ENERGY DESIGN PLUGIN: AN ENERGYPLUS PLUGIN FOR SKETCHUP*

Peter G. Ellis[1], Paul A. Torcellini[1], and Drury B. Crawley[2]
[1]National Renewable Energy Laboratory, Golden, CO
[2]United States Department of Energy, Washington, DC

ABSTRACT

This paper describes the Energy Design Plugin, a new software plugin that aims to integrate simulation as a tool during the earliest phases of the design process. The plugin couples the EnergyPlus whole-building simulation engine to the Google SketchUp™ drawing program. Leveraging the powerful SketchUp application programming interface, we developed a plugin that extends the capabilities of SketchUp to allow EnergyPlus building models to be developed in 3-D while taking advantage of all of the native SketchUp capabilities, including intuitive tools, different rendering modes, and realistic shading. The model geometry can be saved to create an EnergyPlus input file. Existing input files can be opened, edited in the SketchUp environment, and saved again. Already well-established as a popular tool among architects and designers, SketchUp offers a familiar, easy-to-use interface that, when coupled with the plugin, could make building energy simulation more accessible for architects, designers, and students during the design process.

KEYWORDS

energy simulation, EnergyPlus, SketchUp, graphical user interface, design process, conceptual phase

INTRODUCTION

Although building energy simulation is a useful tool for predicting performance and comparing design options, most energy simulation occurs too late in the design process. In the traditional design process, the energy engineer uses simulation (if at all) as a tool for equipment sizing and code compliance only after the architect has completed the architectural design. Part of the problem is that existing simulation tools are not practical for the design process. Ideally, the design team would use building energy simulations to guide the architectural design from the earliest phases of the project. Experience with real buildings has shown that low-energy design is not intuitive and that simulation should therefore be an integral part of the design process (Torcellini et al. 1999; Hayter et al. 2001). But this is usually not possible because the development of the energy model that describes the building design is time-consuming and requires a skilled specialist.

EnergyPlus is a whole-building energy simulation program developed by the U.S. Department of Energy (DOE). EnergyPlus is the next generation of building simulation program and offers many advanced simulation capabilities (Crawley et al. 2004). However, EnergyPlus is a simulation engine only—it does not have its own graphical user interface (GUI). Manually entering detailed, 3-D geometry data can be difficult and prone to errors. Third-party GUIs for EnergyPlus are approaching maturity, but are not necessarily aimed at the earliest phases of the design process.

In 2005, we began a task to find ways to integrate simulation into the earliest phases of the design process. Work on the Energy Design Plugin began as a pilot project to explore the coupling of EnergyPlus to a commercially-available 3-D drawing package. The concept was to leverage the capabilities of an established GUI and build on an existing market and user base, instead of developing an entirely new program from scratch.

We began further development of the plugin in 2007 with the objective to deploy a version of the Energy Design Plugin for public use. It was decided that the plugin and its source code would be released under an open source license to encourage collaborative development on future versions.

SKETCHUP

Google SketchUp™ 6 is a 3-D drawing program that offers the advanced visualization capabilities of more expensive computer-aided design (CAD) packages, but with a much simpler and more intuitive interface that facilitates the rapid sketching of designs. SketchUp is available in free and professional versions for Microsoft Windows or Mac OS X platforms. (The plugin will work with either free or professional versions, but currently only on Windows. A Mac OS X plugin is under development for a future release.)

The hallmark of SketchUp is its easy-to-use GUI. The program enables a user to easily manipulate and edit designs in 3-D. As with a CAD program, the user can still accurately measure distances and add dimension markings. The program also features a variety of rendering options, including bitmap textures, shadowing, and x-ray mode, as well as traditional rendering modes such as black-and-white line drawings, or a rough "sketchy" style that imitates a hand-drawn architectural draft. By entering the longitude, latitude, date, and time, SketchUp can perform shadowing studies for a project. The shadowing feature can be useful for examining passive solar building designs.

