

Best Practices Guidebook on Advanced Occupant Modeling

Sara Gilani, William O'Brien

Department of Civil
and Environmental
Engineering,
Carleton University

Basic theories

Why do occupants matter in buildings?

Occupants have been widely recognized as a significant influential factor on the building energy performance, which lead to the gap between the predicted and real energy performance of buildings. For example, De Wilde (2014) reported that the measured electricity energy use of a case study deviated by 30% from what was simulated.

Additionally, occupants may affect the energy use of identical buildings in a wide range. For example, the annual non-HVAC electricity energy consumption in Saldanha and Beausoleil-Morrison's (2012) study on 12 residential buildings varied by a factor of five. Simulation analysis by Haldi and Robinson (2011) showed that the energy demands of similar offices varied by a factor of two.

Therefore, truly representing occupants' presence and behaviour in a simulation-aided design process is very important. If occupants are not taken into account properly as they use buildings in reality, simulation results and the resulting simulation-aided building designs are prone to two risks (Gilani et al., 2016): (1) non-representative predicted building performance that may not accurately reflect what is going on in reality, and (2) poor design decision-making based on the non-representative predicted building performance.

What are occupant models?

Occupants are active agents, rather than passive. They find a way to restore their comfort conditions if they don't feel comfortable in their environments. They may adapt buildings to their comfort through interacting with buildings' systems and components. For example, they may switch on lights, change thermostat setpoint, open or close an operable window, and open or close window shades. Furthermore, they may adapt themselves to their environments. For example, they may change their clothes or drink cold or hot beverages. These kinds of behaviors are called adaptive behaviors (Gunay et al., 2013). On the hand, there are occupant behaviors which are not to restore comfort, rather they are motivated by contextual factors (O'Brien and Gunay, 2014). These behaviors are classified as non-adaptive behaviors (Gunay et al., 2013). For example, occupants switch off their computers or lights more likely when they are going for a vacation (Gunay et al., 2016).

Adaptive and non-adaptive behaviors can be modeled through four main occupant modeling approaches. These modeling approaches can be categorized as: (1) **static** or **dynamic**, and (2) **deterministic** or **stochastic** (aka probabilistic) (O'Brien et al., 2018).

Static occupant models do not capture the impacts that a building and its occupants can have on each other; whereas **dynamic** models mimic two-way interactions between a building and its occupants. Static models are most suitable for non-adaptive behaviors. Dynamic models are most suitable for adaptive behaviors as they can mimic environmental conditions that trigger occupants' actions on changing the state of buildings' systems and components.

Deterministic models give the same results every time a simulation is run. So, just one simulation run is adequate if these models are used in simulation. In contrast to deterministic models, **stochastic** models yield different results every time a simulation is run. This variation in the results is because the parameters of occupant models are chosen randomly based on the properties (i.e. mean and standard deviation) of the models' parameters. Therefore, multiple simulation runs are required when probabilistic occupant models are used in simulation.

How are occupants simulated now?

The most common occupant models used in industry practice are static-deterministic models. Averaged schedules and nominal density (e.g. for lighting, electric equipment, and occupancy) which vary hourly and daily are the form of these models used for code compliance. These models are easily implemented in simulation while they impose two main limitations. First, they do not account the two-way impact that a building and its occupants can have on each other. For example, occupant may close window shades of smaller windows less frequently as they cause a lower rate of glare than larger windows. However, static-deterministic models assume manually-controlled window shades are open all time. The other limitation of static-deterministic models is that they neglect the difference between occupants. Consequently, the uncertainty of occupant behavior is not taken into account and we will not have the opportunity to provide a robust building design (i.e. a design which is less affected by occupant behavior) (O'Brien and Gunay, 2015).

What are advanced occupant model forms?

Advanced occupant models which are well known among the current occupant behavior researcher community are the stochastic models. These models can have a range of statistical forms. We discuss here the most common forms that the current stochastic occupant models have. The most common occupant model forms are: (1) Markov chain models, (2) Bernoulli models, and (3) survival models (Parys et al., 2011).

Using **Markov chain models**, we predict whether an occupant takes an action in the next timestep or event. There are two types of Markov chain models: (1) discrete-time, and (2) discrete-event. **Discrete-time** Markov models use environmental conditions in the current timestep as predictors to predict whether an occupant takes an action. For example, indoor illuminance in the current timestep is used to predict whether an occupant feels a building space is dark enough to turn on lights. **Discrete-event** Markov models use a specific event to predict whether an occupant takes an action. For example, Reinhart's (2004) light switch-on model simulate occupants in a way that they are more likely to turn on lights when they arrive a building space.

Despite Markov chain models, **Bernoulli models** predict the state of a building system or component. For example, Haldi and Robinson (2009) developed window opening models using the Bernoulli model form to predict whether a window is open, rather than if an occupant open or close a window.

The other model form, which is also common in occupant modeling, is **survival model**. This model form is used to predict the duration of a state right before an event happens. For example, Wang et al.'s (2005) occupancy model uses a survival model to predict how long an occupant will go for a lunch or coffee break.

The common statistical model form which is used for Markov chain and Bernoulli models are **logistic regression** models, as the dependent variable is just two categories: whether an action happens or not. Survival models can take different distribution functions. For example, they can have the **exponential** or **Weibull** distribution.

Which occupant modeling approach to use?

Deciding which occupant modeling approaches are most suitable for which application depends on: (1) What is our aim from simulating a building or a room-level model? (2) How big is the building we simulate? (3) What type of building we simulate?

Here, we provide a guideline on which occupant modeling approaches are most suitable for each use case for the design of office buildings. See Table 1 for a summary of applications of the four occupant modeling approaches that we have discussed.

Whole building energy prediction: If our main objective of simulation is predicting the annual average energy use of a medium to large-scale building, static-deterministic models give reasonable estimation. For example, Gilani et al. (2018) showed that static-deterministic lighting use models provides a reasonable approximation of the annual lighting energy use of buildings larger than 100 offices.

Building system and plant equipment sizing: If predicting the peak energy use of a medium to large-scale building for HVAC equipment sizing is our aim of using simulation, static-stochastic models are most suitable models to avoid oversizing system and plant equipment. For example, O'Brien et al. (2018) found that the static-stochastic models give similar predictions to static-deterministic (i.e. standard schedules) for zone-level equipment sizing; however, the system and plant equipment were oversized using the static-deterministic models. In other words, when the distribution (uncertainty) of the predicted energy use (e.g. peak loads) is important, static-stochastic models are most suitable.

Net-zero energy buildings: If we want to design a net-zero energy building design, we will be interested in the energy use of the whole building. In addition, the absolute energy use of the building is the output that we want to obtain from simulation. Therefore, the accuracy of energy prediction is important. In these cases, static-stochastic and dynamic-stochastic models are most suitable to consider the impact of the uncertainty of occupant behaviors on the design and energy use. With these models, we can design more energy-efficient buildings and provide a more accurate estimation of the building's energy use. For example, Abdelalim and O'Brien (2018) used static-stochastic models for PV system sizing of a net-zero building. They found that the PV sizing is highly affected by the uncertainty from occupants to achieve a net-zero energy building.

Occupant comfort: When we want to analyze occupants' comfort, we are mostly interested in the room-level energy and comfort analysis. As we discussed earlier, occupants undertake adaptive behaviors at the room level to mitigate their discomfort. So, when we want to design a building which is more robust to occupant behavior, we try to find a room design alternative that most occupants feel comfortable in it. In other words, we try to reduce occupants' interactions (which can be used as an indicator of occupants' discomfort) with zone-level's components to increase the robustness of a room-level design to occupant behavior. In such cases, we are interested in the distribution of energy use and number of occupants' interactions. So, for these kinds of simulation outputs, dynamic models, either deterministic or stochastic, are most suitable. For example, O'Brien and Gunay (2015) performed an analysis on using dynamic-stochastic models to achieve a robust office design by reducing occupants' interactions with buildings. Their analysis showed that fixed exterior shading can reduce occupants' use of interior roller shades and lights.

Façade design: When we want to design a building façade, occupant comfort is an important factor that we need to consider. Occupants may take adaptive behaviors to mitigate their discomfort. So, it is important how we design fixed and moveable shades (for daylighting and visual comfort) and how we design operable windows (for thermal comfort and indoor air quality). We are looking for a façade design for each room that an occupant feels comfortable in the space and the energy use is at the minimum level. Also, we want to design a room in which most occupants feel comfortable; so, we are interested in the distribution of the simulation outputs. Therefore, for designing buildings' façade, we are using dynamic models, either deterministic or stochastic, for adaptive behaviors instead of using static models. For example, Gilani et al. (2016) showed that how the near-optimal window size for daylighting using dynamic-stochastic models is

different from the static-deterministic model prediction. For non-adaptive behaviors that an occupant may take in a room, such as using plug-in appliance that can affect the energy use of a room, we can still use static models.

Table 1. Application of the four occupant modeling approaches for each use case.

Use case	Modeling approach			
	Static		Dynamic	
	Deterministic	Stochastic	Deterministic	Stochastic
Whole building energy prediction	✓			
Building system and plant equipment sizing		✓		
Net-zero energy buildings (e.g. PV sizing)		✓		✓
Occupant comfort			✓	✓
Façade design			✓	✓

Workflow overview

First of all, we will make the model in SketchUp Make. We will add additional inputs to the model in OpenStudio. Then, we simulate occupants by adding occupant models to the building model in EnergyPlus. We will also see how to postprocess the data and visualize them, and how useful dynamic occupant models are for designing a building (Figure 1).

