

Test d'interopérabilité de protocoles : de la formalisation des critères d'interopérabilité à la génération des tests

Soutenance de thèse
Alexandra DESMOULIN

Equipe : DIONYSOS
Encadrant : César VIHO

IRISA UNIVERSITÉ DE RENNES I

Contexte

```

    graph TD
      A[Ecriture des normes de Protocoles réseaux (organismes de normalisation)] --> B[Implémentation des composants réseaux]
      B --> C[Test des composants de protocoles]
      C --> D[Déploiement des composants dans des réseaux réels]
  
```

IRISA UNIVERSITÉ DE RENNES I

Contexte : tests d'implémentations de protocoles

```

    graph TD
      S1([Spécification S1]) -- "Test de conformité I1-S1" --> I1[Implémentation Testée I1]
      S2([Spécification S2]) -- "Test de conformité I2-S2" --> I2[Implémentation I2]
      I1 <--> |"Test d'interopérabilité I1-I2"| I2
  
```

IRISA UNIVERSITÉ DE RENNES I

Contexte : tests d'interopérabilité de protocoles

- Test d'interopérabilité
 - Objectif: interfonctionnement des implémentations
 - Complémentaire avec le test de conformité
 - Pratiques actuelles:
 - Écriture manuelle des tests sur la base de spécifications informelles
- Objectifs de cette thèse:
 - Proposer une approche formelle pour le test d'interopérabilité
 - Préciser la notion d'interopérabilité
 - Proposer des méthodes et algorithmes pour la génération des tests

IRISA UNIVERSITÉ DE RENNES I

Plan de la soutenance

- I. Définitions préliminaires
- II. Test d'interopérabilité one-to-one
- III. Test d'interopérabilité multi-implémentations
- IV. Bilan et perspectives

IRISA UNIVERSITÉ DE RENNES I

Plan de la soutenance

- I. Préliminaires
 - i. Approche formelle
 - ii. Modèle formel
- II. Test d'interopérabilité one-to-one
- III. Test d'interopérabilité multi-implémentations
- IV. Bilan et perspectives

IRISA UNIVERSITÉ DE RENNES I

Étapes du test d'interopérabilité

1. Génération de cas de test d'interopérabilité abstraits
2. Compilation des tests abstraits en tests exécutables
3. Exécution des tests sur les implémentations à tester (IUT)

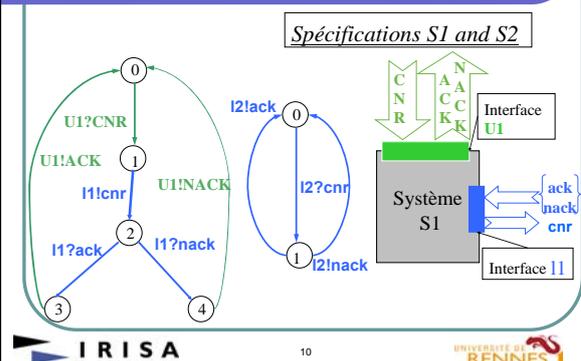
Génération des cas de tests abstraits

- I. Identification de l'architecture de test
- II. Choix de la notion d'interopérabilité
- III. Génération des cas de tests d'interopérabilité abstraits
 1. Spécifications formelles des systèmes
 2. Objectifs de test

Modèle formel

- IOLTS: Input-Output Labelled Transition System
- Utilisé pour la modélisation des spécifications et implémentations
- Différencie entrées, sorties et évènements internes

