

Intégration de propriétés PSL dans des spécifications hétérogènes

Mattis LE FALHUN
Ossama MOUSTAFA



SOMMAIRE

I. CONCEPTS, OUTILS, ET MÉTHODES UTILISÉS

1.1 PSL

1.2 SysML

1.3 Modelio

1.4 Méthodologie

II. PROPOSITION

2.1 Systèmes hétérogènes

2.2 Intégration de PSL

2.3 Déductions et ajouts de propriétés PSL

III. EXPÉRIMENTATION

3.1 Traductions SysML-PSL

3.2 Application à un système

IV. BILAN

PLAN

I. CONCEPTS, OUTILS, ET MÉTHODES UTILISÉS

- 1.1 PSL
- 1.2 SysML
- 1.3 Modelio
- 1.4 Méthodologie

II. PROPOSITION

- 2.1 Systèmes hétérogènes
- 2.2 Intégration de PSL
- 2.3 Déductions et ajouts de propriétés PSL

III. EXPÉRIMENTATION

- 3.1 Traductions SysML-PSL
- 3.2 Application à un système

IV. BILAN

1.1 PSL

- ▶ **PSL** : Property Specification Language



1.2 SysML

- ▶ PSL : Property Specification Language
- ▶ **SysML** : Systems Modeling Language (UML 3.0)



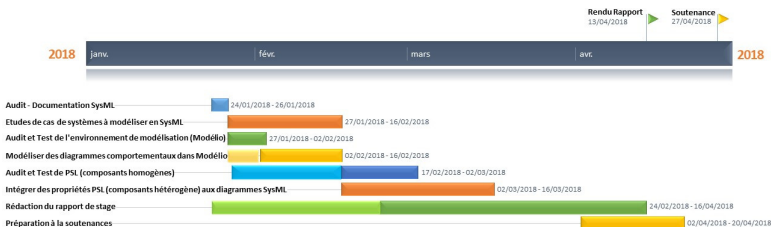
1.3 MODELIO

- ▶ PSL : Property Specification Language
- ▶ SysML : Systems Modeling Language (UML 3.0)
- ▶ **Modelio** : Outil de modélisation SysML



1.4 MÉTHODOLOGIE

- ▶ PSL : Property Specification Language
- ▶ SysML : Systems Modeling Language (UML 3.0)
- ▶ Modelio : outil de modélisation SysML.
- ▶ **Méthodologie**



PLAN

I. CONCEPTS, OUTILS, ET MÉTHODES UTILISÉS

- 1.1 PSL
- 1.2 SysML
- 1.3 Modelio
- 1.4 Méthodologie

II. PROPOSITION

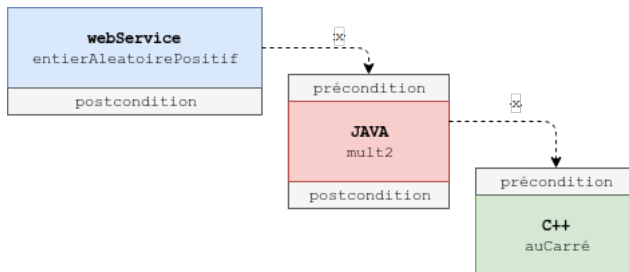
- 2.1 Systèmes hétérogènes
- 2.2 Intégration de PSL
- 2.3 Déductions et ajouts de propriétés PSL

III. EXPÉRIMENTATION

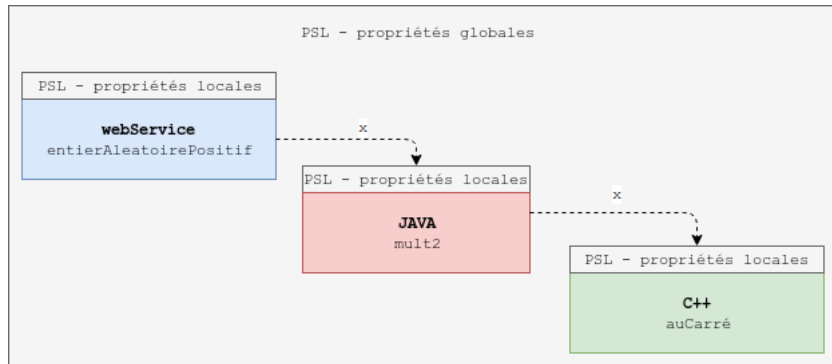
- 3.1 Traductions SysML-PSL
- 3.2 Application à un système

