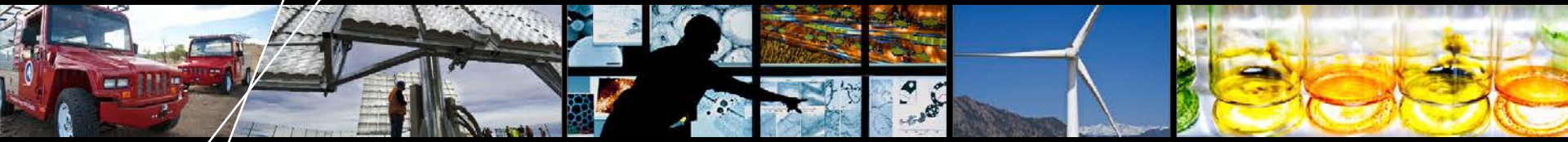
NREL
NATIONAL RENEWABLE ENERGY LABORATORY

SCRIPTED BUILDING ENERGY MODELING AND ANALYSIS



SimBuild 2012, Madison, WI

Daniel Macumber, NREL

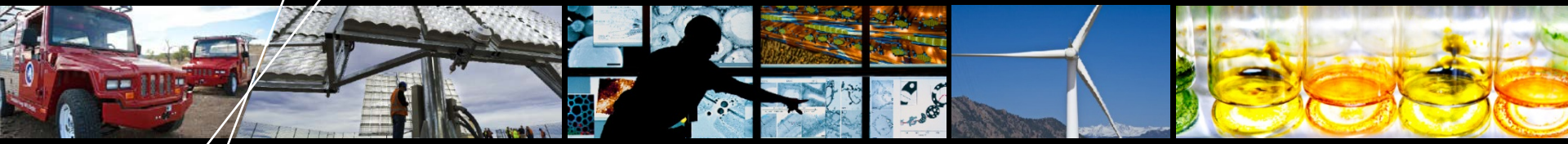
August 2, 2012

NREL/PR-5500-55863

Outline

- **Problem Statement and Solution**
- **Measures in OpenStudio**
  - What is an OpenStudio Measure?
  - Structured Rulesets
  - Freeform Scripts
- **Use Cases for OpenStudio Measures**
  - Interactive Measure Application
  - Noninteractive Measure Application
  - Composing Parameter Spaces With Measures
- **OpenStudio Measures and the BCL**
- **Conclusions**

NREL
NATIONAL RENEWABLE ENERGY LABORATORY



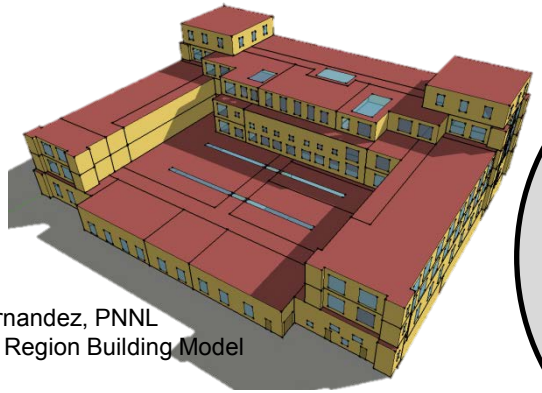
Problem Statement and Solution

Problem Statement and Solutions

- **Building energy analysis is often time-intensive, error-prone, and nonreproducible**
  - GUIs help
  - Libraries of data help
  - Still need to automate tedious workflows
- **Entire energy analyses can be scripted end to end using the OpenStudio Ruby API**
- **Common tasks within an analysis can be automated using OpenStudio Measures**

OpenStudio Approach

Credit: David Goldwasser / NREL

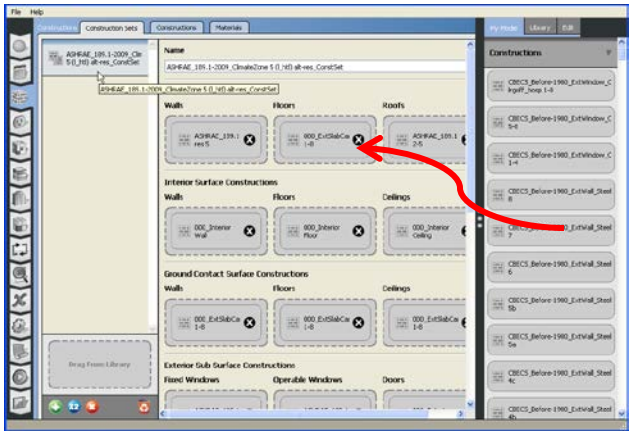


Credit: Nick Fernandez, PNNL
GSA Heartland Region Building Model

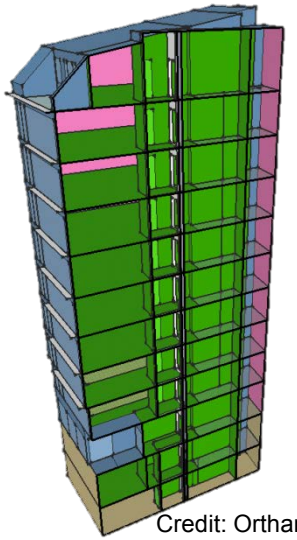
Use measures interactively from GUIs or noninteractively as part of model's simulation workflow

