

Contents

CONTEXT C0	2
CONTEXT C2	3
CONTEXT C3	4
MACHINE M0	5
MACHINE M1	7
MACHINE M2	11
MACHINE M3	18
MACHINE M4	21
MACHINE M5	24
MACHINE M6	29

CONTEXT C0

SETS

TRAIN

CONSTANTS

a

b

WAY

AXIOMS

axiom1: $\{a, b\} \subseteq \mathbb{N}$

axiom2: $a < b$

axiom3: $WAY = a \dots b$

axiom4: $b - a \geq 20$

END

CONTEXT C2**EXTENDS** C0**SETS**

STATES

CONSTANTS

TTD

VSS

OCCUPIED

FREE

UNKNOWN

AMBIGUOUS

AXIOMS**axiom1:** $TTD \subseteq \mathbb{P}_1(WAY)$ **axiom2:** $union(TTD) = WAY$ **axiom3:** $inter(TTD) = \emptyset$ **axiom4:** $\forall ttd. (ttd \in TTD \Rightarrow (\exists p, q. (p .. q \subseteq WAY \wedge p < q \wedge ttd = p .. q)))$ **axiom5:** $VSS \subseteq \mathbb{P}_1(WAY)$ **axiom6:** $union(VSS) = WAY$ **axiom7:** $inter(VSS) = \emptyset$ **axiom8:** $\forall vss. (vss \in VSS \Rightarrow (\exists p, q, ttd. (ttd \in TTD \wedge p .. q \subseteq ttd \wedge p < q \wedge vss = p .. q)))$ **axiom9:** $partition(STATES, \{OCCUPIED\}, \{FREE\}, \{UNKNOWN\}, \{AMBIGUOUS\})$ **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: $connectedTrain \in TRAIN \leftrightarrow BOOL$
inv0.2: $front \in dom(connectedTrain) \rightarrow WAY$
inv0.3: $rear \in dom(connectedTrain) \rightarrow WAY$
inv0.4: $\forall tr. (tr \in dom(rear) \Rightarrow rear(tr) < front(tr))$

EVENTS**Initialisation****begin**

act1: $connectedTrain := \emptyset$
act2: $front := \emptyset$
act3: $rear := \emptyset$

end**Event** MoveTrainOnTrack $\langle \text{ordinary} \rangle \hat{=}$ **any**

tr
len
n_rear

where

grd1: $tr \in connectedTrain^{-1}[\{TRUE\}]$
grd2: $len \in \mathbb{N}_1$
grd3: $front(tr) + len \in WAY$
grd4: $tr \in dom(rear) \Rightarrow n_rear = rear \Leftarrow \{tr \mapsto rear(tr) + len\}$
grd5: $tr \notin dom(rear) \Rightarrow n_rear = rear$

then

act1: $front(tr) := front(tr) + len$
act2: $rear := n_rear$

end**Event** _connectTrain $\langle \text{ordinary} \rangle \hat{=}$ **any**

tr
fr
re
integer

where

grd0: $TRAIN \setminus dom(connectedTrain) \neq \emptyset$
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 $\langle \text{ordinary} \rangle \hat{=}$ **any**

tr

where

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

then

```

    act1:  $front := \{tr\} \triangleleft front$ 
    act2:  $rear := (\{TRUE \mapsto \{tr\} \triangleleft rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$ 
    act3:  $connectedTrain := \{tr\} \triangleleft connectedTrain$ 
end
Event _toggleTrainConnexionStatus  $\langle ordinary \rangle \hat{=}$ 
any
    tr
    integer
    re
where
    grd0:  $dom(connectedTrain) \neq \emptyset$ 
    grd1:  $tr \in dom(connectedTrain)$ 
    grd2:  $integer \in BOOL$ 
    grd3:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (re \in WAY \wedge re < front(tr))$ 
then
    act1:  $connectedTrain := (\{TRUE \mapsto connectedTrain \triangleleft \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \triangleleft \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$ 
    act2:  $rear := (\{TRUE \mapsto (\{TRUE \mapsto rear \triangleleft \{tr \mapsto re\}, FALSE \mapsto \{tr\} \triangleleft rear\})(bool(integer = TRUE)), FALSE \mapsto rear\})(bool(connectedTrain(tr) = FALSE))$ 
end
END

```

MACHINE M1**REFINES** M0**SEES** C0**VARIABLES**

connectedTrain

front

rear

MA

MAtemp

INVARIANTS**inv1.1:** $MA \in \text{dom}(\text{connectedTrain}) \leftrightarrow \mathbb{P}(WAY)$ **inv1.2:** $\forall tr. (tr \in \text{dom}(MA) \Rightarrow (\exists p, q. (p \dots q \subseteq WAY \wedge p \leq q \wedge MA(tr) = p \dots q)))$ **inv1.3:** $\forall tr. (tr \in \text{dom}(MA) \Rightarrow \text{front}(tr) \in MA(tr))$ **inv1.4:** $\forall tr. (tr \in \text{dom}(\text{rear}) \cap \text{dom}(MA) \Rightarrow \text{rear}(tr) \in MA(tr))$ **inv1.5:** $\forall tr1, tr2. ((\{tr1, tr2\} \subseteq \text{dom}(MA) \wedge tr1 \neq tr2) \Rightarrow MA(tr1) \cap MA(tr2) = \emptyset)$ **inv1.6:** $MAtemp \in \text{dom}(\text{connectedTrain}) \leftrightarrow \mathbb{P}(WAY)$ **inv1.7:** $\forall tr. (tr \in \text{dom}(MAtemp) \Rightarrow (\exists p, q. (p \dots q \subseteq WAY \wedge p \leq q \wedge MAtemp(tr) = p \dots q)))$ **SYSML/KAOS PROOF OBLIGATIONS****sysml_kaos_po.G1-Guard=>G-Guard:** (theorem)
$$\begin{aligned}
& \forall tr, p, q, len. ((\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (p \dots q \subseteq WAY \wedge p \leq q) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in WAY) \\
&) \Rightarrow \\
& (\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in WAY) \\
&)
\end{aligned}$$
remplacement de toute reference a MAtemp par $((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})$ **sysml_kaos_po.G1-Post=>G2-Guard:** (theorem)
$$\begin{aligned}
& \forall tr, p, q, len. ((\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (p \dots q \subseteq WAY \wedge p \leq q) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in p \dots q) \\
&) \Rightarrow \\
& (\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}] \cap \text{dom}((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})) \\
& \wedge (\text{front}(tr) \in ((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})(tr)) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in ((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})(tr)) \\
& \wedge (((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})(tr) \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge \text{front}(tr) + len \in ((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})(tr) \\
&)
\end{aligned}$$
remplacement de toute reference a MA par $((\{tr\} \triangleleft MA) \cup \{tr \mapsto MAtemp(tr)\})$ **sysml_kaos_po.G2-Post=>G3-Guard:** (theorem)
$$\begin{aligned}
& \forall tr, len. ((\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}] \cap \text{dom}(MAtemp)) \\
& \wedge (\text{front}(tr) \in MAtemp(tr))
\end{aligned}$$

$$\begin{aligned}
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in \text{MAtemp}(tr)) \\
& \wedge (\text{MAtemp}(tr) \cap \text{union}(\text{ran}(\{tr\} \triangleleft \text{MA})) = \emptyset) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge \text{front}(tr) + len \in \text{MAtemp}(tr) \\
&) \Rightarrow \\
& (\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}] \cap \text{dom}(((\{tr\} \triangleleft \text{MA}) \cup \{tr \mapsto \text{MAtemp}(tr)\}))) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in ((\{tr\} \triangleleft \text{MA}) \cup \{tr \mapsto \text{MAtemp}(tr)\})(tr)) \\
&)) \\
\text{sysml_kaos_po.G3-Post} \Rightarrow \text{G-Post: } \langle \text{theorem} \rangle \\
\forall tr, len. (\\
& (\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}] \cap \text{dom}(\text{MA})) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in \text{MA}(tr)) \\
&) \Rightarrow \\
& (\\
& (\text{front}(tr) + len = \text{front}(tr) + len) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow ((\{TRUE \mapsto \text{rear} \triangleleft \{tr \mapsto \text{rear}(tr) + len\}, FALSE \mapsto \text{rear}\}) = (\{TRUE \mapsto \text{rear} \triangleleft \{tr \mapsto \text{rear}(tr) + len\}, FALSE \mapsto \text{rear}\}))) \\
&) \\
&)
\end{aligned}$$

