Heinrich Heine
Universität
Düsseldorf

ProB

Harnessing the Power of Prolog to Bring Formal Models and Mathematics to Life

Michael Leuschel and the STUPS Team
University of Düsseldorf

What's in a name?

Prolog

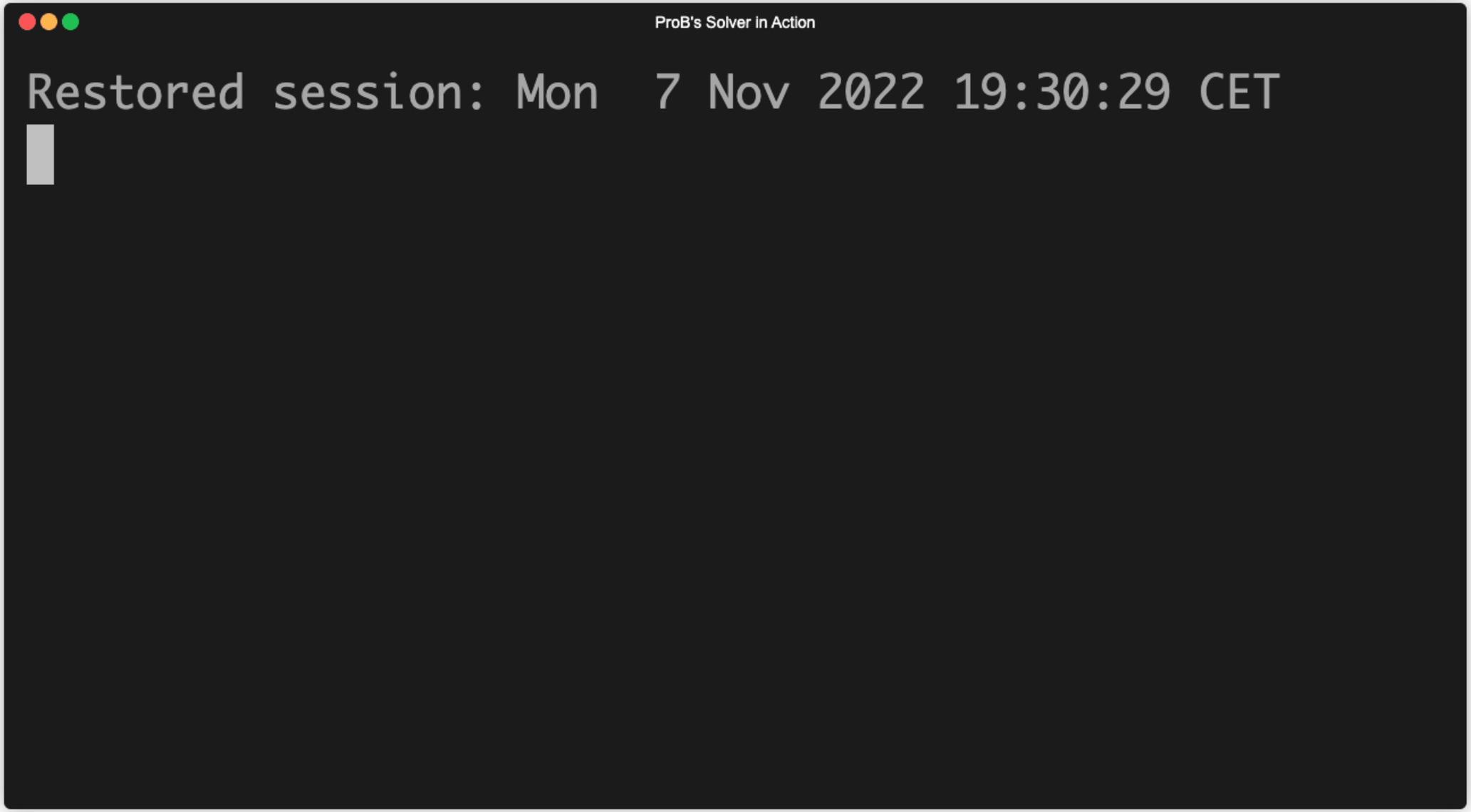
B

A validation tool for the formal method: B

Prolog

Built on a constraint solver for predicate logic,
set theory and arithmetic,
written in SICStus Prolog

ProB's Solver in Action



ProB's Prolog Constraint Solver

Demo using Jupyter

Prolog Day 2022

We highlight some of the features of ProB's constraint solving kernel written in Prolog, dealing with unbounded arithmetic, higher-order and infinite sets:

Some Features

Automatically detecting infinite sets:

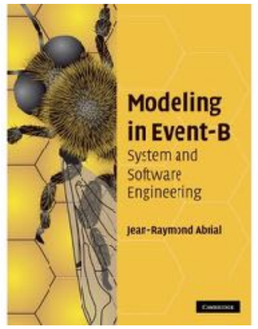
What is the B formal method?

