formalmind

science for systems engineering

michael leuschel

jens bendisposto, michael jastram, daniel plagge, lukas ladenberger
thierry lecomte, luis-fernando mejia

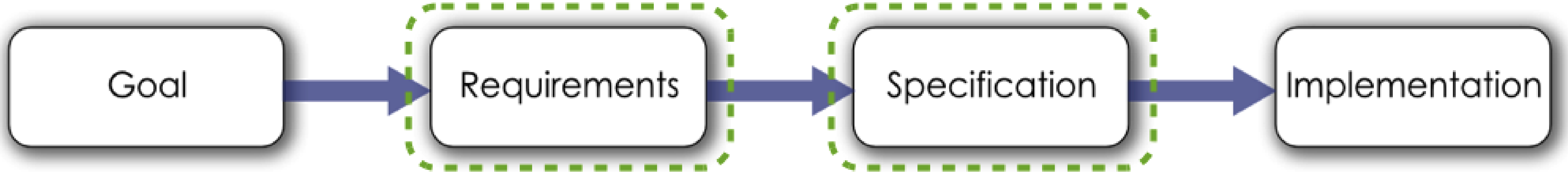
Overview

- 1. formal mind and its tools
- 2. data validation
- 3. the future

background

- gmbh (limited), based in düsseldorf, germany
- spin-off from university of düsseldorf
- expert services: formal verification, requirements management & engineering
- open source software: ProB & ProR

Systems Development today



Requirements Engineering

formalmind

science for systems engineering

ProR

Requirements Engineering Platform

Formal Specifications

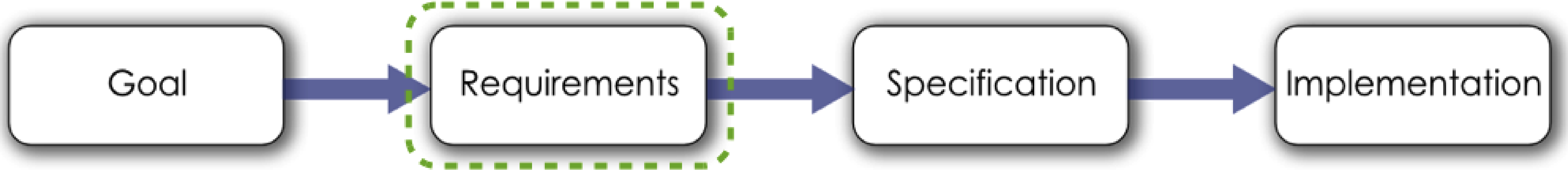
formalmind

science for systems engineering

ProB

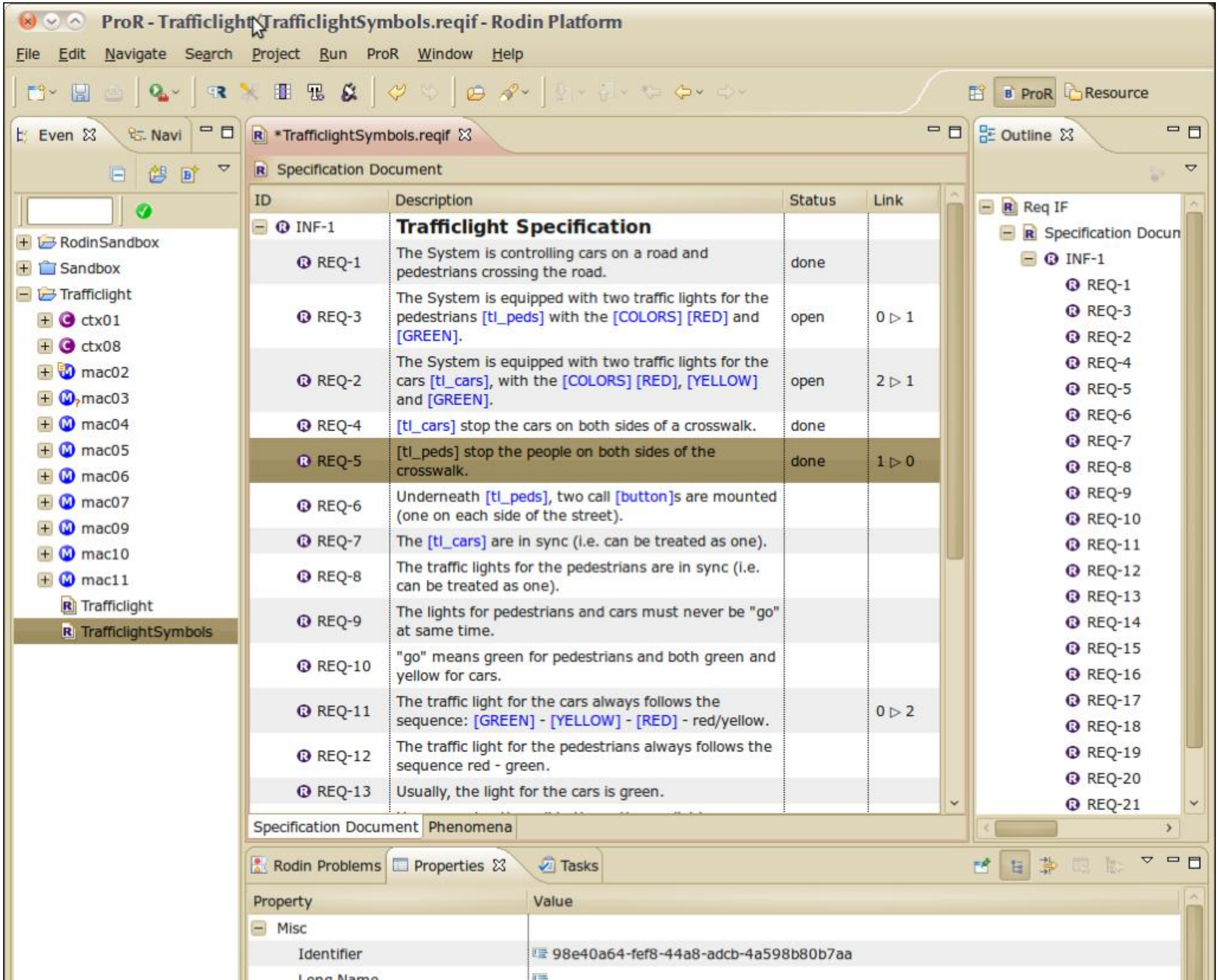
Formal Validation Platform

Requirements Engineering



- ProR
  - Optimizing Communications
  - Integration into existing processes
  - Interoperability with the ReqIF standard

ProR http://pror.org/

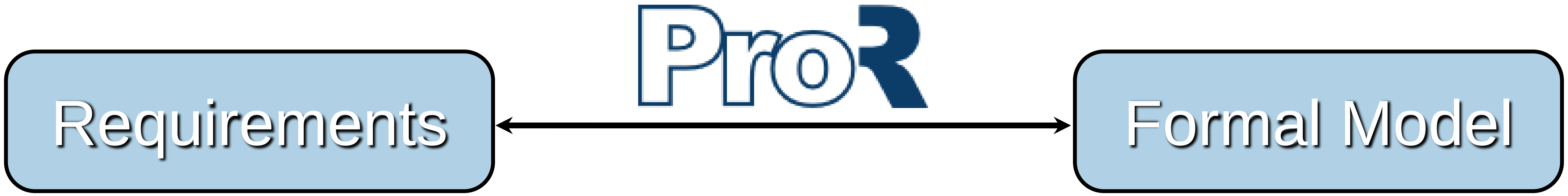


ID	Description	Status	Link
INF-1	**Trafficlight Specification**		
REQ-1	The System is controlling cars on a road and pedestrians crossing the road.	done	
REQ-3	The System is equipped with two traffic lights for the pedestrians [tl_peds] with the [COLORS] [RED] and [GREEN].	open	0 ▷ 1
REQ-2	The System is equipped with two traffic lights for the cars [tl_cars], with the [COLORS] [RED], [YELLOW] and [GREEN].	open	2 ▷ 1
REQ-4	[tl_cars] stop the cars on both sides of a crosswalk.	done	
REQ-5	[tl_peds] stop the people on both sides of the crosswalk.	done	1 ▷ 0
REQ-6	Underneath [tl_peds], two call [button]s are mounted (one on each side of the street).		
REQ-7	The [tl_cars] are in sync (i.e. can be treated as one).		
REQ-8	The traffic lights for the pedestrians are in sync (i.e. can be treated as one).		
REQ-9	The lights for pedestrians and cars must never be "go" at same time.		
REQ-10	"go" means green for pedestrians and both green and yellow for cars.		
REQ-11	The traffic light for the cars always follows the sequence: [GREEN] - [YELLOW] - [RED] - red/yellow.		0 ▷ 2
REQ-12	The traffic light for the pedestrians always follows the sequence red - green.		
REQ-13	Usually, the light for the cars is green.		

Property	Value
Misc	
Identifier	98e40a64-fef8-44a8-adcb-4a598b80b7aa
Long Name	

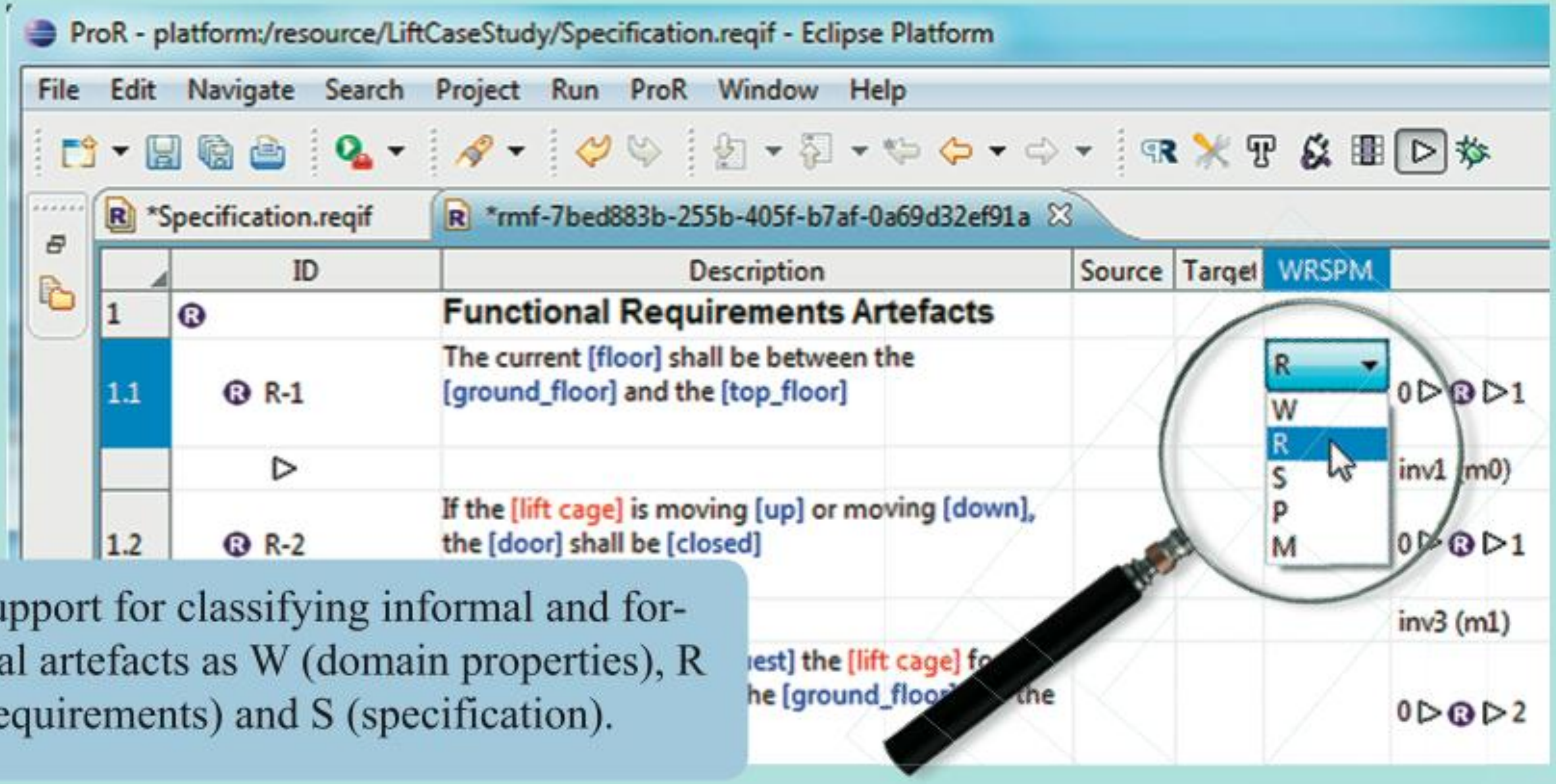
ReqIF support

Traceability to B



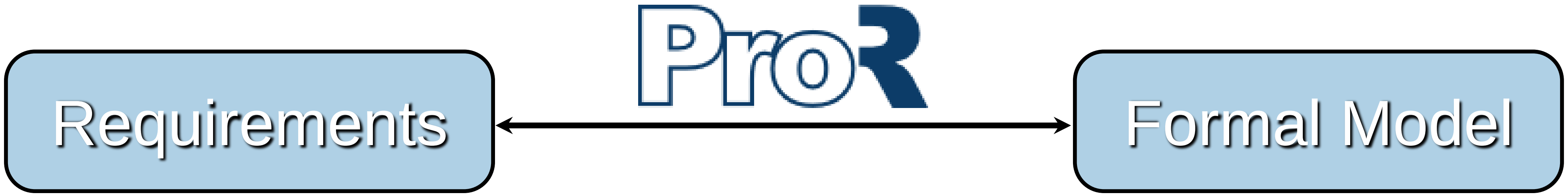
1

Classifying informal and formal artefacts


ProR - platform:/resource/LiftCaseStudy/Specification.reqif - Eclipse Platform
File Edit Navigate Search Project Run ProR Window Help
*Specification.reqif
*rmf-7bed883b-255b-405f-b7af-0a69d32ef91a
R
W
R
S
P
M

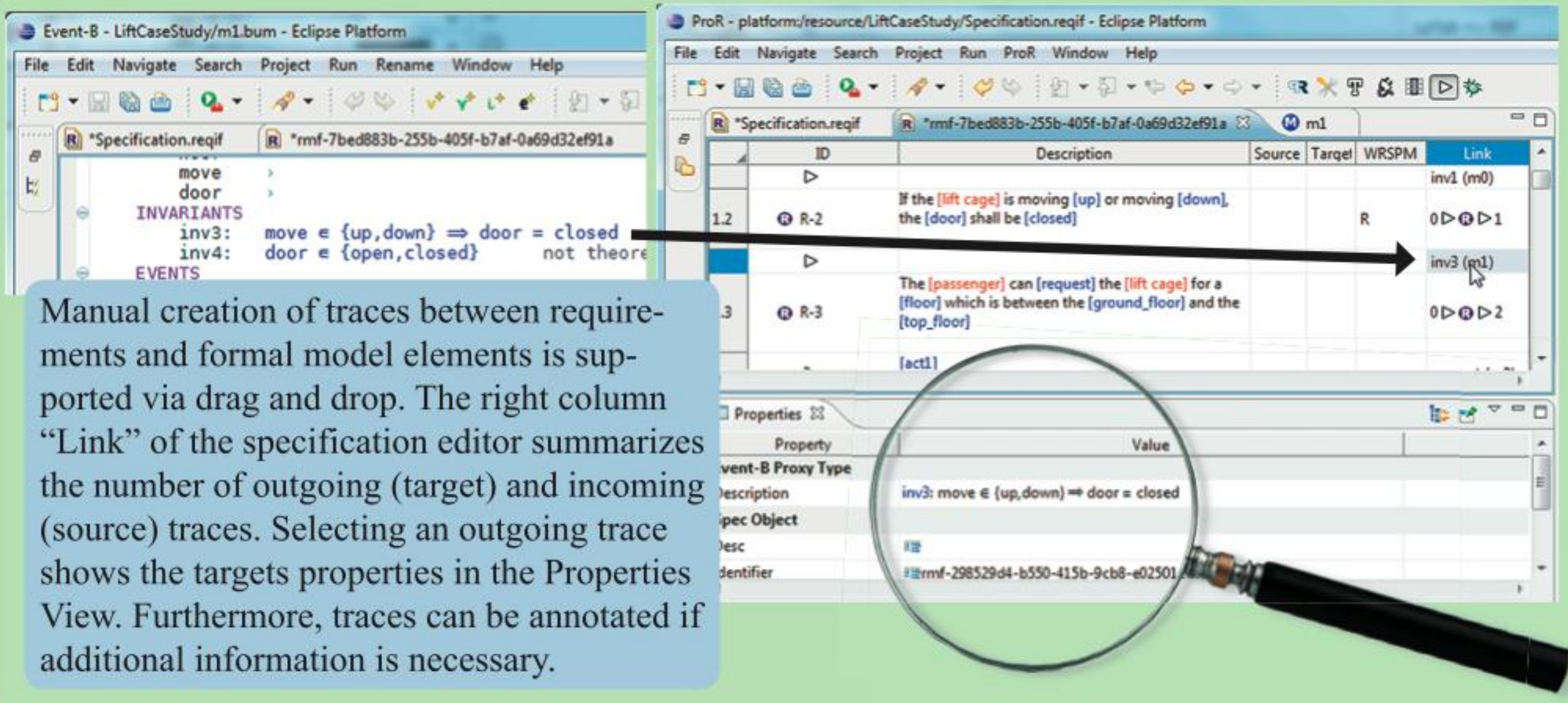
	ID	Description	Source	Target	WRSPM	
1	Ⓡ	**Functional Requirements Artefacts**				
1.1	Ⓡ R-1	The current [floor] shall be between the [ground_floor] and the [top_floor]			R	0 ▷ Ⓡ ▷ 1
	▷					inv1 (m0)
1.2	Ⓡ R-2	If the [lift cage] is moving [up] or moving [down], the [door] shall be [closed]				0 ▷ Ⓡ ▷ 1
						inv3 (m1)
		…est] the [lift cage] f… …he [ground_floo… the				0 ▷ Ⓡ ▷ 2

Support for classifying informal and formal artefacts as W (domain properties), R (requirements) and S (specification).



2

Annotated traces to modelling elements

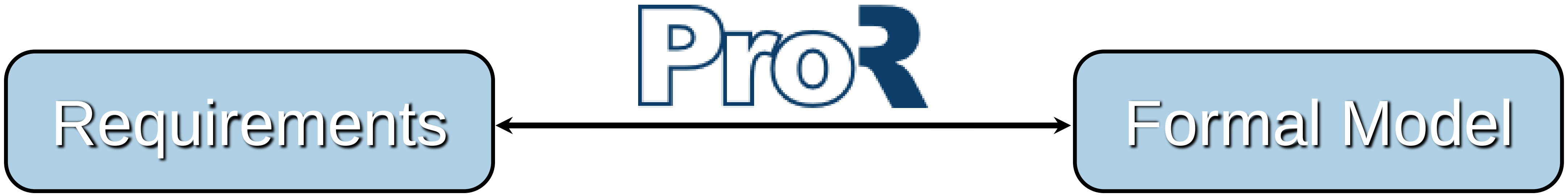

Event-B - LiftCaseStudy/m1.bum - Eclipse Platform
File Edit Navigate Search Project Run Rename Window Help
*Specification.reqif
*rmf-7bed883b-255b-405f-b7af-0a69d32ef91a
ProR - platform:/resource/LiftCaseStudy/Specification.reqif - Eclipse Platform
File Edit Navigate Search Project Run ProR Window Help
*Specification.reqif
*rmf-7bed883b-255b-405f-b7af-0a69d32ef91a
m1
Properties

```
        move    ›
        door    ›
    INVARIANTS
        inv3:   move ∈ {up,down} ⇒ door = closed
        inv4:   door ∈ {open,closed}        not theore
    EVENTS
```

	ID	Description	Source	Target	WRSPM	Link
	▷					inv1 (m0)
1.2	R-2	If the [lift cage] is moving [up] or moving [down], the [door] shall be [closed]			R	0▷R▷1
	▷					inv3 (m1)
.3	R-3	The [passenger] can [request] the [lift cage] for a [floor] which is between the [ground_floor] and the [top_floor]				0▷R▷2
		[act1]				

Property	Value
vent-B Proxy Type	
escription	inv3: move ∈ {up,down} ⇒ door = closed
pec Object	
esc	
dentifier	rmf-298529d4-b550-415b-9cb8-e02501

Manual creation of traces between requirements and formal model elements is supported via drag and drop. The right column “Link” of the specification editor summarizes the number of outgoing (target) and incoming (source) traces. Selecting an outgoing trace shows the targets properties in the Properties View. Furthermore, traces can be annotated if additional information is necessary.

