



Génération de tests à partir de modèles SysML

Fabrice BOUQUET

13 Novembre 2012

4^{ème} journée SysML France - Mulhouse



Contexte VETESS



- VETESS : validation de systèmes embarqués **VE**hicule par génération automatique de **TE**sts à partir des Spécifications SysML (<http://lifc.univ-fcomte.fr/vetess/>)
- De juin 2008 à août 2010
- Financé par le FUI (Fond Unique Interministériel)
- Labélisé pôle de compétitivité « Véhicule du futur »





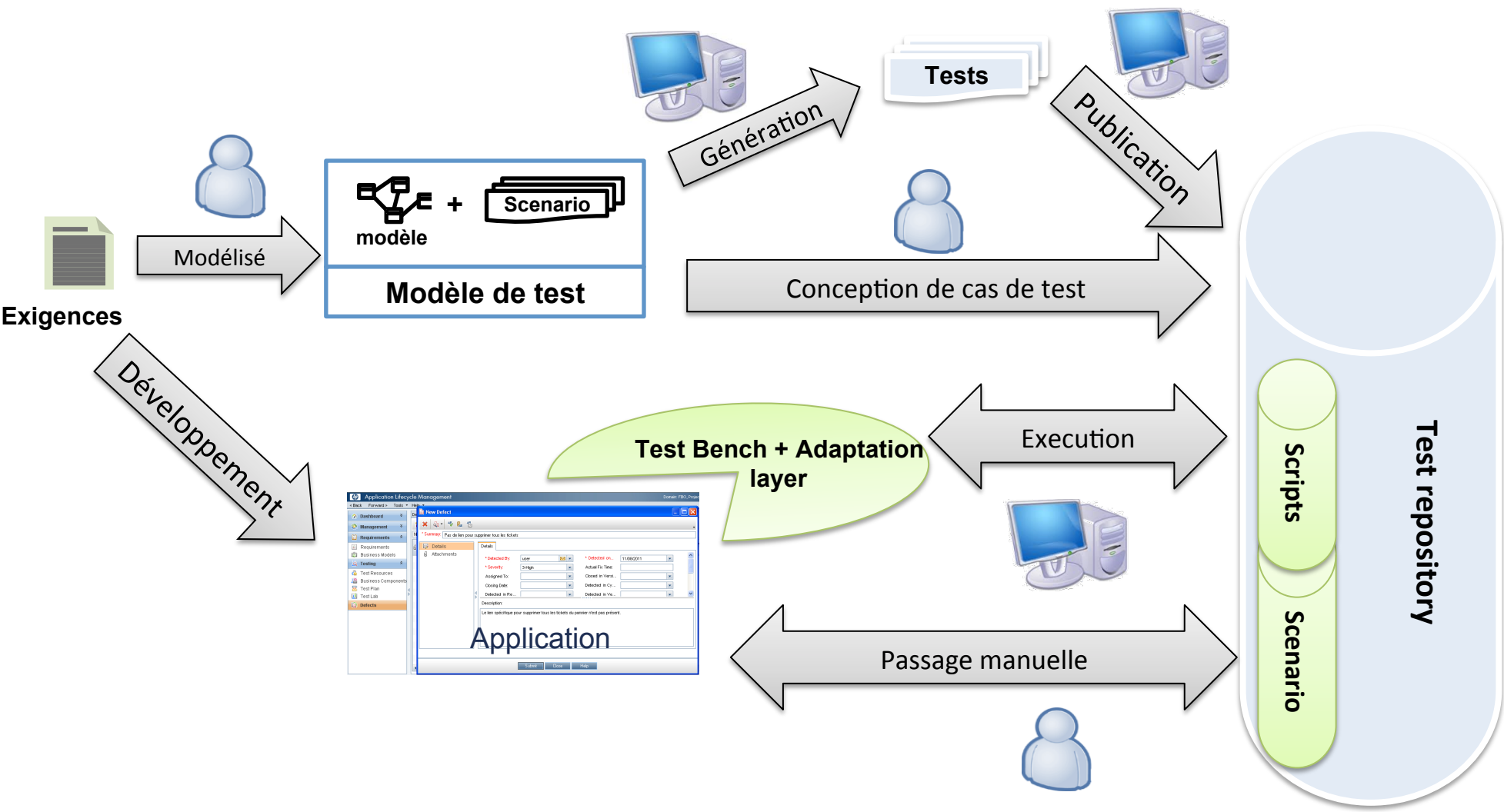
- Partenaires :
 - PSA Peugeot-Citroën : constructeur automobile
 - Smartesting : éditeur logiciel (Test Designer)
 - Clemessy : éditeur de bancs de tests (TestInView)
 - MIPS : Model-Driven Engineering (MDE)
 - FEMTO-ST / DISC : Model-Based Testing (MBT)
- Objectifs :
 - Validation de systèmes embarqués véhicule
 - Génération de tests à partir de modèles (MBT)
 - Chaîne automatisée et adaptable : modèle -> banc

Sommaire



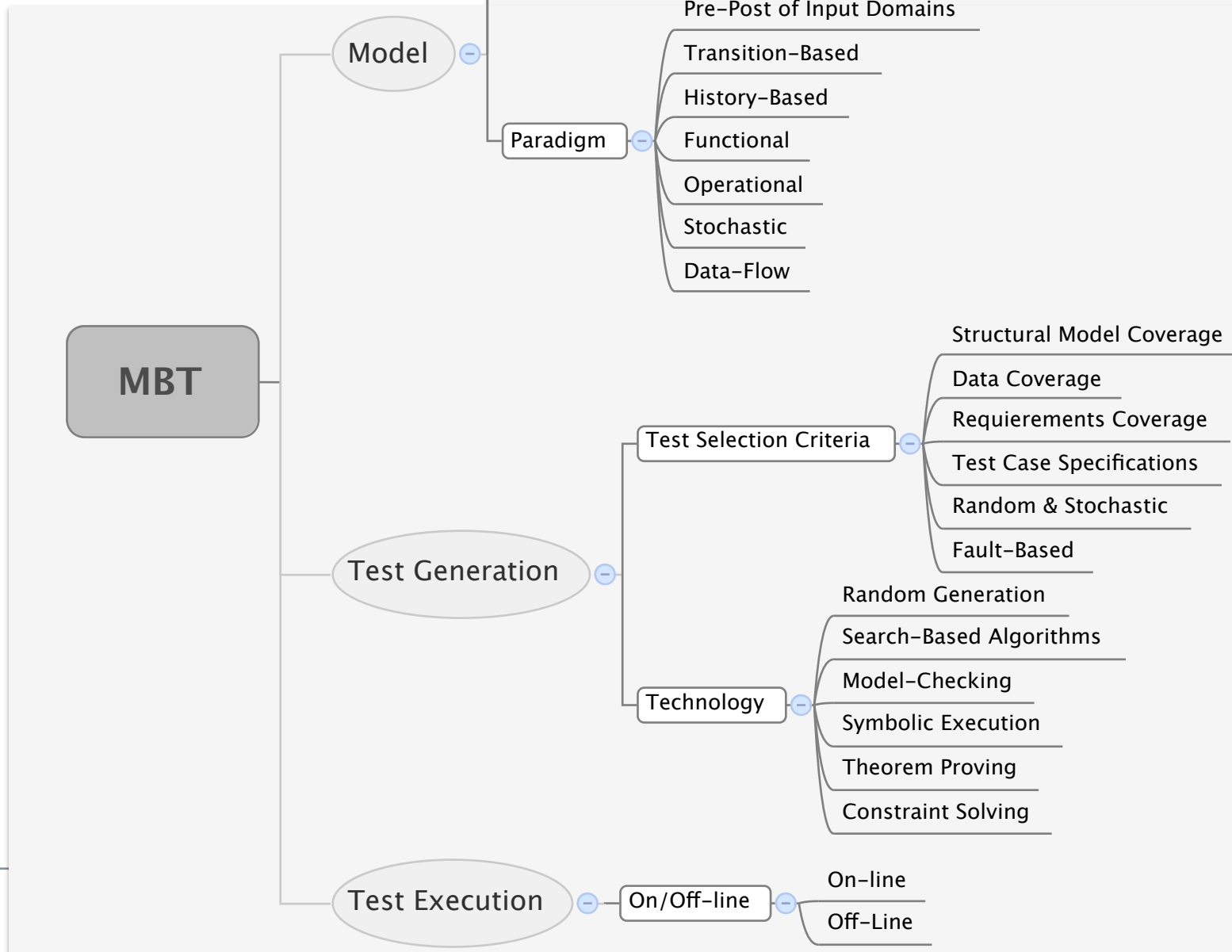
- ① Model-Based Testing
- ② Langages
- ③ Exemple fil rouge
- ④ Critère de couverture
- ⑤ Exécution
- ⑥ Bilan

