Overview of IEA EBC Annex 60
-
A brief introduction to Modelica

Michael Wetter, LBNL, Berkeley, CA

October 17, 2016

Lawrence Berkeley National Laboratory

Strategic initiative:
LBNL leads 42 institutes, 16 countries collaboration

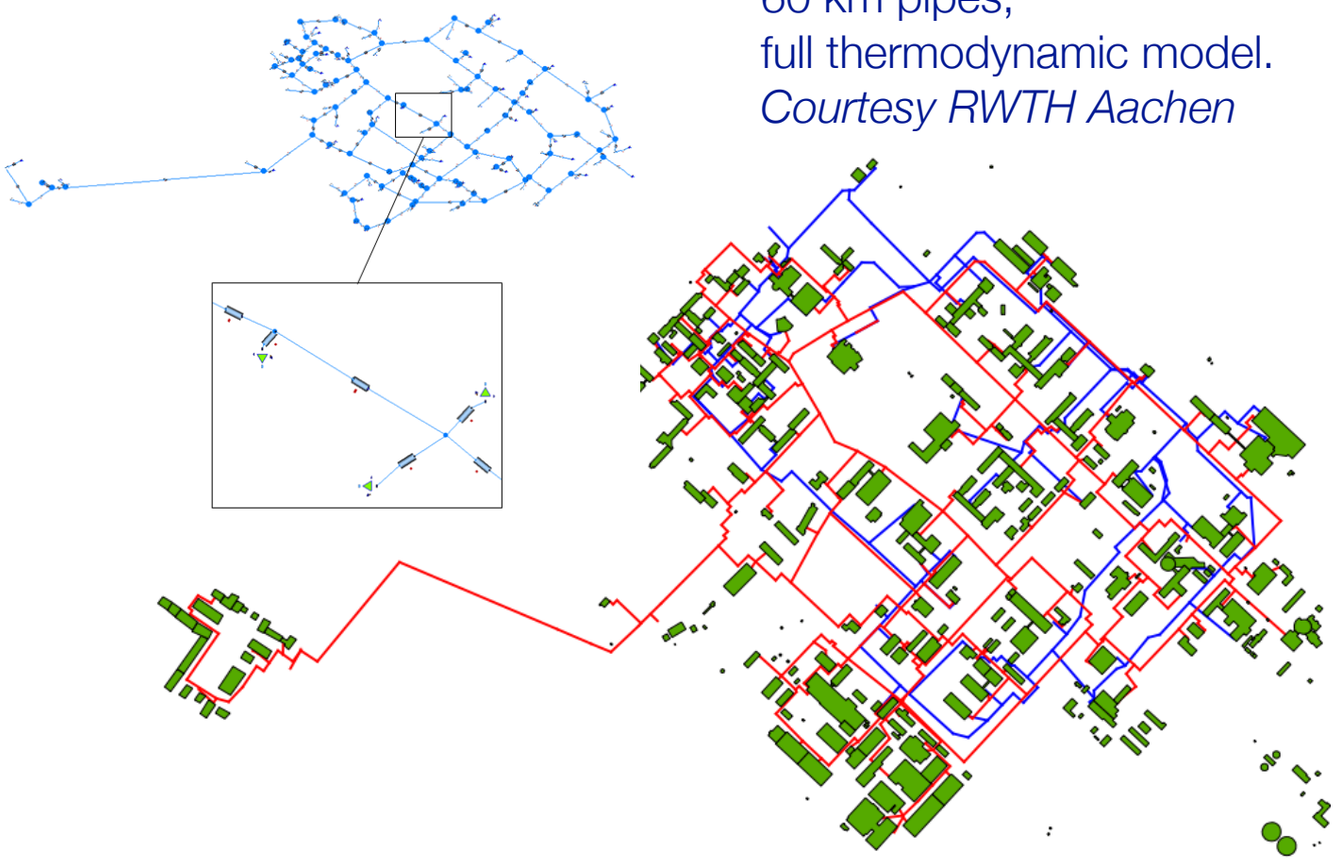
EBC
Energy in Buildings and Communities Programme

IEA EBC Annex 60

New generation computational tools for buildings and community energy systems

Duration: 2012-2017

Operating agents:
Michael Wetter (LBNL) and
Christoph van Treeck (RWTH Aachen).



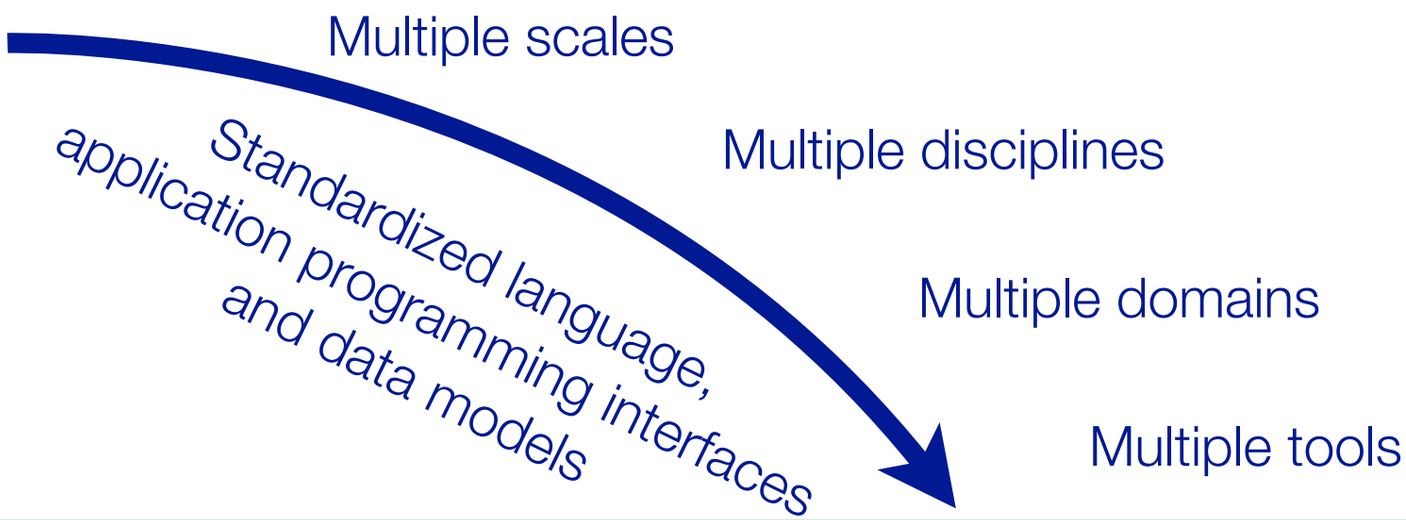
District heating for 200 buildings,
60 km pipes,
full thermodynamic model.
Courtesy RWTH Aachen

Participation:
38 institutes from 16 countries:
Austria, Belgium, Brazil, China, Denmark, France, Germany, Ireland, Italy, the Netherlands,
Slovakia, Spain, Sweden, Switzerland, United Arab Emirates and the USA.

BERKELEY LAB Lawrence Berkeley National Laboratory

RWTH AACHEN UNIVERSITY

Annex 60 overview



Objective

Building and community energy grids

designed & operated as integrated, robust, performance-based systems

Subtask 2

Applications on building design, district design, model-use during operation

Subtask 1

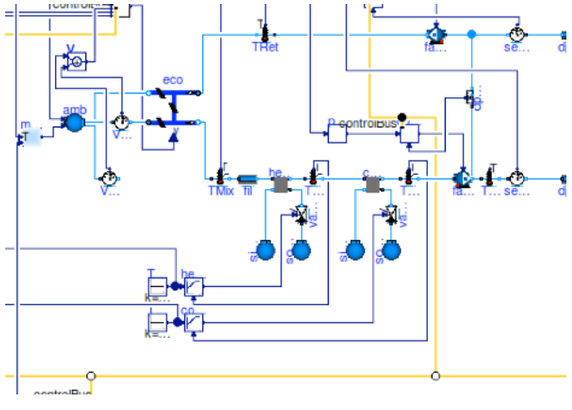
Technology development

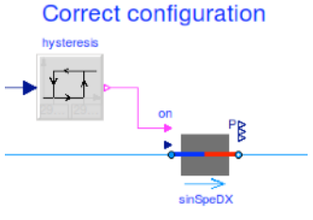
Energy and control systems modeling libraries	**Co-simulation & model-exchange tools and interfaces**	**BIM translators**	**Automation**
Modelica. Free and open-source. Standardized interfaces. Buildings, districts, controls.	Functional Mockup Interface standard. FMI interfaces in existing simulators. Co-simulation algorithms.	Standardized model data exchange. Modelica/BIM interfaces.	Python modules for work-flow automation.

Subtask 1: Technology development

Energy in Buildings and Communities Programme

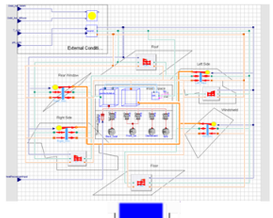
1.1 Modelica model libraries

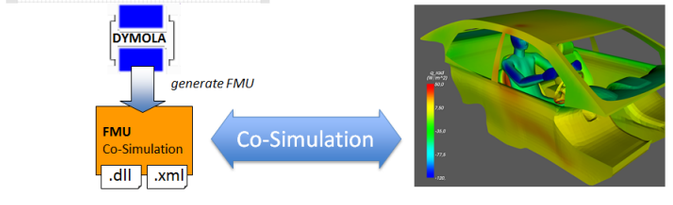


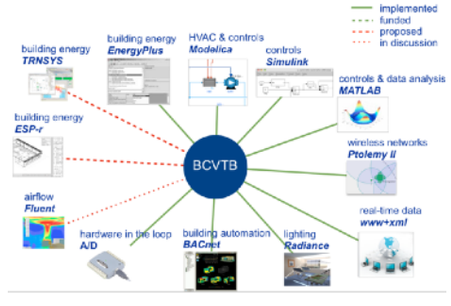


Validated Modelica models & user guide.
Harmonize fragmented duplicative efforts.
Designers, manufacturers, students.

