Contents

CONTEXT C0
SETS
    TRAIN
CONSTANTS
    a
    b
    WAY
AXIOMS
    axiom1: $\{a, b\} \subseteq \mathbb{N}$
    axiom2: $a < b$
    axiom3: $WAY = a\,..\,b$
    axiom4: $b - a \geq 20$
END

CONTEXT C2
EXTENDS C0
SETS

STATES

CONSTANTS

TTD
VSS
OCCUPIED
FREE
UNKNOW
AMBIGUOUS

AXIOMS

axiom1: $TTD \subseteq \mathbb{P}_1(WAY)$
axiom2: $union(TTD) = WAY$
axiom3: $inter(TTD) = \varnothing$
axiom4: $\forall ttd \cdot (ttd \in TTD \Rightarrow (\exists p, q \cdot (p \mathbin{..} q \subseteq WAY \land p < q \land ttd = p \mathbin{..} q)))$
axiom5: $VSS \subseteq \mathbb{P}_1(WAY)$
axiom6: $union(VSS) = WAY$
axiom7: $inter(VSS) = \varnothing$
axiom8: $\forall vss \cdot (vss \in VSS \Rightarrow (\exists p, q, ttd \cdot (ttd \in TTD \land p \mathbin{..} q \subseteq ttd \land p < q \land vss = p \mathbin{..} q)))$
axiom9: $partition(STATES, \{OCCUPIED\}, \{FREE\}, \{UNKNOW\}, \{AMBIGUOUS\})$

END

CONTEXT C3
EXTENDS C2
SETS

TIMER_STATUS

CONSTANTS

INACTIVE

STARTED

EXPIRED

AXIOMS

axm1: $partition(TIMER\_STATUS, \{INACTIVE\}, \{STARTED\}, \{EXPIRED\})$

END

MACHINE M0
SEES C0
VARIABLES

connectedTrain
front
rear

INVARIANTS

inv0_1: $connectedTrain \in TRAIN \nrightarrow BOOL$
inv0_2: $front \in dom(connectedTrain) \rightarrow WAY$
inv0_3: $rear \in dom(connectedTrain) \nrightarrow WAY$
inv0_4: $\forall tr\cdot(tr \in dom(rear) \Rightarrow rear(tr) < front(tr))$

EVENTS
Initialisation

begin
act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
end

Event MoveTrainOnTrack ⟨ordinary⟩ $\widehat{=}$

any
tr
len
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in WAY$
then
act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr)+len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$

any
tr
fr
re
integer
where
grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
then
act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
end

Event _exitTrain ⟨ordinary⟩ $\widehat{=}$

any
tr
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
then
act1: $front := \{tr\} \lhd front$
act2: $rear := (\{TRUE \mapsto \{tr\} \lhd rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $connectedTrain := \{tr\} \lhd connectedTrain$

end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$

grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Lleftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Lleftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

END

MACHINE M1
REFINES M0
SEES C0
VARIABLES

connectedTrain
front
rear
MA
MAtemp

INVARIANTS

inv1_1: $MA \in dom(connectedTrain) \rightarrow\!\!\!\!\!\!+ \;\mathbb{P}(WAY)$
inv1_2: $\forall tr\cdot(tr \in dom(MA) \Rightarrow (\exists p, q\cdot(p\,..\,q \subseteq WAY \land p \leq q \land MA(tr) = p\,..\,q)))$
inv1_3: $\forall tr\cdot(tr \in dom(MA) \Rightarrow front(tr) \in MA(tr))$
inv1_4: $\forall tr\cdot(tr \in dom(rear) \cap dom(MA) \Rightarrow rear(tr) \in MA(tr))$
inv1_5: $\forall tr1, tr2\cdot((\{tr1, tr2\} \subseteq dom(MA) \land tr1 \neq tr2) \Rightarrow MA(tr1) \cap MA(tr2) = \varnothing)$
inv1_6: $MAtemp \in dom(connectedTrain) \rightarrow\!\!\!\!\!\!+ \;\mathbb{P}(WAY)$
inv1_7: $\forall tr\cdot(tr \in dom(MAtemp) \Rightarrow (\exists p, q\cdot(p\,..\,q \subseteq WAY \land p \leq q \land MAtemp(tr) = p\,..\,q)))$

SYSML/KAOS PROOF OBLIGATIONS

sysml_kaos_po_G1-Guard=>G-Guard: ⟨theorem⟩

$\forall tr, p, q, len\cdot(($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (p\,..\,q \subseteq WAY \land p \leq q)$
$\land (front(tr) \in p\,..\,q)$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q)$
$\land (p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in WAY)$
$) \Rightarrow$
$($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in WAY)$
$))$

remplacement de toute reference a MAtemp par $((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p..q\})$

sysml_kaos_po_G1-Post=>G2-Guard: ⟨theorem⟩

$\forall tr, p, q, len\cdot(($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (p\,..\,q \subseteq WAY \land p \leq q)$
$\land (front(tr) \in p\,..\,q)$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q)$
$\land (p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in p\,..\,q)$
$) \Rightarrow$
$($
$(tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p\,..\,q\})))$
$\land (front(tr) \in ((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p\,..\,q\})(tr))$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in ((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p\,..\,q\})(tr))$
$\land (((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p\,..\,q\})(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (len \in \mathbb{N}_1)$
$\land front(tr) + len \in ((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p\,..\,q\})(tr)$
$))$

remplacement de toute reference a MA par $((\{tr\} \lhd MA) \cup \{tr \mapsto MAtemp(tr)\})$

sysml_kaos_po_G2-Post=>G3-Guard: ⟨theorem⟩

$\forall tr, len\cdot(($
$(tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp))$
$\land (front(tr) \in MAtemp(tr))$

