Landing Gear System in Hybrid Event-B — Nominal Regime: Machines and Interfaces I

Richard Banach

Received: by the Almighty / Accepted: in good faith

Abstract A digest of the machines of the Landing Gear Case Study.

Keywords Landing Gear Case Study · Hybrid Event-B

1 Guiding Principle

The most important principle is that each ‘autonomous’ part of the system, i.e. each portion capable of behaving or reacting on its own, is housed in its own machine. This is to allow each part to have its own pliant behaviour, even though in this case study, there are more or less no continuous properties mentioned in the requirements [1].

Besides that, the approach is top-down, at each stage modelling only those parts of the system that are visible at that level.

2 Notational Conventions

A brief summary of the main notational conventions used in the case study.

For ease of orientation, each figure of the development includes a summary of the constituent machines and interfaces at the current level.

Level $n_\ldots$ is used as a prefix for names of machines and interfaces. Once introduced at level $n$, a machine or interface stays in the development at successive levels until refined or decomposed further.

Keywords REFINES, DECOMPOSES, REFINESandDECOMPOSES indicate that a machine has been refined from an earlier level predecessor, or decomposed from an earlier level predecessor, or that both decomposition and refinement are involved in the development step from the earlier level predecessor. The same considerations apply to individual events.

‘$X$’ as a suffix to an input parameter means that it is a stimulus received from the external environment, and is not synchronised with any corresponding output in the whole system model.

‘$s$’ as a suffix to an event name means that it is a synchronised event, which is synchronised with an event of the same name in another machine, using an input parameter in one machine and an output parameter in the other.

$\text{clk}_\ldots$ is the name of a clock.

$s\text{ens}_\ldots$ is the name of a sensor.

3 The Actual Machines ... are below

References

MACHINE Level0_PilotAndLightsNominal
VARIABLES
  handle,
  green, orange
INVARIANTS
  handle ∈ {UP, DOWN}
  green, orange ∈ {ON, OFF}
EVENTS
  INITIALISATION
    STATUS ordinary
    BEGIN
      handle := DOWN
      green, orange := ON, OFF
    END
  PliTrue
    STATUS pliant
    COMPLY INVARIANTS
    END
  PilotGearUP
    STATUS ordinary
    ANY in?
    WHERE in? = pilotGearUP_X ∧
      handle = DOWN
    THEN
      handle := UP
    END
  PilotGearDOWN
    STATUS ordinary
    ANY in?
    WHERE in? = pilotGearDOWN_X ∧
      handle = UP
    THEN
      handle := DOWN
    END
... ...

Fig. 1 Level 0: Nominal top level mode machine for the landing gear system. System consists of:
MACHINE Level1_TopLevelNominal
REFINES Level0_PilotAndLights
VARIABLES
  handle, green, orange, gearsMoving, gearsLocked
INVARINTS
  handle ∈ {UP, DOWN}
  green, orange ∈ {ON, OFF}
  gearsMoving ∈ BOOL
  gearsLocked ∈ BOOL
  gearsMoving ↔ orange = ON
  gearsLocked ↔ green = ON
EVENTS
INITIALISATION
  STATUS ordinary
  REFINES INITIALISATION
  BEGIN
    handle := DOWN
    green, orange := ON, OFF
    gearsMoving := FALSE
    gearsLocked := TRUE
  END
PliTrue
  STATUS pliant
  REFINES PliTrue
  COMPLY INVARINTS
END
PilotGearUP
  STATUS ordinary
  REFINES PilotGearUP
  ANY in?
  WHERE in? = pilotGearUP_X ∧
    handle := DOWN
  THEN
    handle := UP
END
PilotGearDOWN
  STATUS ordinary
  REFINES PilotGearDOWN
  ANY in?
  WHERE in? = pilotGearDOWN_X ∧
    handle = UP
  THEN
    handle := DOWN
END
GearStartMoving
  STATUS ordinary
  REFINES GearStartMoving
  ANY in?
  WHERE in? = gearStartMoving_X ∧ ¬gearsMoving
  THEN
    orange := ON
    gearsMoving := TRUE
END
GearStopMoving
  STATUS ordinary
  REFINES GearStopMoving
  ANY in?
  WHERE in? = gearStopMoving_X ∧ gearsMoving
  THEN
    orange := OFF
    gearsMoving := FALSE
END
GearIsLockedDown
  STATUS ordinary
  REFINES GearIsLockedDown
  ANY in?
  WHERE in? = gearIsLockedDown_X ∧ ¬gearsLocked ∧
  THEN
    green := ON
    gearsLocked := TRUE
END
GearNotLockedDown
  STATUS ordinary
  REFINES GearNotLockedDown
  ANY in?
  WHERE in? = gearNotLockedDown_X ∧ gearsLocked
  THEN
    green := OFF
    gearsLocked := FALSE
END

