Chapter 1
Introduction to Building Energy Modeling

1.1 Why Modeling?

There is good reason that so much attention is paid to the concept of mathematical model in engineering and physics curriculum. Simple regressions derived from empirical data, differential equations based on first-principles, or detailed computational fluid dynamic simulations each provide an analytical framework that yields insight into the behavior of physical systems. In turn, those insights can lead to design decisions that have real impact on safety, cost, and performance of the cars we drive, the power grids that deliver our electricity, and the energy efficiency of the buildings we live and work in.

The cost/benefit of modeling has varied across markets and over time. A few examples include:

- **1960s Aerospace**: The aerospace industry was early to embrace model-based design in order to manage the incredible cost of prototyping aircraft while protecting the lives of test pilots.
- **1970s Automotive**: Increasingly stringent fuel efficiency and emissions standards, coupled with reliance on complex electronic controls drove engine and car manufacturers to adopt sophisticated model-centric processes to minimize development cost and time to market.
- **1980s Financial**: While statistical analysis had long been used to assess trends and risk in financial markets, widespread integration of computers into financial transactions put greater pressure on real-time modeling and analytics to maximize profit in both short and long terms.
- **2000s Power**: Beside the issues of load growth and emissions reduction, the power sector was faced with myriad challenges ranging from increased penetration of renewable energy resources to the introduction of demand response strategies that introduced volatility to the grid. Increasingly sophisticated models of generation, transmission, distribution, and demand systems were required to plan capital expenditures, schedule power reserves, etc.

L. Brackney et al., *Building Energy Modeling with OpenStudio*,
https://doi.org/10.1007/978-3-319-77809-9_1

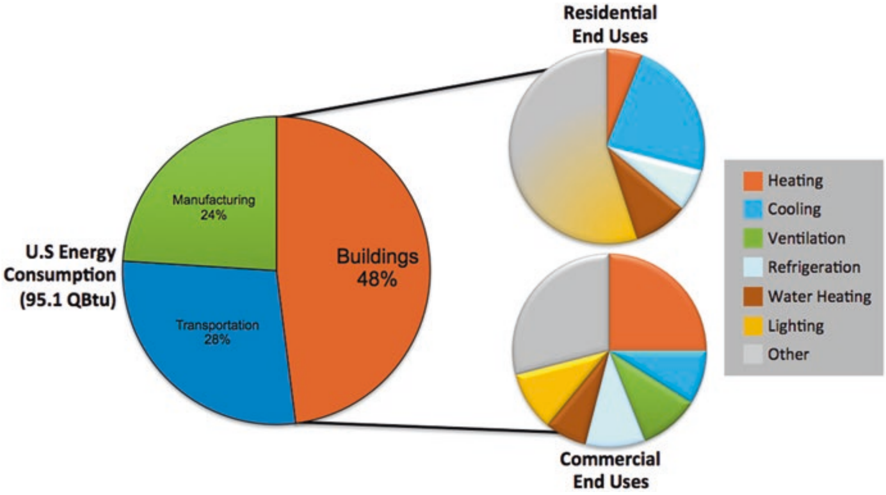


Fig. 1.1 U.S. Energy consumption by sector with end use breakdowns. (Data source U.S. Energy Information Administration 2012)

That is not to say these (and other) sectors did not make use of mathematical models earlier – they did. This brief list is meant to point out significant historical events such as the space race, the 70s fuel crisis, advent of the personal computer, etc. that were significant drivers towards the adoption of rigorous mathematical modeling to meet market challenges. Fortuitously, these needs were enabled by improvements in the computing capability required to perform increasingly sophisticated analysis.

So, what of the topic of this book, the built environment? In a 2012 U.S. Energy Information Administration survey buildings consumed nearly half of the 95.1 Quadrillion BTUs of energy produced in the United States. Figure 1.1 shows overall consumption by sector along with a breakdown of end uses in both residential and commercial buildings.

Given the increased urgency in curbing global greenhouse gas emissions, reducing the carbon footprint of new and existing buildings has become a priority in many jurisdictions. This is evident in increasingly stringent energy efficient building codes and standards,[1,2] voluntary performance certification programs like LEED™ (Leadership in Energy and Environmental Design),[3] the Architecture 2030

[1] American Society of Heating, Refrigeration, and Air-Conditioning Engineers (ASHRAE) Standards and Guidelines Overview (https://www.ashrae.org/standards-research--technology/standards--guidelines).

