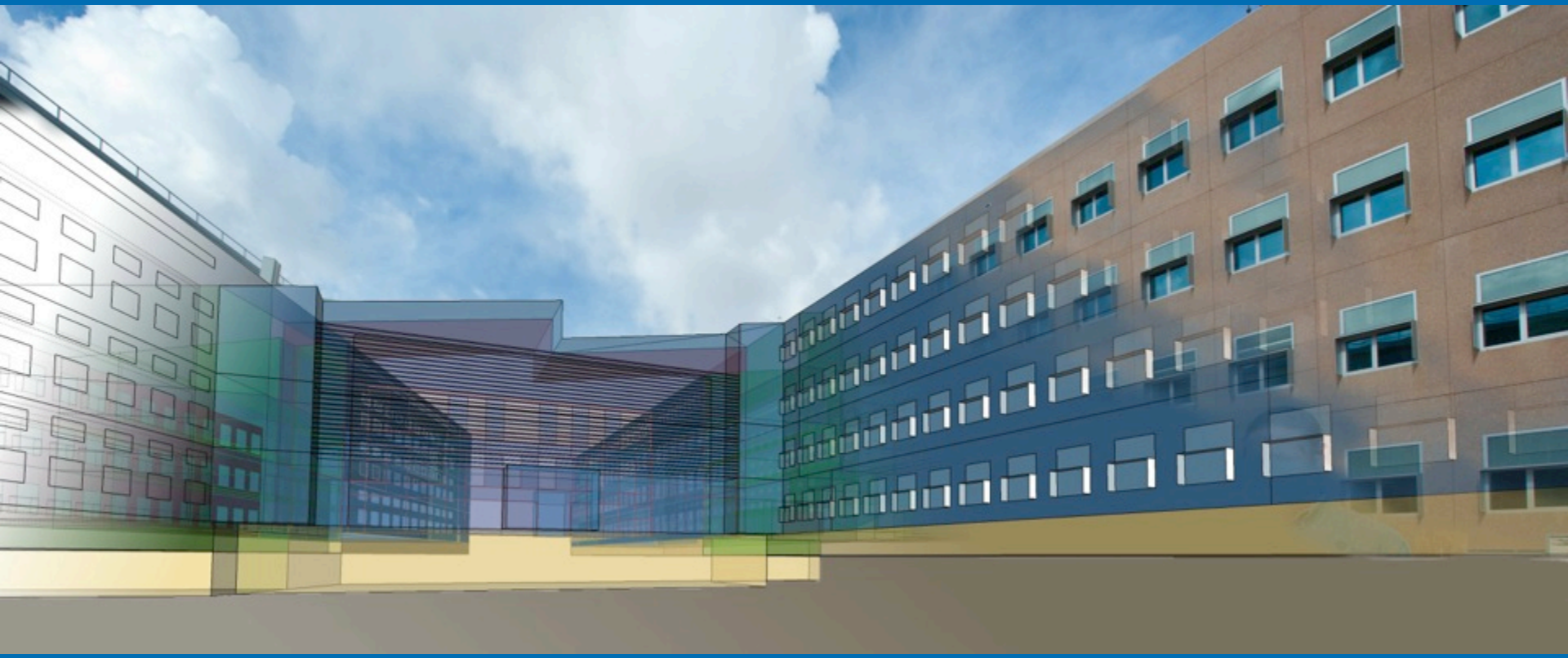
NREL
NATIONAL RENEWABLE ENERGY LABORATORY

OpenStudio-Radiance Update



Rob Guglielmetti
NREL Commercial Buildings Group

14th Annual International Radiance Workshop
August 17-19, 2015

Agenda

- OpenStudio and DOE's BEM Ecosystem
- OpenStudio Measures
- Daylighting and BEM rant
- OpenStudio/Radiance Features Update
- NREL/GitHub mirror of Radiance Source Repository

Inspirational Quotations

"To nature, who gives us marvelous complexity, all the while making it look easy."

- Greg Ward, "Rendering with Radiance"

"Without lamps, there'd be no light."

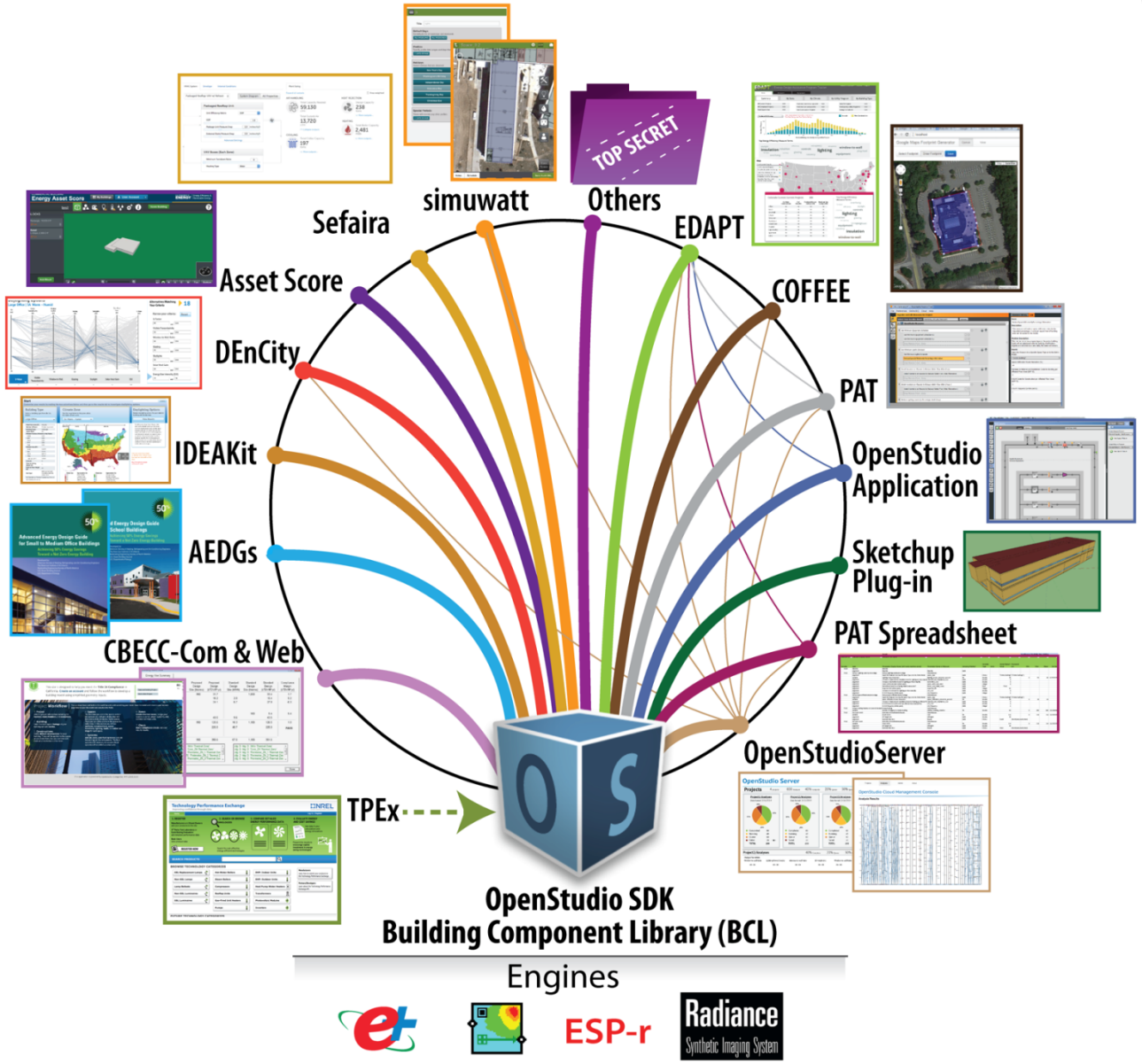
- John Bender, "The Breakfast Club"

(obviously, not a lighting designer)

OpenStudio's Energy Modeling Ecosystem

OpenStudio is an open source platform that bridges the gap between capable but complex engines and the easy-to-use applications that drive energy savings.

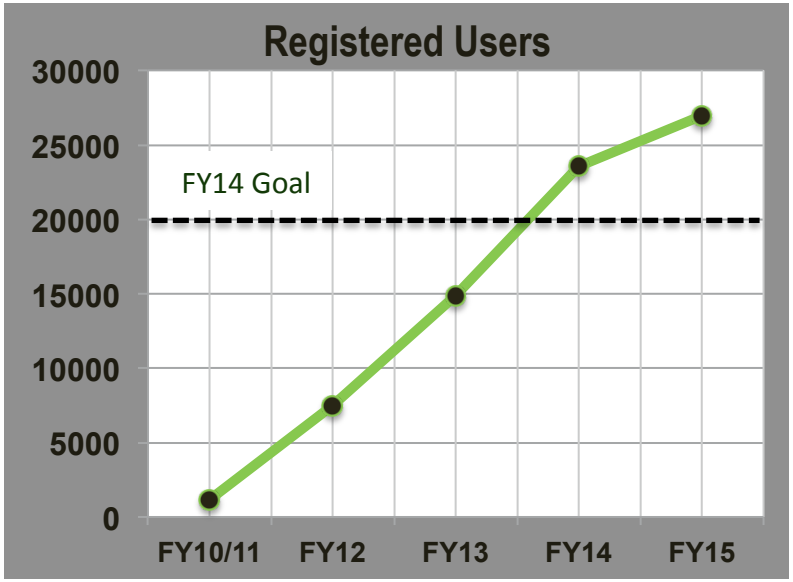
The Technology Performance Exchange (TPEx) and Building Component Library (BCL) provide the raw data that powers the ecosystem.



Who Uses OpenStudio?

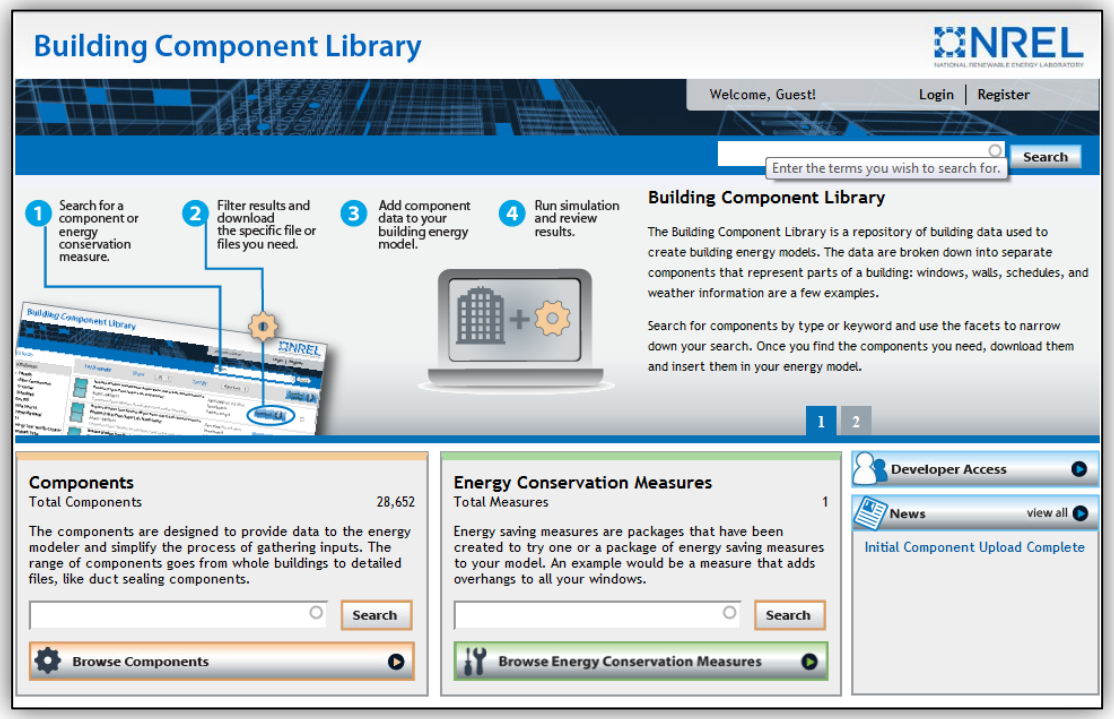
The market for OpenStudio is diverse and includes:

- Developers wishing to quickly and cost-effectively bring new software to market;
- A&E practitioners who use applications for new construction and retrofit projects;
- Utilities trying to reduce costs of incenting EE, improve realization rates, and assess technology potentials;
- Researchers designing and assessing new technologies; and
- Students who are the next generation of building designers.



