OPENSTUDIO: AN OPEN SOURCE INTEGRATED ANALYSIS PLATFORM

Rob Guglielmetti, Dan Macumber, and Nicholas Long
National Renewable Energy Laboratory, Golden, CO USA

ABSTRACT

High-performance buildings require an integrated design approach for all systems to work together optimally; systems integration needs to be incorporated in the earliest stages of design for efforts to be cost and energy-use effective. Building designers need a full-featured software framework to support rigorous, multidisciplinary building simulation. An open source framework—the OpenStudio Software Development Kit (SDK)—is being developed to address this need. In this paper, we discuss the needs that drive OpenStudio's system architecture and goals, provide a development status report (the SDK is currently in alpha release), and present a brief case study that illustrates its utility and flexibility.

INTRODUCTION

High-performance building design requires an integrated design approach to reach ever-loftier energy use goals, satisfy more stringent building codes, and score well against high-performance building rating systems. In general, major building systems (e.g., lighting and air-conditioning) affect each other's operation and energy flows, so many systems and strategies cannot be evaluated as individual line items.

Simulation that verifies design performance and code compliance has become an essential component of the integrated design process. Tools such as EnergyPlus (Crawley et al. 2008) and Radiance (Ward 1994) rigorously evaluate energy and daylighting design strategies and technologies, but have not yet been integrated into a coherent whole-building simulation methodology. Major building systems such as daylighting controls and HVAC, are well modeled by Radiance and EnergyPlus, respectively. They can no longer be modeled in isolation; however, simulations, like building designs, must be integrated to accurately predict building performance (Hirsch et al. 2011).

We present the OpenStudio SDK, an open source analysis platform and toolkit that facilitates integrated whole-building energy analysis. OpenStudio leverages the EnergyPlus and Radiance simulation tools and provides a framework for conducting integrated whole-building energy analysis. It features a holistic building model (.osm) that stores model and site data across multiple simulation tools and a fully open Application Programming Interface (API). It manages simulation parameters across multiple simulation tools (e.g., EnergyPlus and Radiance), and leverages Google SketchUp for graphical user interface (GUI)-based CAD input.

PROBLEM STATEMENT

The needs of the building energy simulationist extend beyond the simulation community. We need to assist designers and engineers, challenge status quo building design, enable the testing and development of new energy efficiency technologies, and promote the evolution of building energy metrics and codes. A whole-building energy simulation toolset is needed to generate, simulate, and analyze building models. It needs to offer:

- Speed
- Simplicity
- GUI interaction
- Integration of multiple simulation engines (for increased simulation fidelity of disparate systems)
- Command line interface functionality
- Batch processing via GUI or scripts
- Optimization and sensitivity analysis
- Rule-based code compliance checking
- Data display and reporting

These requirements cover a wide spectrum of needs and users. Two key concepts for meeting these ends are detailed here.

Integrated Simulations

EnergyPlus and DOE-2 can produce timely results, but have well-known limitations in their daylight simulation algorithms. Both EnergyPlus and DOE-2 engines cannot accurately model daylight flux transfer into complex spaces, and can model only a limited number of daylight redirection technologies. Meanwhile, a fully capable daylight and electric lighting simulation system—Radiance—has been validated (Mardaljevic 2000) and continues to be developed; recent additions have facilitated annual simulations with arbitrarily deployed complex fenestration systems (CFS) (Heshong et al. 2010).

Daylighting and whole-building energy simulations have been used successfully in the design and validation of high-performance buildings (An and Mason 2010, Guglielmetti et al. 2010). The simulation *results* from these efforts were integrated; however, the simulation *process* was not. Multiple designers, architects, and engineers, working in multiple offices, collaboratively built up the simulation results using multiple models that ran through multiple simulation programs on multiple computer systems and platforms. Despite numerous coordination conference calls and email messages, several model/simulation parameters became disconnected. An integrated simulation requires an easily shared, centralized building model.

