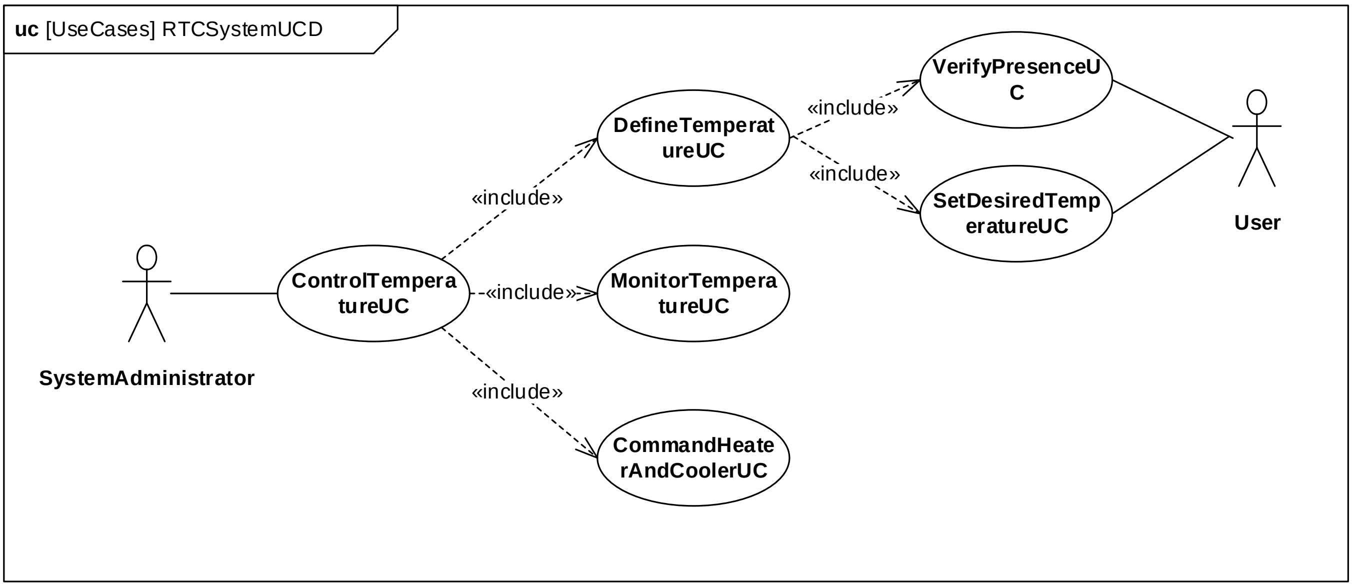
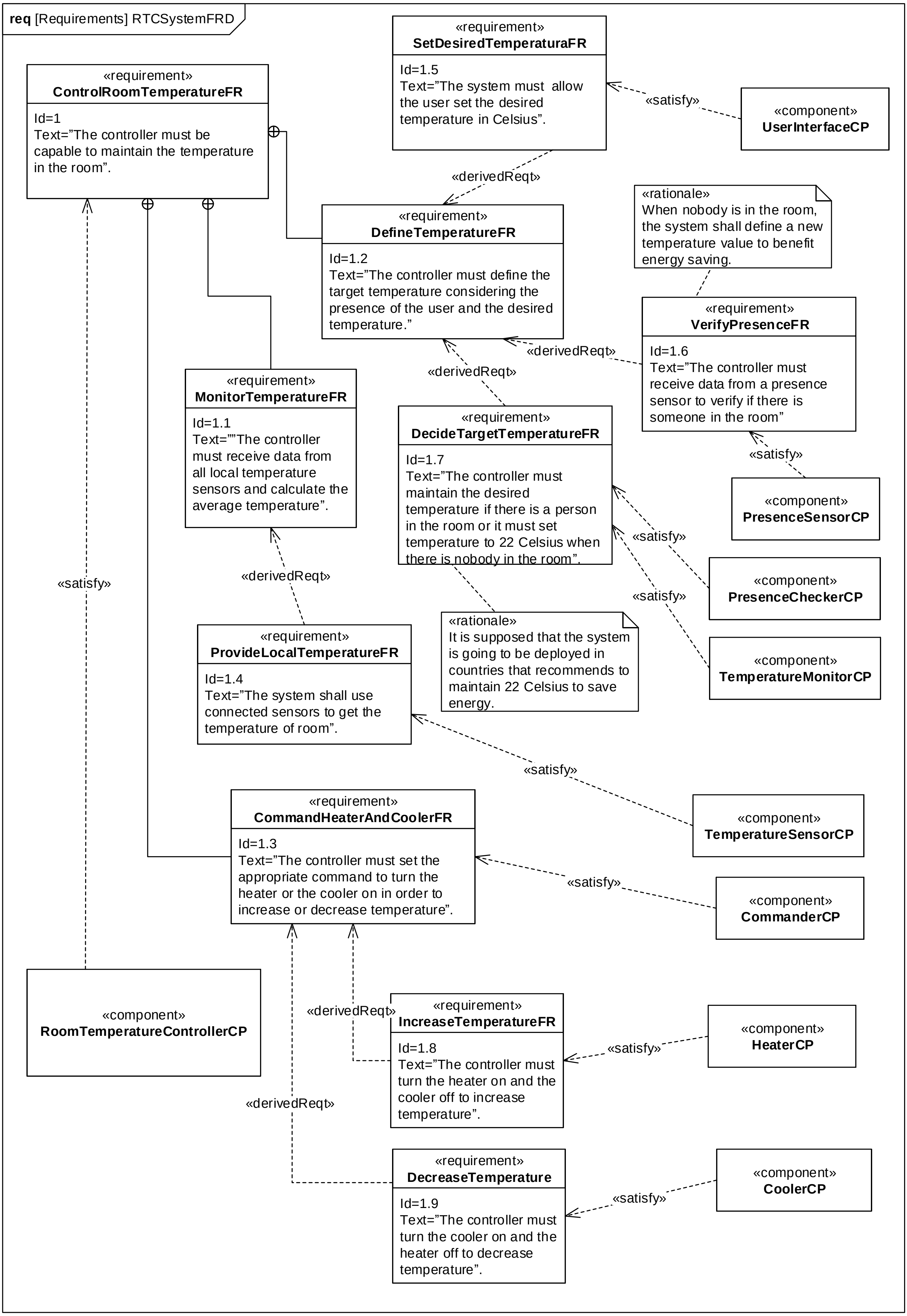

uc [UseCases] RTCSystemUCD
SystemAdministrator
ControlTemperatureUC
«include»
DefineTemperatureUC
«include»
MonitorTemperatureUC
«include»
CommandHeaterAndCoolerUC
«include»
VerifyPresenceUC
«include»
SetDesiredTemperatureUC
User



req [Requirements] RTCSystemFRD

«requirement»
ControlRoomTemperatureFR
Id=1
Text=”The controller must be capable to maintain the temperature in the room”.

«requirement»
SetDesiredTemperaturaFR
Id=1.5
Text=”The system must allow the user set the desired temperature in Celsius”.

«component»
UserInterfaceCP

«requirement»
DefineTemperatureFR
Id=1.2
Text=”The controller must define the target temperature considering the presence of the user and the desired temperature.”

«rationale»
When nobody is in the room, the system shall define a new temperature value to benefit energy saving.

«requirement»
VerifyPresenceFR
Id=1.6
Text=”The controller must receive data from a presence sensor to verify if there is someone in the room”

«requirement»
MonitorTemperatureFR
Id=1.1
Text=””The controller must receive data from all local temperature sensors and calculate the average temperature”.

«requirement»
DecideTargetTemperatureFR
Id=1.7
Text=”The controller must maintain the desired temperature if there is a person in the room or it must set temperature to 22 Celsius when there is nobody in the room”.

«component»
PresenceSensorCP

«component»
PresenceCheckerCP

«component»
TemperatureMonitorCP

«requirement»
ProvideLocalTemperatureFR
Id=1.4
Text=”The system shall use connected sensors to get the temperature of room”.

«rationale»
It is supposed that the system is going to be deployed in countries that recommends to maintain 22 Celsius to save energy.

«requirement»
CommandHeaterAndCoolerFR
Id=1.3
Text=”The controller must set the appropriate command to turn the heater or the cooler on in order to increase or decrease temperature”.

