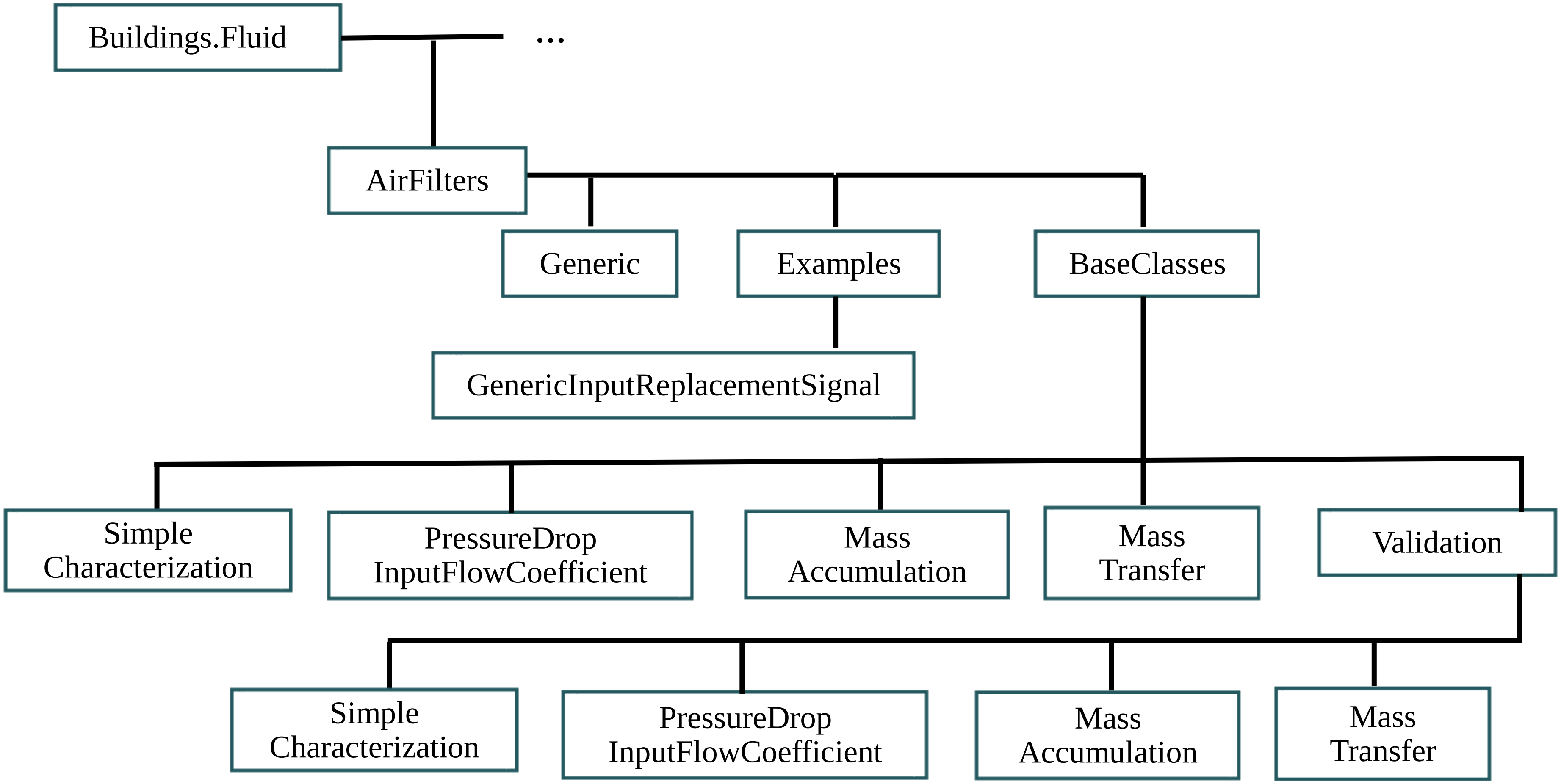
Model Structure



SimpleCharacterization

A block calculates the filter efficiency and the flow coefficient correction factor based on the mass of the contaminant held by the filter.

