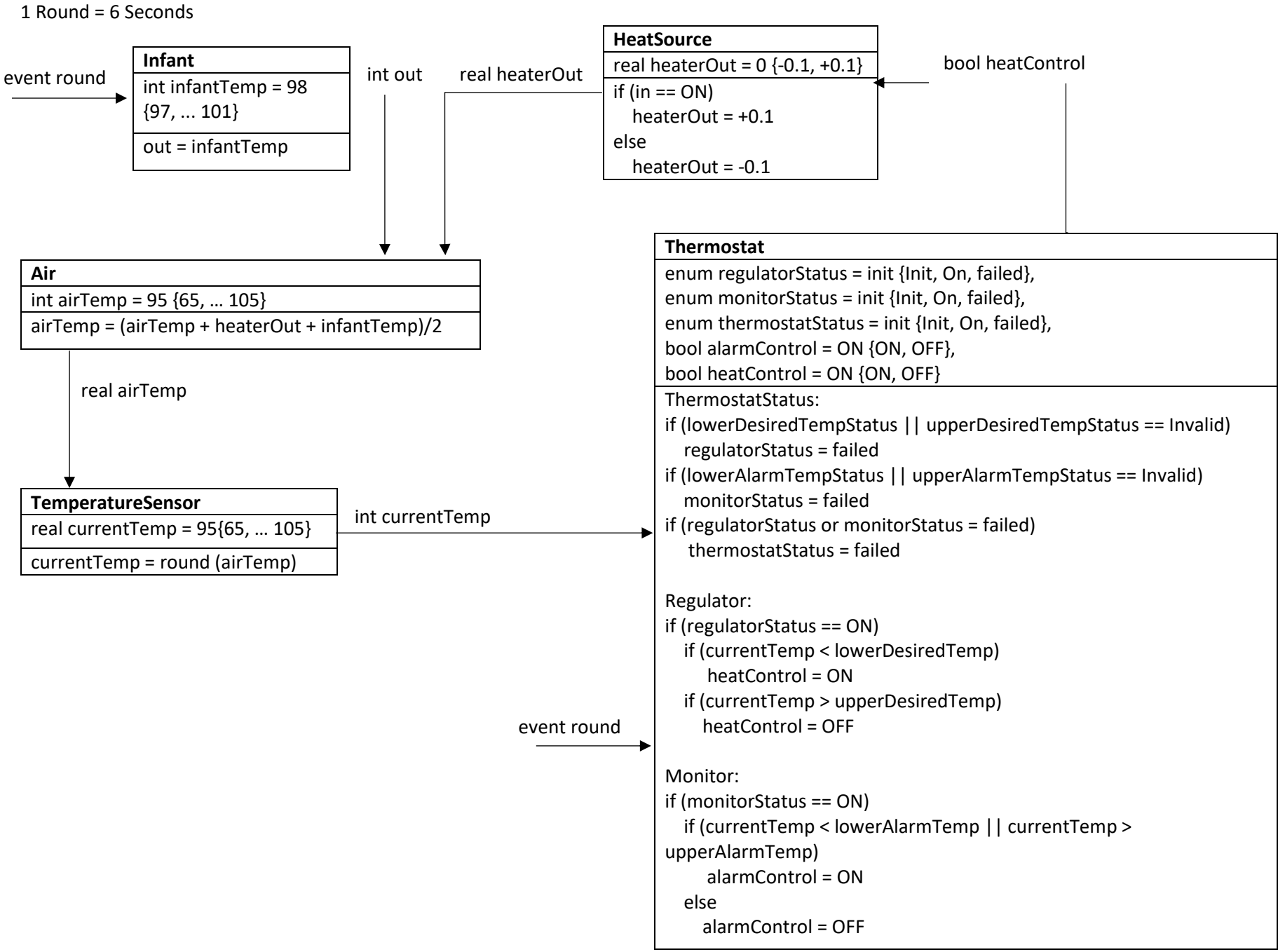
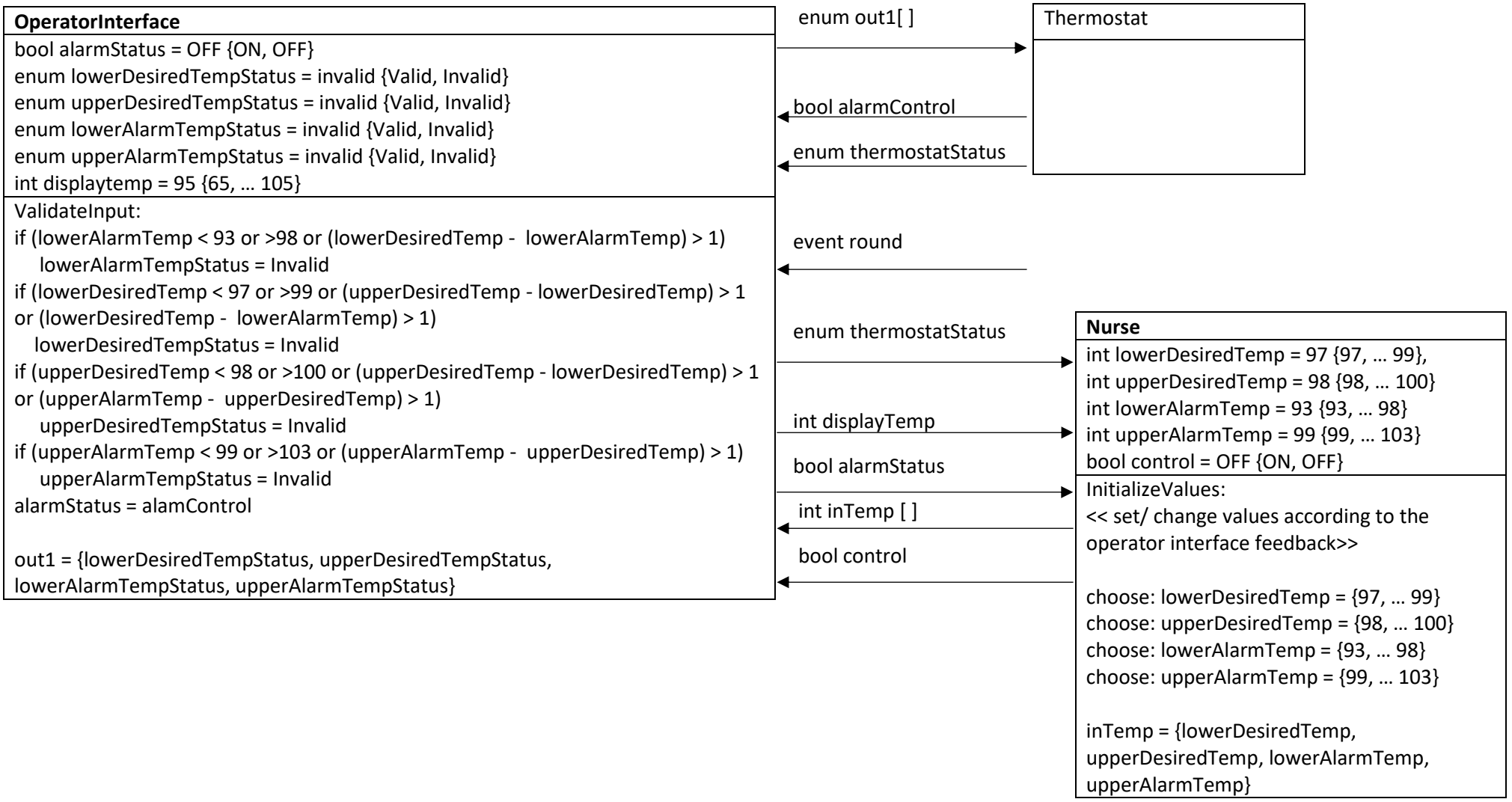

1 Round = 6 Seconds
event round
Infant
int infantTemp = 98
{97, ... 101}
out = infantTemp
int out
real heaterOut
HeatSource
real heaterOut = 0 {-0.1, +0.1}
if (in == ON)
heaterOut = +0.1
else
heaterOut = -0.1
bool heatControl
Air
int airTemp = 95 {65, … 105}
airTemp = (airTemp + heaterOut + infantTemp)/2
real airTemp
TemperatureSensor
real currentTemp = 95{65, … 105}
currentTemp = round (airTemp)
int currentTemp
Thermostat
enum regulatorStatus = init {Init, On, failed},
enum monitorStatus = init {Init, On, failed},
enum thermostatStatus = init {Init, On, failed},
bool alarmControl = ON {ON, OFF},
bool heatControl = ON {ON, OFF}
ThermostatStatus:
if (lowerDesiredTempStatus || upperDesiredTempStatus == Invalid)
regulatorStatus = failed
if (lowerAlarmTempStatus || upperAlarmTempStatus == Invalid)
monitorStatus = failed
if (regulatorStatus or monitorStatus = failed)
