

# CostoTest (un état des lieux)

Pascal ANDRE, Jean-Marie Mottu

AELOS / LINA – UMR CNRS 6241  
{Firstname.Lastname}@univ-nantes.fr

Exposés AeLoS

# Plan de l'exposé

- 1 Introduction
- 2 Processus illustré
- 3 Implantation
- 4 Bilan et perspectives

# Plan

- 1 Introduction
- 2 Processus illustré
- 3 Implantation
- 4 Bilan et perspectives

# Introduction / contexte et motivations

**Contexte** : Test d'applications à composants et services logiciels distribués

- Cible : Niveau modèle vs. niveau code
- Exécutabilité, animation
- Assistance à la construction des tests (harnais)
- Gestion de l'environnement de tests
- Partage du support de développement - banalisation
- Support : COSTO/Kmelia

⇒ revoir l'exposé exposé **Gencode** ([gencode.pdf](#)) : contexte, bibliographie...

# Introduction / cible

## Test de modèles

- détection des erreurs en amont (moins coûteux)
- abstraction des contraintes de programmation et de déploiement (PIM)
- évolutivité et réactivité dans les premières itérations du développement
- généricité du test (unitaire, intégration, recette)
- abstraction du test ( $x\text{Unit} \equiv \text{JUnit} \cup \text{SUnit} \cup \dots$ )
- test non intrusif (encapsulation)

**Harnais** (TPIM) + traçabilité

# Introduction / Exécutabilité

## Pouvoir exécuter des cas de tests à partir de modèles

- génération de code automatique
- enchaîner les exécutions de jeu de test pour un test donné
- évolution des programmes : pas d'intervention manuelle
- évolutivité des cas de test : réutiliser les données (XML)
- variabilité : mutation, bibliothèques de test
- portabilité ?

**Harnais** génération de code + bancs de test

# Introduction / Assistance

## Aider le testeur à construire ses applications

- partir d'une intention, même incomplète, et d'un modèle d'application
- créer des applications de test est une activité d'ingénierie et de développement
- non intrusion dans les programmes de base
- découverte des données pour l'initialisation
- découverte des services pour l'interaction
- API ouverte de manipulation de modèles
- généricité du test (unitaire, intégration, recette)
- abstraction du test ( $xUnit \equiv JUnit \cup SUnit \cup \dots$ )
- test non intrusif (encapsulation)
- qualité via la mutation

Harnais construction + vérification

# Introduction / environnement

## Gestion de modèles et cas de test

- gérer jeux de données et résultats
- gérer variantes d'un test (mutation)
- gérer cohérence et couverture des tests (banc de test)
- domaine du test (unitaire, intégration, recette)
- classification, réutilisation
- évolution...

## Banc de test bonnes pratiques



# Introduction / banalisation

## Un modèle de test = un modèle

- réutiliser TOUT l'environnement de travail (outillage) : édition, vérification, transformation, génération de code, documentation...
- un test = un composant (de l'intention au cas de test)
- évolutivité et réactivité dans les premières itérations du développement
- généricité du test (unitaire, intégration, recette)
- abstraction du test ( $xUnit \equiv JUnit \cup SUnit \cup \dots$ )
- test non intrusif (encapsulation)

Harnais (TPIM) + traçabilité

# Introduction / Support

## COSTO/Kmelia (boîte à outils COSTO) :

- Kmelia : modèle et langage pour composants et services logiciels
  - structure : Composants/Interfaces/Service/Architectures/Composites
  - calculs : actions
  - dynamique : LTS + communication
- Boîte à outils COSTO (Famille de plugins Eclipse)
  - Editions, vérification langage
  - Vérification de propriétés de modèles (structure, dynamique, fonctionnels)
  - Génération de code pour composants et services distribués
    - Framework plain Java (Class, threads, buffers)
    - traçabilité
    - exécution, animation (Sémantique opérationnelle)
  - Documentation
- Wiki, forge

## COSTO/Kmelia Maîtrise de l'environnement

# Introduction / focus

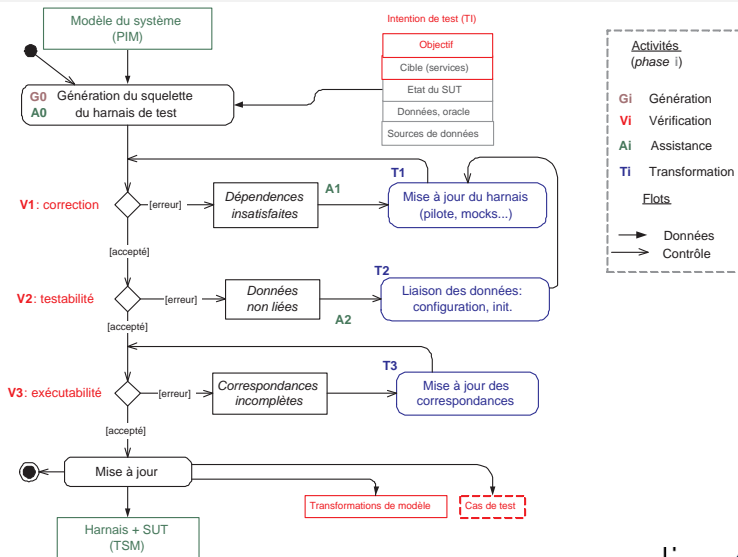
## CostoTest :

- Modèle, processus
  - Assistance au test de modèles à composants et services [?]
  - Building Test Harness From Service-based Component Models [?]
- Outils - COSTOTest : A Tool for Building and Running Test Harness for Service-based Component Models
  - modèle de test
  - processus complet implanté
  - exécutabilité (données en XML)
  - étude de cas
- Expérimentations
  - comparaison avec JUnit
  - en cours
- Forge et documentation
  - <http://www.lina.sciences.univ-nantes.fr/aelos/wiki/doku.php/costo> :test :start
  - **Application**

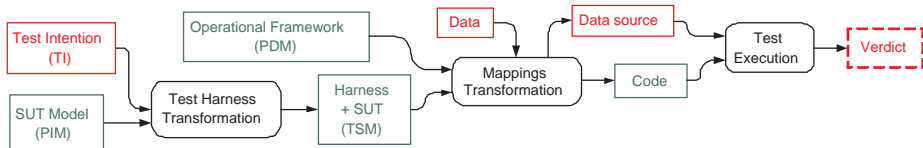
# Plan

- 1 Introduction
- 2 Processus illustré**
- 3 Implantation
- 4 Bilan et perspectives

