

# Formalisations pour les compositions de services

---

**Aurélie Hurault**

6 Juillet 2018

*Soutenance en vue de l'obtention de l'Habilitation à Diriger les Recherches*

---

IRIT - Équipe ACADIE

Toulouse INP - ENSEEIHT

## JURY

BERNADETTE CHARRON-BOST

Directrice de recherche – CNRS LIX

CATHERINE DUBOIS

Professeur des universités – ENSIIE

PASCAL POIZAT

Professeur des universités – Université Paris Nanterre

SANDRINE BLAZY

Professeur des universités – Université Rennes 1

BERTRAND MEYER

Professeur – Politecnico di Milano et Innopolis University

MARC SHAPIRO

Directeur de recherche – INRIA

PHILIPPE QUÉINNEC

Professeur des universités – Université de Toulouse

Rapporteur

Rapporteur

Rapporteur

Examinatrice

Examineur

Examineur

Correspondant

## Positionnement

---

Il est 10h00 et ce matin certains :

- se sont réveillés (de bonne heure) grâce à leur smartphone ;
- ont consulté la météo toulousaine sur leur téléphone ;
- ont ouvert et démarré leur voiture grâce à une carte ;
- ont suivi le GPS jusqu'à l'aéroport ;
- ont eu leurs plaques d'immatriculation scannées pour entrer sur le parking de l'aéroport ;
- ont consulté les panneaux d'affichage pour connaître la porte d'embarquement ;
- ont payé leur petit-déjeuner à une caisse automatique ;
- ont fait valider leur carte d'embarquement sur leur téléphone ;
- ont pris un avion ;
- ont consulté le compteur du taxi ;
- sont passés à l'accueil chercher un badge d'accès pour les tourniquets ;
- ont les yeux rivés sur ces transparents projetés grâce au système multimédia.

Il est 10h00 et ce matin certains :

- se sont réveillés (de bonne heure) grâce à leur smartphone ;
- ont consulté la météo toulousaine sur leur téléphone ;
- ont ouvert et démarré leur voiture grâce à une carte ;
- ont suivi le GPS jusqu'à l'aéroport ;
- ont eu leurs plaques d'immatriculation scannées pour entrer sur le parking de l'aéroport ;
- ont consulté les panneaux d'affichage pour connaître la porte d'embarquement ;
- ont payé leur petit-déjeuner à une caisse automatique ;
- ont fait valider leur carte d'embarquement sur leur téléphone ;
- ont pris un avion ;
- ont consulté le compteur du taxi ;
- sont passés à l'accueil chercher un badge d'accès pour les tourniquets ;
- ont les yeux rivés sur ces transparents projetés grâce au système multimédia.

Domaine critique : normes de certification

Il est 10h00 et ce matin certains :

- se sont réveillés (de bonne heure) grâce à leur smartphone ;
- ont consulté la météo toulousaine sur leur téléphone ;
- ont ouvert et démarré leur voiture grâce à une carte ;
- ont suivi le GPS jusqu'à l'aéroport ;
- ont eu leurs plaques d'immatriculation scannées pour entrer sur le parking de l'aéroport ;
- ont consulté les panneaux d'affichage pour connaître la porte d'embarquement ;
- ont payé leur petit-déjeuner à une caisse automatique ;
- ont fait valider leur carte d'embarquement sur leur téléphone ;
- ont pris un avion ;
- ont consulté le compteur du taxi ;
- sont passés à l'accueil chercher un badge d'accès pour les tourniquets ;
- ont les yeux rivés sur ces transparents projetés grâce au système multimédia.

Domaine non critique : aucune garantie

**Besoin de garantie quelle que soit la criticité de l'application.**

## Modélisation formelle

- Des éléments manipulés
- Du problème à résoudre

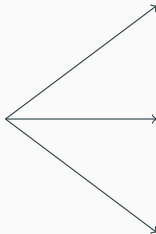
## Outils et méthodes prouvés

Fournir à l'utilisateur non expert des méthodes formelles des outils

- Faciles d'utilisation
- Automatisés
- Prouvés

## Domaine d'application

- Composition de services



Météo France



Weather Pro



La chaîne météo

## Diversité des services météorologiques

- Même fonctionnalité
- Probablement une API et différentes interactions avec l'environnement
- Qualités de service différentes

## Modélisation des services

- Signature : nom et types des entrées et des sorties
- Sémantique : fonctionnalité réalisée par le service
- Comportement : interaction avec l'environnement
- Qualité de service



# La composition de services



Stations d'observation météorologiques



Météo France

module d'assimilation de données



Serveur

Site Météo France



Supercalculateur

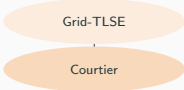
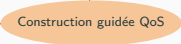
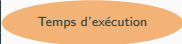
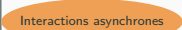

ARPEGE

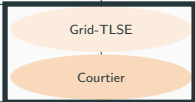
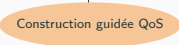
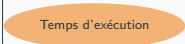


ALADIN

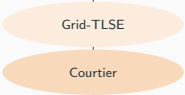
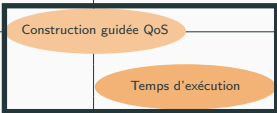
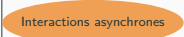

AROME

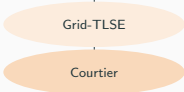
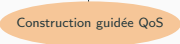