Fig. 2 Level 1: Nominal top level model for the landing gear system with internal variables. System consists of:
**INTERFACE** Level2\_Comp\_IF

**VARIABLES**
- green, orange,
- gearsMoving, gearsLocked

**INVARIANTS**
- green, orange ∈ \{ON, OFF\}
- gearsMoving ∈ BOOL
- gearsLocked ∈ BOOL
- gearsMoving ⇔ orange = ON
- gearsLocked ⇔ green = ON

**INITIALISATION**
BEGIN
- green, orange := ON, OFF
- gearsMoving := FALSE
- gearsLocked := TRUE
END

---

**MACHINE** Level2\_PilotNominal

**REFINES** and **DECOMPOSES** Level1\_TopLevelTimingNominal

**CONNECTS** Level2\_LightsGears\_IF

**VARIABLES**
- handle

**INVARIANTS**
- handle ∈ \{UP, DOWN\}

**EVENTS**

**INITIALISATION**
BEGIN
- handle := DOWN
END

**PilotGearUP**
- STATUS ordinary
- DECOMPOSES PilotGearUP
- ANY in?
- WHERE in? = pilotGearUP \_X ∧ handle = DOWN
- THEN
- handle := UP
END

**PilotGearDOWN**
- STATUS ordinary
- DECOMPOSES PilotGearDOWN
- ANY in?
- WHERE in? = pilotGearDOWN \_X ∧ handle = UP
- THEN
- handle := DOWN
END

---

**Fig. 3** Level 2: Computer interface for nominal decomposed top level nominal timing model. System consists of:

---

**Fig. 4** Level 2: Decomposed nominal top level model: pilot machine. System consists of:
MACHINE Level2_CompNominal
REFINESandDECOMPOSES Level1_TopLevelTimingNominal
CONNECTS Level2_CompIF

VARIABLES
handlecmp

INVARIANTS
handlecmp ∈ {UP, DOWN}

EVENTS
INITIALISATION
BEGIN
handlecmp := DOWN
END
PilotGearUP ... ... ...
PilotGearUP_S
STATUS ordinary
REFINESandDECOMPOSES PilotGearUP
BEGIN
handlecmp := UP
END

PilotGearDOWN_S
STATUS ordinary
REFINESandDECOMPOSES PilotGearDOWN
BEGIN
handlecmp := DOWN
END

... ... ...

GearStartMoving_S
STATUS ordinary
DECOMPOSES GearStartMoving
ANY in?
WHERE in? = gearStartMoving_X ∧ ¬gearsMoving
THEN gearsMoving := TRUE
END

GearStopMoving_S
STATUS ordinary
DECOMPOSES GearStopMoving
ANY in?
WHERE in? = gearStopMoving_X ∧ gearsMoving
THEN gearsMoving := FALSE
END

GearIsLockedDown_S
STATUS ordinary
DECOMPOSES GearIsLockedDown
ANY in?
WHERE in? = gearIsLockedDown_X ∧ ¬gearsLocked
THEN gearsLocked := TRUE
END

GearNotLockedDown_S
STATUS ordinary
DECOMPOSES GearNotLockedDown
ANY in?
WHERE in? = gearNotLockedDown_X ∧ gearsLocked
THEN gearsLocked := FALSE
END

END

Fig. 5 Level 2: Decomposed nominal top level model: computing machine. System consists of:

Level2_CompIF Level2_PilotNominal Level2_CompNominal

... ... ...