Model-Based Testing

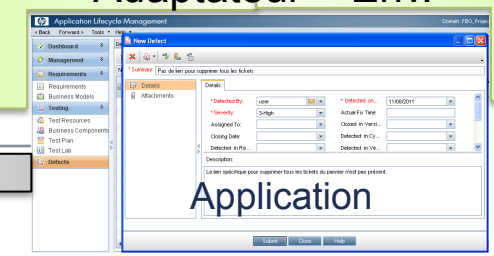
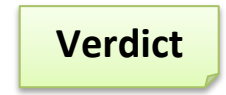
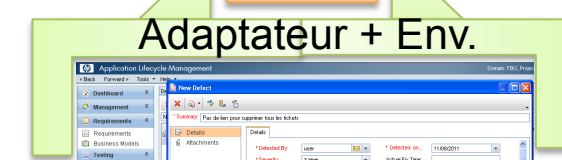
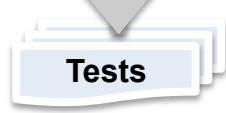
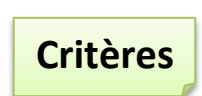
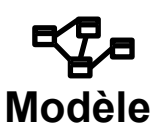
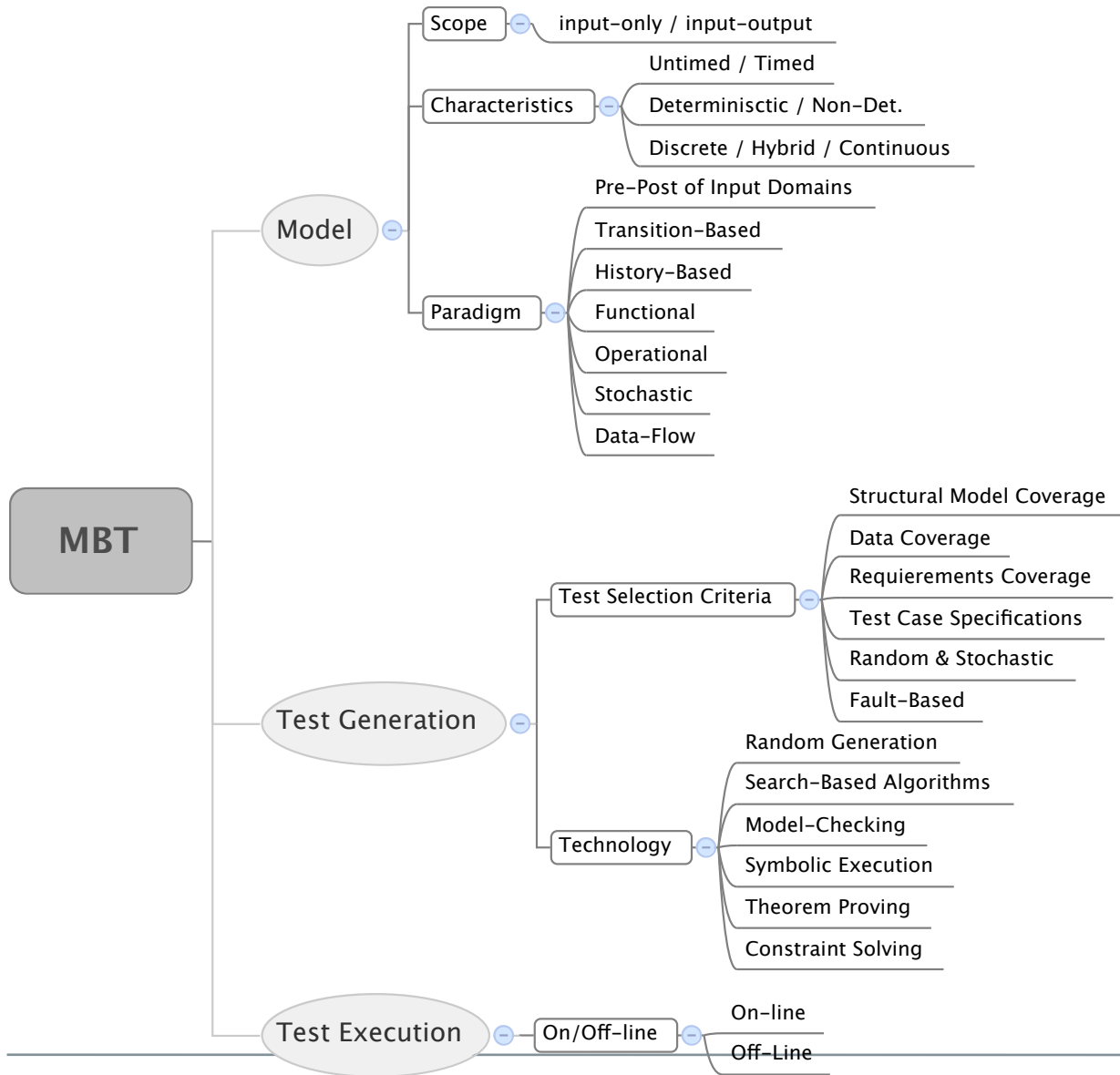


Taxonomie MBT

[Utting & al 11]