## Problématiques liées à la composition de services

- Découverte avec ou sans plan
- Sélection
- Validation
- Adaptation

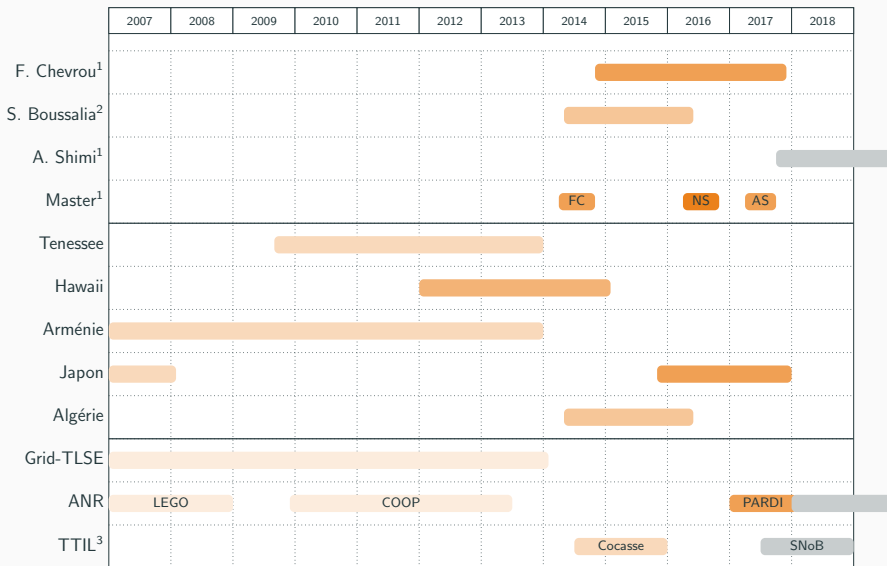
	Sémantique	Signature	Qualité de service	Comportement
Découverte	 <p>Grid-TLSE</p> <p>Courtier</p>	 <p>Construction guidée QoS</p>		
Sélection			 <p>Temps d'exécution</p>	
Validation				 <p>Interactions asynchrones</p>
Adaptation				 <p>Priorités applicatives</p>

	Sémantique	Signature	Qualité de service	Comportement
Découverte	 <p>Grid-TLSE Courtier</p>		 <p>Construction guidée QoS</p>	
Sélection			 <p>Temps d'exécution</p>	
Validation				 <p>Interactions asynchrones</p>
Adaptation				 <p>Priorités applicatives</p>

	Sémantique	Signature	Qualité de service	Comportement
Découverte	 <p>Grid-TLSE</p> <p>Courtier</p>			
Sélection			 <p>Construction guidée QoS</p> <p>Temps d'exécution</p>	
Validation				 <p>Interactions asynchrones</p>
Adaptation				 <p>Priorités applicatives</p>

	Sémantique	Signature	Qualité de service	Comportement
Découverte	 <p>Grid-TLSE Courtier</p>	 <p>Construction guidée QoS</p>		
Sélection			 <p>Temps d'exécution</p>	
Validation				 <p>Interactions asynchrones Priorités applicatives</p>
Adaptation				

# Encadrements, collaborations et projets



<sup>1</sup> Co-encadrement Philippe Quéinnec

<sup>2</sup> Co-encadrement Allaoua Chaoui

<sup>3</sup> Toulouse Tech InterLabs

	Sémantique	Signature	Qualité de service	Comportement
Découverte	<div>Grid-TLSE</div> <div>Courtier</div>	<div>Construction guidée QoS</div>		
Sélection			<div>Temps d'exécution</div>	
Validation				<div>Interactions asynchrones</div>
Adaptation				<div>Priorités applicatives</div>



**Modélisation formelle :  
Découverte et sélection des  
compositions**

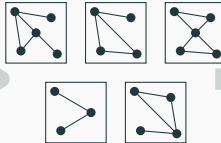
---

Librairies de services de calcul



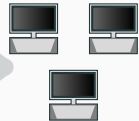
Découverte

Application complexe  
Compositions de services



Sélection

Environnement  
d'exécution



## Problèmes

- Comment construire la/les composition(s) de services à partir des services ?
- Comment choisir la meilleure composition pour un environnement donné ?

## Particularité

- Domaine d'application : algèbre linéaire

### Objectif

- Découverte d'une composition, sans plan connu, pour obtenir une fonctionnalité souhaitée

### Contraintes

- Description des services et de la fonctionnalité rendue réalisable par les experts en algèbre linéaire
- Automatisation
- Correction et complétude

### La bonne idée

- Tirer profit des particularités de l'algèbre linéaire

## Solution

- Domaine d'application
  - Signature hétérogène avec sous-typage
  - Équations donnant la sémantique des opérateurs (TAA)
- Services et requête : termes sur une signature hétérogène avec sous-typage
- Algorithme : unification équationnelle
  - Trouver  $\sigma$  tel que :  $\sigma(t_1) =_E \sigma(t_2)$

## Exemple

- Domaine  
 $\Sigma = \{0, I : \rightarrow Matrix, +, * : Matrix \times Matrix \rightarrow Matrix\}$   
 $\mathcal{E} = \{x, y, z : Matrix : x * I = x, x * 0 = 0, x + 0 = x, x + y = y + x, x + (y + z) = (x + y) + z, x * (y * z) = (x * y) * z, \dots\}$
- Services  
 $dgemm(x, y, z : Matrix) = x * y + z : Matrix, daxpy(x, y : Matrix) = x + y : Matrix$
- Requête  
 $r = \{a, b, c, d : Matrix\} : a * b + c * d$
- Compositions solutions  
 $daxpy(dgemm(a, b, 0), dgemm(c, d, 0)), dgemm(a, b, dgemm(c, d, 0)), \dots$

+

- S'étend à tout domaine applicatif pouvant être décrit par une signature hétérogène avec sous-typage
- Opérationnel : intégration dans des intergiciels de grilles ([J. Grid Comput'13, J. Supercomp'13, VECPAR'10, VECPAR'08])

—

- Domaines applicatifs restreints
- Non efficace

### Objectif

Sélectionner la meilleure (temps d'exécution) composition parmi celles proposées par le courtier

### Contraintes

- Automatisation
- Prise en compte de toute architecture matérielle (multicœur, parallèle, ...)
- Solution proche de l'optimale

### La bonne idée

- Ne pas construire à la main un modèle du temps d'exécution (modèle de la composition de service et de l'environnement d'exécution),
- mais laisser à l'apprentissage automatique le soin de le faire.