# CostTest - processus (spécification)

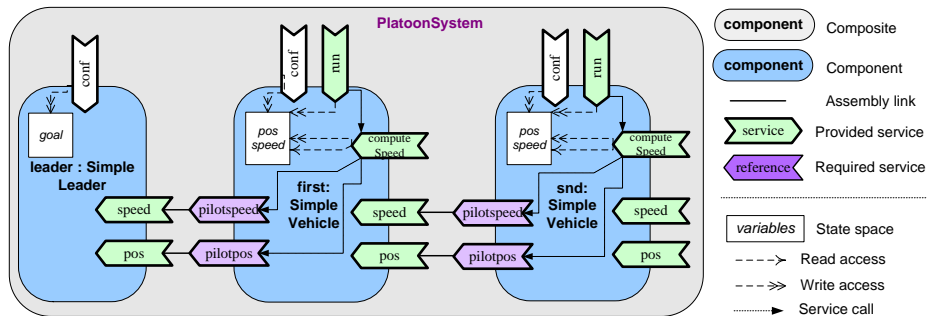


# CostTest - processus (implantation)



- 1 construction
- 2 mutation (optionnel)
- 3 exécution

# CostTest - entrées 1/2



- 1 assemblage de composants
- 2 vérification
- 3 exécution

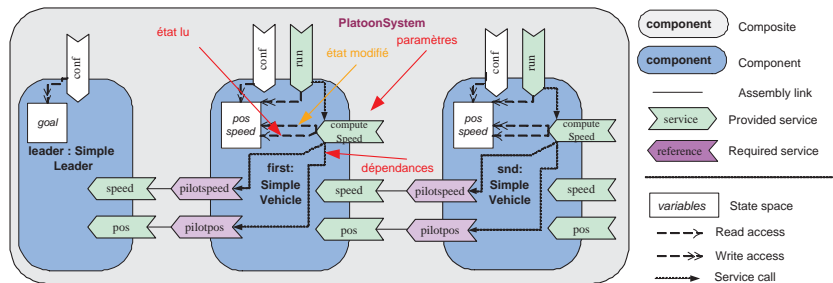
## CostTest - entrées 2/2

```
Kml PlatoonTestIntention.kcp
|TEST_INTENTION PlatoonTestIntention
DESCRIPTION "test of the service computSpeed, covering control flow"
USES {PLATOONTESTLIB}
INPUT VARIABLES
  lastpos:Integer;
  vspeed:Integer;
  safeDistance:Integer;
  pilotpos:Integer;
  pilotspeed:Integer;
OUTPUT VARIABLES
  speed:Integer;
  oracledata:Integer;
ORACLE
  speed=oracledata
```

- 1 tester un service offert
- 2 déterminer le contexte
- 3 brancher le contexte sur les données

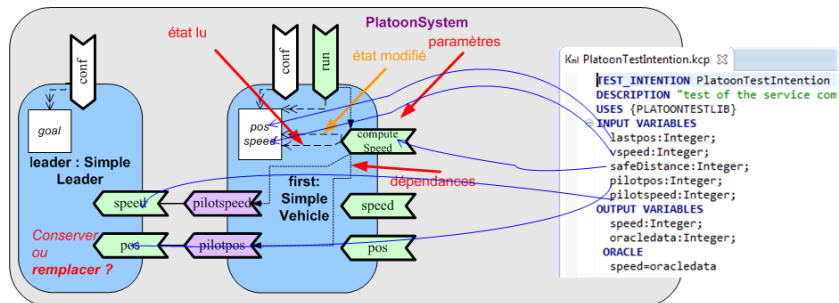


# CostTest - fixer le périmètre 1/4



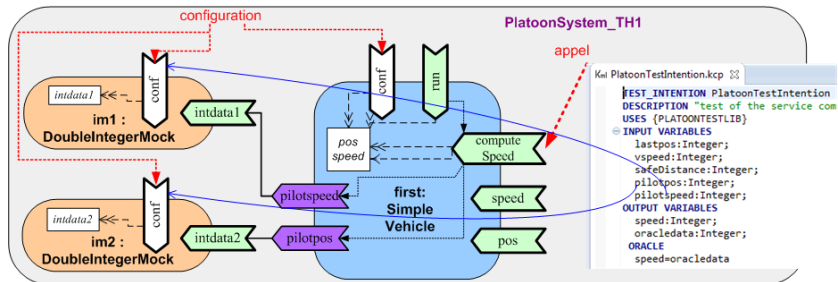
- 1 fixer le périmètre
- 2 lier à l'intention
- 3 cohérence, complétude

# CostTest - lier à l'intention 2/4



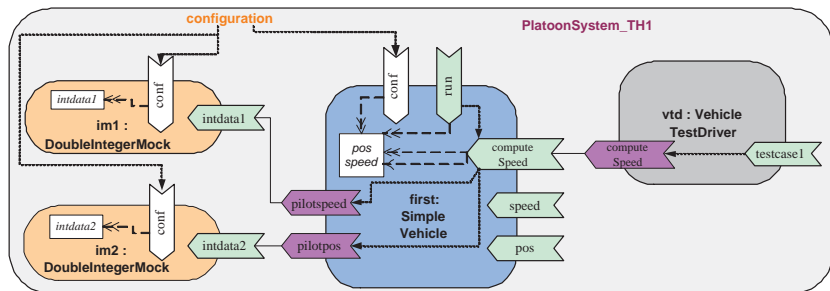
- 1 paramètres, variables d'états (init, accesseurs), dépendances
- 2 convergence intention / application de test (pourcentages)
- 3 cohérence, complétude

# CostTest - satisfaire les besoins 3/4



- 1 fournisseurs de services requis
- 2 itérer sur l'étape 2/4 :  
paramètres, variables d'états (init, accesseurs), dépendances
- 3 convergence intention / application de test
- 4 cohérence, complétude (pourcentages)

