

AIDE AU DÉVELOPPEMENT DE MODÈLES DEVS PAR INTÉGRATION DE TESTS COMPORTEMENTAUX DANS LA SIMULATION

Doctorant : *Timothée ville*

Directeur de thèse : *Jean-François Santucci*

Co-directeur de thèse : *Laurent Capocchi*

SOMMAIRE

SOMMAIRE

- Introduction

SOMMAIRE

- Introduction
- Vue d'ensemble

SOMMAIRE

- Introduction
- Vue d'ensemble
- Contributions

SOMMAIRE

- Introduction
- Vue d'ensemble
- Contributions
- Conclusion et perspectives

INTRODUCTION

INTRODUCTION

- Problématique

INTRODUCTION

- Problématique
- Projet TIC
 - Le formalisme DEVS et DEVSimPy

INTRODUCTION

- Problématique
- Projet TIC
 - Le formalisme DEVS et DEVSimPy
- Méthodes "AGILE" TDD et BDD

INTRODUCTION :

PROBLÉMATIQUE

- Processus de design de modèle
- Phase de test souvent relégué á la fin du processus de design à cause du coût en temps relativement important
- Qualité aboutissante médiocre

INTRODUCTION :

PROJET TIC : LE FORMALISME DEVS ET DEVSIMPY

- Développement de concepts génériques
- Mise en oeuvre sous DEVS avec DEVSimPy
- DEVS \Rightarrow Formalisme
 - Pr. Zeigler
 - Spécifications de systèmes complexes
- DEVSimPy
 - GUI pour DEVS
 - Python

INTRODUCTION :

MÉTHODES "AGILE" TDD ET BDD

- Utilisation du programme avant qu'il existe
- TDD :
 - Le développeur doit se poser les bonnes questions
- BDD :
 - Le développeur n'est plus seul à se poser ces questions
 - Quand j'ai [conditions initiales]
et que je fais [action]
alors j'obtiens [résultat]

VUE D'ENSEMBLE

VUE D'ENSEMBLE

- Contexte

VUE D'ENSEMBLE

- Contexte
- Solution proposée

VUE D'ENSEMBLE :

CONTEXTE

VUE D'ENSEMBLE :

CONTEXTE

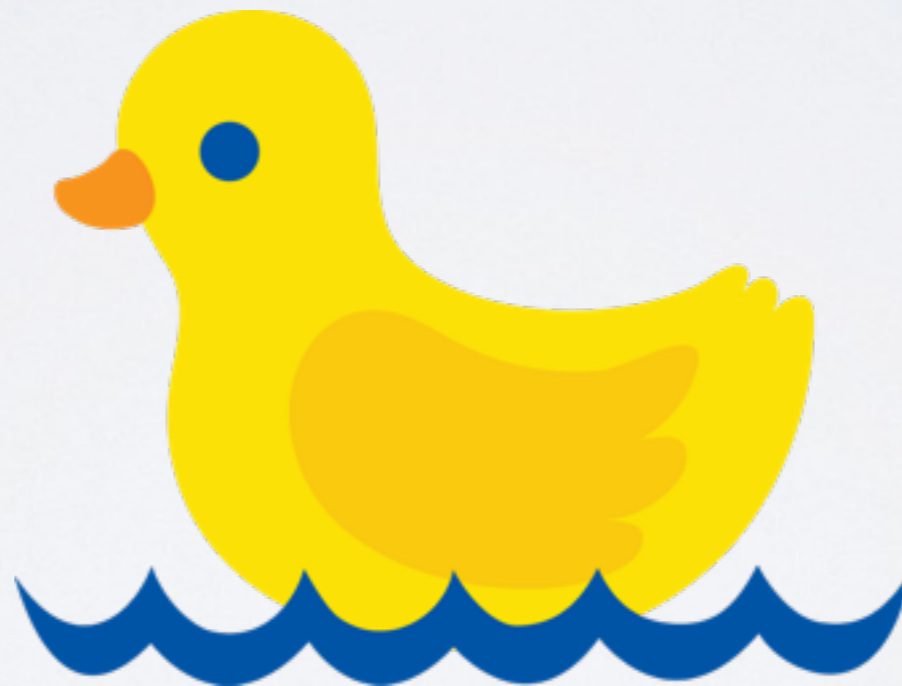
Pouvez
vous me modéliser
un canard?



VUE D'ENSEMBLE :

CONTEXTE

Pouvez
vous me modéliser
un canard?



VUE D'ENSEMBLE :

CONTEXTE

Pouvez
vous me modéliser
un canard?



Il faut un peu de temps avant
que votre canard ai un comportement
normal.



VUE D'ENSEMBLE :

CONTEXTE

Pouvez
vous me modéliser
un canard?



Il faut un peu de temps avant
que votre canard ai un comportement
normal.



VUE D'ENSEMBLE :

SOLUTION PROPOSÉE

VUE D'ENSEMBLE :

SOLUTION PROPOSÉE

Pouvez
vous me modéliser
un canard?