EVENTS

Initialisation

begin

act1: $\text{connectedTrain} := \emptyset$
act2: $\text{front} := \emptyset$
act3: $\text{rear} := \emptyset$
act4: $\text{MA} := \emptyset$
act5: $\text{MAtemp} := \emptyset$

end

Event ComputeTrainMA $\langle \text{ordinary} \rangle \hat{=}$

any

tr
 p
 q
 len

where

grd1: $tr \in \text{connectedTrain}^{-1}[\{TRUE\}]$
grd2: $p \dots q \subseteq \text{WAY} \wedge p \leq q$
grd3: $\text{front}(tr) \in p \dots q$
grd4: $tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q$
grd5: $p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft \text{MA})) = \emptyset$
grd6: $len \in \mathbb{N}_1$
grd7: $\text{front}(tr) + len \in \text{WAY}$

then

act1: $\text{MAtemp}(tr) := p \dots q$

end

Event AssignMAtoTrain $\langle \text{ordinary} \rangle \hat{=}$

any

tr
 len

where

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


```

    grd6:  $front(tr) + len \in MAtemp(tr)$ 
  then
    act1:  $MA(tr) := MAtemp(tr)$ 
  end
Event MoveTrainFollowingItsMA <ordinary>  $\hat{=}$ 
refines MoveTrainOnTrack
  any
    tr
    len
    n_rear
  where
    grd1:  $tr \in connectedTrain^{-1}[\{TRUE\}] \cap dom(MA)$ 
    grd2:  $len \in \mathbb{N}_1$ 
    grd3:  $front(tr) + len \in MA(tr)$ 
    grd4:  $tr \in dom(rear) \Rightarrow n\_rear = rear \triangleleft \{tr \mapsto rear(tr) + len\}$ 
    grd5:  $tr \notin dom(rear) \Rightarrow n\_rear = rear$ 
  then
    act1:  $front(tr) := front(tr) + len$ 
    act2:  $rear := n\_rear$ 
  end
Event _connectTrain <ordinary>  $\hat{=}$ 
extends _connectTrain
  any
    tr
    fr
    re
    integer
  where
    grd0:  $TRAIN \setminus dom(connectedTrain) \neq \emptyset$ 
    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:  $rear := (\{TRUE \mapsto rear \triangleleft \{tr \mapsto re\}, FALSE \mapsto rear\})(integer)$ 
  end
Event _toggleTrainConnexionStatus <ordinary>  $\hat{=}$ 
extends _toggleTrainConnexionStatus
  any
    tr
    integer
    re
  where
    grd0:  $dom(connectedTrain) \neq \emptyset$ 
    grd1:  $tr \in dom(connectedTrain)$ 
    grd2:  $integer \in BOOL$ 
    grd3:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (re \in WAY \wedge re < front(tr))$ 
    grd4:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (tr \in dom(MA) \wedge re \in MA(tr))$ 
  then
    act1:  $connectedTrain := (\{TRUE \mapsto connectedTrain \triangleleft \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \triangleleft \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$ 
    act2:  $rear := (\{TRUE \mapsto (\{TRUE \mapsto rear \triangleleft \{tr \mapsto re\}, FALSE \mapsto \{tr\} \triangleleft rear\})(bool(integer = TRUE)), FALSE \mapsto rear\})(bool(connectedTrain(tr) = FALSE))$ 
  end
Event _exitTrain <ordinary>  $\hat{=}$ 

```

```
extends _exitTrain
  any
    tr
  where
    grd1: tr ∈ connectedTrain-1[{TRUE}]
  then
    act1: front := {tr} ⋈ front
    act2: rear := ({TRUE ↦ {tr} ⋈ rear, FALSE ↦ rear})(bool(tr ∈ dom(rear)))
    act3: connectedTrain := {tr} ⋈ connectedTrain
    act4: MA := ({TRUE ↦ {tr} ⋈ MA, FALSE ↦ MA})(bool(tr ∈ dom(MA)))
    act5: MAtemp := ({TRUE ↦ {tr} ⋈ MAtemp, FALSE ↦ MAtemp})(bool(tr ∈ 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: $stateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}$

inv2.3: $\forall ttd, tr. ((tr \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr) \in ttd) \Rightarrow stateTTD(ttd) = OCCUPIED)$

inv2.4: $\forall ttd, tr. ((tr \in dom(rear) \wedge ttd \in TTD \wedge (rear(tr) .. front(tr)) \cap ttd \neq \emptyset) \Rightarrow stateTTD(ttd) = OCCUPIED)$

inv2.5: $\forall tr1, tr2. ((tr1 \in dom(rear) \wedge tr2 \in dom(rear) \wedge tr1 \neq tr2) \Rightarrow (rear(tr1) .. front(tr1)) \cap (rear(tr2) .. front(tr2)) = \emptyset)$

inv2.6: $\forall tr1, tr2, ttd. ((tr1 \in dom(rear) \wedge tr2 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr2 \wedge ttd \in TTD \wedge rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge front(tr2) \in ttd) \Rightarrow front(tr2) < rear(tr1))$

inv2.7: $\forall tr1, tr2, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr2 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr2 \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow front(tr2) \notin ttd)$

SYSML/KAOS PROOF OBLIGATIONS

sysml_kaos_po.G1-Guard=>G-Guard: *(theorem)*

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

sysml_kaos_po.G2-Guard=>G-Guard: *(theorem)*

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

$$\begin{aligned}
& \wedge (\text{union}(vsss) = p1 \dots q1) \\
& \wedge (p1 \geq \text{front}(tr)) \\
& \wedge (vsss1 \subseteq VSS) \\
& \wedge (\text{union}(vsss1) = p0 \dots (p1 - 1)) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0) \\
& \wedge (tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0) \\
& \wedge (p \dots q \subseteq \text{union}(vsss \cup vsss1)) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in WAY) \\
& \wedge (tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q) \\
&) \Rightarrow \\
& (\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (p \dots q \subseteq WAY \wedge p \leq q) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in WAY) \\
&) \\
& \text{sysml_kaos_po.G1-Post} \Rightarrow \text{G-Post: } \langle \text{theorem} \rangle \\
& \forall tr, p, q, len, ttds, ttds1, p0, p1, q1. ((\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (ttds \subseteq \text{stateTTD}^{-1}[\{FREE\}]) \\
& \wedge (\text{union}(ttds) = p1 \dots q1) \\
& \wedge (p1 \geq \text{front}(tr)) \\
& \wedge (ttds1 \subseteq TTD) \\
& \wedge (\text{union}(ttds1) = p0 \dots (p1 - 1)) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0) \\
& \wedge (tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0) \\
& \wedge (p \dots q \subseteq \text{union}(ttds \cup ttds1)) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (len \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + len \in WAY) \\
& \wedge (tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q) \\
&) \Rightarrow \\
& (\\
& (p \dots q = p \dots q) \\
&) \\
&) \\
& \text{sysml_kaos_po.G2-Post} \Rightarrow \text{G-Post: } \langle \text{theorem} \rangle \\
& \forall tr, p, q, len, vsss, vsss1, p0, p1, q1, \text{newstateVSS}. ((\\
& (\text{newstateVSS} \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}) \\
& \wedge (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (vsss \subseteq \text{newstateVSS}^{-1}[\{FREE\}]) \\
& \wedge (\text{union}(vsss) = p1 \dots q1) \\
& \wedge (p1 \geq \text{front}(tr)) \\
& \wedge (vsss1 \subseteq VSS) \\
& \wedge (\text{union}(vsss1) = p0 \dots (p1 - 1)) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0) \\
& \wedge (tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0) \\
& \wedge (p \dots q \subseteq \text{union}(vsss \cup vsss1)) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset)
\end{aligned}$$