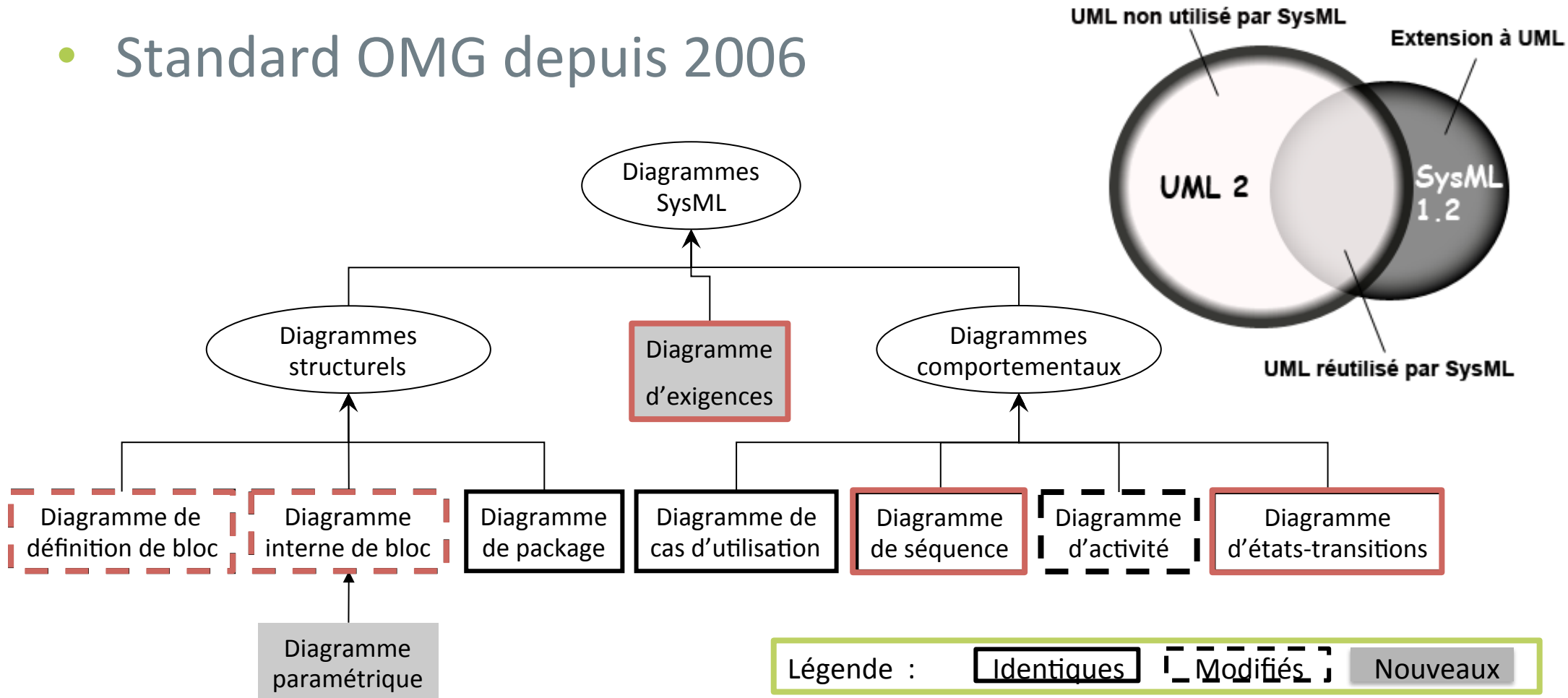


MBT - Outillé



Modélisation SE SysML (1.2)

- Profil UML pour la modélisation des systèmes
- Standard OMG depuis 2006



Exemple fil rouge

- Arrêt d'urgence d'un train
 - Deux actionneurs :
 - Un pour le conducteur
 - Un pour les passagers
 - Effets attendus :
 - Si train à l'arrêt : prévenir uniquement la régulation
 - Si train en marche : arrêter le train et déclencher l'alarme

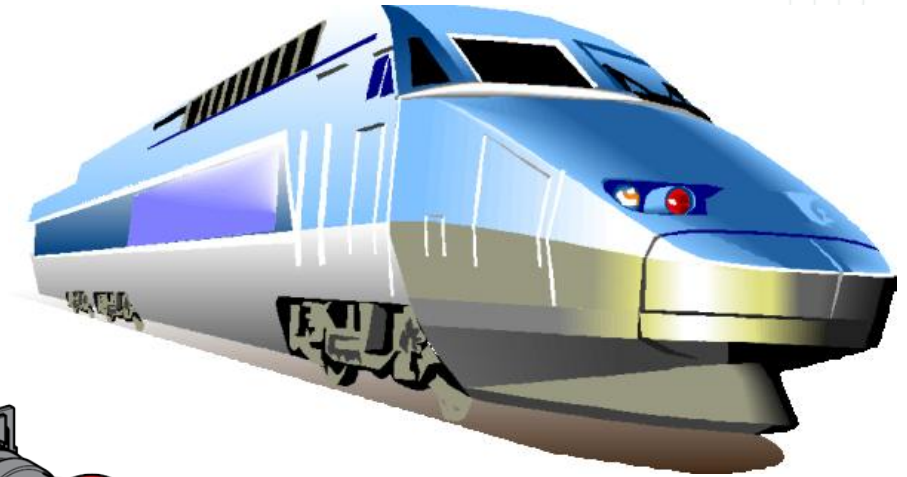
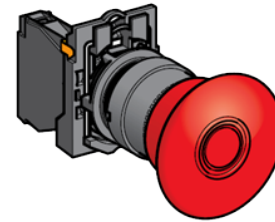


Diagramme de bloc

- Représentation statique du système/environnement :

- blocs
- propriétés
- opérations
- associations
- signaux
- ports de flux
- spécifications de flux

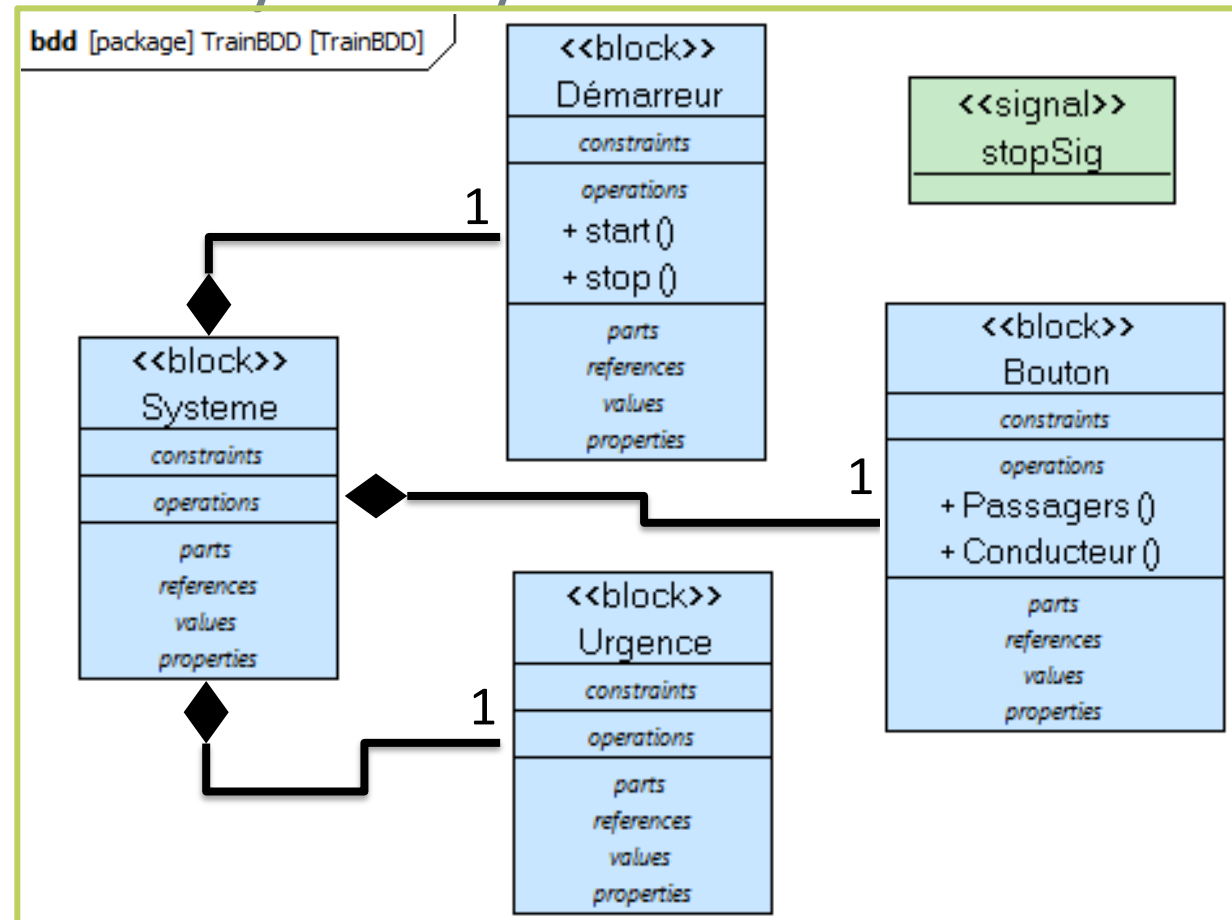


Diagramme interne de bloc

- Représentation des interconnexions entre les blocs
- Liens entre les ports des blocs

