

P. Janssen, P. Loh, A. Raonic, M. A. Schnabel (eds.), *Protocols, Flows and Glitches, Proceedings of the 22nd International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2017*, 137-147. © 2017, The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

ALGORITHMIC-BASED ANALYSIS

Design and Analysis in a Multi Back-end Generative Tool

ANTÓNIO LEITÃO¹, RENATA CASTELO BRANCO² and
CARMO CARDOSO³

^{1,2,3}INESC-ID, Instituto Superior Técnico, Universidade de Lisboa,
Portugal

¹antonio.menezes.leitao@ist.utl.pt ^{2,3}{renata.castelo.branco|c.machado.cardoso}@tecnico.ulisboa.pt

Abstract. Estimating a building's performance is part of the engineering and architecture discipline. Nowadays, this estimation is done using analysis tools. In many cases, these analysis tools require specialized building models that are simplifications of the actual models. Unfortunately, the adaptations that need to be done to an existing model are tiresome and make the architect less willing to evaluate variations of the building design. Moreover, in the case of buildings with complex shapes, the analyses tend to be less reliable. These problems also occur when algorithmic approaches are used to generate the building design, as the algorithmic script needs to be adapted to satisfy the requirements of the analysis tool, or the manual adaptation of the generated model needs to be repeated each time the script is executed. To solve these issues we propose Algorithmic-Based Analysis. This is a Generative Design method that, utilizing a single algorithmic-based representation of a building, can generate not only the traditional CAD or BIM model, but also specialized models for use in different kinds of analysis.

Keywords. Generative Design; Building Performance; Analysis; Performance-based Design; Algorithmic-Based Analysis.

1. Introduction

3D modeling applications changed the design paradigm, altering the way architects approach building design. This shift opened the door for yet another technological leap: the development of parametric tools driven by algorithmic processes. Using an algorithmic approach, instead of creating a model of the intended design, architects can implement an algorithmic description. This description, when executed by a computer, creates the actual design and, when appropriate, variations of that

design. This allows architects to not only achieve new shapes (Yusuf 2012), but also to more rapidly change those shapes (Lopes & Leitão 2011). This approach enables faster exploration of variations of the envisioned design and promotes a more informed search for the optimal variations. As can be expected, the pursuit of higher levels of complexity in buildings' geometry has challenged fabrication and construction methods (Kolarevic 2011). This has made building designs less predictable from, e.g., the thermal, lighting, and acoustics points of view. As a result, it became critical to have an evaluation of buildings' performance.

Analyzing a building's performance has long been part of the engineering discipline. In the past, these analyses have been done through tiresome and error-prone manual calculations. These calculations have recently been implemented in specialized computational tools that perform them using data extracted directly from a 3D model of the building, as well as from information regarding the building's location and materials. The use of these tools has greatly reduced the amount of time needed to perform analysis, as well as the rate of human error.

Analysis tools require a geometric model of the building that, unfortunately, is frequently different from the one produced in the 3D modeling tool, thus requiring a translation process (Aghemo et al. 2013). Usually, this forces the analyst to do some preparation work before an analysis can be done, which becomes tiresome when buildings are generated using algorithmic approaches, as it needs to be repeated for each variation of the building. Given that the time and effort spent in the preparation for analysis are not negligible, performing repetitive analyses inhibits the designer's desire to perform more than just a few of them. This deterrence drastically reduces the potential of the algorithmic approach for finding the design that maximizes the desired building's performance.

In order to solve this conflict between the capability to algorithmically generate a large number of design variations and the effort needed to prepare each variation for analysis, we propose a set of modeling operations that can not only generate the usual 3D model, but also generate specialized versions that are pre-prepared for analysis. Moreover, to deal with the fact that different analyses might require different representations of a building, we propose using an implementation of the modeling operations that is dependent on the modeling goal. This means that a 3D model that is algorithmically generated for, e.g., rendering, employs different geometric objects than the same model when generated for, e.g., daylight analysis.

In this paper, we present an implementation of this proposal, which we call Algorithmic-Based Analysis. We show its potential for bringing architecture closer to performance-based design, a process where the building is conceived with the aim of fulfilling performance requirements such as thermal loads, lighting or structural behavior.

