A Hybrid Event-B Study of Lane Centering

Richard Banach[1] and Michael Butler[2]

[1]School of Computer Science, University of Manchester, UK
[2]Electronics and Computer Science, University of Southampton, UK

Contents

1. Hybrid and Cyber-Physical Systems

Nowadays, computing devices get ever smaller.

Nowadays, computing devices get ever more distributed and interconnected.

This enables ever easier/routine coupling of computing devices to the physical environment.

- **Hybrid Systems** — discrete + continuous behaviour.
- **Cyber-Physical Systems** — distributed embedded behaviour.

Requires an extension of the usual discrete transition frameworks for faithful modeling.

- Most existing work is automaton based.
- Most existing work focuses on **verification** rather than top-down development

1. Hybrid and Cyber-Physical Systems

Nowadays, computing devices get ever smaller.

Nowadays, computing devices get ever more distributed and interconnected.

This enables ever easier/routine coupling of computing devices to the physical environment.

- **Hybrid Systems** — discrete + continuous behaviour.
- **Cyber-Physical Systems** — distributed embedded behaviour.

Requires an extension of the usual discrete transition frameworks for faithful modeling.

- Most existing work is automaton based.
- Most existing work focuses on **verification** rather than top-down development ... enhance Event-B.

2. Discrete Event-B

Event-B is a simplification of the Classical B-Method that was one of the earliest 'full process' top-down development methodologies. A typical Event-B model has the following characteristics:

- static contexts
- commands – guards (no preconditions)
- commands – actions (deterministic, nondeterministic)
- invariants

Straightforward trace style semantics, policed by proof obligations.

- intended for industrial application

Example

MACHINE *Nodes*
SEES *NCtx*
VARIABLES *nod*
INVARIANTS
  $nod \in \mathbb{P}(NSet)$
EVENTS
  INITIALISATION
    STATUS ordinary
    BEGIN $nod := \emptyset$ END
  AddNode
    STATUS ordinary
    ANY n
    WHERE $n \in NSet - nod$
    THEN $nod := nod \cup \{n\}$
    END
END

CONTEXT *NCtx*
SETS *NSet*
CONSTANTS aa, bb, cc, dd
AXIOMS
  $NSet = \{aa, bb, cc, dd\}$
END

3. A Framework for Hybrid Systems

Integrating formal reasoning in discrete and continuous domains requires a suitable semantic framework, which:

- is expressive enough for continuous applications;
- defaults cleanly for discrete reasoning.

3. A Framework for Hybrid Systems

Integrating formal reasoning in discrete and continuous domains requires a suitable semantic framework, which:

- is expressive enough for continuous applications;
- defaults cleanly for discrete reasoning.

- Time is an interval $\mathcal{T}$ of the reals $\mathbb{R}$.
- There are **mode variables** (piecewise constant), and **pliant variables** (piecewise continuously varying).
- $\mathcal{T}$ partitions into a sequence of left-closed right-open intervals, $\langle [t_0 \ldots t_1), [t_1 \ldots t_2), \ldots \rangle$, such that (all) discontinuous changes take place at some boundary point t_i.

In an interval $[t_i \ldots t_{i+1})$, the mode variables will be constant, but the pliant variables will change continuously, subject to:

I Zeno: there is a constant δ_{Zeno}, such that for all i needed, $t_{i+1} - t_i \geq \delta_{\mathsf{Zeno}}$.

II Limits: for every variable x, and for every time $t \in \mathcal{T}$, the left limit $\lim_{\delta \to 0} x(t-\delta)$ written $\overrightarrow{x(t)}$ and right limit $\lim_{\delta \to 0} x(t+\delta)$, written $\overleftarrow{x(t)}$ (with $\delta > 0$) exist, and for every t, $x(t) = \overleftarrow{x(t)}$.

III Differentiability: The behaviour of every pliant variable x in the interval $[t_i \ldots t_{i+1})$ is given by the solution of a well posed initial value problem $\mathcal{D}\, xs = \phi(xs, t)$. "Well posed" means $\phi(xs, t)$ has uniformly bounded Lipschitz constants (w.r.t. xs), and $\phi(xs, t)$ is measurable in t.

