

Two B methods : Classical B and Event B

Marielle Petit-Doche

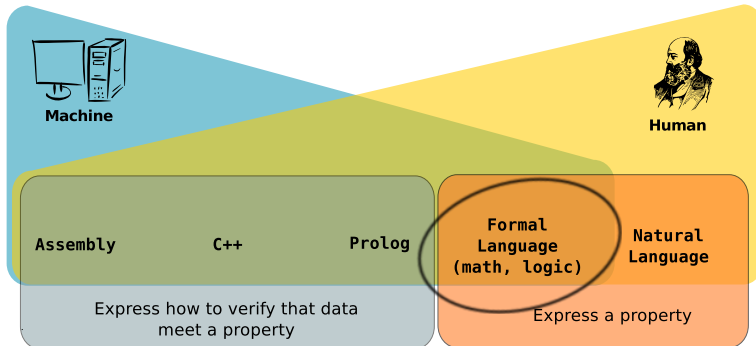
Systerel

April 16th, 2013

Work licensed under Creative Common Attribution-ShareAlike 3.0 Unported License



Why a formal language ?



Best trade-off between human and machine for expression of properties

Two B approaches

- ▶ Both languages developed by Jean-Raymond Abrial
- ▶ *Classical B* (in the late 1980s) :
how the system works
⇒ functional description of a software
- ▶ *Event B* (in the late 1990s) :
why the system works
⇒ system safety requirements
- ▶ Same basic notation based on first order logic and set theory

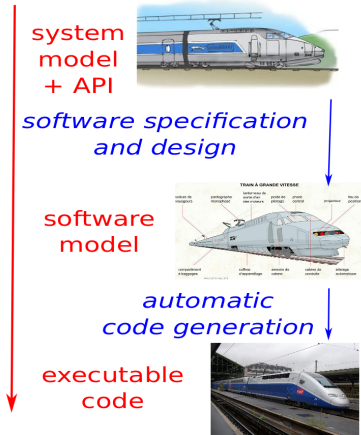
Classical B

Main usage in railway industry

Classical B

Software Design :

- ▶ full functional description
- ▶ some safety constraints
- ▶ structured model
- ▶ refinement
- ▶ deterministic model to generate code (translation)



Classical B approach

- ▶ Language : first order logic + set theory (inherited from Z notation, Hoare logic,...)
- ▶ Structured models : Modules to describe the software architecture
- ▶ Operations, Variables, Invariants
- ▶ Correct by construction approach
- ▶ Formal proof (partially interactive)
- ▶ Automatic code translation (C, Ada,...)
- ▶ Industrial tool : Atelier B (now free but closed source)

Summary

- + Mature approach
- + Well adapted to design critical software : from software informal specification to automatic code generation
- + Easy to maintain
- + Formal verification
- + Industrial approaches and tools adapted to EN50128 requirements
- Methodology and Know-how not shared
- Stable input needed
- Industrial tools are not open source

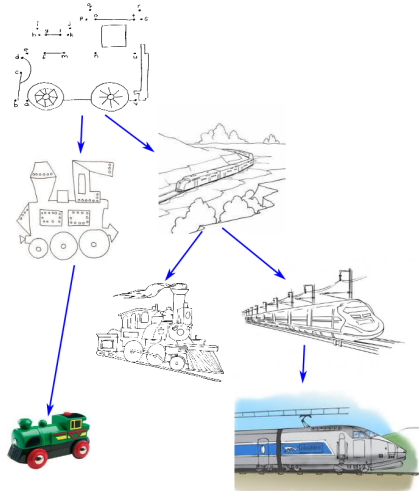
Event B

How to use the event B language

Event B

System analysis :

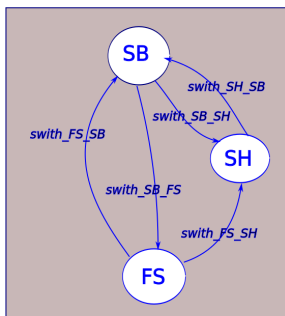
- ▶ initial non deterministic model
- ▶ gradual refinements : several levels of abstraction
- ▶ proof of safety requirements
- ▶ decomposition in sub-systems



Event B in a quick view

- ▶ Seminal article : B Conference 1996
- ▶ *Modeling in Event-B* [Abrial2010]
- ▶ Same basic formal language as classical B
- ▶ Consistency of the model
- ▶ Choice of abstraction level
- ▶ Formal proof (interactive and automatic)
- ▶ Tool : Rodin : open-source tool based on Eclipse
- ▶ www.event-b.org

Approach : System behaviour via Events



VARIABLES

State

INVARIANTS

$State \in \{SB, SH, FS, IS\}$

EVENTS

...

