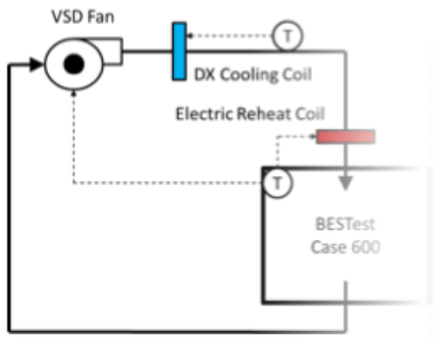
OpenBuildingControl

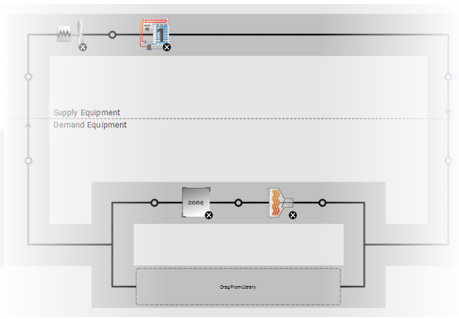
https://obc.lbl.gov

Michael Wetter, Jianjun Hu, Milica Grahovac, Philip Haves, Paul Ehrlich

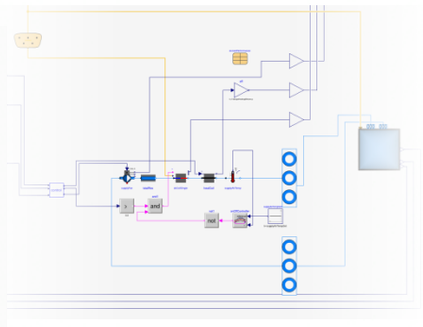
January 22, 2018



Design



Model



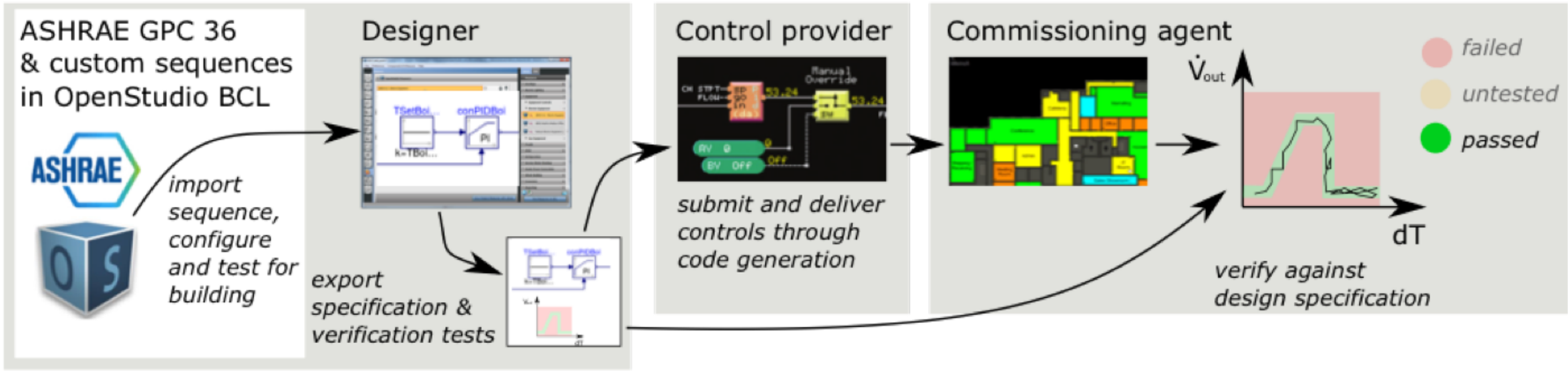
Simulate

Operate

BERKELEY LAB

Lawrence Berkeley National Laboratory

OpenBuildingControl: Design and implement control sequences error-free and at lower cost to owner



Codify best practice | Design | Implement | Verify against original design

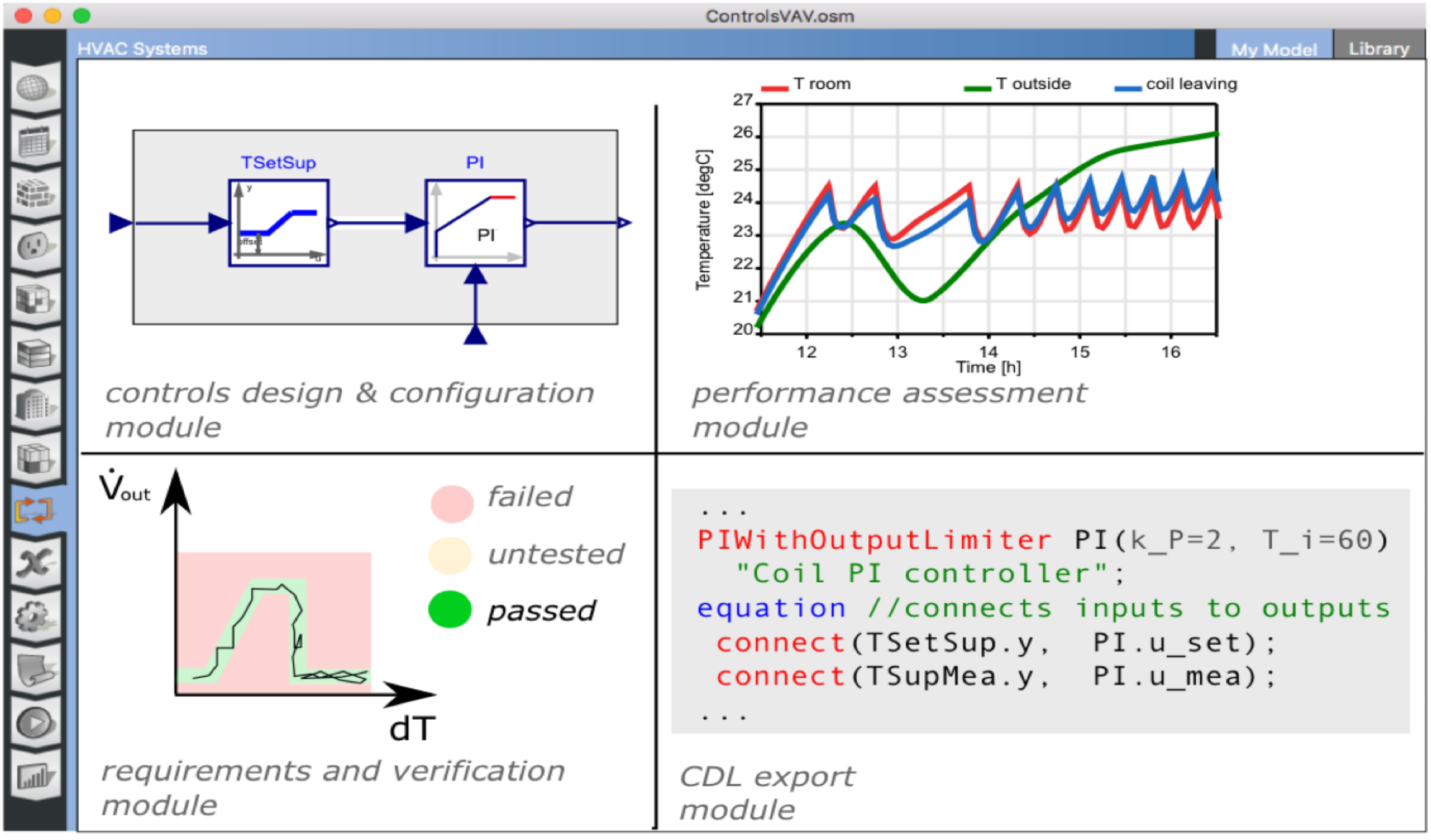
BACnet standardizes communication.

OpenBuildingControl will standardize:

- basic functional building blocks that are used to compose sequences and tests,
- expressing control sequences,
- expressing functional verification tests,

for bidding, automatic implementation and automated functional testing.

OpenBuildingControl: Design and implement control sequences error-free and at lower cost to owner



Points list + Bidding documents + Operator manual + …

Control Description Language

For specifications:
http://obc.lbl.gov/specification/cdl.html#sec-cdl

To browse the blocks:
http://simulationresearch.lbl.gov/modelica/releases/latest/help/Buildings_Controls_OBC_CDL.html

What is CDL?

A language used to specify control sequences and verification tests.

- Non-proprietary
- Open-source
- Non-vendor specific

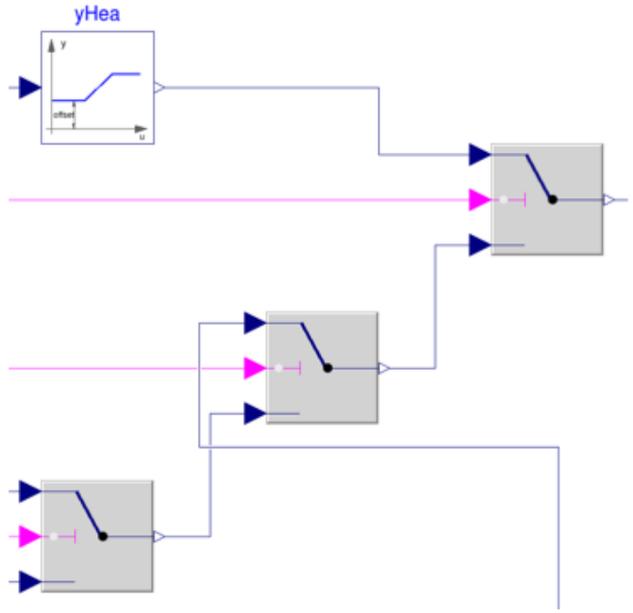
Not a control sequence.

Control sequences are specified, in a declarative way, *using* CDL.

What is CDL?

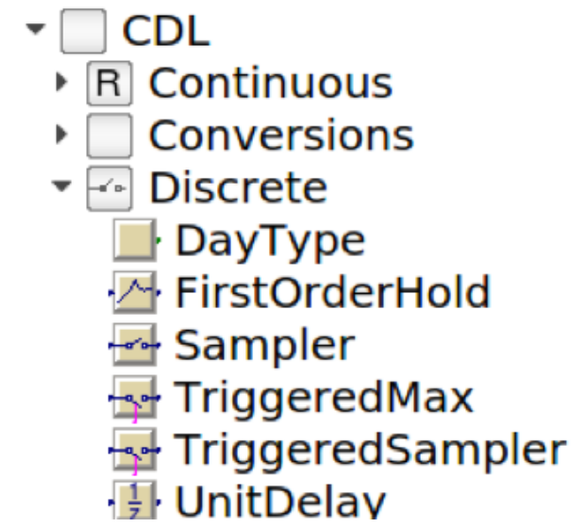
A declarative, open-standard, open-source, non-vendor-specific, language for expressing block-diagrams for controls (and requirements)

A graphical language for rendering these diagrams.



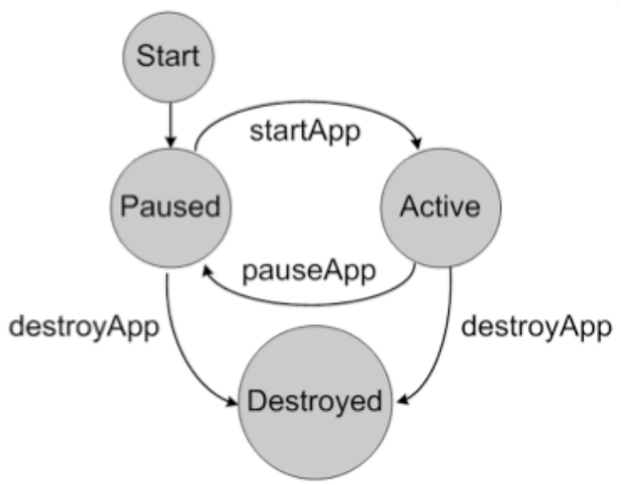
A library with elementary input/output blocks that should be supported [through a translator] by CDL-compliant control providers.

Example: CDL has an adder with inputs `u1` and `u2`, gains `k1` and `k2`, and output
`y = k1*u1 + k2*u2`.



A syntax for documenting the control blocks and diagrams.

