Integration of an algorithmic BIM approach in a traditional architecture studio

Inês Caetano *, António Leitão
INESC-ID/Instituto Superior Técnico, University of Lisbon, R. Alves Redol 9, 1000-029 Lisboa, Portugal

A B S T R A C T

Algorithmic BIM (A-BIM) is a design paradigm that merges the potentialities of both Algorithmic Design (AD) and Building Information Modelling (BIM). This paper describes how the A-BIM approach was integrated into the design workflow of two traditional design studios, to develop a set of parametric facades for a residential building, from which we automatically extracted material quantities and construction details.

This work demonstrates how the combination of AD with BIM influenced the whole design process and the selection of the final solution. The limitations found during the entire process are also addressed, such as tight deadlines and financial constraints. Finally, the pros and cons of using an A-BIM process compared to a traditional BIM approach are discussed, as is the implementation of this paradigm in a traditional design practice.

We also show how the efficiency of the A-BIM process can be greatly increased by the use of an Integrated Development Environment for AD supporting the generation of 3D models in both Computer-Aided Design (CAD) and BIM applications.

1. Introduction

The Building Information Modelling (BIM) paradigm awoke from the idea of modelling a building as an assembly of components. Even though some architects claimed that the BIM idea was highly limiting because it reversed their natural order of designing (Chase, 2005), its use has grown throughout the years because its advantages outweigh the disadvantages. This increasing use of BIM technologies has been changing the architectural practice, complementing Computer-Aided Design (CAD) tools, and even replacing them at certain design levels. BIM allows architects to instil the buildings’ components of their 3D models not only with geometrical information, but also with the corresponding constructive information. This results in a 3D model that includes the building’s construction information, which is important for the collaboration among different specialities, the analysis and optimization processes during the design stage, and the construction stage (Asl, Bergin, Menter, & Yan, 2014).

On the other hand, Algorithmic Design (AD), a design approach that allows the creation of shapes through algorithms (Terzidis, 2003), has been empowering architects’ creativity by allowing the generation of several solutions in a short period of time. Moreover, AD also enables the fabrication of highly complex projects that would be very difficult and time-consuming to produce manually (Kolarevic, 2003). Even though most AD tools were initially developed for CAD environments, there are already some extensions that allow the integration of an AD approach inside BIM tools, therefore combining the potentialities of both approaches (Humppi & Osterlund, 2016) and allowing the emergence of a new design approach: Algorithmic BIM (A-BIM) (Feist, Barreto, & Ferreira, 2016).

Several large-scale studios have already incorporated AD practices within their workflow (De Kestelier & Peters, 2013), whereas with small-scale design studios the reality is not the same. Nevertheless, it is still possible for the latter to benefit from the advantages of AD in productivity, cost/time reduction, experimental freedom, among others (Santos, Lopes, & Leitão, 2012). In this perspective, this paper focuses on the integration of AD into the design process of a traditional, BIM-based design studio. We present a methodology for facade design using an AD approach for BIM (A-BIM) and its evaluation in a real case scenario. The work developed results from a collaborative process between our team...
of specialists in AD and two Portuguese studios (Atelier dos Remédios and FOR-A Architects) with no previous experience in AD or A-BIM.

2. Algorithmic BIM

Due to the changeable nature of architectural design (Bukhari, 2011), architects require a design process that supports change. Unfortunately, when using traditional CAD tools, too much time and effort is required to modify designs and models, limiting the exploration of more intricate solutions. Nevertheless, recent technologies already allow the development and proliferation of more complex design solutions, new patterns, and advanced fabrication methods (Kolarevic, 2003). This includes AD (Terzidis, 2003), which has been changing the way architecture is practiced (Imbert et al., 2012; Heijden, Levelle, & Riese, 2015). AD requires the designer to develop an intermediate step between the idea and the design, in which he produces the algorithmic description of the intended design (Leitão, 2013). When executed by a computer, the developed algorithm creates the actual design.

By combining AD with a BIM-based approach to design, new paradigms, such as Algorithmic-Aided BIM (Humppi & Österlund, 2016) and Algorithmic-Based BIM (Feist et al., 2016), emerge. There are already some tools available that support such approaches: Dynamo (Autodesk, 2016), Generative-Components (BentleySystems, 2003); RosettaBIM (Feist et al., 2016; Ferreira & Leitão, 2015; Lopes & Leitão, 2011); RhinoBIM (VirtualBuildTechnologiesLLC, 2015), Hummingbird (Guttmann & Meador, 2012), and Lyrebird (LMNArchitects, 2014) are examples of tools that allow the exploration of AD approaches within BIM environments.

