

Supervisors: ir. R.C.G.M. (Roel) Loonen, TU/e
prof. dr. A.M. (António) Leitão, IST
ir. S.B. (Samuel) de Vries, TU/e
prof. dr. ir. J.L.M. (Jan) Hensen, TU/e

PARAMETRIC APPROACHES FOR SIMULTANEOUS DESIGN AND PERFORMANCE PREDICTION OF ADAPTIVE FAÇADES

Building performance simulation can serve as support for decision-making in the preliminary design phase of an architectural project, aiding the selection process between different design alternatives. As traditional simulation tools are based on static design and adaptive façades encompass an intended movement of construction elements, there is a lack of supporting tools that can correctly evaluate the performance of such buildings in an early stage. The presented ongoing research focuses on the use of parametric models for analyzing energy consumption in buildings with adaptive façades, combining architectural design and performance prediction in a single task. Grasshopper's plugin Honeybee is used to perform an EnergyPlus simulation of a small office space, for which we test adaptive shading control systems that vary in terms of trigger variables and opacity states. Honeybee's simulation results are generally within a range of 1-21% discrepancy from the same simulation ran through EnergyPlus. Shades with an intermediate opacity level are observed at a higher performance level than simplified on/off options, as they allow for continuous solar absorption while avoiding excess heat to the office's interior.

1. Introduction

Reducing the environmental impact of future buildings, as societies progress towards the application of sustainable practices, comprises the prediction of future energy costs and the compliance with regulations and indoor comfort levels. In the geometric study of complex projects, generative design approaches allow for the exploration of building designs that would otherwise not have been considered (Terzidis, 2006). As the concept of adaptive façades rises into consideration, it is important to find a reliable way to predict the combined benefits and constraints of specific propositions, which may vary in terms of materials, components and systems.

Pre-construction performance-based approaches allow the architect to continually assess geometry options while comparing them to the original design intent. However, assessing a building's performance is still not widely done at an early design stage. Moreover, modeling a dynamic façade is different from modeling a static element, as the geometry patterns are constantly shifting in the design outcome (Moloney, 2011). Therefore, the principal aim of this research is to put in practice the use of parametric models for the analysis of energy consumption in buildings with adaptive façades, combining architectural design and multidisciplinary optimization processes.

1.1. Tools and Workflows

Performance-driven design should follow a procedure that focuses not only on the aesthetics and functionality, but also on a wider understanding of the environmental context of the project. Choosing an analysis tool implies considering the flexibility given to the various BPS software available, regarding model resolution, solution algorithms and user-friendliness (Loonen, Favoino, Hensen & Overend, 2017). The methodology developed by Sharaidin, Burry, and Salim (2012) analyzes this problematic inside a wide research context, using Grasshopper to assist the decision-making process at the preliminary design phase. In parallel, Shi and Yang (2013) explore the integration of performance simulation programs – namely Ecotect, Radiance, and EnergyPlus – into the Rhinoceros/Grasshopper context, which denotes the need for the addition of program development and code writing into the proposed workflow. Recently, both Karssies (2017) and Strunge (2017) make use of a set of BESTest cases to assess the validity of basic functionalities in the interface of Grasshopper’s plugin Honeybee – although showing a wider parameter flexibility for early-stage design¹, overall deviations are denoted in the produced output when compared to the baseline results.

While not all architects are familiar with custom-written computer code, there is a growing potential regarding the use of algorithmic approaches throughout the design process. Focusing on the performance of the building from an early project phase emphasizes a comprehensive optimization of various quantifiable performance outcomes. Using the same script to produce both parametric and analytical models would allow them to be immediately usable by the corresponding analysis tool, reducing the amount of additional work from the project team (Leitão, Castelo Branco & Cardoso, 2017). Earlier research projects started to combine the totality of the project-making procedure into an integrated algorithmic approach, developing a methodology that combines the description of not only the CAD/BIM model, but also of the required analytical inputs into a single script (Castelo Branco, & Leitão, 2017).

1.2. Rosetta

The presented work is part of a collaboration with the Lisbon-based research group ADA², composing a small portion of a generative design and analysis workflow which can be consulted in Annex I.

Performance analysis tools often require a simplified version of the original CAD/BIM model to simulate. The programming environment Rosetta (Lopes & Leitão, 2011) supports several programming languages for the connection of a series of CAD/BIM and analysis tools. Through an abstraction layer, the common operations for geometric modeling are translated into the corresponding operations of each specific tool, commonly known as back-ends, allowing the generation of equivalent models in each tool. Currently, the Rosetta-supported analysis tools include Radiance and DAYSIM, which analyze lighting performance. A connection that allows for direct energy simulations is still to be put in practice – therefore, the development of a back-end for EnergyPlus is seen as a next step, as it will allow for the execution of thermal simulations.

