IBPSA Project 2
Task 2 and 3 Virtual Progress Meeting

Updates from LBNL

06/25/2026

David Blum

Ettore Zanetti

IBPSA Project 2

Updating Modelica Libraries and Compilation of Test Cases

Background

- Test cases originally developed using MSL v3.2.3, compatible versions of IBPSA, Buildings, and IDEAS Modelica librariers, and compiled using JModelica
- Since JModelica stopped being supported (2020), and MSL v4 is non-backwards compatible with MSL v3:
  - 7 of 9 test cases have been stuck at MSL v3, old Buildings (v8.x) and IDEAS
  - We've had no open-source, license-free FMU compiler option to update them
- OpenModelica is an open-source, license-free compiler option, but with limited capabilities for CS FMU export
- Dymola exports FMUs with license requirements, unless have binary model export add-on
- Since years have passed since test cases have been updated to recent Modelica libraries, significant work is required to update the models and test them

→ But at this point something needs to be done to keep up with Modelica ecosystem…

Updating Modelica Libraries and Compilation of Test Cases

PR #799: "Update test cases to Modelica 4.0 and compilation toolchain"

- Update all test cases to MSL v4, Buildings 12.1.0, and IDEAS 4.0, except `multizone_office_complex_air` and `multizone_office_simple_hydronic` (which are already MSL v4)
- Compile all repository's public test case FMUs with Dymola with binary model export option, except `testcase1`, which is the only FMU that can be successfully compiled with OpenModelica and simulated with pyfmi within BOPTEST
- Unit testing Dockerfile: Remove support for JModelica and add support for OpenModelica, and build from worker container to ensure same environment
- Parser: Remove support for JModelica and update support for Dymola, OpenModelica, and Optimica, including new CLI arguments for tool, algorithm (solver), and tolerance.
- Update the worker container from Ubuntu 20 to Ubuntu 24
- Some other changes to test cases that came up during the update
- Benchmark baseline KPIs and time to simulate

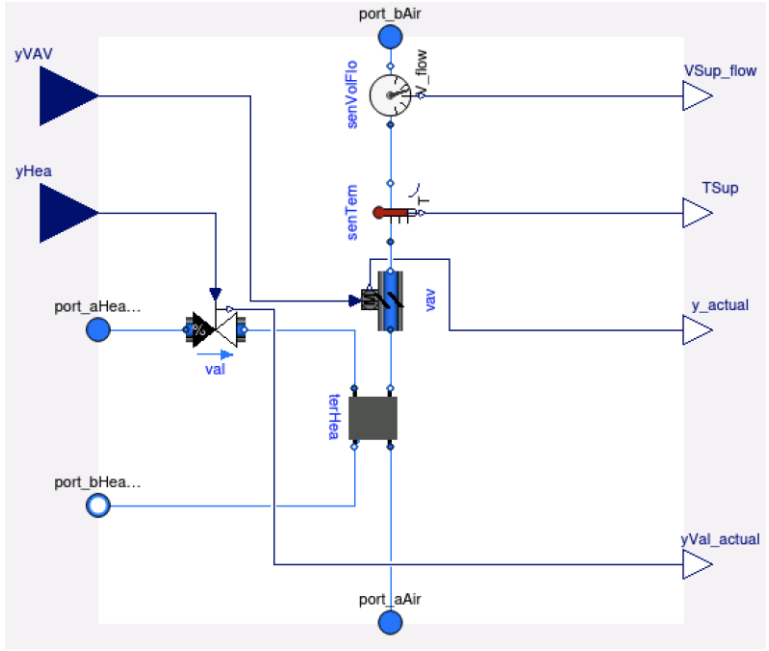
See releasenotes

Updating Modelica Libraries and Compilation of Test Cases

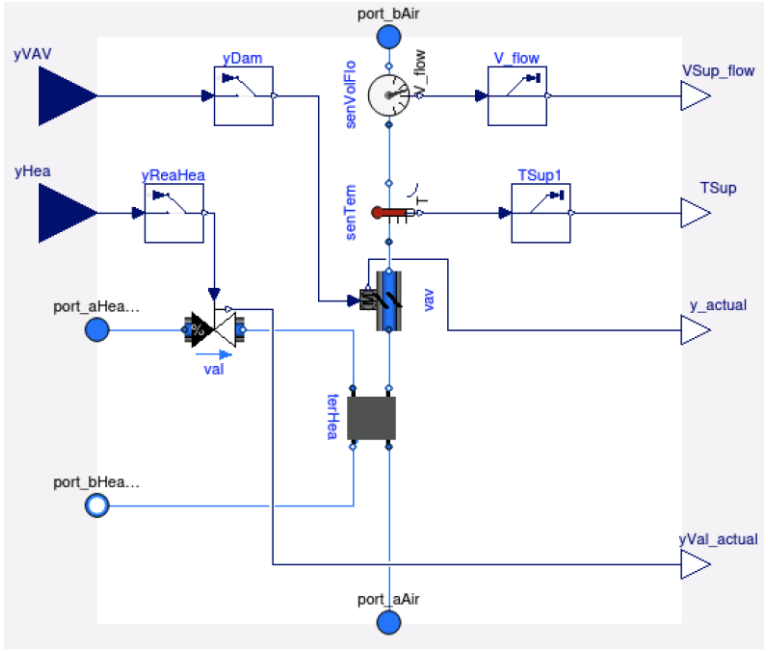
Notes on test case compilation future:

- If have Dymola with binary model export, use to compile any CD FMUs like before.
- If have Dymola with no binary model export, and creating/customizing a test case, could copy license file into BOPTEST worker container upon `compose up`, so pyfmi has access to it (if single seat license, only one worker at a time).
- Future of OpenModelica is brighter:
  - Pyfmi developers may soon be fixing the bug that prevents BOPTEST use of ME FMUs (see PR #412)
  - OpenModelica is much better at exporting ME FMUs than CS FMUs
  - Most test cases can compile and simulate in OpenModelica itself
- Optimica (Impact) works too, though haven't kept up on FMU export licensing requirements.

Use of Modelica “Break” Statements



Consider have this.



For BOPTEST, want this.

- Need to duplicate and/or edit model with signal exchange blocks
- Dependency on original model broken – can’t inherit updates
- Original model can get cluttered with a lot of signal exchange blocks

Use of Modelica “Break” Statements

```
model VAVReheatBox
  extends Buildings.Examples.VAVReheat.BaseClasses.VAVReheatBox(
    break connect(vav.y, yVAV),
    break connect(yHea, val.y),
    break connect(senTem.T, TSup),
    break connect(senVolFlo.V_flow, VSup_flow));

  parameter String zone="1" "Zone designation, required if KPIs is
AirZoneTemperature, RadiativeZoneTemperature,
OperativeZoneTemperature, RelativeHumidity, or CO2Concentration";

  Buildings.Utilities.IO.SignalExchange.Overwrite yDam(
    description="Damper position setpoint for zone " + zone,
    u(unit="1", min=0, max=1)) "Damper position setpoint"
    annotation
(Placement(transformation(extent={{-60,70},{-40,90}})));
```

Using the “break” statement.

Using break statements could allow for:

- If the original model is updated, the updates are passed through to the test case model
- Don't have to maintain an original model with all the signal exchange blocks, which can be graphically cumbersome
- Opens up potential for automated scripting and templating for adding signal exchange blocks

Tested with Dymola 2025x and OpenModelica v1.26 and seems to work.