[2] California Energy Commission (CEC) Title 24 Energy Efficiency Standards (http://www.energy.ca.gov/title24/).

[3] Leadership in Energy and Environmental Design (LEED™) Certification Overview (https://www.usgbc.org/leed).

Commitment,[4] and a host of utility incentive programs that are driving adoption of energy efficiency (EE) technologies. As in other industries, system modeling is a valuable tool for transforming the built environment. That said, why aren't analytical design tools already in widespread use?

Unlike other sectors, the buildings industry does not generally benefit from an "economy of scale." A car manufacturer may amortize investment in analytical capability across many products sold. On the other hand, buildings are usually "one offs" produced by a diverse group of stakeholders including architects, engineering firms, construction companies, owners, and occupants – none of which could would recoup investment during a project. Assuming that foundational modeling capability even existed, the perceived value of building energy modeling (BEM) varies wildly across these stakeholders, and a prospective building owner may not be inclined to procure a model for their project even though it could reduce their long-term operating expenses.

1.2 A Brief History of Building Energy Modeling and Simulation

The path to what we now recognize as BEM and simulation running on personal computers or high performance computing clusters has its roots in the 1960s. Digital computing had evolved to the point that ordinary and partial differential equations representing exterior and interior heat transfer involving buildings could be solved numerically. Unfortunately, even the most rudimentary analysis was restricted to academics or federal employees with access to the mainframe computers of the day. The U.S. Postal Service is recognized as one of the first federal agencies to make significant investment in dedicated software that could assess the thermal performance of buildings.[5]

Additional federal agencies began making their own investments in building energy simulation codes in the early 1970s, largely in response to geopolitical and economic turmoil of the time. The U.S. National Bureau of Standards (NBS), now known as the National Institute of Standards and Technology, created NBSLD, the National Bureau of Standards Load Determination program.[6] NBSLD was a groundbreaking FORTRAN program successfully used to study the growing building energy consumption problem for NBS, the Department of Housing and Urban Development (HUD), and the Department of Defense (DOD).

Around this time, DOD adapted NBSLD's source code into a new simulation software that would be called the Building Loads Analysis and System Thermodynamics program or BLAST.[7] In parallel, the Department of Energy (DOE)

[4] American Institute of Architects (AIA) 2030 Commitment Overview (https://www.aia.org/resources/6616-the-2030-commitment).

[5] Lau and Ayres (1979).

[6] Kusuda (1976).

[7] Hittle (1977).

elected to adapt the Postal Service software into its own FORTRAN code that would come to be known as DOE-2[8] developed at Lawrence Berkeley National Laboratory (LBNL). In 1998, DOD discontinued BLAST and DOE halted support for DOE-2, although DOE-2 development would continue for a number of years with funding from the California Public Utilities Commission (CPUC).

In 1997 and 1998, DOE co-authored several papers, including "Beyond BLAST and DOE-2: EnergyPlus, a New-Generation Energy Simulation Program.[9]" These described in some detail, a technical comparison of the approaches used by both BLAST and DOE-2 along with DOE's roadmap for creating new software that would draw on the best of both predecessors. The new simulation engine would be coded from scratch and known as EnergyPlus. Originally written in FORTRAN, EnergyPlus was converted to C++ in 2014 through a code-contribution from Autodesk. EnergyPlus is made available under a permissive BSD-style open source license in collaboration with multiple national laboratories, universities, contractors, and companies[10] like Trane and Carrier.

Part of the original EnergyPlus design philosophy was that it would be a simulation "engine" only, taking as input a text file and returning simulation results. EnergyPlus text Input Data Files (IDFs) for buildings of any significant complexity were tens of thousands of lines long, requiring precise specification of building geometry, constructions, individual thermal loads, detailed heating ventilation and air conditioning (HVAC) interconnections, and more. Crafting these files by hand was a time consuming and error prone undertaking. It was always DOE's intention that third-party software developers would step up to create graphical user interfaces (GUI) that would author IDFs. Unfortunately, with DOE-2 and its freely available eQUEST GUI available, along with the advent of new simulation engines like ESP-r and ApacheSIM and their own interfaces, DOE investment in a state of the art engine like EnergyPlus was not reaching the marketplace.