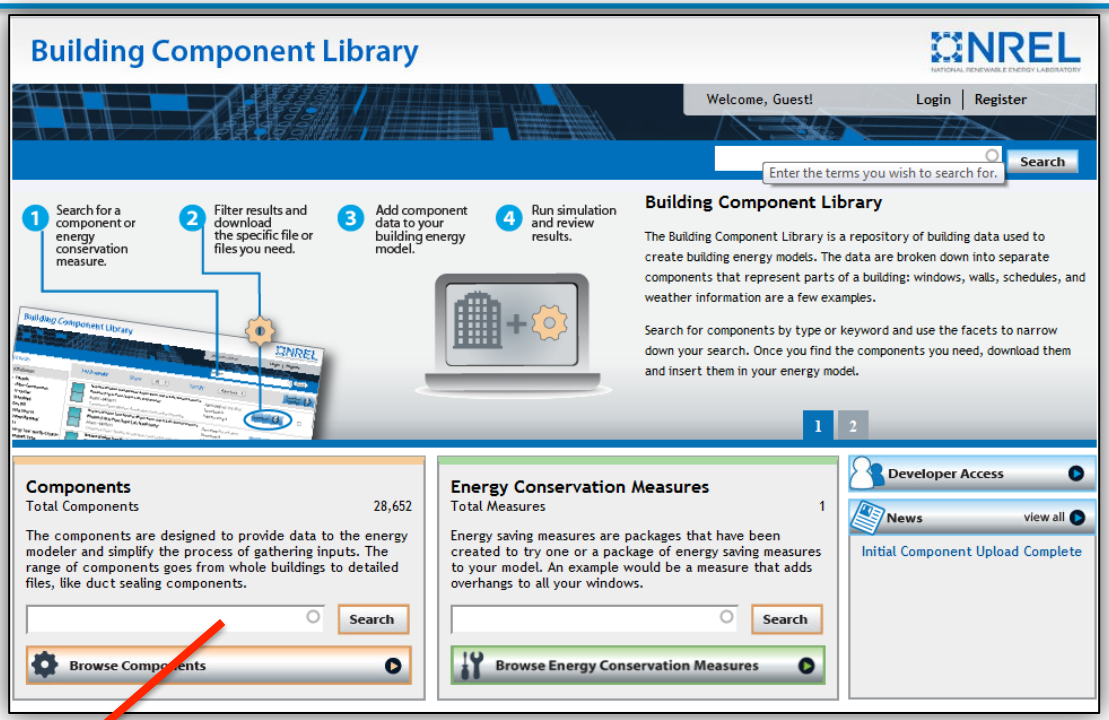
OpenStudio Partners (Partial List)

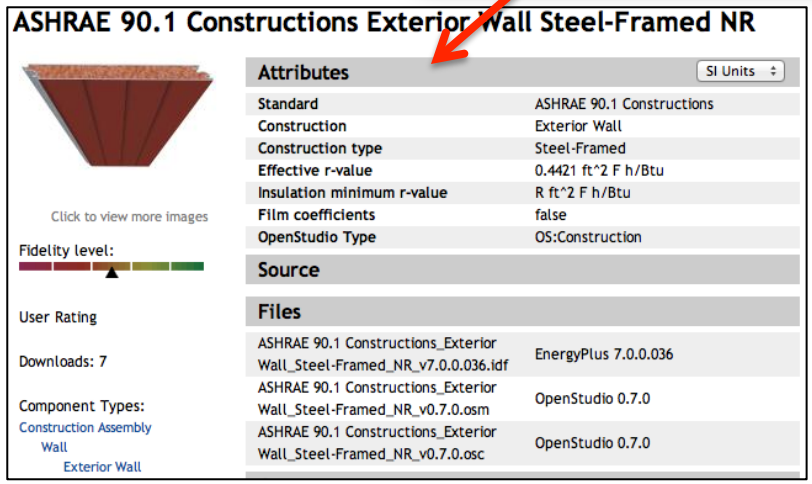
Foundations: The Building Component Library (BCL)



- An Internet-connected source of building energy modeling data:
  - Enables drag-and-drop modeling for **quick** technology evaluations
  - Provides **consistent**, detailed inputs to drive decision-making
  - **Searchable** readily available within applications
  - The BCL is key to OpenStudio's **extensibility**

The BCL: A Source for Reusable Input Data…

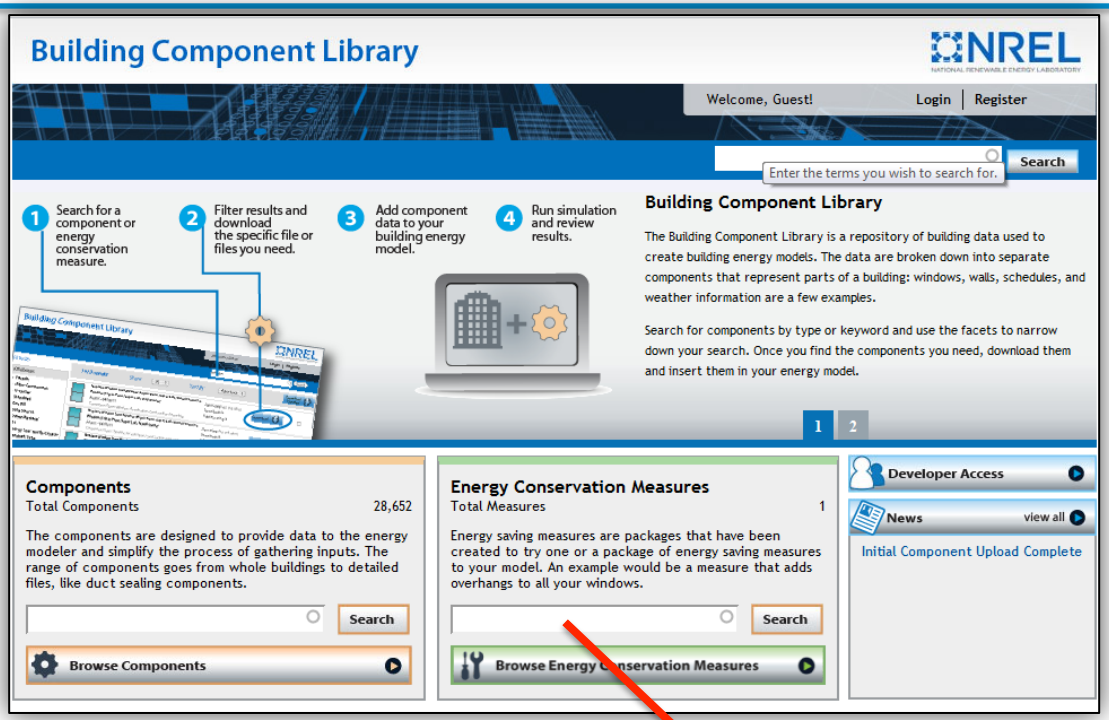


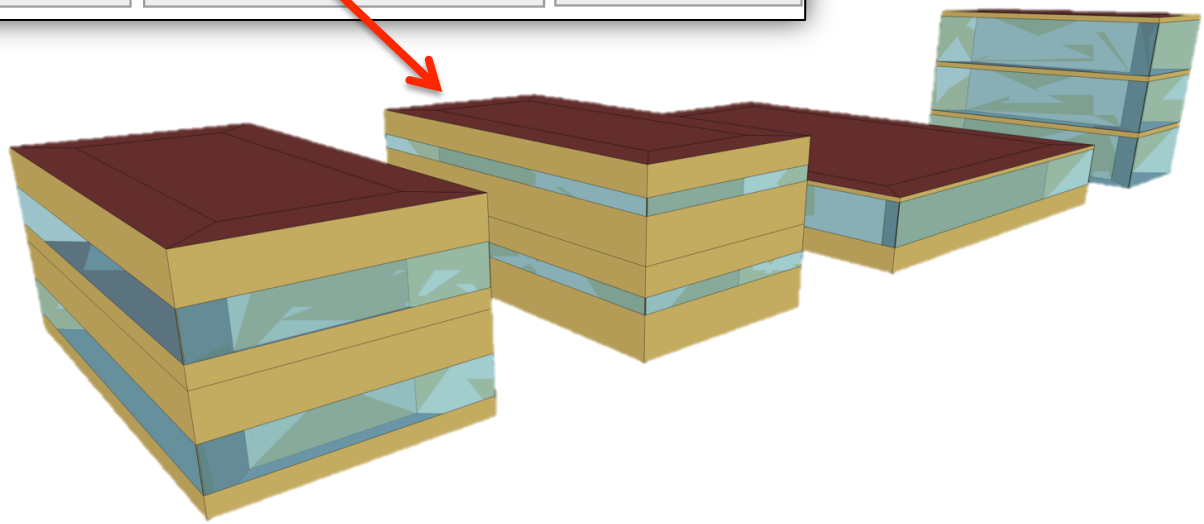


Components:

- Assembled to form complete energy models
- Include constructions, lights, schedules, weather data, PV components, and more

…and a “Measures” Repository



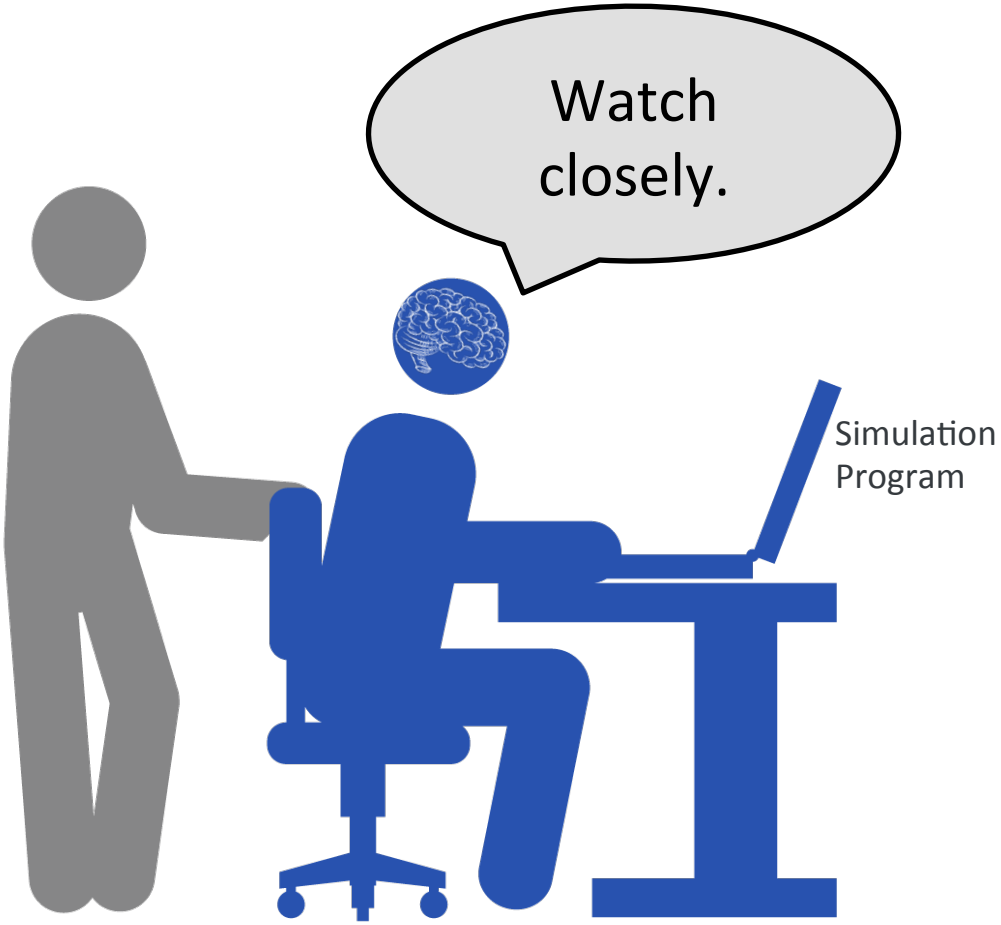


Measures:

- Contain logic needed to transform an energy model easily and consistently
- Can be applied singly or as part of a parametric analysis

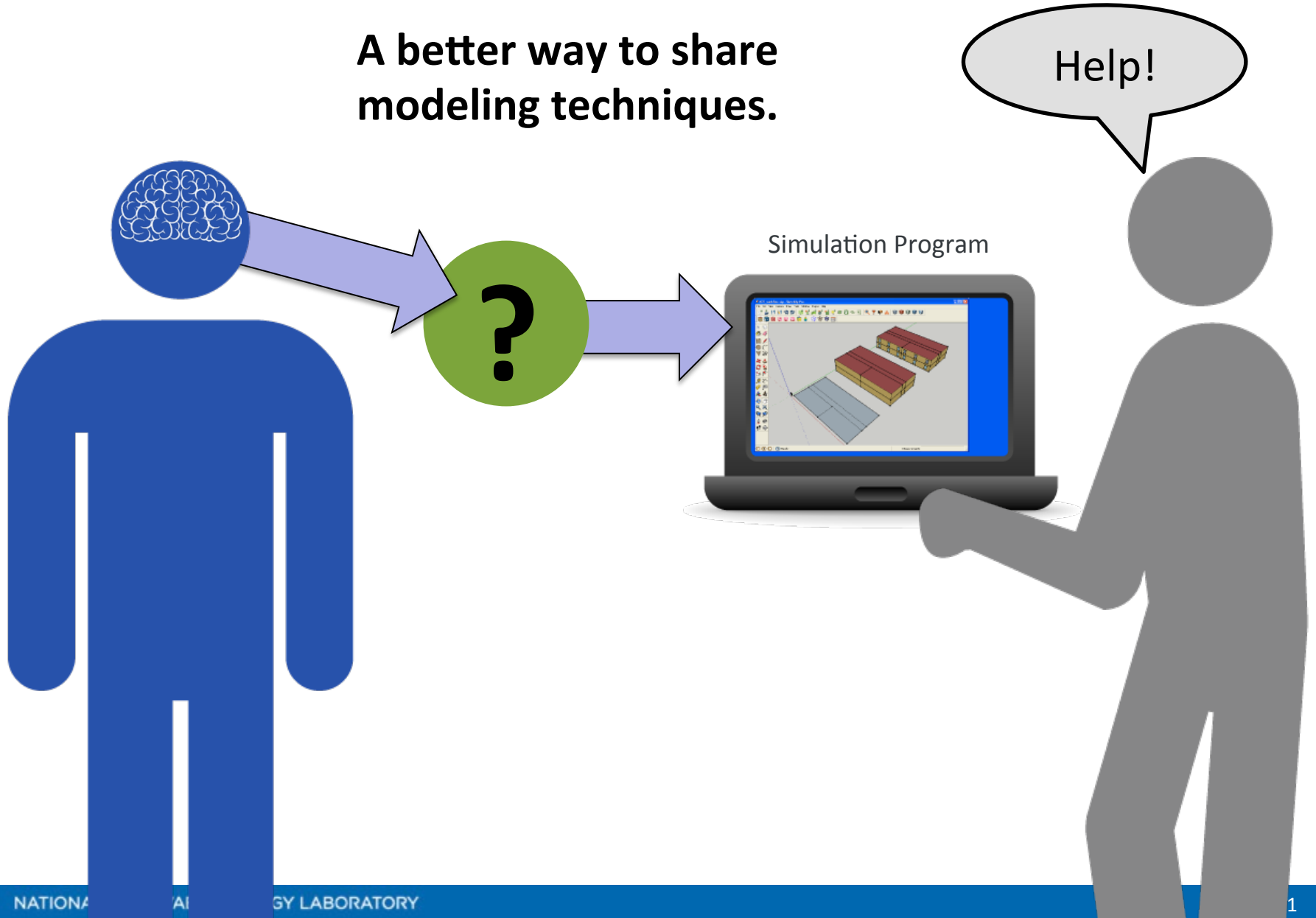
Modeling Today

Knowledge is passed one-to-one:
Inconsistent, not scalable, not transparent



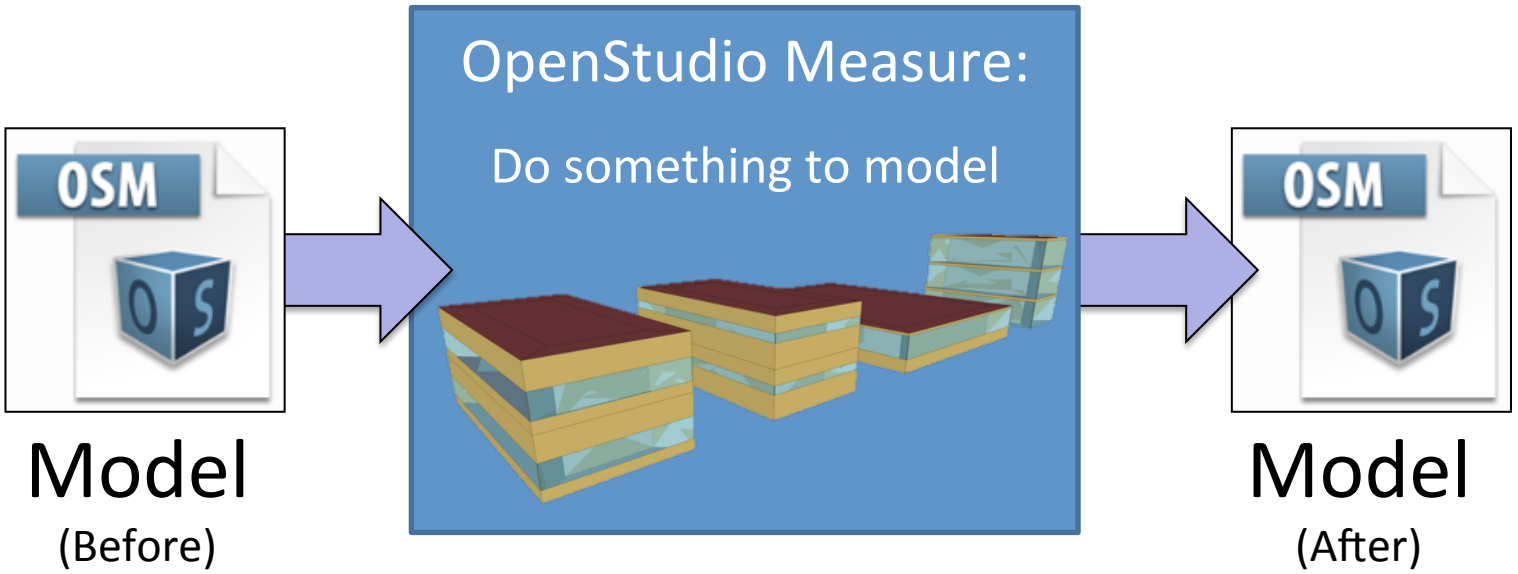
What We've Needed

A better way to share modeling techniques.