$\land (tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr))$
$\land (MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (len \in \mathbb{N}_1)$
$\land front(tr) + len \in MAtemp(tr)$
$) \Rightarrow$
$($
$(tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(((\{tr\} \lhd MA) \cup \{tr \mapsto MAtemp(tr)\})))$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in ((\{tr\} \lhd MA) \cup \{tr \mapsto MAtemp(tr)\})(tr))$
$))$

sysml_kaos_po_G3-Post=>G-Post: ⟨theorem⟩
$\forall tr, len \cdot ($
$($
$(tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA))$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in MA(tr))$
$) \Rightarrow$
$($
$(front(tr) + len = front(tr) + len)$
$\land (tr \in dom(rear) \Rightarrow ((\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\}) = (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})))$
$)$
$)$

EVENTS

Initialisation

begin
act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
end

Event ComputeTrainMA ⟨ordinary⟩ $\widehat{=}$

any
tr
p
q
len
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $p \,..\, q \subseteq WAY \land p \leq q$
grd3: $front(tr) \in p \,..\, q$
grd4: $tr \in dom(rear) \Rightarrow rear(tr) \in p \,..\, q$
grd5: $p \,..\, q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd6: $len \in \mathbb{N}_1$
grd7: $front(tr) + len \in WAY$
then
act1: $MAtemp(tr) := p \,..\, q$
end

Event AssignMAtoTrain ⟨ordinary⟩ $\widehat{=}$

any
tr
len
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MAtemp)$
grd2: $front(tr) \in MAtemp(tr)$
grd3: $tr \in dom(rear) \Rightarrow rear(tr) \in MAtemp(tr)$
grd4: $MAtemp(tr) \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd5: $len \in \mathbb{N}_1$

grd6: $front(tr) + len \in MAtemp(tr)$

then

act1: $MA(tr) := MAtemp(tr)$

end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$

refines MoveTrainOnTrack

any

tr

len

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$

grd2: $len \in \mathbb{N}_1$

grd3: $front(tr) + len \in MA(tr)$

then

act1: $front(tr) := front(tr) + len$

act2: $rear := (\{TRUE \mapsto rear \Lleftarrow \{tr \mapsto rear(tr)+len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$

end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$

extends _connectTrain

any

tr

fr

re

$integer$

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$

grd1: $tr \in TRAIN \setminus dom(connectedTrain)$

grd2: $fr \in WAY$

grd3: $integer \in BOOL$

grd4: $integer = TRUE \Rightarrow re \in WAY$

grd5: $re < fr$

then

act1: $connectedTrain(tr) := TRUE$

act2: $front(tr) := fr$

act3: $rear := (\{TRUE \mapsto rear \Lleftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$

end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$

extends _toggleTrainConnexionStatus

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$

grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Lleftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Lleftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

Event _exitTrain ⟨ordinary⟩ $\widehat{=}$

extends _exitTrain

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$

then

act1: $front := \{tr\} \lhd\!\!\!- front$

act2: $rear := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$

act3: $connectedTrain := \{tr\} \lhd\!\!\!- connectedTrain$

act4: $MA := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$

act5: $MAtemp := (\{TRUE \mapsto \{tr\} \lhd MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$

end

END

MACHINE M2
REFINES M1
SEES C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS

INVARIANTS

inv2_1: $stateTTD \in TTD \rightarrow \{OCCUPIED, FREE\}$

inv2_2: $stateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

inv2_3: $\forall ttd, tr \cdot ((tr \in dom(front) \setminus dom(rear) \land ttd \in TTD \land front(tr) \in ttd) \Rightarrow stateTTD(ttd) = OCCUPIED)$

inv2_4: $\forall ttd, tr \cdot ((tr \in dom(rear) \land ttd \in TTD \land (rear(tr) \mathbin{..} front(tr)) \cap ttd \neq \varnothing) \Rightarrow stateTTD(ttd) = OCCUPIED)$

inv2_5: $\forall tr1, tr2 \cdot ((tr1 \in dom(rear) \land tr2 \in dom(rear) \land tr1 \neq tr2) \Rightarrow (rear(tr1) \mathbin{..} front(tr1)) \cap (rear(tr2) \mathbin{..} front(tr2)) = \varnothing)$

inv2_6: $\forall tr1, tr2 \cdot ((tr1 \in dom(rear) \land tr2 \in dom(front) \setminus dom(rear) \land tr1 \neq tr2) \Rightarrow front(tr2) < rear(tr1))$

inv2_7: $\forall tr1, tr2, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr2 \in dom(front) \setminus dom(rear) \land tr1 \neq tr2 \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr2) \notin ttd)$

SYSML/KAOS PROOF OBLIGATIONS

sysml_kaos_po_G1-Guard=>G-Guard: ⟨theorem⟩

$\forall tr, p, q, len, ttds, ttds1, p0, p1, q1 \cdot (($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (ttds \subseteq stateTTD^{-1}[\{FREE\}])$
$\land (union(ttds) = p1 \mathbin{..} q1)$
$\land (p1 \geq front(tr))$
$\land (ttds1 \subseteq TTD)$
$\land (union(ttds1) = p0 \mathbin{..} (p1 - 1))$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\land (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\land (p \mathbin{..} q \subseteq union(ttds \cup ttds1))$
$\land (p \mathbin{..} q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (front(tr) \in p \mathbin{..} q)$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathbin{..} q)$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in WAY)$
$\land (tr \notin dom(MAtemp) \lor MAtemp(tr) \neq p \mathbin{..} q)$
$) \Rightarrow$
$($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (p \mathbin{..} q \subseteq WAY \land p \leq q)$
$\land (front(tr) \in p \mathbin{..} q)$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathbin{..} q)$
$\land (p \mathbin{..} q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in WAY)$
$))$

sysml_kaos_po_G2-Guard=>G-Guard: ⟨theorem⟩

$\forall tr, p, q, len, vsss, vsss1, p0, p1, q1, newstateVSS \cdot (($
$(newstateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$\land (tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (vsss \subseteq newstateVSS^{-1}[\{FREE\}])$

