

Integrated Algorithmic Design

A single-script approach for multiple design tasks

Renata Castelo Branco¹, António Leitão²

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

^{1,2}{renata.castelo.branco|antonio.menezes.leitao}@ist.utl.pt

Many great architectural endeavors today engage in a multi software approach, as each specialty involved needs a different software, and different task required from the architect, such as 3D modeling, analysis or rendering, also benefit from the use of different tools. Combining them in the same process is not always a successful endeavor. A more effective portability mechanism is needed, and Algorithmic Design (AD) has the potential to become one. This paper explores the advantages of the algorithmic approach to the design process, and proposes a methodology capable of integrating the different tools and paradigms currently used in architecture. The methodology is based on the development of a computer program that describes not only the intended model, but also additional tasks, such as the required analysis and rendering. It takes advantage of CAD, BIM and analysis tools, with little effort when it comes to the transition between them.

Keywords: *Algorithmic Design, CAD, BIM, Analysis tools*

INTRODUCTION

The design process of an architectural creation has seen many changes over time, and more so over the past few decades, as new software emerges nearly every day. Representation methods are amongst the ones that shifted the most (Kalay, 2004).

After centuries of producing precise technical drawings, perspectives and models by hand, architects found these tasks facilitated by Computer Aided Design (CAD) software. Designed to help creative users in an “interactive performance of ‘man-machine’ problem solving engine” (Llach, 2013: 18), CAD tools support sketching, drafting and image-altering, as well as 3D modeling and rendering (Brandon and McLain-Kark, 2001). Further along came the Building Information Modeling (BIM) paradigm, hailed as

one of the most promising developments in Architecture, Engineering, and Construction (AEC) industries, as it pledges to bring the three closer together in a more integrated design and construction process (Eastman et al., 2008).

Despite the new methodologies introduced by these systems, architects still struggled to fully engage in a thoroughly digital process. A few decades ago, modeling tools were often incorporated only into later stages of the design process, where the production of precise and detailed construction drawings and specification documentation usually took place (Brandon and McLain-Kark, 2001).

In order for modeling tools to become more than documental assistants and actually play a deeper part in the design process, architects must be al-

lowed to utilize them with the same authority they might apply to a pen or a compass. Even though the software spectrum is quite wide, each one has a particular workflow to which the user must adapt, conditioning his creative process. Currently, the stakes are changing, partly due to Algorithmic Design (AD), an algorithmic approach to architectural design. With the use of AD the user is able to transcend the limitations the software might impose on him (Terzidis, 2006), and make use of the modelling tools in his own way.

Adding to this collection of tools, another important group of software has been setting its ground in the design processes of the architectural agenda - analysis tools. These tools brought forth a new concept: Performance-based design - a design process informed by a deeper analysis and understanding of the environmental context of the project (Oxman and Oxman, 2014).

Never before have there been so many, or so diverse tools, techniques and methods for design. Architects are spoilt for choice and, because of this, post-digital design has become a task of curious manipulation, speculation and experiment (Sheil, 2008). The current architectural design process already makes use of different paradigms, like CAD and BIM, and many other tools. However, they do not effectively communicate with each other. Our goal is to merge some of the most relevant paradigms and tools in a seamless workable process that architects can follow.

PORTABILITY ISSUE

The chronological order in which all referred tools have entered the market does not entail their sequential replacement. Each of these tools came as a response to specific needs that have arisen over time, and both CAD and BIM tools present distinct advantages to the modeling process.

Nowadays, practitioners are embracing the BIM paradigm, as it not only promotes a faster development of the models with the use of pre-modeled elements available in the libraries or families, but also

automates time-consuming tasks such as arranging maps of quantities and costs, allowing the architects to perform alterations in their projects with real time feedback on the production's final numbers and expenses. In a CAD environment, there are no object libraries of families immediately available in the program. Hence, the user can either model the geometries from scratch or use pre-modeled blocks. CAD blocks, nevertheless, are not as flexible as BIM objects. While these present a set of parameters that allow several variations of the object to be created, CAD blocks must be modified by hand if more complex variations than for instance, scaling and moving, are desired.

