



A-BIM: Algorithmic-based Building Information Modelling

Sofia Teixeira de Vasconcelos Feist

Thesis to obtain the Master of Science Degree in

Architecture

Supervisor: Prof. Dr. António Paulo Teles de Menezes Correia Leitão

Examination Committee

Chairperson: Prof. Francisco Manuel Caldeira Pinto Teixeira Bastos

Supervisor: Prof. Dr. António Paulo Teles de Menezes Correia Leitão

Members of the Committee: Prof^a Ana Paula Filipe Tomé

May 2016

1 INTRODUCTION

Affordable Computer-Aided Design (CAD) systems became available to architectural offices in the 1970s, consisting mainly of geometry-based drafting and modeling systems. Despite these tools supporting more efficient working processes in architectural design by improving drawing production and supporting continuous, digitally-driven design processes (Kolarevic, 2003), the creation and manipulation of highly complex geometries can still be a challenging task using manual means.

The introduction of programming in architecture allowed architects to efficiently conceive and explore complex geometries using algorithmic processes as active agents for form generation in the design process. In particular, Algorithmic Design (AD), a programming-based approach to design, uses algorithmic processes to generate forms and shapes.

AD introduced a new field of design exploration in architecture, allowing architects to explore a whole domain of “*unpredictable*” forms which would have been difficult to explore using manual means (Terzidis, 2003). In addition, because the design generated with AD is typically parameterized, a wide range of different solutions can be quickly generated and tested by providing different values to the parameters, thus supporting exploration and optimization in the design process (Kolarevic, 2003). Finally, AD also brought improvements to architectural design by enabling the automation of repetitive, time-consuming tasks that had to be manually executed before, thus relieving architects from tedious and error-prone work.

To take advantage of AD, several tools and programming environments were introduced to design softwares, enabling architects to develop programs that generate models in CAD applications. Nowadays, various architectural projects have been successfully completed using processes that included algorithmic design phases (see Figure 1).

Recently, however, Building Information Modelling (BIM) tools have been replacing former geometry-based CAD applications in architectural design. BIM tools introduce a new design methodology in architecture consisting on the development of an intelligent, 3D model of a building, containing all relevant data for design and construction (Eastman et. al., 2008). BIM has the potential to bring many improvements to architectural design and, for that reason, many major private and government owners all over the world have started to mandate the use of BIM in their projects as a mean to drive the migration to BIM in the building industry (Bernstein et. al., 2014).

BIM can still benefit from AD and, for that reason, like with former CAD tools, several programming tools have been recently made available to enable the use of AD with BIM applications. By combining AD with the BIM methodology, a new approach to design emerges, one that we call A-BIM, acronym for Algorithmic-based Building Information Modelling.

In this thesis, we define, explore and evaluate A-BIM in the context of architectural design. Our contributions are: (1) a programming methodology for A-BIM that overcomes the differences between programming for CAD and BIM



Figure 1 – Examples of buildings designed through algorithmic processes. Above: The National Aquatics Center. Below: Beijing’s National Stadium (source: <http://www.arup.com/>).

tools; and (2) a comparative study of three different but related design approaches through a case study, namely an algorithmic approach to CAD, the proposed A-BIM approach, and a manual BIM approach. In the end, we evaluate A-BIM in relation to the other two approaches, and we explain the benefits and drawbacks of working with A-BIM.

2 ALGORITHMIC-BASED BUILDING INFORMATION MODELLING

A-BIM is an algorithmic approach to BIM that allows the generation of BIM models through algorithms. This approach creates a paradigm shift in the design process, since the designer, instead of developing the model directly in the BIM application, develops the algorithm that generates the model in the BIM application.

Similarly to BIM, the model generated with A-BIM contains relevant information for design and construction and is constrained by parametric and associative rules. However, unlike BIM, the source of all information in the model is the algorithm that generates the model, which can be shared and developed concurrently between the different members of the design team and offers a more flexible, controllable, and integrated way of managing the project's data.

A-BIM requires the formalization of the design intent in order to construct the algorithm of the proposed design solution. This, in turn, requires programming knowledge in order to translate the algorithm into instructions that can be understood by a computer, using a programming language, so that it can execute them.

However, from a programming standpoint, A-BIM requires a different approach from the one needed for geometry-based CAD. Due to the differences between working with CAD and BIM tools, the corresponding programming methodologies will also differ.

In the following section, we describe a programming methodology that we found appropriate for BIM, while comparing it to the programming methodology needed for CAD.

2.1. From CAD to BIM

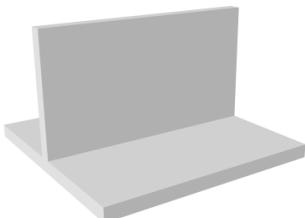


Figure 2 – A generic slab and wall.

One major difference between working with geometry-based CAD and with BIM is that a BIM tool does not just create geometry; it creates digital representations of building components containing all the semantic information and data related to that component (Eastman et. al., 2008). As an example, consider a generic slab and wall (see Figure 2): although they are geometrically similar, semantically they are two distinctly different building components. When programming for CAD, that slab and wall can be both created using the same generic geometric box operation available in every CAD tool, only with different parameters. On the other hand, when programming for BIM creating these objects requires specific operations with semantics matching each different building component.

