Contents

CONTEXT steam_boiler_controller_context

SETS

SteamBoiler

CONSTANTS

SB

AXIOMS

axm: $SB \in SteamBoiler$

axm1: $SteamBoiler = \{SB\}$

END

CONTEXT steam_boiler_controller_context2
EXTENDS steam_boiler_controller_context
SETS

Sensor
Actuator
Data_Set_1
Data_Set_2
Data_Set_3

CONSTANTS

SteamBoilerSensors
SteamBoilerActuators
Min1
Max1
Min2
Max2
degraded
normal
rescue
defective
nondefective
close
open

AXIOMS

axm0: $finite(SteamBoiler)$
axm1: $SteamBoilerSensors \in SteamBoiler \leftrightarrow Sensor$
axm2: $\forall x \cdot (x \in Sensor \Rightarrow card(SteamBoilerSensors^{-1}[\{x\}]) = 1)$
axm3: $SteamBoilerActuators \in SteamBoiler \leftrightarrow Actuator$
axm4: $\forall x \cdot (x \in Actuator \Rightarrow card(SteamBoilerActuators^{-1}[\{x\}]) = 1)$
axm5: $\{Min1, Max1, Min2, Max2\} \subseteq \mathbb{N}$
axm6: $partition(Data\_Set\_1, \{degraded\}, \{normal\}, \{rescue\})$
axm7: $partition(Data\_Set\_2, \{defective\}, \{nondefective\})$
axm8: $partition(Data\_Set\_3, \{close\}, \{open\})$
p1·1: $Min1 < Max1$
p1·2: $Min2 \leq Min1$
p1·3: $Max1 \leq Max2$

END

CONTEXT steam_boiler_controller_context3
EXTENDS steam_boiler_controller_context2
CONSTANTS

SU
SteamUnit
WU
WaterUnit
PC
PumpController
P
Pump
Controls

AXIOMS

axm0: $finite(Sensor) \land finite(Actuator)$
axm1: $\forall sb \cdot (sb \in SteamBoiler \Rightarrow card(SteamBoilerActuators[\{sb\}]) = 1)$
axm2: $\forall sb \cdot (sb \in SteamBoiler \Rightarrow card(SteamBoilerSensors[\{sb\}]) = 3)$
axm6: $partition(Sensor, WaterUnit, PumpController, SteamUnit)$
axm4: $SU \in SteamUnit$
axm5: $SteamUnit = \{SU\}$
axm7: $WU \in WaterUnit$
axm8: $WaterUnit = \{WU\}$
axm10: $PC \in PumpController$
axm11: $PumpController = \{PC\}$
axm12: $Pump \subseteq Actuator$
axm13: $P \in Pump$
axm14: $Pump = \{P\}$
axm15: $Controls \in Pump \rightarrowtail PumpController$
axm16: $Controls = \{P \mapsto PC\}$
axm17: $SteamBoilerActuators = \{SB \mapsto P\}$
axm18: $SteamBoilerSensors = \{SB \mapsto WU, SB \mapsto SU, SB \mapsto PC\}$

END

MACHINE steam_boiler_controller
SEES steam_boiler_controller_context
VARIABLES
waterLevel
INVARIANTS
inv: $waterLevel \in SteamBoiler \rightarrow \mathbb{N}$
EVENTS
Initialisation
begin
act: $waterLevel :\in \{SB\} \rightarrow \mathbb{N}$
end
Event ControlWaterLevel ⟨ordinary⟩ $\widehat{=}$
any
wlvl
where
grd: $wlvl \in \mathbb{N}$
then
act: $waterLevel(SB) := wlvl$
end
END

MACHINE steam_boiler_controller2
REFINES steam_boiler_controller
SEES steam_boiler_controller_context2
VARIABLES