Event *switch_SB_SH* $\hat{=}$

when

State = *SB*

then

State := *SH*

end

Event *switch_SB_FS* $\hat{=}$

when

State = *SB*

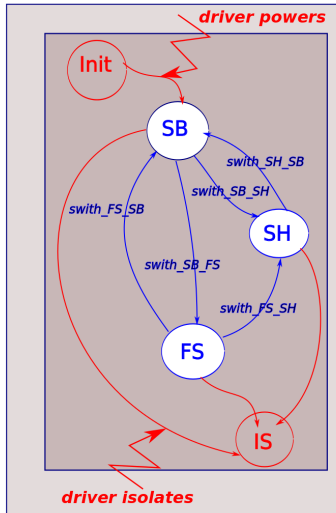
then

State := *FS*

end

...

Closed model : Controller + environment



VARIABLES

State

INVARIANTS

$State \in \{SB, SH, FS, IS\}$

EVENTS

Initialisation

begin

init_State : $State := SB$

end

...

Event *driver_isolates* $\hat{=}$

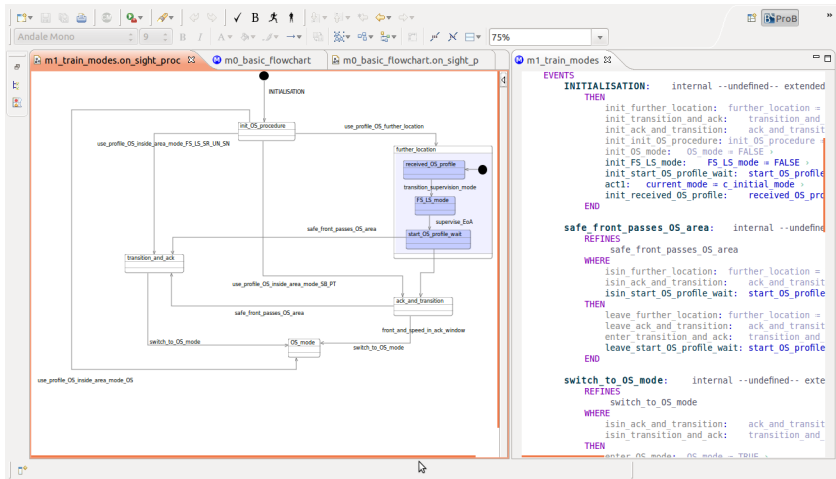
begin

enter_IS : $State := IS$

end

END

State machine



Requirement traceability

```

ms_global_inputs  #subset-final.req#  ms_safe_radio  #2
END
receive_information_compatible: extended ordinary i
REFINES
  receive_information_compatible
ANY
  l_partner >
WHERE
  grd3: l_partner # contacted not theorem >
  grd4: l_partner # system_version_compatible not theorem >
  grd5: l_partner # ER_connections not theorem >
THEN
  act1: outgoing_sessions = outgoing_sessions u
  act2: contacted = contacted \ {l_partner} >
END

receive_information_incompatible: extended ordinary i
REFINES
  receive_information_incompatible
ANY
  l_partner >
WHERE
  grd3: l_partner # contacted not theorem >
  grd4: l_partner # system_version_compatible not theorem >
  grd5: l_partner # ER_connections not theorem >
THEN
  act1: outgoing_sessions = outgoing_sessions u
  act2: contacted = contacted \ {l_partner} >
  act3: terminating_sessions = terminating_sessions u
END

receive_contact_order_accept: extended ordinary i
REFINES
  receive_contact_order_accept