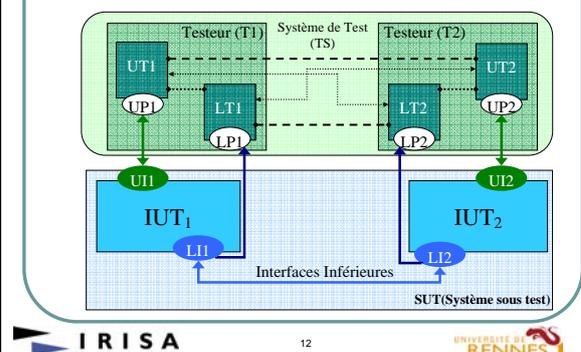
Exemple de spécifications

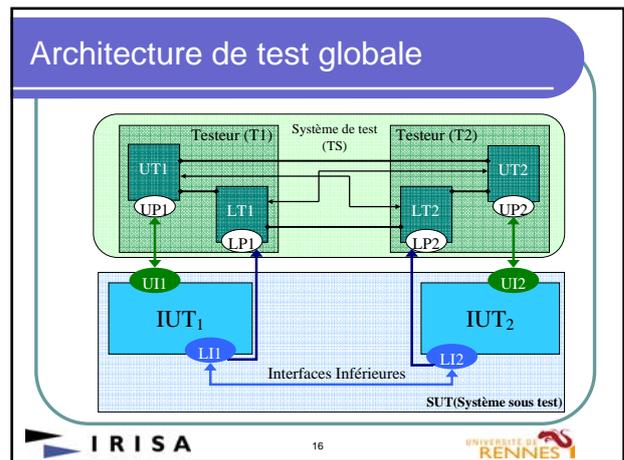
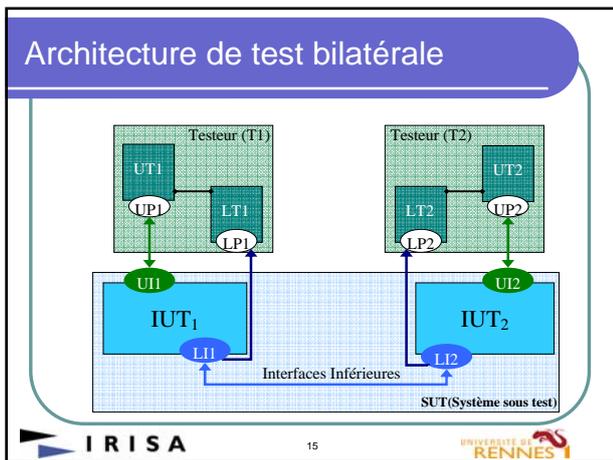
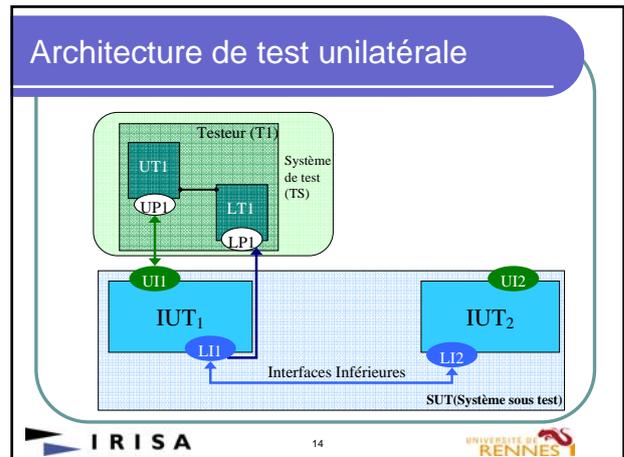
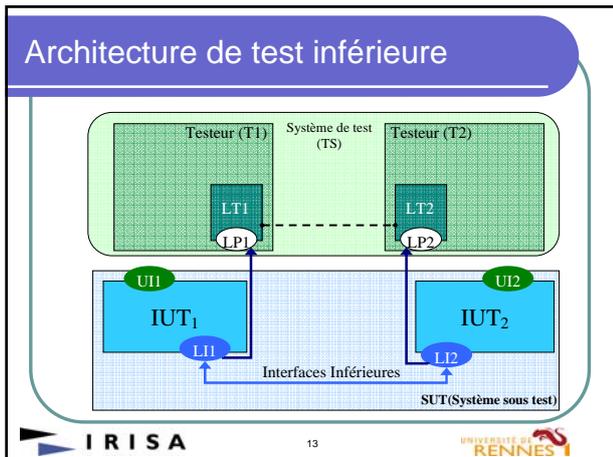


Plan de la soutenance

- I. Préliminaires
- II. Test d'interopérabilité one-to-one
 - i. Architectures de test
 - ii. Définitions formelles
 - iii. Génération de test
- III. Test d'interopérabilité multi-implémentations
- IV. Bilan et perspectives