OpenStudio and SketchUp Plug-in

Model creation and inspection



Drag and drop components from BCL into current model



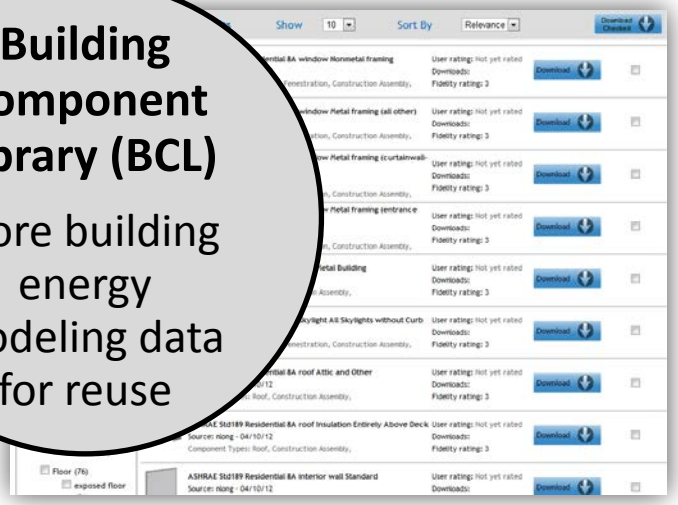
Credit: Orthan Veli Kazanci

OpenStudio Measures

Workflow automation and model transformations

Building Component Library (BCL)

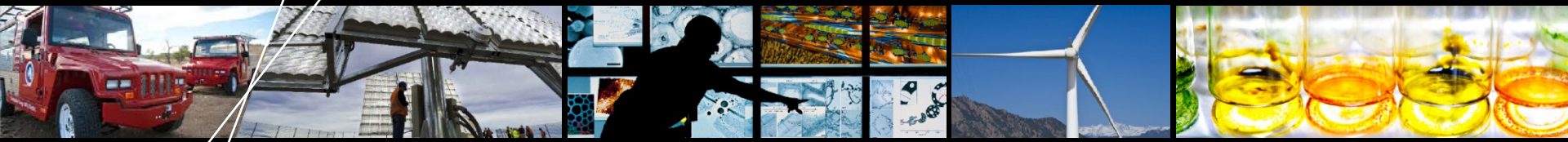
Store building energy modeling data for reuse



