

LEVERAGING OPENSTUDIO'S APPLICATION PROGRAMMING INTERFACES

Nicholas Long¹, Brian Ball¹, Larry Brackney¹, David Goldwasser¹, Andrew Parker¹, Jennifer Elling², Oliver Davis³, and Dale Kruchten⁴

¹National Renewable Energy Laboratory, Golden, CO, USA

²Xcel Energy, Denver, CO, USA

³concept3D, Boulder, CO, USA

⁴National Grid, Waltham, MA, USA

ABSTRACT

OpenStudio development efforts have focused on providing application programming interfaces (APIs) that enable users to extend OpenStudio without compiling the open source libraries. This paper discusses the basic purposes and functionalities of the core libraries that have been wrapped with APIs, including the building model, results processing, advanced analysis, uncertainty quantification, and data interoperability through translators.

Several building energy modeling applications have been produced using OpenStudio's API and software development kits, including the U.S. Department of Energy's Asset Score calculator, a mobile-based audit tool, an energy design assistance reporting protocol, and a portfolio-scale incentive optimization analysis application. This paper discusses each application briefly and describes how the APIs were leveraged for various uses, including high-level modeling, data transformations from detailed building audits, error checking and quality assurance of models, and high-performance computing for mass simulations.

INTRODUCTION

The last several years have seen a number of exciting developments that make simulation engines such as EnergyPlus (USDOE 2013) and Radiance (Ward 1994) accessible to a broader range of design professionals and researchers. Since OpenStudio's initial release in 2010, the application programming interfaces (APIs) and software development kits (SDKs) have evolved to support a number of use cases beyond its initial goal of accelerating the development of practitioner-focused building energy modelling graphical user interfaces (Weaver et al., 2012). OpenStudio abstraction of EnergyPlus objects makes it substantially easier to articulate new energy models and automate analyses (Hale et al., 2012). The Building Component Library (BCL) has also emerged as an integral web service that further simplifies model articulation and transformation through data exchange (Fleming et al., 2012). This paper focuses on recent application examples of the OpenStudio API and SDK to create new web and mobile applications along with automated, large-scale portfolio analyses that support a variety of

market needs for new commercial building construction and retrofit.

WEB APPLICATIONS

OpenStudio's programmatic bindings enable developers to use their favourite languages. The Ruby bindings were initially produced out of necessity to support SketchUp's Ruby-based API (<http://www.sketchup.com/intl/en/developer/>). Ruby has proven to be extremely versatile, enabling quick prototyping of new modelling and analysis workflows using a simple and intuitive interpreted language. It provides a convenient pathway for extending basic OpenStudio functionality, and can be used to produce web applications when used within the popular Ruby on Rails framework (<http://rubyonrails.org>).