thermostatStatus = failed
Regulator:
if (regulatorStatus == ON)
if (currentTemp < lowerDesiredTemp)
heatControl = ON
if (currentTemp > upperDesiredTemp)
heatControl = OFF
event round
Monitor:
if (monitorStatus == ON)
if (currentTemp < lowerAlarmTemp || currentTemp >
upperAlarmTemp)
alarmControl = ON
else
alarmControl = OFF


OperatorInterface
bool alarmStatus = OFF {ON, OFF}
enum lowerDesiredTempStatus = invalid {Valid, Invalid}
enum upperDesiredTempStatus = invalid {Valid, Invalid}
enum lowerAlarmTempStatus = invalid {Valid, Invalid}
enum upperAlarmTempStatus = invalid {Valid, Invalid}
int displaytemp = 95 {65, … 105}
ValidateInput:
if (lowerAlarmTemp < 93 or >98 or (lowerDesiredTemp - lowerAlarmTemp) > 1)
lowerAlarmTempStatus = Invalid
if (lowerDesiredTemp < 97 or >99 or (upperDesiredTemp - lowerDesiredTemp) > 1
or (lowerDesiredTemp - lowerAlarmTemp) > 1)
lowerDesiredTempStatus = Invalid
if (upperDesiredTemp < 98 or >100 or (upperDesiredTemp - lowerDesiredTemp) > 1
