

## Une première passerelle UML vers Signal

### 1. Modélisation UML du Régulateur Ballon

### 2. Principes de la passerelle « Génération Signal »

- a. Principe général de la traduction
- b. Génération du programme principal
- c. Génération d'un process « chart »
- d. Génération d'un process

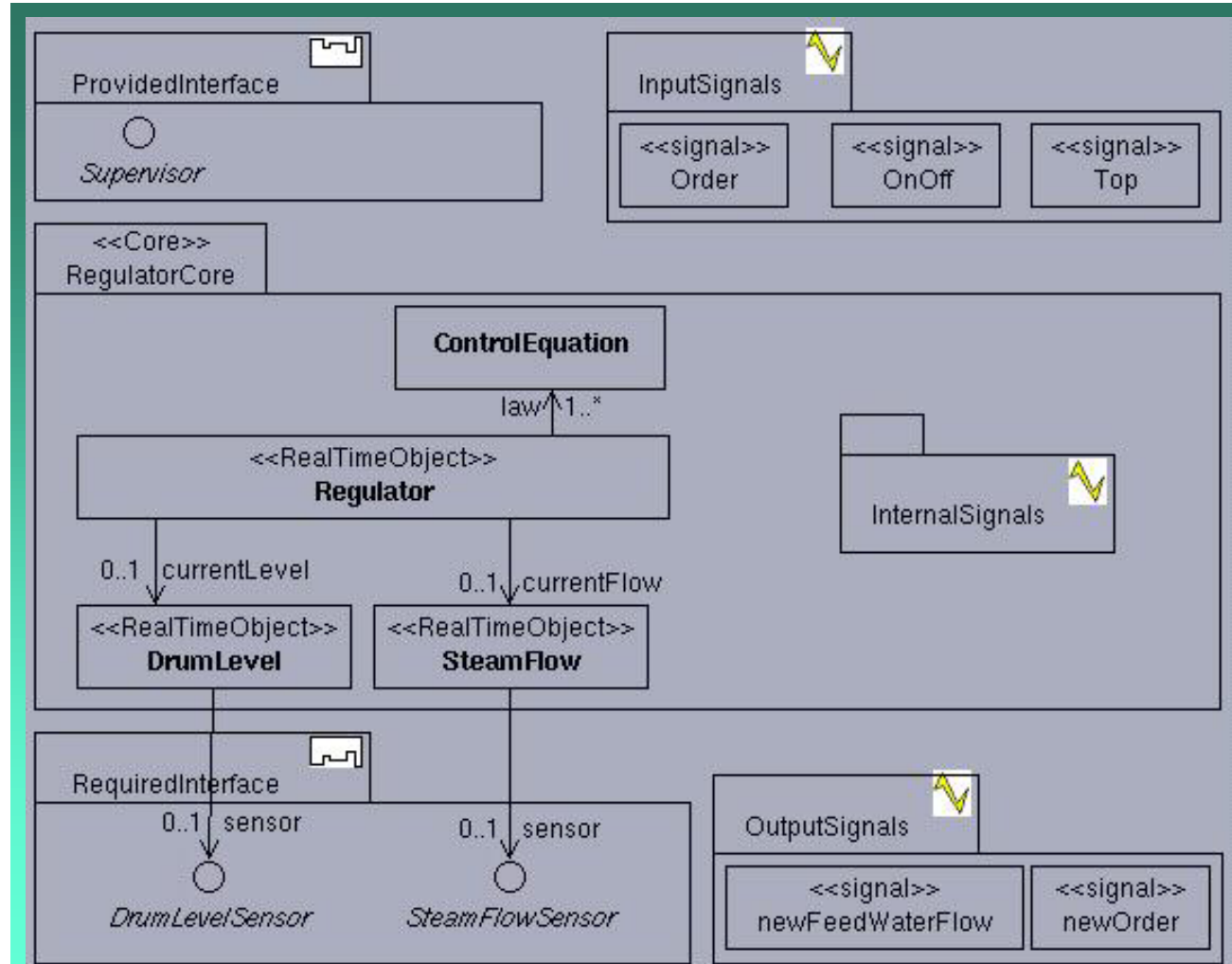
### 3. Remarques sur la passerelle

## Modélisation UML du Régulateur Ballon

## Rappel du cahier des charges

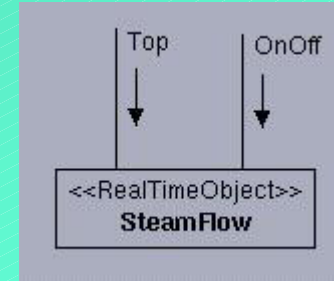
