Présentation du projet INKA aux journnées RNTL 2004

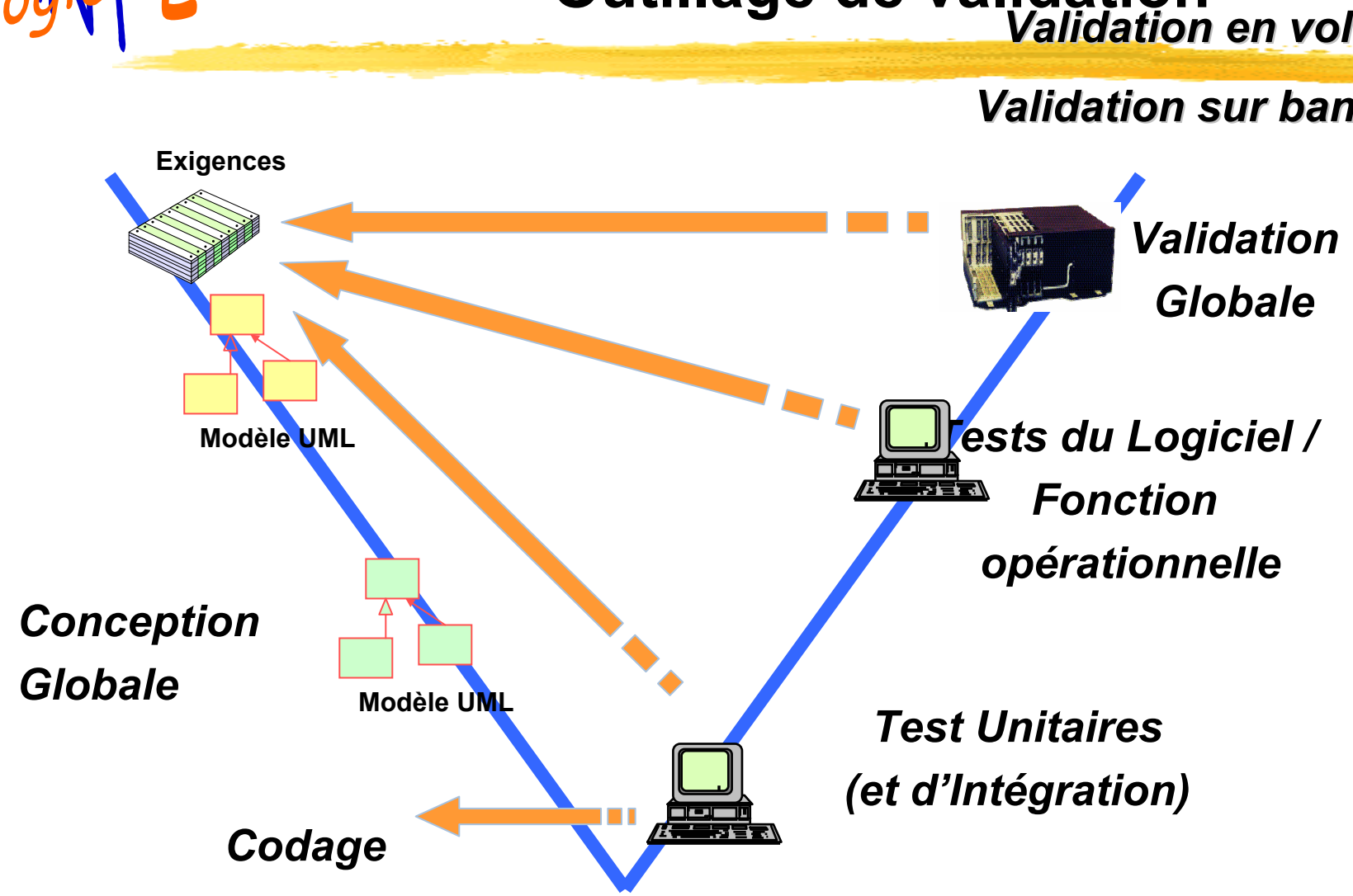
Génération automatique de cas de test structurel pour des programmes écrits en C/C++