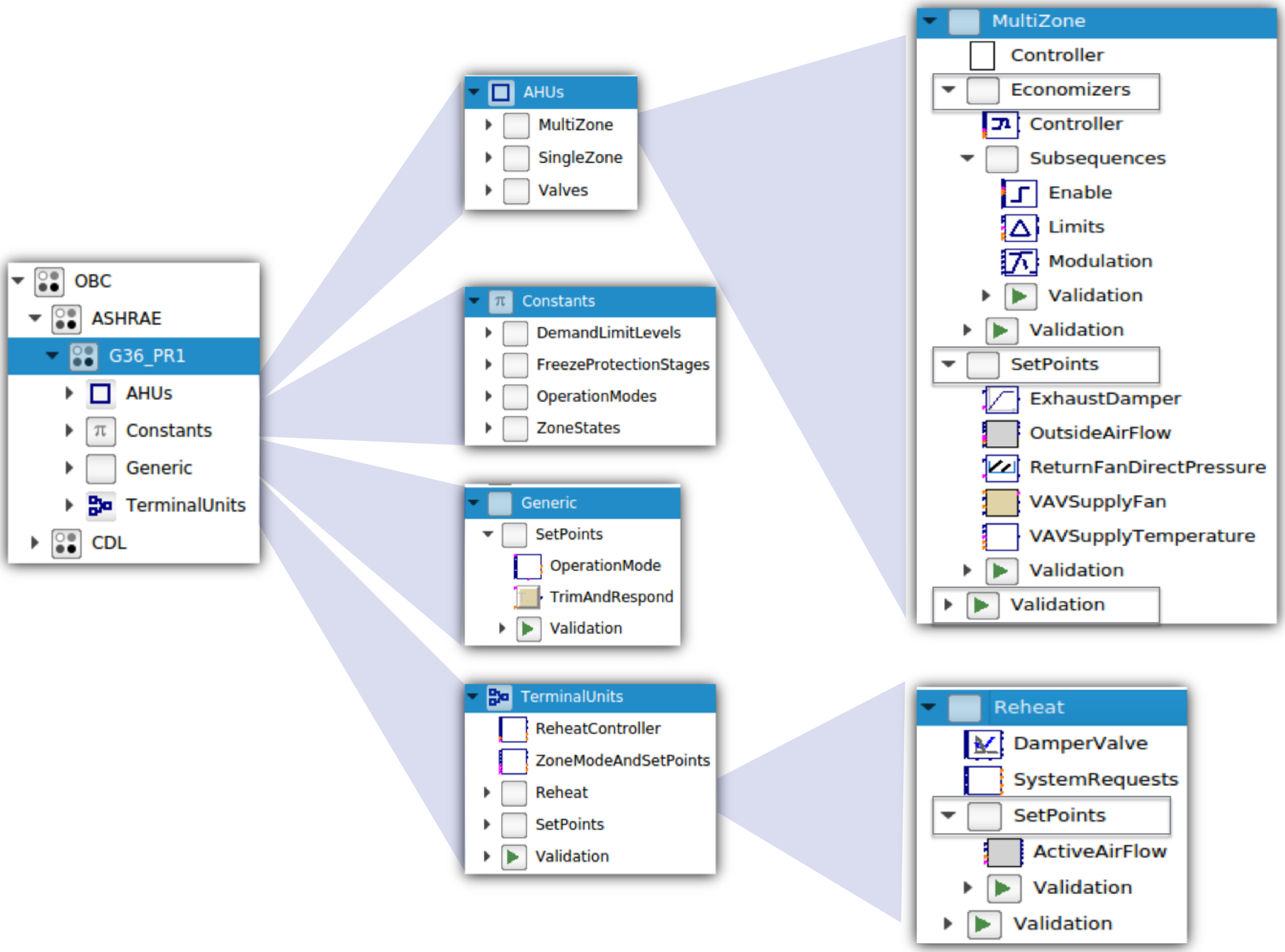
A model of computation that describes the interaction among the blocks.



Case study

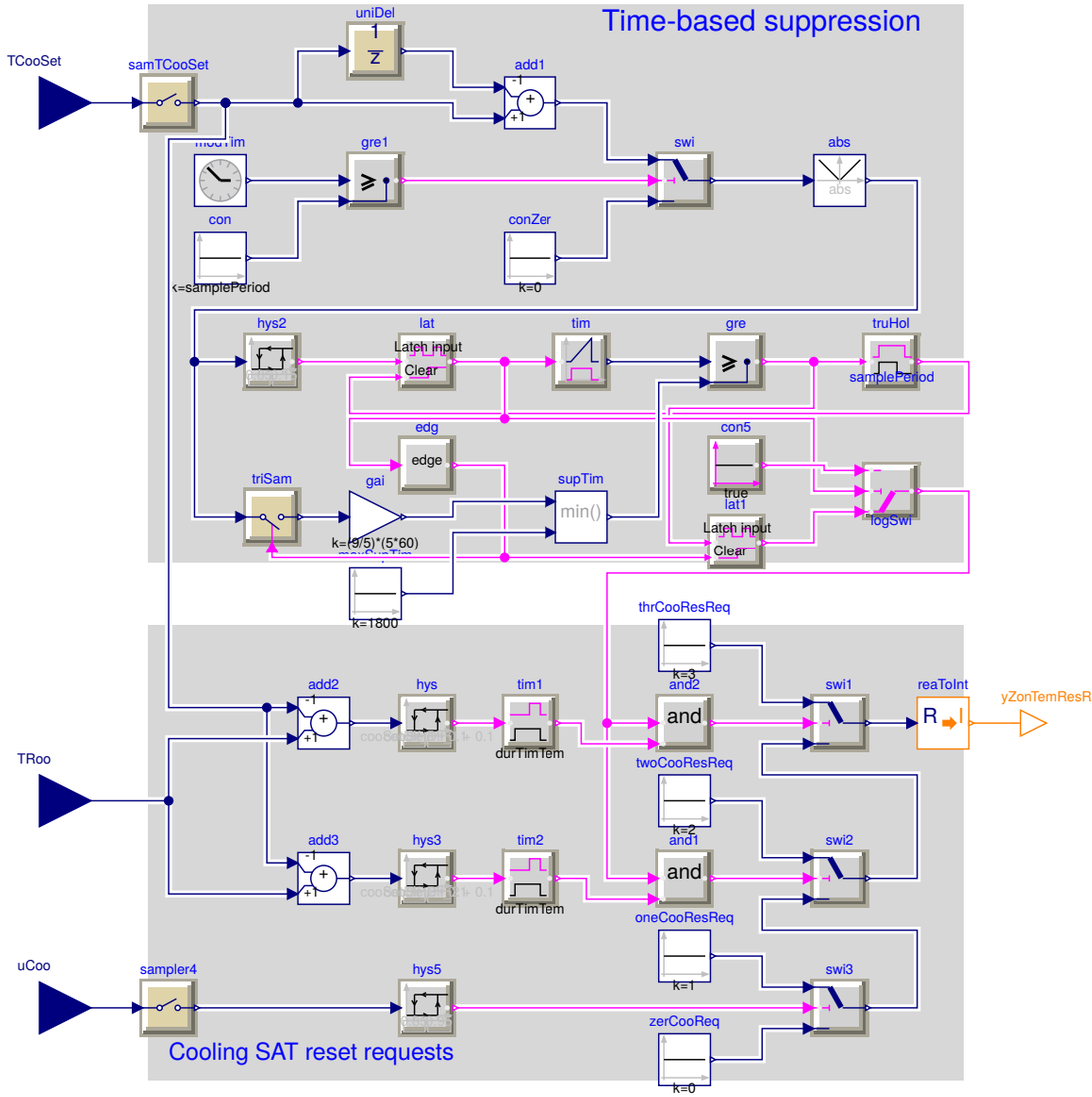
For report, see
http://obc.lbl.gov/specification/example.html or
https://github.com/lbl-srg/obc/tree/master/meetings/2017-11-tag

Implemented part of Guideline 36 sequences in CDL



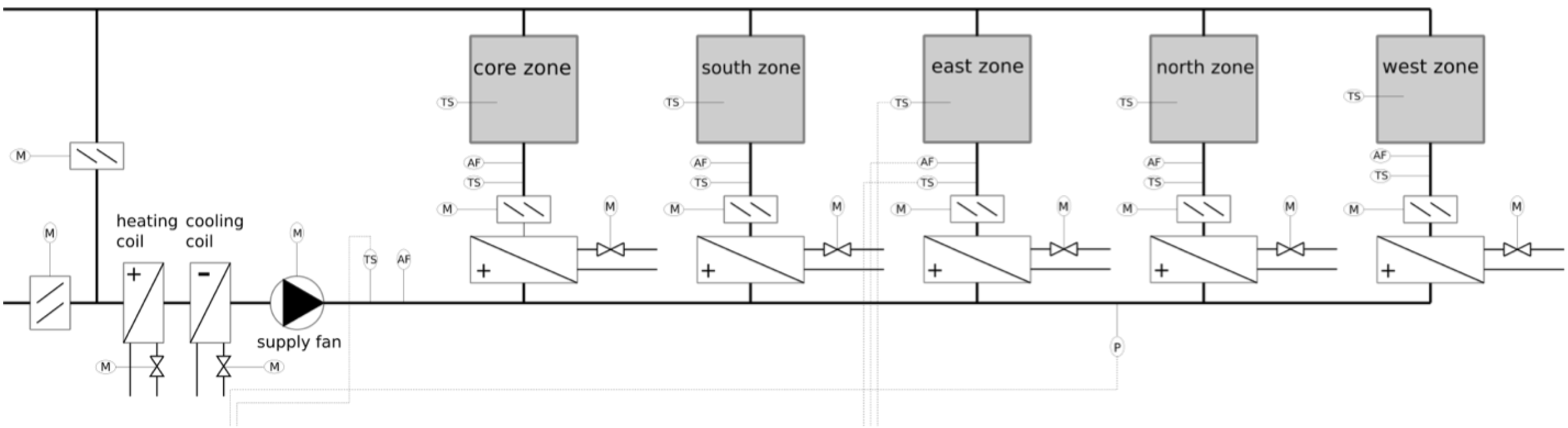
Case study: Modeling multi-zone VAV controls and equipment

- Full airflow network.
- Wind pressure driven infiltration.
- All flows based on flow friction, damper positions and fan curves.
- 4,000 components, 40,000 variables, adaptive time step, state/time events.