«component»
TemperatureSensorCP

«component»
CommanderCP

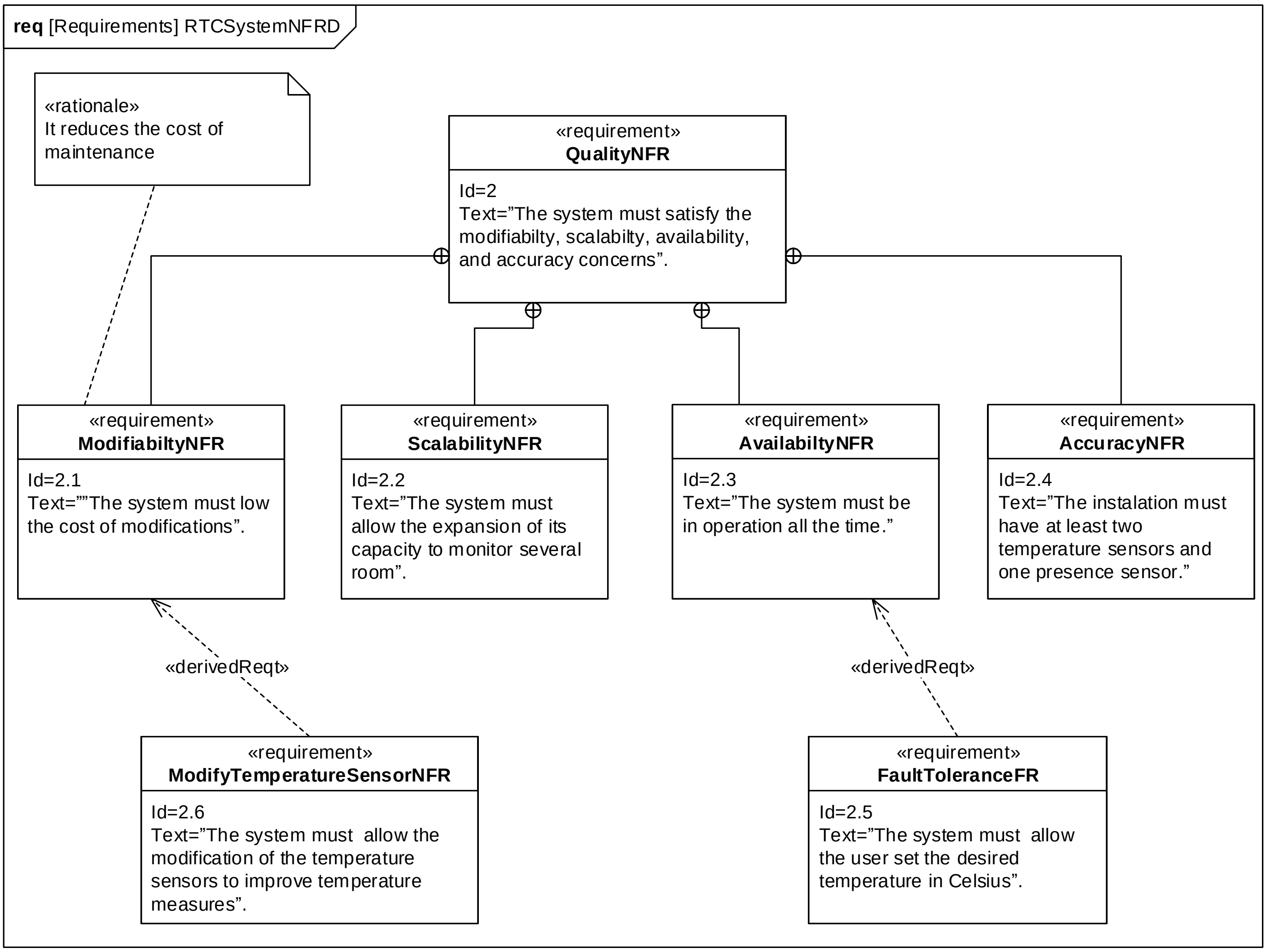
«component»
RoomTemperatureControllerCP

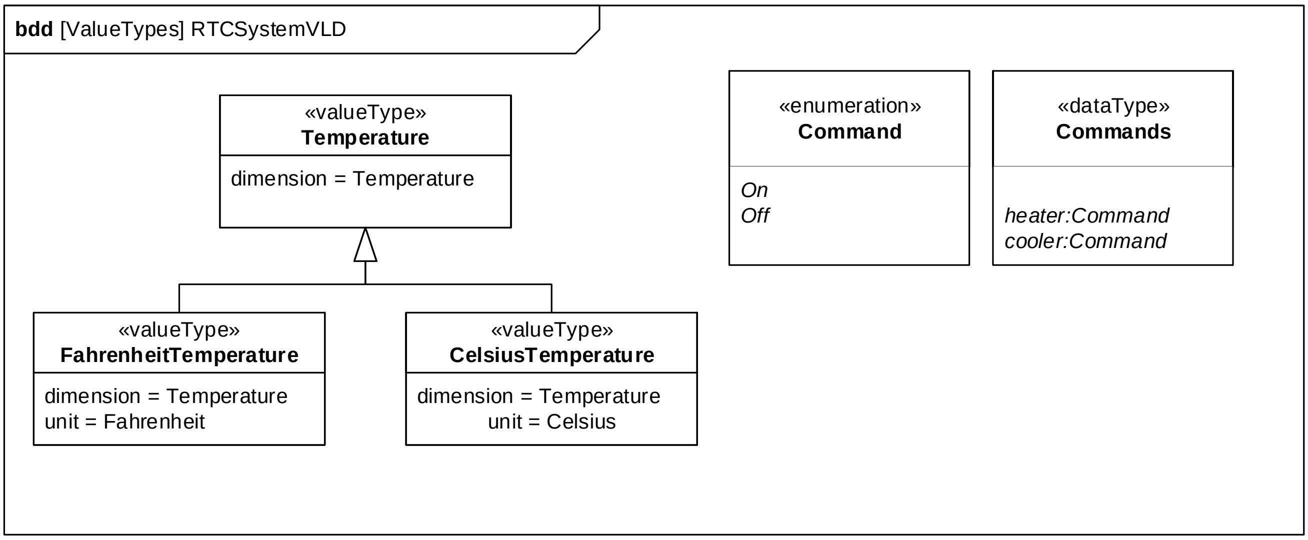
«requirement»
IncreaseTemperatureFR
Id=1.8
Text=”The controller must turn the heater on and the cooler off to increase temperature”.

«component»
HeaterCP

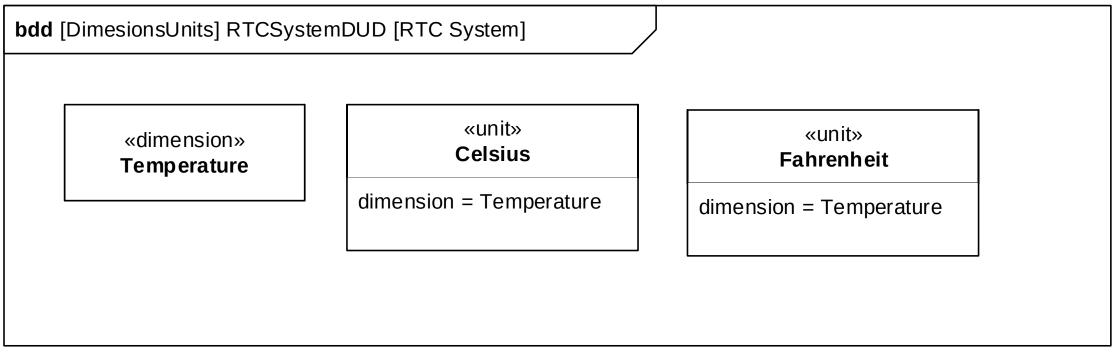
«requirement»
DecreaseTemperature
Id=1.9
Text=”The controller must turn the cooler on and the heater off to decrease temperature”.

«component»
CoolerCP

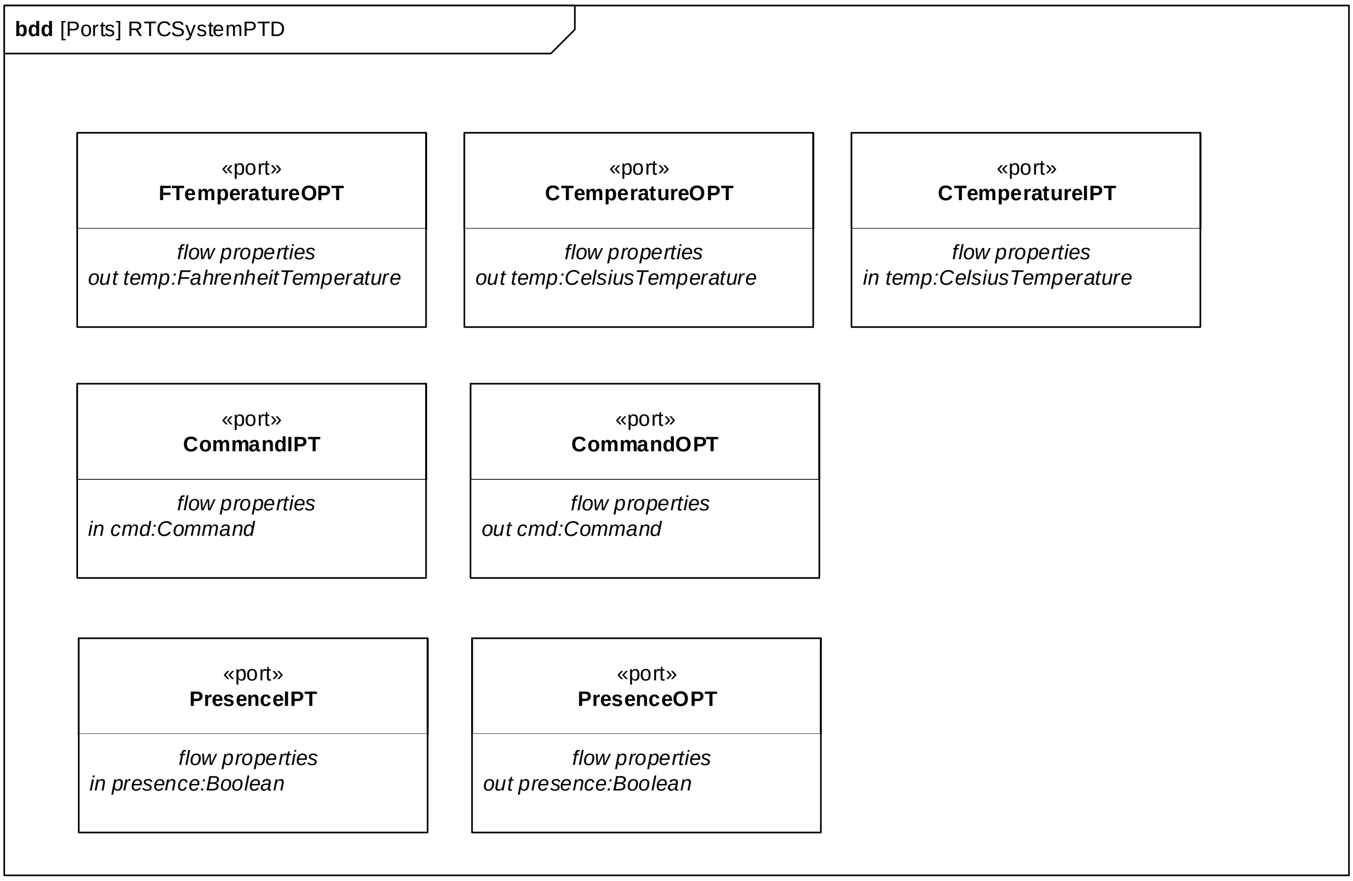

req [Requirements] RTCSystemNFRD
«rationale»
It reduces the cost of maintenance
«requirement»
QualityNFR
Id=2
Text=”The system must satisfy the modifiabilty, scalabilty, availability, and accuracy concerns”.
«requirement»
ModifiabiltyNFR
Id=2.1
Text=””The system must low the cost of modifications”.
«requirement»
ScalabilityNFR
Id=2.2
Text=”The system must allow the expansion of its capacity to monitor several room”.
«requirement»
AvailabiltyNFR
Id=2.3
Text=”The system must be in operation all the time.”
«requirement»
AccuracyNFR
Id=2.4
Text=”The instalation must have at least two temperature sensors and one presence sensor.”
«derivedReqt»
«derivedReqt»
«requirement»
ModifyTemperatureSensorNFR
Id=2.6
Text=”The system must allow the modification of the temperature sensors to improve temperature measures”.
«requirement»
FaultToleranceFR
Id=2.5
Text=”The system must allow the user set the desired temperature in Celsius”.