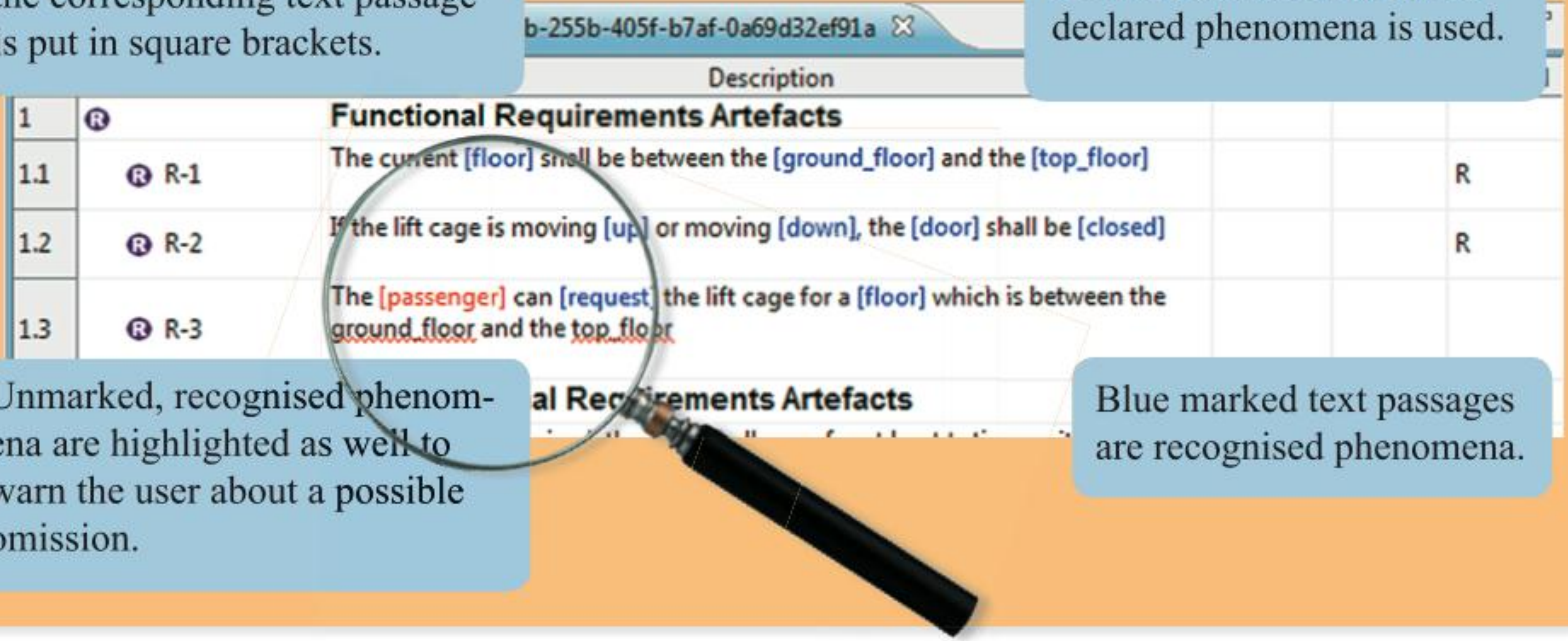


3

Tracing of phenomena used in artefacts

In order to add a uses-trace for an phenomenon to an artefact, the corresponding text passage is put in square brackets.

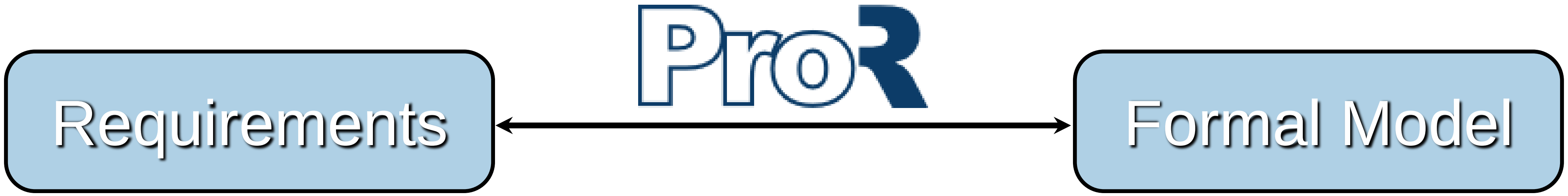
Red marked text passages reminds the user that an un-declared phenomena is used.



		Description			
1	Ⓡ	**Functional Requirements Artefacts**			
1.1	Ⓡ R-1	The current [floor] shall be between the [ground_floor] and the [top_floor]			R
1.2	Ⓡ R-2	If the lift cage is moving [up] or moving [down], the [door] shall be [closed]			R
1.3	Ⓡ R-3	The [passenger] can [request] the lift cage for a [floor] which is between the ground_floor and the top_floor			
		al Requirements Artefacts			

Unmarked, recognised phenomena are highlighted as well to warn the user about a possible omission.

Blue marked text passages are recognised phenomena.



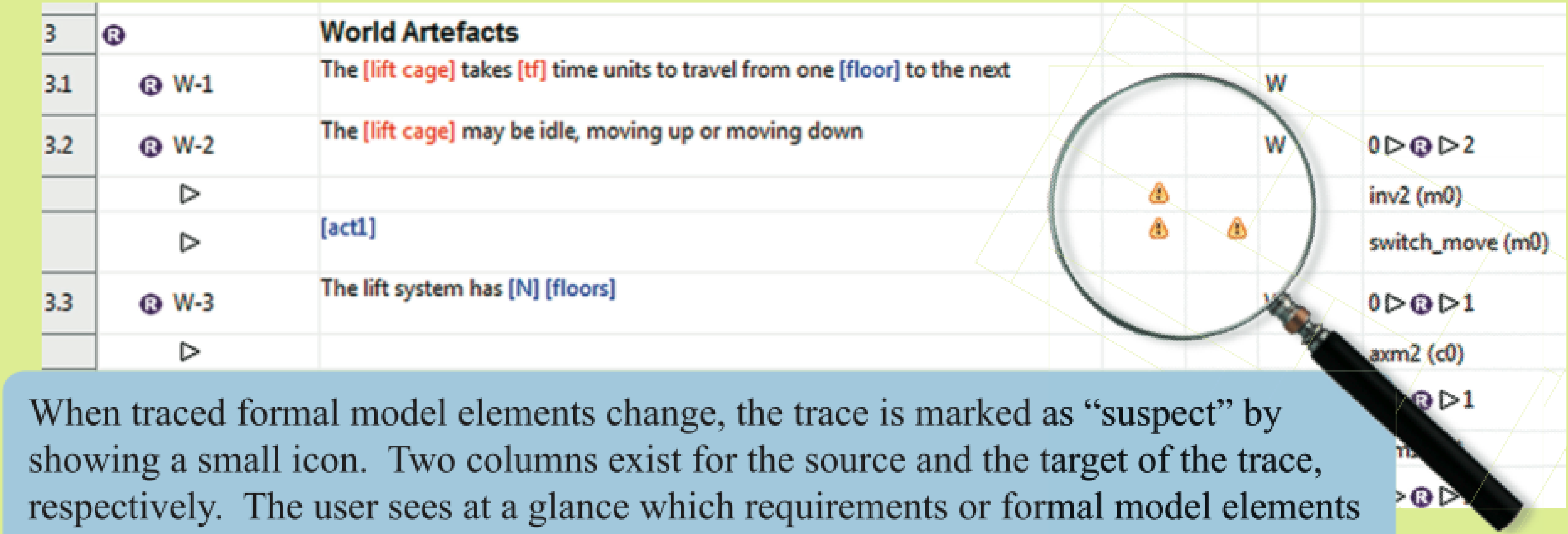
Requirements

ProR

Formal Model

4

Change management

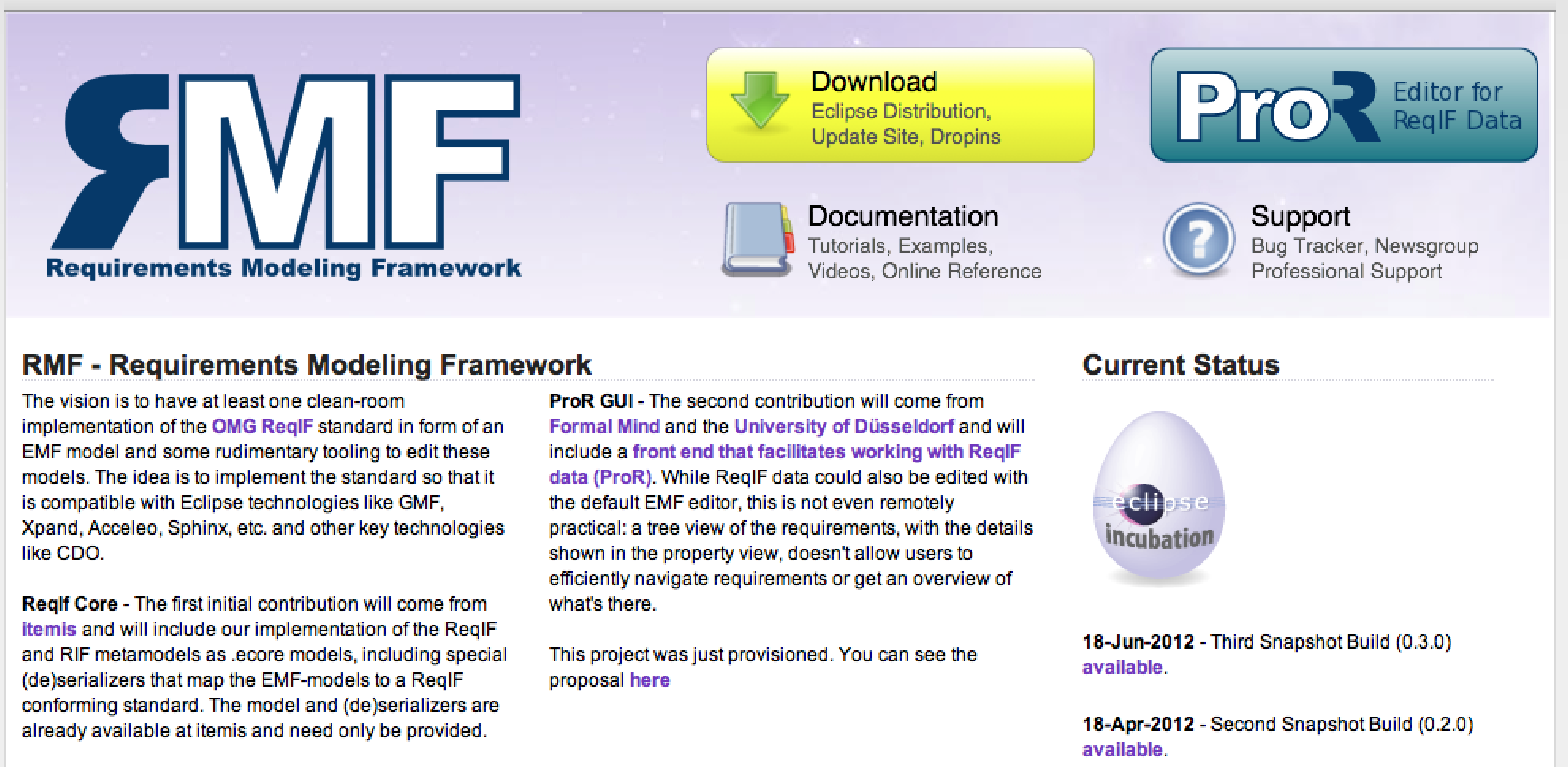


3	Ⓡ		World Artefacts				
3.1		Ⓡ W-1	The [lift cage] takes [tf] time units to travel from one [floor] to the next			W	
3.2		Ⓡ W-2	The [lift cage] may be idle, moving up or moving down			W	0 ▷ Ⓡ ▷ 2
		▷		⚠			inv2 (m0)
		▷	[act1]	⚠	⚠		switch_move (m0)
3.3		Ⓡ W-3	The lift system has [N] [floors]			W	0 ▷ Ⓡ ▷ 1
		▷					axm2 (c0)
							Ⓡ ▷ 1
							Ⓡ ▷

When traced formal model elements change, the trace is marked as "suspect" by showing a small icon. Two columns exist for the source and the target of the trace, respectively. The user sees at a glance which requirements or formal model elements need to be revalidated. This is particularly useful if the requirements document becomes large. By double-clicking on the "suspect" icon, the user can mark the trace as "revalidated" and the icon will be removed.

now: Eclipse Foundation Project !

http://www.eclipse.org/rmf/


RMF
Requirements Modeling Framework
Download
Eclipse Distribution, Update Site, Dropins
ProR
Editor for ReqIF Data
Documentation
Tutorials, Examples, Videos, Online Reference
Support
Bug Tracker, Newsgroup Professional Support
eclipse
incubation

RMF - Requirements Modeling Framework

The vision is to have at least one clean-room implementation of the OMG ReqIF standard in form of an EMF model and some rudimentary tooling to edit these models. The idea is to implement the standard so that it is compatible with Eclipse technologies like GMF, Xpand, Acceleo, Sphinx, etc. and other key technologies like CDO.

Reqlf Core - The first initial contribution will come from itemis and will include our implementation of the ReqIF and RIF metamodels as .ecore models, including special (de)serializers that map the EMF-models to a ReqIF conforming standard. The model and (de)serializers are already available at itemis and need only be provided.

ProR GUI - The second contribution will come from Formal Mind and the University of Düsseldorf and will include a front end that facilitates working with ReqIF data (ProR). While ReqIF data could also be edited with the default EMF editor, this is not even remotely practical: a tree view of the requirements, with the details shown in the property view, doesn't allow users to efficiently navigate requirements or get an overview of what's there.

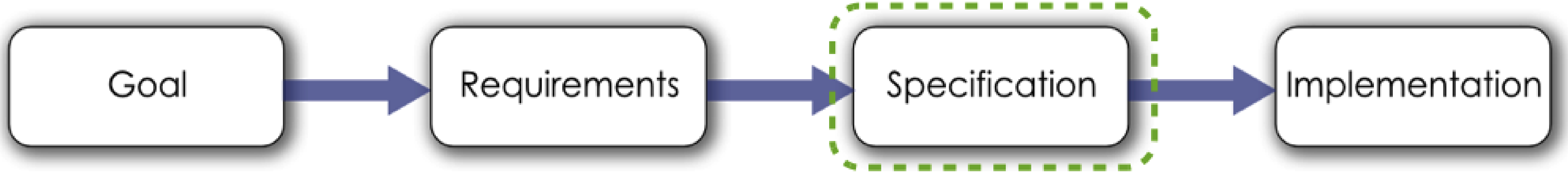
This project was just provisioned. You can see the proposal here

Current Status

18-Jun-2012 - Third Snapshot Build (0.3.0) available.

18-Apr-2012 - Second Snapshot Build (0.2.0) available.

Formal Specifications

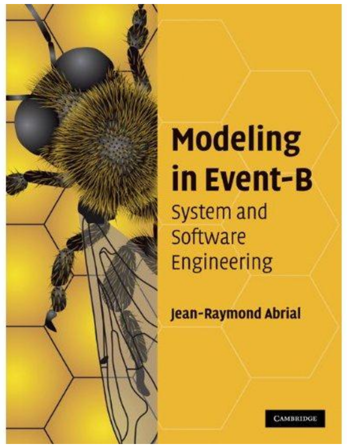

Goal

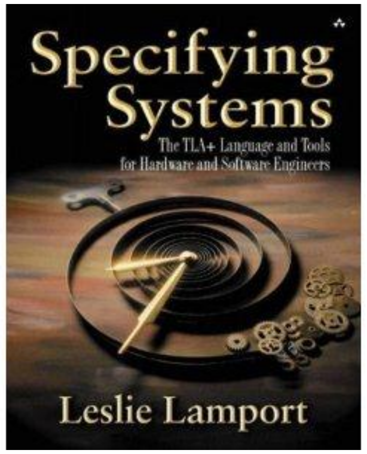
Requirements

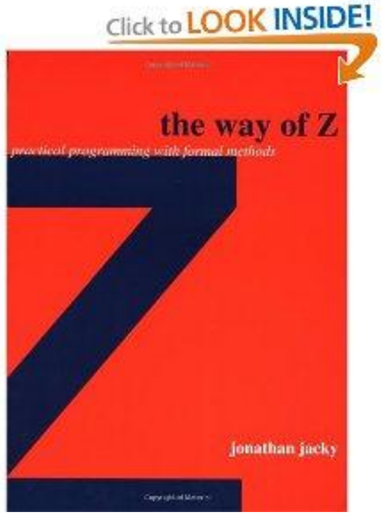
Specification

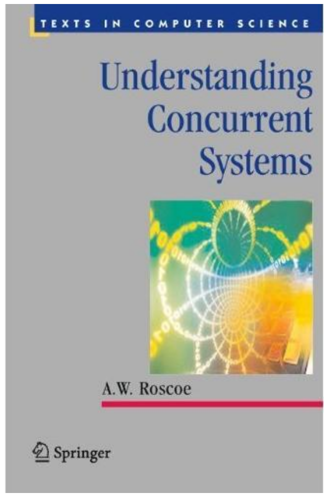
Implementation

- ProB
  - More efficiency in validating formal specifications
  - Optimizing existing tool chains
  - Supports compliance with safety standards




Specifying
Systems
The TLA+ Language and Tools
for Hardware and Software Engineers
Leslie Lamport





Validation tool for high-level formal models

ProB

Animation

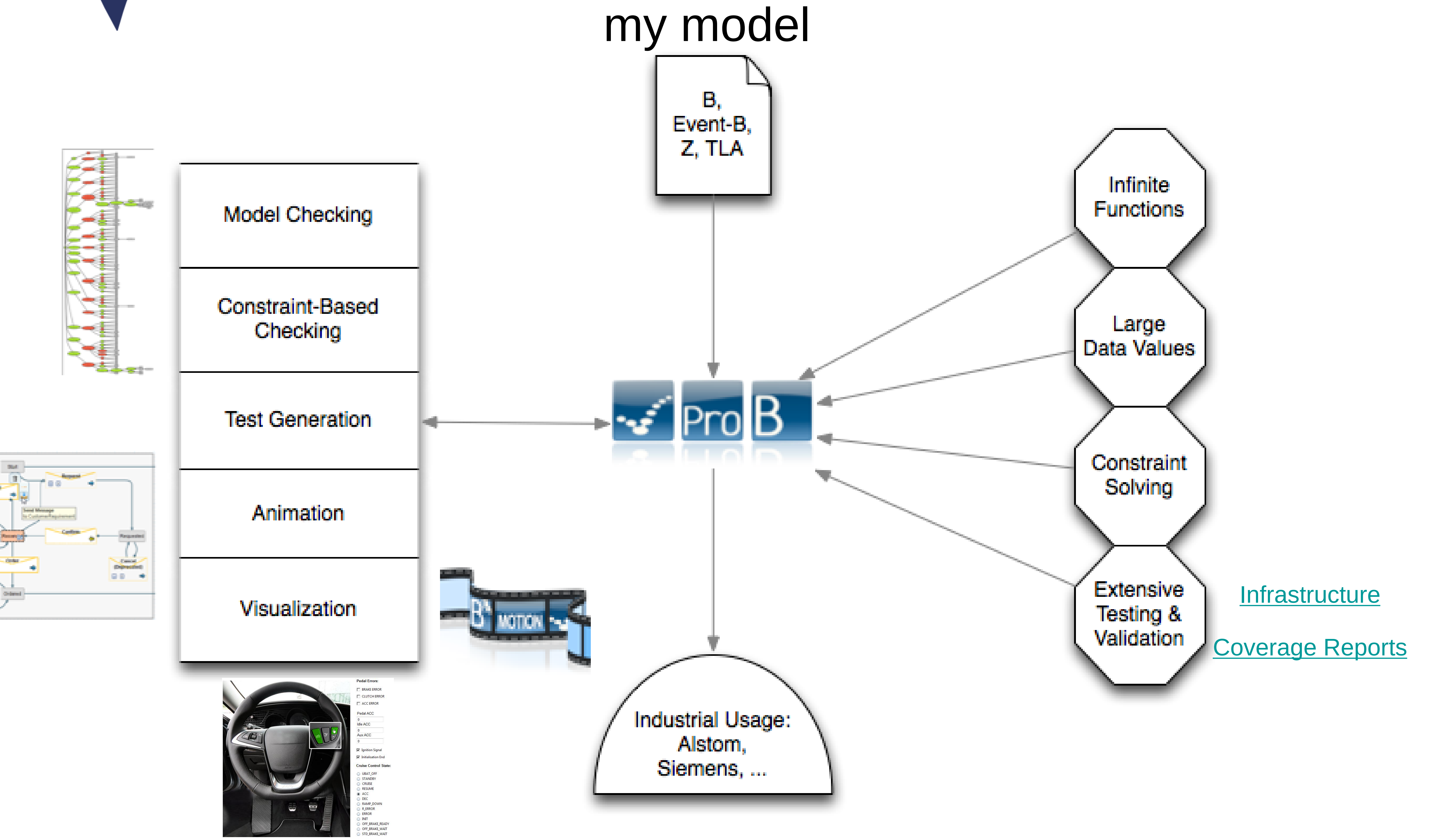
Model Checking

Constraint-Based Checking

Constraint Solver for
Predicate Logic, Arithmetic, Set theory,
Relations, Functions, Sequences, ...