```


$$\wedge (front(tr) \in p .. q)$$


$$\wedge (tr \in dom(rear) \Rightarrow rear(tr) \in p .. q)$$


$$\wedge (len \in \mathbb{N}_1)$$


$$\wedge (front(tr) + len \in WAY)$$


$$\wedge (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p .. q)$$


$$\Rightarrow$$


$$(\wedge (p .. q = p .. q))$$


$$\wedge$$


$$(\wedge (p .. q = p .. q))$$

remplacement de MAtemp par  $((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p .. q\})$ 
sysml_kaos_po.G1-Post=>not(G2-Guard): (theorem)

$$\forall tr, p, q, len, ttds, ttds1, p0, p1, q1. (($$


$$(tr \in connectedTrain^{-1}[\{TRUE\}])$$


$$\wedge (ttds \subseteq stateTTD^{-1}[\{FREE\}])$$


$$\wedge (union(ttds) = p1 .. q1)$$


$$\wedge (p1 \geq front(tr))$$


$$\wedge (ttds1 \subseteq TTD)$$


$$\wedge (union(ttds1) = p0 .. (p1 - 1))$$


$$\wedge (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$$


$$\wedge (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$$


$$\wedge (p .. q \subseteq union(ttds \cup ttds1))$$


$$\wedge (p .. q \cap union(ran(\{tr\} \triangleleft MA)) = \emptyset)$$


$$\wedge (front(tr) \in p .. q)$$


$$\wedge (tr \in dom(rear) \Rightarrow rear(tr) \in p .. q)$$


$$\wedge (len \in \mathbb{N}_1)$$


$$\wedge (front(tr) + len \in WAY)$$


$$\wedge (tr \notin dom(MAtemp) \vee MAtemp(tr) \neq p .. q)$$


$$\Rightarrow$$


$$\neg(\exists vsss, vsss1, newstateVSS. ($$


$$(newstateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\})$$


$$\wedge (tr \in connectedTrain^{-1}[\{TRUE\}])$$


$$\wedge (vsss \subseteq newstateVSS^{-1}[\{FREE\}])$$


$$\wedge (union(vsss) = p1 .. q1)$$


$$\wedge (p1 \geq front(tr))$$


$$\wedge (vsss1 \subseteq VSS)$$


$$\wedge (union(vsss1) = p0 .. (p1 - 1))$$


$$\wedge (tr \in dom(rear) \Rightarrow rear(tr) \geq p0)$$


$$\wedge (tr \notin dom(rear) \Rightarrow front(tr) \geq p0)$$


$$\wedge (p .. q \subseteq union(vsss \cup vsss1))$$


$$\wedge (p .. q \cap union(ran(\{tr\} \triangleleft MA)) = \emptyset)$$


$$\wedge (front(tr) \in p .. q)$$


$$\wedge (tr \in dom(rear) \Rightarrow rear(tr) \in p .. q)$$


$$\wedge (len \in \mathbb{N}_1)$$


$$\wedge (front(tr) + len \in WAY)$$


$$\wedge (tr \notin dom(((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p .. q\})) \vee ((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p .. q\})(tr) \neq p .. q)$$


$$\wedge$$


$$\wedge$$


$$\wedge$$

remplacement de MAtemp par  $((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p .. q\})$ 
sysml_kaos_po.G2-Post=>not(G1-Guard): (theorem)

$$\forall tr, p, q, len, vsss, vsss1, p0, p1, q1, newstateVSS. (($$


$$(newstateVSS \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\})$$


$$\wedge (tr \in connectedTrain^{-1}[\{TRUE\}])$$


$$\wedge (vsss \subseteq newstateVSS^{-1}[\{FREE\}])$$


$$\wedge (union(vsss) = p1 .. q1)$$


$$\wedge (p1 \geq front(tr))$$


$$\wedge (vsss1 \subseteq VSS)$$


$$\wedge (union(vsss1) = p0 .. (p1 - 1))$$


```

$$\begin{aligned}
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0) \\
& \wedge (tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0) \\
& \wedge (p \dots q \subseteq \text{union}(\text{vsss} \cup \text{vsss1})) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (\text{len} \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + \text{len} \in \text{WAY}) \\
& \wedge (tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q) \\
&) \Rightarrow \\
& \neg(\exists ttds, ttds1. (\\
& (tr \in \text{connectedTrain}^{-1}[\{TRUE\}]) \\
& \wedge (ttds \subseteq \text{stateTTD}^{-1}[\{FREE\}]) \\
& \wedge (\text{union}(ttds) = p1 \dots q1) \\
& \wedge (p1 \geq \text{front}(tr)) \\
& \wedge (ttds1 \subseteq TTD) \\
& \wedge (\text{union}(ttds1) = p0 \dots (p1 - 1)) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0) \\
& \wedge (tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0) \\
& \wedge (p \dots q \subseteq \text{union}(ttds \cup ttds1)) \\
& \wedge (p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset) \\
& \wedge (\text{front}(tr) \in p \dots q) \\
& \wedge (tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q) \\
& \wedge (\text{len} \in \mathbb{N}_1) \\
& \wedge (\text{front}(tr) + \text{len} \in \text{WAY}) \\
& \wedge (tr \notin \text{dom}(((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})) \vee ((\{tr\} \triangleleft MAtemp) \cup \{tr \mapsto p \dots q\})(tr) \neq p \dots q) \\
&) \\
&) \\
&)
\end{aligned}$$

EVENTS

Initialisation

begin

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

end

Event $\text{ComputeTrainMAFollowingTTDStates}$ $\langle \text{ordinary} \rangle \hat{=}$

refines ComputeTrainMA

any

tr
 $ttds$
 p
 q
 $ttds1$
 $p0$
 $p1$
 $q1$

len $ttds1$ designe l'ensemble des ttd sur lesquels le train est susceptible de se trouver

where

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

```

    grd7:  $tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0$ 
    grd8:  $tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0$ 
    grd9:  $p \dots q \subseteq \text{union}(\text{tt ds} \cup \text{tt ds1})$ 
    grd10:  $p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset$ 
    grd11:  $\text{front}(tr) \in p \dots q$ 
    grd12:  $tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q$ 
    grd13:  $\text{len} \in \mathbb{N}_1$ 
    grd14:  $\text{front}(tr) + \text{len} \in WAY$ 
    grd15:  $tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q$ 
  then
    act1:  $MAtemp(tr) := p \dots q$ 
  end
Event ComputeTrainMAFollowingVSSStates <ordinary>  $\hat{=}$ 
refines ComputeTrainMA
  any
    tr
    vsss
    p
    q
    vsss1
    p0
    p1
    q1
    newstateVSS
    len vsss1 designe l'ensemble des vss sur lesquels le train est susceptible de se trouver
  where
    grd0:  $\text{newstateVSS} \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}$ 
    grd1:  $tr \in \text{connectedTrain}^{-1}[\{TRUE\}]$ 
    grd2:  $vsss \subseteq \text{newstateVSS}^{-1}[\{FREE\}]$ 
    grd3:  $\text{union}(vsss) = p1 \dots q1$ 
    grd4:  $p1 \geq \text{front}(tr)$ 
    grd5:  $vsss1 \subseteq VSS$ 
    grd6:  $\text{union}(vsss1) = p0 \dots (p1 - 1)$ 
    grd7:  $tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \geq p0$ 
    grd8:  $tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0$ 
    grd9:  $p \dots q \subseteq \text{union}(vsss \cup vsss1)$ 
    grd10:  $p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset$ 
    grd11:  $\text{front}(tr) \in p \dots q$ 
    grd12:  $tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q$ 
    grd13:  $\text{len} \in \mathbb{N}_1$ 
    grd14:  $\text{front}(tr) + \text{len} \in WAY$ 
    grd15:  $tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q$ 
  then
    act1:  $MAtemp(tr) := p \dots q$ 
    act2:  $\text{stateVSS} := \text{newstateVSS}$ 
  end
Event MoveTrainFollowingItsMA <ordinary>  $\hat{=}$ 
extends MoveTrainFollowingItsMA
  any
    tr
    len
    n_rear
    tt ds
  where
    grd1:  $tr \in \text{connectedTrain}^{-1}[\{TRUE\}] \cap \text{dom}(MA)$ 
    grd2:  $\text{len} \in \mathbb{N}_1$ 
    grd3:  $\text{front}(tr) + \text{len} \in MA(tr)$ 
    grd4:  $tr \in \text{dom}(\text{rear}) \Rightarrow n\_rear = \text{rear} \triangleleft \{tr \mapsto \text{rear}(tr) + \text{len}\}$ 
    grd5:  $tr \notin \text{dom}(\text{rear}) \Rightarrow n\_rear = \text{rear}$ 

```

```

    grd6:  $tlds \subseteq stateTTD^{-1}[\{FREE\}]$ 
    grd7:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((front(tr) + len \in ttd) \vee (tr \in dom(n\_rear) \wedge ((n\_rear(tr) .. front(tr) + len) \cap ttd \neq \emptyset))) \Rightarrow ttd \in tlds)$ 
    grd8:  $tr \in dom(n\_rear) \Rightarrow (\forall tr1. ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) .. front(tr1)) \cap (n\_rear(tr) .. front(tr) + len) = \emptyset))$ 
    grd9:  $tr \in dom(n\_rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge n\_rear(tr) .. (front(tr) + len) \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < n\_rear(tr)))$ 
    grd10:  $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge (front(tr) + len) \in ttd) \Rightarrow front(tr) + len < rear(tr1)))$ 
    grd11:  $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$ 

  then
    act1:  $front(tr) := front(tr) + len$ 
    act2:  $rear := n\_rear$ 
    act3:  $stateTTD := stateTTD \Leftarrow (tlds \times \{OCCUPIED\})$ 
  end