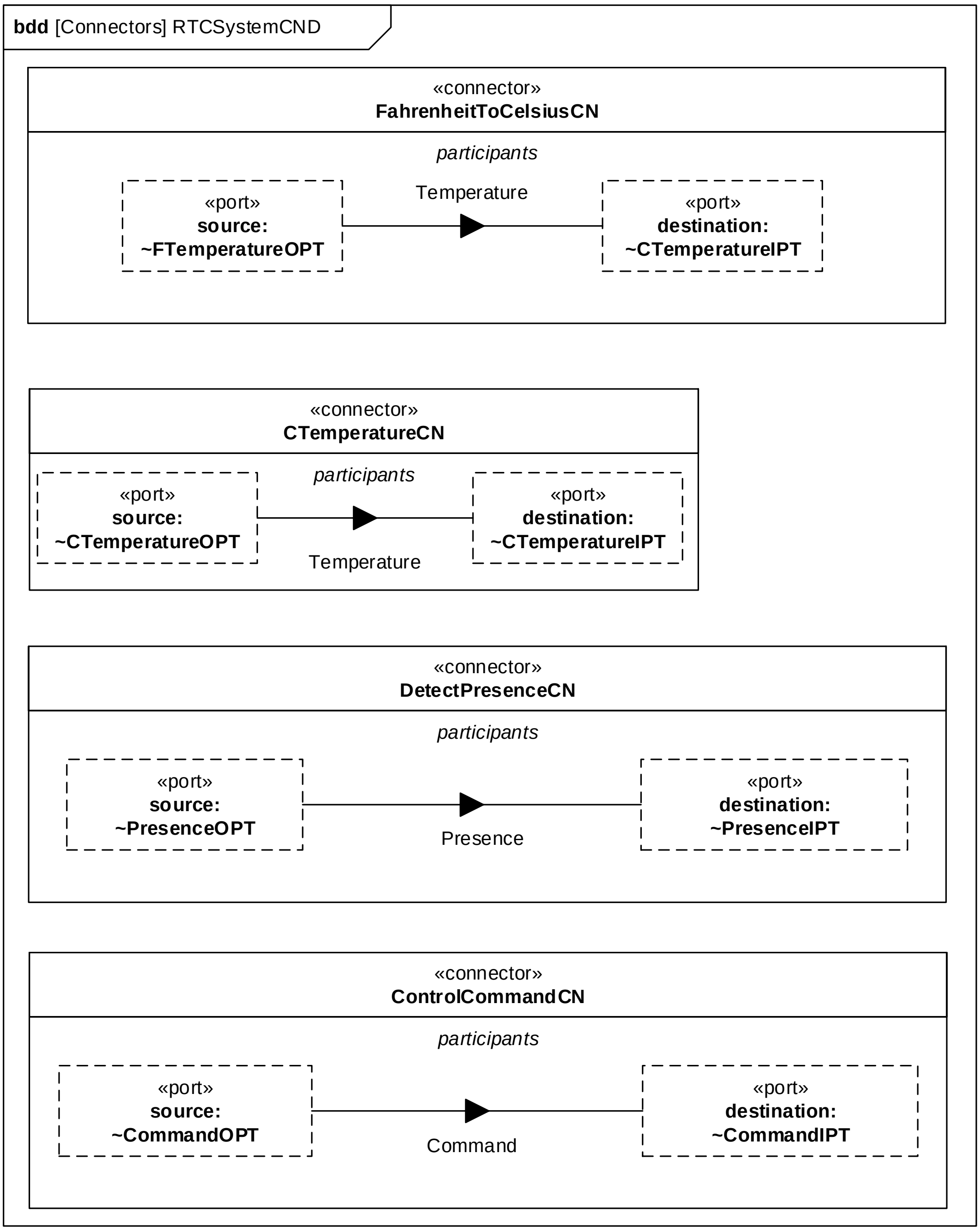


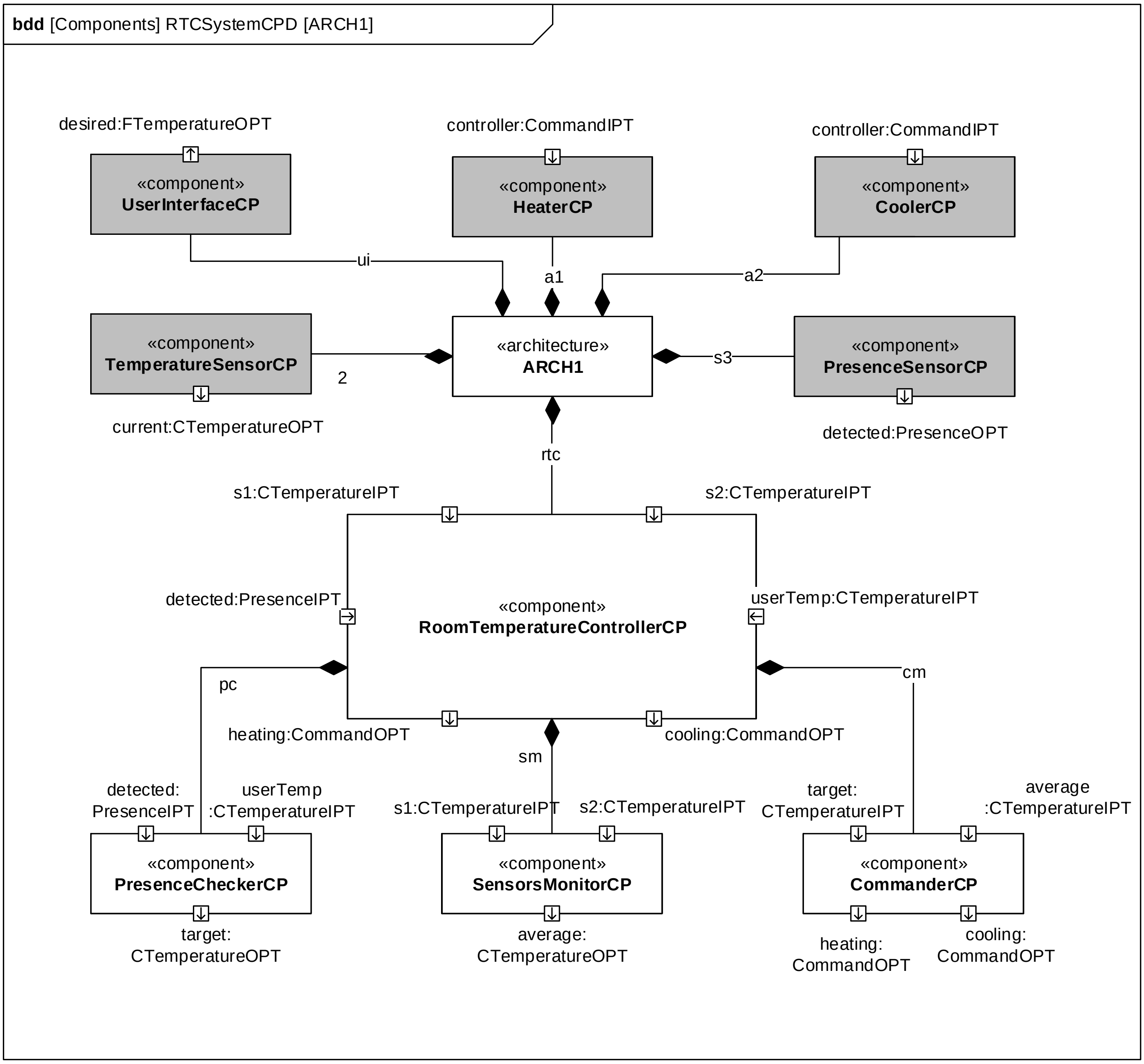
bdd [ValueTypes] RTCSystemVLD
«valueType»
Temperature
dimension = Temperature
«valueType»
FahrenheitTemperature
dimension = Temperature
unit = Fahrenheit
«valueType»
CelsiusTemperature
dimension = Temperature
unit = Celsius
«enumeration»
Command
On
Off
«dataType»
Commands
heater:Command
cooler:Command



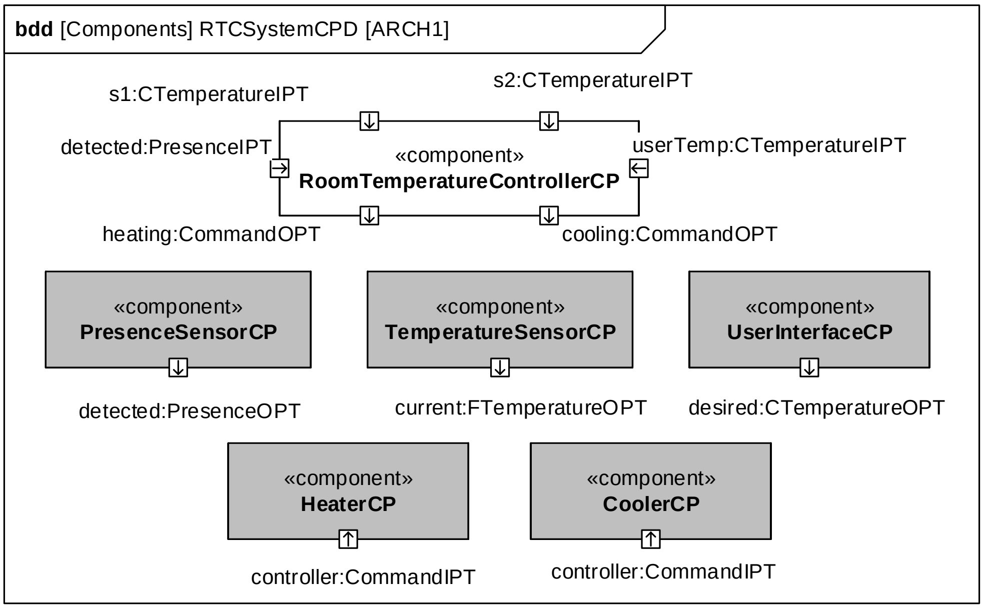
bdd [DimesionsUnits] RTCSystemDUD [RTC System]
«dimension»
Temperature
«unit»
Celsius
dimension = Temperature
«unit»
Fahrenheit
dimension = Temperature

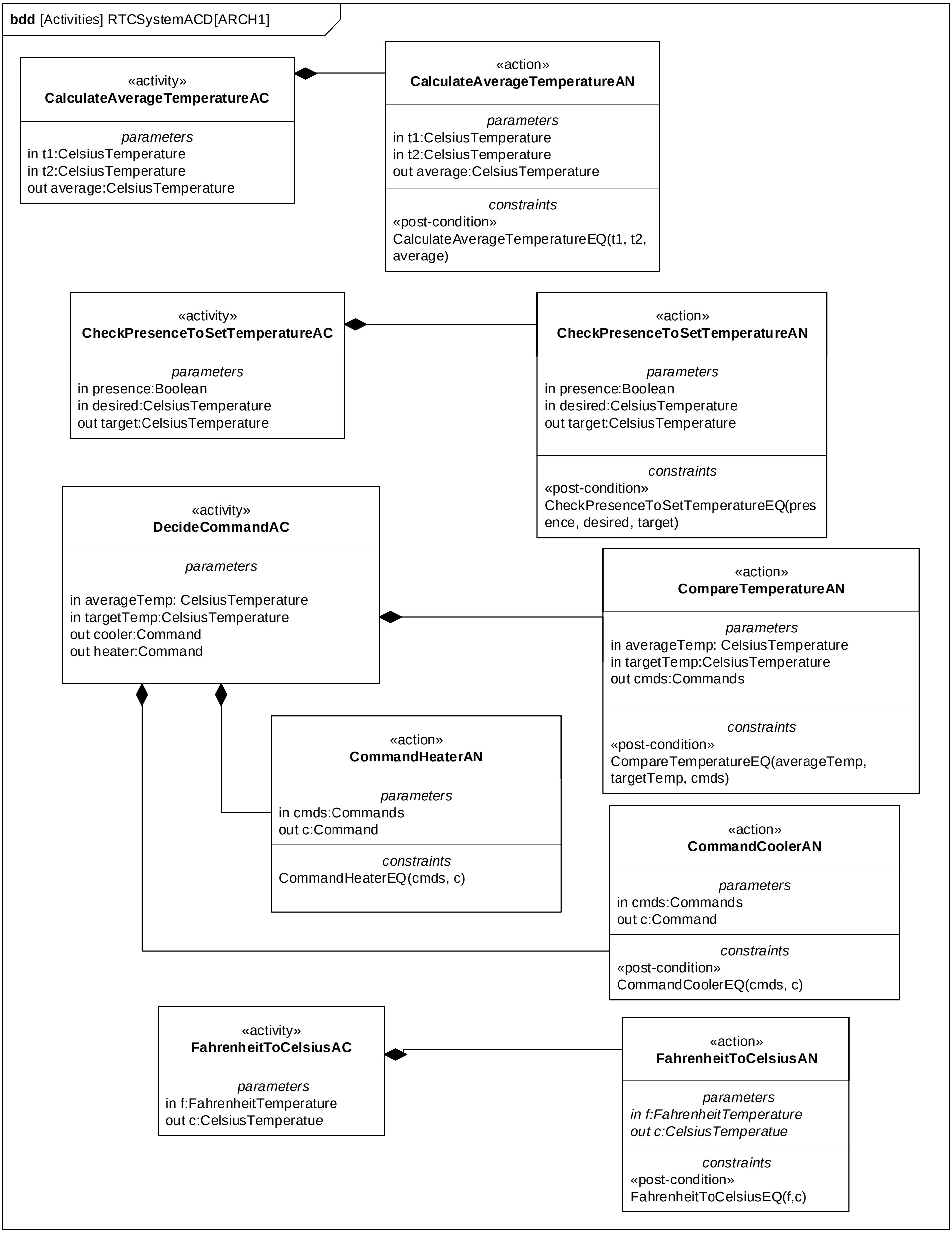

bdd [Ports] RTCSystemPTD
«port»
FTemperatureOPT
flow properties
out temp:FahrenheitTemperature
«port»
CTemperatureOPT
flow properties
out temp:CelsiusTemperature
«port»
CTemperatureIPT
flow properties
in temp:CelsiusTemperature
«port»
CommandIPT
flow properties
in cmd:Command
«port»
CommandOPT
flow properties
out cmd:Command
«port»
PresenceIPT
flow properties
in presence:Boolean
«port»
PresenceOPT
flow properties
out presence:Boolean