Nevertheless, CAD tools, as free-form surface modeling tools, provide greater freedom in form creation (Zboinska, 2015). This may justify a preference in developing the early stages of the project, such as form and concept experimentation, in these applications. Furthermore, the idea of bringing architecture closer to the construction ideal, may somehow inhibit the creative potential of the designer. In a BIM environment the user is obliged to model in a sequential manner that follows the constructive logic. While this proves to be an advantage in resolving construction issues further ahead in the process, it possibly harms the creative workflow of the architect, which would possibly rather model his design in a completely different order and fashion.

In order to take advantage of both approaches, some practitioners begin their design explorations in CAD environments, and when satisfied with the overall shape, transition to BIM. However, the paradigms are so different, that transferring models from one to the other is hardly a simple task. In most cases portability cannot be achieved at all and the architect end up having to rebuild the models from the start.

The same case is verified regarding analysis tools. They are more important now than ever before, as higher complexity levels are being achieved in building design, making them less predictable, e.g., from the thermal, lighting, and acoustics point of view. Yet, analysis tools suffer from portability issues as

well. Several demand a particular model of the building, different from the one used in the CAD or BIM modeling tool, thus requiring a translation process or, again, a rebuild (Aghemo et al., 2013).

Many great architectural endeavors today engage in a multi software approach, particularly when diverse companies participate in the project, such as engineers from different specialties, contractors, etc, each developing their respective part of the project in their work software. Ateliers Jean Nouvel's 100 11th avenue New York' curtain wall façade, required a variety of software interoperability, namely: Digital Project, Rhinoceros, AutoCAD, CATIA, Robot and Strand. PDF and IGES files were also used (Eastman et al., 2008). The Shanghai Tower, from Gensler and Tongji Architectural Design Institute, used Rhinoceros, Grasshopper and Revit for performance-based design (Kensek and Noble, 2014). The tennis center in Hangzhou sports park was essentially developed in Grasshopper (Miller, 2011), with different sets of scripts producing different files and models for different purposes: a wire frame model for structural analysis in another software and 3D DWGs were exported to Revit (Kensek and Noble, 2014).

A mechanism is needed that connects all these tools in a more effective manner. AD has the potential to become a portability mechanism on its own, since mathematical descriptions of the designs are oblivious to any software. Nevertheless, and for all the limitations it vows to surpass in the use of modeling tools, AD is still coming short when it comes to the issue of portability. A large amount of programming tools is already available in the market, for both beginners and/or more experienced programmers, which connect to either CAD or BIM tools. Some are also capable of connecting to analysis tools as well. Nevertheless, it is still difficult to port AD programs between different CAD and BIM applications (Ferreira and Leitão, 2015).

With so many possibilities regarding software and modeling techniques available, the architect has the unique opportunity of joining the best of differ-

ent approaches together in his own design process. He needs only a tool that allows him to do so in a smooth and continuous way, with no effort when it comes to transitioning between approaches.

INTEGRATED ALGORITHMIC APPROACH

We propose an integrated algorithmic approach to design - a methodology based on the development of a single script that, not only describes the intended model, but also the required analysis. Moreover, it takes advantage of CAD, BIM and analysis tools, with little effort when it comes to the transition between them. The proposed method aims to cover the relevant phases of the design process, exploiting the features of the different tools, without bending to their imposed workflow. Hence, the manner in which the designer seizes the tools becomes part of his personal creative approach.

The methodology presented is outlined in three main stages, explained in the following order: (1) CAD modeling for an initial exploratory form and concept stage - 1st phase; (2) BIM modeling for a more detailed stage in the process - 2nd phase; (3) analysis integration - may occur during the 1st and/or 2nd phase. A scheme of this practical application of the methodology can be seen in Figure 1.