There are **mode transitions** ((any) variable can change discontinuously), and **pliant transitions** (pliant variables can change continuously). We say that a set of rules is **well formed** iff:

- Every enabled mode transition is feasible, i.e. has an after-state, and on its completion enables a pliant transition (but does not enable any mode transition).
- Every enabled pliant transition is feasible, i.e. has a time-indexed family of after-states, and EITHER:
  - (i) During the run of the pliant transition a mode transition becomes enabled. It preempts the pliant transition. ORELSE
  - (ii) During the run of the pliant transition it becomes infeasible: finite termination. ORELSE
  - (iii) The pliant transition continues indefinitely: nontermination.

A mode transition establishes the initial state.

4. Extending Event-B for Hybrid Behaviour

First: keep the discrete transitions *as is*.

Sequence of states σ_i of standard (discrete, i.e. mode) E-B variables becomes a sequence of piecewise constant functions $[t_i \ldots t_{i+1}) \mapsto \sigma_i$

These stitch together to give a state function $\mathcal{T} \to TY$.

Usual E-B transition $\sigma_i \to \sigma_{i+1}$ are from/to:

Before-state: $\overrightarrow{\sigma_{i+1}}$

After-state: σ_{i+1}

4. Extending Event-B for Hybrid Behaviour

First: keep the discrete transitions *as is*.

Sequence of states σ_i of standard (discrete, i.e. mode) E-B variables becomes a sequence of piecewise constant functions $[t_i \ldots t_{i+1}) \mapsto \sigma_i$

These stitch together to give a state function $\mathcal{T} \to TY$.

Usual E-B transition $\sigma_i \to \sigma_{i+1}$ are from/to:

Before-state: $\overrightarrow{\sigma_{i+1}}$

After-state: σ_{i+1}

Use the same semantics for discrete transitions of pliant variables.

Second: allow pliant variables to change according to Carathéodory semantics of ordinary differential equations.

SOLVE $\mathcal{D}\, x = \phi$

(This allows discontinuities in RHS of the ODE, while ensuring absolute continuity of solutions, and (pointwise) validity of the ODE almost everywhere.)

Allow initial value and guard conditions (on before-states (only)) to control enabledness.

General theory ensures the existence a (t-dependent) family of transitions $Q(t_i, t)$ (... for $t \in (t_i \ldots \tilde{t})$), where $\tilde{t} > t_i$, and Q relates $\sigma_i = \sigma(t_i)$ at t_i to $\sigma(t)$ at t.

Preemption/feasibility defines t_{i+1} obeying $t_i < t_{i+1} \leq \tilde{t}$.

Carathéodory formulation underpins other forms of expression for pliant events.

SOLVE $y := E$

Direct assignment to E: semantics via $\mathcal{D}\, y := \mathcal{D}\, E$

Allow additional conditions during the evolution to influence feasibility.

COMPLY *BDApred*

Specifies the family of absolutely continuous behaviours satisfying *BDApred*.

Preemption/feasibility defines t_{i+1} obeying $t_i < t_{i+1} \leq \tilde{t}$.

Formal Semantics (Sketch)