SICStus 4

what can ProB do for me ?



my model

B, Event-B, Z, TLA

Model Checking

Constraint-Based Checking

Test Generation

Animation

Visualization

Infinite Functions

Large Data Values

Constraint Solving

Extensive Testing & Validation

Industrial Usage: Alstom, Siemens, ...

Infrastructure

Coverage Reports

Pedal Errors:

□ BRAKE ERROR

□ CLUTCH ERROR

□ ACC ERROR

Pedal ACC
0

Idle ACC
0

Aux ACC
0

☑ Ignition Signal

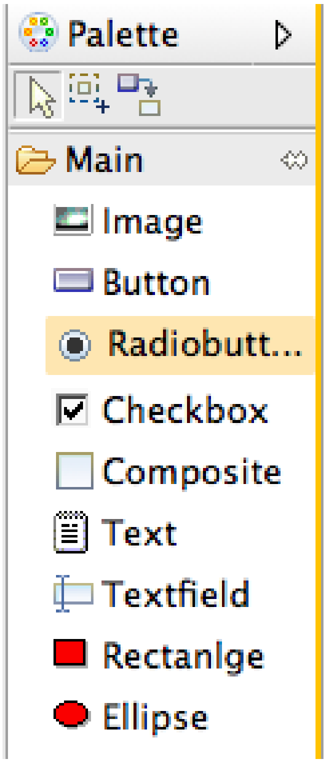
☑ Initialisation End

Cruise Control State:

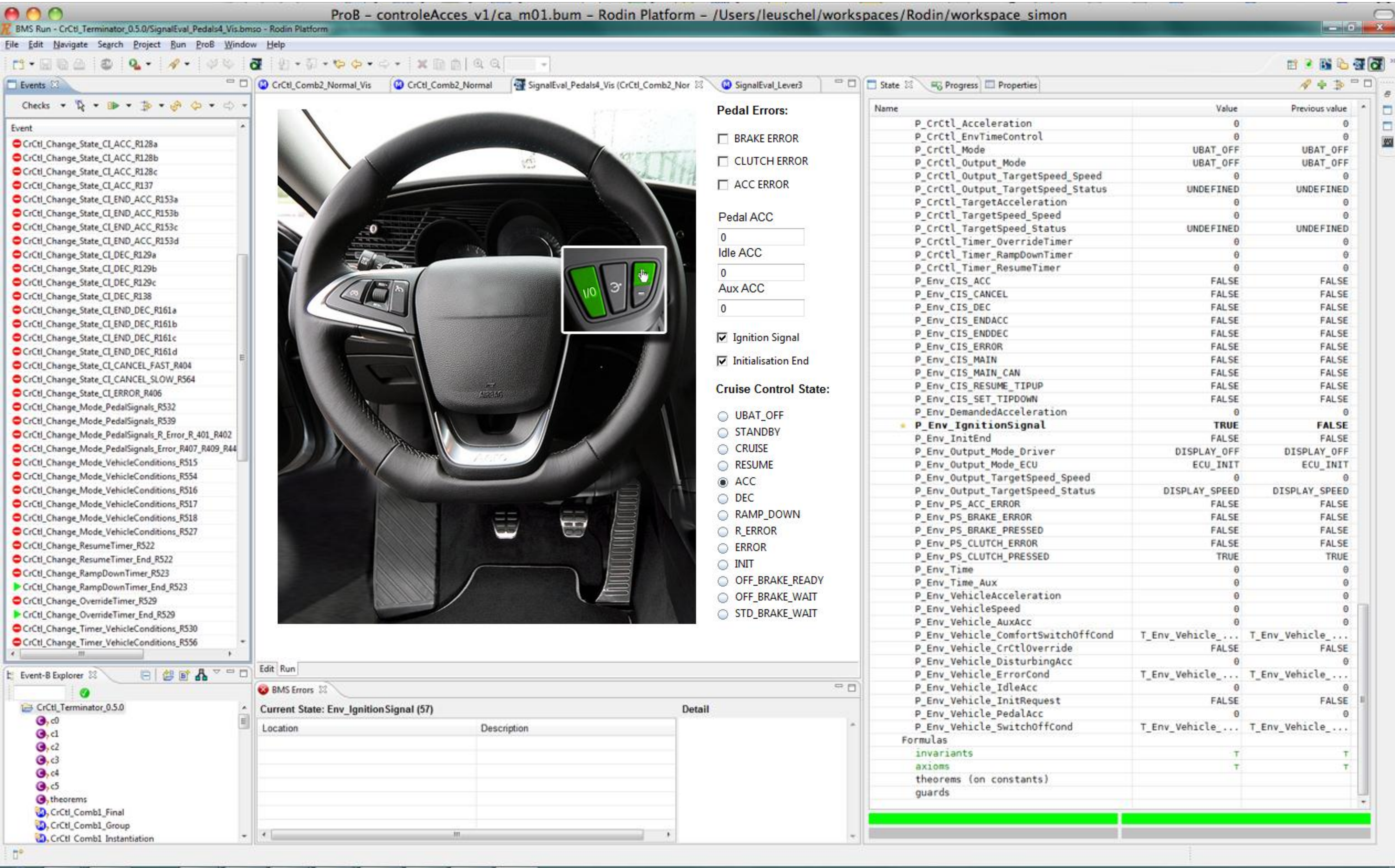
○ UBAT_OFF
○ STANDBY
○ CRUISE
○ RESUME
● ACC
○ DEC
○ RAMP_DOWN
○ R_ERROR
○ ERROR
○ INIT
○ OFF_BRAKE_READY
○ OFF_BRAKE_WAIT
○ STD_BRAKE_WAIT

BMotionStudio

- on top of ProB
- Editor:
  - link graphical elements with B expressions and predicates
- Important so that **domain experts** can detect errors in your models



B MotionStudio



ProB – controleAcces_v1/ca_m01.bum – Rodin Platform – /Users/leuschel/workspaces/Rodin/workspace_simon
BMS Run - CrCtl_Terminator_0.5.0/SignalEval_Pedals4_Vis.bmso - Rodin Platform
File
Edit
Navigate
Search
Project
Run
ProB
Window
Help
Events
Checks
CrCtl_Comb2_Normal_Vis
CrCtl_Comb2_Normal
SignalEval_Pedals4_Vis (CrCtl_Comb2_Nor
SignalEval_Lever3
State
Progress
Properties
Edit
Run
Event-B Explorer
BMS Errors
I/O

Event
- CrCtl_Change_State_CI_ACC_R128a
- CrCtl_Change_State_CI_ACC_R128b
- CrCtl_Change_State_CI_ACC_R128c
- CrCtl_Change_State_CI_ACC_R137
- CrCtl_Change_State_CI_END_ACC_R153a
- CrCtl_Change_State_CI_END_ACC_R153b
- CrCtl_Change_State_CI_END_ACC_R153c
- CrCtl_Change_State_CI_END_ACC_R153d
- CrCtl_Change_State_CI_DEC_R129a
- CrCtl_Change_State_CI_DEC_R129b
- CrCtl_Change_State_CI_DEC_R129c
- CrCtl_Change_State_CI_DEC_R138
- CrCtl_Change_State_CI_END_DEC_R161a
- CrCtl_Change_State_CI_END_DEC_R161b
- CrCtl_Change_State_CI_END_DEC_R161c
- CrCtl_Change_State_CI_END_DEC_R161d
- CrCtl_Change_State_CI_CANCEL_FAST_R404
- CrCtl_Change_State_CI_CANCEL_SLOW_R564
- CrCtl_Change_State_CI_ERROR_R406
- CrCtl_Change_Mode_PedalSignals_R532
- CrCtl_Change_Mode_PedalSignals_R539
- CrCtl_Change_Mode_PedalSignals_R_Error_R_401_R402
- CrCtl_Change_Mode_PedalSignals_Error_R407_R409_R44
- CrCtl_Change_Mode_VehicleConditions_R515
- CrCtl_Change_Mode_VehicleConditions_R554
- CrCtl_Change_Mode_VehicleConditions_R516
- CrCtl_Change_Mode_VehicleConditions_R517
- CrCtl_Change_Mode_VehicleConditions_R518
- CrCtl_Change_Mode_VehicleConditions_R527
- CrCtl_Change_ResumeTimer_R522
- CrCtl_Change_ResumeTimer_End_R522
- CrCtl_Change_RampDownTimer_R523
- ▶ CrCtl_Change_RampDownTimer_End_R523
- CrCtl_Change_OverrideTimer_R529
- ▶ CrCtl_Change_OverrideTimer_End_R529
- CrCtl_Change_Timer_VehicleConditions_R530
- CrCtl_Change_Timer_VehicleConditions_R556

CrCtl_Terminator_0.5.0
- c0
- c1
- c2
- c3
- c4
- c5
- theorems
- CrCtl_Comb1_Final
- CrCtl_Comb1_Group
- CrCtl_Comb1_Instantiation

Pedal Errors:

□ BRAKE ERROR

□ CLUTCH ERROR

□ ACC ERROR

Pedal ACC: 0

Idle ACC: 0

Aux ACC: 0

☑ Ignition Signal

☑ Initialisation End

Cruise Control State:

○ UBAT_OFF
○ STANDBY
○ CRUISE
○ RESUME
● ACC
○ DEC
○ RAMP_DOWN
○ R_ERROR
○ ERROR
○ INIT
○ OFF_BRAKE_READY
○ OFF_BRAKE_WAIT
○ STD_BRAKE_WAIT

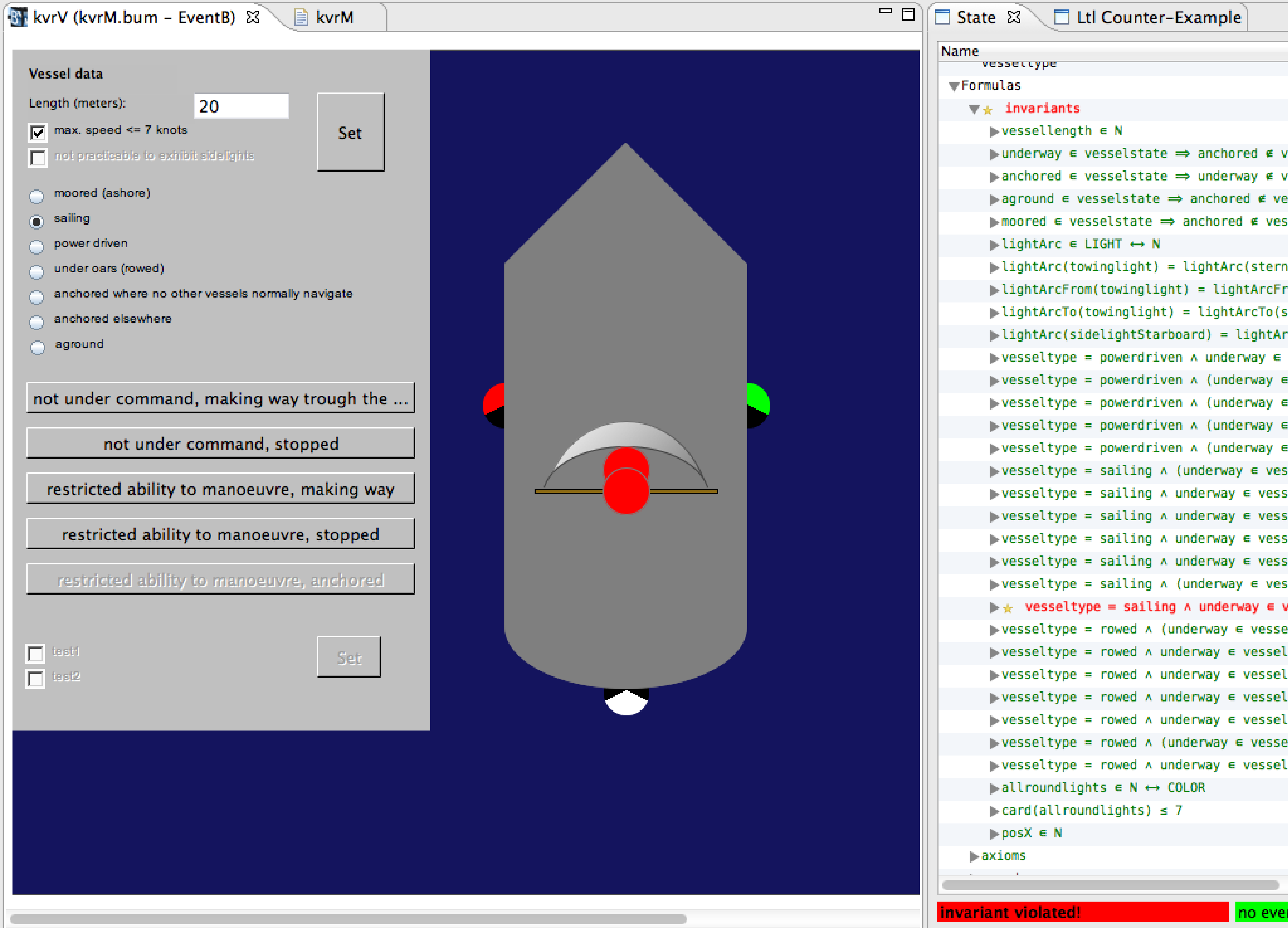
Current State: Env_IgnitionSignal (57)

Detail

Location	Description

Name	Value	Previous value
P_CrCtl_Acceleration	0	0
P_CrCtl_EnvTimeControl	0	0
P_CrCtl_Mode	UBAT_OFF	UBAT_OFF
P_CrCtl_Output_Mode	UBAT_OFF	UBAT_OFF
P_CrCtl_Output_TargetSpeed_Speed	0	0
P_CrCtl_Output_TargetSpeed_Status	UNDEFINED	UNDEFINED
P_CrCtl_TargetAcceleration	0	0
P_CrCtl_TargetSpeed_Speed	0	0
P_CrCtl_TargetSpeed_Status	UNDEFINED	UNDEFINED
P_CrCtl_Timer_OverrideTimer	0	0
P_CrCtl_Timer_RampDownTimer	0	0
P_CrCtl_Timer_ResumeTimer	0	0
P_Env_CIS_ACC	FALSE	FALSE
P_Env_CIS_CANCEL	FALSE	FALSE
P_Env_CIS_DEC	FALSE	FALSE
P_Env_CIS_ENDACC	FALSE	FALSE
P_Env_CIS_ENDDEC	FALSE	FALSE
P_Env_CIS_ERROR	FALSE	FALSE
P_Env_CIS_MAIN	FALSE	FALSE
P_Env_CIS_MAIN_CAN	FALSE	FALSE
P_Env_CIS_RESUME_TIPUP	FALSE	FALSE
P_Env_CIS_SET_TIPDOWN	FALSE	FALSE
P_Env_DemandedAcceleration	0	0
P_Env_IgnitionSignal	**TRUE**	**FALSE**
P_Env_InitEnd	FALSE	FALSE
P_Env_Output_Mode_Driver	DISPLAY_OFF	DISPLAY_OFF
P_Env_Output_Mode_ECU	ECU_INIT	ECU_INIT
P_Env_Output_TargetSpeed_Speed	0	0
P_Env_Output_TargetSpeed_Status	DISPLAY_SPEED	DISPLAY_SPEED
P_Env_PS_ACC_ERROR	FALSE	FALSE
P_Env_PS_BRAKE_ERROR	FALSE	FALSE
P_Env_PS_BRAKE_PRESSED	FALSE	FALSE
P_Env_PS_CLUTCH_ERROR	FALSE	FALSE
P_Env_PS_CLUTCH_PRESSED	TRUE	TRUE
P_Env_Time	0	0
P_Env_Time_Aux	0	0
P_Env_VehicleAcceleration	0	0
P_Env_VehicleSpeed	0	0
P_Env_Vehicle_AuxAcc	0	0
P_Env_Vehicle_ComfortSwitchOffCond	T_Env_Vehicle_...	T_Env_Vehicle_...
P_Env_Vehicle_CrCtlOverride	FALSE	FALSE
P_Env_Vehicle_DisturbingAcc	0	0
P_Env_Vehicle_ErrorCond	T_Env_Vehicle_...	T_Env_Vehicle_...
P_Env_Vehicle_IdleAcc	0	0
P_Env_Vehicle_InitRequest	FALSE	FALSE
P_Env_Vehicle_PedalAcc	0	0
P_Env_Vehicle_SwitchOffCond	T_Env_Vehicle_...	T_Env_Vehicle_...
Formulas		
invariants	T	T
axioms	T	T
theorems (on constants)		
guards		

http://en.wikisource.org/wiki/International_Regulations_for_Preventing_Collisions_at_Sea



Vessel data

Length (meters): 20

☑ max. speed <= 7 knots

□ not practicable to exhibit sidelights

Set

○ moored (ashore)

● sailing

○ power driven

○ under oars (rowed)

○ anchored where no other vessels normally navigate

○ anchored elsewhere

○ aground

not under command, making way trough the ...

not under command, stopped

restricted ability to manoeuvre, making way

restricted ability to manoeuvre, stopped

restricted ability to manoeuvre, anchored

□ test1

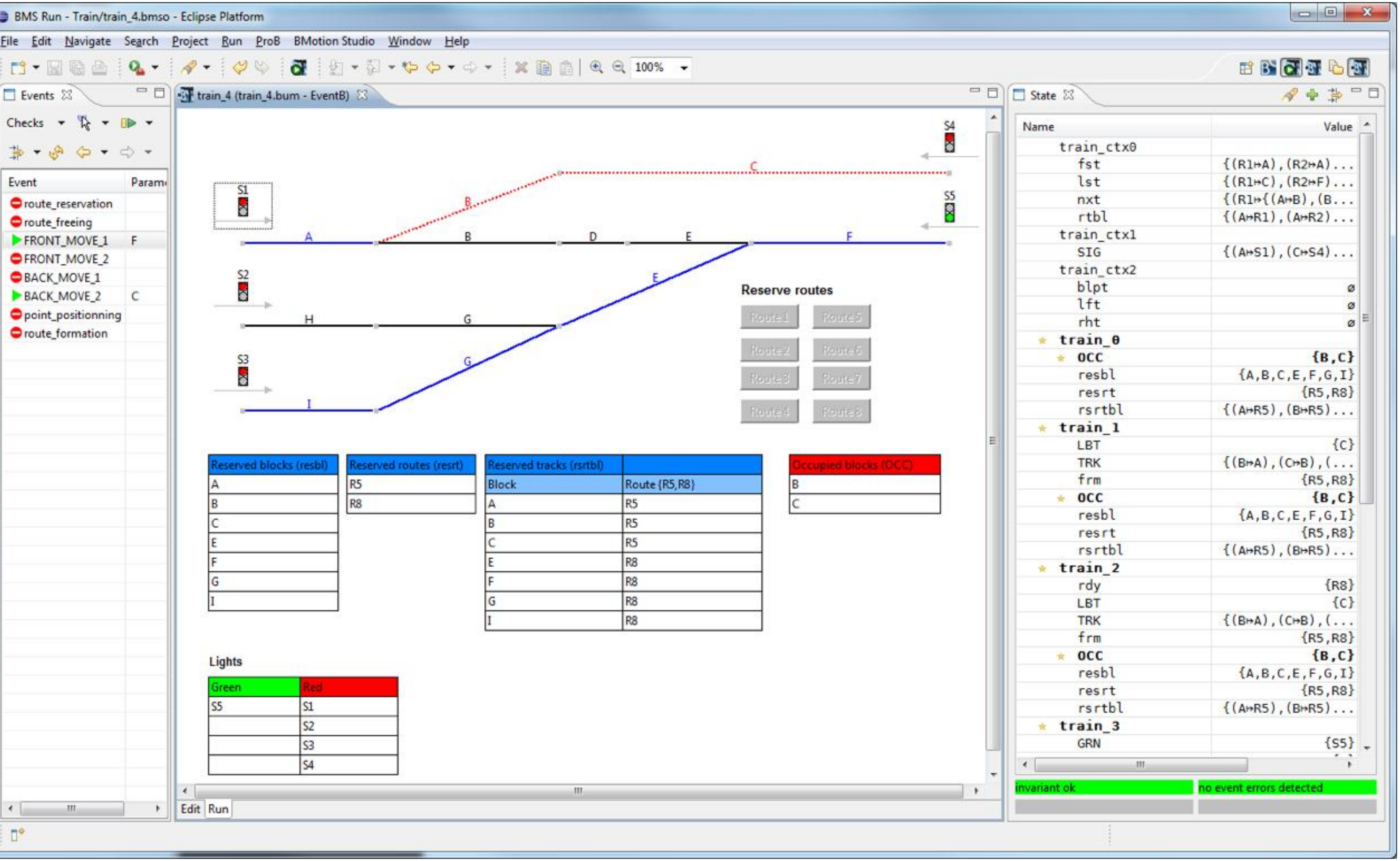
□ test2

Set

```
 vesseltype
▼Formulas
  ▼★ invariants
    ▶vessellength ∈ ℕ
    ▶underway ∈ vesselstate ⇒ anchored ∉ v
    ▶anchored ∈ vesselstate ⇒ underway ∉ v
    ▶aground ∈ vesselstate ⇒ anchored ∉ ve
    ▶moored ∈ vesselstate ⇒ anchored ∉ ves
    ▶lightArc ∈ LIGHT ↔ ℕ
    ▶lightArc(towinglight) = lightArc(stern
    ▶lightArcFrom(towinglight) = lightArcFr
    ▶lightArcTo(towinglight) = lightArcTo(s
    ▶lightArc(sidelightStarboard) = lightAr
    ▶vesseltype = powerdriven ∧ underway ∈
    ▶vesseltype = powerdriven ∧ (underway ∈
    ▶vesseltype = powerdriven ∧ (underway ∈
    ▶vesseltype = powerdriven ∧ (underway ∈
    ▶vesseltype = powerdriven ∧ (underway ∈
    ▶vesseltype = sailing ∧ (underway ∈ ves
    ▶vesseltype = sailing ∧ underway ∈ vess
    ▶vesseltype = sailing ∧ underway ∈ vess
    ▶vesseltype = sailing ∧ underway ∈ vess
    ▶vesseltype = sailing ∧ underway ∈ vess
    ▶vesseltype = sailing ∧ (underway ∈ ves
    ▶★ vesseltype = sailing ∧ underway ∈ v
    ▶vesseltype = rowed ∧ (underway ∈ vesse
    ▶vesseltype = rowed ∧ underway ∈ vessel
    ▶vesseltype = rowed ∧ underway ∈ vessel
    ▶vesseltype = rowed ∧ underway ∈ vessel
    ▶vesseltype = rowed ∧ underway ∈ vessel
    ▶vesseltype = rowed ∧ (underway ∈ vesse
    ▶vesseltype = rowed ∧ underway ∈ vessel
    ▶allroundlights ∈ ℕ ↔ COLOR
    ▶card(allroundlights) ≤ 7
    ▶posX ∈ ℕ
  ▶axioms
```

invariant violated!

