A Refinement Based Approach to Hybrid Systems: Hybrid Event-B

Richard Banach

School of Computer Science, University of Manchester, UK

Contents

1. Discrete Event-B Basics

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

2. 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. Proof Obligations

Event-B machines are defined to be consistent when the POs are provable.

- initialisation feasibility
  $\exists u' \bullet Init_A(u')$
- invariant establishment
  $Init_A(u') \Rightarrow I(u')$
- event feasibility
  $I(u) \wedge grd_{MoEvA}(u, i) \Rightarrow (\exists u' \bullet BApred_{MoEvA}(u, i, u'))$
- invariant preservation
  $I(u) \wedge grd_{MoEvA}(u, i) \wedge BApred_{MoEvA}(u, i, u') \Rightarrow I(u')$

4. Refinement in Event-B

Top-down development in Event-B is achieved via refinement.

- add detail
- restrict nondeterminism
- new events, convergence
- nontrivial retrieve relations via joint invariants

Refinement notion policed by proof obligations.

5. Example, ctd.

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

MACHINE *Edges*
REFINES *Nodes*
SEES *NCtx*
VARIABLES *nod*, *edg*
INVARIANTS
  $nod \in \mathbb{P}(NSet)$
  $edg \in \mathbb{P}(NSet \times NSet)$
  $edg \subseteq nod \times nod$
EVENTS
  INITIALISATION
    STATUS ordinary
    BEGIN $nod := \emptyset$ END
  AddNode
    STATUS ordinary
    REFINES *AddNode*
    ANY n
    WHERE $n \in NSet - nod$
    THEN $nod := nod \cup \{n\}$
    END
  AddEdge
    STATUS convergent
    ANY n, m
    WHERE $\{n, m\} \subseteq nod$
      $n \mapsto m \in NSet \times NSet - edg$
    THEN $edg := edg \cup \{n \mapsto m\}$
    END
VARIANT $\mathrm{card}(NSet \times NSet - edg)$
END

6. Proof Obligations, ctd.

Event-B refinements are defined to be consistent when the POs are provable.

- initialisation feasibility
  $\exists w' \bullet Init_C(w')$
- initialisation relative consistency
  $Init_C(w') \Rightarrow (\exists u' \bullet Init_A(u') \wedge K(u', w'))$
- relative event feasibility
  $\exists u \bullet K(u, w) \wedge grd_{MoEvC}(w, k) \Rightarrow (\exists w' \bullet BApred_{MoEvC}(w, k, w'))$
- guard strengthening
  $I(u) \wedge K(u, w) \wedge grd_{MoEvC}(w, k) \Rightarrow (\exists i \bullet grd_{MoEvA}(u, i))$

6. Proof Obligations, ctd. ...

- joint invariant preservation
  $$I(u) \wedge K(u,w) \wedge grd_{MoEvC}(w,k) \wedge BApred_{MoEvC}(w,k,w') \\ \Rightarrow (\exists i, u' \bullet BApred_{MoEvA}(u,i,u') \wedge K(u',w'))$$

- new events, joint invariant preservation: 'new events refine skip'
  $$I(u) \wedge K(u,w) \wedge grd_{MoEvC}(w,k) \wedge BApred_{MoEvC}(w,k,w') \\ \Rightarrow K(u,w')$$

- new events, convergence
  $$BApred_{NewEvC}(w,k,w') \Rightarrow V(w') < V(w)$$

- old and new events, relative deadlock freedom (using witness)
  $$I(u) \wedge K(u,w) \wedge (\exists u', w' \bullet W(i,k,u,u',w,w')) \wedge \\ [\ grd_{MoEvA1}(u,i) \vee grd_{MoEvA2}(u,i) \vee \ldots \vee grd_{MoEvAN}(u,i)\ ] \\ \Rightarrow grd_{MoEvC1}(w,k) \vee grd_{MoEvC2}(w,k) \vee \ldots \vee grd_{MoEvCM}(w,k)$$

7. Principles for Hybrid Event-B

Discrete Event-B has no time. Need to incorporate time.

