

Outillage pour la modélisation, la vérification et la génération d'applications temporisées et embarquées

P.-E. Hladik, S. Dal Zilio*, O. Pasquier,
S. Pillement, B. Berthomieu



Outillage pour la **modélisation**, la **vérification** et la génération d'applications **temporisées** et embarquées

P.-E. Hladik, S. Dal Zilio*, O. Pasquier,
S. Pillement, B. Berthomieu



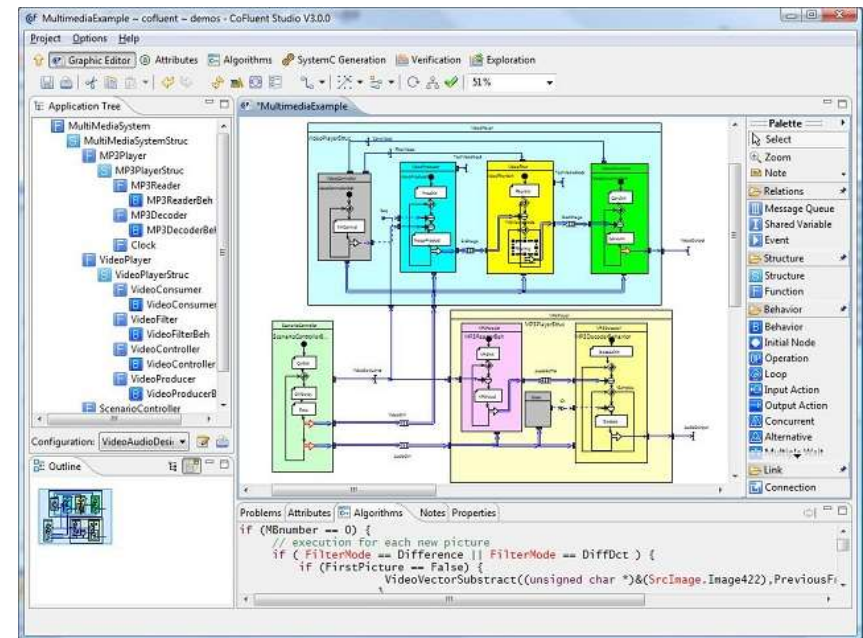
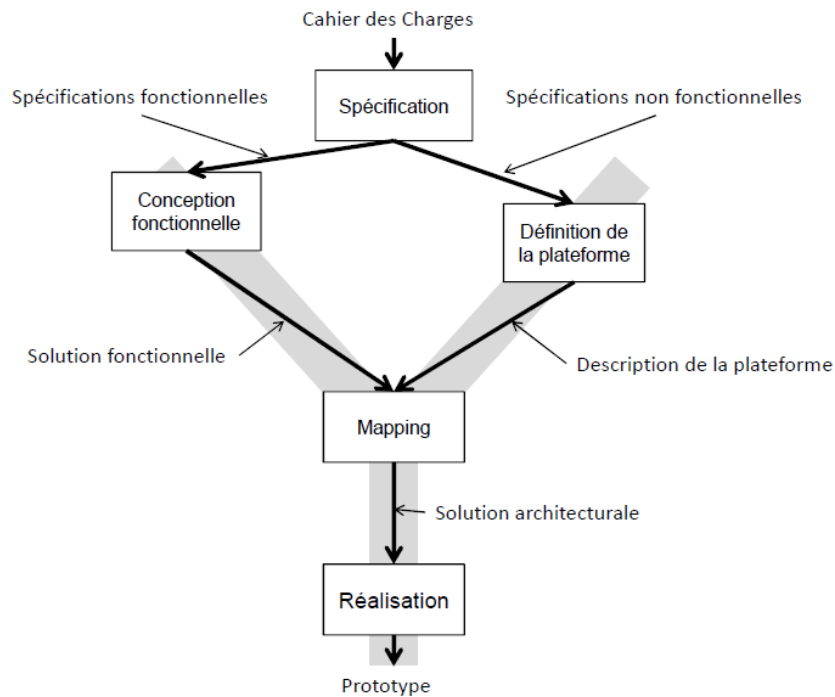
Conception de systèmes embarqués