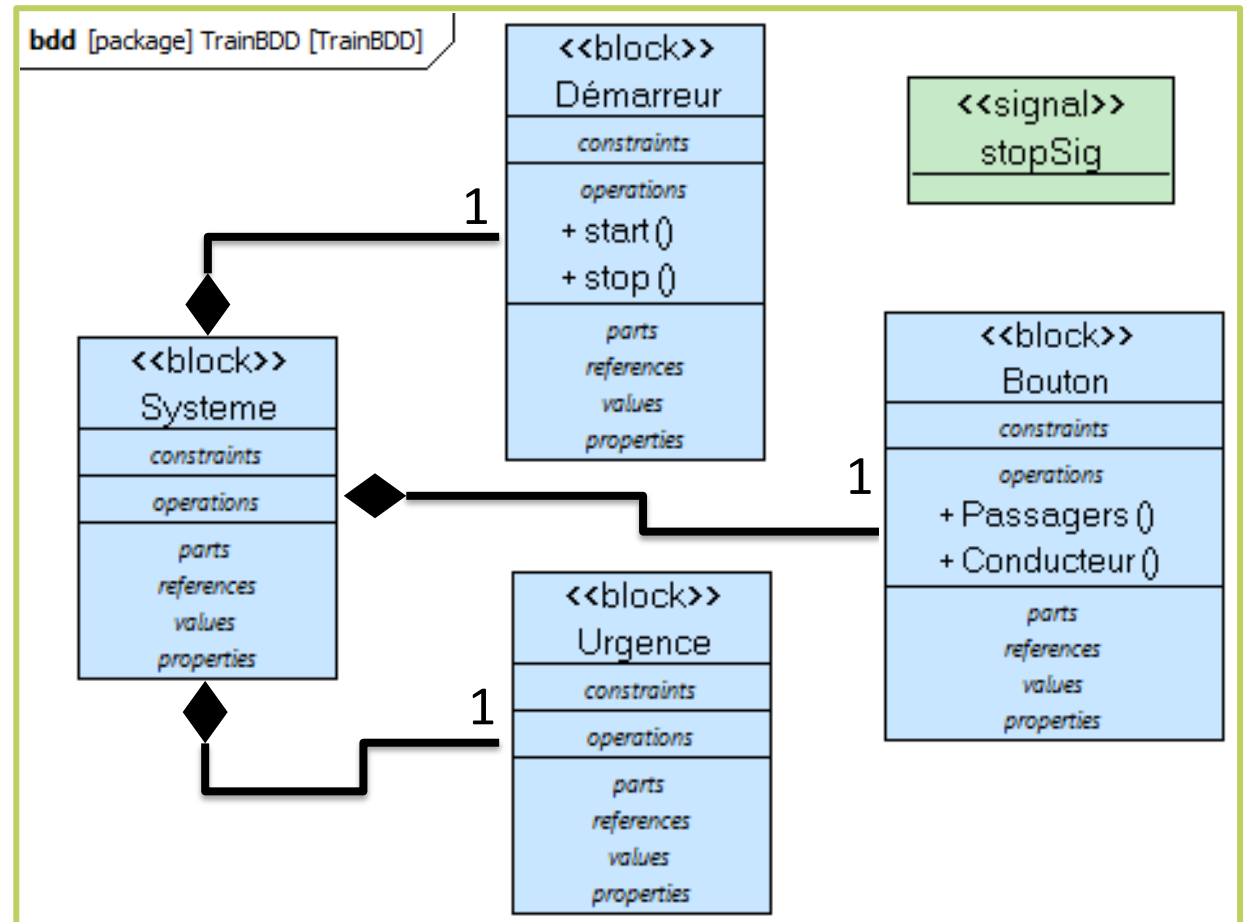
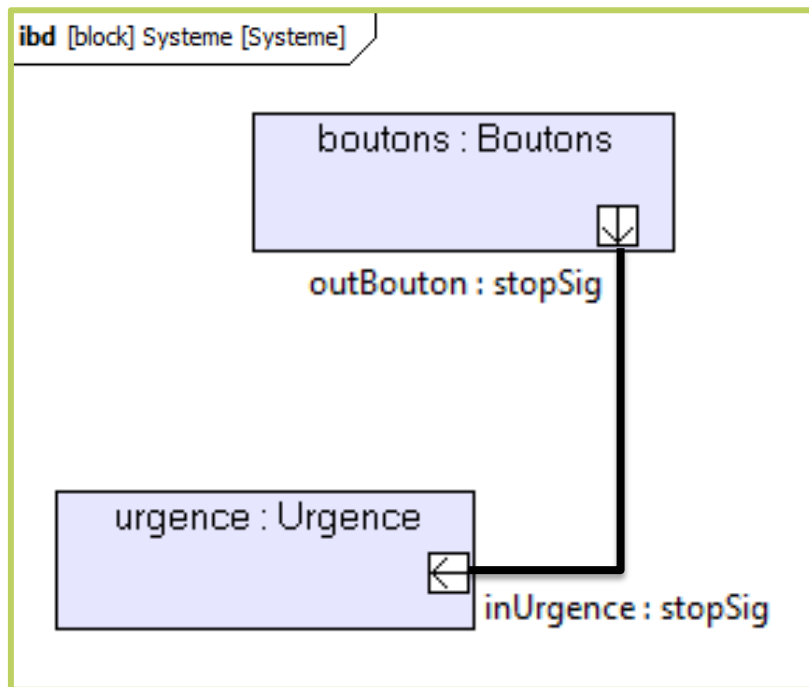
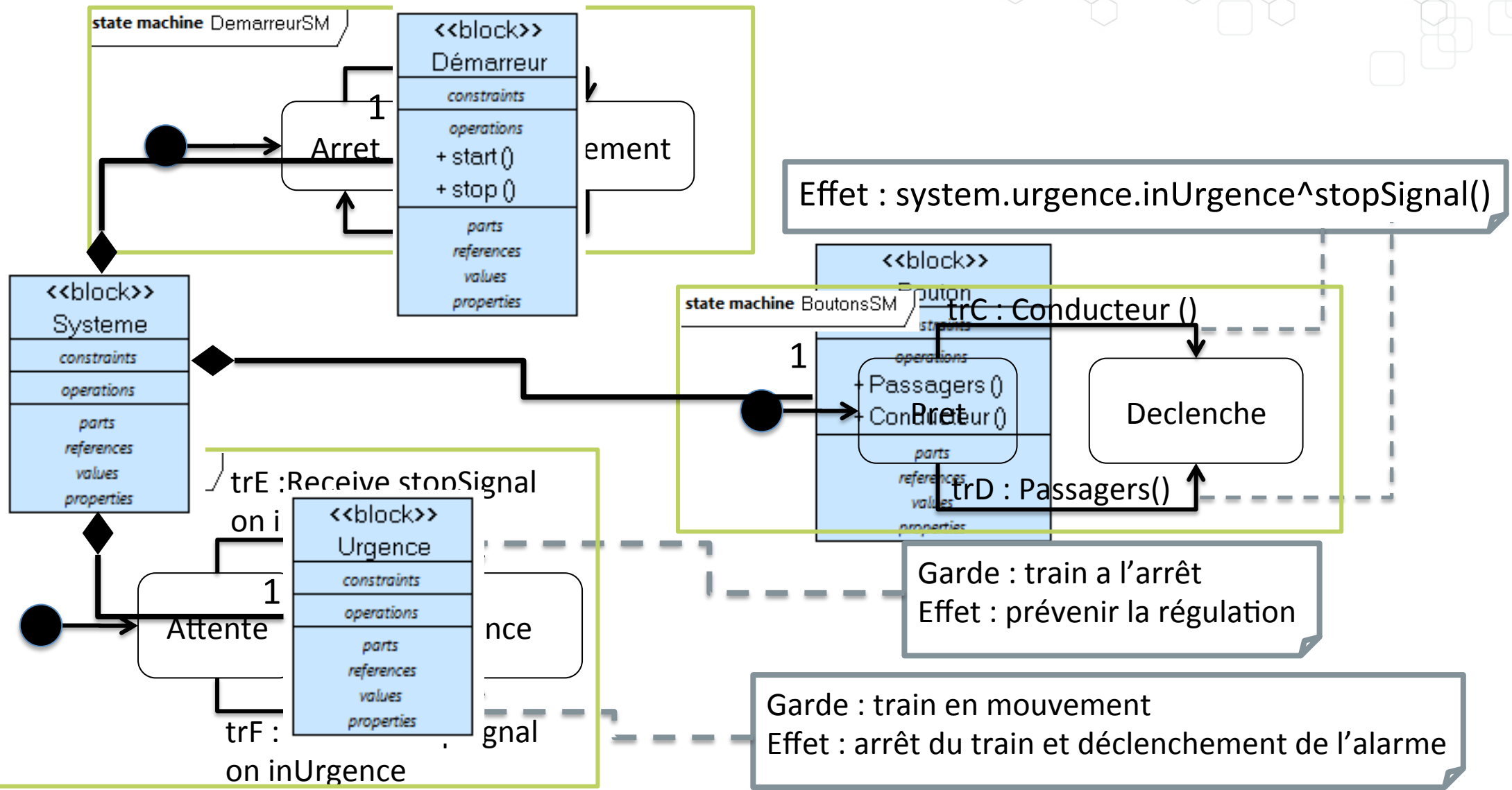