Multiple organizations are actively working to apply OpenStudio to create web services that convert high-level user input into a complete energy model using supporting data and assumptions. Two early examples include the U.S. Department of Energy's (DOE) Asset Score calculator web service (PNNL 2012), being led by researchers at Pacific Northwest National Laboratory. Another example is the VirtualPULSE (<http://www.buildsci.us/virtual-pulse.html>) project led by Penn State researchers associated with the Energy Efficient Buildings Hub (<http://www.eebhub.org>). Figure 1 contains the Ruby script required to fully articulate an energy model from high-level variables and run a simulation. The OpenStudio Plug-in for SketchUp and OpenStudio Application may also be used to inspect the resulting model's geometry and systems (Figure 2) along with simulation results (Figure 3).

```
require 'openstudio'
require 'VirtualPULSEModel'

#create a new model
model = VirtualPULSEModel.new

#add geometry (simple multi-story core/perimeter building)
model.add_geometry({"length" => 100,
  "width" => 50,
  "num_floors" => 3,
  "floor_to_floor_height" => 4,
  "plenum_height" => 1,
  "perimeter_zone_depth" => 3})

#add windows at a given window-to-wall ratio
model.add_windows({"wwr" => 0.4,
```

```

"offset" => 1,
"application_type" => "Above Floor"})

#add HVAC - PVAV w/ Reheat - DX Cooling, Hot Water heat and reheat
model.add_hvac({"fan_eff" => 0.5,
"boiler_eff" => 0.66,
"boiler_fuel_type" => "Natural Gas",
"coil_cool_rated_high_speed_COP" => 5.5,
"coil_cool_rated_low_speed_COP" => 6.6,
"economizer_type" => "Fixed Dry Bulb Temperature Limit",
"economizer_dry_bulb_temp_limit" => 30})

#add thermostats
model.add_thermostats({"heating_setpoint" => 20,
"cooling_setpoint" => 25})

#assign constructions from a local library to the walls/windows 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",
"Climate_zone" => "ClimateZone 1-8",
"NREL_reference_building_primary_space_type" =>
"SmallOffice",
"NREL_reference_building_secondary_space_type" =>
"WholeBuilding"})

#add design days to the model
model.add_design_days()

#save the OpenStudio model (.osm)
model.save_openstudio_osm({"osm_save_directory" => Dir.pwd,
"osm_name" => "exampleVirtualPULSEModel.osm"})

#translate the OpenStudio model (.osm) to an EnergyPlus model (.idf)
model.translate_to_energyplus_and_save_idf({"idf_save_directory" =>
Dir.pwd, "idf_name" => "exampleVirtualPULSEModel.idf"})

#run the EnergyPlus model (.idf)
VirtualPULSEModel::run_energyplus_simulation({"idf_directory" =>
Dir.pwd, "idf_name" => "exampleVirtualPULSEModel.idf"})
    
```

Figure 1 VirtualPULSE Ruby Script

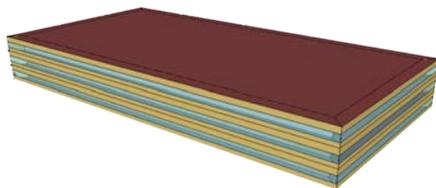


Figure 2 VirtualPULSE Modelled Envelope as rendered with OpenStudio Plug-in for SketchUp

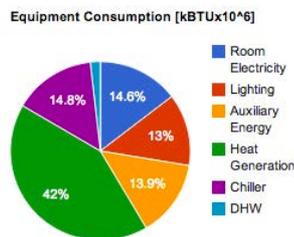


Figure 3 Example VirtualPULSE Simulation Results

This compact, easy-to-read script illustrates the types of high-level parameters that may be used to specify a building energy model with OpenStudio. By comparison, the resulting EnergyPlus input file is many thousands of lines long. In practice, a derivative of this example will be deployed on a

website using Ruby on Rails along with a web form to collect the high-level inputs that are presently “hard coded” in the script.

The VirtualPULSE script accomplishes most of its modelling simplification by calling the BCL’s API to produce detailed space type definitions (building equipment loads, people loads, lighting loads, thermostats, etc.) used in the model. The BCL produces this content “on-demand” based on API arguments, using its own instance of OpenStudio-based scripts with Ruby on Rails bindings (Fleming et al., 2012). The theme of leveraging APIs and distributed services to easily create and simulate models illustrated in this example are further explored in subsequent sections for a range of purposes.

The DOE Asset Score calculator also uses OpenStudio to identify the Asset Score sensitivities associated with key input variables made available to the user via the web interface. To accomplish this, OpenStudio leveraged the R-project, a rich library of algorithms for sampling, uncertainty quantification, sensitivity analysis, and optimization (R-project). In this case, R’s Latin Hypercube Samples (LHS) package is used to sample the high-level input parameter space to identify critical sensitivities affecting the overall asset score produced by the calculator.