Using a programming environment as modeling tool, the architect begins modeling his design intent within the CAD paradigm, and can visualize the result in a CAD tool. He may wish to include performance data in early stages of the design, hence, while still in the "CAD phase" of the process. The programming tool connects to the analysis software and exchanges the necessary data to the algorithmic model. In a more detailed phase of the program, the user shifts to the BIM paradigm, visualizing the modeled geometry in a BIM software. The performance analysis may instead be called upon on this stage, for which the process is identical. Finally, the detail modeled in BIM can not only satisfy construction purposes, but may also include additional decorative elements in order to sell the project's image to a possible client.

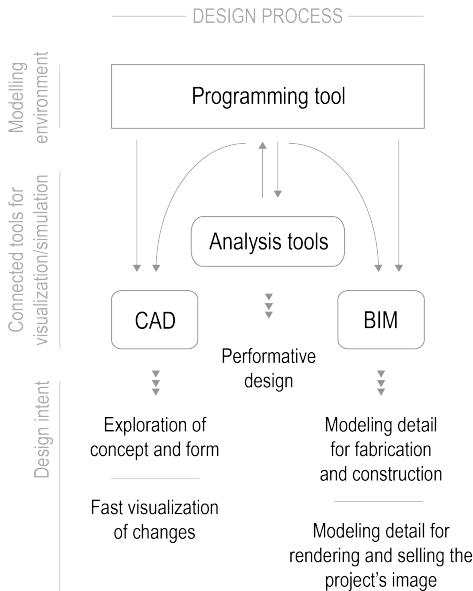


Figure 1
Integrated
Algorithmic Design
methodology

back-ends specialized for the generation of models for analysis, namely in Radiance and EnergyPlus. Most analysis tools require simplified versions of the 3D models, hence, from the same algorithmic description that produces a detailed model in a CAD or BIM application, Rosetta's simulation back-end generates only the simplified version of the essential elements needed for the analysis. For instance, depending on the analysis back-end, slabs, beams and columns might be interpreted as mere planes and surfaces. This automation process spares the architect from the tiresome work of adapting or reconstructing the model for analysis purposes.

Experimenting with Form and Concept

As we have previously discussed, CAD tools are more advantageous in an initial stage of the model as they are more flexible and present a better performance when compared to BIM tools. Not only do they allow for the generation of more complex geometries that some BIMs just cannot process, but they also allow for a constraint free modeling workflow, where no sequences or precedencies are imposed. Furthermore, since they can generate geometry faster, an architect can test a wider range of solutions for their design in a shorter time span.

When modeling with Rosetta, architects are able to write parametric descriptions of their designs that allow a wide range of possible results depending on how the parameters are manipulated. While modelling for CAD, the designer can use almost every operation available in the software, through Rosetta. The abstraction layer contains the common functionalities amongst CADs, such as procedures to create geometric shapes, like circles and boxes, and procedures that apply geometric transformations, including, translations, lofts, extrusions and sweeps (Leitão and Lopes, 2011).

Furthermore, the abstraction layer also contains a set of instructions that convert BIM object functions into operations that CAD back-ends can understand. For example, a beam command with two given points as parameters, is understood by Roset-

Programming Environment

The modelling tool we use is Rosetta (Leitão and Lopes, 2011), a programming environment that supports portable AD, and thus allows the architect to experiment an integrated algorithmic design process. Rosetta supports algorithmic descriptions written in various programming languages and allows the generation of their respective results in a series of CAD and BIM applications. This is possible due to a front-end/back-end architecture: the IDE connects programming languages (front-ends) suitable for beginners, such as Racket, Python, and Processing, to CAD or BIM applications (back-ends), as well as OpenGL for fast visualization. Currently, AD programs written in Rosetta can be generated in SketchUP, Rhinoceros, AutoCAD, Revit, ArchiCAD and OpenGL.

