



Objets, Composants et Aspects

pour la modélisation de systèmes temps réel embarqués

Bachir Djafri, Jean-Marc Delosme
LaMI - UMR CNRS 8042
Université d'Évry Val d'Essonne
Génopole
{nom}@lami.univ-evry.fr

Plan

- ◆ Problématique et Motivations
- ◆ Modèles de traitement (MoC)
- ◆ Approche objet, composant et aspect
- ◆ Modèle, métamodèle et framework
- ◆ Conclusion

Problématique (1)

- ◆ Complexité croissante des systèmes
 - Nombre de composants
 - Taille du code embarqué
 - Puissance de calcul
- ◆ Contraintes non fonctionnelles (temps, surface, consommation, ...)
- ◆ Hétérogénéité croissante des systèmes
 - Numérique, analogique, mécanique,...
 - Contrôle, traitement de signal, ...
 - Différents **MoC**

Problématique (2)

- ◆ Besoin d'une nouvelle classification (MoCs)
- ◆ Expression de tous les **MoCs**, **contraintes** + structuration, abstraction et réutilisation
- ◆ Besoin d'une approche de modélisation

Modèles de traitement

- ◆ Domaine d'application
- ◆ Propriétés partagées par des systèmes ayant des comportements similaires (même domaine)
- ◆ Exemples :
 - Événement discret ;
 - Synchronisme réactif ;
 - Flot de données ;
 - ...

```
init
foreach Event do
    calculer-Sortie();
    mise-à-jour-état();
od
```

Approches de modélisation

◆ Approches hétérogènes

- Combinaison de plusieurs langages
- Nécessité d'une gestion d'interactions entre modèles
- Modèle de coordination

◆ Approches homogènes

- Un seul langage
- Expression des MoCs à l'aide d'un seul langage
- Exemples : Rosetta (nouveau langage), SystemC et PtolemyII (extension de langage).

Approche « idéale »

- ◆ Abstraction
- ◆ Modularité
- ◆ Réutilisabilité
- ◆ Support de plusieurs MoCs
- ◆ Expression de différents types de contraintes
- ◆ Définition de modèles exécutables

- ◆ ... Combinaison de plusieurs paradigmes

Vers une approche multiparadigmes

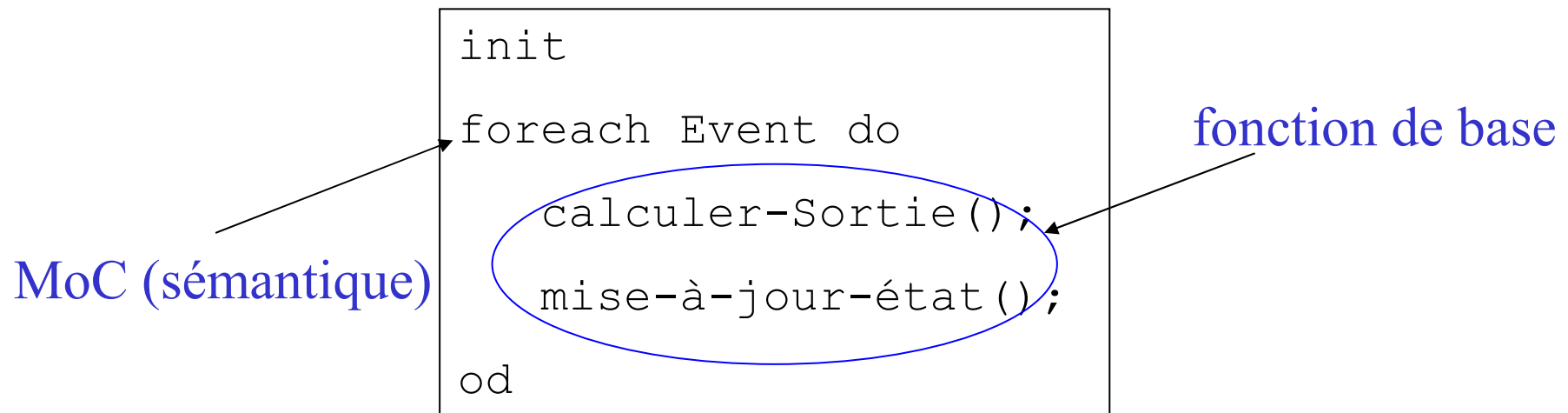
- ◆ Abstraction, modularité, réutilisabilité \Rightarrow Approche Objet
- ◆ Objet + Autonomie \Rightarrow Approche Composant
- ◆ Expression de propriétés non fonctionnelles (MoCs + contraintes) \Rightarrow Approche Aspect
- ◆ Expression des concepts (composant, aspect) \Rightarrow approche Objet

Structuration des modèles

- ◆ Système = ensemble de composants (hiérarchie)
- ◆ Composant = ensemble de sous-composants et/ou ensemble d'Aspects
- ◆ Trois niveaux de modélisation
 - Base (fonctionnel)
 - Contrôle (non fonctionnel)
 - Architecture (liens et intégration, tissage)
 - Interconnections des sous-composants (Base)
 - Tissage local fonction-MoC-Contraintes

Structuration des modèles

- ◆ Séparation des propriétés fonctionnelles (traitement de base) et des propriétés non fonctionnelles (sémantique d'exécution)



Description des modèles

- ◆ Absence de langage et de notation
- ◆ Définition d'un **métamodèle** pour chaque niveau de modélisation
- ◆ Description des concepts de chaque métamodèle à l'aide d'**objets**
- ◆ Définition d'un **framework** extensible (classes)

Métamodélisation

- ◆ Définition d'un métamodèle : ensemble de concepts (outils de modélisation) + liens entre les concepts
- ◆ Abstrait et général
- ◆ Mise en œuvre : langage de modélisation et/ou notation
- ◆ Définition d'un framework (classes)
- ◆ Génération de code exécutable (embarqué)

Conclusion

- ◆ Approche de conception (méthodologie) unifiée
- ◆ Approche multiparadigmes
 - Composant \Rightarrow autonomie, réutilisabilité,...
 - Aspect \Rightarrow flexibilité, séparation fonctionnalité/domaine,...
 - Composition d'aspects
- ◆ Intégration de nouveaux MoCs
- ◆ Transformation de métamodèles