## Modélisation

- Composition de service : espace de caractéristiques composé de
  - nombre d'appels aux services
  - tailles des matrices
  - complexité théorique des services
- Environnement d'exécution : données d'entraînement
- Algorithme : apprentissage automatique

## Méthodologie

- Tests de plusieurs espaces caractéristiques
- Tests d'une grande variété d'algorithmes d'apprentissage automatique
- Tests de plusieurs bibliothèques d'algèbre linéaire (ATLAS et OpenBlas)
- Comparaison avec l'existant : composition sélectionnée meilleure ou équivalente à la solution choisie par Octave

+

- Donne de bons résultats (régression linéaire)
- Apprentissage réalisé sur matrices de petites tailles
- S'adapte à toute architecture (sous réserve de rejouer la phase d'entraînement)

—

- Vérification empirique, pas de garantie formelle
- Spécifique au domaine d'application



## Modélisations des services / composition de services

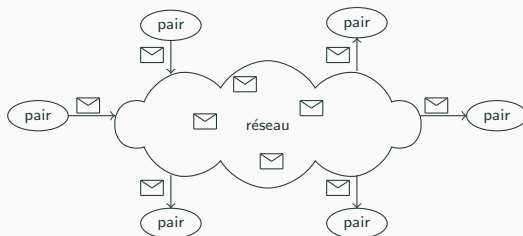
- **Découverte** : Termes sur une signature hétérogène avec sous-typage
- **Sélection** : Agrégat de valeurs numériques

## Algorithmes

- **Découverte** : Unification équationnelle
- **Sélection** : Apprentissage automatique

## **Outils et méthodes prouvés : Vérification des compositions**

---



### Domaine d'application

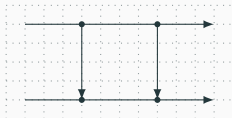
- Système réparti communiquant par messages
- Communication asynchrone

### Objectifs

- Outil automatique et prouvé pour la vérification d'une propriété sur une composition de services
- Formalisation des interactions asynchrones

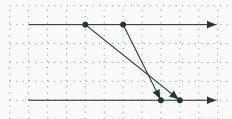
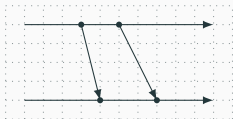
## Communication synchrone

- Rendez-vous entre émission et réception



## Communication asynchrone

- Pas de synchronisation entre émission et réception



## Modèles de communication

- Contraintes sur l'ordre des délivrances

## Objectifs

- Recensement de la diversité des modèles
- Formalisations avec différents niveaux d'abstraction
- Comparaisons

## Diversité des multiplicités

- Point-à-point
- Multicast : 1 vers  $N$
- Mergecast :  $N$  vers 1

## Diversité des contraintes de délivrance

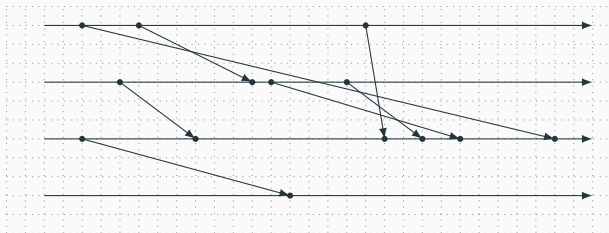
- Contraintes de délivrance applicatives
- Contraintes de délivrance génériques

## Diversité des multiplicités

- Point-à-point
- Multicast : 1 vers  $N$
- Mergecast :  $N$  vers 1

## Diversité des contraintes de délivrance

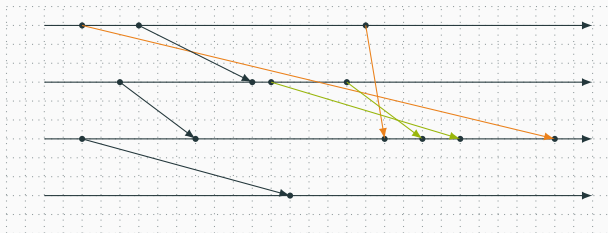
- Contraintes de délivrance applicatives
- Contraintes de délivrance génériques



## Contraintes de délivrance génériques

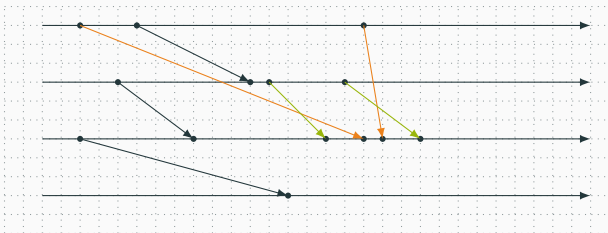
- Asynchrone pur ✓
- $FIFO_{1-1}$
- Causal
- $FIFO_{n-1}$
- $FIFO_{1-n}$
- $FIFO_{n-n}$
- RSC