1.1. ANALYSIS SOFTWARE

Nowadays there are many tools available for daylight, thermal, and energy analysis, most of which, require the digital model to be in a particular format. Changing a 3D model's format many times causes information loss, and, in some cases, the analysis tools cannot even deal with complex geometry (Moon et al. 2011), which

forces the user to remodel a simplified version of his creation.

Ideally, the modeling and the analysis tools should exchange data seamlessly. This is actually possible in some design programs that have analysis tools built into them. For example, Revit, a program that supports the BIM paradigm, provides structural, solar radiation, and thermal analysis. This avoids the need to rebuild the model or export it to other formats. Unfortunately, that is not the case for other important analysis tools, such as Radiance (Fritz & Macneill 2016) or DAYSIM (Reinhart 2016), for daylight analysis, and EnergyPlus or Ecotect, for energy simulation. Each of these programs requires a building modeled according to specific geometric requirements and exported in a particular format.

To overcome this problem, more integrated approaches began appearing in the market, for example, DIVA - Design Iterate Validate Adapt - (Niemasz 2016), a plug-in for Rhinoceros that incorporates the analysis provided by Energy Plus, Radiance, and DAYSIM. The main drawback of DIVA, however, is the amount of adaptation that it requires in the building's model so that it complies with the tool's requirements. This adaptation might require considerable efforts, and is, therefore, unsuitable for generative processes that can produce several different variations of a building. This issue will be further discussed in section "Comparing Approaches".

Despite these shortcomings, DIVA is currently integrated with Grasshopper (Davidson 2016), a Generative Design (GD) tool, allowing for the mechanization of the analysis process. However, this analysis process still requires the modeled building to follow the requirements of Grasshopper and DIVA. Furthermore, the time it takes to do even the simplest analysis makes the process unsuitable for the highly interactive nature of Grasshopper.

As is apparent with the DIVA-Grasshopper integration, the use of GD is not a solution that avoids the need to generate specialized building models for different kinds of analysis. Instead of forcing the designer to manually adapt his models, it now forces him to manually adapt the programs that generate the models. However, if we focus on portable GD (Lopes & Leitão 2011), an approach that allows the same program to generate equivalent models in different CAD/BIM tools, then it becomes possible to use the same modeling operations for different purposes, namely, analysis. Moreover, by implementing specialized analysis operations, that are taken into account by specific analysis tools and are ignored by other tools, we allow the same algorithmic description of a design to be used for advanced analysis without disrupting the model's other uses. We call this approach Algorithmic-Based Analysis.

2. Algorithmic-Based Analysis

In order to support Algorithmic-Based Analysis we first need portable GD. To this end, we used Rosetta (Lopes & Leitão 2011), a programming environment that supports scripts written in various programming languages, and which generates equivalent models in a series of CAD applications that are known as back-ends. Due to the recent integration of BIM back-ends in Rosetta (Feist et al. 2016), it also has operations available for modeling the parts of a BIM model. These oper-

ations include slabs, walls, columns, and beams, among others. Rosetta provides an abstraction layer that encompasses the common operations, such as shape constructors and transformations, amongst the back-ends. The abstraction layer is then translated into the requirements of each back-end. This means that the same script is interpreted differently by each one of them. Despite this fact, equivalent models are seamlessly produced in the various back-ends.

As explained before, analysis tools, such as Radiance and EnergyPlus, require a simplified version of the 3D model. To this end, we extended Rosetta with additional specialized back-ends with the purpose of generating models for analysis. Hence, from the same script that produces a detailed model in a CAD or BIM application, Rosetta's analysis back-ends generate the specialized version of the model that is needed for the desired analysis. For instance, in the case of the back-end tool for Radiance and DAYSIM, slabs, beams, and columns are interpreted as mere planes and surfaces. This information adaptation will also reduce the computational resources required to perform the analysis, and therefore reduce the time that it takes to run the analysis.

In the next section, we describe the Algorithmic-Based Analysis workflow in the context of Rosetta, and we compare this workflow with the well-known DIVA-based workflow. After this section, we present an evaluation of the process in a case-study.