As each EnergyPlus front-end has its own default assumptions and design approach, their use usually requires developed expertise in the field of energy analysis to assure the quality of the simulation. Notwithstanding the need of a BPS background knowledge, linking Rosetta to EnergyPlus through a back-end would allow for the whole design process – from the initial concept to parametric optimization – to be fused into a single script, thus eliminating possible data losses that occur when multiple design and analysis tools are used. With that in mind, it would become possible to run several simulations in shorter timespans without the need to visually reproduce the tested models and, by the end of the process, generate an optimized solution in the chosen visualization tool (Rocha, 2017).

¹ When compared to the DIVA/Archsim combination.

² ADA = Algorithmic Design for Architecture. Publications can be consulted at <http://algorithmicdesign.github.io/publications.html>.

To create a functional back-end between Rosetta and EnergyPlus, it is imperative to acquire sufficient knowledge on the requirements and data processing methods of this tool. While a direct connection is not achieved, we are using Honeybee as an access point to EnergyPlus. To validate this analysis method, we are comparing the results of the Honeybee analysis of a case study to its original simulation results, obtained solely through EnergyPlus. This process will, additionally, serve to identify which functionalities are lacking in Honeybee and consider how they can be fixed through Rosetta, so that this plugin can ultimately be removed from the back-end's workflow.

1.3. Report Outline

In Section 2, the Honeybee modeling and performance assessment of a case study are described. Section 3 focuses on the implementation of an adaptive shading control system for the same model, through the execution of an Energy Management System (EMS). Methodology and results are presented and discussed in both sections. Finally, Section 4 concludes this report by presenting a set of general conclusions and the future outlook for this subject.

2. Energy Analysis

The objectives stated above cannot be achieved without sufficient understanding of the requirements of EnergyPlus. Thus, the first stage of this project is a preliminary study on the functionality of this tool and its simulation methods. The analyzed case study is a reference office space from subtask C³ of IEA SHC⁴ Task 56, which hosts 3 people (9m²/person). The south-facing exterior façade is equipped with solar active technologies, while the remaining five surfaces have adiabatic boundary conditions.

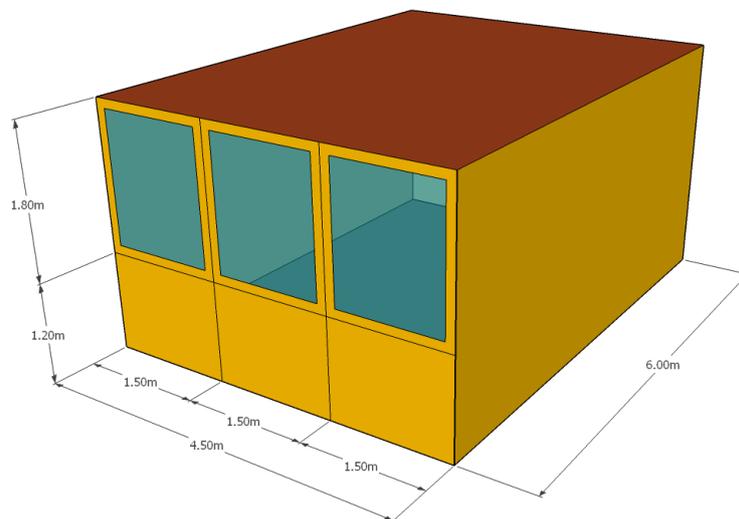


Figure 1. View of the reference office zone.

Internal sensible gains from human presence are set at 70 W/person, with an appliance use of 7 W/m² and a LED lighting system with a specific electrical gain of 10.9 W/m². A contemporaneity index of 0.8 is

³ The aim of this subtask is to analyze the energy contribution of different solar active façade technologies in covering building loads, as well as to evaluate indoor comfort conditions. More information on Task 56 can be found at <http://task56.iea-shc.org/description>.

⁴ International Energy Agency (IEA), Solar Heating and Cooling (SHC) Programme.

assumed to be constant throughout the year (SIA 2024, 2015). During weekends, appliance use is fixed in a standby consumption of 10%, while occupancy and lighting gains are not considered.

Due to infiltration through the façade, a natural air change rate of 0.15 1/h is assumed. A heat recovery unit is further included, with a sensible efficiency of 70% (SIA 2024, 2015). The solar shading consists of generic external shades, which block up to 70% of the solar radiation impinging the façade glass and are activated when the sum of beam and diffuse solar radiation on the external façade exceeds a threshold of 120 W/m². No influence of the urban or environmental context in terms of unwanted shading on the façade is considered. Out of the three reference locations analyzed in Task 56, Stockholm is the chosen one to be applied in this case study.

2.1. Methodology

The geometry for the Honeybee-EnergyPlus (HB-EP) analytic model was produced in Rhinoceros, and further imported to Grasshopper to be defined as a *Honeybee Zone*⁵. Information regarding assumptions and general zone description is introduced as shown in figure 2.