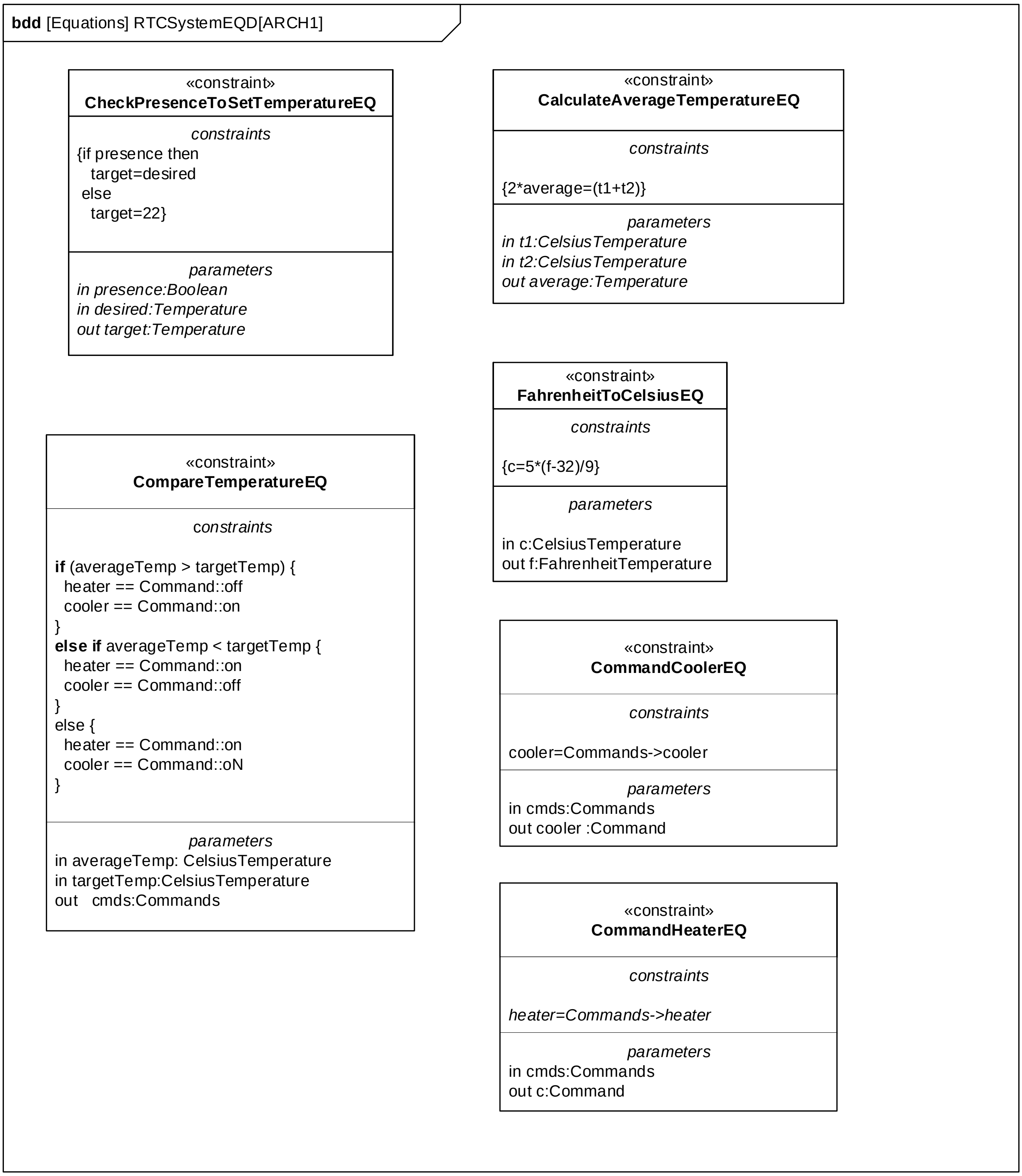

bdd [Connectors] RTCSystemCND
«connector»
FahrenheitToCelsiusCN
participants
«port»
source:
~FTemperatureOPT
Temperature
«port»
destination:
~CTemperatureIPT
«connector»
CTemperatureCN
participants
«port»
source:
~CTemperatureOPT
Temperature
«port»
destination:
~CTemperatureIPT
«connector»
DetectPresenceCN
participants
«port»
source:
~PresenceOPT
Presence
«port»
destination:
~PresenceIPT
«connector»
ControlCommandCN
participants
«port»
source:
~CommandOPT
Command
«port»
destination:
~CommandIPT



bdd [Components] RTCSystemCPD [ARCH1]
desired:FTemperatureOPT
«component»
UserInterfaceCP
controller:CommandIPT
«component»
HeaterCP
controller:CommandIPT
«component»
CoolerCP
ui
a1
a2
«component»
TemperatureSensorCP
2
current:CTemperatureOPT
«architecture»
ARCH1
s3
«component»
PresenceSensorCP
detected:PresenceOPT
rtc
s1:CTemperatureIPT
s2:CTemperatureIPT
detected:PresenceIPT
«component»
RoomTemperatureControllerCP
userTemp:CTemperatureIPT
pc
cm
heating:CommandOPT
cooling:CommandOPT
sm
detected:
PresenceIPT
userTemp
:CTemperatureIPT
s1:CTemperatureIPT
s2:CTemperatureIPT
target:
CTemperatureIPT
average
:CTemperatureIPT
«component»
PresenceCheckerCP
«component»
SensorsMonitorCP
«component»
CommanderCP
target:
CTemperatureOPT
average:
CTemperatureOPT
heating:
CommandOPT
cooling:
CommandOPT


bdd [Components] RTCSystemCPD [ARCH1]
s1:CTemperatureIPT
s2:CTemperatureIPT
detected:PresenceIPT
«component»
RoomTemperatureControllerCP
userTemp:CTemperatureIPT
heating:CommandOPT
cooling:CommandOPT
«component»
PresenceSensorCP
«component»
TemperatureSensorCP
«component»
UserInterfaceCP
detected:PresenceOPT
current:FTemperatureOPT
desired:CTemperatureOPT
«component»
HeaterCP
«component»
CoolerCP
controller:CommandIPT
controller:CommandIPT


bdd [Activities] RTCSystemACD[ARCH1]
«activity»
CalculateAverageTemperatureAC
parameters
in t1:CelsiusTemperature
in t2:CelsiusTemperature
out average:CelsiusTemperature
«action»
CalculateAverageTemperatureAN
parameters
in t1:CelsiusTemperature
in t2:CelsiusTemperature
out average:CelsiusTemperature
constraints
«post-condition»
CalculateAverageTemperatureEQ(t1, t2, average)
«activity»
CheckPresenceToSetTemperatureAC
parameters
in presence:Boolean
in desired:CelsiusTemperature
out target:CelsiusTemperature
«action»
CheckPresenceToSetTemperatureAN
parameters
in presence:Boolean
in desired:CelsiusTemperature
out target:CelsiusTemperature
constraints
«post-condition»
CheckPresenceToSetTemperatureEQ(pres
ence, desired, target)
«activity»
DecideCommandAC
parameters
in averageTemp: CelsiusTemperature
in targetTemp:CelsiusTemperature
out cooler:Command
out heater:Command
«action»
CompareTemperatureAN
parameters
in averageTemp: CelsiusTemperature
in targetTemp:CelsiusTemperature
out cmds:Commands
constraints
«post-condition»
CompareTemperatureEQ(averageTemp, targetTemp, cmds)
«action»
CommandHeaterAN
parameters
in cmds:Commands
out c:Command
constraints
CommandHeaterEQ(cmds, c)
«action»
CommandCoolerAN
parameters
in cmds:Commands
out c:Command
constraints
«post-condition»
CommandCoolerEQ(cmds, c)
«activity»
FahrenheitToCelsiusAC
parameters
in f:FahrenheitTemperature
out c:CelsiusTemperatue
«action»
FahrenheitToCelsiusAN
parameters
in f:FahrenheitTemperature
out c:CelsiusTemperatue
constraints
«post-condition»
FahrenheitToCelsiusEQ(f,c)



bdd [Equations] RTCSystemEQD[ARCH1]

«constraint»
CheckPresenceToSetTemperatureEQ

constraints

```
{if presence then
   target=desired
 else
   target=22}
```

parameters

in presence:Boolean
in desired:Temperature
out target:Temperature

«constraint»
CalculateAverageTemperatureEQ

constraints

{2*average=(t1+t2)}

parameters

in t1:CelsiusTemperature
in t2:CelsiusTemperature
out average:Temperature

«constraint»
FahrenheitToCelsiusEQ

constraints

{c=5*(f-32)/9}

parameters

in c:CelsiusTemperature
out f:FahrenheitTemperature

«constraint»
CompareTemperatureEQ

constraints

```
if (averageTemp > targetTemp) {
  heater == Command::off
  cooler == Command::on
}
else if averageTemp < targetTemp {
  heater == Command::on
  cooler == Command::off
}
else {
  heater == Command::on
  cooler == Command::oN
}
```

parameters

in averageTemp: CelsiusTemperature
in targetTemp:CelsiusTemperature
out   cmds:Commands