- les ADLs: UML/MARTE, SysML, EAST-ADL, AADL
 - focus sur le langage plutôt que sur le processus
 - ... , SyncCharts , ... , Capella , ...
- MCSE—Méthodologie de Conception des Systèmes Embarqués (J.-P. Calvez, 1990) $\rightsquigarrow$ Intel CoFluent Studio
 - + simple et accessible
 - sémantique (!?) utilise des contraintes temporelles

Plan

- La *méthodologie* MCSE (Univ. Nantes)
 - Comportements et fonctions
- La boîte à outils Tina (TPN) (LAAS-CNRS)
 - Le langage pivot Fiacre
- Le mélange des deux
 - Modèle d'exécution « time-triggered »

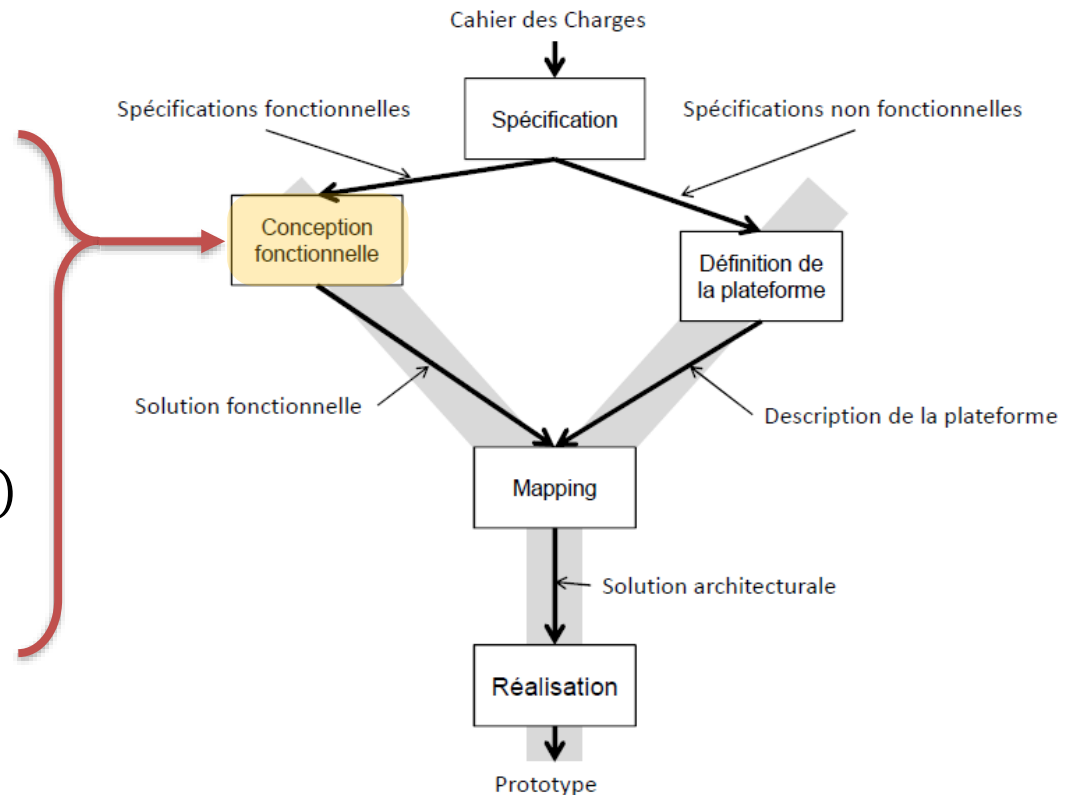
MCSE: Méthodologie de Conception des Systèmes Embarqués