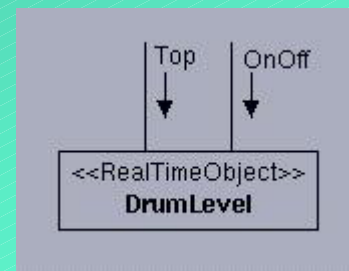
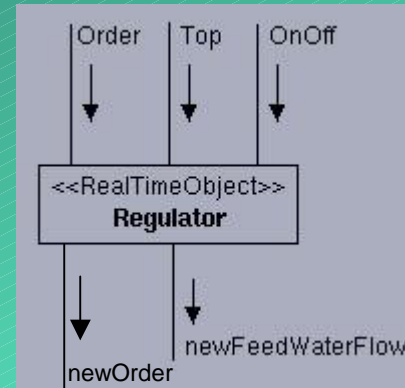
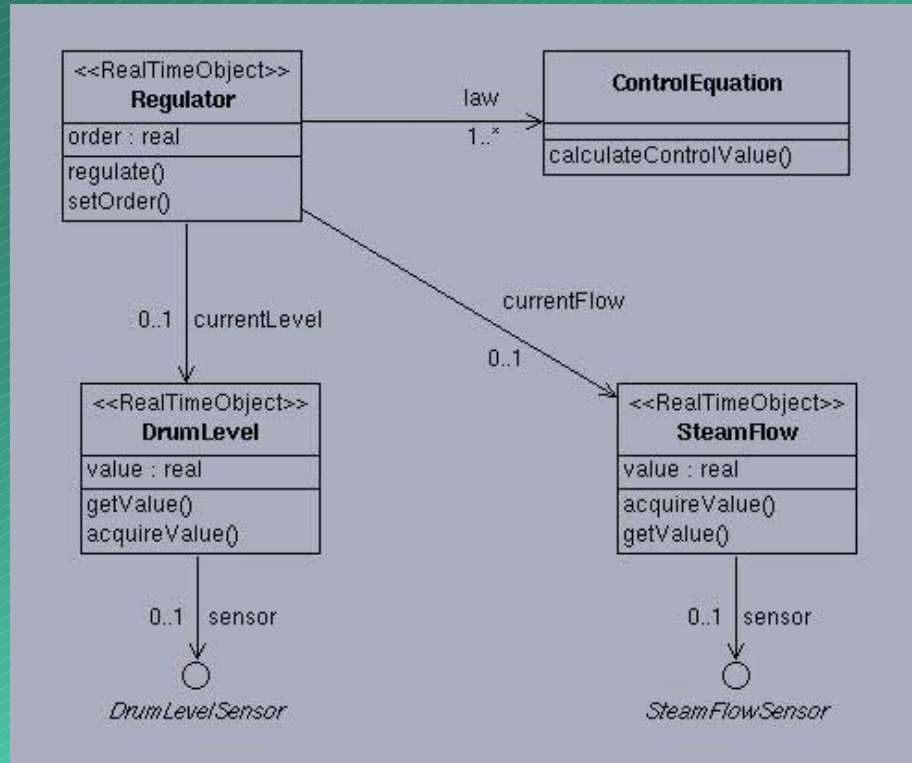
- Acquisition du niveau d'eau (auprès d'un capteur de niveau)
- Acquisition du débit de vapeur (auprès d'un capteur de débit)
- Enregistrement de la consigne de niveau (envoyée par le superviseur)
- Calcul périodique du débit d'eau (à envoyer à la vanne)