Part of the appeal of coupling EnergyPlus to SketchUp is that it is already a well-known and popular tool among architects, designers, and students. Firsthand accounts suggest that SketchUp is widely used by architects during the conceptual phases of projects. An initial design proposal is rapidly "sketched" with SketchUp to show the building form and massing, and then submitted to the client. The client provides feedback to the architect and requests changes. The architect and client might iterate over several SketchUp models until the client is fully satisfied with the design concept. The project then moves forward to design development, where the SketchUp model is exported to become a much more detailed CAD model. The conceptual phase of the design process—when the SketchUp models are being used by architect and client to make decisions about the building form and massing—is precisely when energy simulation can provide the most helpful feedback to influence the design. SketchUp is optimally positioned in the design process workflow for coupling to an energy simulation tool. Once the project moves to the CAD model, it is usually too late or too expensive to revisit the design of the building form and massing.

Application Programming Interface

The strength and flexibility of the SketchUp application programming interface (API) is another reason that the program was selected for the original pilot project. The SketchUp API allows plugins to be created that add custom functionality to the program.

The API provides access to most of the functionality of the SketchUp user interface. It enables custom controls such as menus, toolbars, and specialized drawing tools to be added. The API can interact directly with the SketchUp model to inspect, create, modify, and delete 3-D objects. The underlying paradigm for the API is fully object oriented.

All the API calls are accessed by using the Ruby programming language (Matsumoto 2001; Thomas 2005). Ruby is a high-level, object-oriented language similar to Smalltalk or Lisp, and is an interpreted language like Perl or Python. Because Ruby programs are not compiled, they are often easier and faster to develop. They do, however, require a separate interpreter program to run them. SketchUp provides its own embedded Ruby interpreter to execute all the code for plugins. Another advantage of Ruby, like many interpreted languages, is that it is platform-independent and can run the same code on Windows, Mac OS X, or Linux. Although Ruby has no native GUI capabilities for handling dialog windows or graphics, it can be coupled to several third-party GUI toolkits.

ENERGY DESIGN PLUGIN

An alpha version of the Energy Design Plugin was released in January 2008 to a small group of testers. The first beta version (0.9.3) was publicly released in early April 2008.

Features

The current release of the plugin (0.9.4) has three major features:

- Geometry editing capabilities
- EnergyPlus run manager
- Data visualization capabilities

First and foremost, the plugin is an easy-to-use geometry editor for EnergyPlus. It allows the user to create a building geometry from scratch: add zones, draw heat transfer surfaces, draw windows and doors, draw shading surfaces, etc. All EnergyPlus geometry objects are drawn with the standard tools provided by SketchUp. The SketchUp model can then be saved as an EnergyPlus input file. Existing input files can also be opened with the plugin, edited in the SketchUp environment, and saved again (Figures 1 and 2).

If the input file is complete, the plugin can use the run manager to launch an EnergyPlus simulation. When the simulation is finished, the Annual Building Utility Performance Summary (ABUPS) generated by

EnergyPlus can be automatically displayed in SketchUp. Other reports, such as the error file or report variable data, can be opened automatically in a text editor or in an Excel spreadsheet, respectively.

Finally, the plugin includes capabilities for data visualization. After a simulation is run, the raw data output file (.ESO) generated by EnergyPlus can be read and parsed by the plugin. A rendering manager allows the user to choose report variables that should be associated by surface or by zone. The rendering manager then paints the SketchUp surfaces with a false-color scale according to the value of the report variable at a given date and time. A color scale or a gray scale can be selected. The user selects the date and time using the SketchUp shadow settings for a snapshot visualization of the data. One application could be to render the surface colors by outside surface temperature, or to render the zones by mean air temperature. An animation manager takes data visualization to the next level by automatically stepping time forward or backward at user-defined intervals to create a dynamic, animated rendering of the data.