# CostTest - générer le testeur 4/4

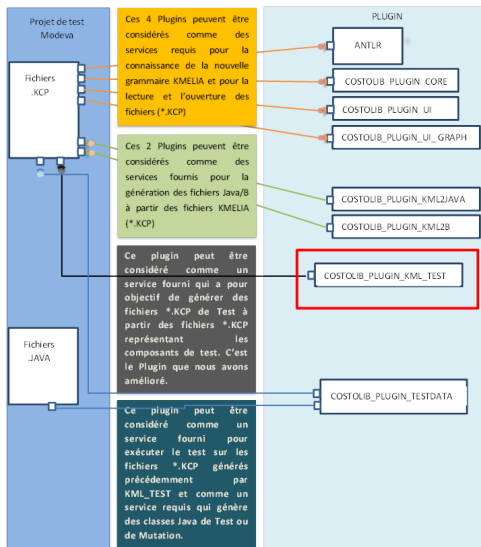


- 1 séquence d'initialisation
- 2 service de lancement du cas de test
- 3 application de test (composants, services, liens, configurations)

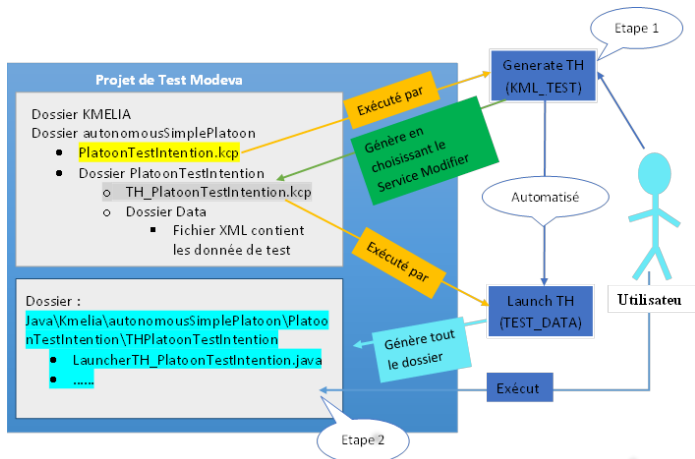
# Plan

- 1 Introduction
- 2 Processus illustré
- 3 Implantation**
- 4 Bilan et perspectives

# CostTest- Architecture



# CostTest - Processus



# CostTest - SUT

Java - CaseStudyCostoTest/kmela/autonomousSimplePlatoon/PlatoonSystem.kcp - Eclipse SDK

File Edit Navigate Search Project Run Component Extractor Window Help

KmI2Java KmI2Lates KmI2MEC4 Mutants sysout makeAGraph KmI2B Generate THOld Generate TH Generate TH M2Alma Launch TH MutationTH

Package Explorer

- CaseStudyCostoTest
  - src
    - mylib
      - PlatoonlibMap.java
      - PlatoonTestlibMap.java
    - JRE System Library [JavaSE-1.6]
    - Referenced Libraries
      - costo.kmI2java.jar
      - jdom-2.0.5.jar
    - kmela
      - autonomousSimplePlatoon
        - PLATOONLIB
          - KeI constants.kic
          - KeI functions.kif
          - PLATOONLIB.javamapping
          - KeI types.kit
        - PlatoonTestIntention
          - data
        - PLATOONTESTLIB
          - KeI functions.kif
          - PLATOONTESTLIB.javamapping
        - tests
          - KeI IntegerMock.kcp
          - KeI PlatoonSystem.kcp
          - KeI PlatoonTestIntention.kcp
          - KeI SimpleDriver.kcp
          - KeI SimpleVehicle.kcp
      - lib
        - collections-generic-4.01.jar
        - costo.kmI2java.jar
        - costo.kmI2java.monitor.jar
        - jdom-2.0.5.jar
        - jung-algorithms-2.0.1.jar
        - jung-api-2.0.1.jar
        - jung-graph-impl-2.0.1.jar
        - jung-visualization-2.0.1.jar
        - testdata-sequences.txt
        - testdata.txt
        - CaseStudyModev1.3

Task List

Find  All  Activate...

Uncategorized

Connect Mylyn

Connect to your task and ALM tools.

Outline

An outline is not available.

Problems  Javadoc  Declaration  Progress

No operations to display at this time.

Writable Insert 1:1

UNIVERSITÉ DE NANTES



# CostTest - TI

Java - CaseStudyCostoTest\kmlia\autonomousSimplePlatoon\PlatoonTestIntention.kcp - Eclipse SDK

File Edit Navigate Search Project Run Component Extractor Window Help

Package Explorer

- CaseStudyCostoTest
  - src
    - mylib
      - PlatoonlibMap.java
      - PlatoonTestlibMap.java
    - JRE System Library [JavaSE-1.6]
    - Referenced Libraries
      - costo.kml2java.jar
      - jdkom-2.0.5.jar
    - kmlia
      - autonomousSimplePlatoon
        - PLATOONLIB
          - Kai constants.klc
          - Kai functions.klf
          - PLATOONLIB.javamapping
          - Kai types.kit
        - PlatoonTestIntention
          - data
            - data\_res.xml
            - data.xml
            - TH\_PlatoonTestIntention.info
            - TH\_PlatoonTestIntention.properties
          - PLATOONTESTLIB
            - Kai functions.klf
            - PLATOONTESTLIB.javamapping
          - tests
            - Kai IntegerMock.kcp
            - Kai PlatoonSystem.kcp
            - Kai PlatoonTestIntention.kcp
            - Kai SimpleDriver.kcp
            - Kai SimpleVehicle.kcp
    - lib
      - collections-generic-4.01.jar
      - costo.kml2java.jar
      - costo.kml2java.monitor.jar
      - jdkom-2.0.5.jar
      - jung-algorithms-2.0.1.jar
      - jung-api-2.0.1.jar
      - jung-qrash-impl-2.0.1.jar

Kai PlatoonSystem.kcp Kai PlatoonTestIntention.kcp

```

TEST_INTENTION PlatoonTest: Intention
DESCRIPTION "the vehicle will stop if it is too close to the previous or
INPUT VARIABLES
  pilotspeed: Integer;
  pilotpos: Integer;
  lastpos: Integer;
  vspeed: Integer;
  safeDistance: Integer;
  vname: String
OUTPUT VARIABLES
  speed: Integer
ORACLE
  speed=0
  
```

Task List

Find All Activate...

Uncategorized

Connect Mylyn  
Connect to your task and ALM tools.

Outline  
An outline is not available.

Problems Javadoc Declaration Progress

No operations to display at this time.