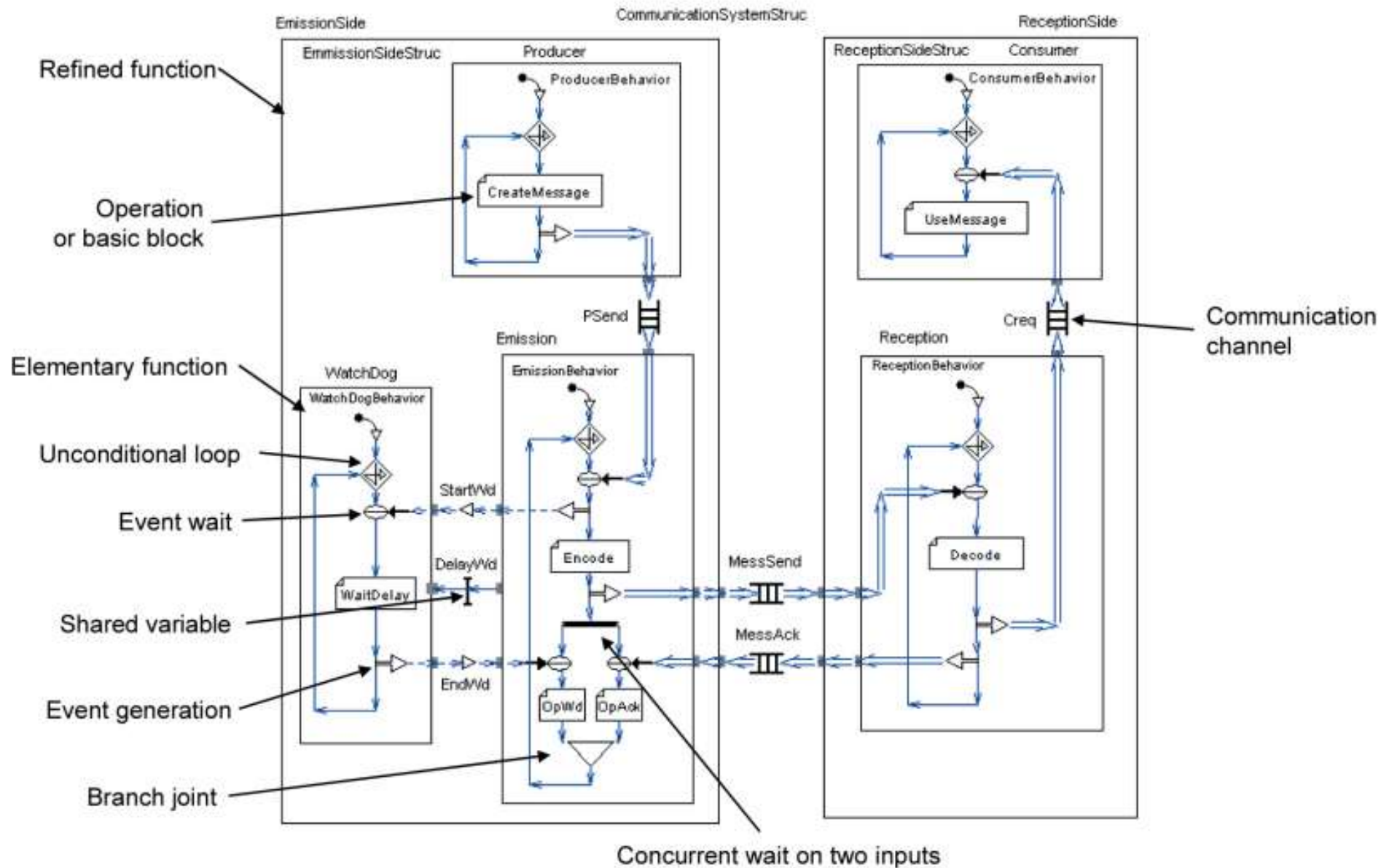


MCSE: Méthodologie de Conception des Systèmes Embarqués

Conception fonctionnelle

1. délimitation des I/O
2. première décomposition fonctionnelle
3. raffinement(s) fonctionnel(s)
4. algorithmes.





Modèle basé LET (Logical Execution Time)