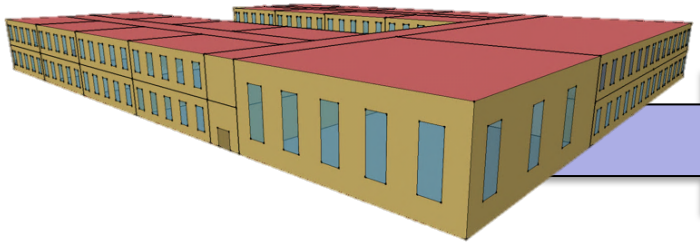


It Exists! – OpenStudio Measures

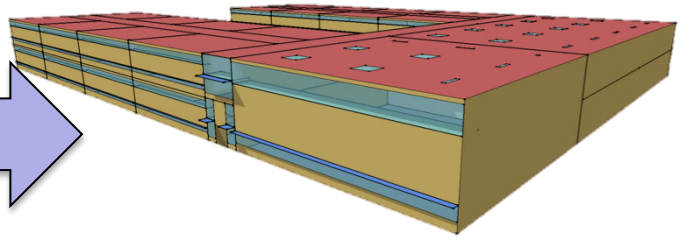
OpenStudio Measures are (generally) short scripts that transform building models **quickly** and **easily**

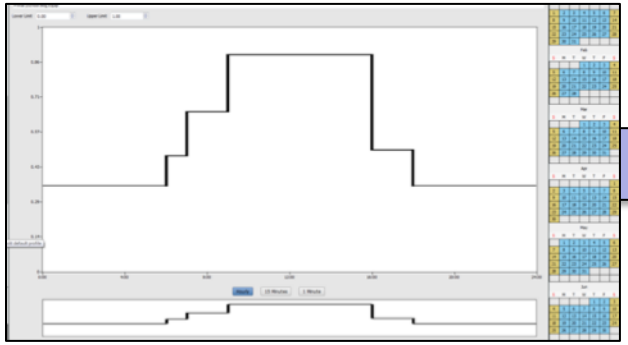


Sample Measures

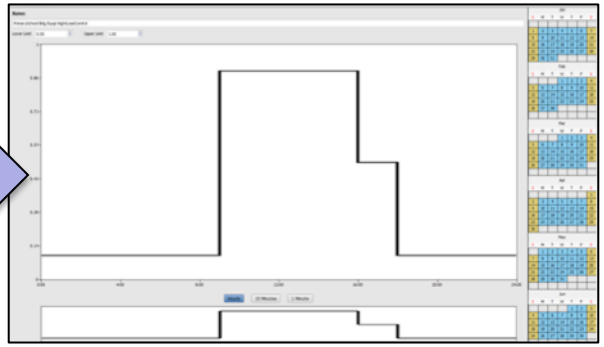


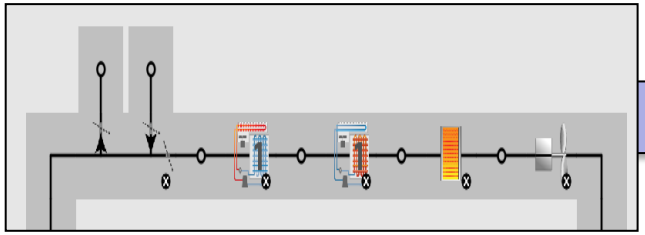
Daylighting Measure



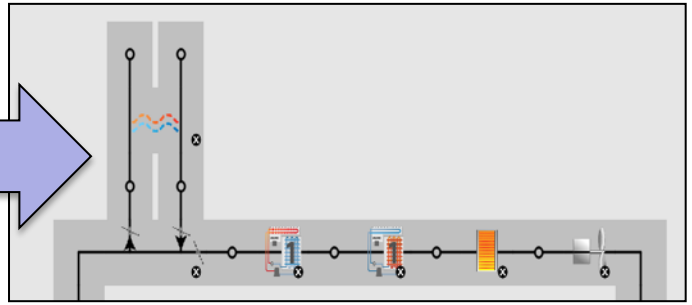


Night Setback Measure

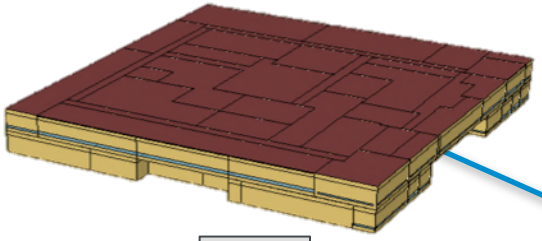




Add Heat Recovery Measure

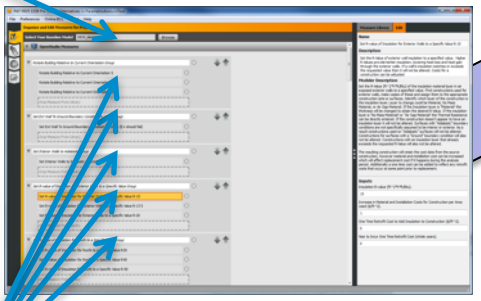


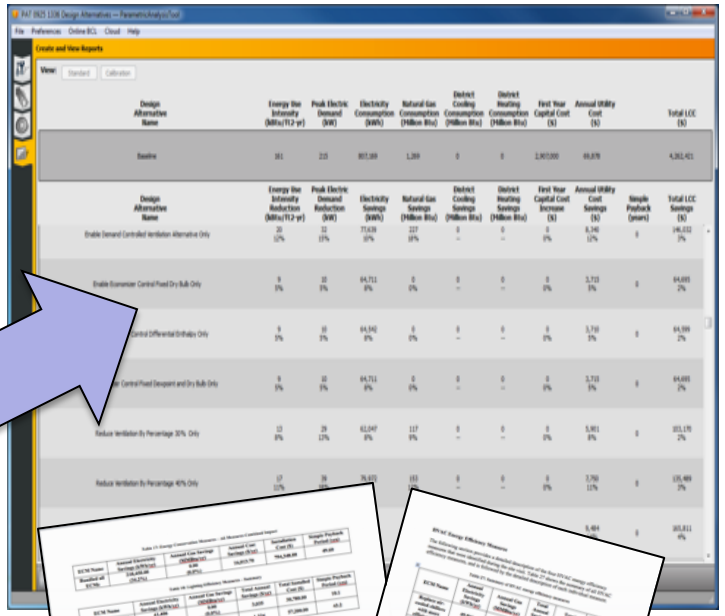
Measures Workflow Example

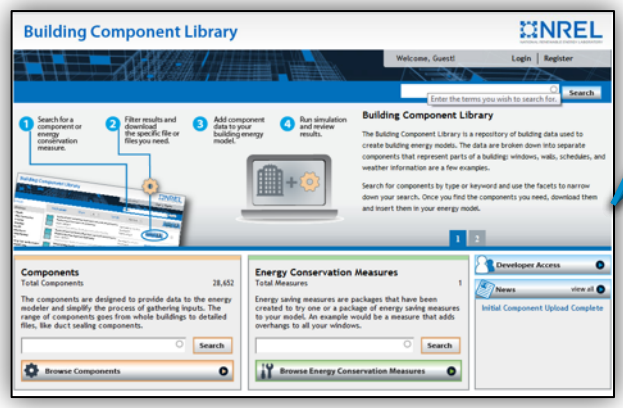


Baseline Model

OpenStudio Parametric Analysis Tool (PAT)







Online Database of Measures

Life cycle analysis identifies cost-effective measures that are automatically documented

simuwatt

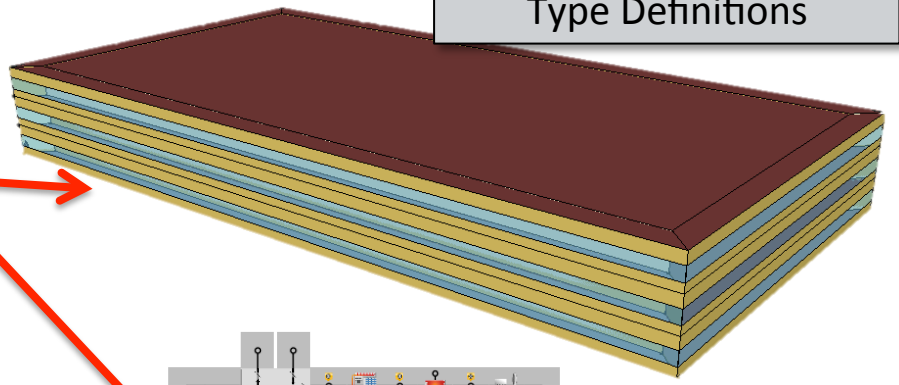
OpenStudio Enables Automated Model Generation…

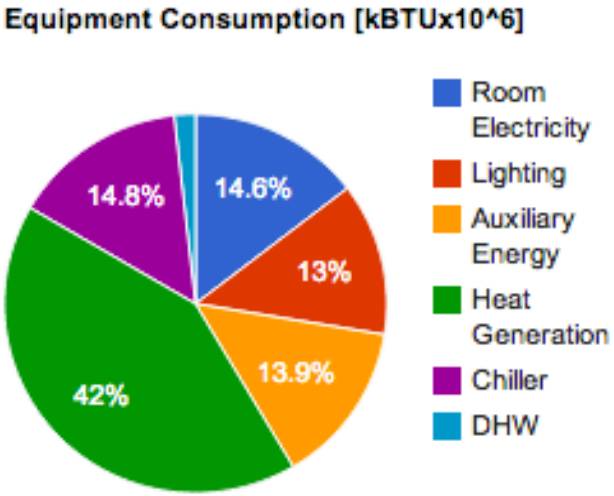
VirtualPULSE_run.rb

```ruby
require 'openstudio'
require 'VirtualPULSEModel'

#create a new model
model = VirtualPULSEModel.new

#add geometry (in this case a simple multi-story core/perimeter building)
model.add_geometry({"length" => 100,
#add windows at a given window-to-wall ratio
model.add_windows({"wwr" => 0.4,
#add HVAC - Packaged VAV w/ Reheat - DX Cooling, Hot Water heat and reheat
model.add_hvac({"fan_eff" => 0.5,
#add thermostats
model.add_thermostats({"heating_setpoint" => 24,
#assign constructions from a local library to the walls/windows/etc. in the model
model.add_constructions({"construction_library_path" => "#{Dir.pwd}/VirtualPULSE_default_constructions.osm"})

#add space type from a remote library (BCL) to the model
model.add_space_type({"NREL_reference_building_vintage" => "ASHRAE_90.1-2004",
#add design days to the model
model.add_design_days()

#save the OpenStudio model (.osm)
model.save_openstudio_osm({"osm_save_directory" => Dir.pwd,
#translate the OpenStudio model (.osm) to an EnergyPlus model (.idf)
model.translate_to_energyplus_and_save_idf({"idf_save_directory" => Dir.pwd,
#run the EnergyPlus model (.idf)
VirtualPULSEModel::run_energyplus_simulation({"idf_directory" => Dir.pwd,
```