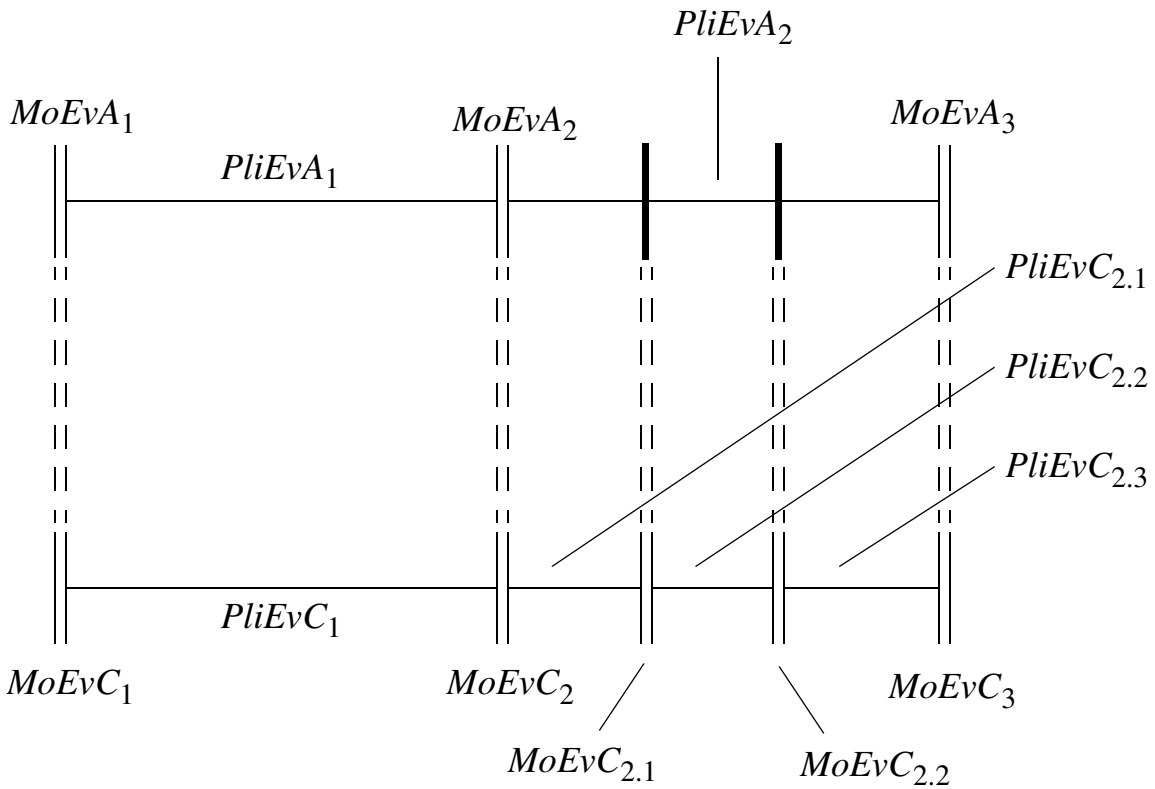
[1] Initialise. (Mode event.) $i := 0$
[2a] CHOOSE an enabled pliant event from each machine that has one. (Consistency.) or else
[2b] CHOOSE a pliant continuation for each machine that has one. (Consistency.) or else
[2b] ABORT if any pliant variable unspecified.
[3] FIND maximal mutually consistent solution on $[t_i \ldots t_{\text{NEW}})$.
[4] FIND earliest mode event preemption point in $(t_i \ldots t_{\text{NEW}})$, if there is one. (If not, finite or infinite termination).
[5] IMPLEMENT mode event preemption; i++; discard solution in $(t_i \ldots t_{\text{NEW}})$.
[6] GOTO **[2]**.

Semantics is a set of behaviours over $[t_0 \ldots t_{\text{FINAL}})$, or VOID.

Refinement

Fundamental Principle:

- In Hybrid Event-B, time moves at the same rate in all models of a refinement chain. Gives tight abstract/concrete coupling.



5. Schematic Syntax

Mode events ... nothing special:

MoEv
ANY $i?, l, o!$
WHERE $grd(\overrightarrow{u}, i?, l)$
THEN $u, o! :| BApred(\overrightarrow{u}, i?, l, \overleftarrow{u'}, o!)$
END

The overarrows constitute semantic decoration.

A full machine:

MACHINE *HyEvBMch*
TIME t
CLOCK *clk*
PLIANT x, y
VARIABLES u
INVARIANTS
$x \in \mathbb{R}$
$y \in \mathbb{R}$
$u \in \mathbb{N}$
EVENTS
INITIALISATION
STATUS ordinary
WHEN
$t = 0$
THEN
$clk := 1$
$x := x_0$
$y := y_0$
$u := u_0$
END
... ...

... ...
MoEv
STATUS ordinary
ANY $i?, l, o!$
WHERE $grd(x, y, u, i?, l, t, clk)$
THEN
$x, y, u, o!, clk :| BApred($
$x, y, u, i?, l, t, clk, x', y', u', o!, clk')$
END
PliEv
STATUS pliant
INIT $iv(x, y, u, t, clk)$
WHERE $grd(u)$
ANY $i?, l, o!$
COMPLY $BDApred(x, y, u, i?, l, o!, t, clk)$
SOLVE $\mathcal{D}\, x = \phi(x, y, u, i?, l, t, clk)$
$y, o! := E(x, u, i?, l, t, clk)$
END
END