Now that we have described what the plugin can do, it is perhaps equally important to clarify what it cannot do. The plugin is not a full-featured interface for EnergyPlus. It does not provide a way to create or edit nongeometry EnergyPlus objects such as materials, constructions, schedules, internal heat gains, and HVAC equipment and systems. Currently, to effectively use EnergyPlus with the plugin, the user must have expertise in building simulation and be willing to work with the EnergyPlus syntax of the text input file. Alternatively, the user could work with the plugin in conjunction with another third-party interface to make the job easier.

The plugin is also not a translator from SketchUp to EnergyPlus. It cannot automatically convert a SketchUp model into an EnergyPlus input file. To use the plugin as intended, the user must have the energy model in mind from the beginning as the SketchUp model is being created.

Audience

The initial target audience for the first release of the plugin is firmly based on EnergyPlus users who wish to rapidly generate building geometry input—often the most time-consuming part of creating an EnergyPlus input file. Prospective EnergyPlus users who were previously put off by the difficulty of manually entering the geometry input will now find it more accessible with the plugin.

Because SketchUp is a popular tool for architects and designers, we have already seen growing demand among professionals who ordinarily would not use simulation but are eager to acquire the appropriate tools. At this stage, the plugin is not ready to go that far mainstream.

If the mission of the Energy Design Plugin is "to integrate energy simulation into the earliest phases of the design process," perhaps implicit in this statement is the requirement that the plugin be easy-to-use for those who are not experts in simulation. After all, simulation experts have not been traditionally involved during the conceptual phase. We might argue that simulation experts *should* be involved from the earliest phases of the project, but the reality is that most projects will not enjoy that luxury. Usually the conceptual phase is the domain of the architect. For future releases, the ultimate goal must be to develop the plugin into a software tool that these professionals would be comfortable using to produce reliable energy simulation results.

Guiding Principles

The overall development of the Energy Design Plugin has been influenced by the following guiding principles:

1. Focus on the conceptual phase

The conceptual phase is a critical time for introducing energy simulation into the design process. In this phase, changes can be made to the building form and massing that can significantly reduce building energy use, yet add little to the cost. By the time the project moves to the later phases of the design process, it is usually too late to make major changes to form and massing.

As one possible option for the conceptual phase, our research in 2005 evaluated integration of energy simulation with CAD tools using Industry Foundation Classes (IFC). Significant research and development has been put into advancing IFC as a means for exchanging building geometry data between CAD programs and energy simulation programs (Bazjanac and Crawley 1997; Karola et al. 2001; O'Sullivan and Keane 2005). Much of this research has specifically targeted EnergyPlus as the energy simulation program.

The fundamental drawback to CAD programs in general is that they are simply not designed for creative design exploration and rapid concept development. This explains why they are seldom used for the conceptual phase. In the end, we decided to develop the plugin around SketchUp because it is the design tool of choice for architects during the conceptual phase.

2. Develop architectural and energy models together

One challenge of the CAD-to-IFC-to-EnergyPlus path is that it is difficult for software to automatically derive an energy model from a full-blown architectural CAD model. In most cases the architectural model contains many details that are irrelevant for energy simulation. The process of trying to identify and simplify the important geometry is complex. Instead of attempting to tackle this problem anew, we opted for an entirely different approach.

The approach we chose for the plugin requires the user to develop the architectural model and the energy model *simultaneously*. The user defines both models explicitly in the same SketchUp environment at the same time. Where appropriate, the two models can share the same geometry. For instance, the user can define a wall as an element of the energy model, yet can also decorate the wall with architectural elements like crown molding, baseboard, and lighting fixtures. Explicitly defining the energy model avoids the difficulties and errors inherent in attempting to automatically infer one model from the other. Instead, the user develops the energy model exactly as required. This approach also offers the added reward of engaging users to begin to think about energy as they create the building design.

The capability to combine the architectural and energy models means that EnergyPlus geometry objects can coexist with purely decorative content such as architectural details, landscaping, Google Earth images, people, furniture, and vehicles—all in the same SketchUp environment.