- In Hybrid Event-B, time is $\mathbb{R}^+$ say, read-only.

Discrete Event-B has no continuous behaviour. Need to incorporate this.

- In Hybrid Event-B, distinguish between mode events and pliant events.
- Demand that in Hybrid Event-B, pliant transitions interleave mode transitions of discrete Event-B. Preemption semantics.
- Demand usual differentiability, Lipschitz, measurability properties of pliant events.
- Demand usual Zeno, càdlàg properties of pliant transitions.

7. Principles for Hybrid Event-B ...

Mode event decorated with semantic interpretation:

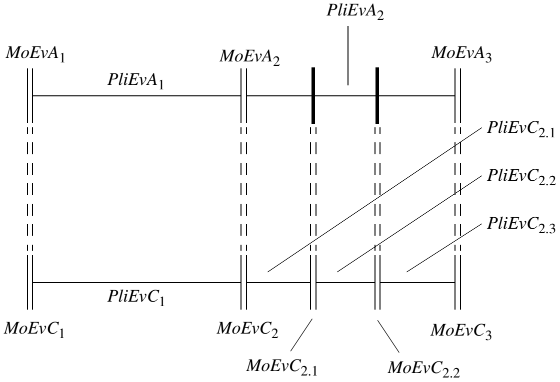
MoEv ANY $\overrightarrow{i}$ WHERE $grd(\overrightarrow{u}, \overrightarrow{i})$ THEN $u := E(\overrightarrow{u}, \overrightarrow{i})$ END	*MoEv* ANY $\overrightarrow{i}$ WHERE $grd(\overrightarrow{u}, \overrightarrow{i})$ THEN $u :\mid BApred(\overrightarrow{u}, \overrightarrow{i}, \overleftarrow{u'})$ END

Left limits for before-values, right limits for after-values.

7. Principles for Hybrid Event-B ... ...

Refinement.

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



8. 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] CHOOSE a constant behaviour for each remaining variable.

[3] FIND maximal mutually consistent solution on $[t_i \ldots t_{\mathrm{NEW}})$.

[4] FIND earliest mode event preemption point in $(t_i \ldots t_{\mathrm{NEW}})$, if there is one. (If not, finite or infinite termination).

[5] IMPLEMENT mode event preemption; $i{+}{+}$; discard solution in $(t_i \ldots t_{\mathrm{NEW}})$.

[6] GOTO **[2]**.

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

9. Examples – 1

```
MACHINE HyEvBMch
TIME t
CLOCK clk
PLIANT x
VARIABLES u
INVARIANTS
  x ∈ ℝ
  u ∈ . . .
EVENTS
  INITIALISATION
    STATUS ordinary
    WHEN t = 0
    THEN clk := 1
         u := u0
         x := x0
    END
... ...
```

```
... ...
    PliEvDE
      STATUS pliant
      INIT iv(x)
      WHEN grd(u)
      ANY i
      WHERE BDApred(x, i, t)
      SOLVE 𝒟 x = φ(x, i, t)
      END
    PliEvNA
      STATUS pliant
      INIT iv(x)
      WHEN grd(u)
      ANY i
      THEN x :| BDApred(x, i, t)
      END
  END
```

9. Examples ... – 2

```
MACHINE  ExUp

TIME  t
CLOCK  clk
PLIANT  x
VARIABLES  md
INVARIANTS
  md ∈ {stat, dyn}
  t ∈ [0 ... ∞)
  x ∈ [0 ... 10]

EVENTS
  INITIALISATION
    STATUS  ordinary

    WHEN  t = 0
    THEN  md := dyn
          x := 0
          clk := 1
    END
  IncPLi
    STATUS  pliant

    WHEN  md = dyn
    SOLVE  𝒟 x = 1
    END
... ...
```

```
... ...
                    IncD
                      WHEN  t ∈ ℕ ∧
                            t ∈ {1 ... 9}
                      THEN  skip
                      END
  Stop
    STATUS  ordinary

    WHEN  t = 10
    THEN  md := stat

    END
  FINAL
    STATUS  pliant final

    WHEN  clk = 11
    THEN  skip
    END
END
```

