A FRAMEWORK FOR DAYLIGHTING OPTIMIZATION IN WHOLE BUILDINGS WITH OPENSTUDIO

Rob Guglielmetti¹, Brian L. Ball, Ph.D.¹

¹Commercial Buildings Group, National Renewable Energy Laboratory, Golden, CO

ABSTRACT

We present a toolkit and workflow for leveraging the OpenStudio (Guglielmetti et al. 2010) platform to perform daylighting analysis and optimization in a whole building energy modeling (BEM) context. We have re-implemented OpenStudio’s integrated Radiance and EnergyPlus functionality as an OpenStudio Measure. The OpenStudio Radiance Measure works within the OpenStudio Application and Parametric Analysis Tool, as well as the OpenStudio Server¹ large scale analysis framework, allowing a rigorous daylighting simulation to be performed on a single building model or potentially an entire population of programmatically generated models. The Radiance simulation results can automatically inform the broader building energy model, and provide dynamic daylight metrics as a basis for decision.

Through introduction and example, this paper illustrates the utility of the OpenStudio building energy modeling platform to leverage existing simulation tools for integrated building energy performance simulation, daylighting analysis, and reportage.

INTRODUCTION

The building energy simulation community recognizes the need for robust lighting simulations, unconstrained by the geometric and algorithm limitations of the building energy model (BEM), which can characterize spatially-honest daylight distributions, glare potential, and lighting control responses, thus informing the energy efficiency and occupant comfort potentials of a given design or strategy. Many options are available to the simulationist for detailed daylighting and electric lighting simulations, and in recent years many tools (OpenStudio, Honeybee, Sefaira, IESve, et al.) have emerged or evolved to allow daylighting analysis within a whole building energy use context. Some of these tools also enable parametric design and optimization of daylighting strategies. While these tools have granted tremendous power to the design team, they also have the potential to create very large parameter spaces – in turn creating a simulation performance bottleneck, model management problem, and data harvesting and visualization issues.

Further, while a raytraced daylighting solution affords greater accuracy, material model diversity, and spatial consistency, these benefits come at a computational cost. For the daylighting simulationist tasked with demonstrating the efficacy of a daylighting solution or technology, the simulation time requirements are often at odds with the design schedule, or, in the case of an R&D effort, preclude the generation of a reasonable population of building models from which to perform an optimization.

A coherent package of tools was needed to allow the creation of detailed building models, spawn generations of perturbed models, leverage commodity cluster computing resources for the simulations, and visualize the results.

The precursor to this work in many ways is the dataset generation and visualization for the IDEAKit project (NREL 2013), an effort to promote the use of daylighting funded by the Bonneville Power Administration. Recognizing the lack of time, computing power, and daylighting domain expertise in the commercial building sector, the project goal was to showcase potential daylighting savings with a pre-computed database of simulation results for a variety of exemplar commercial building types, in a number of climate zones. A variety of daylighting options were proposed, resulting in a relatively large parameter space. At the time, the measures needed to automate the

¹ https://github.com/NREL/OpenStudio-analysis-spreadsheet
model generation, and the analysis framework needed to distribute the work to NREL’s supercomputing resources did not exist. As a result, the suboptimal computing resource shown in Figure 1 was used. Building models were generated with minimal automation, and simulations were manually distributed across the three worker nodes in the author’s living room. The process was tedious and error prone, and the resulting dataset by project’s end was smaller than initially scoped.

![Figure 1 Suboptimal Cluster Computing Setup](image)

Shortcomings notwithstanding, the project did demonstrate the utility of the parallel coordinate plot as a visualization tool for the multivariate data resulting from a typical parametric analysis in high performance daylighting in building design, and highlighted the need for better ways to access the OpenStudio model via the API. In the ensuing years, a number of efforts by the OpenStudio project team have resulted in several tools that bridge the gaps encountered on the IDEAKit project, enabling large-scale analysis for whole building energy modeling.

LARGE-SCALE ANALYSIS TOOLS IN OPENSTUDIO

A rich library of programmatic model creation and modification scripts—dubbed *measures* in OpenStudio parlance (NREL 2015[2])—has been developed over the last few years by NREL that leverages the OpenStudio API. Accessing the Building Component Library (Fleming et al. 2012), modelers can select from an array of existing OpenStudio measures that can modify their initial building model in discrete and repeatable ways. Building characteristics available for parameterization via measure include materials, form, mechanical systems, and schedules.