The current release of the plugin does not implement all the functionality to make this approach a reality, but this capability will be a core feature of version 1.0.

3. Integrate seamlessly with SketchUp

Much of the popularity of SketchUp can be attributed to the design of its GUI. The GUI is the main reason the program is easy to use and can facilitate rapid 3-D drawing.

Where possible, the plugin GUI imitates the SketchUp GUI and strives to integrate seamlessly with it. A familiar GUI helps SketchUp users feel comfortable working with the plugin by providing the same look and feel between SketchUp and the plugin.

The plugin integrates the standard SketchUp tools for drawing and manipulating the view. The plugin automatically recognizes different surface types as they are being drawn. Walls, roofs, and floors are identified by the orientation of their outward normal vectors. Windows and doors are inferred based on their relationship to a base surface.

The plugin is also linked to the SketchUp location settings and unit system settings. When a user changes one of these settings, the plugin is instantly updated with the new values.

4. Allow two-way editing of EnergyPlus input files

The plugin adopts the paradigm of an "editor" for EnergyPlus input files. As an editor, the plugin can be used to write *and* read input files. Most simulation software has a one-way interface: it can only write the input file. Our experience suggests that EnergyPlus users will always want to be able to use other tools such as a text editor to edit their input files. The two-way editing feature of the plugin preserves all the formatting and comments of the input file and changes only the geometry data that were modified through the GUI.

5. Find cross-platform solutions

SketchUp is available on Windows and Mac OS X platforms. Anecdotal evidence suggests that design-oriented professionals such as architects often prefer the Mac OS X platform over Windows. With architects as a key audience for the plugin in future releases, it makes sense to develop a cross-platform solution for the plugin that supports both Windows and Mac OS X.

The current release of the plugin is only available on Windows due to issues with the dialog feature of the SketchUp API for OS X. With more development, a cross-platform solution should be achievable. The Ruby plugin code is already inherently cross-platform. An OS X build of EnergyPlus has been made available since version 2.0.

6. Release as open source

Since the inception of the plugin, we have planned to release it under an open source license. A volunteer-based open source effort may be a viable way for sharing development costs across many collaborators, leveraging scarce federal funding. Open source projects also have a long-standing reputation for producing high-quality software (Fogel 2006).

The current version is released under the GNU General Public License (GPL) (FSF 2007). The license allows anyone to freely copy and redistribute the plugin and its source code and to modify the source code, provided that if it is distributed publicly, the modified plugin must also be released under the GPL.

By embracing open source, we aim to provide a collaborative foundation for building a volunteer community of developers who are interested in improving the plugin. To that end, we wrote all the code with open source in mind. We paid close

attention to the internal plugin architecture, and to useful programming conventions such as detailed comments, consistent naming standards, and consistent formatting. The result is a body of source code that we hope can be readily understood and extended by other developers. The next step will be to set up a project site for the plugin on a publicly-accessible open source web host such as SourceForge.net.

Implementation

The Energy Design Plugin is entirely implemented by interfacing with the SketchUp API. The bulk of the source code is a written in Ruby, but the plugin also contains some code in HTML and JavaScript. Development was facilitated by the API documentation (Google 2008). Note that due to limitations in the API, workarounds were needed in some cases.

A primary job of the plugin is to track the SketchUp geometry as it is drawn by the user. But it is not necessary to track everything the user draws, because the SketchUp model can contain both EnergyPlus geometry and purely decorative geometry. To delineate which geometry is important for EnergyPlus, the plugin relies on the SketchUp concept of a Group. Groups can be created through the normal SketchUp interface as a way to associate or "group" multiple geometric objects together into a single entity. Groups provide a degree of isolation from the rest of the SketchUp model, but can still be opened for editing at any time. In the plugin, when the user creates a new EnergyPlus zone, they are actually creating an empty Group in SketchUp. From this moment forward the plugin knows that it must track changes inside that zone Group. Similarly, shading surfaces that are not associated with a zone (e.g., detached shading surfaces) are also organized into shading Groups.