Just as different CAD and BIM tools were integrated in Rosetta, several analysis tools are currently going through a similar process. The tool is presently being developed to include additional

ta/AutoCAD as a right cuboid placed in space from one point to the other. Much like the beam command, similar abstractions exist for slabs, columns, walls, etc. This means that the architect may choose to shift parts of his program to the BIM paradigm while still modelling for a CAD back-end. This will grant a smoother transition to BIM further ahead in the design process, while still allowing for form experimentation of the elements in a CAD environment.

Transitioning To BIM

Differently from CAD, Modelling in BIM implies some significant loss of freedom, as some interdependencies are required and the order in which the elements are generated matters as well. The speed of the generation is also affected, since BIM applications must process more information in the creation of each object, as well as detect conflicts between the elements.

Nevertheless, as the project evolves into greater detail, shifting to the BIM paradigm has proven to greatly reduce time and effort spent on the modelling process, as well as on the production of documentation. In a BIM environment the architect can take advantage of all the information available in the libraries, saving a lot of time, as he needs not to model every single geometric element, as he would in CAD. In fact, in CAD, architects would not likely reach the level of detail needed for construction nor for detailed renderings, for example. These tasks are most commonly left to construction experts, in the first case, and rendering plus image editing software, in the second. However, BIM tools bring forth the possibility to reach higher levels of detail in the same model and with less effort than ever before for the architect.

Transitioning to BIM, the designer can take advantage of all the semantics embedded in his program, that in CAD was not as useful. Due to all the information included in Rosetta's abstraction layer, with the same program fragment that in a CAD back-end produced only geometry, the architect obtains, in a BIM back-end, more data, such as default mate-

rials and all information associated to it like weight, density, cost, etc.

Incorporating Analysis in The Design

The design intent influences the way through which analysis data is incorporated in the process. For form-finding explorations, optimizations methods are commonly used. This process entails an established communication channel between the analysis tool and the algorithmic one, where the later provides the former with a sequence of variations of the original model, produced according to the results obtained by the analysis. The cycle is repeated until the optimization criterion is met. This is usually a rather time-consuming process and a computationally-intense one. Additionally, it is often difficult to understand if the obtained result represents the global optimum, or simply a slightly better solution (Nguyen, Reiter and Rigo, 2014).

For a more controlled enhancement of the building's performance, architects can opt for a direct approach. When considering few criteria, the design solution that performs best can often be deduced without an optimization algorithm. As a practical example, one can envision an architect wishing to base the design of the façade elements on the building's solar exposition, to guarantee the best thermal or lighting conditions in the interior space. In such a case, he would need only to perform the analysis once in order to collect the solar exposition data, and input it into his algorithmic description of the design. This process, while not so adequate for multi-criteria optimization, is simpler and faster than optimization methods and grants the architect full control over the process and the final result.

Using Rosetta, just as the same script can be interpreted by CAD and BIM applications to generate similar geometries in each one, the same is now possible with analysis tools, like Radiance and DAYSIM. Furthermore, these tools give back information retrieved from the analysis they perform. Using an algorithmic based approach with Rosetta, the results are simultaneously transferred to our program and

can immediately be used in the modelling process, either in a direct input approach or in a continuous loop of information transfer between the model and the analysis tool.

Selling the Product

Modeling non-structural detail for rendering purposes is also a crucial part of the design process. As selling the product's image is of fundamental importance in architectural practice nowadays, the ability to produce detailed renders of the described model is a great asset. We believe the same script that generates all the BIM elements relevant for construction, and simplified geometries representing the model for analysis tools, should also be able to generate a specific model for rendering. This should be a fully finished and furnished model, created for the sole purpose of generating quality renders capable of properly selling the building's image and aimed ambience.

