Formal Development of Cyber-Physical Systems: The Event-B Approach

Paulius Stankaitis
AMBER research group, School of Computing, Newcastle University

The British Computer Society (FACS group)
April 4th, 2023

About Me

- Junior Research Assistant (2014-16) on the SafeCap project: formal methods for a safe and optimum railway,
- PhD work (2016-20, iCase w. Siemens Rail Automation) on formal engineering of heterogeneous railway signalling systems,
- Post-doctoral work (2020-) on the integration of hybridised Event-B and reachability analysis, real-time reachability analysis of autonomous systems and safe AI.

Cyber-Physical Systems

What are Cyber-Physical Systems (CPS)?

- integrate **computation** and **physical** processes,
- **networked** computers control physical systems.

Examples of CPS can be found in many industry sectors[1], [2]:

[1] https://www.phillymag.com/healthcare-news/2019/07/15/medcrypt-hack-proof-medical-devices/
[2] https://sites.rmit.edu.au/cyber-physical-systems/

Cyber-Physical Systems

What are Cyber-Physical Systems (CPS)?

- integrate **computation** and **physical** processes,
- **networked** computers control physical systems.

Examples of CPS can be found in many industry sectors.

Importantly many of these systems are **safety-critical**.

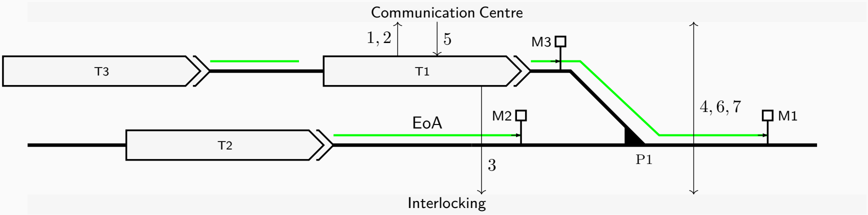
Cyber-Physical Railway Signalling Systems

2 Photo taken from https://www.networkrail-training.co.uk/media/Signaller_Training_01.jpeg

Cyber-Physical Railway Signalling Systems

Railway signalling systems are safety-critical cyber-physical systems:

- European Train Control System (ETCS L0-3, part of ERTMS), Communication-based Train Control (CBTC),

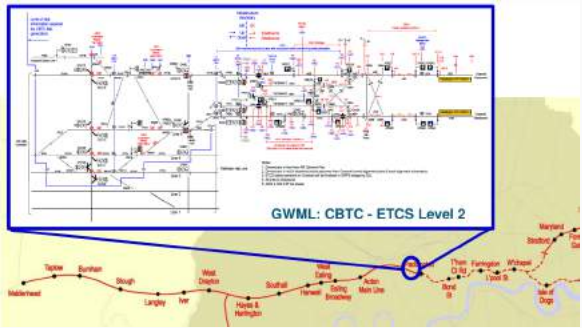


Trains are **hybrid systems** (discrete and continuous behaviour)

Cyber-Physical Railway Signalling Systems

Railway signalling systems are safety-critical cyber-physical systems:

- European Train Control System (ETCS L0-3, part of ERTMS), Communication-based Train Control (CBTC),
- **Heterogeneous** railway signalling networks (Crossrail, Thameslink).

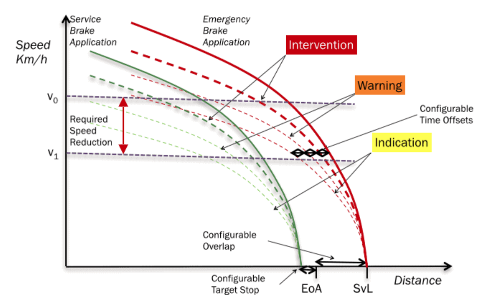


Trains are **hybrid systems** (discrete and continuous behaviour)

Cyber-Physical Railway Signalling Systems

Trains are **hybrid systems** (discrete and continuous behaviour)[3].

- European Vital Computed (EVC) computes braking curves and intervenes if braking curves are breached.



3 https://www.graffica.co.uk/case-studies/hermes-etcs-modelling/

Formal Methods for Railway Signalling Systems

Formal methods have been used in the railway domain, for example:

- The **B** method Paris Metro, Paris Roissy Airport shuttle.

Formal Verification of **control tables** and interlocking software (Solid State Interlocking (SSI)):

- push-button model-checking approaches.