The mechanism for tracking the changes in a Group is the Observer class provided by the API. When the plugin creates a zone Group or shading Group, it creates and attaches an Observer object that tracks all the events inside that Group. When the user draws or erases a surface, the the Observer notifies the plugin, which can update the EnergyPlus input file by adding or deleting the object. Each surface also has an individual Observer object assigned to it. When the user moves or resizes a surface, the surface Observer notifies the plugin and can update the EnergyPlus surface object with the new vertex coordinates. Observers are the heart of the plugin; they ensure that the EnergyPlus input file is always up to date.

Dialog windows are another important part of the plugin GUI. Dialogs allow the user to edit EnergyPlus surface and zone properties, select simulation run settings, and manipulate various other inputs. The current release of the plugin uses the WebDialog class provided by the SketchUp API to create all of its custom dialogs. A WebDialog is a web browser embedded in SketchUp that can be filled with a Dynamic HTML (DHTML) document. DHTML is a ubiqitous web technology that adds dynamic behavior to an HTML web page using the JavaScript language and Cascading Style Sheets. The plugin uses DHTML to create dialogs that contain typical GUI widgets like buttons, radio buttons, check boxes, and text boxes. The result is a plugin dialog that looks very much like a standard Windows dialog.

Future Development: Version 1.0

As we continue to develop the plugin toward version 1.0, we will add more core capabilities and improve its current capabilities. Descriptions of some of the major new features planned for version 1.0 follow:

1. Persistent association between SKP and IDF

The current release of the plugin manages the EnergyPlus input file (.IDF) separately from the SketchUp file (.SKP). A new feature will implement a persistent association between SKP and IDF. When the user opens the SKP file, the IDF file will be opened automatically. When the user saves the SKP file, the IDF file will also be saved automatically. A persistent association will allow the user to include decorative SketchUp elements along with the EnergyPlus objects. This feature is an important step for realizing the full potential for developing the architectural and energy models together in the same SketchUp environment.

2. Automatic interzone connections

The process of assigning interzone connections between heat transfer surfaces is a boring chore under the current release. The user must manually set the boundary conditions on each surface separately. A new feature for automatically detecting interzone connections will allow the user to move one zone to touch another zone and have the plugin automatically set the boundary conditions on the adjoining walls, and subdivide the surfaces as necessary.

3. Advanced dialog GUI

Version 1.0 of the plugin will shift away from using DHTML to create the plugin dialogs. An open source GUI toolkit called wxRuby (2008) has been selected to replace the current DHTML dialogs. WxRuby is a wrapper for the cross-platform wxWidgets C++ library. WxWidgets provides a superior GUI look and feel because it uses the native widgets on each platform.

Advanced dialog capabilities will make it easier to develop some of the GUI-intensive features below.

4. Construction manager

The current release requires the user to use a text editor or other interface to prepare constructions and materials beforehand. A new construction manager feature will allow the user to add, edit, and delete constructions and materials from an EnergyPlus input file, and manage libraries of constructions. The user will be able to assign a bitmap texture to each EnergyPlus material and render the energy model in texture mode. A "paint can" tool will allow the user to apply constructions to surfaces with a single click.

5. Graphical display of simulation results

In the current release, the standard EnergyPlus ABUPS report is used to present the simulation results. A new feature will postprocess the results and show a graphical display of the annual energy use. The results will be presented as a color bar chart with energy delineated by end use. Other graphical presentations may also be possible.

6. Wizard for automatic model generation

The current release is mainly a geometry editor. A new feature will expand the plugin beyond geometry to provide an interactive wizard that can automatically generate a prototype or "starter" model from scratch. The wizard dialogs will collect a set of high-level inputs from the user, including site location, building energy code, total floor area, number of floors, general building shape, building use, and HVAC system type. These data will be uploaded to a web service that shares the same preprocessor backend as the EnergyPlus Example File Generator (EEFG 2008). The preprocessor is a component of a larger multivariate optimization tool (Ellis et al. 2006). The preprocessor will automatically generate a fully-functional EnergyPlus input file from the metadata. The input file will then download to the user's computer and can be opened in SketchUp for further editing.