Approche *asynchrone conduite par le temps*

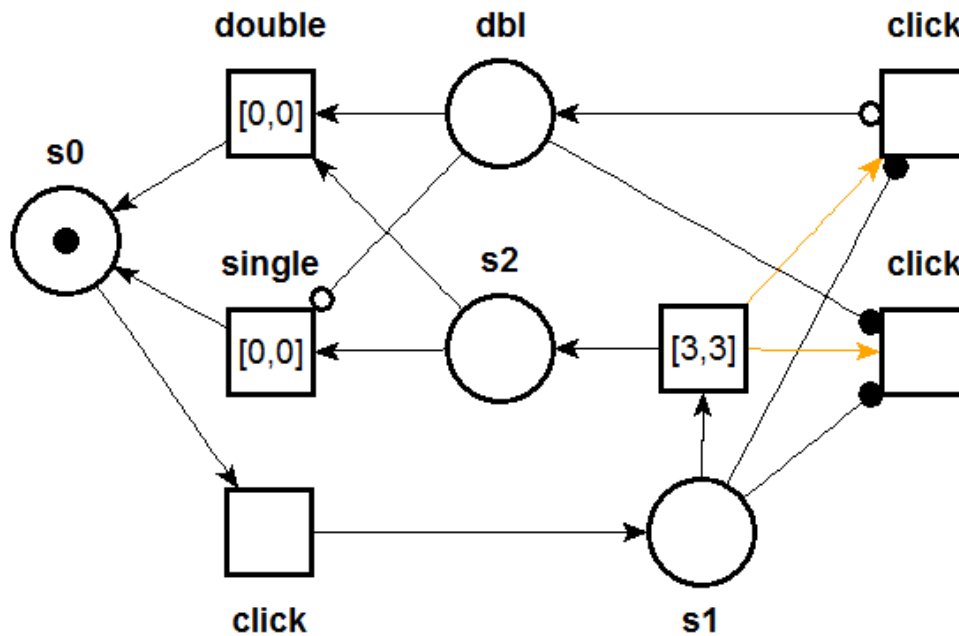
on impose que la date de production des données par une tâche soit « déterministe » et indépendante de la durée du traitement.

- + on peut ignorer les problèmes liés à la préemption (stopwatch)
- on doit s'assurer que les tâches respectent leurs LET (analyse d'ordonnancabilité)

- MCSE: Méthodologie de Conception des Systèmes Embarqués
- **Fiacre et Tina**
- Chaîne de transformation

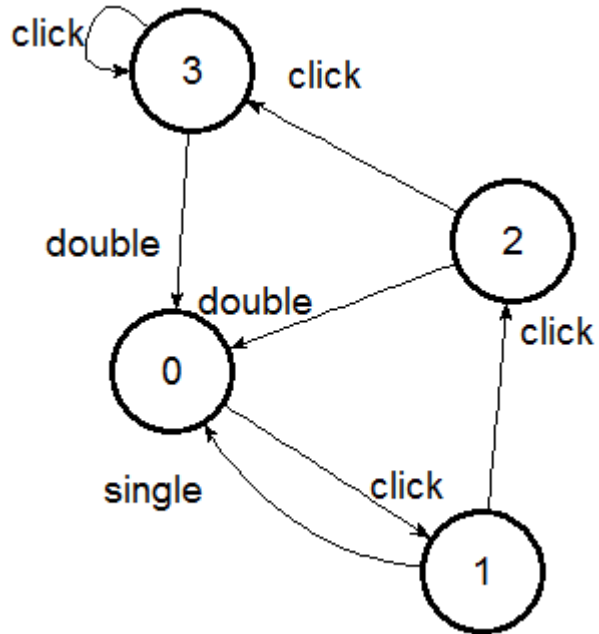
Tina

(www.laas.fr/tina/)



Time Petri Nets [Merlin74]

Model-Checking (CTL, LTL, ...)



class 2

marking

$s0*0 \quad s1*1 \quad dbl*1$

domain

$0 \leq \text{click}$

$0 \leq \text{double} \leq 3$

Always Eventually (single $\vee$ double) **After** first click ?

Fiacre

(www.laas.fr/fiacre/)

```
process Push [click : none,  
              single : none,  
              double : none,  
              delay : none] is
```

```
states s0, s1, s2
```

```
var dbl : bool := false
```

```
from s0 click; to s1
```

```
from s1
```

```
  select
```

```
    click; dbl := true; loop
```

```
  [] delay; to s2
```

```
end
```

```
from s2
```

```
  if dbl then double
```

```
  else single end;
```

```
  dbl := false; to s0
```

```
component Mouse [click : none,  
                  single : none,  
                  double : none] is
```

```
port delay : none in [1,1]
```

```
priority delay > click
```

```
par
```

```
  Push [click, single, double, delay]
```

```
end
```