Static pressure reset requests

Hot water reset requests

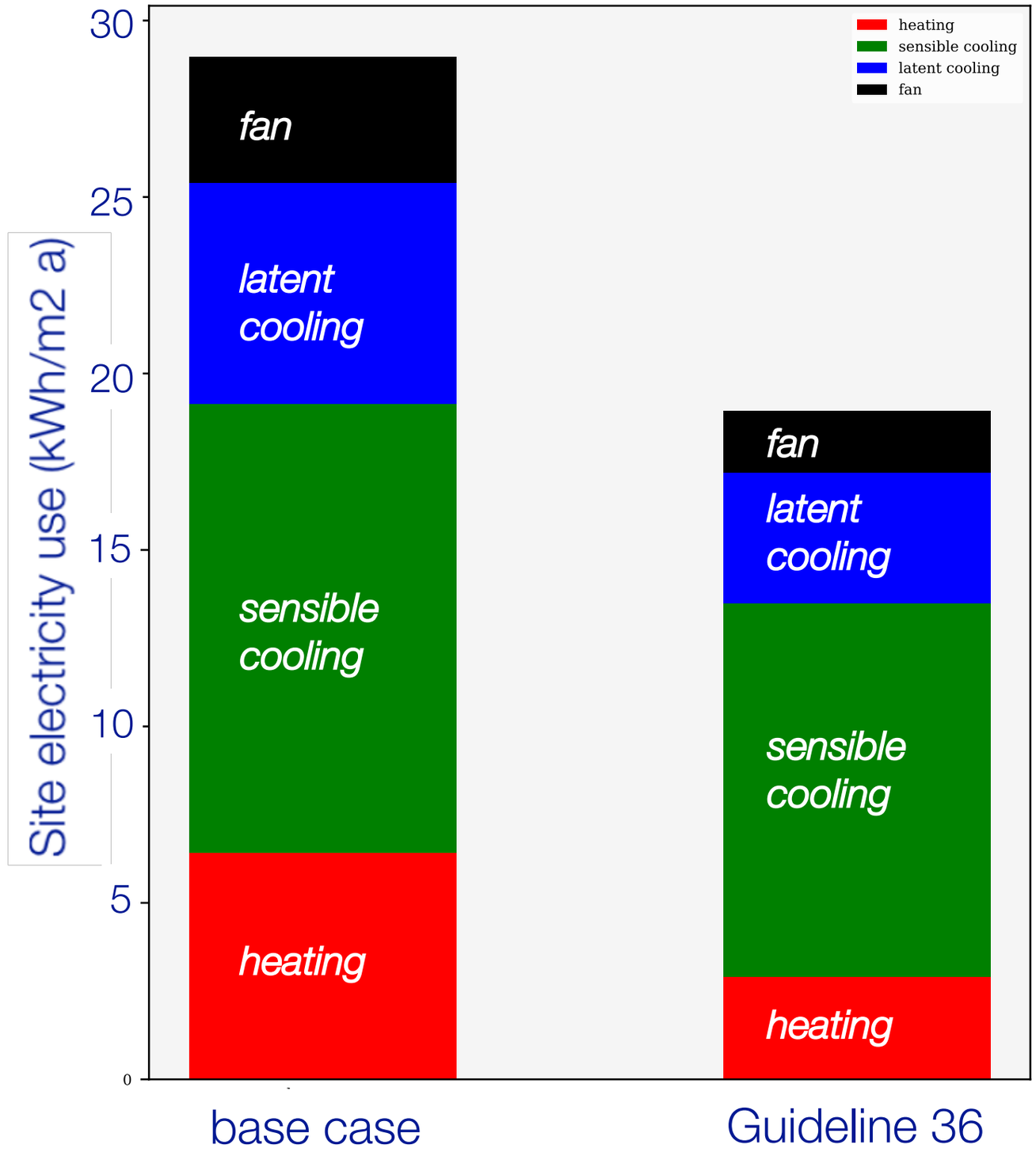


Key take-aways

~30% annual site HVAC energy savings for Chicago, solely due to controls.

Can simulate actual control sequences, with dynamic response.

Packaging of sequences is important, because interpretation and implementation of the sequences was more time-consuming and error-prone than anticipated.



Benefits of OBC approach and tools

Mechanical designer:

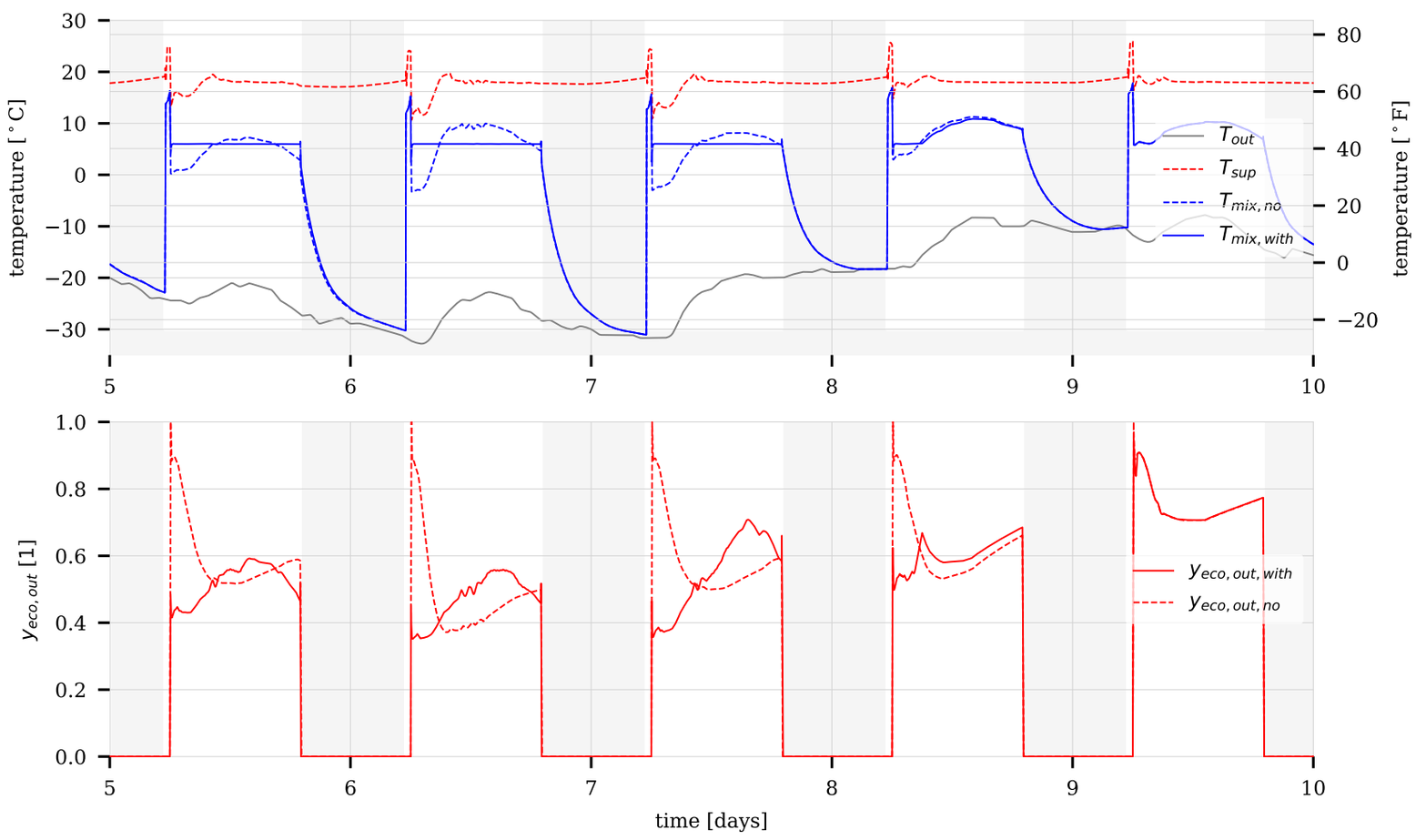
- HVAC design analysis tool
  - Equipment and controls
- Adapt and test sequences for particular building

Controls provider:

- Faster, higher quality, error-free automated implementation

GPC36:

- Formal way to test, compare and deploy sequences



Possible future uses (beyond the current project)

- Publish sequences as **reference implementations** against which vendor implementations can be compared.
- Use as **benchmark for comparing/rating advanced control sequences** such as MPC.
- **Verify G36 savings are robust** through performance assessment across different climates.
- Use OBC to test and prioritize future sequences to be added to G36.
- Potential to work with ASHRAE to move **CDL into a standard**.

Deployment of sequences

Process flow

Mechanical designer

Control provider

Commissioning agent

Q&A-based sequence selector (such as CtrlSpecBuilder)

CDL libraries
- G36
- GSA
- custom-libraries

CDL sequences & mechanical system model

CDL representation to bid for job, and for implementation of real control system

Verification tests

Control vendor bids for project, and later implements sequence semi-automatically

Commissioning agent verifies that design intent is met

Backup

CDL library

Developed CDL library based on review of industrial control libraries.

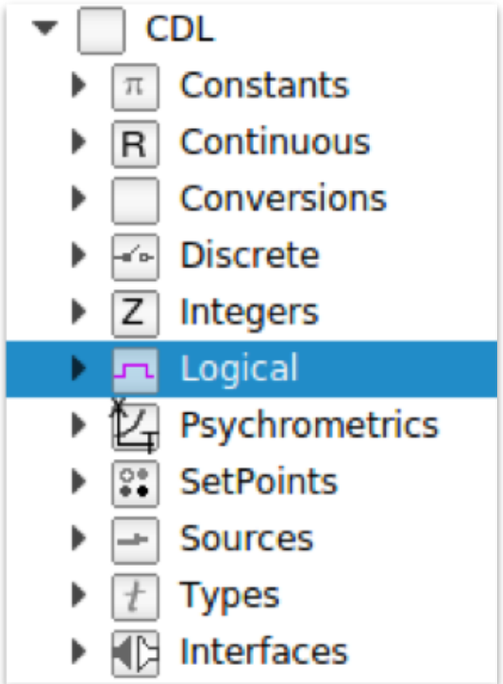
Validated blocks to ensure expected functionalities

- against known results
- across two independent simulators (Dymola and JModelica)

Package name	Description
Constants	Library of constants
Continuous	Library with elementary mathematical functions for continuous variables
Conversions	Library with blocks for type conversion
Discrete	Library of discrete input/output blocks with fixed sample period
Integers	Library with elementary mathematical functions for integer variables
Logical	Library with logical blocks
Psychrometrics	Library with psychrometric blocks
Routing	Package of blocks to combine and extract signals
SetPoints	Package with models for control set points
Types	Package with type definitions
Interfaces	Library with connectors for input and output signals

In CDL library:

- 11 packages
- 134 basic blocks



Browse CDL library at
http://simulationresearch.lbl.gov/modelica/releases/v5.0.0/help/Buildings_Controls_OBC_CDL.html

CDL is used to implement open and proprietary sequences

The standard to be supported by vendors

- CDL
- ASHRAE
- G36

Sequences that come out of ASHRAE projects and can be shared with community.

Custom implementations that are built using the CDL language, and CDL blocks

- GSA
- ARUP
- ALC

GSA preferred sequences, made available through a CDL-complaint implementation.

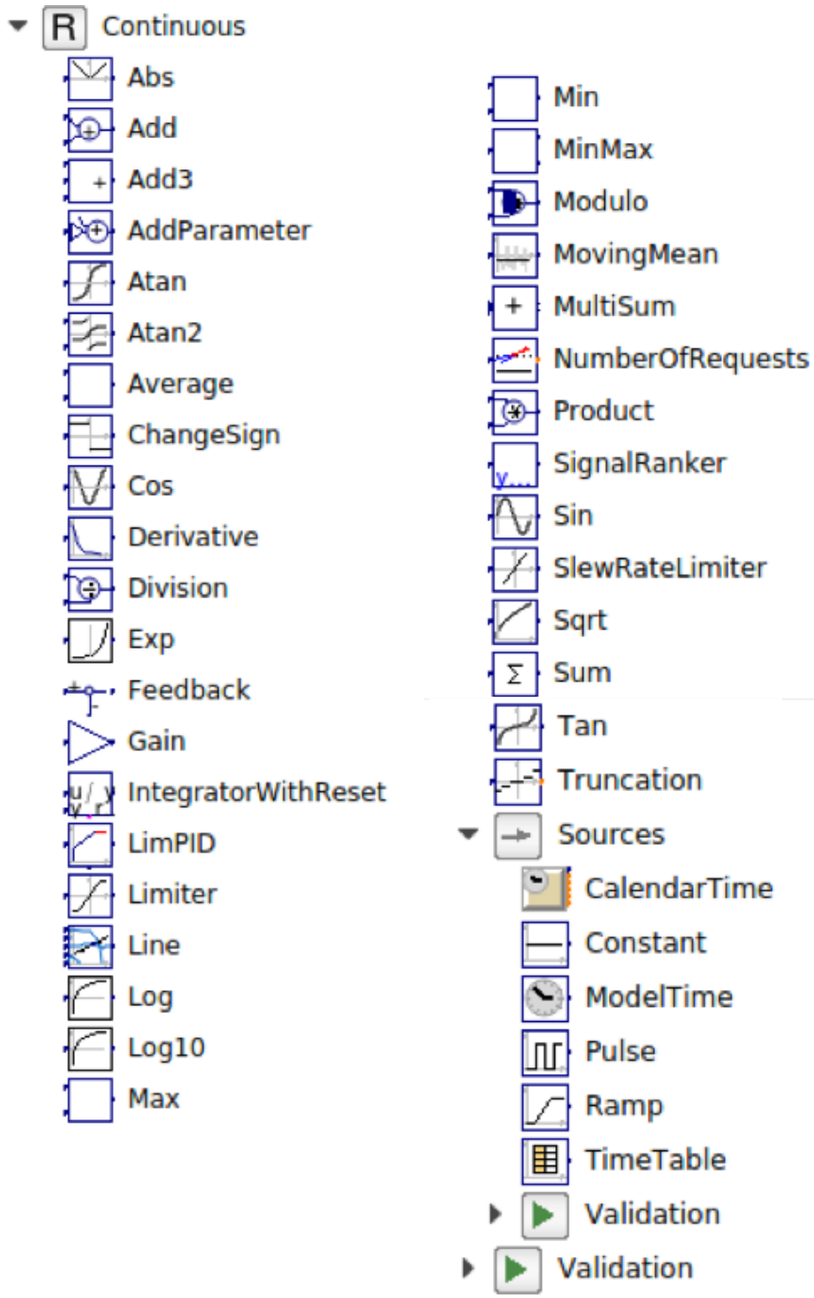
Design firms can share their own (proprietary) implementation across their offices.

