OpenBuildingControl
—
2nd TAG meeting

Michael Wetter
Philip Haves
Jianjun Hu
Milica Grahovac

July 19, 2017

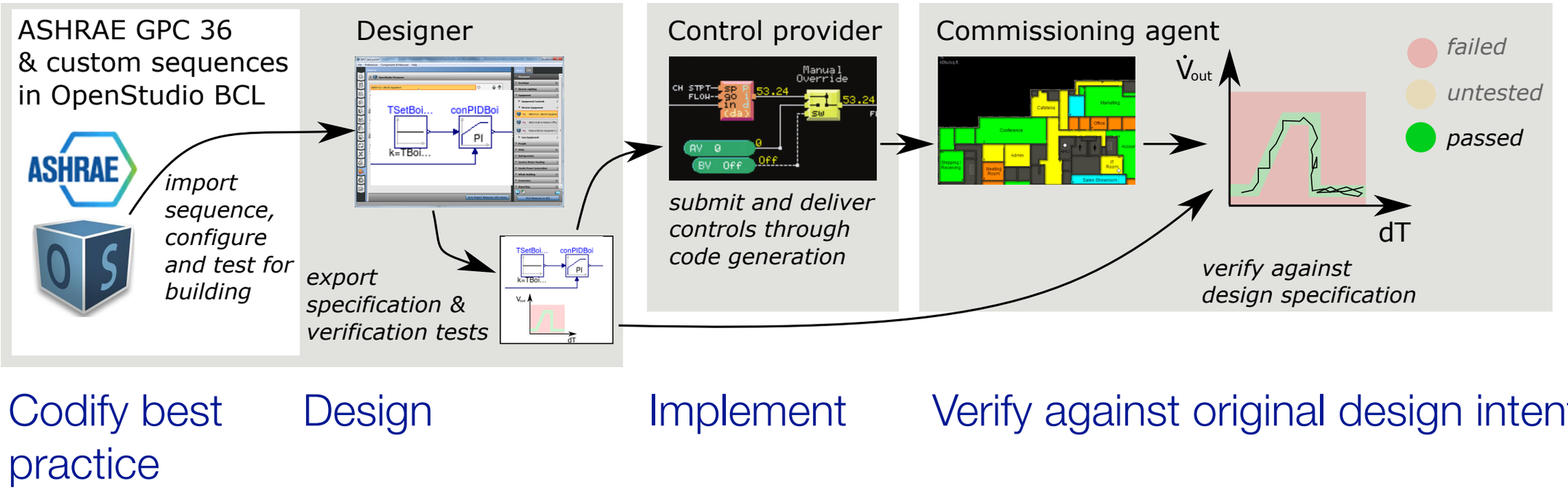
BERKELEY LAB
Lawrence Berkeley National Laboratory

Administrative — CEC cost share

CEC approved full cost-share.

Contract currently in development.

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



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,

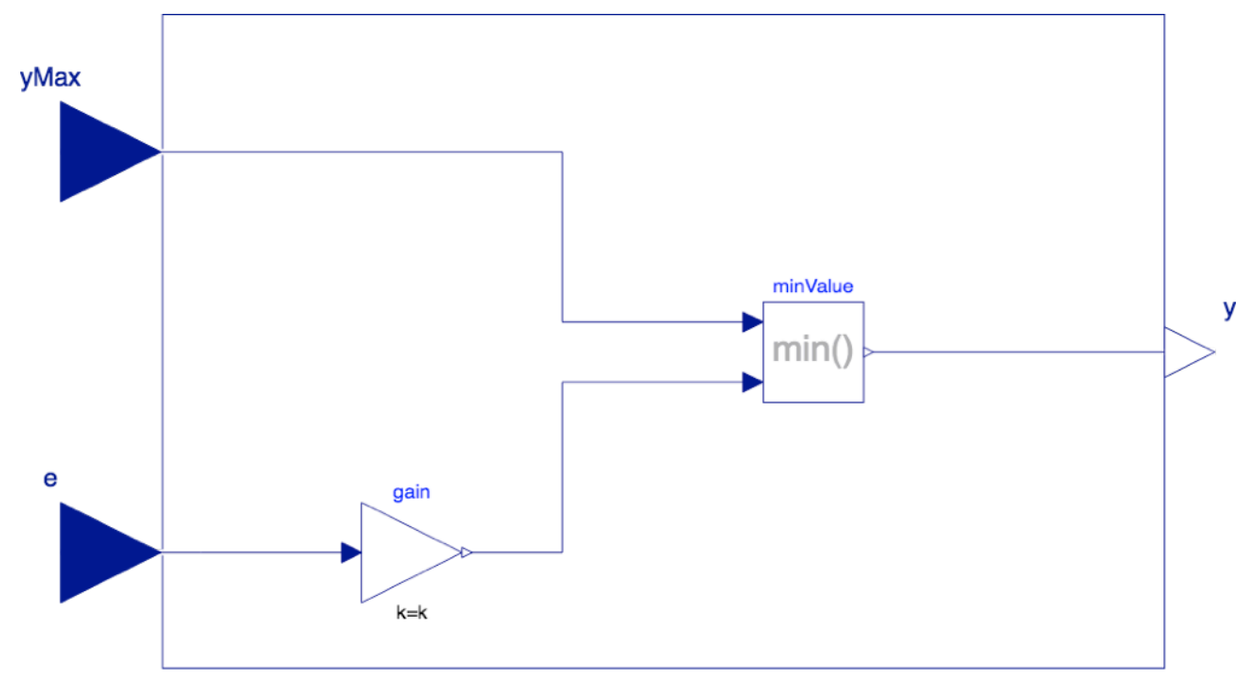
for bidding, automatic implementation and automated functional testing.

Control Description Language

Developed first version of specification for review and further implementation

Proposed

- Syntax
- Permissible data types
- Encapsulation of functionality
- Instantiation
- Connectors
- Connections
- Annotations
- Composite blocks
- Tags
- Model of computations