no ever


BMS Run - Train/train_4.bmso - Eclipse Platform
File Edit Navigate Search Project Run ProB BMotion Studio Window Help
100%
Events
Checks
train_4 (train_4.bum - EventB)
State
S1
S2
S3
S4
S5
A
B
C
D
E
F
G
H
I
Reserve routes
Route1
Route5
Route2
Route6
Route3
Route7
Route4
Route8
Edit
Run

Event	Param
route_reservation	
route_freeing	
FRONT_MOVE_1	F
FRONT_MOVE_2	
BACK_MOVE_1	
BACK_MOVE_2	C
point_positionning	
route_formation	

Reserved blocks (resbl)
A
B
C
E
F
G
I

Reserved routes (resrt)
R5
R8

Reserved tracks (rsrtbl)	
Block	Route {R5,R8}
A	R5
B	R5
C	R5
E	R8
F	R8
G	R8
I	R8

Occupied blocks (OCC)
B
C

Lights

Green	Red
S5	S1
	S2
	S3
	S4

Name	Value
train_ctx0	
fst	{(R1↦A),(R2↦A)...
lst	{(R1↦C),(R2↦F)...
nxt	{(R1↦{(A↦B),(B...
rtbl	{(A↦R1),(A↦R2)...
train_ctx1	
SIG	{(A↦S1),(C↦S4)...
train_ctx2	
blpt	∅
lft	∅
rht	∅
★ **train_0**	
★ **OCC**	**{B,C}**
resbl	{A,B,C,E,F,G,I}
resrt	{R5,R8}
rsrtbl	{(A↦R5),(B↦R5)...
★ **train_1**	
LBT	{C}
TRK	{(B↦A),(C↦B),(...
frm	{R5,R8}
★ **OCC**	**{B,C}**
resbl	{A,B,C,E,F,G,I}
resrt	{R5,R8}
rsrtbl	{(A↦R5),(B↦R5)...
★ **train_2**	
rdy	{R8}
LBT	{C}
TRK	{(B↦A),(C↦B),(...
frm	{R5,R8}
★ **OCC**	**{B,C}**
resbl	{A,B,C,E,F,G,I}
resrt	{R5,R8}
rsrtbl	{(A↦R5),(B↦R5)...
★ **train_3**	
GRN	{S5}

invariant ok

no event errors detected

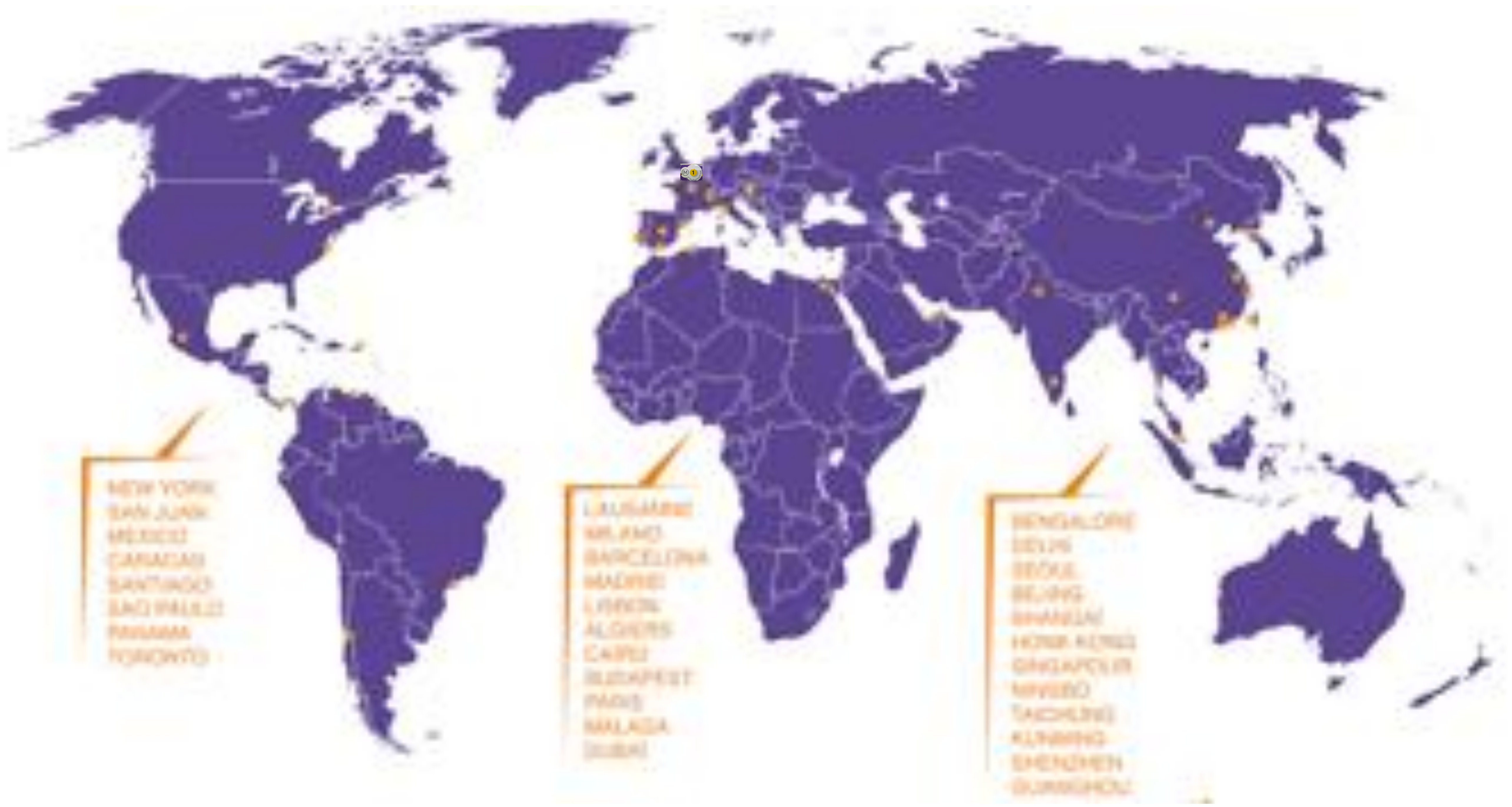
2. Data Validation

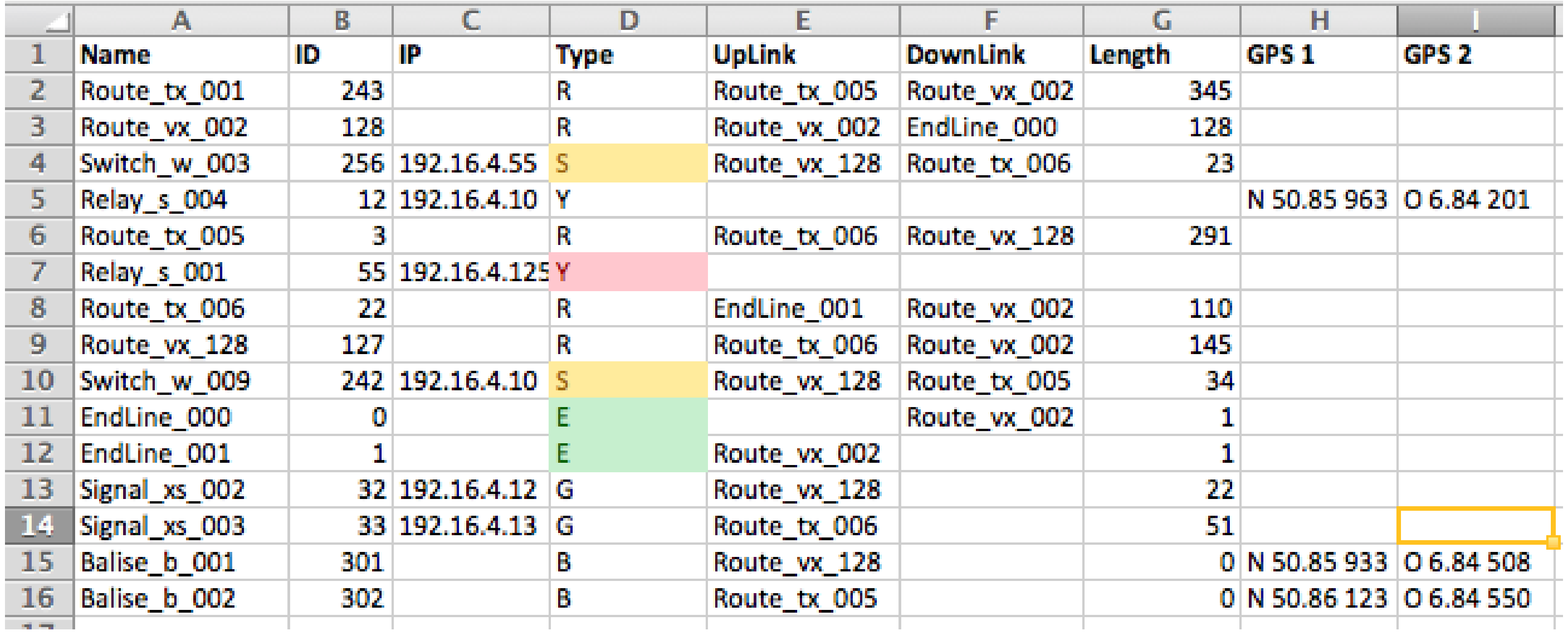
M 1 s'automatise

M
1
s'automatise

ennes
1er train
2e train
La Défense
M
1
s'automatise

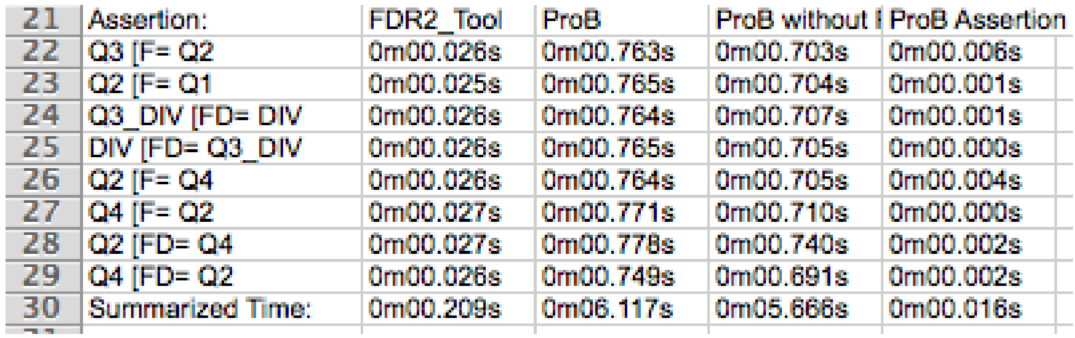
Worldwide implementations (2012) of systems embedding software generated from B models.



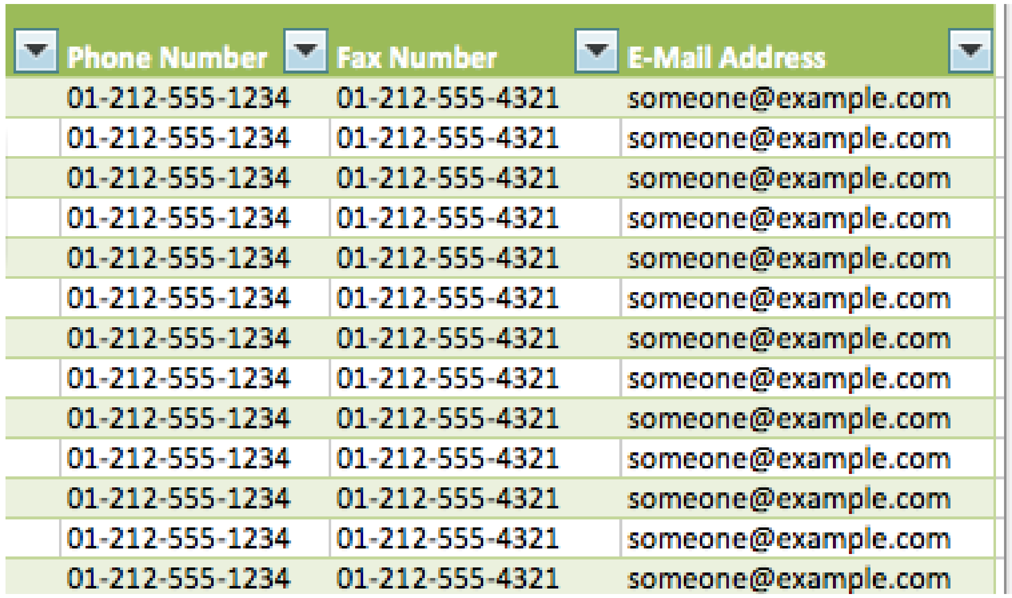


	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

Your Data

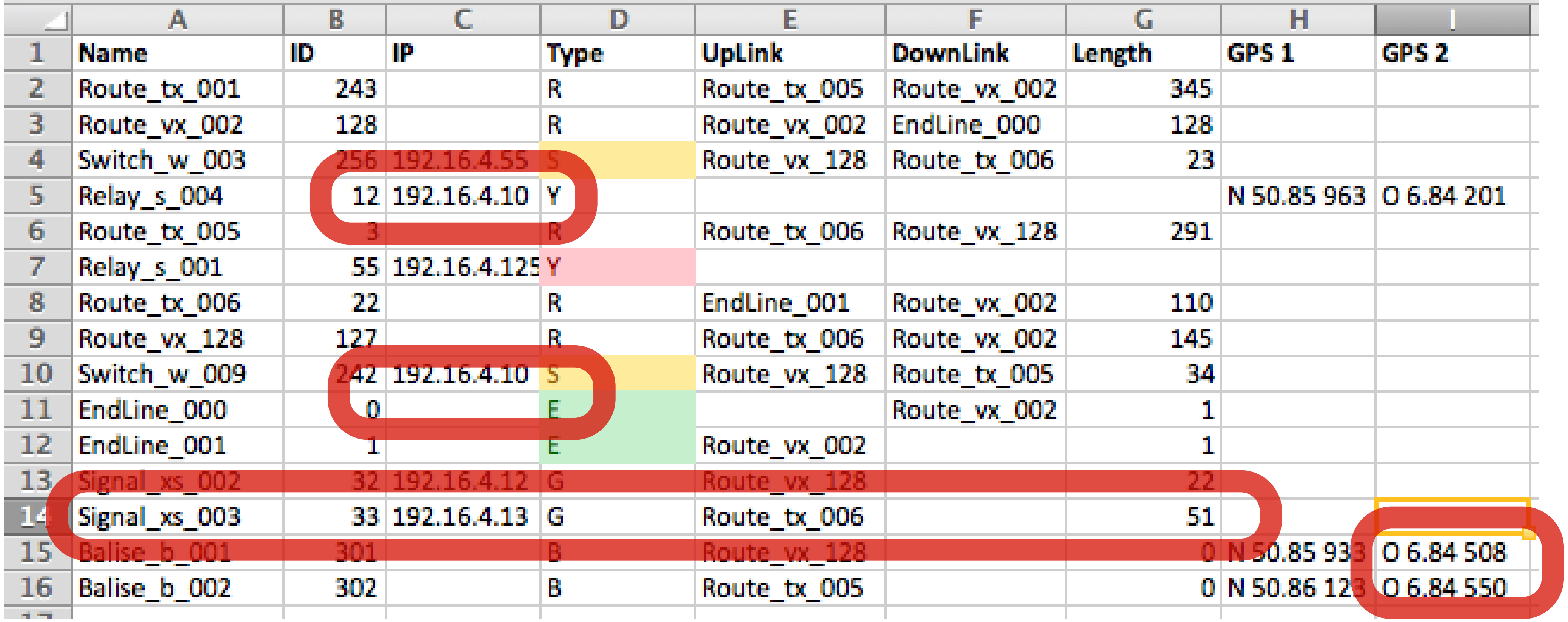


21	Assertion:	FDR2_Tool	ProB	ProB without I	ProB Assertion
22	Q3 [F= Q2	0m00.026s	0m00.763s	0m00.703s	0m00.006s
23	Q2 [F= Q1	0m00.025s	0m00.765s	0m00.704s	0m00.001s
24	Q3_DIV [FD= DIV	0m00.026s	0m00.764s	0m00.707s	0m00.001s
25	DIV [FD= Q3_DIV	0m00.026s	0m00.765s	0m00.705s	0m00.000s
26	Q2 [F= Q4	0m00.026s	0m00.764s	0m00.705s	0m00.004s
27	Q4 [F= Q2	0m00.027s	0m00.771s	0m00.710s	0m00.000s
28	Q2 [FD= Q4	0m00.027s	0m00.778s	0m00.740s	0m00.002s
29	Q4 [FD= Q2	0m00.026s	0m00.749s	0m00.691s	0m00.002s
30	Summarized Time:	0m00.209s	0m06.117s	0m05.666s	0m00.016s



Phone Number	Fax Number	E-Mail Address
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com
01-212-555-1234	01-212-555-4321	someone@example.com

Is it consistent ?

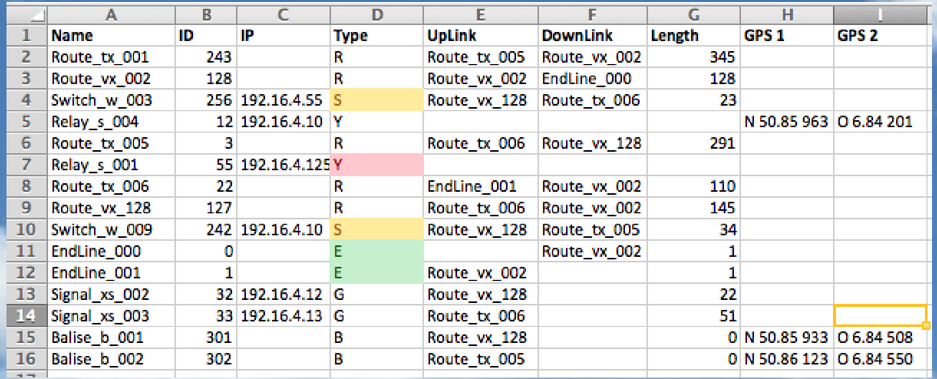


	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

Is it safe ?

Is it correct ?