3. Rationalization processes

When using AD approaches, the architect benefits from a larger design freedom in his creative process but, unfortunately, he is then limited by the physical constraints usually connected with the structural performance and fabrication of the obtained solution (Mesnil, Douthé, Baverel, Léger, & Caron, 2015). Therefore, a current concern of designers when using such approaches is the manufacturing process of the developed solutions, especially when dealing with more complex geometries or patterns.

By resorting to geometric optimization processes (Mesnil et al., 2015), the task of constructing unusual solutions can be simplified. This includes rationalization processes, which imply the adjustment of a geometrically complex design towards a feasible and affordable way of production (Deng et al., 2015). As an example, when facing the manufacturing of a large-scale free-form surface, the designed surface can be rationalized by resorting to a process named panelling (Andrade, Harada, & Shimada, 2017; Eigensatz, Deuss et al., 2010; Eigensatz, Kilian et al., 2010; Flöry & Pottmann, 2010; Fu & Cohen-or, 2010; Son, Fitriani, Kim, Go, & Kim, 2017), which is the segmentation of the surface into simpler elements that can be manufactured at a reasonable cost, while preserving the design intent (Eigensatz, Kilian et al., 2010).

Another possible strategy to deal with the fabrication of complex geometries and patterns is to discretize the obtained design by applying an algorithm to control the range of variation of the parts composing the solution (Dritsas, 2012). This enables the architect to balance the solution’s visual outcome and its fabrication cost, a process that promotes the search for a design solution that fits the original design intent and, simultaneously, is more viable in terms of cost and material waste. Even though most of the related work focused on the optimization of regular polygonal free-form meshes (triangular, quadrangular, and hexagonal meshes) (Eigensatz, Deuss et al., 2010; Fu & Cohen-or, 2010; Pottmann & Liu, 2007; Pottmann, Schiftner, & Wallner, 2008; Singh & Triangle, 2010; Son et al., 2017), other geometries or types of facade panels have also begun to be considered (Jiang, 2015; Mesnil, Douthé, Baverel, & Léger, 2016; Pottmann et al., 2010). In this work, we explore a discretization strategy of a set of facade elements whose geometry was more peculiar.

4. Design project

The work developed in this paper is part of an architectural project of two Portuguese design studios: Atelier dos Remédios and FOR-A Architects. Initially, the aim of the design studios was to create a facade design for a residential building in Lisbon that ensured different degrees of permeability and, simultaneously, communicated the idea of randomness. However, the way this conceptual idea was going to be materialized was still to be defined, both in terms of its geometric pattern and its materiality. Adding to this, there was another important limitation: the design solution had to be developed in six days. Given this scenario, which required both design flexibility and visualization speed to explore different design options, using AD was the most obvious solution. Unfortunately, the architects had no experience with such an approach and, clearly, there was no time to teach them. Therefore, the only possibility was for us, architects with AD skills, to collaborate with them as specialized service providers.

As the deadline was very short and the building design was already advanced when we were contacted, lacking only the parametric facades and related dependencies, the collaboration proceeded as follows: the design studios were responsible for designing the still-missing parts of the project, and we were responsible for developing the algorithmic facades. Moreover, the building’s model was done in Revit, which meant the final model of the algorithmic facades had to be delivered in the same format. Therefore, we decided to develop the building’s facades using an A-BIM approach in which AD was advantageous for the geometric exploration stage, providing both design flexibility and visualization speed, whereas BIM allowed to directly integrate the obtained model into the building’s main model.

The project consisted in a residential building for the urban area of Belém, in Lisbon. The intervention lot had a pre-existence that had to be maintained (Fig. 1A), which was then incorporated into the building’s main facade, the one facing the main street. The building had a U shaped plan to create a central courtyard that provided natural light and pleasant views to all dwellings (Fig. 1B).

The proposed challenge was the development of three algorithmic facades for the residential building (Fig. 1C). The architects’ idea was to create an architectural envelope with various levels of perforation depending on the function of the interior space associated. When the facade coincided with the dwellings’ terraces it should be more permeable, whereas when the facade corresponded to more private areas, i.e. bedrooms and kitchens, it should be less permeable or totally opaque. This concept of permeability-opacity was left open at an initial phase, allowing us to explore a range of different solutions. After obtaining a set of design options and presenting them to the design studios, the architects could evaluate the proposed solutions, suggesting improvements to be then implemented. This iterative interaction between our team and the design studios resulted in a dynamic design process with direct participation and feedback from both parties.

