



Description orientée aspects des systèmes matériels

Assia Touil

Bachir Djafri

LaMI-UMR 8042, CNRS

Université d'Évry Val d'Essonne

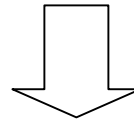
OCM – SI

Le Plan

- Problématique
- Programmation Orientée Aspects
- L'approche composant (VHDL) et la Programmation Orientée Aspects
- Premiers résultats
- Conclusion

Problématique

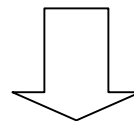
Complexité croissante des systèmes matériels



Tentatives d'amélioration des langages de description matérielle :

Le standard **VHDL** \Rightarrow langage à base de composants

- Abstraction
- Modularité



Travaux effectués / en cours :

Extensions de VHDL vers l'Orienté Objet

Héritage et polymorphisme

\Rightarrow réutilisabilité

Problématique : limites

Ne permet pas une réutilisation maximale

⇒ Mélange des parties fonctionnelles et non fonctionnelles

```
process  
begin
```

```
A <= B or C after T; -- T = 10 ns (T<N)
```

```
wait for N; -- N = 20 ns
```

```
end process;
```

partie fonctionnelle :
affectation d' un signal

partie non fonctionnelle :
la périodicité


Solution :

Application du paradigme de la programmation orientée aspects sur les composants décrits avec VHDL

Programmation orientée aspects

- **Principe** : ‘Séparation des préoccupations’
(*separation of concerns*)

Séparation des propriétés fonctionnelles (quoi) et non fonctionnelles (comment) dans des unités isolées : **Les aspects**

- **Les types d’aspects** 
 - Aspects fonctionnels
 - Aspects non fonctionnels

- **But** :
 - Réutilisabilité
 - Lisibilité

Programmation orientée aspect

Exemple d'une application avec des parties non fonctionnelles avant la séparation des préoccupations :