Control vendors can provide their own specialized sequences, either as open-source, or as compiled (proprietary) I/O blocks.

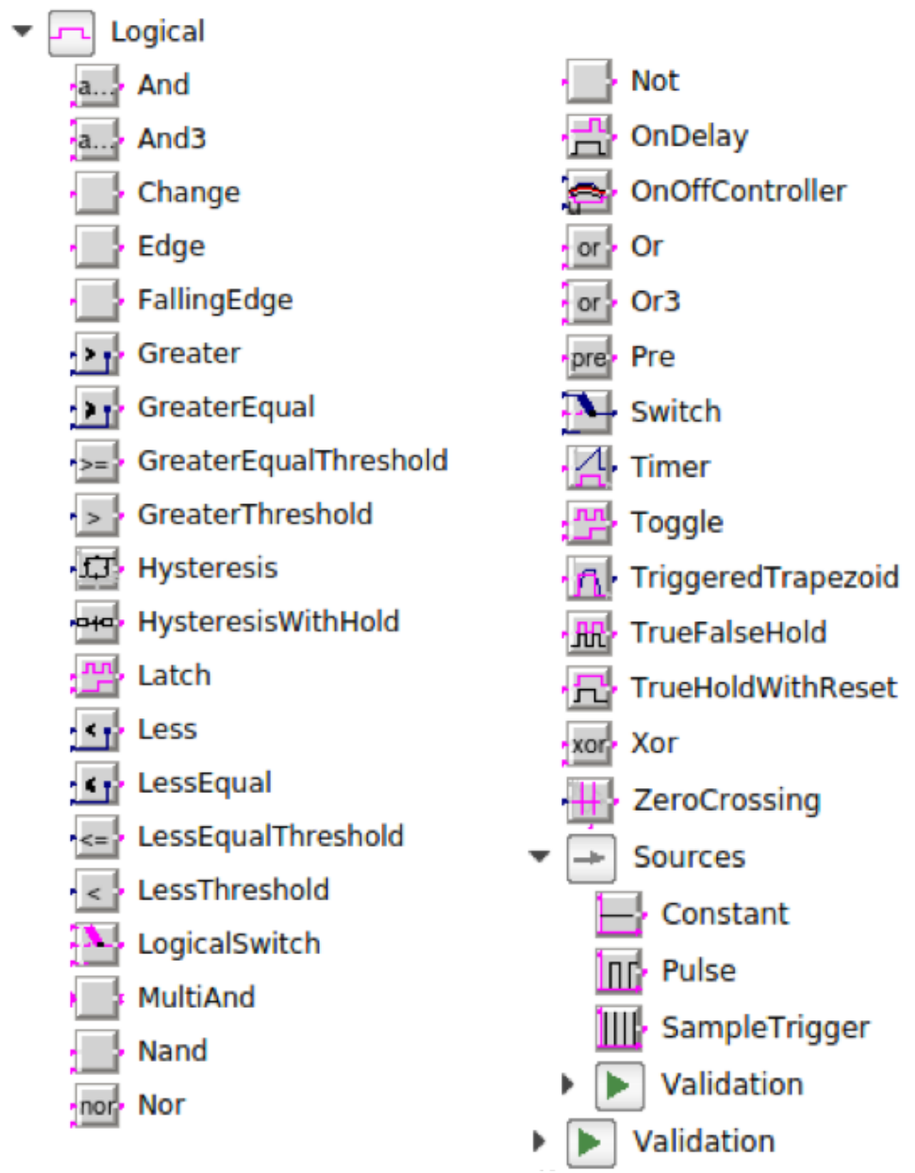
Companies are illustrative

CDL library: Packages

`CDL.Continuous`:
elementary mathematical functions for continuous variables

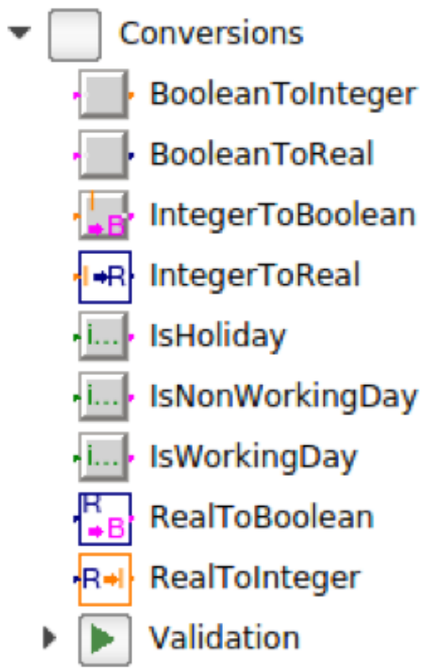


`CDL.Logical`:
elementary mathematical functions for boolean variables

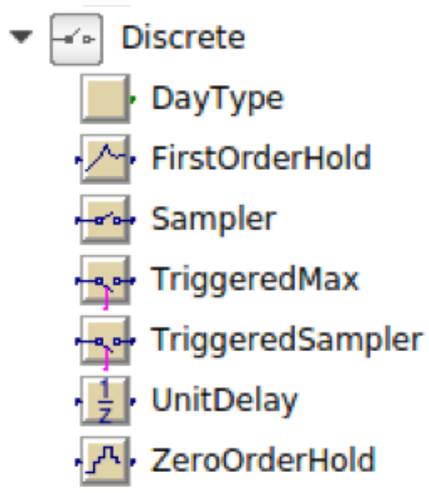


CDL library: Packages

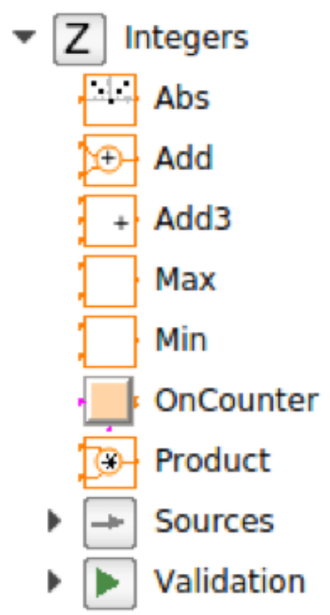
`CDL.Conversions`:
type conversions



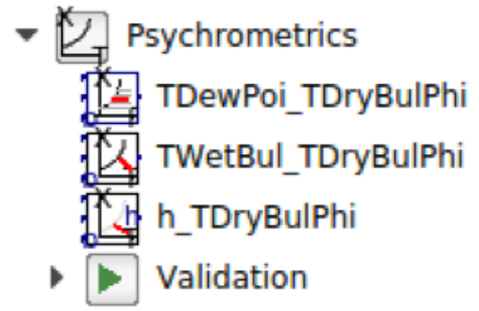
`CDL.Discrete`:
daytype, sample, delay, hold



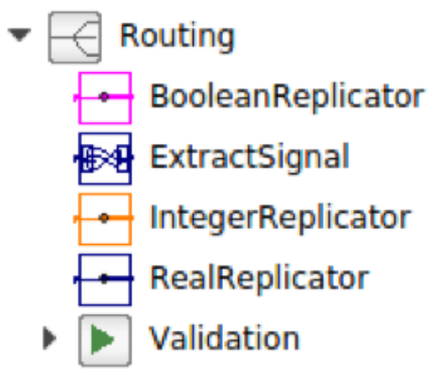
`CDL.Integers`:
mathematical functions for integer variables



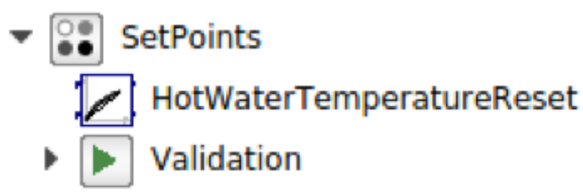
`CDL.Psychrometrics`:
psychrometric calculations



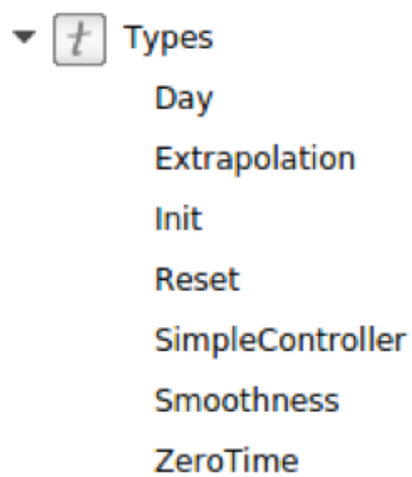
`CDL.Routing`:
combine and extract signals



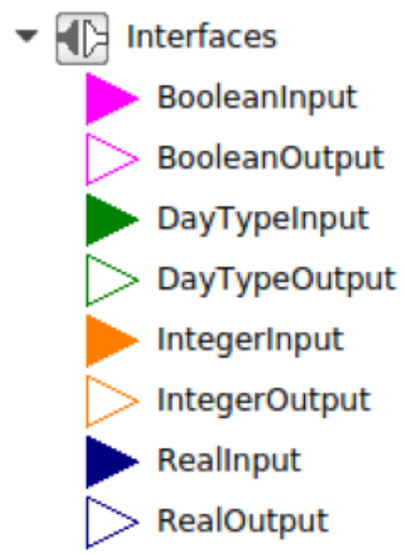
`CDL.Setpoints`:
setpoints for control systems



`CDL.Types`:
type definitions

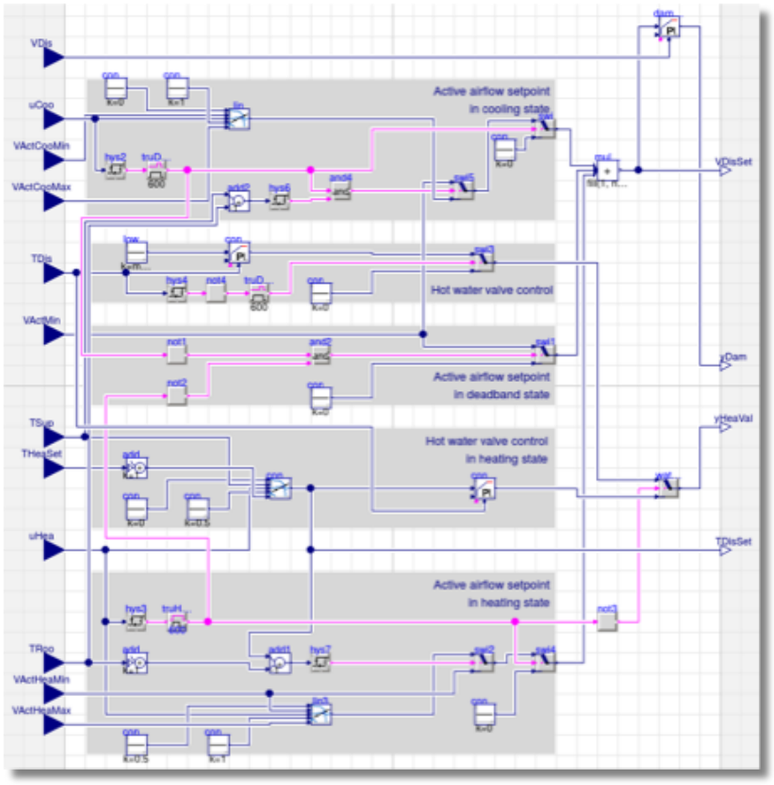


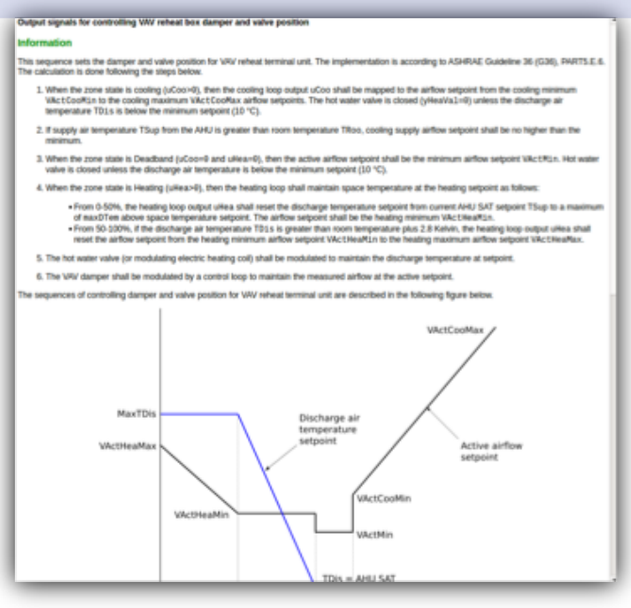
`CDL.Interfaces`:
connectors for input and output signals