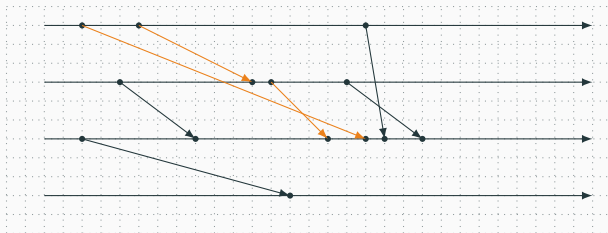
## Contraintes de délivrance génériques

- Asynchrone pur ✓
- $FIFO_{1-1}$  ✗
- Causal
- $FIFO_{n-1}$
- $FIFO_{1-n}$
- $FIFO_{n-n}$
- RSC



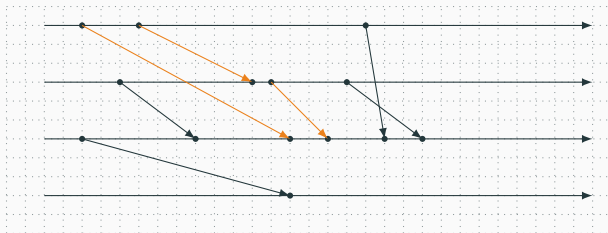
## Contraintes de délivrance génériques

- Asynchrone pur ✓
- $FIFO_{1-1}$  ✓
- Causal
- $FIFO_{n-1}$
- $FIFO_{1-n}$
- $FIFO_{n-n}$
- RSC



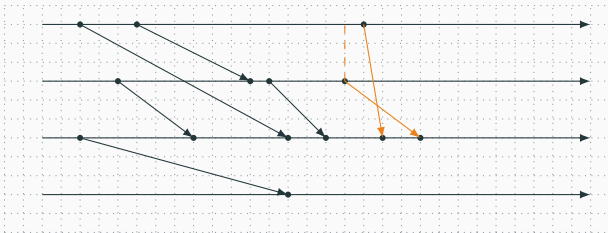
## Contraintes de délivrance génériques

- |                  |   |                |       |
|------------------|---|----------------|-------|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | • RSC |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ |       |
| • Causal         | ✗ | • $FIFO_{n-n}$ |       |



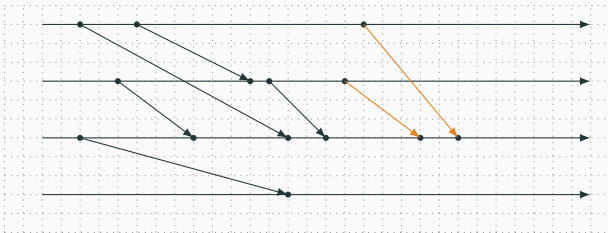
## Contraintes de délivrance génériques

- Asynchrone pur ✓
- $FIFO_{1-1}$  ✓
- Causal ✓
- $FIFO_{n-1}$
- $FIFO_{1-n}$
- $FIFO_{n-n}$
- RSC



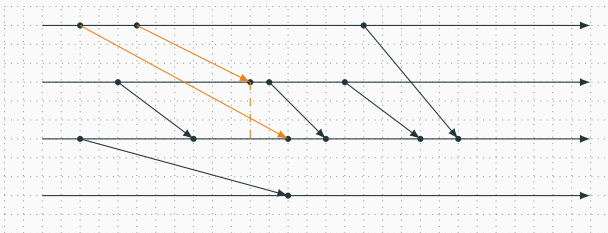
## Contraintes de délivrance génériques

- |                  |   |                |   |       |
|------------------|---|----------------|---|-------|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | × | • RSC |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ |   |       |
| • Causal         | ✓ | • $FIFO_{n-n}$ |   |       |



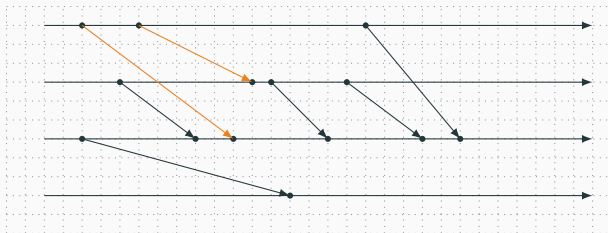
## Contraintes de délivrance génériques

- Asynchrone pur ✓
- $FIFO_{1-1}$  ✓
- Causal ✓
- $FIFO_{n-1}$  ✓
- $FIFO_{1-n}$
- $FIFO_{n-n}$
- RSC



## Contraintes de délivrance génériques

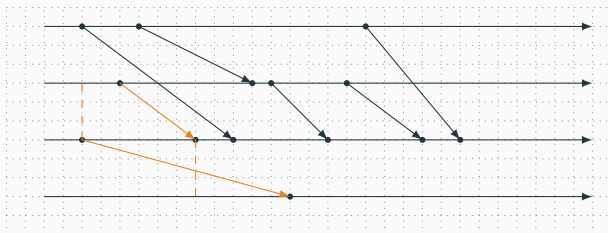
- |                  |   |                |   |       |
|------------------|---|----------------|---|-------|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | ✓ | • RSC |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ | ✗ |       |
| • Causal         | ✓ | • $FIFO_{n-n}$ |   |       |



## Contraintes de délivrance génériques

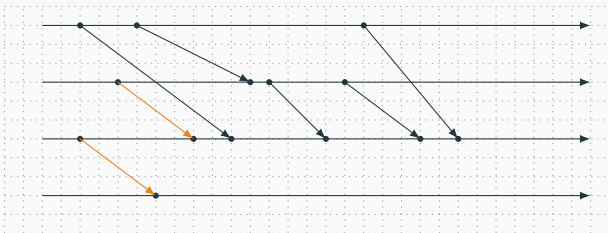
- |                  |   |                |   |       |
|------------------|---|----------------|---|-------|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | ✓ | • RSC |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ | ✓ |       |
| • Causal         | ✓ | • $FIFO_{n-n}$ |   |       |