or (upperAlarmTemp - upperDesiredTemp) > 1)
upperDesiredTempStatus = Invalid
if (upperAlarmTemp < 99 or >103 or (upperAlarmTemp - upperDesiredTemp) > 1)
upperAlarmTempStatus = Invalid
alarmStatus = alamControl
out1 = {lowerDesiredTempStatus, upperDesiredTempStatus,
lowerAlarmTempStatus, upperAlarmTempStatus}
enum out1[ ]
bool alarmControl
enum thermostatStatus
Thermostat
event round
enum thermostatStatus
int displayTemp
bool alarmStatus
int inTemp [ ]
bool control
Nurse
int lowerDesiredTemp = 97 {97, … 99},
int upperDesiredTemp = 98 {98, … 100}
int lowerAlarmTemp = 93 {93, … 98}
int upperAlarmTemp = 99 {99, … 103}
bool control = OFF {ON, OFF}
InitializeValues:
<< set/ change values according to the
operator interface feedback>>
choose: lowerDesiredTemp = {97, … 99}
choose: upperDesiredTemp = {98, … 100}
choose: lowerAlarmTemp = {93, … 98}
choose: upperAlarmTemp = {99, … 103}
inTemp = {lowerDesiredTemp,
upperDesiredTemp, lowerAlarmTemp,
upperAlarmTemp}