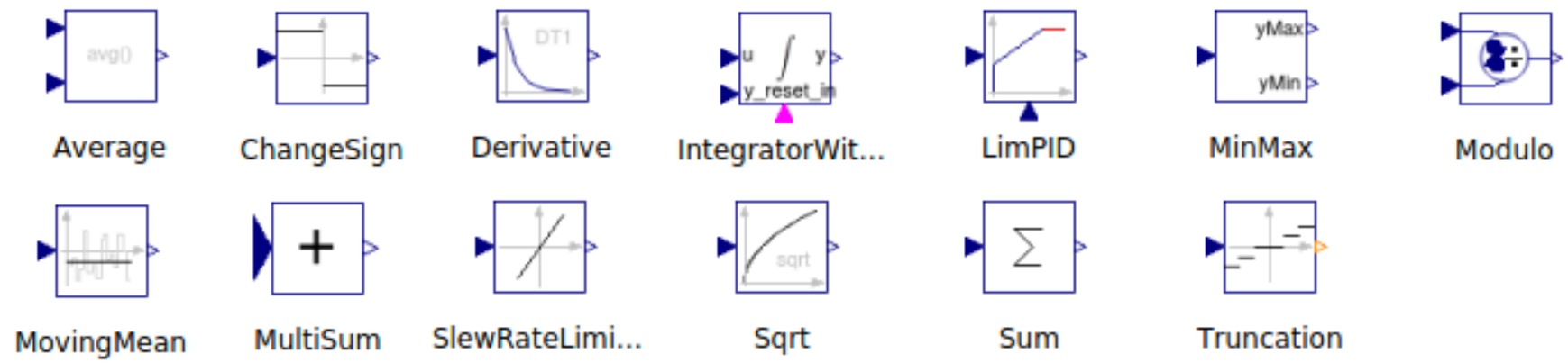


See specification for details: http://obc.lbl.gov/specification/cdl.html

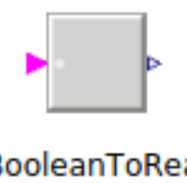
New blocks in CDL library

> 20 new blocks

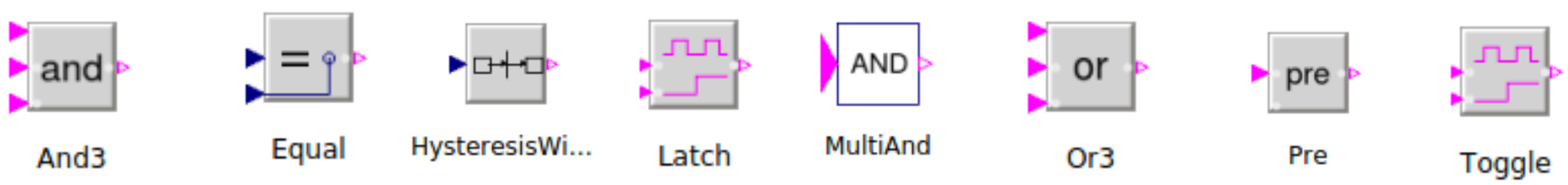
- Continuous



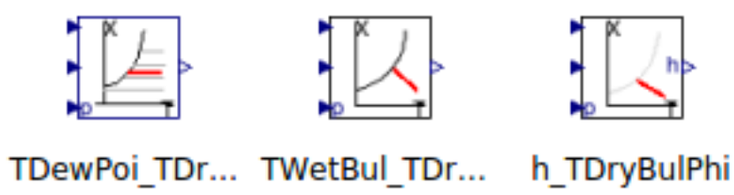
- Conversions



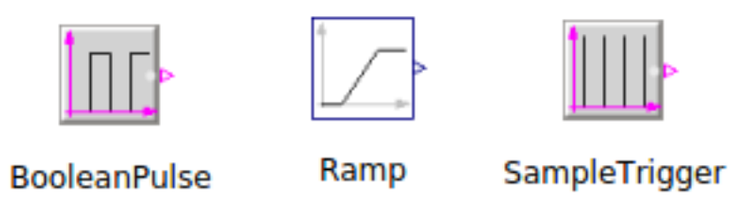
- Logical



- Psychrometric



- Sources



Browse CDL library at

http://obc.lbl.gov/specification/cdl/latest/help/CDL.html

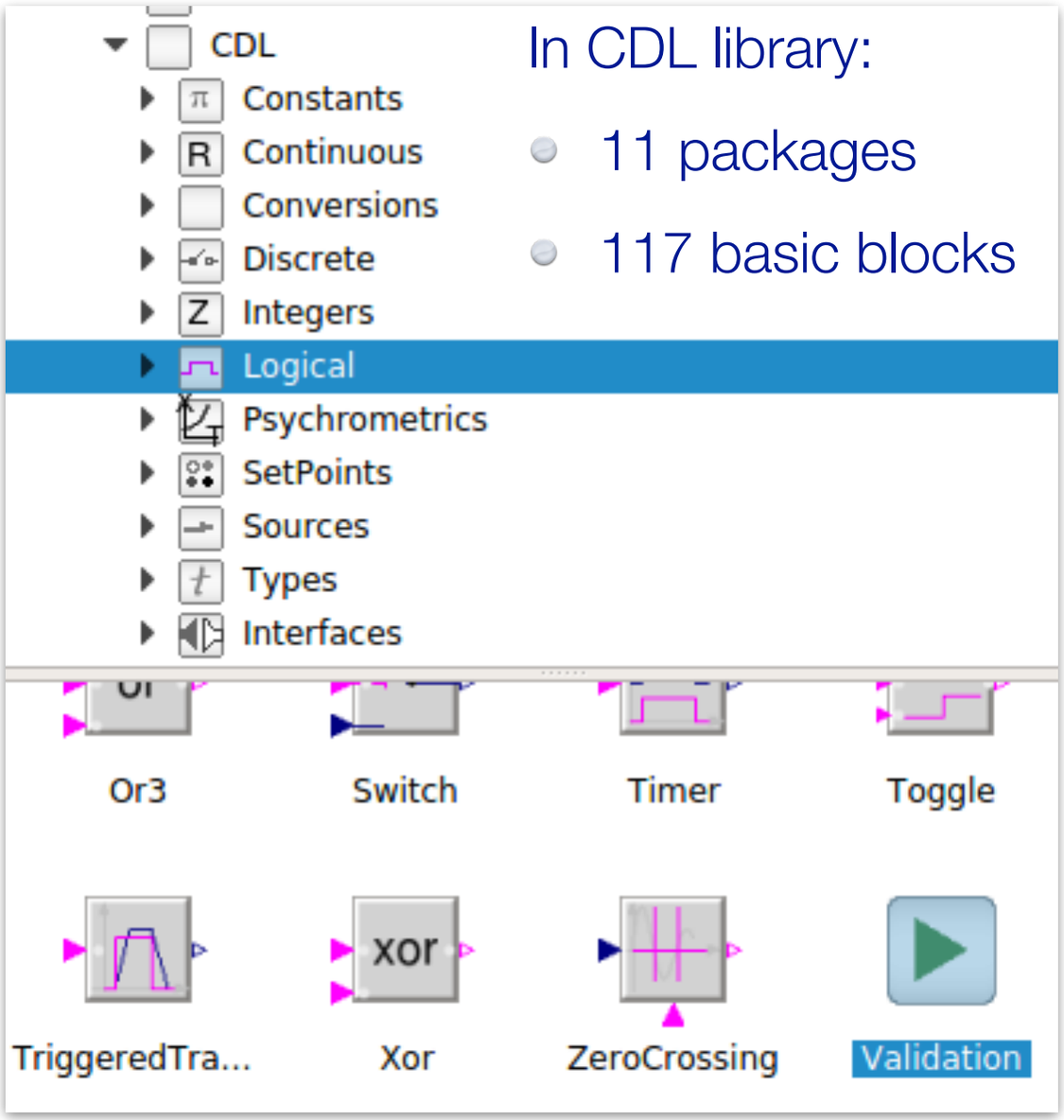
CDL library

Compared CDL library with industrial control library.

Added basic blocks as needed.

Validated blocks to ensure expected functionalities.

Convening group of vendor to review the basic blocks.



In CDL library:

- 11 packages
- 117 basic blocks

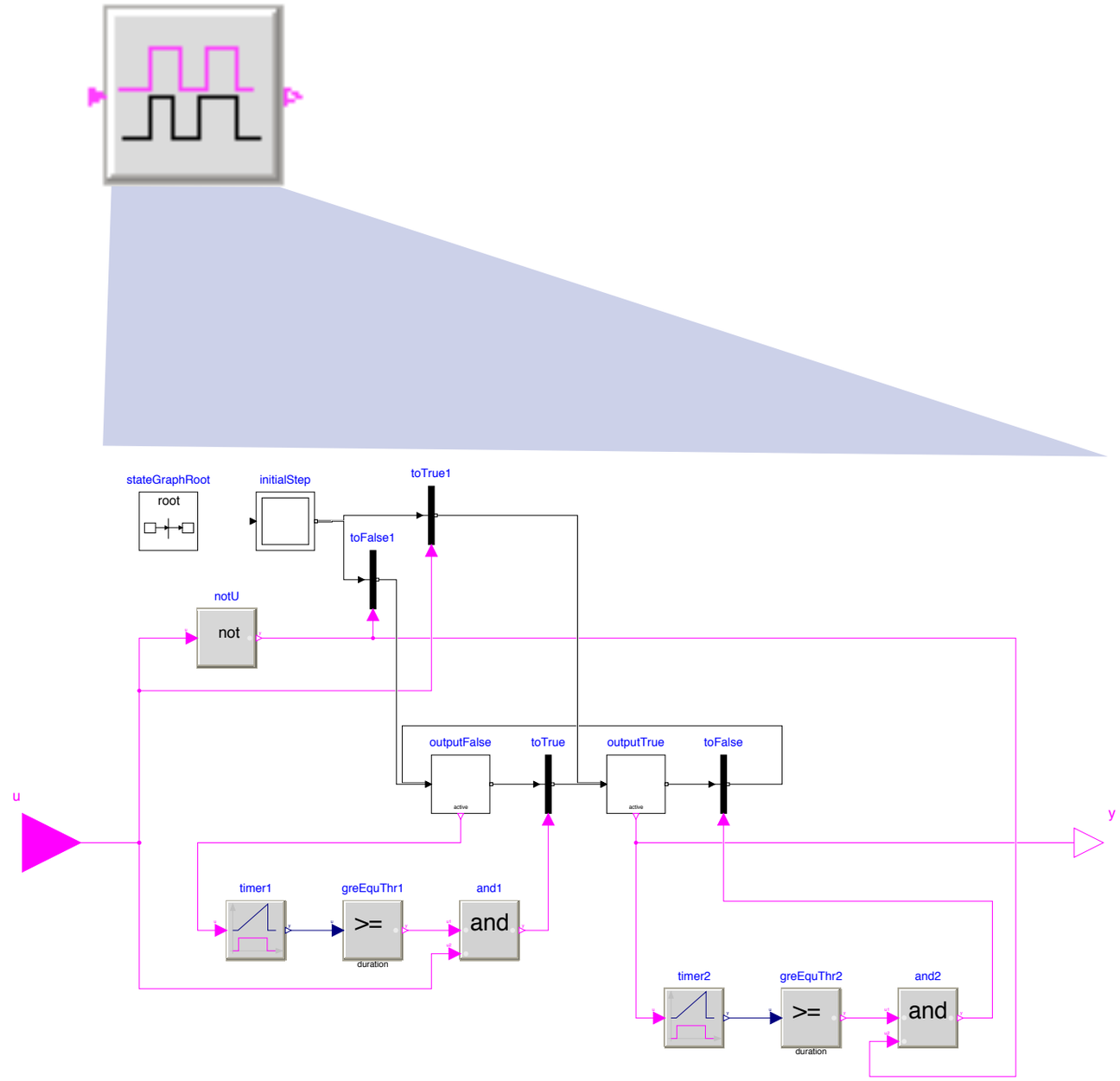
Block that holds an output signal for a minimum time

Block that holds a *true* or *false* signal for at least a defined time period.

Whenever the input *u* switches, the output *y*
switches and remains at that value for at least *duration*
seconds, where *duration* is a parameter.

After *duration* elapsed, the output will be $y = u$.

If this change required changing the value of *y*, then *y* will remain at that value for at least *duration*. Otherwise, *y* will change immediately whenever *u* changes.



http://obc.lbl.gov/specification/cdl/latest/help/CDL_Logical.html#CDL.Logical.TrueFalseHold

Questions to TAG about the CDL

1. Should we allow conditional removal of blocks and connectors?

   `OperationModeSelection` has inputs such as heating set point. How should we deal with controls of systems that have no heating, such as in Miami. Should

   - the inputs be left, and users asked to provide a value (such as 0 degC for heating set point if there is no heating),
   - should the inputs be removed based on a boolean parameter such as `haveHeating`, or
   - should we have three separate sequences,
     one with heating and cooling,
     one with cooling only, and
     one with heating only.

2. How do control blocks such as for optimal start-up retrieve a signal for when the building switches next from unoccupied to occupied mode?

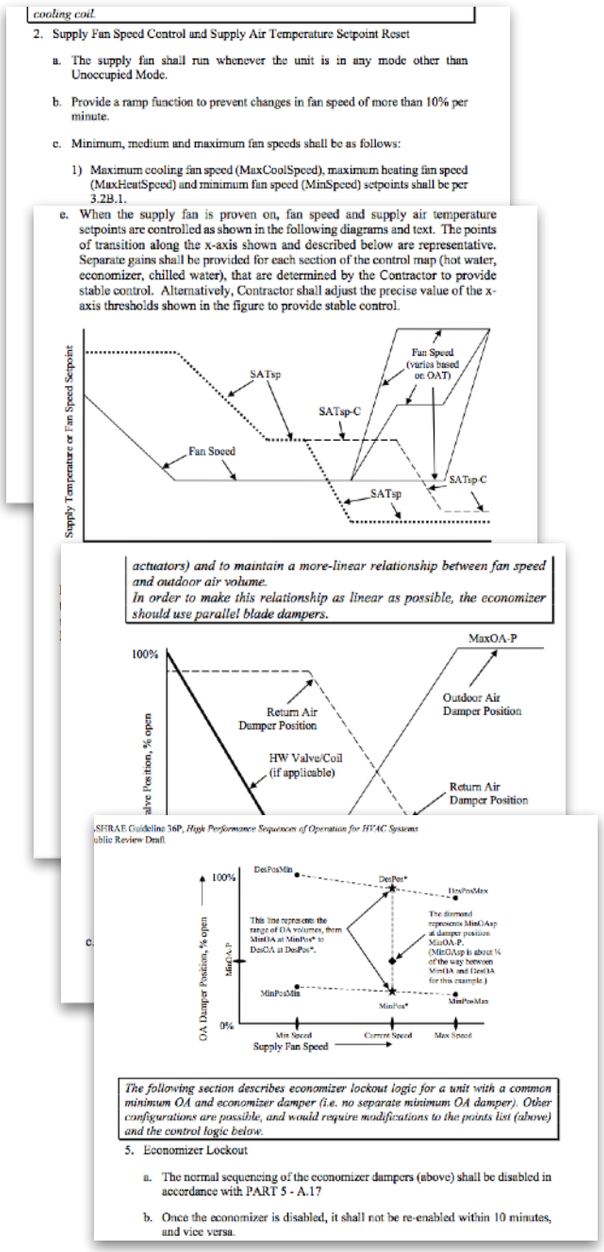
3. Why does ALC constrain the output accuracy to values such as 0.1, 0.01, 0.001?

4. Obtained feedback from ALC, but need feedback from other control providers by September 15 for CDL specification and library of basic blocks.

Sequence Specification

Implement atomic sequences with CDL

ASHRAE Guideline 36



cooling coil.

2. Supply Fan Speed Control and Supply Air Temperature Setpoint Reset

   a. The supply fan shall run whenever the unit is in any mode other than Unoccupied Mode.

   b. Provide a ramp function to prevent changes in fan speed of more than 10% per minute.

   c. Minimum, medium and maximum fan speeds shall be as follows:

      1) Maximum cooling fan speed (MaxCoolSpeed), maximum heating fan speed (MaxHeatSpeed) and minimum fan speed (MinSpeed) setpoints shall be per 3.2B.1.

   e. When the supply fan is proven on, fan speed and supply air temperature setpoints are controlled as shown in the following diagrams and text. The points of transition along the x-axis shown and described below are representative. Separate gains shall be provided for each section of the control map (hot water, economizer, chilled water), that are determined by the Contractor to provide stable control. Alternatively, Contractor shall adjust the precise value of the x-axis thresholds shown in the figure to provide stable control.

actuators) and to maintain a more-linear relationship between fan speed and outdoor air volume.
In order to make this relationship as linear as possible, the economizer should use parallel blade dampers.