VUE D'ENSEMBLE :

SOLUTION PROPOSÉE

Comment
ça se comporte un
canard?



VUE D'ENSEMBLE :

SOLUTION PROPOSÉE

Comment
ça se comporte un
canard?



Je vous écrit le
comportement du canard
(spécifications)



VUE D'ENSEMBLE :

SOLUTION PROPOSÉE

Comment
ça se comporte un
canard?



Je vous écrit le
comportement du canard
(spécifications)



CONTRIBUTIONS :

CONTRIBUTIONS :

- Langage de spécification de test semi-formel

CONTRIBUTIONS :

- Langage de spécification de test semi-formel
- Génération de scénario de test

CONTRIBUTIONS :

- Langage de spécification de test semi-formel
- Génération de scénario de test
- Exécution de test

CONTRIBUTIONS :

LANGAGE DE SPÉCIFICATION DE TEST SEMI-FORMEL

- Comment doit-on écrire ses spécifications?
 - to start hold in generate for time t_0 !
after generate output Job !
from generate go to generate !
when in generate and receive Stop then go to passive !
passivate in passive !
- Langage naturel

CONTRIBUTIONS :

GÉNÉRATION DE SCÉNARIO DE TEST

from generate go to generate !

CONTRIBUTIONS :

GÉNÉRATION DE SCÉNARIO DE TEST

from generate go to generate !



CONTRIBUTIONS :

GÉNÉRATION DE SCÉNARIO DE TEST



CONTRIBUTIONS :

