Using SPARK as a Solver for Modelica

Michael Wetter

Philip Haves

Michael A. Moshier

Edward F. Sowell

July 30, 2008

Overview

- Overview of SPARK, Modelica, OpenModelica, Dymola
- Problem reduction
- SPARK integration in OpenModelica
- SPARK/Dymola benchmark
- Future work

SPARK, Modelica, OpenModelica, Dymola

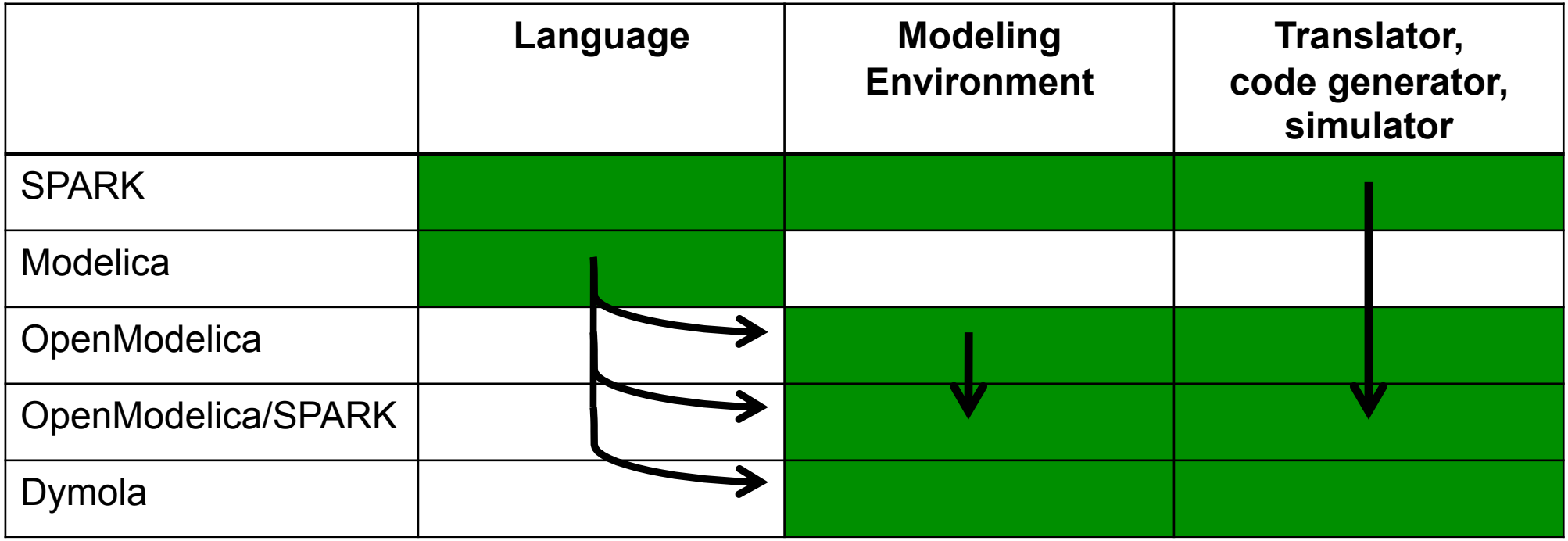
Modelica

- Modeling language developed since 1996 by Modelica Association
  - Model development environments
  - Solvers
- Designed by developers of
  - Allan, Dymola, NMF, ObjectMath, Omola, SIDOPS+, Smile
- Well positioned to become de-facto standard for modeling multi-engineering systems
  - e.g.: ITEA2 (Eurosyslib: 110 man years, 20 partners to develop Modelica libraries, but not for building HVAC systems).
- Supports differential, algebraic and discrete equations
- Declarative instead of imperative
- Object-oriented
- Automatic documentation

SPARK

- Developed since 1989 by LBNL and Ayres Sowell Associates
- Supports differential and algebraic equations
- Declarative instead of imperative
- Uses graph-theoretic methods to:
  - define well-posed problems
  - generate computationally efficient solution sequences

Overview



Questions:

1) How can SPARK be used as a solver for OpenModelica?

2) How does SPARK compare with best in class Modelica simulator?