Dream



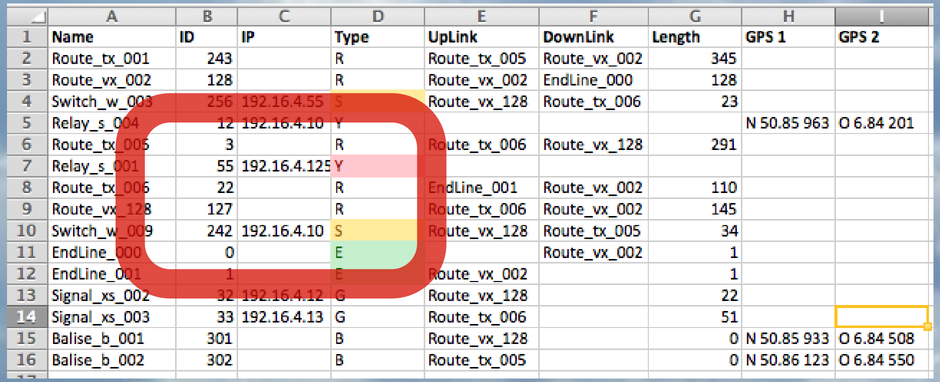
	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

Formal Properties

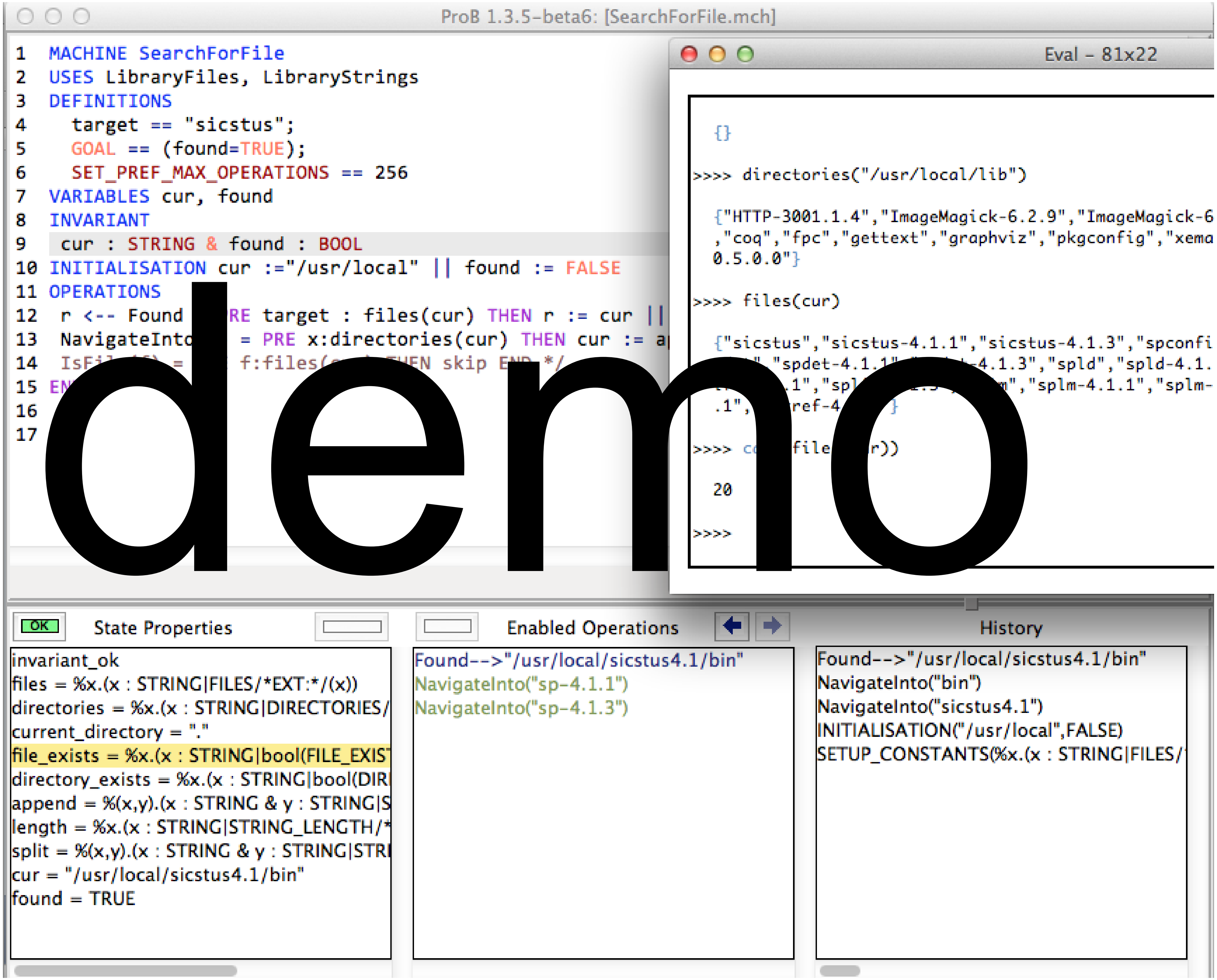
high-level language (B,...)

ProB

Error location & diagnosis



	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550



```
MACHINE SearchForFile
USES LibraryFiles, LibraryStrings
DEFINITIONS
  target == "sicstus";
  GOAL == (found=TRUE);
  SET_PREF_MAX_OPERATIONS == 256
VARIABLES cur, found
INVARIANT
 cur : STRING & found : BOOL
INITIALISATION cur :="/usr/local" || found := FALSE
OPERATIONS
 r <-- Found    RE target : files(cur) THEN r := cur ||
 NavigateInto    = PRE x:directories(cur) THEN cur := a
 IsFi      =     f:files(    THEN skip END */
EN
```

```
  {}

>>>> directories("/usr/local/lib")

  {"HTTP-3001.1.4","ImageMagick-6.2.9","ImageMagick-6
  ,"coq","fpc","gettext","graphviz","pkgconfig","xema
  0.5.0.0"}

>>>> files(cur)

  {"sicstus","sicstus-4.1.1","sicstus-4.1.3","spconfi
       ,"spdet-4.1.1"      -4.1.3","spld","spld-4.1.
       .1","spl               m","splm-4.1.1","splm-
  .1",      ref-4            }

>>>> c     file    r))

  20

>>>>
```

State Properties:

```
invariant_ok
files = %x.(x : STRING|FILES/*EXT:*/(x))
directories = %x.(x : STRING|DIRECTORIES/
current_directory = "."
file_exists = %x.(x : STRING|bool(FILE_EXIS
directory_exists = %x.(x : STRING|bool(DIR
append = %(x,y).(x : STRING & y : STRING|S
length = %x.(x : STRING|STRING_LENGTH/*
split = %(x,y).(x : STRING & y : STRING|STR
cur = "/usr/local/sicstus4.1/bin"
found = TRUE
```

Enabled Operations:

```
Found-->"/usr/local/sicstus4.1/bin"
NavigateInto("sp-4.1.1")
NavigateInto("sp-4.1.3")
```

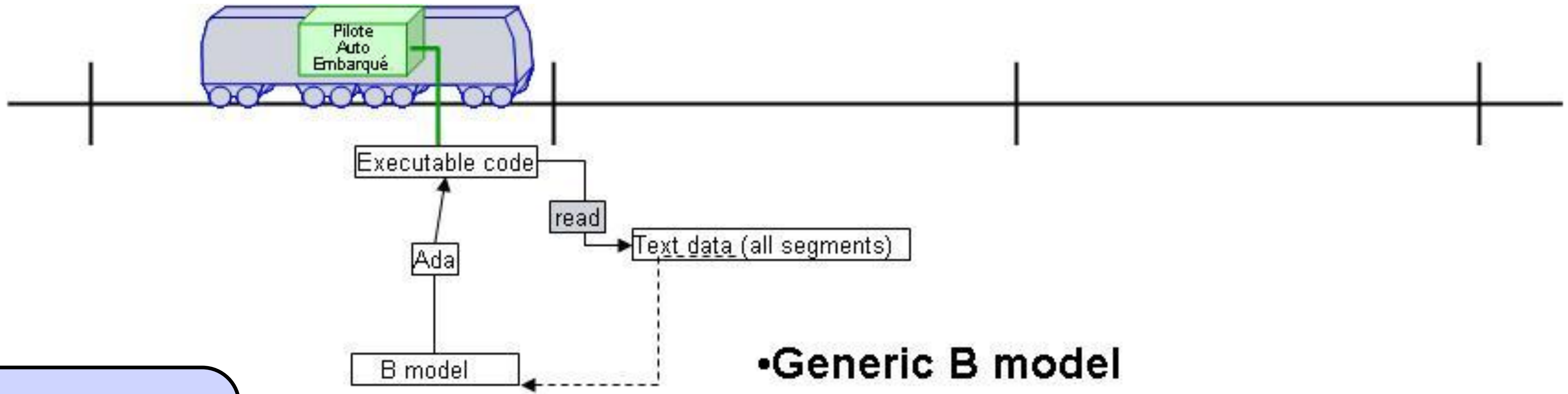
History:

```
Found-->"/usr/local/sicstus4.1/bin"
NavigateInto("bin")
NavigateInto("sicstus4.1")
INITIALISATION("/usr/local",FALSE)
SETUP_CONSTANTS(%x.(x : STRING|FILES/
```

demo

how it all began

- Deploy Project: Scalability issue at Siemens for data validation



•Generic B model

•Text Data with the entire line (all segments)

Problems :

•Make sure the assumptions in the B model are correct for all segments

Considerable work

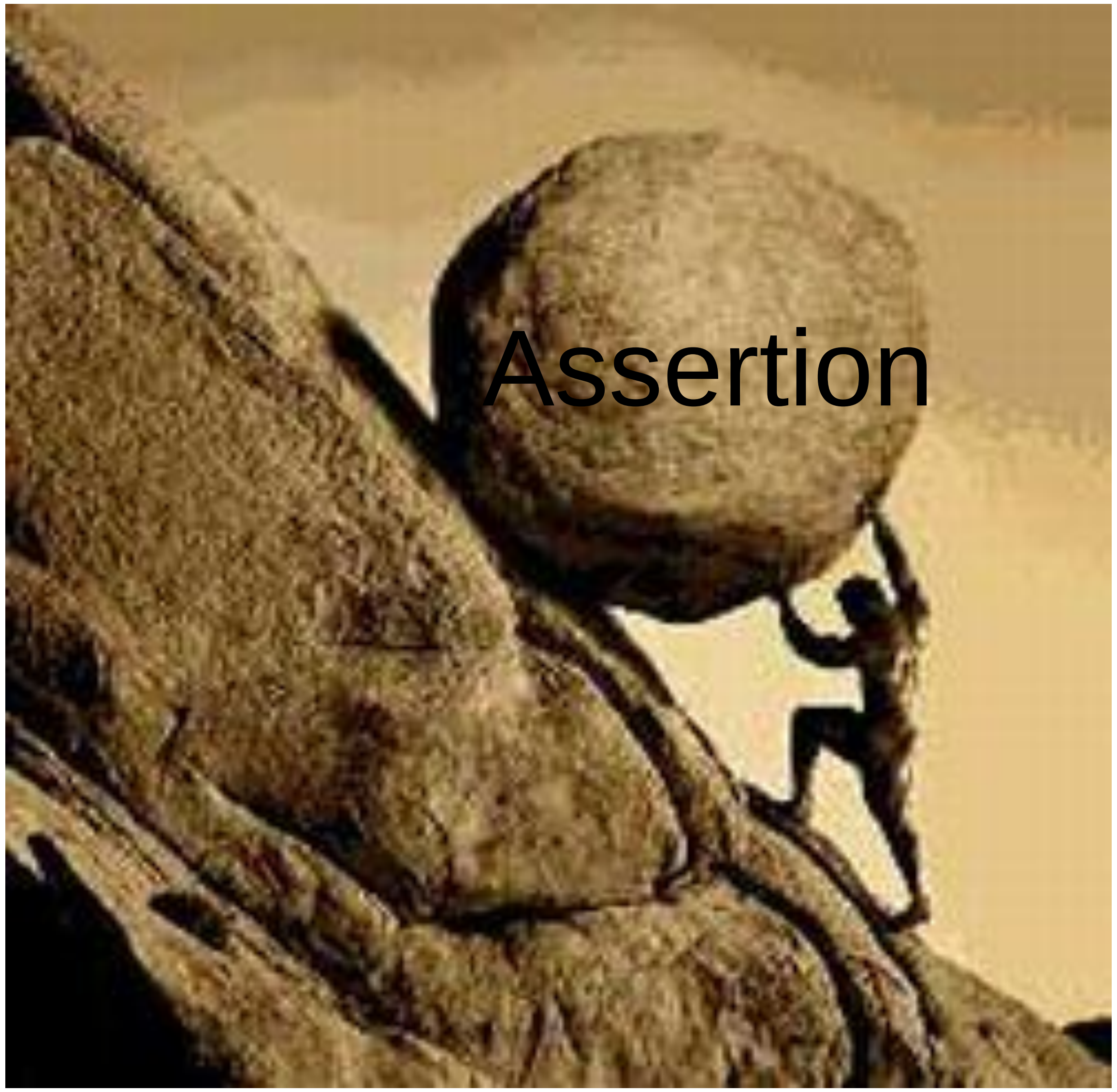
147 Assertions

147 Assertions

t_iti_partiel_acs & bb : cfg_cdv_aig & aa |-> bb : t_iti_partiel_acs
<| cfg_ipart_cdv_transit_dernier_i |> cfg_cdv_aig => bb :
cfg_ipart_cdv_transit_liste_i[(cfg_ipart_cdv_transit_deb(aa) ..
cfg_ipart_cdv_transit_fin(aa))]) cfg_ipart_pc1_adj_i~[{TRUE}] <:
t_iti_partiel_acs cfg_ipart_pc2_adj_i~[{TRUE}] <:
t_iti_partiel_acs cfg_ipart_pc1_adj_i~[{TRUE}] /\
cfg_ipart_pc2_adj_i~[{TRUE}] = {}
cfg_ipart_aig_tild_liste_i~[t_iti_partiel_acs] <: t_liste_acs
cfg_ipart_aig_tild_liste_i~[t_iti_partiel_acs] <: NATURAL
cfg_ipart_aig_liste_i~[t_aig_acs] <: t_liste_acs
cfg_ipart_aig_liste_i~[t_aig_acs] <: NATURAL
cfg_ipart_cdv_transit_liste_i~[cfg_cdv_aig] <: t_liste_acs
cfg_ipart_cdv_transit_liste_i~[cfg_cdv_aig] <: NATURAL
cfg_ipart_cdv_zdest_sscant_liste_i~[cfg_cdv_block] <:
t_liste_acs cfg_ipart_cdv_zdest_sscant_liste_i~[cfg_cdv_block]

147 Assertions

situation before deploy



san juan: 80 assertions had to be checked manually

current situation

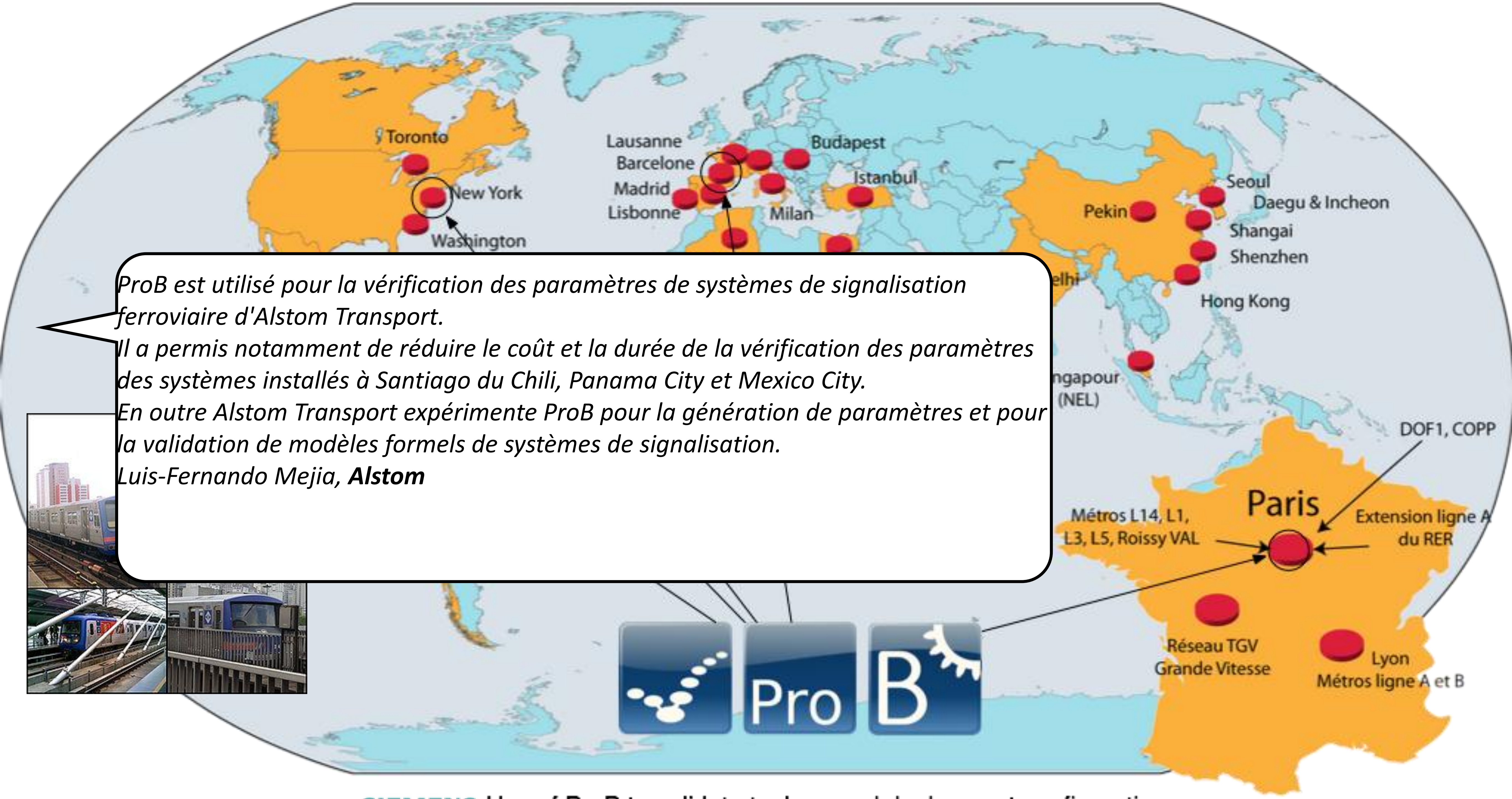
- validation can be done in minutes
- more reliable, better feedback
- much bigger and new problems can be tackled (on-board data)

The work done with ProB is a great success. Thanks to the automatization and ProB, the wayside data validation is quicker, easier and complete

Use of the B Method developped by

CLEARSY
SYTEM ENGINEERING

Metros and Trains equipped with B SIL4 software

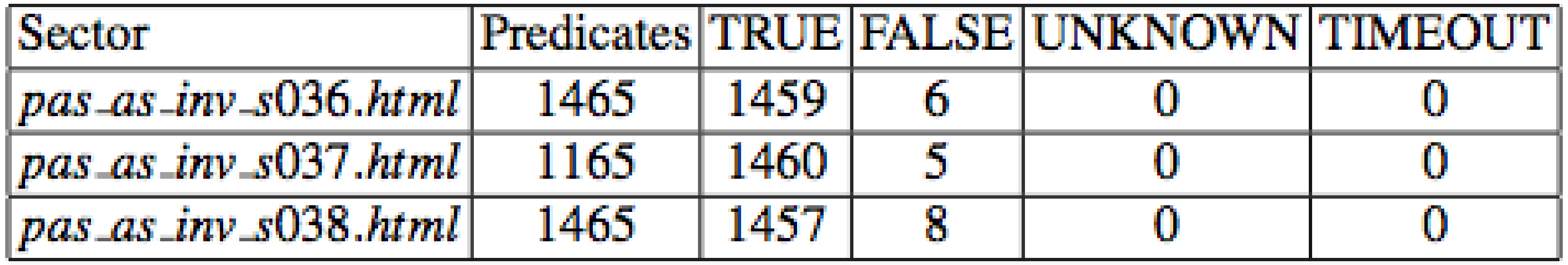


ProB est utilisé pour la vérification des paramètres de systèmes de signalisation ferroviaire d'Alstom Transport.
Il a permis notamment de réduire le coût et la durée de la vérification des paramètres des systèmes installés à Santiago du Chili, Panama City et Mexico City.
En outre Alstom Transport expérimente ProB pour la génération de paramètres et pour la validation de modèles formels de systèmes de signalisation.
Luis-Fernando Mejia, ***Alstom***

SIEMENS Use of ProB to validate toplogy and deployment configuration

São Paulo line 4

- 210 files, >30,000 lines of B
- >2500 assertions, >32,000 properties to be checked
- some very large sets (e.g., 1..65535), some infinite sets (e.g., INTEGER - {x})
- inconsistencies found !