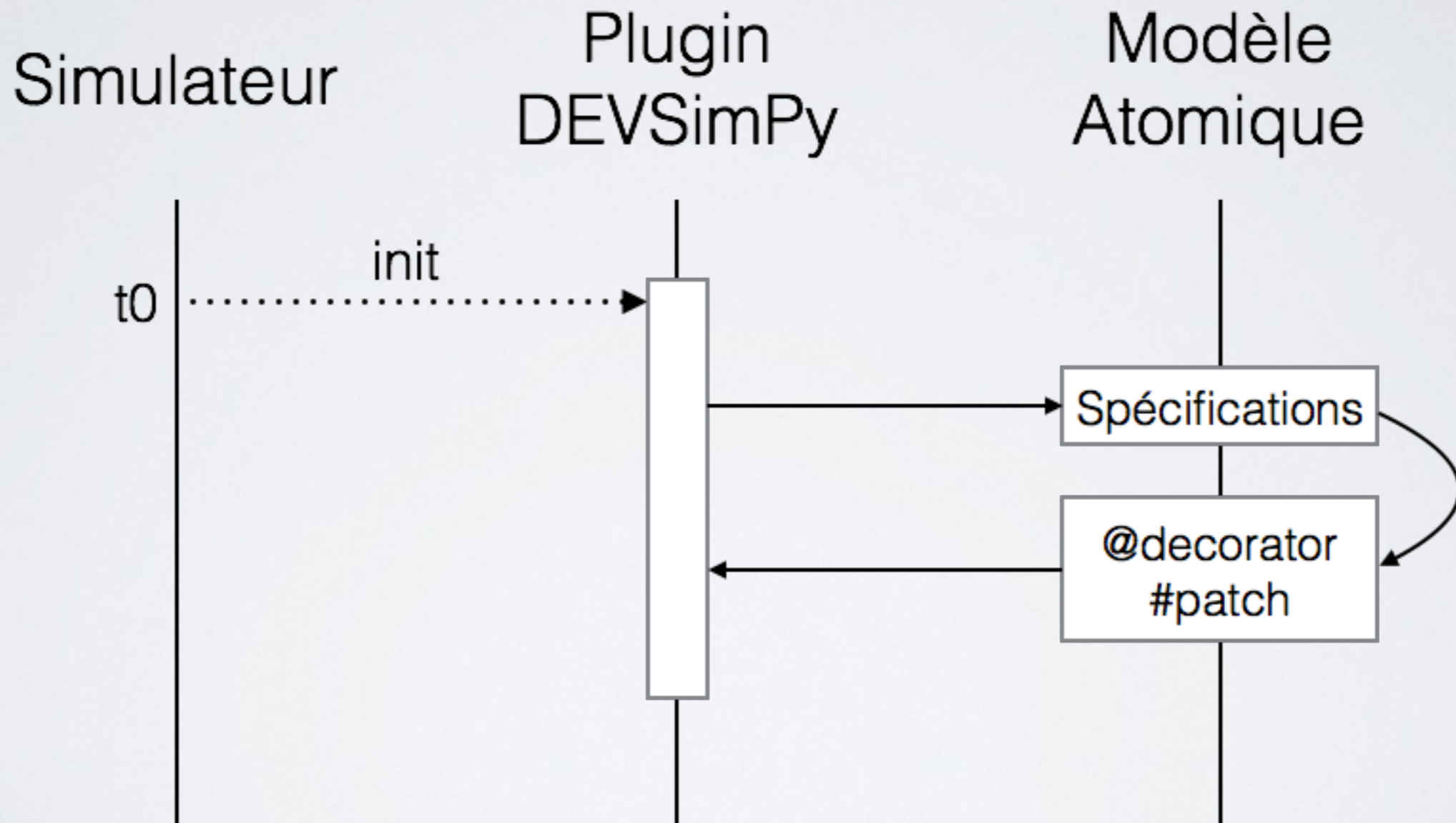
GÉNÉRATION DE SCÉNARIO DE TEST

B.T.G

```
# intTransition(generate) should do : from generate go to generate
def dec_intTransition(intTransition):
    def new_intTransition():
        realStatus = intTransition.__self__.state["status"]
        if realStatus == "generate":
            status = "generate"
            intTransition()
            if realStatus == status:
                print "intTransition[generate] --> OK"
            else:
                print "Error in intTransition function : \
                status should be %s and we have %s"
                %(status, realStatus)
        return intTransition
    return new_intTransition
```

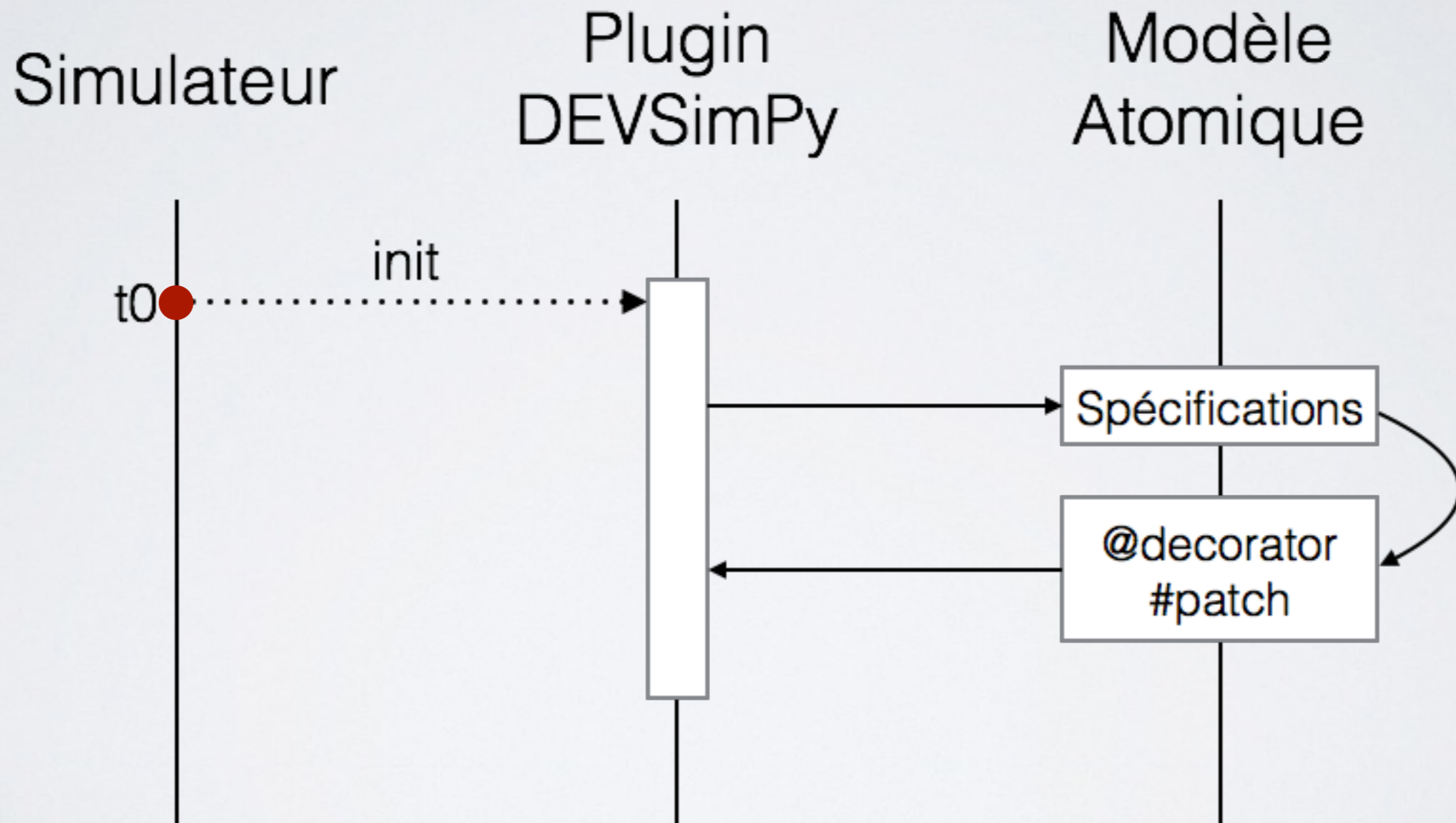
CONTRIBUTIONS :