2.1. ANALYSIS WORKFLOW

Before using an analysis tool, a designer will usually have already modeled their design as a generic 3D model. These models frequently have far more detail than is required for the analysis. This generic model would be a good starting point if the designer's analysis tool could use it, but this is not always the case. Sometimes the analysis tool requires the designer to change the model, e.g., to simplify it. In other cases, the designer must rebuild the model completely. As an example, figure 1 (left) synthesizes the information exchange needed to perform an analysis using DIVA and Radiance. In this exchange, the user begins by building a 3D model in Rhino. After some manual changes to the geometry (1) the model is exported to Radiance and evaluated. It then returns the results back to Rhino where the user can see them (2).

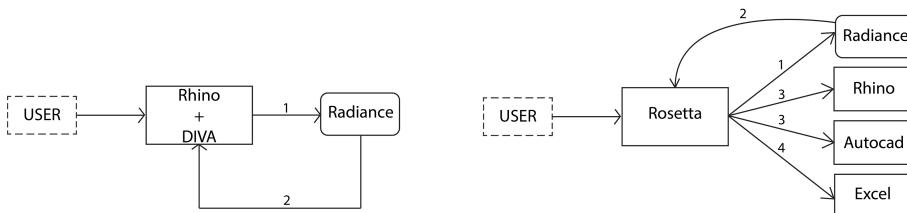


Figure 1. The DIVA workflow (left) and the Rosetta workflow (right).

When using Rosetta's analysis back-end, the geometry of the model is generated in the way required by the analysis tool. The practical result is that, despite the user's script being the same, the model given to the analysis back-end is independent and different from the model produced in the visualization back-end. The model generated by Rosetta contains all required elements and simplifications that the user would otherwise have to do manually. Figure 1 (right) illustrates this process for the Radiance tool. The user scripts his design in Rosetta, with no concerns regarding the analysis tools requirements. (1) Rosetta then sends only the necessary information to Radiance for the analysis. After the analysis is concluded, (2) the results are retrieved and (3) displayed in the 3D modeling backend, e.g., Rhino or AutoCAD, or (4) they are exported for further processing, for example, in Excel.

We call this method Algorithmic-Based Analysis because the level of detail or features of the algorithmically generated building are produced according to the analysis needs. This approach makes it easier for the architect to introduce changes to the design script, which are then automatically translated according to the analysis tools' requirements. Therefore, as change becomes effortless, the architect is more willing to try variations of his model. Moreover, possible misinterpretations and information loss are eliminated. This is due to the fact that the model geometry, that is fed to the analysis software, is generated according to the analysis software's requirements. This makes the analysis results more trustworthy.

3. Evaluation

To evaluate our approach, we selected two distinct case studies: Astana National Library, from Bjarke Ingels Group, and Absolute Towers from MAD Architects. Astana's façade is covered with triangular panels with different openings defined according to the building's light exposition. The Absolute Towers feature smooth, unbroken balconies all around each floor. In addition, as the buildings rise, the slabs rotate, offering them a curvaceous figure. A perspective of the library's structure can be seen in figure 2 on the left, while a detail from one of the towers is shown on the right.

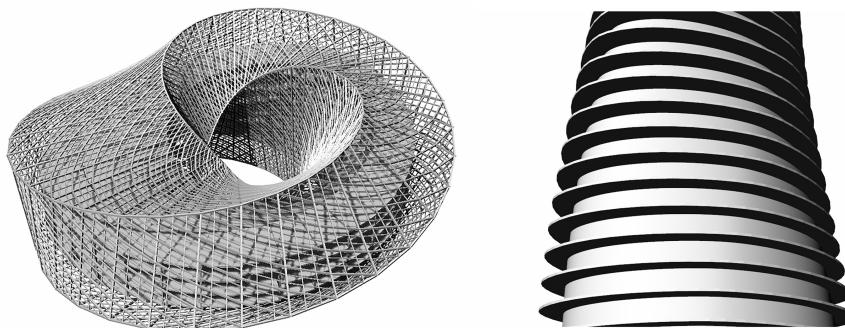


Figure 2. Astana National Library's Structure (left) and Absolute World Towers' detail (right).