«constraint»
CommandCoolerEQ

constraints

cooler=Commands->cooler

parameters

in cmds:Commands
out cooler :Command

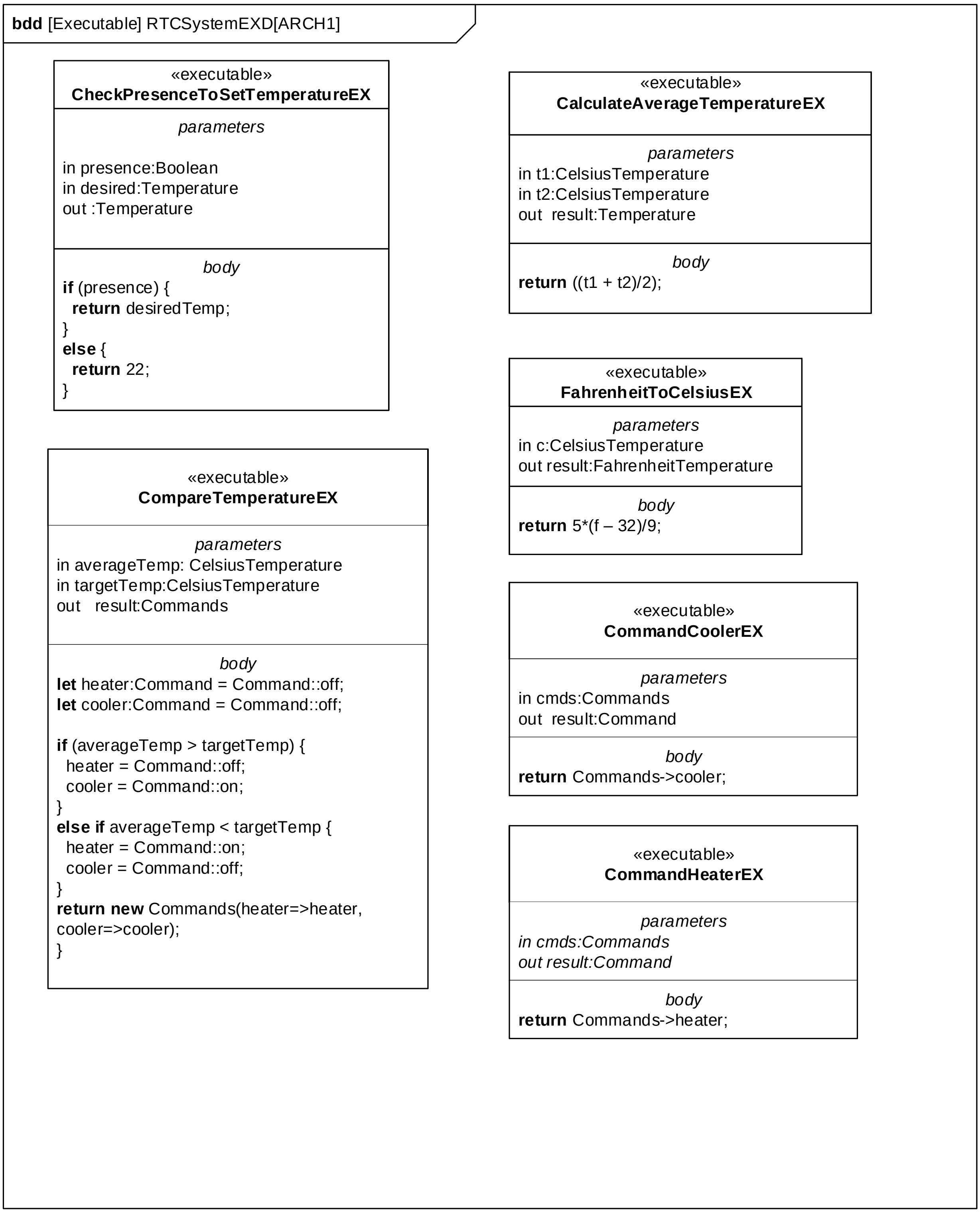
«constraint»
CommandHeaterEQ

constraints

heater=Commands->heater

parameters

in cmds:Commands
out c:Command



bdd [Executable] RTCSystemEXD[ARCH1]

«executable»
CheckPresenceToSetTemperatureEX

parameters

in presence:Boolean
in desired:Temperature
out :Temperature

body

```
if (presence) {
  return desiredTemp;
}
else {
  return 22;
}
```

«executable»
CalculateAverageTemperatureEX

parameters
in t1:CelsiusTemperature
in t2:CelsiusTemperature
out result:Temperature

body

```
return ((t1 + t2)/2);
```

«executable»
FahrenheitToCelsiusEX

parameters
in c:CelsiusTemperature
out result:FahrenheitTemperature

body

```
return 5*(f – 32)/9;
```

«executable»
CompareTemperatureEX

parameters
in averageTemp: CelsiusTemperature
in targetTemp:CelsiusTemperature
out   result:Commands

body

```
let heater:Command = Command::off;
let cooler:Command = Command::off;

if (averageTemp > targetTemp) {
  heater = Command::off;
  cooler = Command::on;
}
else if averageTemp < targetTemp {
  heater = Command::on;
  cooler = Command::off;
}
return new Commands(heater=>heater,
cooler=>cooler);
}
```