EXÉCUTION DE TEST



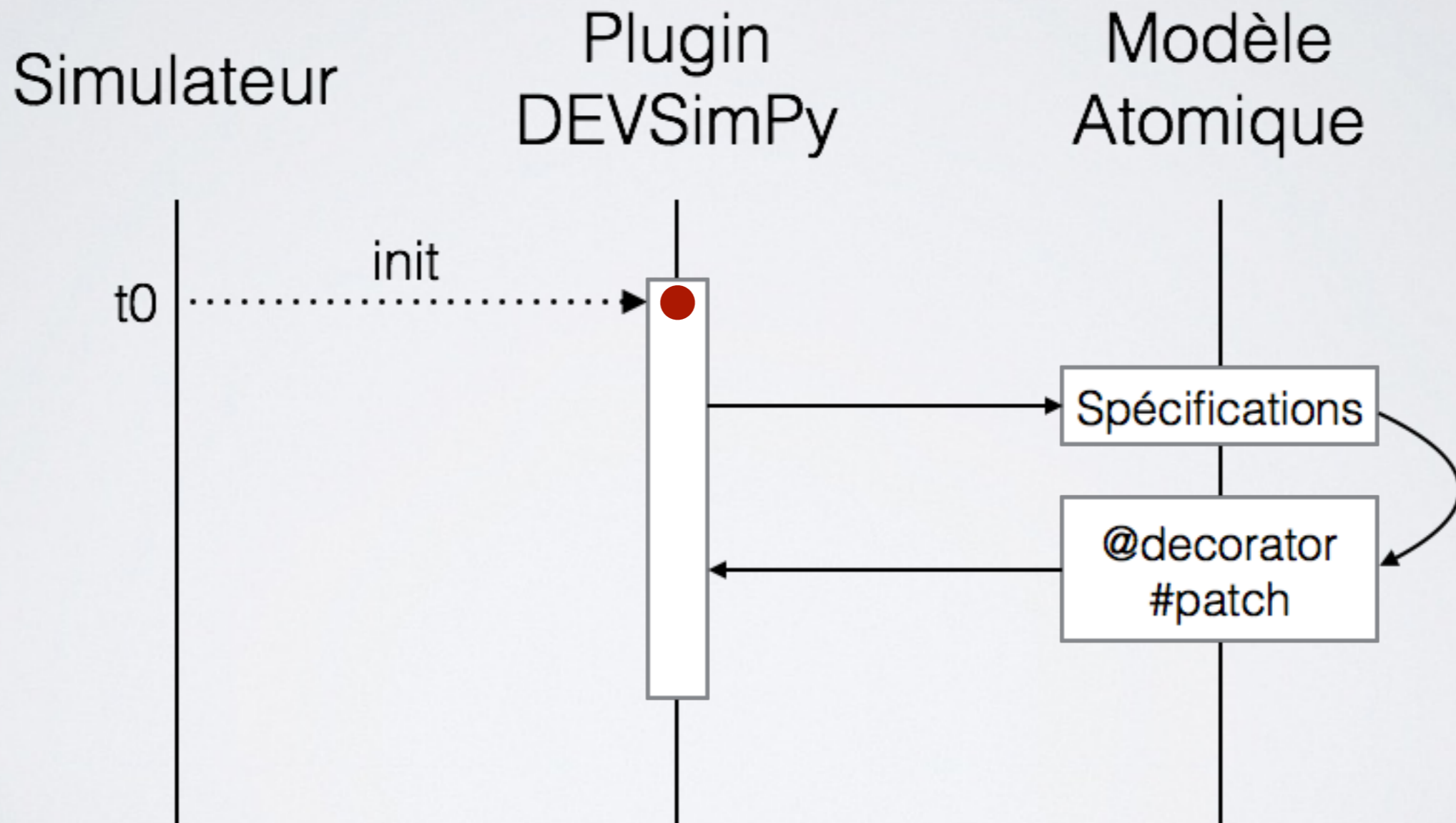
CONTRIBUTIONS :