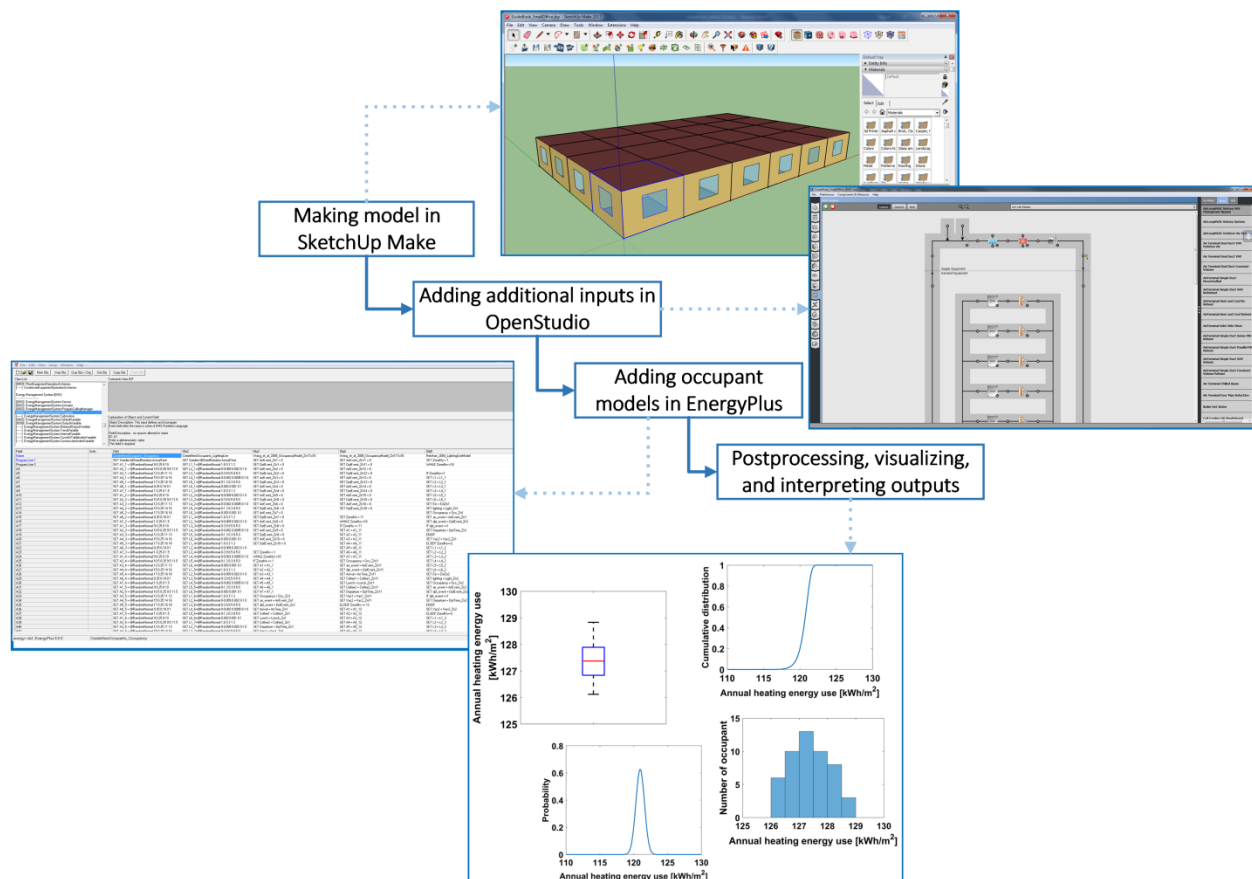


Figure 1. Workflow of preparing and analyzing a model for occupant study.

Getting Started

For getting started, we first download and install three free softwares: SketchUp Make, OpenStudio, and EnergyPlus. Also, we download the Legacy OpenStudio SketchUp Plug-in if our SketchUp Make did not have the OpenStudio extension (check **Extensions** in the menu bar).

1. Download and install [OpenStudio V2.4](#)
2. Download and install [SketchUp Make 2017](#)
3. Download and install [EnergyPlus V8.8](#)
4. Download [Legacy OpenStudio SketchUp Plug-in V1.0.14](#) (if applicable)

Making model in SketchUp Make

For making the model in SketchUp Make, first we check if the SketchUp has the OpenStudio extension from **Extensions** in the menu bar. If the SketchUp did not have the OpenStudio extension, we add the Legacy OpenStudio SketchUp Plug-in to SketchUp Make. We choose **Window > Preferences** from the menu bar. On the right side of **System Preferences**, we enable **Legacy OpenStudio** if we have already downloaded it (Figure 2).

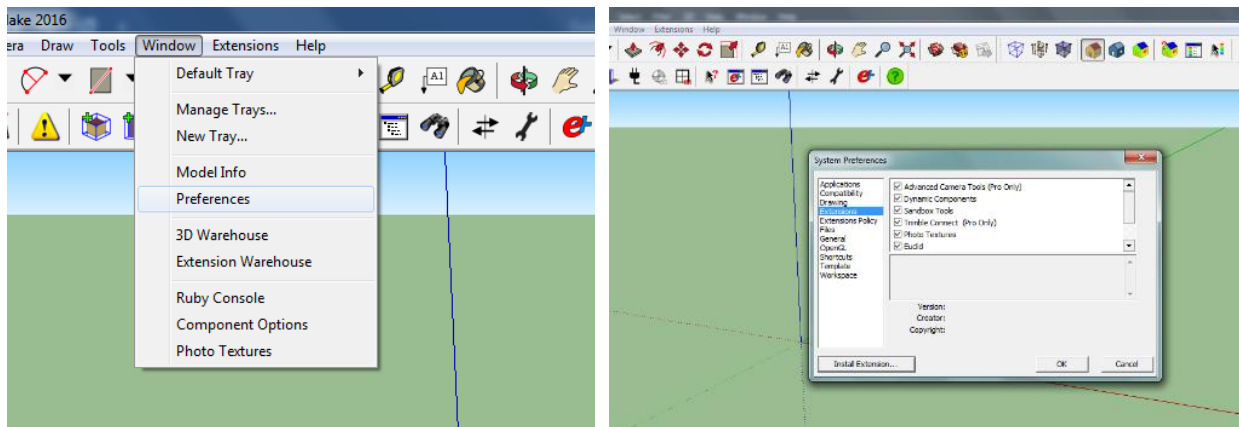


Figure 2. Adding OpenStudio plug-in to SketchUp Make.

Step 1. Making building geometry

Here we will make a model of a one-storey building with independently-controlled 16 perimeter private offices and one core open-plan office in Ottawa, Canada (Figure 3).

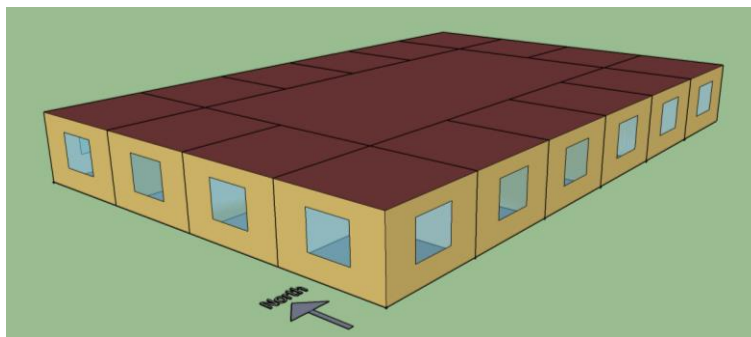


Figure 3. Geometry of building model.

To make the geometry of a building model, first we make the floor plan of the building. Each thermal zone should have its own plan. Note that the true north is aligned with the green axis in SketchUp Make (Figure 4).

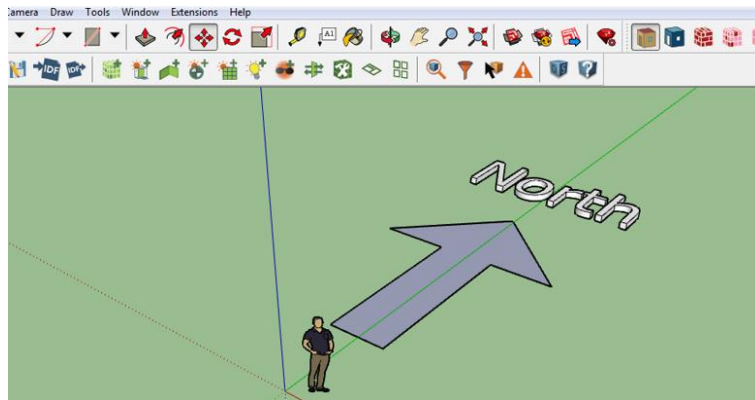


Figure 4. North direction in SketchUp Make.

Let us make the model of the office which is located on the south-west corner. We make the floor plan of this office using the **Rectangle** from the drawing toolbox. Then, we create the space from the floor plan that we have drawn using **Create Spaces from Diagram** from the OpenStudio toolbox (Figure 5).

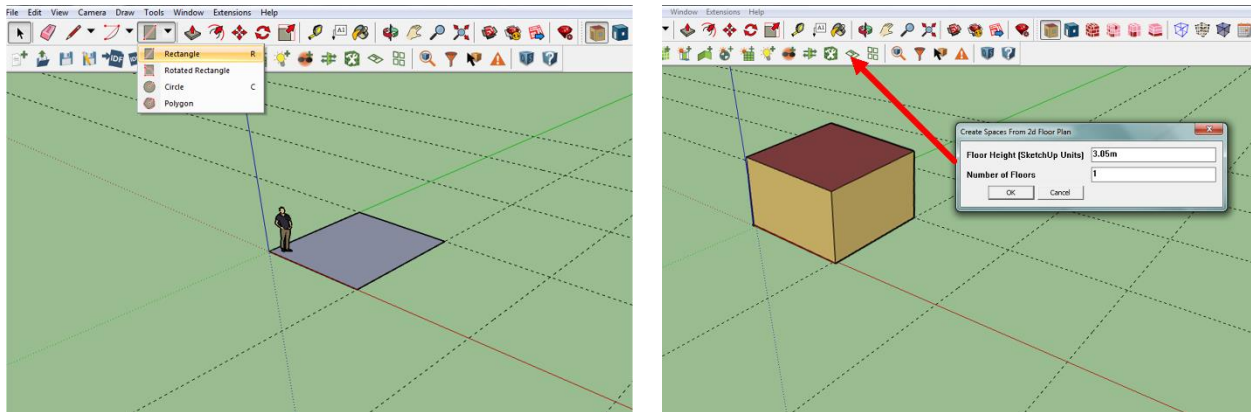


Figure 5. Making a room space in SketchUp Make.

Once we made the space from the floor plan, we make windows in the walls by first drawing them using **Rectangle** and then making them in the walls using **Project Loose Geometry** from the OpenStudio toolbox (Figure 6).