Credit: David Goldwasser / NREL

Share measures in BCL; measures may refer to BCL data

NREL
NATIONAL RENEWABLE ENERGY LABORATORY



Measures in OpenStudio

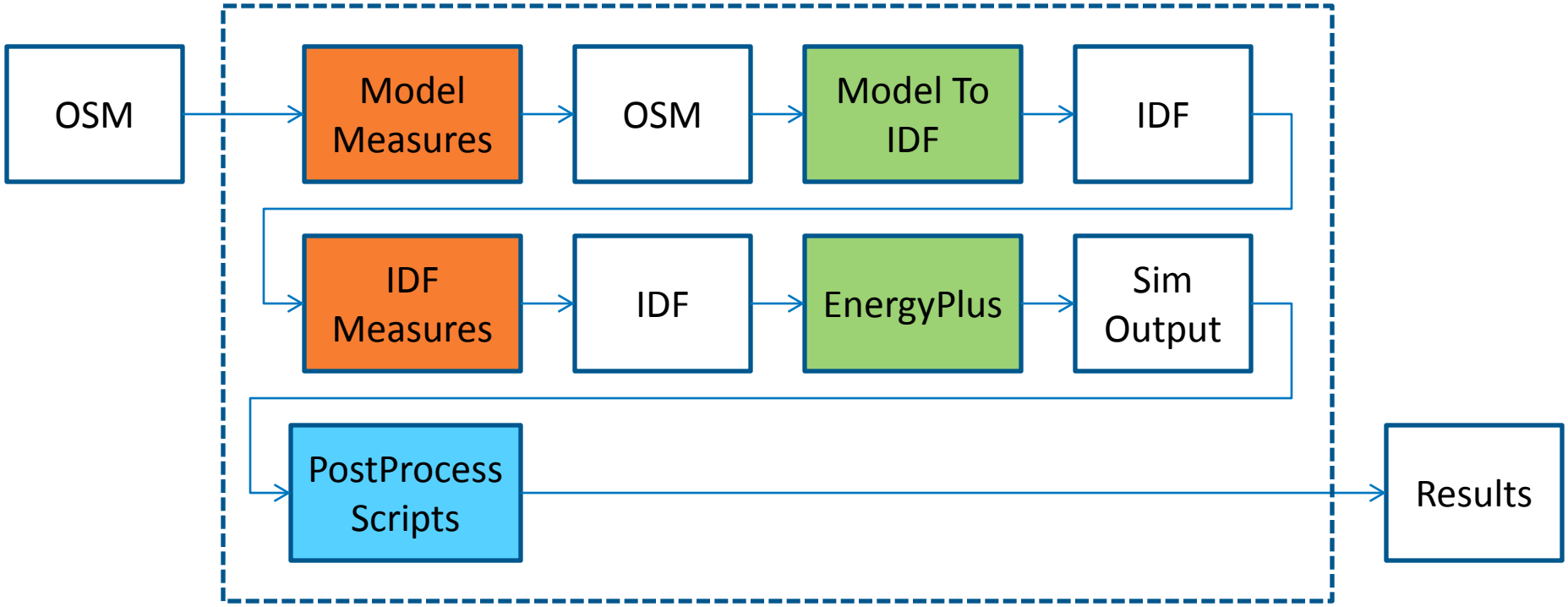
Measures in OpenStudio

- **What is a "Measure" in OpenStudio?**
  - Repeatable transformation of an input building energy model into an output energy model

- **Examples of OpenStudio Measures**
  - Set window to wall ratio to 40% on all façades
  - Replace all T-12 light fixtures with T-5 fixtures
  - Improve fan efficiency from 50% to 70%
  - Import IDF text snippets and set object names

Measures in OpenStudio

- **Users may drag and drop multiple Measures at two points in the OpenStudio simulation workflow**



Workflow automatically applies Measures, runs simulation, and performs postprocessing

Measures in OpenStudio

- **Measures may be written generically to be reused across many building models or written for a specific building model**
- **Measures may take input parameters**
  - WindowToWallRatio Measure takes arguments ‘wwr’, ‘offset’, and ‘application_type’
- **Types of OpenStudio Measures**
  - Structured Rulesets
  - Freeform Scripts

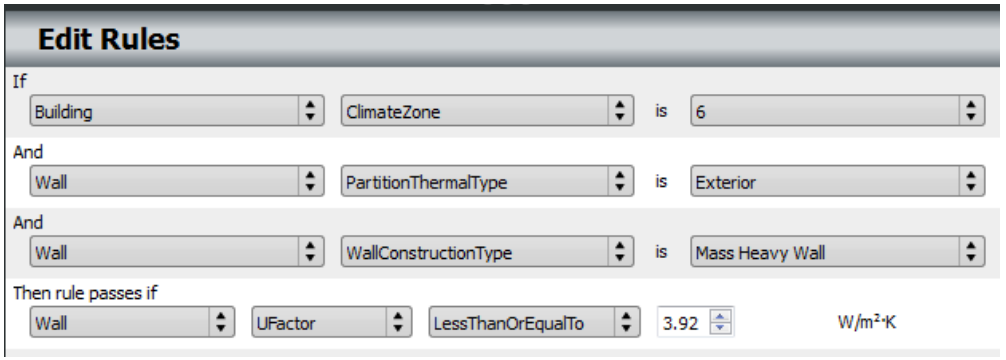
Structured Ruleset

- Attributes allow OpenStudio ModelObject methods to be called through a string key at runtime
  - Surface::windowToWallRatio() == getAttribute("windowToWallRatio")
- Some attributes are read only, others allow read/write
- A Ruleset is composed of Rules
- A Rule is composed of Filters and Actions (there are many types of filters and actions)
- Attribute filters pass or reject objects based on attribute values
- Attribute actions change objects by calling setAttribute
- Structured Ruleset can be serialized to XML format

Example Rule to set WWR on exterior walls:

Filter 1	If type is "OS:Surface"
Filter 2	If getAttribute("surfaceType") is "Wall"
Filter 3	If getAttribute("outsideBoundaryCondition") is "Outdoors"
Action 1	Then setAttribute("windowToWallRatio", 0.4)

Initial Structured Ruleset GUI:



Credit: Elaine Hale/ NREL

Freeform Script

- User defines Ruby class
- Class must have methods ‘name’, ‘arguments’, and ‘run’
- Run method can use any method exposed in the OpenStudio Ruby API
- Freeform Script serialized as Ruby script (text file)

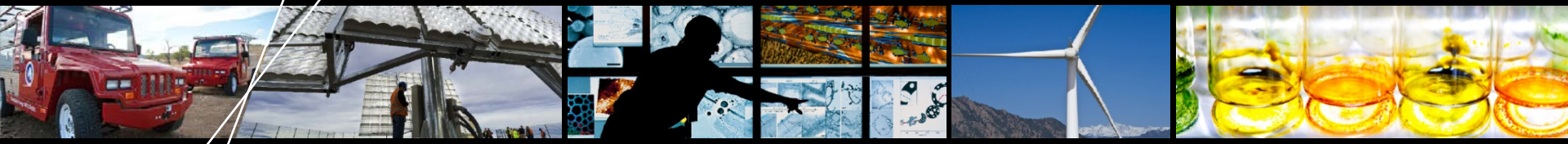
Example code to set WWR on exterior walls:

```ruby
def run(model, runner, arguments)
 wwr = arguments["wwr"]
 offset = arguments["offset"]
 application_type =
     arguments["application_type"]

 heightOffsetFromFloor = nil
 if (application_type.valueAsString ==
     "Above Floor")
   heightOffsetFromFloor = true
 else
   heightOffsetFromFloor = false
 end

 model.getSurfaces.each do |s|
   next if not runner.inSelection(s)
   next if not (s.outsideBoundaryCondition ==
                "Outdoors")
   new_window = s.setWindowToWallRatio(
   wwr.valueAsDouble,
   offset.valueAsDouble,
   heightOffsetFromFloor)
 end
end
```

