

The slide features a black background with white text. On the left side, there is a vertical grey bar with the word 'BorCon' written vertically in white. The main title 'Course # 2104' is centered at the top, followed by 'Design by contract' in a larger font. Below that, the presenter's name 'Olivier Mangez - Cross Systems' is centered. In the bottom right corner, there is a logo for 'Conférence Borland' with the dates 'Paris, 21 & 22 novembre 2002' underneath it.

- **Consultant** chez Cross Systems
 - UML, méthodes associées à UML (RUP, XP)

Erreur

- **Différence entre :**
 - **Ce que doit faire le système**
 - **Ce fait le système**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- **Erreur = différence** entre :
 - Ce que doit faire le système (spec)
 - Ce fait le système (réalité)
- **PB : coder == être dans la réalité**
- Ce qu'on recherche :
 - **Réduire les risques** de bugs
 - **Mettre en évidence** plus facilement les bugs
- Comment ?
 - Introduire les **spec dans le code**

Plan

- **Assertion**
- Design by contract
- Tests unitaires

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Assertion

- Un élément de spécification...
- ... placé dans le code

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Exprimer le « Quoi ? »
- plutôt que le « Comment ? »

Exemple

```
public class AClass {  
    public void aMethod() {  
        ...  
        // ici, squareRoot devrait être positif  
        ...  
    }  
}
```

Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Exprimer, **rendre explicite** :
 - conviction du développeur
 - certaines hypothèses
- Facilite la **maintenance**
- **Système correct** :
 - toutes les assertions sont vraies

Exemple

```
public class AClass {  
    public void aMethod() {  
        ...  
        assert squareRoot > 0;  
        ...  
    }  
}
```

Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- condition = assertion
- `assert` : **exprimer** l'assertion (JDK 1.4)
- pourquoi pas un commentaire ?
 - `assert` : **vérifier** l'assertion
 - On n'est jamais à l'abri d'un bug, mieux vaut être prévenu lorsqu'il survient
- **assertion fausse => bug**
 - prévenir voire arrêter
 - (exemple des instruments de pilotage)
- **Désactivable**
- ***** Démonstration *****

Erreurs de saisie

- Assertion fausse = bug

```
System.out.println("O : valider ; N : annuler");
int character = System.in.read();
// l'utilisateur doit saisir 'O' ou 'N'
assert character == 'N' || character == 'O';
...
```

Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Pour bien comprendre les assertions :
 - 2 contre-exemples

Effets de bord

- Assertion désactivable

```
Stack stack = new Stack();  
...  
// l'élément supérieur devrait être null  
assert stack.pop() == null;  
...
```

Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Les autres langages

- **Java** : **Debug**
- **C / C++** : **assert.h**
- **Delphi** : **Assertion**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Plan

- Assertion
- **Design by contract**
- Tests unitaires

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Design by contract

- **Contrat entre les différents modules**
 - **Documenter**
 - **Simplifier**
 - **Vérifier**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Assertion : utile en POO et en procedural
- Problématique en POO :
 - **module ne sait pas comment va être appelé**
 - => prendre en compte tous les cas
 - => plus complexe (ça coûte cher)
 - => plus de risque de bugs
- Comment éviter cela ?
 - Comme dans la vie : contrat
 - droit et responsabilité de chaque module
 - Mise en œuvre des assertions
- But : **simplifier tout en évitant les erreurs**
 - Ne pas gérer tous les cas
 - exprimer clairement le cadre d'utilisation et le documenter