IV. BILAN

2.1 SYSTÈMES HÉTÉROGÈNES

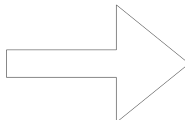
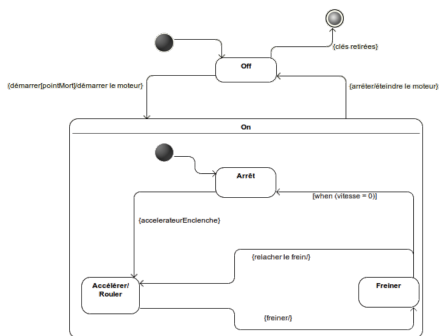


2.2 INTÉGRATION DE PSL



2.3 DÉDUCTIONS ET AJOUTS DE PROPRIÉTÉS PSL

- ▶ Concerne les propriétés **locales**



```
assert always (...);  
assert always (...);  
assert always (...);  
assert always (...);
```

PLAN

I. CONCEPTS, OUTILS, ET MÉTHODES UTILISÉS

- 1.1 PSL
- 1.2 SysML
- 1.3 Modelio
- 1.4 Méthodologie

II. PROPOSITION

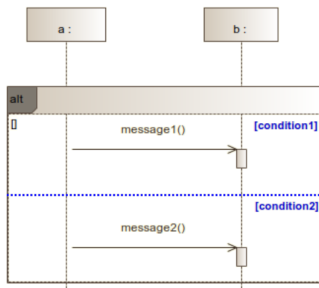
- 2.1 Systèmes hétérogènes
- 2.2 Intégration de PSL
- 2.3 Dédutions et ajouts de propriétés PSL

III. EXPÉRIMENTATION

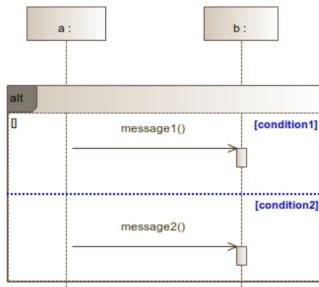
- 3.1 Traductions SysML-PSL
- 3.2 Application à un système

IV. BILAN

3.1 TRADUCTIONS SYSML-PSL

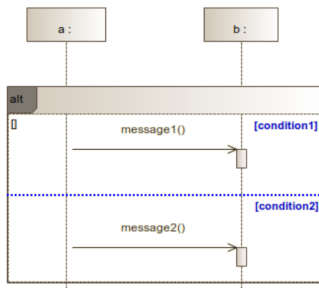


3.1 TRADUCTIONS SYSML-PSL



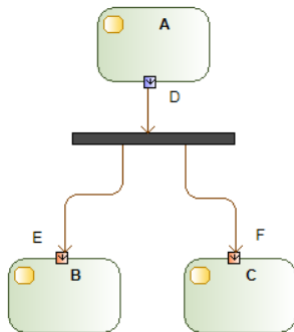
- ▶ PSL du module a : `assert always ((condition1 → next(message1)) & (condition2 → next(message2)));`

3.1 TRADUCTIONS SYSML-PSL

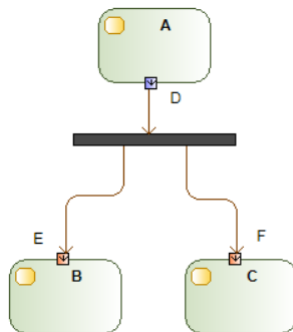


- ▶ PSL du module a : `assert always ((condition1 → next(message1)) & (condition2 → next(message2)))`;
- ▶ PSL du module b : `assert always (message1 → next(!message2) & message2 → next(!message1))`;

3.1 TRADUCTIONS SYSML-PSL

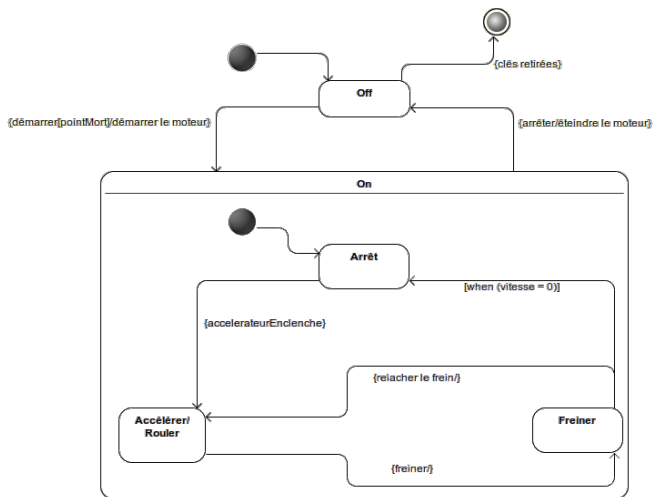


3.1 TRADUCTIONS SYSML-PSL



- ▶ Exemple : D doit respecter E et F.
- ▶ `assert always (D → next(multiple-de-6));`
`assert always (E → next(multiple-de-3));`
`assert always (F → next(multiple-de-2));`