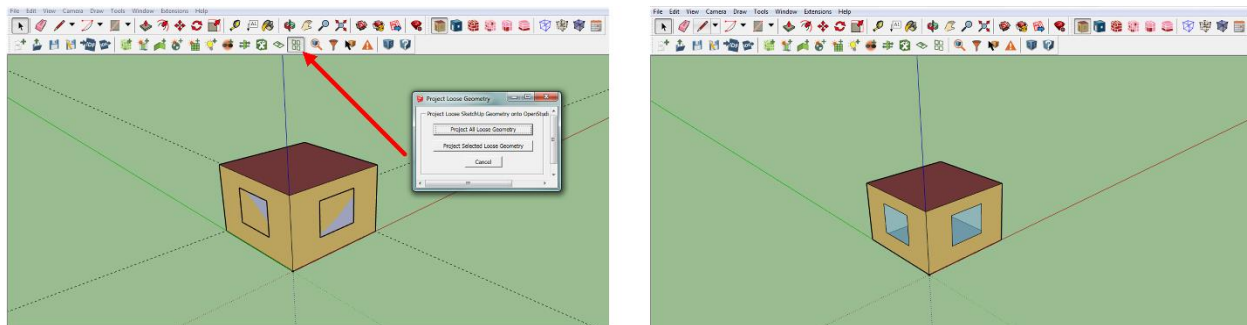


Figure 6. Making fenestrations of a room space in SketchUp Make.

Step 2. Setting attributes of building spaces

Once we made the geometry of the building space, we will identify the attributes of building spaces, such as: space type, building storey, construction set, thermal zone, zone's ideal air loads status, and zone's thermostat. So, we first select the building space. Then, we use **Set Attribute for Selected Spaces** from the OpenStudio toolbox and assign space type, building storey, construction set, thermal zone, and thermostat by choosing the building standard that we want our design complies with and the climate zone of our site (Figure 7).

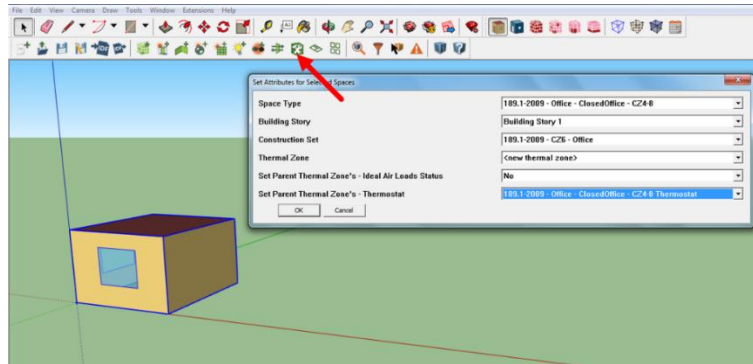


Figure 7. Setting attributes of building spaces in SketchUp Make.

Using the same procedure, we make the geometry of all the spaces and identify their attributes. Note that in our building model, since the offices are controlled independently, we assign new thermal zone to each of these offices.

Additionally, since each of the 16 perimeter offices are controlled independently, the periphery wall of the core office is adjacent to varying thermal conditions because of the different thermal conditions of the perimeter offices. Therefore, we divide the core office to 2 by 16 offices (see Figure 3), so that we can separate the common wall between each perimeter office and the core office from the other walls on the periphery of the core zone. However, we want to have just one thermal zone for the core open-plan office. So, we assign all the walls of the 2 by 16 offices of the core office, except for the walls adjacent to the perimeter offices, as interior partitions.

Step 3. Setting boundary conditions

After we create the building geometry and identify spaces' attributes, we define boundary conditions of each thermal zone. We use **Info Tool** from the OpenStudio toolbox to find the names of each surface of the building spaces (Figure 8).

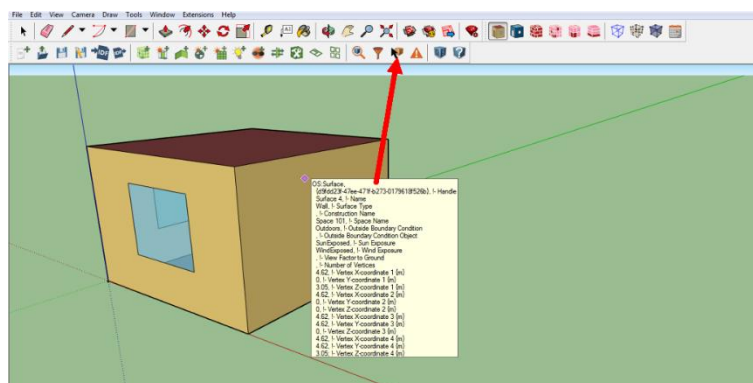


Figure 8. Using "Info Tool" in SketchUp Make for information of surfaces.

When we found surfaces' names, we use **Inspector** from the OpenStudio toolbox to set boundary conditions of each surface (Figure 9). Also, we can edit other objects, such as surface name and type, construction name, and space name.

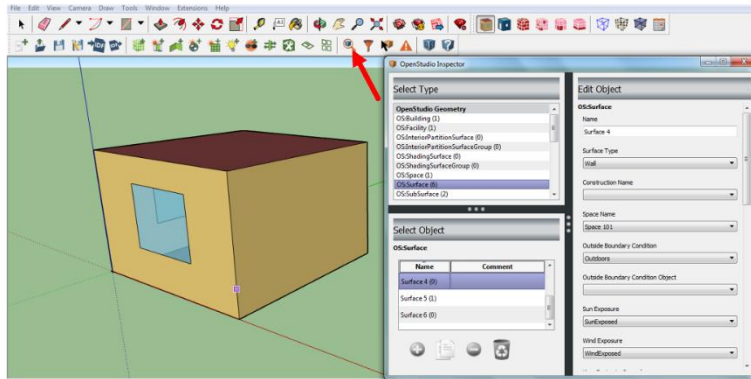


Figure 9. Setting boundary conditions in SketchUp Make.

Step 4. Exporting model to OpenStudio

Once we made the building geometry and set attributes of spaces and boundary conditions, we export the OpenStudio model by selecting **Extensions > OpenStudio > Export > Export OpenStudio Model** (Figure 10).

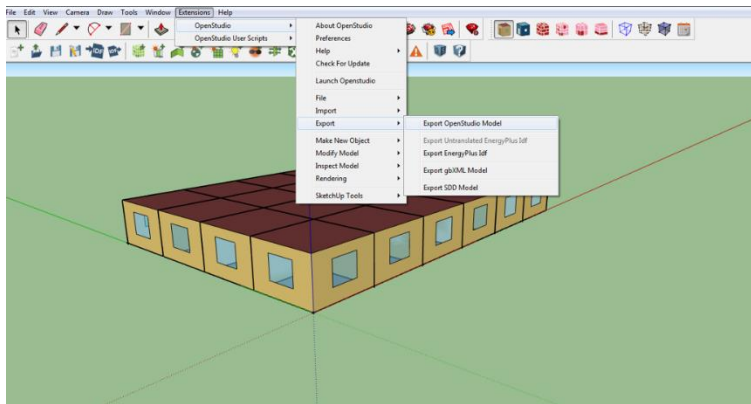


Figure 10. Export model from SketchUp Make to OpenStudio.

Adding additional inputs in OpenStudio

We have made the geometry of our building model and set the attributes of the building spaces and boundary conditions in SketchUp Make. Since we have already identified the spaces' type and construction in SketchUp Make, we do not need more to set schedules, constructions, and loads in OpenStudio. In case we want to modify them, we can use **Schedules, Constructions, Loads, Space Types, Spaces, and Thermal Zones** tabs on the left side of the OpenStudio window. Now, we want to add other inputs, such as building's site, HVAC system, and simulation settings, to our model in OpenStudio.

Step 1. Site

The first step is to define where our building is located. So, we set the EPW file from the weather data folder using **Change Weather File** which we can find in the **Site** tab on the right side of the OpenStudio window (Figure 11). Here, we also set design days in **Design Days** by importing the data from the DDDY

file that is included in the weather data folder. After we import the DDY data, we identify which winter and summer design days we want to use for sizing the HVAC systems.

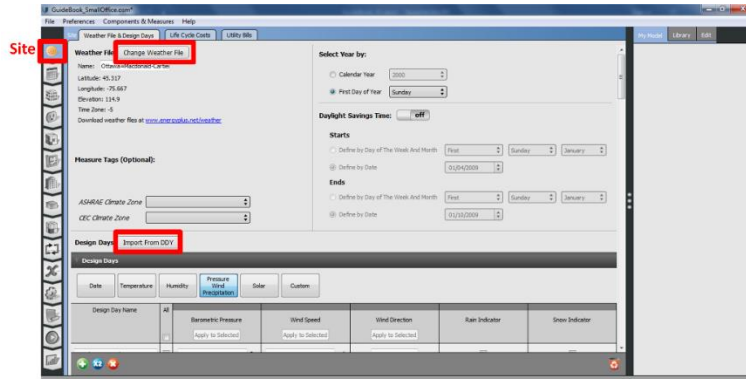


Figure 11. Define site and design days in OpenStudio.

Step 2. HVAC systems

In our building model, we want to have VAV boxes with reheat coil for the heating and cooling demands of each of the 16 perimeter offices and the core office. We assume that one AHU delivers supply air for all the VAV boxes. We also add hot-water baseboards to each thermal zone to deliver partial heating loads.

First, we turn off ideal air loads in the **Thermal Zones** tab if we have already set it as the HVAC system in SketchUp Make (Figure 12).

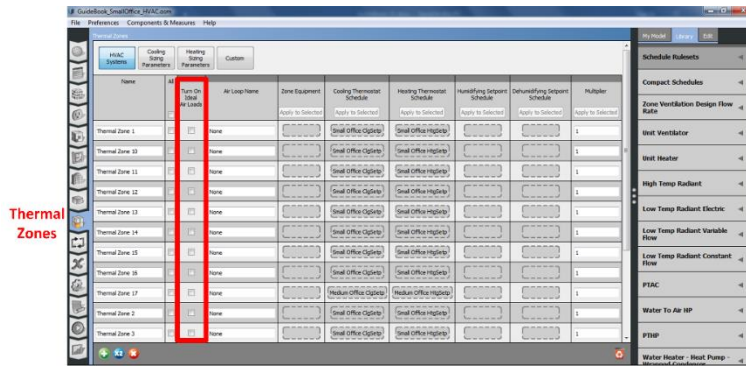


Figure 12. Turn off ideal air loads in SketchUp Make.