Writable Insert 1:1

# CostTest - Intention

Test Harness Creating Process

**Harness**

**Test Intention**

Input Variables		
Variable	Mapping	Controlled
pilotpos : Integer	NONE	<input type="checkbox"/>
pilotspeed : Int...	NONE	<input type="checkbox"/>
safeDistance : ...	NONE	<input type="checkbox"/>
vspeed : Integer	NONE	<input type="checkbox"/>
lastpos : Integer	NONE	<input type="checkbox"/>
vname : String	NONE	<input type="checkbox"/>

Output Variables		
Variable	Mapping	Controlled
speed : Integer	NONE	<input type="checkbox"/>

**Oracle Predicate**

[speed = 0]

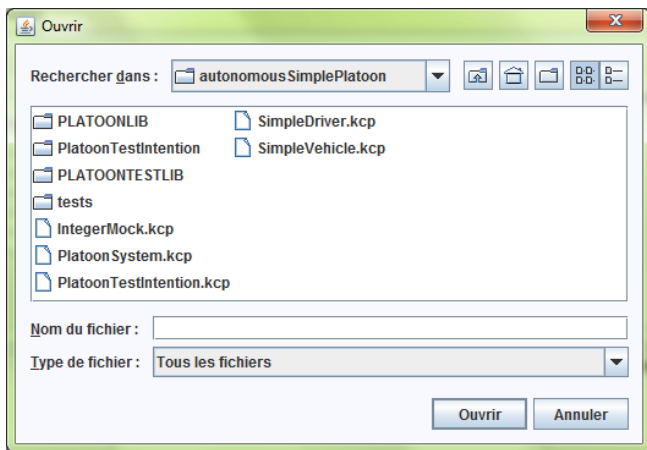
**Construction State**

0 %

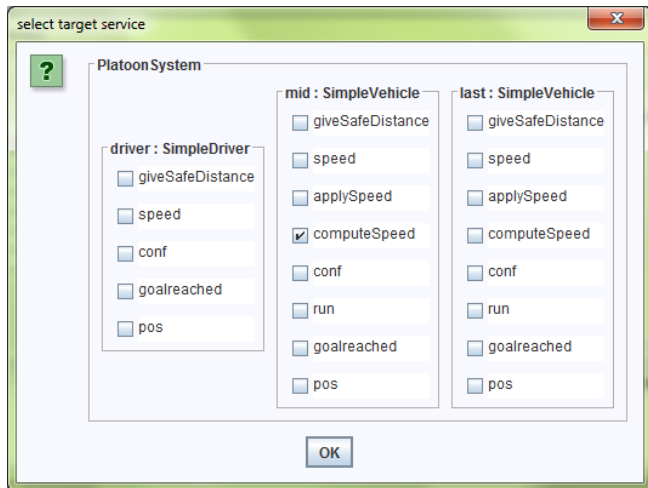
**Test Perimeter**

□

# CostTest - Sélection du système



# CostTest - Sélection des services cibles



# CostTest - convergence

Test Harness Creating Process

**Harness**

**Test Intention**

Input Variables			Output Variables			Oracle Predicate
Variable	Mapping	Controlled	Variable	Mapping	Controlled	
pilotpos : Integer	NONE	<input type="checkbox"/>	speed : Integer	NONE	<input type="checkbox"/>	[speed = 0]
pilotspeed : Int...	NONE	<input type="checkbox"/>				
safeDistance : ...	NONE	<input type="checkbox"/>				
vspeed : Integer	NONE	<input type="checkbox"/>				
lastpos : Integer	NONE	<input type="checkbox"/>				
vname : String	NONE	<input type="checkbox"/>				

**mid\_cut**

Variable assignment for service SimpleVehicle

**Component State**

Variable State	Service Modifier
unassigned lastpos : Integer	assign Setter
unassigned vname : String	
unassigned vspeed : Integer	

**All Provided Services**

computeSpeed Inputs Parameters	unassigned safeDistance : Integer
computeSpeed Outputs Parameters ( Result : Integer )	unassigned Result : Integer

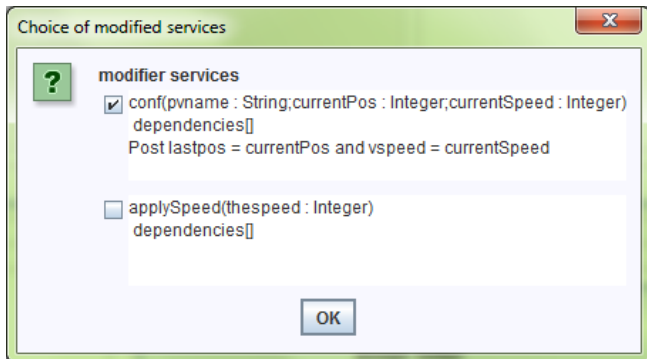
**All Required Services**

unassigned pilotpos : Integer	unassigned
unassigned pilotspeed : Integer	unassigned

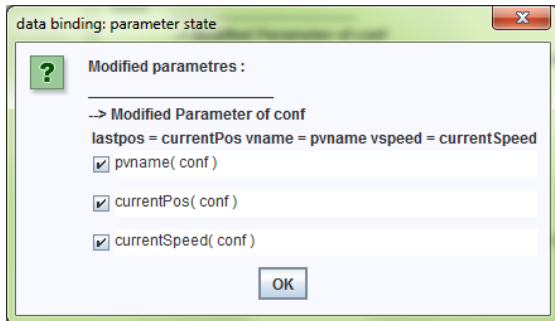
**Construction State**  
0 %

**Test Perimeter**  
mid\_cut : SimpleVehicle[computeSpeed]

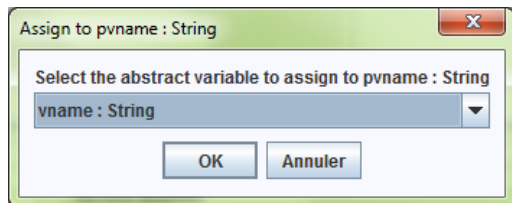
# CostTest - assignSetter



# CostTest - positionnement état



# CostTest - identification intention état





# CostTest - identification intention état

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	NONE	<input type="checkbox"/>
pilotspeed : Int...	NONE	<input type="checkbox"/>
safeDistance : ...	NONE	<input type="checkbox"/>
vspeed : Integer	NONE	<input type="checkbox"/>
lastpos : Integer	NONE	<input type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	NONE	<input type="checkbox"/>

**Oracle Predicate**

[speed = 0]

**mid\_cut**

Variable assignment for service SimpleVehicle

**Component State**

**Variable State**

vname	pvname : String
unassigned	currentPos : Integer
unassigned	currentSpeed : Integer