$\wedge\ (union(vsss) = p1\,..\,q1)$
$\wedge\ (p1 \geq front(tr))$
$\wedge\ (vsss1 \subseteq VSS)$
$\wedge\ (union(vsss1) = p0\,..\,(p1-1))$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\wedge\ (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\wedge\ (p\,..\,q \subseteq union(vsss \cup vsss1))$
$\wedge\ (p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\wedge\ (front(tr) \in p\,..\,q)$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q)$
$\wedge\ (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p\,..\,q)$
$\wedge\ (len \in \mathbb{N}_1)$
$\wedge\ (front(tr) + len \in WAY)$
$\wedge\ (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p\,..\,q)$
$) \Rightarrow$
$($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\wedge\ (p\,..\,q \subseteq WAY \wedge p \leq q)$
$\wedge\ (front(tr) \in p\,..\,q)$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q)$
$\wedge\ (p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\wedge\ (len \in \mathbb{N}_1)$
$\wedge\ (front(tr) + len \in WAY)$
$))$

sysml_kaos_po_G1-Post=>G-Post: ⟨theorem⟩

$\forall tr, p, q, len, ttds, ttds1, p0, p1, q1 \cdot (($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\wedge\ (ttds \subseteq stateTTD^{-1}[\{FREE\}])$
$\wedge\ (union(ttds) = p1\,..\,q1)$
$\wedge\ (p1 \geq front(tr))$
$\wedge\ (ttds1 \subseteq TTD)$
$\wedge\ (union(ttds1) = p0\,..\,(p1-1))$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\wedge\ (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\wedge\ (p\,..\,q \subseteq union(ttds \cup ttds1))$
$\wedge\ (p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\wedge\ (front(tr) \in p\,..\,q)$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q)$
$\wedge\ (len \in \mathbb{N}_1)$
$\wedge\ (front(tr) + len \in WAY)$
$\wedge\ (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p\,..\,q)$
$) \Rightarrow$
$($
$(p\,..\,q = p\,..\,q)$
$)$
$)$

sysml_kaos_po_G2-Post=>G-Post: ⟨theorem⟩

$\forall tr, p, q, len, vsss, vsss1, p0, p1, q1, newstateVSS \cdot (($
$(newstateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$\wedge\ (tr \in connectedTrain^{-1}[\{TRUE\}])$
$\wedge\ (vsss \subseteq newstateVSS^{-1}[\{FREE\}])$
$\wedge\ (union(vsss) = p1\,..\,q1)$
$\wedge\ (p1 \geq front(tr))$
$\wedge\ (vsss1 \subseteq VSS)$
$\wedge\ (union(vsss1) = p0\,..\,(p1-1))$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\wedge\ (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\wedge\ (p\,..\,q \subseteq union(vsss \cup vsss1))$
$\wedge\ (p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$

$\wedge\ (front(tr) \in p \mathrel{..} q)$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathrel{..} q)$
$\wedge\ (len \in \mathbb{N}_1)$
$\wedge\ (front(tr) + len \in WAY)$
$\wedge\ (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p \mathrel{..} q)$
$) \Rightarrow$
$($
$(p \mathrel{..} q = p \mathrel{..} q)$
$)$
$)$
remplacement de MAtemp par (({tr}⩤MAtemp)∪{tr↦ p..q})

sysml_kaos_po_G1-Post=>not(G2-Guard): ⟨theorem⟩

$\forall tr, p, q, len, ttds, ttds1, p0, p1, q1 \cdot (($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\wedge\ (ttds \subseteq stateTTD^{-1}[\{FREE\}])$
$\wedge\ (union(ttds) = p1 \mathrel{..} q1)$
$\wedge\ (p1 \geq front(tr))$
$\wedge\ (ttds1 \subseteq TTD)$
$\wedge\ (union(ttds1) = p0 \mathrel{..} (p1 - 1))$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\wedge\ (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\wedge\ (p \mathrel{..} q \subseteq union(ttds \cup ttds1))$
$\wedge\ (p \mathrel{..} q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\wedge\ (front(tr) \in p \mathrel{..} q)$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathrel{..} q)$
$\wedge\ (len \in \mathbb{N}_1)$
$\wedge\ (front(tr) + len \in WAY)$
$\wedge\ (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p \mathrel{..} q)$
$) \Rightarrow$
$\neg(\exists vsss, vsss1, newstateVSS \cdot ($
$(newstateVSS \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$\wedge\ (tr \in connectedTrain^{-1}[\{TRUE\}])$
$\wedge\ (vsss \subseteq newstateVSS^{-1}[\{FREE\}])$
$\wedge\ (union(vsss) = p1 \mathrel{..} q1)$
$\wedge\ (p1 \geq front(tr))$
$\wedge\ (vsss1 \subseteq VSS)$
$\wedge\ (union(vsss1) = p0 \mathrel{..} (p1 - 1))$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\wedge\ (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\wedge\ (p \mathrel{..} q \subseteq union(vsss \cup vsss1))$
$\wedge\ (p \mathrel{..} q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\wedge\ (front(tr) \in p \mathrel{..} q)$
$\wedge\ (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathrel{..} q)$
$\wedge\ (len \in \mathbb{N}_1)$
$\wedge\ (front(tr) + len \in WAY)$
$\wedge\ (tr \notin dom(((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p \mathrel{..} q\})) \vee ((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p \mathrel{..} q\})(tr) \neq p \mathrel{..} q)$
$)$
$)$
$)$
remplacement de MAtemp par (({tr}⩤MAtemp)∪{tr↦ p..q})

sysml_kaos_po_G2-Post=>not(G1-Guard): ⟨theorem⟩

$\forall tr, p, q, len, vsss, vsss1, p0, p1, q1, newstateVSS \cdot (($
$(newstateVSS \in VSS \to \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$\wedge\ (tr \in connectedTrain^{-1}[\{TRUE\}])$
$\wedge\ (vsss \subseteq newstateVSS^{-1}[\{FREE\}])$
$\wedge\ (union(vsss) = p1 \mathrel{..} q1)$
$\wedge\ (p1 \geq front(tr))$
$\wedge\ (vsss1 \subseteq VSS)$
$\wedge\ (union(vsss1) = p0 \mathrel{..} (p1 - 1))$

