

# Modèle de temps de MARTE & CCSL

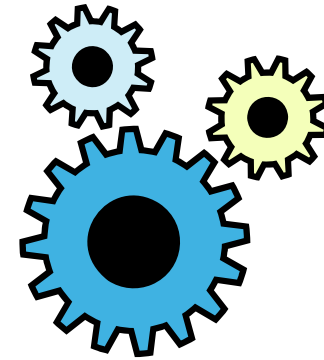
Charles André, Frédéric Mallet

Équipe projet AOSTE / Laboratoire I3S  
Université de NICE / INRIA Méditerranée

# Motivation



DSML



**Temps**

**Modélisation**

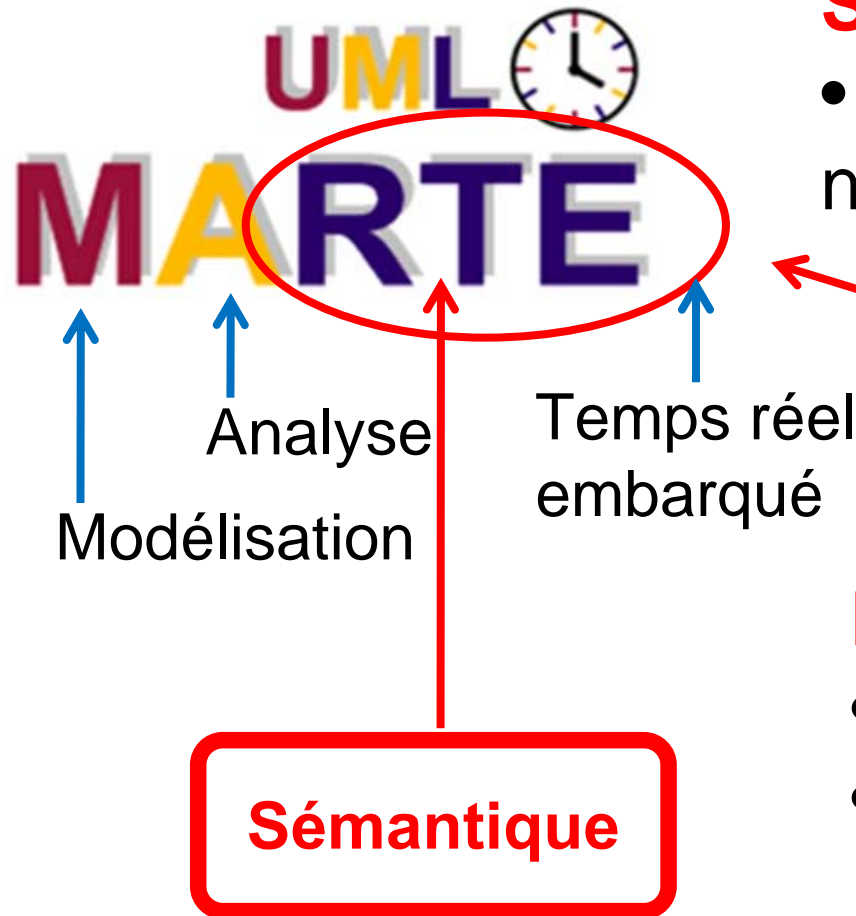
**Analyse**

Performance

Vérification

Systemes Temps réel  
embarqués

# Un profil UML dédié



## Spécification de valeurs

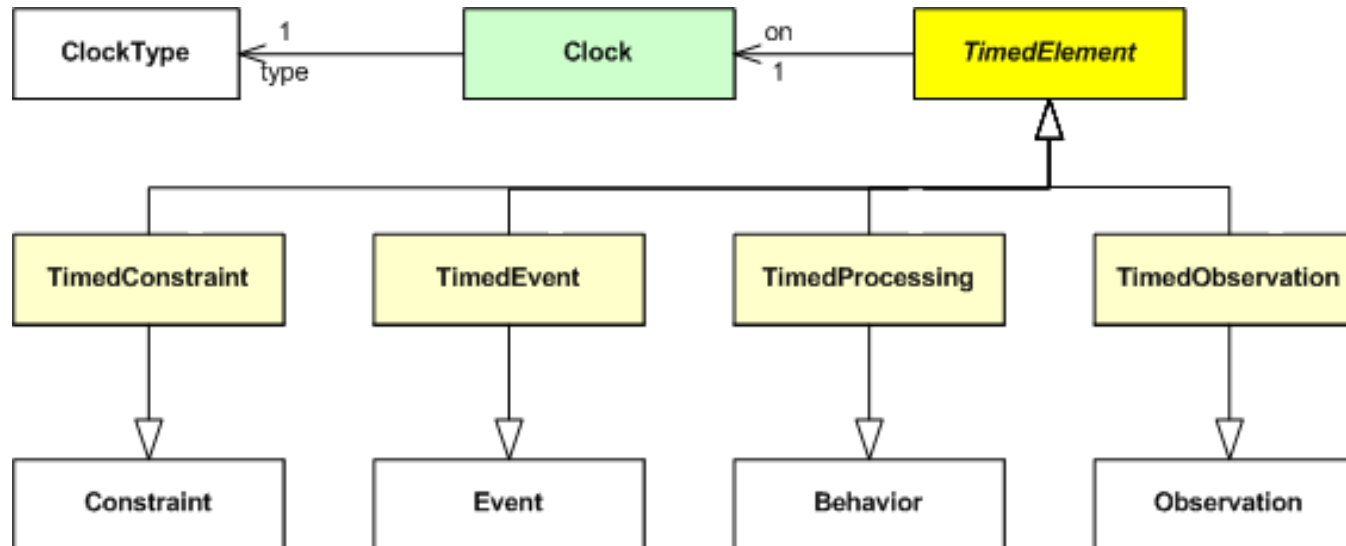
- VSL + NFP (propriétés non fonctionnelles)