Landing Gear System in Hybrid Event-B — Nominal Regime: Machines and Interfaces
INTERFACE Level3_Compl_IF
REFINES Level2_Compl_IF
VARIABLES
  handlecmp,
  green, orange,
  gearsMoving, gearsLocked
handlecmp1, handlecmp2,
gearsMoving1, gearsMoving2,
gearsLocked1, gearsLocked2
INVARIANTS
  handlecmp ∈ {UP, DOWN}
  green, orange ∈ {ON, OFF}
  gearsMoving ∈ BOOL
  gearsLocked ∈ BOOL
  handlecmp1, handlecmp2 ∈ {UP, DOWN}
  handlecmp = handlecmp1 ∧ handlecmp = handlecmp2
... ... ...
MACHINE Level3_CompNominal
REFINES Level2_CompNominal
CONNECTS Level3_CompIF
EVENTS
PilotTrue

PilotGearUP
STATUS ordinary
DECOMPOSES PilotGearUP
BEGIN
handlecmp := UP
END

PilotGearDOWN
STATUS ordinary
DECOMPOSES PilotGearDOWN
BEGIN
handlecmp := DOWN
END

GearStartMoving2
STATUS ordinary
REFINES GearStartMoving2
ANY in?
WHERE in? = gearStartMoving2 X \neg gearsMoving
THEN
gearsMoving := TRUE
gearsMoving2 := TRUE
END

GearStopMoving2
STATUS ordinary
REFINES GearStopMoving2
ANY in?
WHERE in? = gearStopMoving2 X \neg gearsMoving
THEN
gearsMoving := FALSE
gearsMoving2 := FALSE
END

Level2_PilotNominal Level3_CompIF Level3_CompNominal

Fig. 7 Level 3: This level implements the OR from the two computing machines to the abstract computing machine seen by the pilot. Refined nominal computing machines.

System consists of:

• First part.

GearsMovingIn := TRUE
GearsMoving := FALSE
Fig. 8 Level 3: This level implements the OR from the two computing machines to the abstract computing and machine seen by the pilot. Refined nominal computing machines.

System consists of:

Level2_PilotNominal Level3_CompJF Level3_CompNominal
MACHINE Level4_CompNominal
DECOMPOSES Level3_CompNominal
CONNECTS Level3_CompIF
EVENTS
PilotTrue ... ...
PilotGearUP_s
  STATUS ordinary
  DECOMPOSES PilotGearUP_s
  BEGIN
    handlecmp := UP
  END
PilotGearDOWN_s
  STATUS ordinary
  DECOMPOSES PilotGearDOWN_s
  BEGIN
    handlecmp := DOWN
  END
GearStartMoving1_s
  STATUS ordinary
  DECOMPOSES GearStartMoving1_s
  WHEN
    ¬gearsMoving
  THEN
    gearsMoving := TRUE
  END
GearStartMoving2_s
  STATUS ordinary
  DECOMPOSES GearStartMoving2_s
  WHEN
    ¬gearsMoving
  THEN
    gearsMoving := TRUE
  END
GearStopMoving1_s
  STATUS ordinary
  DECOMPOSES GearStopMoving1_s
  WHEN
    gearsMoving
  THEN
    gearsMoving := FALSE
  END
GearStopMoving2_s
  STATUS ordinary
  DECOMPOSES GearStopMoving2_s
  WHEN
    gearsMoving
  THEN
    gearsMoving := FALSE
  END

Fig. 9 Level 4: Decomposed refined nominal computing machine: Comp machine.