$\land (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\land (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\land (p \mathinner{..} q \subseteq union(vsss \cup vsss1))$
$\land (p \mathinner{..} q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (front(tr) \in p \mathinner{..} q)$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathinner{..} q)$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in WAY)$
$\land (tr \notin dom(MAtemp) \lor MAtemp(tr) \neq p \mathinner{..} q)$
$) \Rightarrow$
$\neg(\exists ttds, ttds1 \cdot ($
$(tr \in connectedTrain^{-1}[\{TRUE\}])$
$\land (ttds \subseteq stateTTD^{-1}[\{FREE\}])$
$\land (union(ttds) = p1 \mathinner{..} q1)$
$\land (p1 \geq front(tr))$
$\land (ttds1 \subseteq TTD)$
$\land (union(ttds1) = p0 \mathinner{..} (p1 - 1))$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$
$\land (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$
$\land (p \mathinner{..} q \subseteq union(ttds \cup ttds1))$
$\land (p \mathinner{..} q \cap union(ran(\{tr\} \lhd MA)) = \varnothing)$
$\land (front(tr) \in p \mathinner{..} q)$
$\land (tr \in dom(rear) \Rightarrow rear(tr) \in p \mathinner{..} q)$
$\land (len \in \mathbb{N}_1)$
$\land (front(tr) + len \in WAY)$
$\land (tr \notin dom(((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p \mathinner{..} q\})) \lor ((\{tr\} \lhd MAtemp) \cup \{tr \mapsto p \mathinner{..} q\})(tr) \neq p \mathinner{..} q)$
$)$
$)$
$)$

EVENTS

Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$

end

Event ComputeTrainMAFollowingTTDStates ⟨ordinary⟩ $\widehat{=}$

refines ComputeTrainMA

any

tr
ttds
p
q
ttds1
p0
p1
q1
len ttds1 designe l'ensemble des ttd sur lesquels le train est succeptible de se trouver

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd3: $union(ttds) = p1 \mathinner{..} q1$
grd4: $p1 \geq front(tr)$
grd5: $ttds1 \subseteq TTD$
grd6: $union(ttds1) = p0 \mathinner{..} (p1 - 1)$

grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p \,..\, q \subseteq union(ttds \cup ttds1)$
grd10: $p \,..\, q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p \,..\, q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p \,..\, q$
grd13: $len \in \mathbb{N}_1$
grd14: $front(tr) + len \in WAY$
grd15: $tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p \,..\, q$

then

act1: $MAtemp(tr) := p \,..\, q$

end

Event ComputeTrainMAFollowingVSSStates ⟨ordinary⟩ $\widehat{=}$

refines ComputeTrainMA

any

tr
vsss
p
q
vsss1
p0
p1
q1
newstateVSS
len vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver

where

grd0: $newstateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$
grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p \,..\, q \subseteq union(vsss \cup vsss1)$
grd10: $p \,..\, q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p \,..\, q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p \,..\, q$
grd13: $len \in \mathbb{N}_1$
grd14: $front(tr) + len \in WAY$
grd15: $tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p \,..\, q$

then

act1: $MAtemp(tr) := p \,..\, q$
act2: $stateVSS := newstateVSS$

end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$

extends MoveTrainFollowingItsMA

any

tr
len
ttds

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((front(tr) + len \in ttd) \vee (tr \in dom(rear) \wedge ((rear(tr) + len \,..\, front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$

grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) .. front(tr1)) \cap (rear(tr) + len .. front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then
act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$
extends _connectTrain
any
tr
fr
re
$integer$
ttds

where
grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((fr \in ttd) \lor (integer = TRUE \land ((re .. fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $integer = TRUE \Rightarrow (\forall tr1 \cdot (tr1 \in dom(rear) \Rightarrow (rear(tr1) .. front(tr1)) \cap (re .. fr) = \varnothing))$
grd9: $integer = TRUE \Rightarrow (\forall tr1 \cdot (tr1 \in dom(front) \setminus dom(rear) \Rightarrow front(tr1) < re))$
grd10: $integer = FALSE \Rightarrow (\forall tr1 \cdot (tr1 \in dom(rear) \Rightarrow fr < rear(tr1)))$
grd11: $integer = FALSE \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then
act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$
extends _toggleTrainConnexionStatus
any
tr

where
grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then
act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

Event _exitTrain ⟨ordinary⟩ $\widehat{=}$
extends _exitTrain
any
tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$

then

act1: $front := \{tr\} \mathbin{\lhd\!\!\!-} front$

act2: $rear := (\{TRUE \mapsto \{tr\} \mathbin{\lhd\!\!\!-} rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$

act3: $connectedTrain := \{tr\} \mathbin{\lhd\!\!\!-} connectedTrain$

act4: $MA := (\{TRUE \mapsto \{tr\} \mathbin{\lhd\!\!\!-} MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$

act5: $MAtemp := (\{TRUE \mapsto \{tr\} \mathbin{\lhd\!\!\!-} MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$

end

END

MACHINE M3
REFINES M2
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed

INVARIANTS

inv3_1: $newstateVSScomputed \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