EXÉCUTION DE TEST



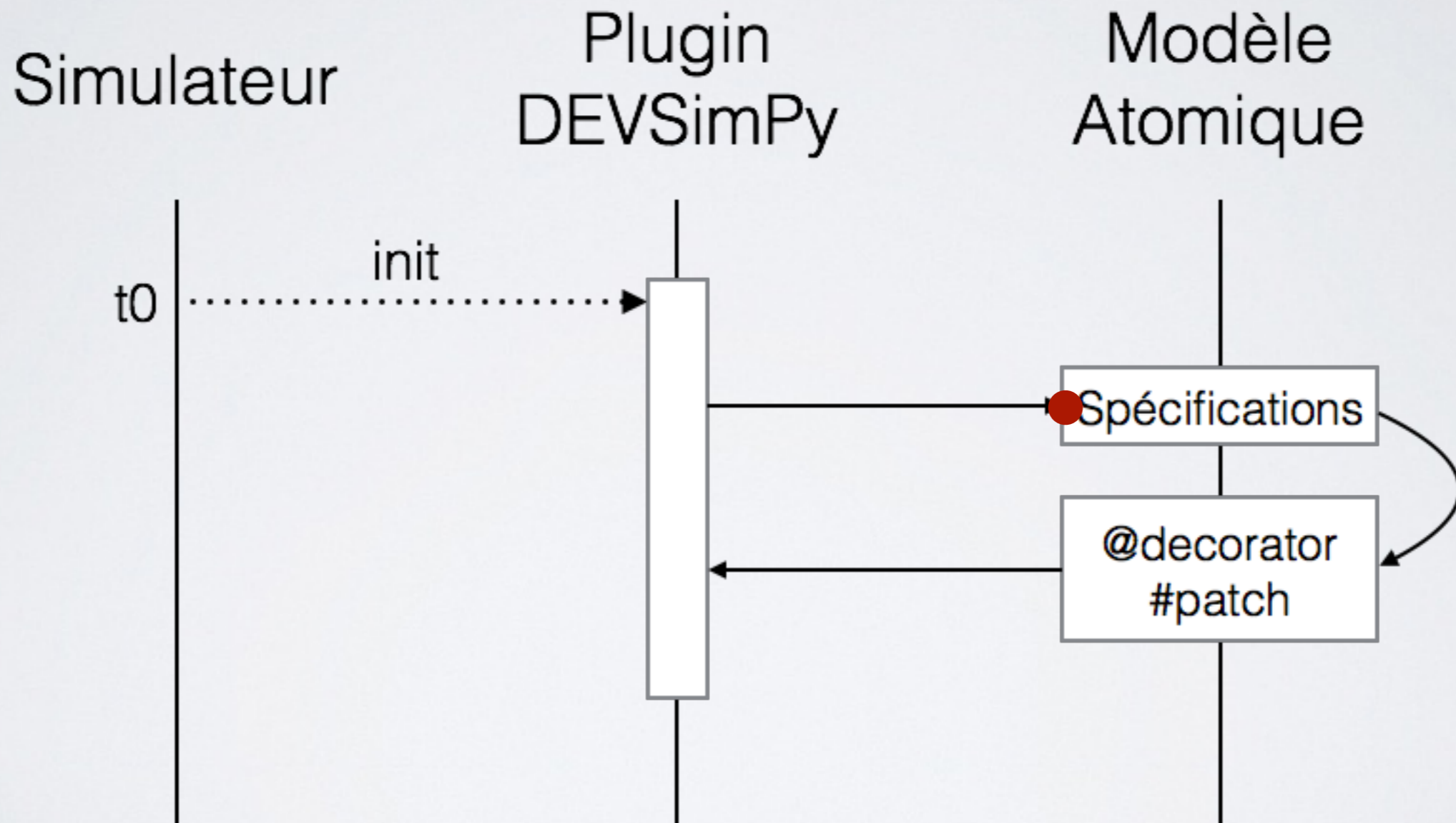
CONTRIBUTIONS :