**assign Setter**

**All Provided Services**

**computeSpeed Inputs Parameters**

unassigned	safeDistance : Integer
------------	------------------------

**computeSpeed Outputs Parameters ( Result : Integer )**

unassigned	Result : Integer
------------	------------------

**All Required Services**

unassigned	pilotpos : Integer	unassigned
unassigned	pilotspeed : Integer	unassigned

**Construction State**

Oracles: [0.0 %]  
mid\_cut: [16.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

**Assign to currentPos : Integer**

Select the abstract variable to assign to currentPos : Integer

lastpos : Integer

OK Annuler

# CostTest - identification intention paramètres

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	NONE	<input type="checkbox"/>
pilotspeed : Int...	NONE	<input type="checkbox"/>
safeDistance : ...	NONE	<input type="checkbox"/>
vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	NONE	<input type="checkbox"/>

**Oracle Predicate**

[speed = 0]

**mid\_cut**

Variable assignment for service SimpleVehicle

Component State

Variable State

vname	pname : String
lastpos	currentPos : Integer
vspeed	currentSpeed : Integer

assign Setter

**All Provided Services**

computeSpeed Inputs Parameters

unassigned	safeDistance : Integer
------------	------------------------

computeSpeed Outputs Parameters ( Result : Integer )

unassigned	Result : Integer
------------	------------------

**All Required Services**

unassigned	pilotpos : Integer	unassigned
unassigned	pilotspeed : Integer	unassigned

**Construction State**

Oracles: [0.0 %]  
mid\_cut: [50.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

Assign to safeDistance : Integer

Select the abstract variable to assign to safeDistance : Integer

safeDistance : Integer

OK Annuler

# CostTest - identification intention résultat

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	NONE	<input type="checkbox"/>
pilotspeed : Int...	NONE	<input type="checkbox"/>
safeDistance : ...	PARAMETER	<input checked="" type="checkbox"/>
vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	NONE	<input type="checkbox"/>

**Oracle Predicate**

[speed = 0]

**mid\_cut**

Variable assignment for service SimpleVehicle

**Component State**

**Variable State**

vname	pvname : String
lastpos	currentPos : Integer
vspeed	currentSpeed : Integer

**Service Modifier**

assign Setter

**All Provided Services**

**computeSpeed Inputs Parameters**

safeDistance	safeDistance : Integer
--------------	------------------------

**computeSpeed Outputs Parameters ( Result : Integer )**

unassigned	Result : Integer
------------	------------------

**All Required Services**

unassigned	pilotpos : Integer	unassigned
unassigned	pilotspeed : Integer	unassigned

**Construction State**

Oracles: [0.0 %]  
mid\_cut: [66.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

**Assign to Result : Integer**

Select the abstract variable to assign to Result : Integer

speed : Integer

OK Annuler

# CostTest - satisfaction des besoins - requis 1

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	NONE	<input type="checkbox"/>
pilotspeed : Int...	NONE	<input type="checkbox"/>
safeDistance : ...	PARAMETER	<input checked="" type="checkbox"/>
vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	STATEVAR	<input checked="" type="checkbox"/>

**Oracle Predicate**

[speed = 0]

**mid\_cut**

Variable assignment for service SimpleVehicle

**Component State**

**Variable State**

vname	pvname : String
lastpos	currentPos : Integer
vspeed	currentSpeed : Integer

**Service Modifier**

assign Setter

**assign provided service**

driver.pos

IntegerMock.IntegerMock67

OK

**All Provided Services**

**computeSpeed Inputs Parameters**

safeDistance	safeDistance : Integer
--------------	------------------------

**computeSpeed Outputs Parameters ( Result : Integer )**

speed	Result : Integer
-------	------------------

**All Required Services**

unassigned	pilotpos : Integer	unassigned
unassigned	pilotspeed : Integer	unassigned

**Construction State**

Oracles: [100.0 %]  
mid\_cut: [66.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

# CostTest - identification intention et requis 1

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	NONE	<input type="checkbox"/>
pilotspeed : Int...	NONE	<input type="checkbox"/>
safeDistance : ...	PARAMETER	<input checked="" type="checkbox"/>
vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	STATEVAR	<input checked="" type="checkbox"/>

**Oracle Predicate**

[speed = 0]

---

mid\_cut IntegerMock67

Variable assignment for service IntegerMock

**Component State**

unassigned

**All Provided Services**

value Parameters

unassigned      Result : Integer

**All Required Services**

**Construction State**

Oracles: [100.0 %]  
mid\_cut: [83.0 %]  
IntegerMock67: [0.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

Assign to value : Integer

Select the abstract variable to assign to value : Integer

pilotpos : Integer

OK    Annuler

# CostTest - identification intention et requis 2

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
pilotspeed : Int...	PARAMETER	<input checked="" type="checkbox"/>
safeDistance : ...	PARAMETER	<input checked="" type="checkbox"/>
vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	STATEVAR	<input checked="" type="checkbox"/>

**Oracle Predicate**

[speed = 0]

mid\_cut IntegerMock67 IntegerMock68

Variable assignment for service IntegerMock

**Component State**

pilotspeed value : Integer

**All Provided Services**

**value Parameters**

unassigned Result : Integer

**All Required Services**

**Construction State**

Oracles : [100.0 %]  
mid\_cut : [100.0 %]  
IntegerMock67 : [100.0 %]  
IntegerMock68 : [100.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

# CostTest - contexte du harnais établi

Test Harness Creating Process

**Harness**

**Test Intention**

**Input Variables**

Variable	Mapping	Controlled
pilotpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
pilotspeed : Int...	PARAMETER	<input checked="" type="checkbox"/>
safeDistance : ...	PARAMETER	<input checked="" type="checkbox"/>
vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

**Output Variables**

Variable	Mapping	Controlled
speed : Integer	STATEVAR	<input checked="" type="checkbox"/>

**Oracle Predicate**

[speed = 0]

mid\_cut IntegerMock67 IntegerMock68

**Variable assignment for service SimpleVehicle**

**Component State**

**Variable State**

vname	pvname : String
lastpos	currentPos : Integer
vspeed	currentSpeed : Integer

**Service Modifier**

assign Setter

**All Provided Services**

**computeSpeed Inputs Parameters**

safeDistance	safeDistance : Integer
--------------	------------------------

**computeSpeed Outputs Parameters ( Result : Integer )**

speed	Result : Integer
-------	------------------

**All Required Services**

unassigned	pilotpos : Integer	IntegerMock.IntegerMock67
unassigned	pilotspeed : Integer	IntegerMock.IntegerMock68