1.2 Co-simulation and model exchange

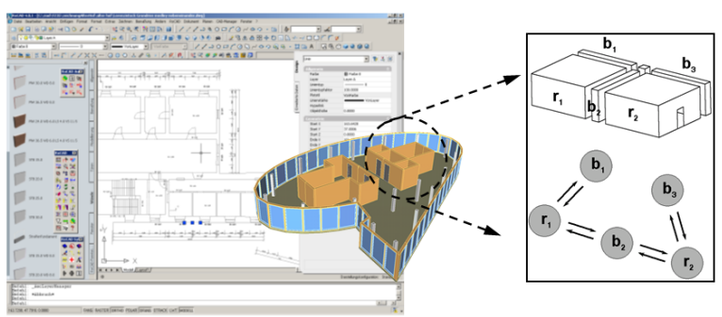






Co-simulation environments.
Integrate FMI interfaces & develop algorithms.
Designers, manufacturers, students.

1.3 Building Information Models



BIM to Modelica translators.
Extend BIM for GIS and controls.
Designers.

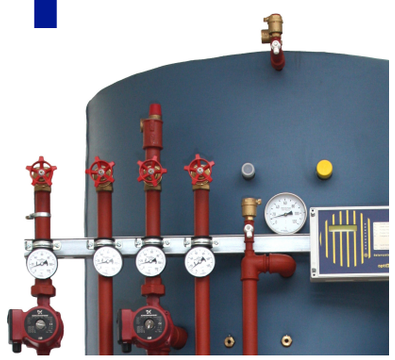
1.4 Workflow automation tools

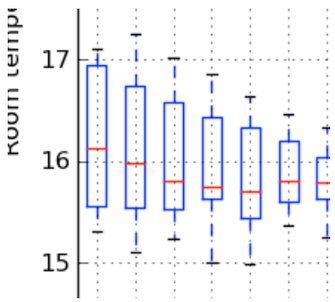
Scripts for pre-processing, running simulations, calling optimizers and post-processing.

Subtask 2: Validation and demonstration

EBC
Energy in Buildings and Communities Programme

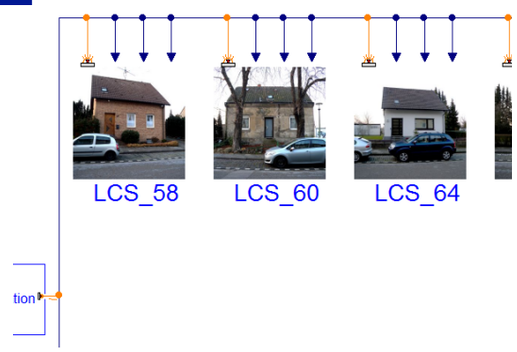
2.1 Detailed design of buildings





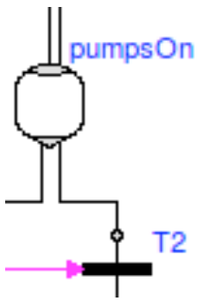
Case study report.
Co-design HVAC, storage & control.
Designers.

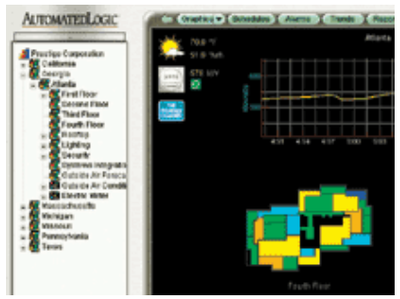
2.2 District energy system design



Case study report.
Utility grid integration (thermal & electrical).
Urban planner, utilities.

2.3 Model use during operation





Software & case study report.
Model-based control & FDD.
Designers, control providers.

Buildings industry is fragmented, builds assets that are sold world-wide, and need to be supported >30 years

Commit yourself to standards, not to tools.

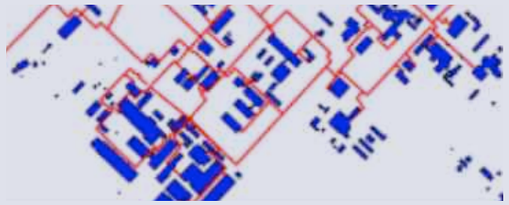
communication

ASHRAE BACnet™

RESTful
• GET
• POST
• PUT
• DELETE

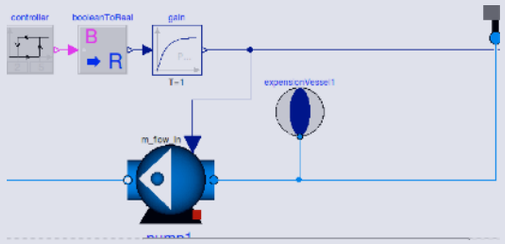
static data

CityGML



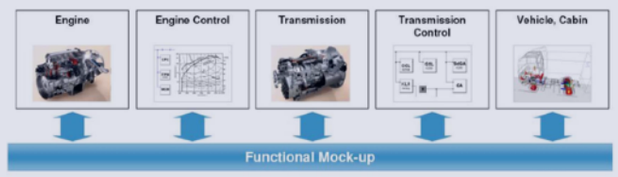
mathematics (behavioral models)

MODELICA



computations (simulators)

fmi FUNCTIONAL MOCK-UP INTERFACE



Cosimulation of the behavioral models and the embedded controller software

> 75M Euro from 2007-15, and growing

Supported by 90 tools

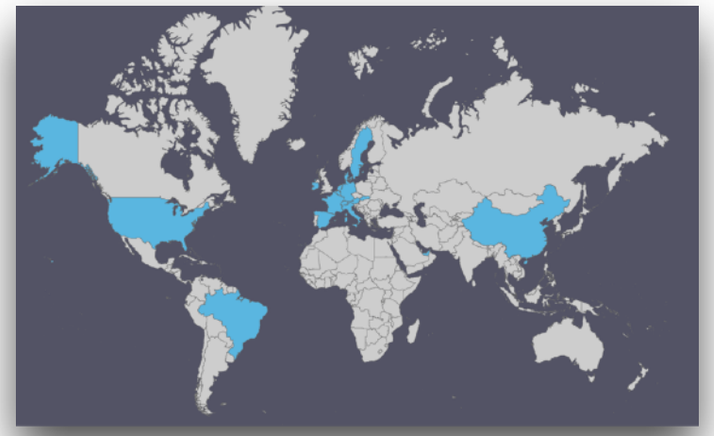
International collaboration on new generation computing tools for buildings and communities, based on open interoperability standards:

IEA EBC Annex 60 2012-17: 42 partners, 16 countries
IBPSA Project 1 2017-22: 15 partners as of now, > 60 person years