System consists of:

| Level2_PilotNominal | Level3_CompIF | Level4_CompNominal | Level4_Comp1Nominal | Level4_Comp2Nominal |
MACHINE Level4_Comp1Nominal
DECOMPOSES Level3_CompNominal1
CONNECTS Level3_CompNominal2

EVENTS
PilotTrue

CONNECTS PilotNominal

IF Level4_CompNominal1

DECOMPOSES GearStartMoving1S

STARTS ordinary

WHERE in? = gearStartMoving1S

THEN gearsMoving1 := TRUE

END

DECOMPOSES GearStopMoving1S

ANY in?

WHERE in? = gearStopMoving1S

THEN gearsMoving1 := FALSE

END

DECOMPOSES GearIsLockedDown1S

ANY in?

WHERE in? = gearIsLockedDown1S

THEN gearsLocked1 := TRUE

END

DECOMPOSES GearNotLockedDown1S

ANY in?

WHERE in? = gearNotLockedDown1S

THEN gearsLocked1 := FALSE

END

Fig. 10 Level 4: Decomposed refined nominal computing machine: Comp1 machine.

System consists of:
MACHINE Level4_Comp2Nominal
DECOMPOSES Level3_CompNominal
CONNECTS Level3_CompIF

EVENTS
PlicTrue ....... .......
PilotGearUP_S
  STATUS ordinary
  DECOMPOSES PilotGearUP_S
  BEGIN
    handlecmp2 := UP
  END
PilotGearDOWN_S
  STATUS ordinary
  DECOMPOSES PilotGearDOWN_S
  BEGIN
    handlecmp2 := DOWN
  END
GearStartMoving_S
  STATUS ordinary
  DECOMPOSES GearStartMoving_S
  WHERE in? = gearStartMoving2_X
  THEN
    gearsMoving2 := TRUE
  END
GearStartMoving2_Second
  STATUS ordinary
  DECOMPOSES GearStartMoving2_Second
  WHERE in? = gearStartMoving2_Second_X
  THEN
    gearsMoving2 := TRUE
  END
GearStopMoving_S
  STATUS ordinary
  DECOMPOSES GearStopMoving_S
  WHERE in? = gearStopMoving2_X
  THEN
    gearsMoving2 := FALSE
  END
GearStopMoving2_Second
  STATUS ordinary
  DECOMPOSES GearStopMoving2_Second
  WHERE in? = gearStopMoving2_Second_X
  THEN
    gearsMoving2 := FALSE
  END

... ... ... ...

GearIsLockedDown_S
  STATUS ordinary
  DECOMPOSES GearIsLockedDown_S
  ANY in?
  WHERE in? = gearIsLockedDown2_X
  THEN
    gearsLocked2 := TRUE
  END
GearIsLockedDown2_Second
  STATUS ordinary
  DECOMPOSES GearIsLockedDown2_Second
  ANY in?
  WHERE in? = gearIsLockedDown2_Second_X
  THEN
    gearsLocked2 := TRUE
  END
GearNotLockedDown_S
  STATUS ordinary
  DECOMPOSES GearNotLockedDown_S
  ANY in?
  WHERE in? = gearNotLockedDown2_X
  THEN
    gearsLocked2 := FALSE
  END
GearNotLockedDown2_Second
  STATUS ordinary
  DECOMPOSES GearNotLockedDown2_Second
  ANY in?
  WHERE in? = gearNotLockedDown2_Second_X
  THEN
    gearsLocked2 := FALSE
  END

END

Fig. 11 Level 4: Decomposed refined nominal computing machine: Comp2 machine.

System consists of:

Level2_PilotNominal | Level3_CompIF | Level4_CompNominal | Level4_Comp1Nominal | Level4_Comp2Nominal
REST IS DONE PER GENERIC Comp MACHINE