In 2008, a researcher at the National Renewable Energy Laboratory (NREL), Peter Ellis, created what he referred to as the OpenStudio SketchUp Plug-In.[11] Google SketchUp, as it was then known, was free and used by a significant number of architects for early concept design. It stood to reason that creating a Plug-In to convert building geometry from SketchUp into IDF would tackle one of the obstacles for modeling and analysis with EnergyPlus. That solved one part of the IDF authoring problem, but what about all of the other data required as input for EnergyPlus? If making the creation of a single model was easier, would it be possible to automate the creation of hundreds or thousands of design alternatives without the use of a laboratory or university super computer? Moreover, was there a way to help third-party software developers unleash the power of EnergyPlus more easily? The next step in the evolution of "OpenStudio" sought to answer these questions and more.

[8] https://buildings.lbl.gov/sites/default/files/lbnl-18046.pdf.

[9] Crawley et al. (1998).

[10] http://energyplus.net.

[11] The SketchUp Plug-In is now called "Euclid" and is available here: http://bigladdersoftware.com/projects/euclid.

1.3 What Is OpenStudio?

For an increasing number of people, it will be hard to recall a time when one didn't use an "app store" to find and download software for personal computers and mobile devices. The advent of operating systems, associated software development kits (SDKs), and app store-like distribution mechanisms has transformed the pace with which software innovations are brought to market. Faced with slow uptake of EnergyPlus, DOE wondered if a similar approach to software development might spur development of BEM-based applications that supported the built environment's many stakeholders.

In 2010, NREL released version 0.1 of a new version of OpenStudio that had been re-envisioned as an SDK that aims to reduce the time and expense of developing new BEM applications.[12] Subsequent quarterly releases increased coverage of EnergyPlus capability, supported DOE's Radiance daylighting analysis engine, added example applications, and introduced entirely new BEM paradigms. OpenStudio has also served as an effective collaboration platform, coordinating contributions from colleagues at Argonne (ANL), Lawrence Berkeley (LBNL), Oak Ridge (ORNL), and Pacific Northwest National Laboratories (PNNL), Pennsylvania State University, National Resources Canada (NRCan), Group 14 Engineering, and others.

Like EnergyPlus, OpenStudio is offered under a BSD-style open source license that allows companies to create and sell their own derivative works built with the SDK. It runs on Windows, Mac, and Linux, and has been used to create web and server applications as well. While the SDK itself is written in C++, code "bindings" allow it to be invoked from other languages including Ruby, C#, Python, and Javascript. As of this writing, nearly two-dozen applications have been produced using the SDK by NREL, other national laboratories, and private sector developers. A few are highlighted in Fig. 1.2, catering to a diversity of end users including building and portfolio owners, architects, engineers, policy makers, and utilities.

In addition to its role in expediting the development of innovative applications, OpenStudio has borrowed a few key concepts from other software domains, adding them to the BEM discourse. We will learn more about them throughout the book, but they are worth introducing here:

- **Application Programming Interface (API)** – Computer systems tend to be built in "layers". Each layer implements some level of functionality and then presents that functionality to the layer above using a convenient API. The API is essentially a software "contract" that relieves developers of higher-level layers from lower-level implementation details and allows layers to evolve largely independently. The OpenStudio SDK contains a rich API that is the basis for creating OpenStudio Measures or full-blown applications. The OpenStudio API also allows client applications to evolve independently from EnergyPlus by insulating the former from changes in the latter. The API is discussed in Chap. 9.

[12] https://www.openstudio.net.