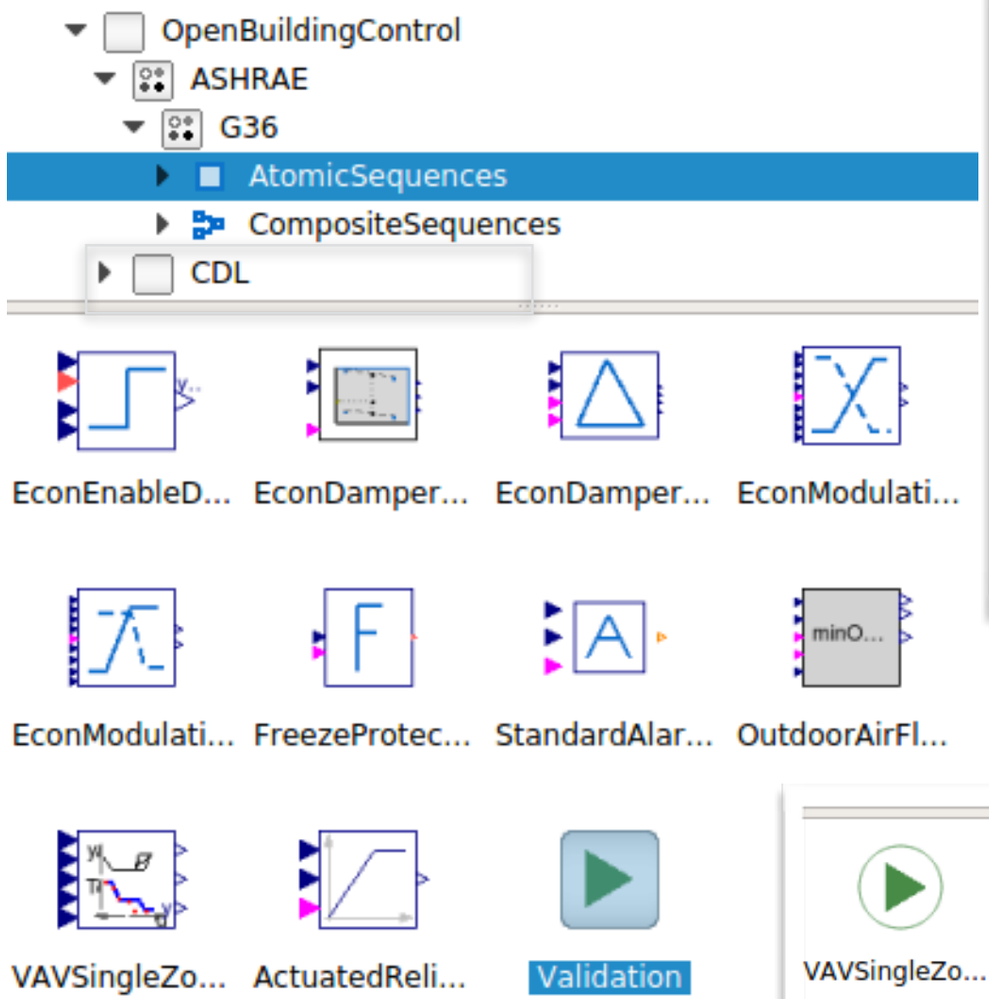
The following section describes economizer lockout logic for a unit with a common minimum OA and economizer damper (i.e. no separate minimum OA damper). Other configurations are possible, and would require modifications to the points list (above) and the control logic below.

5. Economizer Lockout

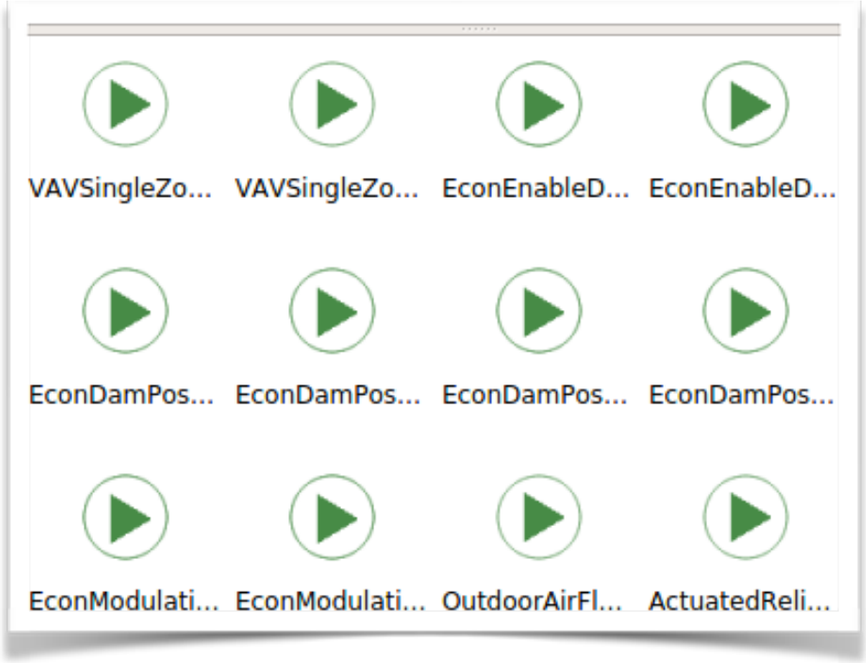
   a. The normal sequencing of the economizer dampers (above) shall be disabled in accordance with PART 5 - A.17

   b. Once the economizer is disabled, it shall not be re-enabled within 10 minutes, and vice versa.

Implementation using CDL library



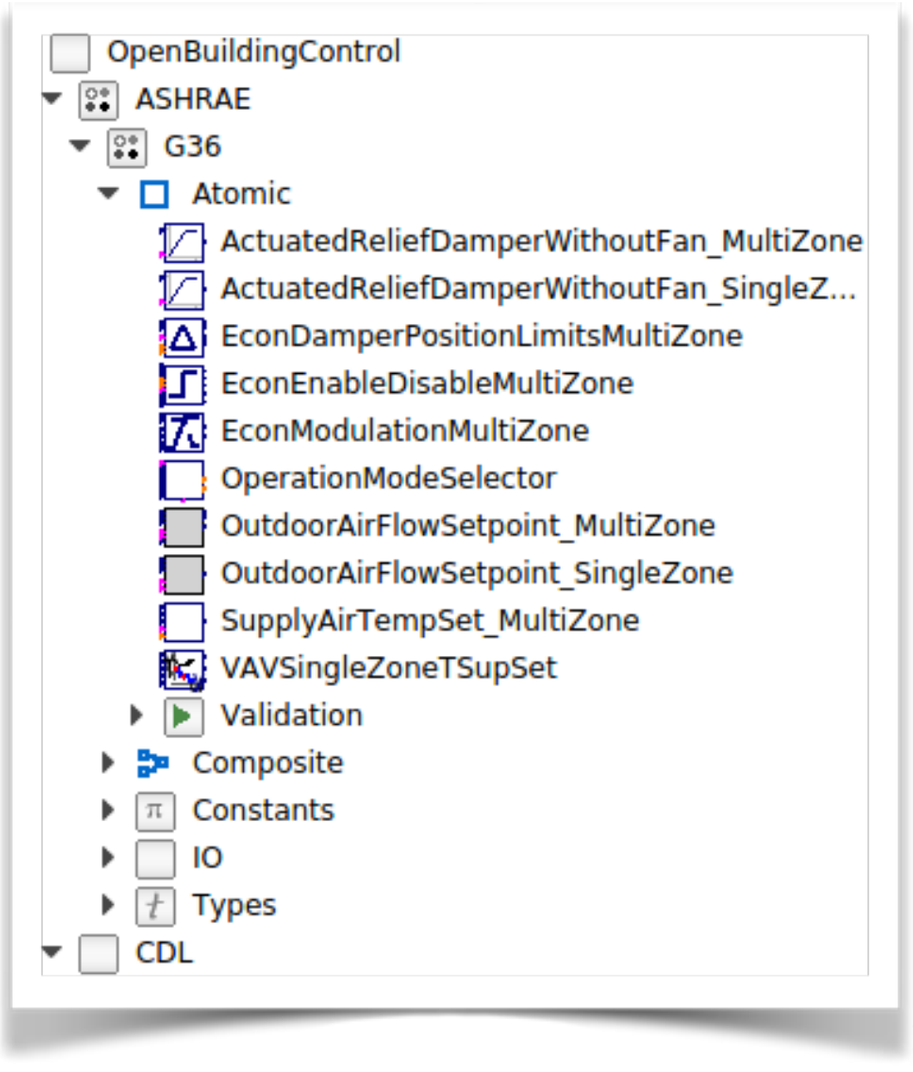

```
CDL.Interfaces.BooleanInput uSupFan "Supply Fan Status, c
  a;
CDL.Interfaces.RealOutput yOutDamPos(min=0, max=1, unit="
                                                      a;
CDL.Interfaces.RealOutput yRetDamPos(min=0, max=1, unit="
                                                      a;
CDL.Continuous.Line outDamPos(limitBelow=true, limitAbove
  "Damper position is linearly proportional to the contrc
  a;
CDL.Continuous.Line RetDamPos(limitBelow=true, limitAbove
  "Damper position is linearly proportional to the contrc
  a;
CDL.Continuous.Constant minSignalLimit(k=damPosController
  "Identical to controller parameter - Lower limit of out
  a;
CDL.Continuous.Constant maxSignalLimit(k=damPosController
  "Identical to controller parameter - Upper limit of out
  a;
CDL.Interfaces.RealInput uHea(min=0, max=1, unit="1")
  "Heating control signal."
  a;
CDL.Interfaces.RealInput uCoo(min=0, max=1, unit="1")
  "Cooling control signal."
  a;
CDL.Interfaces.RealInput uOutDamPosMin(min=0, max=1, unit
  "Minimum economizer damper position limit as returned b
  a;
CDL.Interfaces.RealInput uOutDamPosMax(min=0, max=1, unit
  "Maximum economizer damper position limit as returned b
  a;
```