For the HVAC system, we use **HVAC Systems** tab. We click on the green plus sign on the top left corner of HVAC systems window. A window of a number of HVAC system templates is popped up (Figure 13).

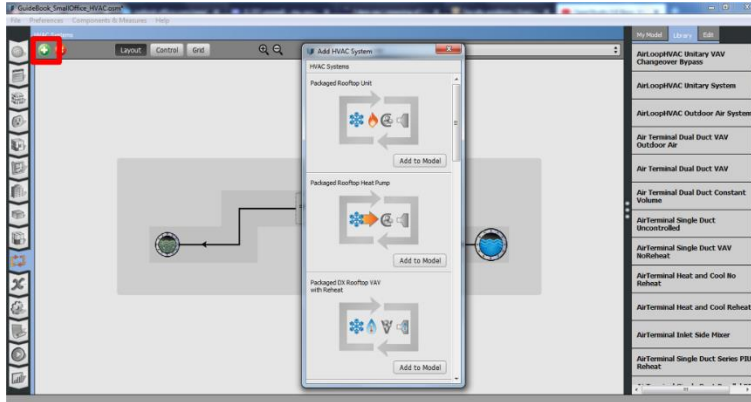


Figure 13. Adding HVAC systems in OpenStudio.

Now, we can choose from the HVAC system templates. For our building model, we add the “Packaged Rooftop VAV with Reheat” from the templates. This template will show the air loop, where we can see the outdoor air, heating and cooling coils, and a fan on the top part. On the bottom, it shows us terminal units in the thermal zones (Figure 14).

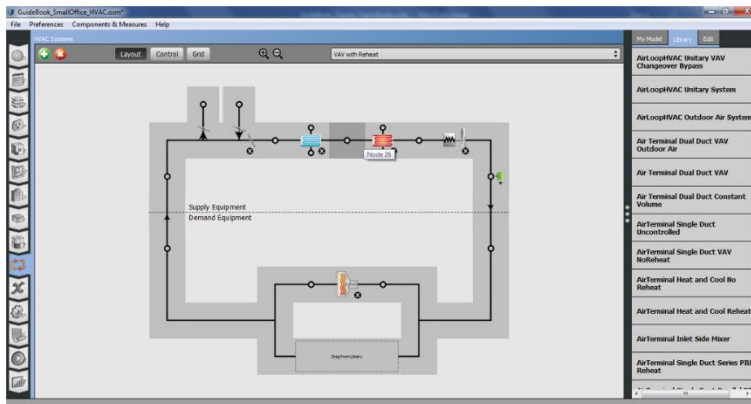


Figure 14. Air loop of a packaged rooftop VAV with reheat coils in OpenStudio.

From **My Model > Thermal Zone** on the right side of the OpenStudio window model, we drag and drop each thermal zone that we want to have a VAV box for it to the bottom part of the air loop, which is the demand side (Figure 15).

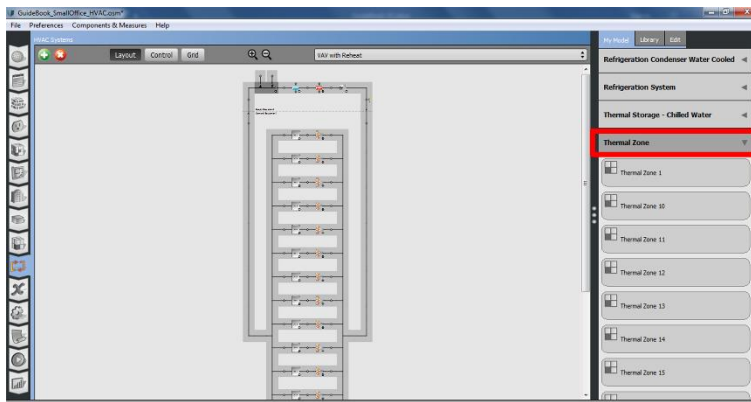


Figure 15. Assigning VAV with reheat coil for thermal zones in OpenStudio.

If we click on the outdoor air system in the air loop, heating or cooling coils, or fans, we can see the information about them in the **Edit** tab on the right side of the OpenStudio window. For example, we want to control outdoor air with an economizer. So, we click on **Air Loop HVAC Outdoor Air System 1** and set the economizer in the **Edit** tab (Figure 16).

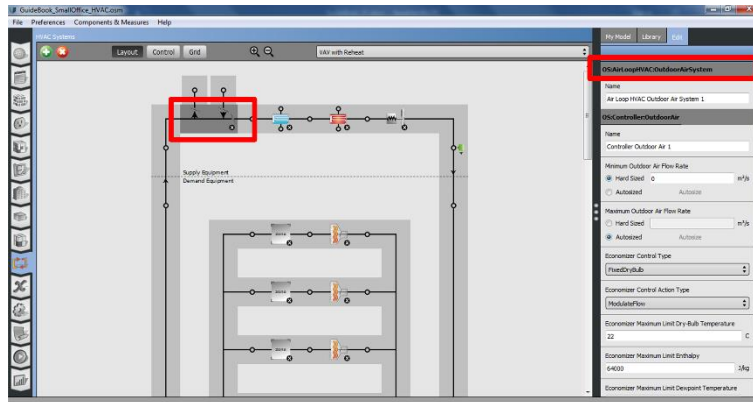


Figure 16. Set economizer for the HVAC outdoor air system in OpenStudio.

Now, we want to add hot-water baseboards to each thermal zone. So, we go back to **Thermal Zones** tab on the left side. From the **Library** on the right side, we drag and drop **Baseboard Convective Water** to each thermal zone (Figure 17).

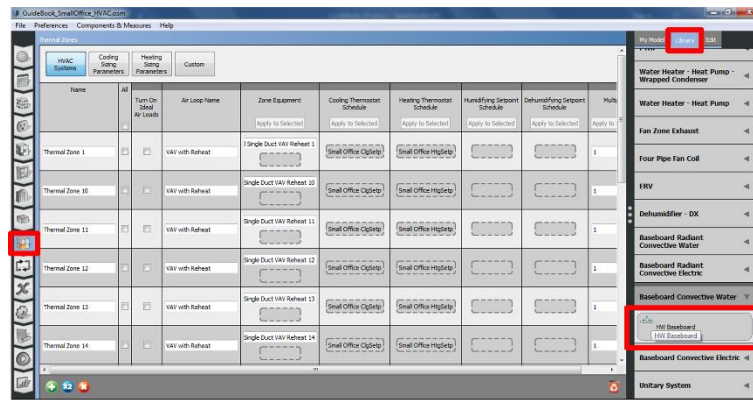


Figure 17. Set hot-water baseboards to thermal zones in OpenStudio.

Once we added the hot-water baseboard to each thermal zone, we click on the baseboard of each thermal zone in **Zone Equipment** and select **Edit > OS:Coil:Heating:Water:Baseboard**. Then, we choose **Hot Water Loop** (Figure 18).

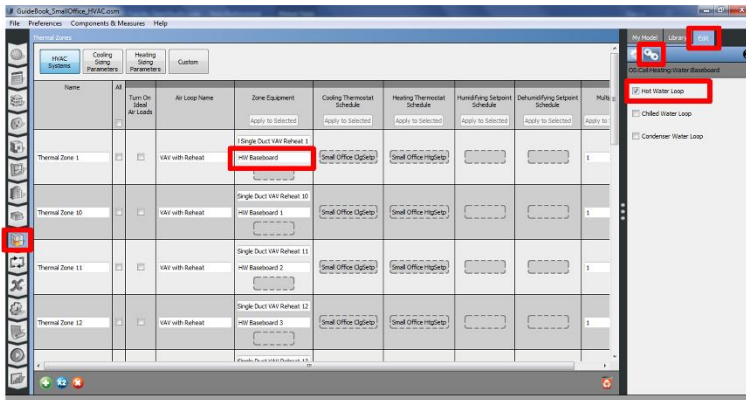


Figure 18. Set hot-water loop for hot-water baseboards of thermal zones in OpenStudio.

Now, if we go back to **HVAC Systems > Hot Water Loop**, we will see that all the hot-water baseboards have been added to the demand side of the hot water loop (Figure 19).

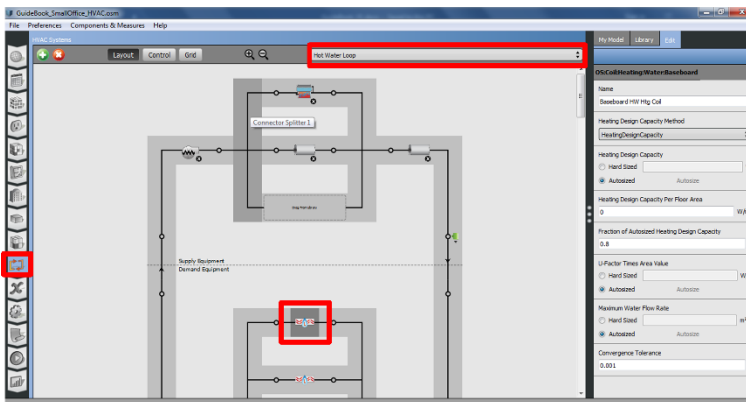


Figure 19. Check hot water loop of the HVAC system in OpenStudio.

Step 3. Simulation settings

Before we start test run in OpenStudio, we use **Simulation Settings** to provide other inputs for simulation runs. For example, we can define run period and simulation controls such as: if we want to do zone/system/plant sizing calculation and run simulation for sizing periods. In our example, we set the annual simulation run to do zone, system, and plant sizing calculation (Figure 20).