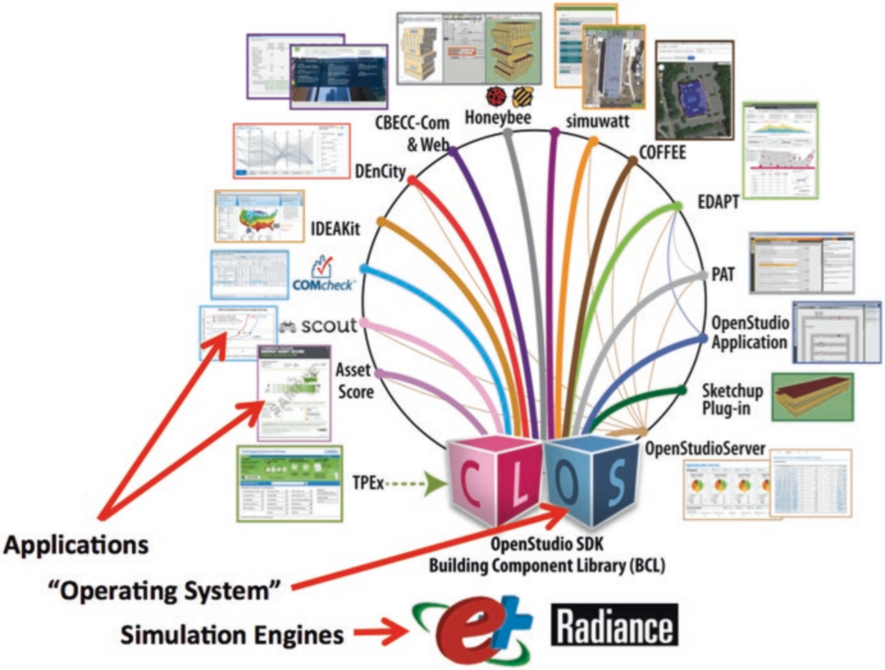


Fig. 1.2 OpenStudio as a BEM operating system. (*Credit: Marjorie Schott*)

- **Hierarchy and Inheritance** – The term "object-oriented" is somewhat broad and encompasses several characteristics. However, one of the main characteristics is the notion of an object hierarchy with parent-child relationships, in which more-specific child objects "inherit" attributes and capabilities from more-general parent objects. OpenStudio makes extensive use of hierarchy and inheritance to improve the efficiency of creating and modifying models. The concepts of hierarchy and inheritance as they apply to BEM will be introduced in Chap. 2.
- **Scripting and Measures** – The OpenStudio API may be invoked from the Ruby scripting language. The OpenStudio SDK itself can execute Ruby programs, essentially creating an extension and automation facility for the SDK that is similar to Visual Basic for Excel. Specially structured OpenStudio Ruby scripts are called Measures because the most common use case is applying an energy efficiency measure (EEM) to a building model to improve simulated performance in the same way the EEM is applied to a building to improve actual performance. However, as we will learn in Chap. 6, OpenStudio Measures are much more powerful and have become one of the core value propositions of the platform.
- **Cloud Computing** – The availability of commodity cloud computing has been game changing for other sectors, and it should be no different for BEM. OpenStudio

makes it easy for anyone[13] to leverage cloud computing for large-scale sampling, optimization, and other analyses. OpenStudio Measures are a key to leveraging cloud computing as they provide a systematic way of defining a large simulation space. We will explore this capability first-hand in Chap. 7.

- **Shared Content and Crowd-Sourcing** – The "open" in OpenStudio comes from open source. Modern open source software projects are largely about freedom to create and commercialize derivative works, but many also try to create a community that encourages—or at least enables—sharing of knowledge and created content. OpenStudio does this via the Building Component Library (BCL), an online repository for OpenStudio content including Measures. The BCL is first introduced in Chap. 2 but will pop up in subsequent Chapters as well.

1.4 Overview of Book Structure

The majority of this text is organized around one of the first example applications built using the OpenStudio SDK, the OpenStudio Application. The "App" supports construction, simulation, and review of individual building energy models. Figure 1.3 is a typical screenshot of the App annotated to highlight its workflow-centric design. Tabs along the left-hand side of the window are generally placed in the order they are used, although not all Tabs are necessarily required for modeling a building. Some Tabs are also broken down into Sub-Tabs identified along the top of the window. In general, the first (left most) Sub-Tab will be most frequently used, while subsequent sections are used in specialized circumstances.

Like the OpenStudio Application, this book is organized according to steps commonly used in a BEM workflow. The Application's Tabs and related Chapters are as follows:

- **Site** – Specify weather, life cycle costs and utility bills (*Chap.* 2)
- **Schedules** – Define schedules applied to building loads (*Chap.* 3)
- **Constructions** – Specify materials, construction assemblies, and sets (*Chap.* 2)
- **Loads** – Define individual building loads (*Chap.* 3)
- **Space Types** – Create profiles for how spaces are occupied (*Chap.* 3)
- **Geometry** – Define the building exterior and interior geometries (*Chap.* 2)
- **Building** – Assign building level defaults and exterior items (*Chaps.* 2, 3, *and* 8)
- **Spaces** – Assign profiles to individual spaces (*Chap.* 3)
- **Thermal Zones** – Group spaces into Thermal Zones and assign Zone Equipment (*Chaps.* 4 *and* 5)

[13] Previously, large-scale BEM analysis was the purview of laboratory or university researchers with access to high performance computers. No small engineering firms can run hundreds of simulations in the space of a few minutes for under $10.