ASHRAE G36 Sequences Library - Status

Sequences currently implemented or in the late development/ review stage:

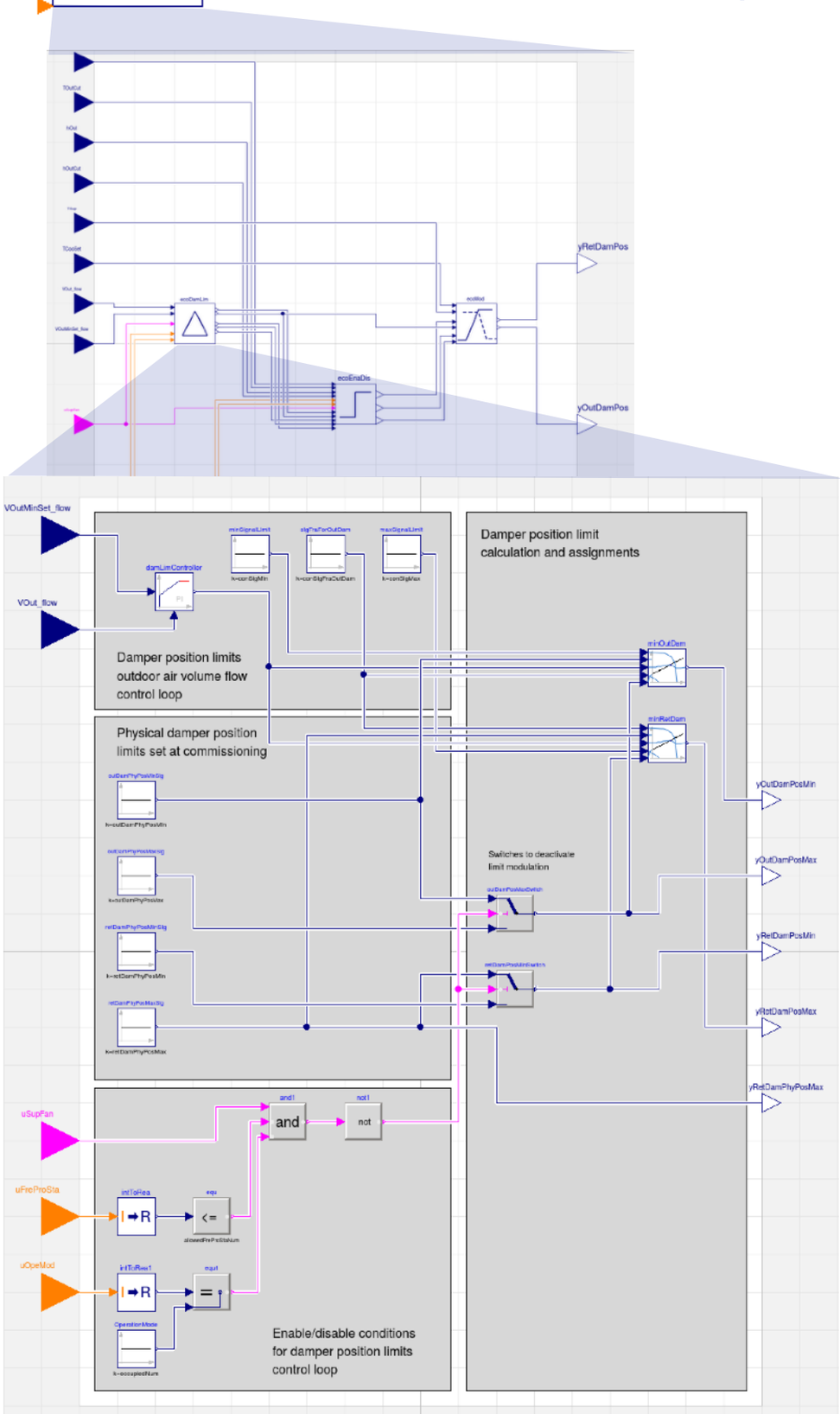
- Single zone VAV AHU Temperature and Fan Speed Set Points;
- VAV AHU Economizer Sequences — Single and Multiple zones, including:
  - High limit lockout and economizer enable/disable
  - Damper position limits for outdoor and return air dampers to satisfy minimum outdoor air requirement
  - Outdoor and return air damper modulation in economizer mode
- System operation modes selector;
- Minimum outdoor airflow setpoint — Single and Multiple zones;
- Actuated relief damper without fan — Single and Multiple zones;