EBC
Energy in Buildings and Communities Programme

INTERNATIONAL BUILDING PERFORMANCE SIMULATION ASSOCIATION

Next steps



EBC IEA EBC Annex 60
Energy in Buildings and Communities Programme

IBPSA Project 1

5 year research phase

2016/17: Planning phase

2016/17: Reporting phase

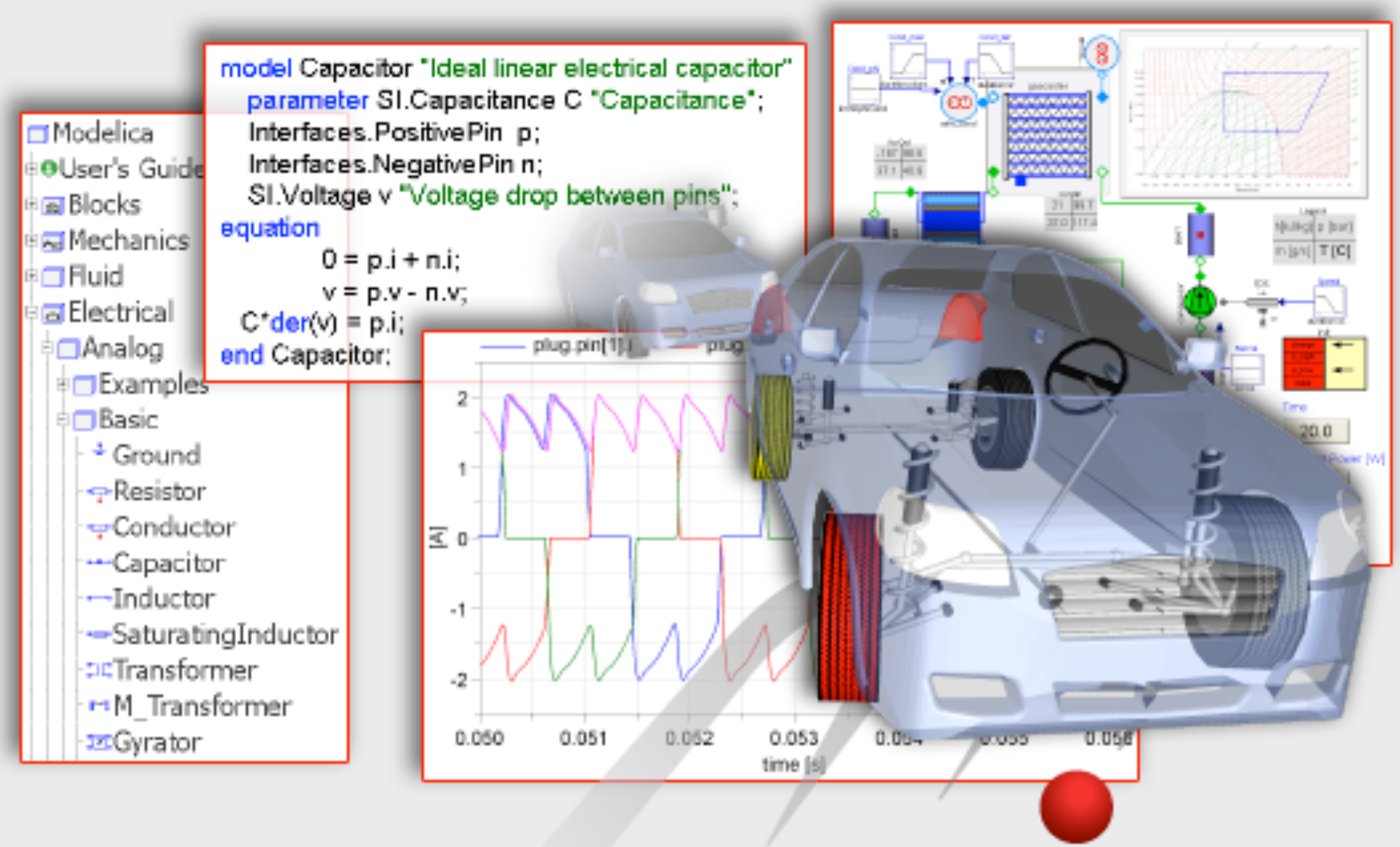
3 year research phase

2012/13: Planning phase

2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022

08/17: Kick-off meeting at Berkeley, CA

https://ibpsa.github.io/project1/


model Capacitor "Ideal linear electrical capacitor"
parameter SI.Capacitance C "Capacitance";
Interfaces.PositivePin p;
Interfaces.NegativePin n;
SI.Voltage v "Voltage drop between pins";
equation
0 = p.i + n.i;
v = p.v - n.v;
C*der(v) = p.i;
end Capacitor;
Modelica
User's Guide
Blocks
Mechanics
Fluid
Electrical
Analog
Examples
Basic
Ground
Resistor
Conductor
Capacitor
Inductor
SaturatingInductor
Transformer
M_Transformer
Gyrator
plug.pin[1].i
[A]
time [s]
MODELICA

Purpose and approach

The purpose is to

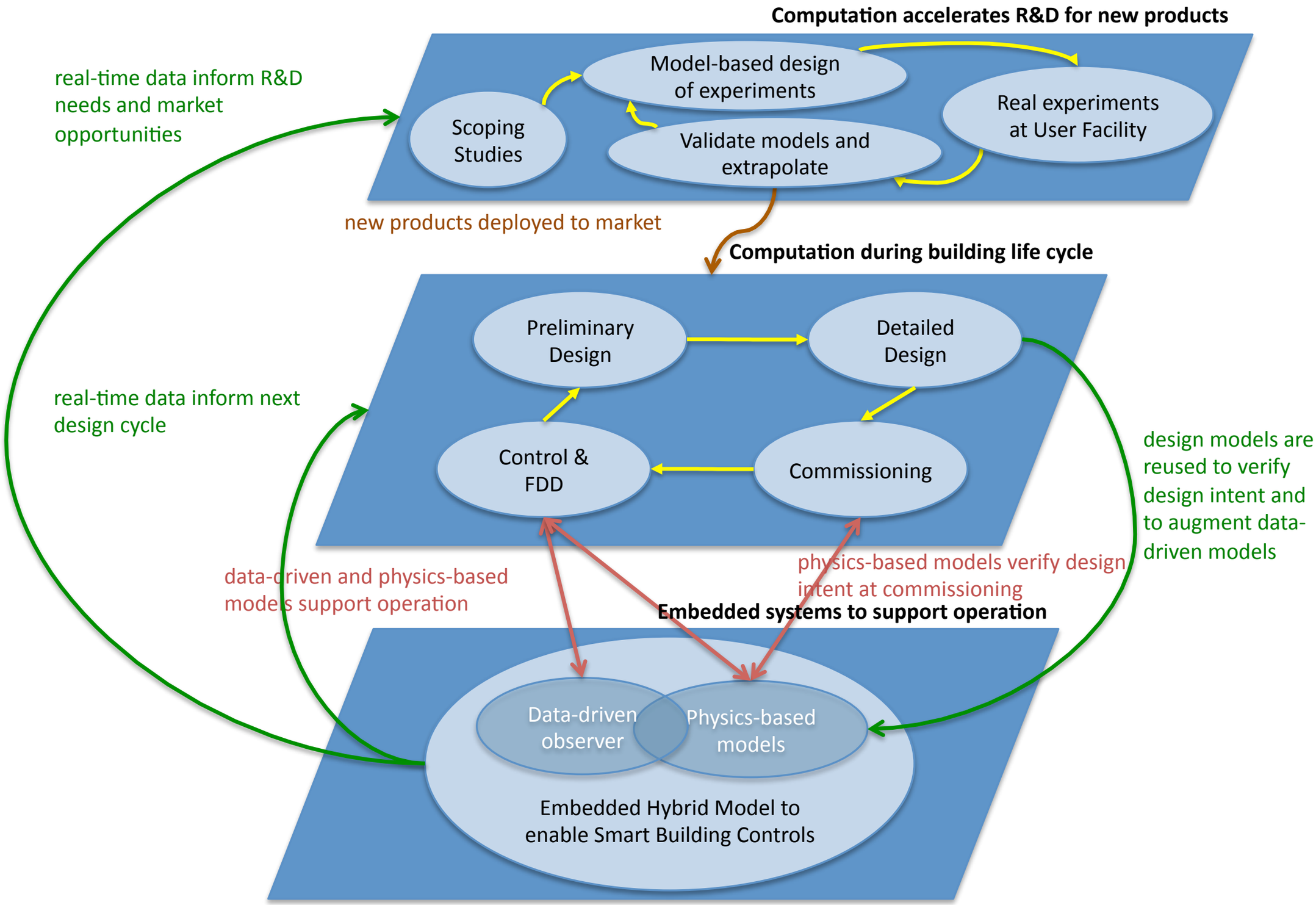
- introduce Modelica
- explain how it differs from other building simulators
- present tool ecosystem

(Applications will be shown afterwards.)

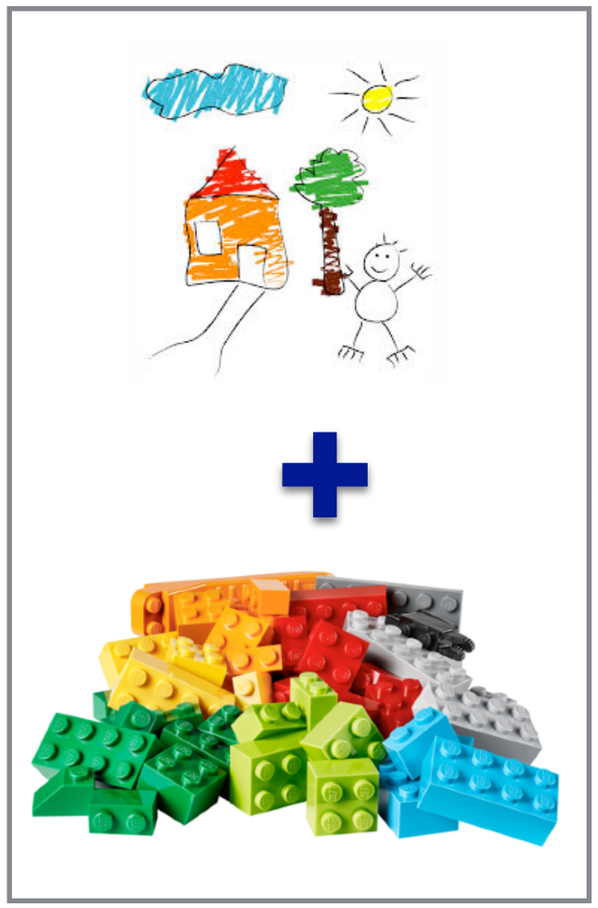
Background material includes:

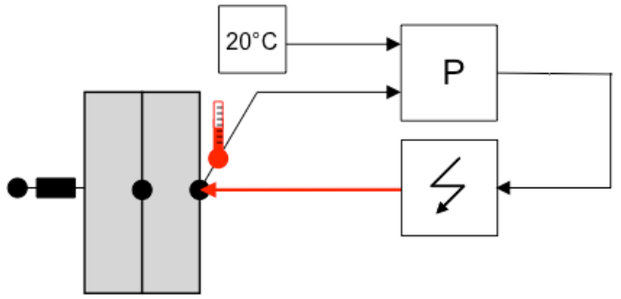
- The online book from Michael Tiller:
  http://book.xogeny.com
- Modelica reference: http://modref.xogeny.com/
- Interactive tour: http://tour.xogeny.com
- Other references:
  http://simulationresearch.lbl.gov/modelica/userGuide/gettingStarted.html