Architectures de test d'interopérabilité





- ### Définitions formelles
- Propriétés sur les spécifications
 - Hypothèses sur les implémentations
 - Définitions formelles de la notion d'interopérabilité
- IRISA 17 UNIVERSITÉ DE RENNES I

- ### Propriétés des spécifications et hypothèses sur les implémentations
- lop-compatibilité des spécifications
 - Pour tout message envoyé par une spécification sur une interface inférieure, la réception correspondante doit être prévue dans l'autre spécification
 - Hypothèses sur les implémentations
 - > Dépendantes du type d'interface
- IRISA 18 UNIVERSITÉ DE RENNES I

Définitions formelles de la notion d'interopérabilité

- Critères d'interopérabilité : propriétés à vérifier par deux implémentations pour être considérées interopérables:
 1. Service (Propriété Pr_Serv) : une IUT doit rendre le service prévu lors de l'interaction
 2. Interaction (Propriété Pr_Int) : capacité des deux IUTs à interagir (envoi et réception de messages)

Critères d'interopérabilité

1. Vérification des sorties (et blocages):
 - Sur toutes les interfaces
 - Comparaison sorties observées par rapport aux sorties prévues dans les spécifications
2. Vérification des entrées
 - Interfaces inférieures
 - Basée sur les dépendances causales entre entrées et sorties

Critères d'interopérabilité : vérification des entrées

- Problème : le testeur ne peut pas observer la prise en compte effective d'une entrée
- Méthode de vérification :
 - calcul des sorties dépendances causales de l'entrée considérée
 - ✓ Utilise un algorithme de recherche en largeur (breadth-first search algorithm)
 - Comparaison entre les sorties observées et les sorties dépendances causales de l'entrée considérée

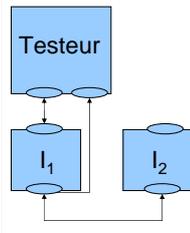
Critères d'interopérabilité

- Cas de l'accès aux deux types d'interfaces:
 - Critère d'interopérabilité globale totale iop_G
 - Critère d'interopérabilité unilatérale totale iop_U
 - Critère d'interopérabilité bilatérale totale iop_B
- Cas de l'accès à un seul type d'interfaces:
 - Critères déduits via des projections

Critère d'interopérabilité unilatérale iop_U

- C1: après une trace de S_1 observée pendant l'interaction des IUTs, toutes les sorties observées sur les interfaces de I_1 doivent être prévues dans S_1
- C2: les entrées correspondant à des messages envoyés par I_2 doivent être effectivement reçus par I_1

Architecture de test unilatérale totale



Critère d'interopérabilité bilatérale iop_B

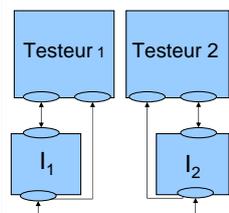
Critère défini par :

➢ $I_1 iop_U I_2$

Et

➢ $I_2 iop_U I_1$

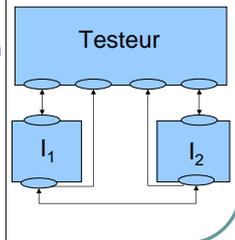
Architecture de test bilatérale totale



Critère d'interopérabilité globale iop_G

- **C1**: après une trace de l'interaction des spécifications, toutes les **sorties** observées durant l'interaction des IUTs doivent être **prévues dans l'interaction des spécifications**

Architecture de test globale totale



- **C2**: les **entrées** correspondant à des messages envoyés par l'une des IUTs doivent être **effectivement reçues** par l'autre

Critères d'interopérabilité : propriétés

- ✓ $iop_B \equiv iop_G$: même capacité de détection de la non-interopérabilité
- ✓ iop_G basé sur $S_1 \parallel_A S_2$, pas iop_B
 - ⇒ iop_B évite l'explosion combinatoire du nombre d'états rencontrée lors d'une approche globale