Input parameter spaces may be sampled according to specific probability density functions to perform quasi-actuarial analyses. Figure 4 illustrates an example in which a nominal small office building (shown as a singular red solution) has been perturbed to simulate hypothetical deteriorated/derated performance (green lognormal distribution) and a potential retrofit (blue normal distribution). Both HVAC coefficient of performance (COP) and effective insulation R-values are computed, and the resulting distributions in energy use intensity (EUI) are computed. The EUI distributions include interactive effects from varying these two input parameters simultaneously. This sample problem is available as a tutorial on the OpenStudio website (<http://openstudio.nrel.gov/openstudio-tutorials/openstudio-uncertainty-quantification-tutorial>).

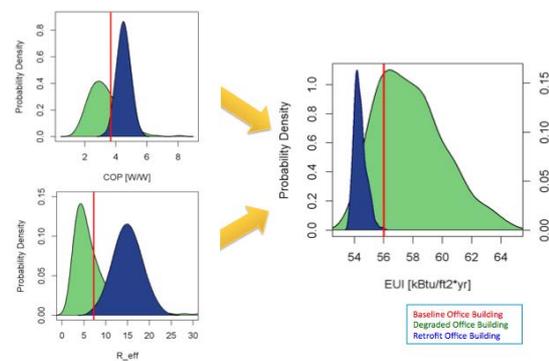


Figure 4 Uncertainty Quantification Example

WEB-INTEGRATED WORKFLOWS

Like a number of USA gas and electric utilities, the Midwest's Xcel Energy offers an energy design assistance (EDA) program that incentivizes energy efficiency for new construction. Pay-for-performance relationships with energy consultants are designed to influence customers who are considering energy conservation measures (ECMs) early in the design process. Increasingly stringent building codes and market conditions have placed significant pressure on EDA programs to reduce program costs and simultaneously increase the quality of outcomes. To maintain its EDA program's economic viability, Xcel Energy has developed an OpenStudio-integrated web service to automate quality assurance and quality checking (QA/QC), program reporting, and communication between all stakeholders.

EDAPT, the Energy Design Assistance Program Tracker, is a web service designed to interoperate with OpenStudio applications used by energy consultants. The site, shown in Figure 5, provides a structured workflow that helps energy program administrators, energy consultants, and customers track approvals and progress of individual projects throughout the EDA life cycle from application, introductory meeting, ECM assessment and selection, and through measurement and verification modelling.

