


## Vers une adaptation dynamique cohérente des composants

Présentée par Audrey Ocello



Equipe Rainbow - Laboratoire I3S 

Journées Composants 2002 :  
« Systèmes à composants adaptables et extensibles »

---

---

---

---

---

---

---

---

### Impacts des adaptations

- Modification des interfaces d'un composant
  - Ajout/retrait de messages
  - Modification de la signature de messages
  - ↑ Changement du « type » du composant
- Modification du comportement d'un composant
  - Nouveaux comportements associés à un message
  - ↑ Composition des différents comportements
- Modification des éléments requis par un composant
  - Connexions/déconnexions de composants
  - ↑ Impacts sur le graphe d'interactions entre composants

JC2002 - 07/11/2002

2

Equipe Rainbow

---

---

---

---

---

---

---

---

### Etat de l'art

	Typage	Composition
<b>Réécriture de code</b> <i>(AspectJ, OpenC++)</i>	changement du type (statique)	relation de priorité, ordre de déclaration
<b>Modification de la hiérarchie de classes</b> <i>(JAC, JavaPod)</i>	ajout seulement	chaînage d'aspects
<b>Composants : approche statique</b> <i>(EJB, CCM)</i>	pas d'évolution dynamique des ports	composition limitée mais automatisée
<b>Composants : approche dynamique</b> <i>(Molène, Fractal)</i>	évolution dynamique des assemblages, « containment relationship »	grain de composition + global (controller)
<b>Langages d'interactions</b> <i>(ISL)</i>	abs. de modification d'interfaces	mécanisme de fusion sur règles d'interactions

JC2002 - 07/11/2002

3

Equipe Rainbow

---

---

---

---

---

---

---

---

## Des types aux rôles

- **Programmation par contrats : Cas d'utilisation**
  - Assertions (pré/post conditions)
- **Modèles de composants : Distinction fournis/requis**
  - CCM, Fractal (interface serveur/client)
- **UML 2.0 : Spécification de composants**
  - Introduction d'automates d'interfaces

↑ **Aborder le problème de la cohérence par le typage**

↑ **Embarquer + d'informations dans le type**

- contrôles sur l'exécution des messages,
- conditionnelles sur les adaptations acceptées ou refusées,
- ...

↑ **Evolution du « type »**

⇒ **Rôles**

JC2002 - 07/11/2002

4

Equipe Rainbow

---

---

---

---

---

---

---

---

## Modèle pour Composants Adaptables (1/2)

- Permet de contrôler l'évolution des composants
- Comprend les étapes :
  - d'assemblage initial de composants
  - d'adaptation des composants

JC2002 - 07/11/2002

5

Equipe Rainbow

---

---

---

---

---

---

---

---

## Modèle pour Composants Adaptables (2/2)

- **Port :**
  - simple : traite un message.
  - générique : décrit un ensemble de ports simples.
- **Rôle :** Fonction, emploi, action, influence exercée.
  - décrit les propriétés attendues de l'entité associée.
  - comporte un ensemble de :
    - ports
    - rôles de composants **requis**
- **Composant :**
  - un rôle lui est associé.
  - adapté à certains moments et à certaines conditions par modification du rôle.

JC2002 - 07/11/2002

6

Equipe Rainbow

---

---

---

---

---

---

---

---

## Exemple d'adaptation

- Rôle de base d'un agenda :
  - ports fournis addRDV(RDV), removeRDV(RDV), print()
- Règle d'adaptation :
  - agenda.addRDV(rdv) -> bd.update(rdv); \_call()
- Rôle d'un agenda modifié par adaptation :
  - ports fournis addRDV(RDV), removeRDV(RDV), print()
  - ports contrôlés : addRDV(RDV)
  - rôles de composants requis : rôle de base de données
  - ports émis : bd.update(RDV)

↑ Modification du graphe d'interactions entre composants  
ajout d'une arrête entre les composants agenda et bd

JC2002 - 07/11/2002

7

Equipe Rainbow

---

---

---

---

---

---

---

---

## Cohérence des adaptations

- Cohérence structurelle (C1)
  - Tout composant requis pour un rôle donné (rôle attendu) doit toujours pouvoir correspondre à ce rôle.
- Cohérence comportementale
  - Locale à un composant (C2) : Compatibilité des contrôles exercés.
  - Locale à une adaptation (C3) : Conservation / restauration de l'état des composants.
  - Globale au réseau (C4) : Cycles, points de non déterminisme.

↑ Cohérence sémantique : Nommage, antagonisme.

JC2002 - 07/11/2002

8

Equipe Rainbow

---

---

---

---

---

---

---

---

## Du modèle aux propriétés vérifiées

	C1	C2	C3	C4
Rôles bien formés	✓			
Composants bien formés	✓			
Sous rôle	✓			
Applicabilité de rôle		✓		
Schémas bien formés	✓			
Interactions bien formées			✓	
Application rôle (post cond)	✓			
Application schéma (post cond)	✓			

JC2002 - 07/11/2002

9

Equipe Rainbow

---

---

---

---

---

---

---

---

## Perspectives ...

- Immersion du modèle formel dans les modèles à composants
  - limites du modèle actuel,
  - apports des propriétés vérifiées au niveau du modèle sur la cible.
- Apports pour les outils de développement
- Rôle : un atout pour la réutilisation et interopérabilité

↑ Avoir un modèle formel en adéquation avec la problématique d'adaptations cohérentes.

---

---

---

---

---

---

---

---