Both models were built using Rosetta making it possible to change several aspects of their design, including the buildings' height, width, diameter, number of floors, façade panels (in Astana's case), and slab rotation (in the Absolute Towers). Regarding the Astana case study, for a better thermal performance of the building, the organization of the panels requires an analysis of the sun exposure. In the Absolute Towers's case, the buildings have non-orthogonal shapes that also benefits from an automated analysis. We analyzed the models using Rosetta's Radiance back-end with a Radiation Map metric.

3.1. COMPARING APPROACHES

In order for the model to be analyzed by DIVA it first needs to be prepared. This preparation involves introducing additional objects, adapting the geometry of the building elements, or restructuring the arrangement of layers. In the end, the model that is used for analysis is not the same that represents the actual building. This causes a maintenance problem, as all changes in the design need to be propagated to either models, or, alternatively, the preparation of the updated model for analysis needs to be repeated. Both options have considerable drawbacks. By making a new Rosetta back-end that automates analysis, the speed of the process is increased and possible setup mistakes are avoided. Moreover, in the event that the analysis options are the same for every iteration of the building, there is no need to re-introduce these options.

3.1.1. Overcoming Setup requirements

When initiating the setup of the model for analysis in Rhino, DIVA requires a ground plane that is composed of an adequate material. However, it is frequently the case that the ground plane is not part of the original model and, thus, the designer might forget to include it. Moreover, despite the considerable influence that the ground plane might have in the analysis, there is no guidance regarding its appropriate size. The same problem happens when using the DIVA-Grasshopper plug-in, as no information is provided regarding whether the ground plane should be modeled, or how it should be modeled. For cases like this, the automatic creation of the necessary surface for the analysis facilitates the analyst's task and speeds up the process of setting up the model. Using our Radiance back-end for Rosetta, the ground plane is automatically generated and its size is computed according to the size of the bounding box of the building. Furthermore, the ground plan will be assigned a default material, which can be easily changed by the designer.

When constructing a 3D model, the designer usually creates most of it using solid objects. DIVA, however, requires all elements for the analysis to be mere surfaces, poly-surfaces, or meshes. If these conditions are not met the program either refuses to perform the analysis or presents distorted results. This means that users that create models using solid objects must simplify them before analysis, e.g., by transforming solids into surfaces. Our approach automates this process by generating the elements in the model according to the requirements of the analysis tool. The user may program using slabs, beams, glass, etc., but, depending on

the back-end being used, those operations will generate solid objects or the corresponding surfaces. In the case of the Radiance back-end, only the relevant surfaces are generated.

When using DIVA-Grasshopper it is possible to execute the analysis on boxes and other solid objects. This represents a considerable advantage when compared to DIVA for Rhino since the user does not need to manually introduce changes to the model. However, in this case, the analysis grids cover all faces of the surface. This can be a disadvantage if the user only intends to analyze one part of the building. Moreover, by having additional surfaces to analyze, the process becomes more time-consuming.

3.1.2. Dealing with Materials

In order to specify the building materials, both DIVA and DIVA-Grasshopper associate materials to layers. This forces the designer to classify the geometric elements according to their material, which can be tiresome and might invalidate an existing layer organization. In Rosetta, materials are not associated with layers. Due to its CAD-BIM portability, all models produced in Rosetta benefit from the BIM approach to material information. Even if the geometry is meant to be generated by a CAD tool, like Rhino, the user can take advantage of default BIM families. As such, Rosetta contains all of the model's material data that is needed for the analysis despite generating only simple geometric elements in the chosen CAD application. For analyses, instead of using layers, the list of materials is extracted automatically from the generated model. This results in the analysis utilizing the actual materials that the designer selected for each element or the materials that were assigned by default.

3.1.3. Dealing with Non-Planar Surfaces

Most simulations offered by DIVA are node-based, meaning that a grid of points must be created to represent sensor nodes placed where the light levels should be evaluated. For each surface, DIVA assigns a single normal vector and computes the location of the sensor nodes at a given distance from the surface, along the direction of the normal vector.