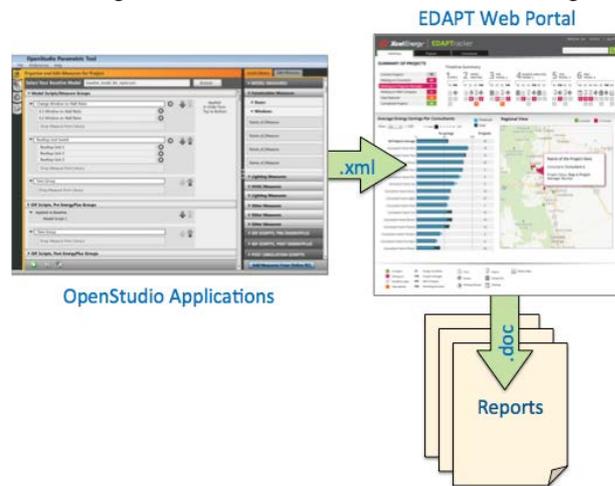


Figure 5 Xcel Energy's EDAPT Web Service

For this application, OpenStudio has specifically been extended to include Ruby scripts that perform a range of automated QA/QC checks that use model inputs and outputs to identify excessive unmet hours, out of range EUIs as shown in Figure 6, and more. These are combined with compliance checks against Xcel Energy protocol requirements such as a test for fuel switching compared to the baseline model and deviation from standard BCL input data prescribed by the protocol. Results are stored in an eXtensible Markup Language (XML) file along with baseline and design alternative model performance that is uploaded to the EDAPT web portal.

```
#checks the EUI for the whole building
def eui_check(model,sql)
  eui_check = Check.new
  building = model.getBuilding
  total_site_energy_G = OpenStudio::Quantity.new(
    sql.totalSiteEnergy.get, $G_unit)
  total_site_energy_kBtu = OpenStudio::convert(
    total_site_energy_G, $kBtu_unit).get
  floor_area_m2 = OpenStudio::Quantity.new(
    building.floorArea, $m2_unit)
  floor_area_ft2 = OpenStudio::convert(floor_area_m2, $ft2_unit).get
  site_EUI = total_site_energy_kBtu / floor_area_ft2
  if site_EUI.value > 200
    eui_check.add_flag(Flag.new($gen_mod, "Site EUI of #{site_EUI}
    looks high. A hospital or lab (high energy buildings) are
    around 200 kBtu/ft^2"))
  end

  if site_EUI.value < 30
    eui_check.add_flag(Flag.new($gen_mod, "Site EUI of #{site_EUI}
    looks low. A high efficiency office building is around 50
    kBtu/ft^2"))
  end

  debug_puts eui_check.flags.inspect
  return eui_check
end
```

Figure 6 EUI Quality Control Ruby Script

EDAPT combines the OpenStudio XML output with project information extracted from the application and other process stages. The user interface uses this information to enable Xcel Energy to assess and track the performance of its entire EDA portfolio. It is also used to automatically generate template documents for energy consultants (e.g., construction documents that enumerate all of the ECMs specified in the model). These documents contain boilerplate information culled from all the project's data, along with standardized end-use plots, comparison tables, etc., which consultants may tailor to support their own customer-focused narratives. In addition to providing a uniform and minimal level of content required by Xcel, the template helps consultants save significant time by avoiding transcription of key simulation results into reports.

MOBILE APPLICATIONS

New mobile workflows are also being enabled by OpenStudio's ability to automate model generation and analysis via web interfaces. An emerging company, concept3D, is developing a new service called "simuwatt™". simuwatt is an application for energy auditors that combines a mobile software-guided workflow, OpenStudio's procedural modelling capability, and the BCL to dramatically reduce the cost of performing consistent, high-quality, investment-grade energy audits (Figure 7).

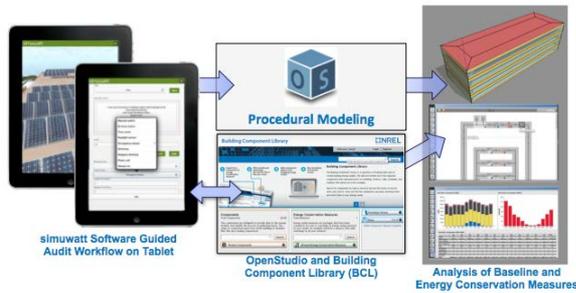


Figure 7 simuwatt Approach to Auditing

Auditors using the mobile application use a tablet computer to define geometry, loads, mechanical systems, schedules, utility rates, and more. Specific pieces of equipment identified in the building are associated with corresponding BCL components. New BCL components may also be entered and uploaded from the tablet, enabling crowd-sourced content generation. Data from the tablet are pushed to the simuwatt server as an XML file. The server automatically translates the audit XML file to a baseline model for use by any OpenStudio-based application (Figure 8).

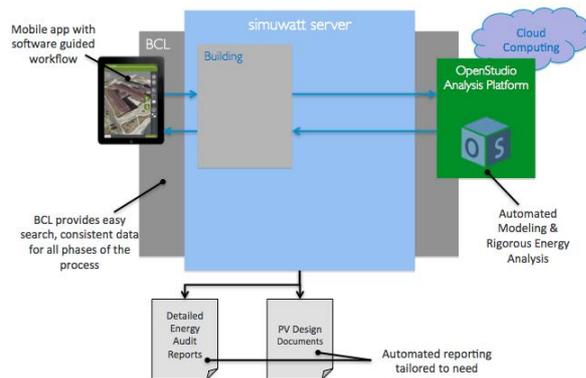


Figure 8 simuwatt Distributed Architecture

The OpenStudio Parametric Analysis Tool (PAT) (NREL 2013) is a user interface that has been works with Xcel’s EDA program, simuwatt, and for general use by modelling practitioners. Shown in Figure 9, it is designed to search, download, and apply BCL measures to the baseline model; manage multiple simulations; present tabular results; and generate the XML output file described in the previous section. This same XML output file is consumed by the simuwatt server, and is used to generate template audit reports. Near-term versions of PAT will support semi-automated model tuning, using calibration measures and cloud computing.