waterLevel
operatingMode
sensorState
sensorInput
actuatorState
actuatorOutput

INVARIANTS

inv1: $operatingMode \in SteamBoiler \rightarrow Data\_Set\_1$
inv2: $sensorState \in Sensor \rightarrow Data\_Set\_2$
inv3: $actuatorState \in Actuator \rightarrow Data\_Set\_2$
inv4: $actuatorOutput \in Actuator \rightarrow Data\_Set\_3$
inv5: $sensorInput \in Sensor \rightarrow \mathbb{N}$
p1,4: $\forall sb \cdot ((sb \in SteamBoiler \wedge operatingMode(sb) = normal) \Rightarrow waterLevel(sb) \in Min1 \mathinner{..} Max1)$
p1,5: $\forall sb \cdot ((sb \in SteamBoiler \wedge operatingMode(sb) \in \{degraded, rescue\}) \Rightarrow waterLevel(sb) \in Min2 \mathinner{..} Max2)$
t1: ⟨theorem⟩
$\forall wlvl, values \cdot ($
$(wlvl \in (\{TRUE \mapsto Min1..Max1, FALSE \mapsto Min2..Max2\})(bool(operatingMode(SB) = normal)))$
$\wedge (values \in (SteamBoilerSensors[\{SB\}] \cap sensorState^{-1}[\{nondefective\}]) \rightarrow \mathbb{N})$
$) \Rightarrow$
$(wlvl \in \mathbb{N})$
$)$
t2: ⟨theorem⟩
$(\forall mode \cdot ($
$(mode \in Data\_Set\_1)$
$\wedge (waterLevel(SB) \in (\{TRUE \mapsto Min1..Max1, FALSE \mapsto Min2..Max2\})(bool(mode = normal)))$
$)) \Rightarrow$
$(\exists wlvl \cdot (wlvl \in \mathbb{N}))$
t3: ⟨theorem⟩
$(\forall actions \cdot ($
$(actions \in (SteamBoilerActuators[\{SB\}] \cap actuatorState^{-1}[\{nondefective\}]) \rightarrow Data\_Set\_3)$
$)) \Rightarrow$
$(\exists wlvl \cdot (wlvl \in \mathbb{N}))$
t4: ⟨theorem⟩
$\forall wlvl, values, mode, actions \cdot ($
$(wlvl \in (\{TRUE \mapsto Min1..Max1, FALSE \mapsto Min2..Max2\})(bool(operatingMode(SB) = normal)))$
$\wedge (values \in (SteamBoilerSensors[\{SB\}] \cap sensorState^{-1}[\{nondefective\}]) \rightarrow \mathbb{N})$
$\wedge (mode \in Data\_Set\_1)$
$\wedge (waterLevel(SB) \in (\{TRUE \mapsto Min1..Max1, FALSE \mapsto Min2..Max2\})(bool(mode = normal)))$
$\wedge (actions \in (SteamBoilerActuators[\{SB\}] \cap actuatorState^{-1}[\{nondefective\}]) \rightarrow Data\_Set\_3)$
$) \Rightarrow$
$(wlvl = wlvl)$
$)$

EVENTS
Initialisation
begin

act1: $waterLevel := \{SB \mapsto Min1\}$
act2: $operatingMode := \{SB \mapsto normal\}$
act3: $sensorState := Sensor \times \{nondefective\}$
act4: $actuatorState := Actuator \times \{nondefective\}$
act5: $actuatorOutput :\in Actuator \rightarrow Data\_Set\_3$
act6: $sensorInput :\in Sensor \rightarrow \mathbb{N}$

end

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

refines ControlWaterLevel

any

wlvl

values

where

grd1: $wlvl \in (\{TRUE \mapsto Min1 \mathinner{..} Max1, FALSE \mapsto Min2 \mathinner{..} Max2\})(bool(operatingMode(SB) = normal))$

grd2: $values \in (SteamBoilerSensors[\{SB\}] \cap sensorState^{-1}[\{nondefective\}]) \pfun \mathbb{N}$

then

act1: $waterLevel(SB) := wlvl$

act2: $sensorInput := sensorInput \domsub values$

end

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

any

mode

where

grd1: $mode \in Data\_Set\_1$

grd2: $waterLevel(SB) \in (\{TRUE \mapsto Min1 \mathinner{..} Max1, FALSE \mapsto Min2 \mathinner{..} Max2\})(bool(mode = normal))$

then

act: $operatingMode(SB) := mode$

end

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

any

actions

where

grd: $actions \in (SteamBoilerActuators[\{SB\}] \cap actuatorState^{-1}[\{nondefective\}]) \tfun Data\_Set\_3$

then

act2: $actuatorOutput := actuatorOutput \domsub actions$

end

END

MACHINE steam_boiler_controller3
REFINES steam_boiler_controller2
SEES steam_boiler_controller_context3
VARIABLES

waterLevel
operatingMode
sensorState
sensorInput
actuatorState
actuatorOutput

INVARIANTS

p2,1: $sensorState(WU) = defective \Rightarrow operatingMode(SB) = rescue$
p2,2: $(sensorState(WU) = nondefective \land defective \in sensorState[\{SU, PC\}] \cup actuatorState[\{P\}]) \Rightarrow operatingMode(SB) = degraded$
p2,3: $(sensorState[\{WU, PC, SU\}] = \{nondefective\} \land actuatorState(P) = nondefective) \Rightarrow operatingMode(SB) = normal$