```

*Subsection 26, Chapter 3.5

| Section | Description | Type | WRSMP | Source | Target | Link |
|----------|---|------|-------|--------|-----------|-----------------|
| 1.49 | 3.5.3.8 When the on-board receives the system version it shall consider the communication session established and: | | | | | |
| 1.50 | a) If one of its supported system versions is compatible with the one sent by trackside, it shall send a session established report, including its telephone numbers, to the trackside. | | | | 0 > 0 > 1 | |
| | b) If none of its supported system versions is compatible with the one sent by trackside, it shall send a version independent message indicating "No compatible version supported". It shall inform the driver and shall terminate the communication session. | | | | 0 > 0 > 1 | |
| 1.51 | When the trackside receives the session established report or the information that no compatible system version is supported by the on-board, it shall consider the communication session established. | | | | | receive_informa |
| 1.52 | 3.5.3.9 Intentionally deleted. | | | | | |
| 1.53 | 3.5.3.9. Intentionally deleted. | | | | | |
| 1.54 | <pre> sequenceDiagram participant On-Board participant Trackside Note over On-Board: Set up of the safe connection According to EUROADIO specifications On-Board->>Trackside: Initiation of a communication session Trackside-->>On-Board: RBC/RSU System Version On-Board->>Trackside: Session established Trackside-->>On-Board: Communication session established for trackside </pre> | | | | | |
| 1.55 | Figure F1: Establishment initiated by on-board | | | | | |
| 3.5.3.10 | If the establishment of a communication session is initiated by the RBC, it shall be performed according to the following: | | | | | |

VnV : Animation or Simulation (1)

The screenshot displays the ProB Animation Editor interface. The main window shows the Event B code for the file `m0_basic_flowchart.o`. The code is structured as follows:

```

init_OS_mode: OS_mode = FALSE >

safe_front_passes_OS_area: internal --undefined-- not extended
WHERE
  isin_further_location: further_location = TRUE TYPING --u
  isin_ack_and_transition: ack_and_transition = TRUE TYPING
THEN
  leave_further_location: further_location = FALSE >
  leave_ack_and_transition: ack_and_transition = FALSE >
  enter_transition_and_ack: transition_and_ack = TRUE >
END

switch_to_OS_mode: internal --undefined-- not extended ordinal
WHERE
  isin_ack_and_transition: ack_and_transition = TRUE TYPING
  isin_transition_and_ack: transition_and_ack = TRUE TYPING
THEN
  enter_OS_mode: OS_mode = TRUE >
  leave_ack_and_transition: ack_and_transition = FALSE >
  leave_transition_and_ack: transition_and_ack = FALSE >
END

front_and_speed_in_ack_window: internal --undefined-- not extended
WHERE
  isin_further_location: further_location = TRUE TYPING --u
THEN
  enter_ack_and_transition: ack_and_transition = TRUE >
  leave_further_location: further_location = FALSE >
END

use_profile_OS_further_location: internal --undefined-- not extended
WHERE
  isin_init_OS_procedure: init_OS_procedure = TRUE TYPING --
THEN
  leave_init_OS_procedure: init_OS_procedure = FALSE >
  enter_further_location: further_location = TRUE >

```

The right-hand pane, titled "Animation Editor", contains two panels: "Events" and "Variables and constants".

Events Panel:

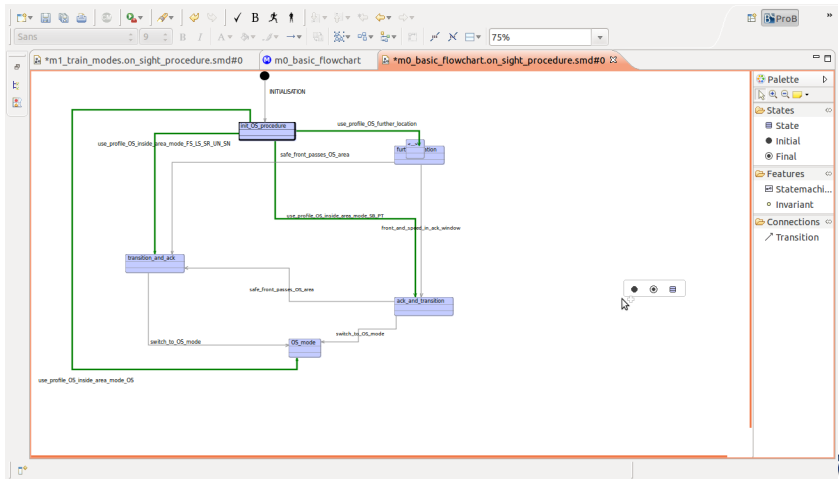
| Ex | ExR | Name |
|----|-----|---|
| ▶ | | INITIALISATION |
| ▶ | ○ | safe_front_passes_OS_area |
| ▶ | ○ | switch to OS mode |
| ▶ | ○ | front and speed in ack window |
| ▶ | ○ | use_profile_OS_further_location |
| ▶ | ○ | use_profile_OS_inside_area_mode_OS |
| ▶ | ○ | use_profile_OS_inside_area_mode_SB_PT |
| ▶ | ○ | use_profile_OS_inside_area_mode_FS_LS_S |

