



Génération de composants testables avec testeur distribué

N icolas Belloir

Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour

Equipe AOC (Agent-Objet-Composant)

Université de Pau et des Pays de l'Adour

France



Contexte

Contexte

Le Built-In Test

La librairie
BIT/J

Démonstration

Conclusion

- v Assemblage de composants réutilisables

- v Composants

- v Validés ou certifiés

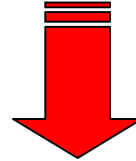
- v Par les **fournisseurs** de composants

- v Dans un **environnement de développement**

- v Intégrés et Déployés

- v Par un **utilisateur** de composant

- v Dans un **environnement de déploiement**



- v Nécessité d'accroître la **testabilité** des composants

Intégration de test dans les composants

Contexte

Le Built-In Test

La librairie
BIT/J

Démonstration

Conclusion

v Approches de test intégré dans les composants

- v Composant auto testable [Wang98]
- v Processus de développement des jeux de test [JDT01]
- v Approche par **composant testable** (RM I) [Gao02]

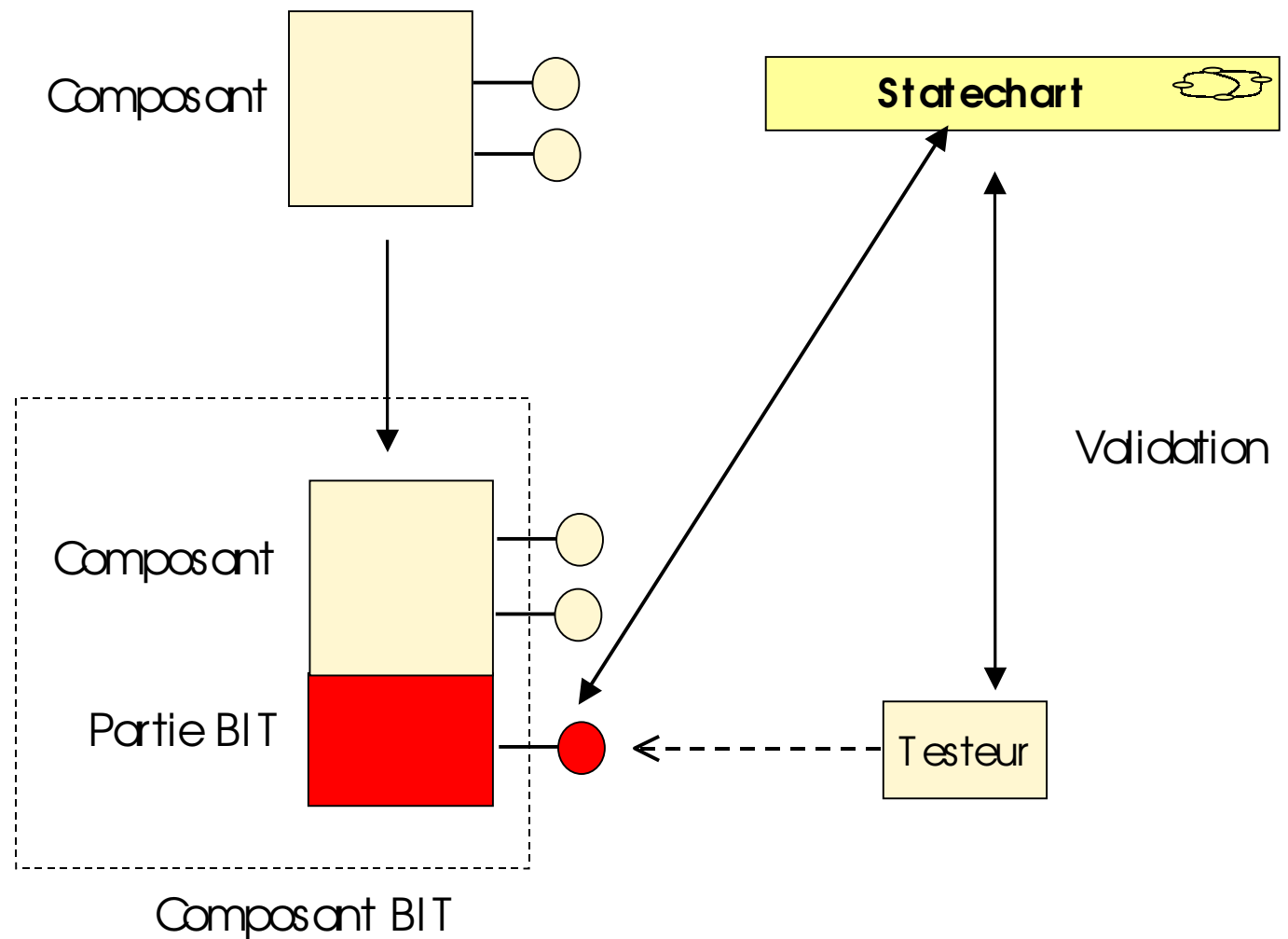
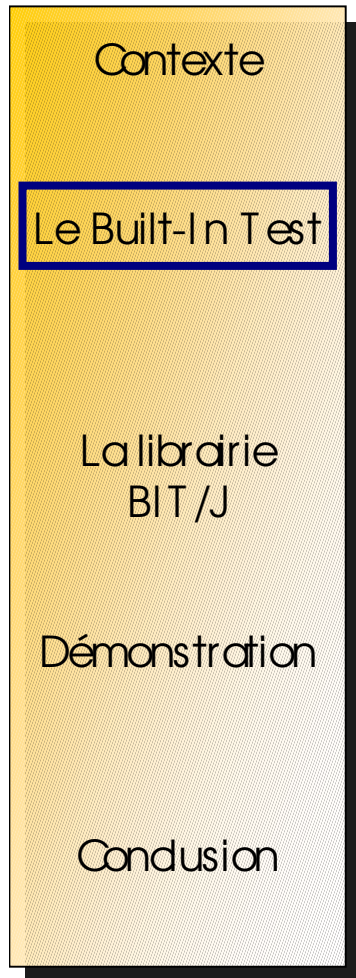
v La technologie **BIT** (Built-In Test)

