KU LEUVEN

Modelica, Dymola and IDEAS Crash Course 2022

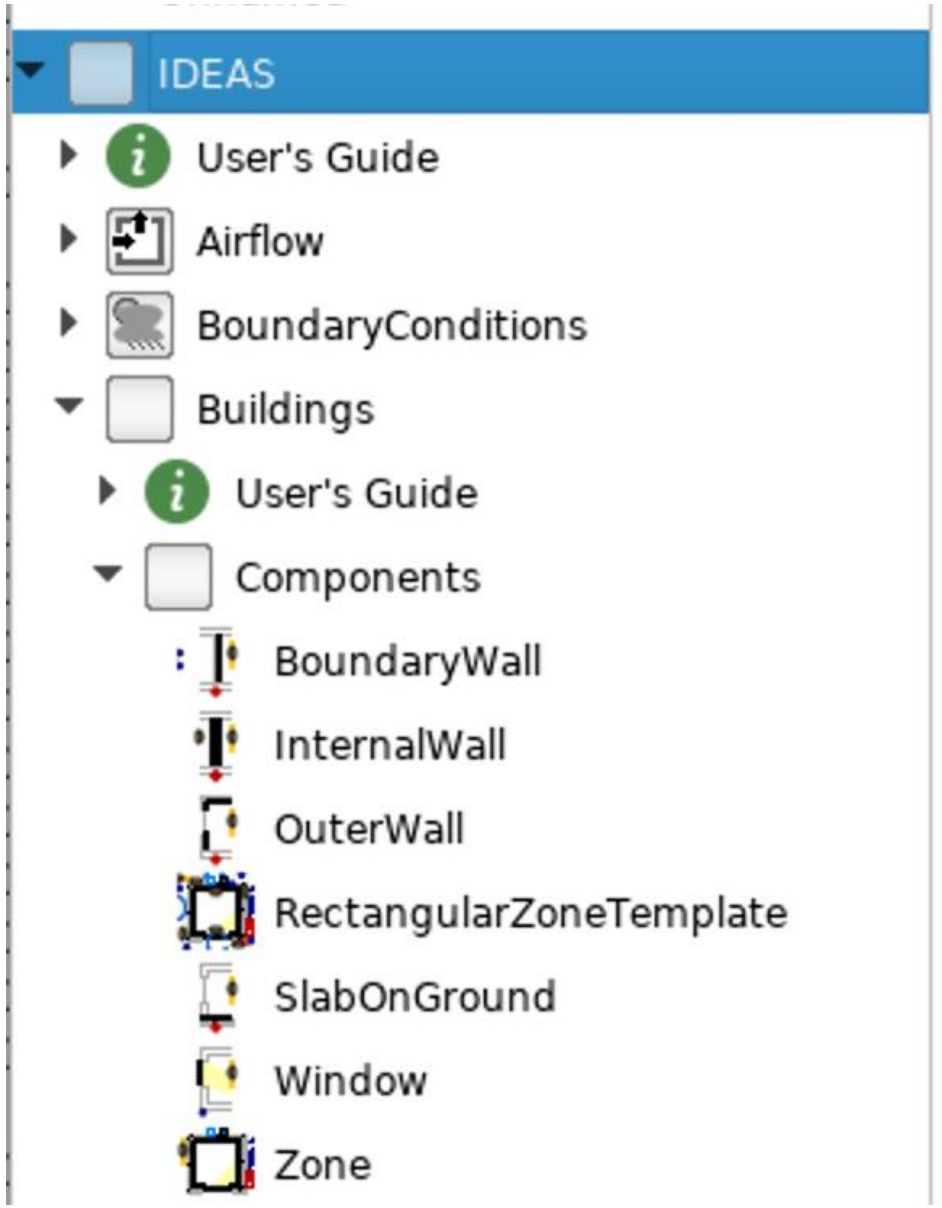
Jelger Jansen
Javier Arroyo
Filip Jorissen
Lucas Verleyen