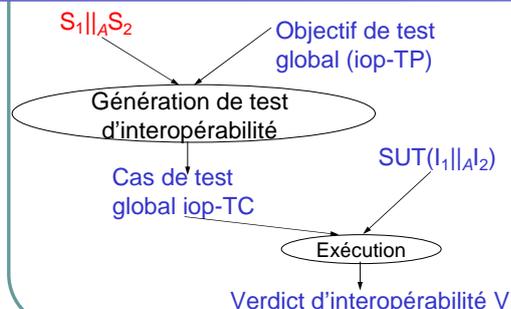
Principes d'une méthode de génération de tests

- Basée sur un **critère** d'interopérabilité
- **Entrées** :
 - **Spécifications** des systèmes à tester
 - un **objectif de test** d'interopérabilité (propriété à tester)
- **Sortie** :
 - **cas de test** d'interopérabilité (unilatéral, bilatéral ou global)

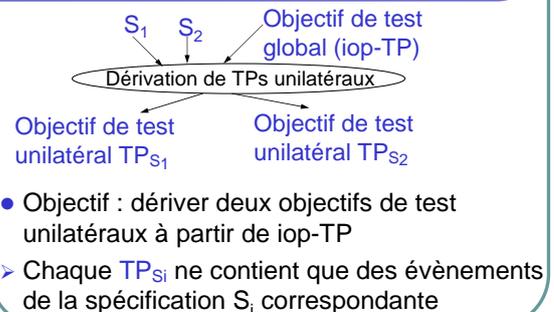
Génération de test

- Architecture de test totale ⇒ deux approches:
 1. Approche globale
 2. Approche bilatérale

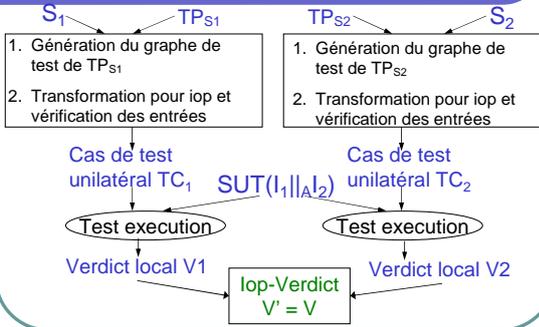
Génération de test : approche globale



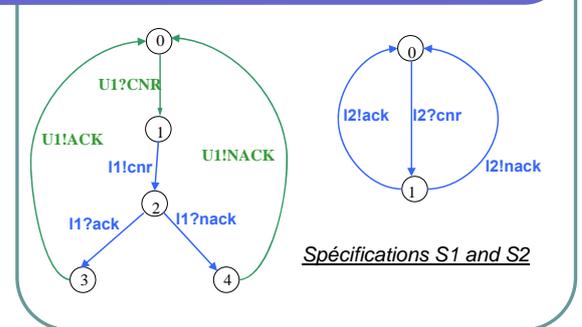
Génération de test : approche bilatérale (étape 1)



Génération de test : approche bilatérale (étape 2)

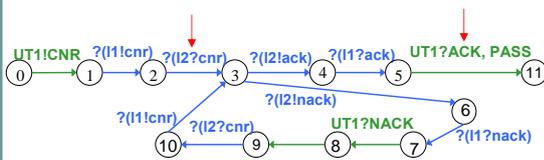


Application des méthodes de génération de tests d'interopérabilité



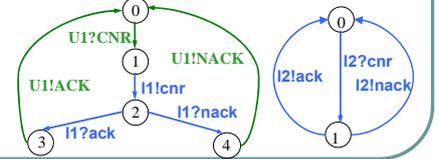
Approche globale : génération du cas de test