Variables and constants Panel:

| Name |
|--------------------|
| init_OS_procedure |
| further_location |
| ack_and_transition |
| transition_and_ack |
| OS_mode |

The bottom status bar indicates the current file is `m0_basic_flowchart`.

VnV : Animation or Simulation (2)



VnV : Formal Proof : Principle

Goal :

- ▶ Show the model is sound
- ▶ Verify properties

Kind of proof obligations :

- ▶ Invariant preservation
- ▶ Non-deterministic action feasibility
- ▶ Guard strengthening (refinement)
- ▶ Well-definedness
- ▶ ...

VnV : Formal Proof : Means

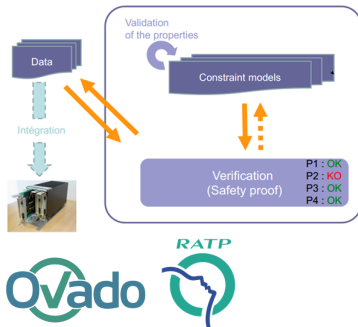
The screenshot displays the Rodin IDE interface, which is used for developing formal proofs in Event-B. The interface is divided into several panes:

- Proof Tree (Left):** A hierarchical view of the proof structure, showing goals and their relationships. The current goal is highlighted.
- Goal (Center):** The main workspace for developing the proof. It shows the goal statement: `partition({TRUE},{init_timer}n{TRUE},{wait_brake_ack}n{TRUE},{brake_until_ack}n{TRUE})`. Below the goal, there are sections for "Selected Hypotheses" and "Goal".
- Event-B Explorer (Right):** A pane showing the Event-B model being proved. It includes a tree view of the model's components, such as `c0_train_mode`, `m4_timer`, and `m3_driver_ack`. The "Proof Obligations" section lists the specific obligations that need to be proved.
- Proof Control (Bottom):** A pane for managing the proof process, including buttons for "New current obligation" and "Rodin Problems".

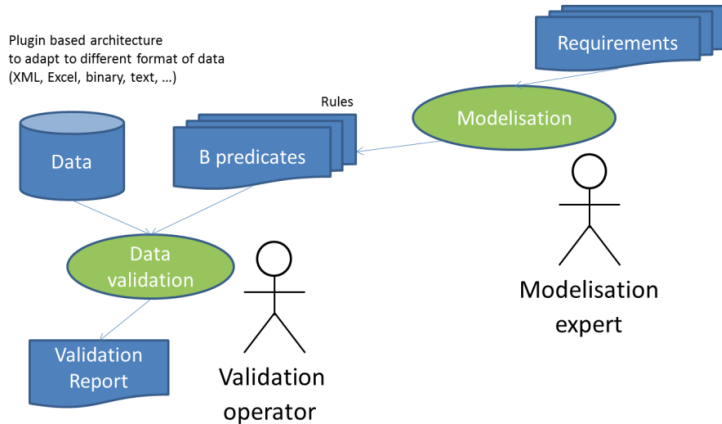
The interface also includes a standard toolbar at the top and a symbols pane at the bottom right.