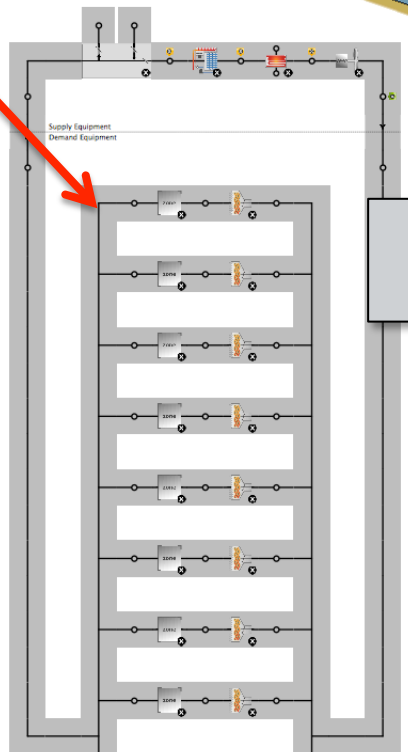
~3 Lines of Ruby Code + Comments

Geometry and Space Type Definitions



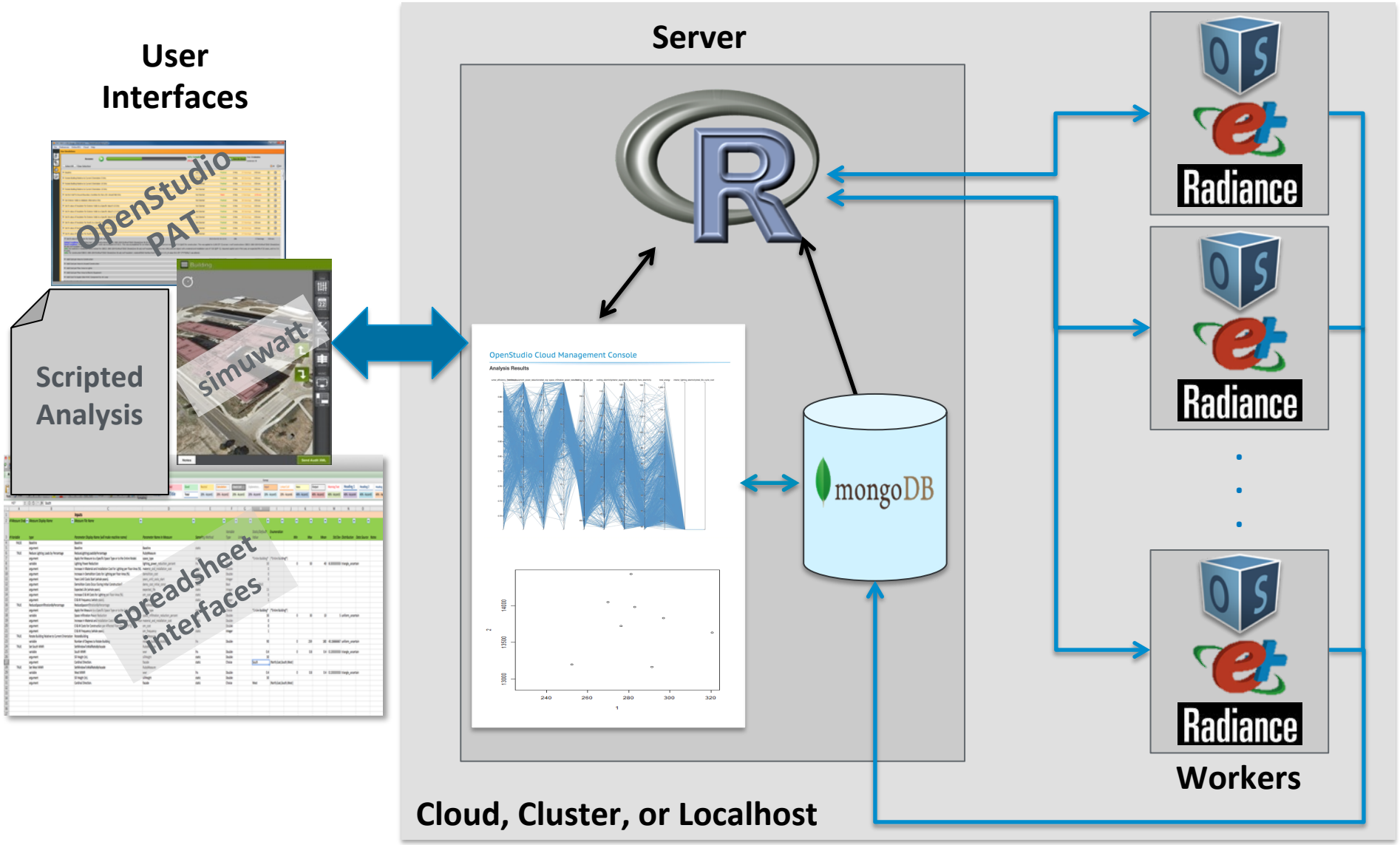


Simulated End Uses



Detailed HVAC and Zoning

…and Cloud-Based Analysis



Yes, but…

- What's that third one there?
- Well, that's Radiance.
- Where's that fit in?
- Oh, yeah…

Radiance Measure

Daylight simulation with Radiance is now available as an OpenStudio measure!

```
# see the URL below for information on how to write OpenStudio measures:
# http://nrel.github.io/OpenStudio-user-documentation/measures/measure_writing_guide/

# design document for this measure is at https://docs.google.com/document/d/16_TLRuhc4VFs2o0gRAp81hRObet7-s6fUEWo3HO7LpE/edit#

require 'fileutils'
require 'csv'
require 'tempfile'
require 'date'

# start the measure
class RadianceMeasure < OpenStudio::Ruleset::ModelUserScript

  # human readable name
  def name
    return "Radiance Daylighting Measure"
  end

  # human readable description
  def description
    return "This measure uses Radiance instead of EnergyPlus for daylighting calculations with OpenStudio."
  end

  # human readable description of modeling approach
  def modeler_description
    return "The OpenStudio model is converted to Radiance format. All spaces containing daylighting objects (illuminance map, daylighting
  end

  # define the arguments that the user will input
  def arguments(model)
    args = OpenStudio::Ruleset::OSArgumentVector.new

    chs = OpenStudio::StringVector.new
    chs << 'Yes'
    chs << 'No'
    apply_schedules = OpenStudio::Ruleset::OSArgument::makeChoiceArgument('apply_schedules', chs, true)
    apply_schedules.setDisplayName('Apply schedules')
    apply_schedules.setDefaultValue('Yes')
    apply_schedules.setDescription('Replace lighting and shading control schedules with schedules computed by Radiance')
    args << apply_schedules
```

Radiance Measure

By moving the Radiance simulation workflows to the OS Measure Framework, users can use Radiance for/with:

- **GUI-based parameter space creation (PAT)**
- **Large-scale analysis (OpenStudio-Server)**
- **Simulations on local systems or on the cloud (AWS, other)**

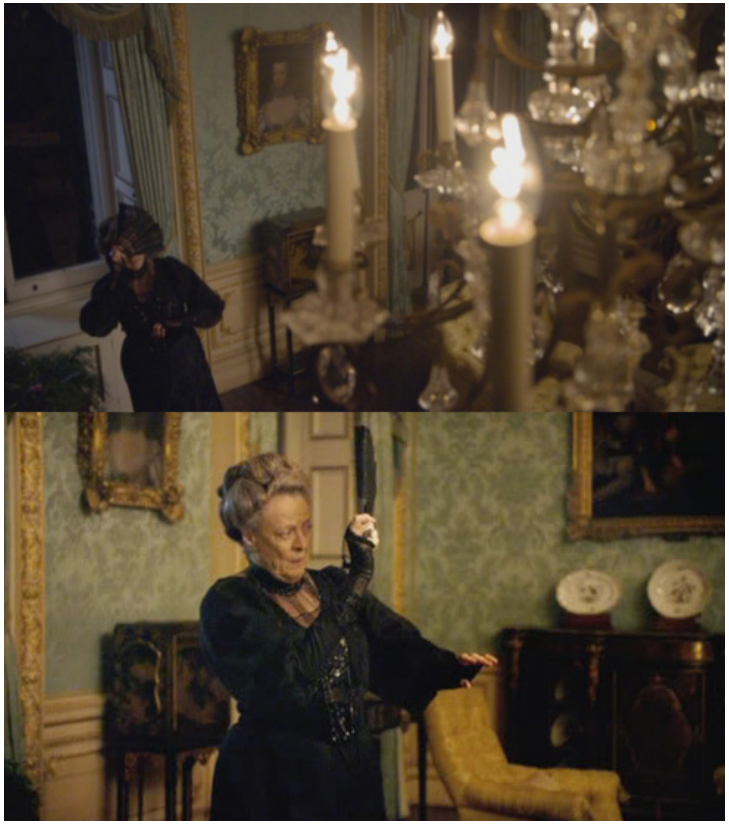
So what?

Evaluating daylight in a space ***constrained by an energy model's notion of architecture***, is a fool's errand!

Climate based daylight modeling and spatial daylight distribution form the foundation of all modern "dynamic daylight metrics", which are here to stay.

And real buildings have ***people*** in them!

Modern metrics require better tools, both for application and refinement



"Oh, dear, such a glare!"

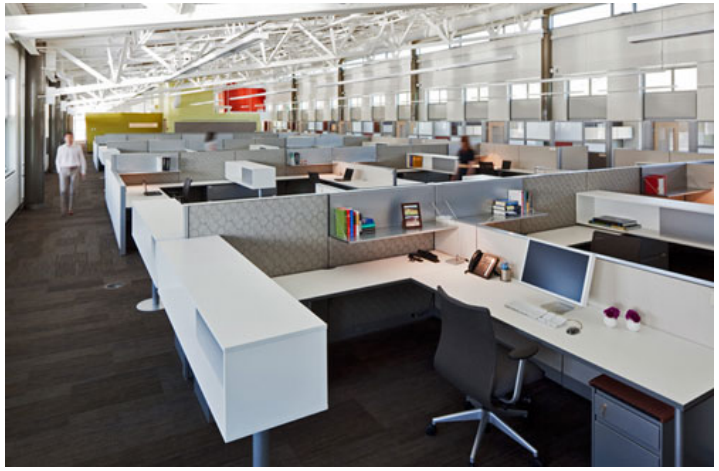
- Violet, Dowager Countess of Grantham, offering her opinion of small-size, intense light sources

Why Radiance?

Real buildings are partitioned by space use, not by HVAC zone!

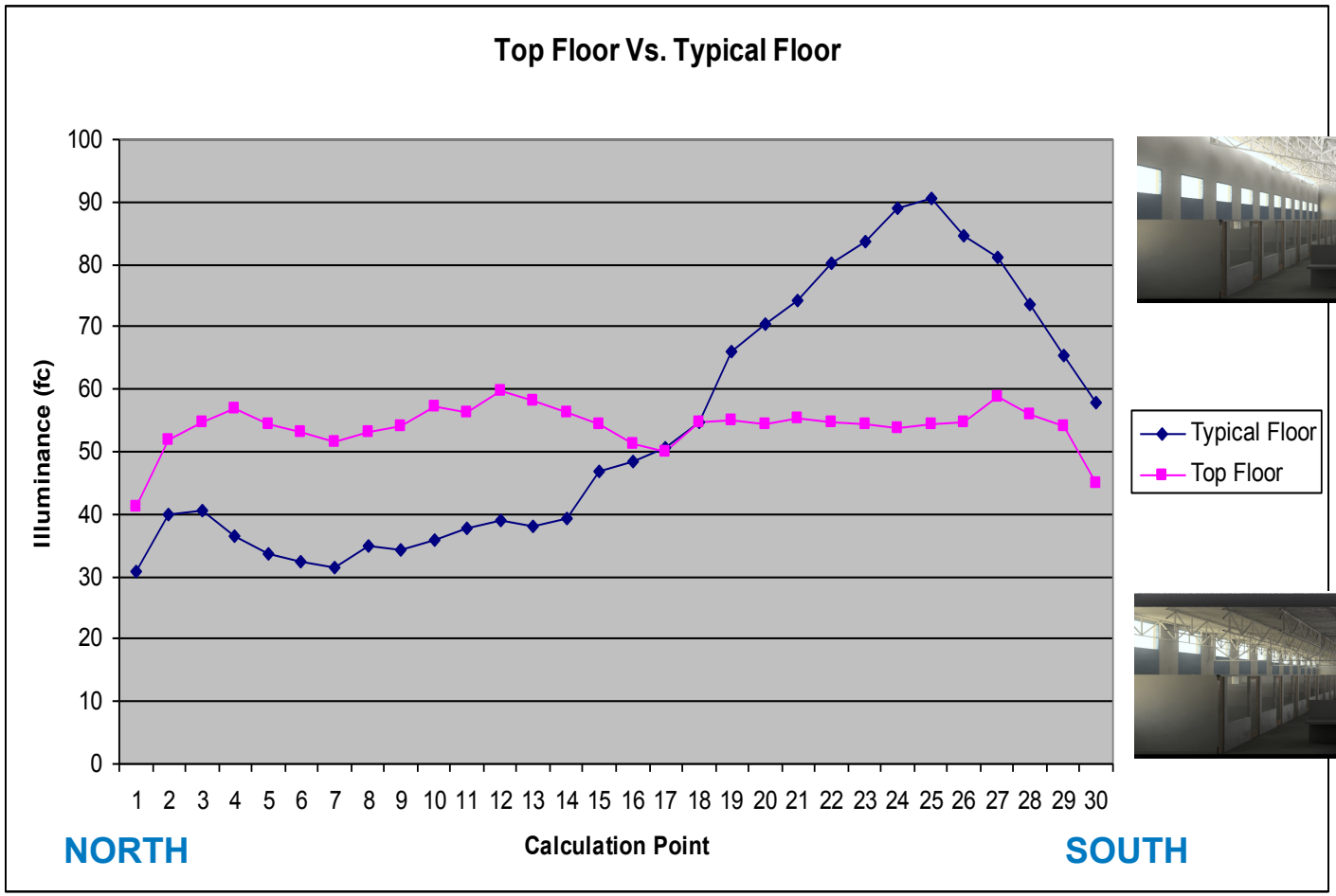
EnergyPlus' sense of interior architectural space has no relationship to the actual interior architecture!

Radiance and the OpenStudio model are not constrained by thermal zone boundaries



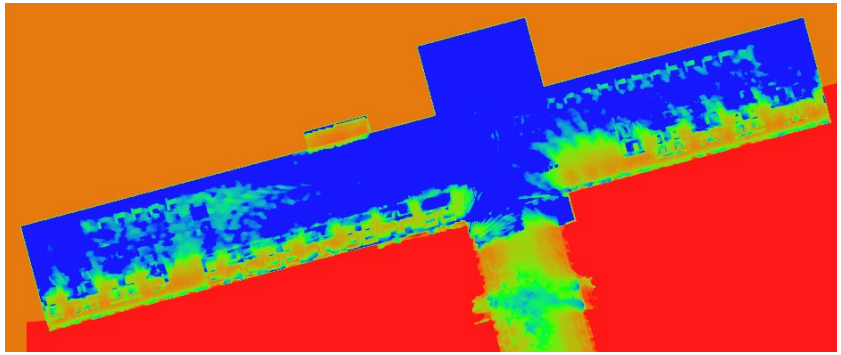
EnergyPlus Daylight Model

Radiance allows Architecturally Honest Daylight Simulation in BEM

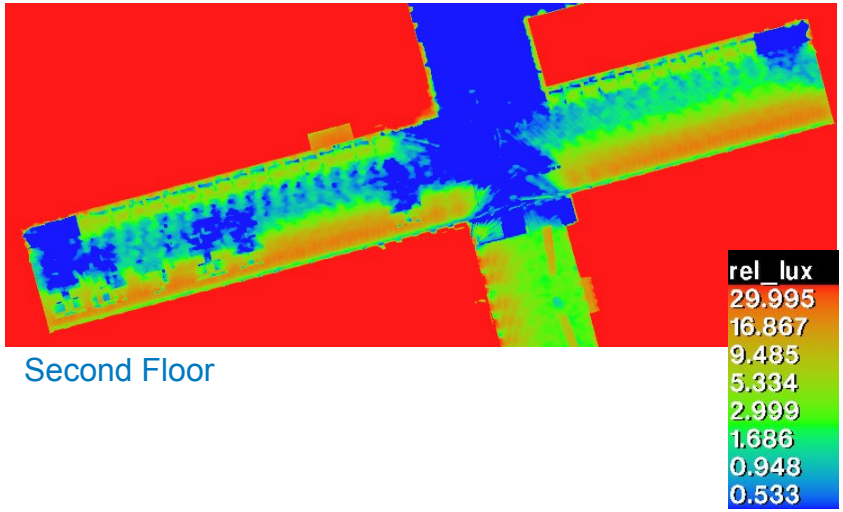


Top vs. Typical Floor Illuminance Distribution

Architecturally Honest Simulation



First Floor



Second Floor

Why (not) Radiance?