EXÉCUTION DE TEST



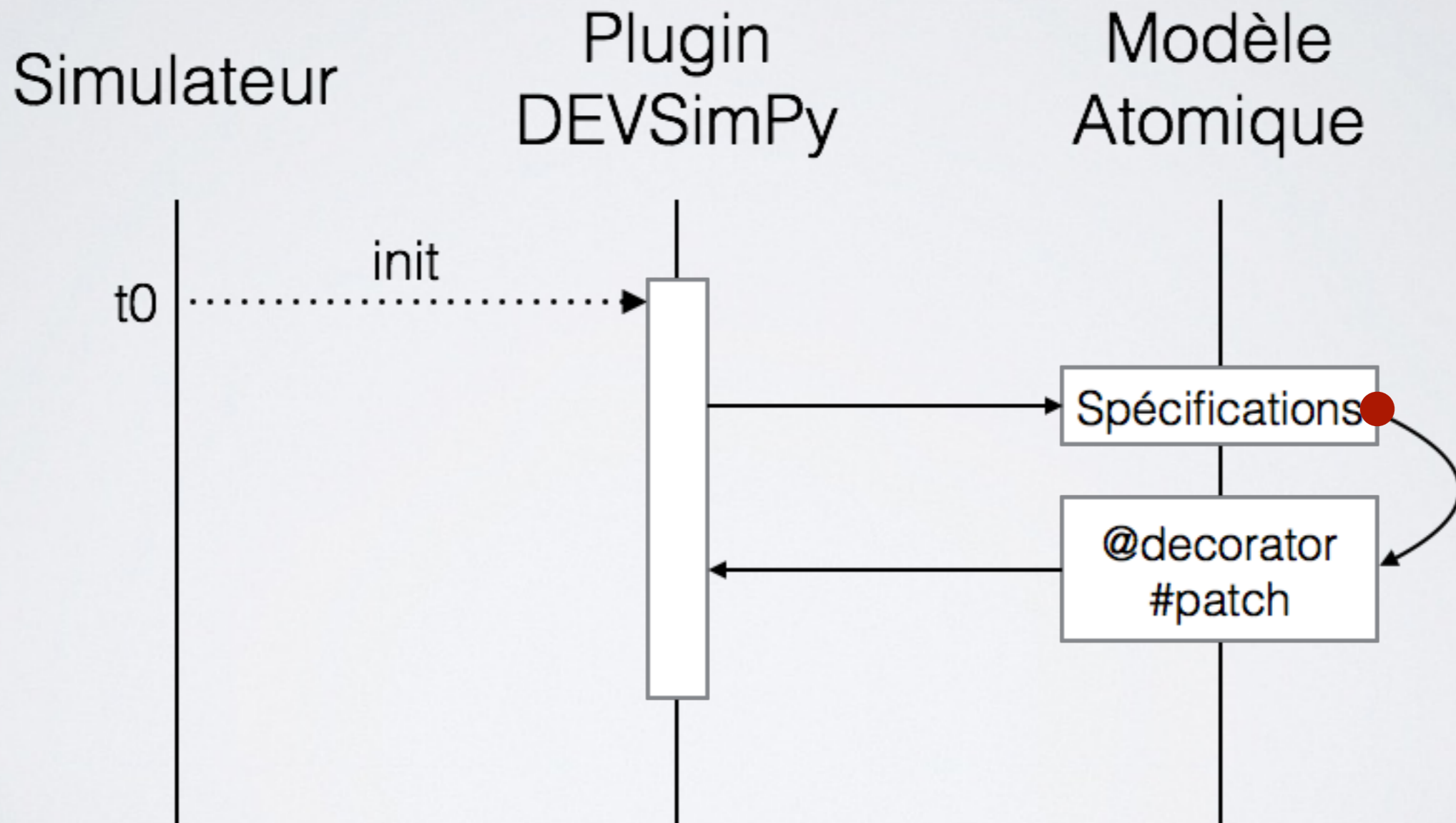
CONTRIBUTIONS :

EXÉCUTION DE TEST



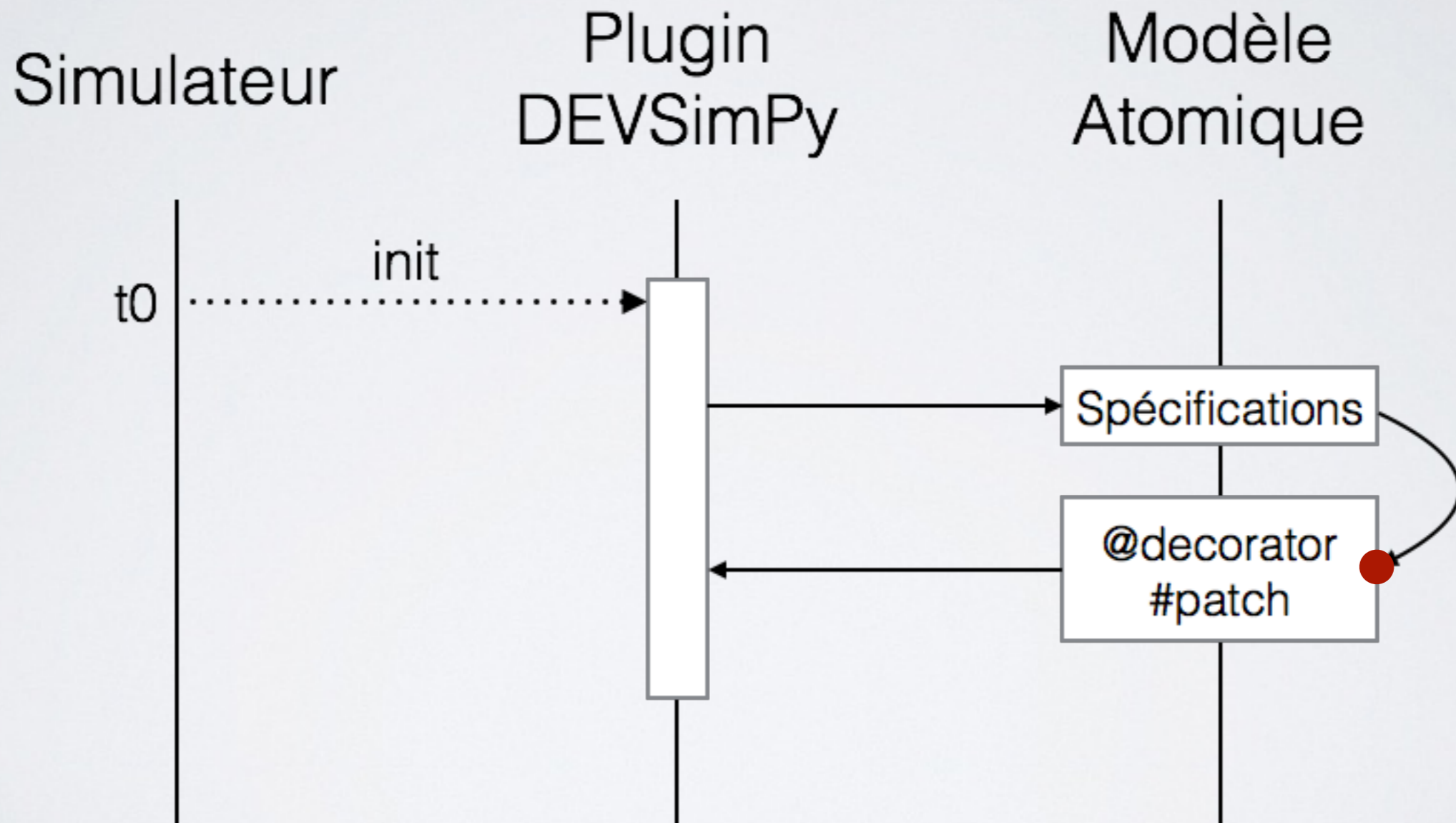
CONTRIBUTIONS :