5. Methodology

The most critical issue in this project was the time limit. Given that we had only six days until completion, we decided to divide
the process in three two-days-long stages: one for materializing the architects’ conceptual idea and present different design alternatives; another for improving and detailing the chosen design solution; a final one for preparing the solution for its subsequent fabrication.

Regarding the first stage, we started by implementing an algorithm that addressed the architects’ idea. Afterwards, new parameters and relations between parameters were added to the algorithm to produce different facade patterns. Since there were no constrains regarding materiality at this stage, the developed solutions took advantage of various materials, namely bricks, tiles, and concrete panels, which allowed the exploration of different strategies to address the concept of permeability. Thanks to the use of AD, two days were sufficient for us to explore multiple solutions with diverse geometries and materials.

After analysing the range of proposed solutions, the design studios asked us to merge the features of two of the presented options: (1) the placement of bricks creating random protrusions and occasional voids, i.e., brick absences (Fig. 2A), and (2) the use of different sized bricks (Fig. 2B). Already with the aim of achieving a higher control over the feasibility and cost of the final solution, they also suggested to rationalize the facade design by reducing the degree of variation of the pattern. Since the designs initially presented (Fig. 2A and B) were composed by elements that created different sized protrusions, the architects decided to reduce this range of variation to only two possible protrusions. Then, they suggested decreasing the number of brick sizes to only two possible dimensions: a small brick (30 × 5 × 10 cm) and a large brick (60 × 5 × 20 cm) with four times the size of the small one. Lastly, they decided the facade voids should only be created by the absence of small bricks, and never of large bricks. In general, the choice between placing one large brick or four small bricks was randomly controlled, as was the creation of voids and the placement of protruded bricks.

In sum, the design studios’ intent was to maintain the initial concept of randomness and irregularity but using more restricted design variables. Fig. 2C is a conceptual representation of the pattern achieved after the iterative process of implementing their suggestions and feedback. It is important to note that this is the type of solution that few architects are willing to produce manually, especially with the existing time constraints.

Comparing with a traditional modelling approach, one of the advantages of using AD is the ability to implement changes more rapidly during the design exploration process, allowing us to iteratively improve the results with much less effort and time. In this case, the architects could instantaneously visualize the resulting models right after the implementation of their feedback. At an initial stage, this allowed to accelerate the search for an acceptable design solution regarding the architects’ intent, by narrowing the design space of the options being developed. Similarly, it also allowed the application of increasingly small design changes at more advanced design stages, enabling a step-by-step improvement of the solution until the architects’ goals were completely reached.

The following stage was the generation of a more detailed facade model adapted to the building’s dimension and to the areas with different permeability degrees. Fig. 3 presents some of the small variations applied to the facade pattern, which included (1) the balance between the placement of a large brick or a set of four small bricks, (2) the percentage of protruded bricks, and (3) the degree of the facade’s permeability in certain areas, i.e. the percentage of absent bricks.

Finally, given we were using an AD tool that supports the generation of equivalent 3D models in both CAD and BIM applications, it was trivial to change the generation target and produce the set of parametric facades directly in Revit. This allowed us to easily incorporate and correctly place the facades in the BIM model of the whole project.

6. CAD/BIM portability

The work presented here was developed using Rosetta, an Integrated Development Environment for AD that supports scripts using different programming languages, such as Python and AutoLisp, and allows the generation of 3D models in a set of CAD and BIM applications, such as AutoCAD, Rhino, and Revit (Lopes & Leitão, 2011).

6.1. Programming for CAD and BIM

One of the advantages of Rosetta comes from its emphasis on portability between CAD and BIM tools. To achieve this portability, the tool provides a set of operations that express the meaning of the objects to create but implements these operations differently, i.e., depending on the design tool being used.

As an example, the operation wall creates a wall-object in a BIM tool, containing both the geometric information and associated semantics. However, for a CAD tool it creates a box with merely
Fig. 2. A and B are two of the design ideas developed at an initial design stage. C shows the obtained facade pattern at a final stage.