ROUTE		INTERLOCKING				CONTROL				
		REQUIRES	SET & LOCKS POINTS		REQUIRES KEYLOCK	ASPECT	SIGNAL AHEAD	REQUIRES TC		
								CLEAR	AT TIME OF CLEARING ONLY	
From	To	ROUTE NORMAL	NORMAL	REVERSE	NORMAL				TC CLEAR OR	OCC FOR TIME
1	3					Y G	3 AT R# 3 AT Y# OR G#			
2	4					Y G	4 AT R# 4 AT Y# OR G#			
3(1)	23	16,24,4(1),4(2),3(2)		103,104	201,202, 203,204	Y+JI	23 AT R#	3T,9T,103T,24T,41T, 23T,104T,8T,4T	42	42 FOR 60 sec
3(2)	15	16,24,4(1),4(2),3(1)	103,104			Y G	15 AT R# 15 AT G#	3T,9T,103T,16T,42T, 15T,104T,8T,4T	41	41 FOR 60 sec
4(1)	16	15,23,3(1),3(2),4(2)	104,103			Y G	16 AT R# 16 AT G#	4T,8T,104T,15T,42T, 16T,103T,9T,3T	41	41 FOR 60 sec
4(2)	24	15,23,3(1),3(2),4(1)		104,103	201,202, 203,204	Y+JI	24 AT R#	4T,8T,104T,23T,41T, 24T,103T,9T,3T	42	42 FOR 60 sec
15	UP BLOCK SECTION	23,4(1),4(2)	104,103			G		15T,104T,8T,4T,2T,TOL		
23	UP BLOCK SECTION	15,4(1),4(2)		104,103		G		23T,104T,8T,4T,2T,TOL		
16	DOWN BLOCK SECTION	24,3(1),3(2)	103,104			G		16T,103T,9T,3T,1T,TOL		
24	DOWN BLOCK SECTION	16,3(1),3(2)		103,104		G		24T,103T,9T,3T,1T,TOL		

[3] Control Table example from S. Vanit-Anunchai: Verification of Railway Interlocking Tables Using Coloured Petri Nets. COORDINATION, 2010.

Formal Methods for Railway Signalling Systems

Formal methods have been used in the railway domain, for example:

– The **B** method Paris Metro, Paris Roissy Airport shuttle.

Formal Verification of control tables and the **interlocking software** (Solid State Interlocking (SSI)):

– automated theorem provers (e.g., The Formal Route company).

```
*QR117B(M)                       / route request block for route R117B(M)
  if R117B(M) a                  / route R117B(M) is available
           USD-CA f,OSC-BA f,OSV-BA f   / sub-route and sub-overlaps are free
        then if OSL-AC l,        / sub-overlap is OSL-AC locked
              P223 fr , P224 fr   / points P223, P224 free to move reverse
            then @P223QR \   / call subroutine P223QR
          if OSD-BC f         / sub-overlap is OSD-BC is free
             LTR04 xs         / latch (boolean flag) not set (false)
             P224 crf         / point P224 commanded reverse or free to move reverse
          then R117B(M) s    / set route set flag for R117B(M)
             USD-AC l , USC-AB l , USB-AB l , OSA-AB l  / set sub-routes/overlaps
             P224 cr          / command point P224 reverse
             LARR xs          / clear latch LARR
             S117 clear bpull    / clear signal button pull flag
             if P223 xcr , P223 rf then / check point states
              @P223QR    / point command subroutine
                 EP230 = 0 \ / reset timer EP230
```

[3] SSI example from Iliasov et al.: Formal Verification of Signalling Programs with SafeCap. SAFECOMP, 2018.

Framework for CPS Design and Analysis

Formal CPS development framework which utilises abstraction and refinement.

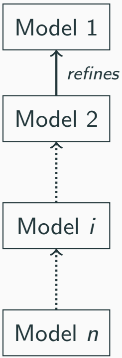
Enables a **multifaceted** CPS design:

- simulation-based system validation and analysis,
- model constraints and safe parameter values via reachability analysis.

Improves **scalability** of formal verification:

- automation of formal verification of hybrid systems,
- challenge of deriving differential invariant.

State-based pivot model (A)

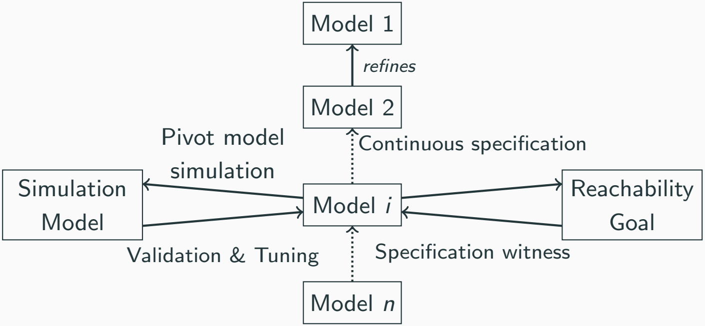


Framework for CPS Design and Analysis

Enables a **multifaceted** CPS design:

- simulation-based system validation and analysis,
- model constraints and safe parameter value discovery via reachability analysis.