9. Examples ... – 2

```
MACHINE ExUpR
REFINES ExUp
TIME t
CLOCK clk
PLIANT w
VARIABLES md
INVARIANTS
  md ∈ {stat, dyn}
  t ∈ [0 ... ∞)
  w ∈ [0 ... 10]
  w = ⌊x⌋
EVENTS
  INITIALISATION
    STATUS ordinary
    REFINES INITIALISATION
    WHEN t = 0
    THEN md := dyn
         w := 0
         clk := 1
    END
  IncPLi
    STATUS pliant
    REFINES IncPLi
    WHEN md = dyn
    THEN skip
    END
... ...
```

```
... ...
  IncD
    WHEN t ∈ ℕ ∧
         t ∈ {1 ... 9}
    THEN w := w + 1
    END
  Stop
    STATUS ordinary
    REFINES Stop
    WHEN t = 10
    THEN md := stat
         w := w + 1
      END
  FINAL
    STATUS pliant final
    REFINES FINAL
    WHEN clk = 11
    THEN skip
    END
END
```

9. Examples ... ... – 3

MACHINE *ExUpQuadR*
REFINES *ExUpQuad*
TIME t
PLIANT x
VARIABLES *md*
INVARIANTS
$md \in \{stat, dyn\}$
$t \in [0 \ldots \infty)$
$x \in [0 \ldots 9]$

EVENTS
INITIALISATION
STATUS ordinary
REFINES *INITIALISATION*
WHEN $t = 0$
THEN $md := dyn$
$x := 0$
END
IncPLi
STATUS pliant
REFINES *IncPLi*
WHEN $md = dyn$
SOLVE $\mathcal{D}\, x = 2\, t$
END
... ...

... ...
IncD
STATUS ordinary

WHEN $t \in \mathbb{N} \wedge$
$t \in \{1 \ldots 2\}$
THEN skip
END

Stop
STATUS ordinary
REFINES *Stop*
WHEN $t = 3$
THEN $md := stat$
END

FINAL
STATUS pliant final

WHEN $t = 3$
THEN skip
END
END

9. Examples … … – 3

MACHINE *ExUpQuadRRet*
RETRENCHES *ExUpQuadR*
TIME t
PLIANT w
VARIABLES *md*
INVARIANTS
  $md \in \{stat, dyn\}$
  $t \in [0 \ldots \infty)$
  $w \in [0 \ldots 9]$
  $x \in \{0, 9\} \Rightarrow x = w$
EVENTS
  INITIALISATION
    STATUS ordinary
    REFINES *INITIALISATION*
    WHEN $t = 0$
    THEN $md := dyn$
      $w := 0$
    END
  IncPLi
    STATUS pliant
    RETRENCHES *IncPLi*
    WHEN $md = dyn$
    THEN skip
    OUT $\sup_{t \in (\mathfrak{t}_L \ldots \mathfrak{t}_R)} |x(t) - w(t)|$
      $\leq 2\,\mathfrak{t}_R + 1$
    END
… …

… …
  IncD
    STATUS ordinary
    RETRENCHES *IncD*
    WHEN $t \in \mathbb{N} \wedge$
      $t \in \{1 \ldots 2\}$
    THEN $w := w + 2\,t + 1$
    OUT $x' = w' \wedge$
      $x - w = 2\,t + 1$
    END
  Stop
    STATUS ordinary
    RETRENCHES *Stop*
    WHEN $t = 3$
    THEN $md := stat$
      $w := w + 2\,t + 1$
    OUT $x' = w' \wedge$
      $x - w = 2\,t + 1$
    END
  FINAL
    STATUS pliant final
    REFINES *FINAL*
    WHEN $t = 3$
    THEN skip
    END
END

10. More Proof Obligations

Hybrid Event-B is highly structured. Lots of new POs ...