Fig. 3. Design variations: A - Percentage of four small bricks. B - Percentage of absent bricks. C - Protrusions' depth size. D - Percentage of protruded bricks.
Similarly, when working in a BIM tool, the window operation inserts a certain BIM-family window in a wall, otherwise, it resorts to Boolean operations to, first, open the area occupied by the window and, then, add its purely geometric elements. This results in an AD program that is more explicit about the design, as it literally mentions the building’s elements being created (walls, windows, slabs, etc.), while being able to quickly generate efficient visual representations in a CAD tool. In practice, for an algorithmic designer, it is easier to describe a design using these higher-level operations than using the more primitive geometric operations, such as boxes, unions, and subtractions.

In sum, programming with A-BIM forces a paradigm shift to be made right from the start, which requires taking into consideration the issues mentioned previously, i.e., making use of the operations specifically designed to handle both CAD and BIM paradigms. Henceforth, the algorithmic development of A-BIM solutions turns out to be as linear as that of AD solutions.

### 6.2. A-BIM workflow

As the time available for the development of the facades was only six days, each of the design stages described previously had to be very short.

Given that CAD tools usually have better performance than BIM tools, we took advantage of Rosetta’s portability to test our algorithms using AutoCAD at an initial stage. This allowed us to generate and visualize a wider range of design variations, which was critical, given the time available for the development of the final project. After each design iteration, the architects analysed the solution obtained and, in return, provided feedback, thus improving the following iterations. This resulted in an iterative process of generation-visualization-regeneration that facilitated the achievement of a design solution better suiting the architects’ design intent, while also meeting the project and site constraints. Finally, when a design solution that satisfied the design studio was reached, we used the same algorithms to directly generate an identical model in the BIM tool they used, without any extra effort (Fig. 4) and avoiding spending time redoing the model from scratch.

The resulting BIM model had the corresponding BIM families, construction information and correct dimensions. This allowed the architects to directly incorporate the set of algorithmic facades in the 3D model containing the whole building (Fig. 5).

Fig. 6A is a conceptual representation of the portability-based workflow. This workflow was important to speed up the whole exploration process as Rosetta’s AutoCAD backend was more than 80× faster than the Revit backend. Although recent developments in Rosetta’s BIM backends have dramatically reduced this difference to around 4×, this is still a large enough difference that justifies the presented workflow. Even though for a single iteration the time difference was only 27 s, it is expected that this difference will increase when dealing with more complex or larger models. Furthermore, in case multiple design variations are tested, the time difference between tools becomes increasingly noticeable (Fig. 6B).

In sum, since the higher the number of design iterations, the more satisfactory the solution becomes, if we used a BIM tool to generate and visualize the design solutions being developed, each iteration would consume more time, therefore reducing the time available for evaluating other design variations. Moreover, as the transition between CAD and BIM environments added almost no time to the whole process, taking advantage of a CAD tool for the design exploration stage was even more advantageous.

### 7. Manufacturing stage

The next stage dealt with the construction of the developed facade solution. As the budget for this project was relatively small, and the availability of advanced manufacturing processes was limited, it was important to balance the different manufacturing strategies in order to solve the existing constructional issues. The bricks used in this facade had a non-conventional size, which meant they had to be pre-fabricated. With the aim of reducing the production cost of these customized bricks, the architects had already limited, at the design exploration stage, the number of brick sizes to only two. Nevertheless, it was important to further focus on the feasibility of the facade solution.

#### 7.1. Brick fabrication

One of the greatest challenges found at this stage was the placement of the bricks, since several characteristics of the pattern were controlled by random factors: (1) the decision between placing a
large brick or four small bricks; (2) the selection between two different protrusion sizes for the placement of each brick; (3) the location of the facade voids.

One of the first limitations found was the depth size of the bricks, which was too narrow to obtain the necessary stability when stacked. To overcome this situation, the architects decided to double this dimension.

Another problem was the existence of small bricks immediately above a facade void, which meant these bricks would have no support. A possible solution was regenerating the facade models with an additional constraint dictating that only large bricks could be placed on top of facade voids. However, this would decrease the degree of variation of the pattern. Another possible strategy was to fabricate sets of three and four small bricks as single units.

Together with the design studio, we decided to follow the second strategy. Still, it created another challenge, which was how to fabricate the smallest number of different moulds, while maintaining the facade pattern's complexity and randomness. As the cost of the mould fabrication usually exceeds the panel cost, it was important to reduce the number of required moulds and increase their reusability. This would, therefore, decrease the overall cost and material waste of the manufacturing process (Eigensatz, Deuss et al., 2010).

After studying this possibility, we realized that the back face of each set of small bricks corresponded to its front pattern inverted, which meant that we could horizontally and vertically rotate them and, as a result, obtain different pattern configurations. Since the same mould could produce up to four brick configurations, we were able to reduce the number of required moulds. Fig. 7 shows some of the typologies created for each configuration of small bricks together with the corresponding possible variations.