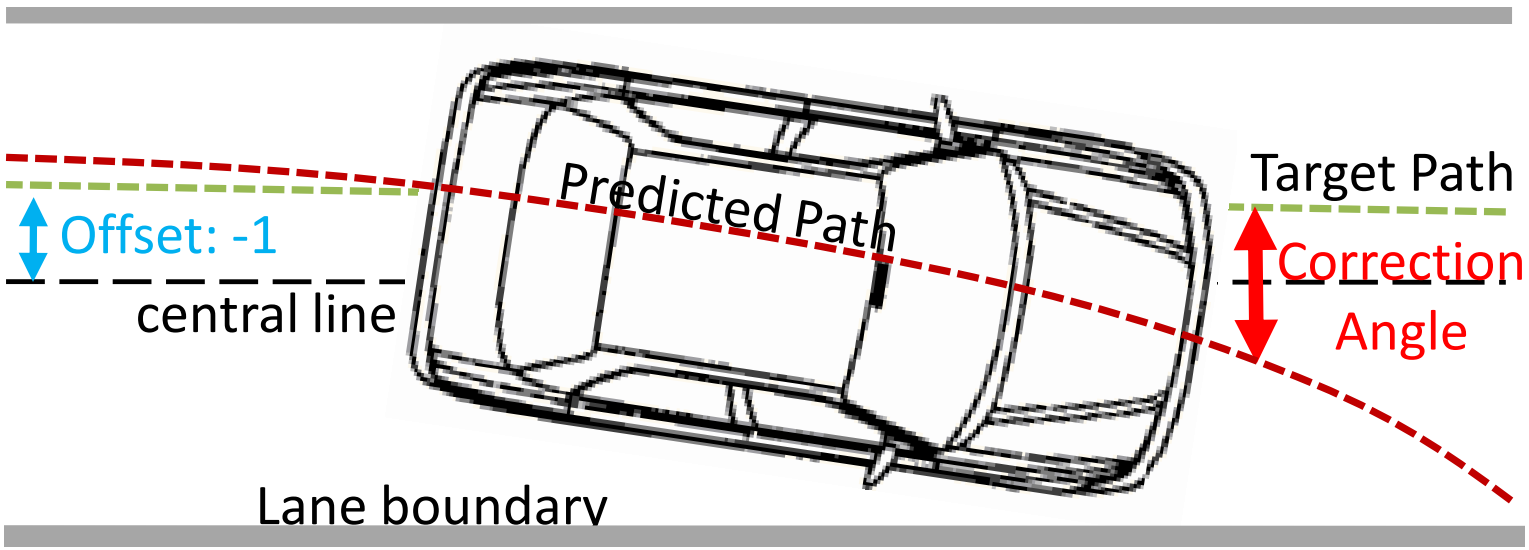
6. Proof Obligations

Like discrete Event-B, Hybrid Event-B semantics are enforced via proof obligations.

- Initialisation.
- Event feasibility: mode/pliant.
- Event invariant preservation: mode/pliant.
- Well-formedness: mode$\rightarrow$pliant/pliant$\rightarrow$mode.
- Refinement, guard strengthening: mode/pliant.
- Refinement, invariant preservation: mode/pliant.
- Relative deadlock freedom: mode/pliant.
- Etc.