Part 3+4: IDEAS

Filip Jorissen

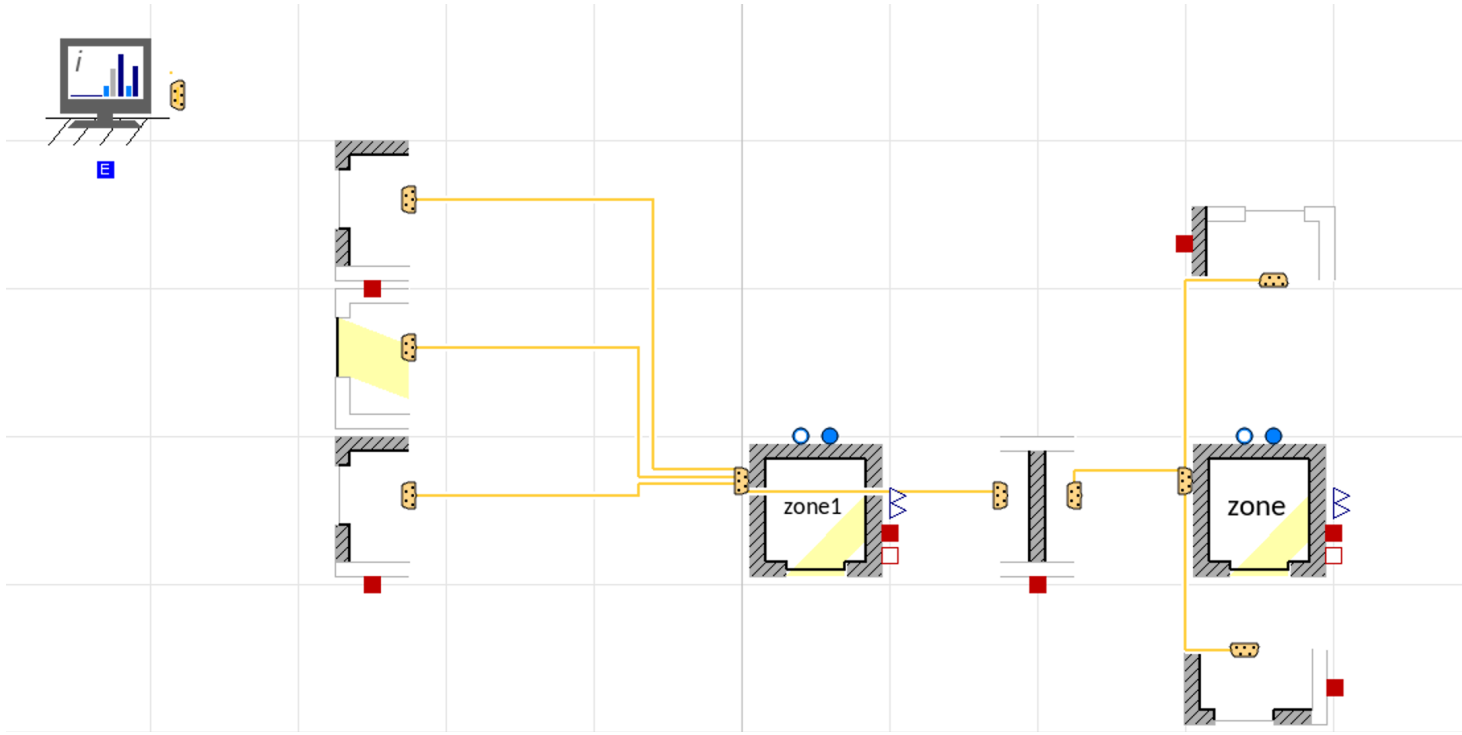
Part 3: IDEAS - principles

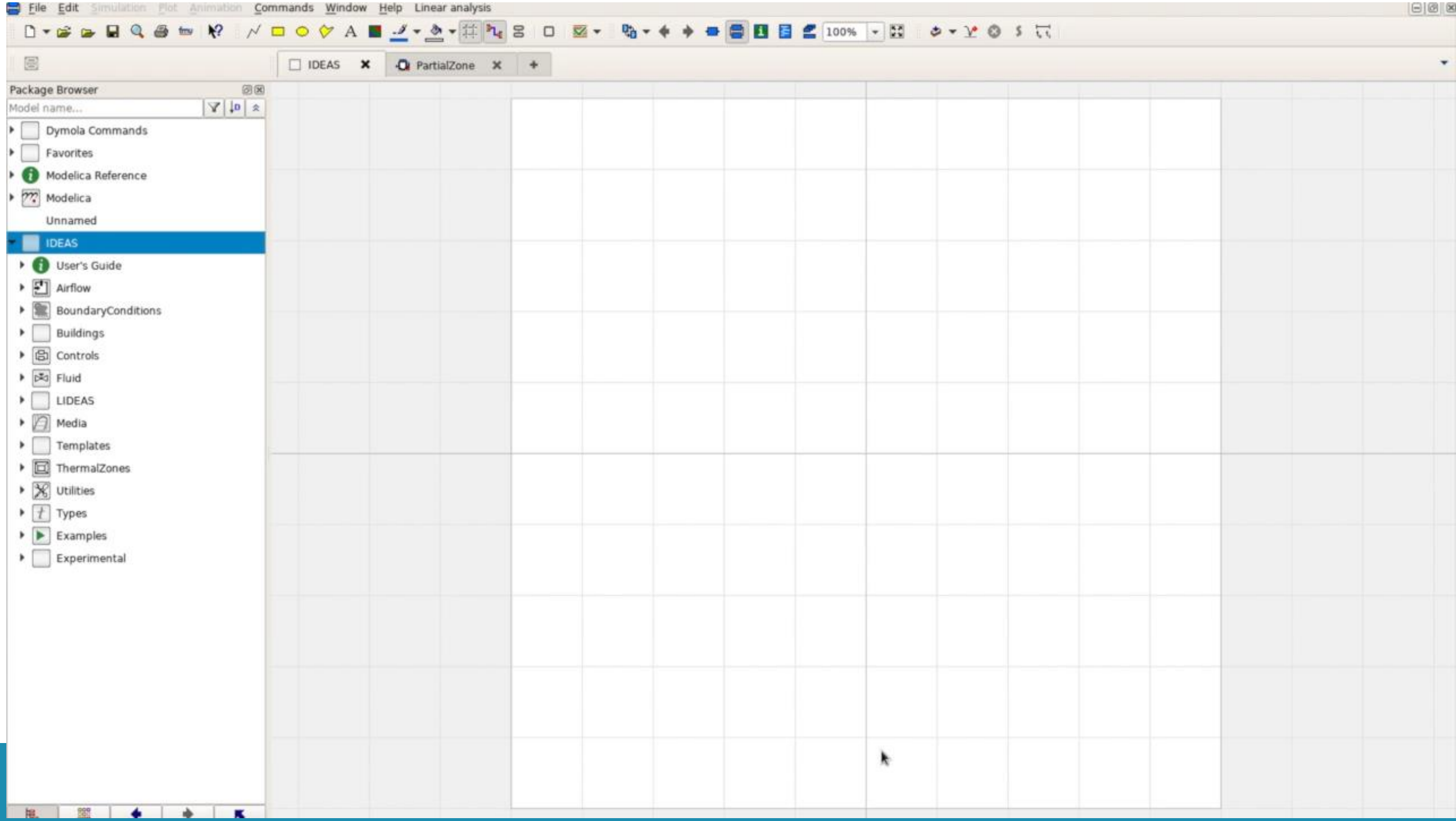
- Overview of components IDEAS.Buildings