Sector	Predicates	TRUE	FALSE	UNKNOWN	TIMEOUT
pas_as_inv_s036.html	1465	1459	6	0	0
pas_as_inv_s037.html	1165	1460	5	0	0
pas_as_inv_s038.html	1465	1457	8	0	0

cf. Chapter

Table 1.3 Paris line 1 (ZC)

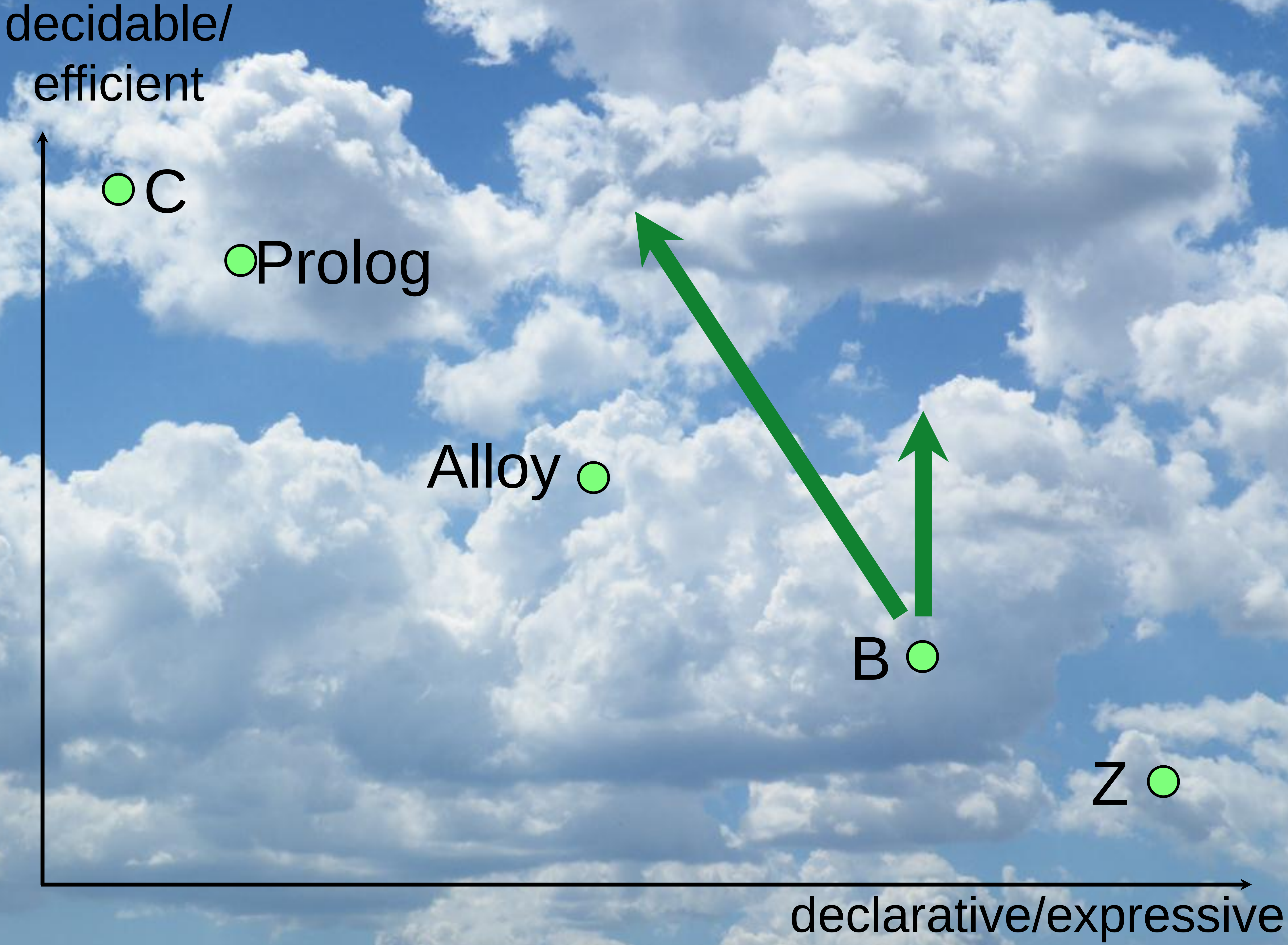
Sector	Predicates	TRUE	FALSE	UNKNOWN	TIMEOUT
pas_as_inv_s011.html	1503	1501	2	0	0
pas_as_inv_s012.html	1503	1498	5	0	0
pas_as_inv_s013.html	1503	1496	7	0	0
pas_as_inv_s014.html	1503	1499	4	0	0
pas_as_inv_s015.html	1503	1498	5	0	0
pas_as_inv_s016.html	1503	1498	5	0	0

Paris line 1 (PAL) . PAL (Pilote Automatique Ligne) is a controller line who realizes the Automatic Train Supervision (ATS) function of CBTC systems. The B models of PAL consisted of 74 files with over 10,000 lines of B. In all 2024 assertions about concrete data of the PAL needed to be checked. ProB found 12 in under 5 minutes. These problems have been examined and confirmed by manual inspection aterward at Siemens.

Why B ?

Why ProB ?

Challenges



decidable/
efficient

C

Prolog

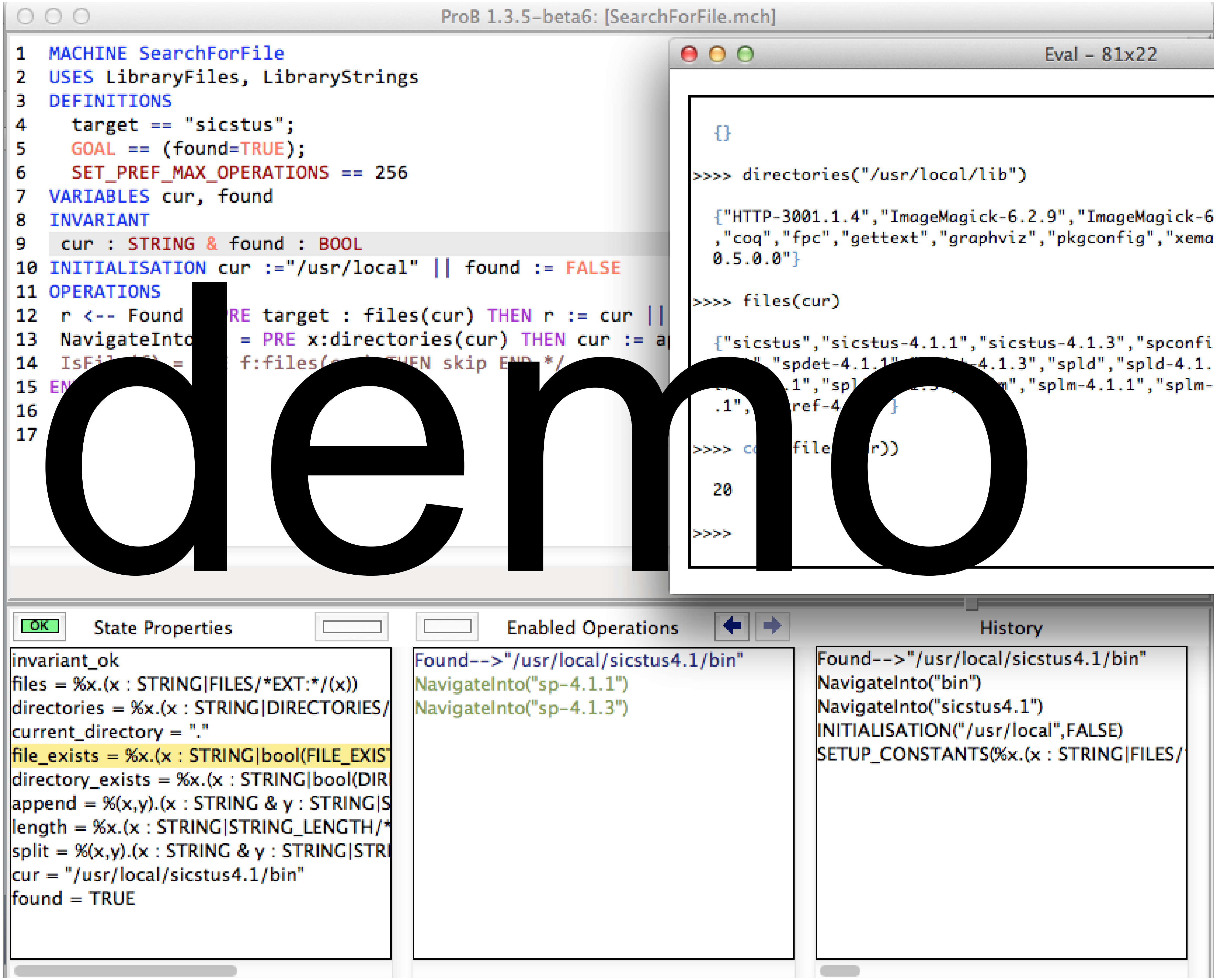
Alloy

B

Z

declarative/expressive

demo



```
MACHINE SearchForFile
USES LibraryFiles, LibraryStrings
DEFINITIONS
  target == "sicstus";
  GOAL == (found=TRUE);
  SET_PREF_MAX_OPERATIONS == 256
VARIABLES cur, found
INVARIANT
 cur : STRING & found : BOOL
INITIALISATION cur :="/usr/local" || found := FALSE
OPERATIONS
 r <-- Found    RE target : files(cur) THEN r := cur ||
 NavigateInto     = PRE x:directories(cur) THEN cur := a
 IsFi        =     f:files(        HEN skip E      */
EN
```

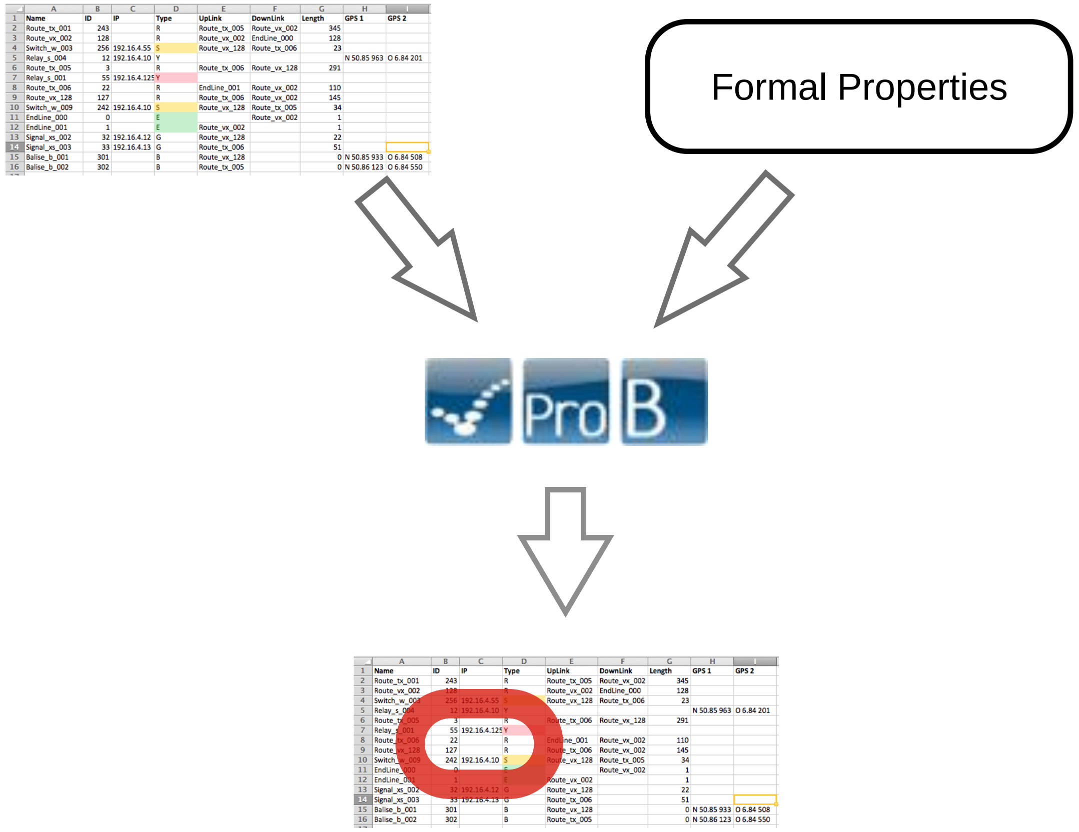
```
 {}

>>>> directories("/usr/local/lib")

 {"HTTP-3001.1.4","ImageMagick-6.2.9","ImageMagick-6
 ,"coq","fpc","gettext","graphviz","pkgconfig","xema
 0.5.0.0"}

>>>> files(cur)

 {"sicstus","sicstus-4.1.1","sicstus-4.1.3","spconfi
         ,"spdet-4.1.1"      -4.1.3","spld","spld-4.1.
         .1","spl              m","splm-4.1.1","splm-
 .1",     ref-4          }

>>>> c     file     r))

 20

>>>>
```

```
invariant_ok
files = %x.(x : STRING|FILES/*EXT:*/(x))
directories = %x.(x : STRING|DIRECTORIES/
current_directory = "."
file_exists = %x.(x : STRING|bool(FILE_EXIS
directory_exists = %x.(x : STRING|bool(DIR
append = %(x,y).(x : STRING & y : STRING|S
length = %x.(x : STRING|STRING_LENGTH/*
split = %(x,y).(x : STRING & y : STRING|STR
cur = "/usr/local/sicstus4.1/bin"
found = TRUE
```

```
Found-->"/usr/local/sicstus4.1/bin"
NavigateInto("sp-4.1.1")
NavigateInto("sp-4.1.3")
```

```
Found-->"/usr/local/sicstus4.1/bin"
NavigateInto("bin")
NavigateInto("sicstus4.1")
INITIALISATION("/usr/local",FALSE)
SETUP_CONSTANTS(%x.(x : STRING|FILES/
```

Why B & ProB



	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

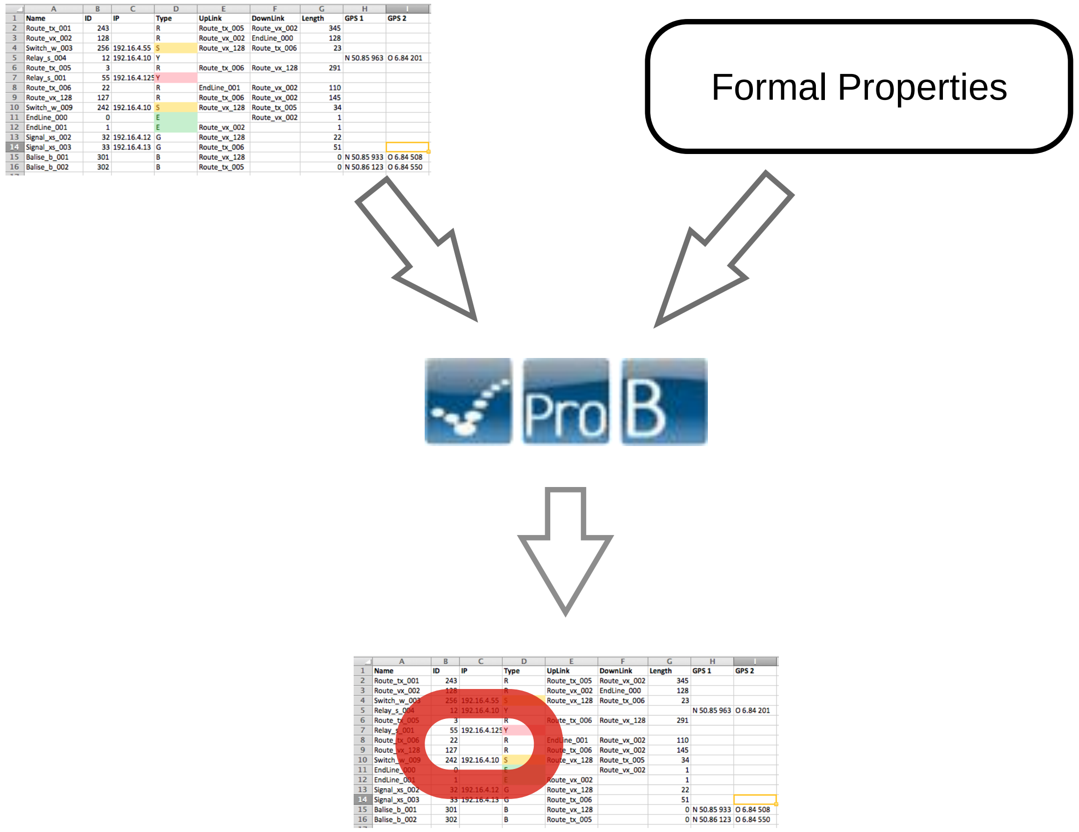
Intelligible
Feedback
on your data

The most expressive language
in the world

> 2,500 years of human
experience

Unambiguous,
Easy to express,
Easy to understand,
Easy to adapt

Why ProB



	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

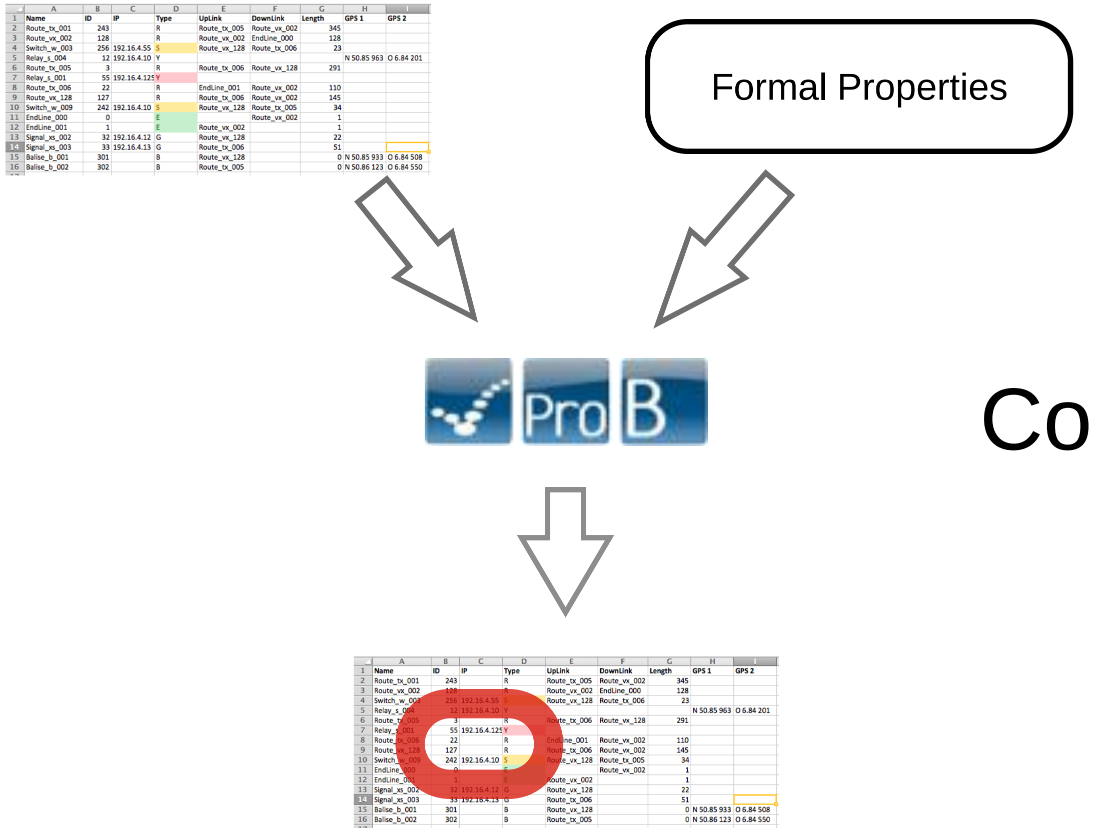
	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

Efficient Engine
Fully Automatic

Validated Engine

In use by
Siemens,
Alstom,...