The possibility of doing so in with a parametric model, allows the architect to take advantage of this marketing strategy at any stage of the design process. If the objects placed in the model are algorithmically anchored to strategic locations of the model, implementing changes in the project's shape will not affect them. The objects are automatically relocated in accordance to the changes made. This means the architect can present his ideas to the client through fully detailed renders, make changes to the model according to client's requests or other needs that may surge in the unfolding of the project, and generate new detailed renders at a click of a button.

Rendering, a task that is normally the last to be executed in a manual approach and, given the time it consumes, one the architect is usually unwilling to repeat many times, is a much faster endeavor when using an algorithmic approach. With integrated algorithmic design, this should be an exercise that, after the initial burden, can be repeated countless times without effort, for as many variations as the model may have.

EVALUATION

To evaluate our approach, we selected as case study an adaptation of Astana National Library, from BIG architects. The shape and concept of the building place it in a category of those that clearly benefit from an algorithmic-based approach to design. Modeling such a composition by hand in either a CAD or a BIM tool would take considerable amounts of time and effort. Furthermore, manually handling changes that might need to be introduced in further stages of modeling is a largely time-consuming task that can be drastically reduced when using AD.

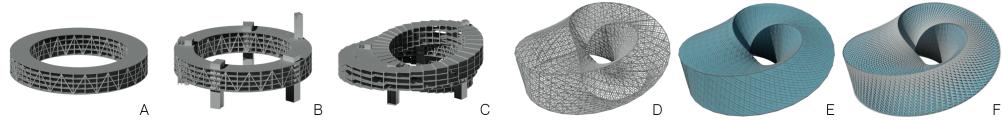
For the evaluation of our methodology we performed all the steps we have previously described, building our model parametrically. This means it is possible to change several aspects of the design, from the building's height, width, and diameter, to the number of floors, slabs, columns, beams, and façade panels, therefore generating other interesting variations of the original Astana Library.

Modeling For CAD

The modelling processes used for our case study can be divided in two main sections: interior blocks and exterior façade. These two elements are relatively autonomous from each other and could be developed in parallel. These elements were conceived in phases, each one increasing the detail of the model and respecting the order of interdependencies that characterize construction and therefore, BIM modelling. This sequential arrangement not only ensures a correct placement of elements throughout the project but also assures the automatic propagation of changes all through the model when the code is modified. CAD applications do not necessarily oblige the architect to have this sort of concerns, but good programming practices dictate that we should not only consider the order in which the elements are created, but also the use of intermediate abstractions that organize the code in a logic and understandable way.

We began by modeling the interior elements. Primarily, we modeled slabs, columns, beams, and

Figure 2
Six of the steps
within Astana's
generation
sequence



walls, all of which could be modelled using BIM operations available for CAD back-ends. Figure 2.A shows the slabs of the middle volume with columns and crossbeams already placed. In 2.B we see the inclusion of the core walls holding the buildings weight. 2.C has the outer volume that circles around the middle one, in and out.

The façade structural framing consisted of a twisting grid of beams, as can be seen in Figure 2.D. Within that framing, triangular panels were added, representing the façade glass (Figure 2.E). Finally, the façade's third layer was done: a composition of triangular photovoltaic panels, secured over the framing, shading the interior, as shown in Figure 2.F. Both panel types were modeled using Rosetta's panel operation, that in CAD back-ends generates a surface and in BIM back-ends, a morph.

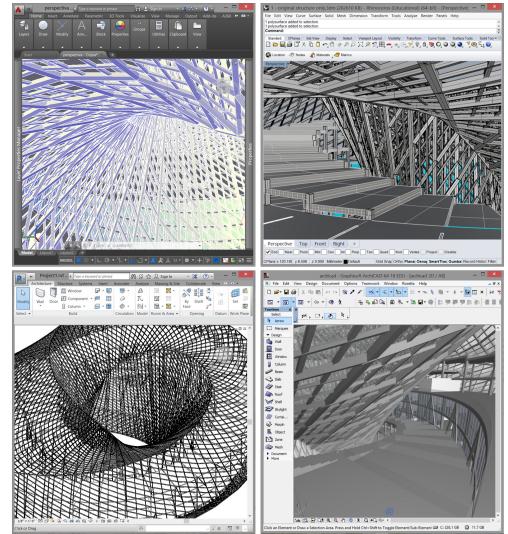
Figure 3
Astana's 3D model
generated in
AutoCAD,
Rhinoceros, Revit
and ArchiCAD