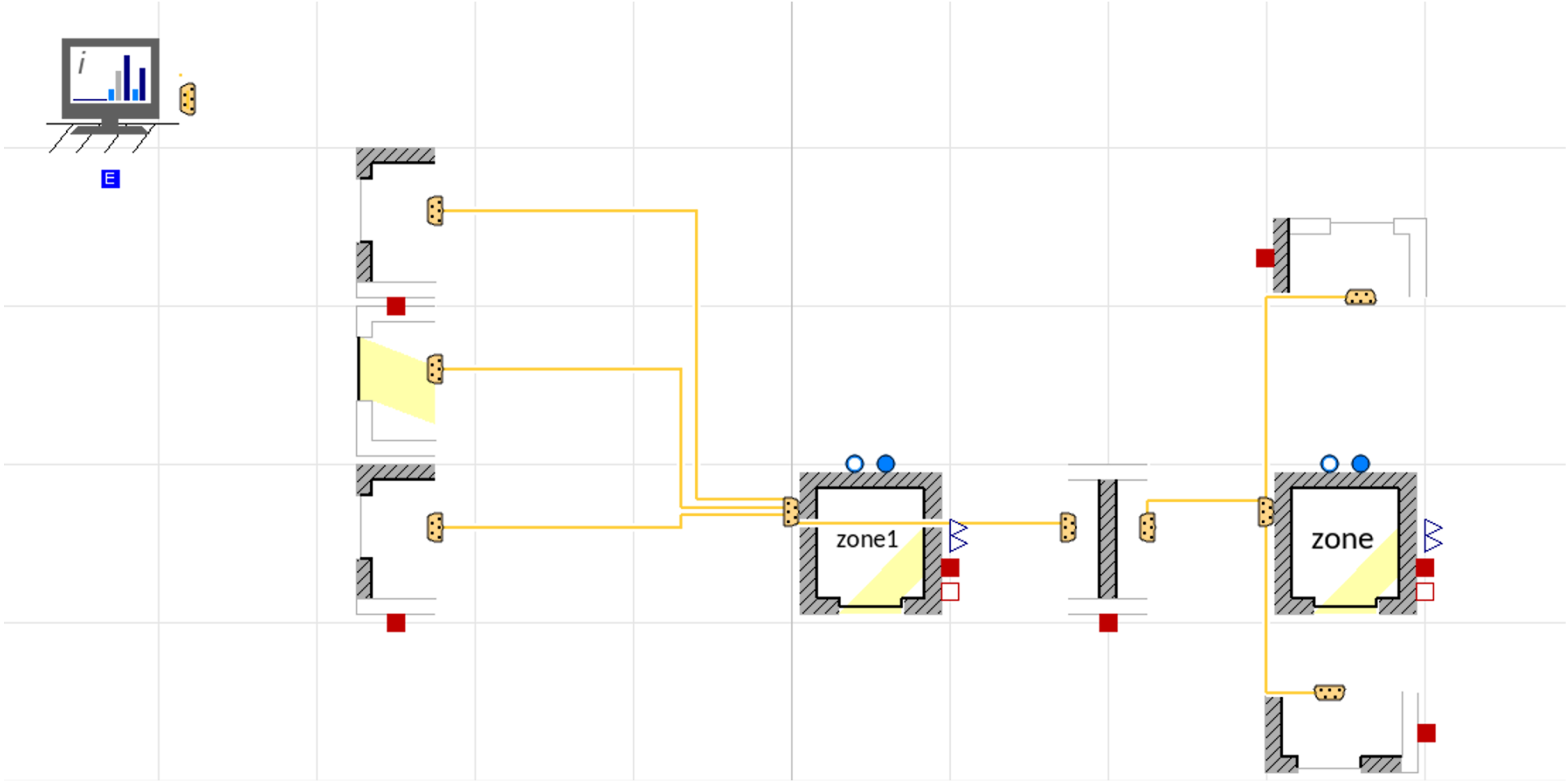
Part 3: IDEAS - principles

- Philosophy
  - Direct mapping between physical objects and components
  - Exception: SimInfoManager
  - Wysiwyg – there is no magic