The OpenStudio approach to Radiance is to provide better quality, climate-based, spatio-temporal daylighting data to inform:

- **General** daylight metrics (UDI! UDI! UDI!)
- Lighting energy savings potential and schedules for a **building energy model (BEM)**

LM-83 is a good start, but is already under scrutiny and revision proposal. Also requires lengthy calculations and detailed CFS data.

NREL has leveraged the OS Model and SDK for **integrated** daylight and building energy modeling, but more detailed daylight modeling, metrics reporting, and tighter linkage with EnergyPlus model objects is needed (and currently not funded) =(
</rant>

OpenStudio/Radiance 3-Phase Workflow

OpenStudio -> Radiance "Forward Translator":

- **Converts model** from OpenStudio (.osm) to Radiance project:
  - Geometry
  - Materials
  - Analysis objects (illuminance grids, lighting control and glare analysis points)
  - Weather (.epw)
  - Schedules (lighting/equipment loads, occupancy, etc)
  - Views

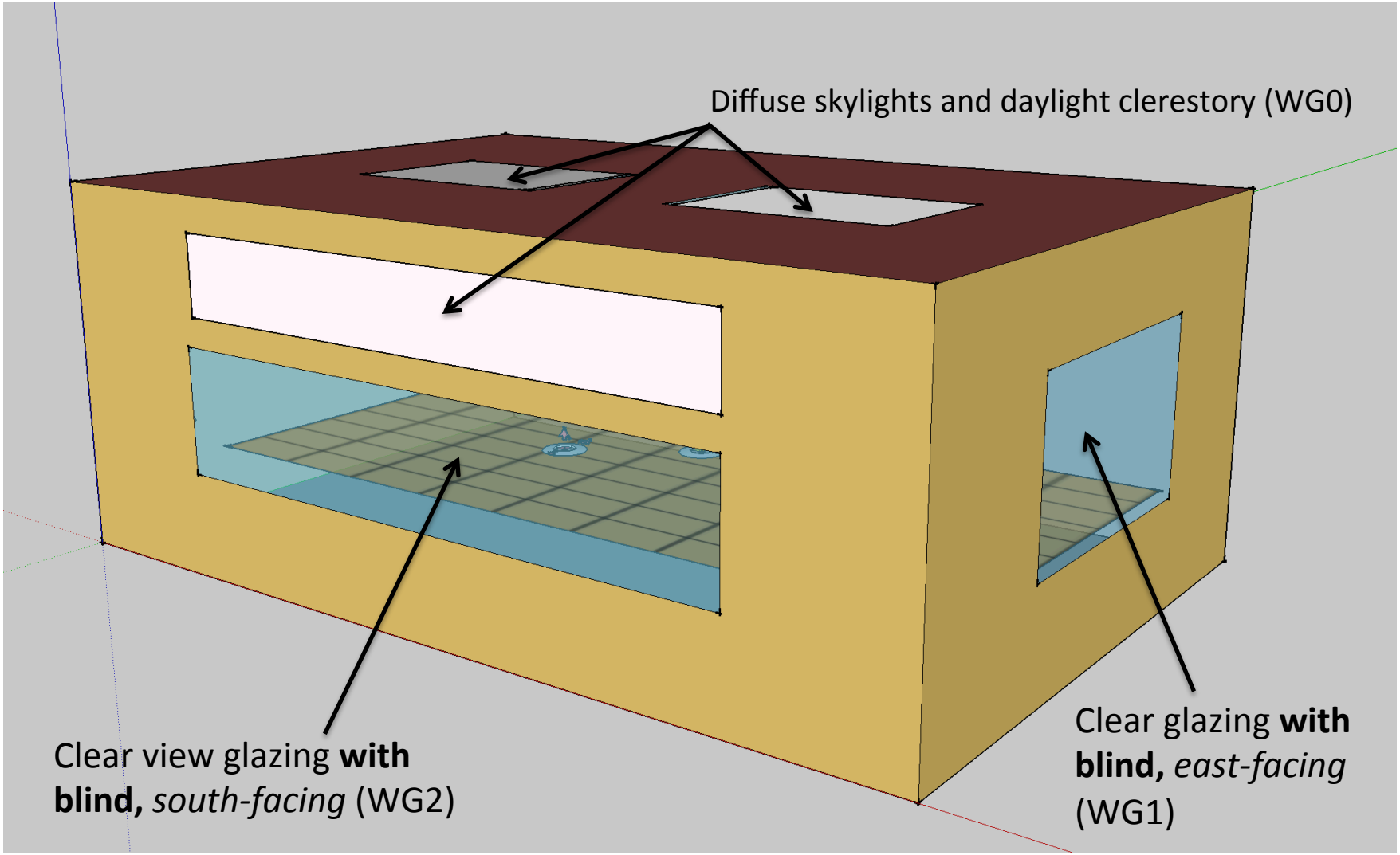
OpenStudio/Radiance 3-Phase Workflow

OpenStudio -> Radiance "Forward Translator":

- **Performs automatic window grouping:**
  - Uncontrolled windows are placed together, calculated as single phase
    - Single phase method
  - Controlled windows are logically grouped by:
    - Space
    - Orientation
    - VLT
    - CFS
      - OpenStudio v.1.9.0 will ship with BSDFs for:
        - Air! (allows to compute all view matrices simultaneously)
        - Blinds
        - Shadecloth
        - Daylight redirecting louvers

Window Grouping Example

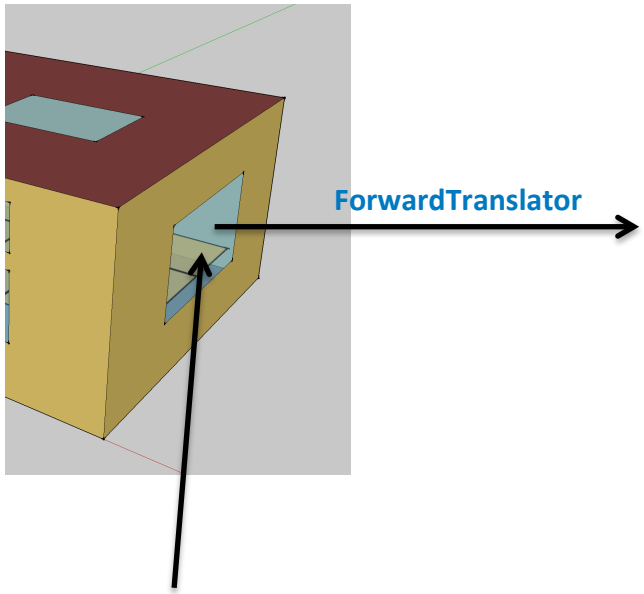
Each window group receives an exterior solar gain sensor for shading control input ("auto shades")



OpenStudio Rfluxmtx Integration

OS::Radiance::ForwardTranslator

- Adds necessary headers for rfluxmtx
- ***Fully automates*** 3-phase matrix generation



Clear glazing **with blind,** *east-facing* (WG1)

```
# OpenStudio Window Group: WG1
#@rfluxmtx h=kf u=Z o=output/dc/WG1.vmx

# SubSurface = East_Window
# Tvis = 0.25 (tn = 0.27)
WG1 polygon East_Window
0
0
12
8.561387500000031 4.110043965327130 2.504345203724800
8.561387500000031 4.110043965327130 0.973995203724805
8.561387500000031 1.633885650643690 0.973995203724805
8.561387500000031 1.633885650643690 2.504345203724800
```

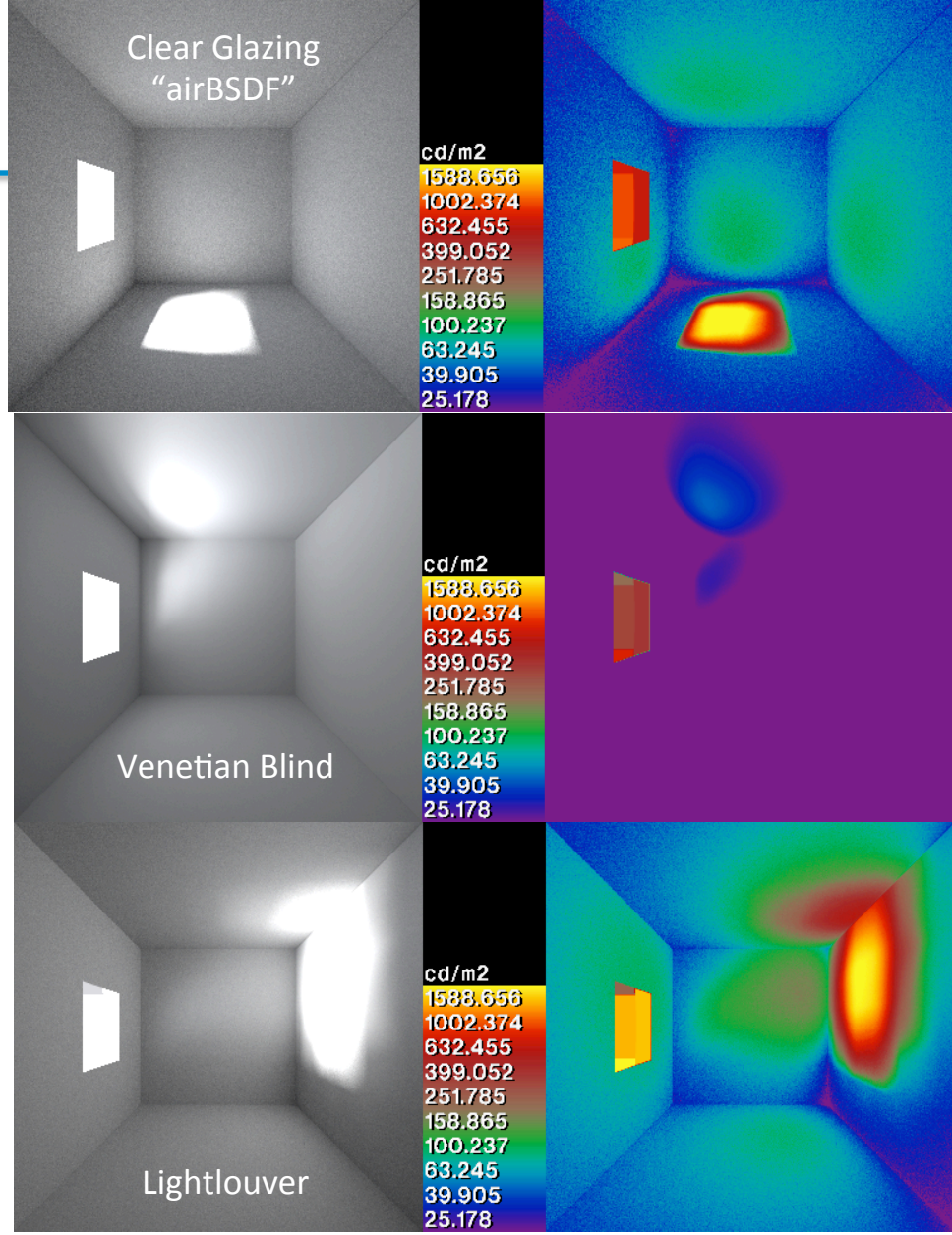
For (much) more on using rfluxmtx, see Andy McNeil's Presentation from the 2014 Radiance International Workshop:
http://www.radiance-online.org/community/workshops/2014-london/presentations/day1/McNeil_BSDFsandPhases.pdf

CFS Support

OpenStudio 1.9.0 will include BSDFs for detailed simulation of complex fenestration systems (CFS):

- Venetian Blinds
- Shadecloth (e.g. Mechoshade)
- Daylight redirecting devices (e.g. Lightlouver)

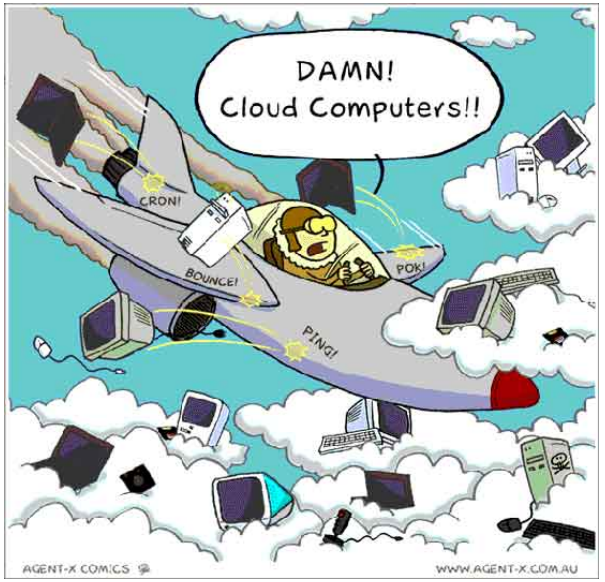
OpenStudio will allow you to use BSDFs as easily as any standard glazing material

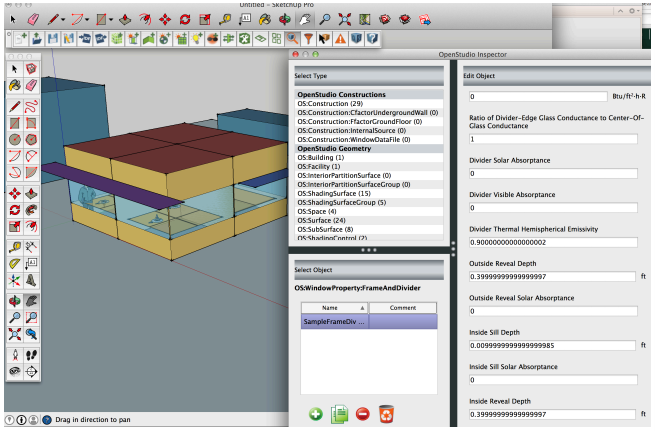


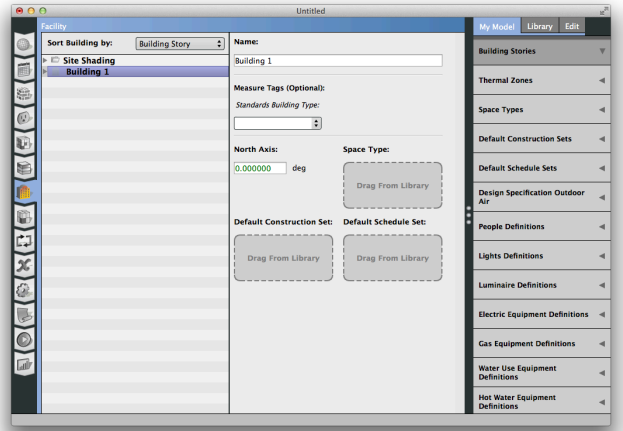
Additional info available at:
http://www.rumblestrip.org/2014/06/03/sanity-testing-bsdfs/ ← Probably NSFW, just sayin'.