Detail Modelling in BIM

After acquiring the overall shape with the essential structural elements, we completed the full transitioning to BIM, as the remaining elements left to model would greatly benefit from pre-existent objects available in BIM libraries. As previously mentioned, the basic BIM operations, can also be interpreted by CAD back-ends thanks to the Rosetta's abstraction layer. This means that our script was, at this stage, capable of generating equivalent geometries in both CAD and BIM applications, namely AutoCAD, Rhinoceros, ArchiCAD and Revit (Figure 3).

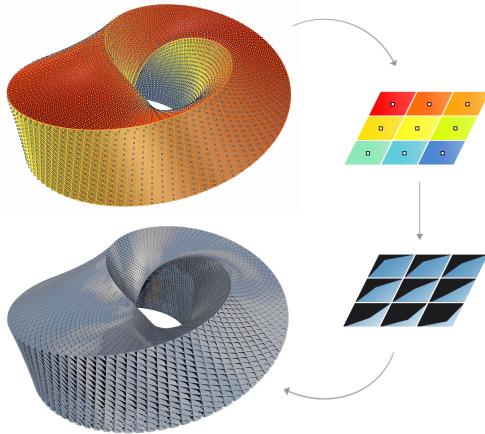
From this point on, we began modeling building elements of a more detailed nature, using tool-specific operations which meant a loss of portability. Library parts or families are exclusive to each BIM application, and while Rosetta was able to find matching characteristics in the most common structural elements that allowed the implementation of common operations in the abstraction layer, it does

not go so far as to allow the portability of every existing element in both libraries (ArchiCAD and Revit back-ends). Hence, when using Rosetta as the programming environment for integrated algorithmic-based design, users must choose one back-end to program for, when delving into detail, and at the expense of portability, they will then be able to use all objects available in the BIM library of that back-end. It is to note that the model can always be generated in other back-ends at any given moment, only tool-specific operations will not work for all.



In Astana's case, we opted for ArchiCAD, the most recent of Rosetta's back-ends. Primarily, we modeled the staircases and the glass wall for the interior volumes. The BIM operations used were the stairs and the curtain wall. However, as we were still conducting some experiments in CAD, regarding the position of these elements, we had to model them in the CAD

paradigm as well. The task ended up serving as proof to the advantages of using BIM objects.



To model the staircases parametrically we had to compute the generation of each step separately and no attention was paid to railing or finishing layers. In comparison, the stair command for BIM is capable of generating the staircase as a single set and it gives automatic access to all these details. The curtain-wall, that in ArchiCAD presents a parameterized framing besides the glass panels, was reduced to single surfaces representing the glass. Other elements such as doors and elevators were modeled in BIM only for all the detail they entail.

Incorporating Analysis

As mentioned above, an arrangement of triangular panels forms the exterior layer of the façade. These panels are in fact photovoltaic lattices that, not only provide passive shading, but also absorb energy from the sun. Due to the wrapping and twisting of the façade geometry, the thermal imprint has a wide range of intensities along the Möbius strip. Hence, the architects decided to base the design of these triangles on the façade's thermal map, creating an ecological pattern grounded on solar impact.

For a similar effect, we set Rosetta to perform a daylight analysis on the building's façade, using the Radiance back-end. Rosetta automatically places sensor nodes on the façade wrap for each panel and performs the simulation, in this case a Radiation Map metric on the winter solstice. The resulting radiation values for each node are then returned to our script. There, they are converted to a one to nine scale values and integrated in the panel modeling function. Each panel can have one of nine possible dimensions, creating larger to smaller openings in the façade pattern. Figure 4 presents a scheme of this conversion.