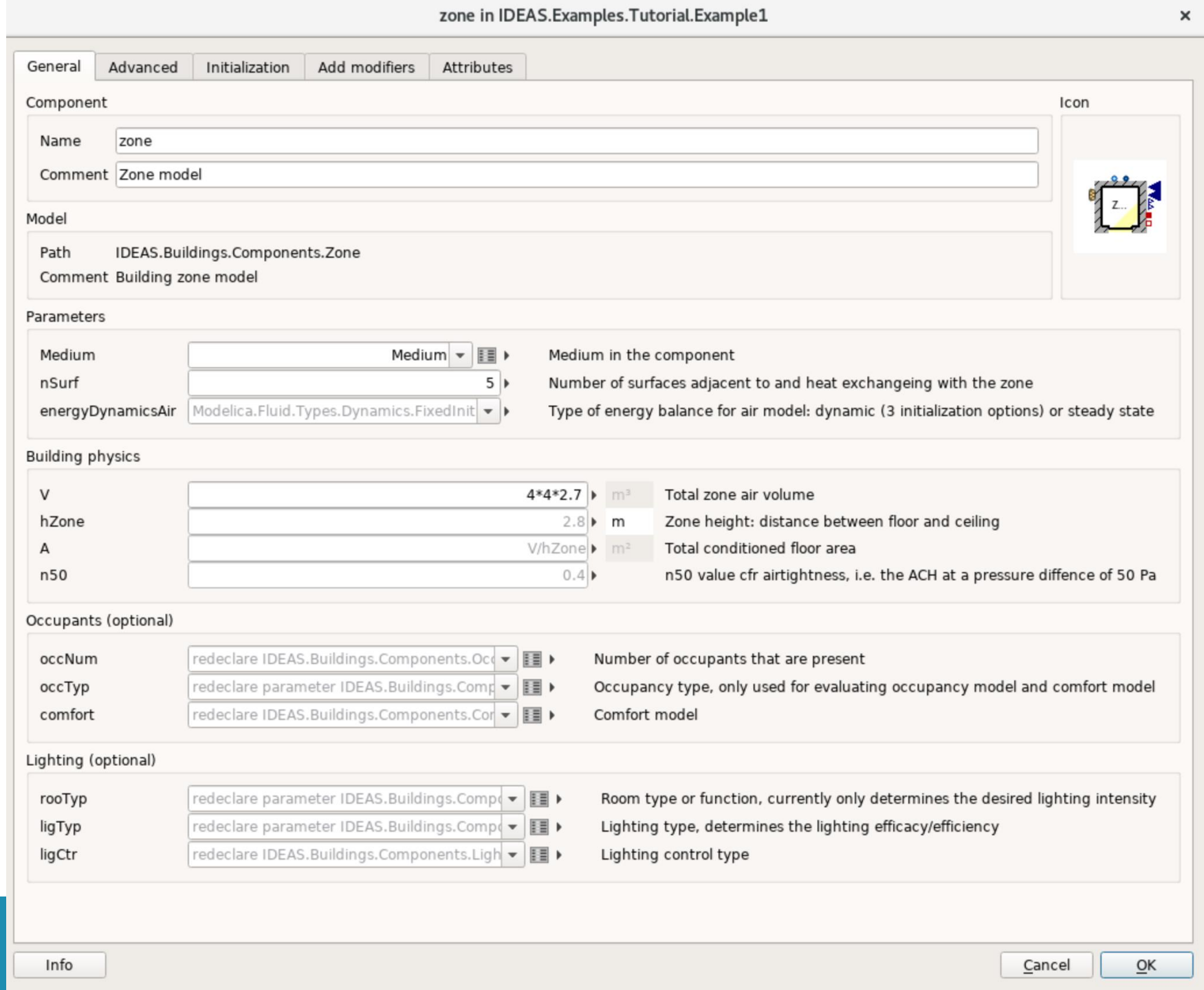

File Edit Simulation Plot Animation Commands Window Help Linear analysis
100%
IDEAS
PartialZone
Package Browser
Model name...
Dymola Commands
Favorites
Modelica Reference
Modelica
Unnamed
IDEAS
User's Guide
Airflow
BoundaryConditions
Buildings
Controls
Fluid
LIDEAS
Media
Templates
ThermalZones
Utilities
Types
Examples
Experimental