Annotations

## Modélisation du temps

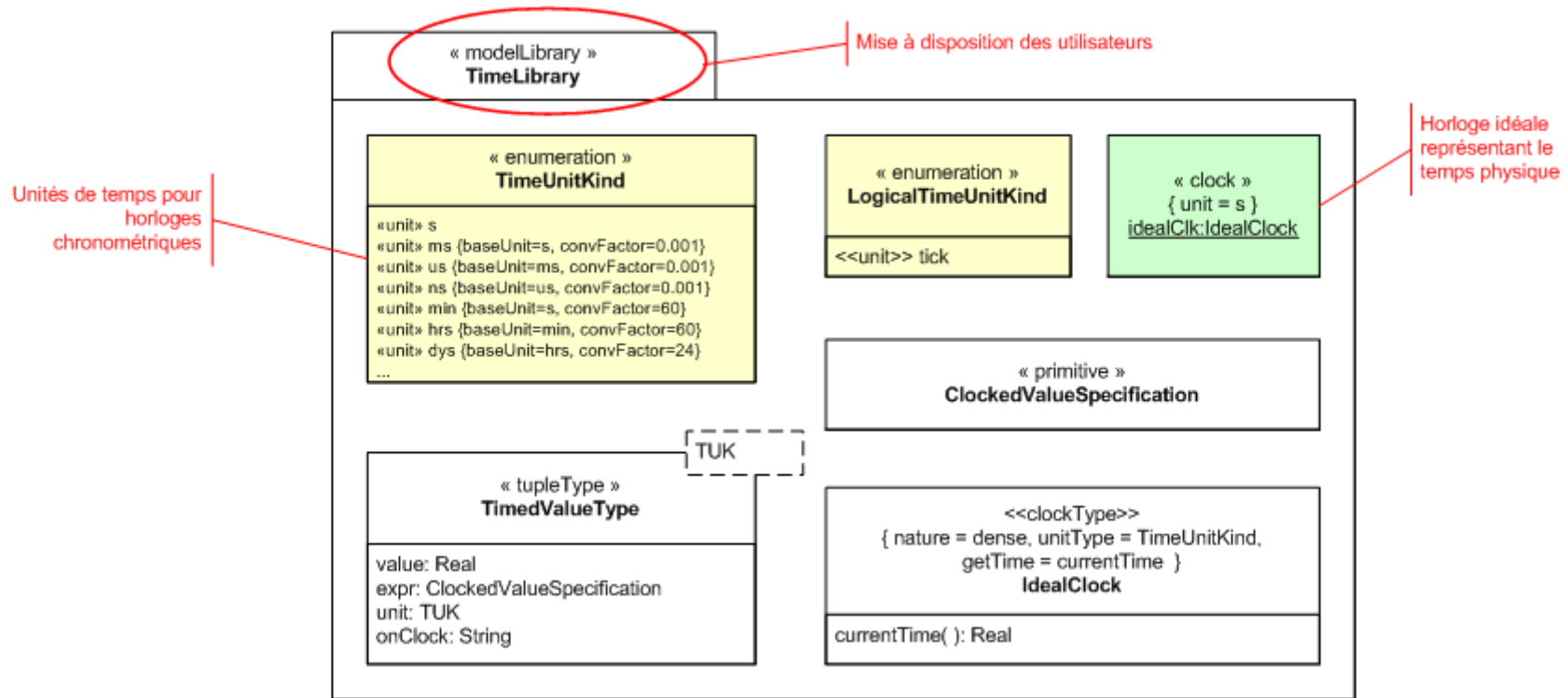
- chronométrique/logique
- discret/continu

# Temps: Metamodelè



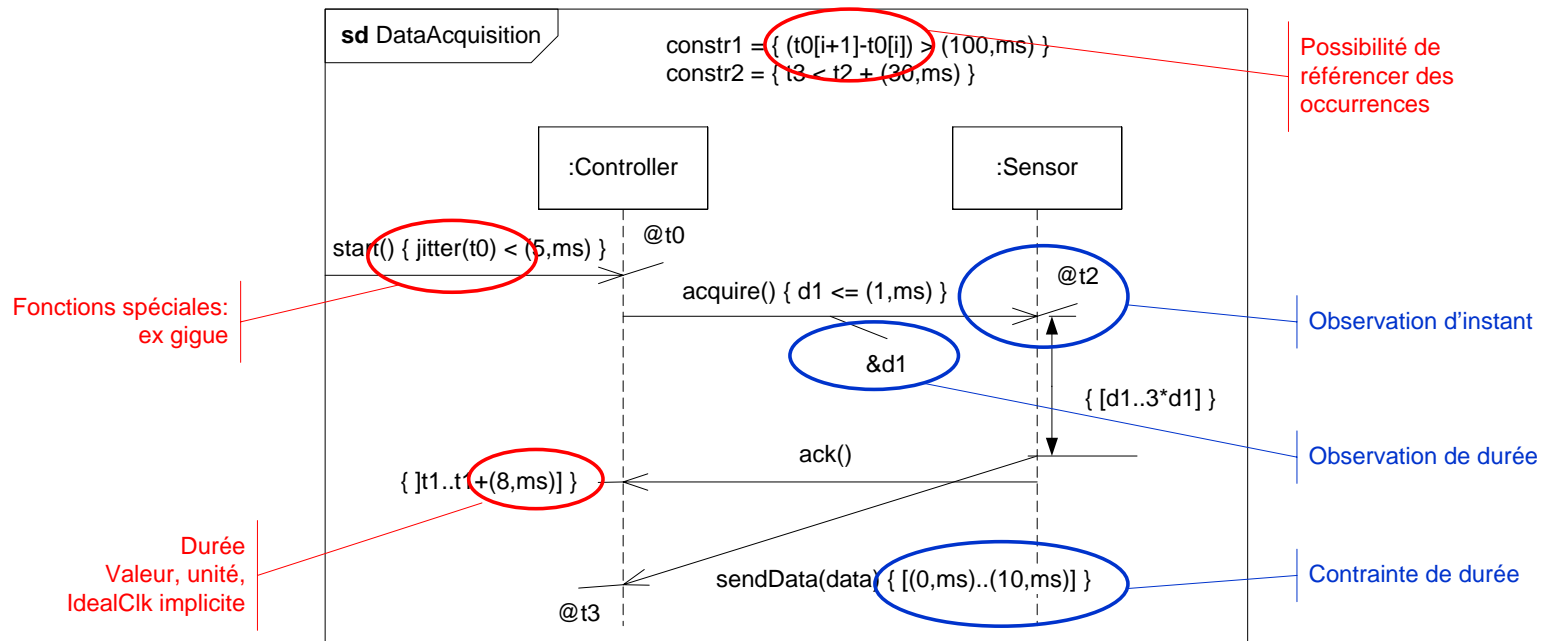
- Métamodèle simplifié du temps de MARTE. Clock, TimedElement.