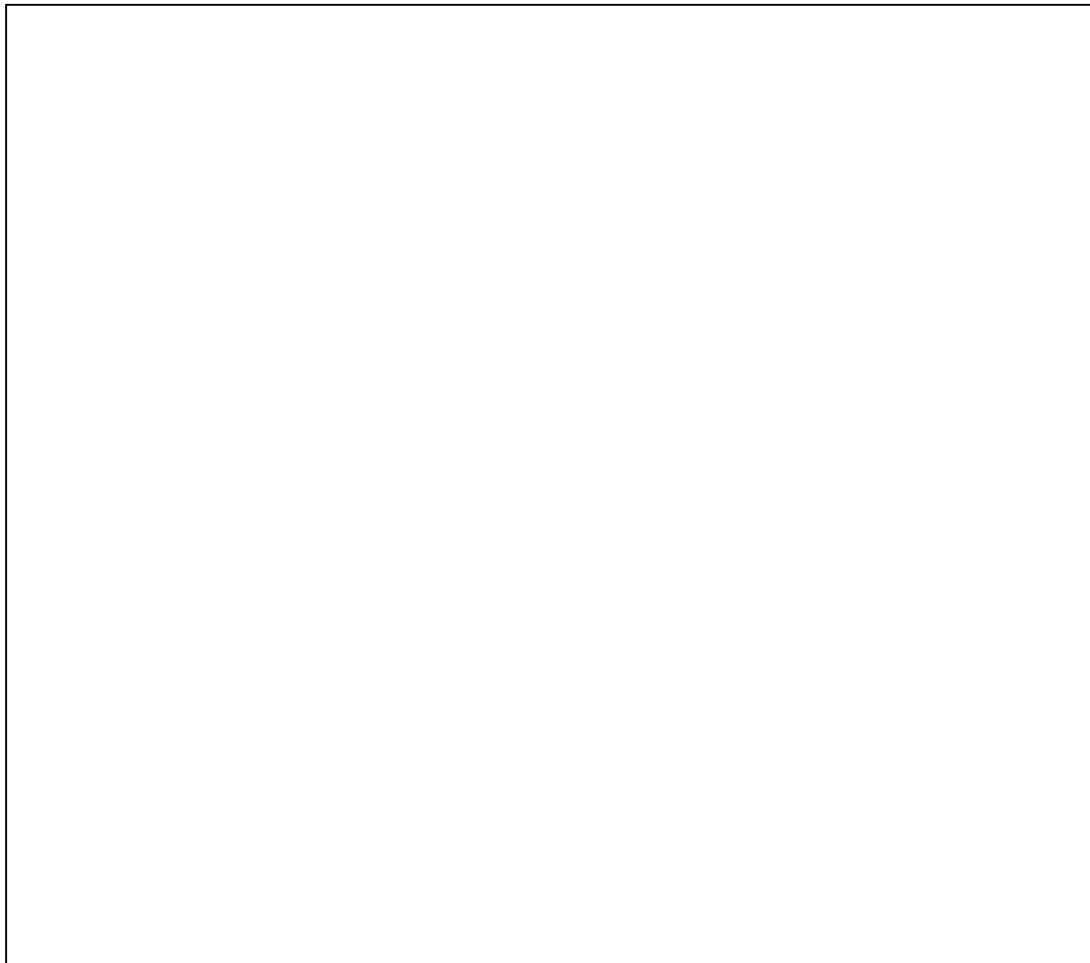
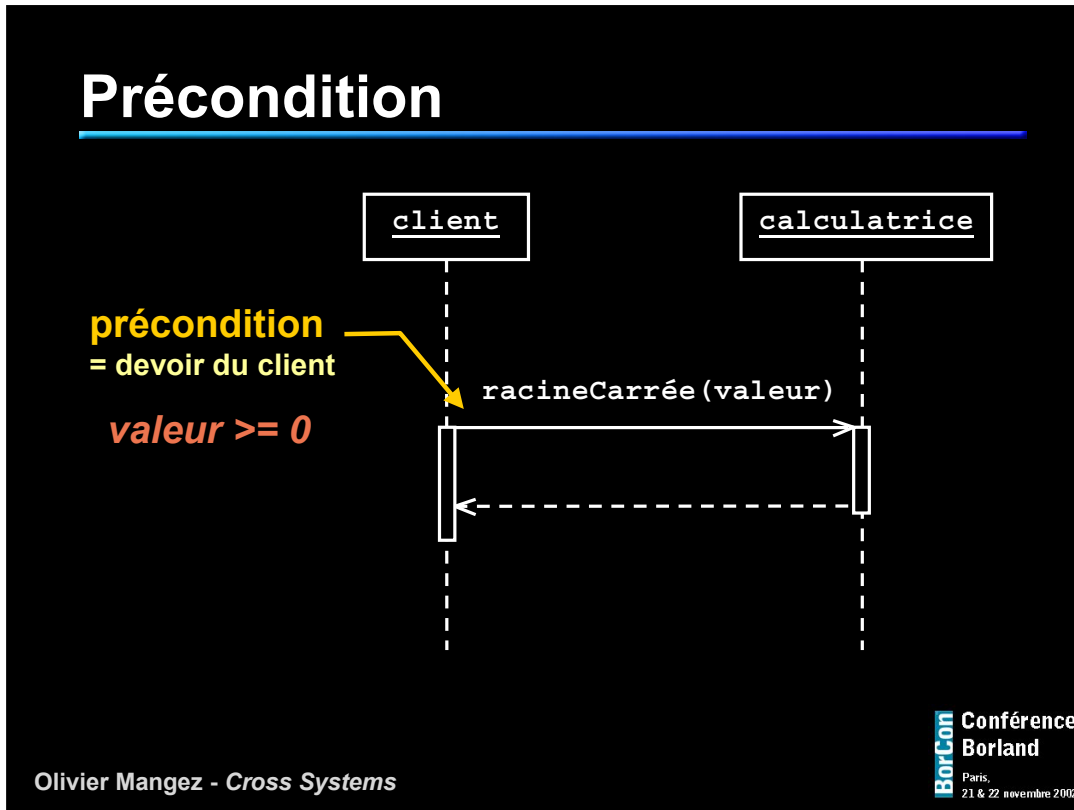
Précondition