Event _connectTrain <ordinary>  $\hat{=}$ 
extends _connectTrain
any
  tr
  fr
  re
  integer
  tlds
where
  grd0:  $TRAIN \setminus dom(connectedTrain) \neq \emptyset$ 
  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:  $tlds \subseteq stateTTD^{-1}[\{FREE\}]$ 
  grd7:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (integer = TRUE \wedge ((re .. fr) \cap ttd \neq \emptyset))) \Rightarrow ttd \in tlds)$ 
  grd8:  $integer = TRUE \Rightarrow (\forall tr1. (tr1 \in dom(rear) \Rightarrow (rear(tr1) .. front(tr1)) \cap (re .. fr) = \emptyset))$ 
  grd9:  $integer = TRUE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge re .. fr \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < re))$ 
  grd10:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge ttd \in TTD \wedge rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge fr \in ttd) \Rightarrow fr < rear(tr1)))$ 
  grd11:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$ 

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

Event _toggleTrainConnexionStatus <ordinary>  $\hat{=}$ 
extends _toggleTrainConnexionStatus
any
  tr
  integer
  re
  tlds
where
  grd0:  $dom(connectedTrain) \neq \emptyset$ 
  grd1:  $tr \in dom(connectedTrain)$ 
  grd2:  $integer \in BOOL$ 
  grd3:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (re \in WAY \wedge re < front(tr))$ 

```



```

grd4:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (tr \in dom(MA) \wedge re \in MA(tr))$ 
grd5:  $ttds \subseteq stateTTD^{-1}[\{FREE\}]$ 
grd6:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge (connectedTrain(tr) = FALSE \wedge integer = TRUE \wedge$ 
 $((re \dots front(tr)) \cap ttd \neq \emptyset)) \Rightarrow ttd \in ttds)$ 
grd7:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (\forall tr1. ((tr1 \in dom(rear) \wedge tr1 \neq$ 
 $tr) \Rightarrow (rear(tr1) \dots front(tr1)) \cap (re \dots front(tr)) = \emptyset))$ 
grd8:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus$ 
 $dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge re \dots front(tr) \cap ttd \neq \emptyset \wedge front(tr1) \in ttd \Rightarrow front(tr1) < re))$ 
grd9:  $(connectedTrain(tr) = FALSE \wedge integer = FALSE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge tr1 \neq$ 
 $tr \wedge ttd \in TTD \wedge rear(tr1) \dots front(tr1) \cap ttd \neq \emptyset \wedge front(tr) \in ttd \Rightarrow front(tr) < rear(tr1)))$ 
grd10:  $(connectedTrain(tr) = FALSE \wedge integer = FALSE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus$ 
 $dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD) \Rightarrow ((front(tr1) \in ttd \Rightarrow front(tr) \notin ttd) \wedge (front(tr) \in$ 
 $ttd \Rightarrow front(tr1) \notin ttd)))$ 

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

end

Event _exitTrain ⟨ordinary⟩  $\hat{=}$ 
extends _exitTrain
any
|
where
grd1:  $tr \in connectedTrain^{-1}[\{TRUE\}]$ 
then
act1:  $front := \{tr\} \Leftarrow front$ 
act2:  $rear := (\{TRUE \mapsto \{tr\} \Leftarrow rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$ 
act3:  $connectedTrain := \{tr\} \Leftarrow connectedTrain$ 
act4:  $MA := (\{TRUE \mapsto \{tr\} \Leftarrow MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$ 
act5:  $MAtemp := (\{TRUE \mapsto \{tr\} \Leftarrow MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$ 

end
END


```

MACHINE M3**REFINES** M2**SEES** C0,C2**VARIABLES**

connectedTrain
front
rear
MA
MAtemp
stateTTD
stateVSS
newstateVSScomputed

INVARIANTS

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

EVENTS**Initialisation****begin**

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

end**Event** ComputeVSSStates $\langle \text{ordinary} \rangle \hat{=}$ **any**

$\text{newstateVSScomputed1}$

where

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

then

act1: $\text{newstateVSScomputed} := \text{newstateVSScomputed1}$

end**Event** ComputeTrainMAUsingVSSStates $\langle \text{ordinary} \rangle \hat{=}$ **refines** ComputeTrainMAFollowingVSSStates**any**

tr

vsss

p

q

vsss1

p0

p1

q1

len

newstateVSS vsss1 designe l'ensemble des vss sur lesquels le train est susceptible de se trouver

where

grd0: $\text{newstateVSS} = \text{newstateVSScomputed}$

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

grd2: $\text{vsss} \subseteq \text{newstateVSS}^{-1}[\{FREE\}]$

grd3: $\text{union}(\text{vsss}) = p1 \dots q1$

grd4: $p1 \geq \text{front}(\text{tr})$

grd5: $\text{vsss1} \subseteq VSS$

grd6: $\text{union}(\text{vsss1}) = p0 \dots (p1 - 1)$

grd7: $\text{tr} \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(\text{tr}) \geq p0$

```

    grd8:  $tr \notin \text{dom}(\text{rear}) \Rightarrow \text{front}(tr) \geq p0$ 
    grd9:  $p \dots q \subseteq \text{union}(vsss \cup vsss1)$ 
    grd10:  $p \dots q \cap \text{union}(\text{ran}(\{tr\} \triangleleft MA)) = \emptyset$ 
    grd11:  $\text{front}(tr) \in p \dots q$ 
    grd12:  $tr \in \text{dom}(\text{rear}) \Rightarrow \text{rear}(tr) \in p \dots q$ 
    grd13:  $\text{len} \in \mathbb{N}_1$ 
    grd14:  $\text{front}(tr) + \text{len} \in WAY$ 
    grd15:  $tr \notin \text{dom}(MAtemp) \vee MAtemp(tr) \neq p \dots q$ 
  then
    act1:  $MAtemp(tr) := p \dots q$ 
    act2:  $\text{stateVSS} := \text{newstateVSS}$ 
  end
Event _connectTrain <ordinary>  $\hat{=}$ 
extends _connectTrain
  any
    tr
    fr
    re
    integer
    ttds
  where
    grd0:  $TRAIN \setminus \text{dom}(\text{connectedTrain}) \neq \emptyset$ 
    grd1:  $tr \in TRAIN \setminus \text{dom}(\text{connectedTrain})$ 
    grd2:  $fr \in WAY$ 
    grd3:  $\text{integer} \in \text{BOOL}$ 
    grd4:  $\text{integer} = \text{TRUE} \Rightarrow re \in WAY$ 
    grd5:  $re < fr$ 
    grd6:  $ttds \subseteq \text{stateTTD}^{-1}[\{FREE\}]$ 
    grd7:  $\forall ttd. (ttd \in \text{stateTTD}^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (\text{integer} = \text{TRUE} \wedge ((re \dots fr) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
    grd8:  $\text{integer} = \text{TRUE} \Rightarrow (\forall tr1. (tr1 \in \text{dom}(\text{rear}) \Rightarrow (\text{rear}(tr1) \dots \text{front}(tr1)) \cap (re \dots fr) = \emptyset))$ 
    grd9:  $\text{integer} = \text{TRUE} \Rightarrow (\forall tr1, ttd. ((tr1 \in \text{dom}(\text{front}) \setminus \text{dom}(\text{rear}) \wedge ttd \in \text{TTD} \wedge re \dots fr \cap ttd \neq \emptyset \wedge \text{front}(tr1) \in ttd) \Rightarrow \text{front}(tr1) < re))$ 
    grd10:  $\text{integer} = \text{FALSE} \Rightarrow (\forall tr1, ttd. ((tr1 \in \text{dom}(\text{rear}) \wedge ttd \in \text{TTD} \wedge \text{rear}(tr1) \dots \text{front}(tr1) \cap ttd \neq \emptyset \wedge fr \in ttd) \Rightarrow fr < \text{rear}(tr1)))$ 
    grd11:  $\text{integer} = \text{FALSE} \Rightarrow (\forall tr1, ttd. ((tr1 \in \text{dom}(\text{front}) \setminus \text{dom}(\text{rear}) \wedge ttd \in \text{TTD} \wedge \text{front}(tr1) \in ttd) \Rightarrow fr \notin ttd))$ 
  then
    act1:  $\text{connectedTrain}(tr) := \text{TRUE}$ 
    act2:  $\text{front}(tr) := fr$ 
    act3:  $\text{rear} := (\{ \text{TRUE} \mapsto \text{rear} \triangleleft \{tr \mapsto re\}, \text{FALSE} \mapsto \text{rear} \})(\text{integer})$ 
    act4:  $\text{stateTTD} := \text{stateTTD} \triangleleft (ttds \times \{OCCUPIED\})$ 
  end
Event _toggleTrainConnexionStatus <ordinary>  $\hat{=}$ 
extends _toggleTrainConnexionStatus
  any
    tr
    integer
    re
    ttds
  where
    grd0:  $\text{dom}(\text{connectedTrain}) \neq \emptyset$ 
    grd1:  $tr \in \text{dom}(\text{connectedTrain})$ 
    grd2:  $\text{integer} \in \text{BOOL}$ 
    grd3:  $(\text{connectedTrain}(tr) = \text{FALSE} \wedge \text{integer} = \text{TRUE}) \Rightarrow (re \in WAY \wedge re < \text{front}(tr))$ 
    grd4:  $(\text{connectedTrain}(tr) = \text{FALSE} \wedge \text{integer} = \text{TRUE}) \Rightarrow (tr \in \text{dom}(MA) \wedge re \in MA(tr))$ 
    grd5:  $ttds \subseteq \text{stateTTD}^{-1}[\{FREE\}]$ 
    grd6:  $\forall ttd. (ttd \in \text{stateTTD}^{-1}[\{FREE\}] \wedge (\text{connectedTrain}(tr) = \text{FALSE} \wedge \text{integer} = \text{TRUE} \wedge ((re \dots \text{front}(tr)) \cap ttd \neq \emptyset)) \Rightarrow ttd \in ttds)$ 

```