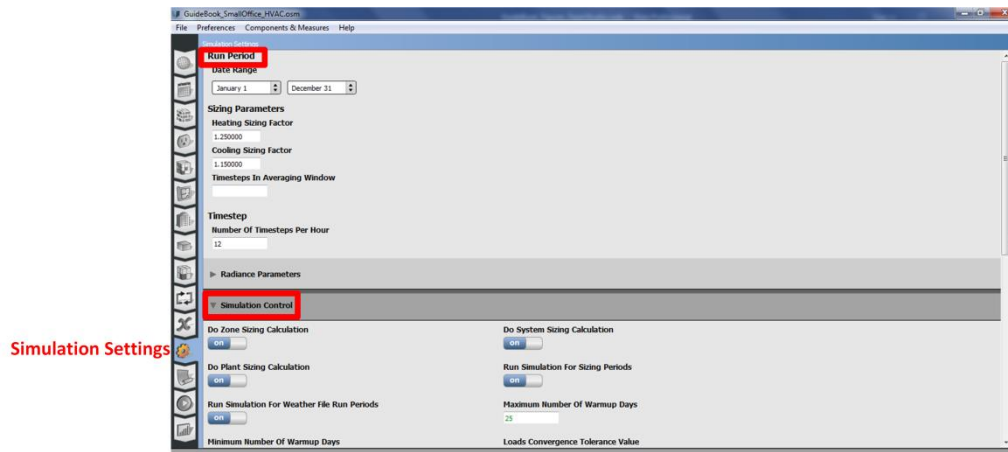


Figure 20. Set simulation inputs in OpenStudio.

Step 4. Test run in OpenStudio

Once we defined site, HVAC systems, and simulation settings in OpenStudio, we do a test run in OpenStudio using **Run Simulation** tab on the left side of the OpenStudio window (Figure 21).

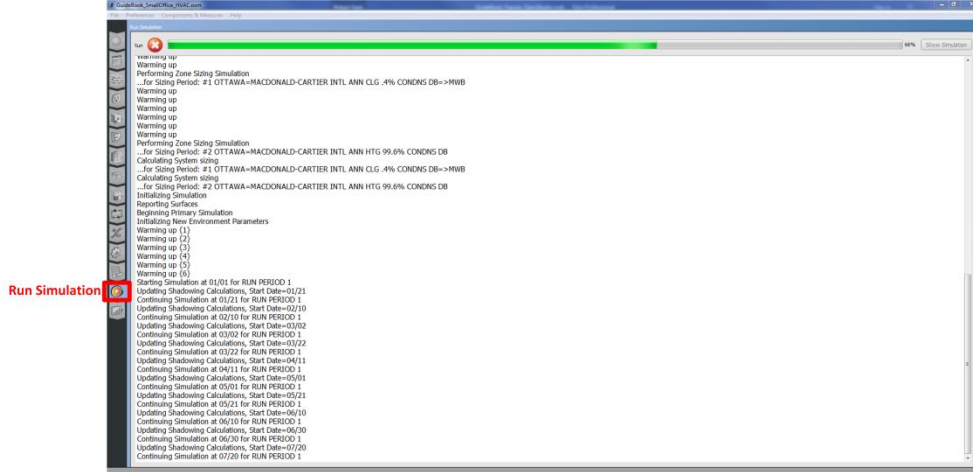


Figure 21. Run simulation in OpenStudio.

Step 5. Export model to EnergyPlus

We use **File > Export > IDF** to export our model from OpenStudio to EnergyPlus. Once we export the model as IDF file, we will add occupant models to the model by working on the IDF file in EnergyPlus (Figure 22).

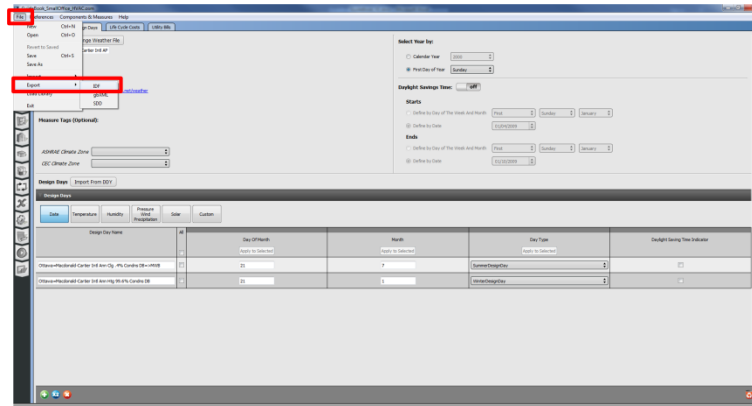


Figure 22. Export model from OpenStudio to EnergyPlus.

Adding occupant models in EnergyPlus

In this section, we learn how to add occupant models to the building model that we have already made. We will use the Energy Management System (EMS) application of EnergyPlus. As an example on how to use the EMS application, we will go through the procedure of writing a program for occupants' presence and lights use. For these two domains, we will implement two models from previous studies including: (1) Wang et al.'s (2005) occupancy model, and (2) Reinhart's (2004) light switch model. Since these models have been driven from data collected in private perimeter offices, we will use them for the 16 private perimeter offices. However, we will use the standard-based assumptions for the open-plan core office.

Step 1. EMS variables

For writing a program in the EMS application, we input the required independent variables into the program that we write to get the dependent variables from it. So, the first step is to define all variables that we need to have in our EMS programs. Types of variables that we use in our EMS programs are: [sensors](#), [actuators](#), [built-in variables](#), and [global variables](#).

In Table 2, we see a list of all the sensors, actuators, and built-in and global variables that we should have for the occupancy and lighting use models in the EMS application of EnergyPlus. We will see how to define sensors and actuators in EnergyPlus in the next steps.

Table 2. EMS variables which are used in the example EMS programs.

Program	Variables			
	Sensors	Actuator	Built-in variable	Global variable
Wang et al.'s (2005) occupancy model		Occ_Zn(1 to 16)	CurrentTime, DayOfWeek	Seeder, ZoneNo, A1, A2, A3, A4, A5, A6, A7, Occupancy, arr_event, dpt_event, Arrival, Coffee1, Lunch, Coffee2, Departure, Vac1, Vac2, handle, ArrTime_Zn(1 to 16), Coffee1_Zn(1 to 16), Lunch_Zn (1 to 16), Coffee2_Zn(1 to 16), DptTime_Zn(1 to 16), Vac1_Zn(1 to 16), Vac2_Zn(1 to 16), A1_(1 to 16), A2_(1 to 16), A3_(1 to 16), A4_(1 to 16), A5_(1 to 16), A6_(1 to 16), A7_(1 to 16), ArrEvent_Zn(1 to 16), DptEvent_Zn(1 to 16)
Reinhart's (2004) light switch model	EinZn(1 to 16),	Light_Zn(1 to 16)	CurrentTime, DayOfWeek	Ein, lighting, L1, L2, L3, L4, L5, L6, L1_(1 to 16), L2_(1 to 16), L3_(1 to 16), L4_(1 to 16), L5_(1 to 16), L6_(1 to 16)

Step 2. EMS sensor

EMS sensors are independent variables that we can find them in the field **EnergyManagementSystem:Sensor** under the group of **Energy Management System (EMS)** (Figure 23). In our case, sensors are the environmental variables or the global variables that we ourselves define them in EnergyPlus and call them in EMS programs.

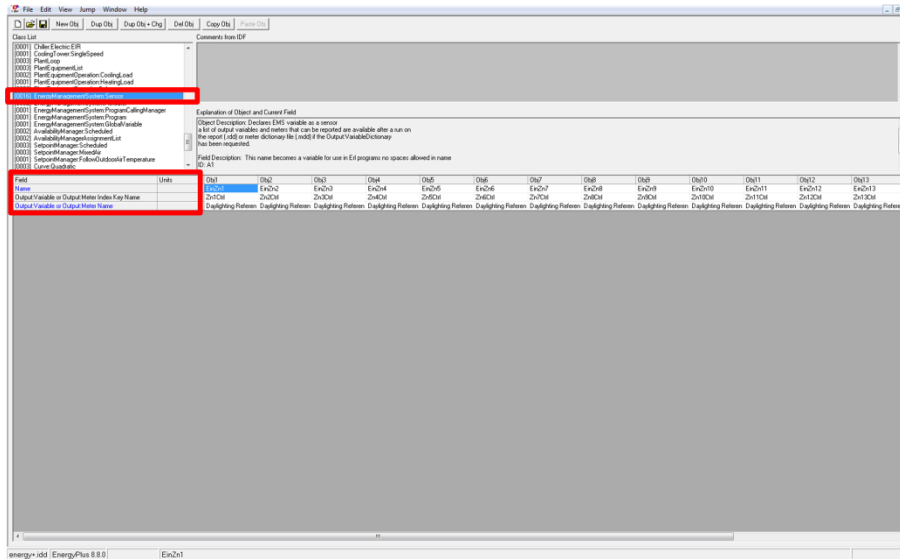


Figure 23. Set EMS sensors in EnergyPlus.

A list of the sensors that we have for the EMS program of our case study is presented in Table 3.

Table 3. EMS sensors.

Field	Occupant model	
	Wang et al.'s (2005) occupancy model	Reinhart's (2004) light switch model
Name	-	EinZn(1 to 16)
Output:Variable or Output:Meter Index Key Name	-	Zn(1 to 16)Ctrl
Output:Variable or Output:Meter Name	-	Schedule:Constant

There is a challenge in how we should get indoor daylighting illuminance of the perimeter offices at each timestep. To tackle this challenge, we use **Daylighting:Controls** under the group of **Daylighting** in EnergyPlus (Figure 24). So, we define a daylighting control for each perimeter office and we call them *Zn(1 to 16)Ctrl*. As we see in Table 3, we refer to these daylighting controls as the **Output:Variable or Output:Meter Index Key Name** for sensing indoor daylighting illuminance in each perimeter office.

For defining a daylighting control in each perimeter, we should also set a sensor in each perimeter office. So, we use **Daylighting:ReferencePoint** under the group of **Daylighting** (see Figure 24). We set a daylight sensor at the center of each perimeter office at the height of 0.8 (i.e. desktop height). We set the illuminance setpoint as a very large value (e.g. 100000 lx), so that the daylighting controls do not control lights in the perimeter offices as we want to use Reinhart's (2004) light switch model for simulating how occupants switch on/off the lights.