Formal Methods

- Mathematical techniques to produce correct software and systems: B, TLA+, Z, Alloy, CSP, …
- Highly recommended for safety critical applications, e.g. for SIL3/SIL4 railway applications by norm EN50128

B Formal Method

Industrial Applications



Mathematical Foundation

Tool Support

B Formal Method

Line 14

B was used for software for **L14** and **L1**
ProB was used to validate configuration of **L1**

Paris (AFP) – **Labour unions have called a major one-**
threatens to paralyse Paris public transport Thursda
action to demand relief from the French governmen

The RATP transport operator for the capital has warned of particularly severe disruptions for metro and suburban rail lines, with bus and tram services also impacted by the protest for higher wages.

Seven metro lines will be fully closed and another seven will only operate at peak hours, RATP announced.

Only lines 1 and 14 – which are fully automated with no drivers -- would operate normally but risk becoming overcrowded, the RATP said.

https://www.rfi.fr

Origins of B

- Train protection system SACEM for Paris RER Line A, sketch of the B-Method by Jean-Raymond Abrial, 1989 project by Alstom, RATP, SNCF to develop tools and train engineers
- **Paris Metro Line 14** contract won by Matra Transport (now Siemens)
  - **1995**: B tools industrialised by Digilog (now CLEARSY) leading to Atelier-B
  - ready by end 1998: 110 kLOC B model
  - Still in version 1.0, “no single issue caused by software”

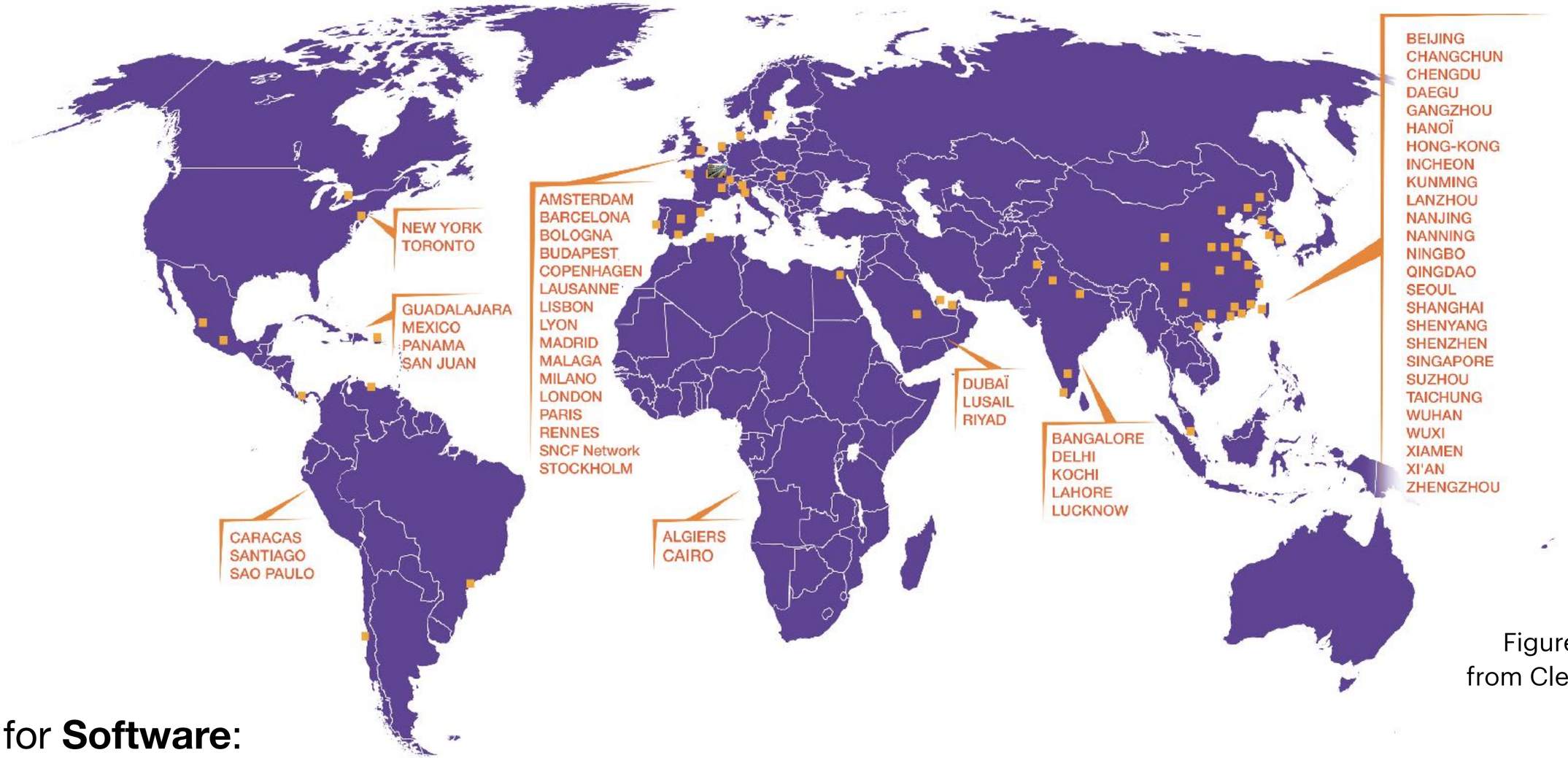


Figure:
from ClearSy

- B for **Software**:
  - about 30% of CBTC systems worldwide employ the B formal method
  - Urbalis 400, Alstom, over 100 metro lines worldwide, 25% of worldwide CBTC market