Parameters	Comment	Unit	Min	Max	Default
mCon_nominal	Nominal mass of the contaminant held by the filter	kg	1E-3	n/a	n/a
epsFun	Filter efficiency curve	1	n/a	n/a	{1}
b	Resistance coefficient	1	1+1E-3	n/a	1.5

Variables	Comment	Unit	Type
mCon	Mass of the contaminant held by the filter	kg	Input
Phi	Relative mass of the contaminant	1	Protected
eps	Filter efficiency	1	Output
kCor	Flow coefficient correction factor	1	Output

Model Implementations:

$$Phi = Buildings.Utilities.Math.Functions.smoothMin(x1 = 1, x2 = \frac{mCon}{mCon\_nominal}, deltaX = 0.1)$$

$$eps = Buildings.Utilities.Math.Functions.polynomial(epsFun, Phi)$$

$$kCor = b\verb|^|(Phi)$$

PressureDropInputFlowCoefficient

Model of a flow resistance with a varying flow coefficient.

Parameters	Comment	Unit	Min	Max	Default
m_flow_nominal	Nominal mass flow rate	kg/s	n/a	n/a	n/a
dp_nominal	Nominal pressure drop	Pa	n/a	n/a	n/a

Variables	Comment	Unit	Type
kCor	Flow coefficient correction factor	1	Input
k	Flow coefficient	(kg.m)^(1/2)	Protected
m_flow	Mass flow rate	kg/s	Output
dp	Pressure drop	Pa	Output

Model Implementations:

Leveraging the implementation of *Buildings.Fluid.FixedResistances.PressureDrop* with the following changes

- Changing k from a parameter into a variable

```
~~  final parameter Real k = if computeFlowResistance then~~
~~        m_flow_nominal_pos / sqrt(dp_nominal_pos) else 0~~
~~    "Flow coefficient, k=m_flow/sqrt(dp), with unit=(kg.m)^(1/2)";~~
 Real k "Flow coefficient, k=m_flow/sqrt(dp), with unit=(kg.m)^(1/2)";
```

PressureDropInputFlowCoefficient (Cont’l)

Model Implementations:

- Removing the intermediate variable coeff

```
   ~~final parameter Real coeff=~~
~~    if linearized and computeFlowResistance~~
~~    then if from_dp then k^2/m_flow_nominal_pos else m_flow_nominal_pos/k^2~~
~~    else 0~~
~~    "Precomputed coefficient to avoid division by parameter";~~
```

- Updating the equation

```
~~  if computeFlowResistance then~~
~~    if linearized then~~
~~      if from_dp then~~
~~        m_flow = dp*coeff;~~
~~      else~~
~~        dp = m_flow*coeff;~~
~~      end if;~~

  if computeFlowResistance then
    k = m_flow_nominal_pos / sqrt(dp_nominal_pos/kCor);
    if linearized then
      if from_dp then
        m_flow = dp*(k^2/ m_flow_nominal_pos);
      else
        dp = m_flow * (m_flow_nominal_pos /k^2);
      end if;
```