**Construction State**

Oracles : [100.0 %]  
mid\_cut : [100.0 %]  
IntegerMock67 : [100.0 %]  
IntegerMock68 : [100.0 %]

**Test Perimeter**

mid\_cut : SimpleVehicle[computeSpeed]

# CostTest - Génération du harnais

Test Harness Creating Process

**Harness**

Select Target test Alt-S

Check concrete Mappings Alt-C

Generate and Launch TH Alt-G    Controlled

Variable	Mapping	Controlled
speed : Integer	STATEVAR	<input checked="" type="checkbox"/>

Oracle Predicate  
[speed = 0]

Generate Alt-G   

quit Alt-Q   

vspeed : Integer	PARAMETER	<input checked="" type="checkbox"/>
lastpos : Integer	PARAMETER	<input checked="" type="checkbox"/>
vname : String	PARAMETER	<input checked="" type="checkbox"/>

mid\_cut    IntegerMock67    IntegerMock68

Variable assignment for service SimpleVehicle

Component State

Variable State

vname	pvname : String
lastpos	currentPos : Integer
vspeed	currentSpeed : Integer

Service Modifier

assign Setter

All Provided Services

computeSpeed Inputs Parameters

safeDistance	safeDistance : Integer
--------------	------------------------

computeSpeed Outputs Parameters ( Result : Integer )

speed	Result : Integer
-------	------------------

All Required Services

unassigned	pilotpos : Integer	IntegerMock.IntegerMock67
unassigned	pilotspeed : Integer	IntegerMock.IntegerMock68

Construction State

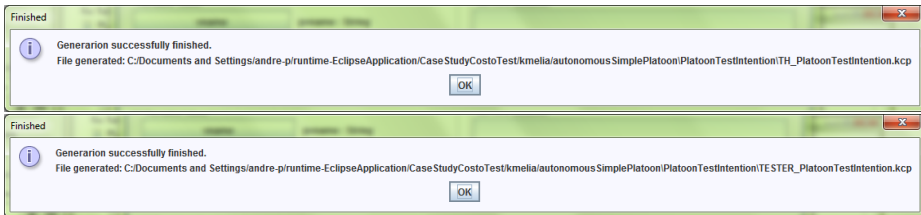
Oracles: [100.0 %]  
mid\_cut [100.0 %]  
IntegerMock67: [100.0 %]  
IntegerMock68: [100.0 %]

Test Perimeter

mid\_cut : SimpleVehicle[computeSpeed]

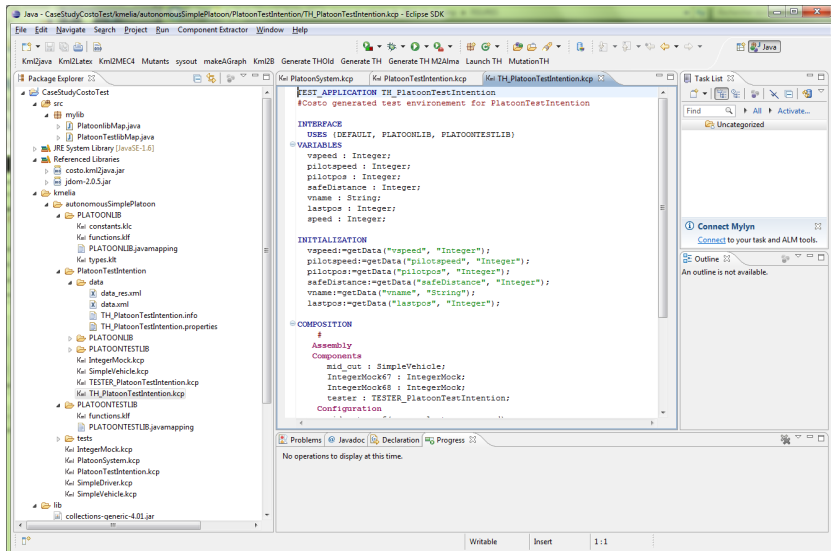


# CostTest - Génération du harnais



mutation possible

# CostTest - Application de test



The screenshot shows the Eclipse IDE interface for a Java project named 'CaseStudyCostoTest'. The Package Explorer on the left displays the project structure, including packages like 'PLATOONLIB' and 'TH\_PlatoonTestIntention'. The main editor window shows the source code for 'TH\_PlatoonTestIntention.kcp', which defines an interface with variables and initialization methods. The bottom status bar indicates 'No operations to display at this time.'

```

TEST_APPLICATION TH_PlatoonTestIntention
#Costo generated test environment for PlatoonTestIntention

INTERFACE
USES (DEFAULT, PLATOONLIB, PLATOONTESTLIB)
@VARIABLES
vspeed : Integer;
pilotspeed : Integer;
pilotpos : Integer;
safedistance : Integer;
vname : String;
lastpos : Integer;
speed : Integer;

INITIALIZATION
vspeed:=getData("vspeed", "Integer");
pilotspeed:=getData("pilotspeed", "Integer");
pilotpos:=getData("pilotpos", "Integer");
safedistance:=getData("safedistance", "Integer");
vname:=getData("vname", "String");
lastpos:=getData("lastpos", "Integer");

@COMPOSITION
#
Assembly
Components
mid_cut : SimpleVehicle;
IntegerMock67 : IntegerMock;
IntegerMock68 : IntegerMock;
tester : TESTER_PlatoonTestIntention;
Configuration
    
```

# CostTest - Génération du code et lien aux données

The screenshot shows an IDE window with a Package Explorer on the left and a code editor on the right. The Package Explorer shows a project structure for 'CaseStudyCostoTest' with various source files and libraries. The code editor displays the generated code for 'TH\_PlatoonTestIntention.kcp', which includes an interface definition, initialization code, and component configuration. A dialog box is overlaid on the code editor, asking: "Data file already exist Do you want to use this one for test process ?" with 'OK' and 'NO' buttons.

```

TEST_APPLICATION TH_PlatoonTestIntention
#Costo generated test environment for PlatoonTestIntention

INTERFACE
  USES (DEFAULT, PLATOONLIB, PLATOONTESTLIB)
  VARIABLES
    vspeed : Integer;
    pilotspeed : Integer;
    pilotcpo : Integer;
    safeDistance : Integer;
    vname : String;
    lastcpo : Integer;
    speed : Integer;

  INITIALIZATION
    vspeed:=getData("vspeed", "Integer");
  PRE-CONFIGURATION
  COMPONENTS
    mid_cut : SimpleVehicle;
    IntegerMock67 : IntegerMock;
    IntegerMock68 : IntegerMock;
    tester : TESTER_PlatoonTestIntention;
  Configuration
  