FM Success Story

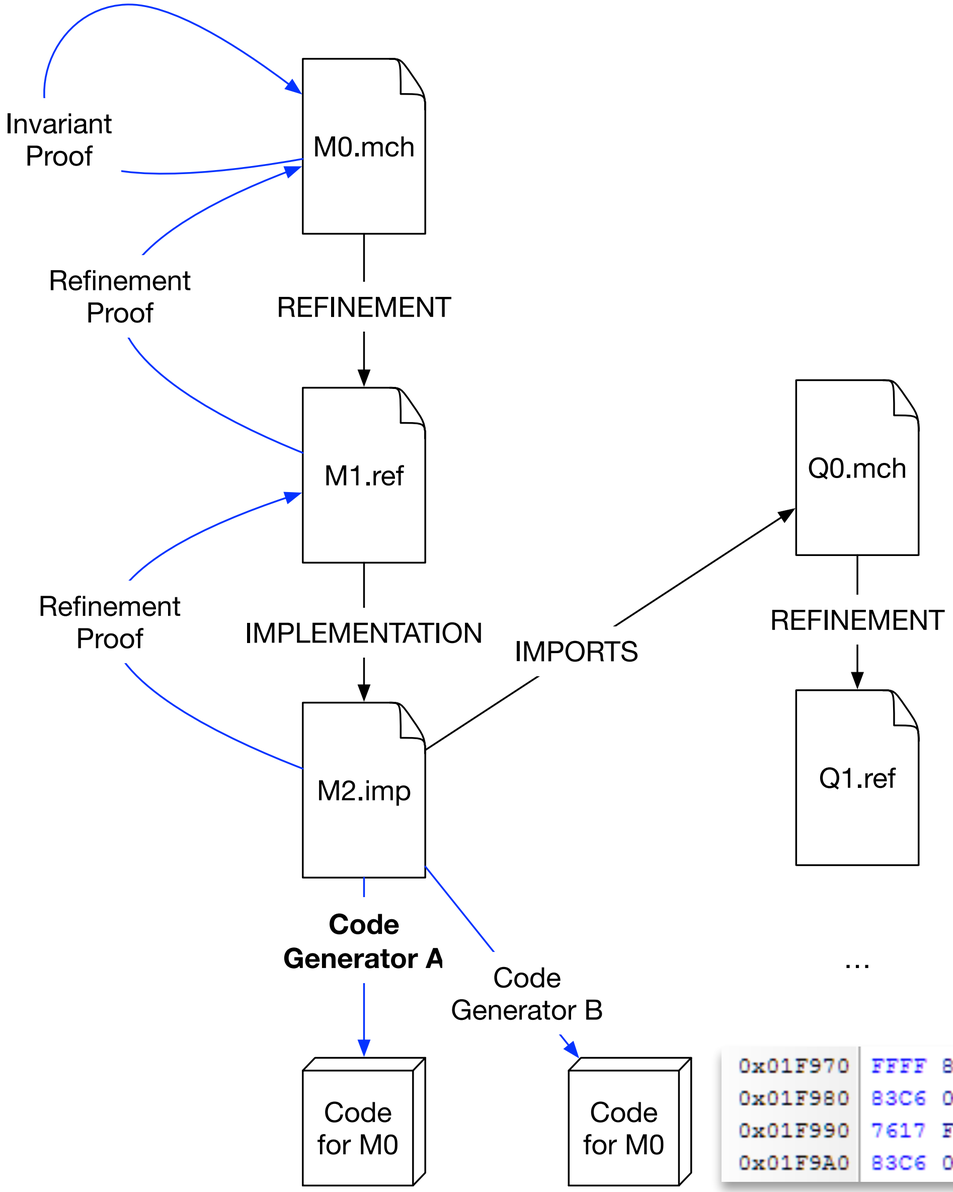
Validation with ProB

B Development Process

High-Level Formal Model

Refinement

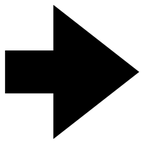
Code Correct by Construction




```
0x01F970  FFFF 8B4C 2440 89C5 8D7D 0C8B
0x01F980  83C6 0C8D 1485 0000 0000 8D42
0x01F990  7617 F7C7 0400 0000 740F 8B41
0x01F9A0  83C6 0489 450C 8D42 04FC 89C1
```

But who guarantees the correctness of the high-level specification?