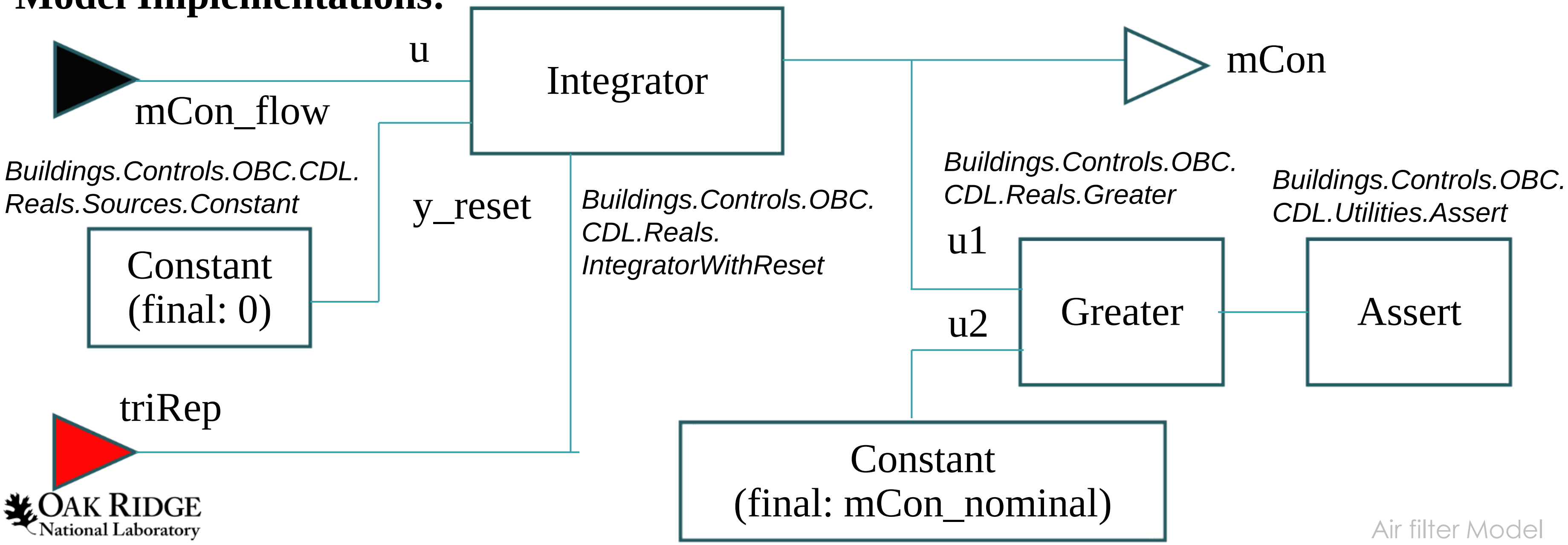
MassAccumulation

A block calculates the mass of the contaminant held by the filter.

Parameters	Comment	Unit	Min	Max	Default
mCon_nominal	Nominal mass of the contaminant held by the filter	kg	1E-3	n/a	n/a

Variables	Comment	Unit	Type
mCon_flow	Rate of the contaminant mass captured by the filter	kg/s	Input
triRep	Replacing the filter when trigger becomes true	n/a	Input
mCon	Mass of the contaminant held by the filter	kg	Output

Model Implementations:



MassTransfer

A block adds the trace substances into the stream.

Variables	Comment	Unit	Type
mCon_flow[Medium.nC]	Mass flow rate of the contaminant	kg/s	Input
eps	Transfer efficiency	1	Input
port_a	Fluid connector a	n/a	Fluid connector
port_b	Fluid connector b	n/a	Fluid connector

Model Implementations:

```
port_b.C_outflow = inStream(port_a.C_outflow) + eps * mCon_flow;
port_a.C_outflow = inStream(port_b.C_outflow) - eps * mCon_flow;
// Mass balance (no storage).
port_a.Xi_outflow = inStream(port_b.Xi_outflow);
port_b.Xi_outflow = inStream(port_a.Xi_outflow);
port_a.m_flow = -port_b.m_flow;
// Pressure balance (no pressure drop).
port_a. p = port_b.p;
// Energy balance (no heat exchange).
port_a.h_outflow = inStream(port_b.h_outflow);
port_b.h_outflow = inStream(port_a.h_outflow);
```

Generic

A generic model of air filter, which captures the impacts of the mass of the contaminant held by the filters on the filter efficiency and pressure drops. This model can be used with air medium models that have extra substances other than air and water.