- pliant event feasibility

  $I(u(\mathbb{t}_{\rm L})) \wedge iv_{PliEvA}(u(\mathbb{t}_{\rm L})) \wedge grd_{PliEvA}(u(\mathbb{t}_{\rm L}))$
  $\quad \Rightarrow (\exists \mathbb{t}_{\rm R} > \mathbb{t}_{\rm L} \bullet (\forall \mathbb{t}_{\rm L} < t < \mathbb{t}_{\rm R}, i(t) \bullet$
  $\quad\quad (\exists u(t) \bullet BDApred_{PliEvA}(u(t), i(t), t) \Rightarrow PliEvA(u(t), i(t), t))))$

- pliant event invariant preservation

  $I(u(\mathbb{t}_{\rm L})) \wedge iv_{PliEvA}(u(\mathbb{t}_{\rm L})) \wedge grd_{PliEvA}(u(\mathbb{t}_{\rm L})) \wedge$
  $(\forall \mathbb{t}_{\rm L} < t < \mathbb{t}_{\rm R} \bullet BDApred_{PliEvA}(u(t), i(t), t) \wedge PliEvA(u(t), i(t), t) \Rightarrow I(u(t))))$

10. More Proof Obligations ...

- well-formedness: mode disables mode, enables pliant

$\exists u_0, i_0 \bullet BApred_{MoEv}(u_0, i_0, u) \wedge I(u)$
$\Rightarrow \neg[\ \exists i \bullet grd_{MoEv1}(u, i) \vee grd_{MoEv2}(u, i) \ldots grd_{MoEvN}(u, i)\ ] \wedge$
$[\ (iv_{PliEv1}(u) \wedge grd_{PliEv1}(u)) \vee (iv_{PliEv2}(u) \wedge grd_{PliEv2}(u)) \vee \ldots \vee$
$(iv_{PliEvM}(u) \wedge grd_{PliEvM}(u))\ ]$

- well-formedness: **nonfinal** pliant enables mode

$I(u(\mathbb{t}_{\mathrm{L}})) \wedge grd_{PliEv}(u(\mathbb{t}_{\mathrm{L}})) \wedge$
$(\forall \mathbb{t}_{\mathrm{L}} < t < \mathbb{t}_{\mathrm{R}} \bullet BDApred_{PliEv}(u(t), i(t), t) \wedge PliEv(u(t), i(t), t))$
$\Rightarrow \neg[\ \exists i, \mathbb{t}_{\mathrm{L}} < \tilde{t} < \mathbb{t}_{\mathrm{R}} \bullet$
$grd_{MoEv1}(u(\tilde{t}), i) \vee grd_{MoEv2}(u(\tilde{t}), i) \vee \ldots \vee grd_{MoEvN}(u(\tilde{t}), i)\ ] \wedge$
$[\ \exists i \bullet grd_{MoEv1}(\overrightarrow{u(\mathbb{t}_{\mathrm{R}})}, i) \vee grd_{MoEv2}(\overrightarrow{u(\mathbb{t}_{\mathrm{R}})}, i) \vee \ldots \vee grd_{MoEvN}(\overrightarrow{u(\mathbb{t}_{\mathrm{R}})}, i)\ ]$

10. More Proof Obligations ... ...

POs for refinement.

- relative event feasibility

$$(\exists u(\mathbb{t}_{\mathrm{L}}) \bullet I(u(\mathbb{t}_{\mathrm{L}})) \wedge K(u(\mathbb{t}_{\mathrm{L}}), w(\mathbb{t}_{\mathrm{L}})) \wedge iv_{PliEvC}(w(\mathbb{t}_{\mathrm{L}})) \wedge grd_{PliEvC}(w(\mathbb{t}_{\mathrm{L}}))$$
$$\Rightarrow (\exists \mathbb{t}_{\mathrm{R}} > \mathbb{t}_{\mathrm{L}} \bullet (\forall \mathbb{t}_{\mathrm{L}} < t < \mathbb{t}_{\mathrm{R}}, k(t) \bullet$$
$$(\exists w(t) \bullet BDApred_{PliEvC}(w(t), k(t), t) \Rightarrow PliEvC(w(t), k(t), t)))))$$

