

# Ingénierie dirigée par les modèles II

Frédéric Boulanger

17 novembre 2022

# Rappels

**Modèle** représentation simplifiée (abstraction) d'un système pour un objectif donné

- ▶ modèle d'analyse (système pré-existant)
- ▶ modèle de conception (système à concevoir)
- ▶ modèle explicatif
- ▶ modèle prédictif
- ▶ modèle exécutable/constructif

**Système** objet d'étude, caractérisé par une structure et des interactions entre ses composants

**Conception** processus composé d'une suite de raffinements de modèles depuis la spécification jusqu'à l'implémentation

**MDE/IDM** approche de la conception dans laquelle les modèles capturent la majeure partie de l'effort de conception

# Rappels

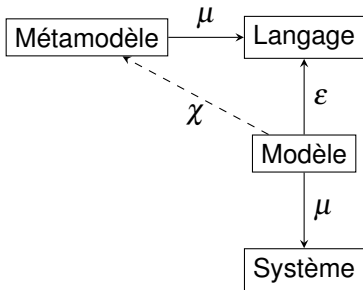
## Relations entre modèles

$M \xrightarrow{\mu} S$  M représente (est un modèle de) S

$M \xrightarrow{\varepsilon} L$  M appartient au (est écrit dans le) langage L

$M \xrightarrow{\delta} P$  M se décompose en (contient) P

$M \xrightarrow{\chi} MM$  M est conforme au métamodèle MM



# Sémantique des modèles

# Sémantique des modèles



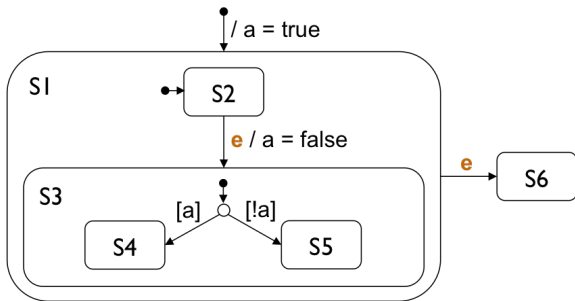
## Sémantique des modèles



Quel est le sens de « jaguar » ?

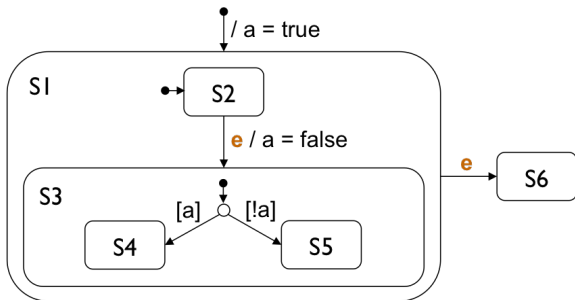
# Nécessité d'une sémantique précise

## Un diagramme Statechart



# Nécessité d'une sémantique précise

## Un diagramme Statechart



L'évènement  $e$  peut mener à :

- ▶ S4 avec UML : la transition vers S1 a priorité et met  $a$  à `true`
- ▶ S5 avec Rhapsody : la transition de S2 vers S3 a priorité
- ▶ S6 avec Stateflow : la transition vers S6 préempte S1

# Nécessité d'une sémantique précise

## Les 3 sémantiques sont *correctes*

Le problème est que la sémantique est définie *implicitement* par l'outil !

Que se passe-t-il si :

- ▶ le *concepteur* pense selon la sémantique UML
- ▶ le générateur de code interprète le modèle comme Rhapsody
- ▶ la vérification est faite selon la sémantique Stateflow ?

# Nécessité d'une sémantique précise

## Les 3 sémantiques sont *correctes*

Le problème est que la sémantique est définie *implicitement* par l'outil !

Que se passe-t-il si :

- ▶ le *concepteur* pense selon la sémantique UML
- ▶ le générateur de code interprète le modèle comme Rhapsody
- ▶ la vérification est faite selon la sémantique Stateflow ?

## La sémantique d'un modèle devrait être :

- ▶ *explicite*, pour éliminer les doutes sur l'interprétation
- ▶ *bien définie*, de façon à pouvoir vérifier des propriétés du modèle

# Nécessité d'une sémantique précise

## Les 3 sémantiques sont *correctes*

Le problème est que la sémantique est définie *implicitement* par l'outil !

Que se passe-t-il si :

- ▶ le *concepteur* pense selon la sémantique UML
- ▶ le générateur de code interprète le modèle comme Rhapsody
- ▶ la vérification est faite selon la sémantique Stateflow ?

## La sémantique d'un modèle devrait être :

- ▶ *explicite*, pour éliminer les doutes sur l'interprétation
- ▶ *bien définie*, de façon à pouvoir vérifier des propriétés du modèle

## Sémantique formelle

Sémantique définie de telle manière qu'un modèle puisse être traité de façon automatique et consistante par des programmes.