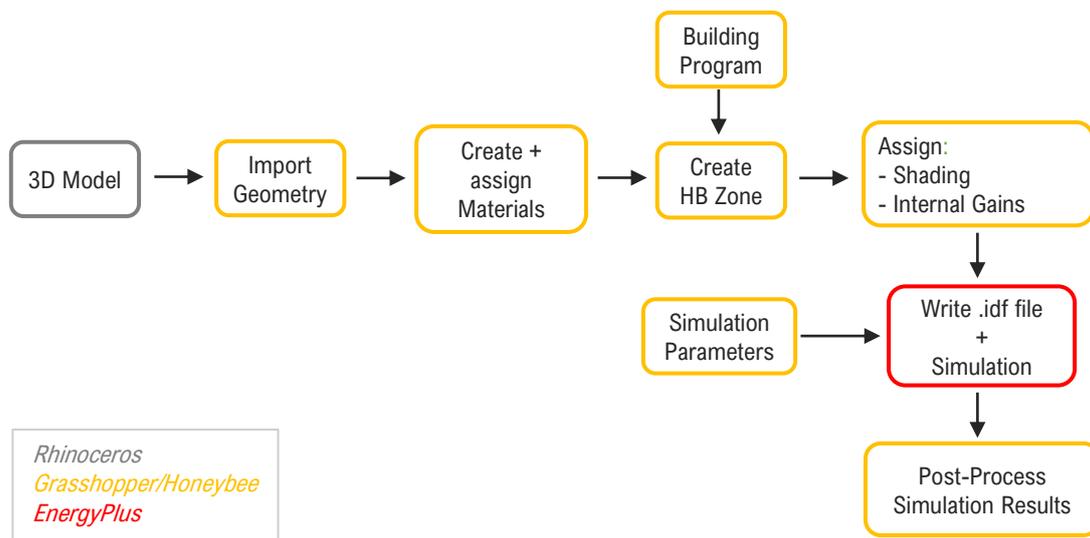


Figure 2. Honeybee-EnergyPlus analysis process.

Simulation parameters include the requested outputs and specific calculation methods. After writing the new .idf file, the simulation is performed, gathering the results in a temporary file address that is overwritten each time Honeybee completes a simulation. Finally, for post-processing purposes, the data is normalized by floor area and summed into the requested demands, gains and losses.

The indoor environment is conditioned by an Ideal Air Loads system, calculating heating (Q_{heat}) and cooling (Q_{cool}) demands within a setpoint temperature of 21°C and 25°C, respectively. The simulation runs for an entire year in an hourly timestep. A sizing calculation is performed for the office zone, considering design days and sizing periods from the chosen weather file. HVAC system and plant sizing calculations are disregarded. Shadow assessment computes the incoming solar energy at every single timestep of the simulation.

⁵ Honeybee zones gather all of the properties necessary for an energy simulation assigned to them.

2.2. Results and discussion

To validate the Honeybee interface capabilities regarding the production of an accurate energetic assessment, the results of the HB-EP simulation are compared to the original reference results of Task 56, simulated solely with EnergyPlus. For the purpose of this report, we are assuming an acceptable range of 10-20%⁶ of variation between results. Results are laid out in a specific energy balance format, as shown in Figure 3. Solar gains (Q_{sol}), internal gains (Q_{gint}) and heating demand (Q_{heat}) are denoted with a positive sign, whereas ventilation losses (Q_{vent}), infiltration losses (Q_{inf}), transmission (Q_{trans}) losses and cooling (Q_{cool}) demand are negative terms.

Heating demands amount to 16.77 kWh/(m²y) and 18.03 kWh/(m²y), respectively for EnergyPlus and HB-EP. Cooling demands for the two simulations are equal to 32.69 kWh/(m²y) and 26.83 kWh/(m²y). Although within an acceptable range, a possible cause for the divergence between both results might reside in the implementation of the building description, which can vary according to the user's own interpretation of data and simplification choices – in this case, we chose to disregard the no-mass aluminum window frame in the Honeybee model, given that it would have to substitute the original exterior layer material of the exterior wall. Other known issues are related to bugs from the plugin itself, as we are dealing with an open-source program that is in constant development and does not stand in an extent where sensible/latent and radiant/convective distinctions are fully implemented for all the energy variables.

It is important to emphasize how minor changes in the Honeybee script can result in significant discrepancies between simulation outcomes. The major discrepancy between both simulation outputs is found in the solar gains, which decreased circa 10 kWh/(m²y) (20.9%) in the HB-EP simulation. This may be justified by the impossibility of inserting some variables for the glass material in Honeybee – namely back side solar and visible reflectance at normal incidence, U-value⁷ and g-value.