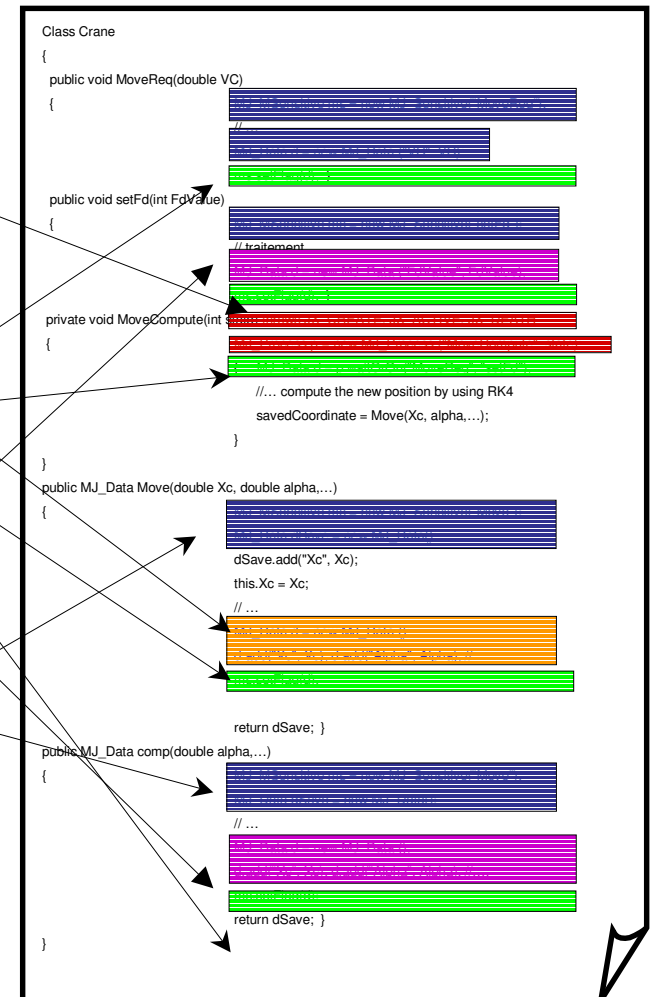
■ concurrence

■ sécurité

■ synchronisation

■ Contraintes de temps

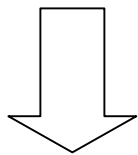
■ communication



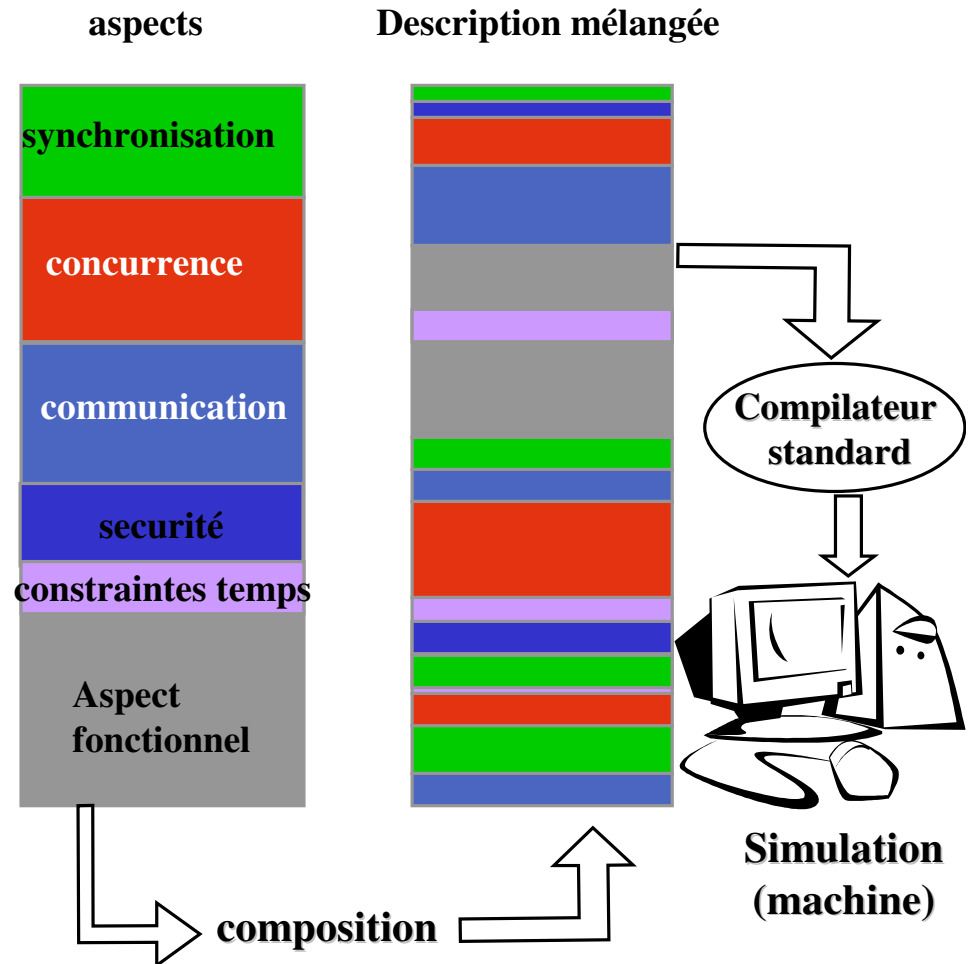
Programmation orientée aspect

*Réalisation de
l'application :*

définition des aspects



composition des aspects



L'approche composant et la Programmation orientée aspects

VHDL

- Langage de description de systèmes matériels (composants matériels)
- Puissant : description et simulation
- Basé sur l'approche composant
 - ⇒ composant : entité / architecture
 - ⇒ entité : interface
 - ⇒ architecture : comportement
 - ⇒ signaux : connecteurs
 - ⇒ concurrence
- Entité / architecture : description du comportement du système et de ses composants

L'approche composant et la Programmation orientée aspects

- Introduction de la programmation orientée aspects :
 - meilleure structuration des parties fonctionnelles et non fonctionnelles
- Ensemble de règles sans modifications de la syntaxe:
 - Description de chaque aspect dans une entité/architecture

Premiers résultats

Les étapes à suivre :

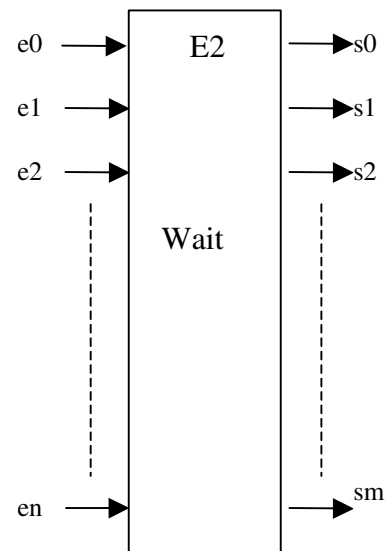
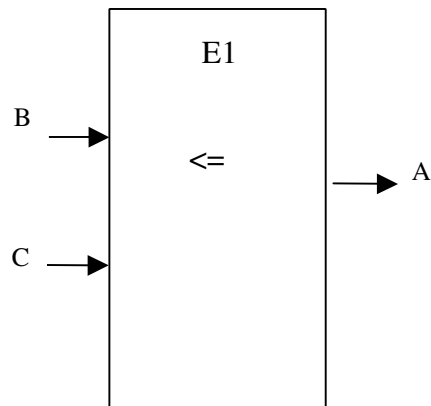
1. Identification des parties fonctionnelles et non fonctionnelles