Optimization and Automation for Massive Insight

An integrated design tool will empower more design and engineering firms to produce high-performance buildings. But most new construction projects do not have and cannot afford an integrated design team, or a sustainable design consultant, to guide the process toward a high-performance result. A tool is also needed for existing buildings, many of which are ripe for energy efficiency retrofits. The U.S. Department of Energy (DOE 2003) reported that existing buildings in the United States numbered 4.9 million; new construction projects totaled just 57,000. This disparity has increased in the wake of the economic downturn. There is enormous opportunity for energy efficiency improvements in the existing building stock, yet retrofit projects seldom have the budget for a fully integrated design team with sophisticated simulation domain knowledge to evaluate all the options. Stakeholders also often lack the patience to wait for minutes or hours for a simulation to run to completion, yet they represent—and are responsible for—a significant percentage of commercial building stock that is ripe for energy efficiency improvements.

The *Advanced Energy Design Guide* (AEDG) series (ASHRAE 2011) provides an easier entry point for stakeholders by including climate-specific energy efficiency design guidance for a number of building types. The recommendations are derived from, and backed by, large simulation datasets (Hale et al. 2009, Long et al. 2010), generated by the automated optimization engine Opt-E-Plus (NREL 2010). Unfortunately, many would-be energy efficiency champions find the AEDGs impenetrable. A "friendlier" front-end to these massive data stores, and *easier methods for researchers and industry to develop them*, are needed. Further, the ability to perform multivariate and multiobjective optimization analysis should not be the exclusive domain of large research facilities, hardware requirements notwithstanding.

SOLUTION DESCRIPTION

Overview

OpenStudio is an open source analysis platform that facilitates integrated whole-building energy analysis. It leverages the EnergyPlus and, optionally, the Radiance simulation engines, and provides a framework for conducting integrated whole-building energy analysis.

OpenStudio is open source under the Lesser GNU Public License[1] and cross platform, running on several Windows, Linux, and Macintosh operating systems.

OpenStudio is written in C++; the development team embraces object-oriented programming and an agile development process. Bindings to C# and Ruby are provided via SWIG.[2]

A plug-in to Google SketchUp enables users to create building geometry and a variety of other input data objects needed by EnergyPlus and Radiance. All building geometry and simulation parameters are stored in a single coordinated building model, an OpenStudio Model (.osm).

When desired (or dictated by building shape or daylighting technology), Radiance is used either with the daylight coefficient method, or to

[1] www.gnu.org/licenses/lgpl.html (accessed July 2011)
[2] www.swig.org (accessed May-2011)

generate discrete timestep simulations, to compute annual daylight availability, glare, and lighting control response; this high-resolution analysis of the lighting solution is in turn used to inform the EnergyPlus model of the electric lighting schedule in lieu of using the split flux or radiosity-based DElight methods available in EnergyPlus.

A number of GUI applications in the OpenStudio Platform facilitate the creation of mechanical systems; simulation job and workflow management on local workstations and high-performance computing (HPC) clusters; and results visualization.

Methods for optimizing economic and energy metrics over architectural, and mechanical design variables (for instance), and rules-based code compliance checking, are under development.

System Architecture

At the core of OpenStudio Platform are the OpenStudio Model, its attendant model data format (.osm), and objects and methods for creating and accessing OpenStudio model objects. The design of the OpenStudio Building Model was developed in an iterative process in response to a list of requirements developed for the overall platform. Requirements included:

- Enabling the user to specify a building at multiple levels ranging from very high (~10 parameters for a whole-building model) down to the full capabilities of the EnergyPlus simulation engine.
- Creating, editing, perturbing, and saving building models or individual components.
- Processing external data dictionary-based queries (e.g., California's Title 24 nomenclature) or other commands for energy standards applications.

These requirements are very broad and some (such as processing data dictionary commands and high-level building parameters) were pulled from the Building Model and placed in other functional blocks of the OpenStudio Architecture (Figure 1) (the Standards Interface and Translator blocks, respectively). The need remained to support two-way translation to EnergyPlus, one-way translations to other simulation engines (e.g., Radiance), and supporting user interfaces (graphical and scripting). Based on these requirements, we used the EnergyPlus data model as a starting point to develop the OpenStudio Building Model. Because other simulation engines and user interfaces require new or different data than the EnergyPlus data model, the Building Model is free to add or subtract data objects that are documented to the same level of rigor required by EnergyPlus.