EVENTS
Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$

end

Event ComputeVSSStates ⟨ordinary⟩ $\widehat{=}$

any

newstateVSScomputed1

where

grd0: $newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

Event ComputeTrainMAUsingVSSStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeTrainMAFollowingVSSStates

any

tr
vsss
p
q
vsss1
p0
p1
q1
len
newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est succeptible de se trouver

where

grd0: $newstateVSS = newstateVSScomputed$
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $vsss \subseteq newstateVSS^{-1}[\{FREE\}]$
grd3: $union(vsss) = p1 \,..\, q1$
grd4: $p1 \geq front(tr)$
grd5: $vsss1 \subseteq VSS$
grd6: $union(vsss1) = p0 \,..\, (p1 - 1)$
grd7: $tr \in dom(rear) \Rightarrow rear(tr) \geq p0$

grd8: $tr \notin dom(rear) \Rightarrow front(tr) \geq p0$
grd9: $p\,..\,q \subseteq union(vsss \cup vsss1)$
grd10: $p\,..\,q \cap union(ran(\{tr\} \lhd MA)) = \varnothing$
grd11: $front(tr) \in p\,..\,q$
grd12: $tr \in dom(rear) \Rightarrow rear(tr) \in p\,..\,q$
grd13: $len \in \mathbb{N}_1$
grd14: $front(tr) + len \in WAY$
grd15: $tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p\,..\,q$

then

act1: $MAtemp(tr) := p\,..\,q$
act2: $stateVSS := newstateVSS$

end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$
extends _connectTrain

any

tr
fr
re
$integer$
$ttds$

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (integer = TRUE \wedge ((re\,..\,fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $integer = TRUE \Rightarrow (\forall tr1\cdot(tr1 \in dom(rear) \Rightarrow (rear(tr1)\,..\,front(tr1)) \cap (re\,..\,fr) = \varnothing))$
grd9: $integer = TRUE \Rightarrow (\forall tr1\cdot(tr1 \in dom(front) \setminus dom(rear) \Rightarrow front(tr1) < re))$
grd10: $integer = FALSE \Rightarrow (\forall tr1\cdot(tr1 \in dom(rear) \Rightarrow fr < rear(tr1)))$
grd11: $integer = FALSE \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then

act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$

end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$
extends _toggleTrainConnexionStatus

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$
extends MoveTrainFollowingItsMA

any

tr
len
$ttds$

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$

grd2: $len \in \mathbb{N}_1$

grd3: $front(tr) + len \in MA(tr)$

grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$

grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr) + len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr) + len \,..\, front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$

grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr) + len \,..\, front(tr) + len) = \varnothing))$

grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$

grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$

grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then

act1: $front(tr) := front(tr) + len$

act2: $rear := (\{TRUE \mapsto rear \mathbin{<\!\!+} \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$

act3: $stateTTD := stateTTD \mathbin{<\!\!+} (ttds \times \{OCCUPIED\})$

end

Event _exitTrain ⟨ordinary⟩ $\widehat{=}$

extends _exitTrain

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$

then

act1: $front := \{tr\} \lhd\!\!\!- front$

act2: $rear := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$

act3: $connectedTrain := \{tr\} \lhd\!\!\!- connectedTrain$

act4: $MA := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$

act5: $MAtemp := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$

end

END

MACHINE M4
REFINES M3
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed

EVENTS
Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$

end

Event ComputeVSSStatesFollowingTTDStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStates

any

newstateVSScomputed1

where

grd0: $newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

Event ComputeVSSStateswoTTDStates ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStates

any

newstateVSScomputed1

where

grd0: $newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$
extends _connectTrain

any

tr
fr
re
integer
ttds

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$

grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (integer = TRUE \wedge ((re \,..\, fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $integer = TRUE \Rightarrow (\forall tr1\cdot(tr1 \in dom(rear) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (re \,..\, fr) = \varnothing))$
grd9: $integer = TRUE \Rightarrow (\forall tr1\cdot(tr1 \in dom(front) \setminus dom(rear) \Rightarrow front(tr1) < re))$
grd10: $integer = FALSE \Rightarrow (\forall tr1\cdot(tr1 \in dom(rear) \Rightarrow fr < rear(tr1)))$
grd11: $integer = FALSE \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then
act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$
end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$
extends _toggleTrainConnexionStatus
any
tr
where
grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$
then
act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$
end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$
extends MoveTrainFollowingItsMA
any
tr
len
$ttds$
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((front(tr)+len \in ttd) \vee (tr \in dom(rear) \wedge ((rear(tr)+len \,..\, front(tr)+len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr)+len \,..\, front(tr)+len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr1) < rear(tr)+len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow front(tr)+len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow front(tr)+len \notin ttd))$
then
act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto rear(tr)+len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$
end

Event _exitTrain ⟨ordinary⟩ $\widehat{=}$
extends _exitTrain
any
tr
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$

then

act1: $front := \{tr\} \lhd front$
act2: $rear := (\{TRUE \mapsto \{tr\} \lhd rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $connectedTrain := \{tr\} \lhd connectedTrain$
act4: $MA := (\{TRUE \mapsto \{tr\} \lhd MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$
act5: $MAtemp := (\{TRUE \mapsto \{tr\} \lhd MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$

end

END

MACHINE M5
REFINES M4
SEES C0,C2
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed SYSML/KAOS PROOF OBLIGATIONS

INVARIANTS

`sysml_kaos_po_G1-Guard=>G-Guard`: ⟨theorem⟩

$\forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1 \cdot (($
$(vss = stateVSS^{-1}[\{UNKNOW\}])$
$\wedge (partition(vss, vss1, vss2, vss3, vss4))$
$\wedge (newstateVSScomputed1 = stateVSS \lhd\!\!- ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\})))$
$) \Rightarrow$
$($
$(newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$))$