Simulation-based analysis (C)	State-based pivot model (A)	Reachability analysis (B)

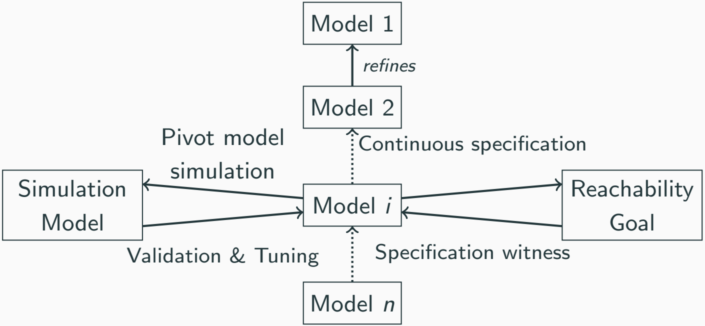


Framework for CPS Design and Analysis

Improves **scalability** of formal verification:

- automation of formal verification of hybrid systems,
- challenge of deriving differential invariant.

Simulation-based analysis (C) | State-based pivot model (A) | Reachability analysis (B)



From Event-B to Hybridised Event-B

From Event-B to Hybridised Event-B

The **B** method:

- formal software development method proposed by J.-R. Abrial.

The **Event-B** method:

- evolution of the **B** method for formal system-level modelling and verification.
- key features of the **Event-B** method:
  - set-theoretic modelling notation,
  - refinement- and proof- driven approach,
  - good tool support (Eclipse-based Rodin platform, ProB model checker, Theory plug-in, SMT solvers).

Both methods are used in academia and industry (e.g., Siemens Transportation, ALSTOM, CLEARSY and others)

From Event-B to Hybrid Event-B

The structure of **Event-B** models:

- a *context* holds static information about the system,
- a *machine* describes dynamic system aspects,
- properties about the system can be expressed as invariants (e.g. inv_2),
- 10 different types of possible proof obligations,
- (discrete) **Event-B** model verification automation has been significantly improved.

CONTEXT ctx0
SETS
  CRS
CONSTANTS
  m
AXIOMS
  axm_0 finite(CRS)
  axm_0 $m \in \mathbb{N}1$
  axm_0 $m \leq \text{card}(CRS)$
END

From Event-B to Hybrid Event-B

The structure of **Event-B** models:

- a *context* holds static information about the system,
- a *machine* describes dynamic system aspects,
- properties about the system can be expressed as invariants (e.g. inv_2),
- 10 different types of possible proof obligations,
- (discrete) **Event-B** model verification automation has been significantly improved.

MACHINE m0
VARIABLES
x
INVARIANTS
inv_1 $x \in \mathbb{N}$
inv_2 $\mathbf{x \leq 11}$
EVENTS
INITIALISATION
THEN
act_1 : $x := 0$
END
Increment
WHERE
grd_1 : $x \leq 10$
THEN
act_1 : $x := x + 1$
END
END

From Event-B to Hybrid Event-B

MACHINE m0
VARIABLES
x
INVARIANTS
$\text{inv}_1 \quad x \in \mathbb{N}$
$\text{inv}_2 \quad x \leq 11$
EVENTS
INITIALISATION
THEN
$\text{act}_1 : \quad x := 0$
END
Increment
WHERE
$\text{grd}_1 : \quad x \leq 10$
THEN
$\text{act}_1 : \quad x := x + 1$
END
END

Invariant Preservation Rule

Axioms
Invariants
Event Guards
Event BAP
$\vdash$
Modified Specific Invariant

$x \in \mathbb{N}$	$x \in \mathbb{N}$
$x \leq 10$	$x = 0$
$\vdash$	$\vdash$
$x + 1 \leq 11$	$x \leq 11$

From Event-B to Hybrid Event-B

MACHINE m0
VARIABLES
x
INVARIANTS
inv_1 $x \in \mathbb{N}$
inv_2 $x \leq 11$
EVENTS
INITIALISATION
THEN
act_1 : $x := 0$
END
Increment
WHERE
grd_1 : $\top$
THEN
act_1 : $x :| x' = x + 1 \land \mathbf{x' + 1 \leq 11}$
END
END

Feasibility

Axioms
Invariants
Event Guards
$\vdash$
$\exists v' \cdot$ Event BAP

Note: Rewriting act_1 with *such that* and strengthening before-after predicate we can automatically prove inv_2 but need to prove feasibility.

