REQUIREMENTS DOCUMENT Lab 2

Group 12

Assumptions:

- Only maximum pressure as invariant, lower pressure is allowed (only maximum pressure is critical for safety)
- Events keep the pressure in a preferred interval

inv1	Pressure <= 65
inv2	¬(pressure > 60 ∧ (nextState = TimerState ∨ nextState = SensorState)) ∨ heater = off ∨ Mode=SUPERVISED
inv3	¬(pressure > 50 ∧ pressure ≤ 55 ∧ (nextState = TimerState ∨ nextState = SensorState)) ∨ heater = high ∨ Mode=SUPERVISED
inv4	¬(pressure > 55 ∧ pressure ≤ 60 ∧ (nextState = TimerState ∨ nextState = SensorState)) ∨ heater = low ∨ Mode=SUPERVISED
inv5	¬(conTimeStamp < senTimeStamp ∧ nextState = SensorState) ∨ boilerOn = FALSE
inv6	∀x·(x ∈ ValidAddresses ⇔ x ≤ MaxAddress)

Variables: types in rodin as invariants

V10	Pressure [natural]
V15	senTimeStamp [natural]
V20	conTimeStamp [natural]
V30	boilerOn [bool]
V40	heater {high, low, off}
V50	nextState {ControllerState, SensorState, TimeState}
V60	Users ⊆ USERSET
V70	USERSET ≠ ∅
V80	ValidAddresses ⊆ Addresses
V90	MaxAddress ∈ ValidAddresses
V100	Roles = {Supervisor, Operator}

V110	Modes = {AUTOMATIC, SUPERVISED, MONITORED}
V120	Mode ∈ Modes
V130	Permissions ∈ Users → Roles

Events: set the heater state according to the current temperature

controlOff	P > 60 and nextState = ControllerState and Mode≠SUPERVISED and address is valid => h := off; increase TimeStamp; nextState = TimeStater
ControlHigh	P > 50 && P <= 55 nextState = ControllerState and and address is valid=> h := high; increase TimeStamp; nextState = TimeStater
ControlLow	P > 55 && P <= 60 nextState = ControllerState and Mode≠SUPERVISED and address is valid => h := low;; increase TimeStamp; nextState = TimeStater
SensorOff	Increase timestamp and change pressure accordingly, set package values
SensorLow	Increase timestamp and change pressure accordingly, set package values
SensorHigh	Increase timestamp and change pressure accordingly, set package values
controlTHigher	Synchronize controller local timestamp
sensorTHigher	Synchronize sensor local timestamp
Shutdown	Set boilerOn:=FALSE in case of invalid incoming timestamp of package
ManualShutdown	Allows setting of boilerOn being set to FALSE if Mode=SUPERVISED or MODE=MONITORED
Chernobyl	Sets heater to a random state if MODE=SUPERVISED
AddUser	Add newUser ∈ USERSET with newRole ∈ Roles to Users respectively newUser ↦ newRole to Permissions if newUser is not yet added to Users
switchMode Supervisor	If user has supervisor permissions, change Mode to newMode with newMode ≠ Mode
switchMode Operator	If user has operator permissions, change Mode to newMode with newMode ≠ Mode and Mode ≠ SUPERVISED