The *Parametric Analysis Tool* (Macumber et al. 2014) gives the modeler a GUI for applying these measures in structured and repeatable ways, and sending the entire simulation problem space to lab-scale supercomputing resources and (equally powerful) commodity cloud resources such as Amazon EC2.

Building upon this foundation, the *OpenStudio Analysis Spreadsheet* tool (NREL 2015[1]) was developed to expand the scope of measure-driven BEM beyond the single building to the utility portfolio and national scales. With this tool, the simulationist can apply statistical sampling algorithms to parameterize and perform sensitivity analysis, optimization, and calibration of their proposed designs. Likewise, product developers, utilities, and researchers can explore parameter spaces related to materials, operations, and incentives. Development of this work, outside of the constraints of the C++ based and GUI driven OpenStudio applications, has progressed rapidly.

*OpenStudio Server* (Long et al. 2014) manages the creation of the simulation compute resources, be they local or cloud resources such as Amazon EC2; distributes the simulations across the resource space; and collates the results and stores the results in a so-called “noSQL” Mongo database. Simulation status and results are all served in a web browser/viewer, which allows the user to explore the data at a high level through a variety of visualizations, or to delve deeper by downloading entire model/datasets for detailed inspection, or further analysis/reporting. The resource allocation process is abstracted; by seamlessly leveraging the scalable commodity-priced compute resources of the Amazon EC2 web service, OpenStudio Server makes large-scale analysis possible for the resource-constrained student or small engineering firm, while larger institutions may run OpenStudio Server on their own clusters.

Lastly, one measure deserves special mention as it forms the basis for the large-scale analyses the preceding tools allow. NREL has developed a robust *OpenStudio Prototype Buildings Measure* that can generate a variety of exemplar prototype building models. Based upon the NREL Commercial Reference Buildings definitions (Deru et al. 2011), these building models represent the majority of US commercial building stock, in a variety of code-compliant vintages. This single measure takes three inputs – building type, climate zone, and vintage – and from these inputs, an entire OpenStudio building energy model is generated. The output model is representative of a typical building of its kind, with mechanical systems, building materials, and schedules all representative of the selected climate zone and building vintage (and thus the prevailing building codes). The significance of this...
work, while ongoing and not currently public, is difficult to overstate.

While the OpenStudio application has allowed the simulationist to use Radiance as the lighting simulation engine for individual building simulations, the suite of tools presented here has been available for building energy analysis with OpenStudio/EnergyPlus only. Thus, in order to take advantage of this wealth of functionality for proper daylighting analysis, the OpenStudio Radiance workflow has been refactored as an OpenStudio measure.

RADIANCE DAYLIGHTING MEASURE

The original Radiance-based OpenStudio simulation workflow was created using the same API that measures leverage, which facilitated its refactoring into a proper OpenStudio Measure. In essence, the existing work was simply wrapped with the following API call:

```java
Class RadianceMeasure <
OpenStudio::Ruleset::ModelUserScript
```

The Radiance Daylighting Measure is now included with the OpenStudio installer package, and is available for application to all OpenStudio models directly from the Measures Tab in the OpenStudio application. The entire OpenStudio Radiance workflow is now more modular, has greater optional functionality that is exposed to the user, and is available for insertion into a parametric or optimization workflow via either the Parametric Analysis Tool, or the OpenStudio Analysis Spreadsheet and OpenStudio Server frameworks.

USE CASES FOR RADIANCE IN BEM

The Case for Spatial Accuracy

When Radiance is used to calculate the daylighting in the OpenStudio workflow, the geometry model is (or can be) spatially accurate—representing the actual architecture—rather than a geometric representation of the energy model’s thermal zones. This allows for the exploration of actual daylighting challenges and scenarios, such as deep-penetration sidelighting, atria, ‘borrowed light’, and other strategies that are often necessary to make the value proposition for daylighting.