Dymola/SPARK Comparison

Features and availability

	Dymola	SPARK
Partitioning		
Tearing		
Inline-integration		
Index reduction		
Automatic differentiation		
Availability	$1-10k	Executable free (kernel is not open-source)

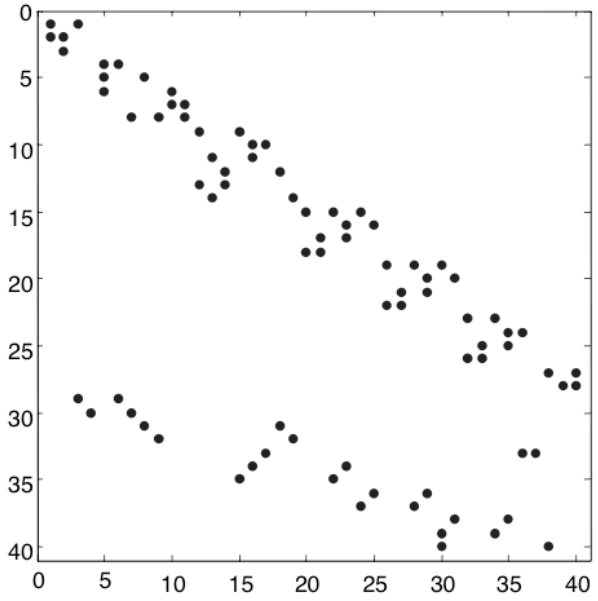
Partitioning or Component Reduction

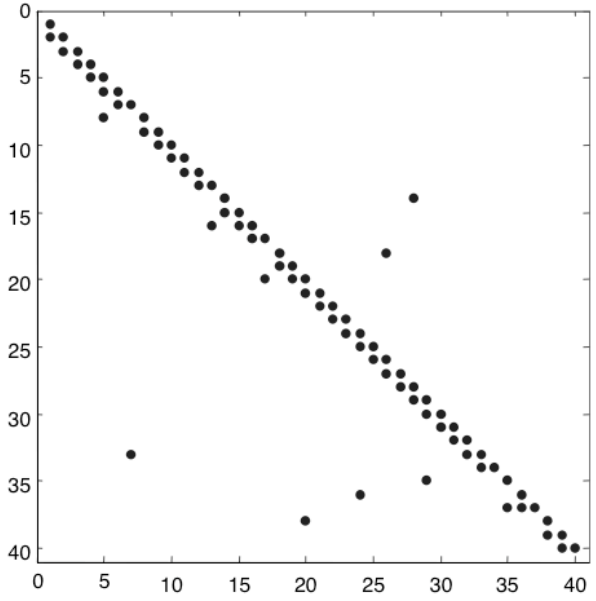
$$1: \quad Q = T_1 - T_2$$
$$2: \quad 0 = T_1$$
$$3: \quad f(t) = T_1 + T_2$$

	T_1	T_2	Q
1			
2			
3			

	T_1	T_2	Q
2			
3			
1			

Application to an electric circuit model





Bunus and Fritzson (2004)