«executable»
CommandCoolerEX

parameters
in cmds:Commands
out result:Command

body

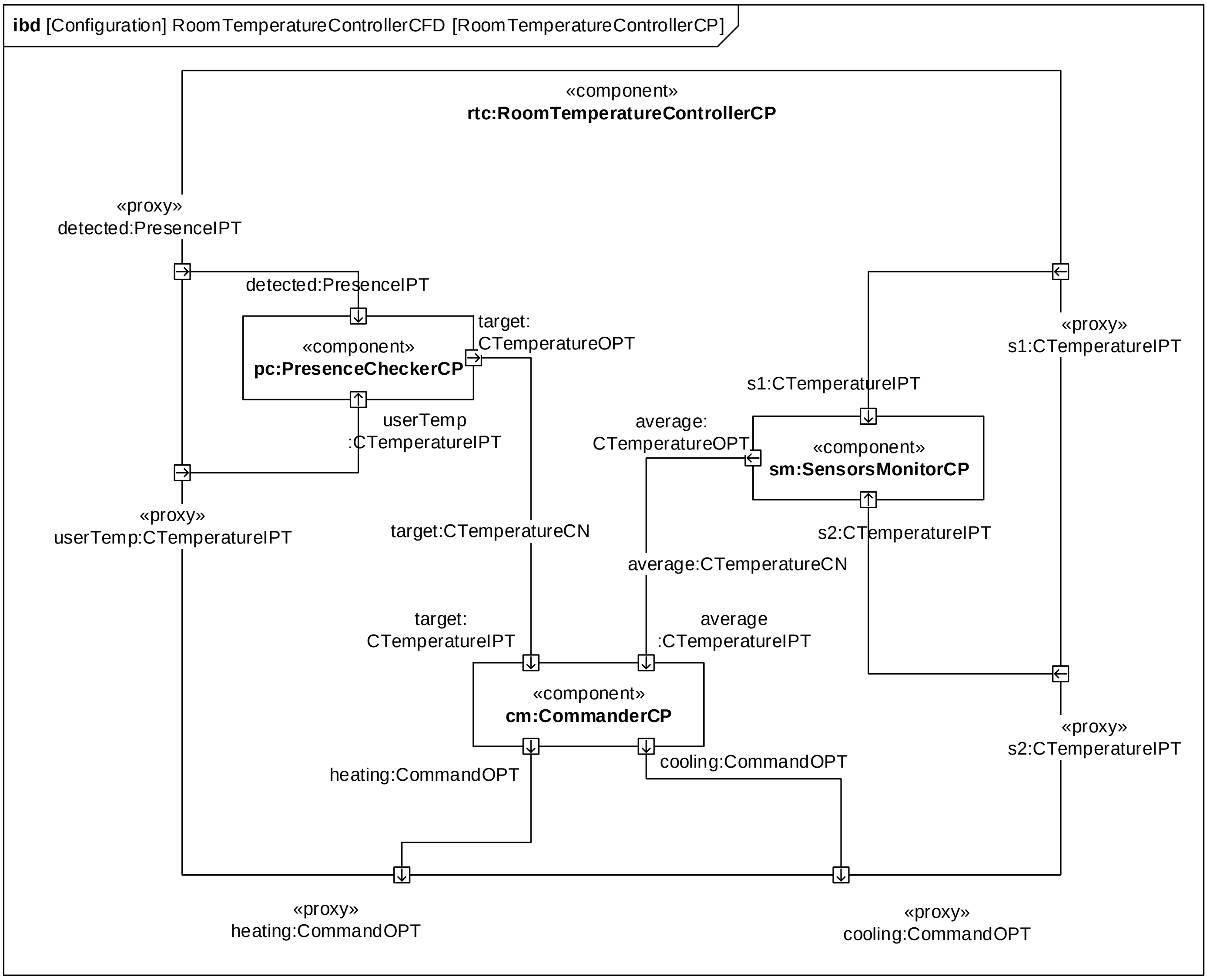
```
return Commands->cooler;
```

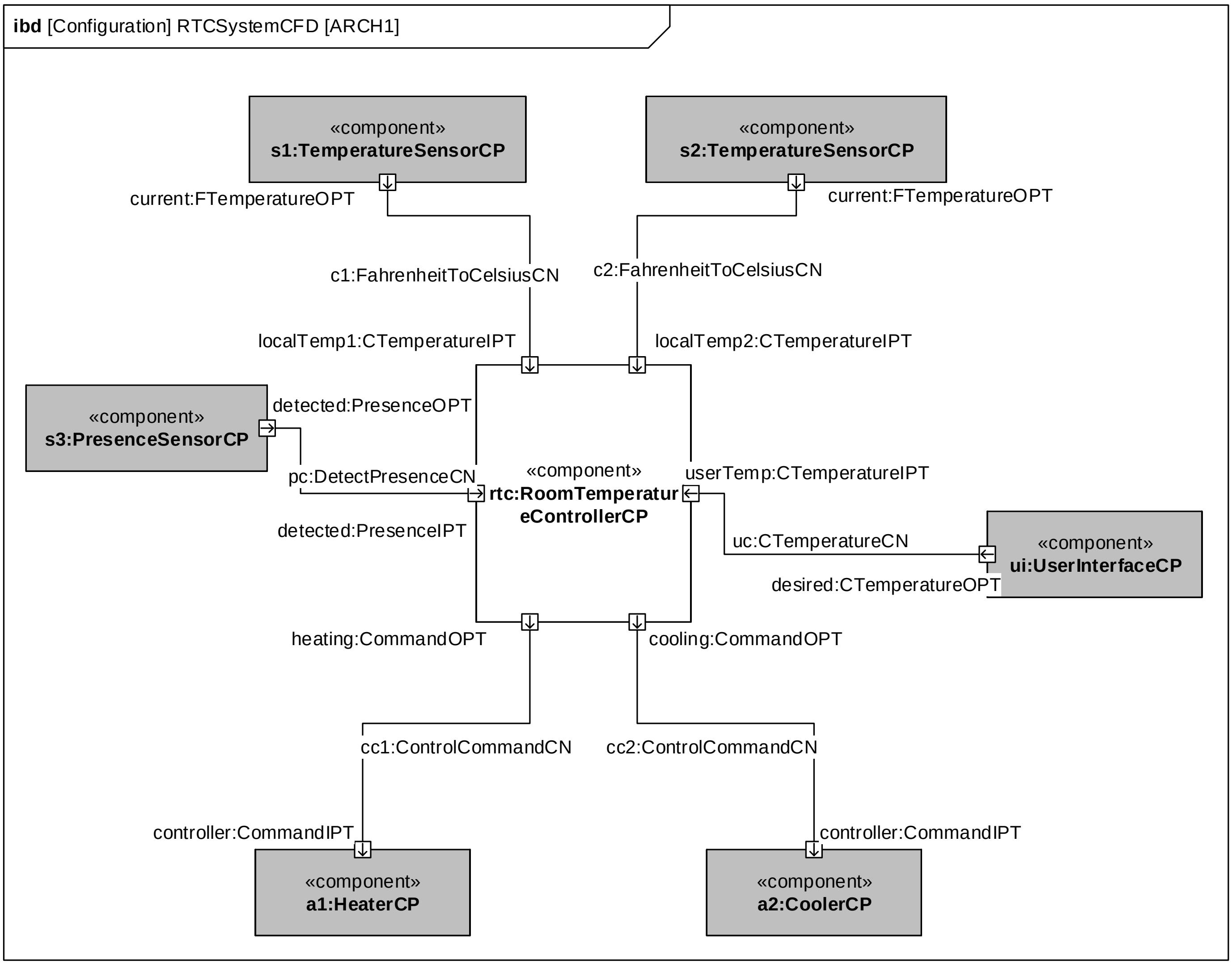
«executable»
CommandHeaterEX

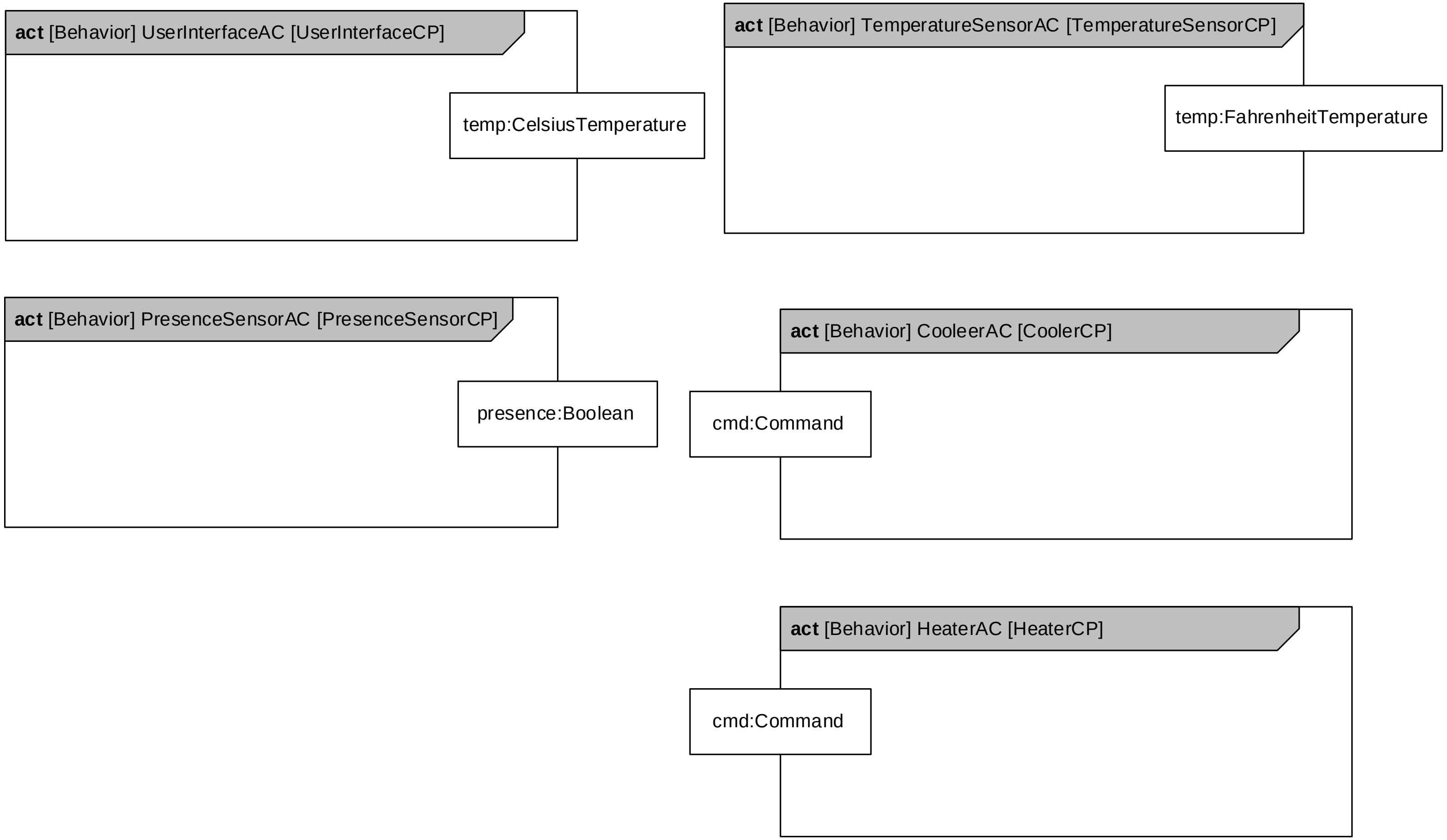
parameters
in cmds:Commands
out result:Command

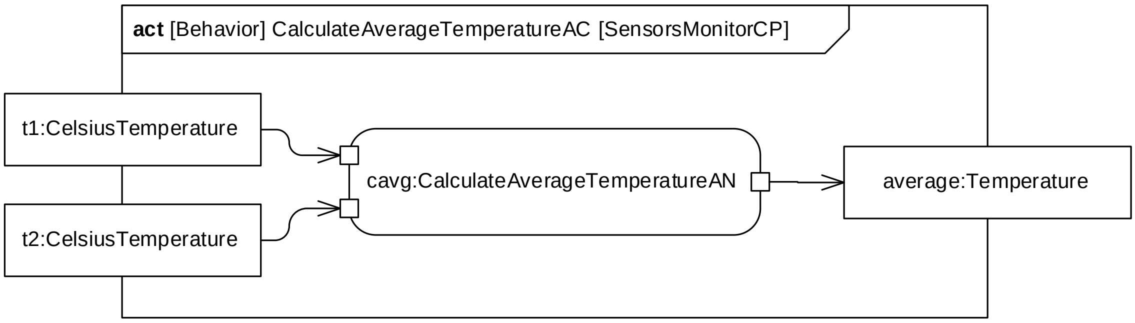
body

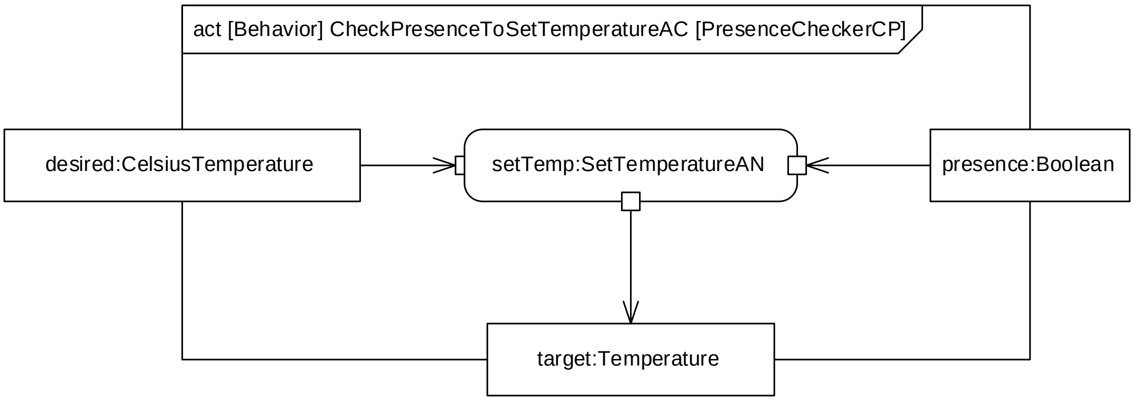
```
return Commands->heater;
```

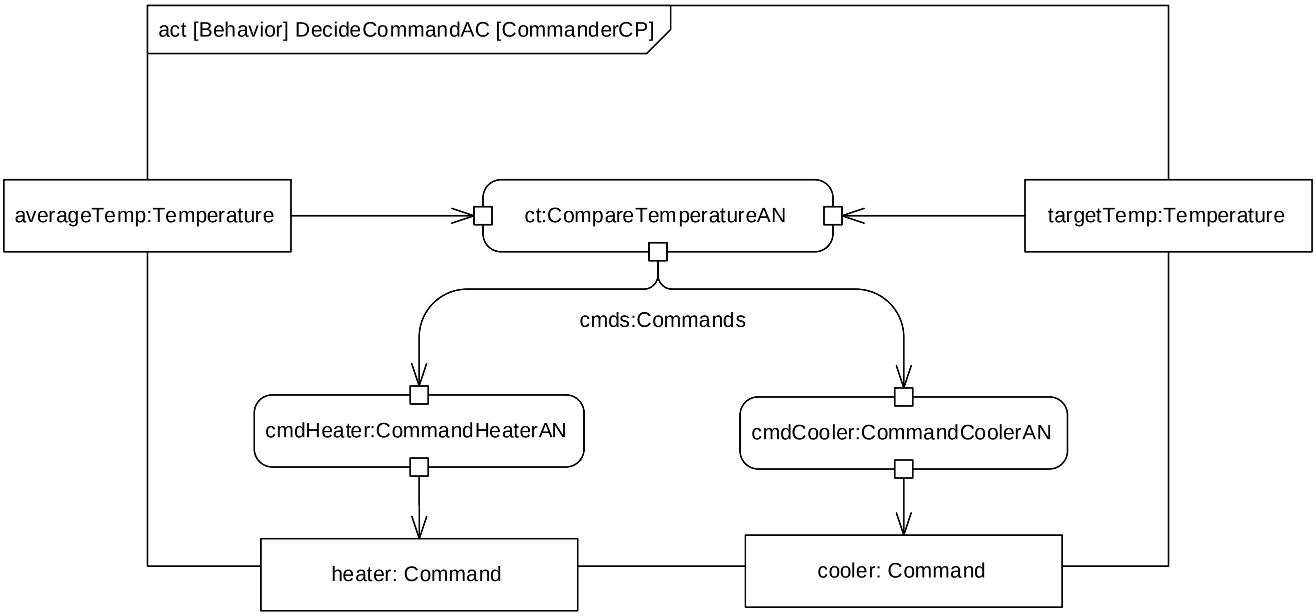

ibd [Configuration] RoomTemperatureControllerCFD [RoomTemperatureControllerCP]
«component»
rtc:RoomTemperatureControllerCP
«proxy»
detected:PresenceIPT
detected:PresenceIPT
target:
CTemperatureOPT
«component»
pc:PresenceCheckerCP
userTemp
:CTemperatureIPT
«proxy»
userTemp:CTemperatureIPT
target:CTemperatureCN
«proxy»
s1:CTemperatureIPT
s1:CTemperatureIPT
average:
CTemperatureOPT
«component»
sm:SensorsMonitorCP
s2:CTemperatureIPT
average:CTemperatureCN
target:
CTemperatureIPT
average
:CTemperatureIPT
«component»
cm:CommanderCP
heating:CommandOPT
cooling:CommandOPT
«proxy»
s2:CTemperatureIPT
«proxy»
heating:CommandOPT
«proxy»
cooling:CommandOPT

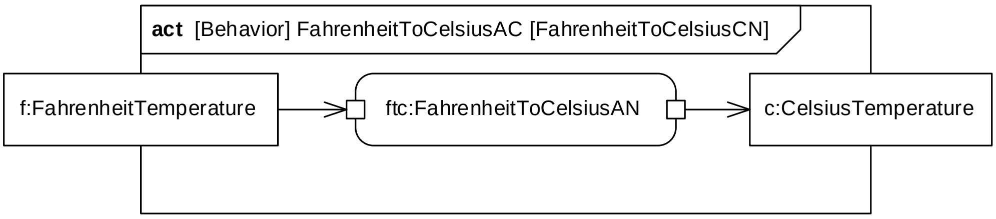
ibd [Configuration] RTCSystemCFD [ARCH1]
«component»
s1:TemperatureSensorCP
«component»
s2:TemperatureSensorCP
current:FTemperatureOPT
current:FTemperatureOPT
c1:FahrenheitToCelsiusCN
c2:FahrenheitToCelsiusCN
localTemp1:CTemperatureIPT
localTemp2:CTemperatureIPT
«component»
s3:PresenceSensorCP
detected:PresenceOPT
pc:DetectPresenceCN
detected:PresenceIPT
«component»
rtc:RoomTemperatureControllerCP
userTemp:CTemperatureIPT
uc:CTemperatureCN
«component»
ui:UserInterfaceCP
desired:CTemperatureOPT
heating:CommandOPT
cooling:CommandOPT
cc1:ControlCommandCN
cc2:ControlCommandCN
controller:CommandIPT
controller:CommandIPT
«component»
a1:HeaterCP
«component»
a2:CoolerCP