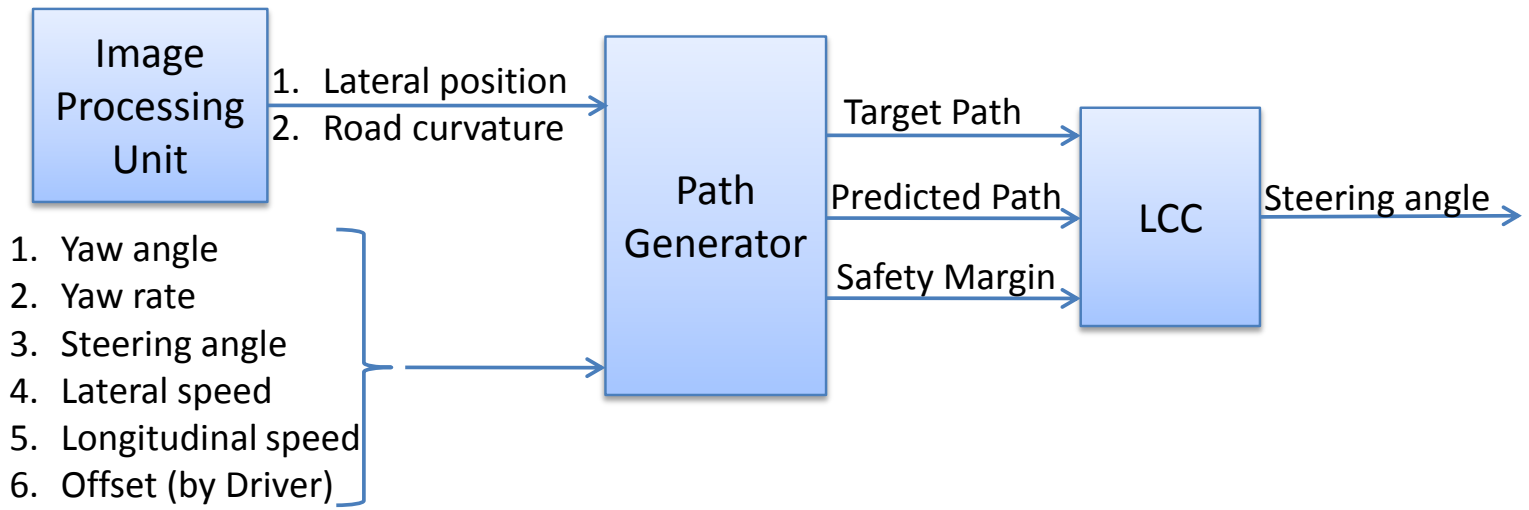
7. Lane Centering

Overview.



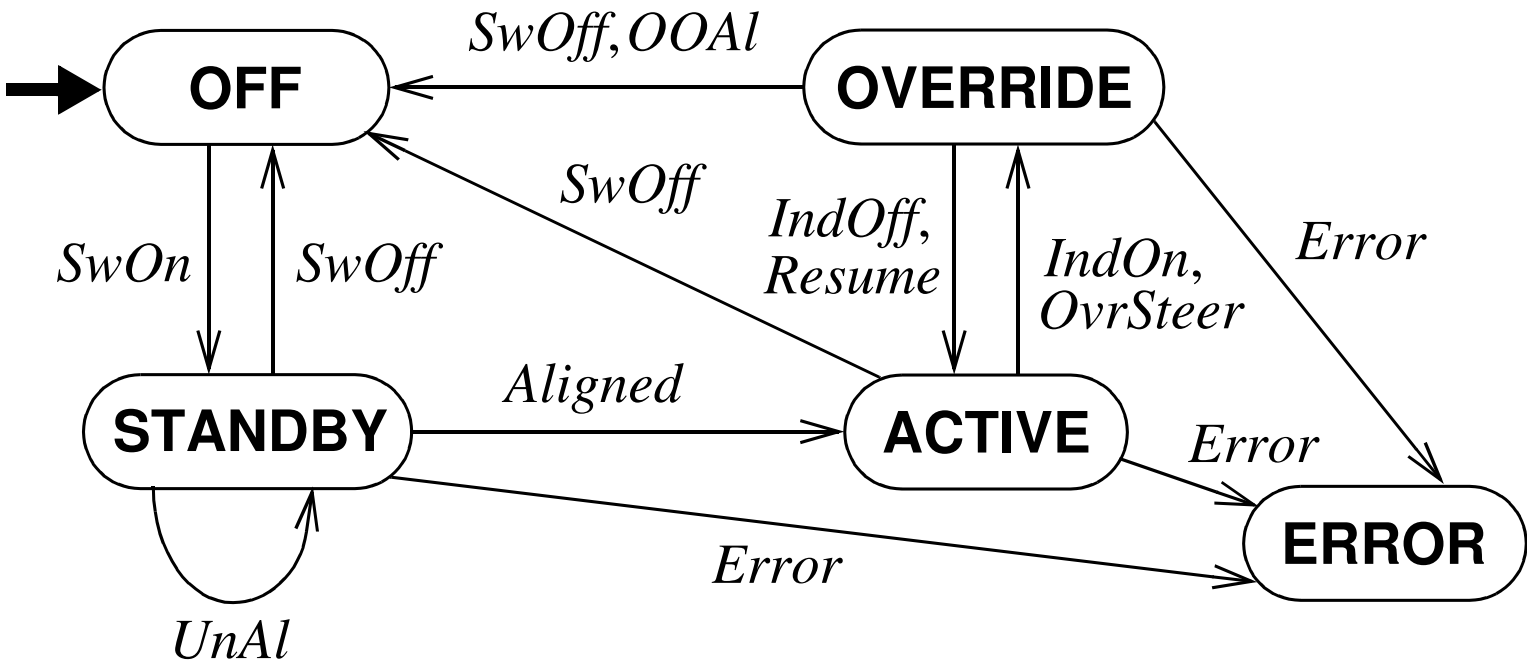
7. Lane Centering

Overview.