Part 3: IDEAS - Multizone



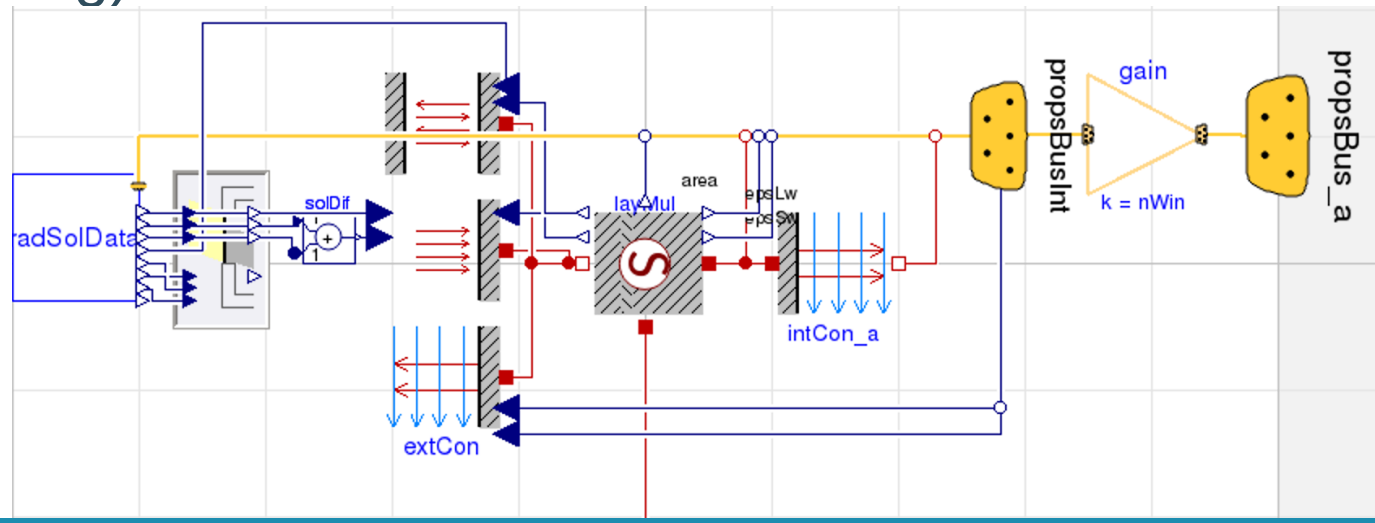
Part 3: IDEAS

- Parameterization



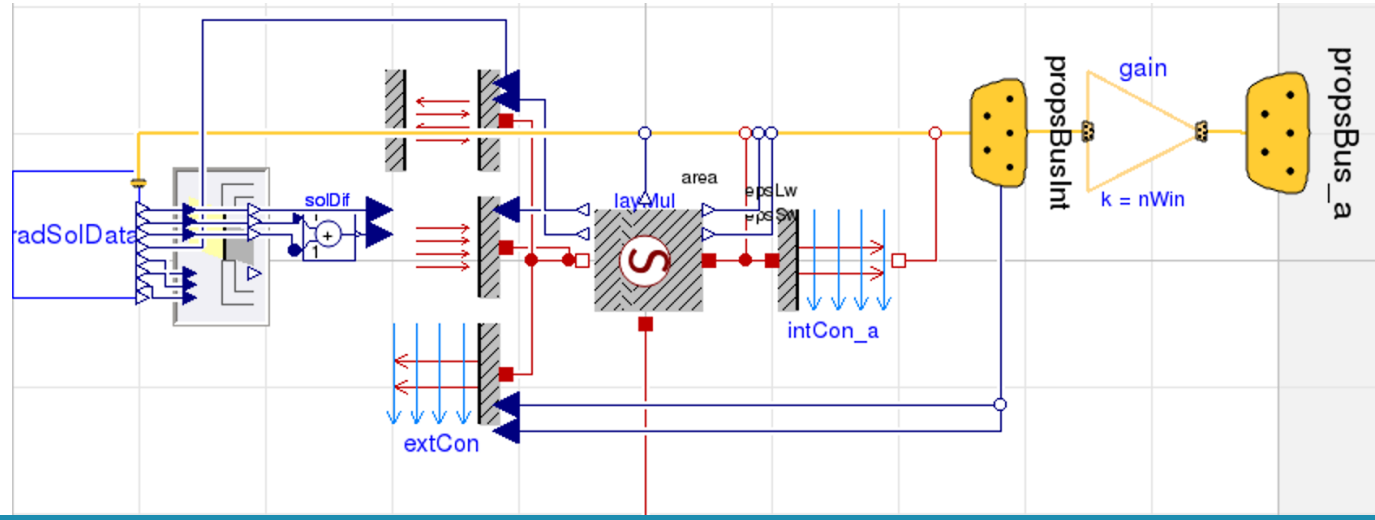
Part 3: IDEAS - physics

- Main building physics:
  - Conduction, thermal mass
  - Convective heat transfer
  - Radiative heat transfer
  - Shortwave heat gains (incl. shading)
  - Internal heat gains:
    occupants, lighting



Part 3: IDEAS - physics

- Demo



Part 3: IDEAS – advanced Modelica concepts

- ‘Extending’ models:
  - Imports all equations from the extended model
  - Allows modifications/extensions on top of that model

Valve 1

```
model TwoWayLinear "Two way valve with linear flow characteristics"
  extends BaseClasses.PartialTwoWayValveKv(phi=l + y_actual*(1 - l));
```

