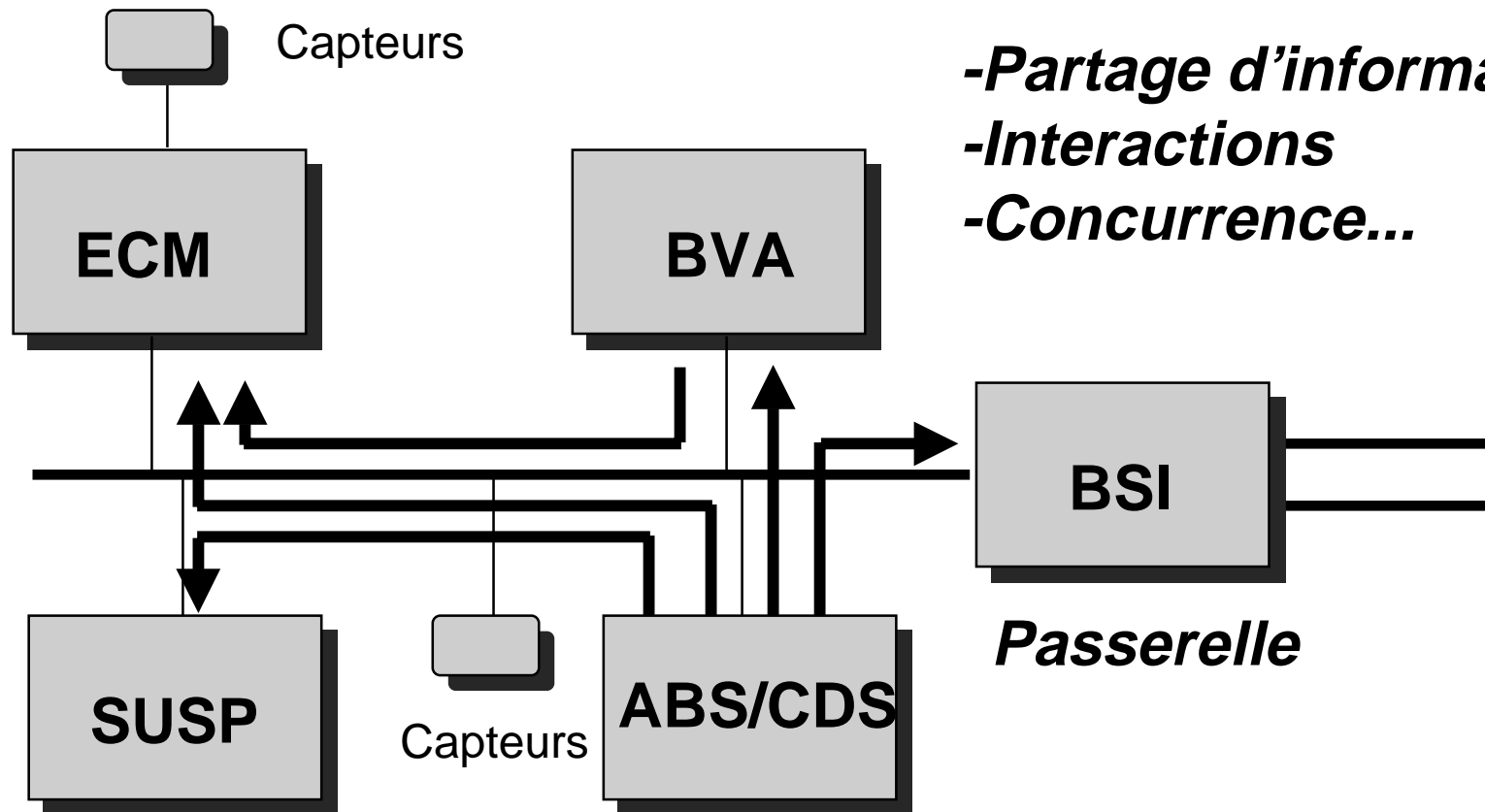
Communication

Instruments
Audio, Téléphone
Aide à la navigation

Environnement

Climatisation
Bruit

Complexité des systèmes



- Partage d'informations*
- Interactions*
- Concurrence...*

Définition des besoins et objectifs

✓ **Exprimer des besoins non-ambigus**

- Compréhension entre donneur d 'ordre et fournisseurs

✓ **Maîtriser toute la chaîne de développement**

- Spécification du système
- Implémentation sur cibles embarquées
- Validation des modèles produits

→ Diminuer les erreurs (ou oublis)

→ Évaluer les coûts de réalisation

→ Évaluer les coûts de modification

Plan de la présentation

✓ Contexte et besoins

- ⊗ Contraintes imposées par le partenaire industriel
- ⊗ Choix définissant le cadre de travail

✓ Présentation de la notion de profil UML

- ⊗ Définition des extensions de UML
- ⊗ Organisation de ces extensions

✓ Le profil ACCORD/UML

Mécanismes d'extension de UML

- ✓ **Stéréotypes (Stereotype)**
 - Ajout indirect d'éléments au méta-modèle
 - « thread » ou « process » sur **Classifier**
- ✓ **Valeurs marquées (Tagged value)**
 - Ajout de propriétés à une méta classe
 - {documentation = “ ... ” } sur **Element**
- ✓ **Contraintes (Constraint)**
 - Ajout ou modification de « Well-Formedness Rules »
 - {destroyed}, {new} ou {transient} sur **Instance**

Mécanismes d'extension de UML

- ✓ **Stéréotypes (Stereotype)**
- ✓ **Valeurs marquées (Tagged value)**
 - Ajout indirect d'éléments au méta-modèle
 - « thread » ou « process » sur **Classifier**
 - Ajout de propriétés à une méta classe
- ✓ **Contraintes (Constraint)**
 - {documentation = " ... " } sur **Element**
 - Ajout ou modification de « Well-Formedness Rules »

Besoin de structuration
{destroyed}, {new} ou {transient} sur **Instance**
χ notion de profil dans UML 1.3