Implement sequences with CDL

Documented and validated sequences





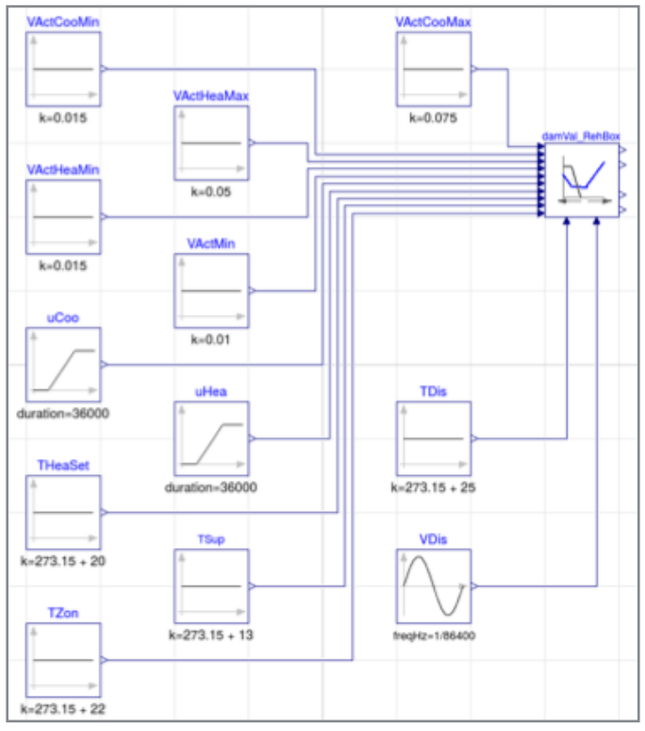
Output signals for controlling VAV reheat box damper and valve position

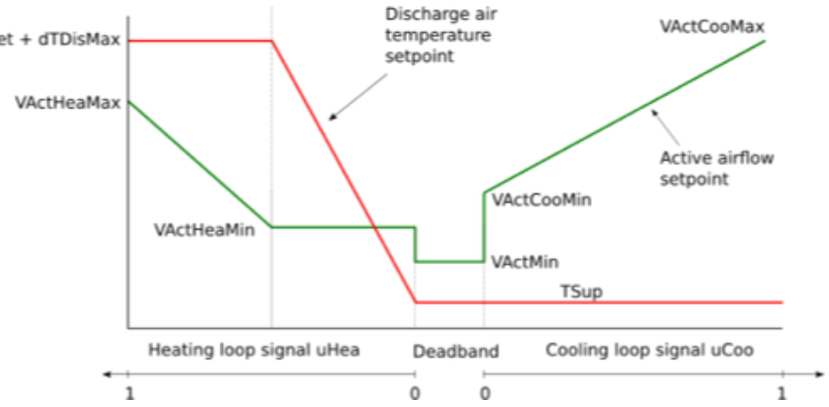
Information

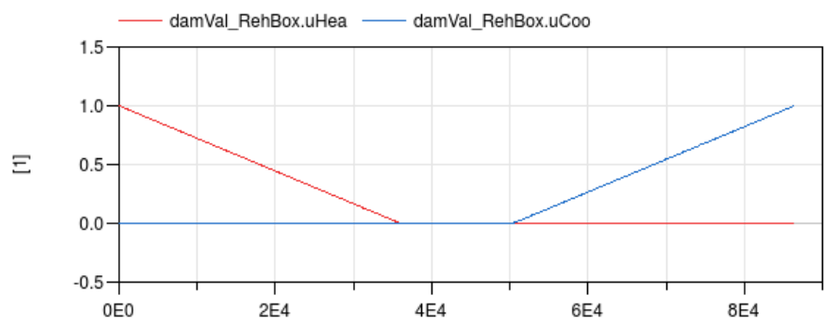
This sequence sets the damper and valve position for VAV reheat terminal unit. The implementation is according to ASHRAE Guideline 36 (G36), PART5.E.6. The calculation is done following the steps below.

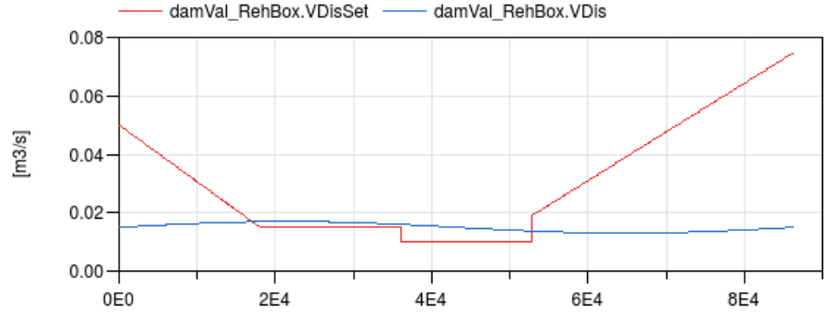
1. When the zone state is cooling (uCoo>0), then the cooling loop output uCoo shall be mapped to the airflow setpoint from the cooling minimum VActCooMin to the cooling maximum VActCooMax airflow setpoints. The hot water valve is closed (yHeaVal=0) unless the discharge air temperature TDis is below the minimum setpoint (10 °C).
2. If supply air temperature TSup from the AHU is greater than room temperature TRoo, cooling supply airflow setpoint shall be no higher than the minimum.
3. When the zone state is Deadband (uCoo=0 and uHea=0), then the active airflow setpoint shall be the minimum airflow setpoint VActMin. Hot water valve is closed unless the discharge air temperature is below the minimum setpoint (10 °C).
4. When the zone state is Heating (uHea>0), then the heating loop shall maintain space temperature at the heating setpoint as follows:
   - From 0-50%, the heating loop output uHea shall reset the discharge temperature setpoint from current AHU SAT setpoint TSup to a maximum of maxDTem above space temperature setpoint. The airflow setpoint shall be the heating minimum VActHeaMin.
   - From 50-100%, if the discharge air temperature TDis is greater than room temperature plus 2.8 Kelvin, the heating loop output uHea shall reset the airflow setpoint from the heating minimum airflow setpoint VActHeaMin to the heating maximum airflow setpoint VActHeaMax.
5. The hot water valve (or modulating electric heating coil) shall be modulated to maintain the discharge temperature at setpoint.
6. The VAV damper shall be modulated by a control loop to maintain the measured airflow at the active setpoint.

The sequences of controlling damper and valve position for VAV reheat terminal unit are described in the following figure below.