NREL
NATIONAL RENEWABLE ENERGY LABORATORY

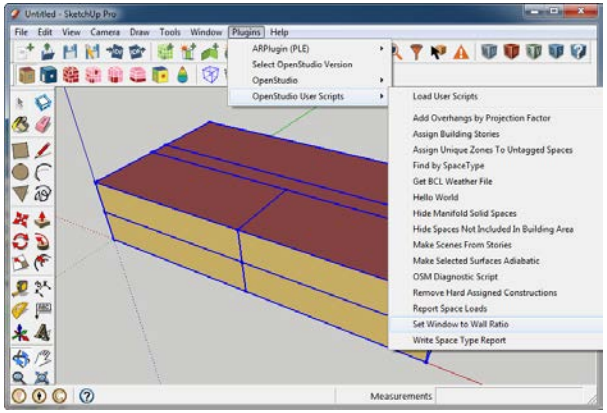


Use Cases for OpenStudio Measures

Interactive Measure Application

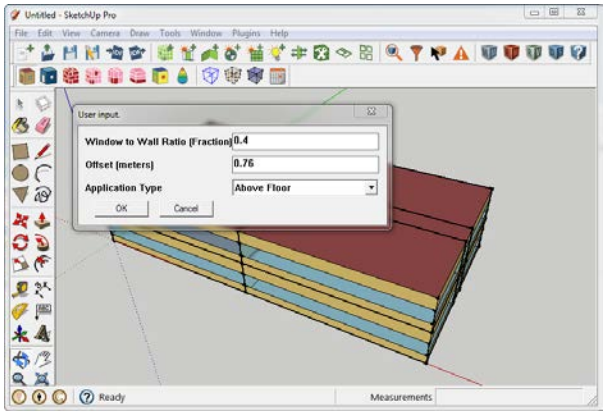
- User can run script at will
- Input arguments are collected at runtime
- User's selection is passed to the script
- Example scripts available in SketchUp plug-in
  - Includes WindowToWallRatio Measure