Boiler | BoilerMachine | Access | AccessMachine

```
CONTEXT
    Boiler  ›
SETS
    HeaterState  ›
    SystemState  ›
CONSTANTS
    high  ›
    low  ›
    off  ›
    SensorState  ›
    ControllerState  ›
    TimerState  ›
    TMAX  ›
    Addresses  ›
    ValidAddresses  ›
    MaxAddress  ›
AXIOMS
    axm1:  high ∈ HeaterState not theorem ›
    axm2:  low ∈ HeaterState not theorem ›
    axm3:  off ∈ HeaterState not theorem ›
    axm4:  partition(HeaterState,{high},{low},{off}) not theorem ›
    axm5:  SensorState ∈ SystemState not theorem ›
    axm6:  ControllerState ∈ SystemState not theorem ›
    axm17:  TimerState ∈ SystemState not theorem ›
    axm7:  partition(SystemState, {ControllerState}, {SensorState}, {TimerState}) not theorem ›
    axm8:  TMAX ∈ ℕ not theorem ›
    axm9:  TMAX = 5 not theorem ›
    axm11:  Addresses ⊆ ℕ not theorem ›
    axm12:  ValidAddresses ⊆ Addresses not theorem ›
    axm13:  MaxAddress ∈ ValidAddresses not theorem ›
    axm14:  MaxAddress = 100 not theorem ›
    axm15:  ∀x·(x ∈ ValidAddresses ⇔ x ≤ MaxAddress) not theorem ›
END
```

Rodin Platform

BMotion Studio ProB Window Help

Quick Access

BoilerMachine Access AccessMachine

```
MACHINE
  BoilerMachine
SEES
  Boiler
VARIABLES
  pressure
  senTimeStamp
  conTimeStamp
  heater
  boilerOn
  nextState
  Address
INVARIANTS
  inv6:  pressure ∈ ℕ not theorem
  inv7:  senTimeStamp ∈ ℕ not theorem
  inv8:  Address ∈ ℕ not theorem
  inv11: conTimeStamp ∈ ℕ not theorem
  inv9:  heater ∈ HeaterState not theorem
  inv12: boilerOn ∈ BOOL not theorem
  inv1:  pressure ≤ 65 not theorem
  inv2:  ¬(pressure > 60 ∧ (nextState = TimerState ∨ nextState = SensorState)) ∨ heater = off not theorem
  inv3:  ¬(pressure > 50 ∧ pressure ≤ 55 ∧ (nextState = TimerState ∨ nextState = SensorState)) ∨ heater = high not theorem
  inv4:  ¬(pressure > 55 ∧ pressure ≤ 60 ∧ (nextState = TimerState ∨ nextState = SensorState)) ∨ heater = low not theorem
  inv10: nextState ∈ SystemState not theorem
  inv5:  ¬(conTimeStamp < senTimeStamp ∧ nextState = SensorState) ∨ boilerOn = FALSE not theorem
EVENTS
  INITIALISATION:  not extended ordinary
  THEN
    act1:  pressure≔51
    act3:  senTimeStamp ≔ 0
    act7:  conTimeStamp ≔ 0
    act4:  heater≔high
    act5:  boilerOn≔TRUE
    act6:  nextState≔SensorState
    act8:  Address ≔ 0
  END
```

ID: 294-238-9756 Stop Share

Rodin Platform

BMotion Studio ProB Window Help

Quick Access

Boiler BoilerMachine Access AccessMachine

```
  controlOff:  not extended ordinary
  ANY
    delta
  WHERE
    grd1:  pressure > 60 ∧ nextState = ControllerState not theorem
    grd2:  delta≤1.TMAX not theorem
    grd3:  boilerOn = TRUE not theorem
    grd40: Address ∈ ValidAddresses not theorem
  THEN
    act1:  heater≔off
    act3:  conTimeStamp ≔ senTimeStamp + delta
    act4:  nextState ≔ TimerState
  END

  controlHigh:  not extended ordinary
  ANY
    delta
  WHERE
    grd1:  pressure > 50 ∧ pressure ≤ 55 ∧ nextState = ControllerState not theorem
    grd2:  delta≤1.TMAX not theorem
    grd3:  boilerOn = TRUE not theorem
    grd40: Address ∈ ValidAddresses not theorem
  THEN
    act1:  heater≔high
    act3:  conTimeStamp≔ senTimeStamp + delta
    act4:  nextState ≔ TimerState
  END

  abstractChernobyl:  not extended ordinary
  ANY
    delta
  WHERE
    grd1:  nextState = ControllerState not theorem
    grd5:  delta ∈ 1.TMAX not theorem
  END
```