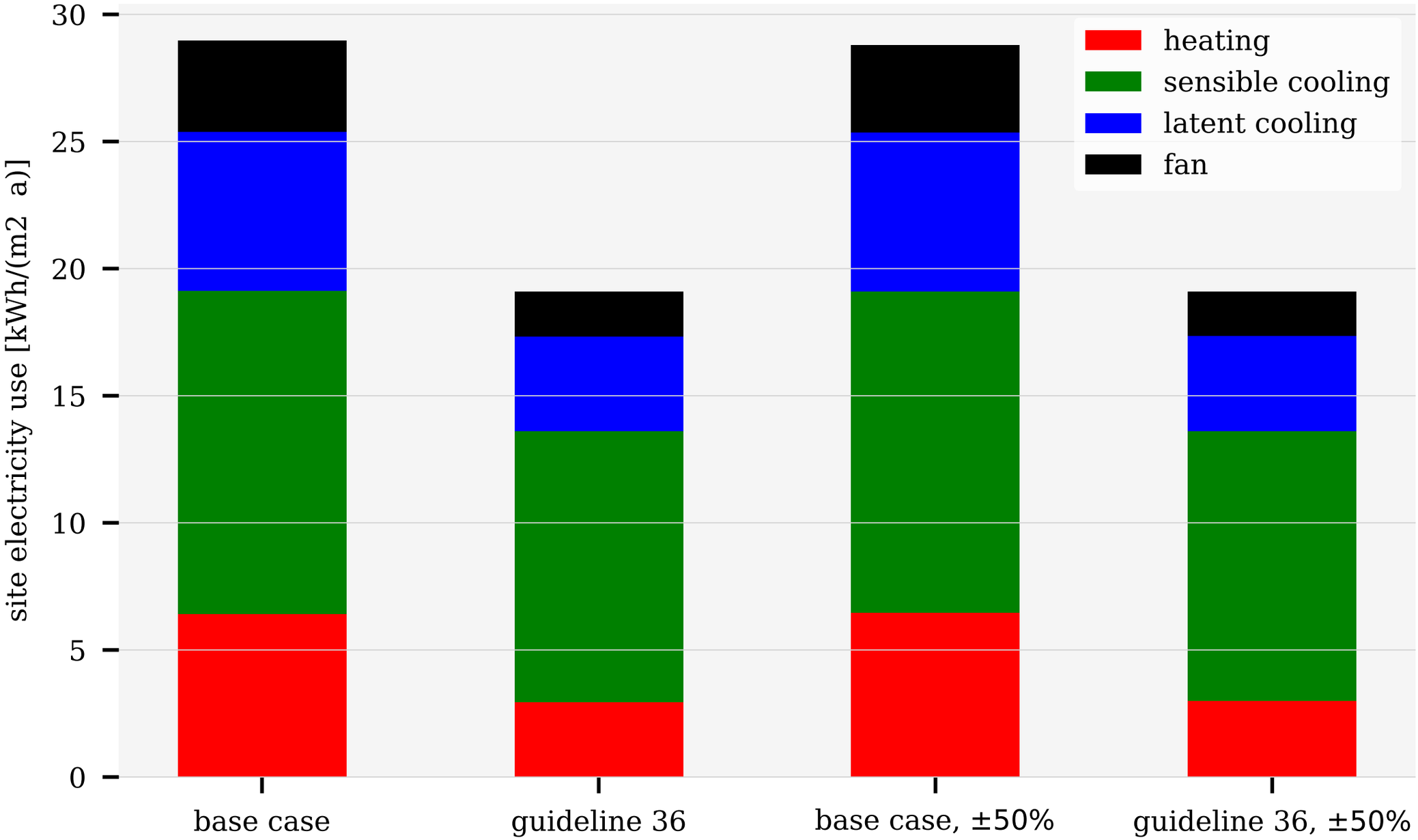






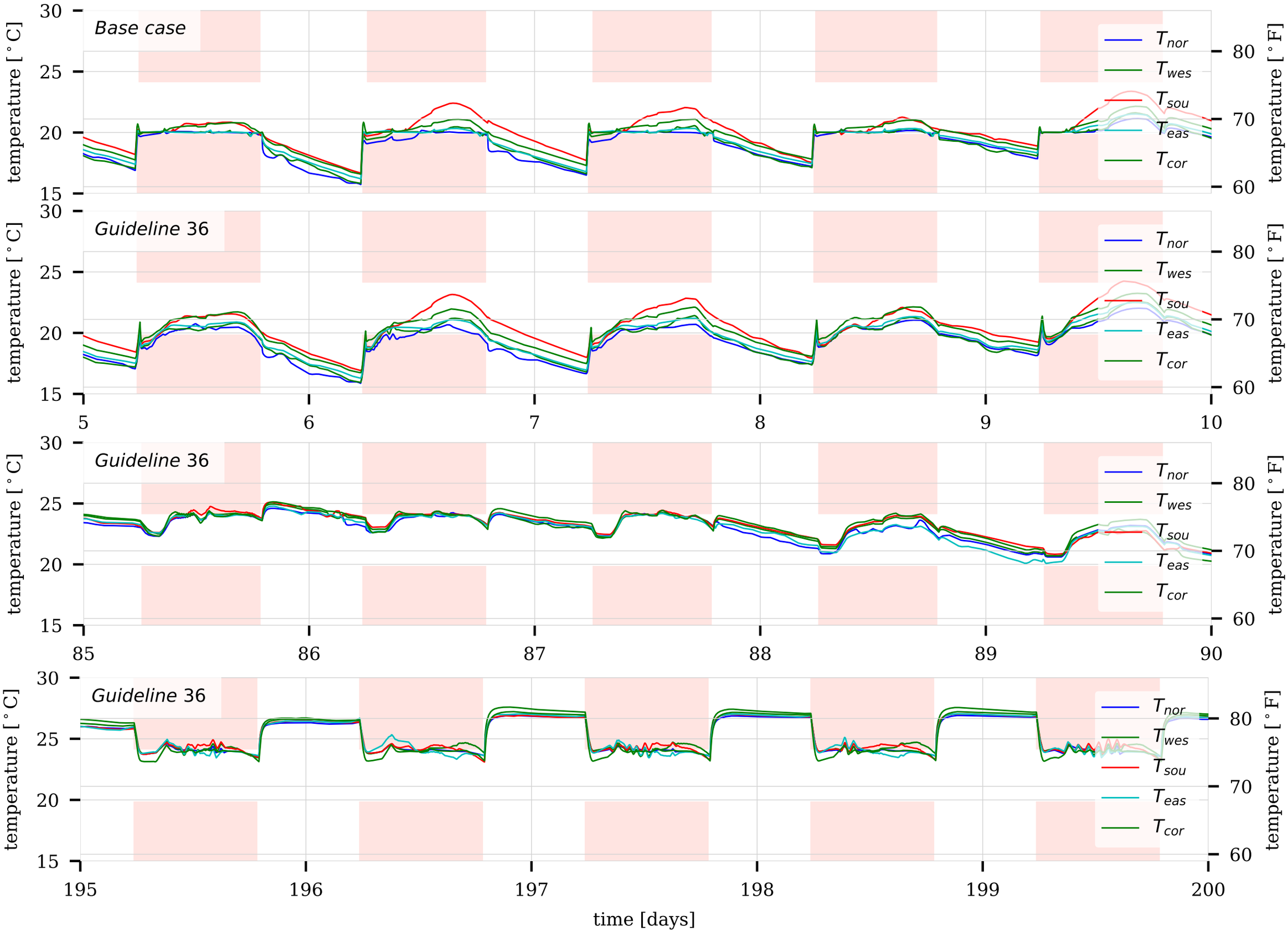
[Buildings.Controls.OBC.ASHRAE.G36_PR1.TerminalUnits.Reheat.DamperValve](#)

Main results



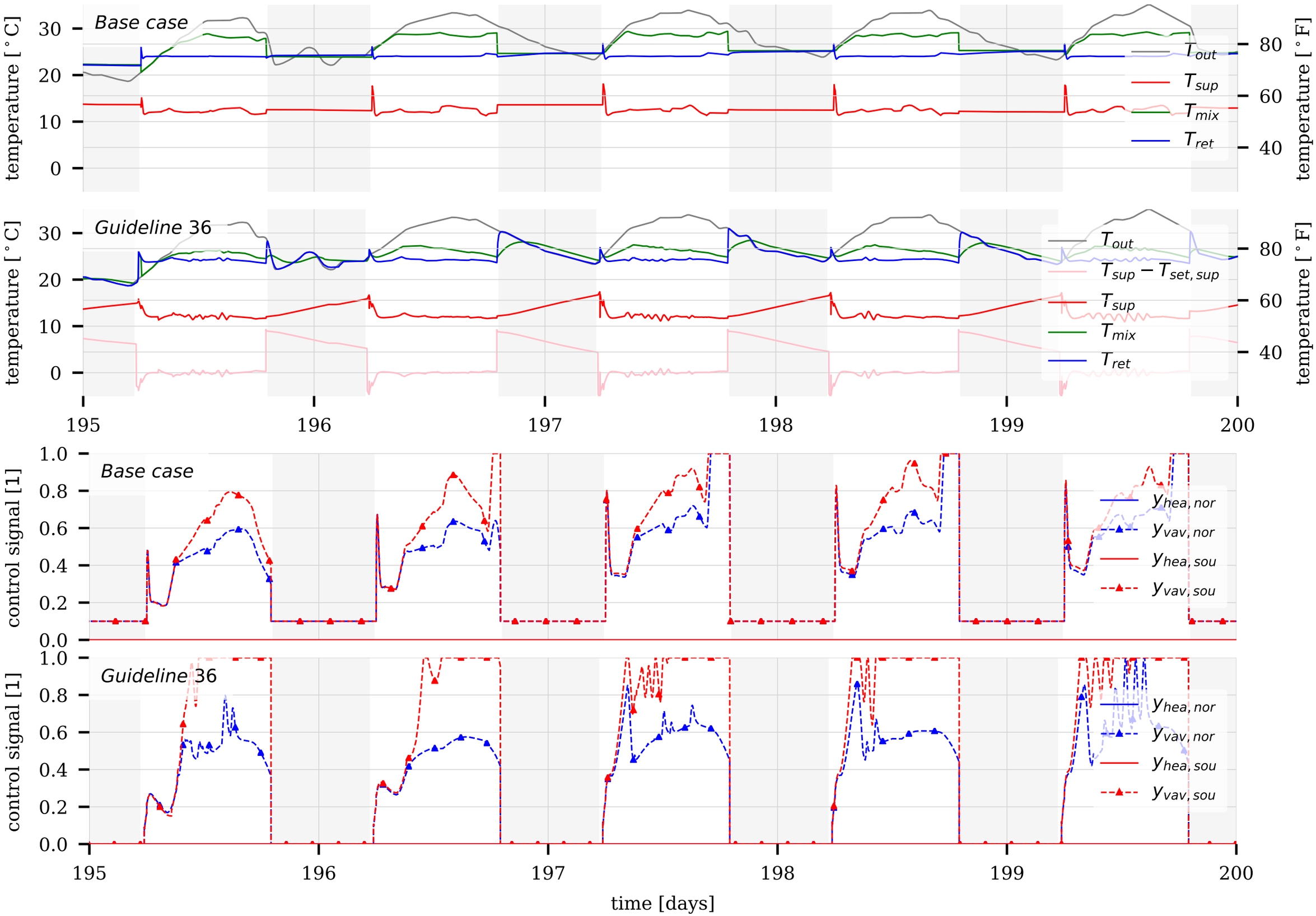
Example output

See http://obc.lbl.gov/specification/example.html
for more trajectories and discussion

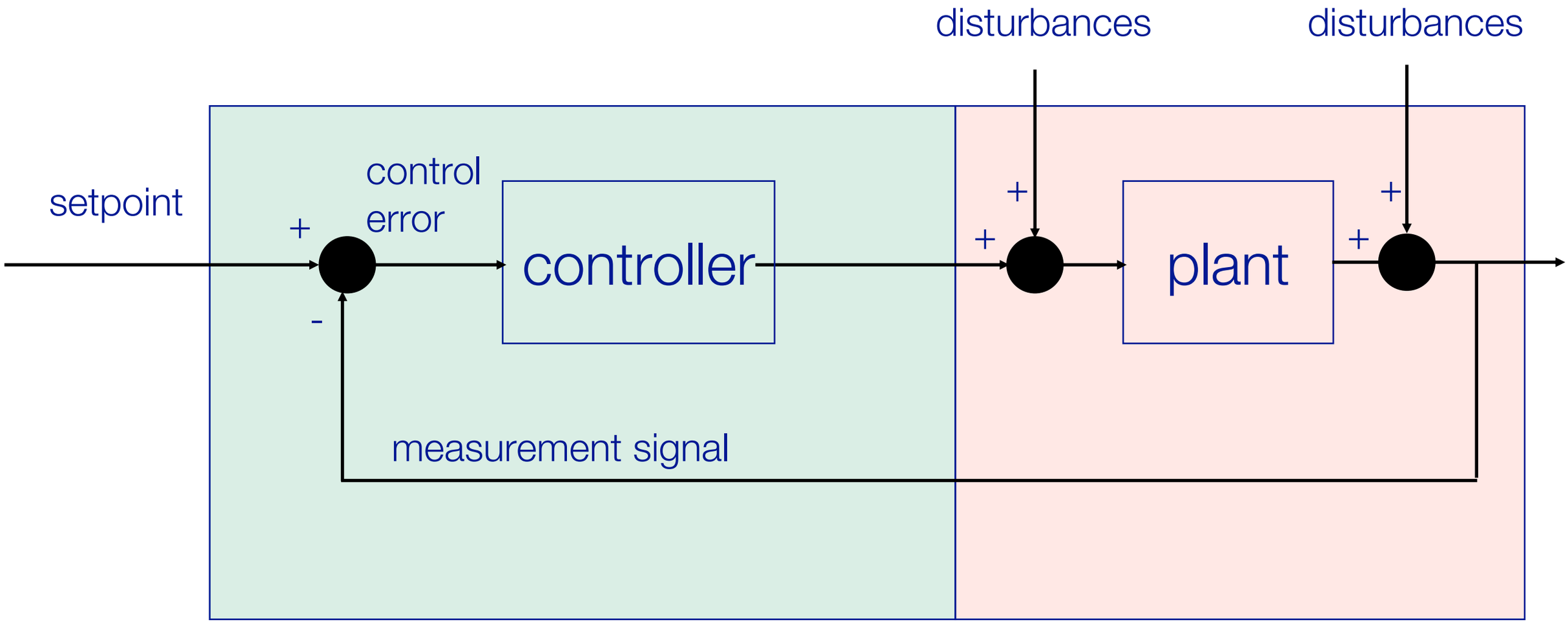


Example output

See http://obc.lbl.gov/specification/example.html for more trajectories and discussion



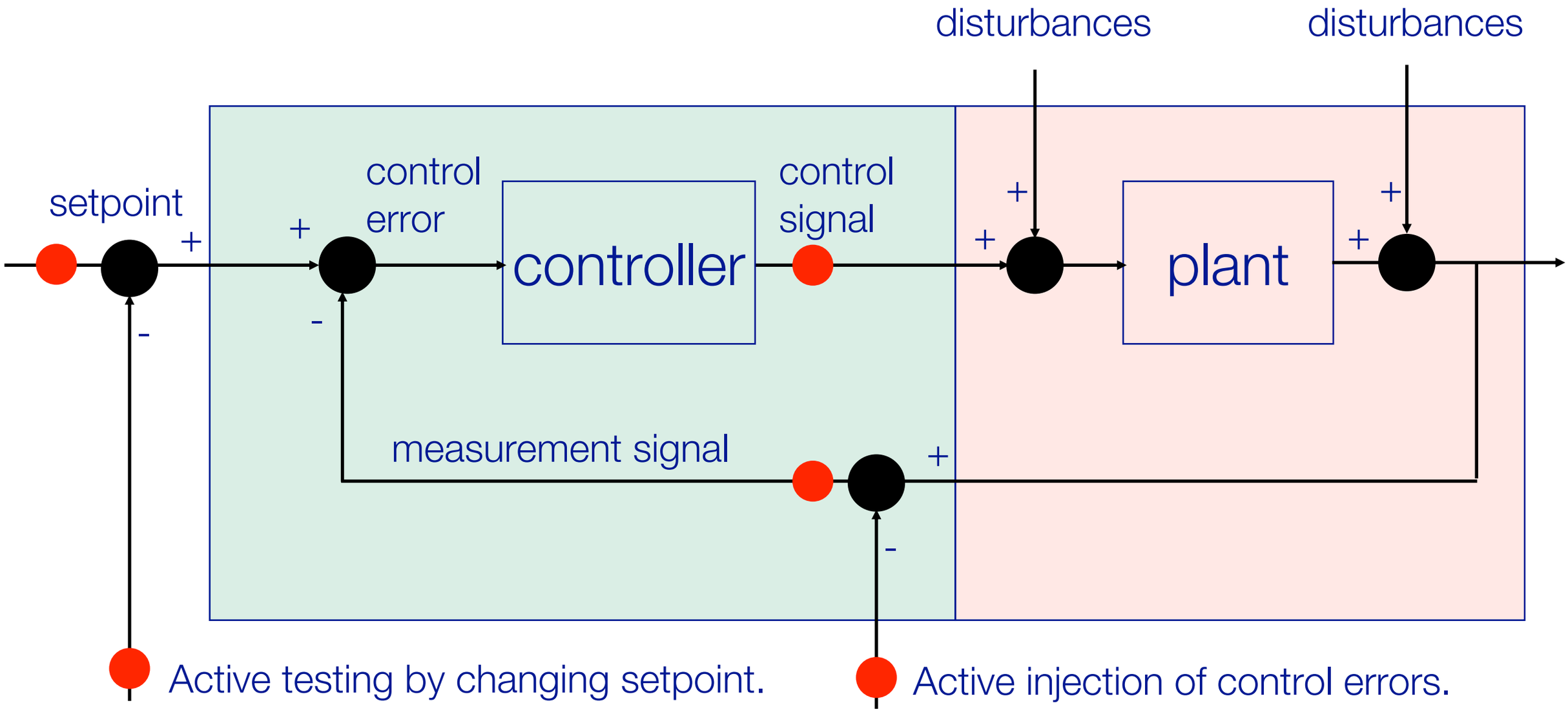
What should be verified?