```

    grd7:  (connectedTrain(tr) = FALSE ∧ integer = TRUE) ⇒ (∀tr1.((tr1 ∈ dom(rear) ∧ tr1 ≠
      tr) ⇒ (rear(tr1) .. front(tr1)) ∩ (re .. front(tr)) = ∅))
    grd8:  (connectedTrain(tr) = FALSE ∧ integer = TRUE) ⇒ (∀tr1, ttd.((tr1 ∈ dom(front) \
      dom(rear) ∧ tr1 ≠ tr ∧ ttd ∈ TTD ∧ re .. front(tr) ∩ ttd ≠ ∅ ∧ front(tr1) ∈ ttd ⇒ front(tr1) < re))
    grd9:  (connectedTrain(tr) = FALSE ∧ integer = FALSE) ⇒ (∀tr1, ttd.((tr1 ∈ dom(rear) ∧ tr1 ≠
      tr ∧ ttd ∈ TTD ∧ rear(tr1) .. front(tr1) ∩ ttd ≠ ∅ ∧ front(tr) ∈ ttd ⇒ front(tr) < rear(tr1)))
    grd10: (connectedTrain(tr) = FALSE ∧ integer = FALSE) ⇒ (∀tr1, ttd.((tr1 ∈ dom(front) \
      dom(rear) ∧ tr1 ≠ tr ∧ ttd ∈ TTD) ⇒ ((front(tr1) ∈ ttd ⇒ front(tr) ∉ ttd) ∧ (front(tr) ∈
      ttd ⇒ front(tr1) ∉ ttd)))
  then
    act1: connectedTrain := ({TRUE ↦ connectedTrain ⇐ {tr ↦ FALSE}, FALSE ↦ connectedTrain ⇐
      {tr ↦ TRUE}})(bool(connectedTrain(tr) = TRUE))
    act2: rear := ({TRUE ↦ ({TRUE ↦ rear ⇐ {tr ↦ re}, FALSE ↦ {tr} ⇐ rear})(bool(integer =
      TRUE)), FALSE ↦ rear})(bool(connectedTrain(tr) = FALSE))
    act3: stateTTD := stateTTD ⇐ (ttds × {OCCUPIED})
  end
Event MoveTrainFollowingItsMA ⟨ordinary⟩ ≐
extends MoveTrainFollowingItsMA
  any
    tr
    len
    n_rear
    ttds
  where
    grd1:  tr ∈ connectedTrain-1[{TRUE}] ∩ dom(MA)
    grd2:  len ∈ ℕ1
    grd3:  front(tr) + len ∈ MA(tr)
    grd4:  tr ∈ dom(rear) ⇒ n_rear = rear ⇐ {tr ↦ rear(tr) + len}
    grd5:  tr ∉ dom(rear) ⇒ n_rear = rear
    grd6:  ttds ⊆ stateTTD-1[{FREE}]
    grd7:  ∀ttd.(ttd ∈ stateTTD-1[{FREE}] ∧ ((front(tr) + len ∈ ttd) ∨ (tr ∈ dom(n_rear) ∧
      ((n_rear(tr) .. front(tr) + len) ∩ ttd ≠ ∅))) ⇒ ttd ∈ ttds)
    grd8:  tr ∈ dom(n_rear) ⇒ (∀tr1.((tr1 ∈ dom(rear) ∧ tr1 ≠ tr) ⇒ (rear(tr1) .. front(tr1)) ∩
      (n_rear(tr) .. front(tr) + len) = ∅))
    grd9:  tr ∈ dom(n_rear) ⇒ (∀tr1, ttd.((tr1 ∈ dom(front) \ dom(rear) ∧ tr1 ≠ tr ∧ ttd ∈ TTD ∧
      n_rear(tr) .. (front(tr) + len) ∩ ttd ≠ ∅ ∧ front(tr1) ∈ ttd ⇒ front(tr1) < n_rear(tr)))
    grd10: tr ∈ dom(front) \ dom(rear) ⇒ (∀tr1, ttd.((tr1 ∈ dom(rear) ∧ tr1 ≠ tr ∧ ttd ∈ TTD ∧
      rear(tr1) .. front(tr1) ∩ ttd ≠ ∅ ∧ (front(tr) + len) ∈ ttd ⇒ front(tr) + len < rear(tr1)))
    grd11: tr ∈ dom(front) \ dom(rear) ⇒ (∀tr1, ttd.((tr1 ∈ dom(front) \ dom(rear) ∧ tr1 ≠ tr ∧ ttd ∈
      TTD ∧ front(tr1) ∈ ttd ⇒ front(tr) + len ∉ ttd))
  then
    act1: front(tr) := front(tr) + len
    act2: rear := n_rear
    act3: stateTTD := stateTTD ⇐ (ttds × {OCCUPIED})
  end
Event _exitTrain ⟨ordinary⟩ ≐
extends _exitTrain
  any
    tr
  where
    grd1:  tr ∈ connectedTrain-1[{TRUE}]
  then
    act1: front := {tr} ⇐ front
    act2: rear := ({TRUE ↦ {tr} ⇐ rear, FALSE ↦ rear})(bool(tr ∈ dom(rear)))
    act3: connectedTrain := {tr} ⇐ connectedTrain
    act4: MA := ({TRUE ↦ {tr} ⇐ MA, FALSE ↦ MA})(bool(tr ∈ dom(MA)))
    act5: MAtemp := ({TRUE ↦ {tr} ⇐ MAtemp, FALSE ↦ MAtemp})(bool(tr ∈ dom(MAtemp)))
  end
END

```

MACHINE M4**REFINES** M3**SEES** C0,C2**VARIABLES**

connectedTrain
 front
 rear
 MA
 MAtemp
 stateTTD
 stateVSS
 newstateVSScomputed

EVENTS**Initialisation****begin**

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

end**Event** ComputeVSSStatesFollowingTTDStates *ordinary* $\hat{=}$ **refines** ComputeVSSStates**any**

newstateVSScomputed1

where*grd0*: $newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}$ **then***act1*: $newstateVSScomputed := newstateVSScomputed1$ **end****Event** ComputeVSSStateswoTTDStates *ordinary* $\hat{=}$ **refines** ComputeVSSStates**any**

newstateVSScomputed1

where*grd0*: $newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}$ **then***act1*: $newstateVSScomputed := newstateVSScomputed1$ **end****Event** _connectTrain *ordinary* $\hat{=}$ **extends** _connectTrain**any***tr**fr**re**integer**ttds***where***grd0*: $TRAIN \setminus dom(connectedTrain) \neq \emptyset$ *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:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (integer = TRUE \wedge ((re .. fr) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
    grd8:  $integer = TRUE \Rightarrow (\forall tr1. (tr1 \in dom(rear) \Rightarrow (rear(tr1) .. front(tr1)) \cap (re .. fr) = \emptyset))$ 
    grd9:  $integer = TRUE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge re .. fr \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < re))$ 
    grd10:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge ttd \in TTD \wedge rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge fr \in ttd) \Rightarrow fr < rear(tr1)))$ 
    grd11:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$ 
  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>  $\hat{=}$ 
extends _toggleTrainConnexionStatus
  any
    tr
    integer
    re
    ttds
  where
    grd0:  $dom(connectedTrain) \neq \emptyset$ 
    grd1:  $tr \in dom(connectedTrain)$ 
    grd2:  $integer \in BOOL$ 
    grd3:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (re \in WAY \wedge re < front(tr))$ 
    grd4:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (tr \in dom(MA) \wedge re \in MA(tr))$ 
    grd5:  $ttds \subseteq stateTTD^{-1}[\{FREE\}]$ 
    grd6:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge (connectedTrain(tr) = FALSE \wedge integer = TRUE \wedge ((re .. front(tr)) \cap ttd \neq \emptyset)) \Rightarrow ttd \in ttds)$ 
    grd7:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (\forall tr1. ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) .. front(tr1)) \cap (re .. front(tr)) = \emptyset))$ 
    grd8:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge re .. front(tr) \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < re))$ 
    grd9:  $(connectedTrain(tr) = FALSE \wedge integer = FALSE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge front(tr) \in ttd) \Rightarrow front(tr) < rear(tr1)))$ 
    grd10:  $(connectedTrain(tr) = FALSE \wedge integer = FALSE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD) \Rightarrow ((front(tr1) \in ttd \Rightarrow front(tr) \notin ttd) \wedge (front(tr) \in ttd \Rightarrow front(tr1) \notin ttd))))$ 
  then
    act1:  $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$ 
    act2:  $rear := (\{TRUE \mapsto (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto \{tr\} \Leftarrow rear\})(bool(integer = TRUE)), FALSE \mapsto rear\})(bool(connectedTrain(tr) = FALSE))$ 
    act3:  $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$ 
  end
Event MoveTrainFollowingItsMA <ordinary>  $\hat{=}$ 
extends MoveTrainFollowingItsMA
  any
    tr
    len
    n_rear
    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:  $tr \in dom(rear) \Rightarrow n\_rear = rear \Leftarrow \{tr \mapsto rear(tr) + len\}$ 
    grd5:  $tr \notin dom(rear) \Rightarrow n\_rear = rear$ 
    grd6:  $ttds \subseteq stateTTD^{-1}[\{FREE\}]$ 
    grd7:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((front(tr) + len \in ttd) \vee (tr \in dom(n\_rear) \wedge$ 
         $((n\_rear(tr) .. front(tr) + len) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
    grd8:  $tr \in dom(n\_rear) \Rightarrow (\forall tr1. ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) .. front(tr1)) \cap$ 
         $(n\_rear(tr) .. front(tr) + len) = \emptyset))$ 
    grd9:  $tr \in dom(n\_rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge$ 
         $n\_rear(tr) .. (front(tr) + len) \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < n\_rear(tr)))$ 
    grd10:  $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge$ 
         $rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge (front(tr) + len) \in ttd) \Rightarrow front(tr) + len < rear(tr1)))$ 
    grd11:  $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in$ 
         $TTD \wedge front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$ 
  then
    act1:  $front(tr) := front(tr) + len$ 
    act2:  $rear := n\_rear$ 
    act3:  $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$ 
  end
Event _exitTrain <ordinary>  $\hat{=}$ 
extends _exitTrain
  any
     $tr$ 
  where
    grd1:  $tr \in connectedTrain^{-1}[\{TRUE\}]$ 
  then
    act1:  $front := \{tr\} \Leftarrow front$ 
    act2:  $rear := (\{TRUE \mapsto \{tr\} \Leftarrow rear, FALSE \mapsto rear\})(bool(tr \in dom(rear)))$ 
    act3:  $connectedTrain := \{tr\} \Leftarrow connectedTrain$ 
    act4:  $MA := (\{TRUE \mapsto \{tr\} \Leftarrow MA, FALSE \mapsto MA\})(bool(tr \in dom(MA)))$ 
    act5:  $MAtemp := (\{TRUE \mapsto \{tr\} \Leftarrow MAtemp, FALSE \mapsto MAtemp\})(bool(tr \in dom(MAtemp)))$ 
  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)*

$$\begin{aligned} & \forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1. ((\\ & (vss = stateVSS^{-1}[\{UNKNOWN\}]) \\ & \wedge (partition(vss, vss1, vss2, vss3, vss4)) \\ & \wedge (newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))) \\ &) \Rightarrow \\ & (\\ & (newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}) \\ &)) \end{aligned}$$
sysml_kaos_po.G2-Guard=>G-Guard: *(theorem)*

$$\begin{aligned} & \forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1. ((\\ & (vss = stateVSS^{-1}[\{OCCUPIED\}]) \\ & \wedge (partition(vss, vss1, vss2, vss3, vss4)) \\ & \wedge (newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))) \\ &) \Rightarrow \\ & (\\ & (newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}) \\ &)) \end{aligned}$$
sysml_kaos_po.G3-Guard=>G-Guard: *(theorem)*

$$\begin{aligned} & \forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1. ((\\ & (vss = stateVSS^{-1}[\{AMBIGUOUS\}]) \\ & \wedge (partition(vss, vss1, vss2, vss3, vss4)) \\ & \wedge (newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))) \\ &) \Rightarrow \\ & (\\ & (newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}) \\ &)) \end{aligned}$$
sysml_kaos_po.G4-Guard=>G-Guard: *(theorem)*

$$\begin{aligned} & \forall vss, vss1, vss2, vss3, vss4, newstateVSScomputed1. ((\\ & (vss = stateVSS^{-1}[\{FREE\}]) \\ & \wedge (partition(vss, vss1, vss2, vss3, vss4)) \\ & \wedge (newstateVSScomputed1 = stateVSS \Leftarrow ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup (vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))) \\ &) \Rightarrow \\ & (\\ & (newstateVSScomputed1 \in VSS \rightarrow \{OCCUPIED, FREE, UNKNOWN, AMBIGUOUS\}) \\ &)) \end{aligned}$$
sysml_kaos_po.G1-G2-G3-G4-G5-Post=>G-Post : *(theorem)*

$$\begin{aligned} & \forall vss1, vss11, vss12, vss13, vss14, vss2, vss21, vss22, vss23, vss24, vss3, vss31, vss32, vss33, vss34, vss4, vss41, vss42, vss43, \\ & (\\ & (vss1 = stateVSS^{-1}[\{UNKNOWN\}]) \end{aligned}$$

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

EVENTS

Initialisation

begin

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

end

Event ComputeStatesOfVSSinUnknowState $\langle \text{ordinary} \rangle \hat{=}$

refines ComputeVSSStatesFollowingTTDStates

any

vss
vss1
vss2
vss3
vss4
newstateVSScomputed1

where

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

then

act1: *newstateVSScomputed* := *newstateVSScomputed1*

end

Event ComputeStatesOfVSSinOccupiedState $\langle \text{ordinary} \rangle \hat{=}$

refines ComputeVSSStatesFollowingTTDStates

any

vss
vss1
vss2
vss3
vss4
newstateVSScomputed1