ID: 294-238-9756 Stop Share

in Platform

BMotion Studio ProB Window Help

Quick Access

BoilerMachine Access AccessMachine

```
  controlLow:  not extended ordinary
  ANY
    delta
  WHERE
    grd1:  pressure > 55 ∧ pressure ≤ 60 ∧ nextState = ControllerState not theorem
    grd2:  delta≤1.TMAX not theorem
    grd3:  boilerOn = TRUE not theorem
    grd40: Address ∈ ValidAddresses not theorem
  THEN
    act1:  heater≔low
    act3:  conTimeStamp≔ senTimeStamp + delta
    act4:  nextState ≔ TimerState
  END

  sensorHigh:  not extended ordinary
  ANY
    pDelta
    tDelta
    rAddress
  WHERE
    grd1:  nextState = SensorState ∧ heater=high not theorem
    grd2:  pDelta≠0.3 not theorem
    grd4:  tDelta≤1.TMAX not theorem
    grd3:  boilerOn = TRUE not theorem
    grd5:  rAddress ∈ 0.MaxAddress not theorem
  THEN
    act1:  pressure ≔ pressure + pDelta
    act3:  nextState ≔ ControllerState
    act4:  senTimeStamp ≔ senTimeStamp + tDelta
    act5:  Address ≔ rAddress
  END

  sensorLow:  not extended ordinary
  ANY
    pDelta
    tDelta
```

ID: 294-238-9756 Stop Share

Platform

Motion Studio ProB Window Help

Quick Access

BoilerMachine Access AccessMachine

```
ANY
  pDelta
  tDelta
  rAddress
WHERE
  grd1:  nextState = SensorState ∧ heater=low not theorem
  grd2:  pDelta≠0.2 not theorem
  grd4:  tDelta≤1.TMAX not theorem
  grd3:  boilerOn = TRUE not theorem
  grd5:  rAddress ∈ 0.MaxAddress not theorem
THEN
  act1:  pressure ≔ pressure − pDelta
  act3:  nextState ≔ ControllerState
  act4:  senTimeStamp ≔ senTimeStamp + tDelta
  act5:  Address ≔ rAddress
END

sensorOff:  not extended ordinary
ANY
  pDelta
  tDelta
  rAddress
WHERE
  grd1:  nextState = SensorState ∧ heater=off not theorem
  grd2:  pDelta≠0.2 not theorem
  grd4:  tDelta≠1.TMAX not theorem
  grd3:  boilerOn = TRUE not theorem
  grd5:  rAddress ∈ 0.MaxAddress not theorem
THEN
  act1:  pressure ≔ pressure − pDelta
  act3:  nextState ≔ ControllerState
  act4:  senTimeStamp ≔ senTimeStamp + tDelta
  act5:  Address ≔ rAddress
END

sensorTHigher:  not extended ordinary
WHERE
```