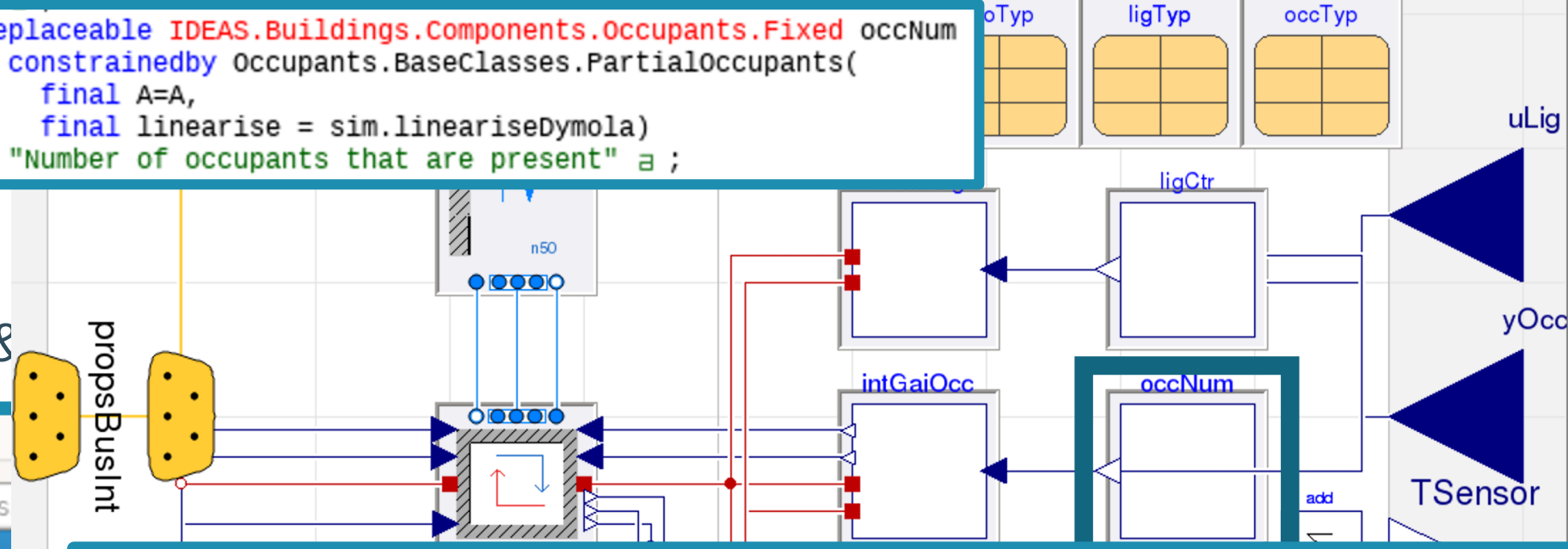
Valve 2

```
model TwoWayPolynomial "Two way valve with polynomial characteristic"
  extends IDEAS.Fluid.Actuators.BaseClasses.PartialTwoWayValveKv(
    phi=l + pol_y*(1 - l));

  parameter Real[:] c
    "Polynomial coefficients, starting with fixed offset";

protected
  constant Integer nP = 100
    "Number of points for initial algorithm test";
  Real pol_y = sum(c.*{y_actual^i for i in 0:size(c,1)-1})
    "Polynomial of valve control signal";
```

Part 3: IDEA

- Redeclare models &

```
replaceable IDEAS.Buildings.Components.Occupants.Fixed occNum
  constrainedby Occupants.BaseClasses.PartialOccupants(
    final A=A,
    final linearise = sim.linariseDymola)
  "Number of occupants that are present" a ;
```



Occupants (optional)

occNum — redeclare IDEAS.Buildings

occTyp

comfort

- Number of occupants
- Occupancy defined by
- Fixed number of occu
- Number of occupants
- Number of occupants

Lighting (optional)

rooTyp — redeclare parameter IDEA

ligTyp — redeclare parameter IDEA

ligCtr — redeclare IDEAS.Buildings

Occupants (optional)

occNum — rede

occTyp — rede

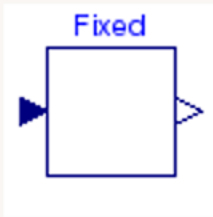
General | Add modifiers | Attributes

Component

Name: zone.occNum

Comment:

Icon



Model

Path: IDEAS.Buildings.Components.Occupants.Fixed

Comment: Fixed number of occupants

Parameters

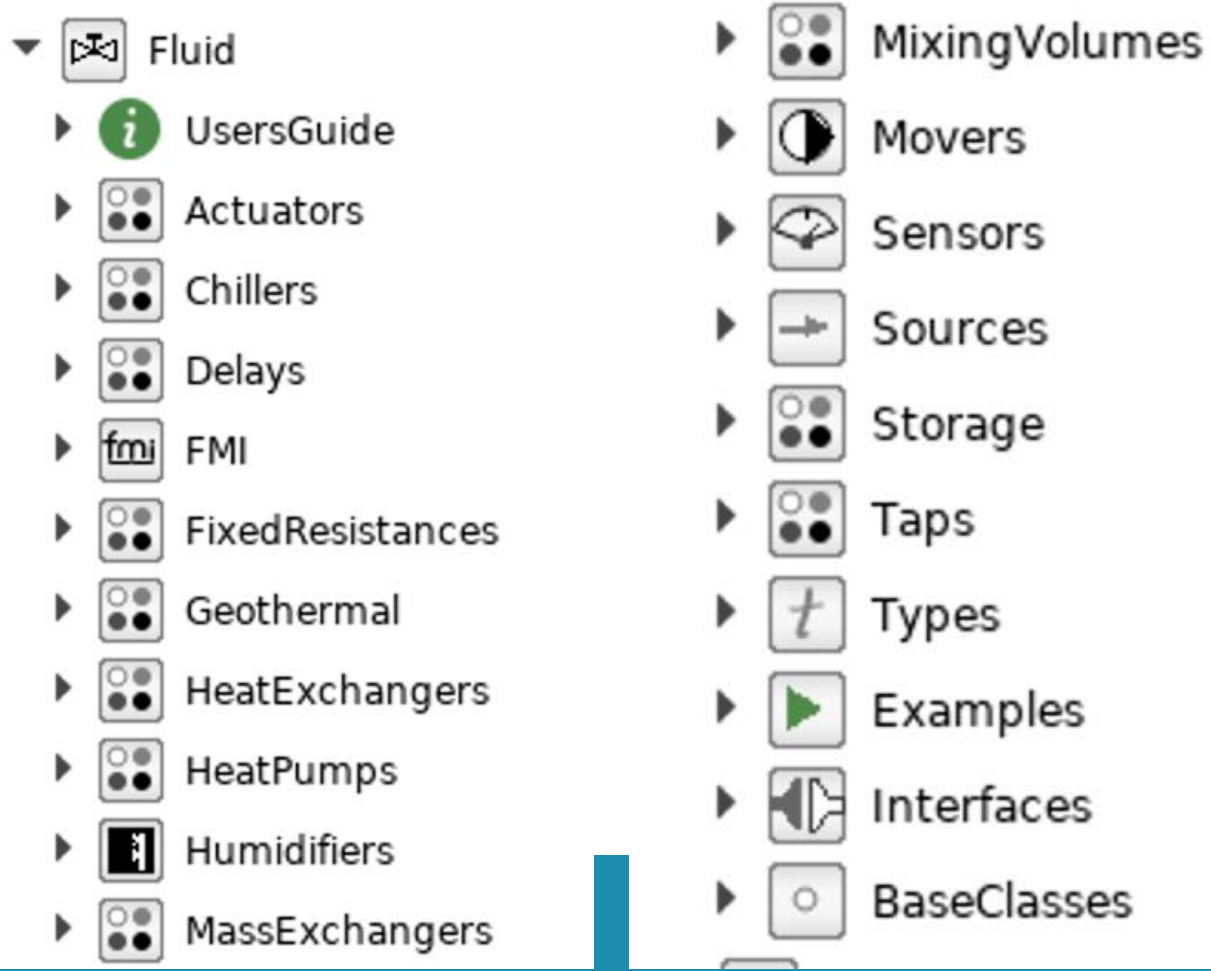
nOccFix: 0 — Fixed number of occupants

Part 3: IDEAS – envelope exercise

- Exercise (simulate at each step – compare result between steps):
  1. Create a single-room house
     1. 8 m by 4 m
     2. Double glazing 3*1.4 window
     3. Brick walls (BESTEST heavy wall), concrete floor (BESTEST heavy floor)
  2. Extend: add closed screens
  3. Extend: add a single occupant and LED lighting
  4. Create and redeclare realistic occupant schedule
  5. Split the zone in two zones using the RectangularZoneTemplate
     1. Zone 1: north facing window
     2. Zone 2: south facing window

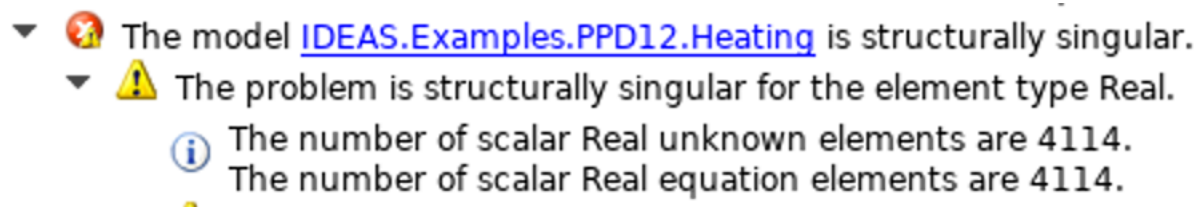
Part 4: IDEAS – HVAC & utilities overview

- IDEAS.Fluid, IDEAS.Media, IDEAS.Utilities



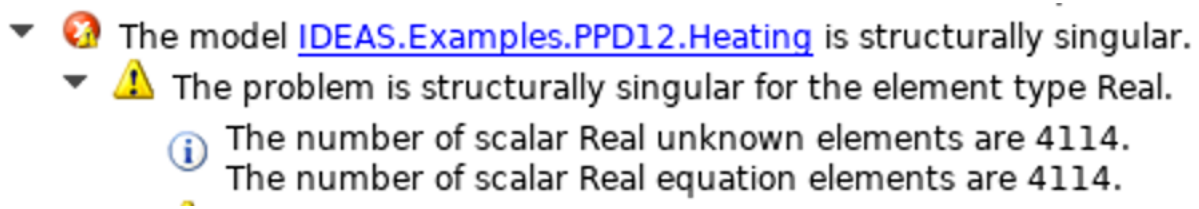
Part 4: some Modelica fundamentals

- Modelica is a generic modelling language
  - I.e. not tailored to buildings
- Important language requirements:
  - # equations = # variables
  - Equations should not contradict each other (e.g. adding both x = 1 and x = 0)
  - Solution to equations should exist (e.g. don't try $x^2 = -1$)
- These requirements are abstract and are hidden from the user by library developers. However, they pop up sometimes, leading to unclear errors like:



Part 4: some Modelica fundamentals

- These requirements are abstract and are hidden from the user by library developers. However, they pop up sometimes, leading to unclear errors like:



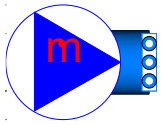
- IDEAS.Buildings is fairly robust as long as each zone propsBus connector is connected to exactly one surface propsBus connector.
- IDEAS.Fluid pressure drop circuits can require some experience:
  - Set absolute pressure in flow circuits
  - Don’t oversimplify pressure drops