- A l'appel d'une méthode
- Devoirs du client
- Bénéfices du fournisseur

Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Conditions vérifiées à l'appel d'une méthode
 - **Devoir du client** :
 - Violation = erreur chez le client
 - **Bénéfice du fournisseur** :
 - Il n'est pas obligé de vérifier la précond
 - Il n'est pas obligé d'avoir un comportement spécifique si la précond est fausse
- Une précondition peut porter sur
 - Les paramètres
 - Les attributs



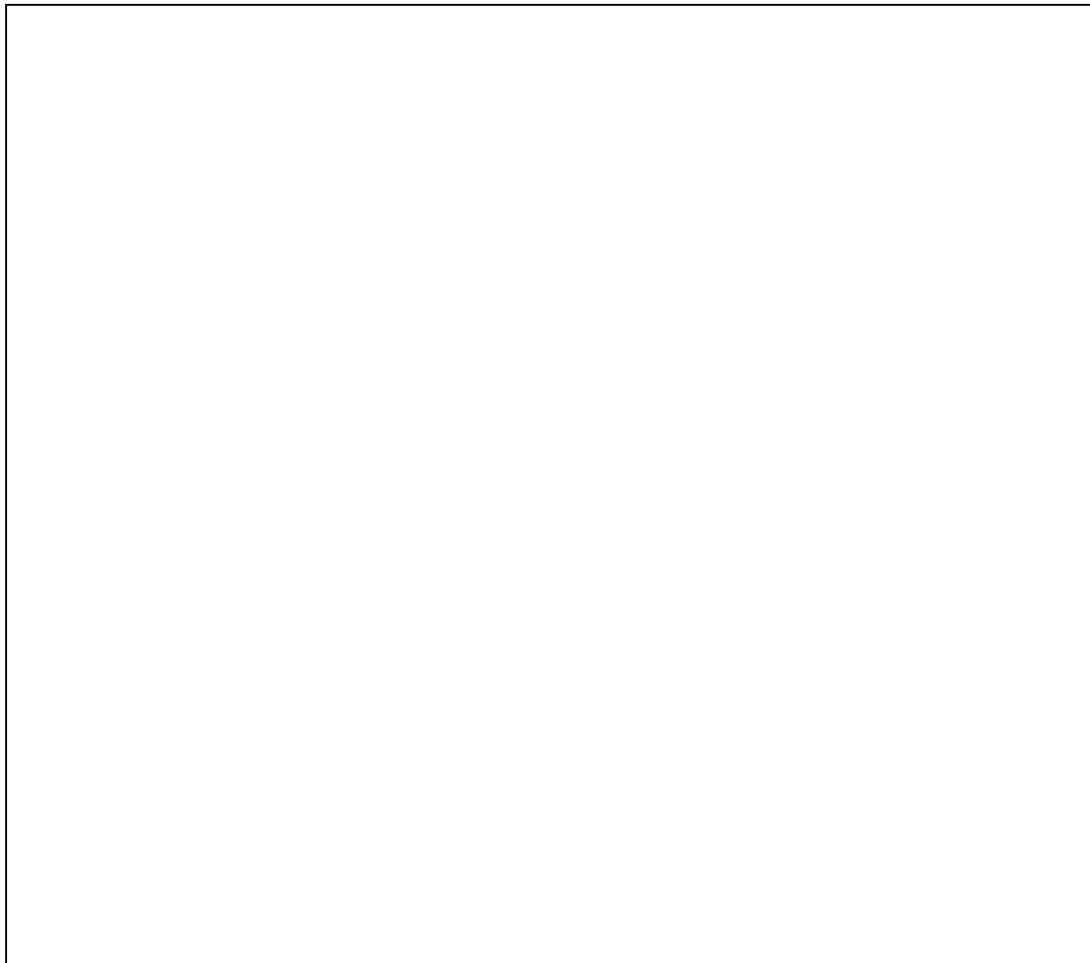