Problems Properties Tasks Symbols

ID: 294-238-9756 Stop Share

Rodin Platform

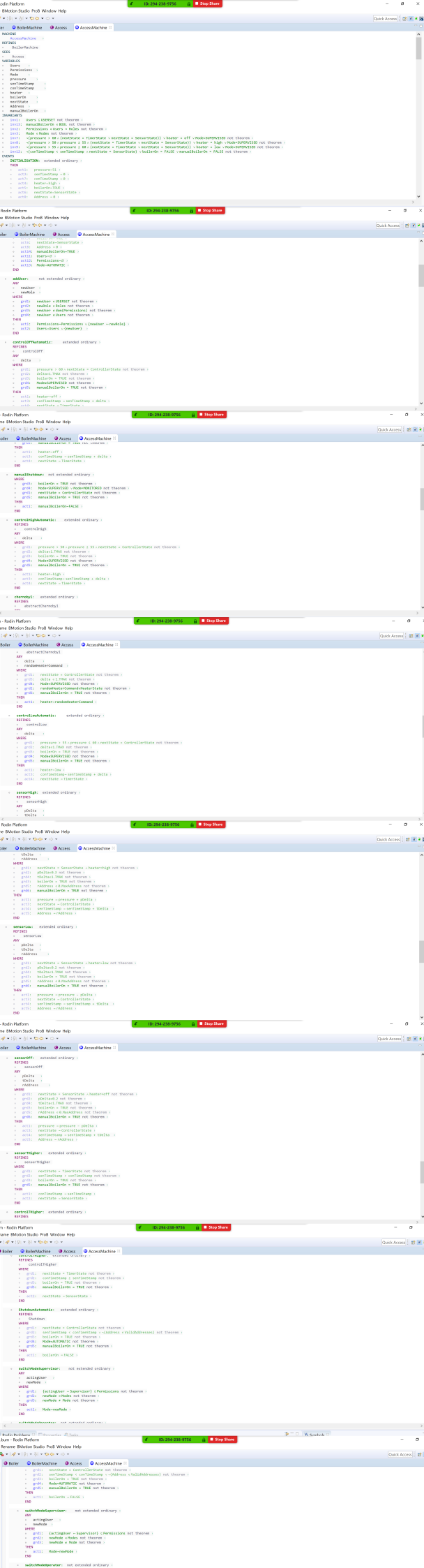
BMotion Studio ProB Window Help

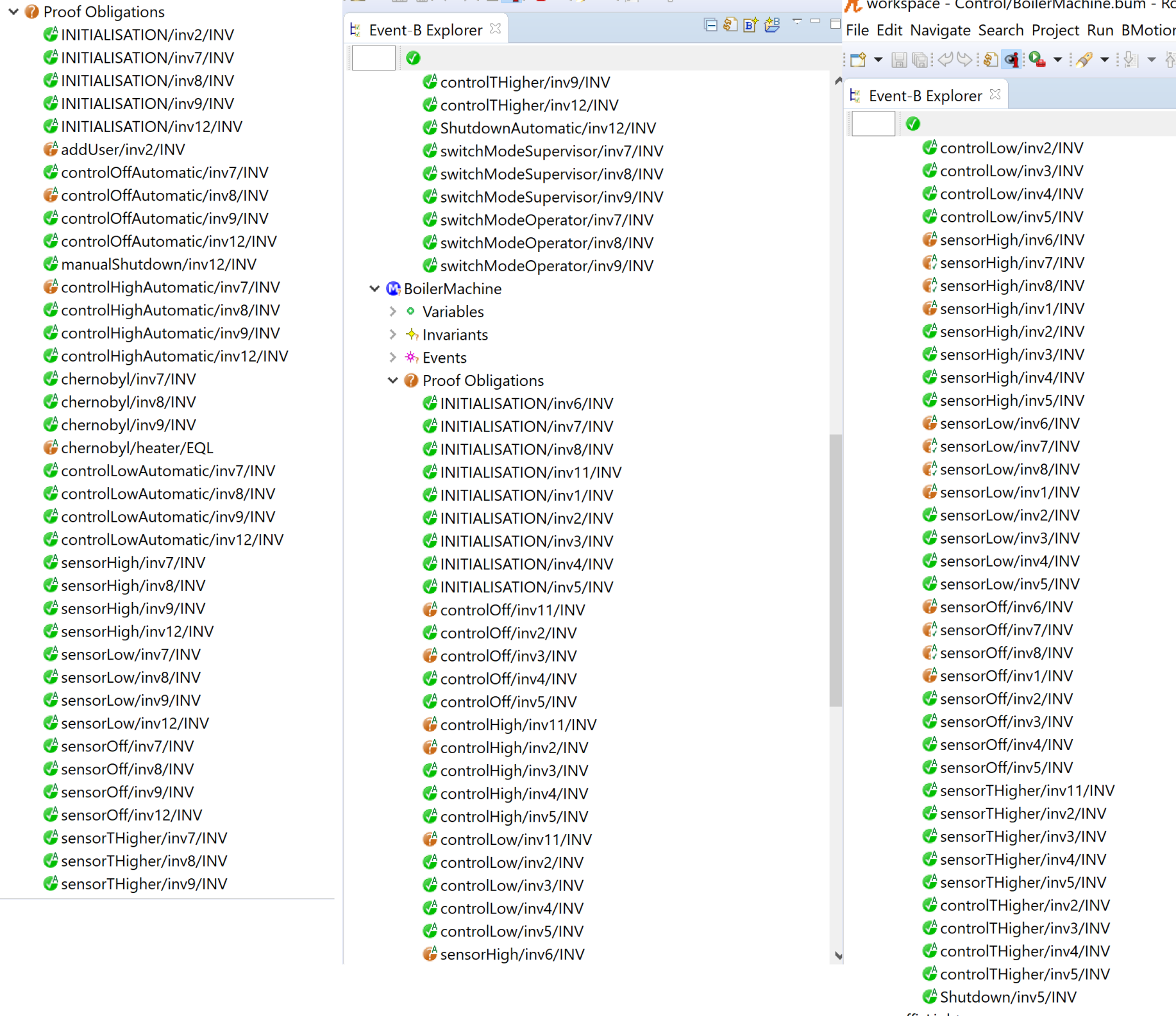
Quick Access

Boiler BoilerMachine Access AccessMachine

```
  THEN
    act1:  pressure ≔ pressure − pDelta
    act3:  nextState ≔ ControllerState
    act4:  senTimeStamp ≔ senTimeStamp + tDelta
    act5:  Address ≔ rAddress
  END

  sensorTHigher:  not extended ordinary
  WHERE
    grd1:  nextState = TimerState not theorem
    grd2:  senTimeStamp > conTimeStamp not theorem
    grd3:  boilerOn = TRUE not theorem
  THEN
    act1:  conTimeStamp ≔ senTimeStamp
    act2:  nextState ≔ SensorState
  END

  controlTHigher:  not extended ordinary
  WHERE
    grd1:  nextState = TimerState not theorem
    grd2:  conTimeStamp ≥ senTimeStamp not theorem
    grd3:  boilerOn = TRUE not theorem
  THEN
    act2:  nextState ≔ SensorState
  END

  Shutdown:  not extended ordinary
  WHERE
    grd1:  nextState = ControllerState not theorem
    grd2:  senTimeStamp < conTimeStamp ∨ ¬(Address ∈ ValidAddresses) not theorem
    grd3:  boilerOn = TRUE not theorem
  THEN
    act1:  boilerOn ≔ FALSE
  END

END
```

```
CONTEXT
    Access ›
EXTENDS
    Boiler
SETS
    Roles ›
    Modes ›
    USERSET ›
CONSTANTS
    Supervisor ›
    Operator ›
    AUTOMATIC ›
    SUPERVISED ›
    MONITORED ›
AXIOMS
    axm1: partition(Roles, {Supervisor},{Operator}) not theorem ›
    axm2: partition(Modes, {AUTOMATIC},{SUPERVISED},{MONITORED}) not theorem ›
    axm3: USERSET ≠ ∅ not theorem ›
    axm4: finite(USERSET) not theorem ›
END
```