Simulation has applicability beyond building design

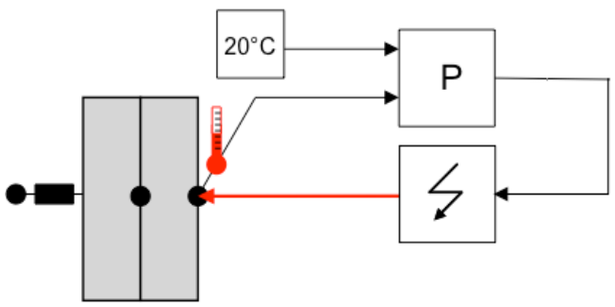


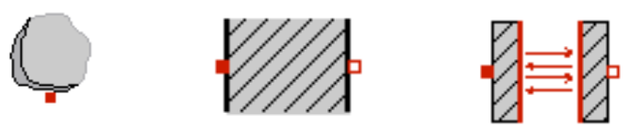
Modeling vs. simulation

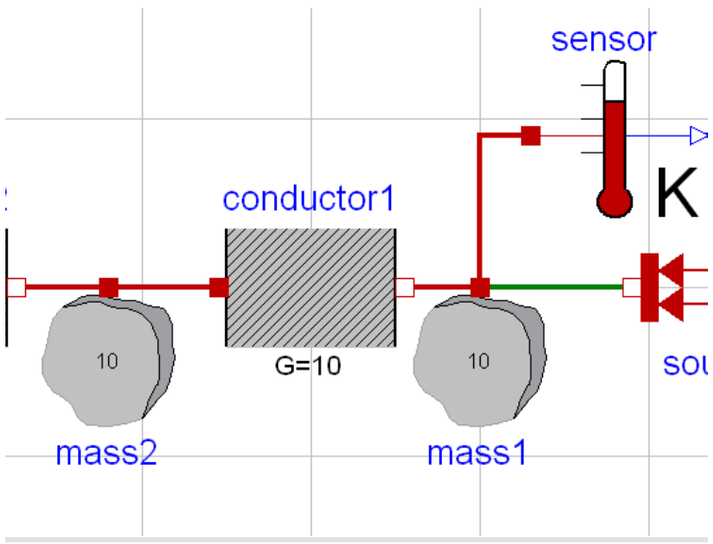


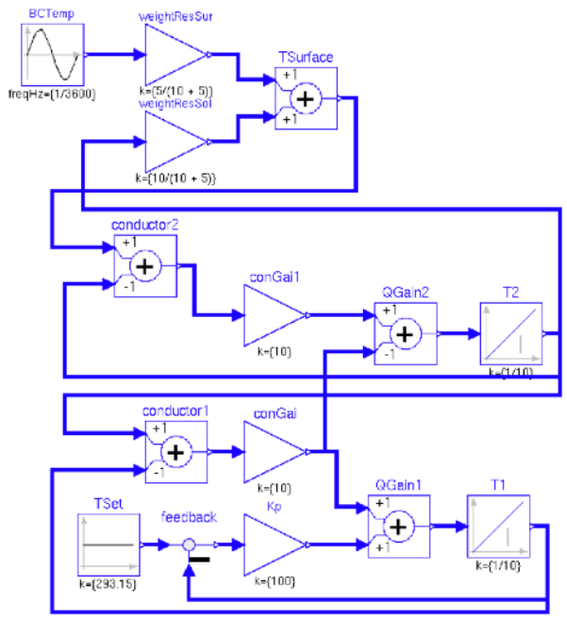



```
integer :: i                                    ! Loop counter
integer, parameter :: lun = 6                   ! logical unit number

double precision, parameter :: amp = 5          ! Amplitude
double precision :: time                        ! Simulation time

character(LEN=*), parameter  :: FMT = "(4F14.7)"

! Initialize variables
dt = tFin/N
T1 = T0
T2 = T0
time = 0
iCom = 1
open (lun, FILE='results.txt')
! Perform integration
do i = 1, N, 1
   TBC = T0 + amp * dsin(2*3.14159*time/3600)
   TSur = T2 * ( G / (G+h) ) + TBC * ( h / (G+h) )
   QSou = Kp * (TSet - T1)
   QCon1 = G * (T2 - T1)
```

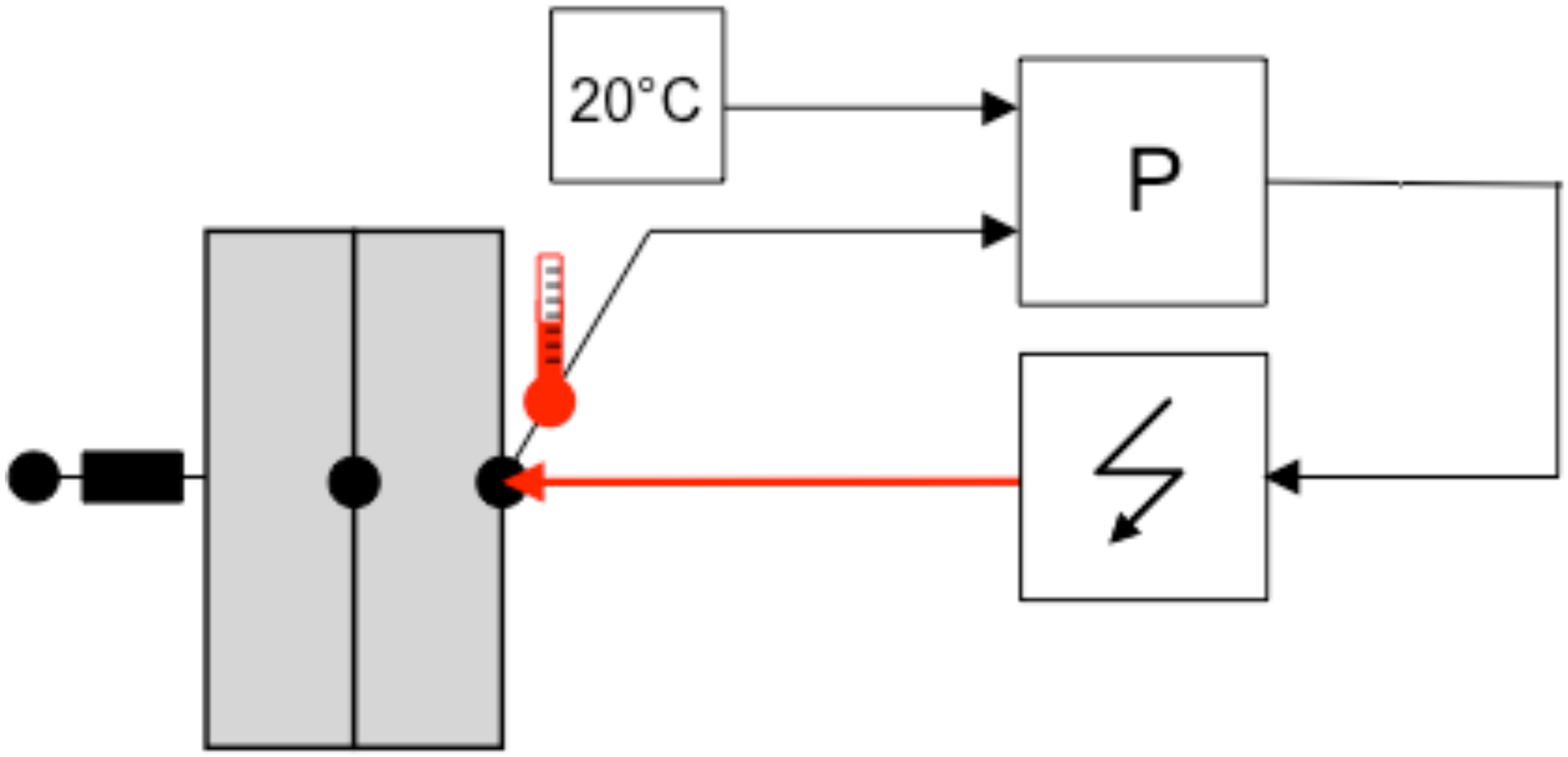








Small demo



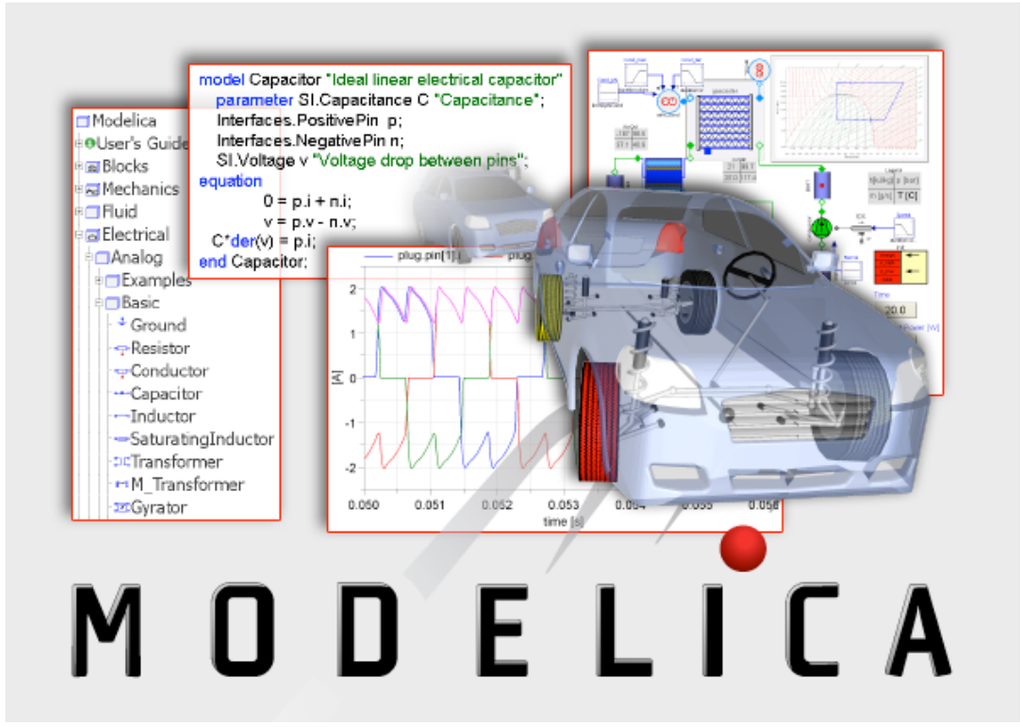
What is Modelica?