High-Level Formal Model



Visualisation

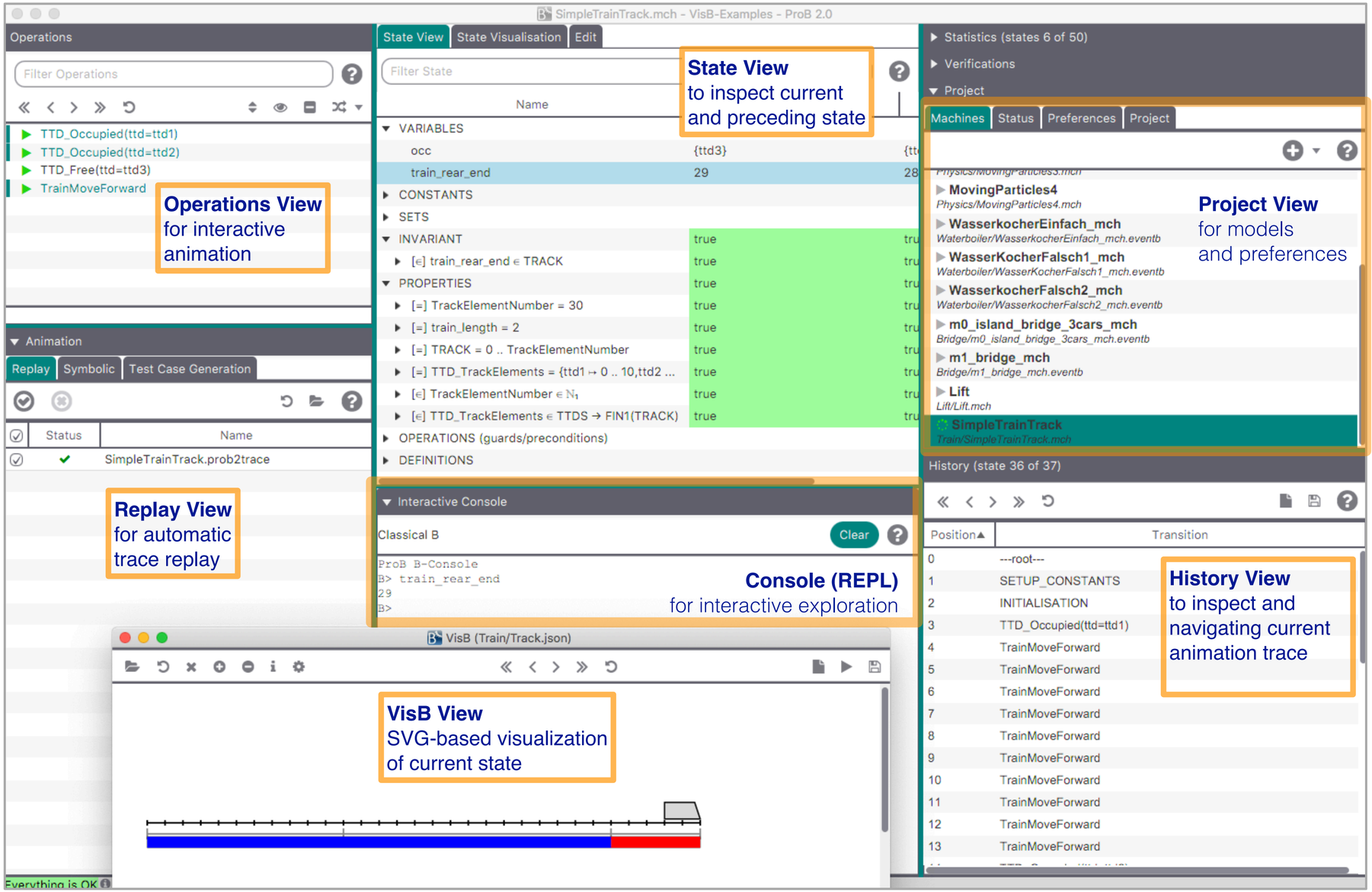
Animation

Model Checking

Refinement

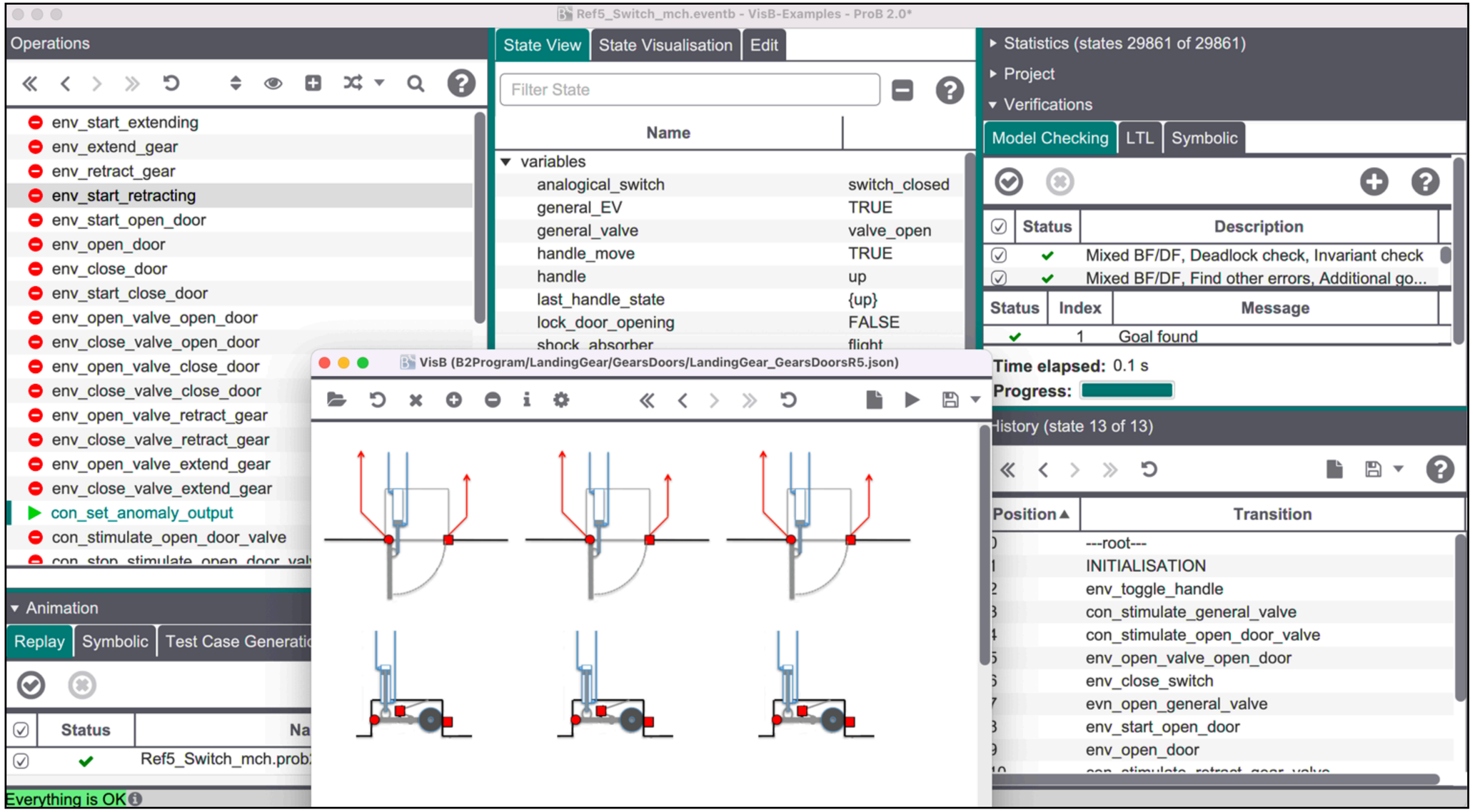
Code Correct by Construction

ProB2-UI



https://prob.hhu.de

Demo


Ref5_Switch_mch.eventb - VisB-Examples - ProB 2.0*
Operations
env_start_extending
env_extend_gear
env_retract_gear
env_start_retracting
env_start_open_door
env_open_door
env_close_door
env_start_close_door
env_open_valve_open_door
env_close_valve_open_door
env_open_valve_close_door
env_close_valve_close_door
env_open_valve_retract_gear
env_close_valve_retract_gear
env_open_valve_extend_gear
env_close_valve_extend_gear
con_set_anomaly_output
con_stimulate_open_door_valve
State View
State Visualisation
Edit
Filter State
Name
variables
analogical_switch switch_closed
general_EV TRUE
general_valve valve_open
handle_move TRUE
handle up
last_handle_state {up}
lock_door_opening FALSE
shock_absorber flight
Statistics (states 29861 of 29861)
Project
Verifications
Model Checking
LTL
Symbolic
Status
Description
Mixed BF/DF, Deadlock check, Invariant check
Mixed BF/DF, Find other errors, Additional go...
Status
Index
Message
1
Goal found
Time elapsed: 0.1 s
Progress:
VisB (B2Program/LandingGear/GearsDoors/LandingGear_GearsDoorsR5.json)
History (state 13 of 13)
Position
Transition
---root---
INITIALISATION
env_toggle_handle
con_stimulate_general_valve
con_stimulate_open_door_valve
env_open_valve_open_door
env_close_switch
evn_open_general_valve
env_start_open_door
env_open_door
Animation
Replay
Symbolic
Test Case Generatio
Status
Ref5_Switch_mch.prob
Everything is OK

Can we use the high-level specification before a complete implementation?