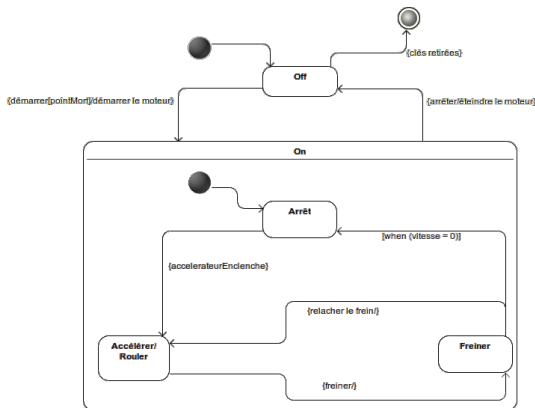
3.2 APPLICATION À UN SYSTÈME



3.2 APPLICATION À UN SYSTÈME

Trois grandes fonctionnalités :

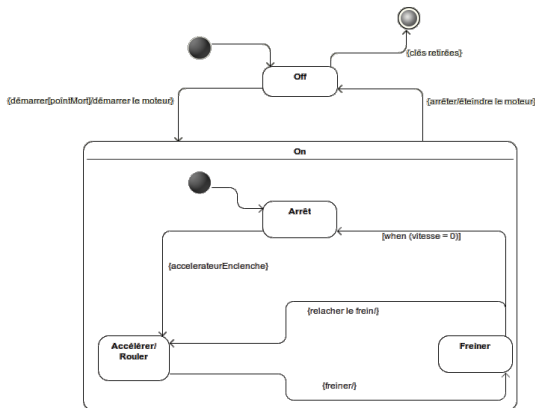
- ▶ Démarrage
- ▶ Accélération
- ▶ Freinage



3.2 APPLICATION À UN SYSTÈME

Trois grandes fonctionnalités :

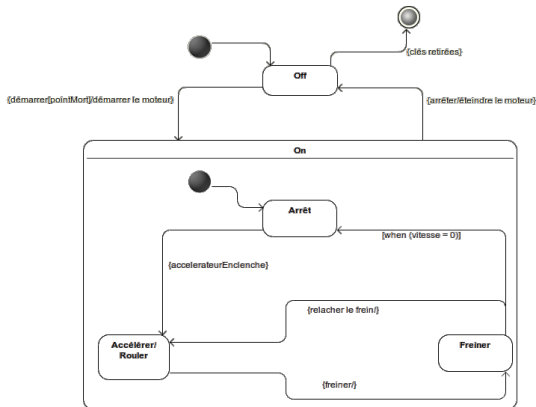
- ▶ Démarrage → STATE_DEMARRAGE
- ▶ Accélération → STATE_ACCELERATION
- ▶ Freinage → STATE_FREINAGE



3.2 APPLICATION À UN SYSTÈME

Démarrage :

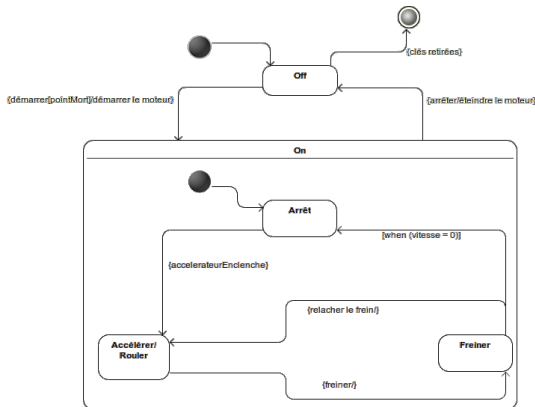
- ▶ Précondition : on est au point mort
- ▶ Postcondition : Le moteur est démarré et on est plus dans l'état démarrage



3.2 APPLICATION À UN SYSTÈME

Démarrage :

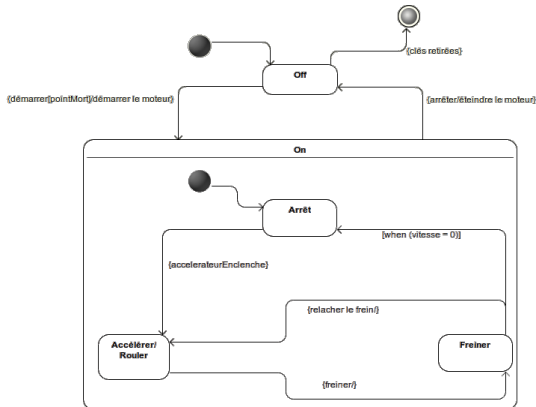
- ▶ **Précondition** : `assert always ((pointMort) before!_ STATE_DEMARRAGE);`
- ▶ **Postcondition** : `assert always (STATE_DEMARRAGE[*] → next (moteurDemarre & !STATE_DEMARRAGE))`