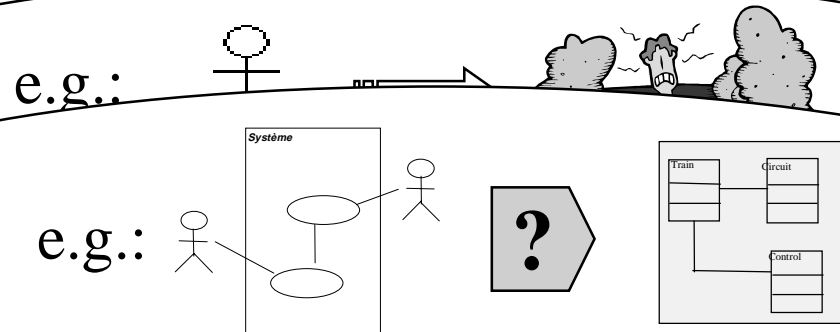
Définition d'un profil UML

✓ Objectif

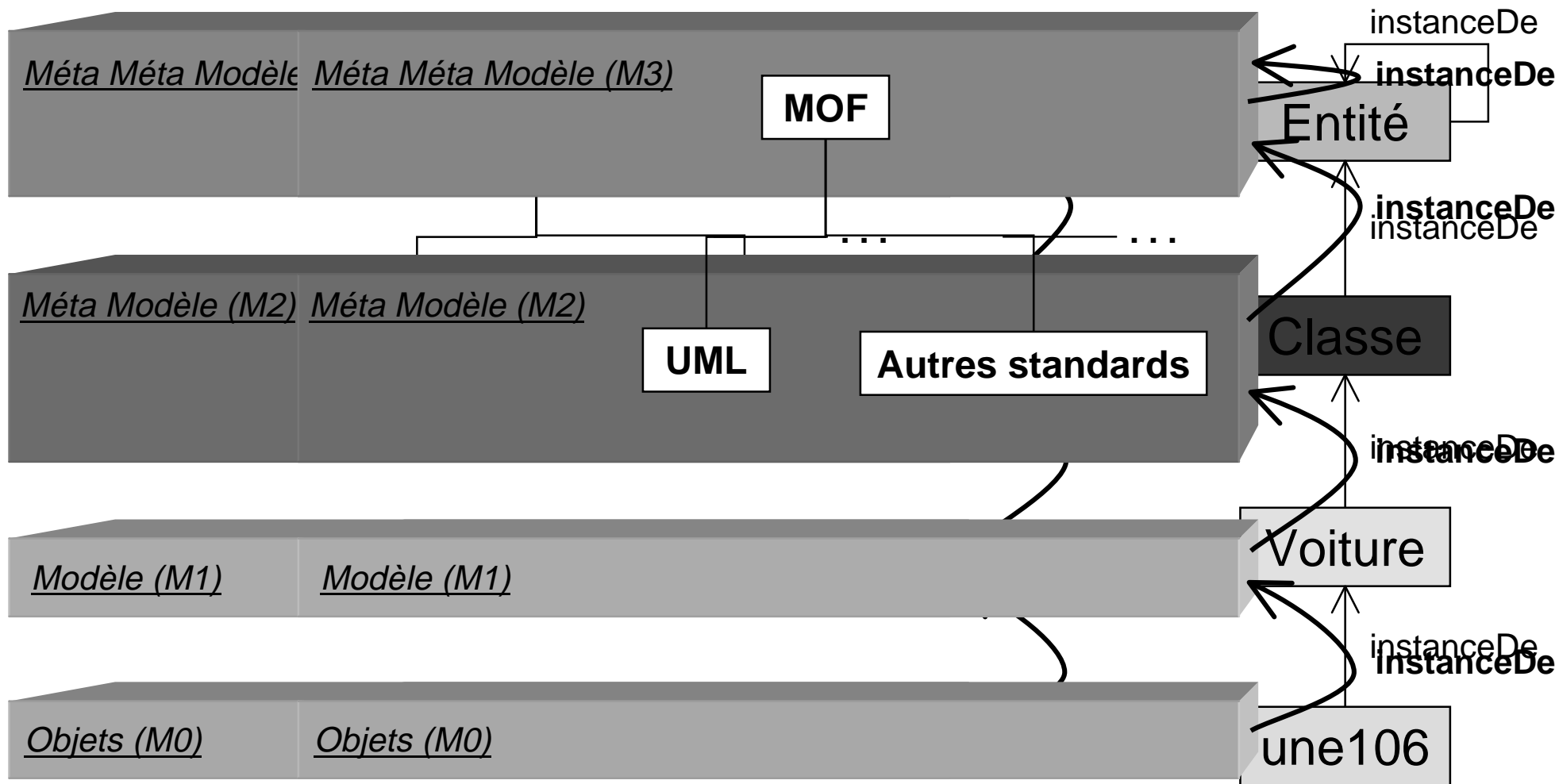
Spécialiser un langage de Modélisation (UML) servant de référence et d'application particulière. Méta-éléments : Stéréotypes, tagged values et « Semantics Variation Points » et

✓ Un profil peut :

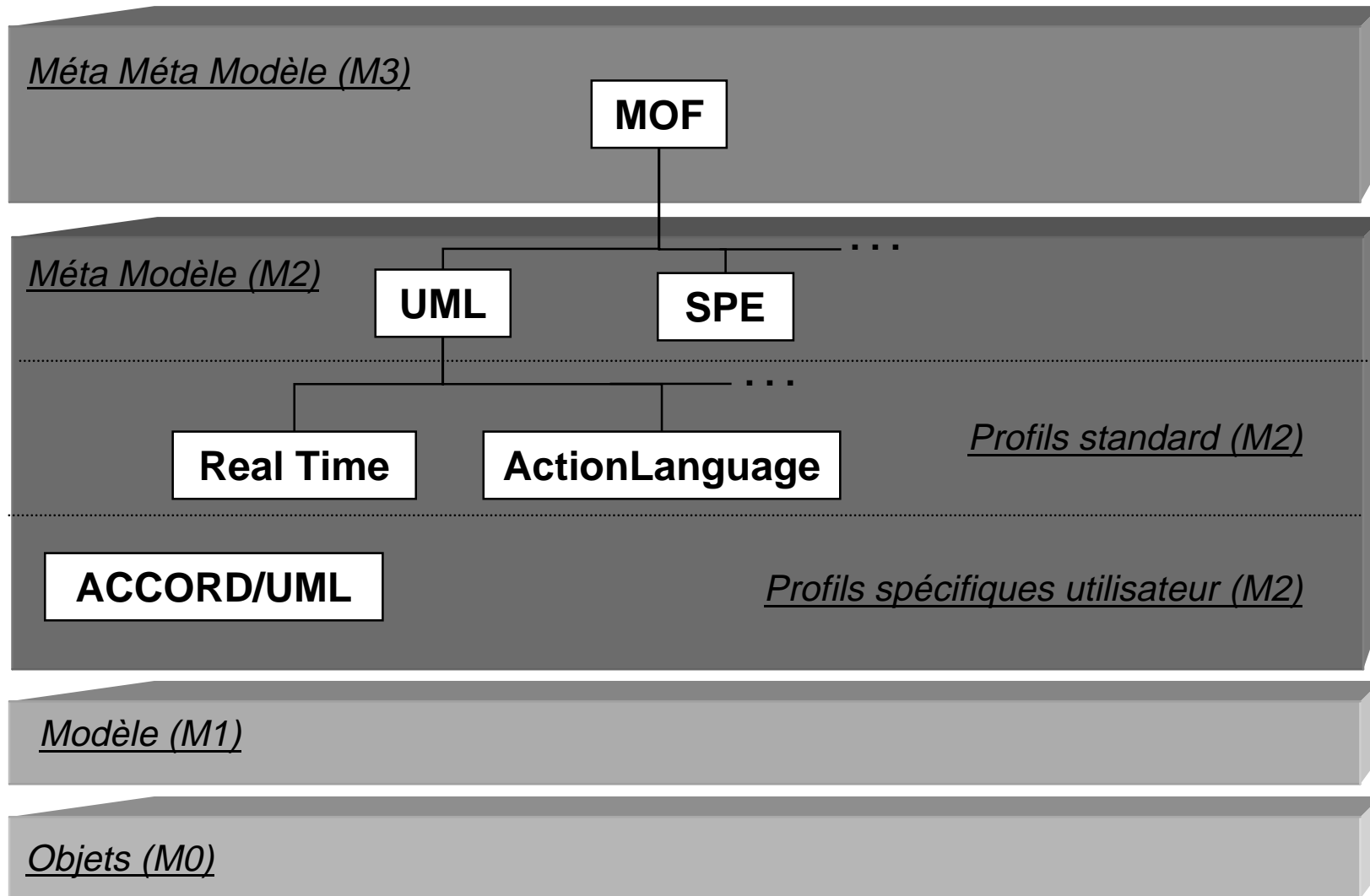
- les éléments sélectionnés
- des mécanismes d'extension
- des descriptions sémantiques
- des notations supplémentaires,
- des règles de transformation, de validation ou de présentation.