```

where
  grd1:  $vss = stateVSS^{-1}[\{OCCUPIED\}]$ 
  grd2:  $partition(vss, vss1, vss2, vss3, vss4)$ 
  grd3:  $newstateVSScomputed1 = stateVSS \triangleleft ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup$ 
     $(vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))$ 
then
  act1:  $newstateVSScomputed := newstateVSScomputed1$ 
end
Event ComputeStatesOfVSSinAmbiguousState  $\langle ordinary \rangle \hat{=}$ 
refines ComputeVSSStatesFollowingTTDDStates
any
  vss
  vss1
  vss2
  vss3
  vss4
  newstateVSScomputed1
where
  grd1:  $vss = stateVSS^{-1}[\{AMBIGUOUS\}]$ 
  grd2:  $partition(vss, vss1, vss2, vss3, vss4)$ 
  grd3:  $newstateVSScomputed1 = stateVSS \triangleleft ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup$ 
     $(vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))$ 
then
  act1:  $newstateVSScomputed := newstateVSScomputed1$ 
end
Event ComputeStatesOfVSSinFreeState  $\langle ordinary \rangle \hat{=}$ 
refines ComputeVSSStatesFollowingTTDDStates
any
  vss
  vss1
  vss2
  vss3
  vss4
  newstateVSScomputed1
where
  grd1:  $vss = stateVSS^{-1}[\{FREE\}]$ 
  grd2:  $partition(vss, vss1, vss2, vss3, vss4)$ 
  grd3:  $newstateVSScomputed1 = stateVSS \triangleleft ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup$ 
     $(vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOWN\}))$ 
then
  act1:  $newstateVSScomputed := newstateVSScomputed1$ 
end
Event _connectTrain  $\langle ordinary \rangle \hat{=}$ 
extends _connectTrain
any
  tr
  fr
  re
  integer
  ttDs
where
  grd0:  $TRAIN \setminus dom(connectedTrain) \neq \emptyset$ 
  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:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (integer = TRUE \wedge ((re \dots fr) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
    grd8:  $integer = TRUE \Rightarrow (\forall tr1. (tr1 \in dom(rear) \Rightarrow (rear(tr1) \dots front(tr1)) \cap (re \dots fr) = \emptyset))$ 
    grd9:  $integer = TRUE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge re \dots fr \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < re))$ 
    grd10:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge ttd \in TTD \wedge rear(tr1) \dots front(tr1) \cap ttd \neq \emptyset \wedge fr \in ttd) \Rightarrow fr < rear(tr1)))$ 
    grd11:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$ 

