A Modular Building Controls Virtual Test Bed for the Integration of Heterogeneous Systems

Michael Wetter and Philip Haves

Building Technologies Department
Energy and Environmental Technologies Division
Lawrence Berkeley National Laboratory

July 31, 2008
Need for Innovation

Grueneich, 2008
Overview

Problem
- Analysis tools don’t integrate with each other
- Why do we write *simulation programs* when we need a *model*?

Approach
- Ptolemy II middleware

Examples

Next steps
Problem
How do we address integration challenges of energy, control and communication systems?

Energy
Exploit building dynamics (thermal/humidity storage). Account for part load system performance.

Controls
Optimal thermal storage management. Multi-variate controls (temperature, humidity, IAQ, light, peak power, energy).

Communication
Density of sensors, computing nodes and actuators increases. Controls vendors prefer not to close control loops with fast dynamics over networks because network may jam unpredictably.

Situation
Today’s building simulation tools are quasi-steady state. Most building control systems don’t work well. Disconnect - design and operations. - disciplines.
Problem: Non-integrated domain specific tools
Aim

1) Integrated multidisciplinary analysis
2) Model-based system level design & optimization for design phase
3) Model deployment during building operation to optimize performance
Requirements from Industry Group

**Functional Requirements**

- Shorter time-to-market, model creation “on the fly,” transparent model assumptions & limitations
- Model use during operation
- Performance assessment of controls
- Local & supervisory control
- Optimization-based control
- Embedded systems

**Technology Needs**

- Object-oriented equation-based models
- Acausal encapsulated models
- Prototypical buildings simulated in validated tools
- Frequency analysis, dynamic plant models, hybrid automata
- Models amendable for optimization
- Different models of computation
Today’s monolithic building simulation programs

Innovation:  
- Adding new models takes **months**.
- Many **low energy systems** cannot be simulated.

Controls:  
- HVAC models are **quasi steady-state**.
- “Control” is based on “**requested energy**,” not on actual feedback control.

Collaboration:  
- Not designed for collaborative research.

Future environment for modeling and simulation

Provides a **modular extensible platform** to stimulate R&D.

Lets users **rapidly add** and **share** new component and system models.

Allows **model extraction** from “design” for reuse during “operations.”
Structure the problem the way you think, not how you compute a solution

Traditional approach

```
procedural

class name

Data

procedures

IF (GetMixerInputFlag) THEN
  CALL GetMixerInput
  GetMixerInputFlag = false.
ENDIF

CALL GetConnectorList(ConnectorListName, Connectoid, Conn
  IF (Connectoid%ConnectorType(1) == MIXER) THEN
    Count = FindItemInList(Connectoid%ConnectorName(1), Mixe
    IF (PRESENT(MixerNumber)) MixerNumber = MixerNumber +
    IF (Count == 0) THEN
      CALL ShowFatalError('GetLoopMixer: No Mixer Found=
    ENDIF
```

New approach

```
equation-based

port \((q_1, T_1)\)  \quad q_1 = K (T_1 - T_2) \quad 0 = q_1 + q_2

actor-oriented

<table>
<thead>
<tr>
<th>actor name</th>
<th>states</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>input data</td>
<td>ports</td>
<td>output data</td>
</tr>
</tbody>
</table>

3 studies show 3-5 times faster development time using equation-based modeling.

80% of source code is for data management.
Approach

Modeling vs. simulation

Physical System \(\xrightarrow{\text{Modeling}}\) Mathematical Model \(\xrightarrow{\text{Model Compilation}}\) Simulation Program \(\xrightarrow{\text{Model Simulation}}\) Trajectory Behavior

Cellier & Kofman, 2006

However…

- There is still a need to use **legacy** simulation programs.
- How do we **link** new modeling approaches with legacy simulation programs using co-simulation?

Legacy simulation tools:

- 100+ man years development time.
- Some models have been extensively validated.
Ptolemy II

Open-source software framework for modeling & simulation of heterogeneous systems

Author: Michael Wetter
Ptolemy II

- Used as **middleware** for LBNL’s Building Controls Virtual Test Bed
- **Open-source, platform independent**, developed over last 10+ years at UC Berkeley (EECS)

**Architecture**
- **Actors** that process input tokens
- **Links** that define actor connectivity
- **Model of Computations** that define actor interactions
  - Synchronous Data Flow
  - Discrete Event
  - Finite State Machine
  - Continuous Time
  - etc.
BCVTB Implementation

Modular architecture built around Ptolemy II

**Ptolemy II**
Open-source software laboratory for heterogeneous systems.

BCVTB extensions to interface external programs

API to link programs to Ptolemy II

Legend
- Implemented
- Planned

API: Application Programming Interface

- EnergyPlus
- MATLAB/Simulink
- Dymola
- BACnet
- A/D
Ex: Simple Room Heater, Implemented in C
Ex.: Natural Ventilation in SF Fed. Bldg.

All software modules reusable without code modification

Ptolemy II

MATLAB/Simulink

EnergyPlus

Real-time output

This model illustrates how to link EnergyPlus with Simulink.
At each EnergyPlus zone time step, sensor values are sent from EnergyPlus to Simulink, and control signals are sent from Simulink to EnergyPlus.
This model illustrates how to link EnergyPlus with Simulink. At each EnergyPlus zone time step, sensor values are sent from EnergyPlus to Simulink, and control signals are sent from Simulink to EnergyPlus.

Output simulation time and wall clock time. This is for illustration purposes only and not needed by the above model.

Expression
input: 86400
SimulationTime
0.606944444444
WallClockTime
RunTime
17.624

Author: Michael Wetter
EnergyPlus linked to BCVTB

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<BCVTB-variables>
  <variable source="EnergyPlus">
    <EnergyPlus name="ZONE SOUTH"
      type="ZONE/SYS AIR TEMPERATURE"/>
  </variable>
  <variable source="Ptolemy">
    <EnergyPlus dayschedule="SP-TCooling"/>
  </variable>
  <variable source="Ptolemy">
    <EnergyPlus dayschedule="SP-THeating"/>
  </variable>
</BCVTB-variables>
```
MATLAB/Simulink-based controls development, linked to BCVTB

Implementation of controls algorithm

Library with BSD socket interface

.runtime coupling
Modeling of HVAC Systems

Modelica Buildings Library.

Open Source Development: https://gaia.lbl.gov/bir

Buildings
Controllers
Fluids (resistances, actuators, fans, sensors)
Heat exchangers
(cooling towers, coils)
Mass exchangers
(humidifiers)
Media (perfect gases, constant property water)
Plants (furnaces, heat pumps)
Utilities (mathematics, psychrometrics, file readers, file printers, sockets)
Modeling of HVAC Systems

Textual & graphical editor to define dynamic component models

Modelica simulation environment

Modelica component library

Packages
- Actuators
- Components
- Delays
- Examples
- FixedResistance
- Interfaces
- Media
- Movers
- Sensors
- Utilities

Runtime coupling
Adding your own simulator

- Download the BCVTB
  https://gaia.lbl.gov/bcvtb

- Integrate in your simulator:
  1: `establishclientsocket(...)`
  2: `exchangewithsocket(...)`
    [each time step]
  3: `closeipc(...)`

Examples are provided for

- EnergyPlus, MATLAB/Simulink, C, FORTRAN 90
- Apple, Linux, Windows XP
Next Steps

- Add interfaces to
  - Modelica, A/D converter, BACnet
- Development, testing and performance assessment of HVAC control sequences
- Expand Modelica HVAC component library