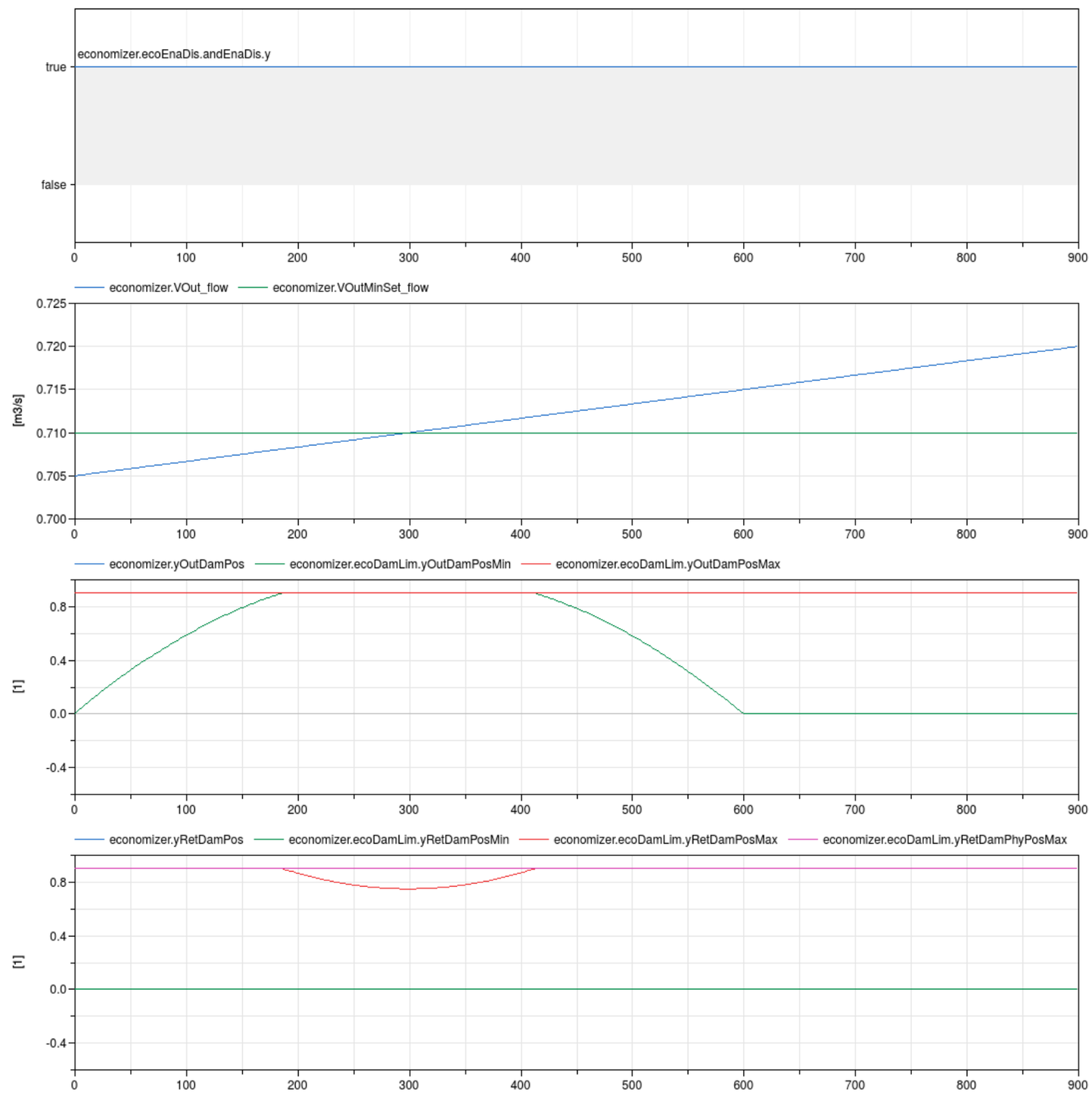


Guideline G36 sequences in the early development stage:

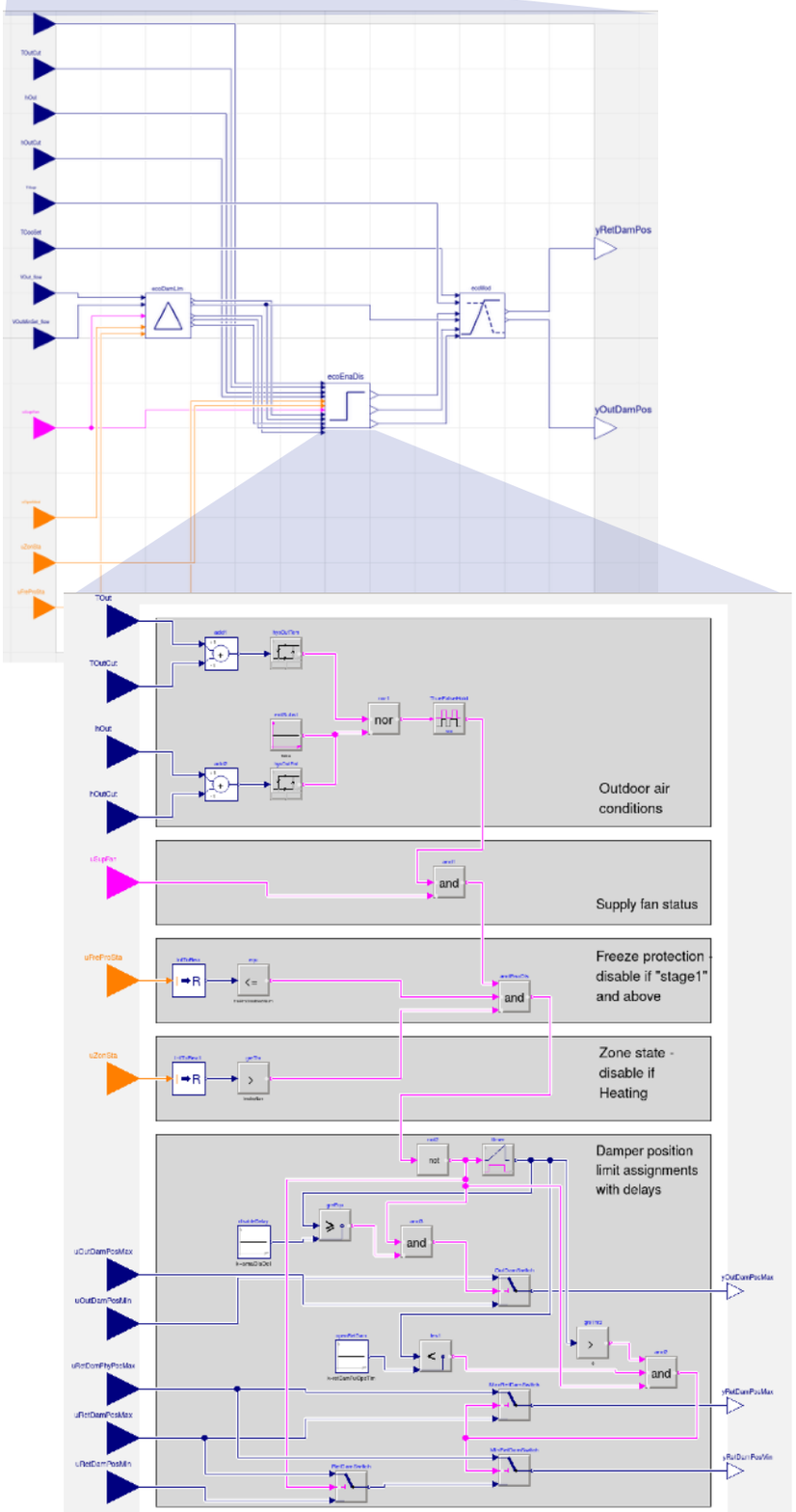
- Freeze Protection;
- Standard Alarms;
- Multizone VAV AHU supply air temperature set point;
- Relief fan control;
- Heating and cooling valve control loops