From Event-B to Hybridised Event-B

The Rodin Theory plug-in allows extending the **Event-B** mathematical language:[4]

THEORY Seq
TYPE PARAMETERS A
OPERATORS
seq expression $seq(a : \mathbb{P}(A))$
direct definition
$seq(a : \mathbb{P}(A)) \triangleq \{n, f \cdot n \in \mathbb{N} \wedge f \in 1..n \rightarrow a | f\}$

⋮

AXIOMS
seqsIsFinite $\forall s, a \cdot a \subseteq A \wedge s \in seq(a) \Rightarrow finite(s)$

⋮

PROOF RULES

⋮

END

[4] Event-B theory example based on https://wiki.event-b.org/index.php/Theory_Plug-in

From Event-B to Hybridised Event-B

Hybrid systems are dynamical systems that exhibit discrete and continuous behaviour:

- a hybrid automaton model is used for describing hybrid systems.

The **Event-B** method for hybrid systems:

- Banach et al. Hybrid Event-B: Core Hybrid Event-B I: Single Hybrid Event-B machines
  - new *pliant* events for continuous actions,
  - approach is not tool supported.
- Dupont et al. Correct-by-Construction Design of Hybrid Systems Based on Refinement and Proof (PhD thesis)
  - new Event-B theories (Reals, continuous functions, differential equations, theory of approximations),
  - hybrid system modelling and refinement patterns (generic hybrid Event-B model).

From Event-B to Hybridised Event-B

Hybrid systems are dynamical systems that exhibit discrete and continuous behaviour:

- a hybrid automaton model is used for describing hybrid systems.

The **Event-B** method for hybrid systems:

- Banach et al. Hybrid Event-B: Core Hybrid Event-B I: Single Hybrid Event-B machines
  - new *pliant* events for continuous actions,
  - approach is not tool supported.
- **Dupont et al. Correct-by-Construction Design of Hybrid Systems Based on Refinement and Proof (PhD thesis)**
  - new Event-B theories (Reals, continuous functions, differential equations, theory of approximations),
  - hybrid system modelling and refinement patterns (generic hybrid Event-B model).

From Event-B to Hybridised Event-B

THEORY DiffEq **IMPORT** Functions
TYPE PARAMETERS E, F
DATATYPES
DE(F) **constructors** **ode**(f, η_0, t_0), ...
OPERATORS
solutionOf *predicate* ($D : \mathbb{P}(\mathbb{R})$, $\eta : \mathbb{R} \nrightarrow F$, $\mathcal{E}$: **DE**(F)) ...
Solvable *predicate* ($D : \mathbb{P}(\mathbb{R})$, $\mathcal{E}$: **DE**(F)) ...
CBAP *predicate* ($t, t' : \mathbb{R}^+$, $x_p, x_p' : \mathbb{R} \nrightarrow F$, $\mathcal{P} : \mathbb{P}((\mathbb{R} \nrightarrow F) \times (\mathbb{R} \nrightarrow F))$, $H : \mathbb{P}(F)$)
...
:∼ *predicate* ($t, t' : \mathbb{R}^+$, $x_p, x_p' : \mathbb{R} \nrightarrow F$, $\mathcal{E}$: **DE**(F), $H : \mathbb{P}(F)$)
well−definedness condition **Solvable**($[t, t'], \mathcal{E}$)
direct definition **solutionOf**($[t, t'], x_p', \mathcal{E}$) $\wedge \ldots$
...
AXIOMS
CauchyLipschitz: —— *external*
$\forall \mathcal{E}, D, D_F \cdot \mathcal{E} \in$ **DE**(F) $\wedge \ldots \Rightarrow$ **Solvable**($D, \mathcal{E}$)
...