Tearing or Cut-Set Reduction

Suppose we have an equation

$$0 = f(x),\ x \in \Re^n,\ n > 1,$$

that can be written in the form

$$(1)\quad L\,x^1 = \hat{f}^1(x^2),$$

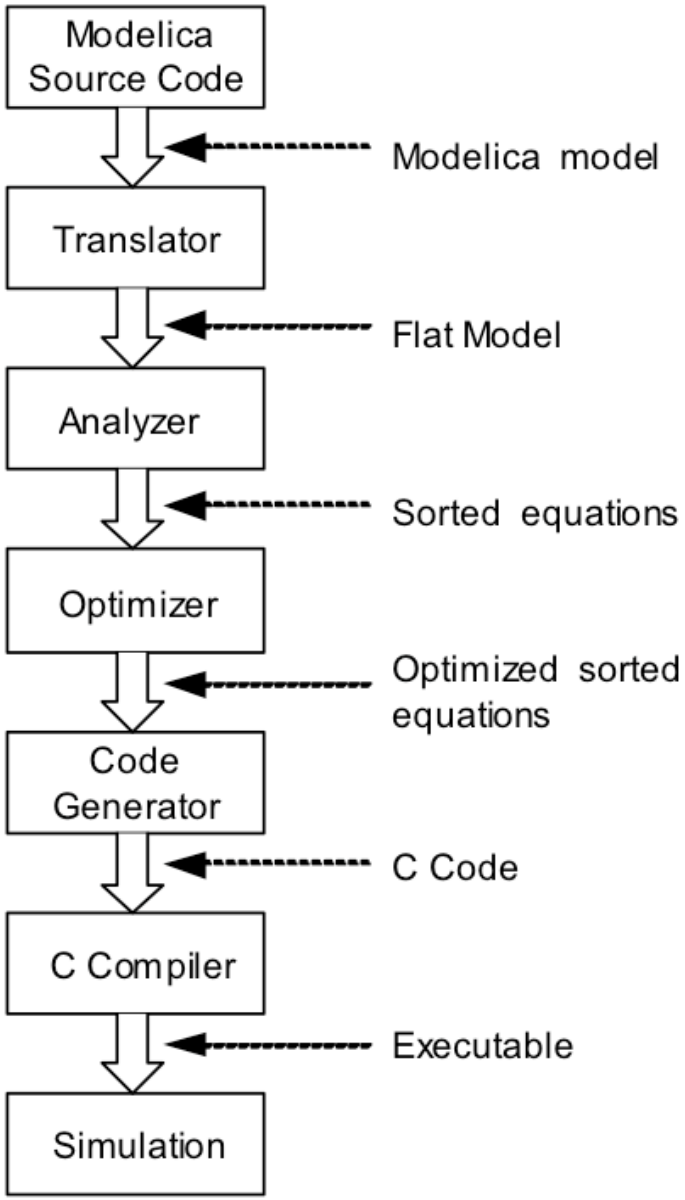
$$(2)\qquad 0 = \hat{f}^2(x^1, x^2),$$

where *L* is a lower triangular matrix with constant non-zero diagonals.

Now, pick a guess value for x^2, solve (1) for x^1, and compute a new value for x^2 from (2).
Iterate until x^2 converges to a solution.

SPARK/OpenModelica Integration

SPARK Integration



SPARK integration

Match variables with equations

Derive equation inverses

Use SPARK solver
- symbolic processing
- numerical solver

Fritzson *et al. (2007)*

Integration

- Modify the open source Modelica compiler to
  - Construct internal data structures corresponding to SPARK classes (individual equations and sub-systems)
  - Generate SPARK class definitions
  - Use another open source system (e.g., SAGE) to perform needed algebraic manipulations
- Use Modelica annotations (part of the Modelica standard) to give user access to SPARK controls
- Modify SPARK API to integrate diagnostics into OpenModelica (future work)
- Build algorithmic code generator (near future work)

Current Status

- Compiler now emits basic version of SPARK model
  - No user control via annotation
  - Does not provide code for re-use
- Compiler supports data structures corresponding to SPARK classes (sub-models) including user controls.
- Need to build code generator for the latter (straightforward task)