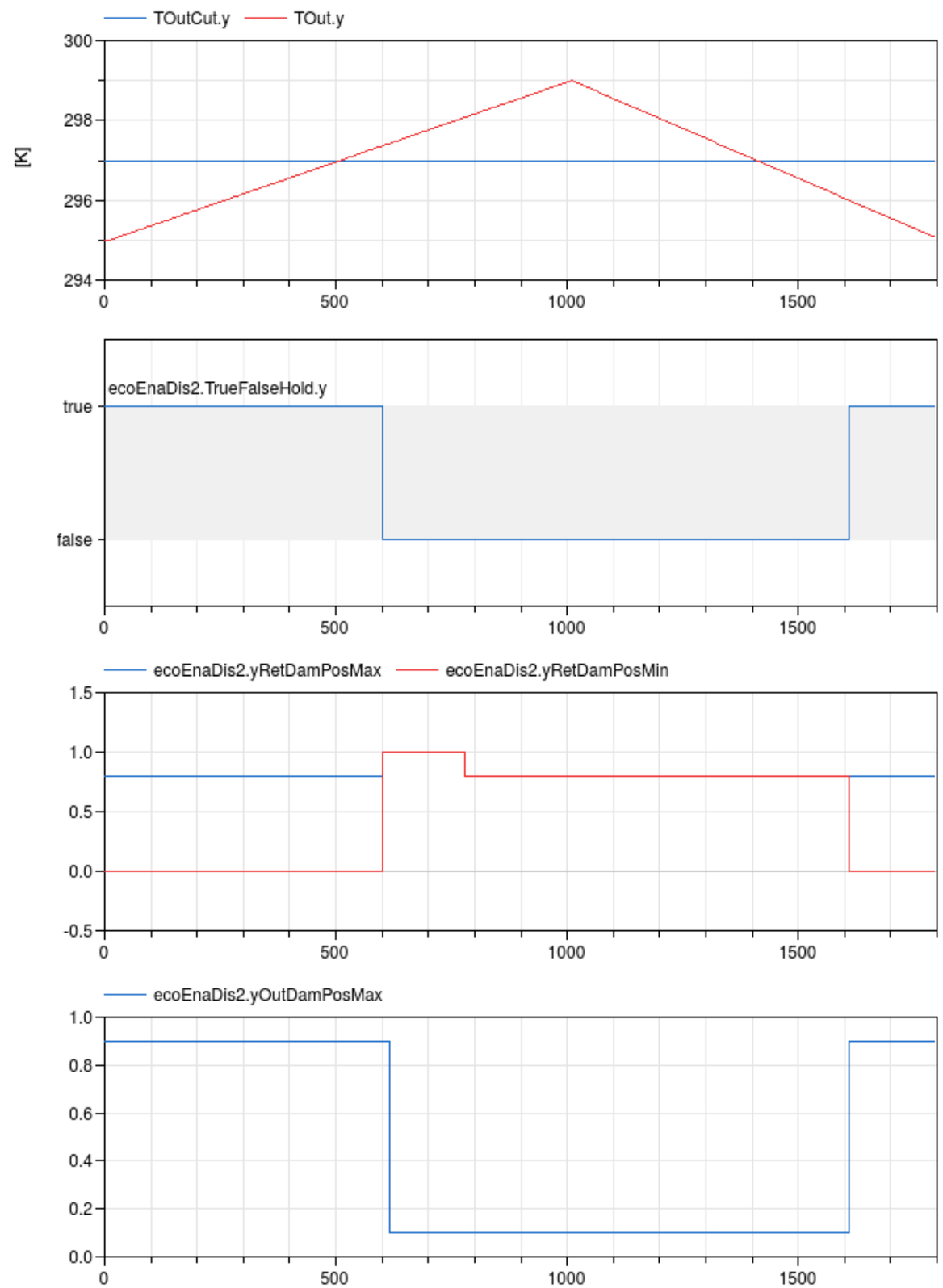
G36 Library Example: Multiple Zone VAV AHU Economizer OA and RA Damper Limits for Minimal Outdoor Air Control



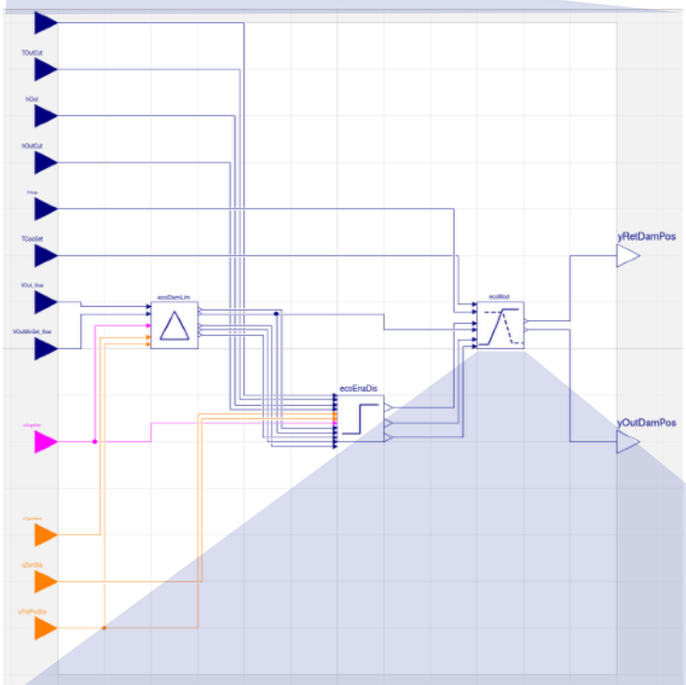


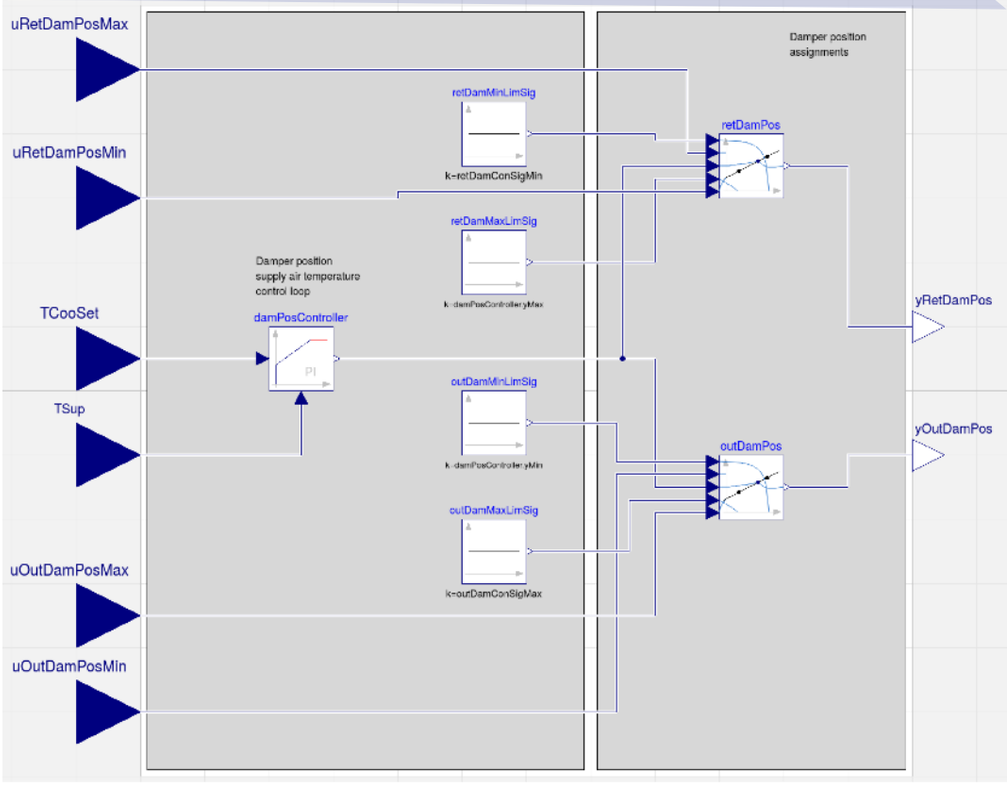
G36 Library Example: Multiple Zone VAV AHU Economizer Economizer Enable/Disable