The need for such architectural model verisimilitude was a driving factor in the use of Radiance in the design development of the NREL RSF net-zero energy building (Guglielmetti et al., 2010), where a 60’ floor plate is sufficiently daylit to the point where the electric lighting power density is reduced as much as 90% for the majority of operating hours. Were the design team to rely completely upon the BEM, the daylighting—so critical to the overall energy efficiency design—would have been sold short from the outset; daylight would only have been available to the 15’ deep perimeter thermal zone, as its boundary is represented by an opaque polygon in the energy model.

Conversely, interior architectural details such as furniture and especially tall cubic partitions can turn ‘open floor plan’ designs into daylight-restrictive spaces that do not perform well from the daylighting perspective. We believe BEM should illustrate bad design as well as good, and a tool that over- or understates the effectiveness of a solution is of no use to the simulationist or the design team. Hence this linkage of a high quality daylighting simulation engine (Radiance) with a high quality energy modeling engine (EnergyPlus) through OpenStudio.

Dynamic Daylighting Metrics Support

The last several years have also witnessed the emergence of new (and the refinement of existing) metrics for describing daylight quantity and quality (IESNA 2012, Wienold 2009). These metrics are spatial in nature; daylight flux in the spatial and temporal distribution, and luminous intensity (again in a spatial context) must be computed to inform these metrics, some of which form the basis for new sustainable building rating systems, and building codes.

We believe these metrics can inform design much the way net site energy and other ‘bottom line’ metrics can, and as such form the other use case for the Radiance measure; the OpenStudio Radiance measure can report the following metrics on the building or per-space basis:

- Daylight Autonomy (DA)
- Continuous Daylight Autonomy (cDA)
- Useful Daylight Illuminance (UDI)
- Spatial Daylight Autonomy (sDA)

The OpenStudio model also supports the placement of one or more glare sensor objects in the building model. The glare sensor has an origin and one or more views (vectors) defined, and daylight illuminance is calculated for each view at each sensor. From this the simplified daylight glare probability (DGPs) is calculated (Wienold 2009). If more than one view is defined, they are uniformly arrayed about the glare sensor origin, and the mean, max and min are all reported, in deference to the adaptive zone concept (Jakubiec and Reinhart 2012).

The reader will recognize the absence of the Annual Sunlight Exposure (ASE) metric from the list of those calculated by the OpenStudio Radiance measure. This is due to the intractability of the necessary calculation; this metric requires high-resolution bidirectional scatter distribution functions (BSDFs) or detailed fenestration models to produce the proper input data for the metric, which placed the feasibility of large scale climate based
daylighting analysis out of reach in the author’s opinion. Further, current research indicates the ASE algorithm, which dictates very specific shade control operation, is unrealistic, making the metric oftentimes contraindicative. We believe the UDI metric is an excellent proxy for daylight penetration, distribution, and glare potential, while only requiring the view-independent quantity of illuminance as the input.

WORKFLOW
Building Model and Parameter Space Definition
Building models are created singularly using the OpenStudio SketchUp plugin, or programmatically using one or more scripts which leverage the OpenStudio API. The choice of whether to build a model manually or automatically is up to the user.

In the former case, the user has tremendous flexibility to create detailed and arbitrarily shaped building models, optionally with interior geometric details added. For example, the OpenStudio Interior Partition Surface gives the modeler an object with which to define interior architectural details such as furniture, columns, partitions, and walls (other than thermal zone boundaries). These objects are translated to a Radiance model and participate in the lighting calculation, resulting in a more accurate spatial daylight distribution dataset, from which honest daylight metrics can be generated. Conversely, the modeler can define thermal zone boundaries that have no architectural counterpart with air walls, which the Radiance translator ignores. The end result is an architecturally honest Radiance geometry model.

Once the simulationist has created the initial building model (.osm), it can be further manipulated through the use of additional measures; these typically modify a discrete element of the building model. As discussed, a large and growing body of measures exists on the NREL Building Component Library, and daylighting-relevant measure examples include:

• Set window visible transmittance
• Set window to wall ratio by facade
• Add overhangs to model

Any of the aforementioned measure actions can also be called on a subset of the model, e.g.: “add overhangs to the south facade”, or “set window-to-wall ratio to 65% on north windows”. When used with the Parametric Analysis Tool or the OpenStudio Analysis Spreadsheet, these measures allow for repeatable and rapid model generation of a single model or an entire population representing a complex parameter space. For example with the Analysis Spreadsheet tool, one can request a given measure’s variable(s) be applied to a range rather than a fixed value, or that the tool perform a sensitivity analysis to determine the parameters and ranges most worthy of simulation time to investigate.