Rodin Platform
ID: 294-238-9756
Stop Share
BMotion Studio Prob Window Help
Quick Access
BoilerMachine
Access
AccessMachine
MACHINE
AccessMachine
REFINES
BoilerMachine
SEES
Access
VARIABLES
Users
Permissions
Mode
pressure
sentTimeStamp
confTimeStamp
heater
boilerOn
nextState
Address
manualBoilerOn
INVARIANTS
inv1: Users ⊂ USERSET not theorem
inv10: manualBoilerOn ∈ BOOL not theorem
inv2: Permissions ∈ Users ↔ Roles not theorem
inv3: Mode = Modes not theorem
inv4: ¬(pressure > 80 ∧ (nextState = TimerState ∨ nextState = SensorState)) ⇒ heater = off ⇒ Mode=SUPERVISED not theorem
inv5: ¬(pressure < 50 ∧ pressure ≥ 15 ∧ (nextState = TimerState ∨ nextState = SensorState)) ⇒ heater = low ⇒ Mode=SUPERVISED not theorem
inv6: ¬(pressure < 15 ∧ pressure ≥ 0 ∧ (nextState = TimerState ∨ nextState = SensorState)) ⇒ heater = low ⇒ Mode=SUPERVISED not theorem
inv7: ¬(confTimeStamp = sentTimeStamp ∧ nextState = SensorState) ∧ boilerOn = FALSE ⇒ manualBoilerOn = FALSE not theorem
EVENTS
INITIALISATION: extended ordinary
THEN
act1: pressure≔51
act11: sentTimeStamp ≔ 0
act7: confTimeStamp ≔ 0
act4: heater≔high
act5: boilerOn≔TRUE
act6: nextState≔SensorState
act8: Address ≔ 0
END
addUser: not extended ordinary
ANY
newUser
newRole
WHERE
grd1: newUser ∈ USERSET not theorem
grd2: newRole ∈ Roles not theorem
grd3: newUser ∉ dom(Permissions) not theorem
grd4: newUser ∈ Users not theorem
THEN
act1: Permissions≔Permissions ∪ {newUser ↦ newRole}
act2: Users≔Users ∪ {newUser}
END
control2OffAutomatic: extended ordinary
REFINES
control2Off
ANY
delta
WHERE
grd1: pressure > 60 ∧ nextState = ControllerState not theorem
grd2: delta∈1‥TMAX not theorem
grd3: boilerOn = TRUE not theorem
grd4: Mode=SUPERVISED not theorem
grd5: manualBoilerOn = TRUE not theorem
THEN
act1: heater≔off
act2: confTimeStamp ≔ sentTimeStamp + delta
act3: nextState ≔ TimerState
END
manualShutdown: not extended ordinary
WHERE
grd1: boilerOn = TRUE not theorem
grd2: Mode=SUPERVISED ∨ Mode=MONITORED not theorem
grd3: nextState = ControllerState not theorem
grd4: manualBoilerOn = TRUE not theorem
THEN
act1: manualBoilerOn≔FALSE
END
control2HighAutomatic: extended ordinary
REFINES
control2High
ANY
delta
WHERE
grd1: pressure > 90 ∧ pressure ≤ 95 ∧ nextState = ControllerState not theorem
grd2: delta∈1‥TMAX not theorem
grd3: boilerOn = TRUE not theorem
grd4: Mode=SUPERVISED not theorem
grd5: manualBoilerOn = TRUE not theorem
THEN
act1: heater≔high
act2: confTimeStamp ≔ sentTimeStamp + delta
act3: nextState ≔ TimerState
END
chernobyl: extended ordinary
REFINES
abstractChernobyl
ANY
delta
randomHeaterCommand
WHERE
grd1: nextState = ControllerState not theorem
grd2: delta∈1‥TMAX not theorem
grd4: Mode=SUPERVISED not theorem
grd5: randomHeaterCommand=heater1state not theorem
grd6: manualBoilerOn = TRUE not theorem
THEN
act1: heater≔randomHeaterCommand
END
control1LowAutomatic: extended ordinary
REFINES
control1Low
ANY
delta
WHERE
grd1: pressure > 55 ∧ pressure ≤ 60 ∧ nextState = ControllerState not theorem