# Syntaxe et sémantique

## Modèle informatique

- ▶ un modèle est considéré en tant que système (transformation, validation)
- ▶ un modèle est écrit dans un langage de modélisation (UML, Simulink, ...)
- ▶ un langage de modélisation est considéré en tant que système (éditeurs, grammarware)
- ▶ un langage de modélisation est décrit par un métamodèle (modèle du langage)

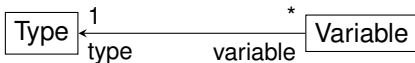
## Métamodèle = syntaxe seule

- ▶ syntaxe abstraite (concepts et relations)
- ▶ syntaxe concrète (mots-clefs, éléments graphiques)

# Syntaxes abstraite et concrète

## Exemple

- ▶ syntaxe abstraite (concepts et relations)



- ▶ syntaxe concrète (notation à l'aide de symboles)

```
int i;
```

- ▶ autre syntaxe concrète textuelle

```
i: int;
```

- ▶ syntaxe concrète graphique

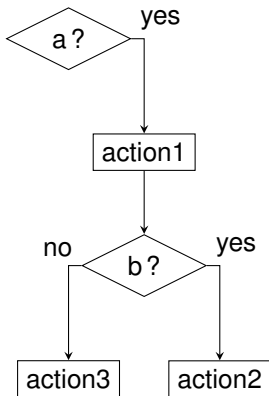


# Nécessité d'une syntaxe non ambiguë

```
if (a) then  
do action1  
if (b) then  
do action2  
else  
do action3
```

# Nécessité d'une syntaxe non ambiguë

```
if (a) then  
do action1  
if (b) then  
do action2  
else  
do action3
```

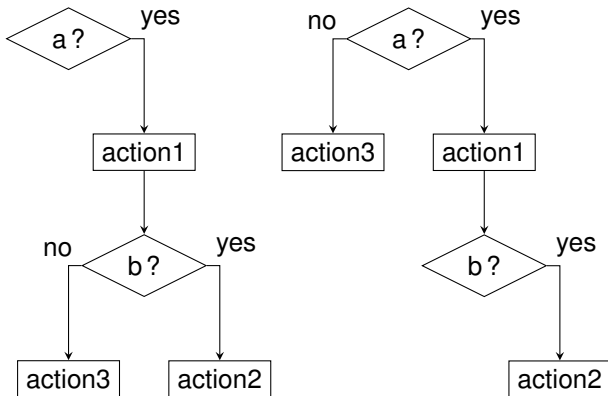


## Nécessité d'une syntaxe non ambiguë

```

if (a) then
do action1
if (b) then
do action2
else
do action3

```



# Sémantique

**Sémantique** : sens donné aux éléments syntaxiques

Se définit par une association entre les éléments syntaxiques (signifiants) et les éléments du domaine sémantique (signifiés)

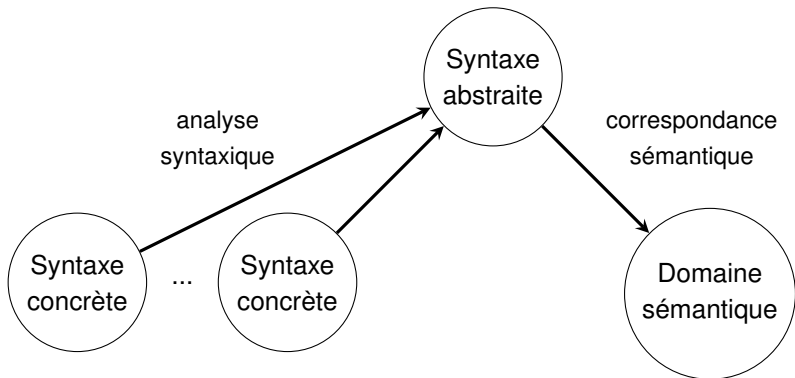
## Domaine sémantique

- ▶ autre langage (traduction)
- ▶ théorie mathématique

## Définition de la sémantique

1. définition formelle de la syntaxe abstraite
2. choix d'un domaine sémantique
3. association entre syntaxe abstraite et domaine sémantique

## Définition de la sémantique d'un langage



# Point clefs pour la sémantique des modèles

## Sémantique

- ▶ définition non ambiguë de la syntaxe
- ▶ choix d'un domaine sémantique
- ▶ définition d'une correspondance entre domaines syntaxique et sémantique

Souvent donnée par une transformation de modèle

## Sémantique statique

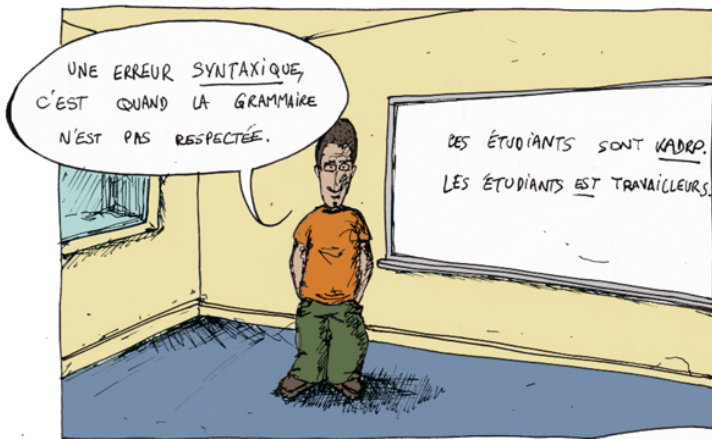
- ▶ exprime des contraintes sur les modèles
- ▶ n'est pas capturée par la syntaxe
- ▶ exemple : une variable est déclarée avant d'être utilisée

Peut être exprimée sous forme d'invariants OCL par exemple.

# Lexèmes, syntaxe et sémantique



# Lexèmes, syntaxe et sémantique



## Lexèmes, syntaxe et sémantique