`sysml_kaos_po_G2-Guard=>G-Guard`: ⟨theorem⟩

$\forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1 \cdot (($
$(vss = stateVSS^{-1}[\{OCCUPIED\}])$
$\wedge (partition(vss, vss1, vss2, vss3, vss4))$
$\wedge (newstateVSScomputed1 = stateVSS \lhd\!\!- ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\})))$
$) \Rightarrow$
$($
$(newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$))$

`sysml_kaos_po_G3-Guard=>G-Guard`: ⟨theorem⟩

$\forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1 \cdot (($
$(vss = stateVSS^{-1}[\{AMBIGUOUS\}])$
$\wedge (partition(vss, vss1, vss2, vss3, vss4))$
$\wedge (newstateVSScomputed1 = stateVSS \lhd\!\!- ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\})))$
$) \Rightarrow$
$($
$(newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$))$

`sysml_kaos_po_G4-Guard=>G-Guard`: ⟨theorem⟩

$\forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1 \cdot (($
$(vss = stateVSS^{-1}[\{FREE\}])$
$\wedge (partition(vss, vss1, vss2, vss3, vss4))$
$\wedge (newstateVSScomputed1 = stateVSS \lhd\!\!- ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\})))$
$) \Rightarrow$
$($
$(newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\})$
$))$

`sysml_kaos_po_G1-G2-G3-G4-G5-Post=>G-Post` : ⟨theorem⟩

$\forall vss1, vss11, vss12, vss13, vss14, vss2, vss21, vss22, vss23, vss24, vss3, vss31, vss32, vss33, vss34, vss4, vss41, vss42, vs$
$($
$(vss1 = stateVSS^{-1}[\{UNKNOW\}])$

$\wedge\ (partition(vss1, vss11, vss12, vss13, vss14))$
$\wedge\ (vss2 = stateVSS^{-1}[\{OCCUPIED\}])$
$\wedge\ (partition(vss2, vss21, vss22, vss23, vss24))$
$\wedge\ (vss3 = stateVSS^{-1}[\{AMBIGUOUS\}])$
$\wedge\ (partition(vss3, vss31, vss32, vss33, vss34))$
$\wedge\ (vss4 = stateVSS^{-1}[\{FREE\}])$
$\wedge\ (partition(vss4, vss41, vss42, vss43, vss44))$
$) \Rightarrow$
$($
$(stateVSS \Leftarrow ((vss11 \times \{OCCUPIED\}) \cup (vss12 \times \{FREE\}) \cup (vss13 \times \{AMBIGUOUS\}) \cup (vss14 \times \{UNKNOW\})))$
$\Leftarrow (stateVSS \Leftarrow ((vss21 \times \{OCCUPIED\}) \cup (vss22 \times \{FREE\}) \cup (vss23 \times \{AMBIGUOUS\}) \cup (vss24 \times \{UNKNOW\})))$
$\Leftarrow (stateVSS \Leftarrow ((vss31 \times \{OCCUPIED\}) \cup (vss32 \times \{FREE\}) \cup (vss33 \times \{AMBIGUOUS\}) \cup (vss34 \times \{UNKNOW\})))$
$\Leftarrow (stateVSS \Leftarrow ((vss41 \times \{OCCUPIED\}) \cup (vss42 \times \{FREE\}) \cup (vss43 \times \{AMBIGUOUS\}) \cup (vss44 \times \{UNKNOW\})))$
$\in VSS \rightarrow \{OCCUPIED, FREE, UNKNOW, AMBIGUOUS\}$
$)$
$)$

EVENTS

Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$

end

Event ComputeStatesOfVSSinUnknowState ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStatesFollowingTTDStates

any

vss
vss1
vss2
vss3
vss4
newstateVSScomputed1

where

grd1: $vss = stateVSS^{-1}[\{UNKNOW\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$
grd3: $newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

Event ComputeStatesOfVSSinOccupiedState ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStatesFollowingTTDStates

any

vss
vss1
vss2
vss3
vss4
newstateVSScomputed1

where
grd1: $vss = stateVSS^{-1}[\{OCCUPIED\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$
grd3: $newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$
then
act1: $newstateVSScomputed := newstateVSScomputed1$
end

Event ComputeStatesOfVSSinAmbiguousState ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStatesFollowingTTDStates
any
vss
vss1
vss2
vss3
vss4
newstateVSScomputed1
where
grd1: $vss = stateVSS^{-1}[\{AMBIGUOUS\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$
grd3: $newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$
then
act1: $newstateVSScomputed := newstateVSScomputed1$
end

Event ComputeStatesOfVSSinFreeState ⟨ordinary⟩ $\widehat{=}$
refines ComputeVSSStatesFollowingTTDStates
any
vss
vss1
vss2
vss3
vss4
newstateVSScomputed1
where
grd1: $vss = stateVSS^{-1}[\{FREE\}]$
grd2: $partition(vss, vss1, vss2, vss3, vss4)$
grd3: $newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$
then
act1: $newstateVSScomputed := newstateVSScomputed1$
end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$
extends _connectTrain
any
tr
fr
re
integer
ttds
where
grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$
grd1: $tr \in TRAIN \setminus dom(connectedTrain)$
grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$

grd7: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \land ((fr \in ttd) \lor (integer = TRUE \land ((re \,..\, fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $integer = TRUE \Rightarrow (\forall tr1\cdot(tr1 \in dom(rear) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (re \,..\, fr) = \varnothing))$
grd9: $integer = TRUE \Rightarrow (\forall tr1\cdot(tr1 \in dom(front) \setminus dom(rear) \Rightarrow front(tr1) < re))$
grd10: $integer = FALSE \Rightarrow (\forall tr1\cdot(tr1 \in dom(rear) \Rightarrow fr < rear(tr1)))$
grd11: $integer = FALSE \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then

act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \lhd\!\!- \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \lhd\!\!- (ttds \times \{OCCUPIED\})$

end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$
extends _toggleTrainConnexionStatus