Bernard BOTELLA

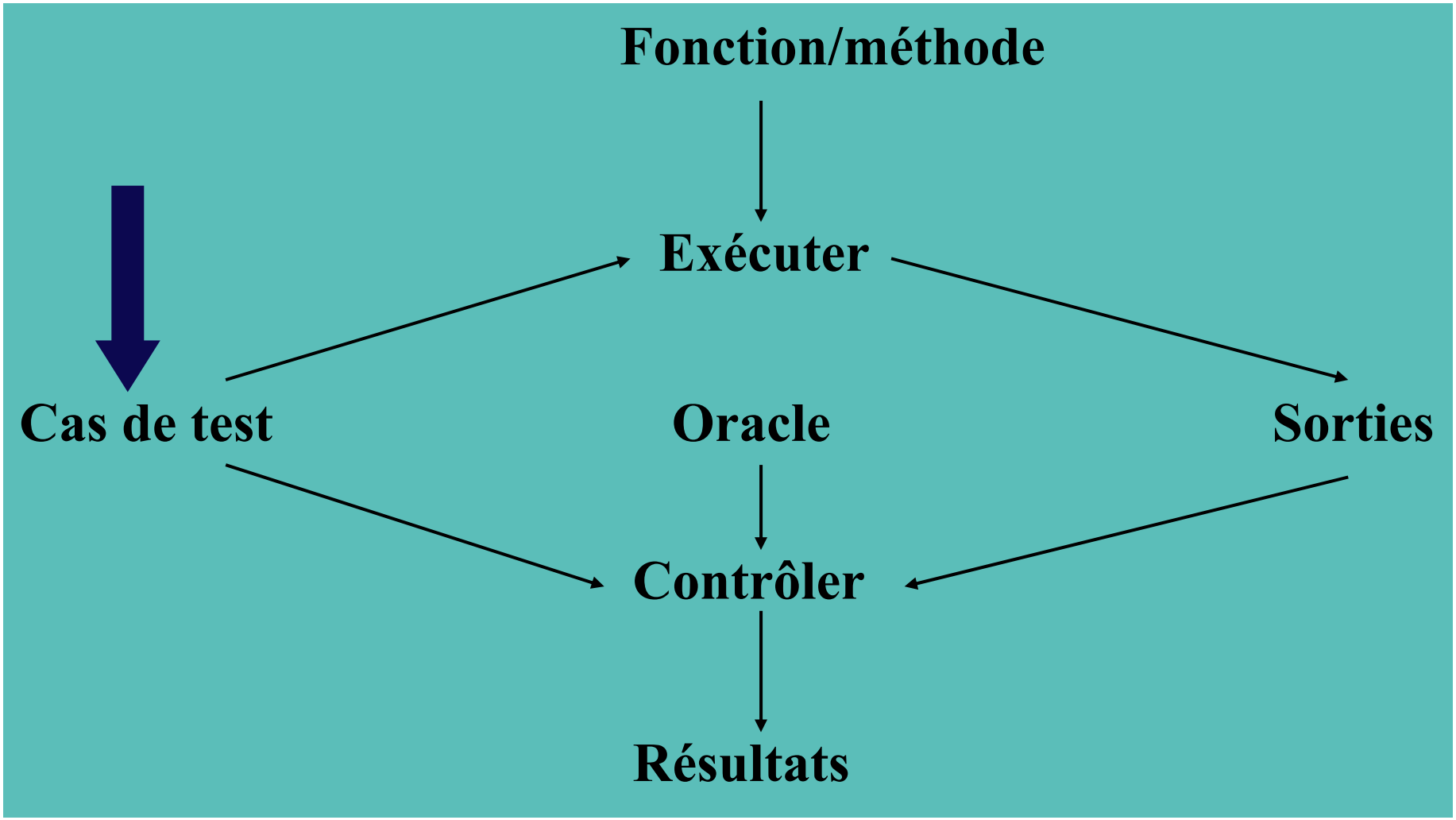
THALES SYSTEMES AEROPORTES

Bernard.Botella@fr.thalesgroup.com

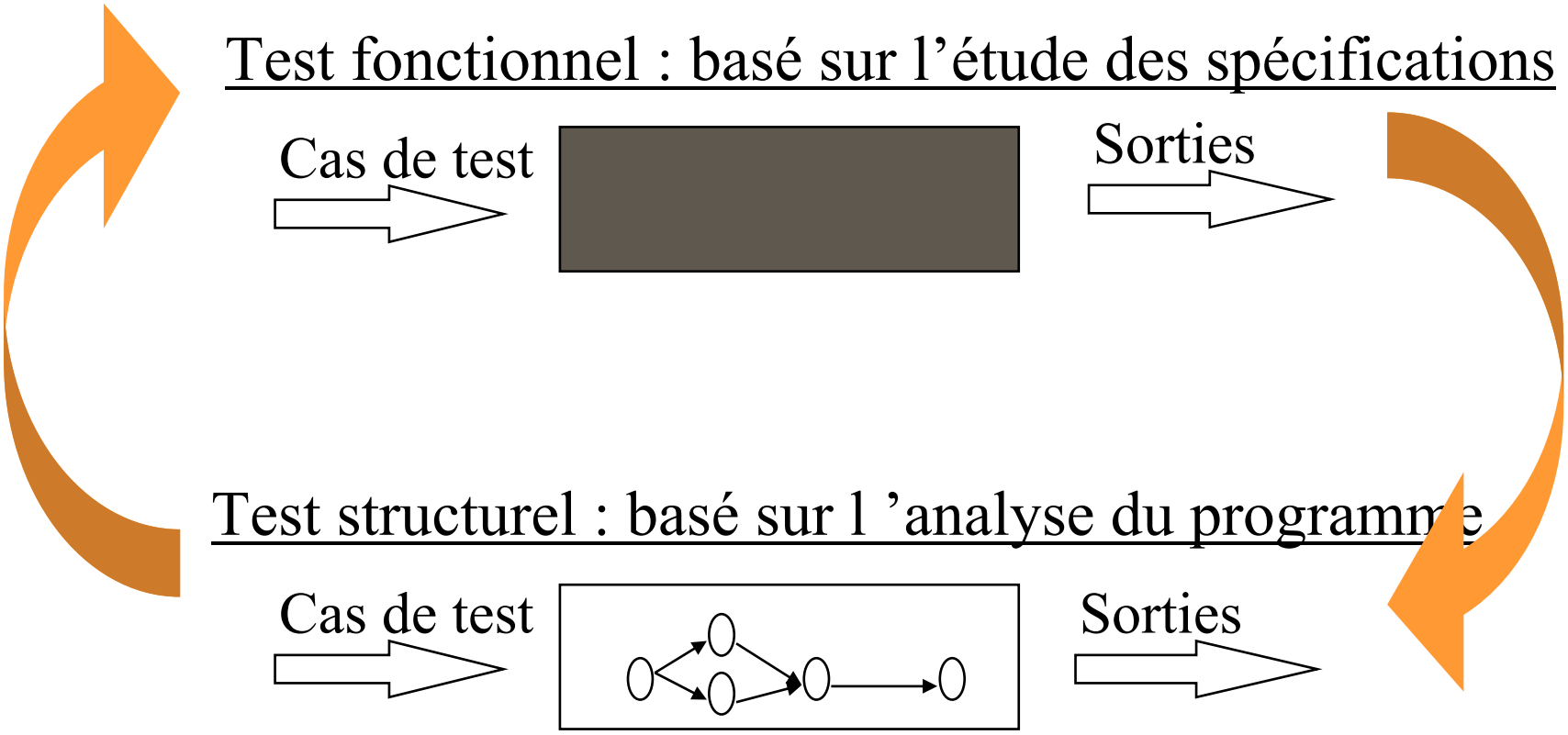