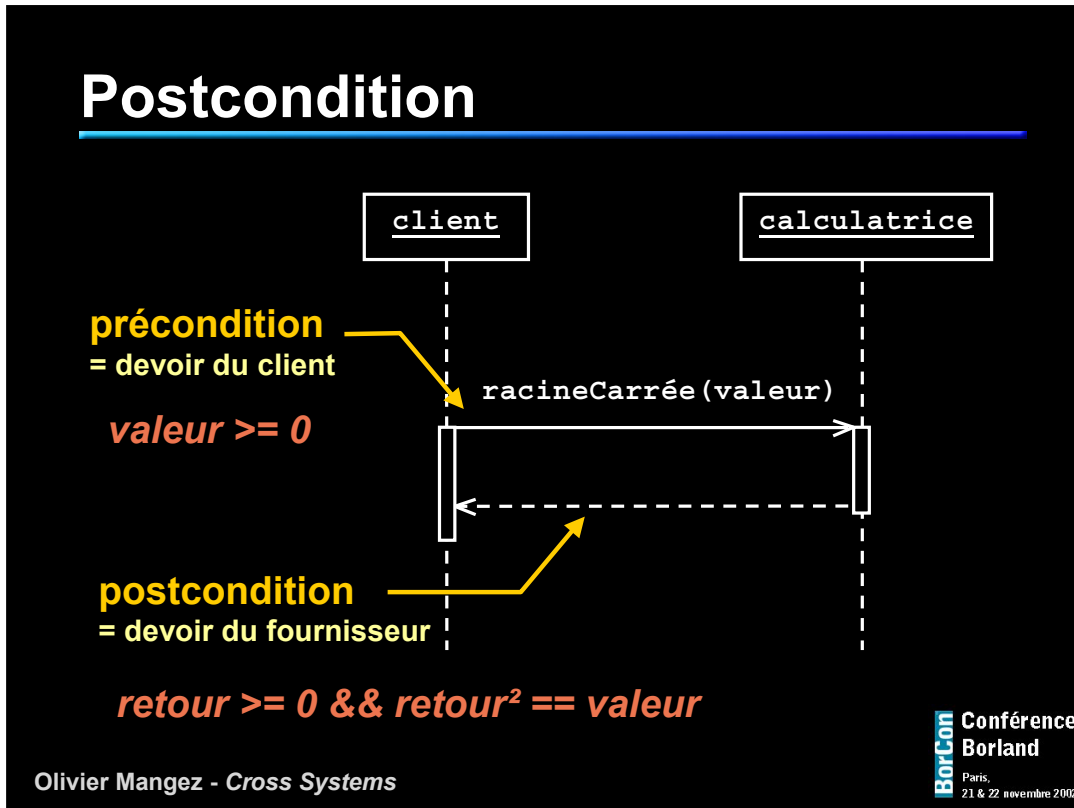
Postcondition

- **Au retour d'une méthode**
- **Devoirs du fournisseur**
- **Bénéfices du client**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Conditions vérifiées à la fin d'une méthode
 - **Devoir du fournisseur**
 - Seulement si les préconditions sont respectées
 - Violation = erreur chez le fournisseur
 - **Bénéfice du client**
- Une postcondition peut porter sur
 - La valeur de retour
 - Les attributs
- Parfois difficile à exprimer précisément