## Diagramme de classes global du modèle d'analyse détaillée (DAM)

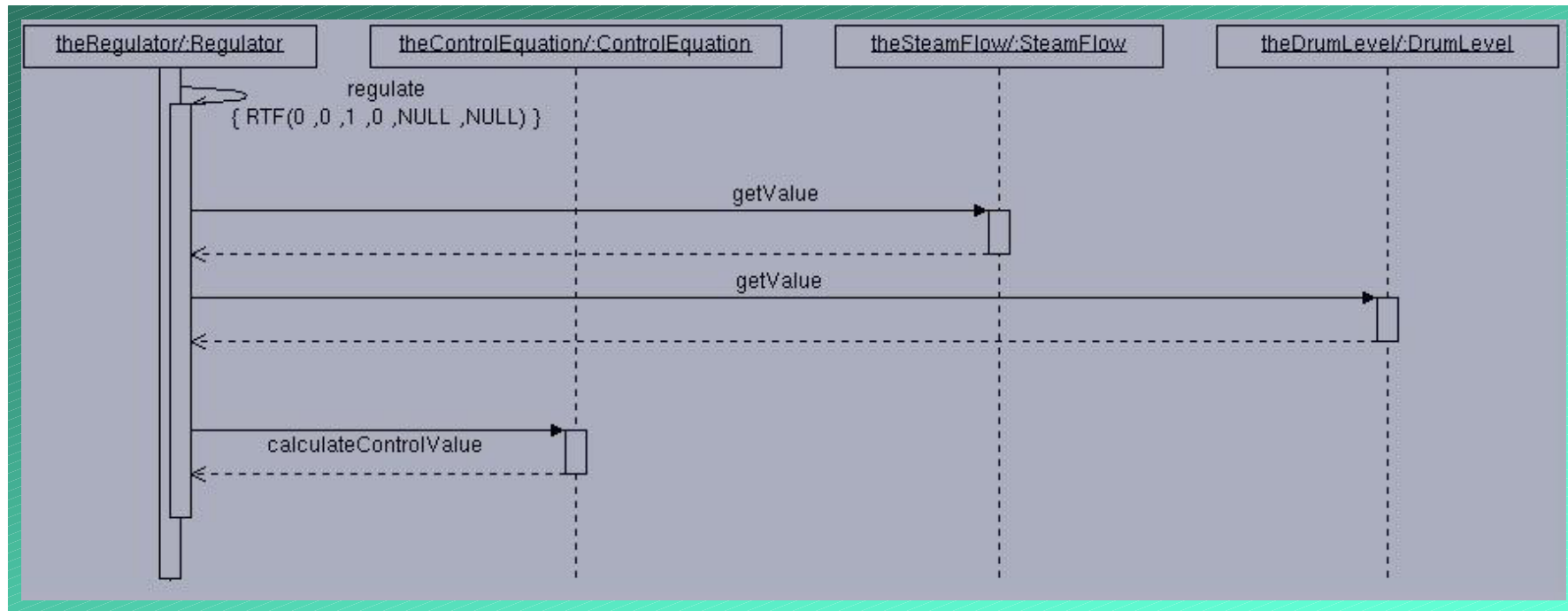


## Diagrammes de classes détaillés

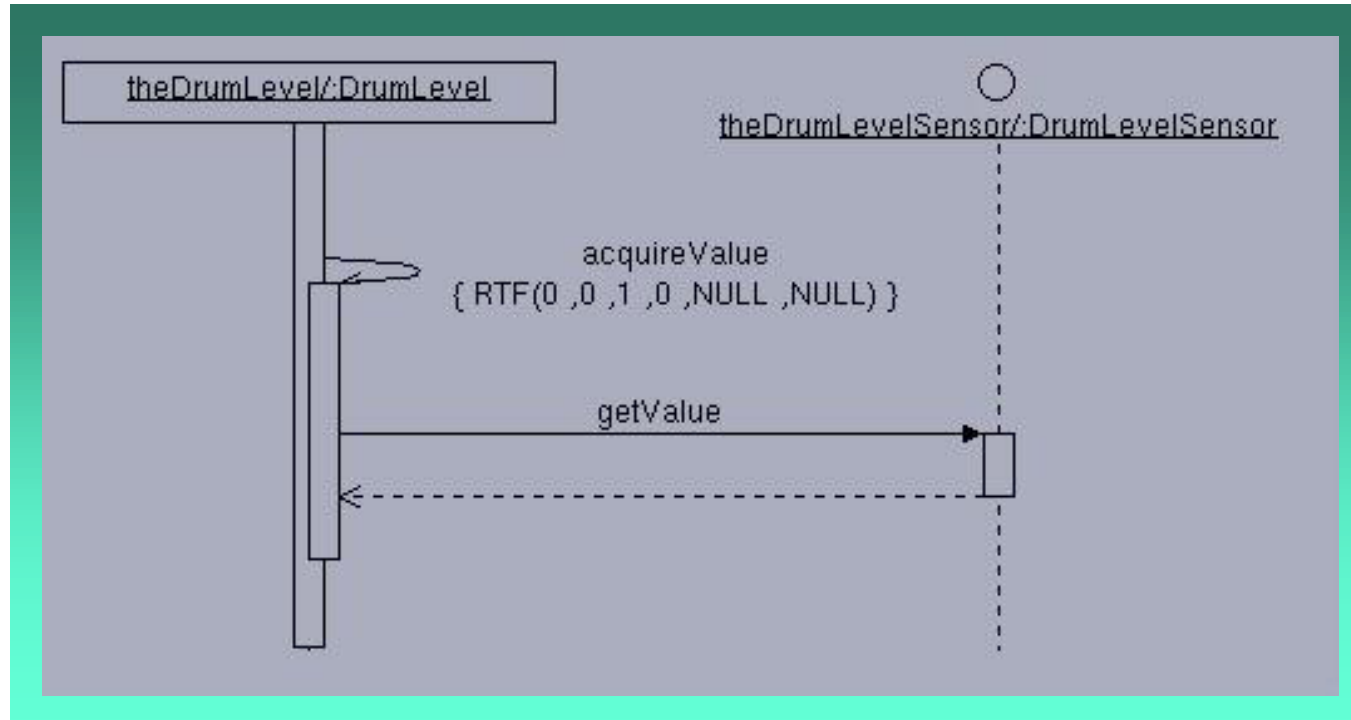
