

## *Lot 3 : Test, Animation*

Partenaires impliqués :  
CRIL, FT R&D, LaBRI, Loria

# Description du lot

---

Thème : Techniques d'exécution de spécifications

- Sous lot 1 : Animation
- Sous lot 2 : Test

# Sous-lot 3.1 : Animation

---

- Buts :
  1. Visualiser des traces d'exécution
    - Forme graphique : Visualisation des différents automates, des localités, transitions, valuations des variables...
    - Générique : trace d'exécution produite par un outil
      - Accessibilité d'un model-checker
      - Générateur de séquence de test
      - ...
  2. Produire des traces d'exécution à partir d'ELAN
    - CRIL - LORIA
    - ELAN : langage à base de règles de réécriture
    - Model-checker pour l'accessibilité avec des règles de réécriture en ELAN

## Sous-lot 3.1 : Etat d'avancement

---

- Fournitures :
  - F3.1.1** : description XML d'une trace d'exécution
- En cours d'études :
  - F3.1.2** : Elan pour générer des traces d'exécutions
  - F3.1.3** : Intégration du simulateur à la plateforme Calife

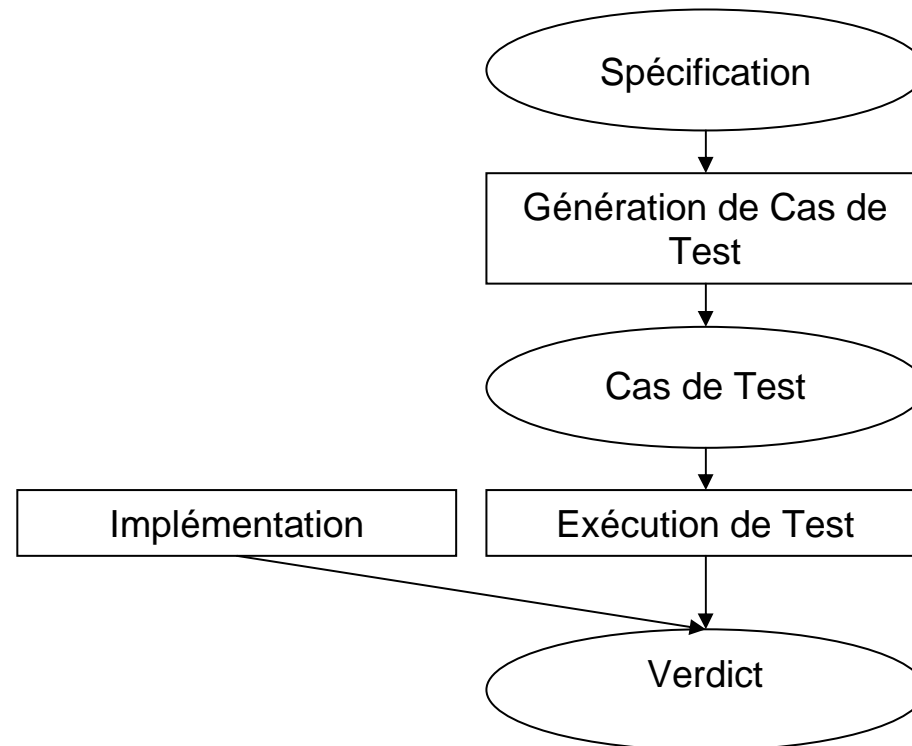
## Sous-lot 3.2 : Test

---

- But : Génération de cas de test
- Différents types de test :
  - **Conformité**
  - **Interopérabilité**  
Tester que plusieurs implantations sous test communiquent, inter opèrent et fournissent les services attendus
  - Robustesse
  - Performance
  - ...

## Sous-lot 3.2 : Activité de test

---



## Sous-lot 3.2 : Modèle

---

- P-Automates restreints avec les paramètres instanciés
    - Types de variables élémentaires (Time, Z)
    - Actions élémentaires (affectation 'linéaire')
    - Prédicats 'linéaires'
- (automate temporisé avec I/O et action)