- use of theories to integrate continuous features
  $\Rightarrow$ *e.g. continuous behaviour using differential equations*
- exploit WD to ensure the correct use of operators/theorems

From Event-B to Hybridised Event-B

Continuous state variables = *functions of time* ($\in \mathbb{R} \nrightarrow S$)
$\Rightarrow$ **continuous evolution** *as CBAP*

$$\mathbf{CBAP}(t, t', x_p, x'_p, \mathcal{P}, H) \equiv$$

$$x_p \mathbin{:}|_{t \to t'} \mathcal{P}(x_p, x'_p) \,\&\, H \equiv$$

$$[0, t[\lhd x'_p = [0, t[\lhd x_p \qquad (\textit{Past Preservation})$$

$$\land \mathcal{P}([0, t] \lhd x_p, [t, t'] \lhd x'_p) \qquad (\textit{Predicate})$$

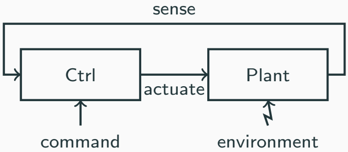
$$\land \forall t^* \in [t, t'], x_p(t^*) \in H \qquad (\textit{Evolution Dom.})$$

Note: *shorthand for differential equations:*

$$x_p \mathbin{:}\sim_{t \to t'} \mathcal{E} \,\&\, H \equiv x_p \mathbin{:}|_{t \to t'} \mathbf{solutionOf}([t, t'], \mathcal{E}, x'_p) \,\&\, H$$

From Event-B to Hybridised Event-B

Hybridised **Event-B** patterns formalise a generic controller-plant-loop hybrid system as Event-B model:



Hybridised **Event-B** machine modelling pattern:

MACHINE Generic
EXTENDS *DiffEquations*
VARIABLES t , x_s , x_p
INVARIANTS
inv$_1$: $t \in \mathbb{R}^+$
inv$_2$: $x_s \in \text{STATES}$
inv$_3$: $x_p \in \mathbb{R} \nrightarrow S$
inv$_4$: $[0, t] \subseteq \mathrm{dom}(x_p)$

- use developed theories (e.g., differential equations),
- explicit time (t),
- discrete state (x_s) + continuous state (x_p, function of time).

From Event-B to Hybridised Event-B

Generic events of hybridised **Event-B** modelling pattern:

Actuate
ANY $\mathcal{P}$, s , H , t'
WHERE
grd$_0$: $t' > t$
grd$_1$: $\mathcal{P} \in (\mathbb{R}^+ \twoheadrightarrow S) \times (\mathbb{R}^+ \twoheadrightarrow S)$
grd$_2$: $\textbf{Feasible}([t, t'], x_p, \mathcal{P}, H)$
grd$_3$: $s \subseteq \text{STATES} \wedge x_s \in s$
grd$_4$: $H \subseteq S \wedge x_p(t) \in H$
THEN
act$_1$: $x_p :|_{t \to t'} \mathcal{P}(x_p, x_p') \,\&\, H$
END

Sense
ANY s , p
WHERE
grd$_1$: $s \in \mathbb{P}1(\text{STATES})$
grd$_2$: $p \in \mathbb{P}(\text{STATES} \times \mathbb{R} \times S)$
grd$_3$: $(x_s \mapsto t \mapsto x_p(t)) \in p$
THEN
act$_1$: $x_s :\in s$
END

- discrete event **Sense** + continuous event **Actuate** (passing of time),
- **Actuate** based on **CBAP**, WD in guard (proved in refinement with guard strengthening),
- Additional generic events **Behave** and **Transition** model changes induced by environment and user.

From Event-B to Hybridised Event-B

New types of proof obligations:

– Continuous invariant preservation: if the invariant is true on $[0, t]$, then it must be true on $[t, t']$, i.e., on the whole duration of the continuous event:

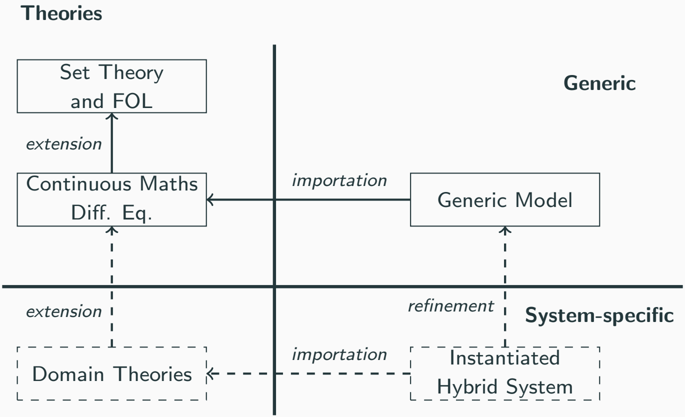
$$\Gamma, \mathcal{I}([0,t] \lhd x_p),\ CBAP(t, t', x_p, x_p', \mathcal{P}, \mathcal{H}) \quad \vdash \mathcal{I}([t,t'] \lhd x_p') \qquad \text{(CINV)}$$

– Continuous feasibility requires to prove that, if the event is triggered, then its action can be performed:

$$\Gamma \vdash \exists t' \cdot t' \in \mathbb{R}^+ \land t' > t \land \textbf{Feasible}([t,t'], x_p, \mathcal{P}, \mathcal{H}_{saf}) \qquad \text{(CFIS)}$$

Important: Proof-obligations related to continuous system behaviour of the model are generally complex and proved interactively.

