

# ATELIER/TUTO EVENT-B/RODIN

## INTRODUCTION À LA MÉTHODE EVENT-B ET SES DIFFÉRENTS OUTILS

🎓 TAPAS-ANR meeting

🏛️ Laboratoire Méthodes Formelles - LMF, Paris-Saclay, 19 November 2025



**Idir AIT SADOUNE**  
[idir.aitsadoune@centralesupelec.fr](mailto:idir.aitsadoune@centralesupelec.fr)

# OUTLINE

- The Event-B method
- The Pro-B animator/model-checker
- The Theory plugin

[Back to the outline](#) - [Back to the begin](#)

# OUTLINE

- The Event-B method
- The Pro-B animator/model-checker
- The Theory plugin

[Back to the outline](#) - [Back to the begin](#)

# USING EVENT-B METHOD

- The use of the **Event-B method** has continued to increase.
  - applied to various applications and domains.
  - railway, automotive, aeronautics, cybersecurity, nuclear-energy, ...
- The **Rodin** platform (an **Eclipse-based IDE**) is intended to support the construction and verification of **Event-B models**.
  - **plugins** for editing, generating proof obligations, proving, animating, model-checking, code generating ...



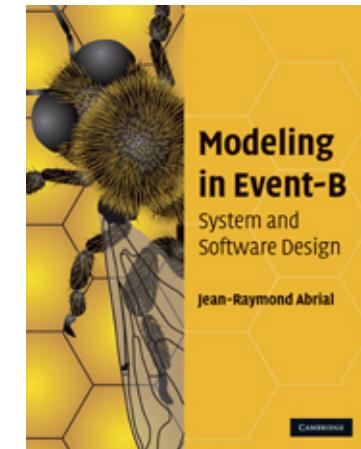
# THE RODIN PLATFORM

- The **Rodin Platform** is an **Eclipse-based IDE** for **Event-B** that provides effective support for refinement and mathematical proof.
- The platform is **open source**, contributes to the **Eclipse framework** and is further extendable with **plugins**.
- **Rodin Platform and Plug-in Installation:**
  - Requires **Java 17**
  - Download the Core: [Rodin Platform file](#) for your platform.
  - Install the [Atelier B Provers plugin](#) from the Atelier B Provers Update site.



# THE EVENT-B METHOD

- The **Event-B method** is an evolution of the **classical B method**.
  - modeling a system by a **set of events** instead of **operations**.
- The **Event-B method** is a **formal method** based on **first-order logic** and **set theory**.
- The **Event-B method** is based on :
  - the notions of pre-conditions and post-conditions (**Hoare**),
  - the **weakest pre-condition** (**Dijkstra**),
  - and the **calculus of substitution** (**Abrial**).
- The **Event-B method** is adapted to analyse **discrete systems**.
  - offers the possibility of modelling **discrete behaviors**.



# THE EVENT-B METHOD

## THE STATE OF A MODEL

- A discrete model is first made of a **state**
- The state is represented by some **constants** and **variables**
- Constants are linked by some **properties**
- Variables are linked by some **invariants**
- Properties and invariants are written using **set-theoretic expressions**

# THE EVENT-B METHOD

## THE EVENTS OF A MODEL (TRANSITIONS)

- A discrete model is also made of a number of events
- An event is made of a guard and an action
- The guard denotes the enabling condition of the event
- The action denotes the way the state is modified by the event
- Guards and actions are written using set-theoretic expressions

# THE EVENT-B METHOD

## A MODEL SCHEMATIC VIEW

**CONTEXT**  $ctx_1$   
**EXTENDS**  $ctx_2$

**SETS**  $s$   
**CONSTANTS**  $c$   
**AXIOMS**  
     $A(s, c)$   
**THEOREMS**  
     $T(s, c)$   
**END**

**MACHINE**  $mch_1$   
**REFINES**  $mch_2$   
**SEES**  $ctx_i$

**VARIABLES**  $v$   
**INVARIANTS**  
     $I(s, c, v)$   
**THEOREMS**  
     $T(s, c, v)$   
**EVENTS**  
     $[events\_list]$   
**END**

*event*  $\hat{=}$   
any  $x$   
where  
     $G(s, c, v, x)$   
then  
     $BA(s, c, v, x, v')$   
end

# THE EVENT-B METHOD

## OPERATIONAL INTERPRETATION

```
Initialize;  
while (some events have true guards) {  
    Choose one such event;  
    Modify the state accordingly  
}
```

- An event execution is supposed to **take no time**
- Thus, **no two events can occur simultaneously**
- When all events have false guards, the **discrete system stops**
- When some events have true guards, **one of them** is chosen non-deterministically and **its action modifies the state**
- The previous phase is **repeated** (if possible)

# THE EVENT-B METHOD

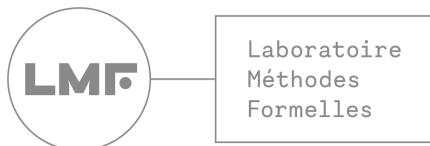
## COMMENTS ON THE OPERATIONAL INTERPRETATION

- Stopping is not necessary: a discrete system may run for ever
- This interpretation is just given here for informal understanding
- The meaning of such a discrete system will be given by the proofs which can be performed on it

# BUILDING LARGE COMPUTERIZED SYSTEMS

## REFINEMENT

- Refinement allows us to build model *gradually*
- We shall build an *ordered sequence* of more precise models
- Each model is a *refinement* of the one preceding it
- A useful analogy: looking through a *microscope*
- *Spatial* as well as *temporal* extensions
- *Data refinement*



# OUTLINE

- The Event-B method
- The Pro-B animator/model-checker
- The Theory plugin

[Back to the outline](#) - [Back to the begin](#)

# OUTLINE

- The Event-B method
- The Pro-B animator/model-checker
- The Theory plugin

[Back to the outline](#) - [Back to the begin](#)

# THANK YOU

[PDF version of the slides](#)

[Back to the begin](#) - [Back to the outline](#)