### Paquetage "Core"



## Diagramme de séquence du scénario “Regulate”



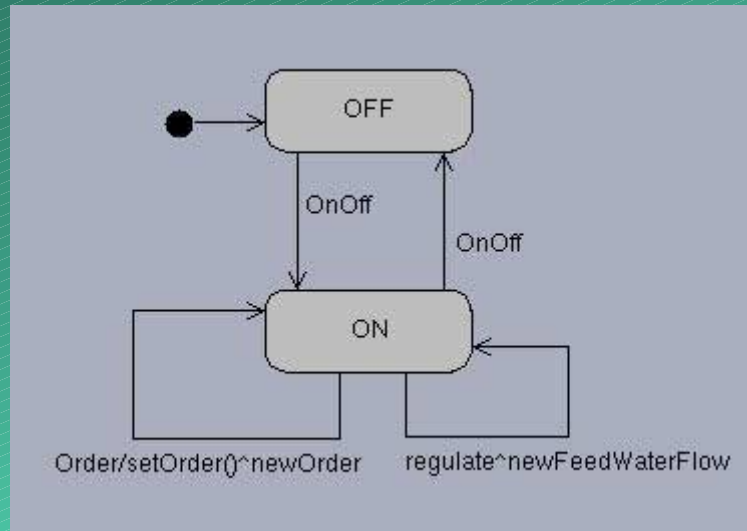
## Diagramme de séquence du scénario “Acquire Drum Level”