Modelica, an open standard that specifies an equation-based, object-oriented modeling language

A modeling, not a programming, language.

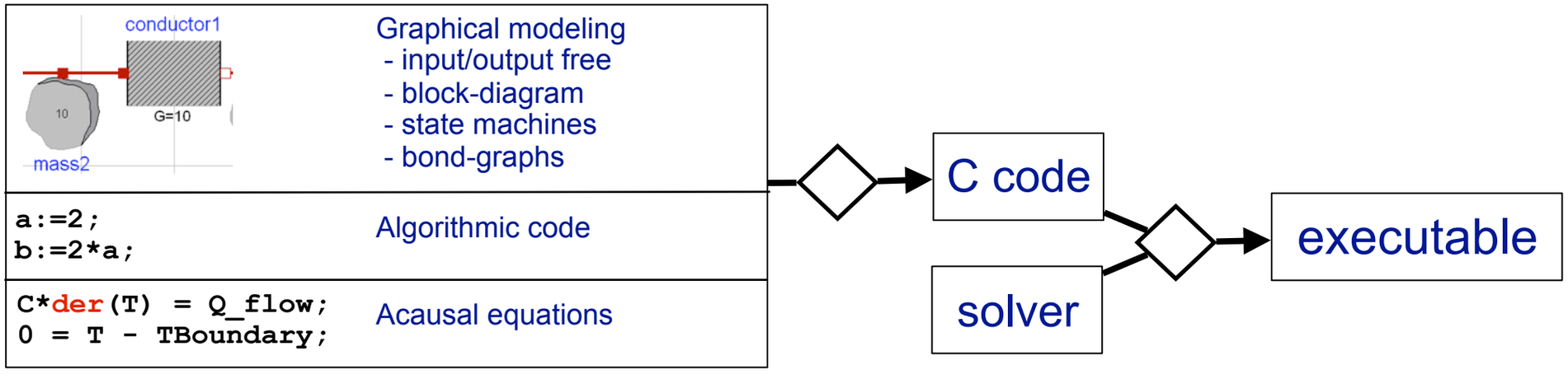
Open, industry-driven standard for modeling multi-physics, engineered systems

Developed since 1996 because conventional approach for modeling was inadequate for integrated engineered systems



Large number of free and commercial libraries, https://modelica.org/libraries

Needs a tool to translate models to an executable.



Separation of equation, solvers, and data input/output yields readable and maintainable models with high reusability

```
model HeatCapacitor "Lumped thermal element storing heat"
```

declare

```
  parameter Modelica.SIunits.HeatCapacity C
     "Heat capacity of element (= cp*m)";
  Interfaces.HeatPort_a port
     "Port with (Q_flow, T)";
```

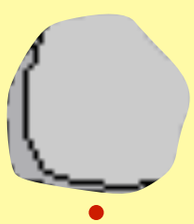
```
equation
```

model

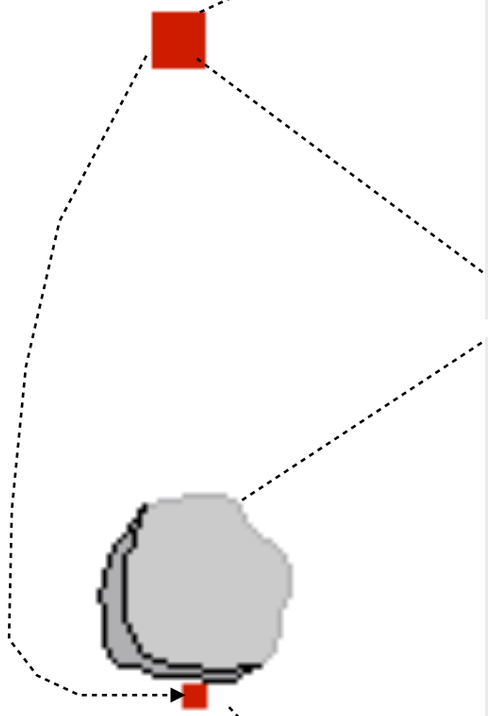
```
  C*der(port.T) = port.Q_flow;
```

```
end HeatCapacitor;
```

iconify

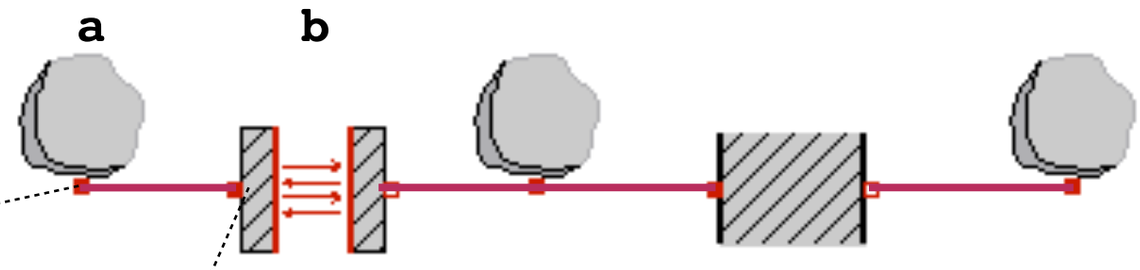


Modularization in object-oriented modeling supports creation of transparent models



```
connector HeatPort_a
  "Thermal port for 1-dim. heat transfer"
  Modelica.SIunits.Temperature T "Port temperature";
  flow Modelica.SIunits.HeatFlowRate Q_flow
    "Heat flow rate (positive if flowing from
     outside into the component)";
end HeatPort_a;
```

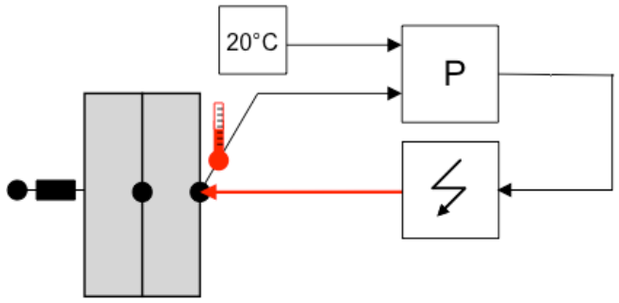
```
model HeatCapacitor "Lumped thermal element storing heat"

  parameter Modelica.SIunits.HeatCapacity C "Heat capacity";
  Modelica.SIunits.Temperature T "Temperature of element";
  Interfaces.HeatPort_a port;

equation
  T = port.T;
  C*der(T) = port.Q_flow;
end HeatCapacitor;
```



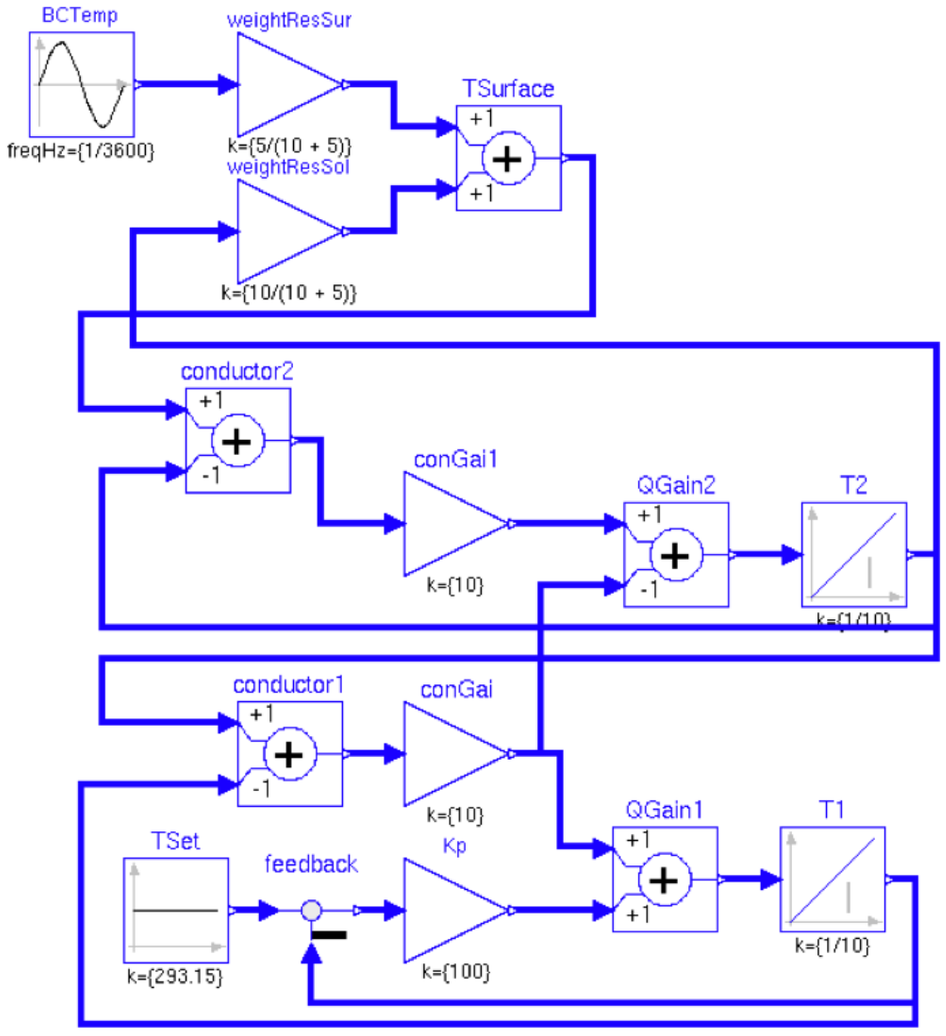

```
connect(a.port, b.port);
```

```
a.port.T = b.port.T;
0 = a.port.Q_flow + b.port.Q_flow;
```