Hybridised Event-B for CPS Design Framework



The following slides present the framework application for developing a cyber-physical railway signalling system.

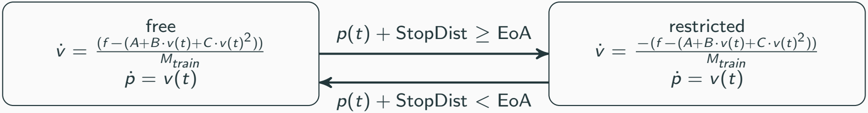
Cyber-Physical Railway Signalling System: Speed Controller

1st refinement of the generic introduces rolling stock.

- A driver (or ATO system) controls a train engine power (tractive force) - f - which yields an acceleration,
- Davis Resistance equation in Equation (1), where A, B, C are fixed parameters and $v(t)$ is the speed of a train at time t:

$$\begin{cases} \dot{v}(t) = \pm(f - (A + B \cdot v(t) + C \cdot v(t)^2))/M_{train} \\ \dot{p}(t) = v(t) \end{cases} \tag{1}$$

- The hybrid automaton model of the train speed controller:



Cyber-Physical Railway Signalling System: Speed Controller

- Properties of the train are gathered in the Train *domain theory*,
- This theory mainly defines the *Davis equation* and its properties

THEORY Trains
OPERATORS
DavisResistance *expression* $(a : \mathbb{R},\ b : \mathbb{R},\ c : \mathbb{R})$
well−definedness condition $a \geq 0,\ b \geq 0,\ c \geq 0$
direct definition $(\lambda v \cdot v \in \mathbb{R} \mid a + bv + cv^2)$
...
THEOREMS
...
END

Cyber-Physical Railway Signalling System: Speed Controller

The context defines the constants of the system:

- Davis coefficients (a, b, c), traction power limits (f_{min}, f_{max})

Also, the context introduces the stopping distance function **StopDist** and controller models.

CONTEXT TrainCtx
CONSTANTS
free_move, restricted_move
StopDist
$a, b, c, f_{min}, f_{max}, f_{dec\_min}$
AXIOMS
$\mathrm{axm}_1: \quad a, b, c \in \mathbb{R}^+$
$\mathrm{axm}_2: \quad f_{min}, f_{max}, f_{dec\_min} \in \mathbb{R}$
$\mathrm{axm}_3: \quad \mathrm{StopDist} \in (\mathbb{R} \times \mathbb{R}^+) \rightarrowtail \mathbb{R}^+$
$\mathrm{axm}_5: \quad \mathrm{partition}(\mathrm{STATES}, \{\mathrm{free\_move}\}, \{\mathrm{restricted\_move}\})$
...

Cyber-Physical Railway Signalling System: Proof Statistic

MACHINE TrainMach **REFINES** Generic
VARIABLES t , x_{st} tp , tv , ta , f , EoA
INVARIANTS

inv$_1$: $tp, tv, ta \in \mathbb{R} \nrightarrow \mathbb{R}$
inv$_2$: $[0, t] \subseteq \mathrm{dom}(tp), \ldots$
inv$_3$: $\mathsf{EoA} \in \mathbb{R}^+$
inv$_4$: $f_{min} \leq f \wedge f \leq f_{max}$
inv$_5$: $x_p = [ta\ tv\ tp]^\top$
saf$_1$: $\forall t^* \cdot t^* \in [0, t] \Rightarrow tp(t^*) \leq \mathsf{EoA}$
phy$_1$: $\forall t^* \cdot t^* \in [0, t] \Rightarrow tv(t^*) \geq 0$

Safety property as: **at all times the train must remain within the issued movement authority**:

- expressed as Event-B invariant saf$_1$,
- an additional physics property phy$_1$.