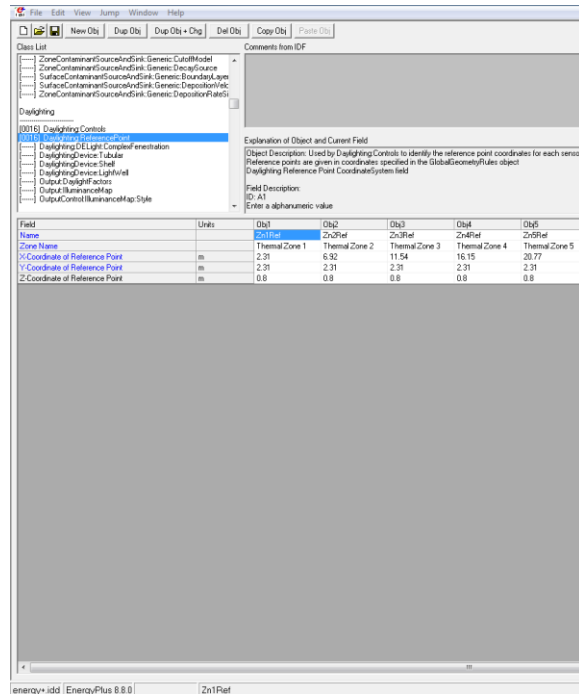
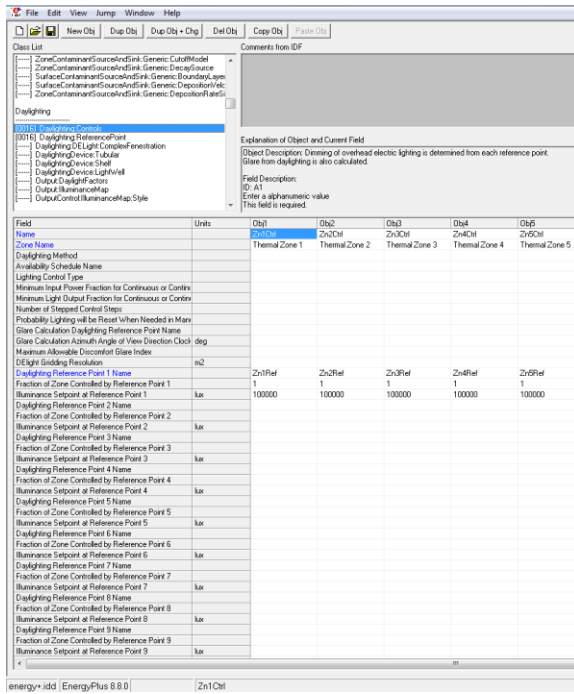


Figure 24. Set daylight control (left) and sensors (right) in EnergyPlus.

Step 3. EMS actuator

EMS actuators are occupants' presence or actions of occupants on building's systems (e.g. lights and thermostats) and components (e.g. window shades and operable windows). So, it is important to connect occupants' presence and states of building's systems and components to EMS actuators. For example, we set the fields **People** and **Lights** under the group of **Internal Gains** in EnergyPlus to adjust occupancy and light states using the schedules that we define for these two internal gain groups in **Schedule:Constant**.

We can find EMS actuators in the field: **EnergyManagementSystem:Actuator** under the group of **Energy Management System (EMS)** (Figure 25).

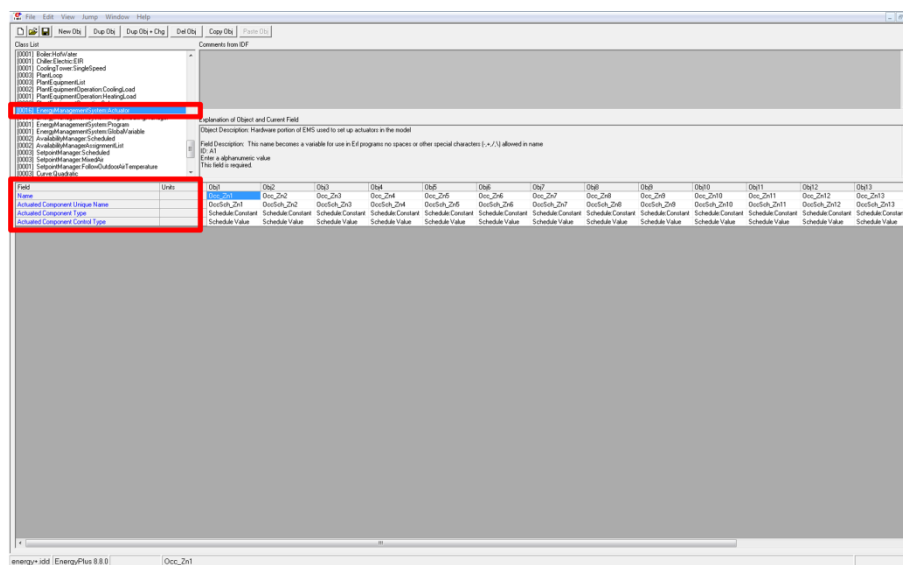


Figure 25. Set EMS actuator in EnergyPlus.

For the occupancy model, we use the variables, which we have already prepared a list of them (see Table 2), to predict whether an occupant is present or absent in the office at each timestep. So, we need to have an actuator for occupancy of each perimeter office to set it as 1 (i.e. occupant was present) or 0 (i.e. occupant was absent) at each timestep. Similarly, we use the EMS program that we write for light switch-on and off actions to predict whether an occupant turns on or off the lights at each timestep. So, we will define an actuator for the light states of each perimeter office to set it as 1 (i.e. lights were on) or 0 (i.e. lights were off) at each timestep using the EMS program for occupants' light switch actions.

There are four fields for defining EMS actuators. In Table 4, we see how we fill in the fields for the perimeter zones. Both occupancy and lights states have the type **Schedule:Constant** and their control type is **Schedule Value**. So, occupancy and light state schedules are the schedules that we define using the field **Schedule:Constant** under the group of **Schedules**. The field **Actuated Component Unique Name** is the name with which we call the schedules of occupancy and lights.

Table 4. EMS actuators.

Field	Occupant model	
	Wang et al.'s (2005) occupancy model	Reinhart's (2004) light switch model
Name	Occ_Zn(1 to 16)	Light_Zn(1 to 16)
Actuated Component Unique Name	OccSch_Zn(1 to 16)	LighSch_Zn(1 to 16)
Actuated Component Type	Schedule:Constant	Schedule:Constant
Actuated Component Control Type	Schedule Value	Schedule Value

Going back to the important note that the occupancy and lights state at each timestep should be actuated using the EMS program that we have written, we set the **Number of People Schedule Name** in the object **People** under the group of **Internal Gains** (Figure 26) to use the occupancy schedules that we called them *OccSch_Zn(1 to 16)*; which is the same as the schedule that we define for the schedules in the object **Schedule:Constant** under the group of **Schedules** (see Figure 26). The **Actuated Component Unique Name** in Table 4 is also the same as the schedule name (*OccSch_Zn(1 to 16)*). We follow the same procedure for lights as well.

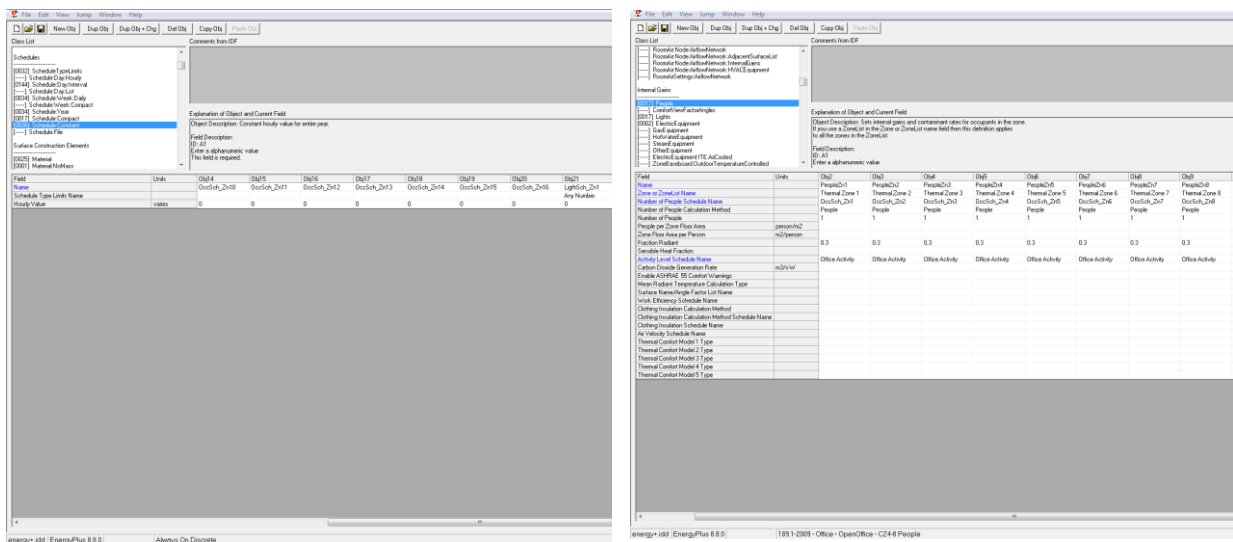


Figure 26. Set occupancy and light schedules in EnergyPlus.

Step 4. EMS built-in variable

In addition to EMS sensors and actuators, we can use built-in variables that EnergyPlus provides us. In the **EnergyPlus EMS Application Guide**, we can find a complete list of all built-in variables (Figure 27). We provided a list of the built-in variables in Table 2.

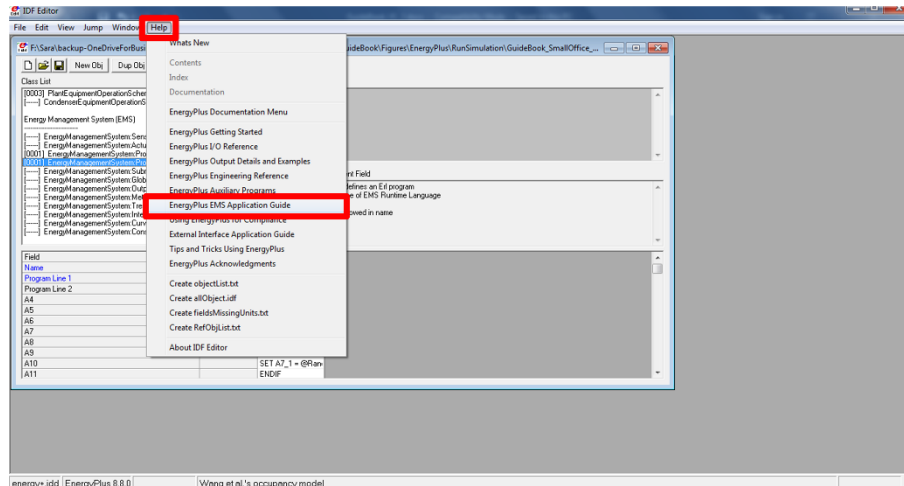


Figure 27. EnergyPlus EMS Application Guide.