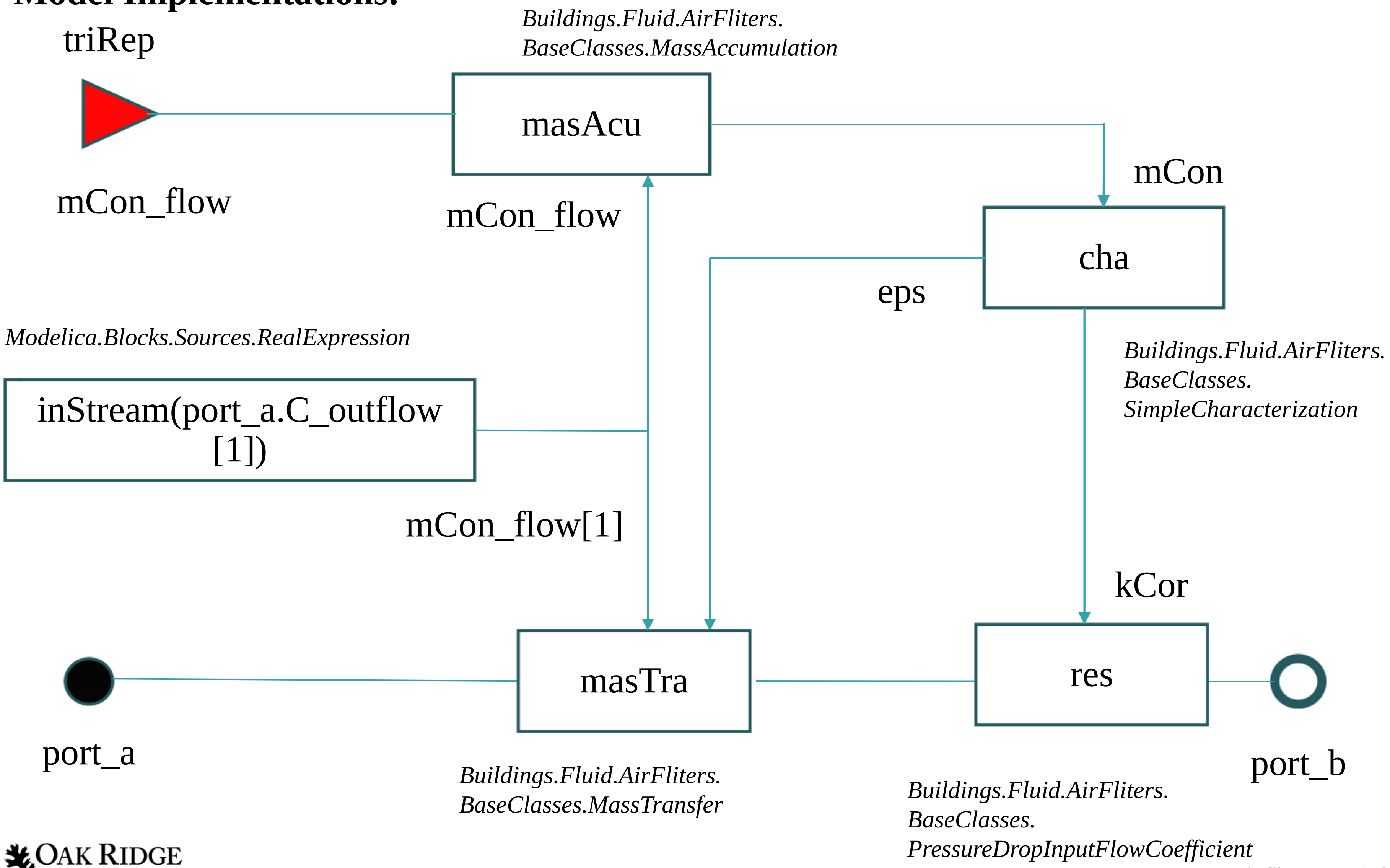
.

Parameters	Comment	Unit	Min	Max	Default
mCon_nominal	Nominal mass of the contaminant held by the filter	kg	1E-3	n/a	n/a
epsFun	Filter efficiency curve	1	n/a	n/a	{1}
b	Resistance coefficient	1	1+1E-3	n/a	1.5
mAir_flow_nominal	Nominal mass air flow rate	kg/s	n/a	n/a	n/a
dp_nominal	Nominal pressure drop	Pa	n/a	n/a	n/a

Variables	Comment	Unit	Type
triRep	Replacing the filter when trigger becomes true	n/a	Input
port_a	Fluid connector a	n/a	Fluid conne ctor
port_b	Fluid connector b	n/a	Fluid conne ctor

Generic (Cont’l)

Model Implementations:



Buildings.Fluid.AirFliters.
BaseClasses.MassAccumulation

triRep

masAcu

mCon

mCon_flow

mCon_flow

cha

eps

Modelica.Blocks.Sources.RealExpression

Buildings.Fluid.AirFliters.
BaseClasses.
SimpleCharacterization

inStream(port_a.C_outflow
[1])

mCon_flow[1]

kCor

masTra

res

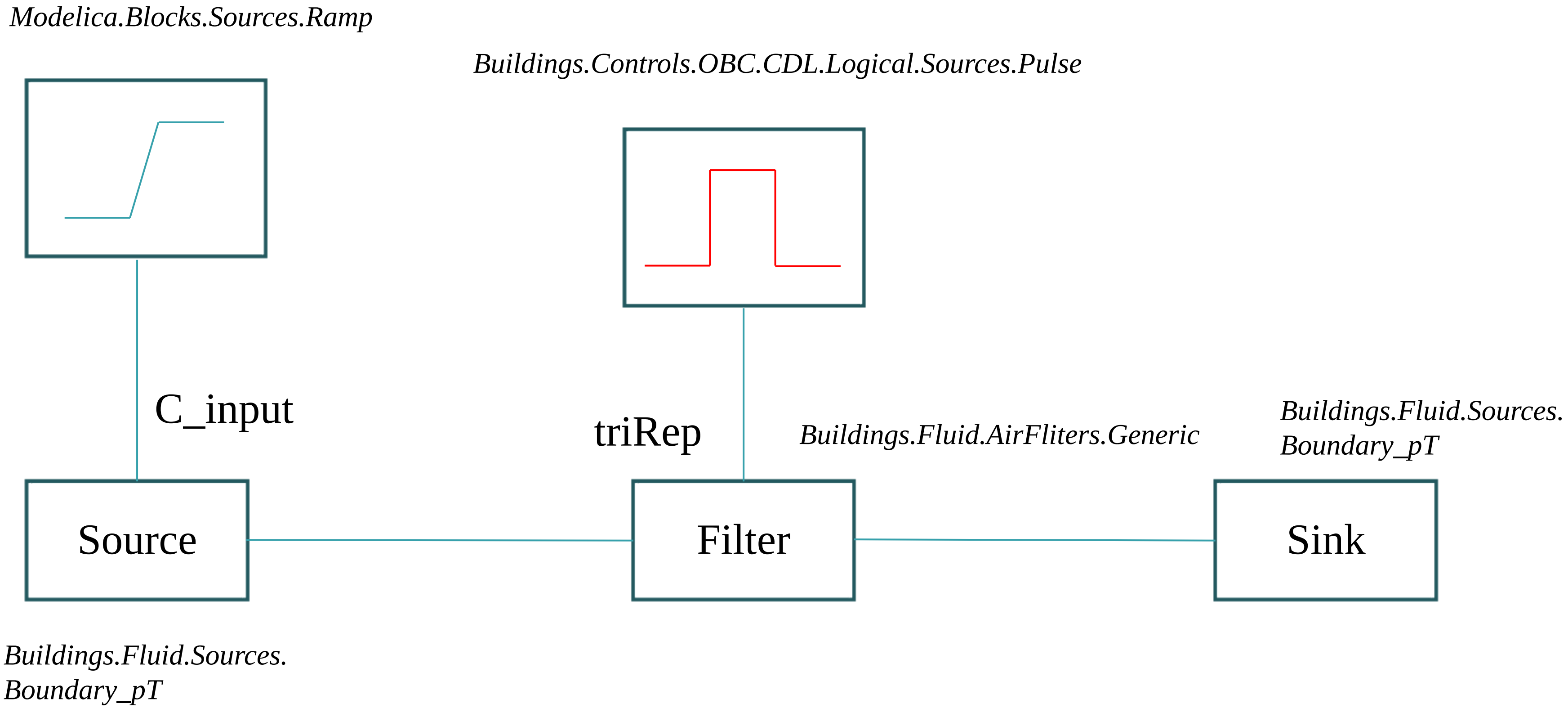
port_a

port_b

Buildings.Fluid.AirFliters.
BaseClasses.MassTransfer

Buildings.Fluid.AirFliters.
BaseClasses.
PressureDropInputFlowCoefficient

Example Design



Modelica.Blocks.Sources.Ramp

Buildings.Controls.OBC.CDL.Logical.Sources.Pulse

C_input

triRep

Buildings.Fluid.AirFliters.Generic

Buildings.Fluid.Sources.
Boundary_pT

Source

Filter

Sink

Buildings.Fluid.Sources.
Boundary_pT