- guard strengthening

$$I(u(\mathbb{t}_{\mathrm{L}})) \wedge K(u(\mathbb{t}_{\mathrm{L}}), w(\mathbb{t}_{\mathrm{L}})) \wedge iv_{PliEv_C}(w(\mathbb{t}_{\mathrm{L}})) \wedge grd_{PliEv_C}(w(\mathbb{t}_{\mathrm{L}}))$$
$$\Rightarrow \left[\ iv_{PliEv_A}(u(\mathbb{t}_{\mathrm{L}})) \wedge \right] grd_{PliEv_A}(u(\mathbb{t}_{\mathrm{L}}))$$

10. More Proof Obligations ... ...

- joint invariant preservation

  $I(u(\mathbb{t}_{\mathrm{L}})) \wedge K(u(\mathbb{t}_{\mathrm{L}}), w(\mathbb{t}_{\mathrm{L}})) \wedge iv_{PliEv_C}(w(\mathbb{t}_{\mathrm{L}})) \wedge grd_{PliEv_C}(w(\mathbb{t}_{\mathrm{L}})) \wedge$
  $\big(\forall \mathbb{t}_{\mathrm{L}} < t < \mathbb{t}_{\mathrm{R}} \bullet BDApred_{PliEv_C}(w(t), k(t), t) \wedge PliEv_C(w(t), k(t), t)$
  $\Rightarrow (\exists u(t), i(t) \bullet BDApred_{PliEv_A}(u(t), i(t), t) \wedge PliEv_A(u(t), i(t), t) \wedge K(u(t), w(t)))\big)$

- old and new pliant events, relative deadlock freedom

  $[\ grd_{PliEv1}(u(\mathbb{t}_{\mathrm{L}})) \vee grd_{PliEv2}(u(\mathbb{t}_{\mathrm{L}})) \vee \ldots \vee grd_{PliEvM}(u(\mathbb{t}_{\mathrm{L}}))\ ] \wedge$
  $I(u) \wedge K(u(\mathbb{t}_{\mathrm{L}}), w(\mathbb{t}_{\mathrm{L}}))$
  $\Rightarrow [\ grd_{PliEv1}(w(\mathbb{t}_{\mathrm{L}})) \vee grd_{PliEv2}(w(\mathbb{t}_{\mathrm{L}})) \vee \ldots \vee grd_{PliEvN}(w(\mathbb{t}_{\mathrm{L}}))\ ]$

10. More Proof Obligations ... ...

POs for retrenchment.

- mode events

$$I(u) \wedge K(u,w) \wedge grd_{MoEvC}(w,k) \wedge BApred_{MoEvC}(w,k,w')$$
$$\Rightarrow (\exists i, u' \bullet BApred_{MoEvA}(u,i,u') \wedge ((K(u',w') \wedge out(u',w',i,u,k,w)) \vee conc(u',w',i,u,k,w)))$$

- pliant events

$$I(u(\mathbb{t}_\mathrm{L})) \wedge K(u(\mathbb{t}_\mathrm{L}),w(\mathbb{t}_\mathrm{L})) \wedge iv_{PliEv_C}(w(\mathbb{t}_\mathrm{L})) \wedge grd_{PliEv_C}(w(\mathbb{t}_\mathrm{L})) \wedge$$
$$\big(\forall \mathbb{t}_\mathrm{L} < t < \mathbb{t}_\mathrm{R} \bullet BDApred_{PliEv_C}(w(t),k(t),t) \wedge PliEv_C(w(t),k(t),t)$$
$$\Rightarrow (\exists u(t), i(t) \bullet BDApred_{PliEv_A}(u(t),i(t),t) \wedge PliEv_A(u(t),i(t),t) \wedge ((K(u(t),w(t)) \wedge out(u(t),w(t),i(t),k(t))) \vee conc(u(t),w(t),i(t),k(t))))\big)$$

11. Conclusions

With a little thought, hybrid ideas fit neatly into Event-B.

BBQ-CPS Project(-to-be?) will:

11. Conclusions

With a little thought, hybrid ideas fit neatly into Event-B.

BBQ-CPS Project(-to-be?) will:

- explore application scenarios
- investigate relevant theoretical properties
- investigate relevant reasoning frameworks
- build these ideas into the Rodin tool