# Bibliothèque Temps de MARTE



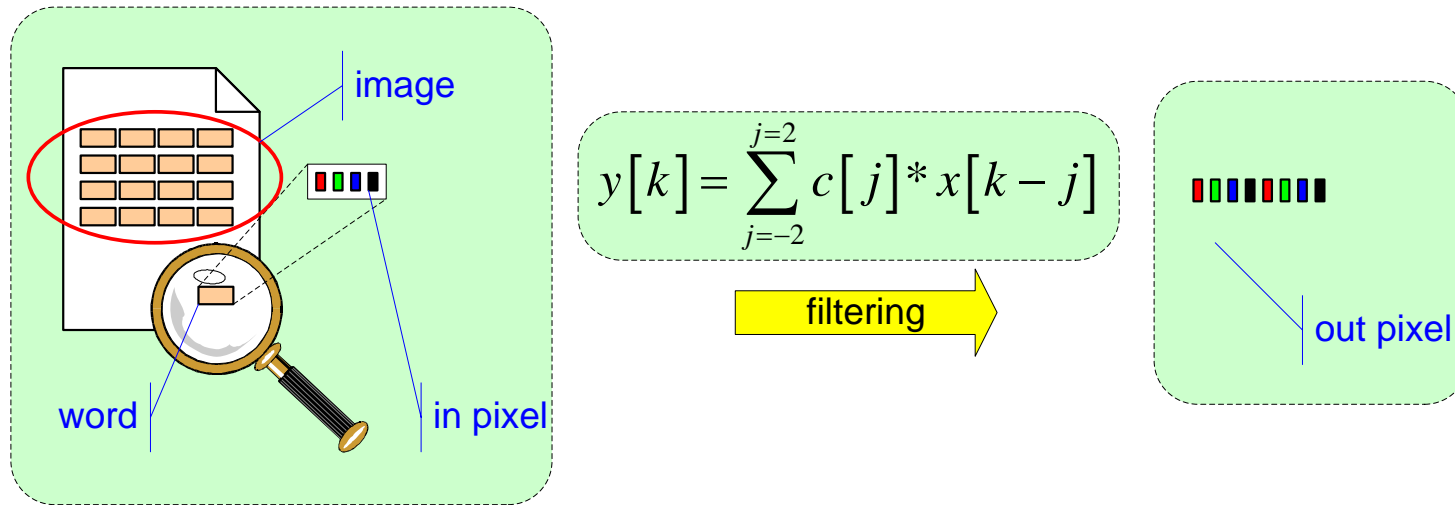
- Bibliothèque définie dans MARTE. Contient en particulier **IdealClk** pour les utilisateurs de temps chronométrique.

# Expressions temporelles VSL



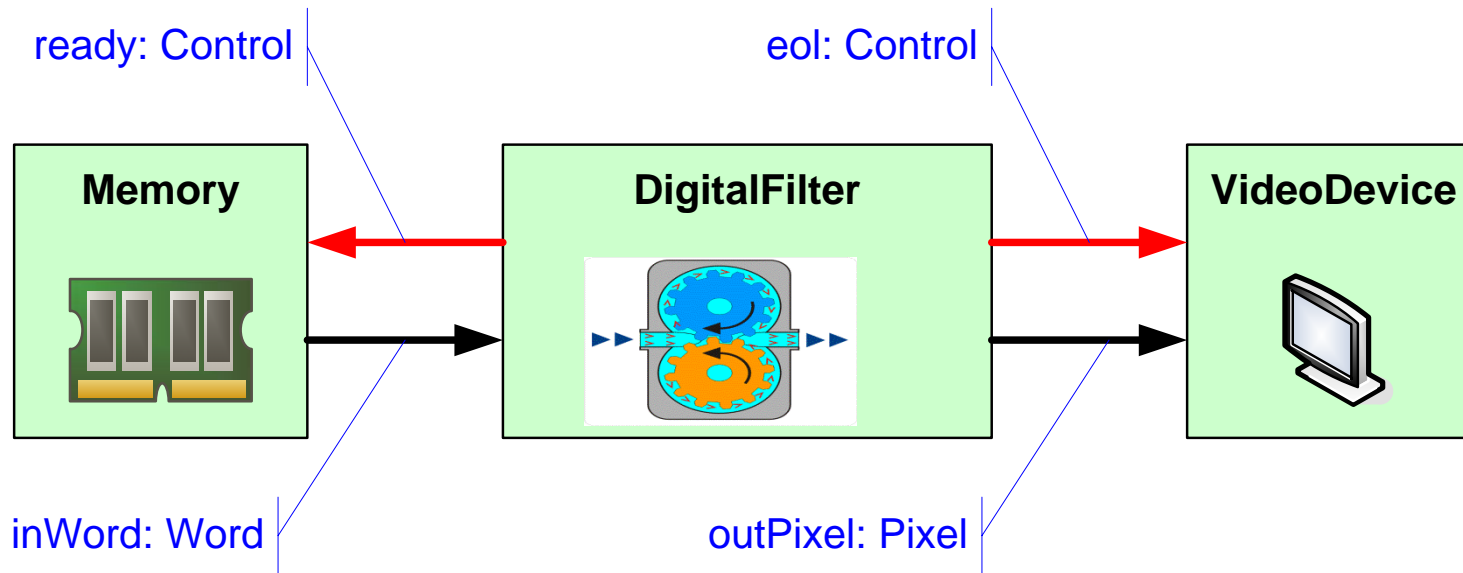
- Exemple de diagramme de séquence avec contraintes temporelles exprimées en VSL