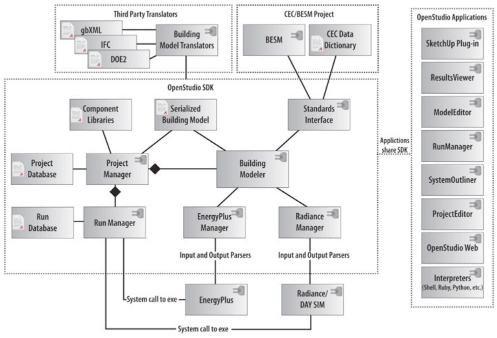
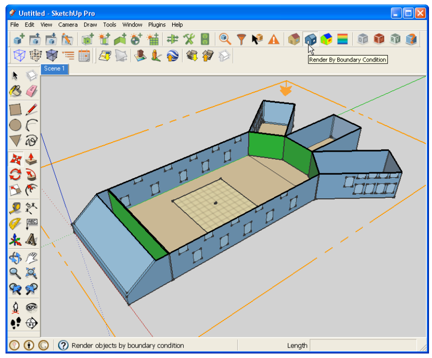


Figure 1 – OpenStudio Architecture

Applications

Several applications leverage the OpenStudio SDK to give the user access to the Model, set up simulation runs, and review data. The first of these tools is the OpenStudio SketchUp Plug-in (Figure 2). This is a continuation of work that began at NREL (Ellis et al. 2008) in an attempt to provide a simple, GUI-based means to rapidly create EnergyPlus geometry input. The Plug-in has been extended to support new OpenStudio objects as they are added to the model.



Credit: Rob Guglielmetti / NREL

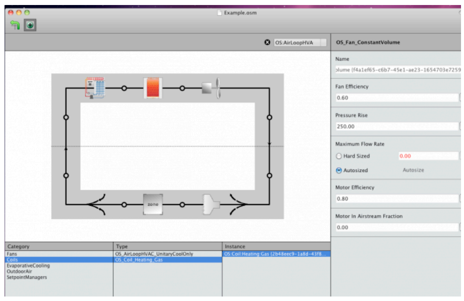
Figure 2 – OpenStudio SketchUp Plug-in

Figure 3 depicts some OpenStudio tool options. OpenStudio can place new .osm objects (e.g., illuminance maps and interior partitions for Radiance simulations) and reading and writing files in the EnergyPlus input data file format. Powerful debugging tools, such as the ability to render zone surfaces by boundary condition, are also available. This render mode and an illuminance map are shown in Figure 2.

Credit: Rob Guglielmetti / NREL

Figure 3 – OpenStudio Toolbars

SystemOutliner (Figure 4) demonstrates using objects and methods exposed by the OpenStudio API to graphically edit mechanical systems, which has historically been a challenging task for EnergyPlus users. The GUI application can generate EnergyPlus input for single-zone air loops; support for multizone systems is currently in development.



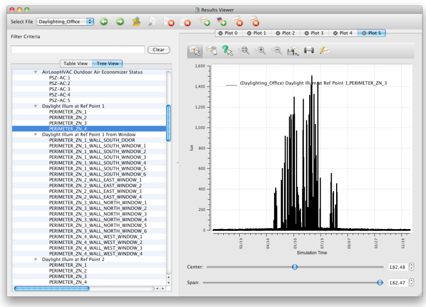
Credit: Rob Guglielmetti / NREL

Figure 4 – SystemOutliner

RunManager manages simulations spread across multiple local processor cores and remote high performance computing resources. It currently supports EnergyPlus simulations and Ruby scripts. Future support for Radiance and other user-defined jobs is planned. Cluster resource management is handled with SLURM.[3]

The ResultsViewer application (Figure 5) displays time series results produced from EnergyPlus report variable requests. ResultsViewer reads the SQLite database output of the EnergyPlus engine and presents the time series data in line and two-dimensional flood plots. It also displays the html output of EnergyPlus in an internal browser window.

[3]Simple Linux Utility for Resource Management https://computing.llnl.gov/linux/slurm/ (accessed August 18, 2011)



Credit: Rob Guglielmetti / NREL

Figure 5 – ResultsViewer

Workflows

These applications can be used interchangeably throughout a project. A typical workflow would begin with using the OpenStudio SketchUp plugin to create a geometric building model with the CAD GUI; the plug-in also includes templates for rapidly populating the model with valid mechanical systems. SystemOutliner could then be used to visually create custom single-zone mechanical systems. Once the user has an OpenStudio model, they could use the RunManager GUI to set up a series of simulations, or use Ruby scripts to access the perturbation and rules processing methods available in OpenStudio. (RunManager can also be used from the command line and within Ruby scripts as needed.) Results are then viewed in ResultsViewer.