Step 5. EMS global variable

If the variables that we use in EMS programs are not sensors, actuators, or built-in variables, we need to define them as global variables. We can define global variables in the object: **EnergyManagementSystem:GlobalVariable** under the group of **Energy Management System (EMS)** (Figure 28).

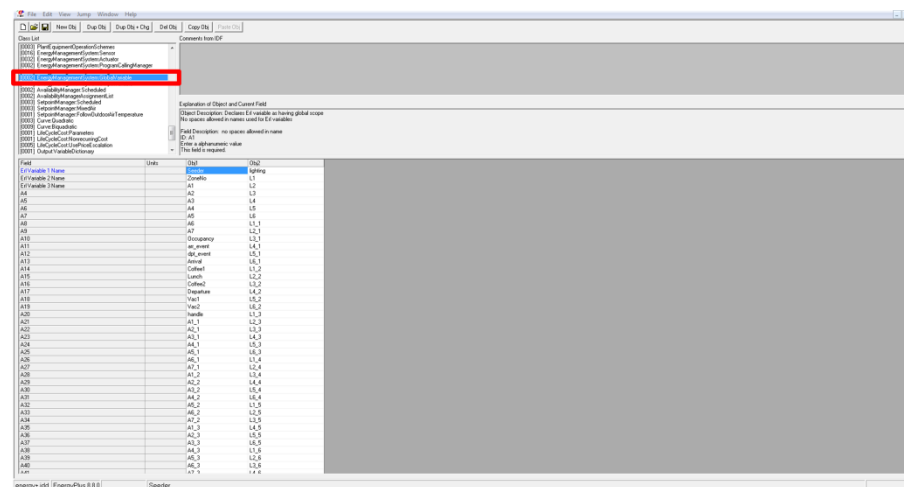


Figure 28. Set EMS global variables in EnergyPlus.

Step 6. EMS program calling manager

For running the EMS programs in EnergyPlus, it is important when we want each EMS program to be run. We know that each simulation run time represents one unique occupant. So, to mimic the habits of each individual occupant, we randomly choose the parameters of the occupant models at the beginning of each annual simulation runs based on the mean and standard deviation of the models' parameters. On the other hand, we want to simulate the randomness of the occupancy and actions for each occupant on each day or

each timestep. So, we write two main programs for occupancy and lighting use in our case study. One program is to generate the random parameters of the models just one time at the beginning of each annual simulation run; the other one is to run the program at the beginning of each timestep (Table 5). We see a list of the programs that we call through the object **EnergyManagementSystem:ProgramCallingManager** under the group of **Energy Management System (EMS)** (Figure 29) in Table 5.

Table 5. EMS program calling manager.

Field	Input	
Name	StartAnnual	EachTimestep
EnergyPlus Model Calling Point	BeginNewEnvironment	BeginTimestepBeforePredictor
Program Name 1	CreateNewOccupants_Occupancy	Wang_et_al_2005_OccupancyModel_Zn1To10
Program Name 2	CreateNewOccupants_LightingUse	Wang_et_al_2005_OccupancyModel_Zn11To16
Program Name 3	-	Reinhart_2004_LightingUseModel

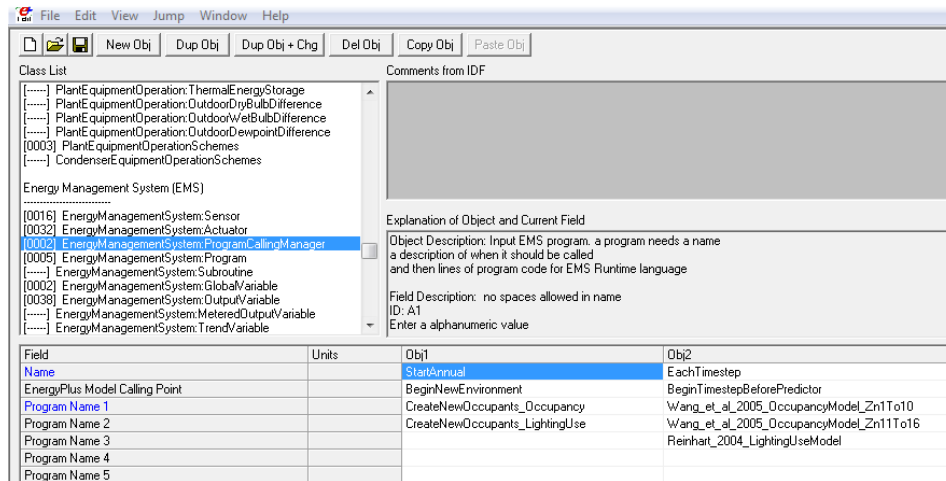


Figure 29. Set EMS program calling manager in EnergyPlus.

Step 7. EMS program

Let us implement Wang et al.'s (2005) occupancy model as an example using the EMS application of EnergyPlus. Arrival, departure, breaks times, and duration of breaks are chosen randomly for each occupant using this model at the beginning of an annual simulation run. We will keep them constant for the whole year for each occupant. In this way, we can mimic the habits of each occupant. On the other hand, the arrival, departure, breaks times, and duration of breaks that each occupant takes may change daily to some extent. So, we generate the arrival, departure, breaks times, and duration of breaks randomly at the beginning of each day for each occupant using the average times and breaks' durations of each occupant. For choosing random arrival, departure, and breaks times and durations of breaks, we need to know the mean and standard deviation of the time that these events happen, so that we can choose them randomly from normal distribution. We assume that the arrival time is $9:00 \pm 15$ minutes and the departure time is $17:00 \pm 15$ minutes. Also, we assume that lunch time is $12:00 \pm 15$ minutes and two coffee breaks (one before and one after lunch) are at $10:30$ and $15:00 \pm 15$ minutes. Each coffee break is for 15 ± 5 minutes

and lunch break is for one hour \pm 15 minutes. In our example, we assume that the two coffee breaks take equally. We choose the duration of the breaks randomly using the exponential distribution probability.

For generating the arrival, departure, breaks times randomly using the normal distribution function, we use the built-in function **@RandomNormal** in the EMS. We also use the built-in function **@SeedRandom** to generate the seed which we need for random number generation for using in **@RandomNormal**. Here, we use the built-in unique variable **ActualTime** for random seeding using **@SeedRandom**. For generating the duration of the breaks using the exponential distribution function, we use the built-in functions: **@RandomUniform** and **@Ln** (Table 6).

Table 6. EMS functions and statements which are used in the example EMS programs

Program	Built-in function	Statements
Wang et al.'s (2005) occupancy model	@SeedRandom, @RandomNormal, @RandomUniform, @Ln	IF, ELSEIF, ELSE, ENDIF, WHILE, ENDWHILE, SET
Reinhart's (2004) light switch model		

Step 8. EMS output variable

If we want to see how our EMS programs actuate the variables, we can use the object **EnergyManagementSystem:OutputVariable** under the group of **Energy Management System (EMS)** to define which variables we want to get their outputs. Once we define them through the EMS output variables, we add their objects using **Output:Variable** under the group of **Output Reporting** (Figure 30).

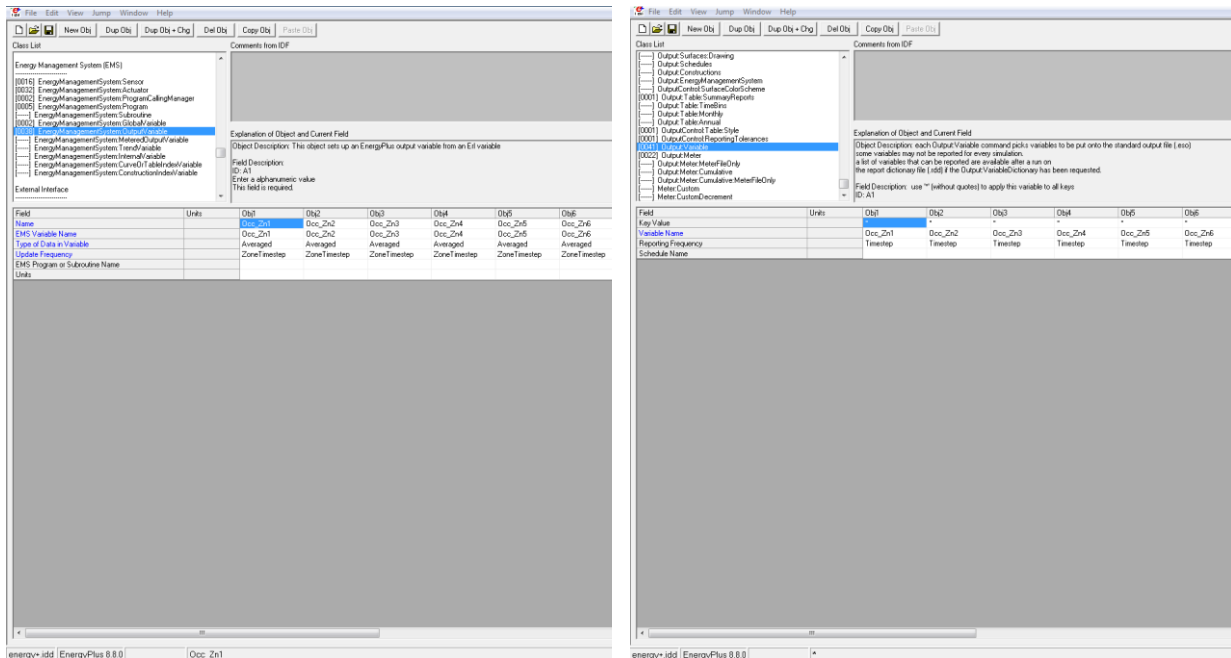


Figure 30. Set EMS output variable in EnergyPlus.

Postprocessing, visualizing, and interpreting the outputs

Here, we will see how we postprocess and visualize the simulation outputs; and we will do a parametric analysis to see how a building design affects occupants and in return, how building energy performance is affected by occupants.

Postprocessing and visualization

In our case study, we implemented two stochastic models. Since stochastic models give different results every time they are simulated, we ran our building model more than one time (50 annual simulation runs). So, final important steps in simulating occupant behavior are how we postprocess our simulation results; and how we can effectively communicate our stochastic results to clients. Here, we will see the most common ways of visualization of the stochastic results.