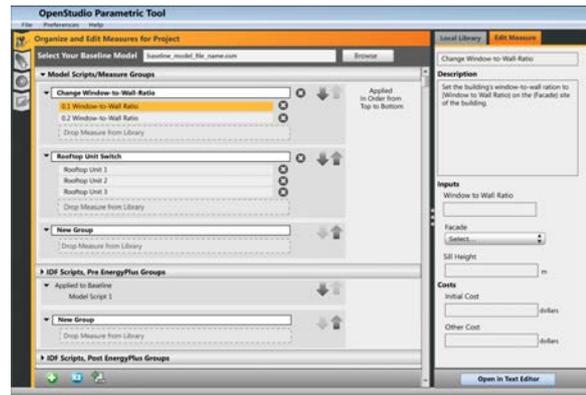


Figure 9 OpenStudio Parametric Analysis Tool

simuwatt, and any OpenStudio-based tool suite, is designed for extensibility as new BCL content is made available. As with model components, measure content creation is expected to be crowd-sourced. PAT provides functionality to author OpenStudio-compatible measure templates, and a measures authoring guide is available to assist authors (<http://openstudio.nrel.gov/openstudio-measure-writing-guide>).

simuwatt is also intended to be a scalable solution that facilitates auditing of single buildings by multiple performers, or multiple building audits that are aggregated to define portfolios (see Figure 10). In addition to audit reports from single buildings, simuwatt is planned to produce portfolio assessment reports that compare baseline performance and identify opportunities to achieve economies of scale when purchasing ECMs for multiple buildings. Audit data, models, and analysis results persist on the server for future use; e.g., when continuing to comply with the U.S. Energy Independence and Security Act 2007 auditing mandate for federal buildings.

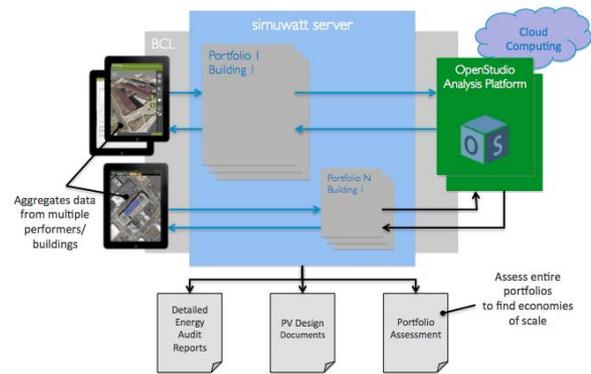


Figure 10 simuwatt Architecture Scalability

LARGE-SCALE ANALYSIS

National Grid, a North East USA utility, focuses on a different portfolio-scale application of OpenStudio designed to optimize incentive and marketing expenditures for their energy efficiency programs. National Grid is developing automated model generation and tuning processes that leverage a combination of geographic information systems

(GIS), public records, customer data, and content from the BCL to feed portfolio optimization algorithms (Figure 11).