3.2 APPLICATION À UN SYSTÈME

Accélération :

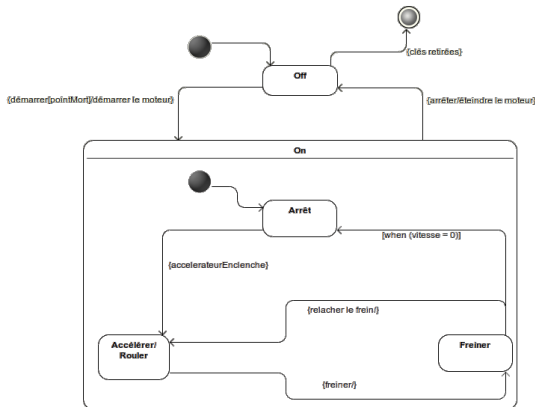
- ▶ Précondition : Le moteur est démarré et la pédale d'accélération est enclenché
- ▶ Postcondition : La pédale d'accélération n'est plus enclenché et on est plus dans l'état d'accélération



3.2 APPLICATION À UN SYSTÈME

Démarrage :

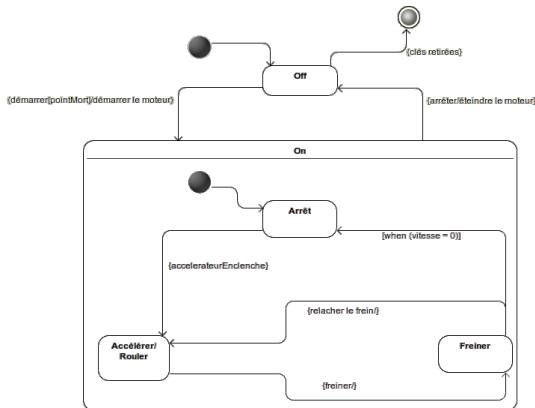
- ▶ **Précondition**: `assert always ((moteurDemarre
accelerateurEnclenche) before!_ STATE_ACCELERATION);`
- ▶ **Postcondition**: `assert always (STATE_ACCELERATION[*] → next
(!accelerateurEnclenche & !STATE_ACCELERATION))`



3.2 APPLICATION À UN SYSTÈME

Freinage :

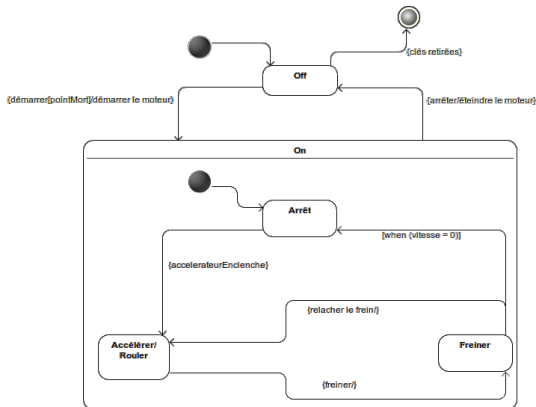
- ▶ Précondition : La pédale de frein est enclenché
- ▶ Postcondition : La pédale de frein n'est plus enclenché et on est plus dans l'état de freinage



3.2 APPLICATION À UN SYSTÈME

Démarrage :

- ▶ **Précondition** : `assert always ((freinEnclenche) before!_STATE_FREINAGE);`
- ▶ **Postcondition** : `assert always (STATE_FREINAGE[*] → next (!freinEnclenche & !STATE_FREINAGE))`



PLAN

I. CONCEPTS, OUTILS, ET MÉTHODES UTILISÉS

- 1.1 PSL
- 1.2 SysML
- 1.3 Modelio
- 1.4 Méthodologie

II. PROPOSITION

- 2.1 Systèmes hétérogènes
- 2.2 Intégration de PSL
- 2.3 Dédutions et ajouts de propriétés PSL

III. EXPÉRIMENTATION

- 3.1 Traductions SysML-PSL
- 3.2 Application à un système

IV. BILAN

4. BILAN DU PROJET

- ▶ Bilan technique

4. BILAN DU PROJET

- ▶ Bilan technique
- ▶ Bilan personnel

4. BILAN DU PROJET

- ▶ Bilan technique
- ▶ Bilan personnel
- ▶ *Avenir du projet*

Des questions ?