EVENTS
Initialisation
begin
act1: $waterLevel := \{SB \mapsto Min1\}$
act2: $operatingMode := \{SB \mapsto normal\}$
act3: $sensorState := Sensor \times \{nondefective\}$
act4: $actuatorState := Actuator \times \{nondefective\}$
act5: $actuatorOutput :\in Actuator \to Data\_Set\_3$
act6: $sensorInput :\in Sensor \to \mathbb{N}$
end
Event ReadWaterUnit ⟨ordinary⟩ $\widehat{=}$
refines ReadInputs
any
wlvl
values
val
where
grd0: $sensorState(WU) = nondefective$
grd1: $val \in \mathbb{N}$
grd2: $values = \{WU \mapsto val\}$
grd3: $wlvl = values(WU)$
grd4: $wlvl \in (\{TRUE \mapsto Min1 \mathinner{..} Max1, FALSE \mapsto Min2 \mathinner{..} Max2\})(bool(operatingMode(SB) = normal))$
then
act1: $waterLevel(SB) := wlvl$
act2: $sensorInput := sensorInput \lhd\!\!+ values$
end
Event ReadInputsInRescueMode ⟨ordinary⟩ $\widehat{=}$
refines ReadInputs
any
wlvl
values
val1
val2
where
grd0: $sensorState(WU) = defective \land sensorState[\{SU, PC\}] = \{nondefective\}$
grd1: $\{val1, val2\} \subseteq \mathbb{N}$
grd2: $values = \{SU \mapsto val1, PC \mapsto val2\}$
grd3: $wlvl \in Min2 \mathinner{..} Max2$
then

act1: $waterLevel(SB) := wlvl$
act2: $sensorInput := sensorInput \lhd values$

end

Event ComputeNextSystemMode ⟨ordinary⟩ $\widehat{=}$
refines ComputeNextSystemMode

any
mode
where
grd1: $mode \in Data\_Set\_1$
grd2: $waterLevel(SB) \in (\{TRUE \mapsto Min1 \mathinner{..} Max1, FALSE \mapsto Min2 \mathinner{..} Max2\})(bool(mode = normal))$
grd3: $sensorState(WU) = defective \Rightarrow mode = rescue$
grd4: $(sensorState(WU) = nondefective \wedge defective \in sensorState[\{SU, PC\}] \cup actuatorState[\{P\}]) \Rightarrow mode = degraded$
grd5: $(sensorState[\{WU, PC, SU\}] = \{nondefective\} \wedge actuatorState(P) = nondefective) \Rightarrow mode = normal$
then
act: $operatingMode(SB) := mode$
end

Event OpenPump ⟨ordinary⟩ $\widehat{=}$
refines SendActionCommand

any
actions
where
grd0: $waterLevel(SB) < (\{TRUE \mapsto Max1, FALSE \mapsto Max2\})(bool(operatingMode(SB) = normal))$
grd1: $actions \in (SteamBoilerActuators[\{SB\}] \cap actuatorState^{-1}[\{nondefective\}]) \rightarrow \{open\}$
then
act2: $actuatorOutput := actuatorOutput \lhd actions$
end

Event ClosePump ⟨ordinary⟩ $\widehat{=}$
refines SendActionCommand

any
actions
where
grd0: $waterLevel(SB) > (\{TRUE \mapsto Min1, FALSE \mapsto Min2\})(bool(operatingMode(SB) = normal))$
grd1: $actions \in (SteamBoilerActuators[\{SB\}] \cap actuatorState^{-1}[\{nondefective\}]) \rightarrow \{close\}$
then
act2: $actuatorOutput := actuatorOutput \lhd actions$
end

END