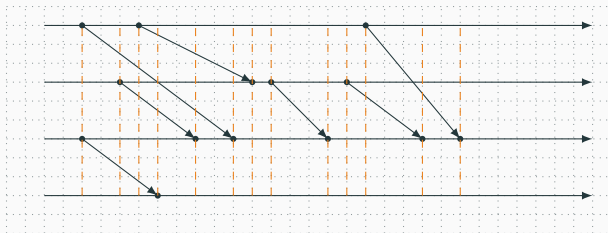
## Contraintes de délivrance génériques

- |                  |   |                |   |       |
|------------------|---|----------------|---|-------|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | ✓ | • RSC |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ | ✓ |       |
| • Causal         | ✓ | • $FIFO_{n-n}$ | ✗ |       |



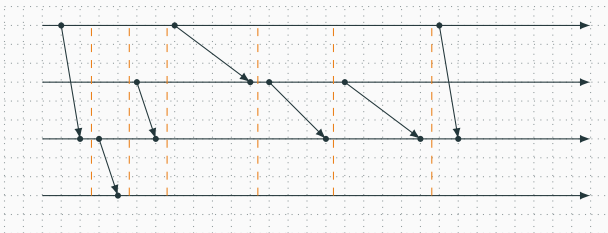
## Contraintes de délivrance génériques

- |                  |   |                |   |       |
|------------------|---|----------------|---|-------|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | ✓ | • RSC |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ | ✓ |       |
| • Causal         | ✓ | • $FIFO_{n-n}$ | ✓ |       |



## Contraintes de délivrance génériques

- |                  |   |                |   |       |   |
|------------------|---|----------------|---|-------|---|
| • Asynchrone pur | ✓ | • $FIFO_{n-1}$ | ✓ | • RSC | × |
| • $FIFO_{1-1}$   | ✓ | • $FIFO_{1-n}$ | ✓ |       |   |
| • Causal         | ✓ | • $FIFO_{n-n}$ | ✓ |       |   |



## Contraintes de délivrance génériques

- |                       |   |                       |   |       |   |
|-----------------------|---|-----------------------|---|-------|---|
| • Asynchrone pur      | ✓ | • FIFO <sub>n-1</sub> | ✓ | • RSC | ✓ |
| • FIFO <sub>1-1</sub> | ✓ | • FIFO <sub>1-n</sub> | ✓ |       |   |
| • Causal              | ✓ | • FIFO <sub>n-n</sub> | ✓ |       |   |

## Formalisation axiomatique

- Exécutions et calculs distribués formalisés comme séquences d'événements
  - Réception sur un pair  $p$  d'un message  $m$  :  $r_p(m)$
  - Émission sur un pair  $p$  d'un message  $m$  :  $s_p(m)$
  - Événement interne :  $\tau$
- Formalisation des modèles : contraintes sur l'ordre des événements
  - Trois ordres : local, causal, global
  - $\text{FIFO}_{1-1}$  :  $s_{p_a}(m_1) \prec s_{p_a}(m_2) \Rightarrow \neg(r_{p_b}(m_2) \prec r_{p_b}(m_1))$
- Formalisation similaire en multicast

## Formalisations opérationnelles

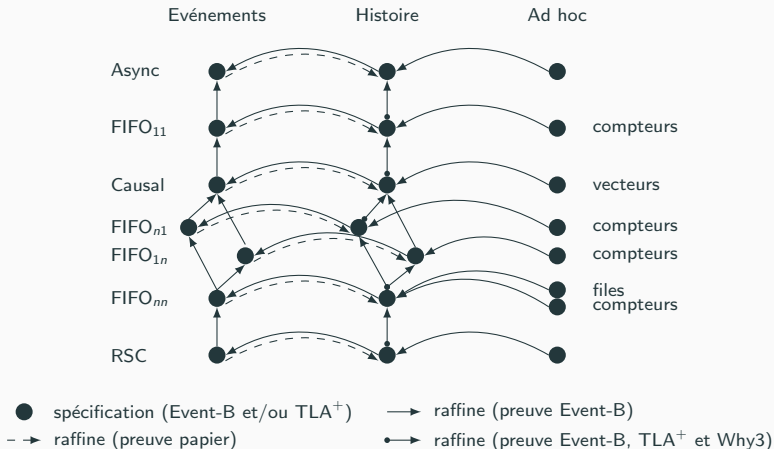
- Utilisées dans le cadre du framework de vérification
- Modélisation des pairs et des modèles de communication : systèmes de transition

## Formalisation opérationnelle unifiée basée histoire

- Formalisation reposant sur :
  - Histoires locales, causales et globales
  - Ensemble de messages en transit
- Formalisation des modèles :
  - Transition d'envoi : contraintes sur la capacité du réseau
  - Transition de réception : contrainte sur les histoires des messages
- Correspondance histoires / ordres

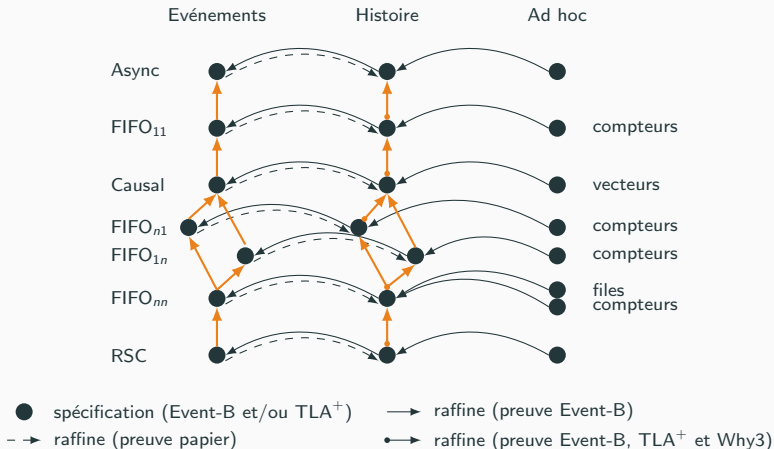
## Formalisations opérationnelles avec structures ad hoc

- Ensemble, file, vecteur d'horloge, compteur, ...