One of the common ways to visualize the stochastic data is a box plot (Figure 31). A box plot consists of a box and lines (aka whiskers), which extend out of the box. The median of the data is shown by the line which is inside the box. The 25 and 75% of the data are the bottom and top lines of the box. In other words, the box includes 50% of the data. The whiskers show the range of all data. There might be some points on a box plot which are outside the whiskers. These points are called outliers. With a box plot, we can illustrate the dispersion of the stochastic data.

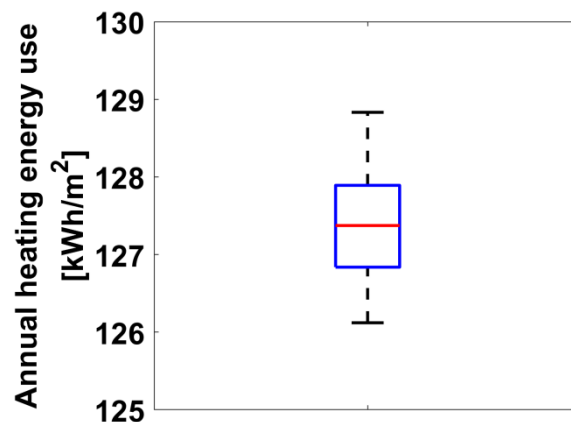


Figure 31. Using box plot for visualization of stochastic results.

The other graph type which we can use for visualization of the stochastic data, is a cumulative distribution function (Figure 32). For each value on the x-axis (i.e. x) of a cumulative distribution, the y-axis shows the probability that the values of X will be lower than or equal to x . For example, we can extract the heating load which is used by a specific percentage of occupants from a cumulative distribution.

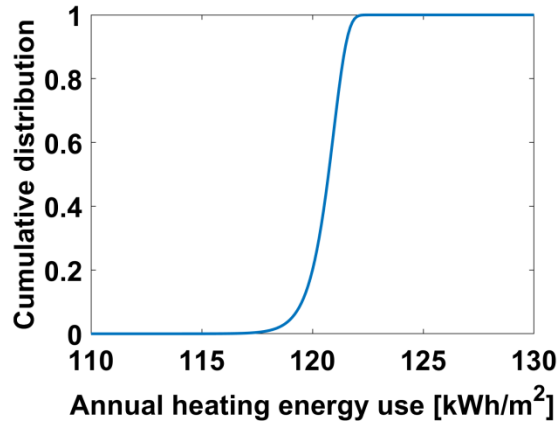


Figure 32. Using cumulative distribution function for visualization of stochastic results.

We can also present the distribution of the stochastic data using a histogram (Figure 33). The y-axis on a histogram shows the number of data that falls into each bin which are shown on the x-axis. A probability distribution is the other graph type that we can illustrate the data distribution (see Figure 33). A probability distribution is similar to a histogram, but the y-axis on a probability distribution shows the probability that each specific value of the x-axis can happen.

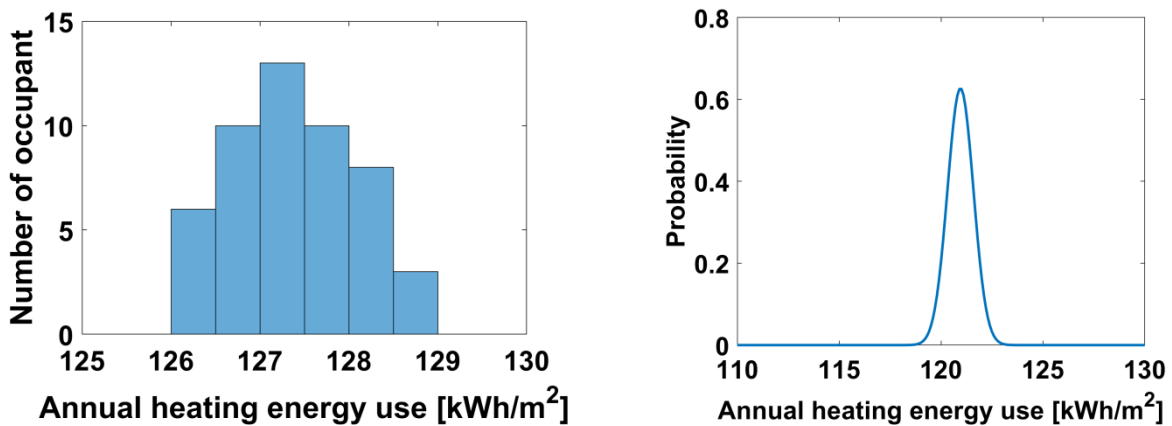


Figure 33. Using histogram (left) and probability distribution (right) for visualization of stochastic results.

Parametric analysis

As we discussed before, we can track the two-way interactions between a building and its occupants using dynamic occupant models. To see how a building and its occupants can affect each other, we will do a parametric analysis here. We analyze the impact of two glazing systems (Table 7) on the lighting energy use of our building model and how frequently occupants switched on/off the lights for the two glazing systems.

Table 7. Glazing systems design parameters

Type	U-factor (kWh/m ²)	SHGC	VT
1	1.82	0.36	0.64
2	1.42	0.48	0.69

We can see the distribution of the lighting energy use and the number of times that occupants switched on/off the lights in the 16 perimeter offices if we design two window types. The first window type has a lower solar heat gain coefficient (SHGC) and visible transmittance (VT) than the second window type. The box plot of the number of light switches (Figure 34) shows that if we use the second window type in our design building, occupants will not use lights in the perimeter offices as frequently as that if we use the first window type. In return, the lighting energy use of the perimeter offices will be lower if we use the second window type (see Figure 34).

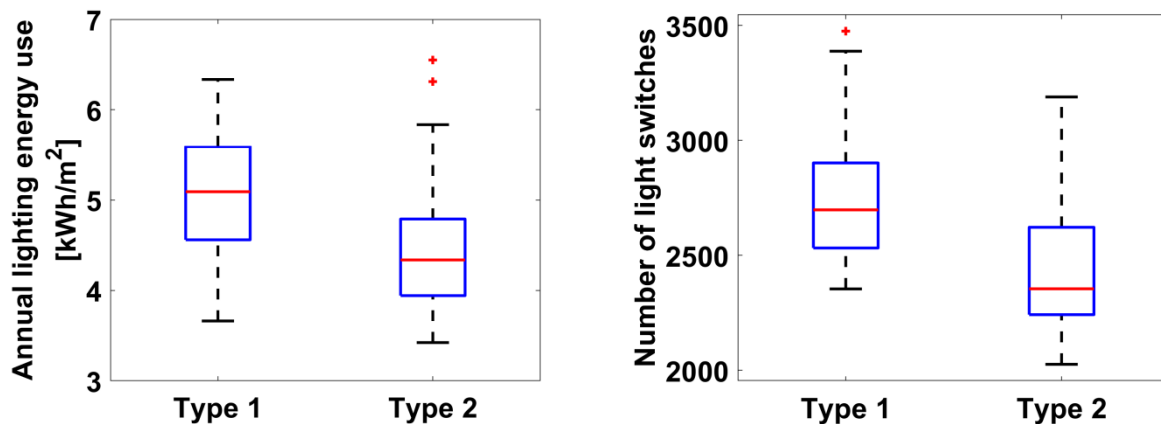


Figure 34. A parametric study on the impact of two window type on lighting energy use (left) and light switches (right).

References

- Abdelalim A, O'Brien W. 2018. An approach towards achieving net-zero energy buildings based on a stochastic tenant model. In: eSim 2018. Montreal, Canada.
- de Wilde P. 2014. The gap between predicted and measured energy performance of buildings: A framework for investigation. *Automation in Construction* 41:40-49.
- Gilani S, O'Brien W, Gunay HB. 2018. Simulating occupants' impact on building energy performance at different spatial scales. *Building and Environment* 132:327-337.
- Gilani S, O'Brien W, Gunay HB, Carrizo JS. 2016. Use of dynamic occupant behavior models in the building design and code compliance processes. *Energy and Buildings: Special Issue on Advances in BEM and Sim* 117:260-271.
- Gunay HB, O'Brien W, Beausoleil-Morrison I. 2013. A critical review of observation studies, modeling, and simulation of adaptive occupant behaviors in offices. *Building and Environment* 70:31-47.
- Gunay HB, O'Brien W, Beausoleil-Morrison I, Gilani S. 2016. Modelling plug-in equipment load patterns in private office spaces. *Energy and Buildings*.
- Haldi F, Robinson D. 2009. Interactions with window openings by office occupants. *Building and Environment* 44:2378-2395.
- Haldi F, Robinson D. 2011. The impact of occupants' behaviour on building energy demand. *Journal of Building Performance Simulation* 4:323-338.
- O'Brien W, Abdelalim A, Abuimara T, Beausoleil-Morrison I, Carrizo JS, Danks R, Gilani S, Gunay HB, Kesik T, Ouf M. 2018. Roadmap for occupant modelling in building codes and standards. In: eSim 2018. Montreal, Canada.
- O'Brien W, Gunay HB. 2014. The contextual factors contributing to occupants' adaptive comfort behaviors in offices—A review and proposed modeling framework. *Building and Environment* 77:77-87.
- O'Brien W, Abdelalim A, Gunay HB. 2018. Development of an office tenant electricity use model and its application for right-sizing HVAC equipment. *Journal of Building Performance Simulation*.
- O'Brien W, Gunay HB. 2015. Mitigating office performance uncertainty of occupant use of window blinds and lighting using robust design. *Building Simulation*:1-16.

- Parys W, Saelens D, Hens H. 2011. Coupling of dynamic building simulation with stochastic modelling of occupant behaviour in offices – a review-based integrated methodology. *Journal of Building Performance Simulation* 4:339-358.
- Reinhart CF. 2004. Lightswitch-2002: a model for manual and automated control of electric lighting and blinds. *Solar Energy* 77:15-28.
- Saldanha N, Beausoleil-Morrison I. 2012. Measured end-use electric load profiles for 12 Canadian houses at high temporal resolution. *Energy and Buildings* 49:519-530.
- Wang D, Federspiel CC, Rubinstein F. 2005. Modeling occupancy in single person offices. *Energy and Buildings* 37:121-126.