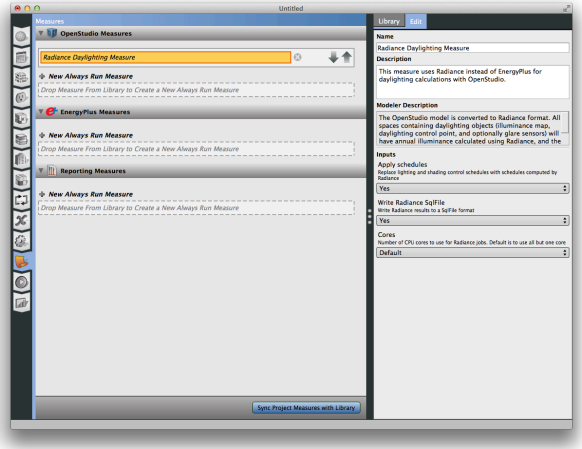
New OpenStudio/Radiance Workflow

1. Create (or import) model in SketchUp
2. Finalize settings in OpenStudio Application (OSApp)
3. Run single model instance in OSApp, and/or:
4. Create parameter space in PAT and run zillions of models:
   - on local system,
     or:
   - in the CLOUD!



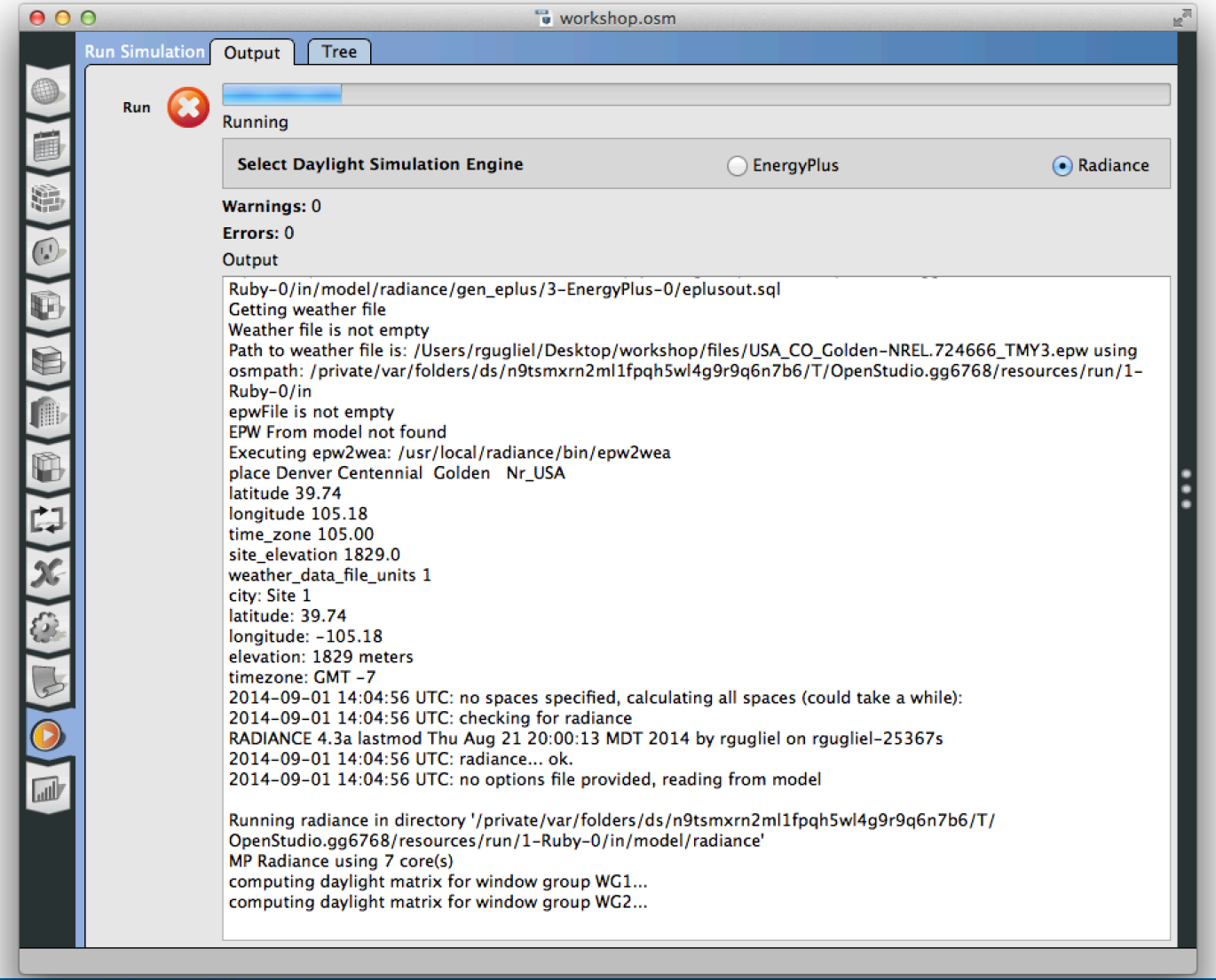






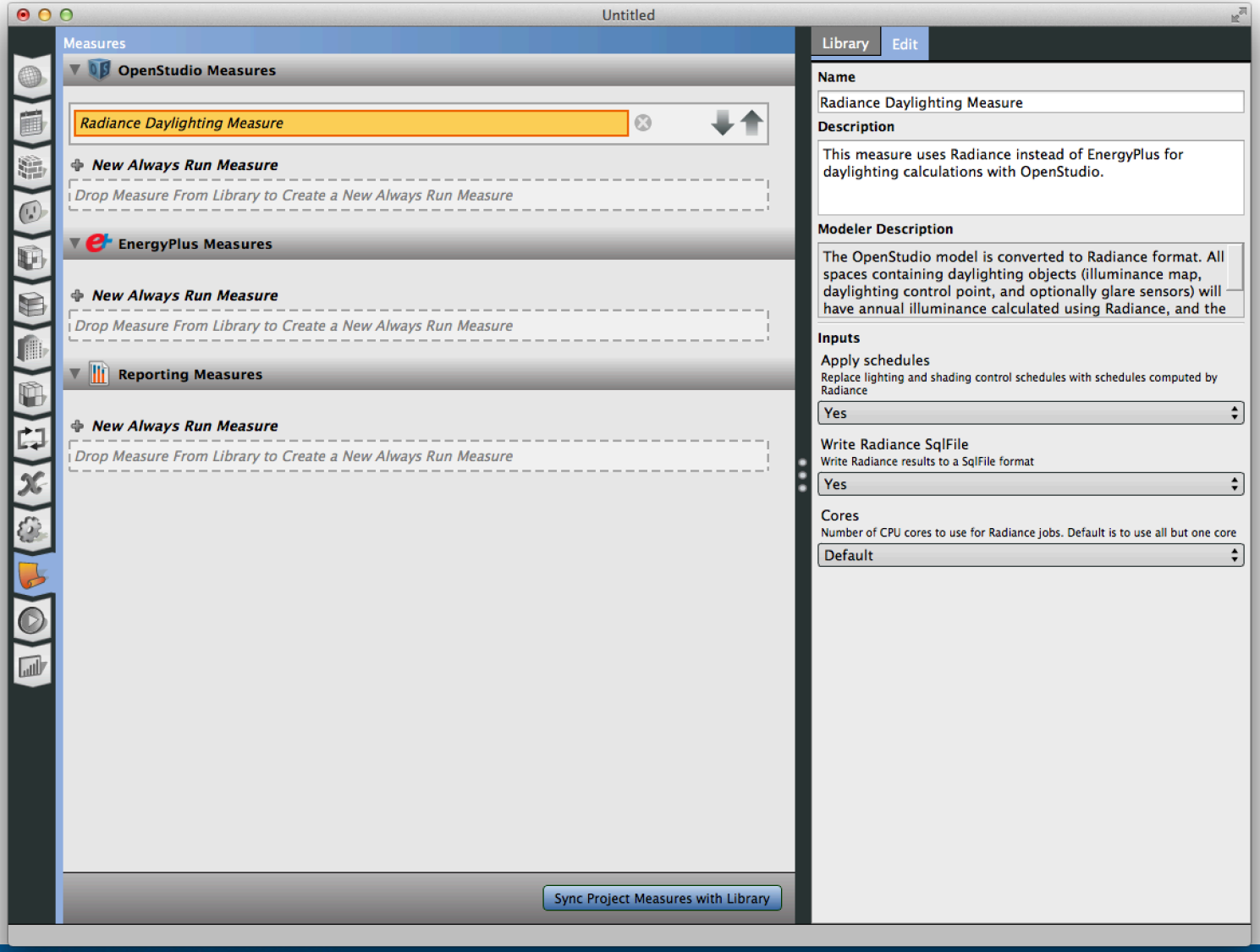
New OpenStudio/Radiance Workflow

Simulation engine select radio buttons are gone…



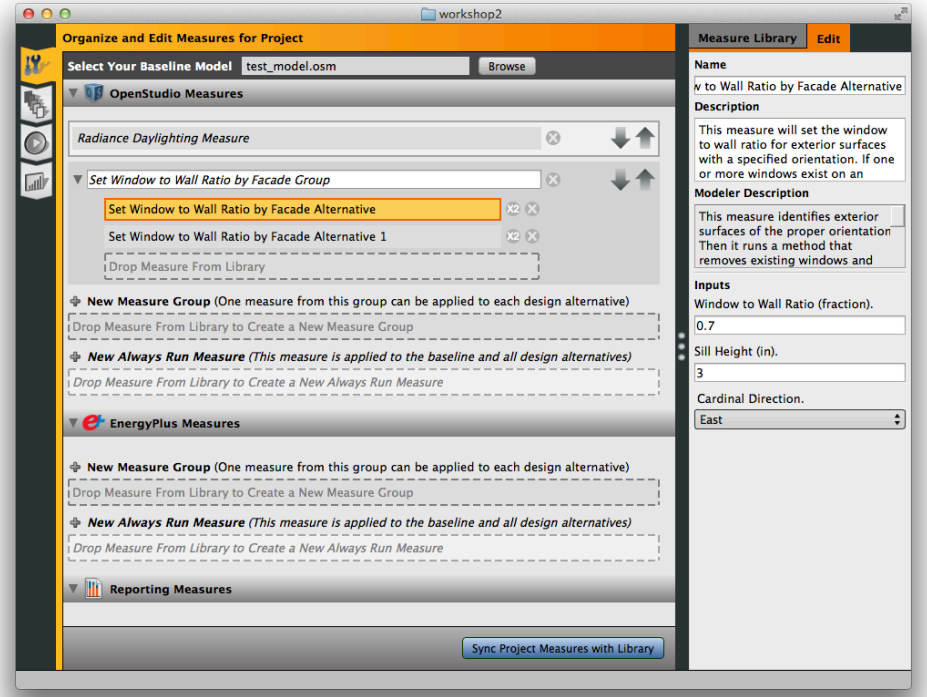
New OpenStudio/Radiance Workflow

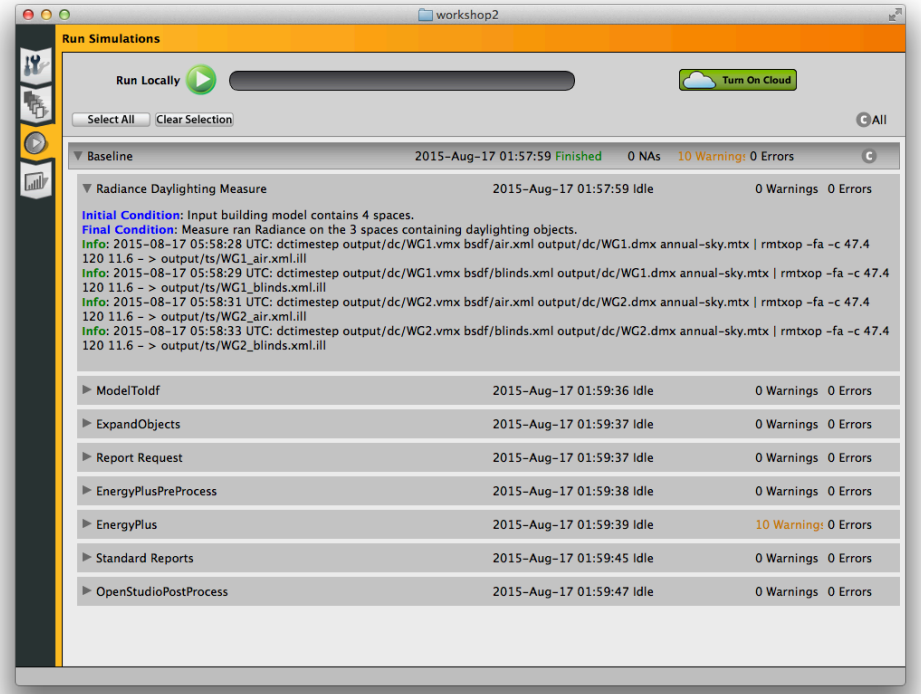
User now adds the Radiance Daylighting Measure to the "stack"