Before WindowToWallRatio Measure



Credit: David Goldwasser / NREL

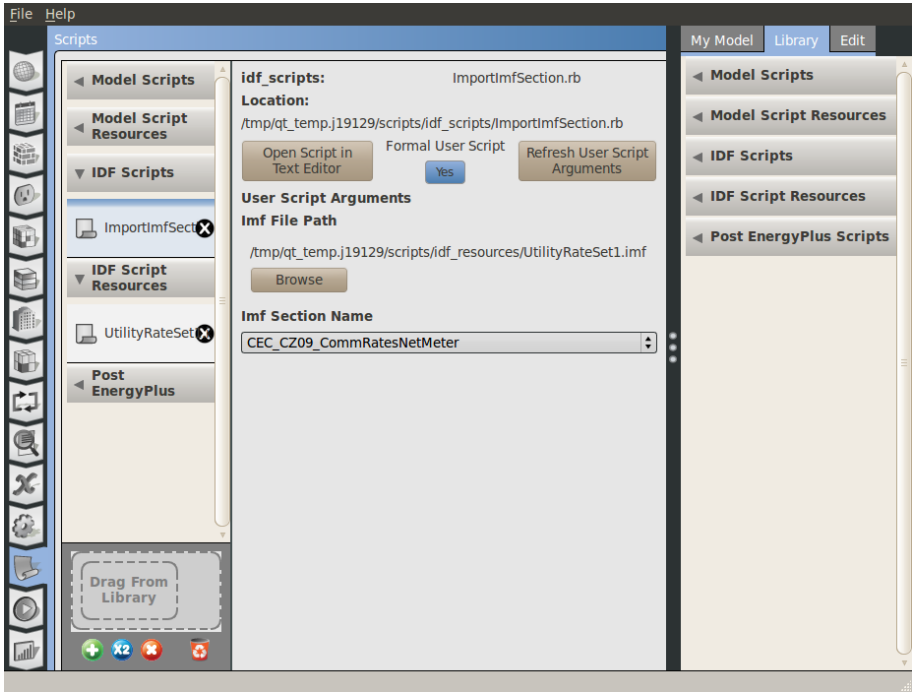
After WindowToWallRatio Measure



Credit: David Goldwasser / NREL

Noninteractive Measure Application

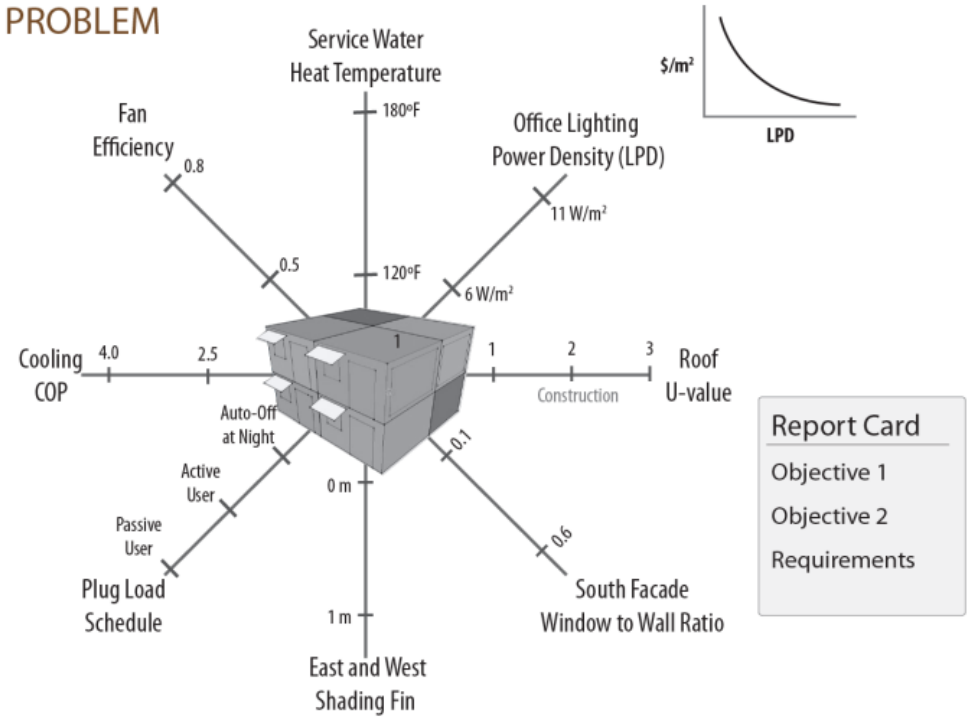
- User must save all input parameters ahead of time
- Script is saved with the model; can be thought of as part of the model
- Script is run every time the model is simulated
- Example scripts available in OpenStudio Application
  - Includes WindowToWallRatio Measure



Credit: David Goldwasser / NREL

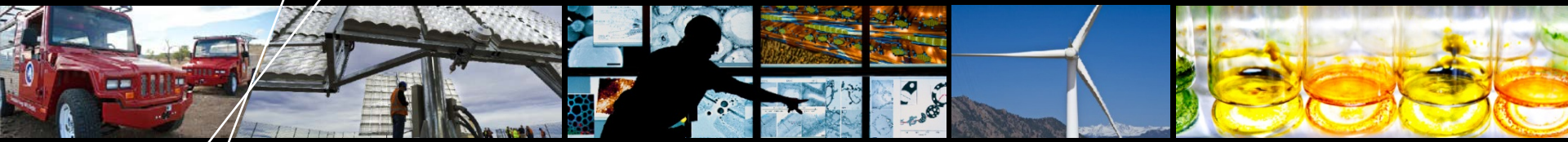
Composing Parameter Spaces with Measures

- Parameter space is composed of several variables
- Measures are used to set a variable to a given value
- Measures are applied as part of the simulation workflow for each model
- Once parameter space is defined, scripted analyses can be run
  - Design of Experiments
  - Optimization
  - Uncertainty Quantification
- Used to generate data for 179D DOE Calculator web tool
  - 250,000 E+ simulations in two weeks
  - Entire analysis rerun in one week after inputs models and measures were revised