High-Level Formal Model

Refinement

Code Correct by Construction

ProB

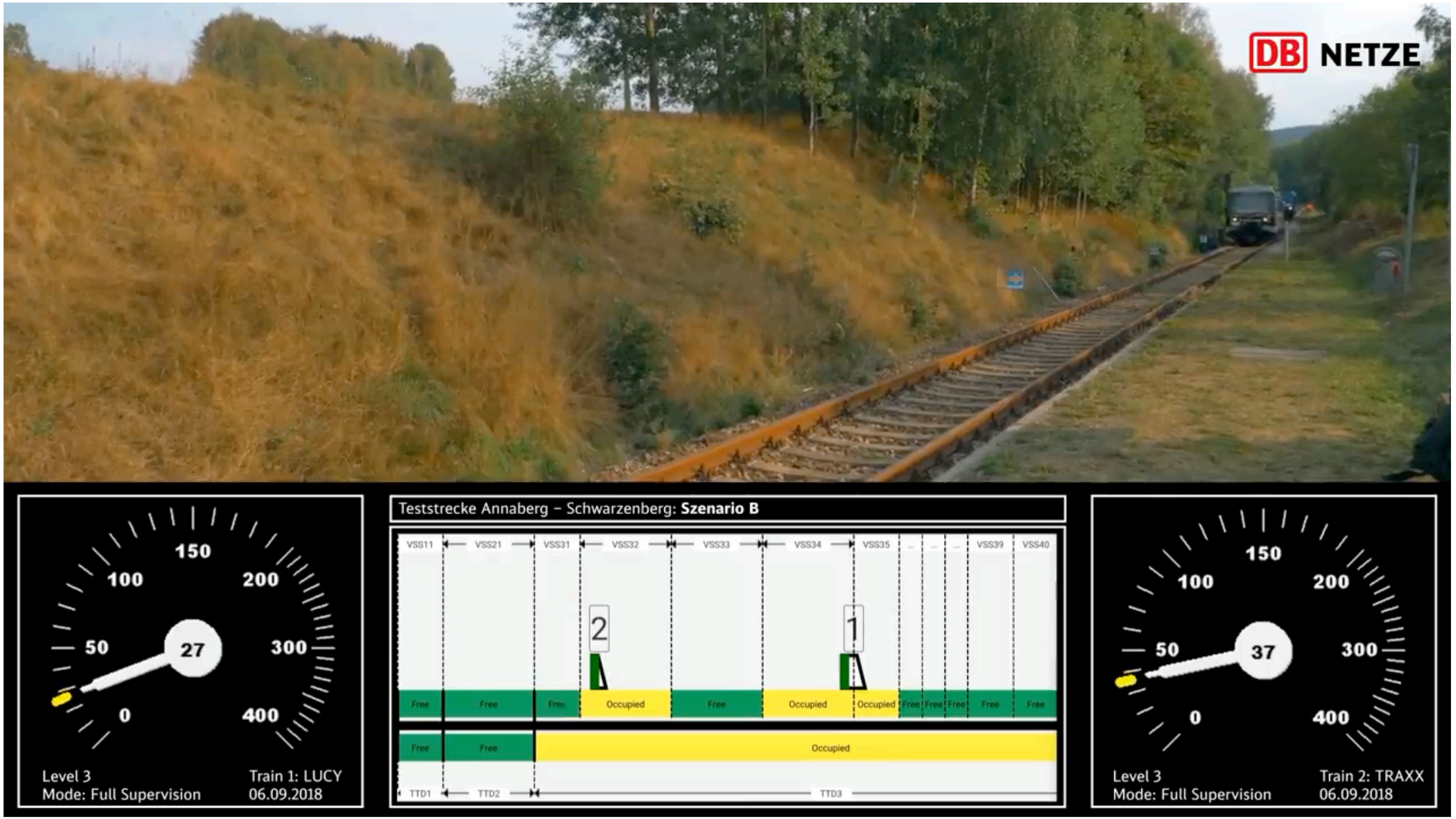
Visualisation

Model Checking

Animation

Execution: Mathematical Model as Runtime Artefact

ProB running in SICStus **Prolog** in real-time executing a formal B model of the Hybrid-Level 3 principles



Train 2 following Train 1 (Lucy) on the same occupied track section, but on different virtual subsections

Source: https://www.youtube.com/watch?v=FjKnugbmrP4

But who guarantees the correct configuration of the final system?