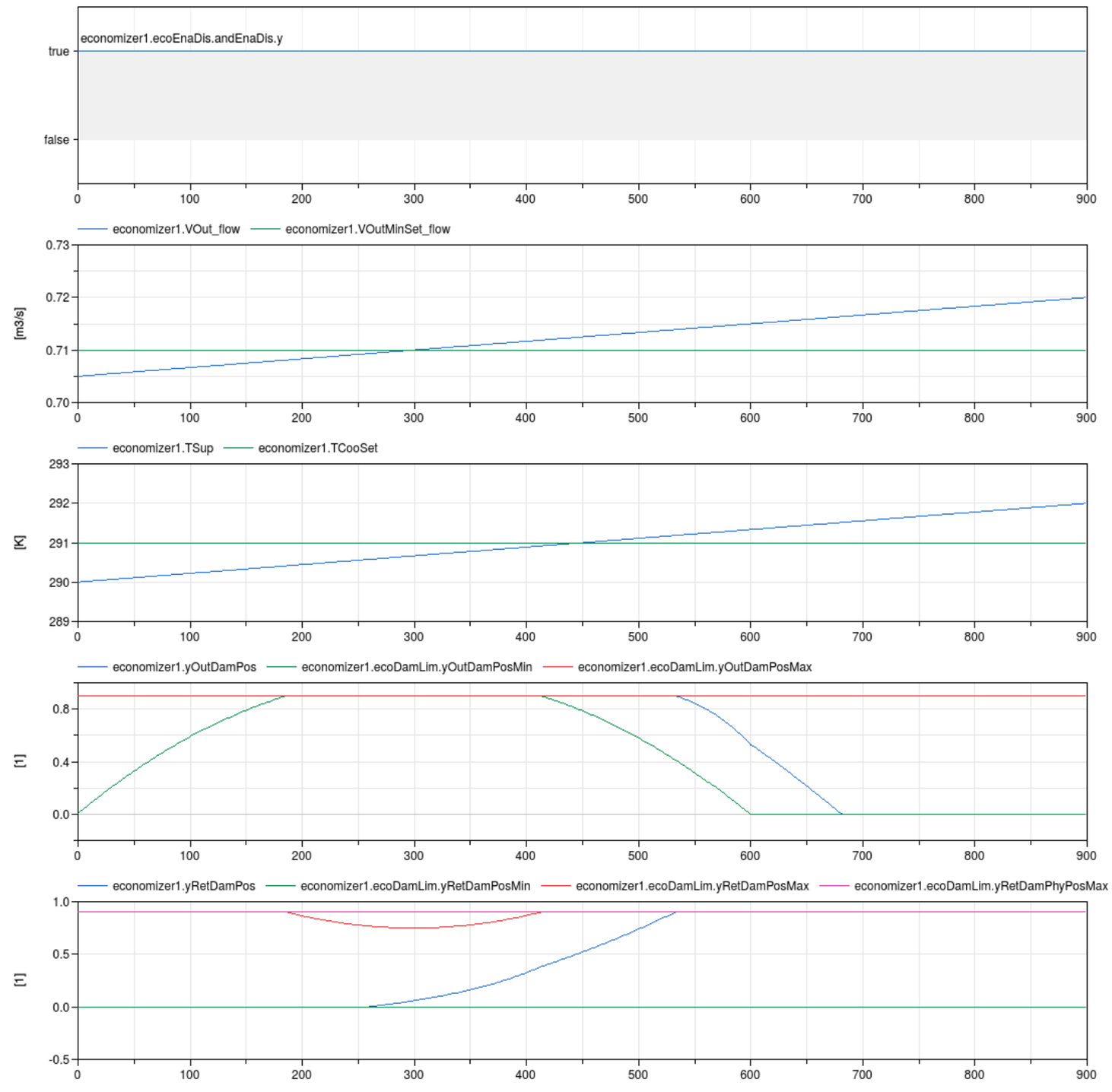




G36 Library Example: Multiple Zone VAV AHU Economizer OA and RA Damper Modulation







Questions to TAG about the sequences

1. Is our current structure (Atomic sequence, Composite sequence) a good way to implement the sequences?

2. Is there any better way to form structure of the sequences library?

Upcoming deadline:

By Q4, release a version of the control library for secondary systems in Modelica.

Tagging

CDL syntax and structure allow the following tags based on the Modelica syntax:

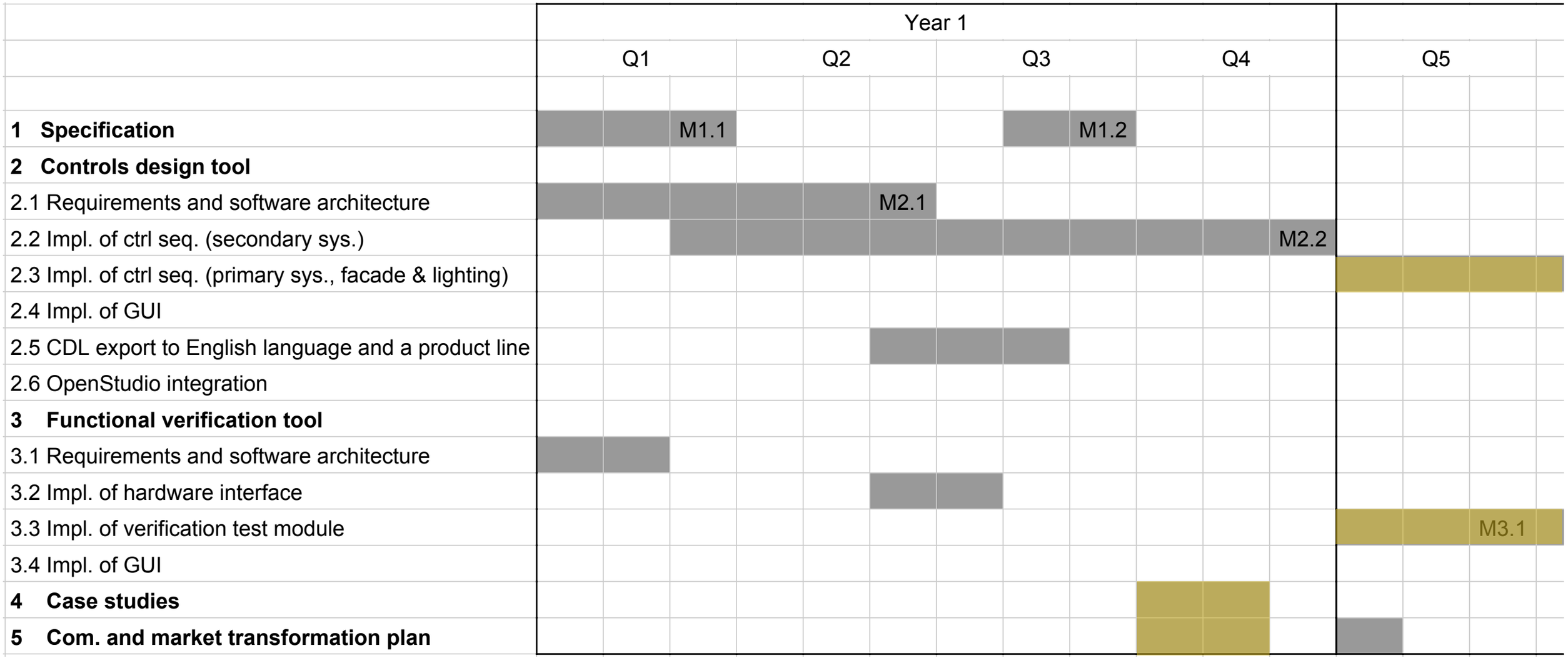
- Numerical value:
  - Binary, Analog, Mode/Status
  - Example: `CDL.Interfaces.RealInput` represents an analog value
- Source:
  - Hardware, Software
  - Example: `CDL.Sources.{Hardware|Software}.`
- Quantity and Unit:
  - Temperature, Pressure, Humidity, Speed or Command/Request/Status
  - Example: `CDL.Interfaces.RealInput TSetZon(unit="K", displayUnit="degC")`

CDL enables implementation of external tagging schemes, such as Brick (http://brickschema.org) and Haystack(http://project-haystack.org/) through vendor annotations. These tags do not influence with the functionality of the tool.

The vendor annotations syntax:

```
annotation :
  annotation "(" [annotations ","]
     __cdl (" [ __cdl_annotation ] ")" [","  annotations] ")"
```

Milestone and progress



See https://github.com/lbl-srg/obc/wiki/2017-07-tag-next-steps for upcoming deadlines