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \lhd\!\!- \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \lhd\!\!- \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$

end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$
extends MoveTrainFollowingItsMA

any

tr
len
$ttds$

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd\cdot(ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr) + len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr) + len \,..\, front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) \,..\, front(tr1)) \cap (rear(tr) + len \,..\, front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1\cdot((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd\cdot((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then

act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \lhd\!\!- \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \lhd\!\!- (ttds \times \{OCCUPIED\})$

end

Event _exitTrain ⟨ordinary⟩ $\widehat{=}$
extends _exitTrain

any

tr

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$

then

act1: $front := \{tr\} \lhd\!\!\!- front$

act2: $rear := (\{TRUE \mapsto \{tr\} \lhd rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $connectedTrain := \{tr\} \lhd connectedTrain$
act4: $MA := (\{TRUE \mapsto \{tr\} \lhd MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$
act5: $MAtemp := (\{TRUE \mapsto \{tr\} \lhd MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$

end

END

MACHINE M6
REFINES M5
SEES C2,C3
VARIABLES

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed
freeVssChangingtoFree
freeVssChangingtoUnknow
freeVssChangingtoOccupied
freeVssChangingtoAmbiguous
muteTimer
waitIntegrityTimer

INVARIANTS

inv6_1: $freeVssChangingtoFree \subseteq VSS$
inv6_2: $freeVssChangingtoUnknow \subseteq VSS$
inv6_3: $freeVssChangingtoOccupied \subseteq VSS$
inv6_4: $freeVssChangingtoAmbiguous \subseteq VSS$
inv6_5: $muteTimer \in TRAIN \rightarrow TIMER\_STATUS$
inv6_6: $waitIntegrityTimer \in TRAIN \rightarrow TIMER\_STATUS$

EVENTS

Initialisation

begin

act1: $connectedTrain := \varnothing$
act2: $front := \varnothing$
act3: $rear := \varnothing$
act4: $MA := \varnothing$
act5: $MAtemp := \varnothing$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOW\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOW\}$
act10: $freeVssChangingtoFree := \varnothing$
act11: $freeVssChangingtoUnknow := \varnothing$
act12: $freeVssChangingtoOccupied := \varnothing$
act13: $freeVssChangingtoAmbiguous := \varnothing$
act14: $muteTimer := TRAIN \times \{INACTIVE\}$
act15: $waitIntegrityTimer := TRAIN \times \{INACTIVE\}$

end

Event ComputeStatesOfVSSinFreeStateWhenTTDisFree ⟨ordinary⟩ $\widehat{=}$

any

vssTtdFree

where

grd1: $vssTtdFree \subseteq stateVSS^{-1}[\{FREE\}]$
grd2: $\forall vss\cdot(vss \in vssTtdFree \Rightarrow vss \subseteq union(stateTTD^{-1}[\{FREE\}]))$

then

act1: $freeVssChangingtoFree := freeVssChangingtoFree \cup vssTtdFree$

end

Event ComputeStatesOfVSSinFreeStateWhenTTDisOccupiedandNoTrainisLocatedandNoMAisIssued ⟨ordinary⟩ $\widehat{=}$

any

vssTtdOccupiedwithNoTrainAndNoMA

where

grd1: $vssTtdOccupiedwithNoTrainAndNoMA \subseteq stateVSS^{-1}[\{FREE\}]$

grd2: $\forall vss\cdot(vss \in vssTtdOccupiedwithNoTrainAndNoMA \Rightarrow vss \subseteq union(stateTTD^{-1}[\{OCCUPIED\}]))$

grd3: $\forall vss, p, q\cdot((vss \in vssTtdOccupiedwithNoTrainAndNoMA \wedge p..q \in TTD \wedge vss \subseteq p..q) \Rightarrow (\forall tr\cdot tr \in connectedTrain^{-1}[\{TRUE\}] \wedge tr \in dom(rear) \Rightarrow (front(tr) < p \vee rear(tr) > q)))$

grd4: $\forall vss, p, q\cdot((vss \in vssTtdOccupiedwithNoTrainAndNoMA \wedge p..q \in TTD \wedge vss \subseteq p..q) \Rightarrow (\forall tr\cdot tr \in connectedTrain^{-1}[\{TRUE\}] \wedge tr \notin dom(rear) \Rightarrow (front(tr) < p \vee front(tr) > q)))$

grd5: $\forall vss, ttd\cdot((vss \in vssTtdOccupiedwithNoTrainAndNoMA \wedge ttd \in TTD \wedge vss \subseteq ttd) \Rightarrow (union(ran(MA)) \cap ttd = \varnothing))$

then

act1: $freeVssChangingtoUnknow := freeVssChangingtoUnknow \cup vssTtdOccupiedwithNoTrainAndNoMA$

end

Event ComputeStatesOfVSSinFreeStateFollowing_1B ⟨ordinary⟩ $\widehat{=}$

any

vss_1B

where

grd1: $vss\_1B \subseteq stateVSS^{-1}[\{FREE\}]$

grd2: $\forall vss\cdot(vss \in vss\_1B \Rightarrow vss \subseteq union(stateTTD^{-1}[\{OCCUPIED\}]))$

grd3: $\forall vss\cdot(vss \in vss\_1B \Rightarrow \exists tr\cdot(tr \in dom(MA) \wedge vss \subseteq MA(tr) \wedge muteTimer(tr) = EXPIRED))$

grd4: $\forall vss, tr, p, q\cdot((vss \in vss\_1B \wedge tr \in dom(MA) \wedge vss \subseteq MA(tr) \wedge muteTimer(tr) = EXPIRED \wedge vss = p..q) \Rightarrow p \geq front(tr))$