Invariant

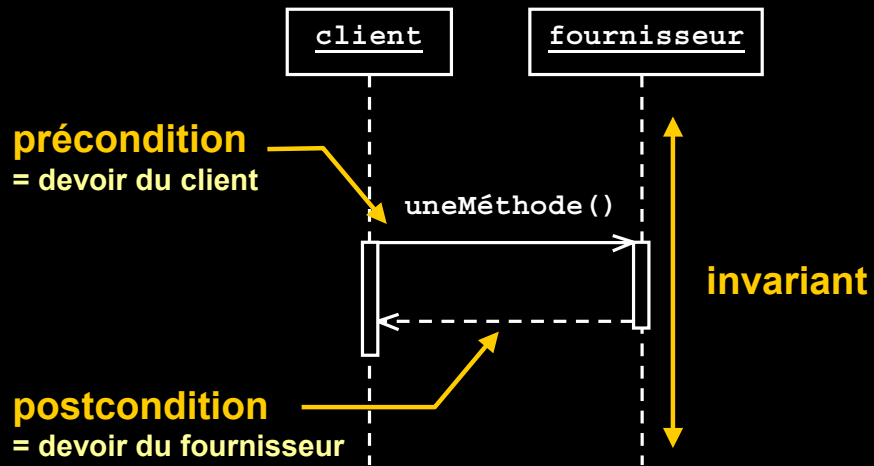
- **Tout au long de la vie de l'objet**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Condition vérifiée tout au long de la vie de l'objet
- Violation = erreur dans le fournisseur

Invariant



Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

iContract

```
public class Calculator {  
    /**  
     * @pre value >= 0.0  
     * @post return >= 0.0  
     *      && return * return = value  
     */  
    public void squareRoot(double value) {  
        ...  
    }  
}
```

Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Plan

- Assertion
- Design by contract
- **Tests unitaires**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

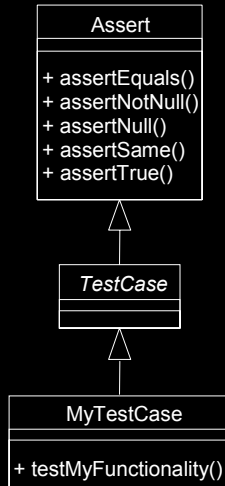
jUnit

- **Tests automatiques**
 - **Exécution**
 - **Interprétation**
- **Ne coder que les tests**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Un simple test



```

public class MyTestCase
    extends TestCase {

    public MyTestCase(String name) {
        super(name);
    }

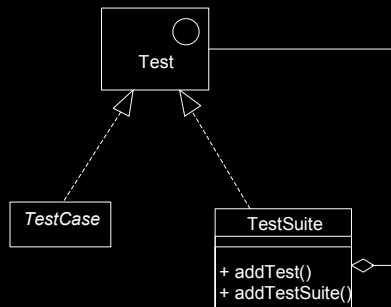
    public void testMyFunctionality() {
        ...
        assertTrue(...);
        ...
    }
}
    
```

Olivier Mangez - Cross Systems

BorCon Conférence
 Borland
 Paris,
 21 & 22 novembre 2002

- Dans le framework JUnit, il existe une classe (`junit.framework.TestCase`) qui permet d'encapsuler un test. Pour créer un test, il faut donc créer une **classe dérivée** de `TestCase` .
- A cette classe, il faut ajouter une méthode du type `public void testXXX()` dans laquelle on placera le test.
- Bien entendu, il faut indiquer les conditions dans lesquelles le test échoue. Pour cela on utilise le mécanisme d'**assertion** fournit par la classe `junit.framework.Assert` .

Un ensemble de tests



Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

- Afin de pouvoir **automatiser l'exécution** des tests, on veut pouvoir tous les exécuter en une seule opération.
- Analogie avec les répertoires :
 - TestCase = fichier
 - TestSuite = répertoire
- Pour cela, il suffit de créer une instance de `junit.framework.TestSuite` et de lui ajouter tous les tests à l'aide de la méthode `addTest()`.