New OpenStudio/Radiance Workflow

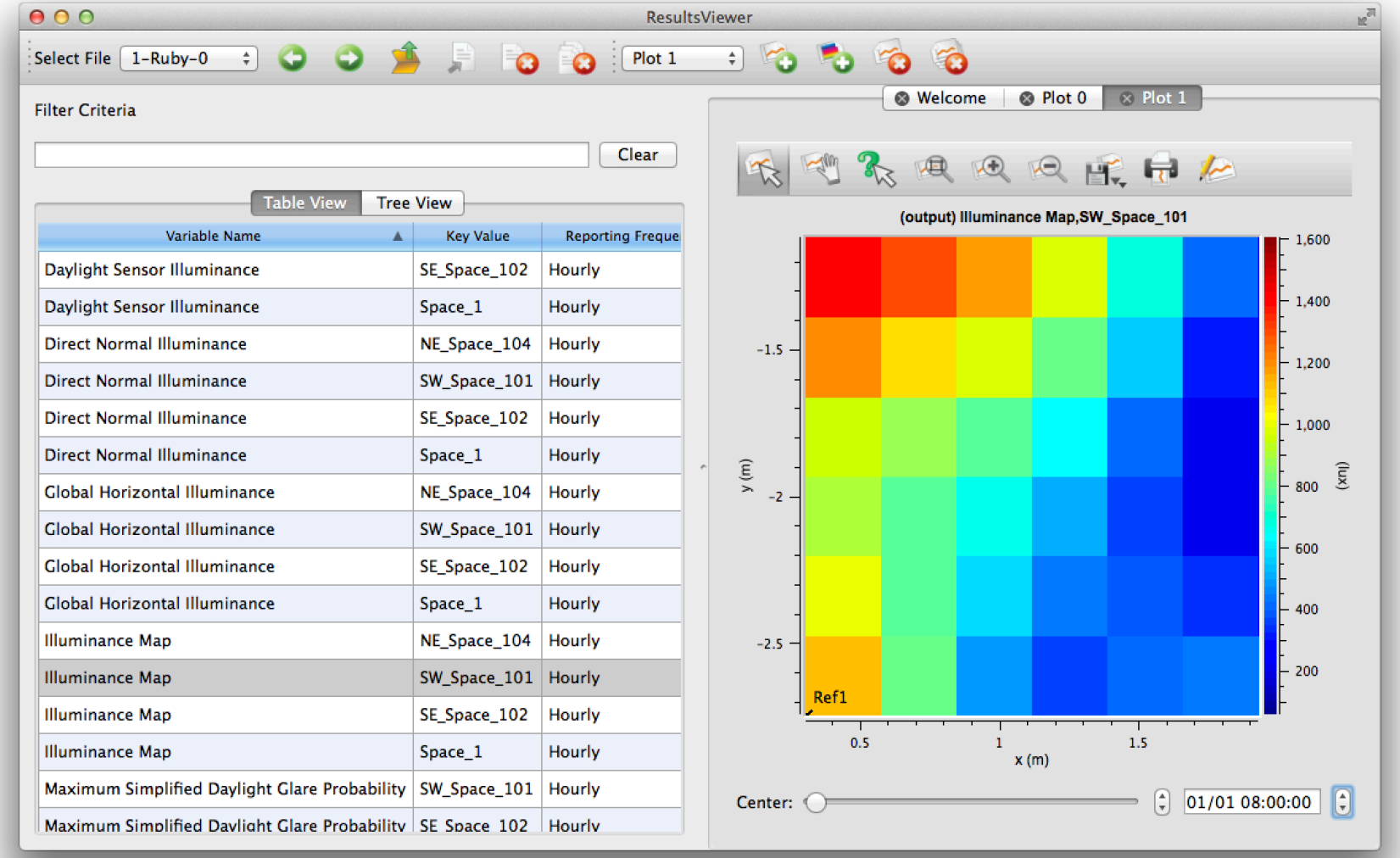
…and optionally loads in PAT for parametric analysis





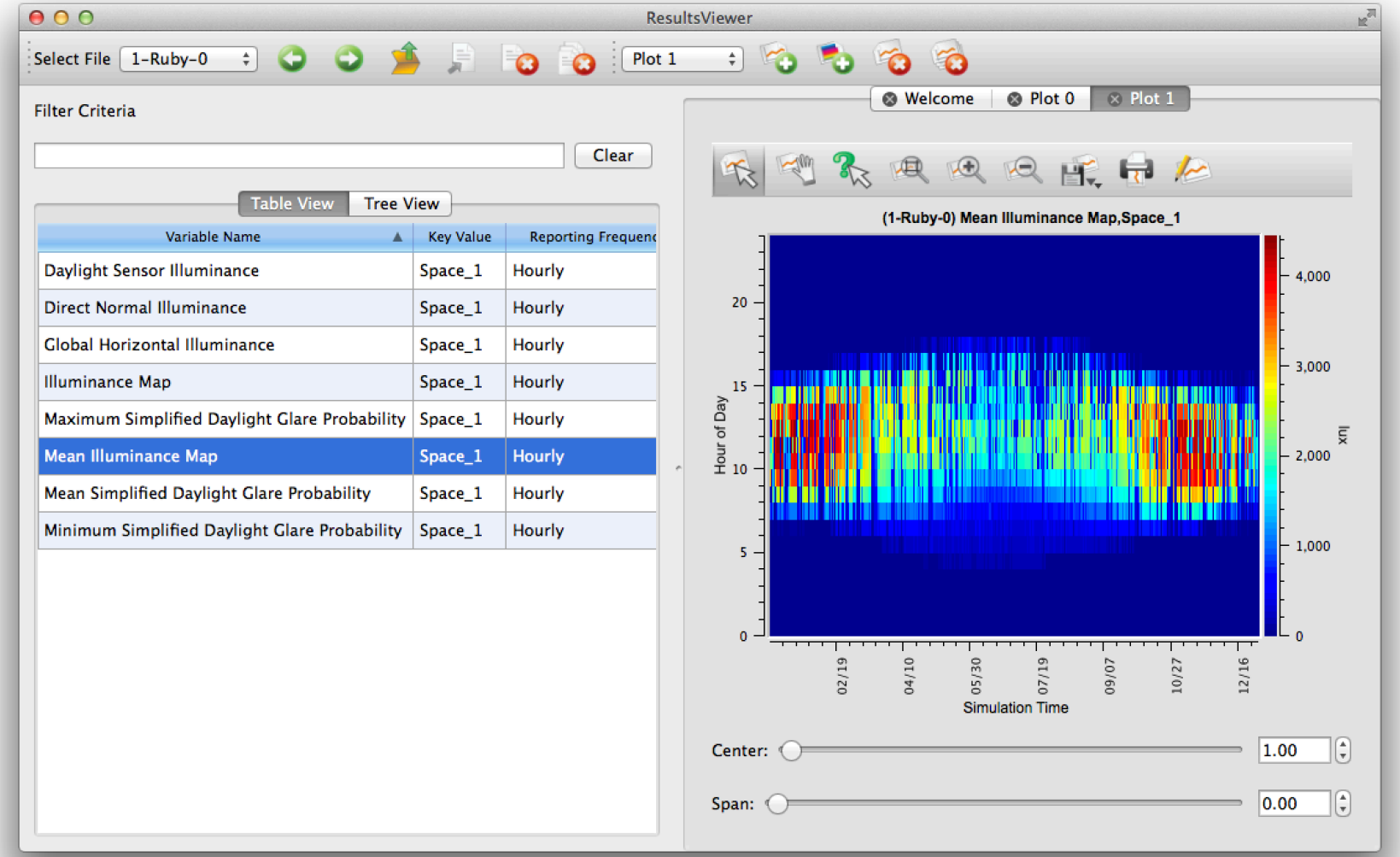
New OpenStudio/Radiance Workflow

Individual space illuminance maps are saved in sql format…



New OpenStudio/Radiance Workflow

Individual space illuminance maps are saved in sql format…



New OpenStudio/Radiance Workflow

…and daylight metrics are stored in CSV format

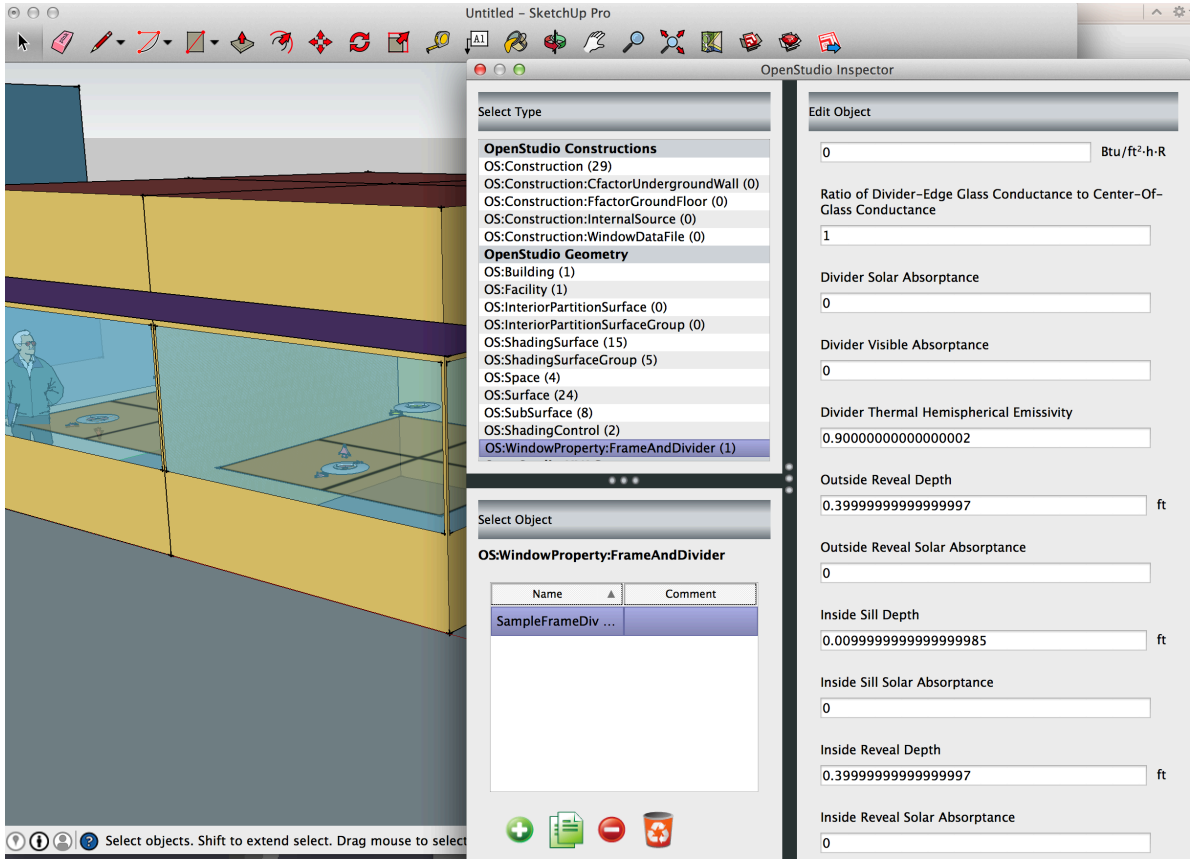
```
# OpenStudio Daylight Metrics Report
# Average daylight autonomy for building daylit spaces: 0.65
# Space data format: [space_name] [metric(setpoint)] [input_hours_range] [metric_value] [hours_met] [input_hours]
NE_Space_104,DA(500),Daylit Hours,0.66,3102,4710
NE_Space_104,DA(500),Occupied Hours,0.47,2481,5270
NE_Space_104,DA(500),Daylit and Occupied Hours,0.66,2475,3735
NE_Space_104,conDA(500),Daylit Hours,0.68,3197,4710
NE_Space_104,conDA(500),0.49,Occupied Hours2565,5270
NE_Space_104,conDA(500),Daylit and Occupied Hours,0.68,2549,3735
NE_Space_104,UDI(100-3000),Daylit Hours,0.45,2134,4710
NE_Space_104,UDI(100-3000),Occupied Hours,0.32,1707,5270
NE_Space_104,UDI(100-3000),Daylit and Occupied Hours,0.45,2549,3735
SW_Space_101,DA(323),Daylit Hours,0.63,2959,4710
SW_Space_101,DA(323),Occupied Hours,0.46,2412,5270
SW_Space_101,DA(323),Daylit and Occupied Hours,0.65,2411,3735
SW_Space_101,conDA(323),Daylit Hours,0.72,3413,4710
SW_Space_101,conDA(323),0.52,Occupied Hours2761,5270
SW_Space_101,conDA(323),Daylit and Occupied Hours,0.74,2750,3735
SW_Space_101,UDI(100-3000),Daylit Hours,0.67,3172,4710
SW_Space_101,UDI(100-3000),Occupied Hours,0.48,2543,5270
SW_Space_101,UDI(100-3000),Daylit and Occupied Hours,0.68,2750,3735
SE_Space_102,DA(323),Daylit Hours,0.63,2953,4710
SE_Space_102,DA(323),Occupied Hours,0.45,2374,5270
SE_Space_102,DA(323),Daylit and Occupied Hours,0.63,2361,3735
SE_Space_102,conDA(323),Daylit Hours,0.71,3327,4710
SE_Space_102,conDA(323),0.51,Occupied Hours2675,5270
SE_Space_102,conDA(323),Daylit and Occupied Hours,0.71,2651,3735
SE_Space_102,UDI(100-3000),Daylit Hours,0.53,2499,4710
SE_Space_102,UDI(100-3000),Occupied Hours,0.38,2025,5270
SE_Space_102,UDI(100-3000),Daylit and Occupied Hours,0.54,2651,3735
```

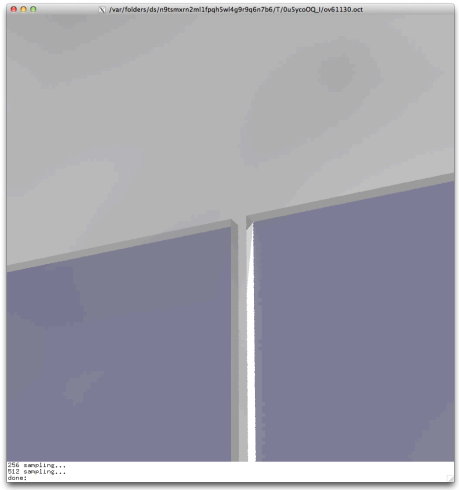

Why don’t you
show them? (demo)

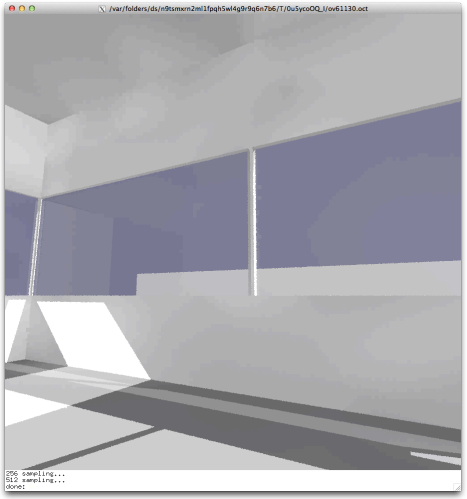
Wall Thickness

OS::Window::FrameAndDivider

- Support for wall thickness, mullions and framing
- Translates to Radiance model (and EnergyPlus)