High-Level Formal Model

Refinement

Code Correct by Construction

ProB

Visualisation

Model Checking

Animation

Execution: Mathematical Model as Runtime Artefact

Data Validation: Ensure correct configuration of system

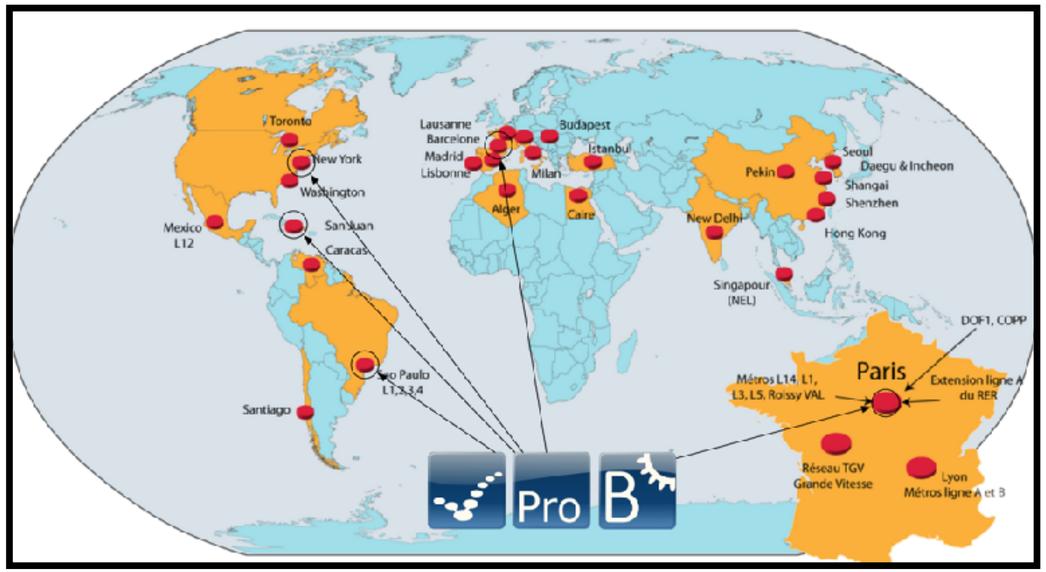
Data Validation at Siemens

One man-month of work done automatically in 3 minutes with ProB and undetected issues found

ProB at Siemens

*The work done with ProB is a **great success for Siemens**:*

- *Thanks to the automatization and **ProB**, the wayside data validation is **quicker**, more **complete** and **easier** than AtelierB.*
- *The on-board data validation, which was not formally proven, is now proven with **ProB**, so the use of formal methods has been extended.*
- *The B experts are only required when problems have been found, whereas before highly skilled people were required for long and fastidious B modification and proof.*
- *The validation of **ProB** itself enables a use in a **SIL4** development.*



Certified as T2 Tool EN50128

Jérôme Falampin, SAS Siemens

cf. Deploy Deliverable D41, June 2011

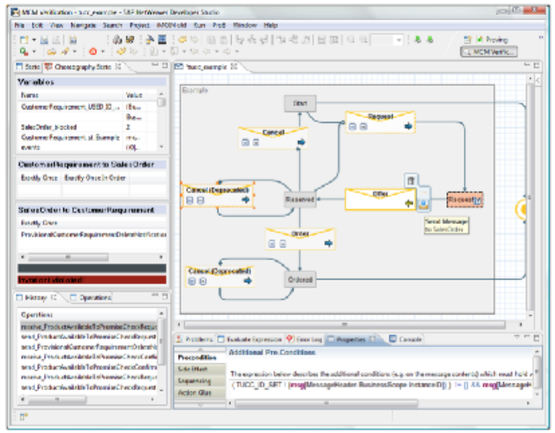
ProB Uses for Data Validation

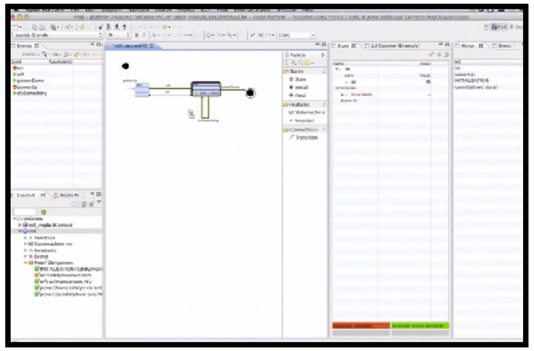
- RDV (Siemens; L1, Barcelona,...): used as primary toolchain
- Ovado, Ovado2 (Systerel, RATP):
  used as secondary toolchain
- DTVT (Clearsy, Alstom),
- Dave (ClearSy, G&E),
- Olaf (ClearSy, SNCF), ... :
  used as primary tool chain

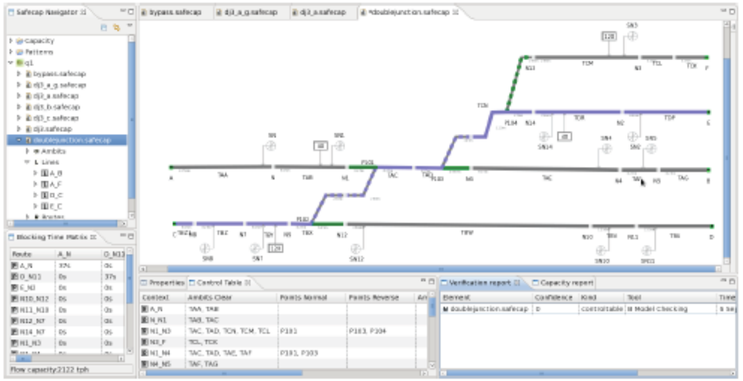
- Caval (ClearSy), Rubin (Thales): currently used as only tool chain