EnergyPlus – Radiance Integration

As mentioned in the Introduction, the daylight simulation algorithms in EnergyPlus and DOE-2 have limitations. Reinhart and Jones (2004) illustrated that the differences in predicted interior illuminances between DOE-2 and Radiance-based DAYSIM were significant, leading to an overstatement of the potential energy savings of a given design. Koti and Addison (2007) found similar results, and proposed "aiding" DOE-2 with DAYSIM-calculated data. They essentially informed the energy simulation of the lighting systems' daylight response determined by a Radiance-based simulation; this methodology is currently in use by practitioners attempting to integrate energy and daylighting simulations (An and Mason 2010).

As DAYSIM and EnergyPlus determine lighting power reductions based directly on the work plane sensor point illuminance, the photosensor controlling the lighting system (which would actually be located on the ceiling) is assumed to have a precise cosine spatial response and be "ideally commissioned" to send the proper response to the lighting control system. The work plane sensor point is assumed to be truly representative of the critical task illuminance set point. In practice, none of these criteria are likely to be met.

The SPOT tool (Architectural Energy Corporation 2008) was created to address this gap. This Radiance-based simulation more faithfully represents a daylight-responsive lighting control system, where the photosensor is located realistically and is commissioned to "read" the task illuminance or another input that has been determined to correlate to the task. SPOT also models the interaction of discrete luminaire zones—a subtle and often confounding factor in commissioning the lighting control systems in real buildings. SPOT offers an option to save a DOE-2 function file from an annual simulation, to facilitate the "assistance" process described by Koti and Addison (2007); Guglielmetti et al. (2010) used this successfully on a recent project.

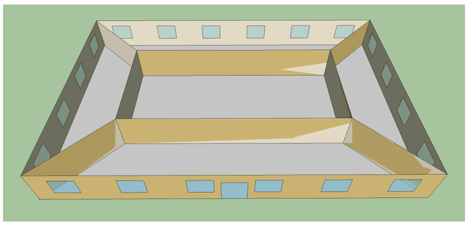
Scheib et al. (in press 2011) further investigated the limitations of EnergyPlus' simulation capabilities. They used the Radiance-based SPOT software to conduct a Radiance daylight simulation and calculate the true photosensor response. They found that calibration factors could be derived from a limited set of SPOT simulations that could be applied to the EnergyPlus computed daylight values. This raises the possibility of using Radiance to compute daylight solutions over a limited range of seasonal conditions and applying those results to the much more rapidly computable daylight solutions produced by EnergyPlus.

All this illustrates that although the daylight calculation in Radiance certainly has value, the electric lighting and control simulations are critical to demonstrating the efficacy of a given daylighting design or system. It is expensive to compute, but we believe there may be ways to use Radiance judiciously to overlay rigorous daylighting simulation on larger EnergyPlus-simulated parameter spaces. Many daylighting strategies also take architectural forms that EnergyPlus cannot handle, and technologies that cannot be evaluated using the EnergyPlus algorithms (because they involve specular surfaces, for instance). For such cases, a rigorous lighting simulation is required.

The time burden of the ray tracing calculations done by Radiance has been called into question. EnergyPlus can compute the daylighting solution for a time step in seconds, whereas a Radiance simulation can take several minutes to several hours. Recent additions to the Radiance suite enable the simulationist to apply the daylight coefficient approach (Tregenza and Waters 1983) to the problem of annual daylight simulation (Bourgeois et al. 2008), and use bidirectional scattering distribution functions (BSDFs) to describe complex fenestration systems and arbitrary window treatments (blinds, shades) in an annual daylighting context (Saxena et al. 2010, Ward et al. 2010). Although the Radiance simulations take longer than the EnergyPlus simulations for the same building, the Radiance results are of much better quality. Radiance can also generate high dynamic range images, suitable for qualitative evaluation, visualization, and glare analysis of a daylighting design.