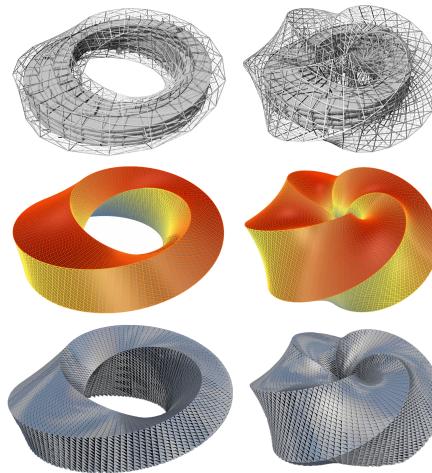


Figure 4
Scheme of radiation values converted to panel sizes in Rosetta

Figure 5
Structure, analysis model, and render of the final panels of variations one (on the left) and two (on the right)

The algorithmic integration of analysis tools offers the architects the possibility to analyze the building's shape at any given time of the design process with no effort at all, allowing for a more informed development of the form. The information exchange system between the design script and the analysis tool, allowed by the algorithmic approach, is even capable of running alone, when stimulated by an optimization algorithm. Regarding Astana National Library, that was not the architects' intent, as they merely pursued a direct use of the analysis information. For this reason, we did not pursue that path. However, the possibility is left open by our proposed approach.

Furthermore, we did take advantage of this feature to analyse a couple of variants to the original model, obtained by changing only a few parameters that define the shape of the mobius strip and the number of panels in the façade. Figure 5 presents two variations we tested, namely their structure, façade analysis and final result for the panel placing.

Render Detail

For this task, we focused on the central volume and we algorithmically filled the space with objects that we felt would naturally decorate a library space. We used several of ArchiCAD's pre-modeled objects, namely book-shelves, book clusters, tables and chairs, people and lights. Most of these objects presented variable parameters, which we modified for Astana, in order to better suit the library's environment. Figure 6 presents a view of the furnished space.

In the case of the objects representing people, we introduced randomness factors that created variations, regarding their clothes and complexions. All elements were parametrically distributed, tending to the shape of the spaces. This means that if the number of floors, or their dimensions are altered, all the mentioned elements will adjust accordingly.



Figure 6
Render of the
libraries interior
space furnished.
View of the spiral
volume

CONCLUSION

This paper presents a design methodology that explores a single-script approach, that includes not only the model's description but also the necessary

steps for its analysis. Using the proposed methodology, the architect may choose to model his design using the CAD or BIM paradigms at any given stage of the design process. Moreover, he can perform analysis of his design directly from the script and incorporate the results in the algorithmic description. This integrated algorithmic approach to design mitigates the current CAD / BIM / analysis tools portability issues that practitioners face today.

Related Work

Besides Rosetta, the chosen tool to evaluate the methodology, there exist other programming tools that allow portable AD. Grasshopper is one of them. This graphical algorithm editor, integrated with Rhino's 3-D modeling tool, offers several plug-ins that connect the program to different BIM and analysis tools. For instance: Lyrebird connects it to Revit; Hummingbird allows the creation of native Revit objects; Rhino-Grasshopper-ArchiCAD makes the connection to Graphisoft's BIM tool; DIVA-Grasshopper allows the setting up of analysis in DIVA's plug-in for Rhinoceros in Grasshopper; LadyBug allows the user to import weather data files from EnergyPlus into Grasshopper; and HonneyBee connects Grasshopper to EnergyPlus, Radiance, Daysim and OpenStudio for building energy and daylighting simulation.