## Sous-lot 3.2 : Approche 'Objectif de test'

---

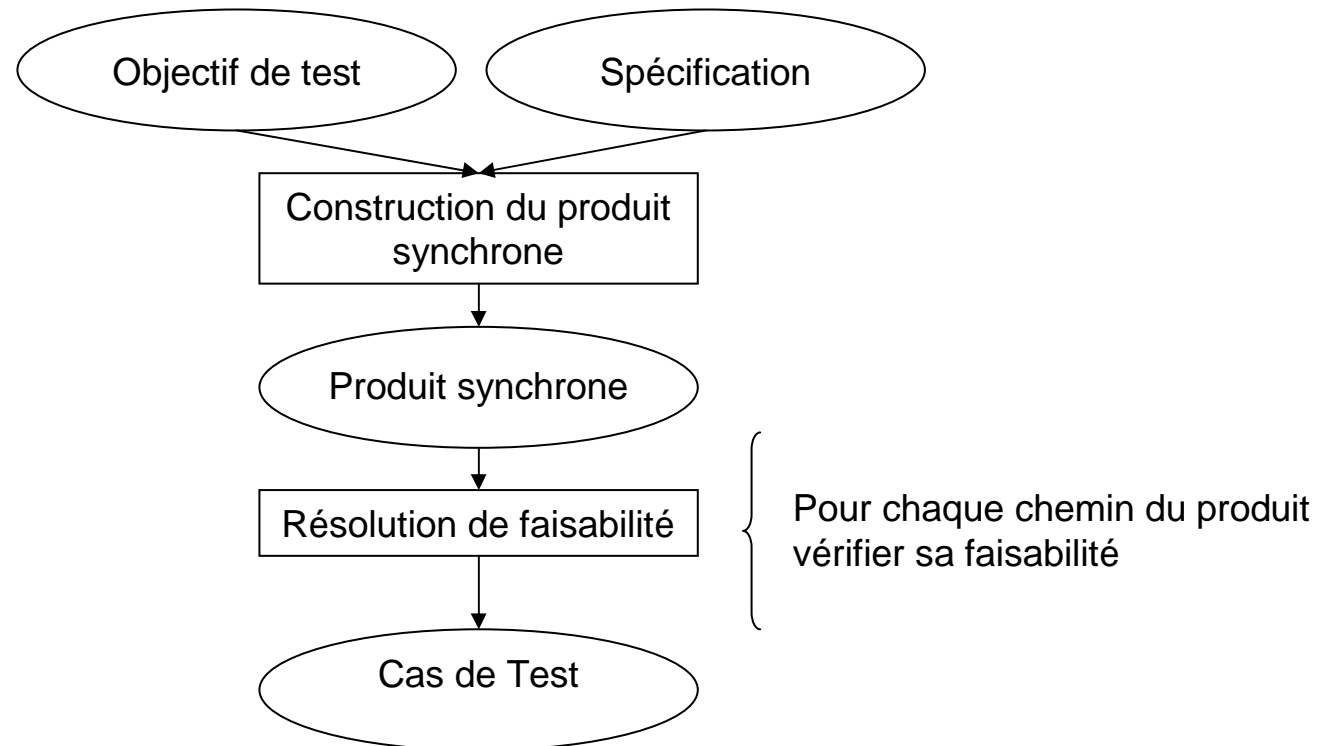
Un objectif de test  $OT$  est un automate déterministe acyclique ayant un ensemble non vide d'états acceptant : il représente une propriété à observer sur le comportement d'une implantation.

- Principe : Trouver une trace d'exécution  $Tr$  de la spécification  $Spéc$  'respectant' l'objectif de test  $OT$ .



## Sous-lot 3.2 : Approche 'Objectif de test'

---



## Sous-lot 3.2 : Faisabilité d'un chemin

---

Etant donné un chemin, décider de l'accessibilité de l'état terminal de ce chemin.

1. Trace symbolique :  $\sigma(\rho) = (a_1\tilde{P}_1)\dots(a_n\tilde{P}_n)$
2. Résolution de contraintes

Le chemin  $\rho$  faisable ssi  $\tilde{P}_1 \wedge \dots \wedge \tilde{P}_n$

1. Programmation linéaire : instantiation des variables
2. Calcul des contraintes atteignables

## Sous-lot 3.2 : Etat d'avancement

---

- Fournitures
  - **F3.2.1** : 'Techniques de tests d'interopérabilité pour les systèmes temporisés'.
- En cours d'études
  - **F.3.2.2** : Prototype *GenTest* de génération de séquences de test d'interopérabilité.  
Implémentation des différents algorithmes (synchronisation, trace symbolique, résolution de contraintes), intégration de *GenTest* dans Calife
- T0+30
  - **F.3.2.3** : Validation du prototype sur des études de cas.  
[Etude de cas à déterminer : PGM (Pragmatic General Multicast)...]