# Utilisation d'horloges logiques



- Exemple de Filtrage numérique d'une image en mémoire vers un périphérique vidéo

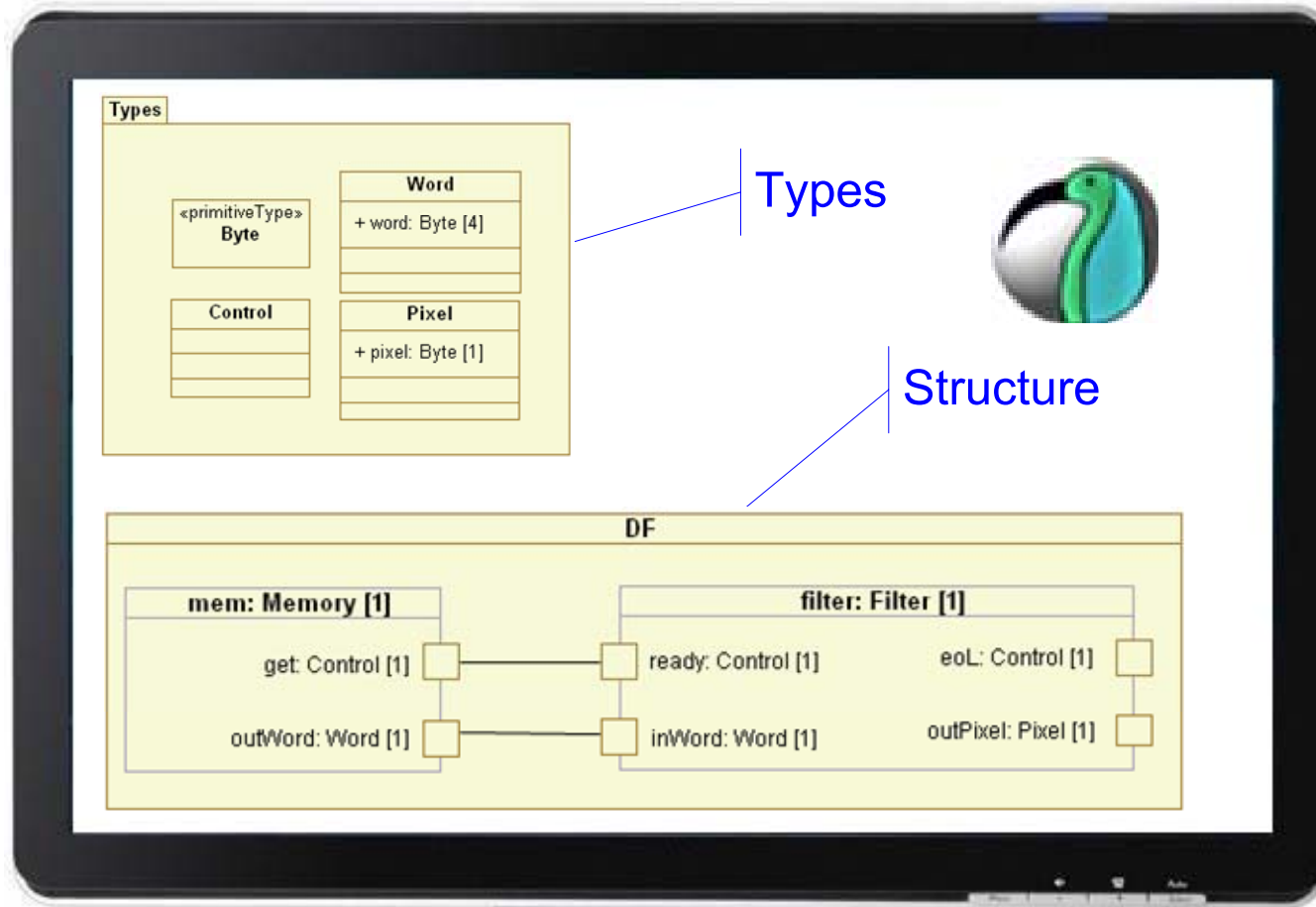
# Modélisation



- Structure  
traitement / communication



# Modélisation UML



Temps avec MARTE, CCSL - NEPTUNE'09

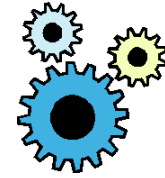


- Structure

Modélisation UML (avec papyrus)