Dymola/SPARK Benchmarking

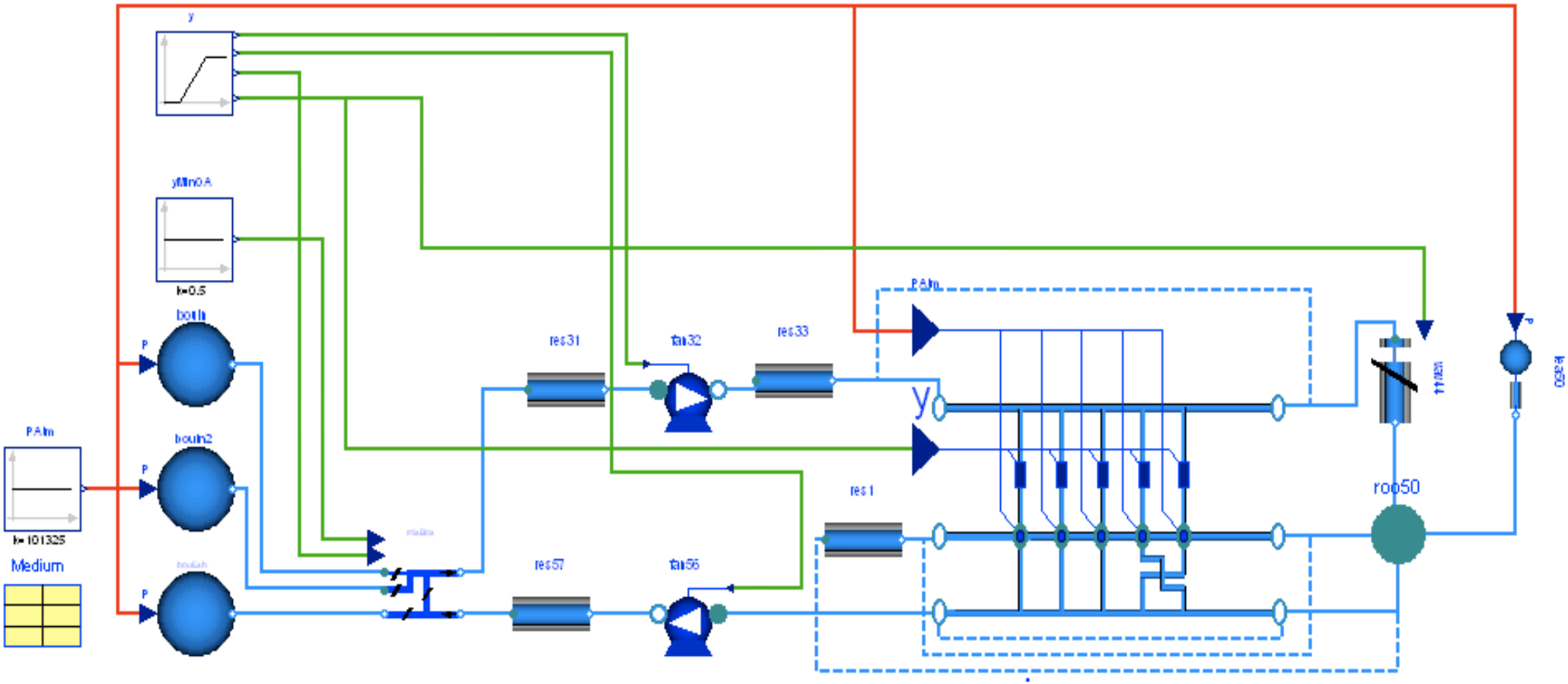
Dymola/SPARK Benchmarking

Test problem:

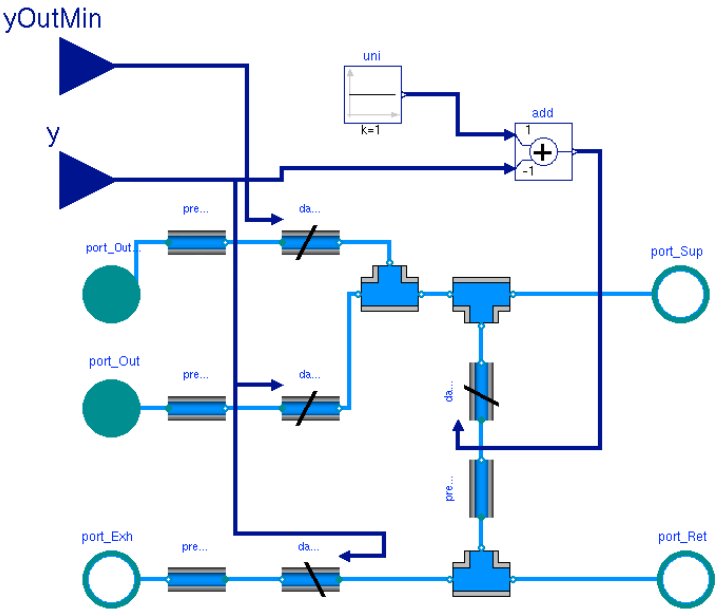
- Airflow network (pressure & mass flow rate only)
- Algebraic system of equations
- Scaled VAV air distribution system (ASHRAE 825-RP)

Dymola/SPARK Benchmarking

Schematic view of VAV System described in ASHRAE 825-RP



Dymola/SPARK Benchmarking



Splitter - FlowNetwork.Components.Splitter - [Modelica Text]

lot Animation Commands Window Help

100%

Packages

- FixedResistanc...
- FlowMachinePo...
- OAMixingBoxMi...
- PrescribedSource
- SimulationTime
- Splitter
- SplitterFixedResis
- Tests
- VAVBoxExponent
- Volume