  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  $\langle ordinary \rangle \hat{=}$ 
extends _toggleTrainConnexionStatus
any
  tr
  integer
  re
  ttds
where
  grd0:  $dom(connectedTrain) \neq \emptyset$ 
  grd1:  $tr \in dom(connectedTrain)$ 
  grd2:  $integer \in BOOL$ 
  grd3:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (re \in WAY \wedge re < front(tr))$ 
  grd4:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (tr \in dom(MA) \wedge re \in MA(tr))$ 
  grd5:  $ttds \subseteq stateTTD^{-1}[\{FREE\}]$ 
  grd6:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge (connectedTrain(tr) = FALSE \wedge integer = TRUE \wedge ((re \dots front(tr)) \cap ttd \neq \emptyset)) \Rightarrow ttd \in ttds)$ 
  grd7:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (\forall tr1. ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) \dots front(tr1)) \cap (re \dots front(tr)) = \emptyset))$ 
  grd8:  $(connectedTrain(tr) = FALSE \wedge integer = TRUE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge re \dots front(tr) \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < re))$ 
  grd9:  $(connectedTrain(tr) = FALSE \wedge integer = FALSE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge rear(tr1) \dots front(tr1) \cap ttd \neq \emptyset \wedge front(tr) \in ttd) \Rightarrow front(tr) < rear(tr1)))$ 
  grd10:  $(connectedTrain(tr) = FALSE \wedge integer = FALSE) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD) \Rightarrow ((front(tr1) \in ttd \Rightarrow front(tr) \notin ttd) \wedge (front(tr) \in ttd \Rightarrow front(tr1) \notin ttd))))$ 

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

  end

Event MoveTrainFollowingItsMA  $\langle ordinary \rangle \hat{=}$ 
extends MoveTrainFollowingItsMA
any
  tr
  len
  n_rear
  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:  $tr \in dom(rear) \Rightarrow n\_rear = rear \Leftarrow \{tr \mapsto rear(tr) + len\}$ 

```

```

    grd5:  $tr \notin \text{dom}(\text{rear}) \Rightarrow n\_rear = \text{rear}$ 
    grd6:  $ttds \subseteq \text{stateTTD}^{-1}[\{FREE\}]$ 
    grd7:  $\forall ttd. (ttd \in \text{stateTTD}^{-1}[\{FREE\}] \wedge ((\text{front}(tr) + \text{len} \in ttd) \vee (tr \in \text{dom}(n\_rear) \wedge$ 
       $((n\_rear(tr) .. \text{front}(tr) + \text{len}) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
    grd8:  $tr \in \text{dom}(n\_rear) \Rightarrow (\forall tr1. ((tr1 \in \text{dom}(\text{rear}) \wedge tr1 \neq tr) \Rightarrow (\text{rear}(tr1) .. \text{front}(tr1)) \cap$ 
       $(n\_rear(tr) .. \text{front}(tr) + \text{len}) = \emptyset))$ 
    grd9:  $tr \in \text{dom}(n\_rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in \text{dom}(\text{front}) \setminus \text{dom}(\text{rear}) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge$ 
       $n\_rear(tr) .. (\text{front}(tr) + \text{len}) \cap ttd \neq \emptyset \wedge \text{front}(tr1) \in ttd) \Rightarrow \text{front}(tr1) < n\_rear(tr)))$ 
    grd10:  $tr \in \text{dom}(\text{front}) \setminus \text{dom}(\text{rear}) \Rightarrow (\forall tr1, ttd. ((tr1 \in \text{dom}(\text{rear}) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge$ 
       $\text{rear}(tr1) .. \text{front}(tr1) \cap ttd \neq \emptyset \wedge (\text{front}(tr) + \text{len}) \in ttd) \Rightarrow \text{front}(tr) + \text{len} < \text{rear}(tr1)))$ 
    grd11:  $tr \in \text{dom}(\text{front}) \setminus \text{dom}(\text{rear}) \Rightarrow (\forall tr1, ttd. ((tr1 \in \text{dom}(\text{front}) \setminus \text{dom}(\text{rear}) \wedge tr1 \neq tr \wedge ttd \in$ 
       $TTD \wedge \text{front}(tr1) \in ttd) \Rightarrow \text{front}(tr) + \text{len} \notin ttd)$ 
  then
    act1:  $\text{front}(tr) := \text{front}(tr) + \text{len}$ 
    act2:  $\text{rear} := n\_rear$ 
    act3:  $\text{stateTTD} := \text{stateTTD} \Leftarrow (ttds \times \{OCCUPIED\})$ 
  end
Event _exitTrain <ordinary>  $\hat{=}$ 
extends _exitTrain
  any
     $tr$ 
  where
    grd1:  $tr \in \text{connectedTrain}^{-1}[\{TRUE\}]$ 
  then
    act1:  $\text{front} := \{tr\} \Leftarrow \text{front}$ 
    act2:  $\text{rear} := (\{TRUE \mapsto \{tr\} \Leftarrow \text{rear}, FALSE \mapsto \text{rear}\})(\text{bool}(tr \in \text{dom}(\text{rear})))$ 
    act3:  $\text{connectedTrain} := \{tr\} \Leftarrow \text{connectedTrain}$ 
    act4:  $MA := (\{TRUE \mapsto \{tr\} \Leftarrow MA, FALSE \mapsto MA\})(\text{bool}(tr \in \text{dom}(MA)))$ 
    act5:  $MAtemp := (\{TRUE \mapsto \{tr\} \Leftarrow MAtemp, FALSE \mapsto MAtemp\})(\text{bool}(tr \in \text{dom}(MAtemp)))$ 
  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: $connectedTrain := \emptyset$
act2: $front := \emptyset$
act3: $rear := \emptyset$
act4: $MA := \emptyset$
act5: $MAtemp := \emptyset$
act6: $stateTTD := TTD \times \{OCCUPIED\}$
act7: $stateVSS := VSS \times \{UNKNOWN\}$
act8: $newstateVSScomputed := VSS \times \{UNKNOWN\}$
act10: $freeVssChangingtoFree := \emptyset$
act11: $freeVssChangingtoUnknow := \emptyset$
act12: $freeVssChangingtoOccupied := \emptyset$
act13: $freeVssChangingtoAmbiguous := \emptyset$
act14: $muteTimer := TRAIN \times \{INACTIVE\}$
act15: $waitIntegrityTimer := TRAIN \times \{INACTIVE\}$

end**Event** ComputeStatesOfVSSinFreeStateWhenTTDisFree *(ordinary)* $\hat{=}$ **any** $vssTtdFree$ **where**