NREL has also developed a measure that automatically adds daylighting controls and illuminance maps to a model based on an analysis of the building’s spaces and their applicability to relevant building codes. When applied to a source model, the model has all the elements necessary for the Radiance measure to function. Figure 2 shows the Small Office prototype building (itself automatically-generated by the Prototype Building Measure) with daylighting controls and illuminance maps that were automatically placed in the model, in this case based on the square footage of the north and south zones and the building vintage’s applicable building code.

Finally, the Radiance Daylighting Measure is applied to the workflow, and the simulation(s) can be initiated.

Model Translation
The OpenStudio-to-Radiance forward translator is the means of generating a valid Radiance model and all elements necessary for a Radiance daylighting simulation from the source OpenStudio model (.osm). The Radiance forward translator converts model geometry, materials/constructions, spatial illuminance calculation points (illuminance maps, in EnergyPlus parlance), daylighting control, window shade control and glare analysis calculation points to valid Radiance
descriptions. In the case of opaque materials, the Radiance plastic primitive is used, and in the case of glazing materials, the Radiance glass primitive is used; if any transmitting materials in the OpenStudio construction have the “is solar diffusing” property, the Radiance trans material is used to define a diffuse transmitter. Reflectance or transmittance is automatically derived from the surface constructions, either taking the inverse of the absorbance or converting the transmittance to transmissivity, respectively.

In order to report dynamic daylight metrics and account for occupant interaction with window shades, the OpenStudio-to-Radiance translator also performs an automated restructuring of the model. All windows in the model are logically grouped by:

- Space
- Orientation
- Visible light transmittance (VLT)
- Unique shading control

This arrangement of window groups allows for the creation of building-wide hourly illuminance schedules for each window group and shade condition. The notions behind the window-grouping rubric were that each orientation receives direct sun at different times of the year and so one might raise shades on one end of the building and lower them on the other, for example. Different spaces may connote different occupancy schedules and tasks, so this criterion also will create a new logical window group. Similarly, different glazing transmittances imply different functions.

The shade control object offers the simulationist the option to simulate the effect of a glare control or daylight redirecting device on a window or group of windows. The device options are:

- Venetian blind
- Shadecloth
- Daylight redirecting device (e.g. Lightlouver)
- Dynamic glazing (e.g. electrochromic glass, or switchable film cartridges)

The modeler can force different behaviors on the same facade by placing different shading controls on specific windows, and assigning different setpoints to each.

The shading devices are represented by either a BSDF (for scattering transmitters) or a by pair of window constructions describing two tint states of a dynamic (state changing) specularly transmitting IGU. Low-resolution (Klems basis) BSDF files are included with OpenStudio for the scattering devices as well as for air. The ‘air BSDF’ (McNeil et al. 2013) allows the use of a single set of daylight coefficients to generate illuminance schedules for both unshaded and shaded conditions, for each window group. Further, all the view matrices for the entire building can be computed in a single step. The use of BSDFs for the scattering media allows for relatively relaxed simulation parameters as compared to what is required to accurately capture redirected daylight with raytracing. The distributions for these BSDFs are presented in Figure 3.

A shading control point is also created (one per window group) and located at the centroid of the largest window in the group. This provides hourly daylight illuminance for each window group, as input to the shade control algorithm.

Once translated, this Radiance model remains intrinsically linked to the source model. Weather data and schedules pertinent to the daylighting simulation (e.g., occupancy, lighting power) remain accessible either in the source model itself, or its support directory, thus minimizing duplication of input.

Note that while the Radiance forward translator is called from within the Radiance measure, this is a core component of the OpenStudio API and as such can be called from any user measure, perhaps as the basis for other modeling and analysis tools that leverage the OpenStudio API, OpenStudio measures, and Radiance (e.g. Honeybee (Roudsari, 2016)).