7. Lane Centering

State transition diagram.



MACHINE LCC_0
VARIABLES $mode$
INVARIANTS
$mode \in \{OFF, STANDBY, ACTIVE, OVERRIDE, ERROR\}$
EVENTS
INITIALISATION
STATUS ordinary
BEGIN
$mode := OFF$
END
SwOn
STATUS ordinary
ANY $in?$
WHERE $in? = swOn \wedge mode = OFF$
THEN $mode := STANDBY$
END
. . . . . .
UnAI
STATUS ordinary
ANY $in?$
WHERE $in? = tryAct \wedge$
$mode = STANDBY$
THEN skip
END
Aligned
STATUS ordinary
ANY $in?$
WHERE $in? = tryAct \wedge$
$mode = STANDBY$
THEN $mode := ACTIVE$
END
. . . . . .
OvrSteer
STATUS ordinary
ANY $in?$
WHERE $in? = ovrSteer \wedge$
$mode = ACTIVE$
THEN $mode := OVERRIDE$
END
Resume
STATUS ordinary
ANY $in?$
WHERE $in? = resume \wedge$
$mode = OVERRIDE$
THEN $mode := ACTIVE$
END
. . . . . .
Error
STATUS ordinary
ANY $in?$
WHERE $in? = error \wedge mode \in$
$\{STANDBY, ACTIVE, OVERRIDE\}$
THEN $mode := ERROR$
END
PliTrue
STATUS pliant
COMPLY *INVARIANTS*
END
END

INTERFACE *LCC_PG_IF*
PLIANT trq, θ_T, d
INVARIANTS
$trq \in \mathbb{R} \wedge |trq| \leq MAX_{trq}$
$\theta_T \in \mathbb{R} \wedge |\theta_T| \leq MAX_\theta$
$d \in \mathbb{R} \wedge |d| \leq MAX_d$
INITIALISATION
$trq \in [-MAX_{trq} \ldots MAX_{trq}]$
$\theta_T := 0$
$d := 0$
END

MACHINE *LCC_1*
REFINES *LCC_0*
CONNECTS *LCC_PG_IF*
VARIABLES *mode*
PLIANT θ
INVARIANTS
$mode \in \{OFF, STANDBY, ACTIVE, OVERRIDE, ERROR\}$
$\theta \in \mathbb{R} \wedge |\theta| \leq MAX_\theta$
EVENTS
INITIALISATION
. . . .
PliDefault
STATUS pliant
REFINES *PliTrue*
WHEN $mode \neq ACTIVE$
COMPLY *INVARIANTS*
END
SwOn
. . . .
SwOff
. . . .

UnAl
STATUS ordinary
REFINES *UnAl*
ANY $in?, out!$
WHERE $in? = tryAct \wedge$
$mode = STANDBY \wedge$
$\neg(|d| < \Delta_d \wedge |\theta - \theta_T| < \Delta_\theta)$
THEN $out! := BEEP$
END
Aligned
STATUS ordinary
ANY $in?$
WHERE $in? = tryAct \wedge$
$mode = STANDBY \wedge$
$(|d| < \Delta_d \wedge |\theta - \theta_T| < \Delta_\theta)$
THEN $mode := ACTIVE$
END
LCC_Active
STATUS pliant
REFINES *PliTrue*
WHEN $mode = ACTIVE$
SOLVE $\mathcal{D}\theta = -C(\theta - \theta_T) - Kd$
END
SwOff_Emrg
STATUS ordinary
REFINES *SwOff*
ANY $out!$
WHEN $mode = ACTIVE \wedge$
$\neg(|d| < \Delta_d \wedge |\theta - \theta_T| < \Delta_\theta)$
THEN $mode := OFF$
$out! := BEEP$
END
. . . .
. . . .

8. Conclusions

Hybrid Event-B gives the capability of addressing continuous concerns in an honest manner ... e.g. closed-loop control.

In future:

- Reasoning framework(s).
- RODIN enhancement.
- Etc.