- v Développée par le projet Européen **Component+**
- v Ajout au composant d'une *interface générique de test*
- v Tests basés sur la notion de **contrat**
- v Tests sur les aspects **comportementaux** des composants



<http://www.component-plus.org>

La technologie BIT : illustration





La librairie BIT /J (1)

Contexte

Le Built-In Test

La librairie
BIT /J

Démonstration

Conclusion

- v Technique : Java, réflexion Java

- v Fonctionnalités (version 1)

- v Mise en œuvre de la technologie BIT
- v Composants "in house"
- v Description du comportement par Statecharts

- v Lacunes

- v Mise en œuvre complexe
- v Non prise en compte des COTS
- v Interface non interactive

- v Fonctionnalités (version 2)

- v Outil de génération automatique
- v Composants COTS
- v Testeur distribué avec interface dynamique

La librairie BIT / J (2)

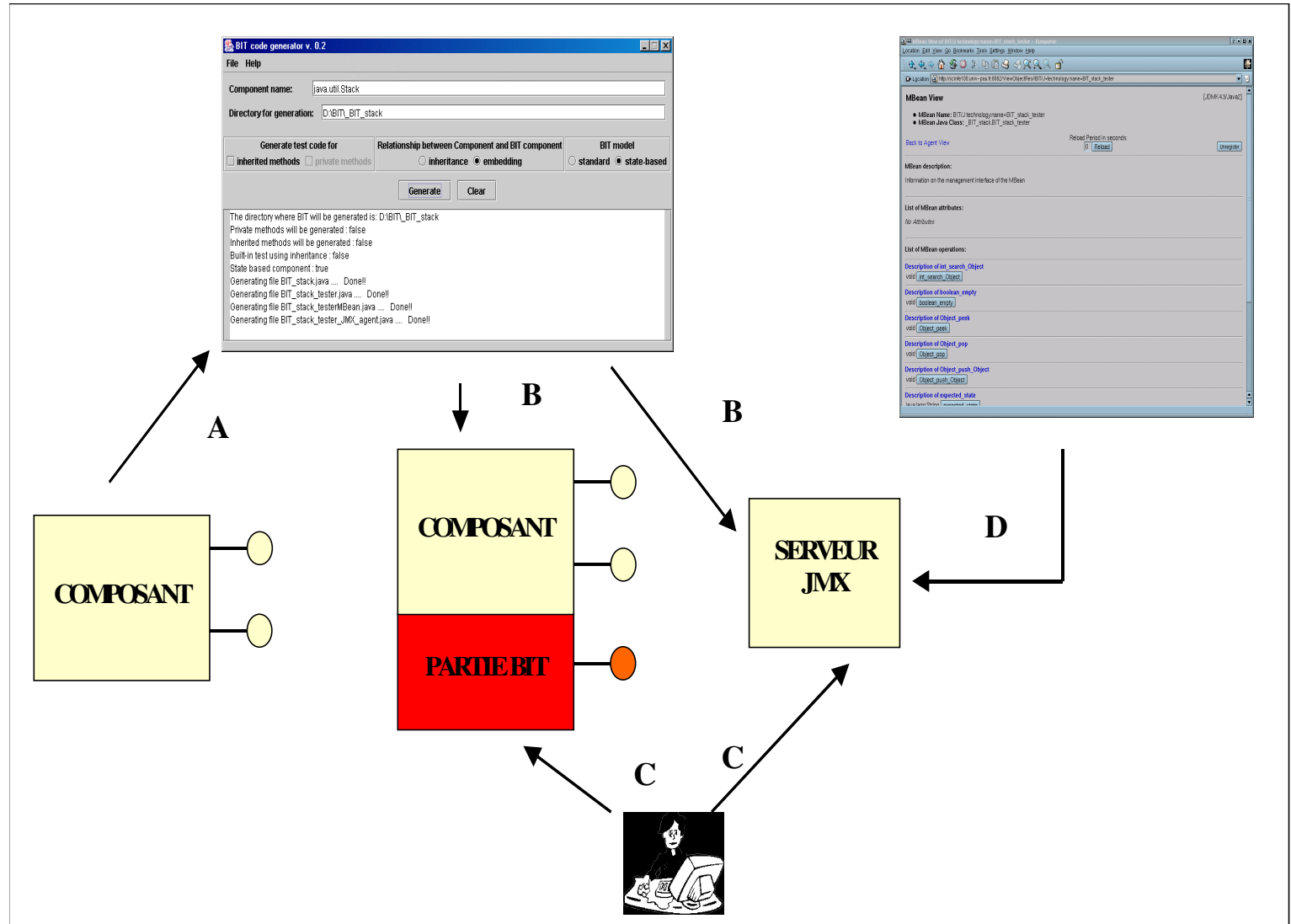
Contexte

Le Built-In Test

La librairie BIT / J

Démonstration

Conclusion



Démonstration

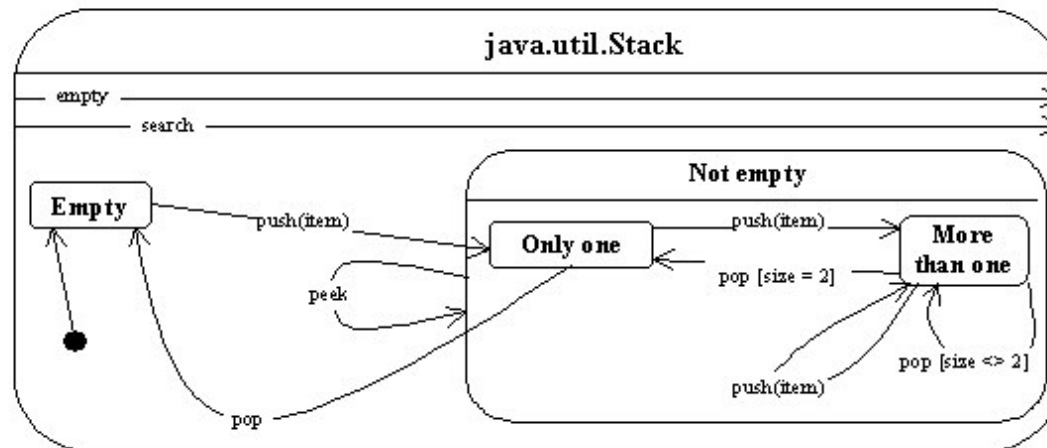
Contexte

Le Built-In Test

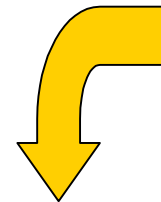
La librairie
BIT / J

Démonstration

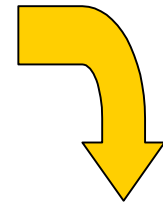
Conclusion



```
_Empty = new BIT_state("Empty");  
_Only_one = new BIT_state("Only one");  
_More_than_one = new BIT_state("More than one");  
_Not_empty = (BIT_state) (_Only_one.xor(_More_than_one)).name("Not empty");  