EXÉCUTION DE TEST



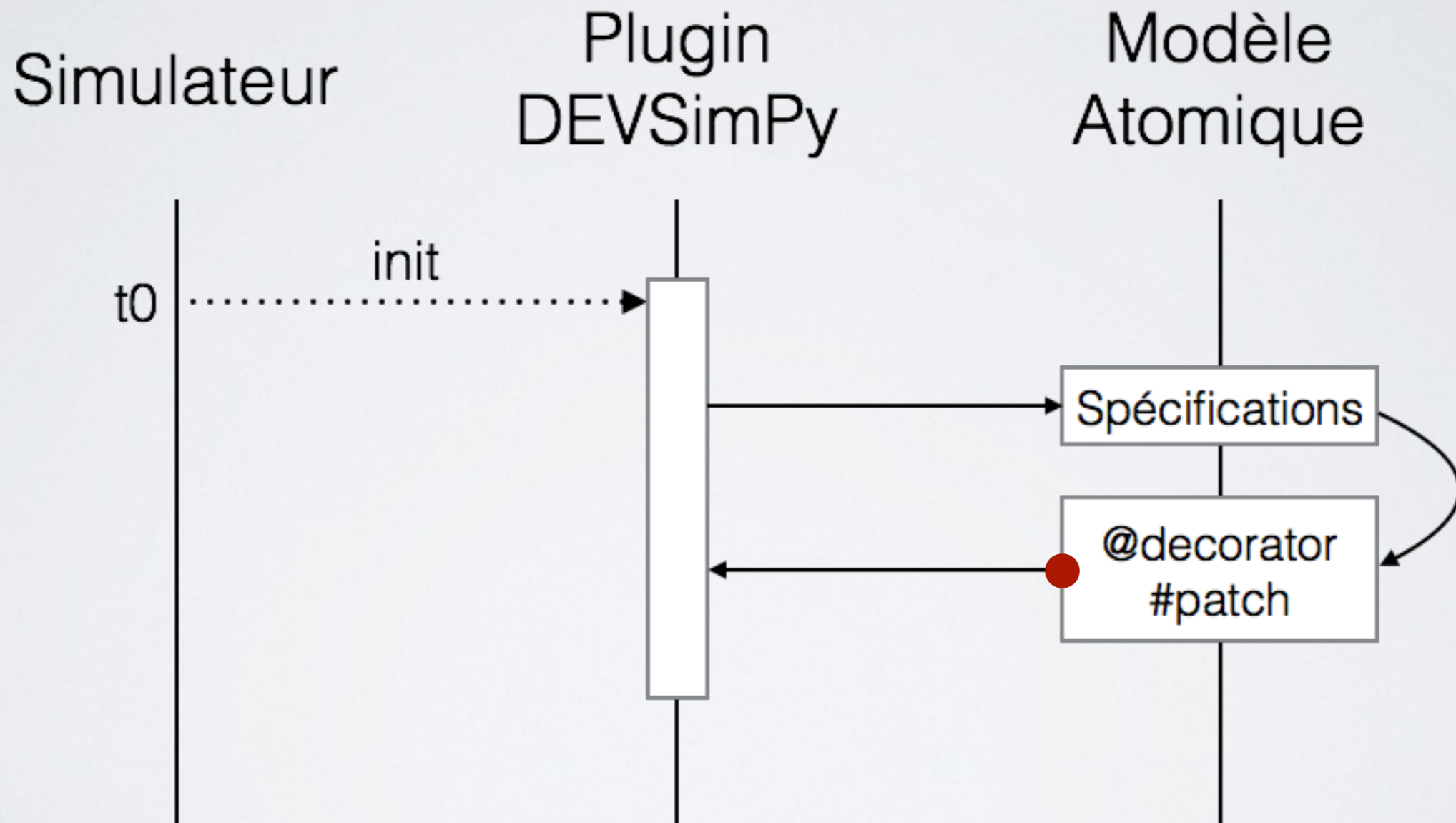
CONTRIBUTIONS :