then

act1: $freeVssChangingtoUnknow := freeVssChangingtoUnknow \cup vss\_1B$

end

Event FullComputeStatesOfVSSinFreeState ⟨ordinary⟩ $\widehat{=}$

refines ComputeStatesOfVSSinFreeState

any

vss
vss1
vss2
vss3
vss4
newstateVSScomputed1

where

grd1: $vss = stateVSS^{-1}[\{FREE\}]$

grd2: $partition(vss, vss1, vss2, vss3, vss4)$

grd3: $freeVssChangingtoFree \subseteq vss2$

lorsque toutes les transitions seront implementees, ceci deviendra une egalite

grd4: $freeVssChangingtoUnknow \subseteq vss4$

lorsque toutes les transitions seront implementees, ceci deviendra une egalite

grd5: $newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$

then

act1: $newstateVSScomputed := newstateVSScomputed1$

end

Event _connectTrain ⟨ordinary⟩ $\widehat{=}$

extends _connectTrain

any

tr
fr
re
integer
ttds

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \varnothing$

grd1: $tr \in TRAIN \setminus dom(connectedTrain)$

grd2: $fr \in WAY$
grd3: $integer \in BOOL$
grd4: $integer = TRUE \Rightarrow re \in WAY$
grd5: $re < fr$
grd6: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd7: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((fr \in ttd) \lor (integer = TRUE \land ((re \mathinner{..} fr) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd8: $integer = TRUE \Rightarrow (\forall tr1 \cdot (tr1 \in dom(rear) \Rightarrow (rear(tr1) \mathinner{..} front(tr1)) \cap (re \mathinner{..} fr) = \varnothing))$
grd9: $integer = TRUE \Rightarrow (\forall tr1 \cdot (tr1 \in dom(front) \setminus dom(rear) \Rightarrow front(tr1) < re))$
grd10: $integer = FALSE \Rightarrow (\forall tr1 \cdot (tr1 \in dom(rear) \Rightarrow fr < rear(tr1)))$
grd11: $integer = FALSE \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow fr \notin ttd))$

then

act1: $connectedTrain(tr) := TRUE$
act2: $front(tr) := fr$
act3: $rear := (\{TRUE \mapsto rear \mathbin{\lhd\!\!\!+} \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$
act4: $stateTTD := stateTTD \mathbin{\lhd\!\!\!+} (ttds \times \{OCCUPIED\})$
act5: $muteTimer(tr) := STARTED$
act6: $waitIntegrityTimer := (\{TRUE \mapsto waitIntegrityTimer \mathbin{\lhd\!\!\!+} \{tr \mapsto STARTED\}, FALSE \mapsto waitIntegrityTimer\})(integer)$

end

Event _toggleTrainConnexionStatus ⟨ordinary⟩ $\widehat{=}$
extends _toggleTrainConnexionStatus

any

tr

where

grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$

then

act1: $connectedTrain := (\{TRUE \mapsto connectedTrain \mathbin{\lhd\!\!\!+} \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \mathbin{\lhd\!\!\!+} \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$
act2: $muteTimer := (\{TRUE \mapsto muteTimer, FALSE \mapsto muteTimer \mathbin{\lhd\!\!\!+} \{tr \mapsto STARTED\}\})(bool(connectedTrain$ $TRUE))$

end

Event MoveTrainFollowingItsMA ⟨ordinary⟩ $\widehat{=}$
extends MoveTrainFollowingItsMA

any

tr
len
$ttds$

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in MA(tr)$
grd4: $ttds \subseteq stateTTD^{-1}[\{FREE\}]$
grd5: $\forall ttd \cdot (ttd \in stateTTD^{-1}[\{FREE\}] \land ((front(tr) + len \in ttd) \lor (tr \in dom(rear) \land ((rear(tr) + len \mathinner{..} front(tr) + len) \cap ttd \neq \varnothing))) \Rightarrow ttd \in ttds)$
grd6: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow (rear(tr1) \mathinner{..} front(tr1)) \cap (rear(tr) + len \mathinner{..} front(tr) + len) = \varnothing))$
grd7: $tr \in dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr) \Rightarrow front(tr1) < rear(tr) + len))$
grd8: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1 \cdot ((tr1 \in dom(rear) \land tr1 \neq tr) \Rightarrow front(tr) + len < rear(tr1)))$
grd9: $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd \cdot ((tr1 \in dom(front) \setminus dom(rear) \land tr1 \neq tr \land ttd \in TTD \land front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$

then

act1: $front(tr) := front(tr) + len$
act2: $rear := (\{TRUE \mapsto rear \mathbin{\lhd\!\!\!+} \{tr \mapsto rear(tr) + len\}, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $stateTTD := stateTTD \mathbin{\lhd\!\!\!+} (ttds \times \{OCCUPIED\})$

act4: $muteTimer(tr) := STARTED$
end
Event expireMuteTimer ⟨ordinary⟩ $\widehat{=}$
any
tr
where
grd0: $dom(connectedTrain) \neq \varnothing$
grd1: $tr \in dom(connectedTrain)$
grd2: $muteTimer(tr) = STARTED$
then
act0: $muteTimer(tr) := EXPIRED$
end
Event _exitTrain ⟨ordinary⟩ $\widehat{=}$
extends _exitTrain
any
tr
where
grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
then
act1: $front := \{tr\} \lhd\!\!\!- front$
act2: $rear := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$
act3: $connectedTrain := \{tr\} \lhd\!\!\!- connectedTrain$
act4: $MA := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$
act5: $MAtemp := (\{TRUE \mapsto \{tr\} \lhd\!\!\!- MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$
act6: $muteTimer(tr) := INACTIVE$
end
END