_BIT_stack = new BIT_state_monitor(_Empty.xor(_Not_empty),"BIT stack");  
_Empty.inputState();
```



```
public java.lang.Object push(java.lang.Object o1) throws Statechart_exception {  
    java.lang.Object result = _stack.push(o1);  
    /* state transitions here: \_BIT\_stack.fires(fromState, toState); */  
    _BIT_stack.fires(_Empty, _Only_one);  
    _BIT_stack.fires(_Only_one, _More_than_one);  
    _BIT_stack.fires(_More_than_one, _More_than_one);  
    _BIT_stack.used_up();  
    return result; }  
}
```





Conclusion

Contexte

Le Built-In Test

La librairie
BIT/J

Démonstration

Conclusion

v Résultats :

- v Prise en compte des CO TS
- v Interface graphique distribuée
- v Outil de génération de code de test
- v Validation de la librairie sur projet industriels

v Perspectives

- v Extension aux composants distribués
- v Intégration dans un processus complet de développement
- v Etude des possibilités de JMX pour la composition
- v Retour sur utilisation (librairie téléchargeable)



Merçi de votre attention

Contexte

Le Built-In Test

La librairie
BIT/J

Démonstration

Conclution

Des questions ?

Quelques liens :

Equipe AOC :

<http://liuppa.univ-pau.fr/themes/aoc/aoc/>

Page perso :

<http://www.univ-pau.fr/~belloir>

Component+ :

<http://www.component-plus.org>

email :

nicolas.belloir@univ-pau.fr