Acausal connectors are used to enable assembling models schematically

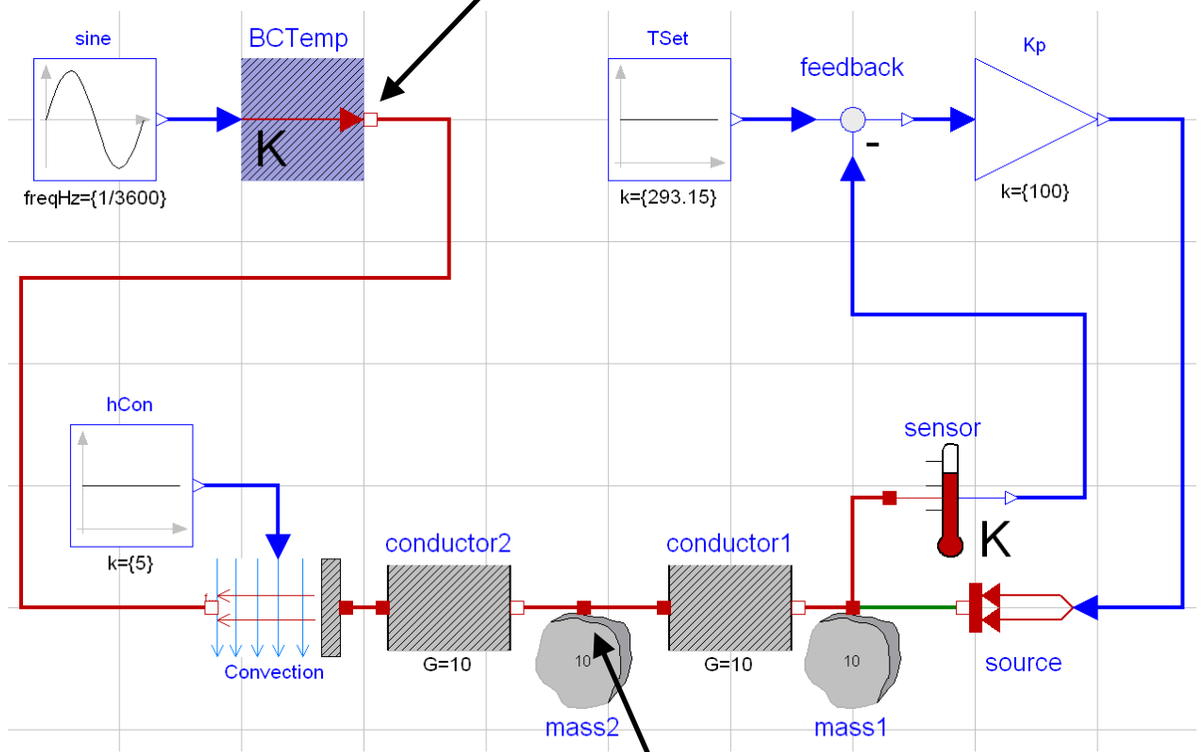


Block Diagram Modeling



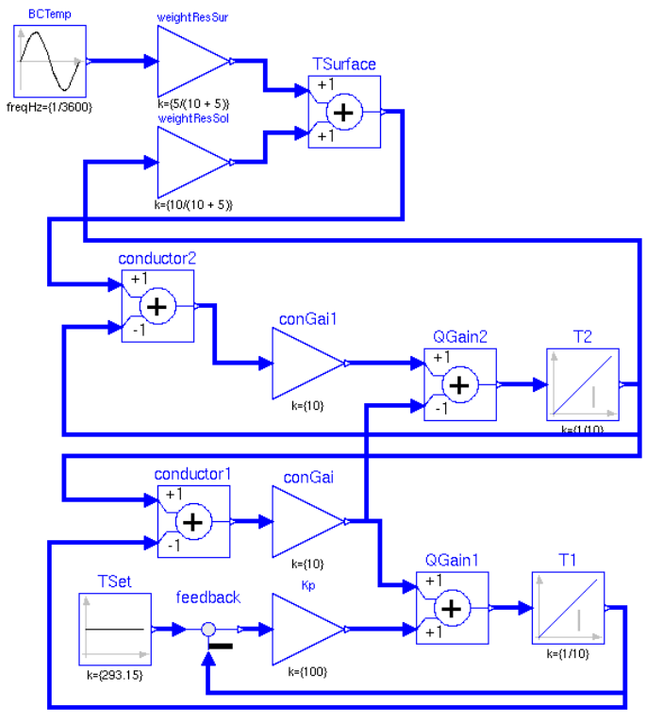
Acausal Modeling

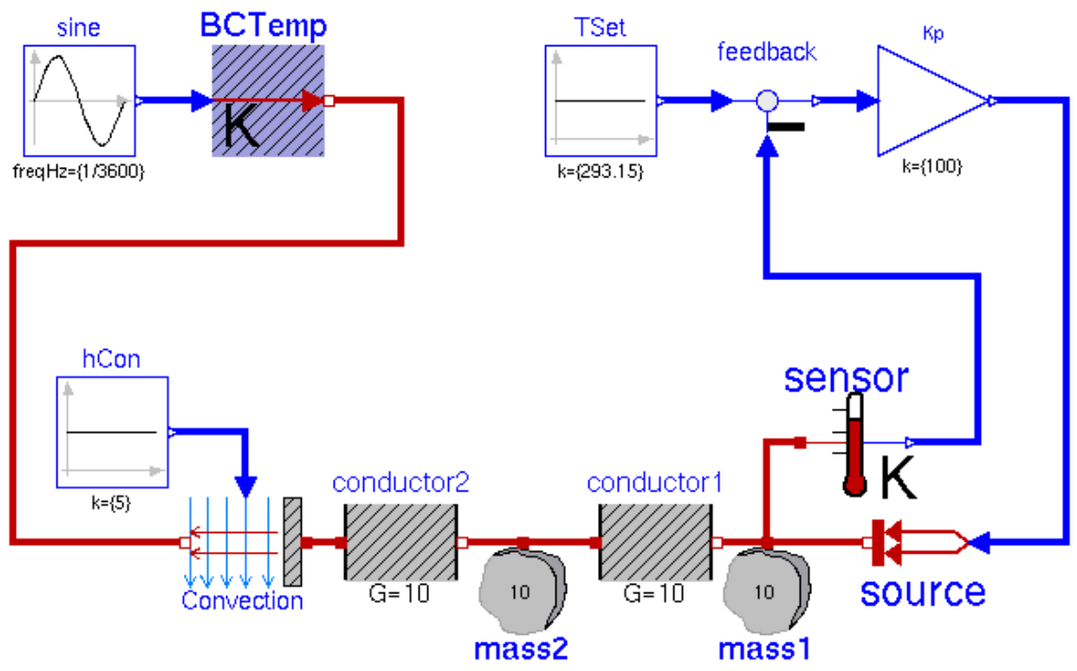
This port carries temperature and heat flow rate. Hence, an electrical pin cannot be connected to it