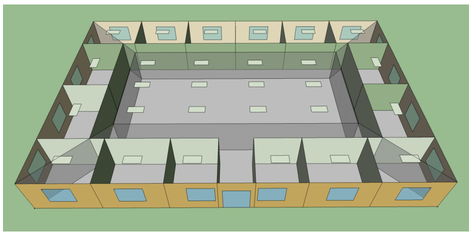
For all these reasons, the integration of Radiance and EnergyPlus simulations has been a key requirement for the OpenStudio project.

The first step toward this end is to add objects and methods to the OpenStudio Model to support daylighting analysis. For example, EnergyPlus models are generally discretized into thermal zones; OpenStudio models also follow this convention. But when evaluating daylight traversal through a building model, the thermal zone boundaries do not necessarily align with architectural massing. To address this issue, an air wall construction denotes a thermal zone boundary that has no architectural counterpart. Conversely, objects are also needed to represent internal objects (e.g., interior walls, furniture, cubicles, and partitions) as they exist in the space plan; such objects are implemented in OpenStudio as *internal partition objects.* Figure 6 depicts an OpenStudio model with typical energy model geometric detail; Figure 7 shows the same model with the additional OpenStudio objects needed for a proper Radiance simulation. The industry standard practice of representing energy model geometry using infinitely thin wall constructions is a further hindrance to producing representative Radiance models from EnergyPlus-type input. Support for wall thickness in the Radiance models is planned, using either additional internal partition objects, shading elements, or BSDFs.



Credit: Rob Guglielmetti / NREL

Figure 6 – Typical Energy Model (.idf) Viewed in OpenStudio with Solid Thermal Zone Boundaries



Credit: Rob Guglielmetti / NREL

Figure 7 – OpenStudio Model (.osm) with Air Wall Constructions, Internal Partitions and Luminaire Objects for Radiance Simulation

Calculation grids and photosensors (referred to as *illuminance maps* and *daylight sensors*, respectively) are available for placement in the model, and a framework for individual luminaire placement (and association to photometric distributions (.ies files) is under development. These objects are all placed or assigned via the OpenStudio SketchUp Plug-in GUI interface. Lastly, a series of scripts were developed to support Radiance analysis in OpenStudio:

- **ModelToRad.rb** converts an OpenStudio .osm model to a valid Radiance model. Geometry, material properties, site, and weather file information are all derived from the .osm, and valid Radiance objects and materials are created. Currently, only Radiance materials "plastic" and "glass" are supported; support for "trans" and the new "bsdf" materials is planned. *Air wall* constructions are ignored, and *illuminance maps* and *daylight sensors* are exported in standard Radiance point/vector arrays. Aperture polygons (e.g., windows and glass doors) are grouped by orientation to facilitate three-phase daylight coefficient matrix binning.
- **DaylightSim.rb** enables the user to perform a variety of simulations with Radiance tools on the converted OpenStudio model. Time step simulations (of views or illuminance maps) are available using standard Radiance tools (e.g., rtrace and rpict), and daylight coefficients can be computed (simple daylight matrix). An annual simulation mode and complete support for the three-phase method with BSDF support is under development.

We used the OpenStudio API to rapidly prototype this functionality in Ruby and demonstrate the general utility of the OpenStudio libraries. The functionality contained in these scripts will ultimately be reimplemented in C++ as part of the OpenStudio core.

Optimization

Ellis et al. (2006) demonstrated that simulation can be used to navigate large parameter spaces in support of optimization analysis for a variety of building types and program goals. They used Opt-E-Plus, a noncommercially available research tool (NREL 2010).

The optimization algorithm used in Opt-E-Plus is being ported to OpenStudio to make it publicly available. The Sequential Search algorithm operates on bi-objective optimization problems over categorical variables. Typically, the two objective functions of interest are cost and energy savings. Additional optimization and uncertainty handling features are under development. For large problems, effective use of these features will often require access to cluster, cloud, or supercomputing resources (not all of which are yet supported by RunManager).

Components containing model data and rules for applying those data to a model will be stored in an online Building Component Library (BCL), which will enable users to use community vetted data to more quickly assemble models and design problems.