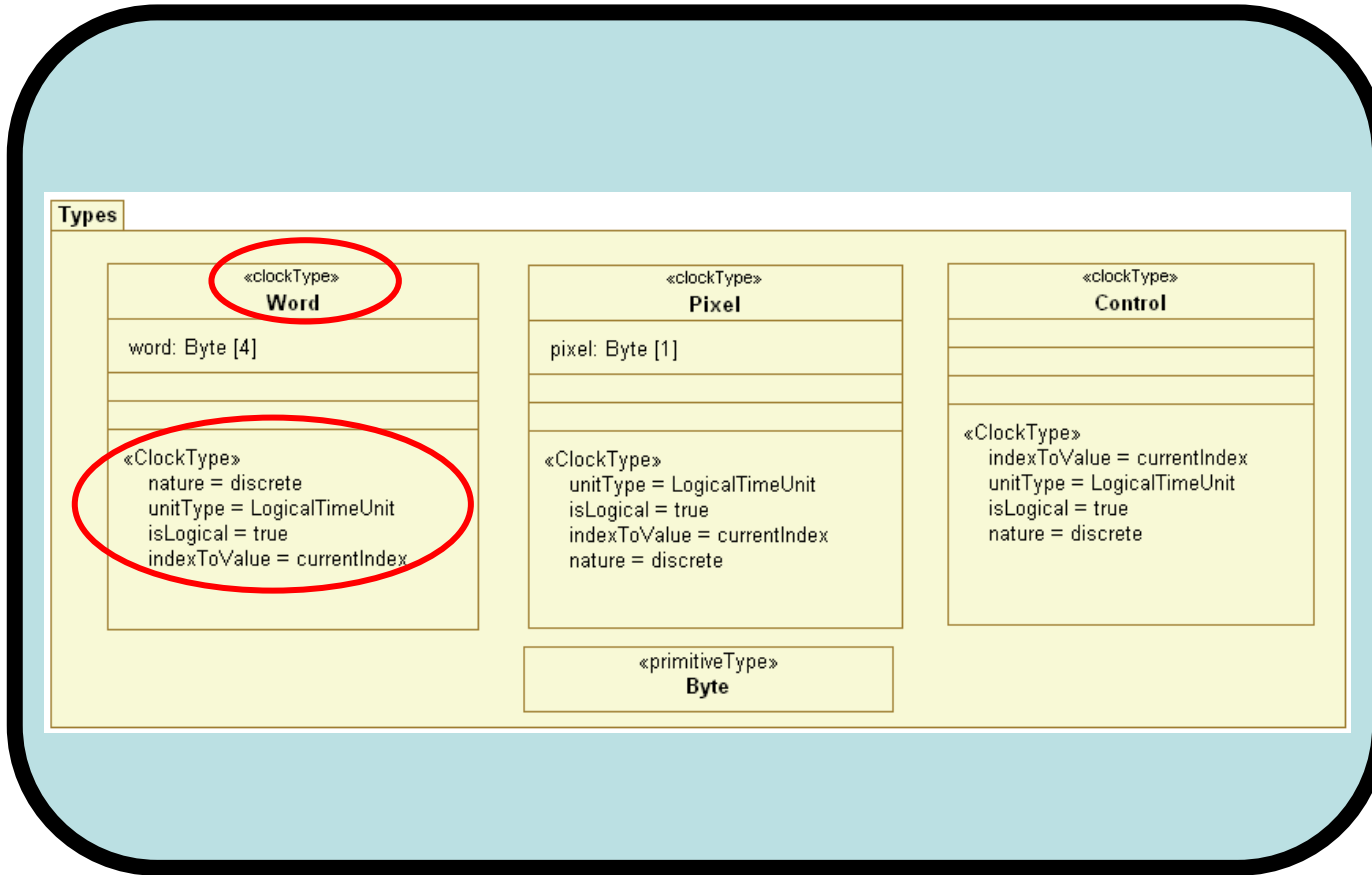
# TimeSquare



- Plug-in Eclipse pour la modélisation et l'analyse de systèmes temps réel embarqués
- Support pour **le sous-profil Temps de MARTE et le langage de spécification de contraintes d'horloges (CCSL)**
- Fonctionnalités:
  1. Application interactive des stéréotypes et éditeur de contraintes (*dans un outil UML, e.g. Papyrus*)
  2. Résolution de contraintes (**calcul d'horloge**) donnant des évolutions temporelles compatibles (*différentes stratégies*)
  3. Visualisation des traces de simulation au format **VCD** (*avec des informations interactives*)
  4. Animation de diagrammes comportementaux UML

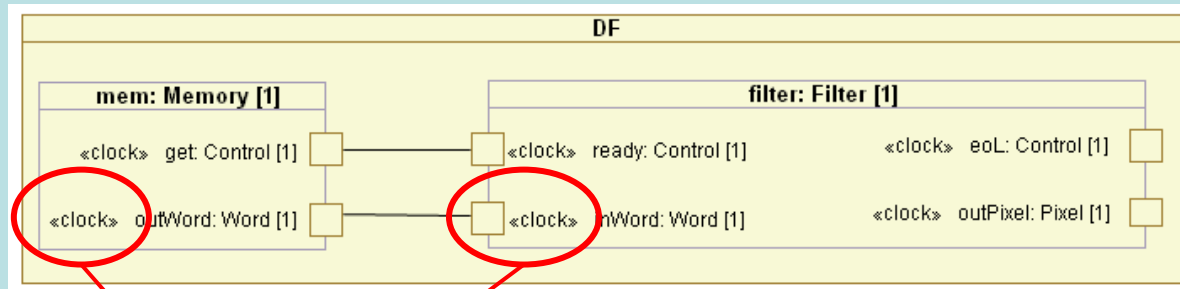
# Modélisation UML+MARTE

Temps avec MARTE, CCSL - NEPTUNE'09



- Application du profil UML MARTE:  
Classes stéréotypées par «clockType»