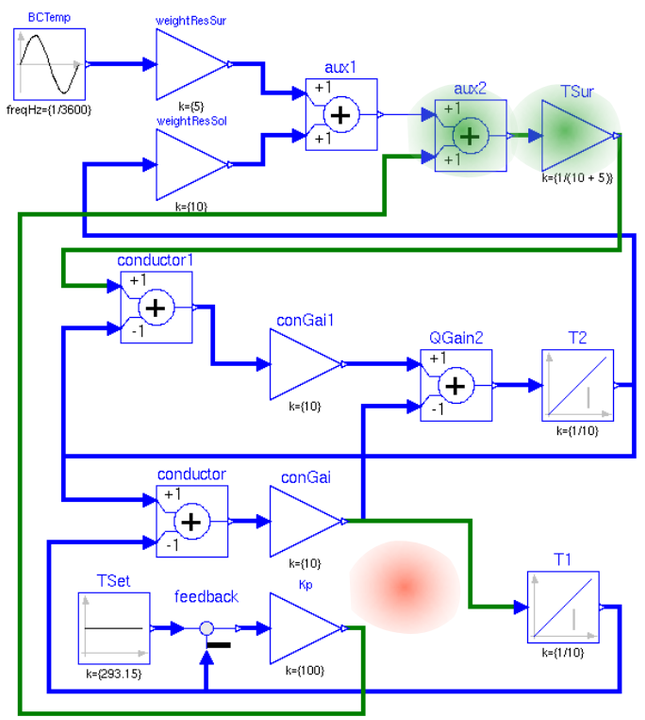


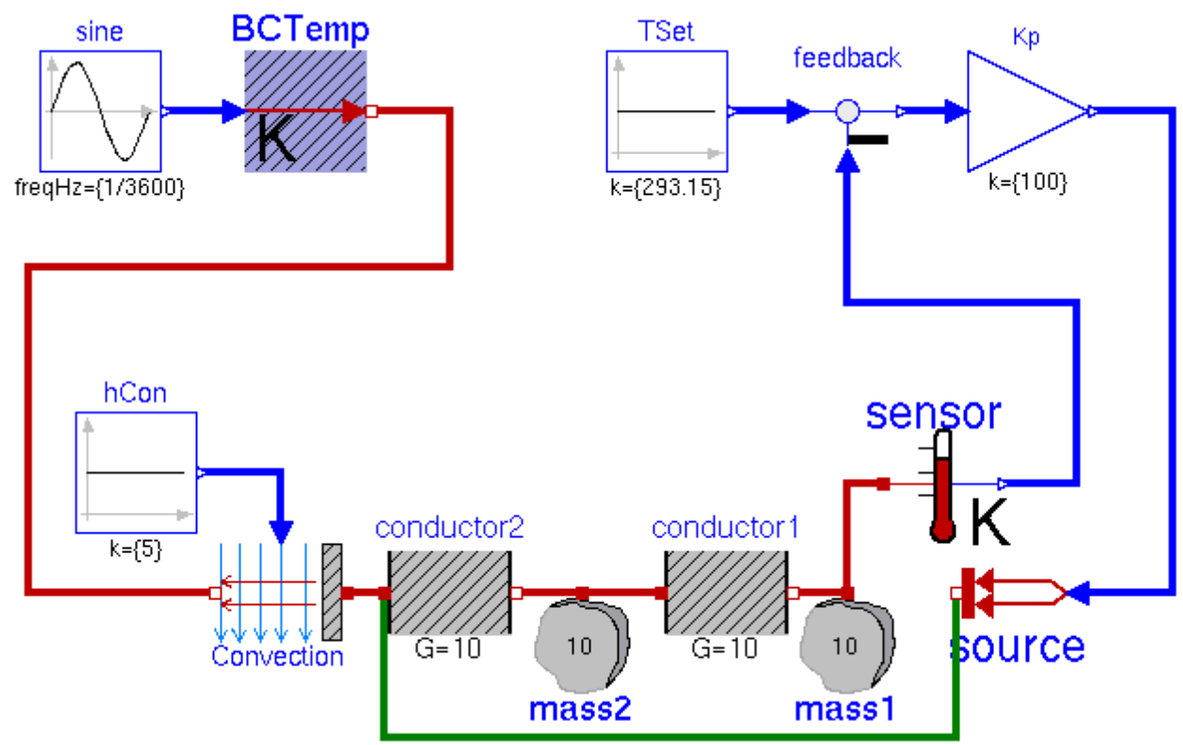
What does it mean to connect three ports?

Acausal components leads to much higher readability and reusability as they allow to declare the system architecture







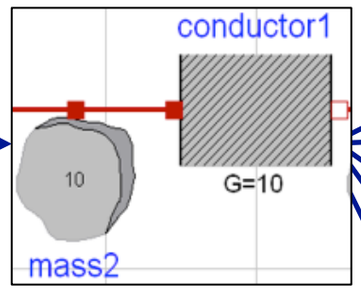


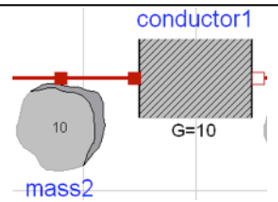
How does it differ from other building simulators?

Separation of concern

Modeling

Specifies the system



	Graphical modeling - input/output free - block-diagram - state machines - bond-graphs
`a:=2;` `b:=2*a;`	Algorithmic code
`C*der(T) = Q_flow;` `0 = T - TBoundary;`	Acausal equations
`external "C"` `y=someCFunction(x);`	C interface

Computation

Solves the equations

Code for time-domain simulation

Code for real-time operation

Limited memory and storage.

Constraints on computing time.

Code for optimization

Differentiation for gradient.

Symbolic processing for collocation.

Code for co-simulation as FMU

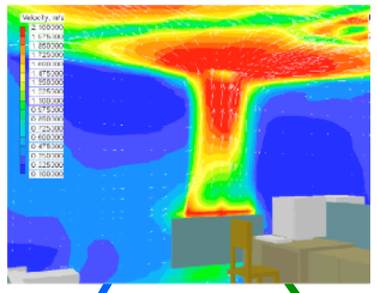
Provide API for model discovery and that returns $x_{k+1}=f(x_k, t_k)$

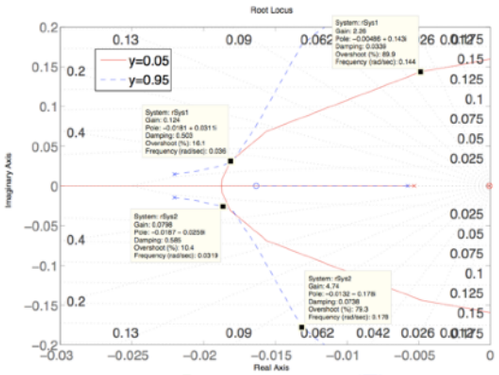
Code for model exchange as FMU

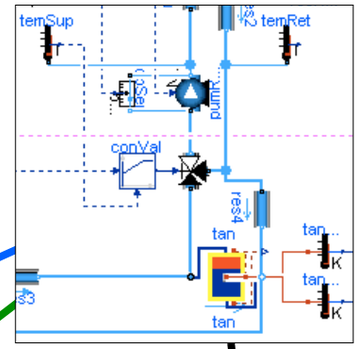
Provide API for model discovery and that expose right-hand-side of dx(t)/dt=f(x(t),t)

Related efforts within the building simulation community include ENET (Low and Sowell, 1982), SPANK (Sowell et al., 1986), SPARK (Buhl et al., 1993) and the Neutral Model Format NMF (Sahlin and Powell, 1989).

Context: Models need to be compatible with up-stream data-formats and serve down-stream applications



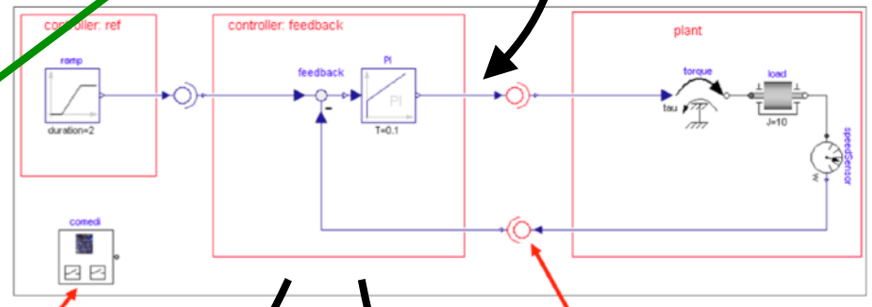




data

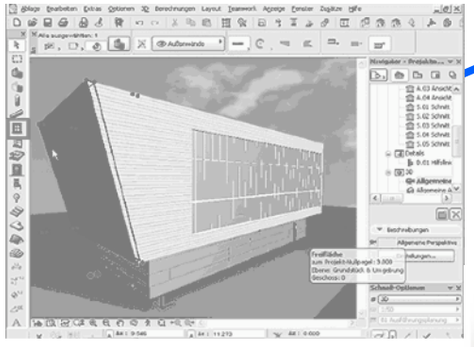
dynamics

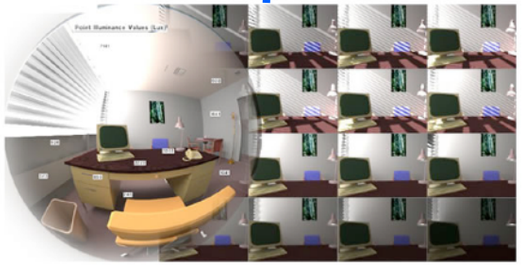
$$\dot{x} = f(x)$$

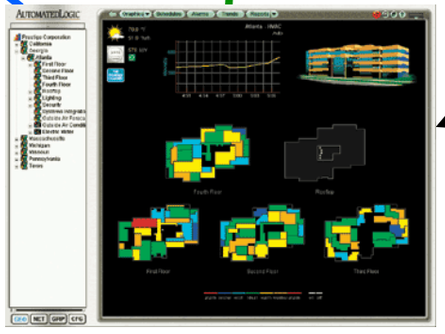


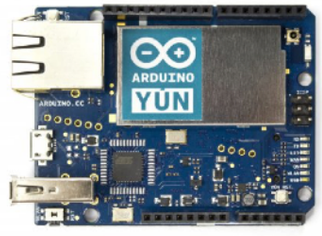
defines configuration of embedded system
(sampling, ECUs, initialization of device drivers, ...)

Defines
- splitting in task/sub-task,
- device drivers,
- references configuration