APPLICATION

179D Tax Deduction Quick Estimate Tool

As an example of the utility of the OpenStudio Platform, we present a brief case study of its application to the creation of a large dataset for a DOE request.

As part of an initiative to promote energy efficiency in U.S. buildings, DOE requested a Web-based tool that would simplify the determination of eligibility for a tax deduction. Section 179D of the U.S. Tax Code provides tax deductions to builders, owners, and lessors who implement energy efficiency measures (EEMs) in commercial buildings. DOE's vision was to provide stakeholders an easy means to evaluate various EEMs on their own buildings against the tax credit requirements, and quickly report success or failure. The Web-based front end accepts simple inputs, and relies on a large precomputed energy analysis dataset for instantaneous reporting. The initial version of the tool considered lighting, HVAC, and envelope EEMs only and on an individual basis; nonetheless, given the number of building types covered by the tax credit, it was still necessary to analyze a large number of cases. The parameter space was:

- 12 building types
- 16 climate zones
- 92–1009 HVAC perturbations (depending on building type and climate)
- 92–1009 envelope perturbations (depending on building type and climate)
- 4 lighting perturbations

This dictated a final count of 226,867 simulations, which were run in a series of batch jobs. NREL researchers used the OpenStudio Ruby bindings to automate the process of creating and simulating these models using NREL's Red Rock HPC cluster. All simulations were run in approximately one week. The entire project (assignment, simulation, data fit, Web front-end programming, and going live) was completed in approximately one month (NREL 2011).

Conclusions and Future work

We have presented an integrated open source, cross-platform analysis platform for high-performance building design that leverages the EnergyPlus and Radiance simulation engines. A holistic model—the OpenStudio object model—contains all model geometry and other input parameters, and several applications are enabled by this object-oriented framework for creating building energy models. An API provides programmatic access in C++, Ruby, and C#.

We have argued for more thorough daylighting analysis than is currently offered in energy analysis software, citing evidence that daylighting simulation tools are required to produce high-performance building designs whose instantiations meet the lofty goals set at the beginning of a project. Integrated daylighting and energy simulation analysis is supported by the holistic OpenStudio building model, which specifically supports EnergyPlus and Radiance model objects and inputs.

The work to integrate Radiance simulation capabilities into OpenStudio workflows is ongoing. There is currently one-way translation from .osm to Radiance format, but support for all relevant OpenStudio model objects (and Radiance materials) requires further development. The ability to set Radiance simulation parameters from the GUI is currently unavailable. These parameters are not stored in the model; this support will be added. Radiance simulation parameterization guidance for end users is also needed, and is planned for future releases.

With the addition of new OpenStudio objects, several geometric model simplifications inherent to energy models, but inaccurate for daylighting simulations, have been overcome for Radiance-ready models. We recognize that infinitely thin wall constructions still present an unacceptable simplification for rigorous daylight simulation. This will be addressed by adding wall thickness and skylight well geometry directly to the OpenStudio model, or by representing this geometry as BSDFs automatically upon model conversion.

The creation of a robust, multiapplication analysis platform has by necessity been rather inward and developer focused. The framework and the API are now at the point where rapid prototyping and testing of additional functionality will be relatively easy. As this is an open source project, we invite the simulation community to join us in refining, testing, and extending OpenStudio.

ACKNOWLEDGEMENTS

The authors would like to thank the entire NREL OpenStudio development team for their work on this project, in particular: Kyle Benne, Elaine Hale, Evan Weaver, Katherine Fleming, David Goldwasser, and Marjorie Schott; Brent Griffith for his work on the 179D tax tool and willingness to share his EnergyPlus expertise; and Larry Brackney and Luigi Gentile Polese for their leadership and management of this software project and development team.

Further thanks to Greg Ward of Anyhere Software, Andrew McNeil of Lawrence Berkeley National Laboratory, and Axel Jacobs of London Metropolitan University, for Radiance tool development, validation work, and usage references.

This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08-GO28308 with the National Renewable Energy Laboratory.

REFERENCES

American Society of Heating, Refrigerating and Air-Conditioning Engineers. 2011. Advanced Energy Design Guides. www.ashrae.org/technology/page/938 (accessed July 2011).

An, J., Mason, S. 2010. Integrating Advanced Daylight Analysis Into Building Energy Analysis. Proc. Fourth National Conference of IBPSA-USA, New York, NY.