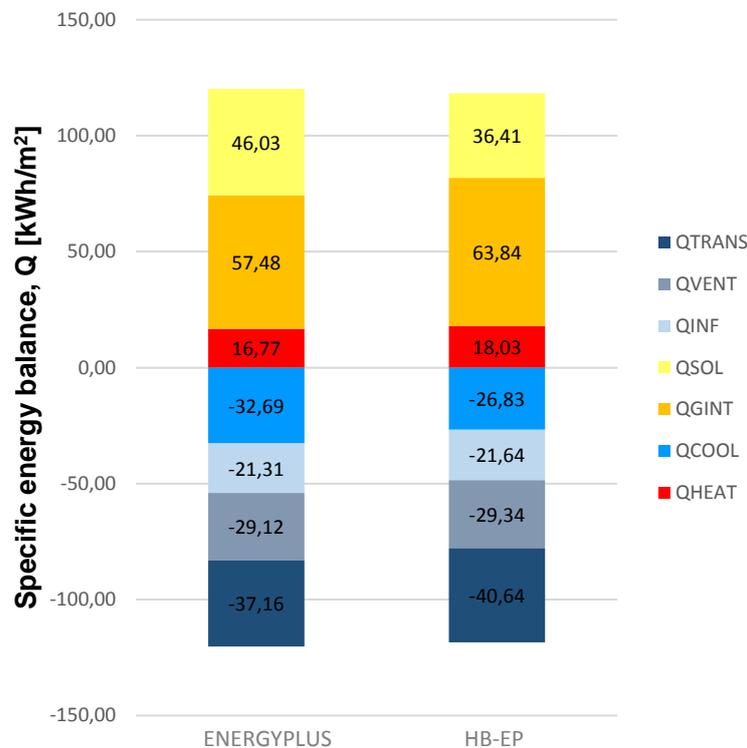


Figure 3. Specific energy balance of the reference office space for Stockholm, simulated with EnergyPlus and Honeybee.

⁶ This range is partly given to differences between simulation tools, as they have their own initial distinct settings and calculation methods that cannot be controlled.

⁷ Although the U-value can be inserted as a glazing property, it is in a separated component from the one that contains the set of customizable inputs such as thickness, reflectance and emissivity. The user cannot apply the two components simultaneously to the model.

The specific electrical demand of the zone for appliances and lights is shown in Figure 4. Appliance use is responsible for 14.50 kWh/(m²y) and 13.72 kWh/(m²y), respectively for EnergyPlus and HB-EP. Light consumption is close to equal in both simulations.

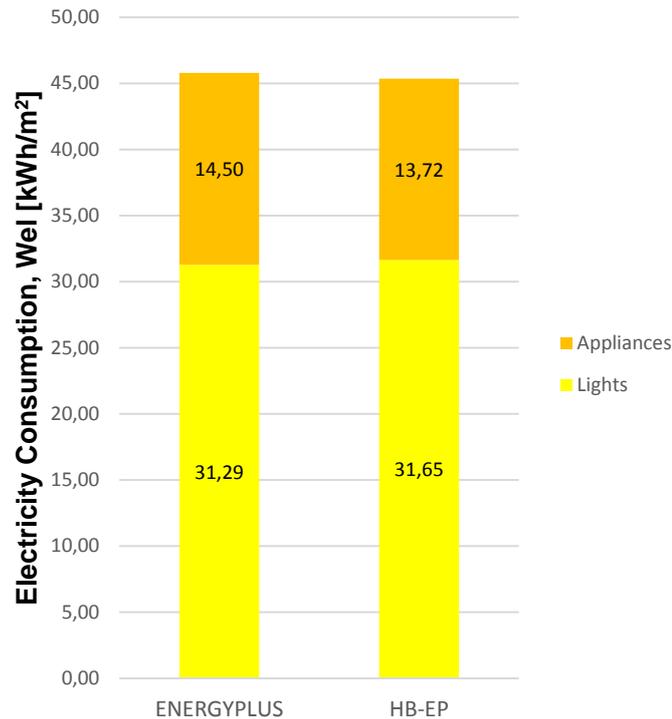


Figure 4. Electricity consumption for appliances and lights.

3. Adaptive Concept

This section is focused on the addition of two responsive shading systems to the office space described in Section 2, which vary in terms of trigger variables and opacity states. The relevance of this study is associated with indoor comfort, as daylighting affects positively the productivity, well-being and health of occupants (Galasiu & Veitch, 2006).

System 1 is inserted by means of an Energy Management System (EMS), a feature of EnergyPlus that provides a way to develop custom control and modeling routines for EnergyPlus models. For this model, the shading system has two states - fully closed and fully open – and it is triggered when the beam direct solar radiation incident on the external façade (vertical plane) exceeds a threshold of 120 W/m².

System 2 is controlled through Grasshopper’s logical operators, with three levels of shade opacity: fully closed (100%), half-open (50%) and fully open (0%). The control scheme is composed by three conditions:

- Close shade whenever the global horizontal radiation exceeds a threshold of 120 Wh/m²;
- Shade is half-open state when the global horizontal radiation exceeds 80 Wh/m² and is below 120 Wh/m²;
- Leave the shade fully open whenever the first two conditions have not triggered.