- **HVAC** – Specify heating, cooling, and water systems for the building (*Chaps.* 4 *and* 5)
- **Variables** – Specify additional simulation reporting variables (*Chaps.* 2, 3, 4, *and* 5)
- **Simulation Settings** – Customize simulation settings (*Chap.* 8 *and Appendix A*)
- **Measures** – Assign OpenStudio Measure scripts to a workflow (*Chaps.* 6 *and* 9)
- **Run Simulations** – Perform a single energy simulation (*Chaps.* 2, 3, 4, 5, *and* 6)
- **Reports** – Review simulation results for a single energy simulation (*Chaps.* 2, 3, 4, 5, *and* 6)

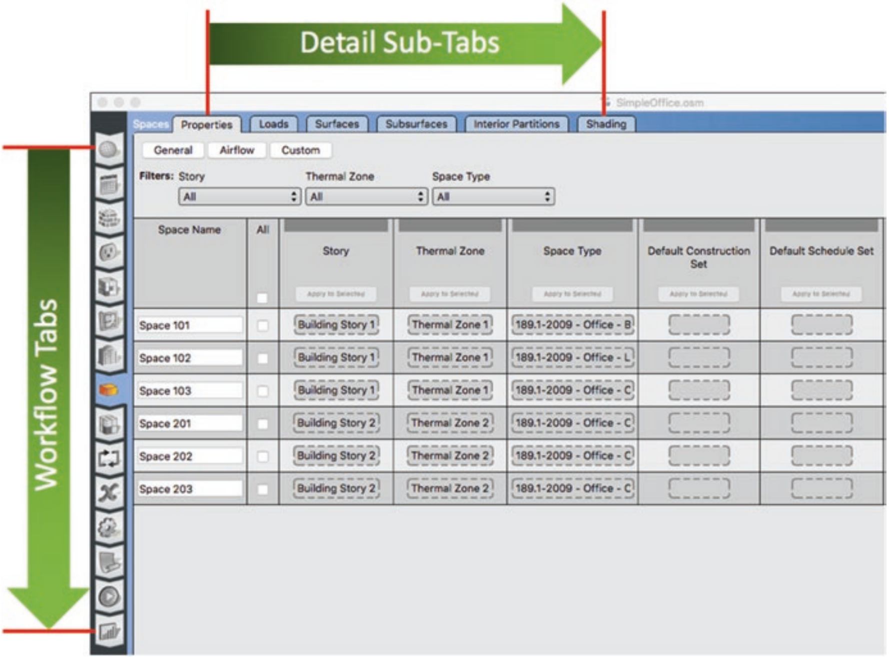
Simulating individual buildings is certainly useful, but the real power of modeling is in the ability to perform comparative analysis of many candidate designs. A second example OpenStudio application aimed at comparative analysis of multiple models is called the Parametric Analysis Tool, or "PAT" for short (Fig. 1.4).

PAT takes one or more models developed using the OpenStudio Application and modifies them using OpenStudio Measures for small or large-scale analysis. PAT is organized according to a tabular workflow as follows:

- **Analysis** – Specify the type of analysis to be performed (*Chaps.* 6 *and* 7)
- **Design Alternatives** – Manually define Design Alternatives for small studies (*Chap.* 6)
- **Outputs** – Specify key analysis outputs for large-scale analysis and visualization (*Chap.* 7)
- **Run Simulations** – Manage local or cloud-based analysis (*Chaps.* 6 *and* 7)
- **Compare Results** – Compare results from small studies (*Chap.* 6)
- **Analysis Server** – Manage and interact with large-scale cloud analyses (*Chap.* 7)

Each Chapter begins with multiple sections describing one or more concepts corresponding to the App or PAT Tabs. Focus then shifts to a series of exercises that are intended to practically demonstrate those same concepts. These are referred to as "Checkpoint" exercises because they are intended to build upon each other successively. That is, Chap. 4's exercise picks up where Checkpoint Four left off at the end of Chap. 3. Each Chapter concludes with some suggestions for additional exercises that allow the reader to explore concepts in greater depth.