## Intuition

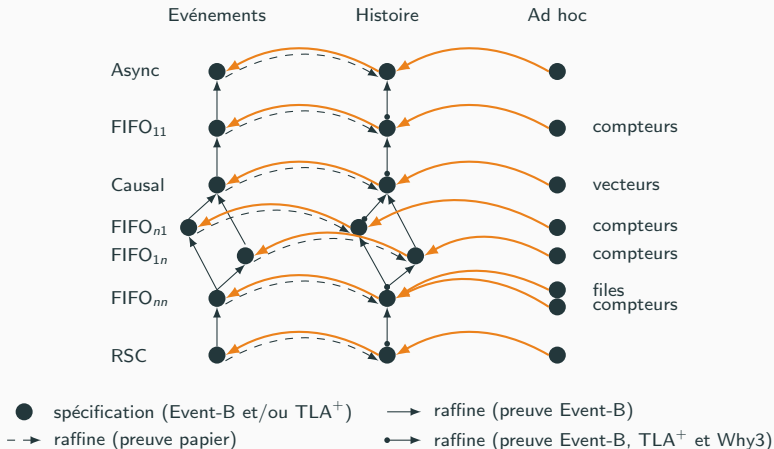
- Liens verticaux : hiérarchie des modèles
- Liens horizontaux : correction et complétude des modèles



## Intuition

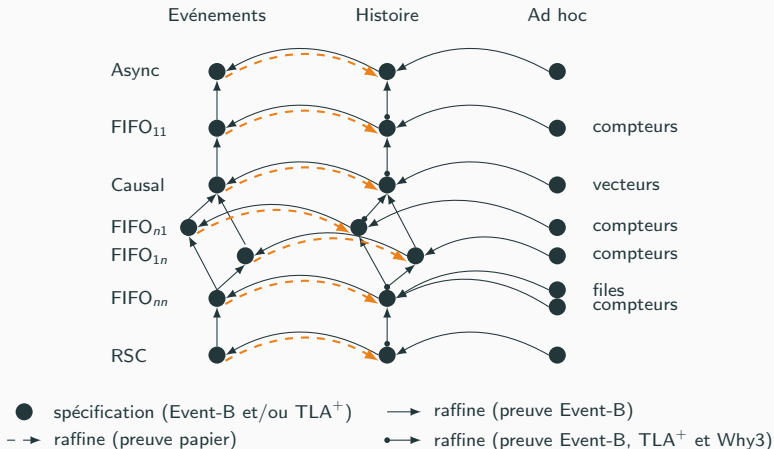
- Liens verticaux : hiérarchie des modèles
- Liens horizontaux : correction et complétude des modèles





## Intuition

- Liens verticaux : hiérarchie des modèles
- Liens horizontaux : **correction** et complétude des modèles



## Intuition

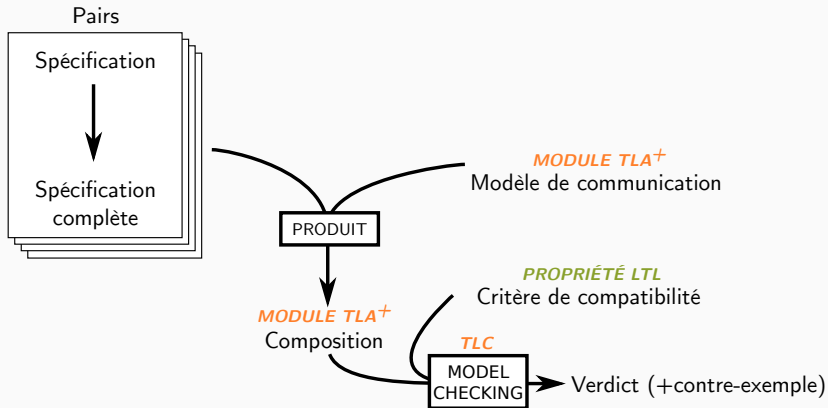
- Liens verticaux : hiérarchie des modèles
- Liens horizontaux : correction et **complétude** des modèles

## Correction

- Validité des traces
- Pas de faux négatif

## Complétude

- Exhaustivité des traces
- Pas de faux positif



<http://vacs.enseeiht.fr/>

Outil et méthode prouvés

+

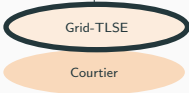
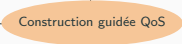

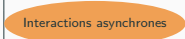

- Formalisation axiomatique
  - Uniforme
  - Hiérarchie étendue
  - Diffusion
- Formalisation opérationnelle
  - Uniforme
  - Outil automatique et fiable
  - Correction et complétude
- Cartographie des modèles
  - Vue globale
  - Preuve des liens

—

- Mécanisation de la preuve de complétude
- Complétude des modèles ad hoc
- Non prise en compte des défaillances

## Autres travaux

---

	Sémantique	Signature	Qualité de service	Comportement
Découverte	 <p>Grid-TLSE</p> <p>Courtier</p>	 <p>Construction guidée QoS</p>		
Sélection			 <p>Temps d'exécution</p>	
Validation				 <p>Interactions asynchrones</p>
Adaptation				 <p>Priorités applicatives</p>

## Problématique

- Découverte d'une composition de services avec plan connu
  - Découverte des services répondant aux contraintes du scénario (plan) et des paramètres d'entrées (matrices).