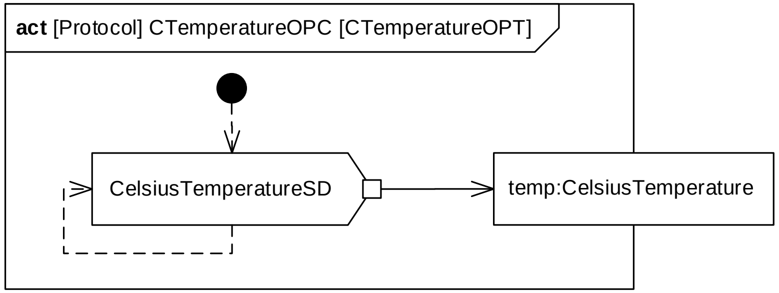

act [Behavior] UserInterfaceAC [UserInterfaceCP]
temp:CelsiusTemperature
act [Behavior] TemperatureSensorAC [TemperatureSensorCP]
temp:FahrenheitTemperature
act [Behavior] PresenceSensorAC [PresenceSensorCP]
presence:Boolean
act [Behavior] CooleerAC [CoolerCP]
cmd:Command
act [Behavior] HeaterAC [HeaterCP]
cmd:Command

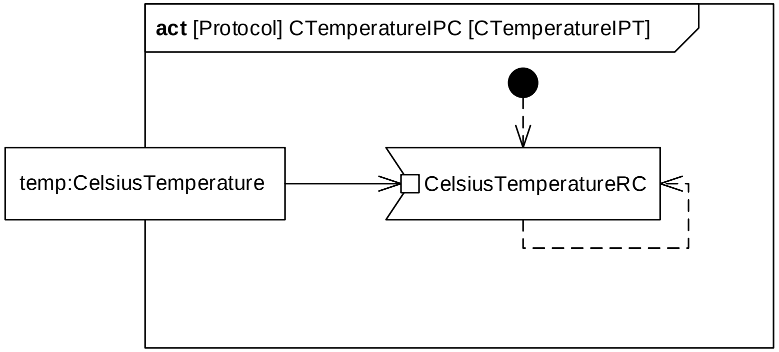

act [Behavior] CalculateAverageTemperatureAC [SensorsMonitorCP]
t1:CelsiusTemperature
t2:CelsiusTemperature
cavg:CalculateAverageTemperatureAN
average:Temperature

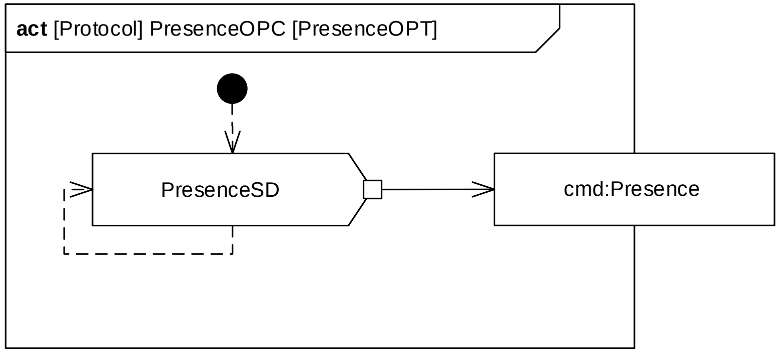

act [Behavior] CheckPresenceToSetTemperatureAC [PresenceCheckerCP]
desired:CelsiusTemperature
setTemp:SetTemperatureAN
presence:Boolean
target:Temperature

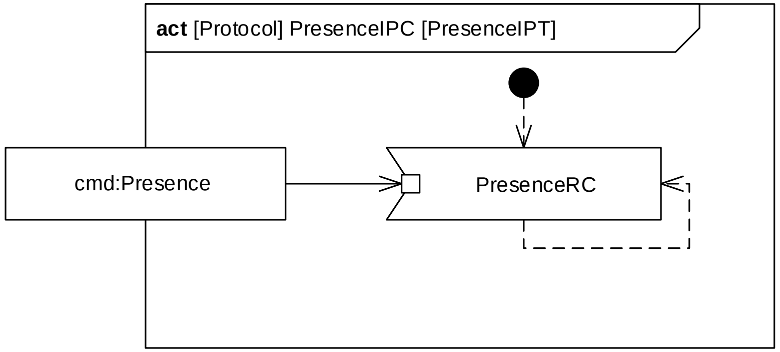

act [Behavior] DecideCommandAC [CommanderCP]
averageTemp:Temperature
ct:CompareTemperatureAN
targetTemp:Temperature
cmds:Commands
cmdHeater:CommandHeaterAN
cmdCooler:CommandCoolerAN
heater: Command
cooler: Command

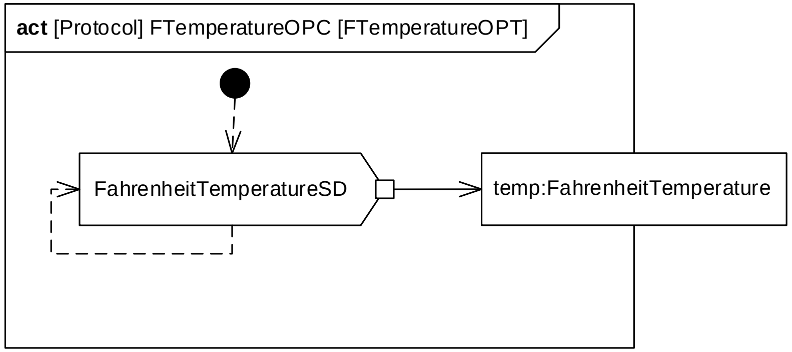

act [Behavior] FahrenheitToCelsiusAC [FahrenheitToCelsiusCN]
f:FahrenheitTemperature
ftc:FahrenheitToCelsiusAN
c:CelsiusTemperature

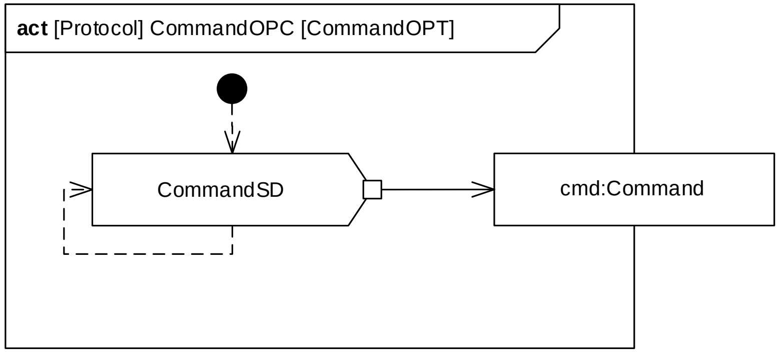

act [Protocol] CTemperatureOPC [CTemperatureOPT]
CelsiusTemperatureSD
temp:CelsiusTemperature

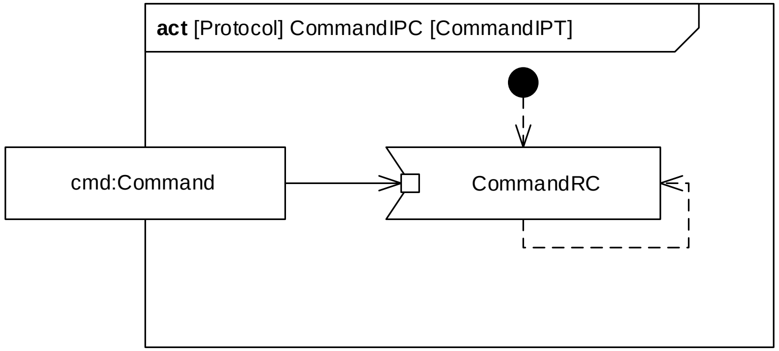

act [Protocol] CTemperatureIPC [CTemperatureIPT]
temp:CelsiusTemperature
CelsiusTemperatureRC

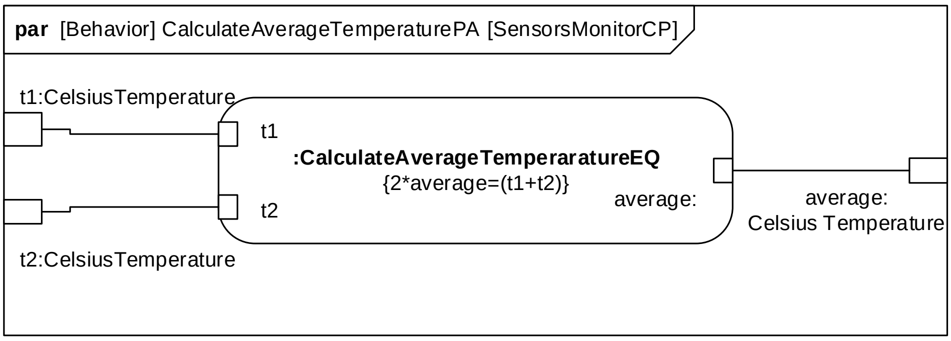

act [Protocol] PresenceOPC [PresenceOPT]
PresenceSD
cmd:Presence

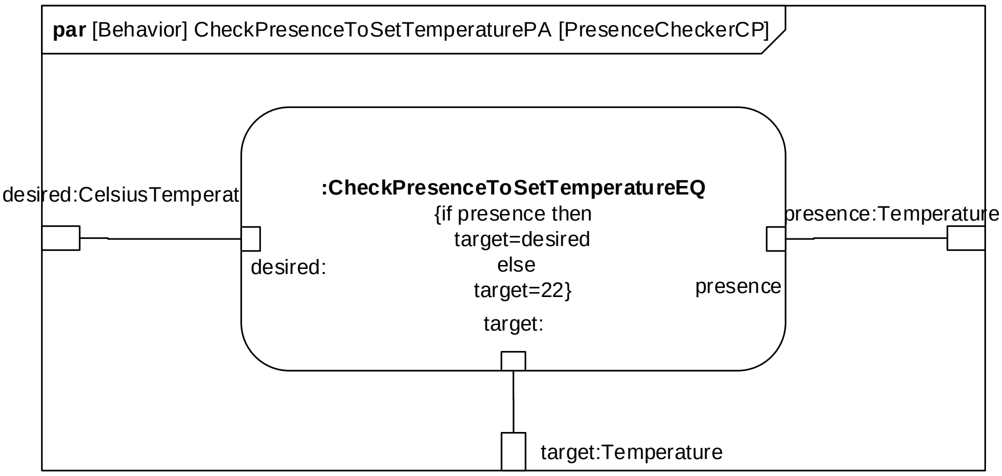

act [Protocol] PresenceIPC [PresenceIPT]
cmd:Presence
PresenceRC

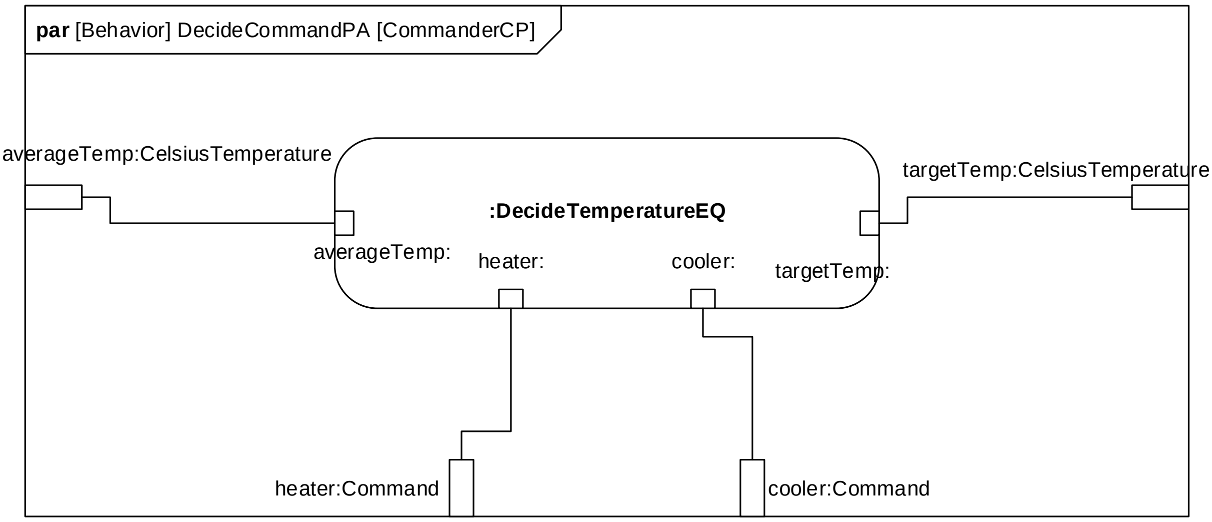

act [Protocol] FTemperatureOPC [FTemperatureOPT]
FahrenheitTemperatureSD
temp:FahrenheitTemperature

