

An Event-B Specification of A1.Context
Generated Date: 7 Apr 2009 @ 04:26:54 PM

CONTEXT A1.Context

This is actually more complex than it need be, but just to show how we might 'build for the future', it defines a fairly complete context!

SETS

TRAINS arbitrary collection of trains that can travel the line
DIR which way is a train going?
STATIONS set of stations on the line
SECTIONS set of tracks. Each track can only take one train at a time

CONSTANTS

up travelling towards Huntingdale
down travelling towards Rowville
Huntingdale 'up' station
Rowville 'down' station
HuntingdaleRowville sections in the initial layout of tracks

AXIOMS

axm1 : $finite(TRAINS)$
cannot have an infinite supply of trains (even if Kosky does get off her bum!)

axm2 : $DIR = \{up, down\}$
up is towards Huntingdale, down is towards Rowville

axm3 : $STATIONS = \{Huntingdale, Rowville\}$
only two stations

axm4 : $SECTIONS = \{HuntingdaleRowville\}$
only one section

axm5 : $card(STATIONS) = 2$
I put this in because Rodin wasn't able to figure it out for itself.

axm6 : $card(TRAINS) \geq 1$

THEOREMS

thm1 : $card(SECTIONS) = card(STATIONS) - 1$
the number of stations must be one more than the number of sections, since there is a station at each end of the section

END

An Event-B Specification of A1_BasicMachine
Generated Date: 7 Apr 2009 @ 04:27:03 PM

MACHINE A1_BasicMachine

SEES A1.Context

VARIABLES

h the trains currently at Huntingdale
r the trains currently at Rowville
th the trains in transit from Huntingdale to Rowville
tr the trains in transit from Rowville to Huntingdale

INVARIANTS

inv1 : $h \subseteq TRAINS$
inv2 : $r \subseteq TRAINS$
inv3 : $th \subseteq TRAINS$
inv4 : $tr \subseteq TRAINS$

EVENTS

Initialisation

begin
 act3 : $h := TRAINS$
 act1 : $r := \emptyset$
 act2 : $th, tr := \emptyset, \emptyset$
end

Event *leaveHuntingdale* $\hat{=}$

any
 train
where
 grd1 : $train \in h$
 choose a train, any train (or, at least, one whose brakes are working!)
 grd2 : $th \cup tr = \emptyset$
 no trains in transit
then
 act1 : $th := \{train\}$
 one train in transit from Huntingdale to Rowville
 act2 : $h := h \setminus \{train\}$
 one less train at Huntingdale
end

Event *arriveRowville* $\hat{=}$

any
 train choose a train, but not just any one ...
where
 grd1 : $train \in th$
 ... it must be coming from Huntingdale
then

act1 : $th := \emptyset$
 no trains in transit now
act2 : $r := r \cup \{train\}$
 "train now arriving on platform ..."

end

Event *leaveRowville* $\hat{=}$

any

train

where

grd1 : $train \in r$
 choose a train from the Rowville stable
grd2 : $th \cup tr = \emptyset$
 no trains in transit

then

act1 : $tr := \{train\}$
 one train in transit from Rowville to Huntingdale
act2 : $r := r \setminus \{train\}$
 one less train at Rowville

end

Event *arriveHuntingdale* $\hat{=}$

any

train choose a train, but not just any one ...

where

grd1 : $train \in tr$
 ... it must be coming from Rowville

then

act1 : $tr := \emptyset$
 no trains in transit now
act2 : $h := h \cup \{train\}$
 train has returned to Huntingdale

end

END

An Event-B Specification of A1_SignallingContext
Generated Date: 7 Apr 2009 @ 04:26:59 PM

CONTEXT A1_SignallingContext

This is a very simple extension of A1.Context, just to add the signal set

EXTENDS A1.Context

SETS

SIGNAL

CONSTANTS

red

green

AXIOMS

axm1 : $SIGNAL = \{red, green\}$

END

An Event-B Specification of A1_SignallingMachine
Generated Date: 7 Apr 2009 @ 04:27:06 PM

MACHINE A1_SignallingMachine

REFINES A1_BasicMachine

SEES A1_SignallingContext

VARIABLES

h the trains currently at Huntingdale
r the trains currently at Rowville
th the trains in transit from Huntingdale to Rowville
tr the trains in transit from Rowville to Huntingdale
sH the signals at Huntingdale
sR the signals at Rowville

INVARIANTS

inv1 : $h \subseteq TRAINS$
inv2 : $r \subseteq TRAINS$
inv3 : $th \subseteq TRAINS$
inv4 : $tr \subseteq TRAINS$
inv5 : $sH \in SIGNAL$
inv6 : $sR \in SIGNAL$

EVENTS

Initialisation

begin
 act1 : $sH := red$
 act2 : $sR := red$
 act3 : $tr := \emptyset$
 act4 : $th := \emptyset$
 act5 : $h := TRAINS$
 initially all trains are at Huntingdale
 act6 : $r := \emptyset$
 and none at Rowville
end

Event *setHuntingdaleRed* $\hat{=}$

begin
 act1 : $sH := red$
end

Event *setRowvilleRed* $\hat{=}$

begin
 act1 : $sR := red$
end

Event *setHuntingdaleGreen* $\hat{=}$

when
 grd1 : $sR = red$
 cannot have conflicting signal
 grd2 : $tr \cup th = \emptyset$
 no trains in transit
then
 act1 : $sH := green$
end

Event *setRowvilleGreen* $\hat{=}$

when
 grd1 : $sH = red$
 cannot have conflicting signal
 grd2 : $tr \cup th = \emptyset$
 no trains in transit
then
 act1 : $sR := green$
end

Event *leaveHuntingdale* $\hat{=}$

refines *leaveHuntingdale*

any
 train
where
 grd1 : $train \in h$
 grd2 : $sH = green$
 grd3 : $tr \cup th = \emptyset$
then
 act1 : $th := \{train\}$
 act2 : $h := h \setminus \{train\}$
 act3 : $sH := red$
end

Event *arriveRowville* $\hat{=}$

refines *arriveRowville*

any
 train
where
 grd1 : $train \in th$
then
 act1 : $th := \emptyset$
 no trains in transit now
 act2 : $r := r \cup \{train\}$
end

Event *leaveRowville* $\hat{=}$

refines *leaveRowville*

```
any
  train
where
  grd1 : train ∈ r
        train leaves from Rowville
  grd2 : tr ∪ th = ∅
        no trains in transit
  grd3 : sR = green
        signal must be green
then
  act1 : tr := {train}
  act2 : r := r \ {train}
        one less train at Rowville
  act3 : sR := red
end
Event arriveHuntingdale ≐
refines arriveHuntingdale
  any
    train
  where
    grd1 : train ∈ tr
  then
    act1 : h := h ∪ {train}
    act2 : tr := ∅
  end
end
END
```