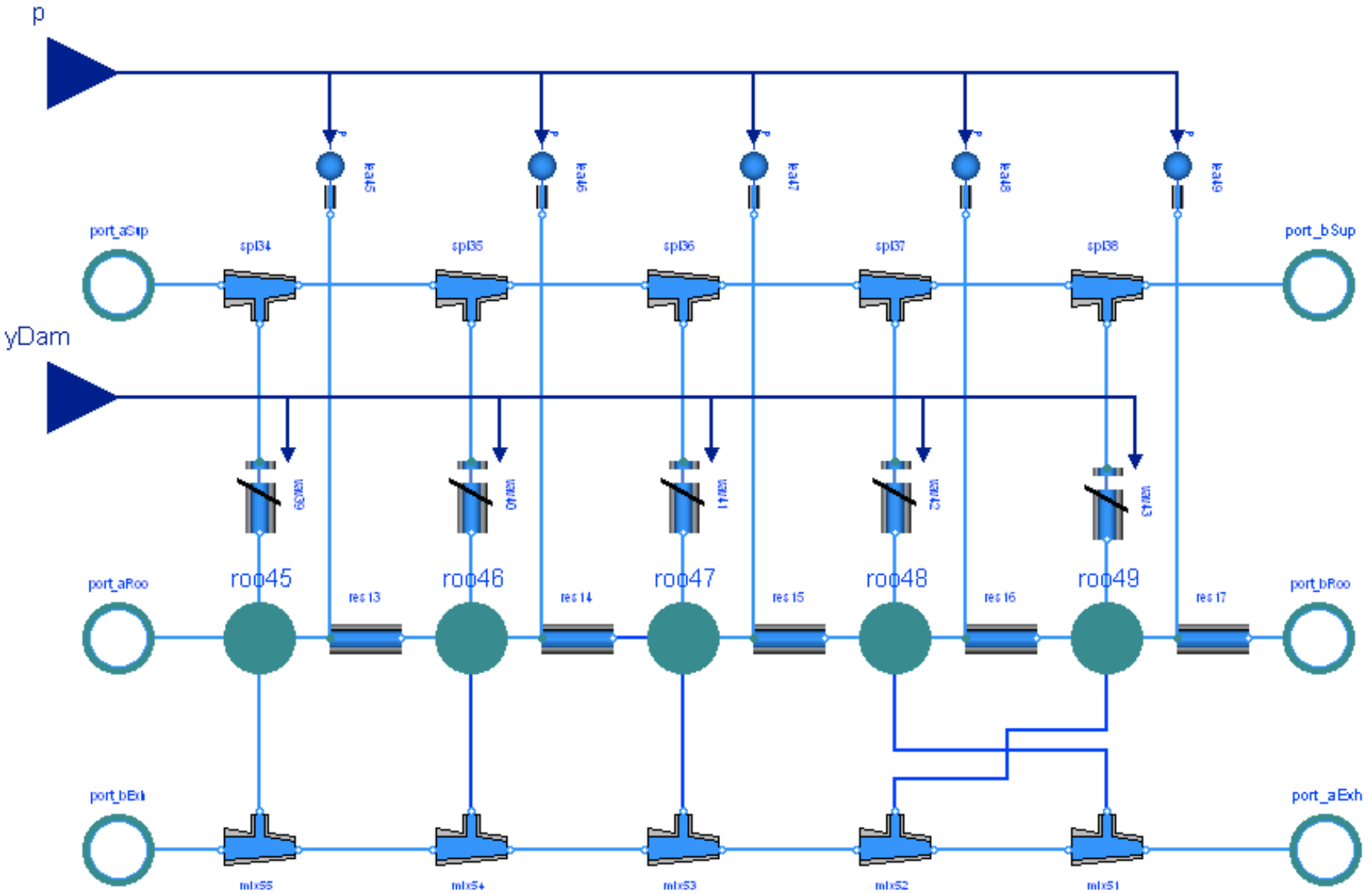
Components

- FlowNetwork.Compo...
  - port_1
  - port_2

```
model Splitter
  "Splitting/joining component with static
     balances for an infinitesimal control volume"

  FlowNetwork.Interfaces.FluidPort_b port_1 ;
  FlowNetwork.Interfaces.FluidPort_b port_2 ;
  FlowNetwork.Interfaces.FluidPort_b port_3 ;

  Modelica.SIunits.AbsolutePressure p(nominal=1E5) "Pressure";
  ;
equation
  port_1.m_flow + port_2.m_flow + port_3.m_flow = 0 "Mass balance";

  // Momentum balances
  port_1.p = p;
  port_2.p = p;
  port_3.p = p;
end Splitter;
```

Line: 3 Modeling Simulation

Dymola/SPARK Benchmarking