Part 4: some Modelica fundamentals

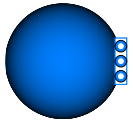
- Modelica.Fluid.Interfaces.FluidPort
  - Potential variable: pressure
  - Flow variable: mass flow rate

```
connector FluidPort
  "Interface for quasi one-dimensional fluid flow in a piping network (incompressible or com

  replaceable package Medium = Modelica.Media.Interfaces.PartialMedium
    "Medium model" ⊒ ;

  flow Medium.MassFlowRate m_flow
    "Mass flow rate from the connection point into the component";
  Medium.AbsolutePressure p "Thermodynamic pressure in the connection point";
  stream Medium.SpecificEnthalpy h_outflow
    "Specific thermodynamic enthalpy close to the connection point if m_flow < 0";
  stream Medium.MassFraction Xi_outflow[Medium.nXi]
    "Independent mixture mass fractions m_i/m close to the connection point if m_flow < 0";
  stream Medium.ExtraProperty C_outflow[Medium.nC]
    "Properties c_i/m close to the connection point if m_flow < 0";
end FluidPort;
```

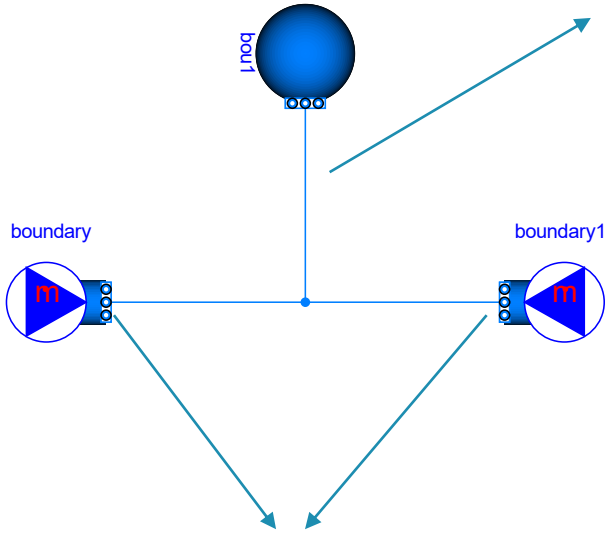
Part 4: some Modelica fundamentals



Sets m_flow = some input



Sets dp = some input

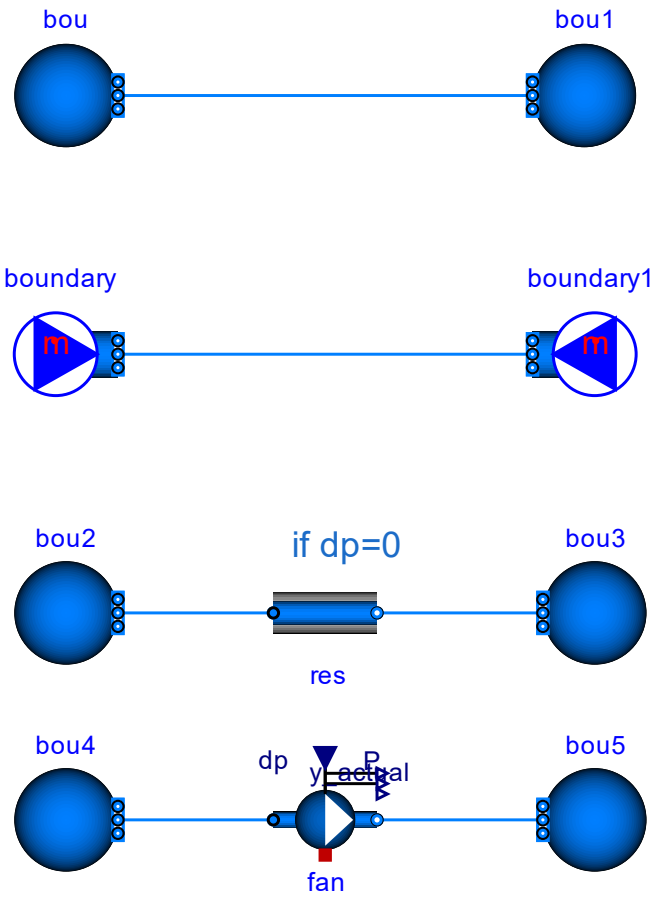


Flow rate equals sum of flow rates set by boundary and boundary1

Pressure equals pressure set by bou1

Part 4: some Modelica fundamentals

- Some illegal circuits:



Further reading

- F. Jorissen, G. Reynders, R. Baetens, D. Picard, D. Saelens, and L. Helsen. Implementation and Verification of the IDEAS Building Energy Simulation Library. *Journal of Building Performance Simulation*, **11** (6), 669-688, 2018. doi: 10.1080/19401493.2018.1428361.
- F. Jorissen, M. Wetter, and L. Helsen. Simulation Speed Analysis and Improvements of Modelica Models for Building Energy Simulation. In 11th International Modelica Conference, pages 59–69, Paris, 2015. doi: 10.3384/ecp1511859.
- F. Jorissen, M. Wetter, and L. Helsen. Simplifications for Hydronic System Models in Modelica. *Journal of Building Performance Simulation*, **11** (6), 639-654, 2019.
- F. Jorissen. *Toolchain for Optimal Control and Design of Energy Systems in Buildings*. PhD thesis, Arenberg Doctoral School, KU Leuven, April 2018

Part 4: IDEAS – HVAC exercise

- Using building envelope of step 5 (see IDEAS.Examples.Tutorial):
  1. Add geothermal heat pump heating system
  2. Add a heat pump controller
  3. Compute energy use and export it in a json file
  4. Add a CO_2-controlled ventilation system