# Modélisation UML+MARTE



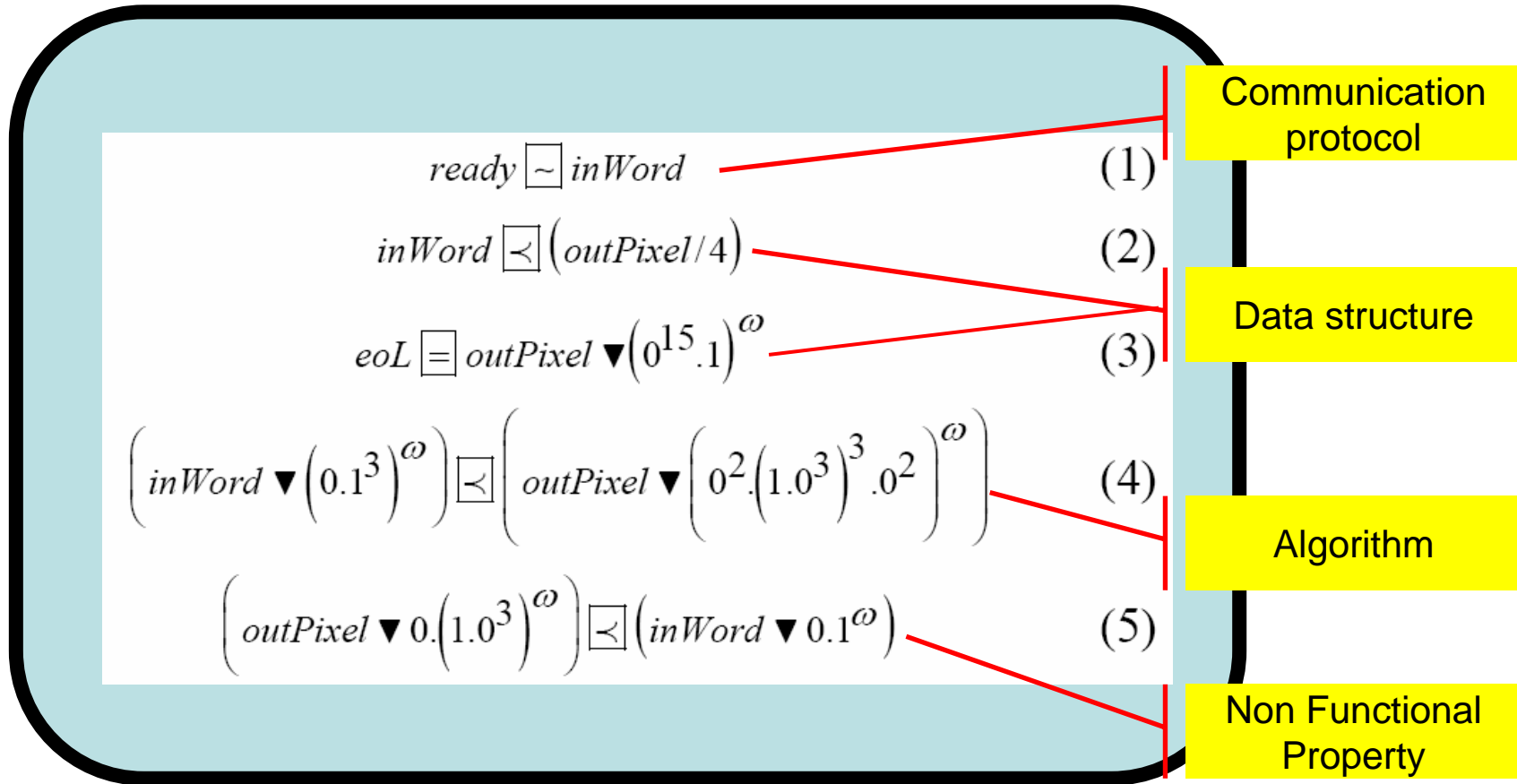
Chaque “tic” d’une horloge logique correspond à une mise à jour de l’information sur le port associé (envoi ou réception)



- Application du stéréotype «clock» sur les ports (qui sont des propriétés UML)