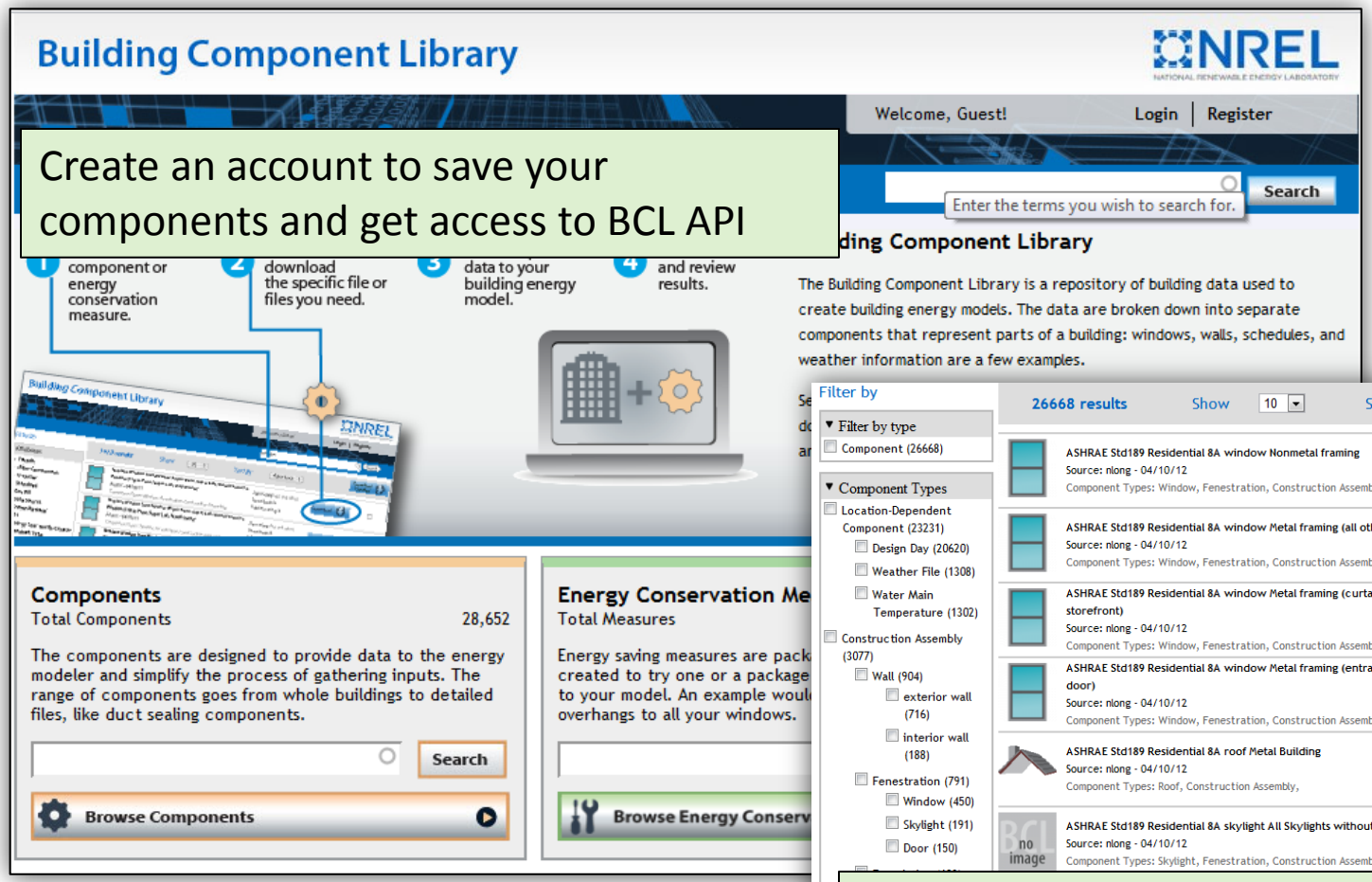
Credit: Marjorie Schott/ NREL

NREL
NATIONAL RENEWABLE ENERGY LABORATORY



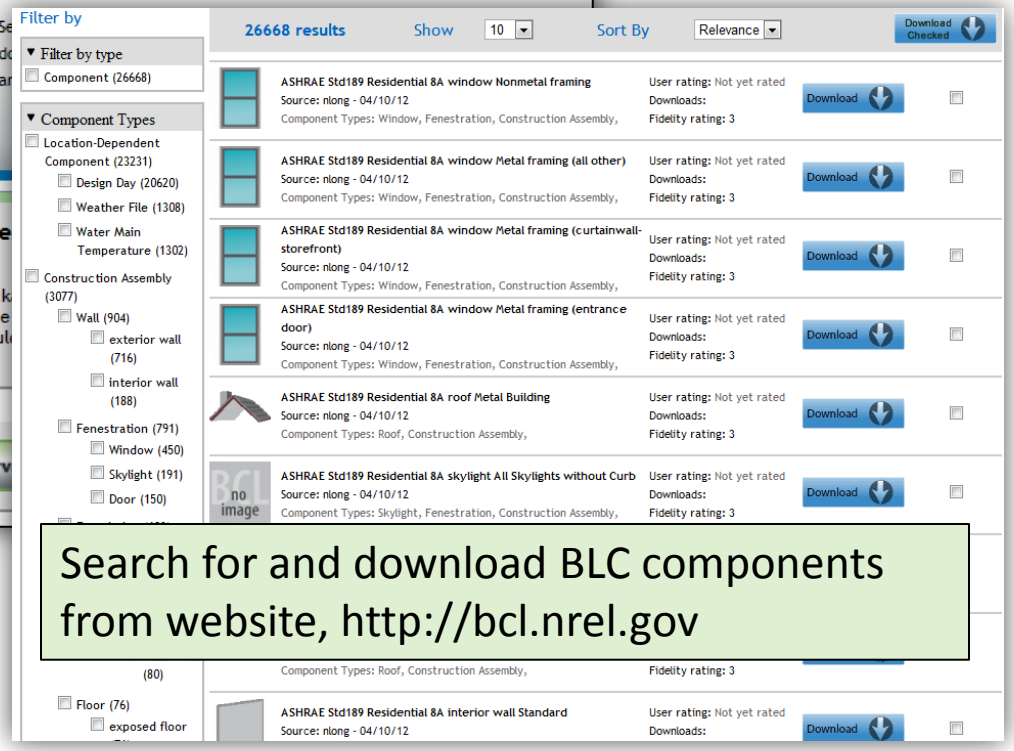
OpenStudio Measures and the BCL

The Building Component Library (BCL)