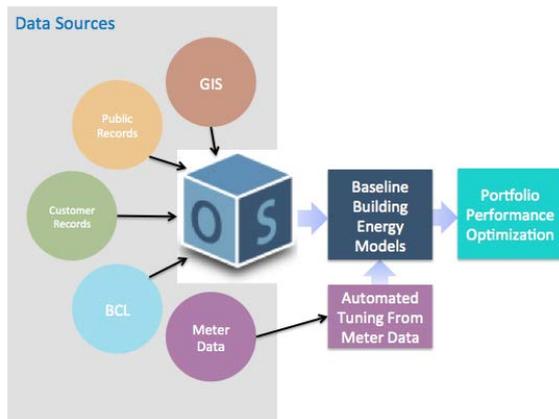


Figure 11 Data Sources for Automated Modelling

High-level data required for baseline model articulation includes: physical address, square footage, number of floors, vintage, building type, and consumption data. A web application (Figure 12) uses the Google Styled Maps API to automatically identify the building footprint from aerial imagery, with options to establish a simple bounding box or user-directed geometry (<https://developers.google.com/maps/documentation/javascript/styling>). All this information is “mashed up” into a single JavaScript Object Notation (JSON) input file (Figure 13) for each customer building.

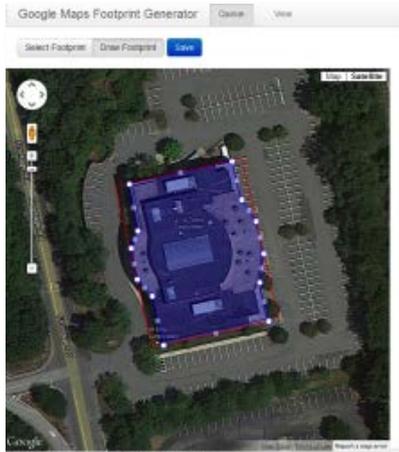


Figure 12 Web Application for Footprint Capture

```
{ "building": {
  "address": {
    "city": "Anytown",
    "state": "NY",
    "street": "1234 RedactedAve",
    "zip": "12345" },
  "geometry": {
    "lat_lng": [12,34],
    "length": 175,
    "rotation": -29.3,
    "sqft": 44414,
    "stories": null,
```

```
"width": 60,
"wwvr": 0.5),
"number_of_customers": 57,
"number_of_non_customers": 59,
"apparent_power": {
  "annual_kwh": 7910294,
  "fuel_type": "electric",
  "hourly_data": [{"time": 1354572784,"value": 0},
    {"time": 1354573084,"value": 0}],
  "max_kw": 1860,
  "monthly_data": [{"time": 1354572784,"value": 0},
    {"time": 1354573084,"value": 0}] },
"reactive_power": {
  "annual_kwh": 7910294,
  "fuel_type": "electric",
  "hourly_data": [{"time": 1354572784,"value": 0},
    {"time": 1354573084,"value": 0}],
  "max_kw": 1860,
  "monthly_data": [{"time": 1354572784,"value": 0},
    {"time": 1354573084,"value": 0}] },
"real_power": {
  "annual_kwh": 7910294,
  "fuel_type": "electric",
  "hourly_data": [{"time": 1354572784,"value": 0},
    {"time": 1354573084,"value": 215}],
  "max_kw": 1860,
  "monthly_data": [{"time": 1354572784,"value": 215},
    {"time": 1354573084,"value": 215}] },
"roof_construction": {"insulation": 0,"type": ""},
"schedule": {},
"type": "office",
"vintage": 1967,
>window_construction": {"insulation": 0,"type": ""} },
"site": { "climate_zone": "5A",
"weather_uid": "0b0da8b1-ae8a-012d-eb00-00ff10a04904"},
"tenants": [{"id": 6767660100,
"name": "",
"sqft": null,
"type": "office"}]}
```

Figure 13 JSON Input File for Automated Modelling with Customer-Identifying Data Removed

The JSON file provides the necessary input to what is, in effect, an OpenStudio-based expert system for automatic model generation of the hundreds of thousands of National Grid customers with minimal human-in-the-loop interaction (Figure 14). A series of OpenStudio measures create a model envelope with core and perimeter zoning, prescribe multiple space types and associated loads based on building type from BCL content, infer construction and mechanical systems based on vintage, and define historical weather input data based on location.