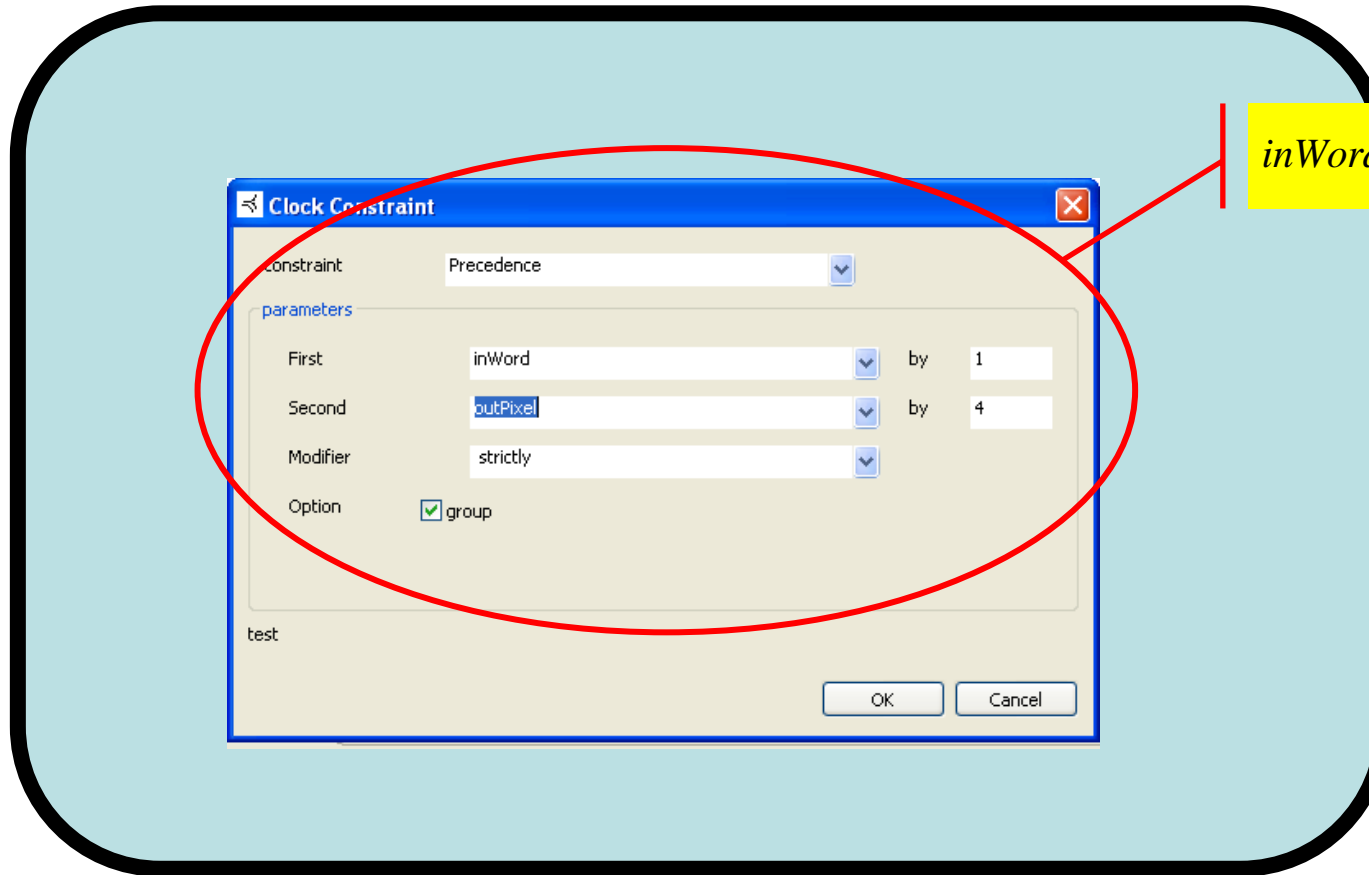


# Contraintes d'horloges



- Spécification des dépendances entre horloges  
(Remarque: ceci est le modèle sous-jacent, pas le format d'entrée de TimeSquare)

# Contraintes d'horloges



- Spécification des contraintes d'horloges via les boîtes de dialogue CCSL de TimeSquare