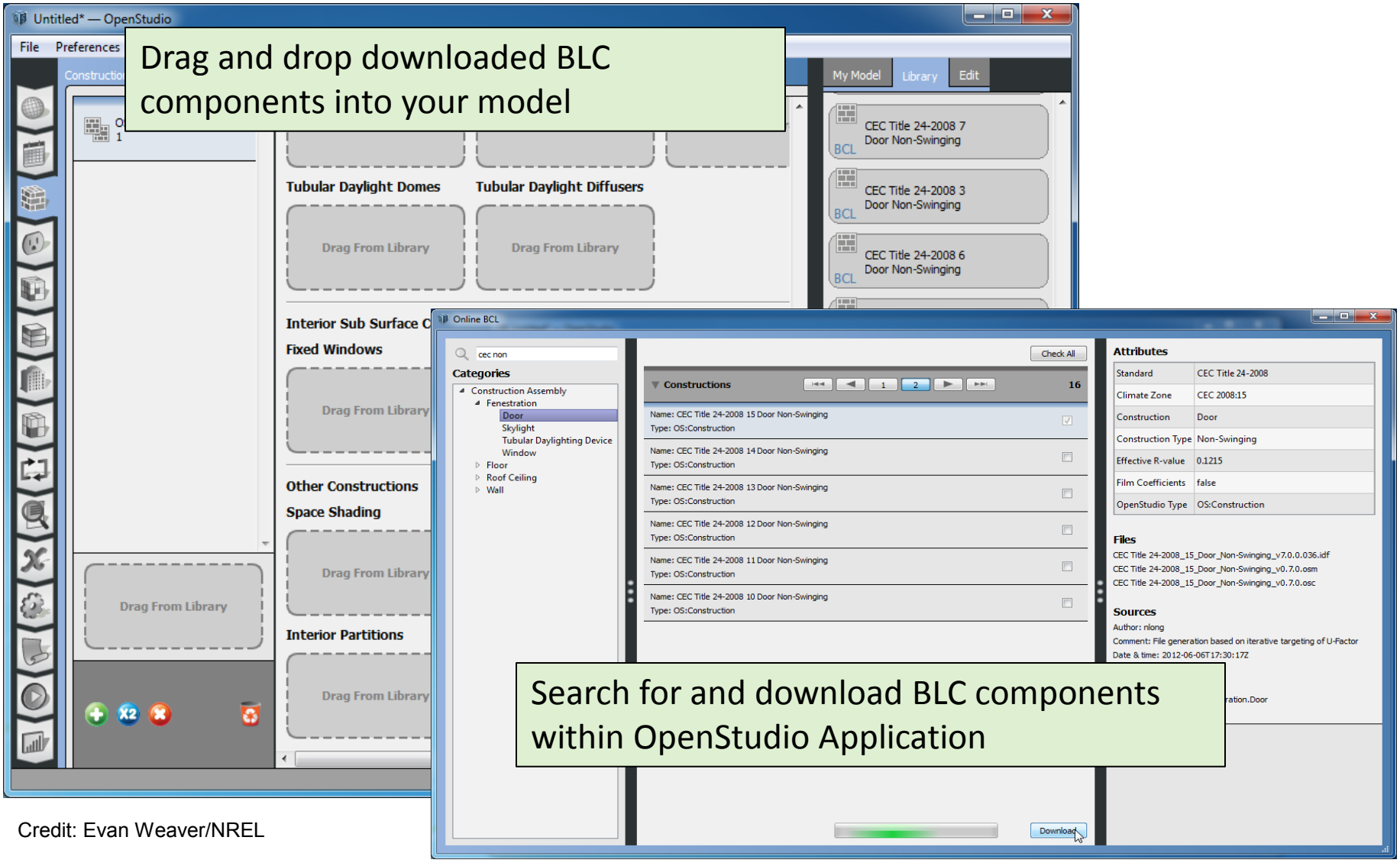
Credit: David Goldwasser/NREL

http://bcl.nrel.gov



Credit: David Goldwasser/NREL

OpenStudio and the BCL

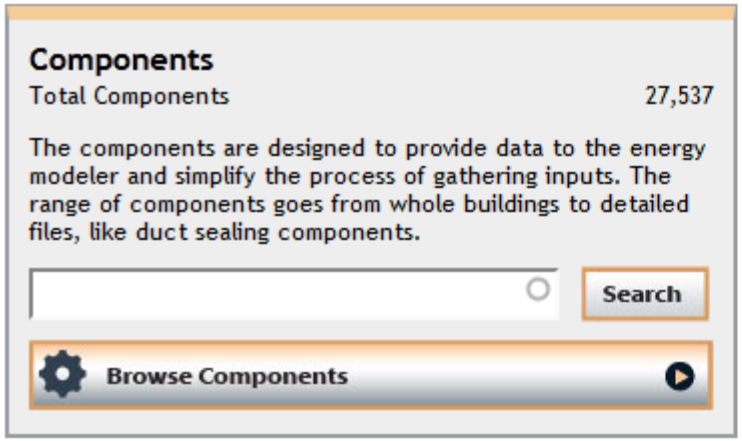


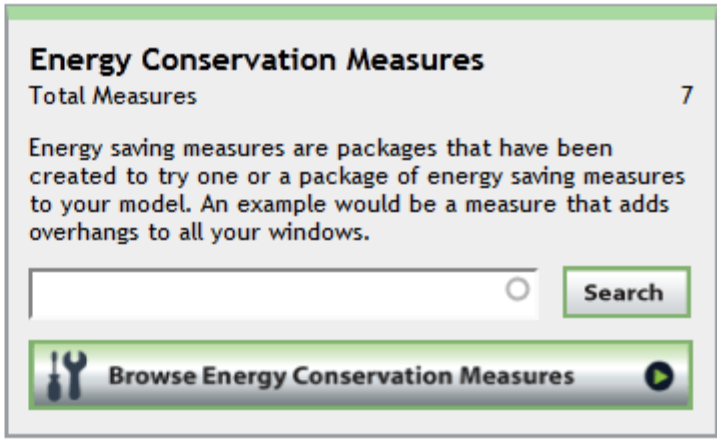
Credit: Evan Weaver/NREL

Credit: Evan Weaver/NREL

OpenStudio Measures and the BCL

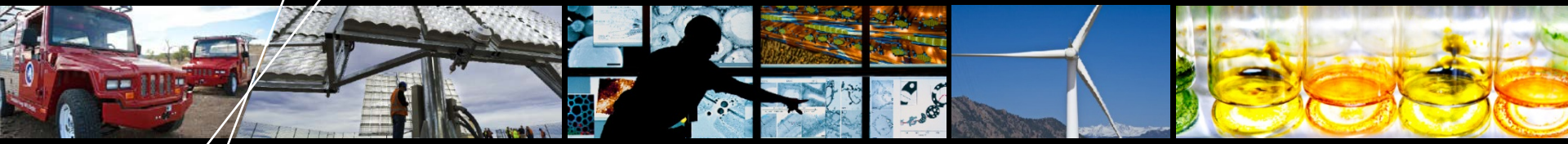
- **BCL allows Measures as well as Components**
  - Measures may refer to Components
  - Components downloaded as needed when Measure runs
  - Measures in BCL still in early stages
- **Future integration with applications planned**
  - Drag and drop Measures into user script libraries, simulation workflow, and parameter spaces





Credit: Daniel Macumber/NREL

NREL
NATIONAL RENEWABLE ENERGY LABORATORY



Conclusions

Conclusions

- GUIs and component libraries reduce time, decrease errors, and improve repeatability in energy modeling
- Entire custom analyses can be scripted end to end using the OpenStudio Ruby API
- Common tasks within an analysis can be automated using OpenStudio Measures