EXÉCUTION DE TEST



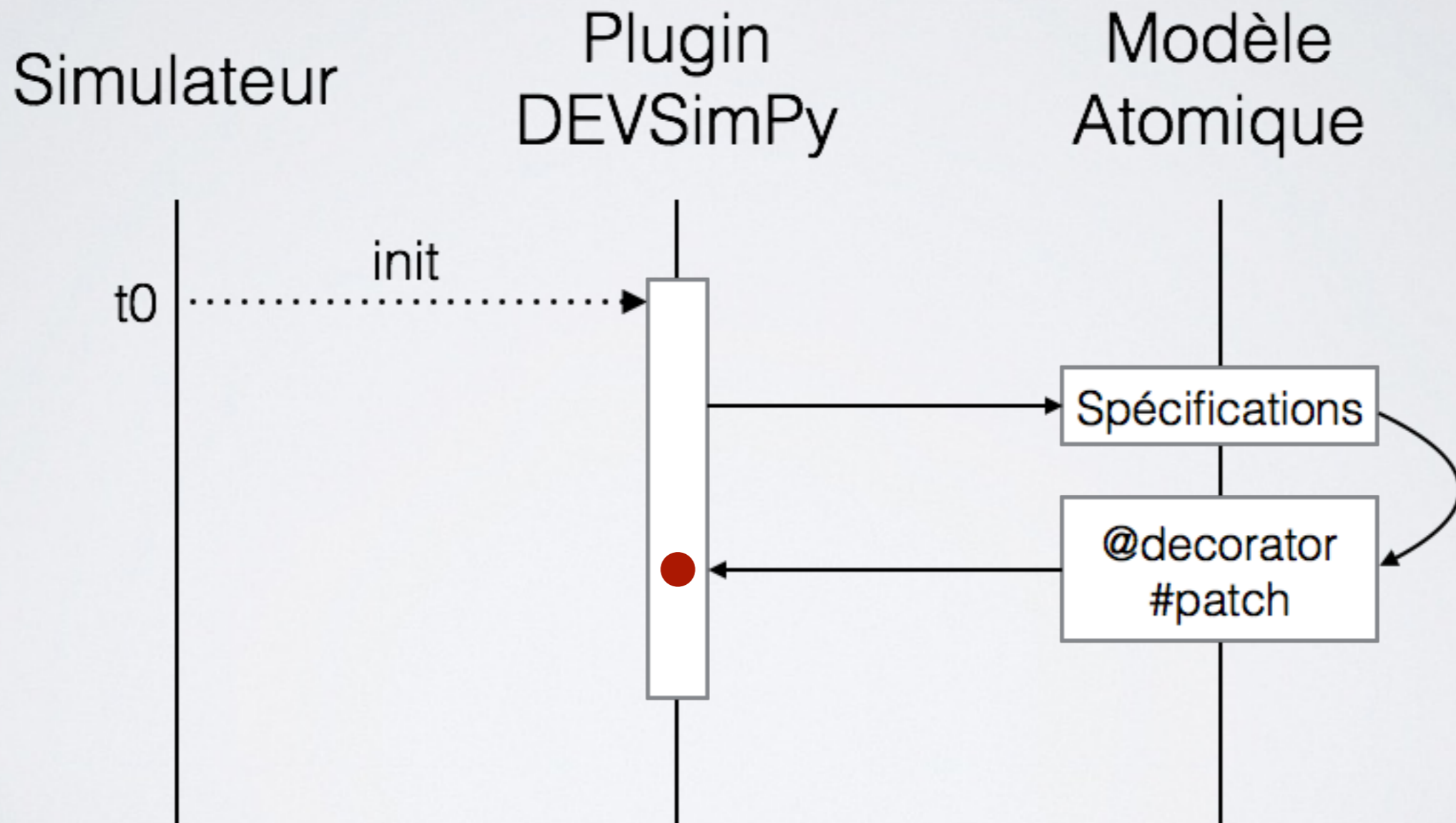
CONTRIBUTIONS :

EXÉCUTION DE TEST



CONTRIBUTIONS :

EXÉCUTION DE TEST



CONCLUSION ET PERSPECTIVES

- Emprunt des méthodes AGILE
- Parallèle avec la modélisation DEVS
 - Nouveau langage de spécification
 - Nouvelles méthodes de génération et d'exécution des scénarios de test
- Futurs travaux sur des modèles plus complexes