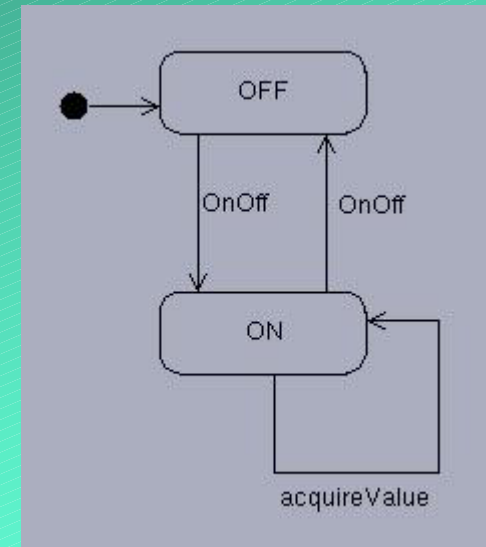


## Diagrammes d'états-transitions

### Classe Regulator



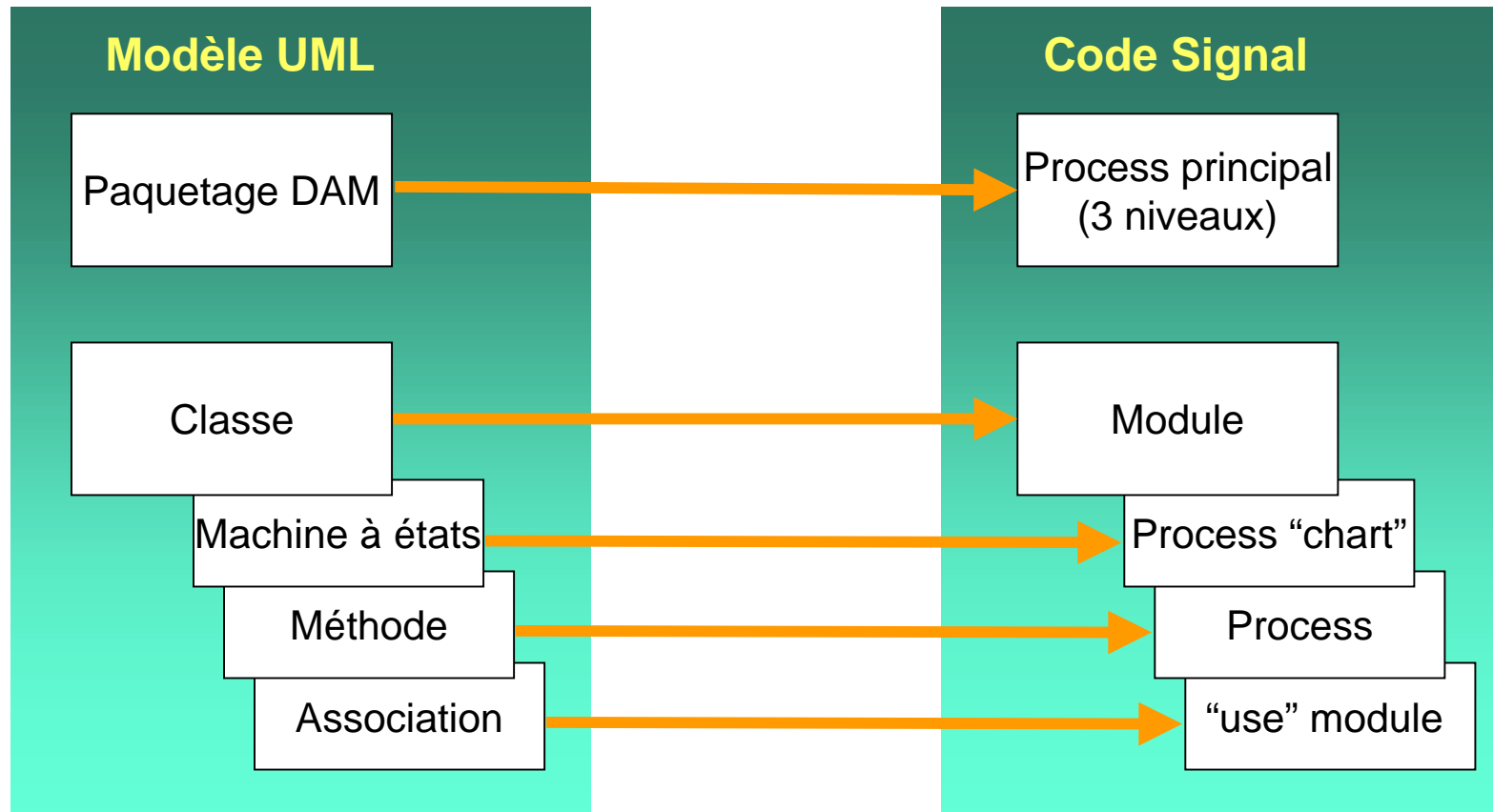
### Classe DrumLevel Classe SteamFlow



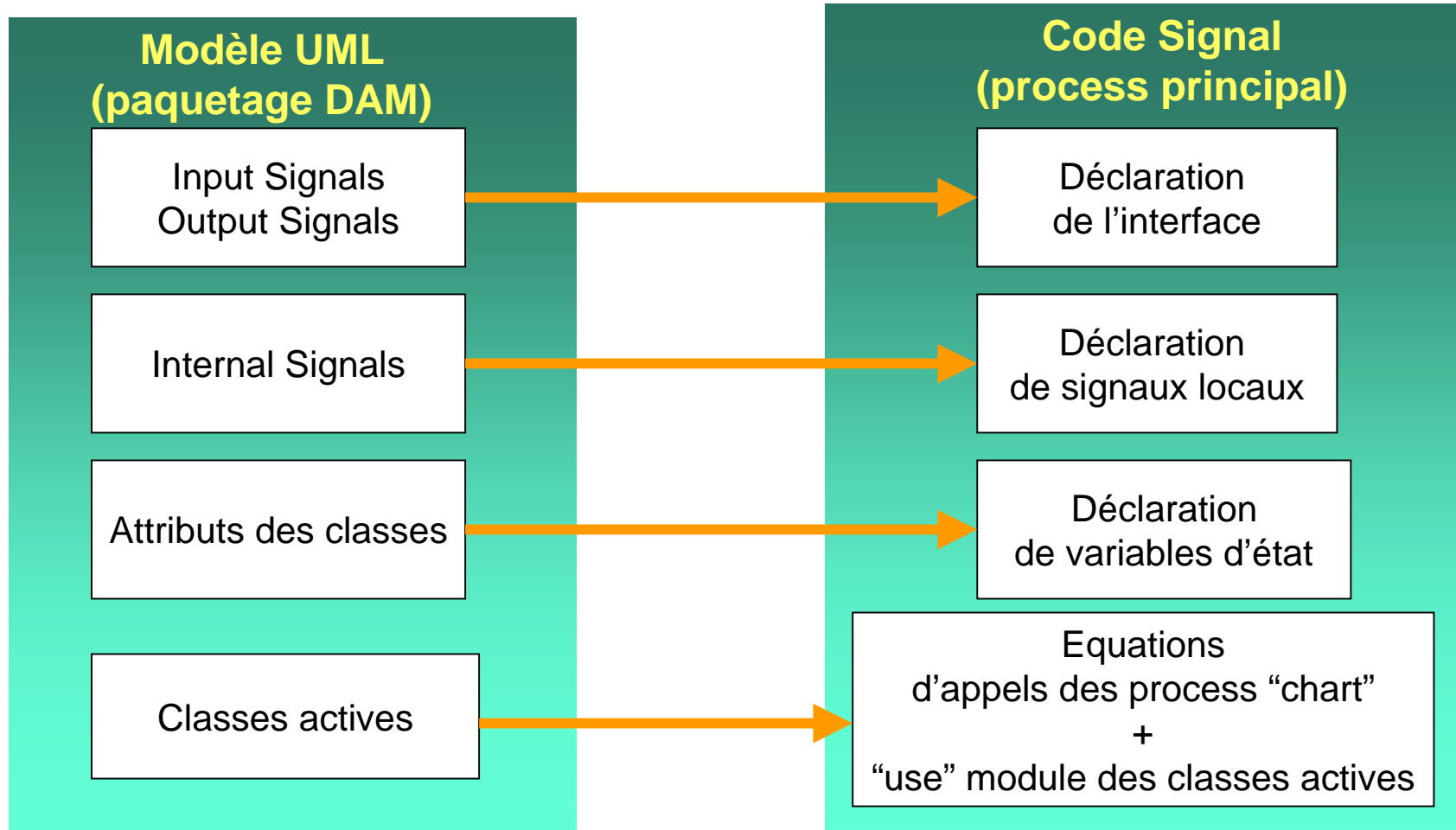


**Une passerelle UML vers Signal**

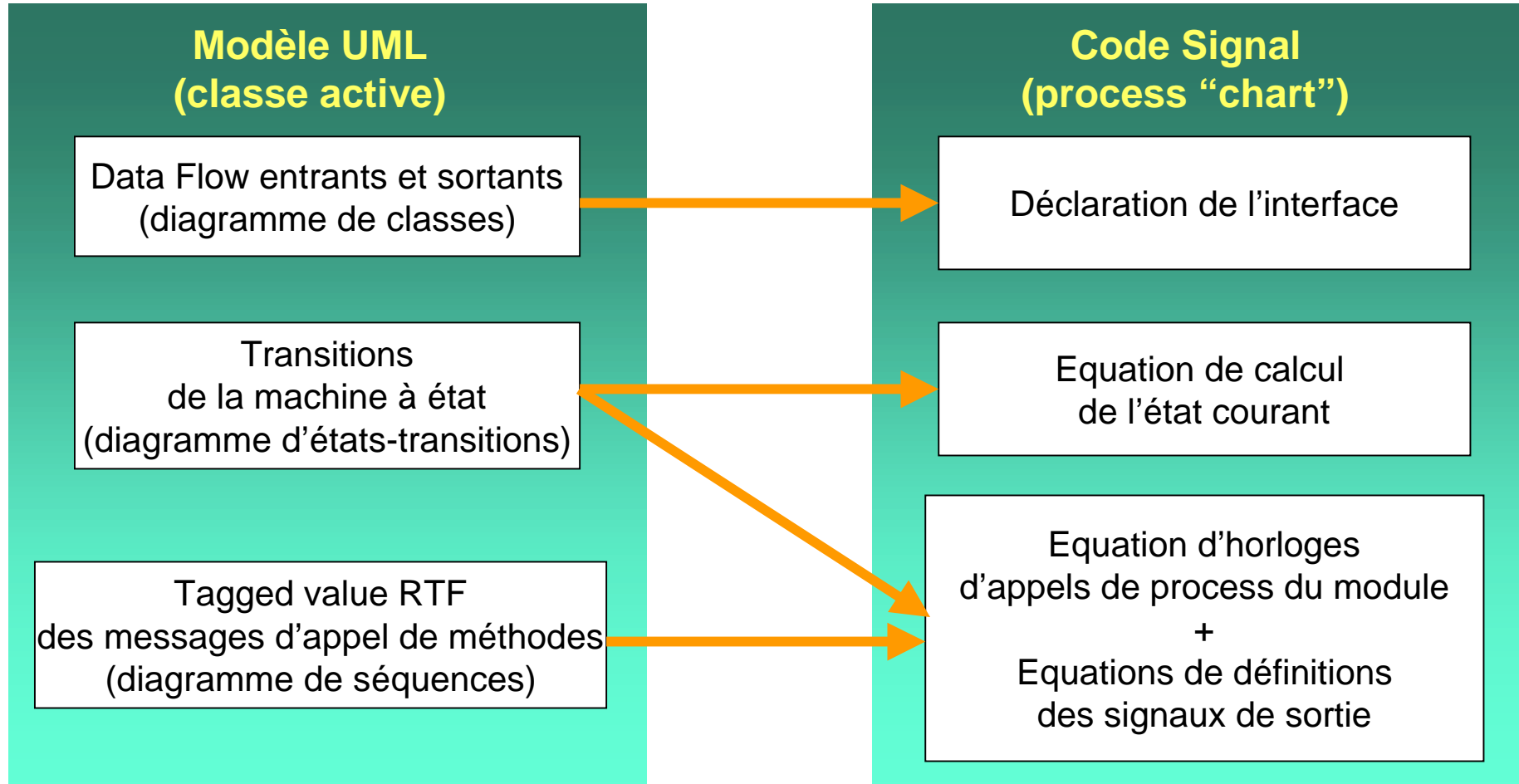