```

Dialog box text: "Data file already exist Do you want to use this one for test process ?" (OK, NO)

# CostTest - Génération du code et lien aux données (console)

```

Java - Eclipse
File Edit Navigate Search Project Run Component Extractor Window Help
Eclipse Application [Eclipse Application] C:\Program Files\Java\jre7\bin\java.exe [13 janv. 2015 10:43:03]
main.VMainParam : vname
ParamModifier : currentPos
transition : initV3-display(vname * * is initialized at currentPos: " + currentPos + " speed: " + currentSpeed)-->initVf
this.comParam : lastPos
ParamModifier : currentSpeed
transition : initV3-display(vname * * is initialized at currentPos: " + currentPos + " speed: " + currentSpeed)-->initVf
this.comParam : vname
ParamModifier : currentSpeed
transition : initV3-display(vname * * is initialized at currentPos: " + currentPos + " speed: " + currentSpeed)-->initVf
this.comParam : vspeed
ParamModifier : currentSpeed
3 = 2
VarFinal [vname : String, currentPos : Integer, currentSpeed : Integer]
[vname : String, currentPos : Integer, currentSpeed : Integer]
Tut targ type Integer
tHook addprovided ****
hide called
tHook addprovided ****
hide called
** Setting all links Harness **
Checking mid_out
* All parameters *
Varvname
VarcurrentPos
VarcurrentSpeed
* All required Links *
* All required Links *
* All required Links *
Checking IntegerHook67
* All parameters *
* All provided *
Varvalue
* tHOOK All required Links *
* tHOOK All required Links *
Checking IntegerHook68
* All parameters *
* All provided *
Varvalue
* tHOOK All required Links *
* tHOOK All required Links *
path:C:\Documents and Settings\andre-p\runtime-EclipseApplication\CaseStudyCostoTest\kmla\autonomousSimplePlatoon\PlatoonTestIntention\IntegerHook.ksp
path:C:\Documents and Settings\andre-p\runtime-EclipseApplication\CaseStudyCostoTest\kmla\autonomousSimplePlatoon\PlatoonTestIntention\IntegerHook.ksp
***** generateVariables START *****
loading C:\Documents and Settings\andre-p\runtime-EclipseApplication\CaseStudyCostoTest\kmla\autonomousSimplePlatoon\PlatoonTestIntention\PLATOONLIB\PLATOONLIB_javanapping
loading C:\Documents and Settings\andre-p\runtime-EclipseApplication\CaseStudyCostoTest\kmla\autonomousSimplePlatoon\PlatoonTestIntention\PLATOONTESTLIB\PLATOONTESTLIB_javanapping
assertT(verdict)costo.kml2java.export.SimpleReplacementWithThisIfTiff79
GENERATING MUTANT FILE
*

```

# CostTest - code généré

The screenshot displays the Eclipse IDE interface. The main editor shows the source code for `LauncherTH_PlatoonTestIntention.java`. The code is as follows:

```

package kmelia.autonomousSimplePlatoon.PlatoonTestIntention.TH.PlatoonTestIntention;

import java.util.Iterator;

public class LauncherTH_PlatoonTestIntention {
    static Document doc;
    static Element racine;
    static SAXBuilder sxb;
    static List<Element> tcase;
    static String dataFilePath = "C:/Documents and Settings/andre

    public LauncherTH_PlatoonTestIntention() {
    }

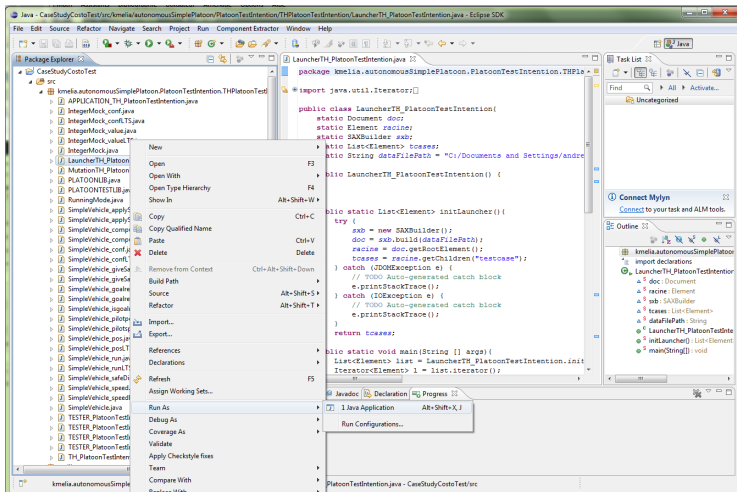
    public static List<Element> initLauncher() {
        try {
            sxb = new SAXBuilder();
            doc = sxb.build(dataFilePath);
            racine = doc.getRootElement();
            tcase = racine.getChildren("testcase");
        } catch (JDOMException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return tcase;
    }

    public static void main(String [] args) {
        List<Element> list = LauncherTH_PlatoonTestIntention.init
        Iterator<Element> l = list.iterator();
    }
}

```

The Package Explorer on the left shows the project structure, including the package `kmelia.autonomousSimplePlatoon.PlatoonTestIntention.TH.PlatoonTestIntention`. The Outline view on the right shows the class hierarchy, including `LauncherTH_PlatoonTestIntention` and its methods.