parameters

```
process Push [click : none,  
              single : none,  
              double : none,  
              delay : none] is
```

```
states s0, s1, s2
```

```
var dbl : bool := false
```

```
from s0 click; to s1
```

```
from s1  
select  
  click; dbl := true; loop  
[] delay; to s2  
end
```

```
from s2  
if dbl then double  
else single end;  
dbl := false; to s0
```

state transition

port declaration

```
component Mouse [click : none,  
                  single : none,  
                  double : none] is
```

```
port delay : none in [1,1]
```

```
priority delay > click
```

```
par
```

```
Push [click, single, double, delay]
```

```
end
```

timing constraints

priorities

process instance

Un exemple simple

process lbuf [put : none, get : none] is

states busy, free, error

from free put ; to busy

from busy

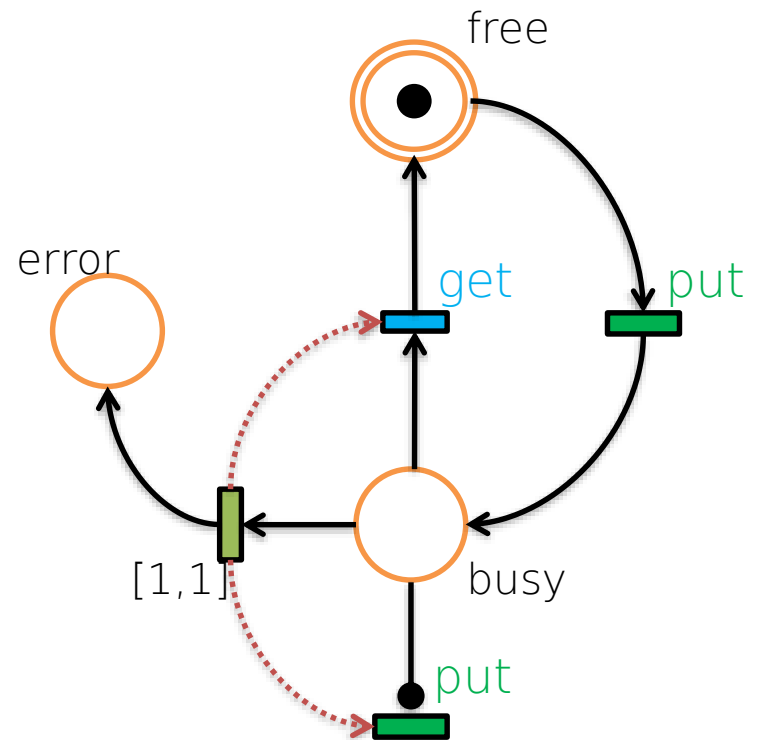
select

put ; loop

[] get ; to free

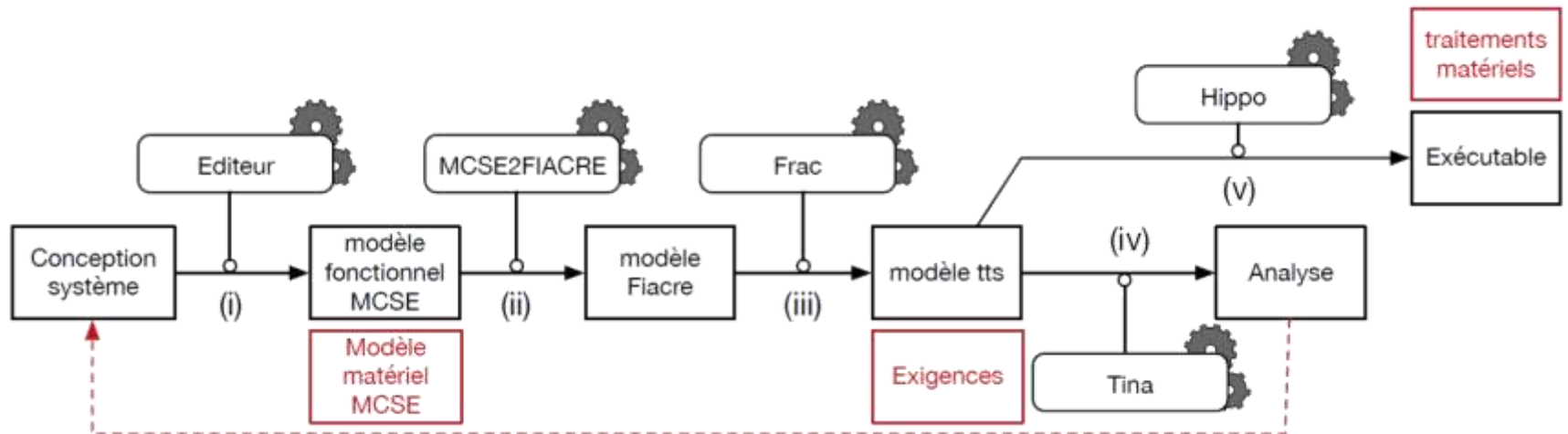
unless wait [1,1] ; to error

end

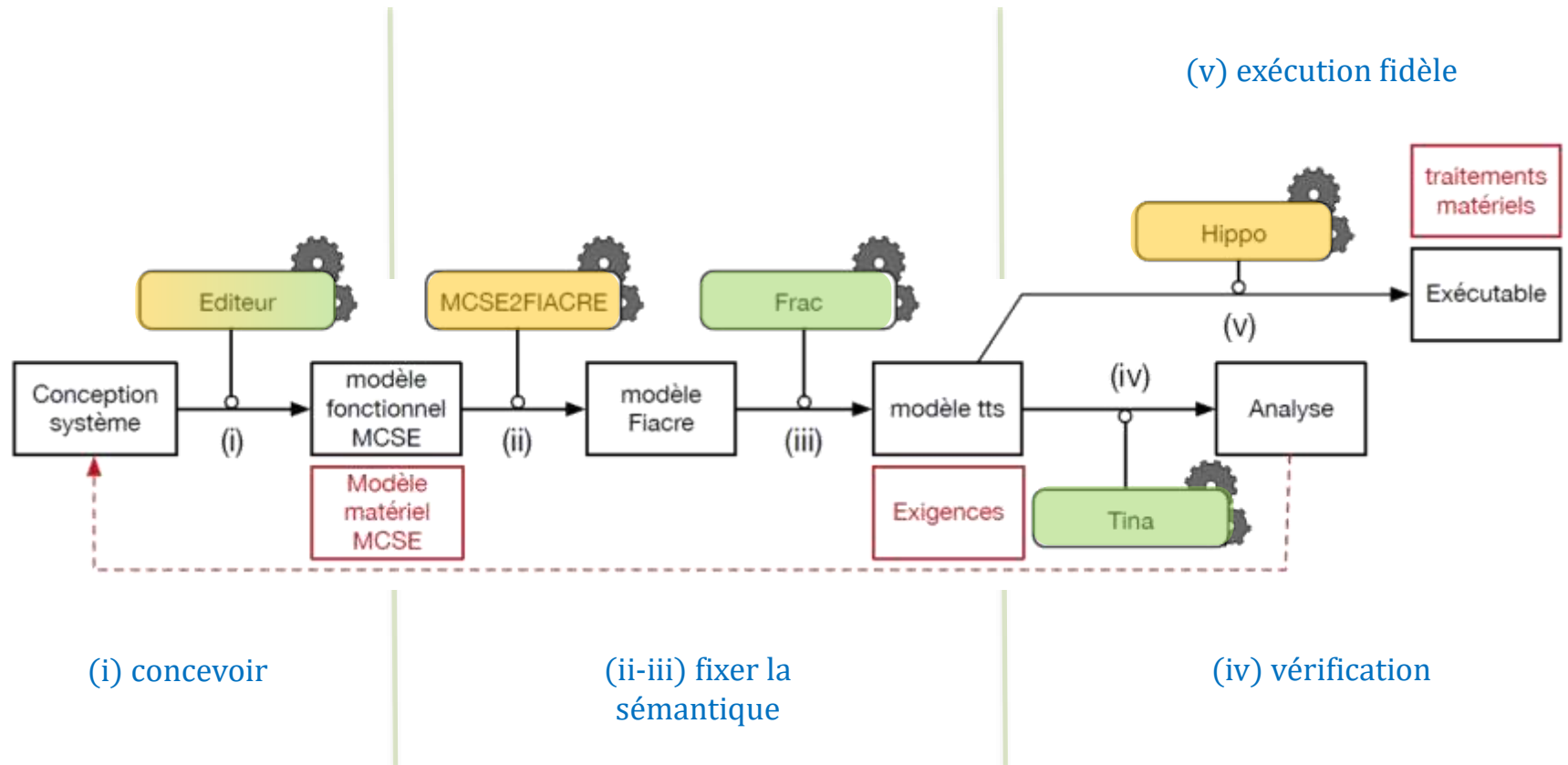


- MCSE: Méthodologie de Conception des Systèmes Embarqués
- Fiacre et Tina
- Chaîne de transformation