grd2: delta∈1‥TMAX not theorem
grd3: boilerOn = TRUE not theorem
grd4: Mode=SUPERVISED not theorem
grd5: manualBoilerOn = TRUE not theorem
THEN
act1: heater≔low
act2: confTimeStamp ≔ sentTimeStamp + delta
act3: nextState ≔ TimerState
END
sensorHigh: extended ordinary
REFINES
sensorHigh
ANY
pDelta
tDelta
tAddress
WHERE
grd1: nextState = SensorState ∧ heater=high not theorem
grd2: pDelta∈0‥1 not theorem
grd3: tDelta∈1‥TMAX not theorem
grd4: boilerOn = TRUE not theorem
grd5: tAddress ∈ dom(Address) not theorem
grd6: manualBoilerOn = TRUE not theorem
THEN
act1: pressure ≔ pressure + pDelta
act2: nextState ≔ ControllerState
act3: sentTimeStamp ≔ sentTimeStamp + tDelta
act4: Address ≔ tAddress
END
sensorLow: extended ordinary
REFINES
sensorLow
ANY
pDelta
tDelta
tAddress
WHERE
grd1: nextState = SensorState ∧ heater=low not theorem
grd2: pDelta∈0‥1 not theorem
grd3: tDelta∈1‥TMAX not theorem
grd4: boilerOn = TRUE not theorem
grd5: tAddress ∈ dom(Address) not theorem
grd6: manualBoilerOn = TRUE not theorem
THEN
act1: pressure ≔ pressure − pDelta
act2: nextState ≔ ControllerState
act3: sentTimeStamp ≔ sentTimeStamp + tDelta
act4: Address ≔ tAddress
END
sensorOff: extended ordinary
REFINES
sensorOff
ANY
pDelta
tDelta
rAddress
WHERE
grd1: nextState = SensorState ∧ heater=off not theorem
grd2: pDelta∈0‥2 not theorem
grd3: tDelta∈1‥TMAX not theorem
grd4: boilerOn = TRUE not theorem
grd5: rAddress ∈ dom(Address) not theorem
grd6: manualBoilerOn = TRUE not theorem
THEN
act1: pressure ≔ pressure + pDelta
act2: nextState ≔ ControllerState
act3: sentTimeStamp ≔ sentTimeStamp + tDelta
act4: Address ≔ rAddress
END
sensorThigher: extended ordinary
REFINES
sensorThigher
WHERE
grd1: nextState = TimerState not theorem
grd2: sentTimeStamp ≥ confTimeStamp not theorem
grd3: boilerOn = TRUE not theorem
grd4: manualBoilerOn = TRUE not theorem
THEN
act1: confTimeStamp ≔ sentTimeStamp
act2: nextState ≔ SensorState
END
control2Higher: extended ordinary
REFINES
control2Higher
ANY
delta
WHERE
grd1: nextState = TimerState not theorem
grd2: confTimeStamp ≥ sentTimeStamp not theorem
grd3: boilerOn = TRUE not theorem
grd4: manualBoilerOn = TRUE not theorem
THEN
act1: nextState ≔ SensorState
END
ShutdownAutomatic: extended ordinary
REFINES
Shutdown
WHERE
grd1: nextState = ControllerState not theorem
grd2: sentTimeStamp < confTimeStamp ∨ ¬(Address ∈ ValidAddresses) not theorem
grd3: boilerOn = TRUE not theorem
grd4: Mode=AUTOMATIC not theorem
grd5: manualBoilerOn = TRUE not theorem
THEN
act1: boilerOn ≔ FALSE
END
switchModeSupervisor: not extended ordinary
ANY
actingUser
newMode
WHERE
grd1: (actingUser ↦ Supervisor) ∈ Permissions not theorem
grd2: newMode ∈ Modes not theorem
grd3: newMode ≠ Mode not theorem
THEN
act1: Mode≔newMode
END
switchModeOperator: not extended ordinary
Problems
Properties
Tasks
Symbols
Rodin - Rodin Platform
Rename BMotion Studio Prob Window Help
Boiler
BoilerMachine
Access
AccessMachine