- Initial integration of Measures with OpenStudio, SketchUp Plug-in, and BCL is complete
- Future work will improve integration with GUIs and generate libraries of shareable Measures

Acknowledgments

Funding Organizations:

- U.S. Department of Energy Building Technologies Program
- California Energy Commission

Coauthors:

- Elaine Hale, Kyle Benne, and David Goldwasser

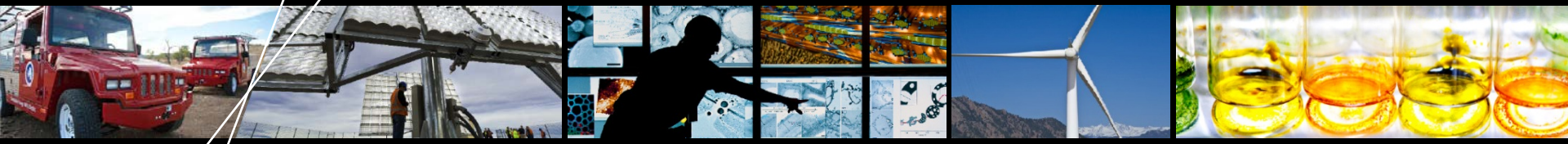
OpenStudio Team Members:

- Brian Ball, Larry Brackney, Luigi Gentile Polese, Nick Long, Marjorie Schott, Alex Swindler, Jason Turner, and Evan Weaver

179D Case Study:

- Brent Griffith, Matt Leach, Eric Bonnema, Katherine Fleming, and Michael Deru

NREL
NATIONAL RENEWABLE ENERGY LABORATORY



Backup Slides

Future Work

- Parameters for Structured Rulesets
- Run Structured Rulesets from OpenStudio and SketchUp Plug-in
- Each measure should generate a report on what it did
- Check if measure applies to given model (alternatively report says it did nothing)
- New measures are added to the BCL over time
- User generated components and measures in BCL