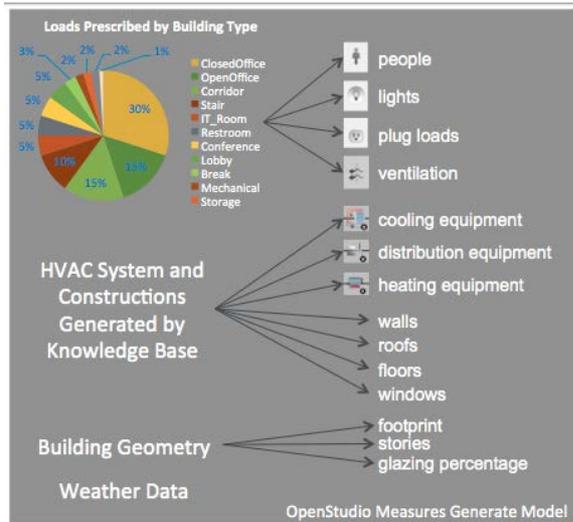


Figure 14 OpenStudio-Based Expert System for Automated Baseline Model Generation

This process has been used on more than a dozen buildings in National Grid’s portfolio. The first, an office building, produced the model shown in Figures 15 and 16, along with the simulated end uses shown in Figure 17.

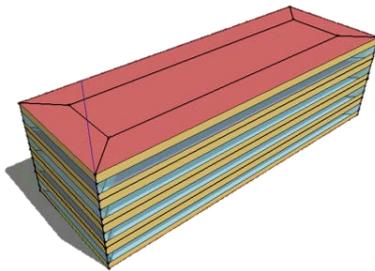


Figure 15 Automated Baseline Model Envelope

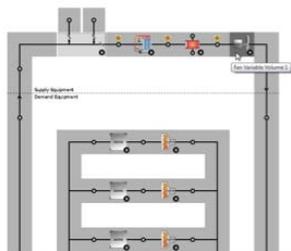


Figure 16 Automated Baseline Model HVAC System

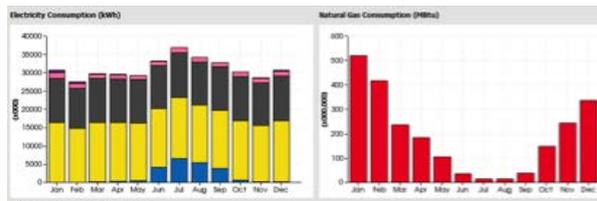


Figure 17 Automated Baseline Model End Uses

Figure 18 shows a comparison of the actual monthly electricity use against the building’s modelled data.

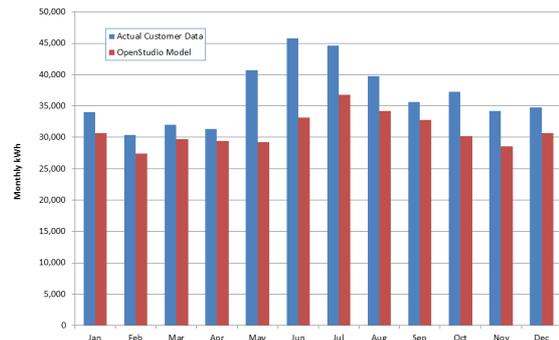


Figure 18 Comparison of Actual and Modelled Data

Comparisons of metered and modelled data in this initial population show a broad range of variation, the worst of which was associated with a building that included unmodelled photovoltaics and medical imaging equipment. Additional heuristics, including precalculated EUI metrics to refine space type allocations and correlation of electricity and gas loads with season to refine HVAC system selection, are planned for addition to the baseline expert system to reduce the initial error bands.

A next step in the project will involve the use of OpenStudio’s analysis and optimization framework to tune baseline models using National Grid’s metered data. As with the simuwatt tool, a constrained set of calibration measures will be applied to each baseline to minimize an objective function that includes monthly and annual electric and gas consumption. These measures will operate on the model at a high level, and will include parameters such as volumetric allocation of space types and wholesale replacement of HVAC systems. This approach relies heavily on OpenStudio’s abstraction of model inputs, and avoids manipulation of low-level EnergyPlus input parameters.