Diagramme d'états-transitions

- Représentation des comportements du système
- Annotations OCL :
 - Formalisation des comportements
 - Pre/post d'opérations, gardes et effets

Diagramme d'états-transitions



Critères de couverture classiques

- Basés sur les éléments graphiques (exemple : *tous les états*)
- Orientés flot de données : utilisation des données (exemple : *toutes les DU*)
- Orientés flot de contrôle : décisions / conditions (exemple : D/CC)

Critère *tous les états*

- Couverture de chaque état modélisé dans le diagramme d'états / transitions

Tous les états

Critère *toutes les transitions*

- Couverture de chaque transition modélisée dans le diagramme d'états / transitions
- Couverture des transitions dans le diagramme de séquences

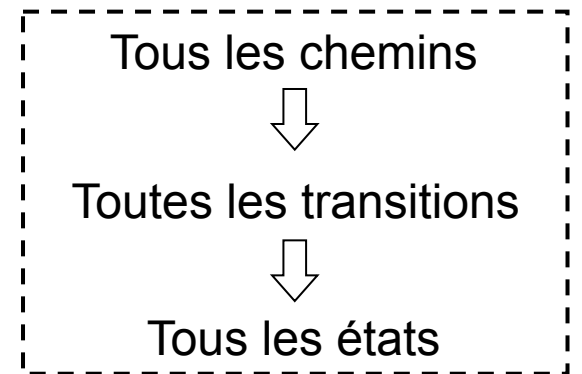
Toutes les transitions



Tous les états

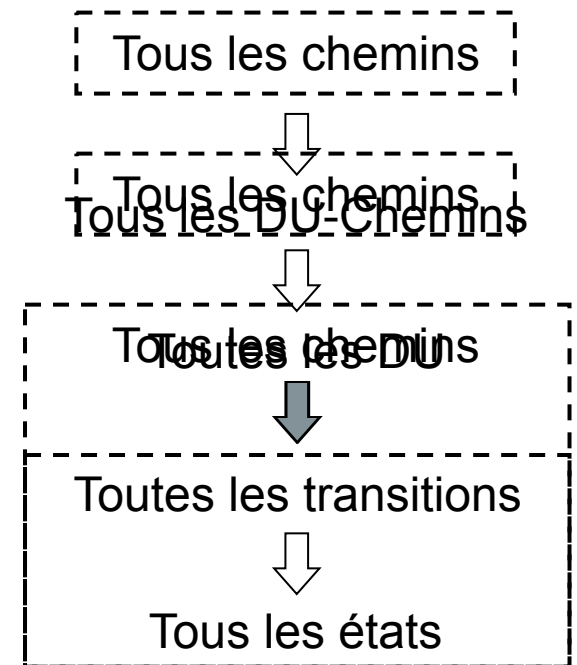
Critère *tous les chemins*

- Couverture de chaque chemin possible dans le diagramme d'états / transitions

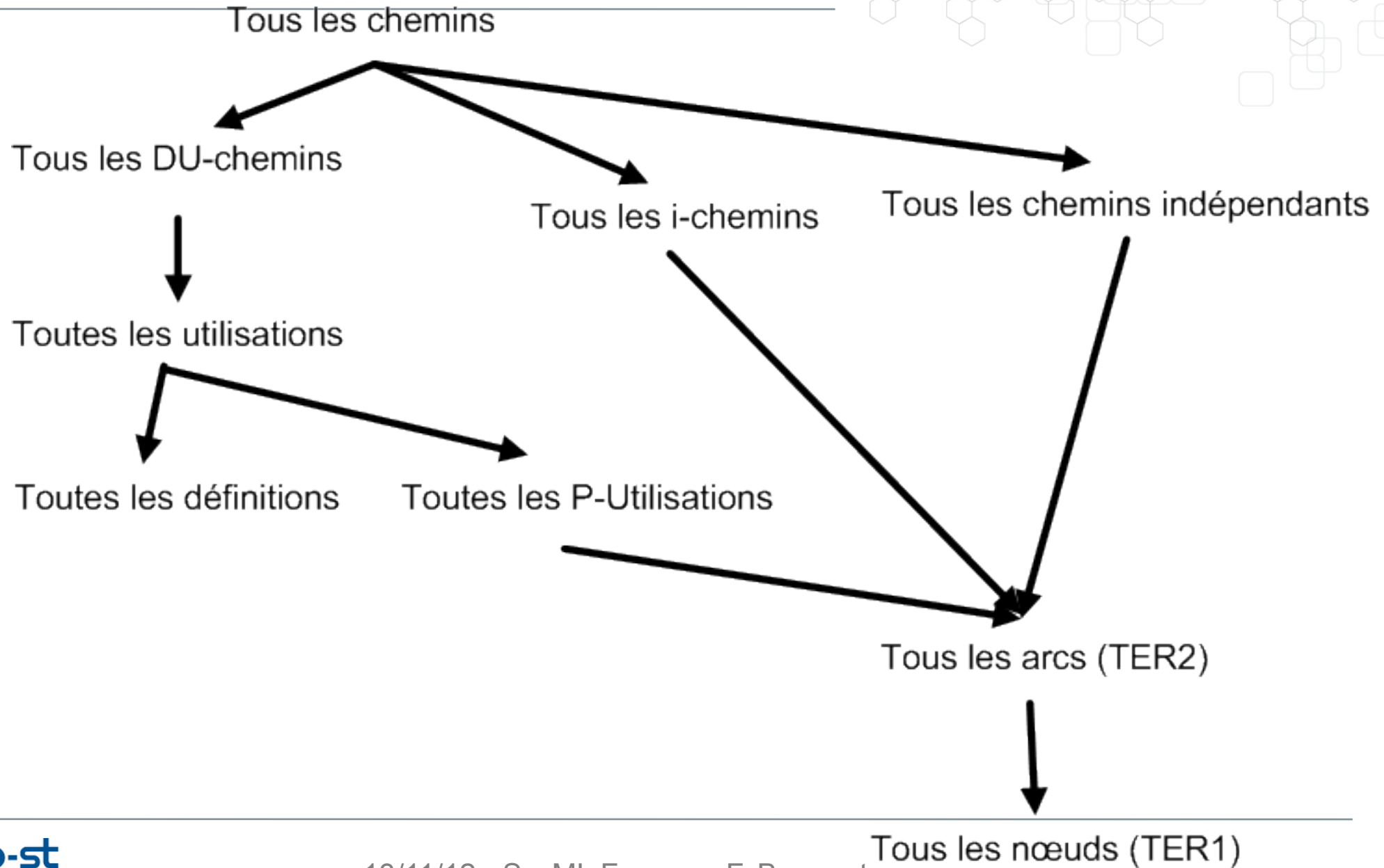