**Daylighting Simulation**

The remainder of the Radiance Measure simply carries out the daylighting simulations dictated by the translated model. All illuminance map and glare sensor points in the model are concatenated and sent as a single input to Radiance, where the following are calculated or derived:

- Static windows view matrix (1)
- Daylight matrix, dynamic window group (1n(groups))
- View matrix, dynamic window group (1n(groups))
- Static windows annual illuminance schedule (1)
- Annual illuminance schedule for each dynamic window group, with shaded BSDF (1n(groups))
- Annual illuminance schedule for each dynamic window group, with “air BSDF” (1n(groups))
- Annual illuminance schedule for all the dynamic window group control sensor points (1)

![Figure 3 BSDF Distributions: Air (left), Blinds (center), Lightlouver (right)](image-url)
For each window group, a blended illuminance schedule is created by taking the hourly illuminance results from the shaded or the unshaded dataset for the window group, depending on whether the shade control setpoint is exceeded or not, and these blended illuminance schedules are combined with the unshaded windows’ illuminance contribution, for a final whole building spatiotemporal daylight illuminance schedule that represents the combined contributions of each window group, as independently controlled by their individual control setpoints. The measure does this for each window group, and each hour. In the next section we discover the scalability issue(s) this methodology presents. Figure 4 is a composite of calculated spatial daylight illuminance plots (for one timestep) in the Small Office model depicted in Figure 2; each row represents a window group, with the left images showing the illuminance for the shades up condition, and the right showing the shades down. The intent of these plots is to illustrate the level of spatial granularity at which the Radiance measure calculates daylight availability; this is required for so-called “borrowed light” to be accounted for in adjacent spaces, independent of differing shade schedules. What is telling about this composite image is the number of separate result matrices required even for a relatively simple building model. In this example, there are four window groups, each with a shade, resulting in eight simulation runs.

If run in debug mode, the Radiance measure also produces renderings for each daylighting control point in the model, allowing the simulationist to confirm the translation process was successful, and that the individual window group materials have been configured properly (for each window group view matrix calculation, all other window groups must be blacked out). If glare sensors are present, perspective view renderings are also created for each sensor’s primary view vector (see Figure 5). Figure 6 shows the debug images produced for a space containing multiple window groups, illustrating how each window group is materially isolated from one another for the purpose of obtaining discrete daylighting data for each unique window group (and shade condition thereof).

From the building-wide annual illuminance schedule, new lighting load schedules are calculated and inserted into the OpenStudio model and used in turn by the EnergyPlus model. The daylight metrics and glare probabilities discussed earlier are calculated at this time as well.

If running on OpenStudio Server, all these results (as well as the status of the simulations themselves) are
stored in a Mongo database and accessed via webservice. The user is presented with a parallel coordinate plot, showing the daylight metrics and building energy use intensity across the entire parameter space – similar to the original IDEAKit data visualization only easier to produce, and available to the general practitioner.

**CASE STUDY**

The author recently conducted a performance analysis of a switchable daylighting film; the goal was to locate optimal conditions for the film across several building types and climates. As the product can be easily installed in existing buildings, a number of building vintages were also investigated. This task was well-suited to the capabilities of the new Radiance Measure and the OpenStudio Server analysis framework, in addition to a few other measures, some of which were specifically written for this analysis.

The aforementioned “Create Prototype Building” and “Add Daylighting Controls” measures were used to automatically generate small and medium office building models for three vintages and seven climate zones (see Table 1). An additional measure was created for this analysis to place the switchable film on each baseline building, for the comparison. For each combination in the parameter space, a baseline model was created as well as a comparison model with the research film applied, for an initial total of 84 models.

These OpenStudio measures, coordinated with the OpenStudio Server framework, generated these models in a matter of minutes, including the sizing runs required to determine the HVAC configurations, vintage-specific materials and constructions, and even the research subject film characteristics – which were dependent upon the final makeup of the generated building models.

This entire problem was sent first to a local server running OpenStudio Server, and then to an Amazon EC2 instance, and in the case of the latter, the results were reported on the OS-Server webservice that is automatically started on the EC2 instance. Parallel coordinate plots were reviewed, and the full results requested from the simulation (in this case: source EUI, lighting EUI and daylight metrics DA, cDA, and UDI) were downloaded locally for further analysis.