Un modèle à quatre couches



Organisation de UML



Exemples issus du profil ACCORD/UML

✓ Règles de présentation

- Choix des diagrammes utilisés dans le contexte du processus de développement ACCORD/UML

✓ Extensions introduites

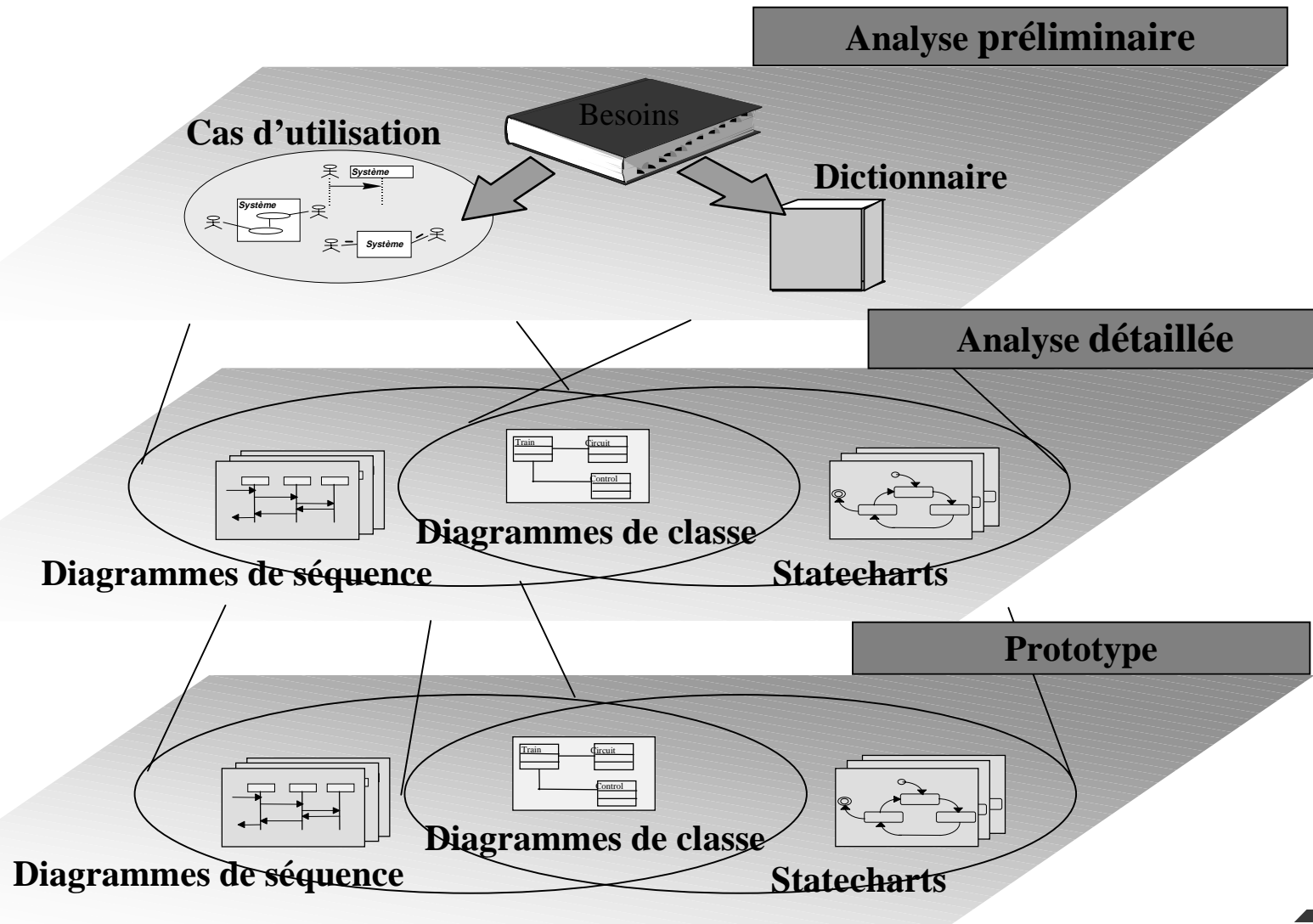
- Le stéréotype « Real Time Object »
- La valeur marquée {RTC=(dl(x,ms), ...)}

✓ Le concept de Signal

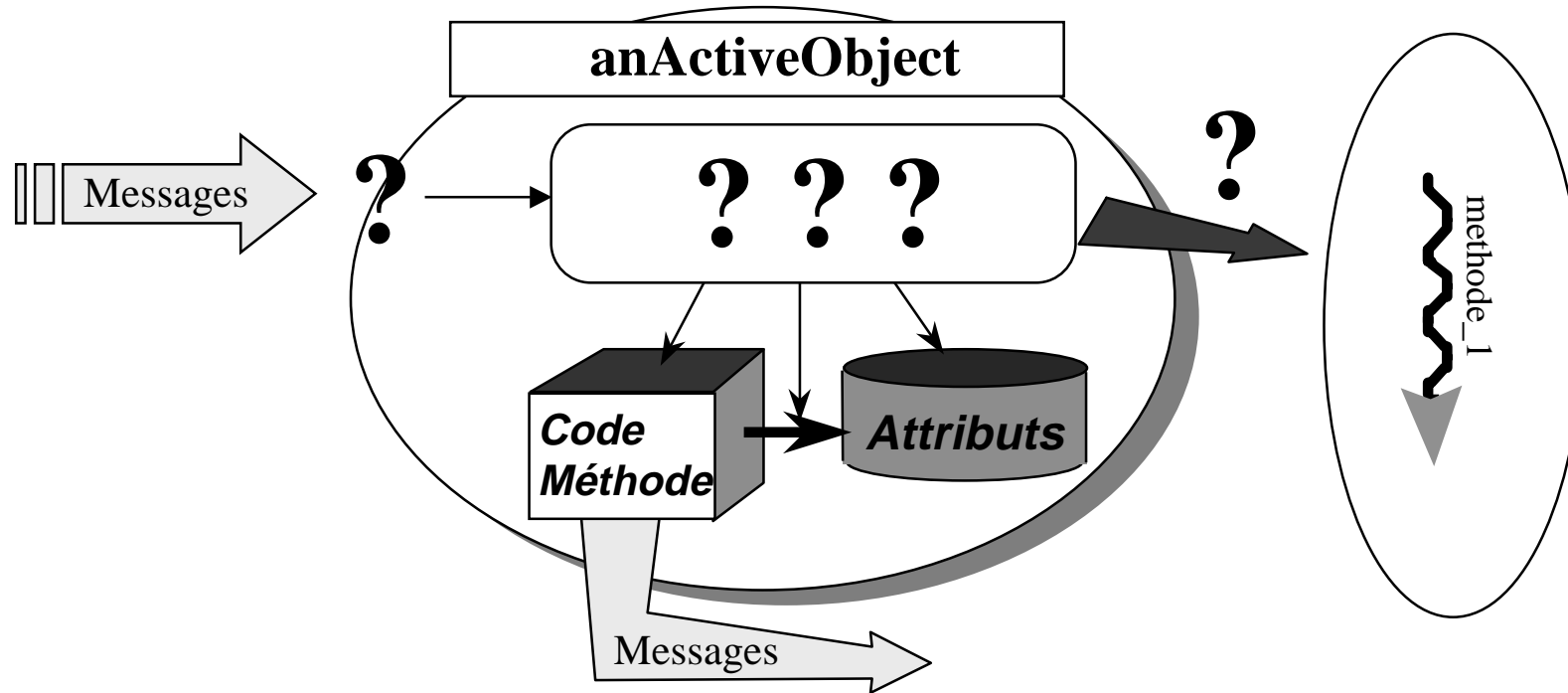
- Aspects sémantiques
- Notations spécifiques introduites
- Génération automatique d'un patron de conception