Exemple :

```
process
begin   A <= B or C after T;
       wait for N;
end process;
```

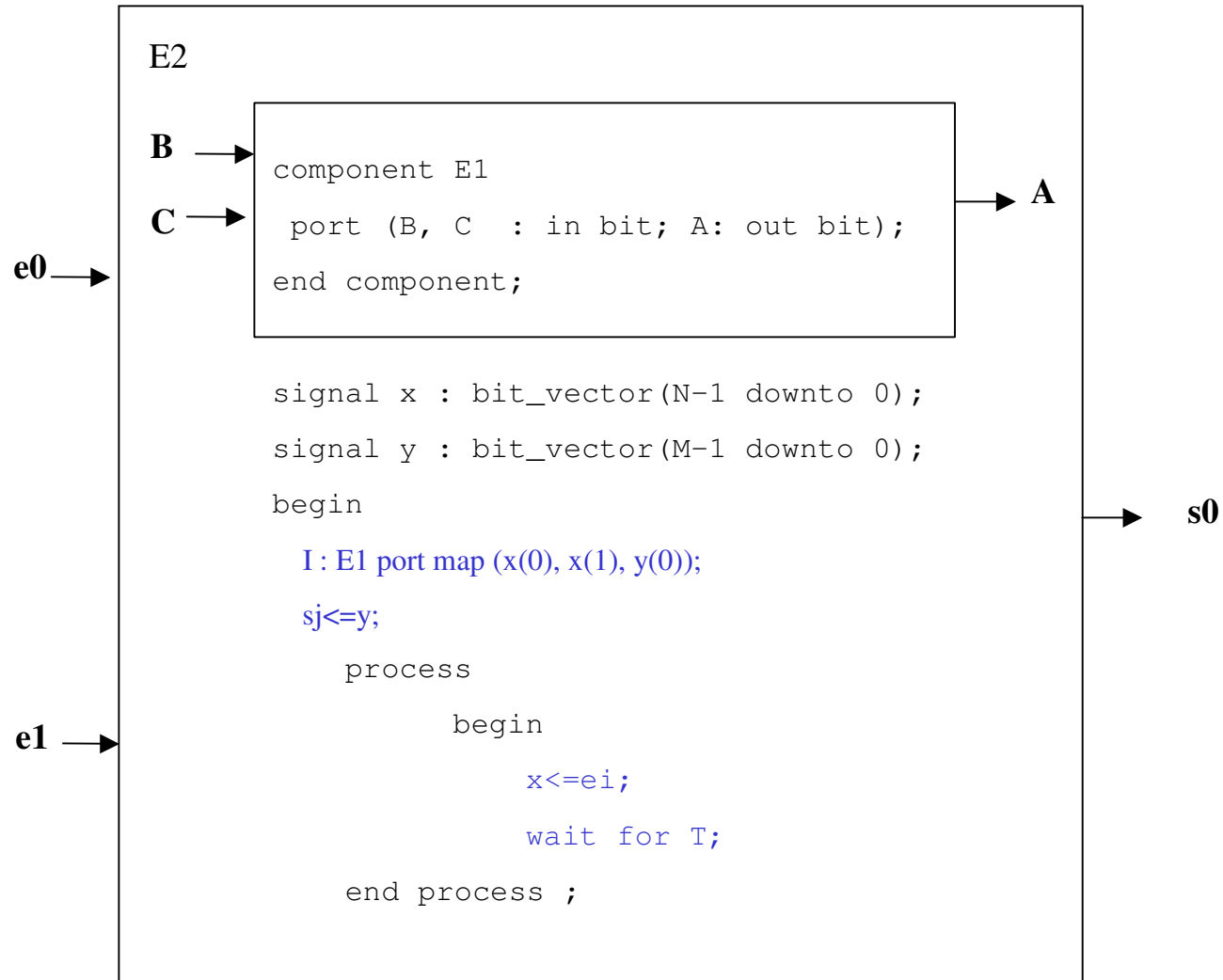
2. Description de ces parties dans des aspects :

Entité/Architecture



Premiers résultats

3. Composition des aspects entre eux



Conclusion

L'intégration des concepts de la programmation orientée aspects dans le langage de description des systèmes matériels *VHDL*

- ◇ approche pour une meilleure réutilisabilité et modularité
- ◇ Une méthodologie de travail

Perspectives :

Analyser les concepts du langage *VHDL*

Exprimer les aspects fonctionnels et non fonctionnels

Configuration \rightleftharpoons Entité/architecture

Exemple : le processus \rightleftharpoons boucle

Composition des aspects non fonctionnels