Appendix A

Models of the Controlled Water Tank System

Contents

A.1 Modelica Model of the Plant

```
model ControlledWaterTankPlant
  "Physical aspect of the controlled water tank"
  inner Modelica.Fluid.System system;
  Modelica.Fluid.Vessels.OpenTank tank(
    nPorts=2,
    height=3,
    crossArea=1,
    redeclare package Medium =
      Modelica.Media.Water.ConstantPropertyLiquidWater,
    portsData={Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
      diameter=0.1,
      height=3),
      Modelica.Fluid.Vessels.BaseClasses.VesselPortsData(
         diameter=0.2)},
    level_start=0);
  Modelica.Fluid.Sources.FixedBoundary source(
    redeclare package Medium =
      Modelica.Media.Water.ConstantPropertyLiquidWater,
    T=system.T_ambient,
    nPorts=1,
    p=2500000);
  Modelica.Fluid.Sources.FixedBoundary sink(
    redeclare package Medium =
      Modelica.Media.Water.ConstantPropertyLiquidWater,
    p=system.p_ambient,
    T=system.T_ambient,
    nPorts=1);
  Modelica.Fluid.Valves.ValveDiscrete valve(
    redeclare package Medium =
      Modelica.Media.Water.ConstantPropertyLiquidWater,
    dp_nominal=100000,
    m_flow_nominal=10);
  Modelica.Blocks.Interfaces.BooleanInput valveInput
    "Control signal of valve state (on/off)";
  Modelica.Blocks.Interfaces.RealOutput levelOutput "Water level";
  Modelica.Blocks.Sources.RealExpression tankLevel(y=tank.level);
equation
  connect(source.ports[1],valve. port_a);
  connect(valve.port_b,tank. ports[1]);
```

```
  connect(sink.ports[1],tank. ports[2]);
  connect(valveInput, valve.open);
  connect(tankLevel.y, levelOutput);
end ControlledWaterTankPlant;
```

A.2 Event-B Specification of the Controller

A.2.1 Context ctx

CONTEXT ctx

CONSTANTS

L

H

LT

HT

AXIOMS

axm3 : $L > 0$

axm4 : $H > L$

axm5 : $LT > L$

axm6 : $HT < H$

axm7 : $HT > LT$

axm1 : $LT = 10$

axm2 : $HT = 20$

END

A.2.2 Abstract Machine wtCtr0

MACHINE wtCtr0

SEES ctx

VARIABLES

$valve$

INVARIANTS

inv1 : $valve \in BOOL$

EVENTS

Initialisation

begin

act1 : $valve := TRUE$

end

Event $openValve \mathrel{\widehat{=}}$

any

l

where

grd1 : $l < LT$

then

act1 : $valve := TRUE$

end

Event $keepValve \mathrel{\widehat{=}}$

any

l

where

grd1 : $l \geq LT \wedge l \leq HT$

then

$skip$

end

Event $closeValve \mathrel{\widehat{=}}$

any

l

where

grd1 : $l > HT$

then

act1 : $valve := FALSE$

end

END

A.2.3 Concrete Machine wtCtr1

MACHINE wtCtr1

REFINES wtCtr0

SEES ctx

VARIABLES

$Read$

$Control$

$DecideClose$

$DecideKeep$

$DecideOpen$

$valve$

$level$

INVARIANTS

typeof_Read : $Read \in BOOL$

typeof_Control : $Control \in BOOL$

typeof_DecideClose : $DecideClose \in BOOL$

typeof_DecideKeep : $DecideKeep \in BOOL$

typeof_DecideOpen : $DecideOpen \in BOOL$

distinct_states_in_sm : $partition(\{TRUE\}, \{Read\} \cap \{TRUE\}, \{Control\} \cap \{TRUE\}, \{DecideClose\} \cap \{TRUE\}, \{DecideKeep\} \cap \{TRUE\}, \{DecideOpen\} \cap \{TRUE\})$

inv1 : $level \in L \mathinner{..} H$
system goal invariant

EVENTS

Initialisation

extended

begin

act1 : $valve := TRUE$

init_Control : $Control := TRUE$

init_Read : $Read := FALSE$

init_DecideClose : $DecideClose := FALSE$

init_DecideKeep : $DecideKeep := FALSE$

init_DecideOpen : $DecideOpen := FALSE$

act2 : $level := 0$

end

Event $openValve \mathrel{\widehat{=}}$

refines $openValve$

when

isin_DecideOpen : $DecideOpen = TRUE$

with

l : $l = level$

then

leave_DecideOpen : $DecideOpen := FALSE$

enter_Control : $Control := TRUE$

act1 : $valve := TRUE$

end

Event $keepValve \mathrel{\widehat{=}}$

refines $keepValve$

when

isin_DecideKeep : $DecideKeep = TRUE$

with

l : $l = level$

then

leave_DecideKeep : $DecideKeep := FALSE$

enter_Control : $Control := TRUE$

end

Event $closeValve \widehat{=}$

refines $closeValve$

when

isin_DecideClose : $DecideClose = TRUE$

with

l : $l = level$

then

leave_DecideClose : $DecideClose := FALSE$

enter_Control : $Control := TRUE$

act1 : $valve := FALSE$

end

Event $readLevel \widehat{=}$

any

l

where

grd1 : $l \geq 0$

then

act1 : $level := l$

end

Event $decideOpen \widehat{=}$

when

isin_Read : $Read = TRUE$

grd1 : $level < LT$

then

leave_Read : $Read := FALSE$

enter_DecideOpen : $DecideOpen := TRUE$

end

Event $decideKeep \widehat{=}$

when

isin_Read : $Read = TRUE$

grd1 : $level \geq LT \wedge level \leq HT$

then

leave_Read : $Read := FALSE$

enter_DecideKeep : $DecideKeep := TRUE$

end

Event $decideClose \widehat{=}$

when

isin_Read : $Read = TRUE$

grd1 : $level > HT$

then

leave_Read : $Read := FALSE$

enter_DecideClose : $DecideClose := TRUE$

end

END