✓ Génération automatique du code temps-réel

Processus de développement de ACCORD/UML



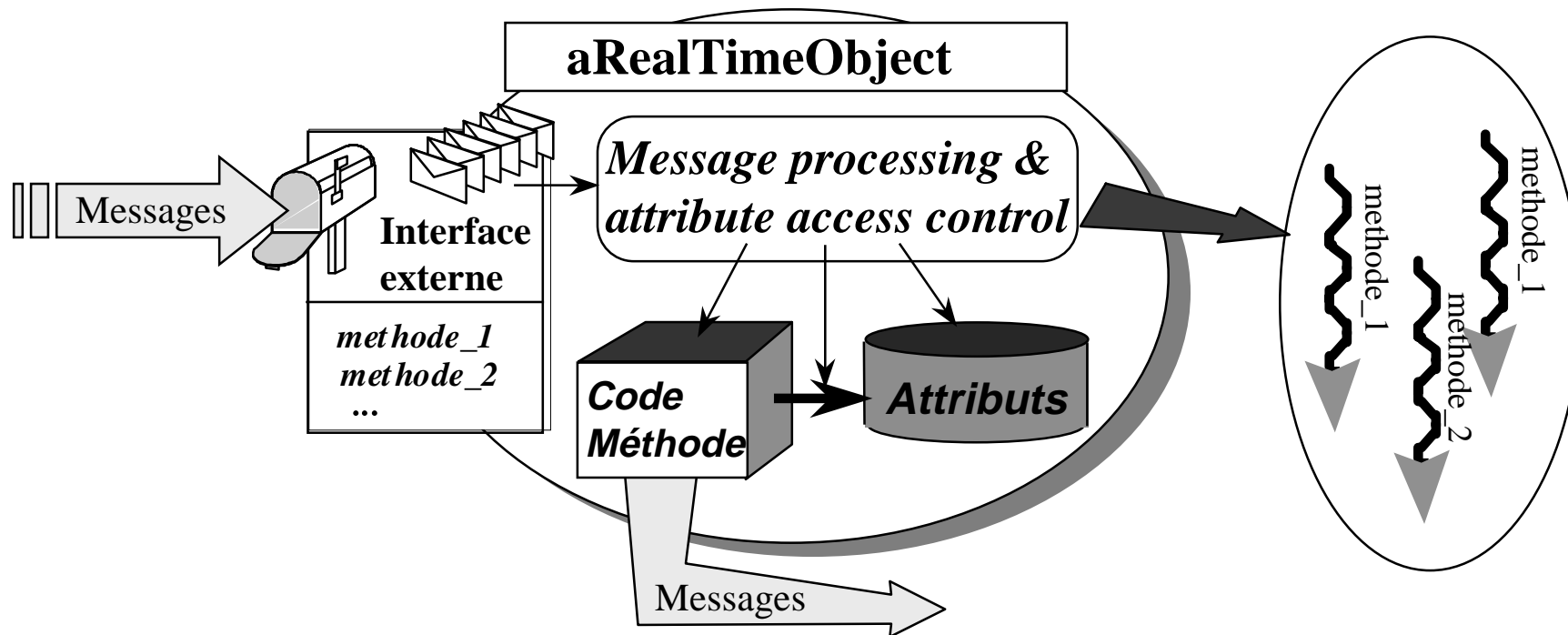
La notion d'objet actif de UML



✓ Caractéristiques principales de l'objet actif de UML

- « mono thread »
- comportement → ?

« Real-Time objects »



✓ Différences majeures avec l'objet actif de UML

- « multi threadé » et « temps-réel »
- comportement clairement et complètement défini

Le concept de message de UML

✓ Définition

Un message définit une communication particulière entre deux instances participant à la réalisation d'une tâche précise.

✓ Un message = une action + un événement

- Appel d'opération (**CallAction** + **CallEvent**)

➔ **Communication synchrone ou asynchrone de type point à point**

- Un signal (**SendAction** + **SignalEvent**)

➔ **Communication asynchrone, ... ?**

Les signaux dans *ACCORD/UML*

✓ Communication de type diffusion

- diffusés à tous les objets déclarés sensibles
- envoi → destinataire inconnu
- réception → émetteur inconnu

✓ La réaction à un signal = exécution d'une méthode

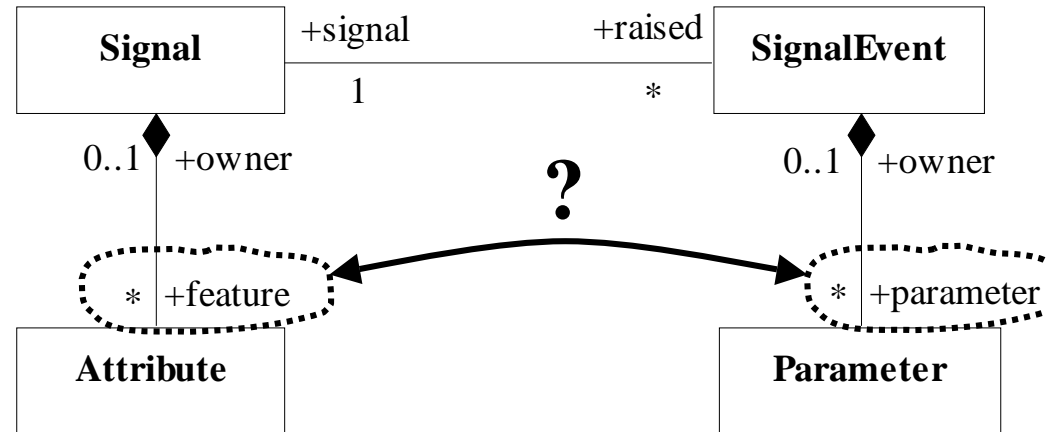
✓ Peut posséder une contrainte de temps

- à l'émission → TV {RTF} sur l'action générant le signal
- à la réception → TV {RTF} sur un événement

Real Time Feature:

{RTF = (dl(xxx, ms), rd(xxx, ms), p(xxx, ms), nbPeriod(xxx), endOfPeriod, outOfTime)}

Ajout de « Well-Formedness Rules »



SignalEvent

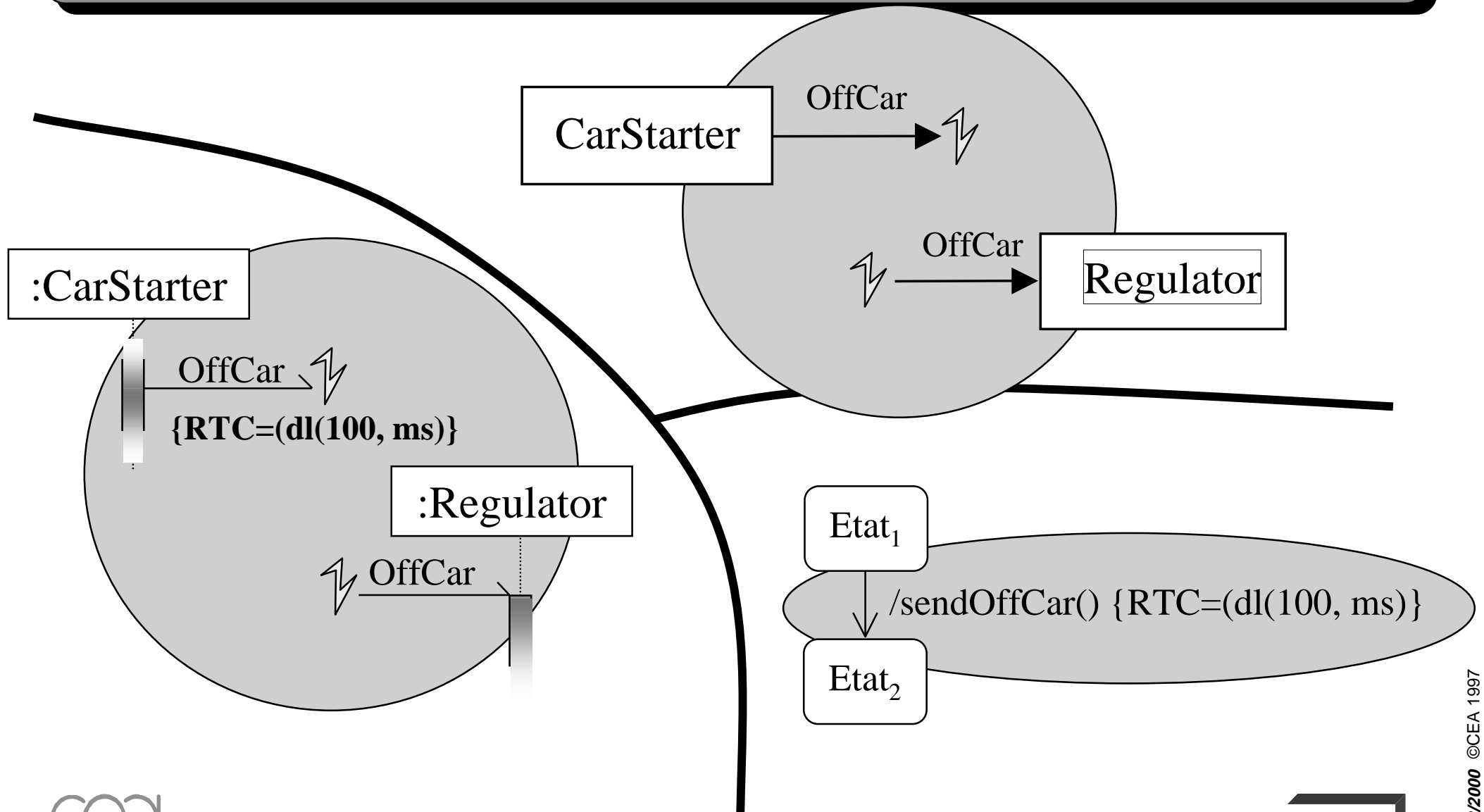
[1] Un événement signal ne possède que des paramètres en entrée.

`self.parameter → forAll (p | p.kind = # in)`

[2] Un événement de type signal possède autant de paramètre que le signal lui même possède d'attributs.

`self.parameter → size = self.signal.allAttributes → size`

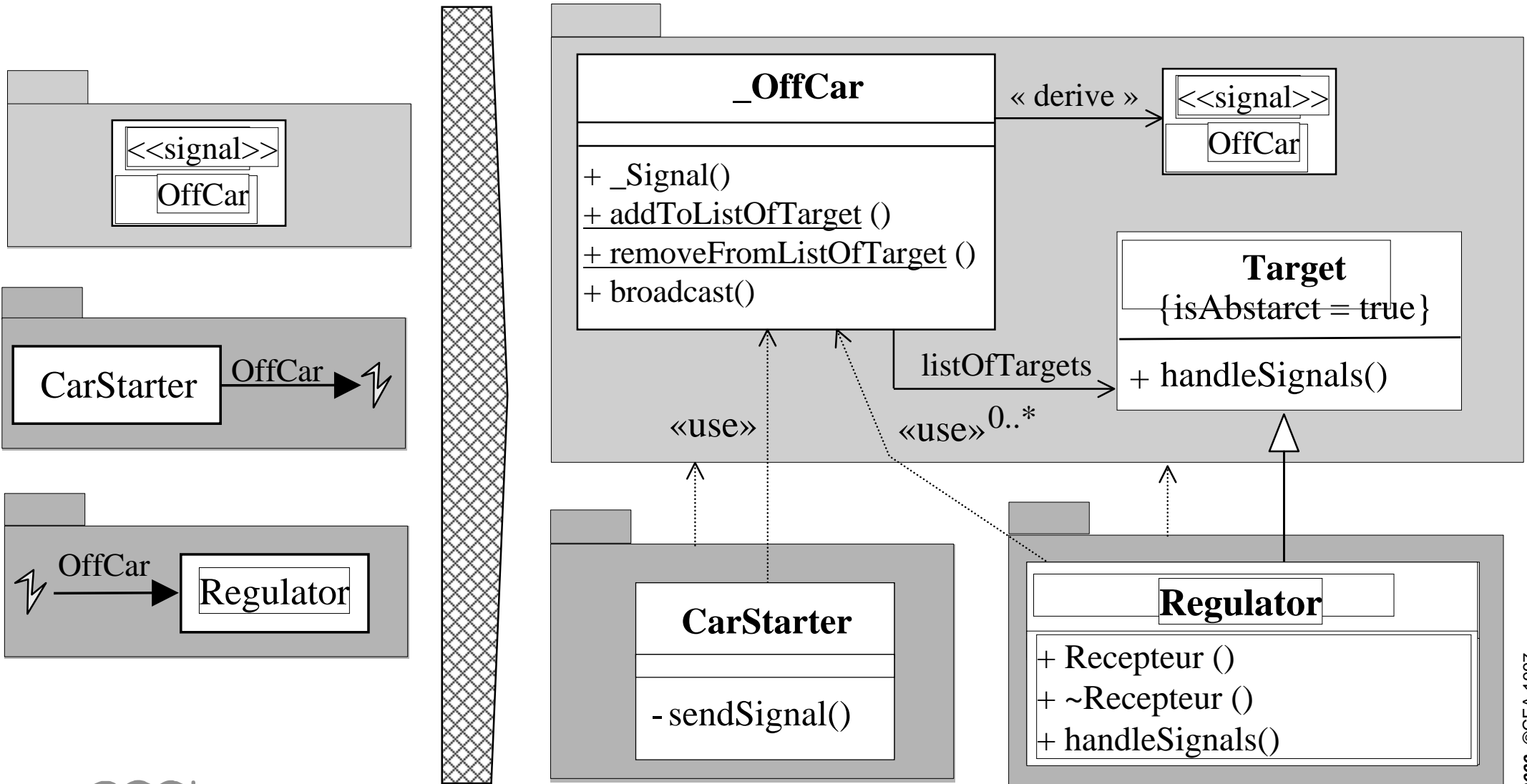
Spécialisation de la notation pour les signaux