# Analyse

Simulation option

Name simulation: DigitalFilter2

Boolean 4 guard

Edit smallBuffer:1

length: 50

Mode: Random

Time Scale: tick

Option:  Pulse

Option:  Ghost

Simulation

Start Next \*\*Next End

Conditional relation parameter

Simulation policy

Launch simulation

UML  
MARTE

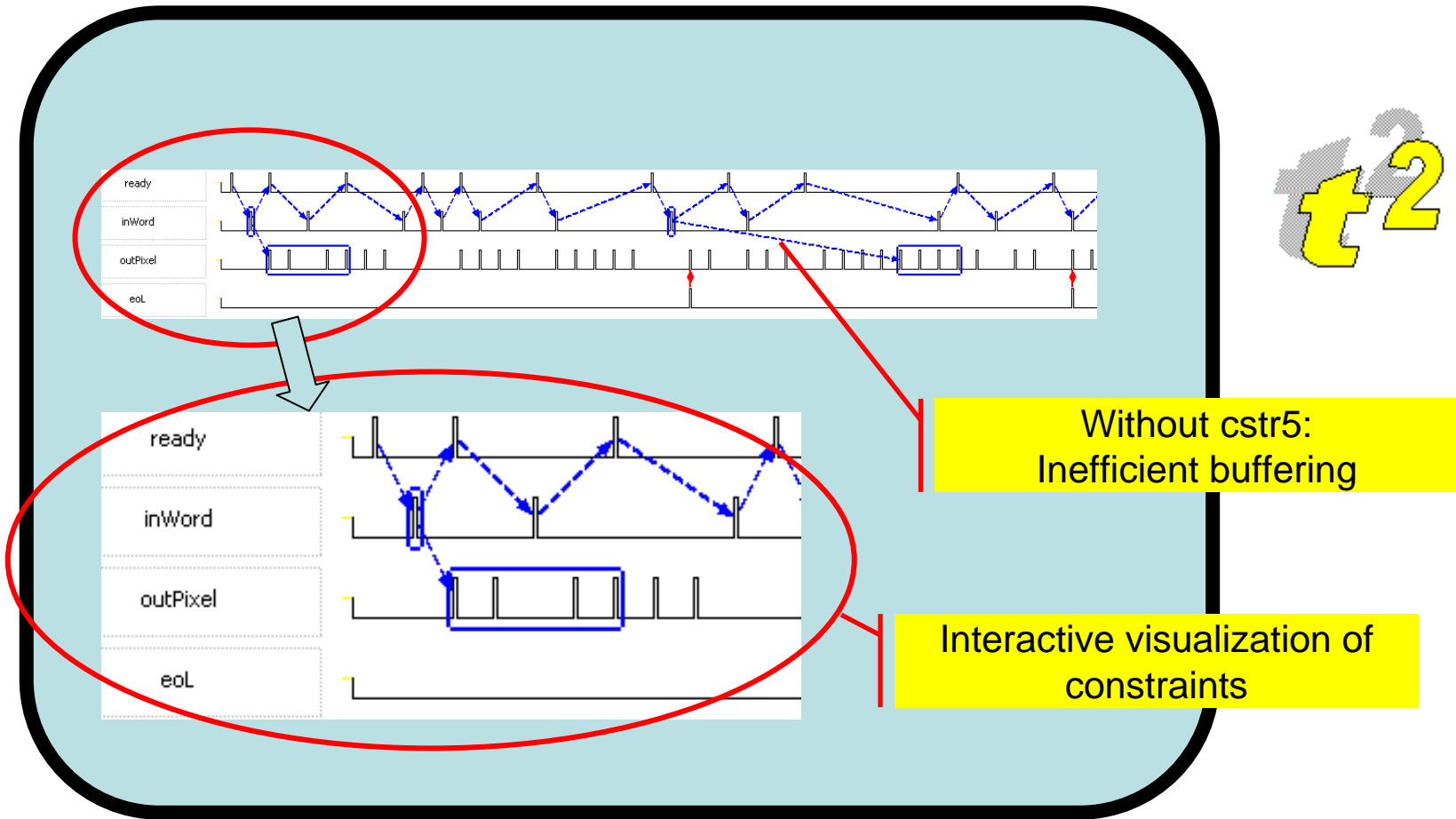
t<sup>2</sup>



- Positionner les paramètres de simulation

# Analyse

$t^2$

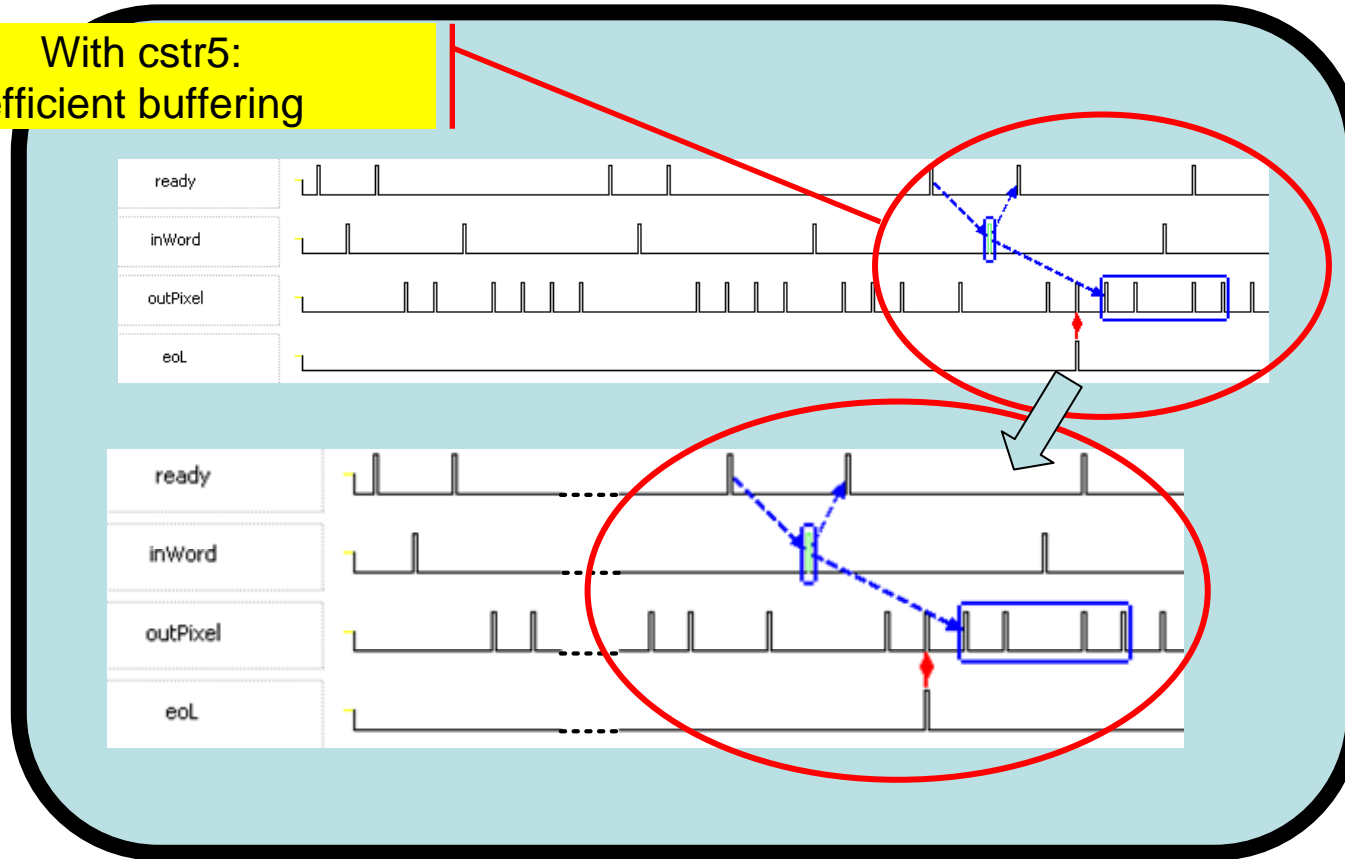


- **Visualiser** et **explorer** les traces de simulation (format VCD + fonctions d'assistance)



# Analyse

With cstr5:  
efficient buffering



$t^2$



- **Optimisation:** imposer une propriété non fonctionnelle (mémoire minimale) par la contrainte cstr 5

# Conclusion

- **Temps en UML/MARTE**: à la fois le temps « physique » (chronométrique) et le temps logique.
- Mise en œuvre: **bénéficie des outils UML** (dont Papyrus)
- Temps + CCSL: **Modèle exécutable** (sémantique formelle) permet simulation et vérification formelle
- CCSL peut être utilisé dans les DSML
- **TimeSquare**: environnement Eclipse dédié disponible