Outillage de validation



Test Unitaire



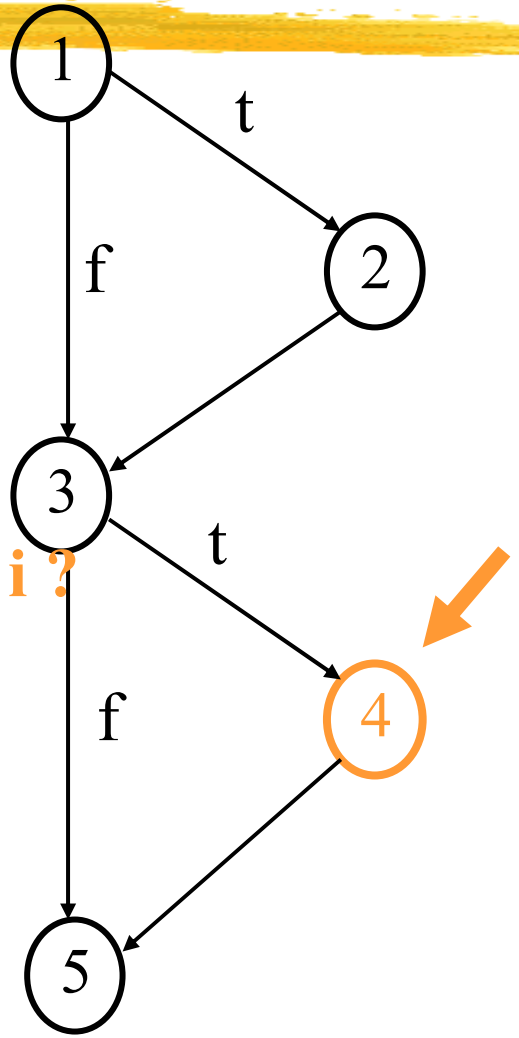
Choix des cas de test



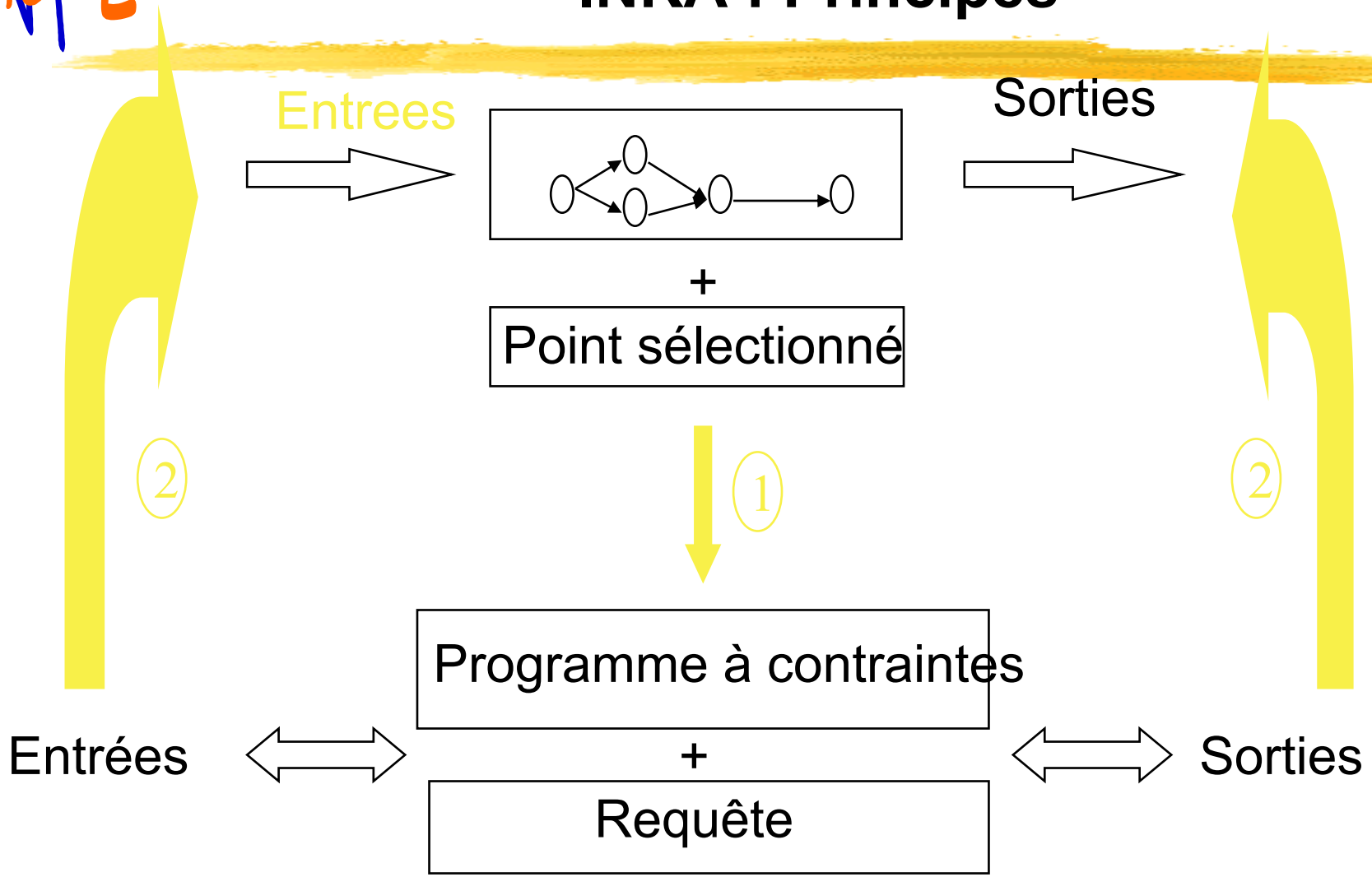
Atteindre un point du programme

```
f( int i )
{
    j := 2
    if( i ≤ 16 )
        j := j * i

    if( j > 8)
        j := 0

    return j
}
```