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

then**act1:** $freeVssChangingtoFree := freeVssChangingtoFree \cup vssTtdFree$ **end****Event** ComputeStatesOfVSSinFreeStateWhenTTDisOccupiedandNoTrainisLocatedandNoMAisIssued *(ordinary)* $\hat{=}$ **any**

```

    vssTtdOccupiedwithNoTrainAndNoMA
  where
    grd1:  $vssTtdOccupiedwithNoTrainAndNoMA \subseteq stateVSS^{-1}[\{FREE\}]$ 
    grd2:  $\forall vss. (vss \in vssTtdOccupiedwithNoTrainAndNoMA \Rightarrow vss \subseteq union(stateTTD^{-1}[\{OCCUPIED\}]))$ 

    grd3:  $\forall vss, p, q. ((vss \in vssTtdOccupiedwithNoTrainAndNoMA \wedge p .. q \in TTD \wedge vss \subseteq p .. q) \Rightarrow$ 
       $(\forall tr. tr \in connectedTrain^{-1}[\{TRUE\}] \wedge tr \in dom(rear) \Rightarrow (front(tr) < p \vee rear(tr) > q)))$ 
    grd4:  $\forall vss, p, q. ((vss \in vssTtdOccupiedwithNoTrainAndNoMA \wedge p .. q \in TTD \wedge vss \subseteq p .. q) \Rightarrow$ 
       $(\forall tr. tr \in connectedTrain^{-1}[\{TRUE\}] \wedge tr \notin dom(rear) \Rightarrow (front(tr) < p \vee front(tr) > q)))$ 
    grd5:  $\forall vss, ttd. ((vss \in vssTtdOccupiedwithNoTrainAndNoMA \wedge ttd \in TTD \wedge vss \subseteq ttd) \Rightarrow$ 
       $(union(ran(MA)) \cap ttd = \emptyset))$ 
  then
    act1:  $freeVssChangingtoUnknow := freeVssChangingtoUnknow \cup vssTtdOccupiedwithNoTrainAndNoMA$ 
  end

Event ComputeStatesOfVSSinFreeStateFollowing_1B ⟨ordinary⟩  $\hat{=}$ 
  any
    vss_1B
  where
    grd1:  $vss\_1B \subseteq stateVSS^{-1}[\{FREE\}]$ 
    grd2:  $\forall vss. (vss \in vss\_1B \Rightarrow vss \subseteq union(stateTTD^{-1}[\{OCCUPIED\}]))$ 
    grd3:  $\forall vss. (vss \in vss\_1B \Rightarrow \exists tr. (tr \in dom(MA) \wedge vss \subseteq MA(tr) \wedge muteTimer(tr) = EXPIRED))$ 
    grd4:  $\forall vss, tr, p, q. ((vss \in vss\_1B \wedge tr \in dom(MA) \wedge vss \subseteq MA(tr) \wedge muteTimer(tr) = EXPIRED \wedge$ 
       $vss = p .. q) \Rightarrow p \geq front(tr))$ 
  then
    act1:  $freeVssChangingtoUnknow := freeVssChangingtoUnknow \cup vss\_1B$ 
  end

Event FullComputeStatesOfVSSinFreeState ⟨ordinary⟩  $\hat{=}$ 
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:  $newstateVSScomputed1 = stateVSS \bowtie ((vss1 \times \{OCCUPIED\}) \cup (vss2 \times \{FREE\}) \cup$ 
       $(vss3 \times \{AMBIGUOUS\}) \cup (vss4 \times \{UNKNOW\}))$ 
  then
    act1:  $newstateVSScomputed := newstateVSScomputed1$ 
  end

Event _connectTrain ⟨ordinary⟩  $\hat{=}$ 
extends _connectTrain
  any
    tr
    fr
    re
    integer
    ttds
  where
    grd0:  $TRAIN \setminus dom(connectedTrain) \neq \emptyset$ 
    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:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((fr \in ttd) \vee (integer = TRUE \wedge ((re \dots fr) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
grd8:  $integer = TRUE \Rightarrow (\forall tr1. (tr1 \in dom(rear) \Rightarrow (rear(tr1) \dots front(tr1)) \cap (re \dots fr) = \emptyset))$ 
grd9:  $integer = TRUE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge re \dots fr \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < re))$ 
grd10:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge ttd \in TTD \wedge rear(tr1) \dots front(tr1) \cap ttd \neq \emptyset \wedge fr \in ttd) \Rightarrow fr < rear(tr1)))$ 
grd11:  $integer = FALSE \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow fr \notin ttd))$ 

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

```

Event $_toggleTrainConnexionStatus$ $\langle ordinary \rangle \hat{=}$

extends $_toggleTrainConnexionStatus$

any

tr
 $integer$
 re
 $ttds$

where

```

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

```

then

```

act1:  $connectedTrain := (\{TRUE \mapsto connectedTrain \Leftarrow \{tr \mapsto FALSE\}, FALSE \mapsto connectedTrain \Leftarrow \{tr \mapsto TRUE\}\})(bool(connectedTrain(tr) = TRUE))$ 
act2:  $rear := (\{TRUE \mapsto (\{TRUE \mapsto rear \Leftarrow \{tr \mapsto re\}, FALSE \mapsto \{tr\} \Leftarrow rear\})(bool(integer = TRUE)), FALSE \mapsto rear\})(bool(connectedTrain(tr) = FALSE))$ 
act3:  $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$ 
act4:  $muteTimer := (\{TRUE \mapsto muteTimer, FALSE \mapsto muteTimer \Leftarrow \{tr \mapsto STARTED\}\})(bool(connectedTrain(tr) = TRUE))$ 
act5:  $waitIntegrityTimer := (\{TRUE \mapsto waitIntegrityTimer \Leftarrow \{tr \mapsto INACTIVE\}, FALSE \mapsto ((\{TRUE \mapsto waitIntegrityTimer \Leftarrow \{tr \mapsto STARTED\}, FALSE \mapsto waitIntegrityTimer \Leftarrow \{tr \mapsto INACTIVE\}\})(integer)))\})(bool(tr \in dom(rear) \wedge re \neq rear(tr)))$ 

```

```

end
Event MoveTrainFollowingItsMA ⟨ordinary⟩ ≐
extends MoveTrainFollowingItsMA
any
  tr
  len
  n_rear
  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:  $tr \in dom(rear) \Rightarrow n\_rear = rear \Leftarrow \{tr \mapsto rear(tr) + len\}$ 
  grd5:  $tr \notin dom(rear) \Rightarrow n\_rear = rear$ 
  grd6:  $ttds \subseteq stateTTD^{-1}[\{FREE\}]$ 
  grd7:  $\forall ttd. (ttd \in stateTTD^{-1}[\{FREE\}] \wedge ((front(tr) + len \in ttd) \vee (tr \in dom(n\_rear) \wedge ((n\_rear(tr) .. front(tr) + len) \cap ttd \neq \emptyset))) \Rightarrow ttd \in ttds)$ 
  grd8:  $tr \in dom(n\_rear) \Rightarrow (\forall tr1. ((tr1 \in dom(rear) \wedge tr1 \neq tr) \Rightarrow (rear(tr1) .. front(tr1)) \cap (n\_rear(tr) .. front(tr) + len) = \emptyset))$ 
  grd9:  $tr \in dom(n\_rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge n\_rear(tr) .. (front(tr) + len) \cap ttd \neq \emptyset \wedge front(tr1) \in ttd) \Rightarrow front(tr1) < n\_rear(tr)))$ 
  grd10:  $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge rear(tr1) .. front(tr1) \cap ttd \neq \emptyset \wedge (front(tr) + len) \in ttd) \Rightarrow front(tr) + len < rear(tr1)))$ 
  grd11:  $tr \in dom(front) \setminus dom(rear) \Rightarrow (\forall tr1, ttd. ((tr1 \in dom(front) \setminus dom(rear) \wedge tr1 \neq tr \wedge ttd \in TTD \wedge front(tr1) \in ttd) \Rightarrow front(tr) + len \notin ttd))$ 
then
  act1:  $front(tr) := front(tr) + len$ 
  act2:  $rear := n\_rear$ 
  act3:  $stateTTD := stateTTD \Leftarrow (ttds \times \{OCCUPIED\})$ 
  act4:  $muteTimer(tr) := STARTED$ 
end
Event expireMuteTimer ⟨ordinary⟩ ≐
any
  tr
where
  grd0:  $dom(connectedTrain) \neq \emptyset$ 
  grd1:  $tr \in dom(connectedTrain)$ 
  grd2:  $muteTimer(tr) = STARTED$ 
then
  act0:  $muteTimer(tr) := EXPIRED$ 
  act1:  $waitIntegrityTimer(tr) := INACTIVE$ 
end
Event expireWaitIntegrityTimer ⟨ordinary⟩ ≐
any
  tr
where
  grd0:  $dom(connectedTrain) \neq \emptyset$ 
  grd1:  $tr \in dom(connectedTrain)$ 
  grd2:  $waitIntegrityTimer(tr) = STARTED$ 
then
  act0:  $waitIntegrityTimer(tr) := EXPIRED$ 
end
Event _exitTrain ⟨ordinary⟩ ≐
extends _exitTrain
any
  tr
where
  grd1:  $tr \in connectedTrain^{-1}[\{TRUE\}]$ 

```



```
    then
      act1: front := {tr}  $\triangleleft$  front
      act2: rear := ({TRUE  $\mapsto$  {tr}  $\triangleleft$  rear, FALSE  $\mapsto$  rear})(bool(tr  $\in$  dom(rear)))
      act3: connectedTrain := {tr}  $\triangleleft$  connectedTrain
      act4: MA := ({TRUE  $\mapsto$  {tr}  $\triangleleft$  MA, FALSE  $\mapsto$  MA})(bool(tr  $\in$  dom(MA)))
      act5: MAtemp := ({TRUE  $\mapsto$  {tr}  $\triangleleft$  MAtemp, FALSE  $\mapsto$  MAtemp})(bool(tr  $\in$  dom(MAtemp)))
      act6: muteTimer(tr) := INACTIVE
      act7: waitIntegrityTimer(tr) := INACTIVE
    end
  END
```