7.2. Brick positioning

Even though the manufacturing of the different brick typologies was solved, their placement on site was still an issue. At this phase, two of the main concerns were the complexity derived from the facade pattern and the need to correctly place the bricks on site by the builders, since the task of positioning each brick typology would be challenging and error-prone.

As the whole facade model was created through AD, it was possible to easily obtain a list with the position of each brick type along the facade. By taking advantage of this information, we could create a scheme of the facade pattern, in which each brick typology was filled with a different colour (see Fig. 8). This scheme aimed at
facilitating the identification of the brick typologies, and their accurate placement on site.

7.3. Brick placement

For the placement and support of the bricks on site, we decided to merge two approaches: (1) the use of metal profiles to maintain the verticality of the stacked bricks, and (2) the application of angle brackets to fix the bricks to the building’s wall. As these elements were also developed algorithmically, we could adjust some details to minimize material waste and improve the facades’ stability and ventilation.

Our solution required the creation of four small grooves on each brick (two above and two below) to then place the metal profiles. These grooves should be centred and distanced according to the protrusions’ size, allowing the placement of the bricks in two different positions: when the brick is protruded, the below and above metal profiles fit the first grooves; otherwise, they fit the second ones (Fig. 9A). This process is repeated for each row of bricks and, simultaneously, some mortar is added to glue the bricks together.

Additionally, to give stability and fix the bricks to the building’s walls, we decided to use angle brackets to hold alternating rows of bricks (Fig. 9B). The obtained solution also had the advantage of creating a thermal isolation zone between the bricks structure and the building’s wall, which is visible in Fig. 9C.

7.4. Brick tallying process

For both cost estimation and the following manufacturing stage, it is important to have the list of quantities of all the facade elements. However, due to the complexity of the design, tallying is not trivial, and it is difficult to get this information from tools such as Revit. To overcome this problem, we extended our algorithm to
include the tallying process. As an example, Fig. 10 shows the exact quantities of each brick typology composing the facade pattern.

Finally, after having (1) the 3D model of all the facade elements (the bricks’ moulds, metal profiles and angle brackets), (2) the corresponding quantities, and (3) the constructive information (position, material, and dimensions), the design studios can proceed with the fabrication of the required elements and the construction of the facade.

8. Evaluation

A-BIM can be evaluated according to different perspectives. One is related to the differences between A-BIM and the traditional design processes. Using algorithms to produce the envisioned designs is clearly the biggest difference between them, which has considerable advantages already discussed in the literature (Bukhari, 2011; De Kestelier & Peters, 2013; Kolarevic, 2003;
The disadvantage, however, is that A-BIM requires specialized programming knowledge, which is currently not yet widespread, especially in traditional design studios.

The case study presented in this paper demonstrated how the application of an A-BIM approach in a traditional architectural studio allowed for a more flexible design process that, simultaneously, suited the design studio's typical workflow. Moreover, the example also evidenced how this approach allowed to gradually improve the design and, therefore, obtain a complete facade solution in a very short amount of time.

It is important to note that, due to the tight deadline of this project (six days), a traditional design approach probably would not lead to a similar final solution. Firstly, it would be tiresome and more time consuming due to the complex nature of the design. Secondly, the obtained model would not be flexible enough to allow the iterative incorporation of design changes, thus hindering the improvements that led to the final design.

On the contrary, the A-BIM approach allowed the incremental improvement of the intermediate solutions by including the architects' feedback almost immediately and enabling the quick visualization of the improved design. Additionally, it allowed the generation of the algorithmic facade model directly in Revit, the BIM tool used by the architects. The resulting model included the corresponding BIM information, which was important for the ensuing manufacturing stage. In the end, the final solution was further developed to include additional construction elements, resulting in a model containing the algorithmic facades' construction information, e.g., the elements' dimensions, material quantities, prices, among other necessary information.

A-BIM can be considered a form of Parametric Design. However, when compared to other parametric approaches used in the past, an important difference emerges: by implementing A-BIM on top of Rosetta, we promote the use of a single script for both CAD and BIM, allowing for different levels of detail in the generated model, directly supporting lighter and faster 3D models (Holzer, 2007) when using a CAD tool or more detailed and heavyweight models when using a BIM tool. Therefore, the A-BIM practitioner can develop and test his AD in both contexts, switching between one and the other depending on his momentary needs for speed or detail.