DIVA's method for node distribution is a reasonable course of action when the model is made of boxes because each surface will be a rectangular plane and, thus, can be described by just one normal vector. Unfortunately, this method is not adequate for non-planar surfaces, which require a vector field. In DIVA's defense, it should be mentioned that the tool's documentation explicitly mentions that "complicated geometry can be pre-meshed before running the metrics." However, no indications are provided regarding this process and it is possible that the designer will not even know which geometry fits in the "complicated" category. Moreover, pre-meshing is yet another manual step that needs to be re-done every time the model changes. Unfortunately, failing to provide a proper description of the surfaces in the model can have a considerable impact on the quality of the analysis. As it happens with most software tools, if the provided description of the model is not accurate, the results will be incorrect, although that might not be obvious for

the designer.

For the final version of the panels from our first case study, the wrapping façade of Astana Library must be analyzed as a whole. Figure 3.A shows the results, of the node's placement of the library's façade, completed in DIVA without the pre-meshing step. As is possible to see, the normal vectors were incorrectly computed for almost all parts of the surface, thus making the location of the analysis sensors incorrect for significant fractions of the building envelope. This, in turn, causes both an incorrect analysis and an incorrect visualization of the results in which DIVA misorients the color-coded quadrangles.

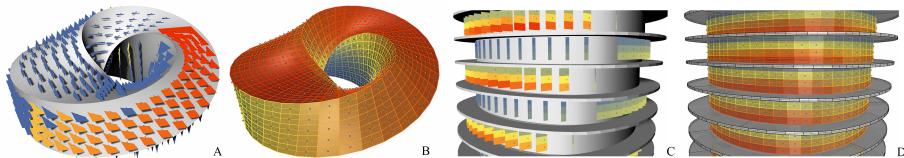


Figure 3. Nodes generated by DIVA and by Rosetta and resulting analysis' color-coded quadrangles in both case studies.

On the other hand, in Rosetta's Radiance back-end, each non-planar surface is automatically meshed according to the node separation, without the designer needing to be aware of this process. Moreover, for each surface, the corresponding vector field is computed, and the correct location and orientation of the sensors is provided to the analysis software. As is possible to see in figure 3.B, all nodes were placed with the same offset from the given surface and the color-coded quadrangles produced by the algorithmic analysis are correctly placed. This contrasts with the placement done by DIVA whereby half of the sensors were located beneath the surface. A similar scenario was encountered when performing the analysis in our second case study, the Absolute Towers. The goal was to analyze the radiation received by the glass walls on each floor. However, these elements are also non-planar and, as DIVA assumes a single normal direction for the entire surface, the color-coded quadrangles were also misplaced. The resulting analysis, presented in figure 3.C is, therefore, equally unreliable. In order to solve this problem, it would be necessary to rebuild the curved surfaces as sets of sufficiently small planar surfaces, an additional laborious step for the designer that is not needed in our approach, allowing for the correct analysis presented in figure 3.D.

3.1.4. Visualization of the Analysis

One of the important limitations of DIVA (or DIVA-Grasshopper) is that it only works in Rhino. On the other hand, the Radiance back-end for Rosetta is independent of the back-end that is used to visualize the results of the analysis. This means that it becomes possible to use whichever back-end the designer prefers. In figure 4, we can see the analysis results visualized in both Rhino and AutoCAD back-ends.

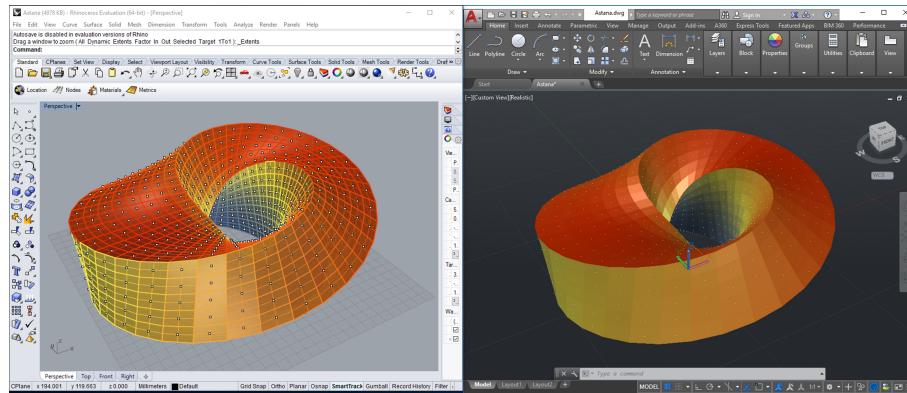


Figure 4. Analysis results generated in AutoCAD and Rhino.