## Modélisation

- Composition de services : workflow mélangeant opérateurs d'exécution (services) et opérateurs de transformation de flux
- Service (solveur linéaire) : métadonnées

## Financements

Projet LEGO (ANR-CICG05-11), Projet FP3C (ANR-JST FP3C), Projet COOP (ANR-09-COSI-001), ...

## Collaborations

- Partenaires académiques : CERFACS, IRIT, LaBRI et LIP-ENS Lyon
- Partenaires industriels : CEA, CNES, EDF et IFP



	Sémantique	Signature	Qualité de service	Comportement
Découverte	<div>Grid-TLSE</div> <div>Courtier</div>	<div>Construction guidée QoS</div>		
Sélection			<div>Temps d'exécution</div>	
Validation				<div>Interactions asynchrones</div>
Adaptation				<div>Priorités applicatives</div>

## Problématique

- Découverte et sélection d'une composition de services dans le but d'optimiser la qualité de service

## Modélisation

- Services / Composition de services : types des entrées et des sorties
- Qualité de service : valeurs numériques agrégées ou non
- Algorithme : algorithme d'optimisation, méthode approchée basée sur l'intelligence en essaim

## Encadrement de thèse

- Sériel Boussalia (thèse soutenue en mai 2016)
- Co-encadrement avec Allaoua Chaoui, université de Constantine II

	Sémantique	Signature	Qualité de service	Comportement
Découverte	<div>Grid-TLSE</div> <div>Courtier</div>	<div>Construction guidée QoS</div>		
Sélection			<div>Temps d'exécution</div>	
Validation				<div>Interactions asynchrones</div>
Adaptation				<div>Priorités applicatives</div>

## Problématique

- Vérifier qu'une composition de services valide une propriété lorsque la communication est asynchrone
- Adapter une composition de services pour qu'une propriété soit validée (inférence d'un modèle de communication garantissant la propriété)

## Extension du framework et des modèles axiomatiques

- Priorité relative des messages
- Utilisation de canaux pour marquer les messages et exprimer les propriétés
- Inférence des contraintes

## Encadrement

- Nathanaël Sensfelder
  - master soutenu en septembre 2016

## Perspectives

---

## Modélisation formelle

- Aider à la généralisation des modélisations formelles.

## Outils et méthodes prouvés

- Fournir des outils et méthodes prouvés.
- Aider à la généralisation des preuves (mécanisées) formelles.

## Méthodologie

- L'enseignement
- L'exemple
- L'interaction avec les développeurs d'assistants de preuve, de vérificateurs de modèles, ...

## Trois domaines d'applications

- Systèmes répartis avec défaillances
  - Projet ANR PARDI (verification of PARAmeterized DIstributed systems)
  - Thèse Adam Shimi
- Systèmes non bloquants
  - Projet Toulouse Tech InterLabs SNOB (Systèmes NON Bloquants)
  - Collaborations :
    - LAAS
    - Instituto de Matematicas, UNAM, Mexico City
- Systèmes interactifs
  - Collaboration : ENAC

## Zoom sur ...

- Modélisation des systèmes répartis avec défaillances
- Vérification des systèmes non bloquants

## Défaillances dans les systèmes répartis

- Inévitables
- De natures diverses : arrêt définitif d'un site, perte de messages, . . .

## Objectif

- Raisonner sur des systèmes distribués asynchrones défaillants paramétrés
  - Résultats génériques sur l'abstraction des défaillances
  - Résultats génériques sur la réduction de l'asynchronie
  - Résultats génériques sur la paramétrisation (nombre de processus, modèle de communication, modèle de défaillance, . . .)

## Pistes

- Se concentrer sur les modèles par tour
- Abstraire les défaillances



## Modèle par tour synchrone

- A chaque tour, les pairs diffusent un message ; reçoivent des messages ; calculent leur nouvel état ; passent au tour suivant.
- Plus facile pour raisonner que sans structure générale
- Couvre une grande partie des algorithmes tolérants aux fautes

## Abstraction des types de défaillance : le modèle Heard-Of

- Prédicat  $HO(p, r)$  représentant l'ensemble des expéditeurs des messages reçus par le pair  $p$  au tour  $r$ .
- Modélise
  - Les différents types de défaillances
  - La dynamicité du réseau de communication
- Peut être forcé ou observé

Version asynchrone ?

## Modèle par tour asynchrone

- Quand le processus décide-t-il de passer au tour suivant ?
  - Synchrone : borne de temps calcul + transfert message
  - Asynchrone : nombre de messages reçus vs risque de blocage
- Modélisation par jeux
  - Environnement : maître du jeu
  - Processus : stratégie de changement de tour
  - Stratégie gagnante : aucun pair n'est bloqué
- Différents types de stratégies
  - Que le présent (sans mémoire)
  - Tout le passé et le présent
  - Un aperçu du futur

Comment construire une stratégie pour un modèle de défaillances ? Optimale ?