VnV : Data Validation (1)

- ▶ Generic systems
- ▶ Lots of data to validate
- ▶ Separate data preparation from data validation
- ▶ Formal model of properties
- ▶ Industrial applications
- ▶ DS-Event-B [Badeau13]
- ▶ ovado.fr

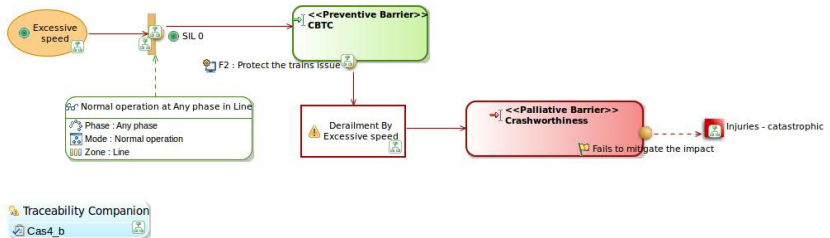


VnV : Data Validation (2)

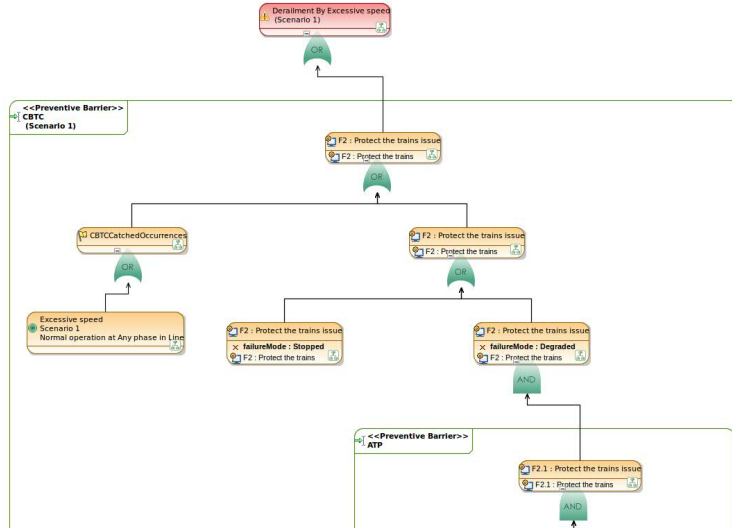


Safety : IMOFIS Safety case

[ERTS2012]



Safety : IMOFIS Safety tree



Safety : IMOFIS requirement management

[ERTS2012]

The screenshot displays the ERTS2012 tool interface. The main window shows the requirements for 'm2_Train_immobile' under the 'departure.requirement' tab. The requirements are organized into sections: MACHINE, REFINES, VARIABLES, INVARIANTS, and EVENTS. The INVARIANTS section contains two theorems: 'inv1: Train immobile = BOOL not theorem' and 'UsNB-02: Le CBTC assure la sécurité des passagers lors des échanges voyageurs (montée et descente).'. The EVENTS section contains an 'INITIALISATION' block with the text 'internal extended ordinary'.

Below the main window, there is a toolbar with icons for 'Requirements', 'Properties', 'Problems', and 'Error Log'. The 'Requirements' tab is active, showing a table of requirements. The table has columns for ID, Name, Category, Derive, and SIL. The first row shows 'UserNeed-02' with the name 'Echanges voyageurs', category 'UserNeed', derive '[UserNeed-01]', and SIL 'SIL 3'.

| ID | Name | Category | Derive | SIL |
|-------------|--------------------|----------|---------------|-------|
| UserNeed-02 | Echanges voyageurs | UserNeed | [UserNeed-01] | SIL 3 |

Safety : IMOFIS event B and proof

*Cas3_a dashboard(Risk Analysis Cas3_a) departure.requirement

platform:/resource/fr.systemel.imofis.sample.lot3.thomas/Departure/Requirements/departure.requirement

Model

- platform:/resource/fr.systemel.imofis.sam
 - Departure
 - UserNeed
 - Arrets commerciaux
 - Echanges voyageurs**
 - SyRS
 - ATC_SyRS
- platform:/resource/fr.systemel.imofis.sam
- platform:/resource/imofis_Cas3_2/m1_Pc
- platform:/plugin/fr.obeo.viewpoint.safety
- platform:/resource/imofis_Cas3_2/m2_Tr

Requirement **Refine** **Satisfy** **Rationale** **Documentation** **Event-B**

Event-B elements

MatchedBy:

| Label | Proved |
|---------------------------|-----------------------|
| m1_Portes/grd1 | -no proof associated- |
| m2_Train_immobile/UsNB-02 | 80.0% |
| m1_Portes/inv1 | -no proof associated- |

