



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Qu'est-ce que la Méthode B ?

Métamodélisation du langage B



Une méthode de preuve formelle

- ▶ La méthode B est une méthode formelle, c'est-à-dire une méthodologie reposant sur un **formalisme mathématique**. Son but est de permettre le raisonnement sur des **systèmes complexes et critiques**, et de **garantir le bon fonctionnement** de ceux-ci à travers des démonstrations logiques.
- ▶ La méthode B repose sur un principe de **raffinements successifs** de modèles. La conception commence par une modélisation abstraite du système et, raffinement après raffinement, un modèle concret, traductible en langage de programmation, se dessine.



Un langage de conception

- ▶ La méthode B est accompagnée par un langage dédié à la modélisation de systèmes, le **langage B**, supporté par le logiciel propriétaire **Atelier B**. Il existe une méthode B étendue par la notion d'événements, appelée **Event-B**. Le langage qui l'accompagne se nomme **B#**, et l'outil dédié est l'environnement libre **Rodin**, fondé sur l'IDE Eclipse.
- ▶ Le langage B événementiel, plus répandu, se concentre sur la description de propriétés d'un système en termes de **théorie des ensembles**. Ainsi, le nombre de passagers d'une rame de métro sera un élément de N , et la température en °C d'un microprocesseur sera un élément de $[-50, 100]$.
- ▶ Au cœur des concepts de la méthode B, le langage B permet de **définir des contraintes**, sous la forme d'invariants logiques et de gardes. Les invariants sont des **prédicats logiques** portant sur les propriétés du système, qui doivent être vérifiés quel que soit l'état de celui-ci. Quant aux gardes, il s'agit de **formules logiques**, qui doivent être vérifiées pour qu'une action puisse avoir lieu.

```
1 MACHINE
2   mac >
3   VARIABLES
4     cars_go >
5     peds_go >
6   INVARIANTS
7     inv1: cars_go ∈ BOOL not theorem >
8     inv2: peds_go ∈ BOOL theorem >
9     inv3: ¬ (cars_go = TRUE ∧ peds_go = TRUE) not theorem >
10  EVENTS
11    INITIALISATION: not extended ordinary >
12    THEN
13      act1: cars_go = FALSE >
14      act2: peds_go = FALSE >
15    END
16
17    set_cars: not extended ordinary >
18    ANY
19      new_value >
20    WHERE
21      grd1: new_value ∈ BOOL not theorem >
22      grd2: new_value = TRUE ⇒ peds_go = FALSE not theorem >
23    THEN
24      act1: cars_go = new_value >
25    END
26
27 END
```

Contact : armand.foucault@telecom-bretagne.eu

- PARTIES PRENANTES -



Openflexo

Équipe PASS

- AUTEUR -

Armand Foucault
Élève Ingénieur
3A Filière 2B

- ENCADRANT -

Jean-Christophe Bach
Maître de conférences
Enseignant-chercheur
Équipe PASS

- ENVIRONNEMENT -