# CostTest - exécution du test



# CostTest - exécution du test (lancement manuel)

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays a project structure for 'krmelia.autonomousSimplePlatoon.PlatoonTestIntention'. The main editor shows the source code for 'MutationTH\_PlatoonTestIntention.kcp', which is a Costo-generated test environment. The code includes an interface with variables (lastpos, vname, vspeed, safeDistance, pilotspeed, pilotpos, speed) and initialization logic. The console at the bottom shows the execution of a test case, resulting in the message: 'Vehicle vehicle15 gets pilotspeed125' and '>>TEST<< TESTER\_PlatoonTestIntention: testcase1 verdict is false'.

```

#Costo generated test environment for Platoon

INTERFACE
USES (DEFAULT, PLATOONLIB, PLATOONTESTLIB)
@VARIABLES
lastpos : Integer;
vname : String;
vspeed : Integer;
safeDistance : Integer;
pilotspeed : Integer;
pilotpos : Integer;
speed : Integer;

INITIALIZATION
lastpos:=getData("lastpos", "Integer");
vname:=getData("vname", "String");
vspeed:=getData("vspeed", "Integer");
safeDistance:=getData("safeDistance", "Integer");
pilotspeed:=getData("pilotspeed", "Integer");
pilotpos:=getData("pilotpos", "Integer");

@COMPOSITION
#
Assembly
Components
    
```

```

MutationTH_PlatoonTestIntention [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13 janv. 2015 21:00:16)

Vehicle vehicle15 gets pilotspeed125
>>TEST<< TESTER_PlatoonTestIntention: testcase1 verdict is false
    
```

# CostTest - exécution du test (console)

```

MutationTH_PlatoonTestIntention [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13 janv. 2015 21:00:16)

Mutant Launcher
Mutant: mutantD1r:
vehicule1 is initialised at currentpos: 27 speed: 121
Vehicle vehicule1 starts
computeSpeed called with safeDistance: 72
Vehicle vehicule1 has lastpos: 27 and vspeed: 121
Vehicle vehicule1 gets pilotspeed00
Vehicle vehicule1 gets pilotspeed130
>>TEST<< TESTER_PlatoonTestIntention::testcase1 verdict is false

vehicule2 is initialised at currentpos: 5 speed: 130
Vehicle vehicule2 starts
computeSpeed called with safeDistance: 72
Vehicle vehicule2 has lastpos: 5 and vspeed: 130
Vehicle vehicule2 gets pilotspe78
Vehicle vehicule2 gets pilotspeed130
>>TEST<< TESTER_PlatoonTestIntention::testcase1 verdict is false

vehicule3 is initialised at currentpos: 0 speed: 120
Vehicle vehicule3 starts
computeSpeed called with safeDistance: 72
Vehicle vehicule3 has lastpos: 0 and vspeed: 120
Vehicle vehicule3 gets pilotspe73
Vehicle vehicule3 gets pilotspeed130
>>TEST<< TESTER_PlatoonTestIntention::testcase1 verdict is false

vehicule4 is initialised at currentpos: 30 speed: 0
Vehicle vehicule4 starts
computeSpeed called with safeDistance: 69
Vehicle vehicule4 has lastpos: 30 and vspeed: 0
Vehicle vehicule4 gets pilotspe100
Vehicle vehicule4 gets pilotspeed115
>>TEST<< TESTER_PlatoonTestIntention::testcase1 verdict is false

vehicule5 is initialised at currentpos: 8 speed: 100
Vehicle vehicule5 starts
computeSpeed called with safeDistance: 69
Vehicle vehicule5 has lastpos: 8 and vspeed: 100
Vehicle vehicule5 gets pilotspe78
Vehicle vehicule5 gets pilotspeed125
>>TEST<< TESTER_PlatoonTestIntention::testcase1 verdict is false

vehicule6 is initialised at currentpos: 0 speed: 5
Vehicle vehicule6 starts
computeSpeed called with safeDistance: 69
Vehicle vehicule6 has lastpos: 0 and vspeed: 5
Vehicle vehicule6 gets pilotspe73
Vehicle vehicule6 gets pilotspeed125
>>TEST<< TESTER_PlatoonTestIntention::testcase1 verdict is false

```



# CostTest - résultat du test

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows the project structure for 'CaseStudyModeval3'. The main editor displays the XML file 'data\_res.xml' with the following content:

```

<?xml version="1.0" encoding="UTF-8"?>
<testHarness nbTestCases="48">
  <testcase>
    <id>4</id>
    <oracledata>130</oracledata>
    <pilotasped>130</pilotasped>
    <pilotpos>100</pilotpos>
    <safeDistance>72</safeDistance>
    <vapeed>121</vapeed>
    <vname>vehicule1</vname>
    <lastpos>27</lastpos>
    <verdict>false</verdict>
  </testcase>
  <testcase>
    <id>2</id>
    <oracledata>130</oracledata>
    <pilotasped>130</pilotasped>
    <pilotpos>78</pilotpos>
    <safeDistance>72</safeDistance>
    <vapeed>130</vapeed>
    <vname>vehicule2</vname>
    <lastpos>5</lastpos>
    <verdict>false</verdict>
  </testcase>
  <testcase>
    <id>9</id>
    <oracledata>130</oracledata>
    <pilotasped>130</pilotasped>
    <pilotpos>73</pilotpos>
    <safeDistance>72</safeDistance>
    <vapeed>120</vapeed>
    <vname>vehicule3</vname>
    <lastpos>0</lastpos>
    <verdict>false</verdict>
  </testcase>
  <testcase>
    <id>4</id>
    <oracledata>10</oracledata>
  </testcase>
</testHarness>
  
```

The bottom console shows the output of the mutant launcher:

```

MutationTH_PlatoonTestIntention [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (13 janv. 2015 21:00:16)
Mutant Launcher
Mutant: mutantDir:
vehicule1 is initialised at currentpos: 27 speed: 121
Vehicle vehicule1 starts
computeSpeed called with safeDistance: 72
Vehicle vehicule1 has lastpos: 27 and vaped: 121
  
```

# Plan

- 1 Introduction
- 2 Processus illustré
- 3 Implantation
- 4 Bilan et perspectives**

# CostTest - Historique

- 1 V1 (GA) : fonctions d'assistance, début construction, API de base
- 2 V2 (M1/Alma) : construction, exécution, documentation wiki
  - <http://www.lina.sciences.univ-nantes.fr/aelos/wiki/doku.php/cost:test:userguide:costotestuginstall>
  - <http://www.lina.sciences.univ-nantes.fr/aelos/wiki/doku.php/cost:test:results:start>
- 3 V3 (M2/Alma) : intégration poursuivie, documentation web
- 4 ...

forges : CIE, Lina

# CostTest - En cours

- 1 Maintenance des plugins COSTO
  - Builder
  - Interface
  - Refactoring
- 2 Intégration des plugins CostoTest
  - Installer
  - Console améliorée
  - Gestion des données de test
- 3 Expérimentations
  - comparaison xUnit (Alma ?)
  - études de cas
- 4 Documentation wiki
  - site Web, wiki
  - Forges
- 5 Evolutions CostoTest
  - Mutation
  - Gestion de bancs de test
  - Méthodologie