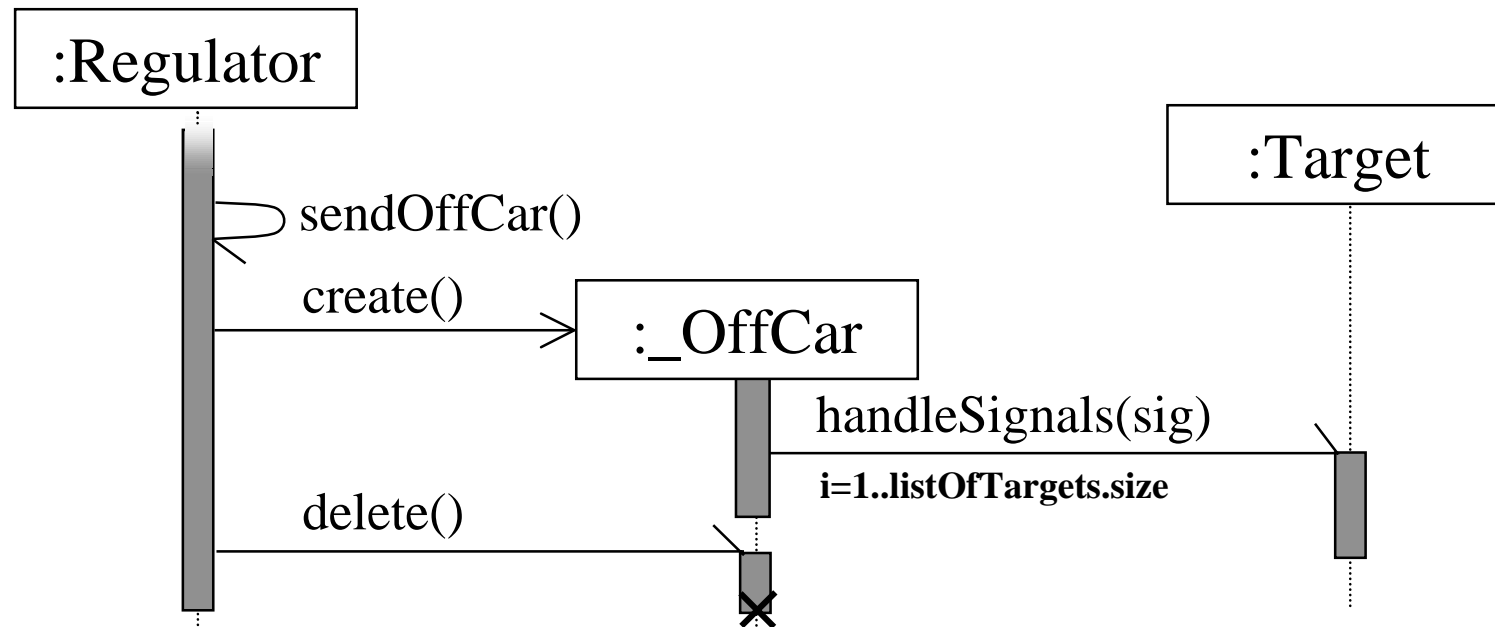
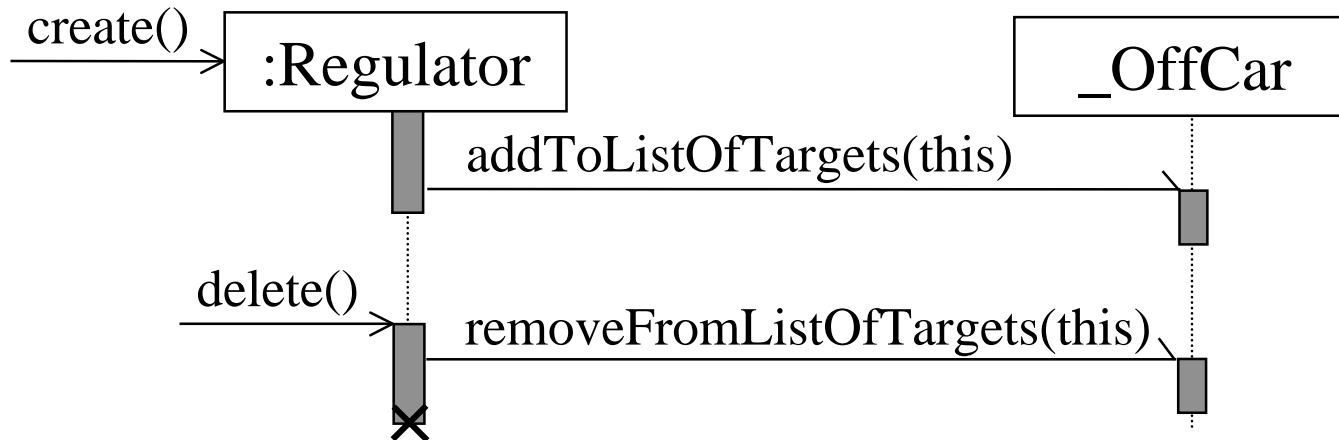
Plan de la présentation

- ✓ **Contexte et besoins**
- ✓ **Présentation de la notion de profil UML**
- ✓ **Le profil ACCORD/UML**
 - Aspects liés à la modélisation
 - Transformation automatique de modèle
 - Génération automatique de code