Insights from these simulations raised additional questions, and analysis was conducted to find the optimal elevation(s) for the film, while the daylighting controls’ effects were also isolated. Two minor code changes to the measures were all that were required to re-run the analysis, now on an expanded parameter space of 170 models. The analysis results are not germane to this paper and subject to non-disclosure agreement, and thus are omitted, however researchers were able to determine the best fit(s) for the subject technology across the most probable building types and climate zones.

**Lessons Learned**

The case study analysis did reveal a number of issues with the current Radiance measure implementation, either directly or with its interaction with OS-Server:

1. The Radiance measure is in a sense a victim of the broader measure ecosystem’s efficacy. It is trivial to generate, programmatically, a single building model of complexity sufficient to overwhelm a user’s system resources.

2. At the time the Radiance measure was written, Windows did not reliably support reading matrix data stored in floating point format. As a result, the measure writes all daylight coefficient data in ASCII format, which makes for very large output files (generally 3x larger than the same data stored as floats); these files can easily overwhelm system memory (RAM) during matrix merge operations with rmatrixop, and available non-volatile storage during a simulation (even on enterprise-class cloud systems).

3. While Radiance is internally multithreaded (on UNIX-based systems), the OpenStudio Server framework is designed to handle job threading on its own. Further, there is no way to schedule or otherwise assign priority to specific jobs, processes, or users, on OS-Server. This led to several instances where the servers’ resources (disk space and/or memory) were exhausted, crashing the asset(s).

A number of stopgaps were added to the Radiance Measure to address these discovered issues:

1. The Radiance measure now triggers an alert to the user when there are either a large number of window groups, calculation points, or both; the

<table>
<thead>
<tr>
<th>Building Type</th>
<th>Small Office, Medium Office</th>
</tr>
</thead>
<tbody>
<tr>
<td>Climate Zone</td>
<td>2A, 3B, 4A, 5A, 6A, 7A, 8A</td>
</tr>
<tr>
<td>Facade Elevation(s)</td>
<td>None, North, East, South, West, South+West, East+South+West</td>
</tr>
<tr>
<td>Daylighting Controls?</td>
<td>yes, no</td>
</tr>
</tbody>
</table>

2. At the time the Radiance measure was written, Windows did not reliably support reading matrix data stored in floating point format. As a result, the measure writes all daylight coefficient data in ASCII format, which makes for very large output files (generally 3x larger than the same data stored as floats); these files can easily overwhelm system memory (RAM) during matrix merge operations with rmatrixop, and available non-volatile storage during a simulation (even on enterprise-class cloud systems).

3. While Radiance is internally multithreaded (on UNIX-based systems), the OpenStudio Server framework is designed to handle job threading on its own. Further, there is no way to schedule or otherwise assign priority to specific jobs, processes, or users, on OS-Server. This led to several instances where the servers’ resources (disk space and/or memory) were exhausted, crashing the asset(s).

A number of stopgaps were added to the Radiance Measure to address these discovered issues:

1. The Radiance measure now triggers an alert to the user when there are either a large number of window groups, calculation points, or both; the
measure will exit on a given model if the number of either exceeds a threshold value that all but guarantees a crashed server.

2. The measure limits the number of concurrent rmtxop processes, and performs merge operations serially rather than in parallel.

Despite the obstacles encountered, the Radiance Measure was (eventually) useful in performing a large-scale analysis of a daylighting device using rigorous lighting simulation.

FUTURE WORK

In direct response to the case study experiment experience, the author has re-started the conversation of the Windows float/ASCII issue on the Radiance developer list, toward a true cross platform float I/O solution in Radiance. The OpenStudio Server queuing system is being reimplemented, and a series of OpenStudio model objects and methods will be added to the SDK and Radiance measure to allow users to define exemplar spaces or thermal zones for Radiance analysis and apply the results to similar areas without the commensurate load on system resources. Added support for user-specified shade schedules and BSDFs, ability to calculate ASE, and a daylight metrics report measure are also planned.

ACKNOWLEDGMENTS

This work was made possible with the support of the US Department of Energy’s Buildings Technology Office. In addition, several features of the Radiance Measure were developed for work under the Wells Fargo Innovation Incubator (IN2) project.

REFERENCES


Illuminating Engineering Society of North America Daylight Metrics Subcommittee. 2015. (Subcommittee meeting presentations and minutes).