Future enhancements to the web service will also allow the plugin to upload an existing input file, automatically generate a code-compliant reference model, and download the revised input file back to SketchUp again.

In the future, the web service will be expanded to automatically generate a code-compliant EnergyPlus building model from an existing input file.

Future Development: Beyond 1.0

Because the plugin is an open source project, the volunteer developer community will likely play a significant role in determining the direction of development beyond version 1.0. We imagine that developers will want to add support for more and more EnergyPlus objects. Besides low-level EnergyPlus inputs, we also imagine that developers will want to work toward a higher level interface for the plugin that makes energy simulation accessible to architects, engineers, and students. To borrow from SketchUp's mantra "3-D for everyone," the ultimate objective will be to provide "energy simulation for everyone." The research question now becomes, "How do we best provide an easy-to-use interface for a complicated energy simulation program such as EnergyPlus?"

CONCLUSION

The development of the Energy Design Plugin is an ongoing experiment in coupling an energy simulation engine to a commercially available 3-D drawing package via an API. As a prospect for integrating simulation into the earliest phases of the design process, the proposition carries benefits and risks.

The plugin clearly benefits from SketchUp's easy-to-use 3-D GUI by saving us from developing this capability from scratch. The plugin also benefits from the extensive user base and popularity that SketchUp enjoys. Architects, a large segment of the SketchUp user base, are the users we want to reach during the design process. Finally, the plugin benefits from SketchUp's unique role as a conceptual phase tool.

The main risks associated with coupling to SketchUp arise because we have no direct control over the future of SketchUp or the SketchUp API. We are forced to develop the plugin within the constraints imposed by the API, including any limitations, missing features, and bugs.

Tallying these benefits and risks, we suggest that the benefits are concrete and indispensable and the risks are hypothetical. There are no real technical barriers that cannot be overcome. Having come this far, we remain satisfied with our decision to couple EnergyPlus to SketchUp.

ACKNOWLEDGMENT

This work was supported by the Building Technologies Program within the Office of Energy Efficiency and Renewable Energy at the U.S. Department of Energy.

REFERENCES

Bazjanac, Vladimir, and Drury B. Crawley. 1997. "The Implementation of Industry Foundation Classes in Simulation Tools for the Building Industry," in *Proceedings of Building Simulation 1997, Volume I*, September 7-10, 1997, Prague, Czech Republic, pp. 203-210.

Crawley, D.B, L.K. Lawrie, C.O. Pedersen, F.C. Winkelmann, M.J. Witte, R.K. Strand, R.J. Liesen, W.F. Buhl, Y.J. Huang, R.H. Henninger, J. Glazer, D.E. Fisher, D.B. Shirey III, B.T. Griffith, P.G. Ellis, and L. Gu. 2004. "EnergyPlus: An Update," *Proceedings of the SimBuild 2004 Conference*, August 4-6, 2004, Boulder, CO.

EEFG. 2008. EnergyPlus Example File Generator. http://www.eere.energy.gov/buildings/energyplus/interfaces_tools.html

Ellis, Peter G., Brent T. Griffith, Nicholas Long, Paul Torcellini, and Drury Crawley. 2006. "Automated Multivariate Optimization Tool for Energy Analysis," *Proceedings of the SimBuild 2006 Conference*, August 2-4, 2006, Cambridge, MA.

Fogel, Karl. 2006. *Producing Open Source Software*. Cambridge: O'Reilly Media. http://producingoss.com

FSF. 2007. Free Software Foundation. "GNU General Public License, Version 3." http://www.fsf.org/licensing/licenses/gpl.html

Google 2008. SketchUp Ruby API Developer's Guide. http://download.sketchup.com/OnlineDoc/gsu6_ruby/Docs/index.html

Hayter, Sheila J., Paul A. Torcellini, Richard B. Hayter, and Ron Judkoff. 2001. "The Energy Design Process for Designing and Constructing High-Performance Buildings," Clima 2000/Napoli 2001 World Congress - Napoli (I), 15-18 September 2001.