Disturbances and plant are only approximately known, and hence should be excluded from the verification of the *control delivery*.

But they should be part of an end-to-end verification of the *building delivery*.

How should we verify?



Active testing by changing setpoint.

Active injection of control errors.

- Red points indicate which signals to verify against a CDL generated response.

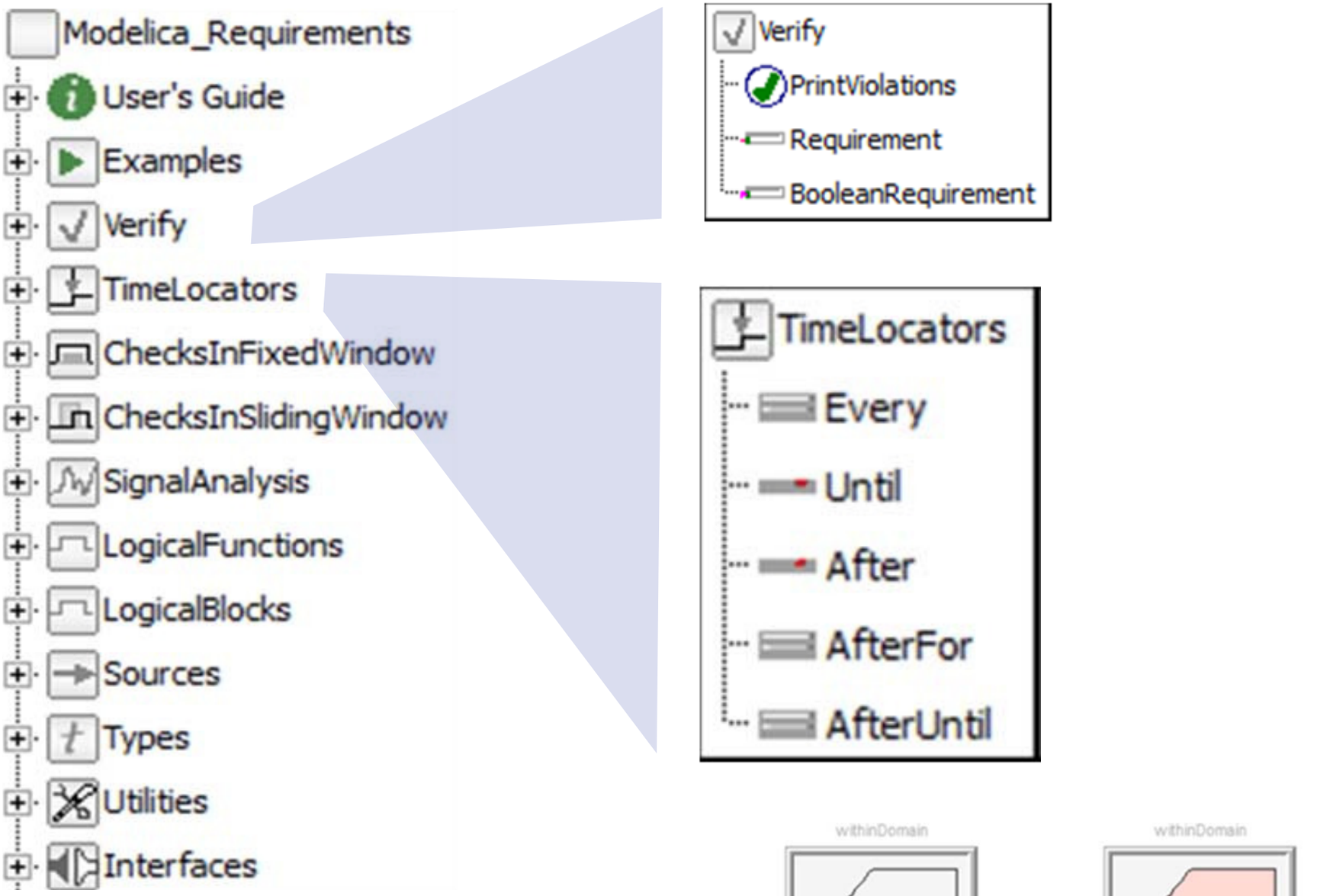
3-valued logic

Violated: Test condition is **violated** at least once.

Untested: Test condition is **undecided** for the complete test period.

Satisfied: Test condition is **satisfied** at least once, and is never violated.

Implementation based on Modelica_Requirements

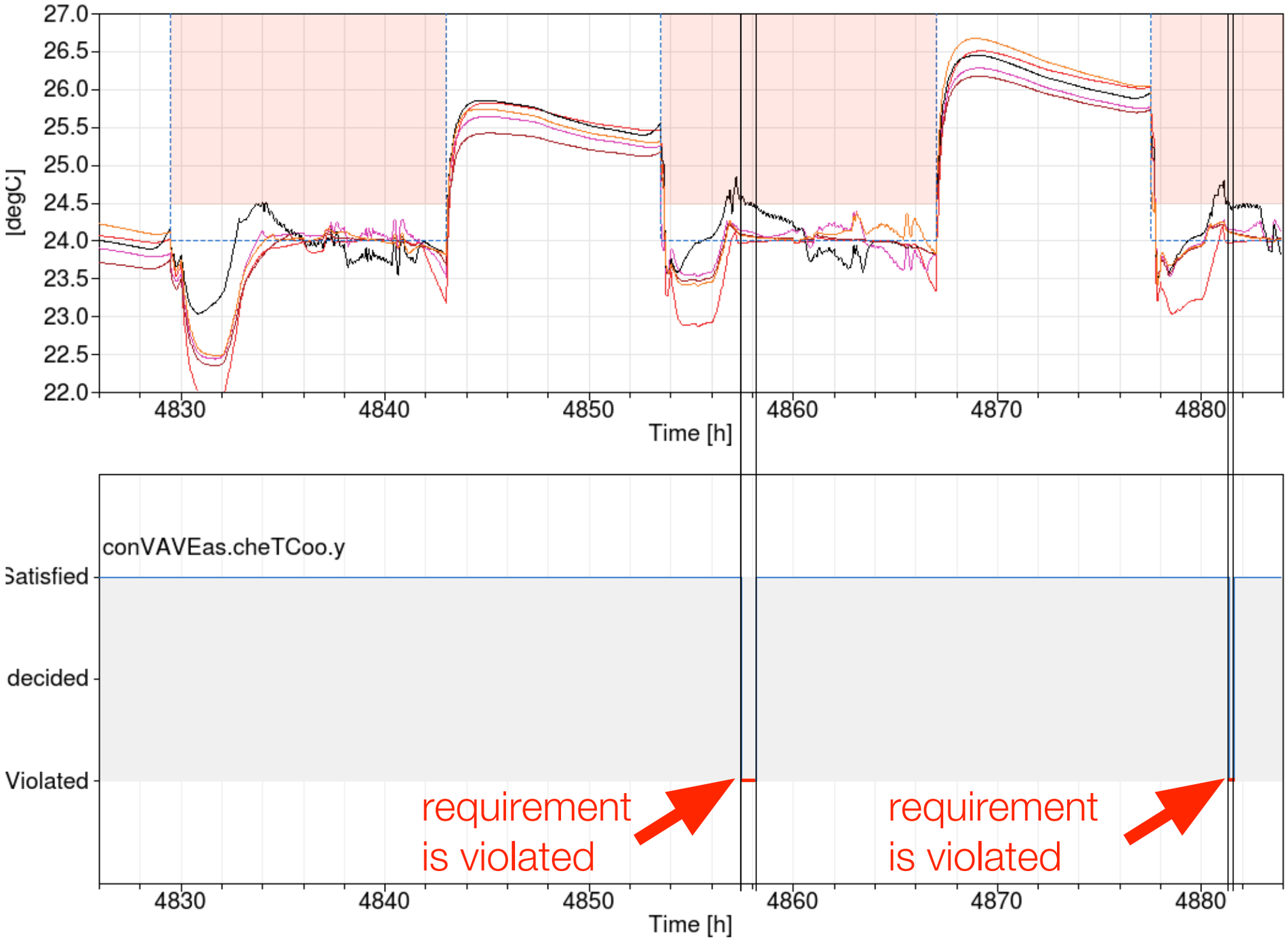


Images from [Otter et al. 2015]

Verification Test

Verification of room air temperature of east zone

Requirement: Room air temperature shall be within (TSet + 0.5 K) for at least 45 min within each 60 min window.