Historically, large-scale analyses such as the one previously described have required access to high performance computing resources. However, using cloud-computing services can enable a large number of practitioners to perform such analyses. The ability to distribute OpenStudio simulations through the OpenStudio RunManager and PAT to various cloud services is currently under development.

CONCLUSION

This paper has presented several case studies that leverage the OpenStudio SDK and BCL to enable web and mobile applications along with portfolio-scale analysis capability. This ability to procedurally generate fully articulated building energy models is being used by multiple DOE National Laboratories, university researchers, and private sector interests to quickly build and analyse models for a range of applications.

The impact of using a standardized model articulation layer from various sources of high-level inputs allows tool developers and end users to focus

on how best to offer incentives and/or focus on the engineering and not the debugging of models. Use of a self-consistent modelling ecosystem creates collateral benefit for stakeholders involved in new construction and retrofit design at many scales. Each use case presented in this paper represents a significant decrease in time spent in modelling, removing barriers for utilities and design firms to use energy modelling to provide actionable guidance to clients.

Rapid and consistent approaches to building energy modelling are crucial to help drive down modelling and operational costs, improve the penetration of emerging technologies through more rigorous analyses and will help designers meet or exceed increasingly stringent codes and standards required for an energy efficient built environment.

ACKNOWLEDGMENT

This work was supported in part by the U.S. Department of Energy Building Technologies Program. The authors appreciate the support of our U.S. DOE sponsors, Amir Roth and Joan Glickman. The authors also thank the rest of the OpenStudio development team including: Kyle Benne, Katherine Fleming, Robert Guglielmetti, Elaine Hale, Daniel Macumber, Luigi Gentile Polese, Marjorie Schott, Alex Swindler, Jason Turner, and Evan Weaver.

REFERENCES

- Fleming K., Long N., Swindler A., 2012. Building Component Library: An Online Repository to Facilitate Building Energy Model Creation, ACEEE Summer Study on Energy Efficient Buildings, August 12-17, Pacific Grove, CA.
- Google. 2013. Google Maps JavaScript API v3: Styled Maps. <https://developers.google.com/maps/documentation/javascript/styling>. Content Last Updated: February 13, 2013.
- Hale E., Macumber D., Weaver E., Shekhar D., 2012. Flexible Framework for Building Energy Analysis, ACEEE Summer Study on Energy Efficient Buildings, August 12-17, Pacific Grove, CA.
- NREL. 2013. OpenStudio. <http://openstudio.nrel.gov>. (accessed January 13, 2013).
- NREL. 2013. OpenStudio Measures Authoring Guide. http://openstudio.nrel.gov/sites/openstudio.nrel.gov/files/OpenStudio%20Measure%20Writing%20Guide_0.pdf. (accessed May 13, 2013).
- NREL. 2013. OpenStudio Parametric Analysis Tool. <http://openstudio.nrel.gov/parametric-analysis-tool-tutorials>. (accessed May 13, 2013)
- NREL. 2013. OpenStudio Uncertainty Quantification Tutorial. <http://openstudio.nrel.gov/openstudio/tutorials/openstudio-uncertainty-quantification-tutorial>. (accessed January 13, 2013).
- PNNL. 2012. Commercial Building Energy Asset Score: Program Overview and Technical Protocol. http://www1.eere.energy.gov/buildings/commercial_initiative/pdfs/energy_asset_score_technical_protocol_phase1.pdf. PNNL-22045.
- R-project. 2013. R-project. <http://www.r-project.org/>. (accessed January 13, 2013).
- USDOE. 2013. EnergyPlus. <http://www.energyplus.gov>. Content Last Updated: February 4, 2013
- Ward, G.J. 1994. The RADIANCE Lighting Simulation and Rendering System. Proc. SIGGRAPH.
- Weaver E., Long N., Fleming K., Schott M., Benne K., Hale E., 2012. Rapid Application Development with OpenStudio, ACEEE Summer Study on Energy Efficient Buildings, August 12-17, Pacific Grove, CA.