Critères *toutes les DU*

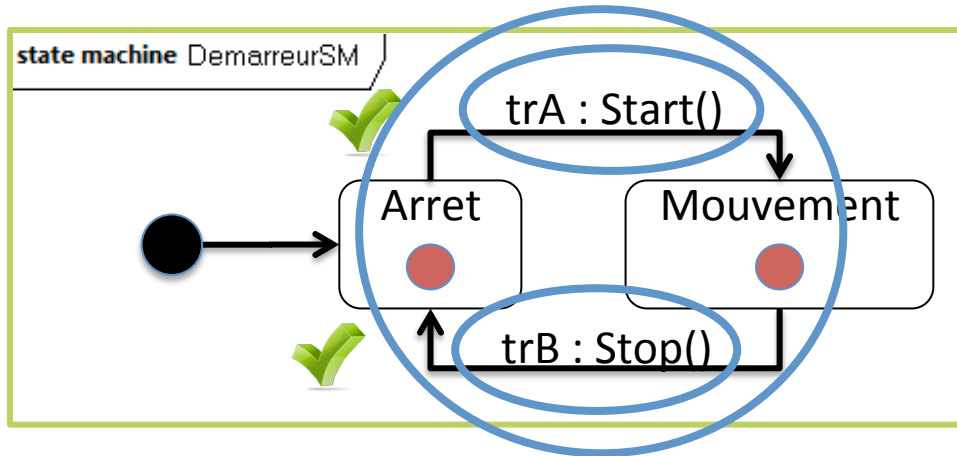
- Définition de variable : affectation d'une variable
- Utilisation de variable : lecture d'une variable
- *Toutes les DU* (flot de données) :
 - Pour chaque définition *def* de *var*
 - Pour chaque utilisation *use* de *var*
 - Couverture de *def* puis *use*
 - sans modification de *var* intermédiaire
- *Tous les DU-chemins* :
 - Couverture de chaque chemin possible
 - de *def* à *use* (sans modification de *var* intermédiaire)



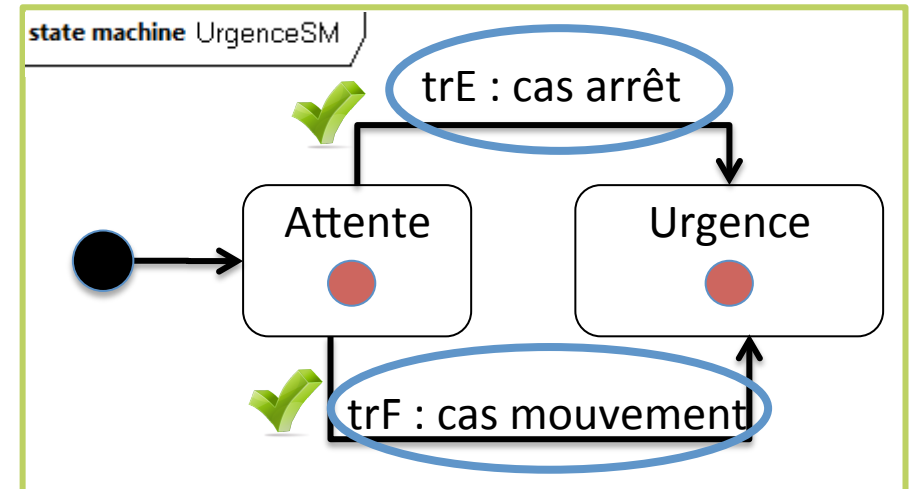
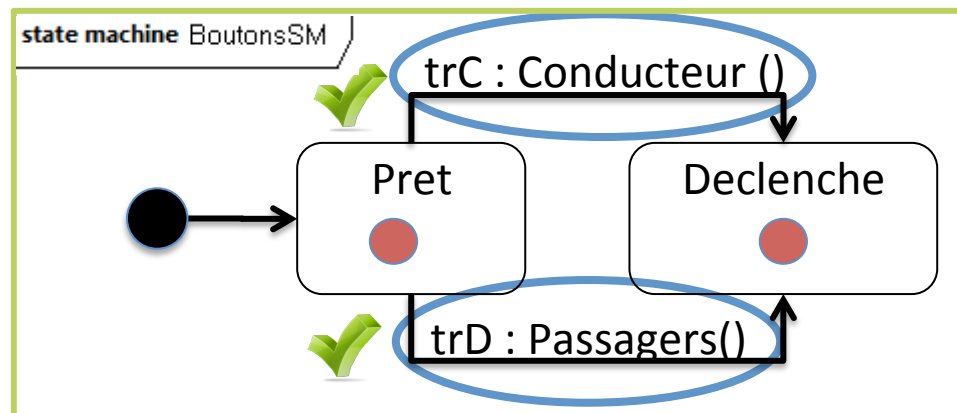
Critères



Analyse – exemple fil rouge



- Trace :
 - Start()getA()trDrC
 - Stop()getB()trDrC
 - [Cas mouvement: trF]



- Critère *toutes les Diversités* **INADAPTE À SOUS-FOURANT**

Critères proposés

- Basés sur la couverture des communications
- Inspirés des critères *Toutes les DU*
- *Toutes les DU_{sig}* :

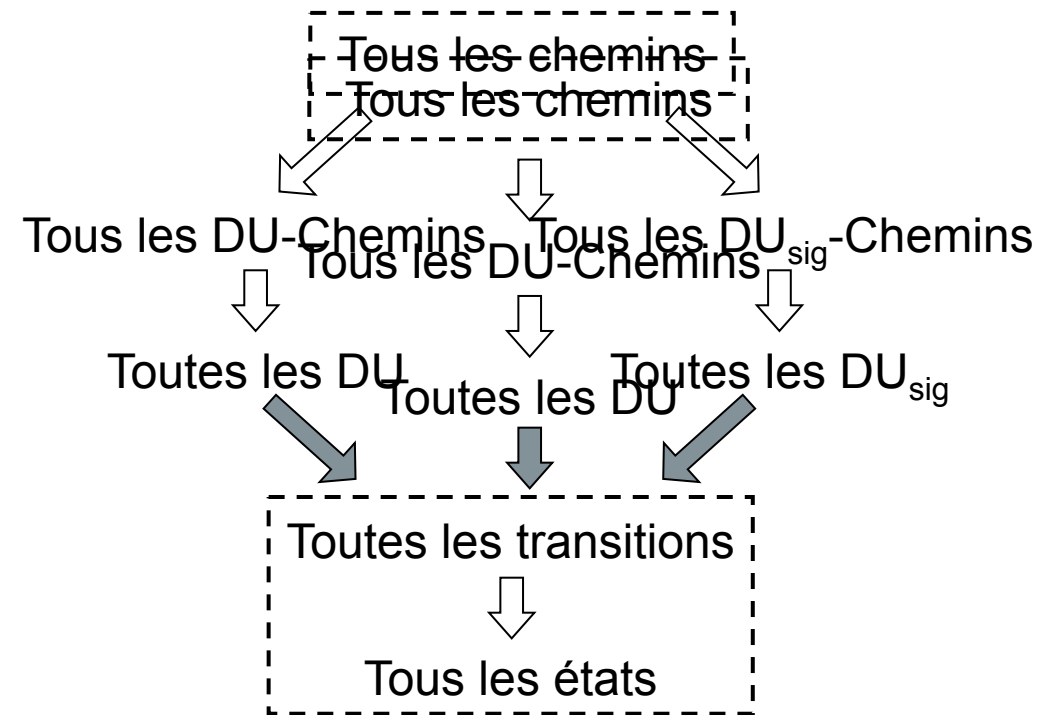
Pour chaque envoi *send*
du signal *S* vers le port *P*