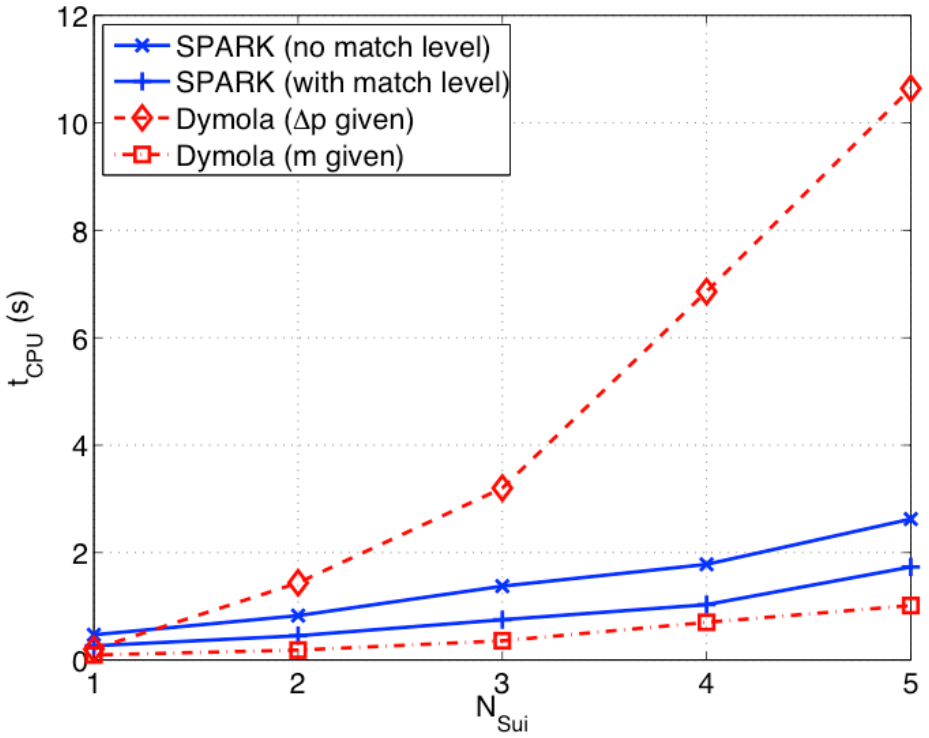
Dymola/SPARK Benchmarking

Symbolic processor

Table 1: Dimension of biggest nonlinear system of equations. For the SPARK, no ML denotes no `MATCH_LEVEL` *and with ML denotes with* `MATCH_LEVEL`.

N_{sui}	SPARK		Dymola	
	no ML	with ML	$\dot{m} = f(\Delta p)$	$\Delta p = g(\dot{m})$
1	21	18	41	26
2	36	34	71	47
3	51	50	101	66
4	66	65	131	87
5	81	80	161	106

Numerical performance



Summary of Benchmarks

- SPARK is compatible with Modelica language
- SPARK has a robust, high performance solver for nonlinear equations, competitive with expensive commercial systems
- OpenModelica has provided a useful testbed for integrating SPARK