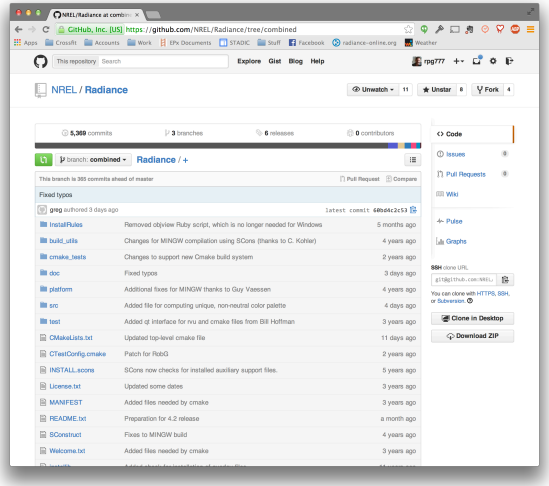


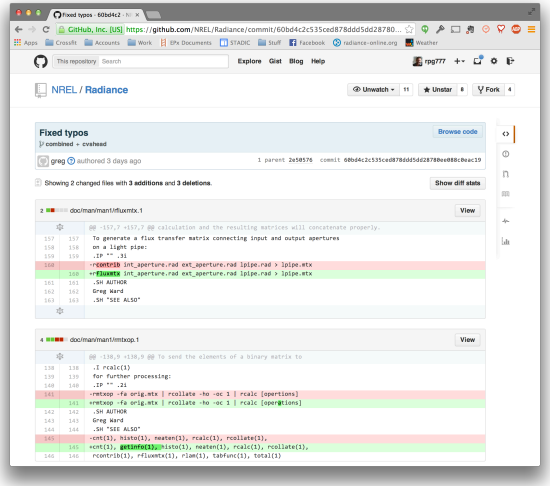


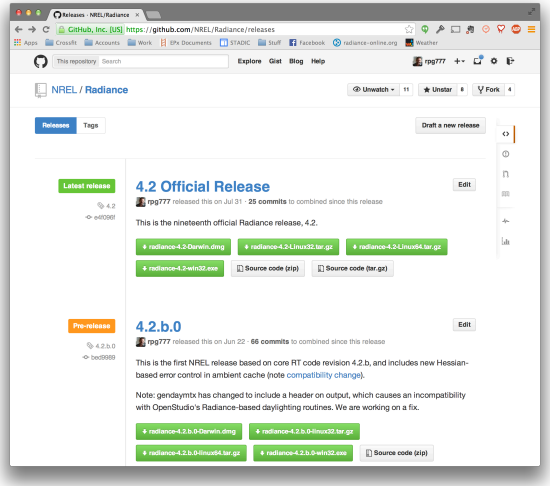
http://nrel.github.io/OpenStudio-user-documentation/tutorials/tutorial_windowproperty_frameanddivider/

Radiance on GitHub

- Mirror of Radiance CVS source code repository
- Refreshed every 15 minutes
- Full revision history
- **Use “combined” branch for latest history!**
- NREL installers (Windows, Mac, Unix)
- Release snapshots
- https://github.com/NREL/Radiance
- Thanks/fist-bumps to Nick Long (NREL), Zack Galbreath (Kitware), and US Department of Energy for this resource!



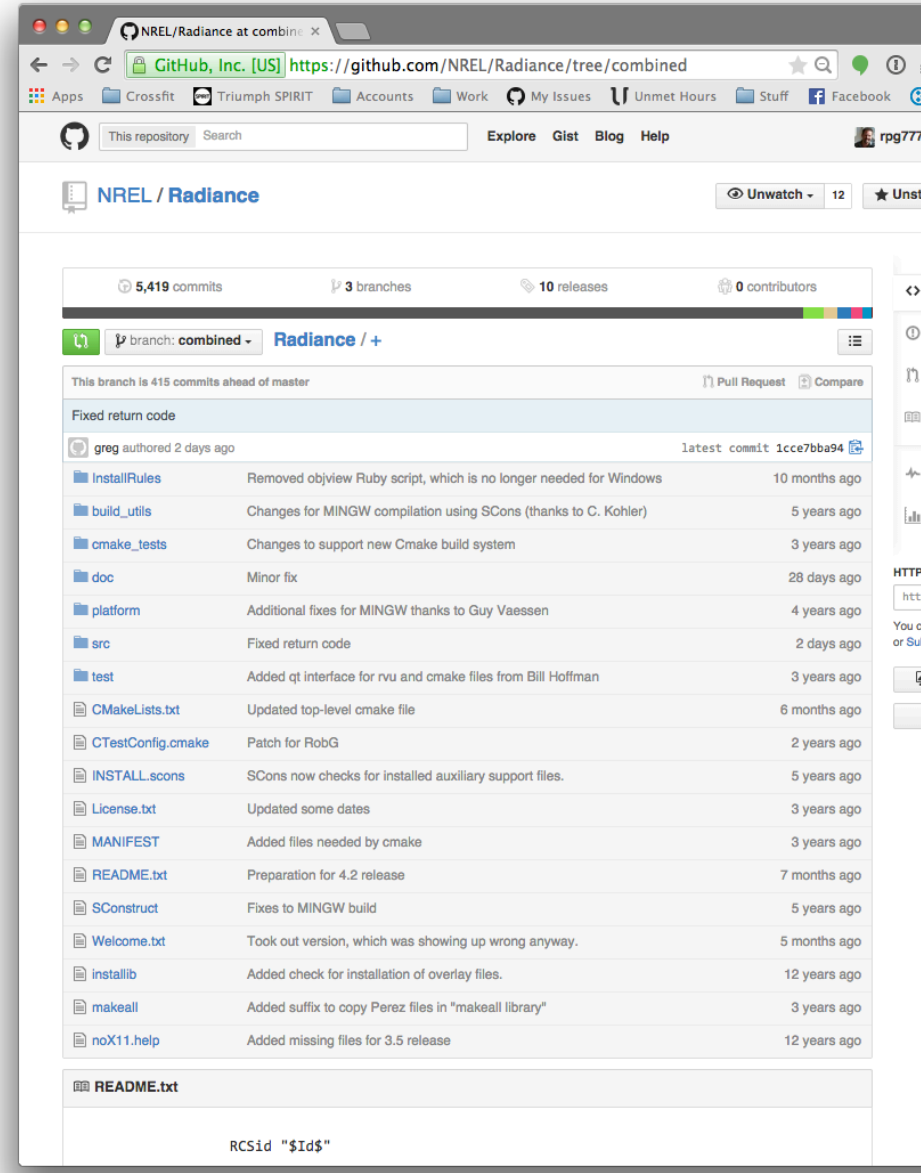




Radiance on GitHub

GitHub mirror of Radiance source repository

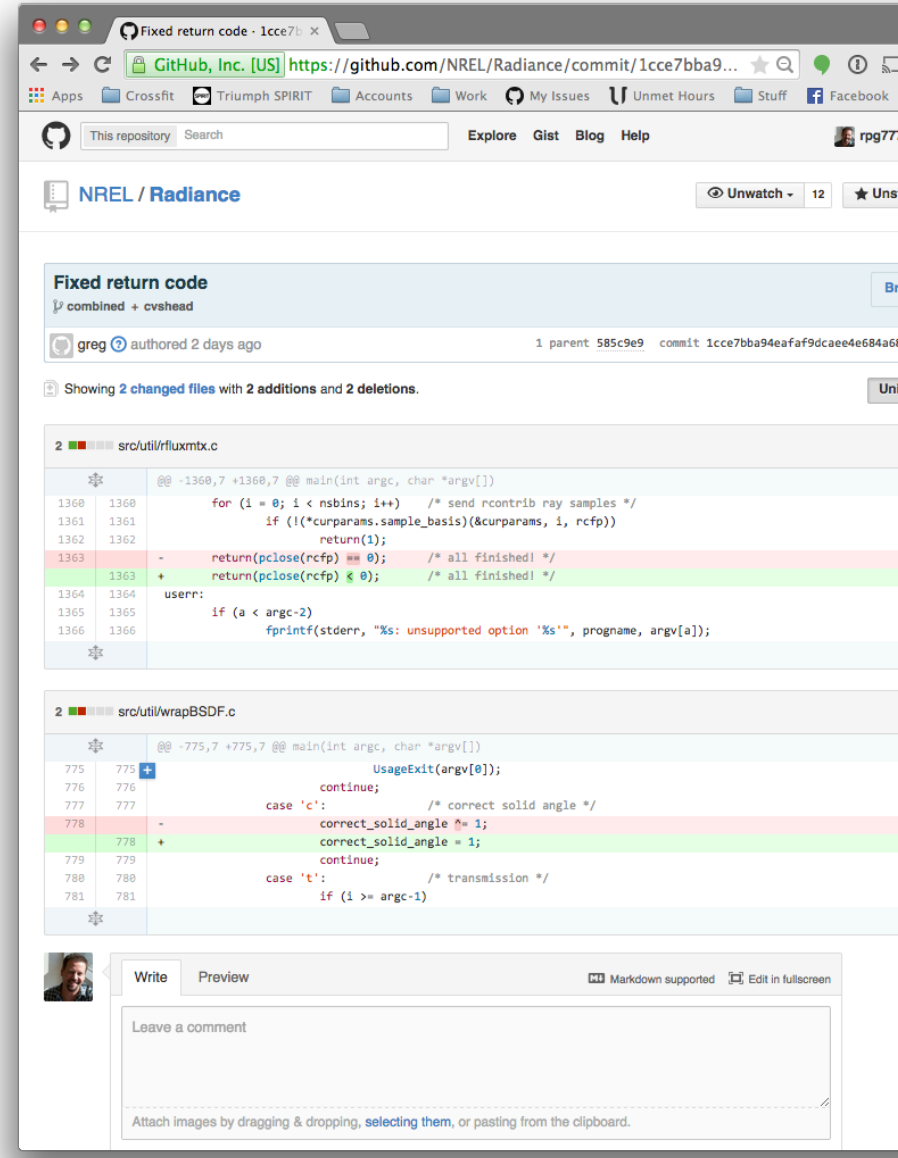
- **Easy access to Radiance source code**
- Release tags
- Revision history
- Installers (Packages)
- https://github.com/NREL/Radiance



Radiance on GitHub

GitHub mirror of Radiance source repository

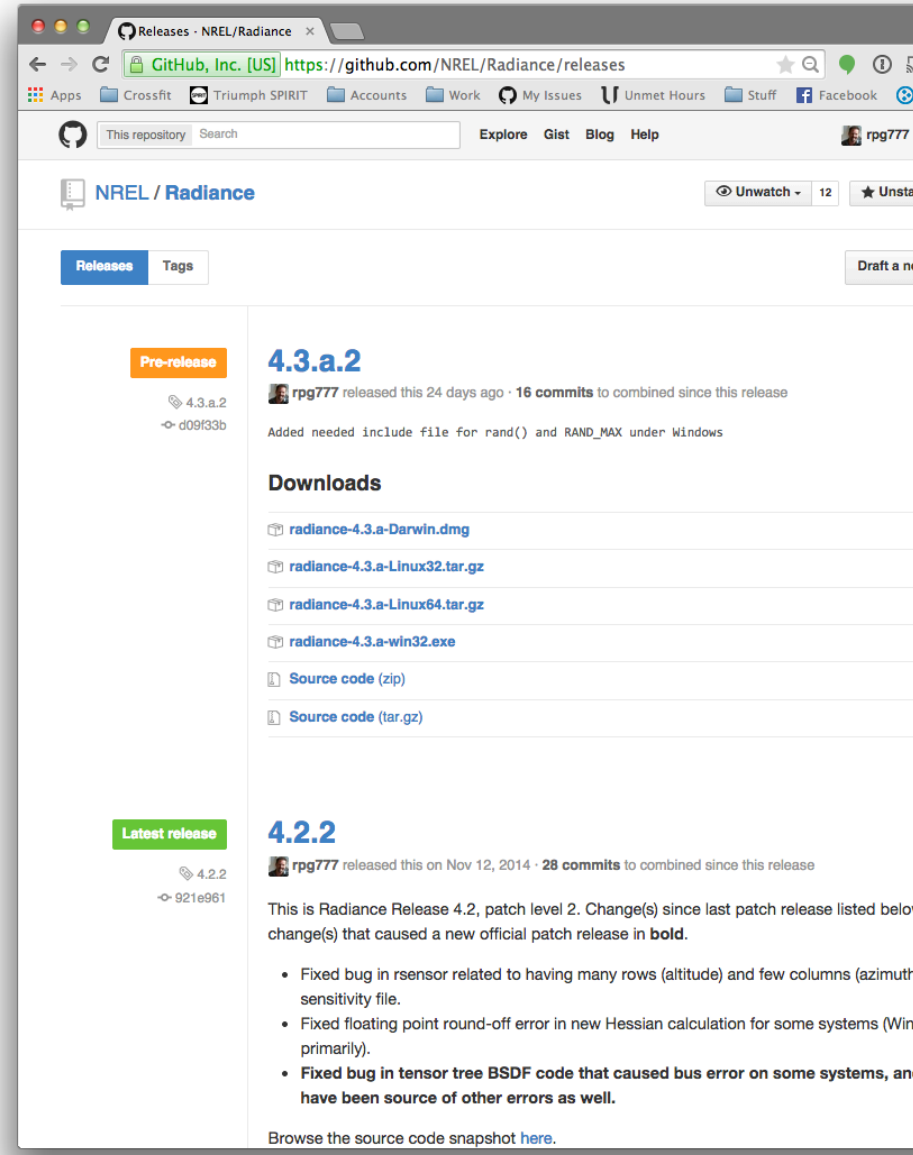
- Easy access to Radiance source code
- Release tags
- **Revision history**
- Installers (Packages)
- https://github.com/NREL/Radiance



Radiance on GitHub

GitHub mirror of Radiance source repository

- Easy access to Radiance source code
- Release tags
- Revision history
- **Installers (Packages)**
- https://github.com/NREL/Radiance



OpenStudio/Radiance Roadmap

For v1.9.0 (September 30, 2015)

- Radiance Measure
- Testing/validation
- Documentation

FY2016 (Begins October 1, 2015)

- Continued support of Radiance source code mirror on GitHub
- Cross platform Radiance installers
- Auto grid generation (either internal tool or CBEIs dxGridmaker)
- Data visualization
  - Report "Measure"
  - JavaScript/Angular.js, etc…
  - ?


How’s that
demo going? (woof.)