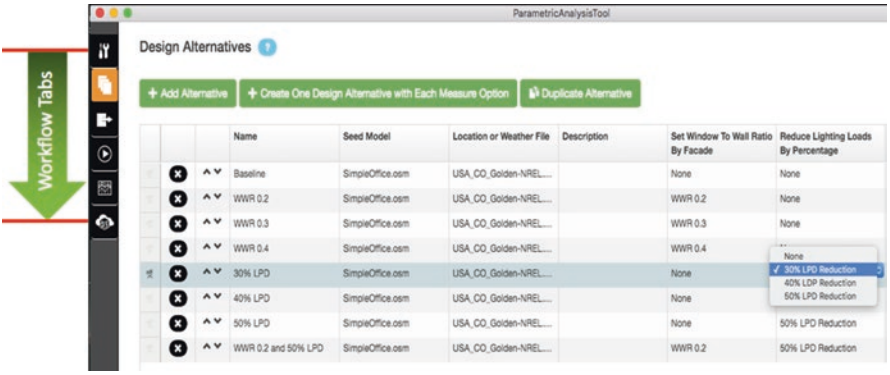
By following the material and performing the exercises in each Chapter, the reader will learn the rudiments of modeling as applied by many BEM tools, and become familiar with the OpenStudio SDK using example applications. Chapter 9 introduces the reader to direct application of the SDK, however it is not intended as a comprehensive guide. Readers requiring more depth on these topics are referred to the resources section of the Appendix for links to in-depth online documentation.



Filters: Story All Thermal Zone All Space Type All

Space Name	All	Story	Thermal Zone	Space Type	Default Construction Set	Default Schedule Set
		Apply to Selected	Apply to Selected	Apply to Selected	Apply to Selected	Apply to Selected
Space 101	□	Building Story 1	Thermal Zone 1	189.1-2009 - Office - B		
Space 102	□	Building Story 1	Thermal Zone 1	189.1-2009 - Office - L		
Space 103	□	Building Story 1	Thermal Zone 1	189.1-2009 - Office - C		
Space 201	□	Building Story 2	Thermal Zone 2	189.1-2009 - Office - C		
Space 202	□	Building Story 2	Thermal Zone 2	189.1-2009 - Office - C		
Space 203	□	Building Story 2	Thermal Zone 2	189.1-2009 - Office - C		

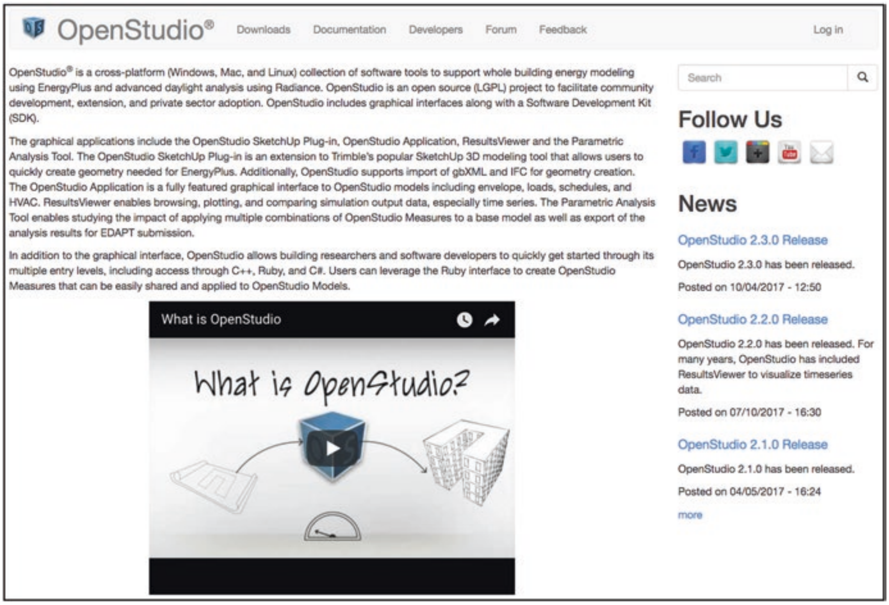
Fig. 1.3 OpenStudio Application workflow and detail



Design Alternatives

			Name	Seed Model	Location or Weather File	Description	Set Window To Wall Ratio By Facade	Reduce Lighting Loads By Percentage
	☒	^ v	Baseline	SimpleOffice.osm	USA_CO_Golden-NREL...		None	None
	☒	^ v	WWR 0.2	SimpleOffice.osm	USA_CO_Golden-NREL...		WWR 0.2	None
	☒	^ v	WWR 0.3	SimpleOffice.osm	USA_CO_Golden-NREL...		WWR 0.3	None
	☒	^ v	WWR 0.4	SimpleOffice.osm	USA_CO_Golden-NREL...		WWR 0.4	
	☒	^ v	30% LPD	SimpleOffice.osm	USA_CO_Golden-NREL...		None	✓ 30% LPD Reduction
	☒	^ v	40% LPD	SimpleOffice.osm	USA_CO_Golden-NREL...		None	
	☒	^ v	50% LPD	SimpleOffice.osm	USA_CO_Golden-NREL...		None	50% LPD Reduction
	☒	^ v	WWR 0.2 and 50% LPD	SimpleOffice.osm	USA_CO_Golden-NREL...		WWR 0.2	50% LPD Reduction