3.1. Methodology

3.1.1. Shading A_EMS

An essential part of the EMS module is the EnergyPlus Runtime Language (ERL), a programming language used to specify the EMS control algorithms. Controlled in a base of SET and IF/ELSE instructions, an EMS

is able to access a wide variety of state variables and utilize it to direct several types of control actions. The purpose is to emulate, inside EnergyPlus, the types of controls possible with the digital energy management systems used in real buildings, without the need to use complex Grasshopper definitions. Its implementation allows to run scripts during simulation time, as well as to override and extend EnergyPlus' capabilities.

For this case study, we are using an EMS script to insert exterior shades that use the beam direct solar radiation, rather than the global solar radiation, as input for the shading sensor - which EnergyPlus does not allow by default -, thus overriding the shading control inserted by Honeybee into the simulation file (.idf) . The EMS script is added to the Task 56 HB-EP model, inserted into the simulator as an additional string⁸ to be appended into the resultant IDF and, posteriorly, perform a new simulation. Below are described the objects that compose the EMS script⁹:

Sensor

Sensors map any EnergyPlus report variables or meters to an EMS variable. In this model, we are defining a variable called 'BeamRadiation', which makes use of the *Surface Outside Face Incident Beam Solar Radiation Rate per Area* output from EnergyPlus as input for the activation of the shading system.

Actuator

This type of object maps control variables to an EMS variable, allowing control signals to be sent to an EnergyPlus object. For this methodology, the object actuated upon is the window surface, defined in the Honeybee model. This way, the original shading control can be overridden by the new shading system we are implementing.

Global Variables

To be accessed by all program objects, the used variables need to be declared as global EMS variables. For this model, 'Shade_Status_On' and 'Shade_Status_Off' are defined.

Program

This set of objects contain the instruction blocks of the ERL code. Two programs are defined in this script: the first associates numerical values to the global variables; the second sets the conditions for the shading system to be activated by means of an IF/ELSE statement. For this case, there are two states for the actuator to take: 'Shade_Status_Off' corresponds to a value of 0.0 and 'Shade_Status_On' corresponds to a value of 3.0. All program objects are automatically run by EnergyPlus at every system time step.

3.1.2. Shading B_Grasshopper

In this approach, the original shading system from Task 56 is removed from the model and replaced by three external shading planes (one for each window), which are modeled in Rhinoceros and imported as Honeybee context surfaces. The transmittance schedule for these shades is manually created by the conditional logic that sets the three possible levels of opacity: the IF/THEN/ELSE statement of the script is described by a set of gates and mathematical operators in Grasshopper, used to perform logical proofs between several comparison tests.

For this shading system, the analyzed variable is the global horizontal radiation (GHR), imported from the weather file as a year-long list of hourly values. This list is evaluated under the condition $GHR \geq 120$ and

⁸ EMS implementation is performed by pasting the EMS script into a panel component in Grasshopper.

⁹ The used EMS script can be found at Annex II.

the list elements that return a 'True' boolean are replaced by the value 0.0, which is equivalent to fully closed shades. The resultant list is then evaluated for the expression $80 \leq GHR < 120$, replacing the corresponding elements by 0.5 for the half-open shading state. Finally, the remainder list elements are replaced by the value 1.0 for fully open shades. When all the elements in the original list are replaced, they are used as input values for the creation of a CSV¹⁰ schedule file. This transmittance schedule is, then, connected to the Honeybee Context component and the simulation is performed.

3.2. Results and discussion



Figure 5. Energy balance comparison between shading systems.

Figure 5 presents the specific energy balance for both the original Honeybee Task56 model and the two tested variations for the shading systems. The outcomes are outlined as follows:

- Shading A shows the worst overall performance, denoting an increase in every simulated variable where the solar gains (26.9%) and cooling demand (21.5%) are the most discrepant values. The reason for the increased energy consumption lies in the variable that activates the shades, as the Task56 model uses a combination of beam and diffuse solar radiation while Shading A only uses the beam solar radiation;
- Ventilation and transmission losses increase in both models, with a range of 1-8% in Shading A and 4-6% in Shading B; the increase on solar gains results in a larger amount of heat loss in both systems;
- The introduction of a half-open status in Shading B reduces the overall energy consumption, which is notoriously visible in the heating (-22.6%) and cooling demands (-17.9%). A 50% opacity level allows for solar absorption and presence of daylight, while avoiding excess heat conduction to the interior of the office.

¹⁰ CSV = Comma Separated Values.

4. Conclusions and future perspectives

The present report aims for the finding of methodologies that combine, in an early design stage, the modeling of dynamic façades with the assessment of the building's energetic performance. The contents of the previous sections belong to a wider workflow that focuses on the back-end connection between the programming environment Rosetta and EnergyPlus (see Annex I). Regarding the case study of Task56, Grasshopper's plugin Honeybee shows an acceptable range of discrepancy from the original EnergyPlus simulation results, with the exception of the solar heat gains which surpass the 20% range. Possible causes for these results are the excessive default assumptions performed by the plugin, the impossibility of introducing a portion of the model's inputs and the presence of bugs in the tool's program, given its open-source condition.