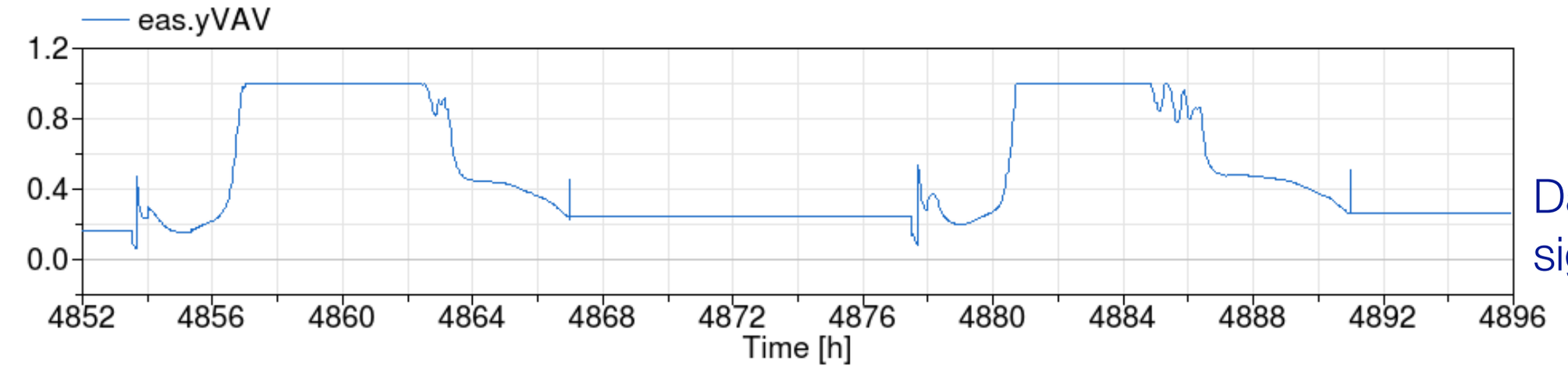


dashed:
cooling set point

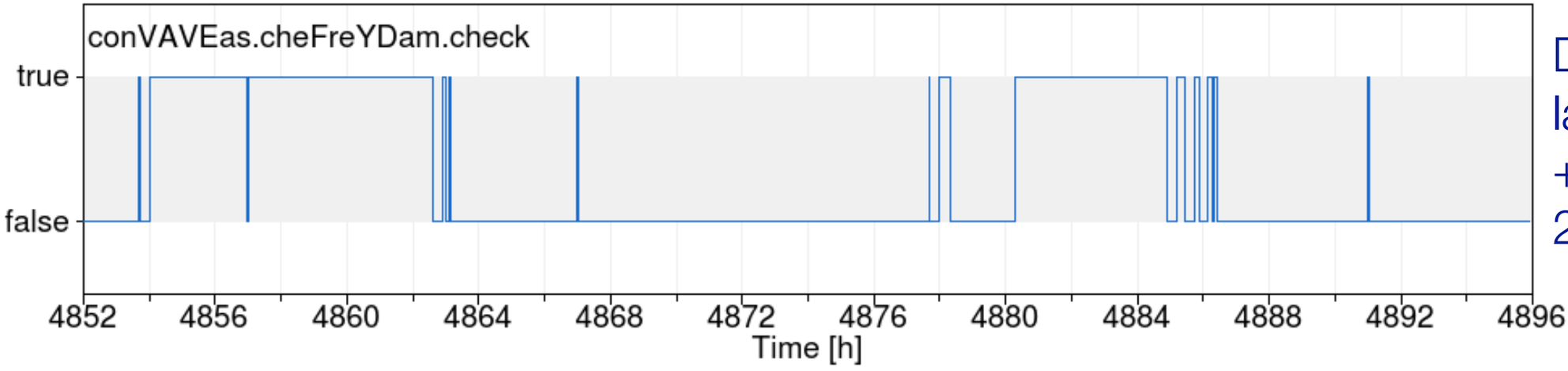
black:
east room
air temperature

Verification of east zone damper signal

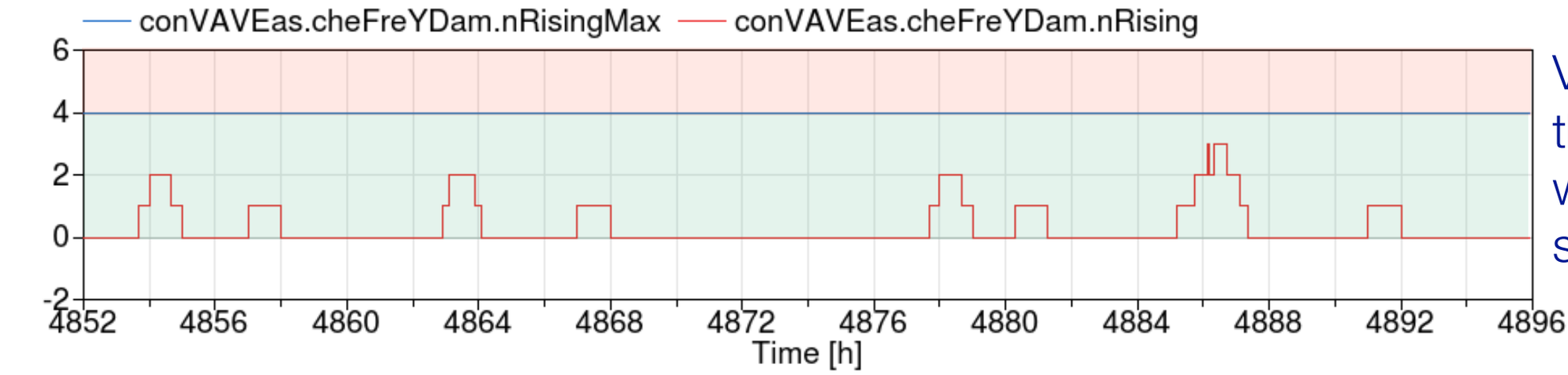
Requirement: Damper signal shall not oscillate more than 4 times per hour between +/- 0.025 (for a 2 minute sample period)



Damper control signal



Detect changes larger than +/- 0.025 between 2 minute samples



Verify no more than 4 changes within 1 hour sliding window

Reports

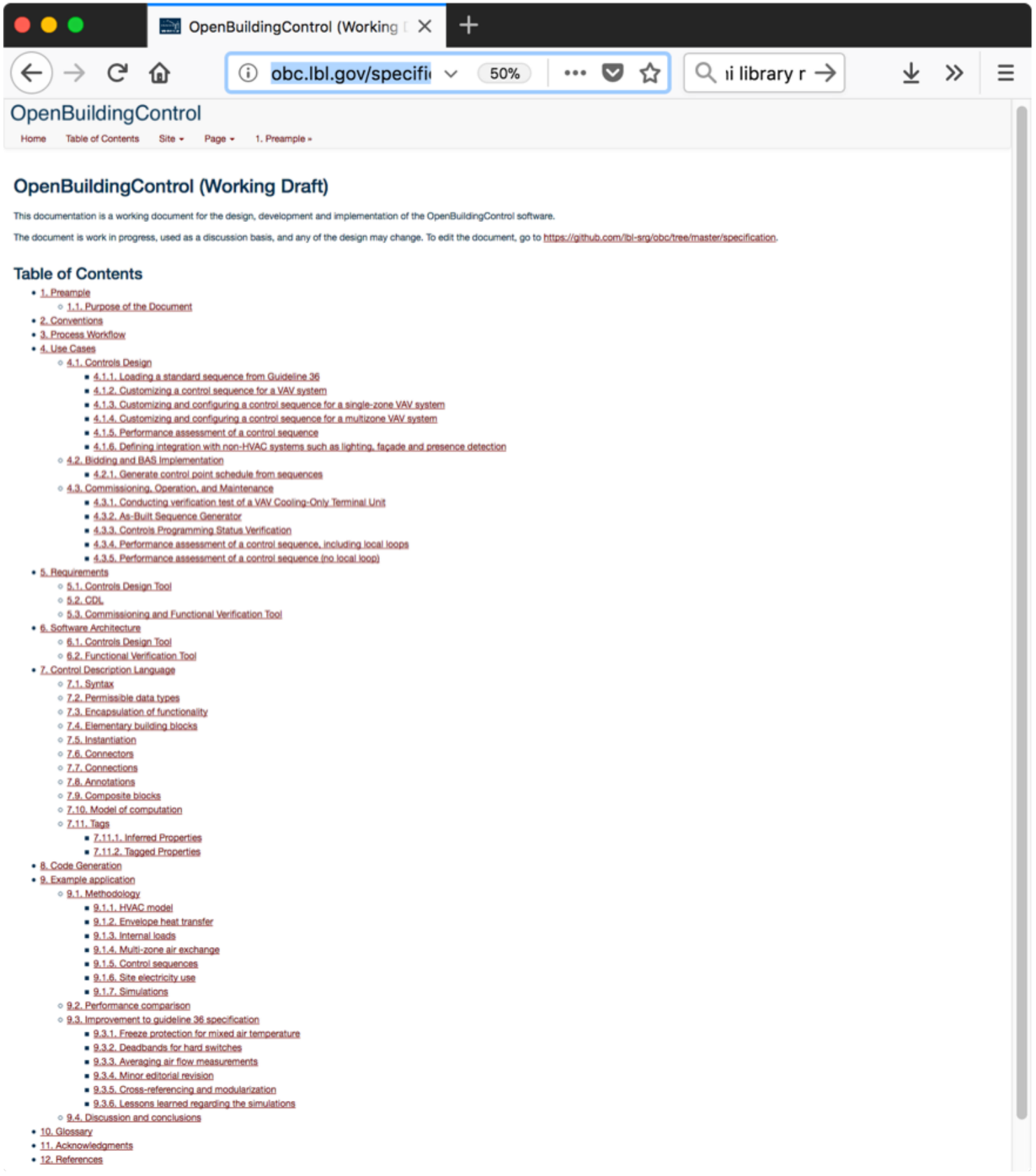
Development requires coordination and compatibility across

- design tool
  - CDL specification
  - CDL library
  - CDL-compliant control sequences
  - CDL-compliant specification for control provider
  - English language representation
- verification tool
  - verification library
  - test specification

To ensure that all elements fit together and that all development is open accessible, reporting with federal agency and stakeholders is done through an integrated document which is continuously updated at:

http://obc.lbl.gov/specification/index.html

Available as html and pdf.

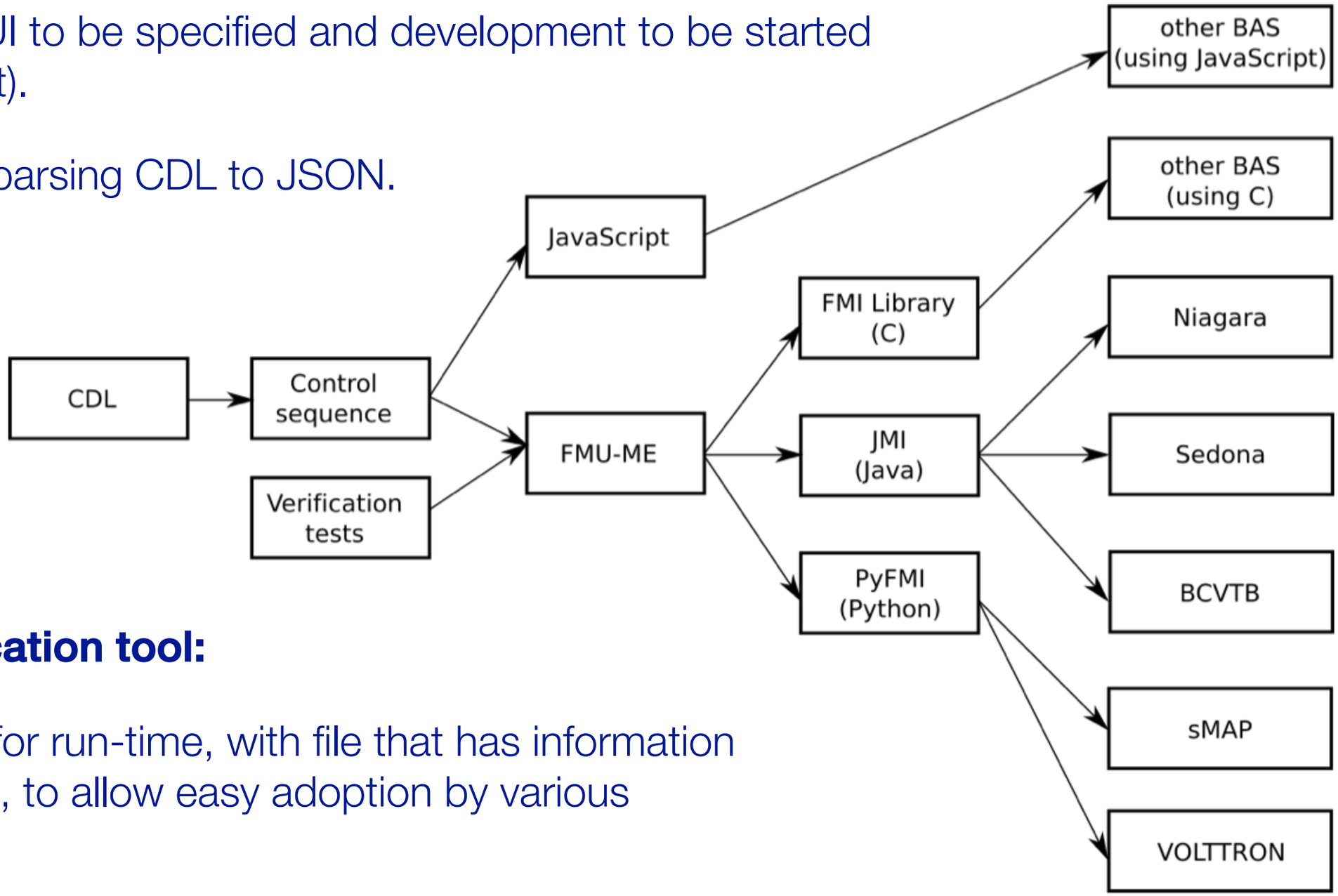


Controls design tool and functional verification tools

Controls design tool:

Schema driven GUI to be specified and development to be started (with SOEP project).

Work ongoing on parsing CDL to JSON.



Functional verification tool:

Use FMI standard for run-time, with file that has information about I/O mapping, to allow easy adoption by various vendors.

http://fmi-standard.org/