La chaîne MCSE2Fiacre

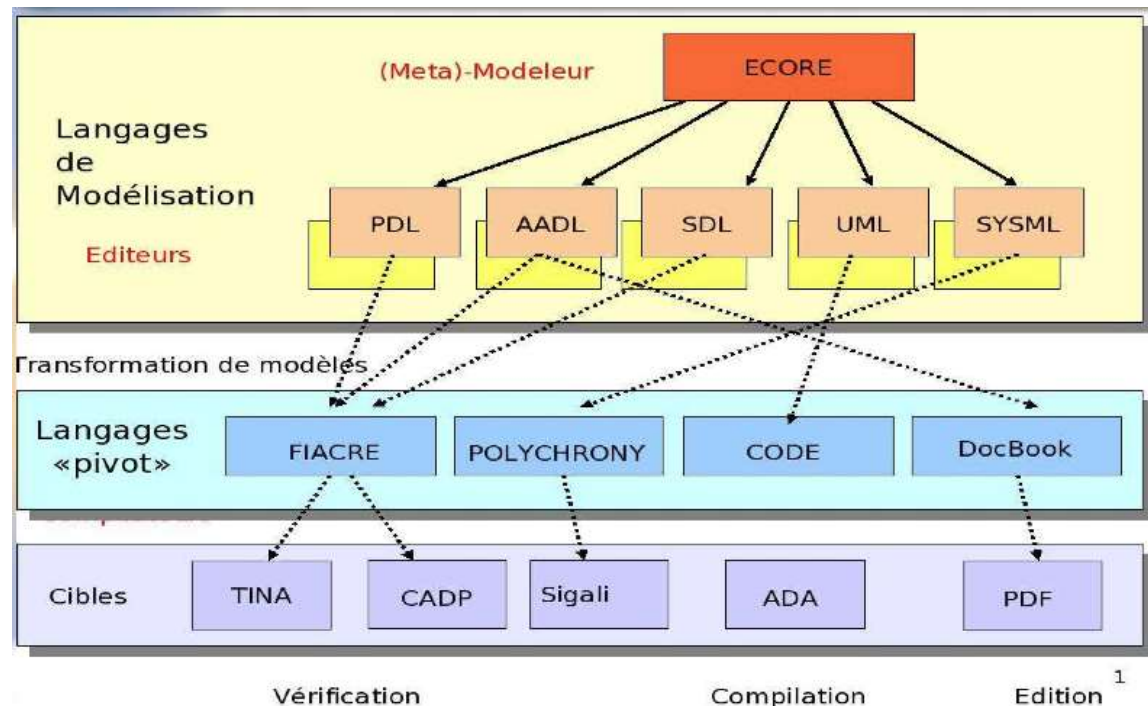


La chaîne MCSE2Fiacre



État de l'art

- Vérification formelle/comportementale pour les ADLs
 - Utilisation d'un langage pivot
 - Travaux (voir e.g. le projet Topcased / Polarsys): ..., BPMN , Genom



État de l'art

- Vérification formelle/comportementale pour les ADLs
- BIP (avec extension temps-réel)
 - focus sur la vérification compositionnelle (DFinder) ; *middleware* d'exécution
- CPAL et langage de simulation
 - Machine virtuelle fidèle pour un “bytecode” formel

Conclusion

- C'est un exemple complet: il y a de la méthodologie ; du temps réel ; du formel ; ...
- Il y a besoin de valider l'ordonnancement
 - méthodes analytiques vs model-checking
- Et pour le futur
 - expression des propriétés sur les modèles ; prise en compte de l'environnement et des contraintes sur le matériel