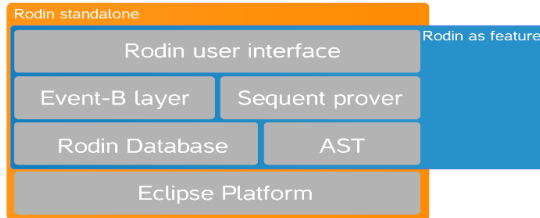
Derived EventB elements

DerivedMatchedBy:

| Label | Proved |
|----------------|-----------------------|
| m1_Portes/inv1 | -no proof associated- |

Rodin platform (1)

www.event-b.org



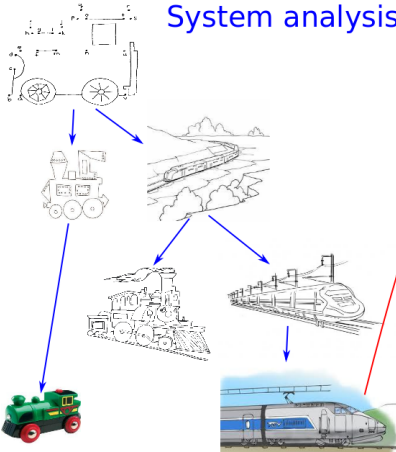
Summary

- + Well adapted to system analysis
- + Abstraction definition
- + Open source approach
- + Possible links to other (semi)-formal approaches
- + Graphical support for design
- + Simulation of the model
- + Extensible approach and platform
 - Little industrial experience
 - Not adapted to software design

Event B / Classical B

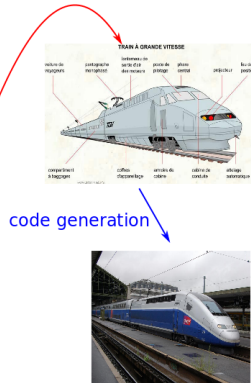
Event B

System analysis



several levels of abstraction

Classical B Software Design

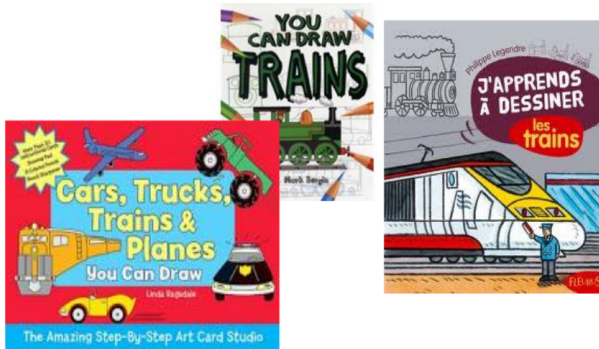


one deterministic
model

Guideline definition

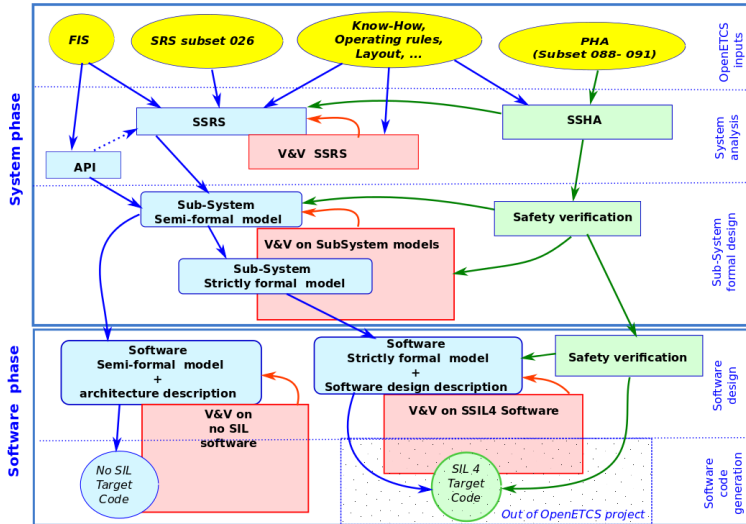
Event B
System analysis

Classical B
Software Design



You need a guideline !

Event B / Classical B in OpenETCS project



Event B / Classical B in OpenETCS project