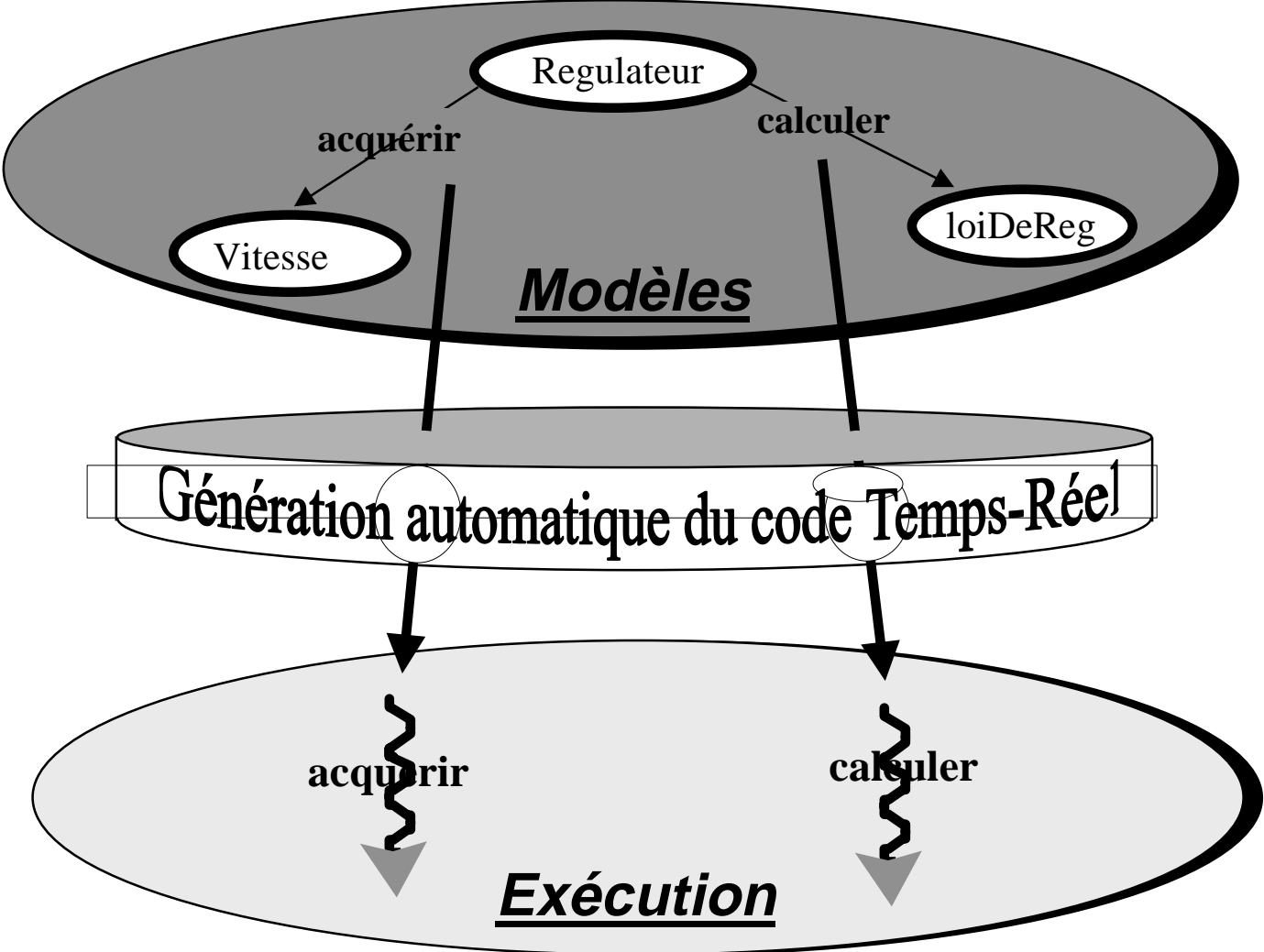
Transformation de modèle, pattern des signaux



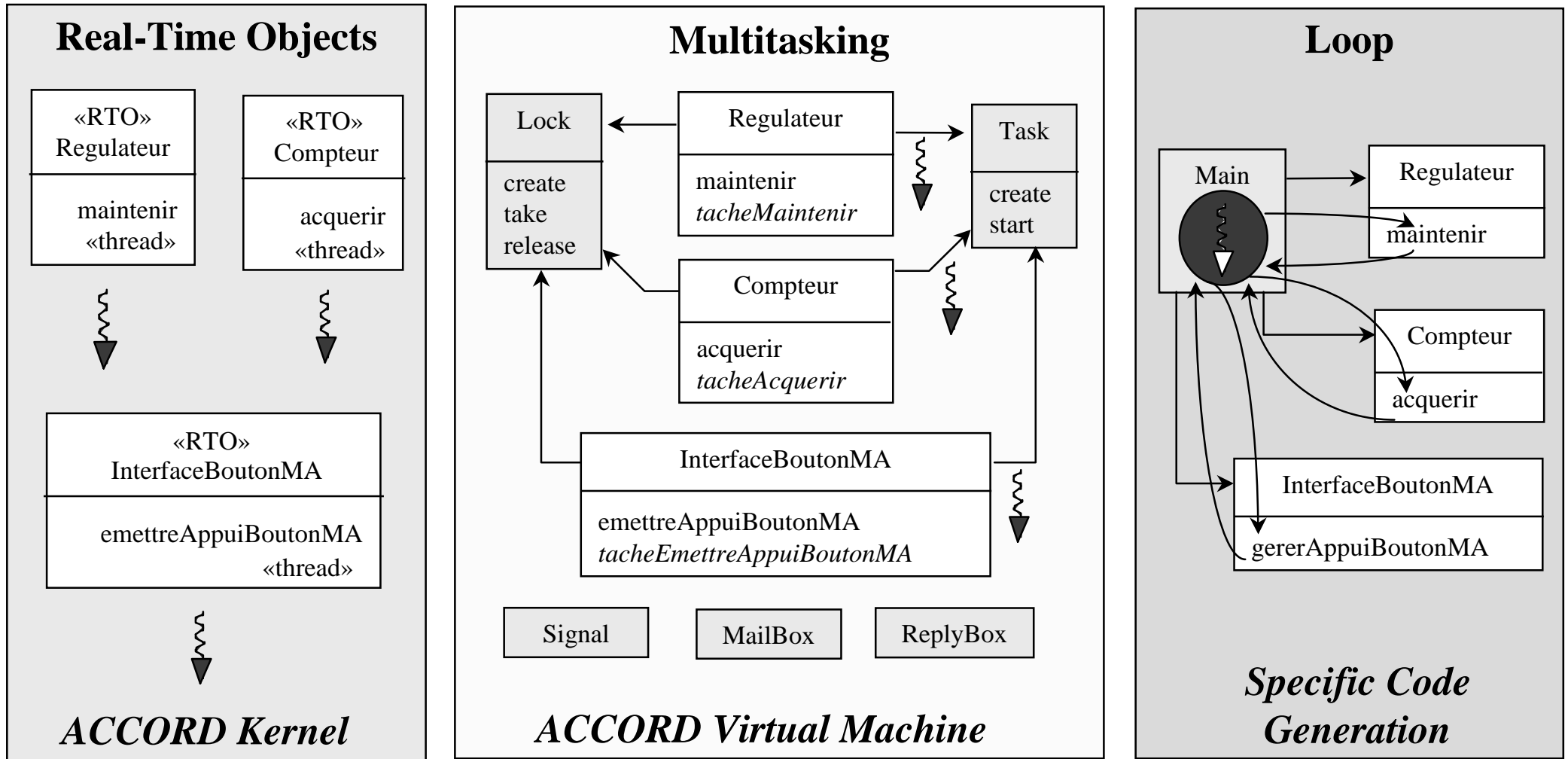
Dynamique du pattern des signaux



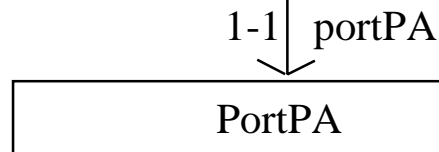
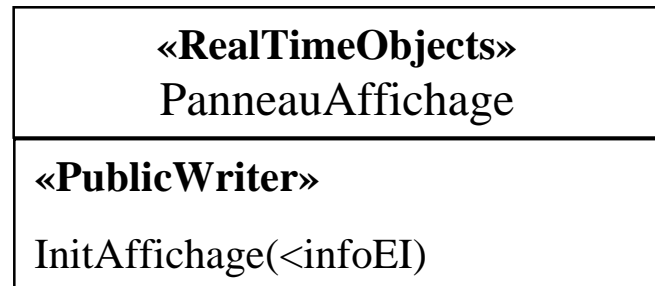
Passage du modèle à sa mise en œuvre multitâche