Karola, Antti, Hannu Lahtela, Reijo Hänninen, Rob Hitchcock, Qingyan Chen, Stephen Dajka, and Kim Hagström. 2001. "BSPro COM-Server—Interoperability Between Software Tools Using Industry Foundation Classes," *Proceedings of Building Simulation 2001*, Rio de Janeiro, Brazil, August 2001, pp. 747-754.

Matsumoto, Yukihiro. 2001. *Ruby in a Nutshell*. Cambridge: O'Reilly Media.

O'Sullivan, Barry, and Marcus Keane. 2005. "Specification of an IFC Based Intelligent Graphical User Interface to Support Building Energy Simulation," *Proceedings of the Ninth International IBPSA Conference*, August 15-18, 2005.

Thomas, Dave, with Chad Fowler and Andy Hunt. 2005. *Programming Ruby: The Pragmatic Programmers' Guide, Second Edition*. The Pragmatic Programmers.

Torcellini, Paul A., Sheila J. Hayter, and Ron Judkoff. 1999. "Low-Energy Building Design—The Process and a Case Study," *ASHRAE Transactions*, V 105, Part 2, pp. 802-810. Atlanta, GA: American Society of Heating Refrigerating and Air-Conditioning Engineers.

wxRuby. 2008. A Ruby wrapper for wxWidgets. http://www.wxruby.org

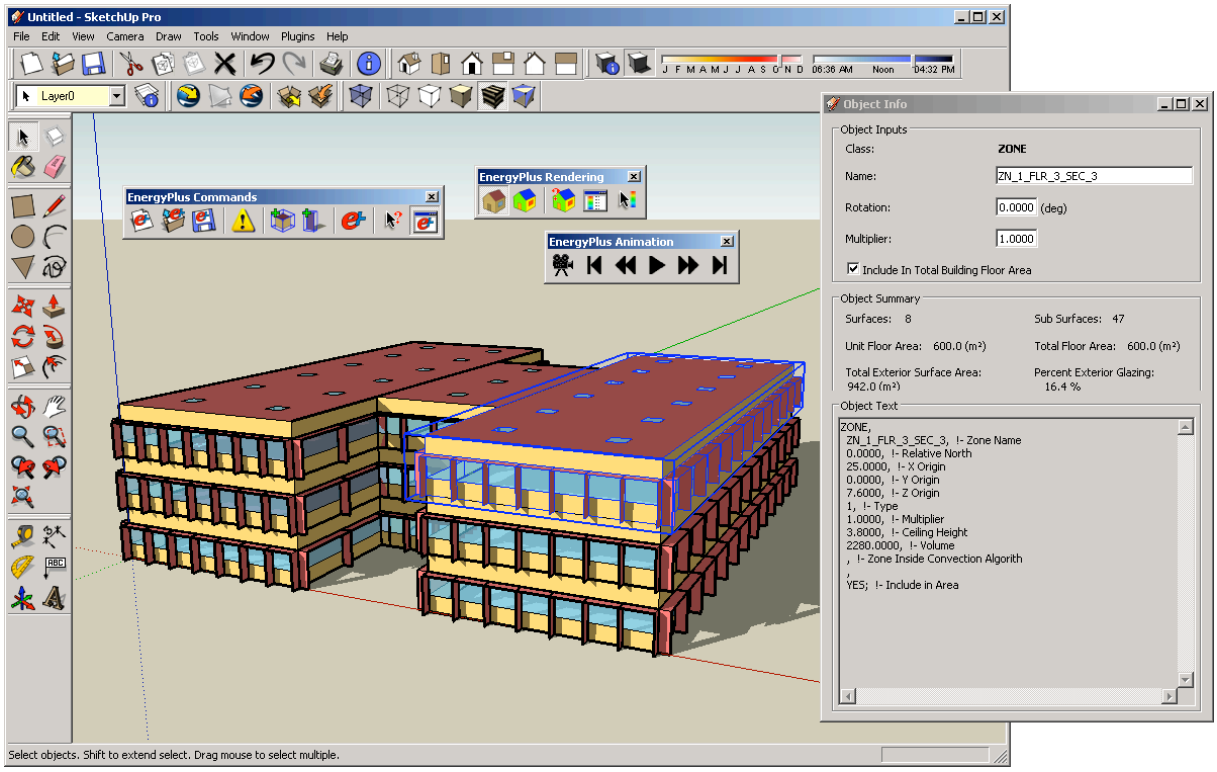

Figure 1 Opening an EnergyPlus input file for editing

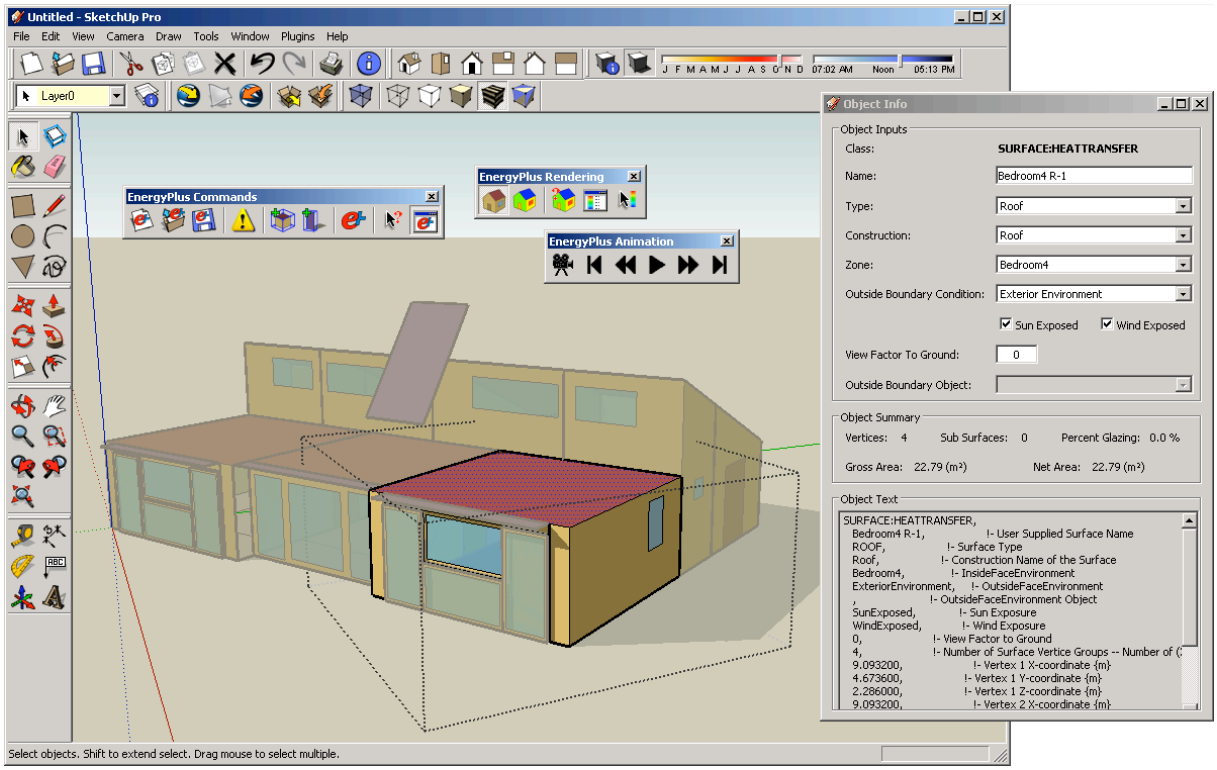

Figure 2 Editing an EnergyPlus zone