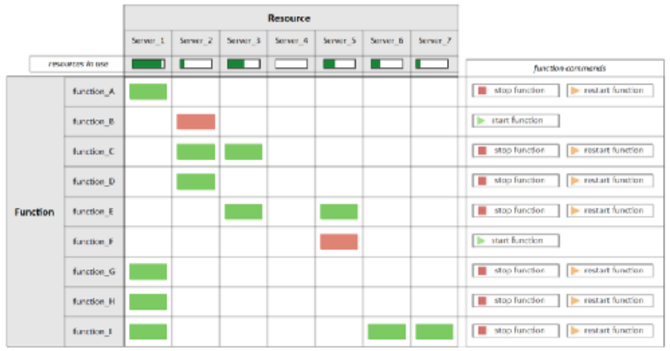
A lot of (other) tools build upon ProB

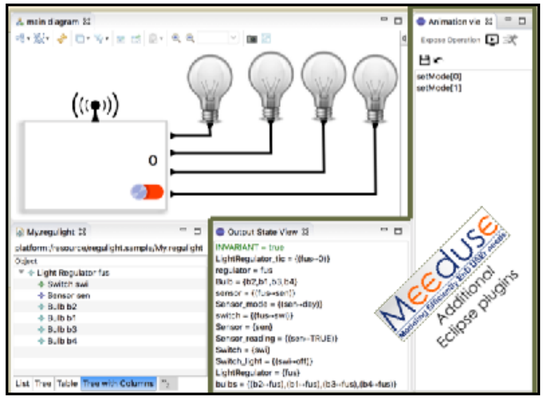
Model checking, simulation, test-case generation, ...

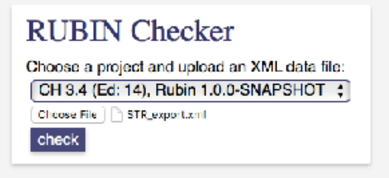


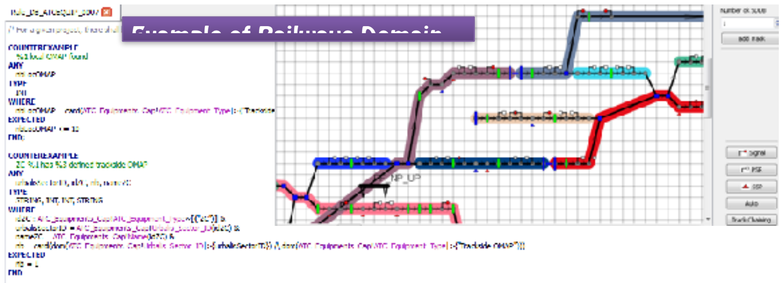












ClearSy Data Solver (Caval), ...

Thanks to Prolog
and Logic Programming

- Convenient to express semantics of specification languages

- Possible to write domain specific solvers, via Prolog’s flexible computation rule

- Compact encoding of analysis, optimisation, type checking and verification rules

- Very fast and robust Prolog systems like SICStus Prolog, with efficient co-routines and fast CLP(FD) library

Conclusion: Prolog

has enabled a tool that

- brings formal mathematics to life
- helps find bugs in safety critical systems
- helps users visualise and understand their models
- is used in industrial applications
- is used for teaching
- is used in research (> 1000 citations)
- is the foundation for many other tools

Thanks for the Support

Alstom (F. Mejia,…)
ClearSy (T Lecomte, R. Lapostelle, E. Mottin,…)
Siemens
Systerel
Thales (N. Nayeri, G. Hemzal,…)

DFG (Gepavas I+II, IVOIRE)
EU (Rodin, Deploy, Advance)

SICStus Prolog (Mats Carlsson, Per Mildner)

hhu.

STUPS Team & Friends

Jens Bendisposto
Carl Friedrich Bolz
Michael Butler
Joy Clark
Ivo Dobrikov
Jannik Dunkelau
Nadine Elbeshausen
Fabian Fritz
Marc Fontaine
Marc Frappier
David Geleßus
Stefan Hallerstede
Dominik Hansen
Christoph Heinzen
Yumiko Jansing
Michael Jastram
Philipp Körner
Sebastian Krings
Lukas Ladenberger
Li Luo
Thierry Massart
Daniel Plagge
Antonia Pütz
Mireille Samia
Joshua Schmidt
David Schneider
Sherin Schneider
Corinna Spermann
Sebastian Stock
Yumiko Takahashi
Edd Turner
Miles Vella
Fabian Vu
Michelle Werth
Dennis Winter