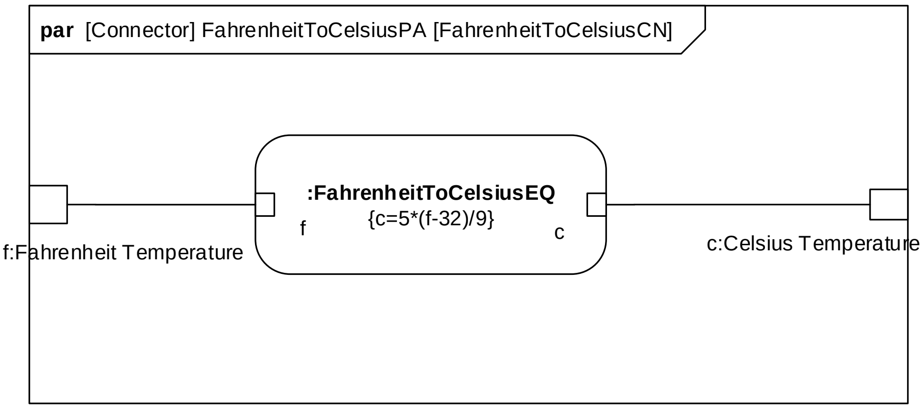

act [Protocol] CommandOPC [CommandOPT]
CommandSD
cmd:Command

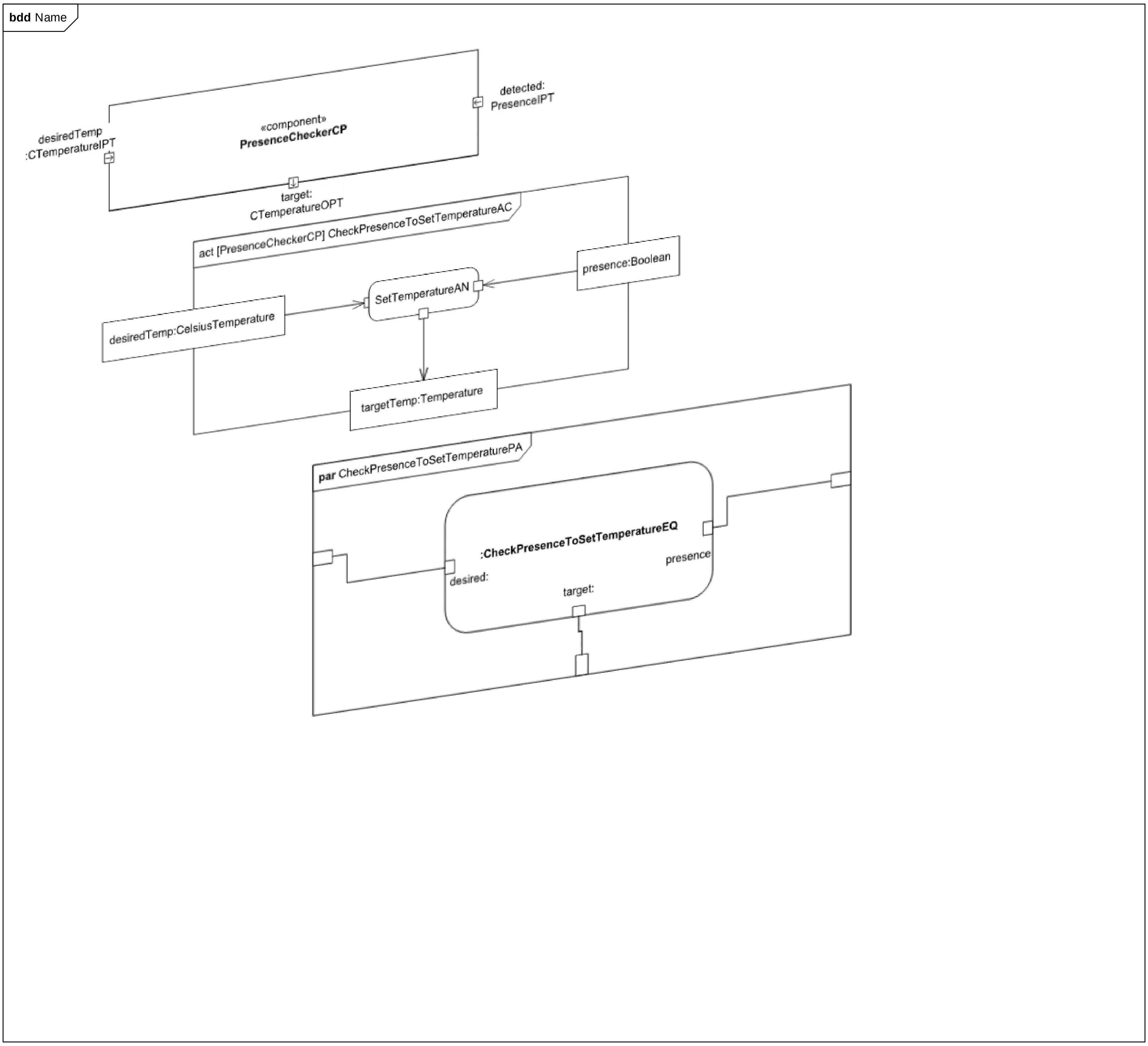

act [Protocol] CommandIPC [CommandIPT]
cmd:Command
CommandRC

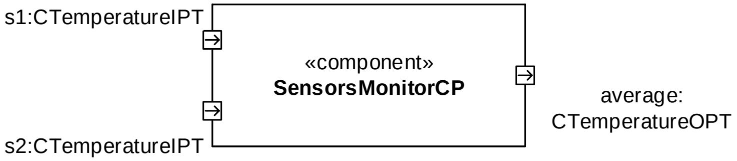

par [Behavior] CalculateAverageTemperaturePA [SensorsMonitorCP]
t1:CelsiusTemperature
t1
:CalculateAverageTemperaratureEQ
{2*average=(t1+t2)}
average:
average:
Celsius Temperature
t2
t2:CelsiusTemperature

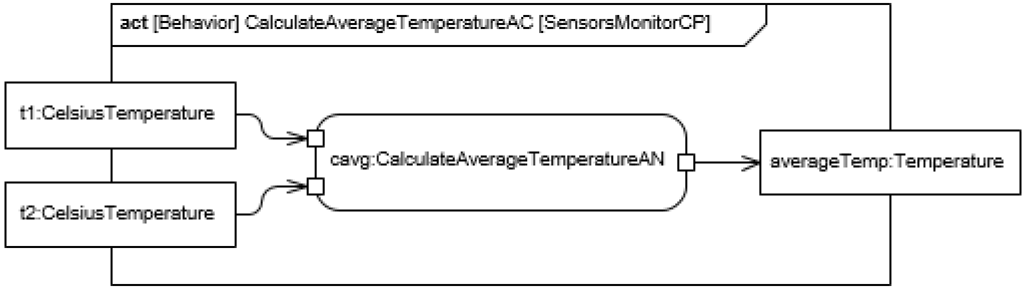

par [Behavior] CheckPresenceToSetTemperaturePA [PresenceCheckerCP]
desired:CelsiusTemperat
:CheckPresenceToSetTemperatureEQ
{if presence then
target=desired
else
target=22}
presence:Temperature
desired:
presence
target:
target:Temperature

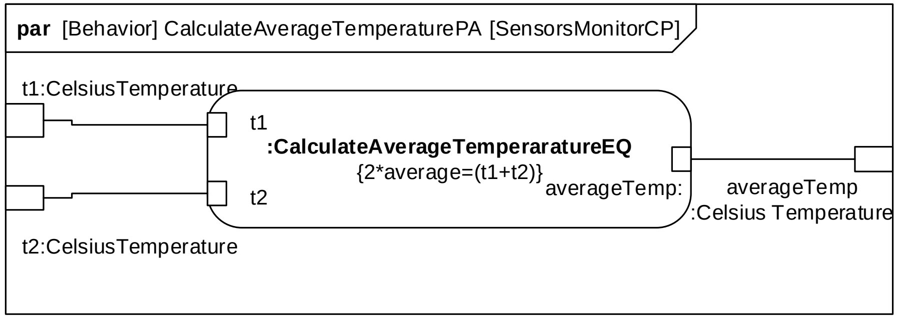

par [Behavior] DecideCommandPA [CommanderCP]
averageTemp:CelsiusTemperature
:DecideTemperatureEQ
targetTemp:CelsiusTemperature
averageTemp:
heater:
cooler:
targetTemp:
heater:Command
cooler:Command


par [Connector] FahrenheitToCelsiusPA [FahrenheitToCelsiusCN]
:FahrenheitToCelsiusEQ
{c=5*(f-32)/9}
f
c
f:Fahrenheit Temperature
c:Celsius Temperature


bdd Name
«component»
PresenceCheckerCP
desiredTemp
:CTemperatureIPT
detected:
PresenceIPT
target:
CTemperatureOPT
act [PresenceCheckerCP] CheckPresenceToSetTemperatureAC
SetTemperatureAN
presence:Boolean
desiredTemp:CelsiusTemperature
targetTemp:Temperature
par CheckPresenceToSetTemperaturePA
:CheckPresenceToSetTemperatureEQ
desired:
presence
target:


s1:CTemperatureIPT
«component»
SensorsMonitorCP
average:
CTemperatureOPT
s2:CTemperatureIPT


act [Behavior] CalculateAverageTemperatureAC [SensorsMonitorCP]
t1:CelsiusTemperature
t2:CelsiusTemperature
cavg:CalculateAverageTemperatureAN
averageTemp:Temperature


par [Behavior] CalculateAverageTemperaturePA [SensorsMonitorCP]
t1:CelsiusTemperature
t1
:CalculateAverageTemperaratureEQ
{2*average=(t1+t2)}
averageTemp:
t2
averageTemp
:Celsius Temperature
t2:CelsiusTemperature