## Principe général de la traduction



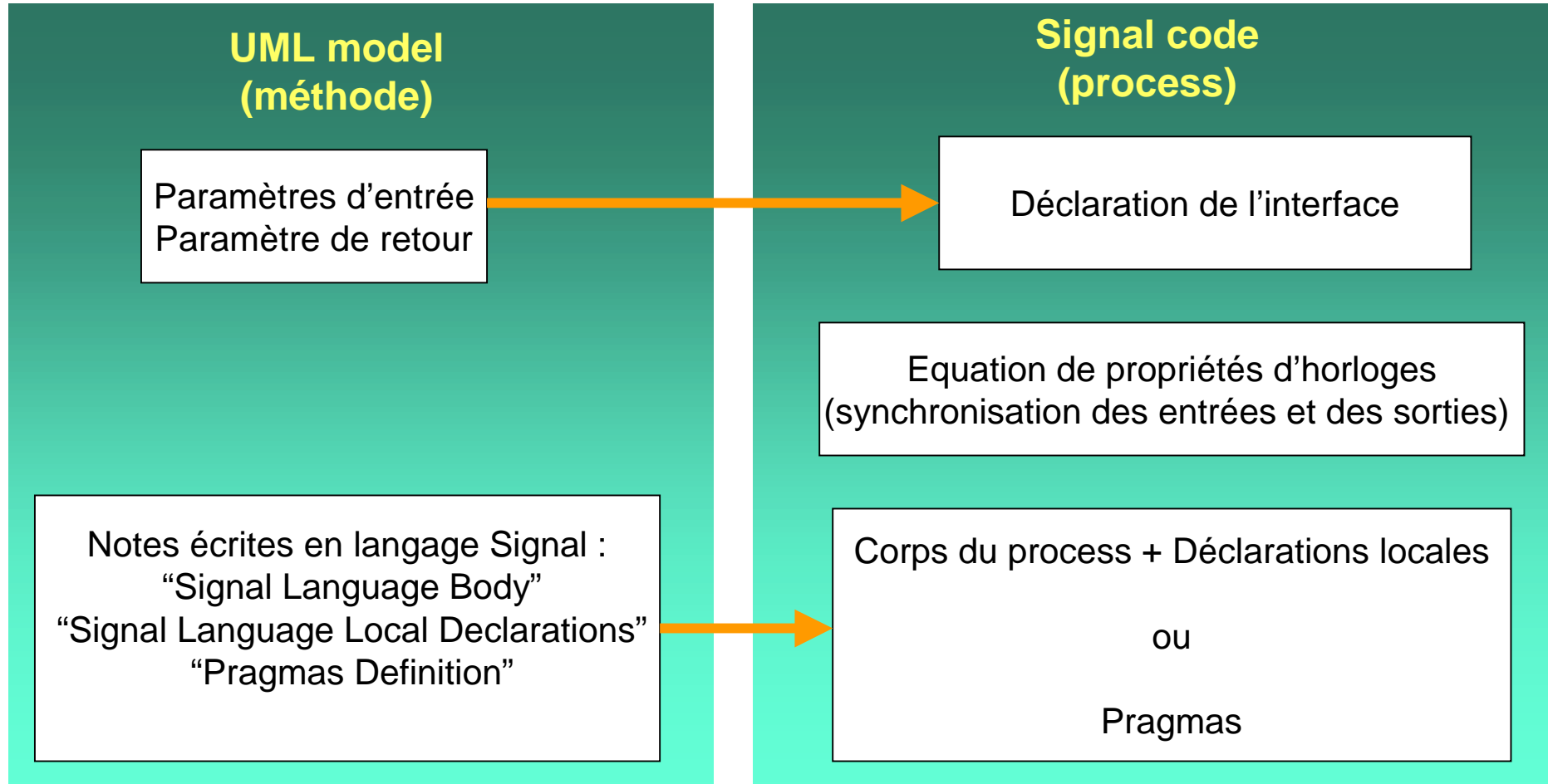
## Génération du process principal



## Génération d'un process « Chart » associé à une classe active



## Génération d'un process associé à une méthode



**Remarques sur la passerelle**

## Remarques sur la passerelle

- Introduction du temps dans le modèle UML avec le signal « Top »
- « Protocol view » de la machine à états exploitée partiellement
- Synchronisation systématique des entrées/sorties d'un process associé à une méthode de classe
- Cas mal traité : un même signal émis dans des transitions différentes
- Tentative de traitement des états composites
- Option de génération avec renvoi des états des classes actives
- Problème de la simultanéité des signaux d'entrée (priorités des transitions)