Pour chaque réception *rec*
du signal *S* sur le port *P*

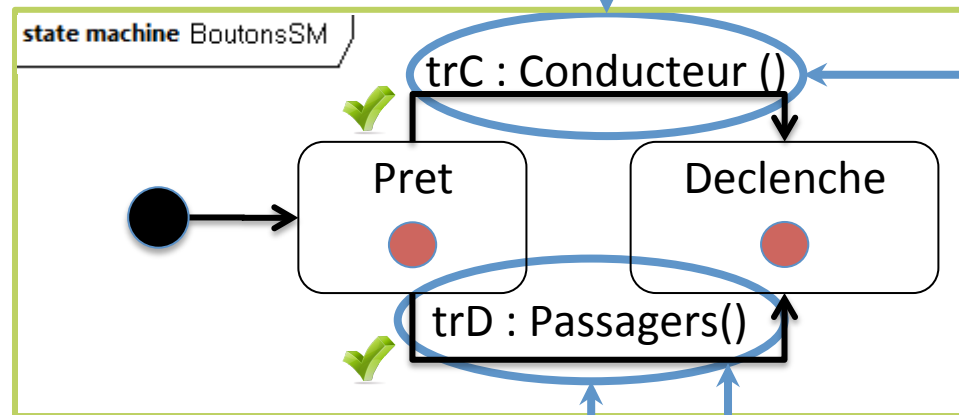
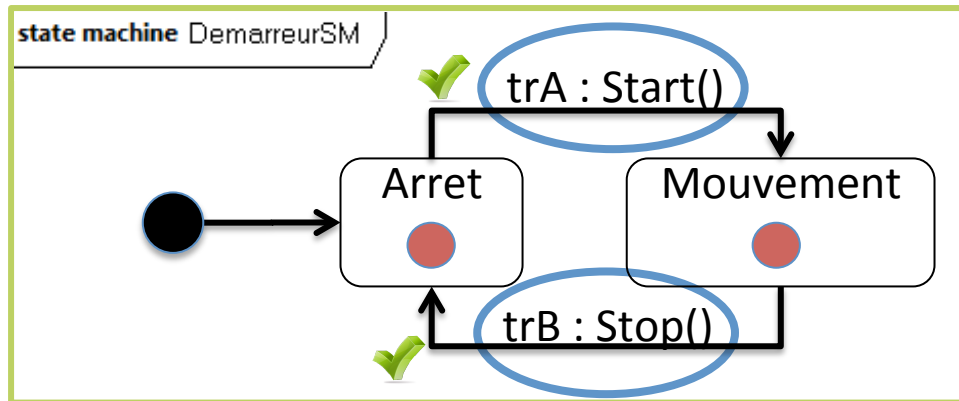
Couverture de *send* puis *rec*
sans passer par un état stable

- *Tous les DU_{sig}-chemins* :

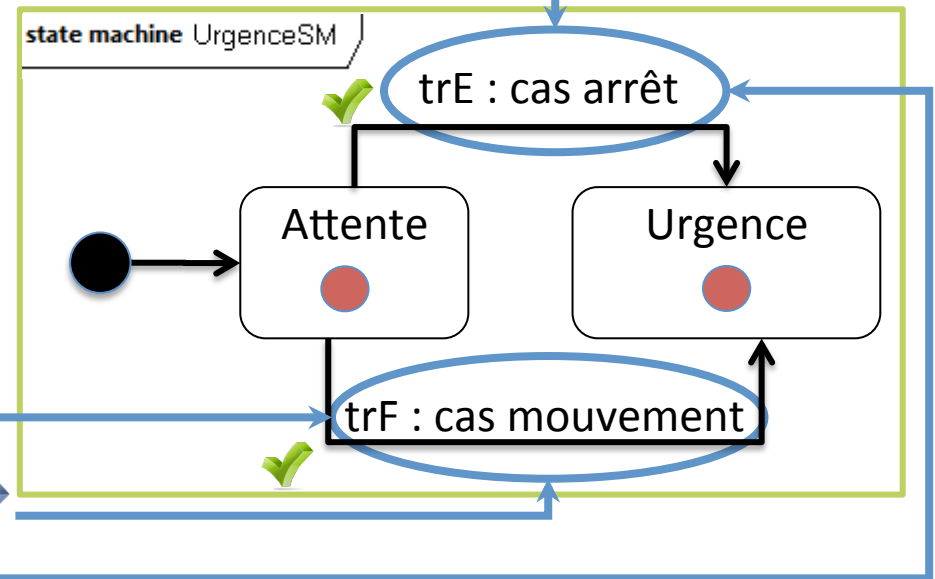
Couverture de chaque chemin possible
de *send* à *rec*



Analyse – exemple fil rouge



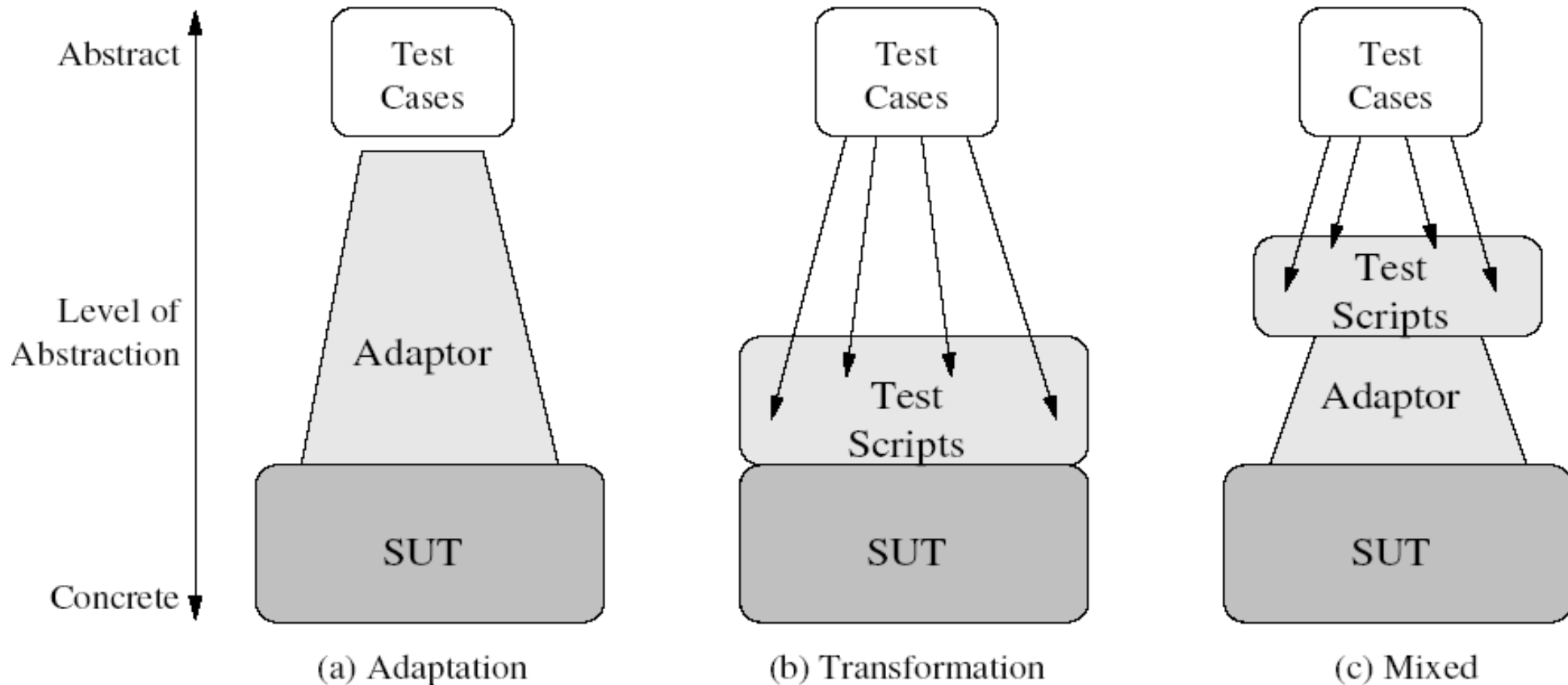
- Trace :
 - Start(trA())trDrC
 - Stop(trB())trDrC
 - [Cas mouvement: trF]