## Domaine d'application

- Systèmes concurrents à mémoire partagée

## Intérêts des systèmes non bloquants

- Résistance à l'arrêt (crash) d'un processus
- Vitesse de progression indépendante de celle des autres activités
- Gain de performance : augmentation du parallélisme

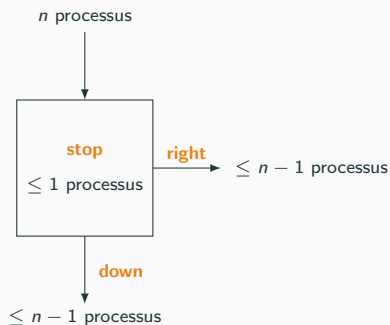
## Problématiques des systèmes non bloquants

- Difficiles à écrire même si concis
- Complexes à analyser même si concis

## Objectif

- Preuve mécanisée des algorithmes non bloquants

## Spécification et implantation du splitter



```
direction(id)
  X := id
  if Y
  then dir := right
  else Y := true
    if (X=id)
    then dir := stop
    else dir := down
    endif
  endif
  return dir
```

## Objectif du splitter

- Partitionner les processus

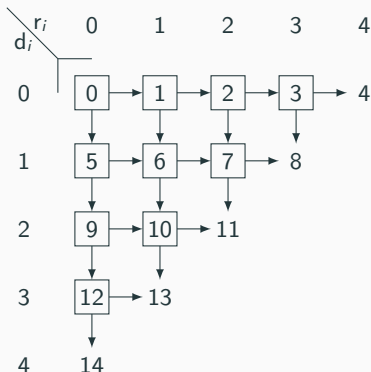
## Preuve du splitter

- Réalisé avec Why3 et TLAPS

- Invariant inductif

```
Prop1 == (pc[0]="spl1" /\ pc[0]="spl0") => (\neg Y)
Prop2 == X=0
Prop3 == pc[0]="spl1" /\ pc[0]="spl0" /\ pc[0]="spl3" /\ pc[0]="spl4" /\ pc[0]="spl5" /\ pc[0]="Done"
Prop4 == rval[0]=None /\ rval[0]=Stop
Prop5 == \A i \in ProcSet : rval[i] # None <=> pc[i]="Done"
Prop6 == (\E i \in ProcSet : (pc[i]="spl4") /\ (pc[i]="spl5") ) => Y
Prop7 == (\E i \in ProcSet : rval[i]=Stop) => Y
Prop8 == \A i \in ProcSet : (pc[i]="spl5" => (X=i /\ pc[X]="spl1" /\ pc[X]="spl2" /\ (pc[X]="Done" /\ rval[X]=Right)))
Prop9 == \A i \in ProcSet : ((pc[i]="Done" /\ rval[i]=Stop) =>
  (X=i /\ pc[X]="spl1" /\ pc[X]="spl2" /\ (pc[X]="Done" /\ rval[X]=Right)))
Prop10 == \A i,j \in ProcSet : ((pc[i]="spl5" /\ pc[j]="spl5") => i=j)
Prop11 == ~(\E i,j \in ProcSet : pc[i]="spl5" /\ pc[j]="Done" /\ rval[j]=Stop)
Prop12 == \A i,j \in ProcSet : (rval[i]=Stop /\ rval[j]=Stop) => i=j
Prop13 == (\E i \in ProcSet : pc[i]="spl2" /\ rval[i]=Right) => Y
Prop14 == Y =>
  ((\E i \in ProcSet : pc[i]="spl4" /\ rval[i]=None) /\ (\E i \in ProcSet : pc[i]="spl5" /\ rval[i]=None)
   /\ (\E i \in ProcSet : pc[i]="spl6" /\ rval[i]=None) /\ (\E i \in ProcSet : pc[i]="Done" /\ rval[i]=Down)
   /\ (\E i \in ProcSet : pc[i]="Done" /\ rval[i]=Stop))
Prop15 == (\E i \in ProcSet : rval[i]#Right)
Prop16 == (pc[X]="spl0" /\ rval[X]=None) /\ (pc[X]="spl1" /\ rval[X]=None) /\ (pc[X]="spl2" /\ rval[X]=None)
  /\ (pc[X]="spl3" /\ rval[X]=None) /\ (pc[X]="spl4" /\ rval[X]=None) /\ (pc[X]="spl5" /\ rval[X]=None)
  /\ (pc[X]="Done" /\ rval[X]=Right) /\ (pc[X]="Done" /\ rval[X]=Stop)
Prop17 == \E i \in ProcSet : rval[i]#Down
```

## Exemple du renommage



```
rename(id)
while (d+r < NP-1 /\ !stop ) do
  X[d][r] := id;
  if Y[d][r]
  then r := r + 1          /* right
  else
    Y[d][r] := TRUE
    if (X[d][r] = id)
    then stop := TRUE      /* stop
    else d := d + 1        /* down
    endif
  endif
endwhile
return (NP*d) + r - (d*(d-1)) \div 2
```

## Spécification du renommage

Chaque processus obtient une valeur de retour (nouveau nom) distincte.

## Preuve du renommage

- Preuve papier par Moir et Anderson
  - Invariant inductif sur le modèle de celui du splitter
- Début de preuve TLAPS
- Problème : n'utilise pas les propriétés prouvées sur le Splitter, car les splitters ne sont pas des boîtes fonctionnelles à cause des entrelacements

## Preuve des algorithmes non bloquants

- Comment faire réapparaître des "modules" pour pouvoir utiliser leur propriété?
- Possibilité de mécaniser cette réduction modulaire?

Besoin de garantie quelle que soit la criticité de l'application.

### Méthodologie

- Modélisation formelle
- Outils et méthodes prouvés

### Composition de services

- Passé : services de calculs ; systèmes répartis sans défaillance
- Futur : systèmes interactifs ; systèmes répartis avec défaillances ; systèmes non bloquants