Cyber-Physical Railway Signalling System: Speed Controller

Sense_to_restricted
REFINES Sense
WHERE
grd$_1$: $tp(t) + \mathrm{StopDist}(ta(t) \mapsto tv(t))) \geq \mathrm{EoA}$
WITH
st : $st = \{\mathrm{restricted\_move}\}$
$p : p = \mathrm{STATES} \times \mathbb{R} \times \{v^* \mapsto p^* \mid p^* + \mathrm{StopDist}(f_{dec\_min} \mapsto v^*) \geq \mathrm{EoA}\}$

THEN
act$_1$: $x_{st} := \mathrm{restricted\_move}$
END

Cyber-Physical Railway Signalling System: Speed Controller

Actuate_move **REFINES** Actuate

ANY t'

WHERE

grd$_1$: $tp(t) + \text{StopDist}(ta(t) \mapsto tv(t)) \leq \text{EoA}$

grd$_2$: $t < t'$

WITH

x_p' : $x_p' = [ta\ tv\ tp]^\top$

$\mathcal{P}$: $\mathcal{P} = \ldots$

H : $H = \ldots$

st : $st = \text{STATES}$

THEN

act$_1$: $ta, tv, tp :|_{t \to t'}$

$\textbf{solutionOf}([t, t'], [tv\ tp]^\top, \text{DavisEquation}(a, b, c, f, t, tv(t), tp(t))) \wedge$

$ta = \dot{tv}$

$\& tp + \text{StopDist}(ta \mapsto tv) \leq \text{EoA} \wedge tv \geq 0$

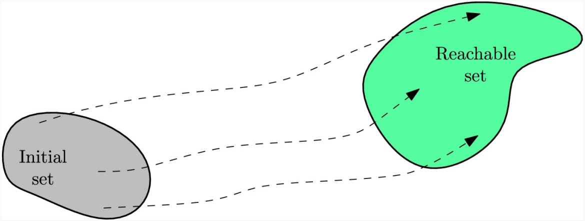
END

Cyber-Physical Railway Signalling System: Proof Statistic

Refinement	PO Type	|POs|	Auto.	Inter.
Speed Controller		55	36	19
	WD	12	12	0
	GRD	11	11	0
	INV	18	10	**8**
	FIS	8	0	**8**
	SIM	6	3	3
Communication		85	71	14
	WD	31	31	0
	GRD	12	7	5
	INV	42	33	9
	FIS	0	0	0
	SIM	0	0	0
Total		140	119	21

Reachability Analysis for Hybridised Event-B

Can **reachability analysis** help to address verification automation challenges of hybridised Event-B models (similar to how ProB model checker is used for discrete systems)?



Computing reachable states of a **hybrid automaton** requires computing *runs* of the hybrid system.

Cyber-Physical Railway Signalling System: Speed Controller

Reachability enabled verification **tactic** of **CINV**:

1. Strengthen actuation events actions such that $H \subseteq \mathcal{I}$,
2. Generating proof-obligation (automatically),
   - 2 CFIS proof obligations were generated (for the free and restricted modes).
3. Translate proof-obligations to reachability analysis tool (JuliaReach, manually),
   - translate other related functions - *StopDist*.
4. Define initial values $\mathcal{X}_0$ for the reachability problem,
5. Compute and check solution produced reachability tool,
   - check existence of an interval $[0, t']$ for which reachset $\mathcal{R}$ of continuous x_p with initial values $\mathcal{X}_0$ satisfies a strengthened local invariant $\mathcal{H}$.

Cyber-Physical Railway Signalling System: Proof Statistic

Refinement	PO Type	|POs|	Auto.	Inter.
Speed Controller		55	36 (48)	19 (7)
	WD	12	12	0
	GRD	11	11	0
	INV	18	10 (14)	8 (4)
	FIS	8	0 (8)	8 (0)
	SIM	6	3	3
Communication		85	71	14
	WD	31	31	0
	GRD	12	7	5
	INV	42	33	9
	FIS	0	0	0
	SIM	0	0	0
Total		140	119	21

To enable model animation and validation we aim to connect hybridised **Event-B** with Simulink/Stateflow.

To validate the speed controller model we (manually) translated it to Simulink/Stateflow.