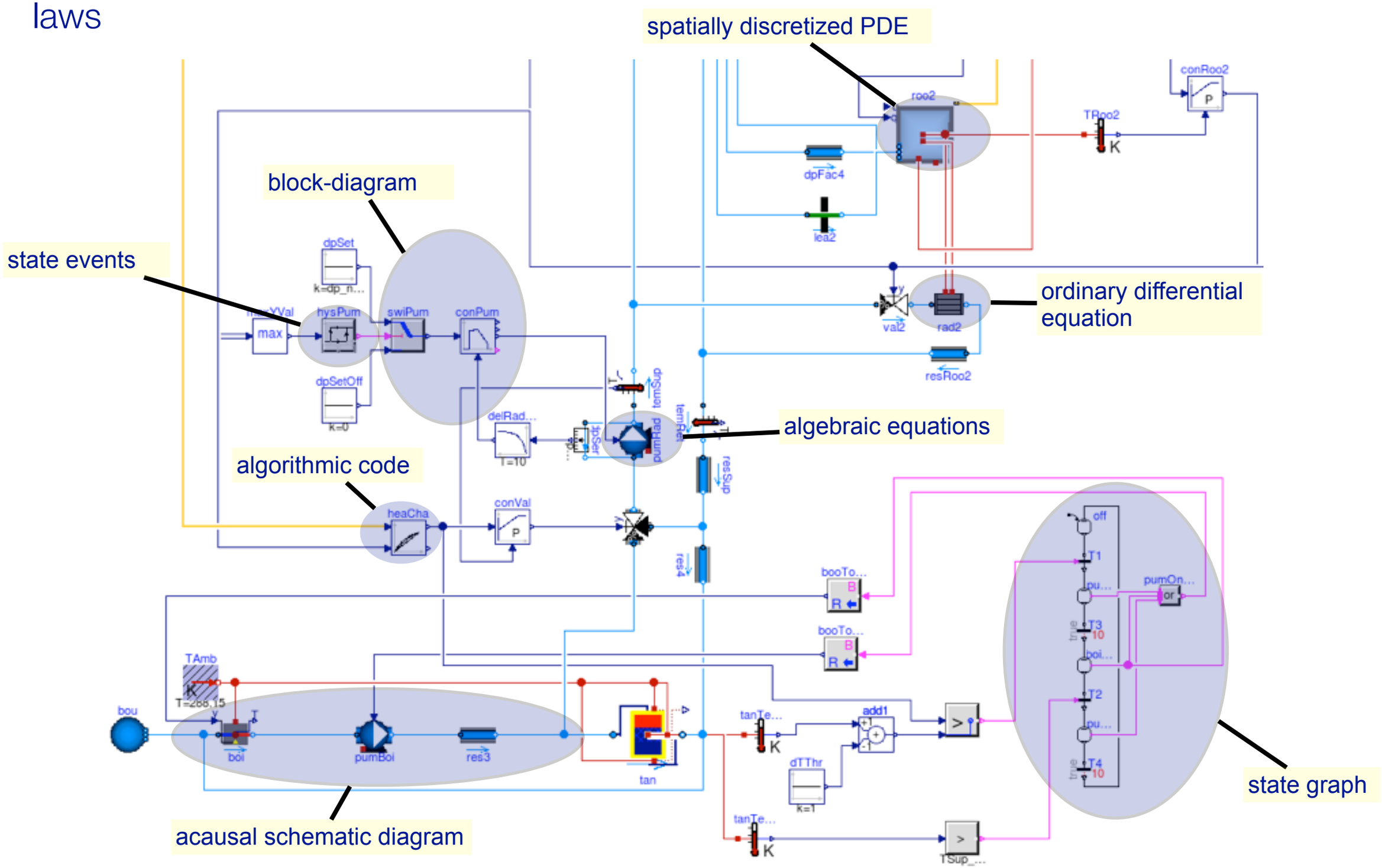






Schematic modeling of complex systems

Complex systems are packaged intuitively: icons represent components and subsystems that evolve differently in time, and lines equate their interface variables *and* impose conservation laws



The above model is composed hierarchically, partially through automatic generation, of 1700 component models.

Support for Optimization - JModelica

Extensible Modelica-based open source platform for optimization,

simulation and analysis of complex dynamic systems.

Main objective:

Create an industrially viable open source platform for optimization of Modelica models.

Offer a flexible platform serving as a virtual lab for algorithm development and research.

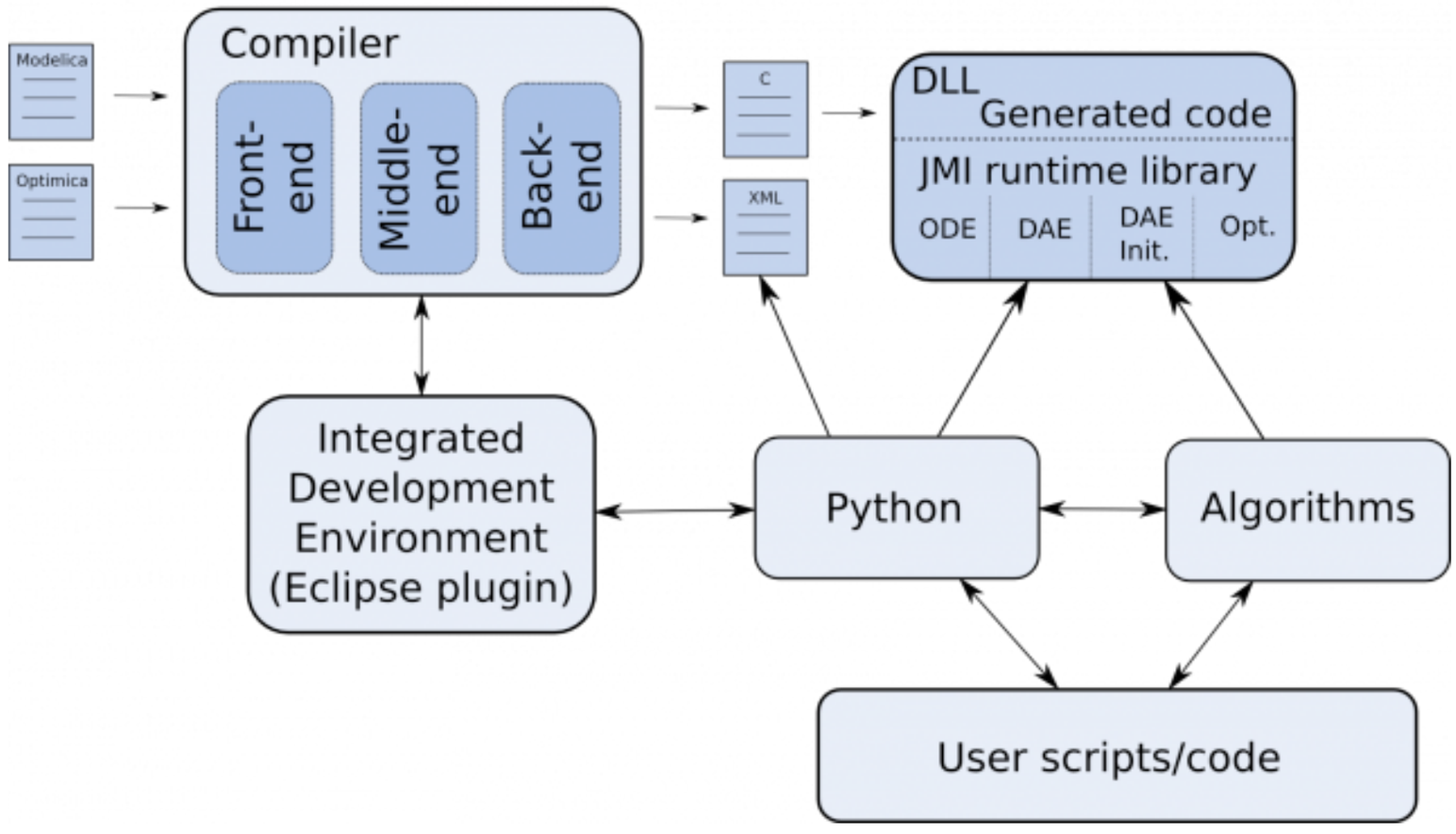
Main features

Optimal control

Parameter estimation

Simulation

Source: http://www.jmodelica.org



Modelica for building and district energy systems (see https://modelica.org/libraries for complete list)

Developed through Annex 60

AixLib (RWTH Aachen)

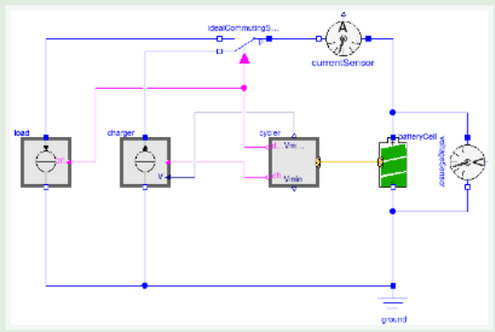
Buildings (LBNL)

BuildingSystem (UdK Berlin)

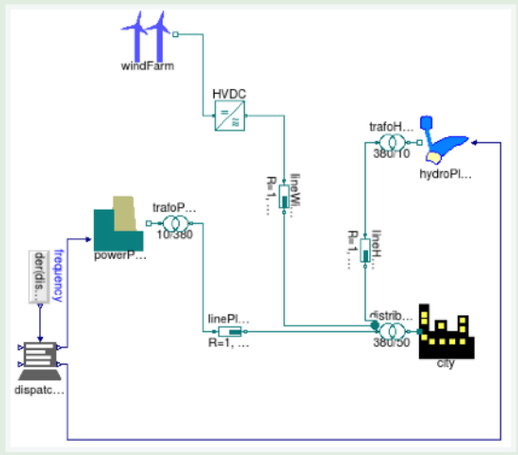
IDEAS (KU Leuven)

Free 3rd party libraries

Energy storage library (batteries)



Power systems library (transmission)



Commercial libraries

Electric power systems

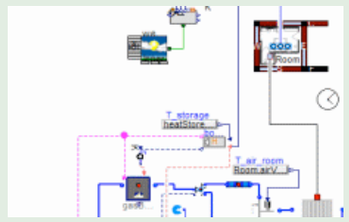
Batteries

Human Comfort

Heat exchanger

Air conditioning

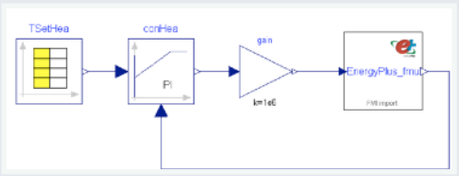
HVAC



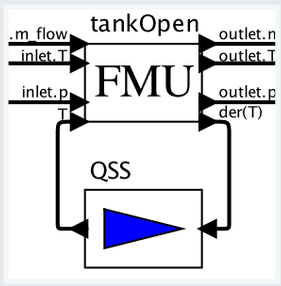
FMI

Simulator deployment and interoperability

FMI export & import in EnergyPlus, Niagara and BCVTB



Co-simulation



More than 80 tools, see https://www.fmi-standard.org/

Further literature

Online book with interactive examples of Michael Tiller at http://book.xogeny.com/ .

Modelica web reference http://modref.xogeny.com/

Interactive Modelica lessons at http://tour.xogeny.com/

The books by Michael Tiller (2001) and Peter Fritzson (2011 and 2004).

The tutorials at https://www.modelica.org/publications.

The Modelica language specification at https://modelica.org/documents.

Although the Modelica Language Tutorial is for an older version (Modelica 1.4), it is still instructive and relevant to understand the concepts of the language.

?