Architectural Energy Corporation. 2008. www.archenergy.com/SPOT (accessed May 2011).

Bourgeois, D., Reinhart, C.F., Ward, G. 2008. A Standard Daylight Coefficient Model for Dynamic Daylighting Simulations. Building Research & Information, 36(1): pp. 68–82.

Crawley, D.B., Hand, J.W., Kummert, M., Griffith, B.T. 2008. Contrasting the Capabilities of Building Energy Performance Simulation Programs. Building and Environment 43(4); pp. 661–673.

DOE, 2003. Commercial Buildings Energy Consumption Survey. www.eia.doe.gov/ emeu/cbecs (accessed May 2011).

Ellis, P.G., B.T. Griffith, N. Long, P. Torcellini1, and D. Crawley. 2006. Automated Multivariate Optimization Tool for Energy Analysis. Proc. Simbuild 2006. Golden, Colo.: National Renewable Energy Laboratory.

Ellis, P.G., Torcellini, P.A., Crawley, D.B. 2008. Energy Design Plug-in: An EnergyPlus Plug-in for SketchUp. Proc. IBPSA-USA SimBuild Conference, Berkeley, CA.

Guglielmetti, R., Pless, S., Torcellini, P. 2010. On the Use of Integrated Daylighting and Energy Simulations to Drive the Design of a Large Net-Zero Energy Office Building. Proc. Fourth National Conference of IBPSA-USA, New York, NY.

Hale, E.T., Leach, M., Hirsch, A, Torcellini, P.A. 2009. General Merchandise 50% Energy Savings Technical Support Document. Golden, CO: National Renewable Energy Laboratory, NREL/TP-550-46100.

Hirsch, A., Pless, S., Guglielmetti, R., Torcellini, P.A. 2011. The Role of Modeling When Designing for Absolute Energy Use Intensity Requirements in a Design-Build Framework. Golden, CO: National Renewable Energy Laboratory, NREL/CP-5500-49067.

Heschong, L., Saxena, M., Higa, R. 2010. Improving Prediction of Daylighting Performance. Proc. ACEEE 2010 Summer Study on Energy Efficiency in Buildings, Pacific Grove, CA.

Koti, R. A., Addison, M. 2007. An Assessment of Aiding DOE-2's Simplified Daylighting Method with DAYSIM's. Proc. Solar 2007 Conference, Cleveland, OH.

Long, N.L., Hirsh, A., Lobato, C., Macumber, D. 2010. Commercial Building Design Pathways Using Optimization Analysis. Proc. ACEEE 2010 Summer Study on Energy Efficiency in Buildings, Pacific Grove, CA.

Mardaljevic, J. 2000. Daylight Simulation: Validation, Sky Models and Daylight Coefficients. PhD thesis, De Montfort University, Leicester, UK.

NREL, 2010. Opt-E-Plus Software for Commercial Building Optimization. Golden, CO: National Renewable Energy Laboratory NREL/FS-550-45620.

NREL, 2011. 179D Easy Calculator (Beta). Golden, CO: National Renewable Energy Laboratory. http://179d.energy.gov (accessed September 2011).

Reinhart, C., Jones, C. 2004. Electric lighting energy savings for an on/off photocell control – a comparative simulation study using DOE2.1 and DAYSIM. Proc. eSim 2004, Vancouver, B.C.

Saxena, M., Ward, G., Perry, T., Heschong, L., Higa, R. 2010. Dynamic Radiance – Predicting Annual Daylighting With Variable Fenestration Optics Using BSDFs. Proc. Fourth National Conference of IBPSA-USA, New York, NY.

Tregenza, P.R., Waters, I.M. 1983. Daylight Coefficients. Lighting Res. Technol. 15(2); pp. 65–71.

Ward, Gregory J. 1994. The RADIANCE Lighting Simulation and Rendering System. Proc. SIGGRAPH.

Ward, G., Mistrick, R., Lee, E.S., McNeil, A., Jonsson, J. 2010. Simulating the Daylight Performance of Complex Fenestration Systems using Bidirectional Scattering Distribution Functions within Radiance. Proc. Annual Conference of the Illuminating Engineering Society, Toronto, CA.