Un même modèle pour plusieurs implantations TR



Description du modèle utilisateur



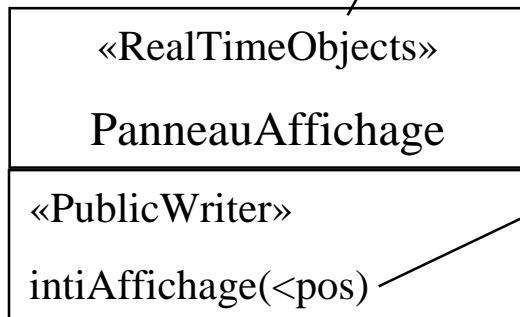
OnOff / InitAffichage(dl=50ms)



Class PanneauAffichage

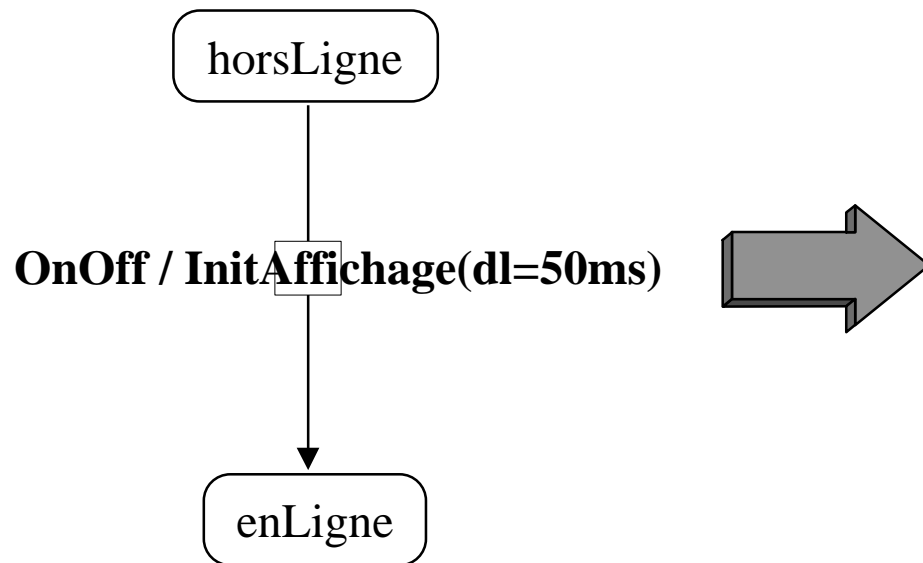
```
{ public :  
    PortPA *portPA;  
    void InitAffichage(EI infoEI)  
    { ##### code user for InitAffichage ###  
      majInfoInitVitesse(infoEI.vit) ;  
      majInfoInitSyst(infoEI.syst);  
      ##### end #####  
    }  
};
```

Génération automatique du code temps-réel 1/2



```
Class PanneauAffichage : public RTO
{ public :
void initAffichage(RTC c, EI infoEI)
    { Request r (iniTask_initAffichage, this, c);
      r << pos;
      allocateTask( r); }
void iniTask_initAffichage (Request *pr)
    { PanneauAffichage *this = pr->This;
      control(pr);
      int arg; (*pr) >> arg;
      codeTask_InitAffichage(arg);}
void codeTask_initAffichage(EI infoEI)
    { majInfoInitTrain(infoEI.train) ;
      majInfoInitSyst(infoEI.syst);}
    ...
}
```

Génération automatique du code temps-réel 2/2



```
⋮  
void handleSignal (OnOff &ev)  
{ if (state == horsLigne)  
  { RTC c (50);  
    InitAffichage( c, ev.infoEI);}  
  }  
};
```

Intérêt de cette démarche par profil UML

✓ Guider

Contrôler

Automatiser



le développement des modèles

✓ Amélioration de la maîtrise de développement d'un produit

✓ Gain en qualité, sûreté et en fiabilité des produits issus d'un tel développement

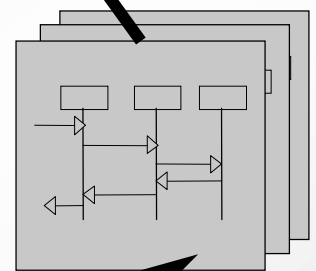
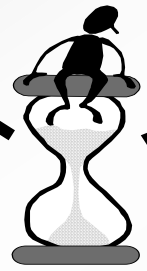
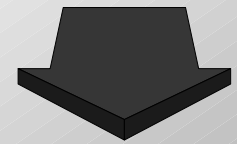
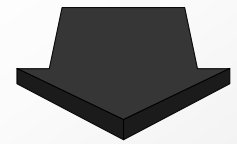
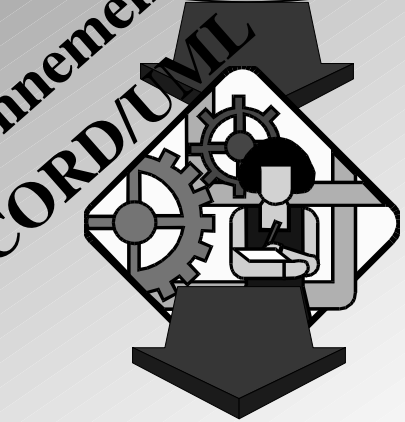
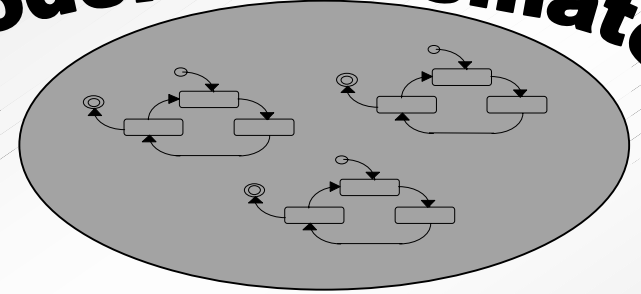
Travaux en cours...

- ✓ **Rédaction d'un profil UML pour le développement de systèmes temps-réel embarqués dans le domaine automobile**
 - Thèse
 - Projet européen WOODDES
 - CEA, PSA, IntraCom, MECEL, I-Logix, Verilog, UPSALA et OFFIS
- ✓ **Travaux sur la validation des modèles issus de ce profil**
 - Génération automatique de tests

Modèle ACCORD/UML

Modèle Statemate

Environnement
ACCORD/UML



J1 = (Vit=120, dist=150)
J2 = (Reg=true, Vit=55)
...

$\neg (\text{Vit} < 100) \wedge (\text{dist} < 200)$
 $(\text{Reg} = \text{true}) \wedge (\text{Vit} \geq 50)$
...



Tests