Why ProB



	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

Formal Properties

	A	B	C	D	E	F	G	H	I
1	Name	ID	IP	Type	UpLink	DownLink	Length	GPS 1	GPS 2
2	Route_tx_001	243		R	Route_tx_005	Route_vx_002	345		
3	Route_vx_002	128		R	Route_vx_002	EndLine_000	128		
4	Switch_w_003	256	192.16.4.55	S	Route_vx_128	Route_tx_006	23		
5	Relay_s_004	12	192.16.4.10	Y				N 50.85 963	O 6.84 201
6	Route_tx_005	3		R	Route_tx_006	Route_vx_128	291		
7	Relay_s_001	55	192.16.4.125	Y					
8	Route_tx_006	22		R	EndLine_001	Route_vx_002	110		
9	Route_vx_128	127		R	Route_tx_006	Route_vx_002	145		
10	Switch_w_009	242	192.16.4.10	S	Route_vx_128	Route_tx_005	34		
11	EndLine_000	0		E		Route_vx_002	1		
12	EndLine_001	1		E	Route_vx_002		1		
13	Signal_xs_002	32	192.16.4.12	G	Route_vx_128		22		
14	Signal_xs_003	33	192.16.4.13	G	Route_tx_006		51		
15	Balise_b_001	301		B	Route_vx_128		0	N 50.85 933	O 6.84 508
16	Balise_b_002	302		B	Route_tx_005		0	N 50.86 123	O 6.84 550

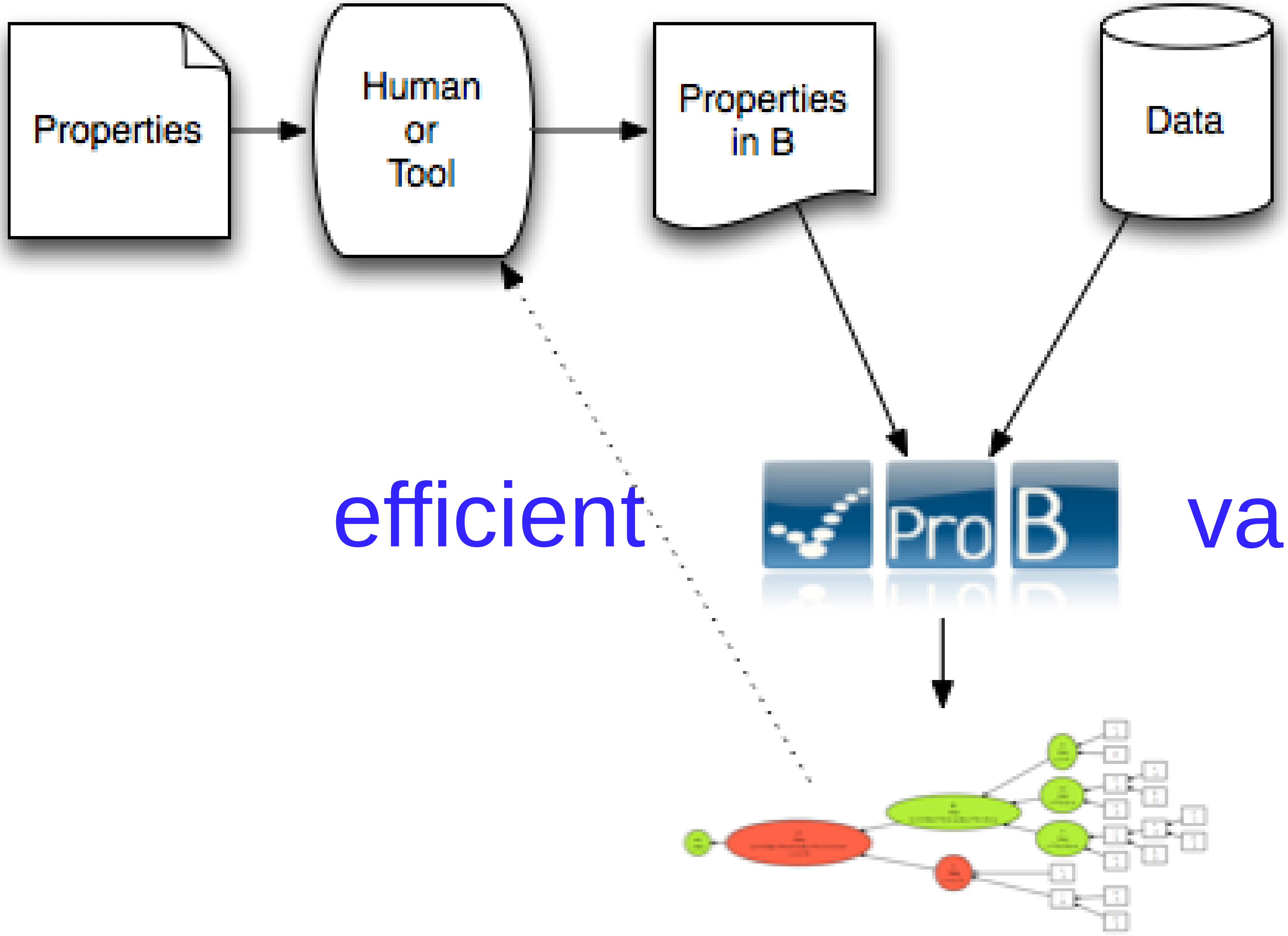
Open Source Core
Command-line interfaces
Java-API

Can fit into your process

formalmind
science for systems engineering

Summary

very expressive



efficient

validated

Infrastructure

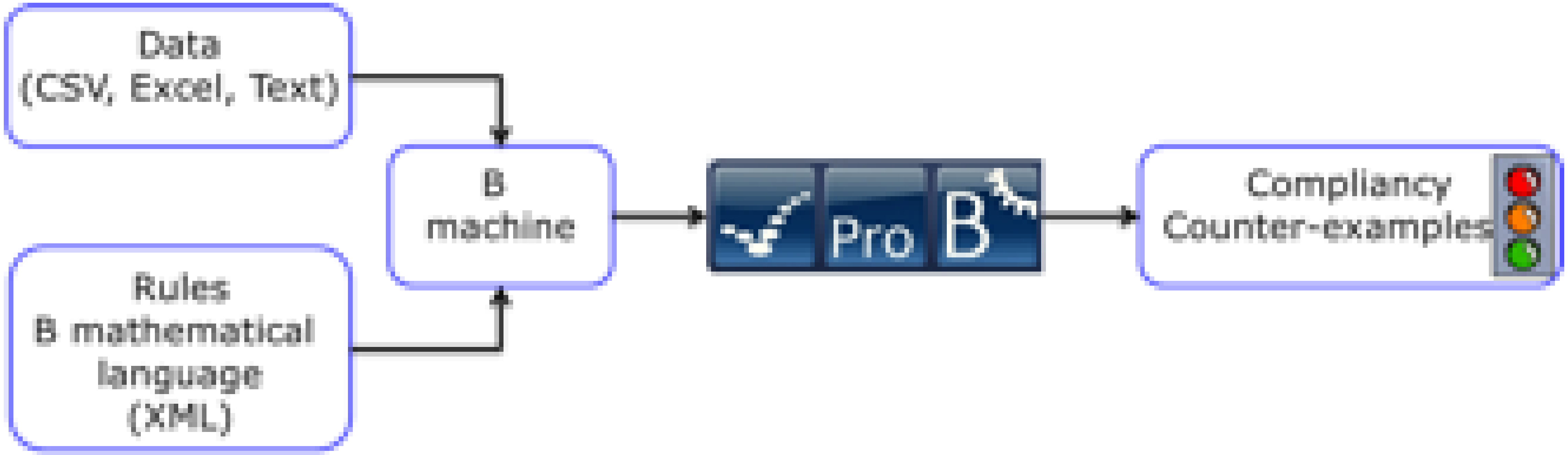
Coverage Reports

good feedback

probcli fits into design flow

DTVT

- Tool based on ProB developed by
  - ClearSy, Alstom, Formal Mind



E_a_trainDynamicDeparture_minimum_speed

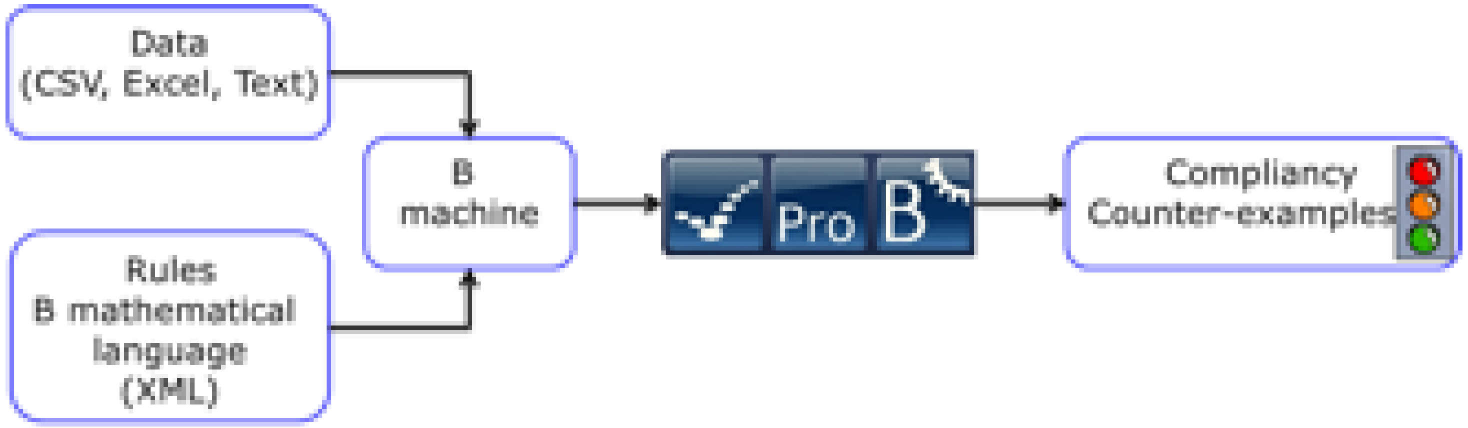
Description :

Train dynamic departure minimum speed

Typage :

```
E_a_trainDynamicDeparture_minimum_speed : INT --> FLOAT
```

Range Excel du domaine : Train_Dynamics!A7:A27
Range Excel du codomaine : Train_Dynamics!AM7:AM27



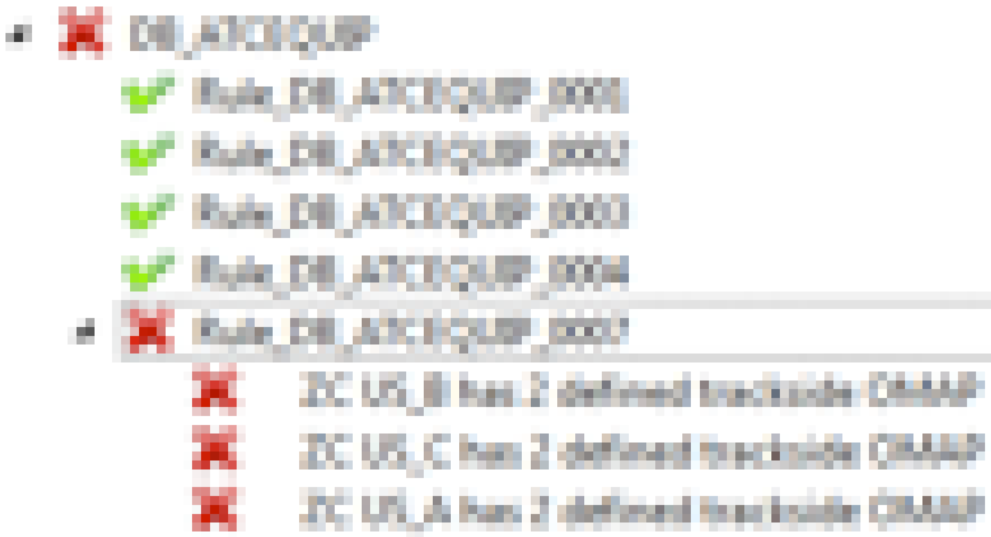
Data
(CSV, Excel, Text)

Rules
B mathematical
language
(XML)

B
machine

Pro B

Compliancy
Counter-examples



DB_ATCEQUIP
Rule_DB_ATCEQUIP_0001
Rule_DB_ATCEQUIP_0002
Rule_DB_ATCEQUIP_0003
Rule_DB_ATCEQUIP_0004
Rule_DB_ATCEQUIP_0007
ZC US_B has 2 defined trackside OMAP
ZC US_C has 2 defined trackside OMAP
ZC US_A has 2 defined trackside OMAP

Propriété VS_CS2 :

Regle associée : 104

Description :

Les zones de freinage ne se recouvrent pas

Expression formelle :

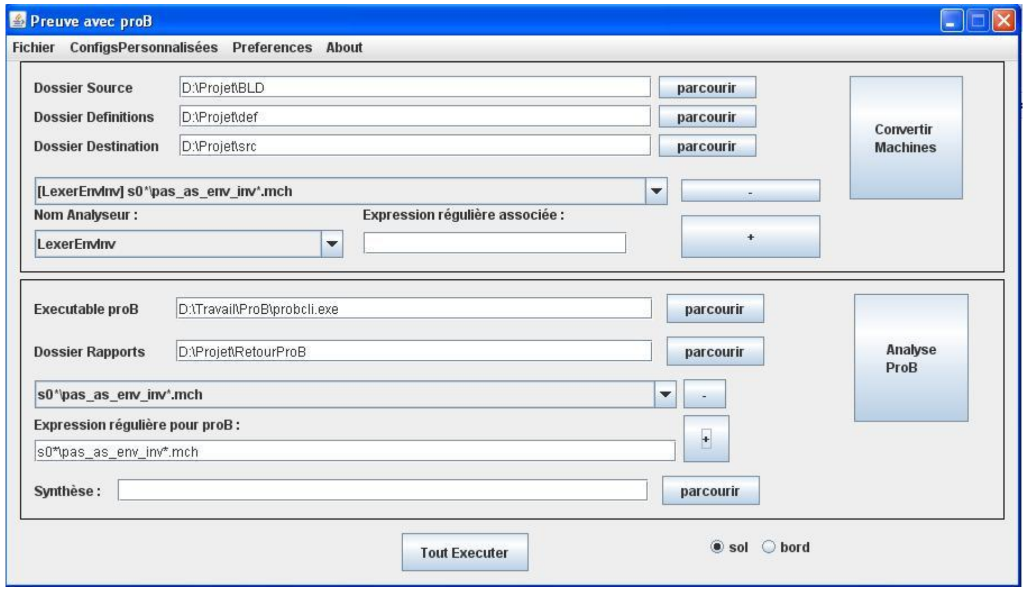
```
!(r1,r2).(r1 : t_regenerativeBraking &
          r2 : t_regenerativeBraking &
          r1 /= r2
          =>
          a_regenerativeBrakingArea(r1)
          |-> a_regenerativeBrakingArea(r2) /: f_areaIntersectArea)
```

the future

- B as a high-level
  - query language
  - constraint-solving language
  - programming language

b as high-level query language

- data validation:
  - Siemens
  - Alstom, ClearSy, ...



Dossier Source: D:\Projet\BLD
Dossier Definitions: D:\Projet\def
Dossier Destination: D:\Projet\src
[LexerEnvInv] s0*\pas_as_env_inv*.mch
Nom Analyseur : LexerEnvInv
Expression régulière associée :
Executable proB: D:\Travail\ProB\probcli.exe
Dossier Rapports: D:\Projet\RetourProB
s0*\pas_as_env_inv*.mch
Expression régulière pour proB : s0*\pas_as_env_inv*.mch
Synthèse :
● sol ○ bord

- many properties can be conveniently expressed in B and now be checked on real data with ProB
- double chain possible: Ovado, ProB-Kodkod, PyProB (in development)

b as a high-level constraint solving language

- very easy to express properties:

@perm p∈ Nodes ⤖ Nodes
@iso ∀x,y•(x∈Nodes ∧ y∈Nodes ⇒ (x↦y∈graph1 ⇔ p(x)↦p(y) ∈ graph2))

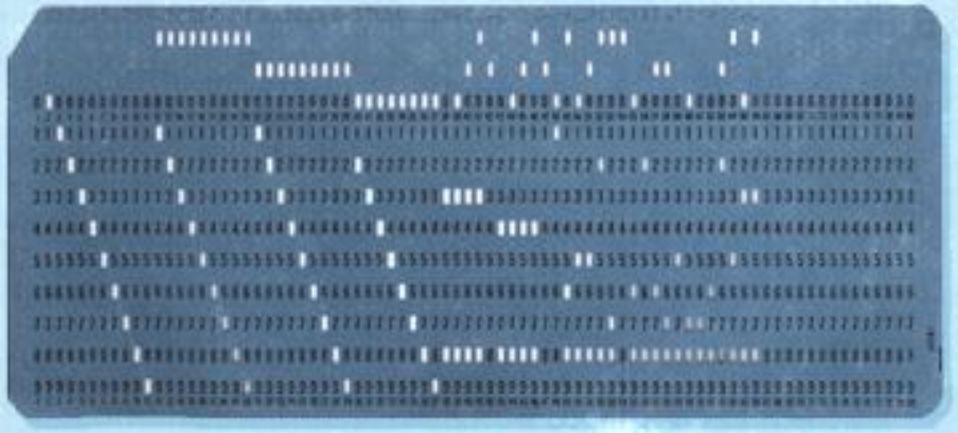
@ctype colour∈ Vtx → 1 · ·maxcol
@alldiff (∀i,j•i↦j∈graph1 ⇒ colour(i) ≠ colour(j))

THE EVOLUTION OF PROGRAMMING

presented by Wikibon

1954 FORTRAN

```
*
C Hello World in Fortran 77
C (lines must be 6 characters indented)
*
    PROGRAM HELLOW
    WRITE(UNIT=*, FMT=*) 'Hello World'
    END
```



1958 LISP

```
(print "Hello World")
```

1959 COBOL

```
DENTIFICATION DIVISION.
PROGRAM-ID. HelloWorld.
AUTHOR. Fabritius.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.

DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
LINKAGE SECTION.

PROCEDURE DIVISION.
DISPLAY "Hello World".
STOP RUN.
```

1962 SIMULA

```
begin
   OutText("Hello World");
   OutImage
end
```

1964 BASIC

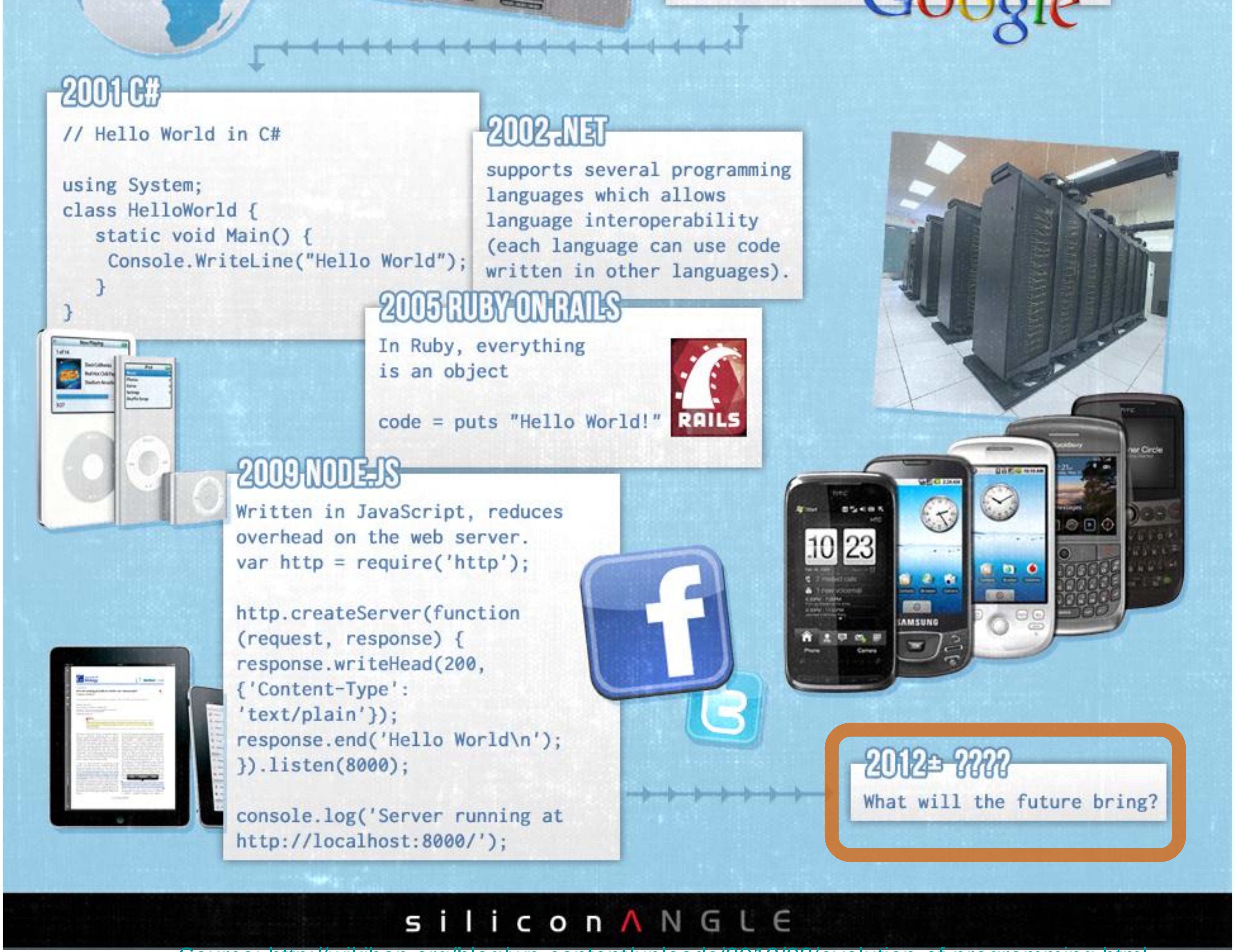
```
PRINT "Hello World"
```

1968 PASCAL

```
PROGRAM HelloWorld;
BEGIN
   WRITELN('Hello World');
END.
```

1983 C±±

```
#include <stdio.h>
```



2001 C#

```
// Hello World in C#

using System;
class HelloWorld {
    static void Main() {
     Console.WriteLine("Hello World");
    }
}
```

2002 .NET

supports several programming languages which allows language interoperability (each language can use code written in other languages).

2005 RUBY ON RAILS

In Ruby, everything is an object

```
code = puts "Hello World!"
```

2009 NODE.JS

Written in JavaScript, reduces overhead on the web server.