valeur du paramètre i ?



INKA : Principes



Le projet INKA

- **Projet RNTL (labellisé 2000) : 2001-2002**

- **Partenaires :**
- **THALES SYSTEMES AEROPORTES**
- **AXLOG,**
- **I3S (Nice) ,**
- **LIFC (Besançon) ,**
- **LSR (Grenoble)**

- **2 AXES :**
  - Industrialisation les travaux commencés chez THALES / I3S
  - Recherche sur les problèmes théoriques posés par les flottants, les structures de données dynamiques

Résultats Industrialisation : INKA V(0).1

- **LANGAGE**
- **traite le langage C**
- **les types « entiers »**
- **les pointeurs sans allocation dynamique vers des types simples**
- **les structures, les tableaux de types simples**

- **FONCTIONS PRINCIPALES**
- **visualisation de la couverture**
- **génération pour toutes_les_instructions , toutes_les_branches**

- **FONCTIONS COMPLEMENTAIRES**
- **expression de contraintes sur les données de tests**
- **gestion des bouchons**
- **intégration dans la chaîne de test**
- **interface : import/export de jeux de tests au format XML**

Résultats de recherche

- **Solutions proposées :**
  - **Solveur flottant**
  - **Utilisation de contraintes ensemblistes pour modéliser les structures de données dynamiques**

- **La réalisation d 'une version d 'INKA incorporant ces résultats est en cours.**
- **Recherche en cours pour l 'extension à C++ (prise en compte des aspects dynamiques du typage, templates, …)**

Perspectives: Test unitaire automatisé

- **Que manque-t-il ?**
  - La formalisation des spécifications
    - fournit l 'oracle pour le test structurel ou fonctionnel
    - permet l 'automatisation de la couverture « fonctionnelle »

- **Peut-on éviter des tests ?**

  projet ***DANOCOPS*** labellisé RNTL 2002

  Détection Automatique de NOn-COnformités Programme-Spécification
    - (Démarrage octobre 2004 : durée 30 mois)
    - (mêmes partenaires)

- **Confronter le modèle des spécifications et le modèle du programme**
- **Trouver automatiquement des cas de défaut**
- **Prouver la conformité**