From the simulation of both shading control systems, it is possible to observe that the insertion of an intermediate opacity stage results in an overall smaller energy balance, with noticeably lower demands for heating and cooling. The chosen type of variable to trigger the shades is equally relevant – using the global horizontal solar radiation values as input for the shade activation shows a lower energy use when compared to the use of direct beam radiation values. It is also important to underline that the implementation of an EMS script is an overall more complex process than the use of logical operators through Grasshopper, and the lack of user experience might result in untrustworthy or incorrect results.

4.1. Future outlook

The process of connecting Rosetta and EnergyPlus in the same workflow is currently in development by ADA, the next step being the assessment of a case study for adaptive façade optimization. However, there is a need for further assessment on the efficacy of the implementation of Energy Management Systems into Honeybee, as other attempts that encompass different actuators for this methodology return incorrect results. It is expected that Honeybee components keep being updated recurrently, to solve the absence of essential EnergyPlus inputs.

References

- Castelo Branco, R. & Leitão, A. (2017). Integrated Algorithmic Design. In *Design Tools: proceedings of the 35th eCAADe conference*, 1, pp. 729–738.
- Galasiu, A., Veitch, J. (2006). Occupant preferences and satisfaction with the luminous environment and control systems in daylight offices: a literature review. *Energy and Buildings*, 38 (7), 728-742.
- Karssies, W. (2017). Optimization workflow regarding daylighting, energy and glare, for performance assessment of new generation semi-transparent photovoltaic façades. MSc Thesis, Technische Universiteit Eindhoven, Eindhoven.
- Leitão, A., Castelo Branco, R. & Cardoso, C. (2017). Algorithmic-based Analysis: Design and Analysis in a Multi Back-end Generative Tool. In *Protocols, Flows, and Glitches: Proceedings of the 22nd Annual Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA)*, Suzhou, pp. 137–147.
- Loonen, R., Favoino, F., Hensen, J. & Overend, M. (2017). Review of current status, requirements and opportunities for building performance simulation of adaptive facades. *Journal of Building Performance Simulation*, 10:2, pp. 205-223.
- Lopes, J. & Leitão, A. (2011). Portable Generative Design for CAD Applications. In *Integration Through Computation: Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, Banff, pp. 196–203.
- Moloney, J. (2011). *Designing Kinetics for Architectural Facades: State Change*. Adingdon: Routledge.

Rocha, C. (2017). Generative Design for Energy Efficiency. MSc Thesis, Instituto Superior Técnico, Lisboa.

Sharaidin, K., Burry, J. & Salim, F. (2012). Integration of Digital Simulation Tools with Parametric Designs to Evaluate Kinetic Façades for Daylight Performance. In *Physical Digitality: proceedings of the 30th eCAADe conference*, 2, pp. 701–709.

Shi, X. & Yang, W. (2013). Performance-driven architectural design and optimization technique from a perspective of architects. *Automation in Construction*, 32, pp. 125-135.

Strunge, J. (2017). Building Performance Simulation in Architectural Design. *Advanced Building Skins Conference*, Switzerland.

Terzidis, K. (2006). *Algorithmic Architecture*. Architectural Press, Oxon and New York.

ANNEX I – ADA Workflow

Figure 5 presents the complete workflow currently being developed by ADA. The blue highlighted set of tasks represents the content of this report. The main code of the building, applied through Rosetta, is explained below. Building geometry information is linked to the HB-EP analysis by means of a Geometry Pipeline¹¹ in Grasshopper.