In its current state, A-BIM promotes a functional programming approach (Leitão & Proença, 2014) but, to speed up the development process (at the expense of verifiability), its implementation uses dynamic typing (Wortmann, Tunc, & Dhabi, 2017). However, neither the use of functional programming nor dynamic typing are absolute requirements for A-BIM. The important differentiating feature is A-BIM's ability to use the same modelling operations to generate equivalent architectural models in different modelling tools.

9. Conclusion

The interconnection between architecture and computer science brought many innovations and improvements to architectural design, namely, new design approaches and methods. The BIM paradigm is an example, which awoke from the idea of modelling a building as an assembly of components. AD is a more recent example that dramatically changes the way architects approach their design ideas, which now must be decomposed into a sequence of operations and geometric interdependencies that critically relies on mathematical knowledge. This might explain why AD is not yet widespread, however, its acceptance in the community has gradually increased thanks to the advantages it brings to the architectural design process. Many architectural schools already offer courses on programming and AD strategies, which, in turn, constitutes a starting point for a new generation of architects that will have the ability to master AD processes (Chase, 2005).

Still, it was only recently that both the BIM and AD paradigms were combined in the architectural practice. Currently, the A-BIM paradigm has been little explored due to the still existing limitations of AD tools for BIM, which suffer from scalability and performance problems when projects become more complex, as well as, the limitations of the BIM tools themselves, which lack flexibility and, therefore, retrain the architect's creative process. Nevertheless, we predict that, no matter how much the AD tools for BIM are improved, for certain types of buildings, BIM itself will become an obstacle, particularly, those that have highly intricate geometries.

Considering the current scenario, we believe that, in a near future, a modelling tool especially designed for algorithmic use in a BIM environment will be needed. Once this tool is created, we anticipate that, with the technological evolution, as well as, the architects' greater experience on AD processes, the A-BIM paradigm will gain ground in the field. Firstly, it supports the desired flexibility and performance of AD in the development of realistic BIM-based design solutions. Secondly, it allows the suppression of the transition process between CAD and BIM tools, which not only saves time and effort from designers, but also protects the solution obtained from the information losses and errors that typically occur during such process. Finally, it makes it possible to incrementally improve the design solution being developed by easily incorporating iterative design changes.

In this paper, we described the design process of a set of algorithmic facades developed using A-BIM, and how this approach was integrated into the design workflow of two traditional studios. Similarly to (Wortmann et al., 2017), we demonstrated how the integration of AD together with BIM influenced the whole design process, the selection of the final facade design, and the rationalization process for the subsequent construction phase. Additionally, we addressed how we dealt with some of the limitations found during the entire process, including tight deadlines and financial constraints. Finally, we explained the pros and cons of using this design approach compared with a traditional BIM approach, and we discussed the integration of this paradigm in the traditional design practice.

This work was developed using Rosetta, a programming environment for AD that supports scripts written in different programming languages and that can generate identical models in different CAD and BIM applications. In the paper, we also demonstrated how this capability was fundamental for speeding up the entire process, allowing us to (1) quickly experiment with designs using efficient CAD tools at an initial design stage, and, when satisfied with the results, (2) switch to the BIM tool used by the studios and generate a complete BIM model of the facade without having to change the corresponding algorithmic description.

Declarations of interest

None.

Acknowledgments

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UID/CEC/50021/2013 and PTDC/ART-DAQ/31061/2017, by the PhD grant under contract of FCT with reference SFRH/BD/128628/2017, and by the PhD grant under contract of University of Lisbon (UL), Instituto Superior Técnico (IST) and the research unit Investigação e Inovação em Engenharia Civil para a Sustentabilidade (CERIS).
References


VirtualBuildTechnologiesLLC (2015). RhinoBIM.


Inês Caetano is a Portuguese architect graduated at Instituto Superior Técnico University of Lisbon), a researcher at INESC-ID, and a PhD student at the same university, exploring the integration of algorithmic design methods in the design, analysis and optimization of facade designs.

António Leitão has a B.Sc in Mechanical Engineering, a M.Sc. in Electronics Engineering, and a PhD in Computer Science and Engineering, all from Instituto Superior Técnico IST) of the University of Lisbon. Currently he is Assistant Professor at the same university, Scientific Coordinator of the Software Engineering Group at INESC-ID, and Coordinator of the Architecture and Computation Group, teaching, lecturing, and researching on bringing together the fields of Computer Science and Architecture.