- Critère *toutes les DU_{sig}*

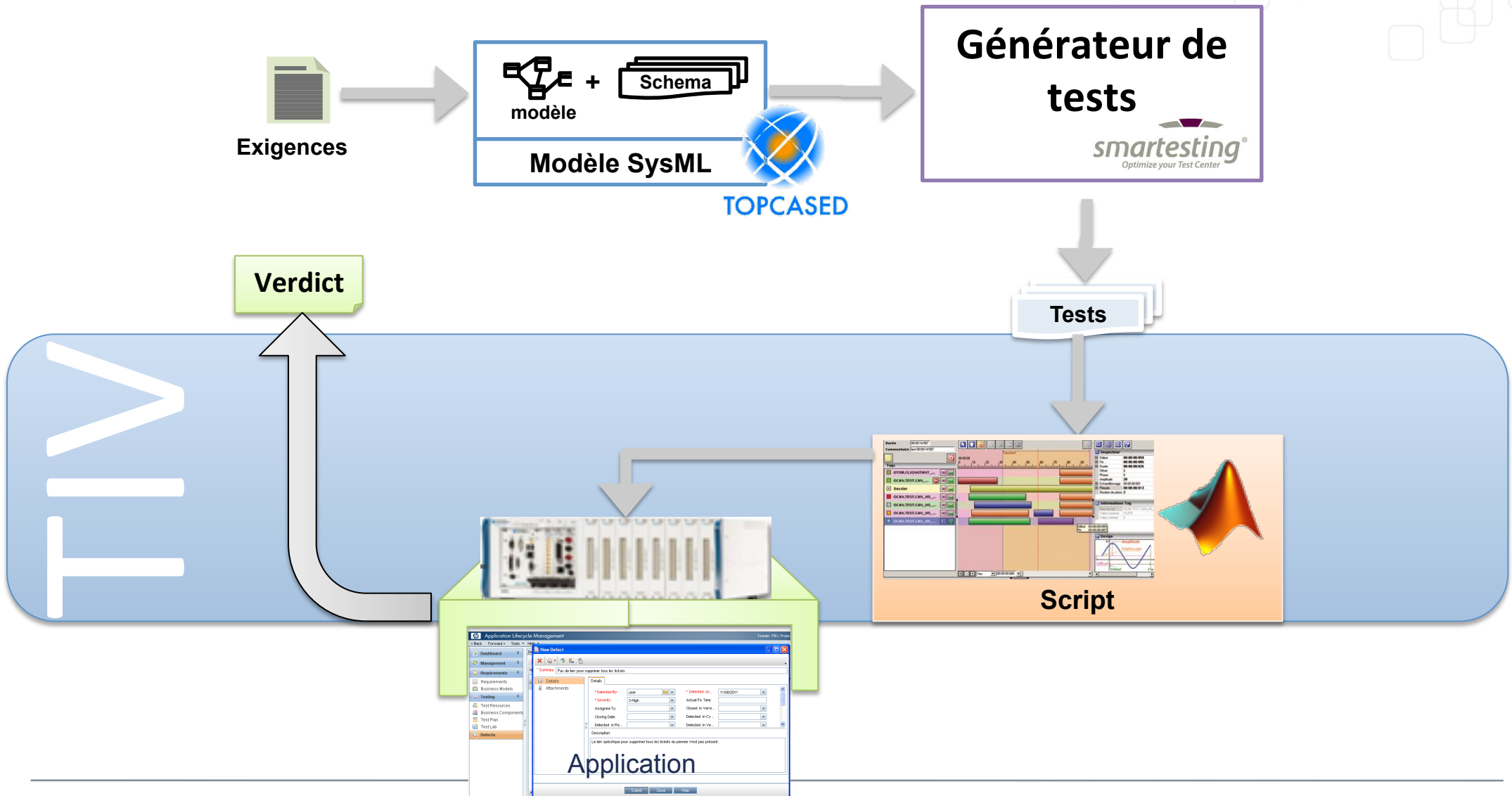
PERTINENT

Model-Based Testing – Exécution

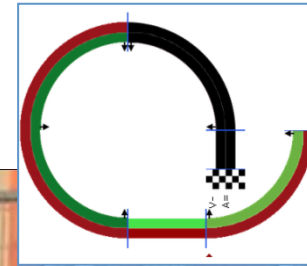
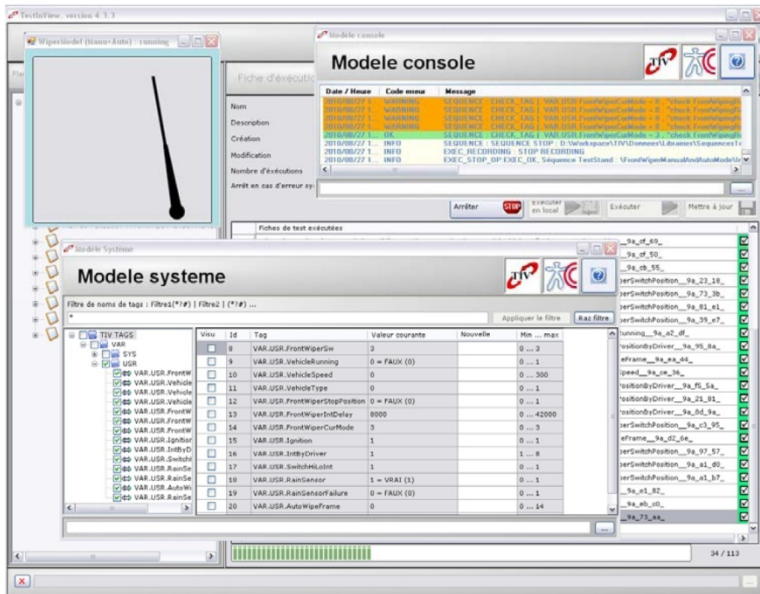


Trois approches pour l'adaptation de test en script de test pour leur exécution

Exécution des tests



Expérimentations



Etude de cas
Front Wiper
(sur simulateur)

Etude de cas
Colonne de direction
(sur banc physique)

Bilan / Perspective



Bilan :

- Méthode de génération de tests outillés
 - Aspect discret
 - Aspect continu (utilisation de MatLab/Simulink)
- Traçabilité exigences / tests

Perspective :

- Evolution SysML 1.3, ALF
- Intégration de Séquences dans l'outils
- Intégration Profil Marte → critères spécifiques
- Interprétation des aspects non discret

Merci pour votre attention...



"Testing is always model-based!"
Robert Binder



Source - <http://model-based-testing.info>