Another important feature is that the analysis results are cached. Given that an analysis can take minutes or even hours to run, it is important to avoid wasting time. To this end, we cache the analysis results, so that when the model did not change relative to the previous analysis, the analysis results can be immediately retrieved with no impact in the performance of the GD process.

4. Conclusions

This paper presented Algorithmic-Based Analysis, a novel approach for the analysis of algorithmically generated buildings. The support the approach we extended a portable GD tool with additional back-ends for different kinds of analysis. Each back-end is specialized to generate a model that satisfy the requirements of each analysis tool without needing any further manual alterations from the user. This not only makes the results more reliable but also encourages architects to test a lot more variations of the intended design. We demonstrated the capabilities of our approach by evaluating the solar radiation on two buildings with complex geometries.

It is important to stress that our approach relies on a GD process and, thus, is not directly applicable to manually created designs. Given the flexibility allowed by Algorithmic-Based approaches, we consider this an advantage. However, it is also important to note that although GD can be very helpful in initial stages of the design process, it does not yet present a good return on the investment required to model a building up to its final stages, when a considerable amount of detail needs to be modeled.

Currently, the supported analysis tools include Radiance and DAYSIM and we are working on preliminary support for Energy Plus. We also plan to include additional back-ends, particularly, for acoustic and structural analysis.

As soon as the different back-ends are in place, we plan to use them for multi-objective optimization. Since the model and the analysis are generated in the same tool there is no information loss when changing the type of analysis. This means the results are more accurate and the building can be optimized according to its performance in all analysis back-ends. We have already tested this approach by

using the Monte Carlo method to optimize the façade of a building according to the Useful Daylight Illuminance (UDI) (Nabil & Mardaljevic 2006). We are now planning to implement more sophisticated optimization algorithms.

Acknowledgements

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

References

- Aghemo, C., Blaso, L., Dalmasso, D., Erba, D., Del Giudice, M., Osello, A., Fracastoro, G., Pellegrino, A. and Ruffino, P.: 2013, Interoperability between building information models and software for lighting analysis, *Building Simulation Applications BSA 2013 1th IBSA Italy Conference*, Bolzano, 69-78.
- Davidson, S.: 2016, "Grasshopper: Algorithmic Modeling for Rhino". Available from <<http://www.grasshopper3d.com/>> (accessed 23rd December 2016).
- Feist, S., Barreto, G., Ferreira, B. and Leitão, A.: 2016, Portable Generative Design for Building Information Modelling, *CAADRIA*, Melbourne, 147-156.
- Fritz, R. and Macneill, A.: 2016, "RADSITE | radiance-online.org". Available from <<https://www.radiance-online.org/about>> (accessed 23rd December 2016).
- Kolarevic, B.: 2001, Designing and manufacturing architecture in the digital age, *Architectural information management, 19th eCAADe*, Helsinki, 117-123.
- Lopes, J. and Leitão, A.: 2011, Portable Generative Design for CAD Applications, *ACADIA 2011*, Banff, Alberta, 196-203.
- Moon, H.J., Choi, M.S., Kim, S.K. and Ryu, S.H.: 2011, Case studies for the evaluation of interoperability between a BIM based architectural model and building performance analysis programs, *12th Conference of International Building Performance Simulation Association*, Sydney, 1521-1526.
- Nabil, A. and Mardaljevic, J.: 2006, Useful daylight illuminances: A replacement for daylight factors, *Energy and buildings*, **38**(7), 905–913.
- Niemasz, J.: 2016, "DIVA for Rhino: Environmental Analysis for Buildings". Available from <<http://diva4rhino.com/>> (accessed 23rd December 2016).
- Reinhart, C.: 2016, "DAYSIM". Available from <<http://daysim.ning.com/>> (accessed 23rd December 2016).
- Stavric, M. and Marina, O.: 2011, Parametric modelling for advanced architecture, *International journal of applied mathematics and informatics*, **5**(1), 9-16.
- Yusuf, H.O.: 2012, *The impact of digital-computational design on the architectural design process*, Master's Thesis, University of Salford.