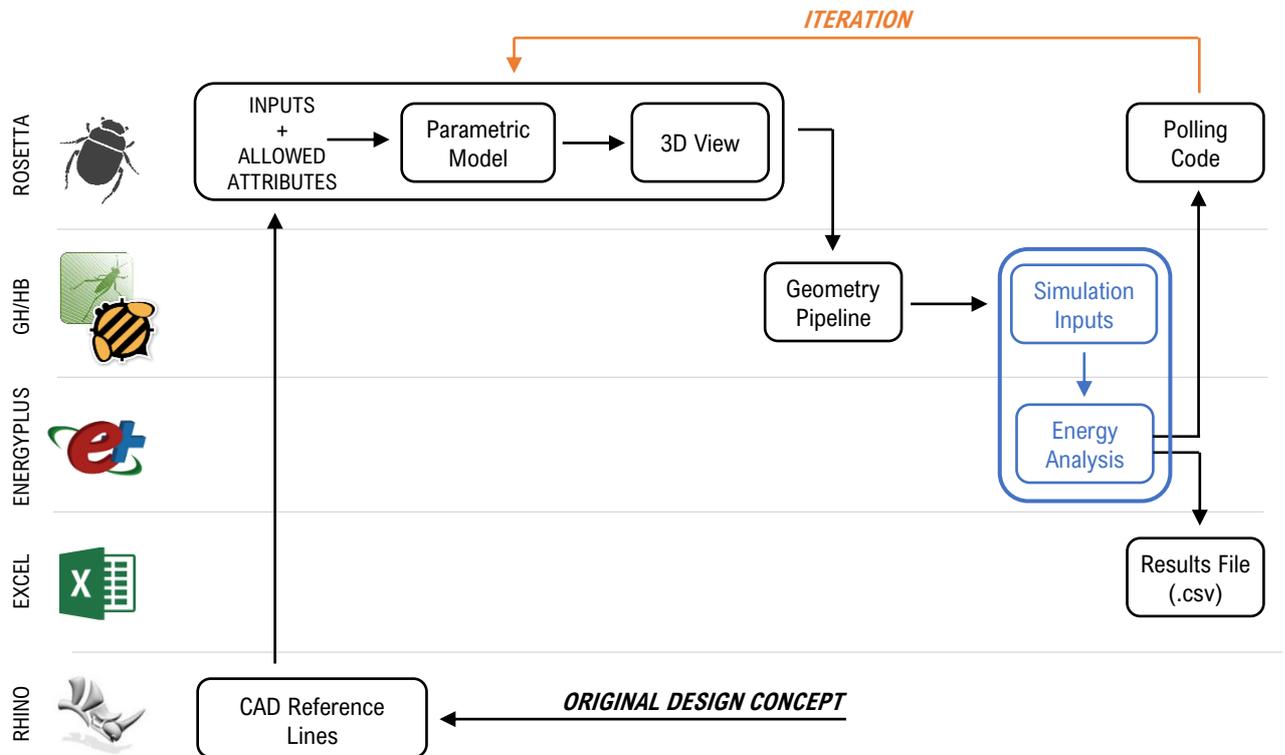


Figure 6 - Workflow

Main code - Building

The script uses the construction geometry modules present in BIM families and a lighting-simulation library to return the necessary surfaces for the analytic model that will be read by the Ladybug-Honeybee plugins. Facade areas are separated considering the program and building use; these sub-areas are related to the analysis zones for energy efficiency simulation. Shading devices and their respective restrictions are defined and, afterwards, we define the functions that create the slabs – the script imports the set of CAD lines that outline the perimeter of the occupied slabs, as well as a total floor slab that disregards the voids correspondent to the circulation areas. The total surface of the slab will be applied in the analytic model in case the indoor comfort and energy efficiency assessments are to be simulated separately.

INPUT: CAD lines from layers 'occupied slabs lines<floor nr.>' and 'slabs lines<floor nr.>'.

Next follows the definition of the functions that create the walls of the building, which are divided into exterior, interior and glazing types.

INPUT: CAD lines from layers 'facade lines<floor nr.>', 'glazing lines<floor nr.>' and 'interior walls<floor nr.>'.

The functions that create the geometry of skylights and roof are also defined.

¹¹ Grasshopper imports the selected layers from Rhinoceros in the beginning of the script.

INPUT: CAD lines from layers 'skylight' and 'slab lines<top floor>'.

Finally, the functions that execute all the building geometry are defined.

```
Pseudocode - Main code  
  
#lang racket  
(require rosetta/lighting-simulation)  
(provide create-building)  
  
Global definitions;  
  
Creation of windows/shading devices;  
  
Auxiliary functions;  
    Function that cleans the environment in the back-end;  
    Function that given a certain n number, returns its logical operator;  
  
Shading functions;  
    Function that defines the shading devices based on size and shading direction;  
  
Slab functions;  
    Function that defines total slab;  
    Function that defines occupied slabs;  
  
Wall functions;  
    Function that defines the exterior walls;  
    Function that defines the interior walls;  
    Function that defines the glazing walls;  
  
Roof functions;  
    Function that defines the skylights;  
    Function that defines the ceiling/roof slab;  
  
Building generation;  
    Slab functions;  
    Function that positions the shading devices;  
    Wall functions;  
    Roof functions;  
  
Create-building function.
```

Figure 7. Mockup script for building generation

Polling Code

The purpose of this script is to generate the indoor comfort and energy efficiency directly with Radiance and EnergyPlus. The code is currently under development. As of now, the analysis is still performed through Grasshopper, aided by the Ladybug-Honeybee plugins, and using the Colibri plugin to aggregate the simulation parameters and performance metrics output into an Excel (.csv)¹² file. Whenever the Geometry Pipeline of the Grasshopper script identifies a change in the input parameters, the data flow is recalculated and new simulations are executed, which are appended to the same file.

¹² CSV = Comma Separated Values

The “polling” code is already able to record simulation parameters and results in a .csv archive, and the replacement of the Colibri plugin is currently under development. This workflow change has the objective of modularizing the set of tasks that are being developed by the research group.