Objectif de test global: $iop-TP = I2?cnr.U1!ACK$

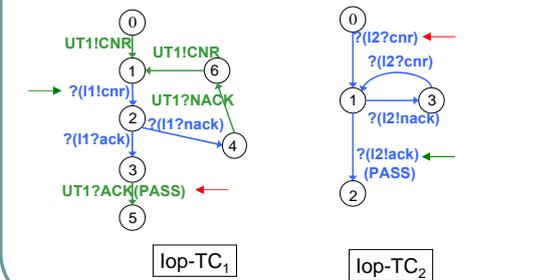


Approche bilatérale : dérivation des objectifs de test unilatéraux

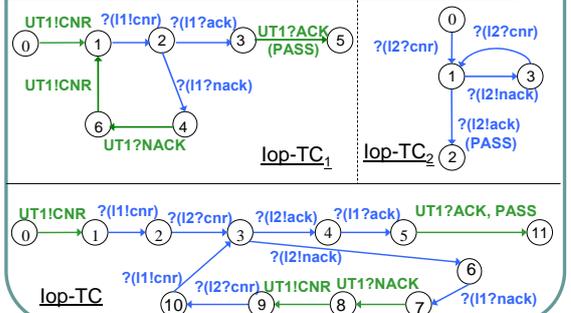
- **Objectif de test global:**
 $iop-TP = I2?cnr.U1!ACK$
- **Objectifs de test unilatéraux :**
 - > $lop-TP1 = I1!cnr.U1!ACK$
 - > $lop-TP2 = I2?cnr.I2!ack$



Approche bilatérale : génération des cas de test



Comparaison des cas de test



Plan de la soutenance

- I. Préliminaires
- II. Test d'interopérabilité one-to-one
- III. Test d'interopérabilité multi-implémentations
 - I. Configuration des testeurs et des implémentations
 - II. Définitions formelles
 - III. Génération de test
- IV. Bilan et perspectives

Test d'interopérabilité multi-implémentations

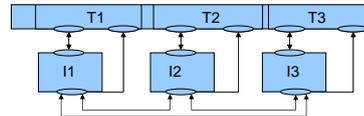
- Problèmes du passage au test d'interopérabilité de N implémentations
⇒ prise en compte de la **topologie**
- Autres étapes
 - 1) Déterminer l'architecture de test
 - 2) Vérifier que les spécifications permettent l'interaction
 - 3) Choisir le critère d'interopérabilité
 - 4) Générer les tests

Topologie et test d'interopérabilité

- Problèmes :
 - Choix de la ou des topologies pour le test
 - Modélisation des topologies utilisées dans les critères d'interopérabilité et dans la génération de test
- Choix d'une topologie ⇒ trouver la **topologie minimale** pour le test
- Modélisation : représentation matricielle

Architectures de test d'interopérabilité multi-implémentations

- Architectures de test multiparties :
 - Architecture de test multi-iop globale
 - Architecture de test multi-iop unilatérale
 - Architecture de test multi-iop multi-unilatérale



Approche formelle pour le test d'interopérabilité multi-implémentations

- l'op-compatibilité des spécifications
- Définition de **critères d'interopérabilité multi-implémentations**, intégrant la topologie considérée
- ➔ Objectif: générer des tests d'interopérabilité à partir de ces critères d'interopérabilité

Plan de la soutenance

- I. Préliminaires
- II. Test d'interopérabilité one-to-one
- III. Test d'interopérabilité multi-implémentations
- IV. Bilan et perspectives

Bilan

- Contexte one-to-one:
 - Définitions formelles de la notion d'interopérabilité : critères d'interopérabilité
 - Méthode bilatérale de génération de tests
 - ✓ Évitant l'explosion combinatoire du nombre d'états
 - ✓ Avec vérification des entrées (dépendances causales)
- Contexte multi-implémentations :
 - Prise en compte de la topologie
 - Généralisation des critères d'interopérabilité

Bibliographie

- Alexandra DESMOULIN, César VIHO. **Quiescence Management improves Interoperability Testing**. *Testcom 2005*, Montréal, Canada, juin 2005.
- Alexandra DESMOULIN, César VIHO. **Formalizing Interoperability testing: Quiescence Management and Test Generation**. *FORTE 2005*, Taipei, Taiwan, octobre 2005.
- Alexandra DESMOULIN, César VIHO. **A New Method for Interoperability Test generation**. *Testcom/Fates 2007*. Tallinn, Estonie, Juin 2007.
- Alexandra DESMOULIN, César VIHO. **Automatic Interoperability Test Case Generation based on Formal Definitions**. *FMICS 2007*. Berlin, Allemagne, Juillet 2007.

Perspectives

- Contexte one-to-one:
 - Approche distribuée pour le test d'interopérabilité
 - Étude des étapes de la phase d'exécution des tests d'interopérabilité
- Contexte multi-implémentations
 - Proposer des algorithmes pour la génération de tests d'interopérabilité

Merci de votre attention!