Proof Obligations
INITIALISATION/inv2/INV
INITIALISATION/inv7/INV
INITIALISATION/inv8/INV
INITIALISATION/inv9/INV
INITIALISATION/inv12/INV
addUser/inv2/INV
controlOffAutomatic/inv7/INV
controlOffAutomatic/inv8/INV
controlOffAutomatic/inv9/INV
controlOffAutomatic/inv12/INV
manualShutdown/inv12/INV
controlHighAutomatic/inv7/INV
controlHighAutomatic/inv8/INV
controlHighAutomatic/inv9/INV
controlHighAutomatic/inv12/INV
chernobyl/inv7/INV
chernobyl/inv8/INV
chernobyl/inv9/INV
chernobyl/heater/EQL
controlLowAutomatic/inv7/INV
controlLowAutomatic/inv8/INV
controlLowAutomatic/inv9/INV
controlLowAutomatic/inv12/INV
sensorHigh/inv7/INV
sensorHigh/inv8/INV
sensorHigh/inv9/INV
sensorHigh/inv12/INV
sensorLow/inv7/INV
sensorLow/inv8/INV
sensorLow/inv9/INV
sensorLow/inv12/INV
sensorOff/inv7/INV
sensorOff/inv8/INV
sensorOff/inv9/INV
sensorOff/inv12/INV
sensorTHigher/inv7/INV
sensorTHigher/inv8/INV
sensorTHigher/inv9/INV
Event-B Explorer
controlTHigher/inv9/INV
controlTHigher/inv12/INV
ShutdownAutomatic/inv12/INV
switchModeSupervisor/inv7/INV
switchModeSupervisor/inv8/INV
switchModeSupervisor/inv9/INV
switchModeOperator/inv7/INV
switchModeOperator/inv8/INV
switchModeOperator/inv9/INV
BoilerMachine
Variables
Invariants
Events
Proof Obligations
INITIALISATION/inv6/INV
INITIALISATION/inv7/INV
INITIALISATION/inv8/INV
INITIALISATION/inv11/INV
INITIALISATION/inv1/INV
INITIALISATION/inv2/INV
INITIALISATION/inv3/INV
INITIALISATION/inv4/INV
INITIALISATION/inv5/INV
controlOff/inv11/INV
controlOff/inv2/INV
controlOff/inv3/INV
controlOff/inv4/INV
controlOff/inv5/INV
controlHigh/inv11/INV
controlHigh/inv2/INV
controlHigh/inv3/INV
controlHigh/inv4/INV
controlHigh/inv5/INV
controlLow/inv11/INV
controlLow/inv2/INV
controlLow/inv3/INV
controlLow/inv4/INV
controlLow/inv5/INV
sensorHigh/inv6/INV
workspace - Control/BoilerMachine.bum - Ro
File Edit Navigate Search Project Run BMotion
Event-B Explorer
controlLow/inv2/INV
controlLow/inv3/INV
controlLow/inv4/INV
controlLow/inv5/INV
sensorHigh/inv6/INV
sensorHigh/inv7/INV
sensorHigh/inv8/INV
sensorHigh/inv1/INV
sensorHigh/inv2/INV
sensorHigh/inv3/INV
sensorHigh/inv4/INV
sensorHigh/inv5/INV
sensorLow/inv6/INV
sensorLow/inv7/INV
sensorLow/inv8/INV
sensorLow/inv1/INV
sensorLow/inv2/INV
sensorLow/inv3/INV
sensorLow/inv4/INV
sensorLow/inv5/INV
sensorOff/inv6/INV
sensorOff/inv7/INV
sensorOff/inv8/INV
sensorOff/inv1/INV
sensorOff/inv2/INV
sensorOff/inv3/INV
sensorOff/inv4/INV
sensorOff/inv5/INV
sensorTHigher/inv11/INV
sensorTHigher/inv2/INV
sensorTHigher/inv3/INV
sensorTHigher/inv4/INV
sensorTHigher/inv5/INV
controlTHigher/inv2/INV
controlTHigher/inv3/INV
controlTHigher/inv4/INV
controlTHigher/inv5/INV
Shutdown/inv5/INV