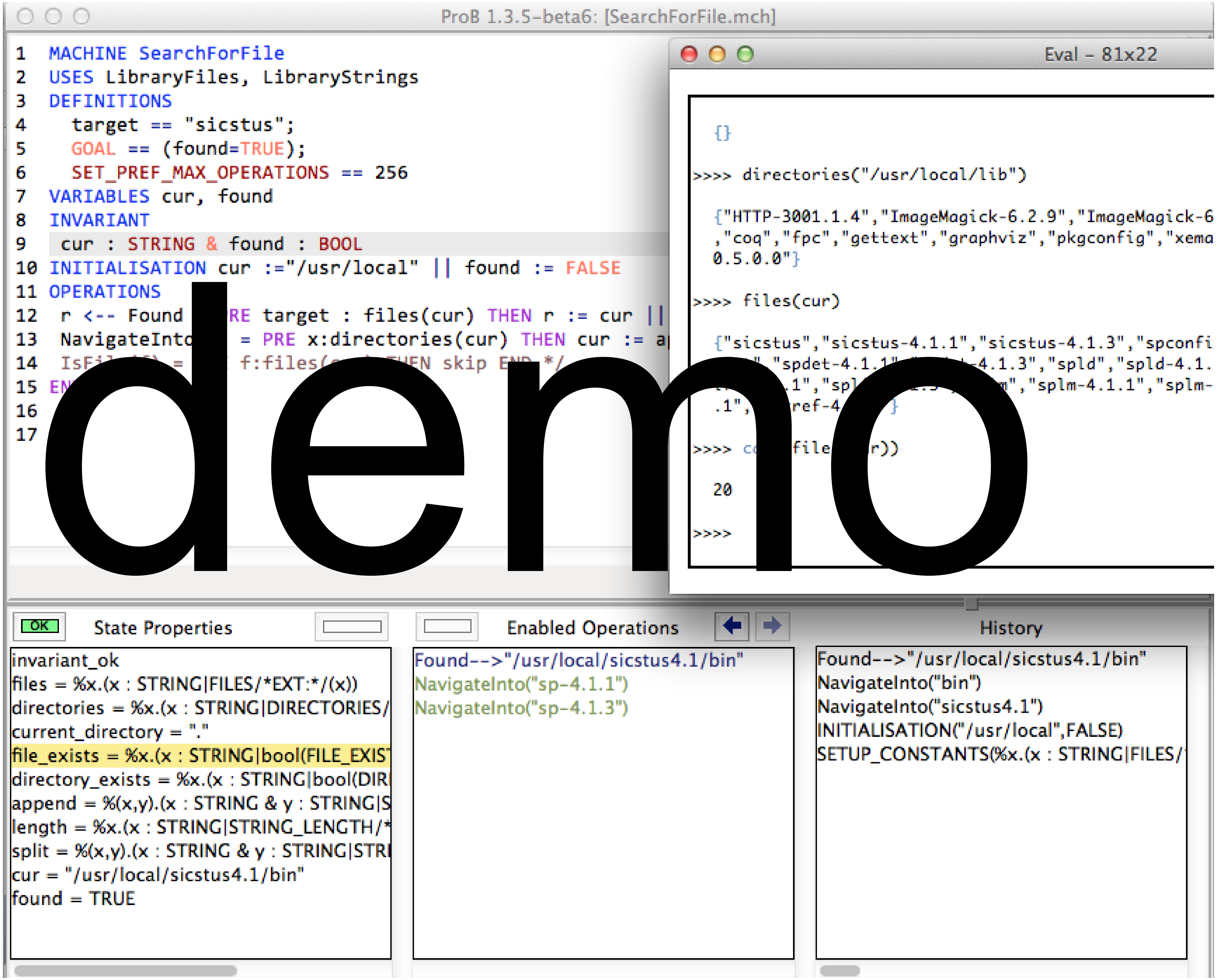
```
var http = require('http');

http.createServer(function
(request, response) {
response.writeHead(200,
{'Content-Type':
'text/plain'});
response.end('Hello World\n');
}).listen(8000);

console.log('Server running at
http://localhost:8000/');
```

2012± ????

What will the future bring?

Source: http://wikibon.org/blog/wp-content/uploads/2012/02/evolution-of-programming.html

demo



```
MACHINE SearchForFile
USES LibraryFiles, LibraryStrings
DEFINITIONS
  target == "sicstus";
  GOAL == (found=TRUE);
  SET_PREF_MAX_OPERATIONS == 256
VARIABLES cur, found
INVARIANT
 cur : STRING & found : BOOL
INITIALISATION cur :="/usr/local" || found := FALSE
OPERATIONS
 r <-- Found      RE target : files(cur) THEN r := cur ||
 NavigateInto      = PRE x:directories(cur) THEN cur := a
 IsFi            f:files(       THEN skip E   */
EN
```

```
 {}

>>>> directories("/usr/local/lib")

 {"HTTP-3001.1.4","ImageMagick-6.2.9","ImageMagick-6
 ,"coq","fpc","gettext","graphviz","pkgconfig","xema
 0.5.0.0"}

>>>> files(cur)

 {"sicstus","sicstus-4.1.1","sicstus-4.1.3","spconfi
      ","spdet-4.1.1"      -4.1.3","spld","spld-4.1.
       .1","sp                m","splm-4.1.1","splm-
 .1",   ref-4       }

>>>> c   file    r))

 20

>>>>
```

```
invariant_ok
files = %x.(x : STRING|FILES/*EXT:*/(x))
directories = %x.(x : STRING|DIRECTORIES/
current_directory = "."
file_exists = %x.(x : STRING|bool(FILE_EXIS
directory_exists = %x.(x : STRING|bool(DIR
append = %(x,y).(x : STRING & y : STRING|S
length = %x.(x : STRING|STRING_LENGTH/*
split = %(x,y).(x : STRING & y : STRING|STR
cur = "/usr/local/sicstus4.1/bin"
found = TRUE
```

```
Found-->"/usr/local/sicstus4.1/bin"
NavigateInto("sp-4.1.1")
NavigateInto("sp-4.1.3")
```

```
Found-->"/usr/local/sicstus4.1/bin"
NavigateInto("bin")
NavigateInto("sicstus4.1")
INITIALISATION("/usr/local",FALSE)
SETUP_CONSTANTS(%x.(x : STRING|FILES/
```

b as a high-level programming language

- Alstom ongoing project:
  - large data (cf data validation)
  - find data (cf constraint solving)
  - computation: infinite functions, recursive functions, external functions (sin, cos, ...)
  - efficiency important

conclusion

- move systems engineering to the next level
- move B to next level
  - ProB, BMotionStudio, ProR

formalmind

science for systems engineering

thank you

- http://www.formalmind.com/

backup slides

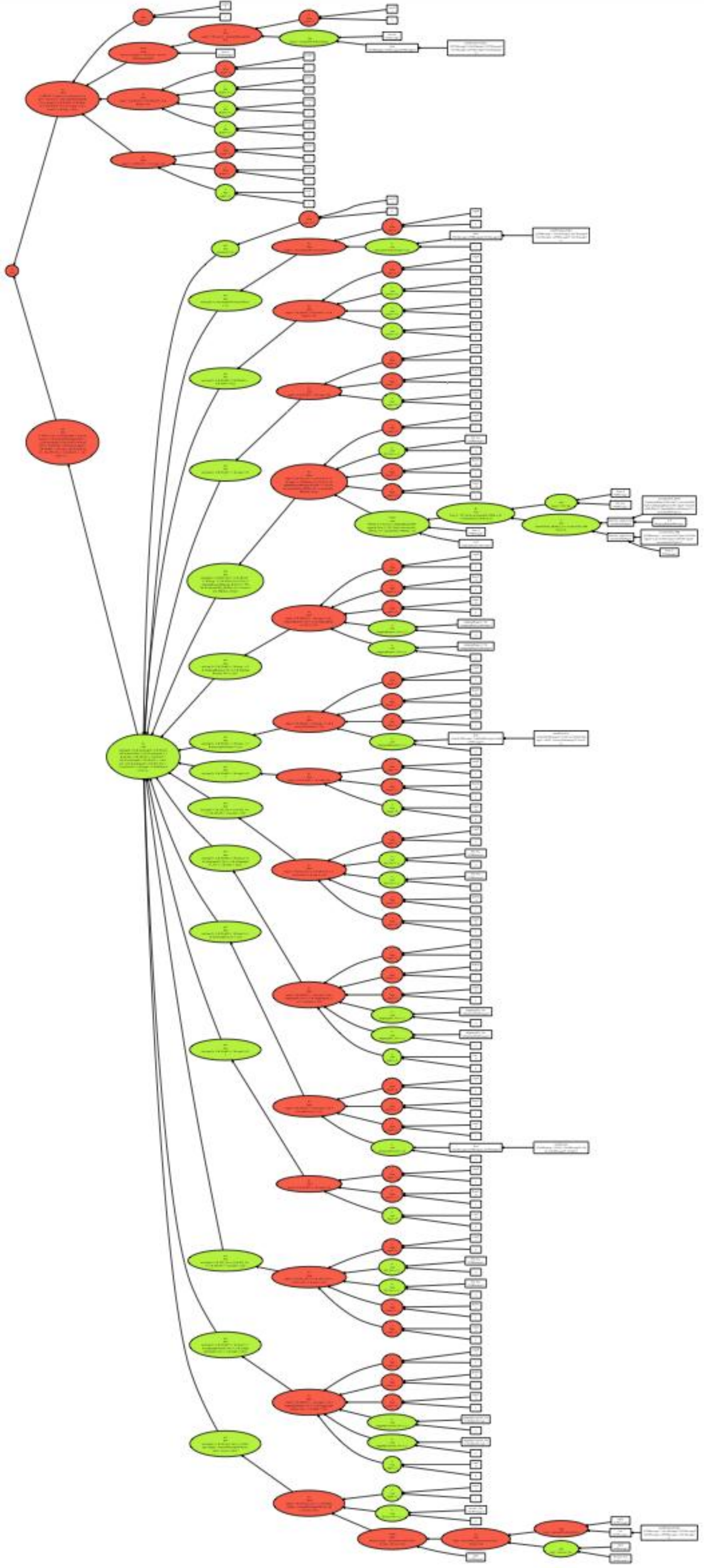
CBC Deadlock Checking

- successful Deploy case study with Bosch

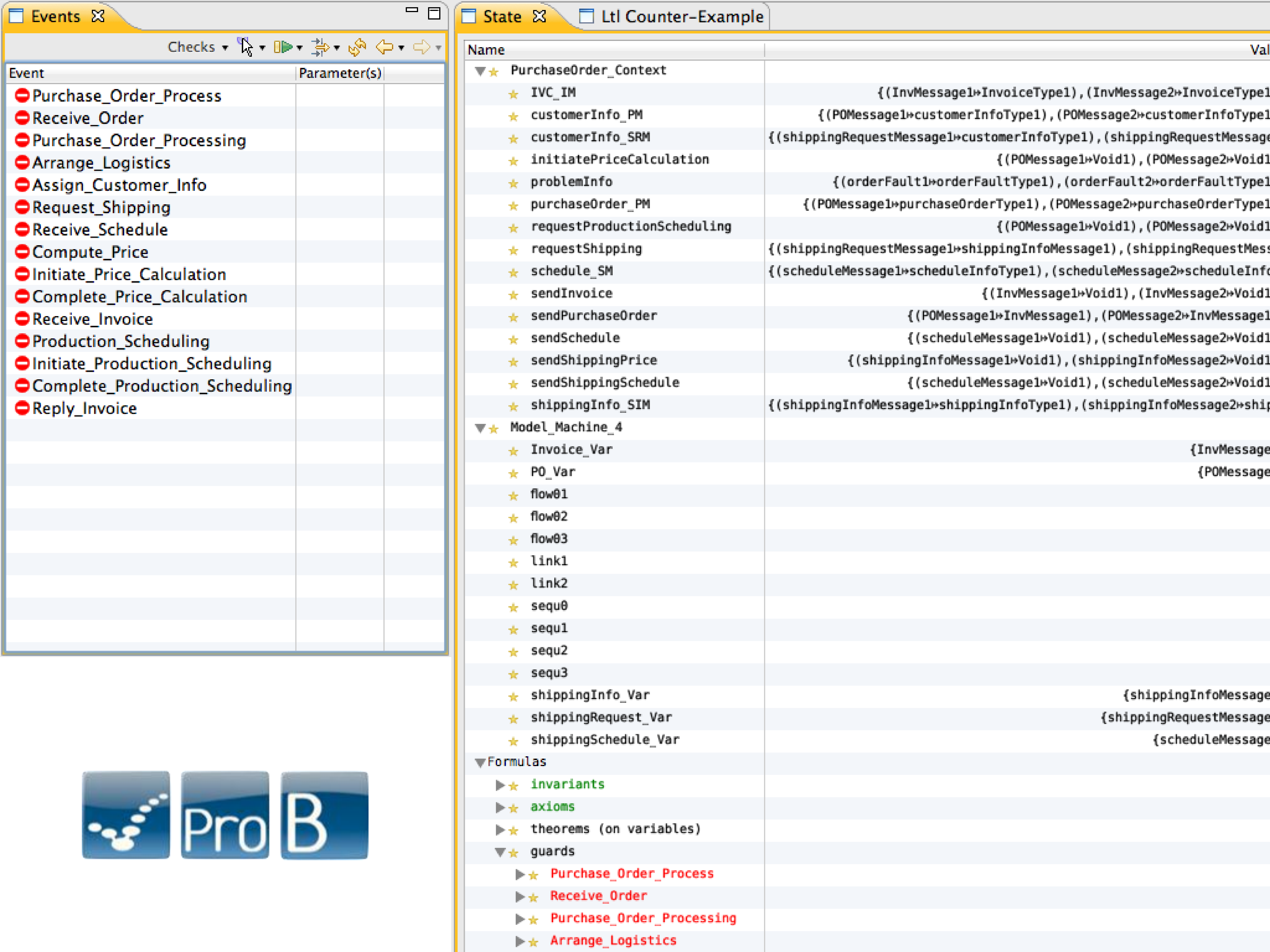
BPEL Example:

SMT Solvers Plugin + Z3:

ProB:



```
$z3 -smt2 dlf1_z3.smt
...WARNING: pulled nested quantifier to be able to find
an useable pattern (quantifier id: k!405)unknown
```

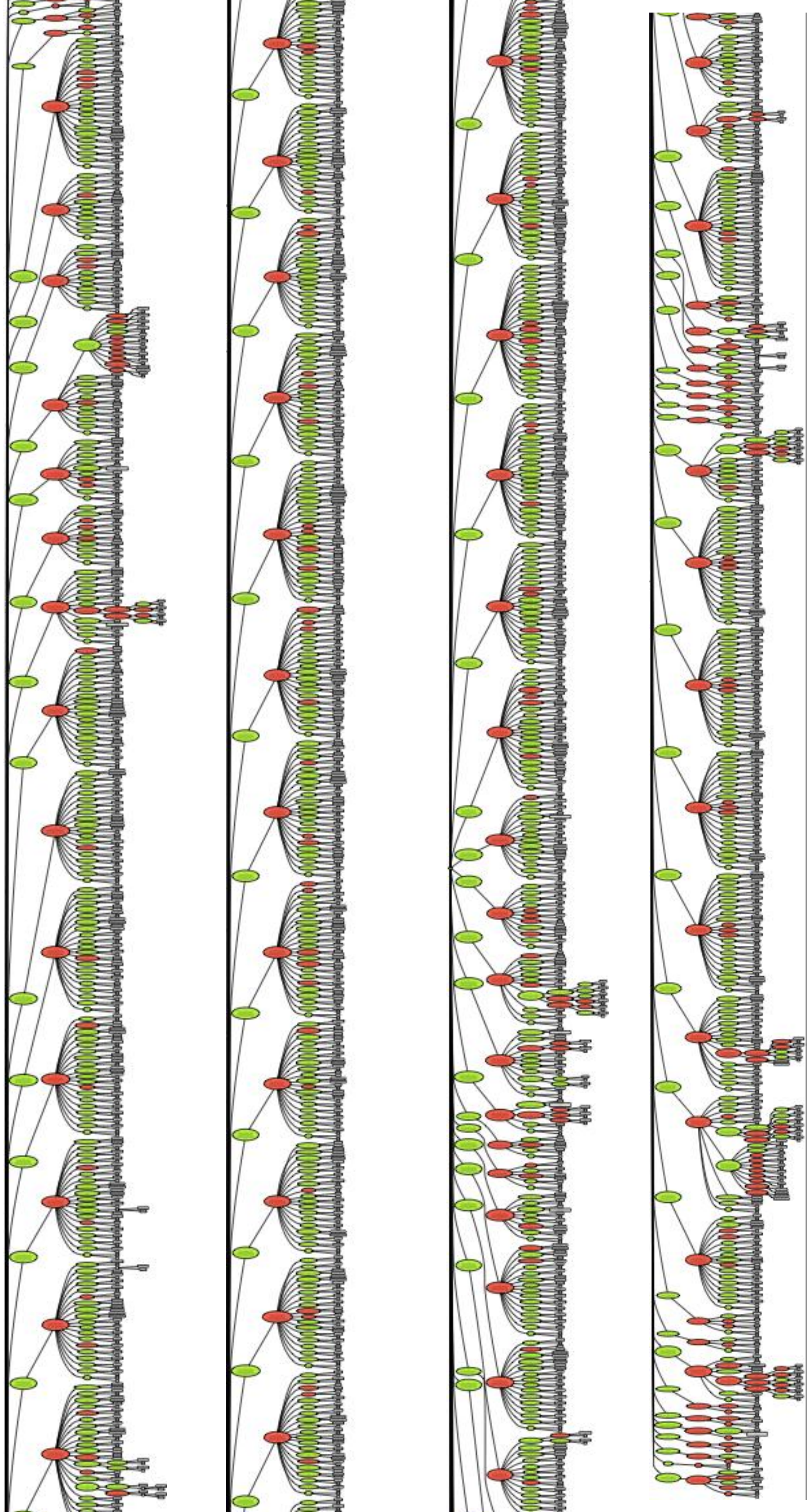


Event	Parameter(s)
Purchase_Order_Process	
Receive_Order	
Purchase_Order_Processing	
Arrange_Logistics	
Assign_Customer_Info	
Request_Shipping	
Receive_Schedule	
Compute_Price	
Initiate_Price_Calculation	
Complete_Price_Calculation	
Receive_Invoice	
Production_Scheduling	
Initiate_Production_Scheduling	
Complete_Production_Scheduling	
Reply_Invoice	

Name	Val
PurchaseOrder_Context	
IVC_IM	{(InvMessage1↦InvoiceType1),(InvMessage2↦InvoiceType1
customerInfo_PM	{(POMessage1↦customerInfoType1),(POMessage2↦customerInfoType1
customerInfo_SRM	{(shippingRequestMessage1↦customerInfoType1),(shippingRequestMessag
initiatePriceCalculation	{(POMessage1↦Void1),(POMessage2↦Void1
problemInfo	{(orderFault1↦orderFaultType1),(orderFault2↦orderFaultType1
purchaseOrder_PM	{(POMessage1↦purchaseOrderType1),(POMessage2↦purchaseOrderType1
requestProductionScheduling	{(POMessage1↦Void1),(POMessage2↦Void1
requestShipping	{(shippingRequestMessage1↦shippingInfoMessage1),(shippingRequestMes
schedule_SM	{(scheduleMessage1↦scheduleInfoType1),(scheduleMessage2↦scheduleInf
sendInvoice	{(InvMessage1↦Void1),(InvMessage2↦Void1
sendPurchaseOrder	{(POMessage1↦InvMessage1),(POMessage2↦InvMessage1
sendSchedule	{(scheduleMessage1↦Void1),(scheduleMessage2↦Void1
sendShippingPrice	{(shippingInfoMessage1↦Void1),(shippingInfoMessage2↦Void1
sendShippingSchedule	{(scheduleMessage1↦Void1),(scheduleMessage2↦Void1
shippingInfo_SIM	{(shippingInfoMessage1↦shippingInfoType1),(shippingInfoMessage2↦shi
Model_Machine_4	
Invoice_Var	{InvMessage
PO_Var	{POMessage
flow01	
flow02	
flow03	
link1	
link2	
sequ0	
sequ1	
sequ2	
sequ3	
shippingInfo_Var	{shippingInfoMessage
shippingRequest_Var	{shippingRequestMessage
shippingSchedule_Var	{scheduleMessage
Formulas	
invariants	
axioms	
theorems (on variables)	
guards	
Purchase_Order_Process	
Receive_Order	
Purchase_Order_Processing	
Arrange_Logistics	

Latest Bosch Experiments

- 78 constants, 121 axioms, 62 variables, 59 invariants, 80 events, 855 guards
- card: 79 ×∞, 1×2^{65}, 1×2^{52}, 11×2^{32}, ...
- 34 pages of A4 formula solved in 1-2 seconds
- So far: no success with SMT/SAT



for Bosch example

```
kodkod.engine.CapacityExceededException: Arity too large (10) for a universe of size 35   at
kodkod.instance.TupleFactory.checkCapacity(TupleFactory.java:266)     at
kodkod.instance.TupleFactory$IntTuple.<init>(TupleFactory.java:325)   at
kodkod.instance.TupleFactory.tuple(TupleFactory.java:88)    at
de.stups.probkodkod.types.TupleType.createAllTuples(TupleType.java:95)          at
de.stups.probkodkod.KodkodAnalysis.extractTupleSet(KodkodAnalysis.java:725)     at
de.stups.probkodkod.KodkodAnalysis.addRelations(KodkodAnalysis.java:666)        at
de.stups.probkodkod.KodkodAnalysis.caseAProblem(KodkodAnalysis.java:246)        at
de.stups.probkodkod.parser.node.AProblem.apply(AProblem.java:75)      at
de.stups.probkodkod.parser.analysis.DepthFirstAdapter.caseAProblemAction(DepthFirstAdapter.java:55) at
de.stups.probkodkod.parser.node.AProblemAction.apply(AProblemAction.java:34)    at
de.stups.probkodkod.parser.analysis.DepthFirstAdapter.caseStart(DepthFirstAdapter.java:34)          at
de.stups.probkodkod.parser.node.Start.apply(Start.java:36)  at
de.stups.probkodkod.KodkodInteraction.interaction(KodkodInteraction.java:54)    at
de.stups.probkodkod.KodkodInteraction.main(KodkodInteraction.java:95)
```