Fig. 1.4 OpenStudio Parametric Analysis Tool (PAT) workflow



OpenStudio® is a cross-platform (Windows, Mac, and Linux) collection of software tools to support whole building energy modeling using EnergyPlus and advanced daylight analysis using Radiance. OpenStudio is an open source (LGPL) project to facilitate community development, extension, and private sector adoption. OpenStudio includes graphical interfaces along with a Software Development Kit (SDK).

The graphical applications include the OpenStudio SketchUp Plug-in, OpenStudio Application, ResultsViewer and the Parametric Analysis Tool. The OpenStudio SketchUp Plug-in is an extension to Trimble's popular SketchUp 3D modeling tool that allows users to quickly create geometry needed for EnergyPlus. Additionally, OpenStudio supports import of gbXML and IFC for geometry creation. The OpenStudio Application is a fully featured graphical interface to OpenStudio models including envelope, loads, schedules, and HVAC. ResultsViewer enables browsing, plotting, and comparing simulation output data, especially time series. The Parametric Analysis Tool enables studying the impact of applying multiple combinations of OpenStudio Measures to a base model as well as export of the analysis results for EDAPT submission.

In addition to the graphical interface, OpenStudio allows building researchers and software developers to quickly get started through its multiple entry levels, including access through C++, Ruby, and C#. Users can leverage the Ruby interface to create OpenStudio Measures that can be easily shared and applied to OpenStudio Models.

Follow Us

News

OpenStudio 2.3.0 Release

OpenStudio 2.3.0 has been released.

Posted on 10/04/2017 - 12:50

OpenStudio 2.2.0 Release

OpenStudio 2.2.0 has been released. For many years, OpenStudio has included ResultsViewer to visualize timeseries data.

Posted on 07/10/2017 - 16:30

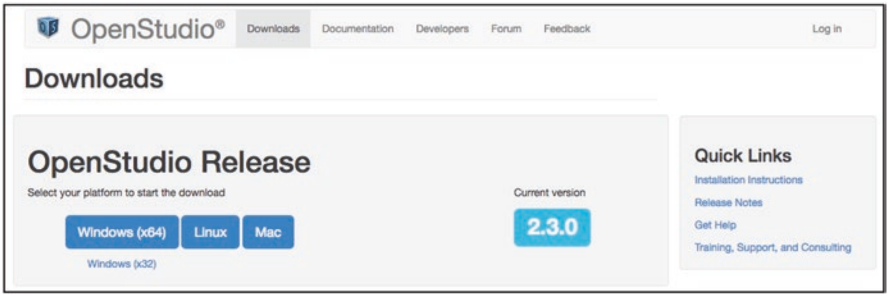
OpenStudio 2.1.0 Release

OpenStudio 2.1.0 has been released.

Posted on 04/05/2017 - 16:24

more

Fig. 1.5 OpenStudio home page (http://openstudio.net)