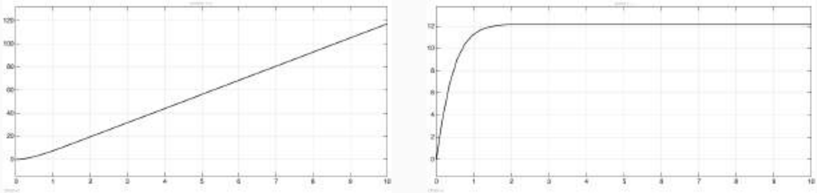


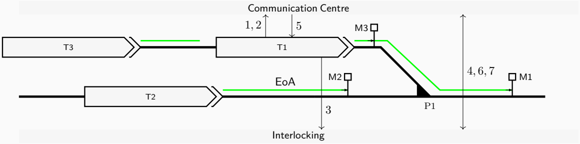
Figure 4: TGV train simulation with Davis equation coefficients for TGV: $a = 25$, $b = 1.188$ and $c = 0.0703728$

Cyber-Physical Railway Signalling System: Other Sub-Systems

2nd refinement introduces other sub-systems of the signalling system:

- communication centres, interlocking and infrastructure,
- communication protocol.

The **generic** railway signalling is based on ETCS Level 3 and CBTC systems.



Communication protocol was modelled by using developed **Event-B** communication modelling patterns.

To formally demonstrate that the generic signalling system issues **safe** movement authority and ensures safe point crossing.

Cyber-Physical Railway Signalling System: Proof Statistic

Refinement	PO Type	|POs|	Auto.	Inter.
Speed Controller		55	36 (48)	19 (7)
	WD	12	12	0
	GRD	11	11	0
	INV	18	10 (14)	8 (4)
	FIS	8	0 (8)	8 (0)
	SIM	6	3	3
Communication		85	71	14
	WD	31	31	0
	GRD	12	7	5
	INV	42	33	9
	FIS	0	0	0
	SIM	0	0	0
Total		140	119	21

Conclusions and Next Steps

In summary:

- The complexity of developing complex CPS can be reduced by using refinement and abstraction.
- Our proposed framework provides a more comprehensive formal CPS development.
- Reachability analysis can help to improve verification automation of hybridised Event-B models.

Next steps in the short-term:

- Facilitate an automatic translation of hybridised Event-B models to JuliaReach,
- develop new Event-B theories.

(Long-term) Future Work

Explore synergies between proof and reachability analysis for CPS system verification and code generation:

- proving single CINV/CFIS proof-obligations (still many open questions),
- proving CPS Event-B sub-models,
- discovering model constraints and safe parameter values,
- discretisation of continuous model and code generation (discovering t').

Acknowledgements

I would like to thank my colleagues:

Alexander Romanovsky (Newcastle University and The Formal Route)

Alexei Iliasov (The Formal Route)

Sergiy Bogomolov (Newcastle University)

Guillaume Dupont (INPT–ENSEEIHT)

Yamine Aït-Ameur (INPT–ENSEEIHT)

Neeraj Kumar Singh (INPT–ENSEEIHT)

Dominic Taylor and Milo Lloyd (Systra Scott Lister)

Cliff Jones (Newcastle University)

Simon Chadwick (Siemens Rail Automation)

Workshop on Formal Engineering of CPS

2nd International Workshop on Formal Engineering of Cyber-Physical Systems (FE-CPS) collocated with TASE 2023 (Bristol, UK), 4-6 July.

Website with CfP: https://www.irit.fr/FE-CPS-2023/

Invited talks: Ana Cavalcanti (University of York, UK) and Claudio Gomes (Aarhus University, Denmark)

References

1. J.-R. Abrial. *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, 2013. ISBN: 1139195883, 9781139195881.
2. J.-R. Abrial. *The B-book: Assigning Programs to Meanings*. New York, USA: Cambridge University Press, 1996. ISBN: 0-521-49619-5.
3. Y. Ait-Ameur et al. "A Refinement-Based Formal Development of Cyber-Physical Railway Signalling Systems". In: *Form. Asp. Comput.* 35.1 (Jan. 2023). ISSN: 0934-5043. DOI: 10.1145/3524052. URL: https://doi.org/10.1145/3524052.
4. R. Banach et al. "Core Hybrid Event-B I: Single Hybrid Event-B machines". In: *Science of Computer Programming* 105 (2015), pp. 92–123. ISSN: 0167-6423. DOI: https://doi.org/10.1016/j.scico.2015.02.003. URL: https://www.sciencedirect.com/science/article/pii/S0167642315000283.
5. G. Dupont et al. "Event-B Hybridation: A Proof and Refinement-Based Framework for Modelling Hybrid Systems". In: *ACM Trans. Embed. Comput. Syst.* 20.4 (May 2021). ISSN: 1539-9087. DOI: 10.1145/3448270. URL: https://doi.org/10.1145/3448270.
6. P. Stankaitis et al. "A Refinement Based Method for Developing Distributed Protocols". In: *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*. 2019, pp. 90–97. DOI: 10.1109/HASE.2019.00023.