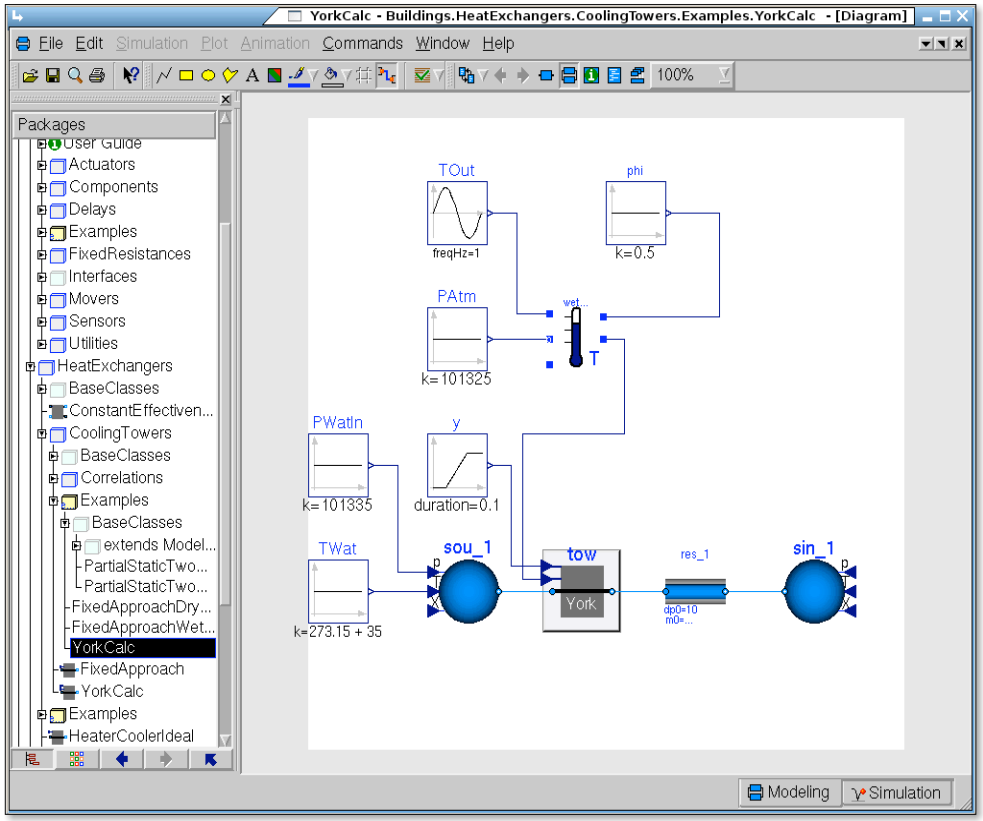
Future work…

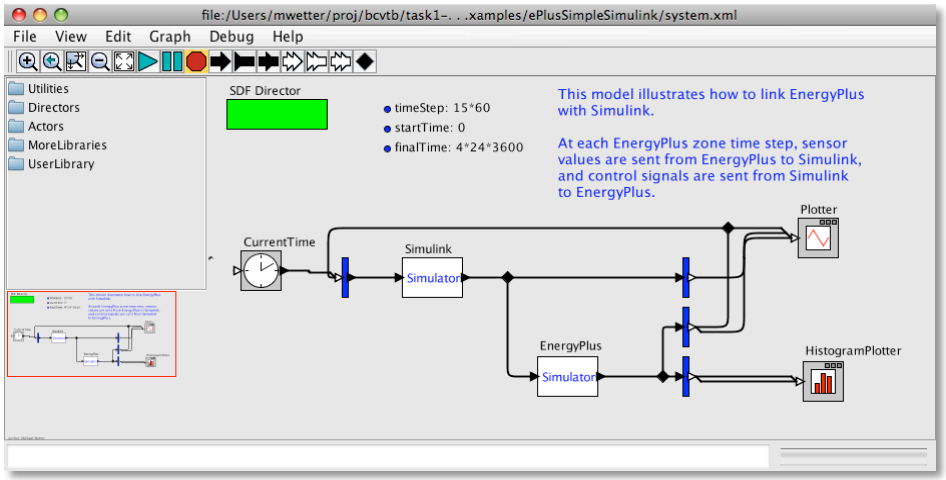
Future Work

SPARK/OpenModelica: Continue integration

Modelica modeling: Open-source buildings library development

Modelica co-simulation: Open source Building Controls Virtual Test Bed development.
Will link Modelica to EnergyPlus.





https://gaia.lbl.gov/bir

https://gaia.lbl.gov/bcvtb