Downloads

OpenStudio Release

Select your platform to start the download

Current version

Quick Links

Installation Instructions

Release Notes

Get Help

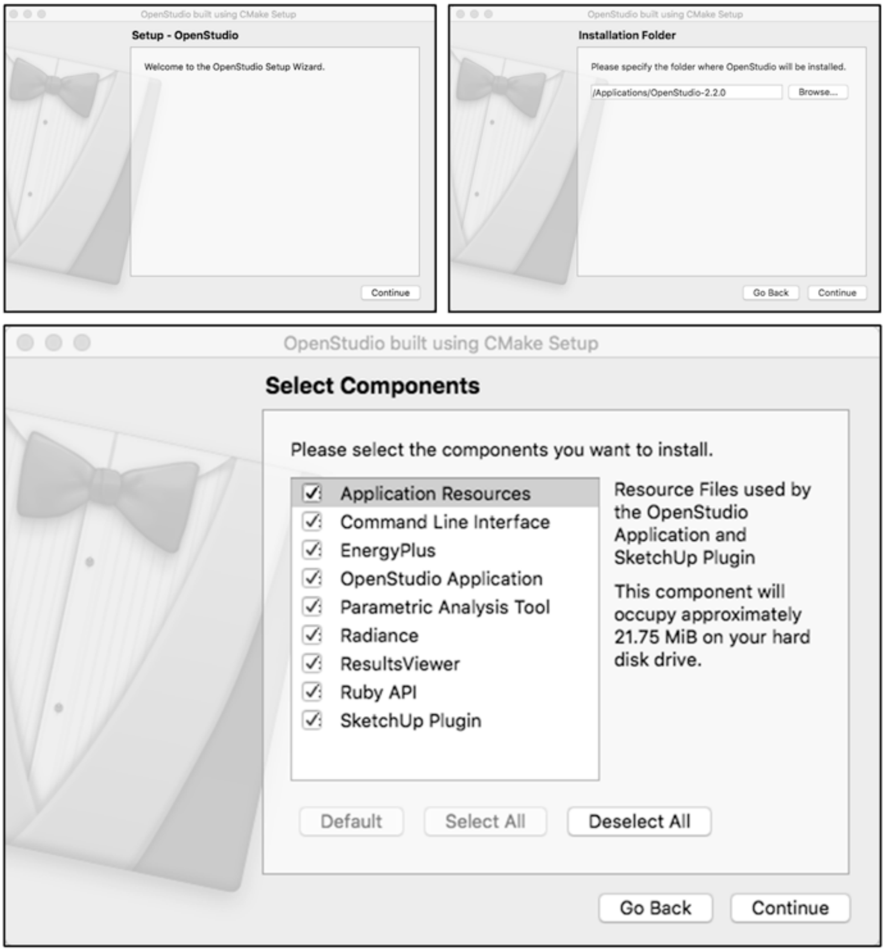
Training, Support, and Consulting

Fig. 1.6 OpenStudio download page

1.5 Installing OpenStudio

The remaining Chapters assume that you have installed OpenStudio and its example Applications on your computer. OpenStudio is a free download available at http://openstudio.net (Fig. 1.5). The OpenStudio website includes downloads, documentation, tutorials, links to instructional YouTube videos, and more.

Clicking the Downloads link at the top of the webpage takes you to the Download page shown in Fig. 1.6. Links for Windows, Linux, and Mac downloads of the latest major release of OpenStudio are available on this page. Major releases are made



OpenStudio built using CMake Setup

Setup - OpenStudio

Welcome to the OpenStudio Setup Wizard.

OpenStudio built using CMake Setup

Installation Folder

Please specify the folder where OpenStudio will be installed.

/Applications/OpenStudio-2.2.0

OpenStudio built using CMake Setup

Select Components

Please select the components you want to install.

- ☑ Application Resources
- ☑ Command Line Interface
- ☑ EnergyPlus
- ☑ OpenStudio Application
- ☑ Parametric Analysis Tool
- ☑ Radiance
- ☑ ResultsViewer
- ☑ Ruby API
- ☑ SketchUp Plugin

Resource Files used by the OpenStudio Application and SketchUp Plugin

This component will occupy approximately 21.75 MiB on your hard disk drive.

Fig. 1.7 OpenStudio installer dialogs on a Mac

every 3 months. Minor release packages are provided more frequently on the Developers page but are not recommended for users as they may include features that are under development and unstable. Select the correct installer for your system and download the package.

Launch the downloaded package and follow the instructions to install OpenStudio (Fig. 1.7). To perform all of the activities described in this book, you will need to install all of the components offered except for the SketchUp Plugin, which is optional. Following a successful installation, it's time to begin learning about BEM and OpenStudio!

References

Crawley D, Lawrie L et al (1998) Beyond BLAST and DOE-2: EnergyPlus, a new-generation energy simulation program. ACEEE Summer Study

Hittle D (1977) The building loads analysis and system thermodynamics program, BLAST, U.S. Army Construction Engineering Research Laboratory, Champaign, IL

Kusuda T (1976) NBSLD, the computer program for heating and cooling loads in buildings, NBS Building Science Series 69

Lau H, Ayres J 1979 Building energy analysis programs. In: Proceedings of the 11th conference on winter simulation, vol 1, pp. 283–289, San Diego, CA

https://www.aia.org/resources/6616-the-2030-commitment

https://www.ashrae.org/standards-research--technology/standards--guidelines

http://bigladdersoftware.com/projects/euclid

https://buildings.lbl.gov/sites/default/files/lbnl-18046.pdf

http://www.energy.ca.gov/title24

http://energyplus.net

https://www.openstudio.net

https://www.usgbc.org/leed