Pseudocode – polling

```
#lang racket
(require csv-reading)
(require "edificio-teste-v1.rkt")

Generic utilities;
  Define csv-columns header names;
  Define the function that, given a list of positions, returns all the geometric elements in their specified positions;

Analysis results;
  Define the function that, given a file path and the name of the parameters to extract from the file, retrieves the
  corresponding parameters from the first row of the file;
  Return first line after headers;
  Collect results from file path, moving the file to a backup directory;

Main program;
  Define the function of the default parameters 'out:ILL Within', 'out:ILL Above', 'out:ILL Below', 'out:Cooling'
  (kWh/m2) and 'out:Heating' (kWh/m2);
  Define the function that waits for a task to finish and only then return values;

Call the function that records all the parameters and results of the analysis in a .csv file;
```

Figure 8. Mockup script for post-processing of the simulation results

ANNEX II – EMS Script

!- ===== ALL OBJECTS IN CLASS: ENERGYMANAGEMENTSYSTEM:SENSOR =====

```
EnergyManagementSystem:Sensor,  
  BeamRadiation,                !- Name  
  EXT_WALL_SRF,                 !- Output:Variable or Output:Meter Index Key Name  
  Surface Outside Face Incident Beam Solar Radiation Rate per Area;    !- Output:Variable or Output:Meter Name
```

!- ===== ALL OBJECTS IN CLASS: ENERGYMANAGEMENTSYSTEM:ACTUATOR =====

```
EnergyManagementSystem:Actuator,  
  WINDOW01_Shading_Deploy_Status, !- Name  
  WINDOW_SRF_01,                 !- Actuated Component Unique Name  
  Window Shading Control,        !- Actuated Component Type  
  Control Status;               !- Actuated Component Control Type
```

```
EnergyManagementSystem:Actuator,  
  WINDOW02_Shading_Deploy_Status, !- Name  
  WINDOW_SRF_02,                 !- Actuated Component Unique Name  
  Window Shading Control,        !- Actuated Component Type  
  Control Status;               !- Actuated Component Control Type
```

```
EnergyManagementSystem:Actuator,  
  WINDOW03_Shading_Deploy_Status, !- Name  
  WINDOW_SRF_03,                 !- Actuated Component Unique Name  
  Window Shading Control,        !- Actuated Component Type  
  Control Status;               !- Actuated Component Control Type
```

!- ===== ALL OBJECTS IN CLASS: ENERGYMANAGEMENTSYSTEM:PROGRAMCALLINGMANAGER =====

```
EnergyManagementSystem:ProgramCallingManager,  
  EMS_PRG_CAL_MAN_NEW_ENVIR,     !- Name  
  BeginNewEnvironment,           !- EnergyPlus Model Calling Point  
  EMS_PRG_INIT_SHD_CNTRL_FLGS;   !- Program Name 1
```

```
EnergyManagementSystem:ProgramCallingManager,  
  EMS_PRG_CAL_MAN_BEFORE_PRED,   !- Name  
  BeginTimestepBeforePredictor,   !- EnergyPlus Model Calling Point  
  EMS_PRG_SHD_CNTRL;             !- Program Name 1
```

!- ===== ALL OBJECTS IN CLASS: ENERGYMANAGEMENTSYSTEM:PROGRAM =====

```
EnergyManagementSystem:Program,  
  EMS_PRG_INIT_SHD_CNTRL_FLGS,   !- Name  
  Set Shade_Status_Off = 1.0,     !- Program Line 1  
  Set Shade_Status_On = 3.0;      !- Program Line 2
```

```
EnergyManagementSystem:Program,  
  EMS_PRG_SHD_CNTRL,             !- Name  
  IF BeamRadiation >= 120, !- Program Line 1  
  Set WINDOW01_Shading_Deploy_Status = Shade_Status_On,  
  Set WINDOW02_Shading_Deploy_Status = Shade_Status_On,  
  Set WINDOW03_Shading_Deploy_Status = Shade_Status_On,  
  ELSE,  
  Set WINDOW01_Shading_Deploy_Status = Shade_Status_Off,  
  Set WINDOW02_Shading_Deploy_Status = Shade_Status_Off,  
  Set WINDOW03_Shading_Deploy_Status = Shade_Status_Off,  
  ENDIF;                          !- <none>
```

!- ===== ALL OBJECTS IN CLASS: ENERGYMANAGEMENTSYSTEM:GLOBALVARIABLE =====

EnergyManagementSystem:GlobalVariable,
Shade_Status_Off; !- Erl Variable 1 Name

EnergyManagementSystem:GlobalVariable,
Shade_Status_On; !- Erl Variable 1 Name

! =====

EnergyManagementSystem:OutputVariable,
BeamRadiation, !- Name
BeamRadiation, !- EMS Variable Name
Averaged, !- Type of Data in Variable
ZoneTimestep; !- Update Frequency

Output:Variable,*,Zone Windows Total Transmitted Solar Radiation Rate,hourly; !- Zone Average [W]
Output:Variable,*,BeamRadiation,Timestep; !- Zone Average [W]
Output:Variable,*,Surface Window Transmitted Solar Radiation Rate,hourly; !- Zone Average [W]