Conclusion

- **DbC vs Tests unitaires**
 - **Redondants ?**
 - **Complémentaires ?**

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Références

- B. Meyer, *Object-oriented software construction*
- A. Hunt, D. Thomas, *The pragmatic programmer*

Olivier Mangez - Cross Systems

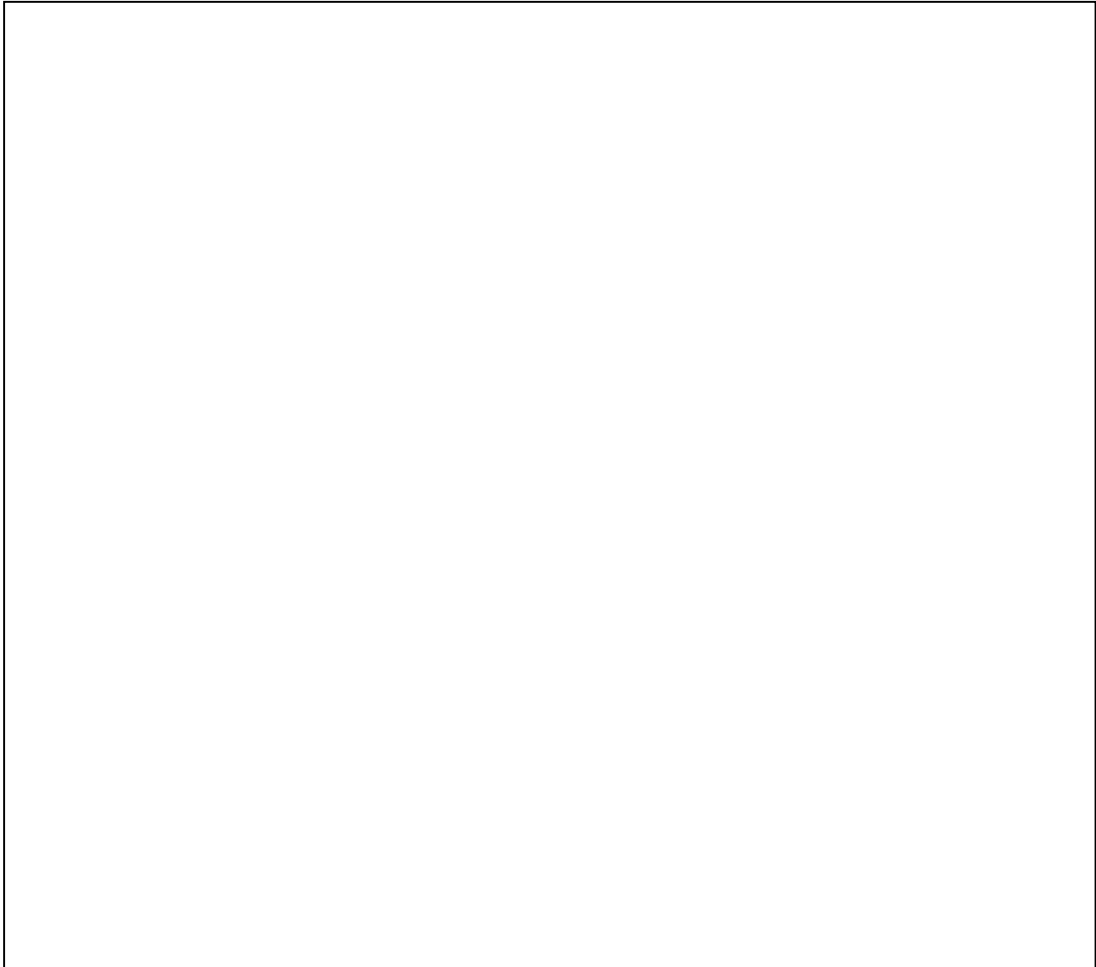
BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

Autre session

- Programmation orientée aspect
 - Vendredi, 10h45

Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002



Contact

- **Olivier Mangez :**

o.mangez@cross-systems.com

www.mangez.info

- **Cross Systems :**

13 rue Fernand Léger, 75020 Paris

Tel : 01 43 58 61 61


Fax : 01 43 58 62 43



Olivier Mangez - Cross Systems

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002

A vos questions !



Olivier Mangez - *Cross Systems*

BorCon Conférence
Borland
Paris,
21 & 22 novembre 2002