Grasshopper is a visual programming languages. This means the user needs no prior knowledge of programming or scripting in order to use it, which is very attractive for architects with little programming experience. However, visual languages lack scalability (Leitão and Santos, 2011). As programs grow in complexity, they become harder to change and understand. Furthermore, the portability asset of Grasshopper's plug-ins has some limitations. While the majority of the program may be common to the various possible models, in order to connect it to each software the user must use specific components offered by each plug-in. This means that significant parts of the program are not portable, and many operations end up being repeated in order to generate the geometry in different software.

Future Work

In the future, we plan to include another feature to Rosetta: the possibility to model project specific elements, when no object provided by the BIM library serves the design purpose. The use of pre-modeled elements should be an advantage to the modelling process but never a limitation to creativity. This asset aims to give the user more freedom, for we believe the architect should not have to feel compelled to change the design, only to adapt to the exiting objects. In order to allow this in our ArchiCAD backend, for instance, we may develop a GDL converter for Rosetta.

It would also be interesting to extend the number of analysis back-ends, namely structural analysis tools, and animation or rendering software as well. The render production we presented serves its purpose well in automating the task in non-final stages of the project where the ambience required is a generic one. However, for later stages of the project where more detail designing wall coatings, flooring, light spots and lamp choices, specifying furniture models, etc, the approach does not seem as fit. Some BIM tools have good rendering engines, but none compare to specialized rendering software.

ACKNOWLEDGMENTS

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, Dalmaso, 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, pp. 69-78
- Brandon, L and McLain-Kark, J 2001, 'Effects of Hand-Drawing and CAD Techniques on Design Development: A Comparison of Design Merit Ratings', *Journal of Interior Design*, 27(2), pp. 26-34
- Eastman, C, Teicholz, P, Sacks, R and Liston, K 2008, *BIM Handbook: A Guide to building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, John Wiley & Sons, Inc., Hoboken, New Jersey
- Ferreira, B and Leitão, A 2015 'Generative Design for Building Information Modeling', *eCAADe (Education and Research in Computer Aided Architectural Design in Europe) and Faculty of Architecture and Urban Planning*, Vienna, pp. 635-644
- Kalay, Y 2004, *Architecture's New Media: Principles, Theories and Methods of Computer-Aided Design*, The MIT Press, Massachusetts
- Kensek, K and Noble, D 2014, *Building Information Modeling: BIM in Current and Future Practice*, John Wiley & Sons, Inc., Hoboken, New Jersey and Canada
- Leitão, A and Lopes, J 2011 'Portable Generative Design for CAD Applications', *ACADIA 2011: Integration through Computation: Proceedings of the 31st annual conference of the Association for Computer Aided Design in Architecture (ACADIA)*, Banff, pp. 196-203
- Leitão, A and Santos, L 2011 'Programming Languages for Generative Design: Visual or Textual?', *eCAADe (Education and Research in Computer Aided Architectural Design in Europe)*, Ljubljana, pp. 549-557
- Llach, D 2013, 'Algorithmic Tectonics: How Cold War Era Research Shaped Our Imagination of Design', *Architectural Design - Computation Works: The Building of Algorithmic Thought*, 222, pp. 16-21
- Miller, N 2011 'The Hangzhou Tennis Center: A Case Study in Integrated Parametric Design', *ACADIA Regional 2011: Parametricism*, Lincoln, Nebraska, USA, pp. 141-148
- Nguyen, AT, Reiter, S and Rigo, P 2014, 'A review on simulation-based optimization methods applied to building performance analysis', *Applied Energy*, 113, pp. 1043-1058
- Oxman, R and Oxman, R 2014, *Theories of the Digital in Architecture*, Routledge, Abingdon and New York
- Sheil, B 2008, 'Protoarchitecture: Between the Analogue and the Digital', *Architectural Design - Protoarchitecture: Analogue and Digital Hybrids*, 78, pp. 6-11
- Terzidis, K 2006, *Algorithmic Architecture*, Architectural Press, Oxon and New York
- Zboinska, M 2015, 'Hybrid CAD/E platform supporting exploratory architectural design', *Computer-Aided Design*, 59, p. 64-84