Fortunately, to increase the legibility of programs, good programming practices already promote the use of intermediate abstractions and these abstractions help the migration from programming for CAD to programming for BIM.

For example, when programming for CAD, we typically implement different user-defined functions for each building component, namely slabs and walls. These abstractions, although useful for organising the program, do not have any additional effect on the CAD tool besides the creation of the corresponding geometric objects. However, when programming for BIM, these abstractions become, in fact, pre-defined operations and, thus, transfer the intended semantics to the generated objects.

One consequence of BIM tools dealing with building components instead of just geometry is that, in comparison with CAD tools, they are more restrictive in the manipulation of the geometry of the created objects and, as such, have limitations regarding geometric modelling operations such as Boolean operations. One reason for this is that BIM tools already handle these operations internally in the creation of certain objects: for instance, when a window is placed in a wall, the subtraction necessary to create the opening in the wall for the window is done automatically. Another, more important, reason is that BIM tools are more sensitive to what can be built or what usually makes architectonic sense. While it might be interesting to see the result of a subtraction between a wall and a stairway, architectonically speaking, it typically makes little sense to subtract a stairway from a wall.

This brings us to another difference between CAD and BIM, which is the fact that BIM building components have parametric and associative rules that dictate their behaviour in the model and provide its architectonic sense. For example, in BIM, a door can only exist hosted in a wall. This means that, to create a door, a wall must be created first. These rules are reflected in the program written for BIM where, in fact, a host wall is one of the parameters required to create the door. This was not the case with CAD, where all objects could be created separately and the order of creation of the elements was irrelevant for the final result.

Finally, another difference between CAD and BIM is the fact that BIM has libraries of pre-modelled, parametric building components, which makes the creation of certain components easier, since their geometry does not need to be created from scratch. For example, a door can be selected from a BIM library, while in CAD all of its subcomponents might need to be modelled. When that modelling is done through programming, it means that a lengthy and complex program might be required, which might take some time to produce. To overcome this disadvantage, CAD tools can import external blocks that are pre-modelled shapes or forms which can facilitate the design process of complex geometry. However, typically, these objects are not parametric as the ones available for BIM tools, restricting the designer's ability to manipulate the geometry as they see fit.

3 CASE STUDY: THE ABSOLUTE WORLD TOWERS

In order to evaluate the capabilities of A-BIM, we selected an architectural case study – the Absolute World Towers (AWTs), two residential twin towers designed by MAD Architects and located in Mississauga, Canada (see Figure 3) – which we modelled using three different but related approaches: (1) an Algorithmic approach to geometry-based CAD (A-CAD); (2) the Algorithmic-based

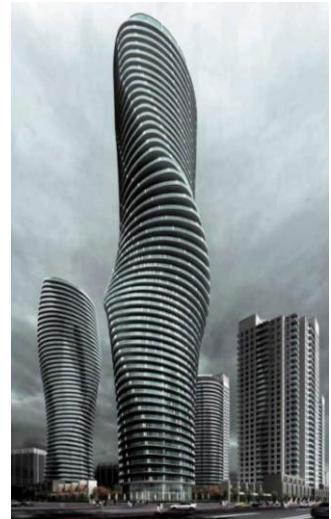


Figure 3 – The Absolute World Towers, designed by MAD Architects (source: <http://www.domusweb.it/>).

Building Information Modelling (A-BIM) approach that we propose, and (3) a manual BIM approach. The aim of this modelling process is to analyze and compare the three different approaches in order to find out the benefits and drawbacks of using A-BIM in relation to the other two.

The manual BIM approach was modelled in Revit while both A-CAD and A-BIM were implemented using Rosetta which is a portable AD tool for both CAD and BIM. This means that, by using Rosetta, we were able to develop our two programs, the one for A-CAD and the one for A-BIM, and generate the model of the AWTs in different CAD and BIM applications. As a result, we were able to test our A-CAD modelling process in both Rhinoceros and AutoCAD and our A-BIM modeling process in both Revit and ArchiCAD, thus providing a more generalized implementation of AD for both CAD and BIM.

The modelling process itself was divided into the different building components that make up the towers as the same component can be modelled in three entirely different ways depending on the approach used. Those are: the levels (while not exactly building components, levels are important 3D BIM elements), slabs, openings in the slabs, walls, roof slab, stairs, and doors.

As an example, let us consider the modelling of the slabs of the AWTs. In the following sections we explain the modelling processes of the slabs in the three aforementioned approaches and compare them in order to find out the benefits and drawbacks of using A-BIM in relation to the other two.

3.1. A-CAD

To create a slab with A-CAD, we had to model its geometry. The method used to achieve this was to define the shape of the slab and extruding the resulting surface with a given thickness (see Figure 4).

In order to generalize the form of the building, we define the shape of the slab using a list of points that, when connected, outlines its boundary (see Figure 5). This list is used to produce the surface needed to create the slab and is provided as a parameter that can be freely defined by the designer. Therefore, by experimenting with different lists with different positioning of the points, we can obtain different slab shapes and easily vary the form of the building.

Each slab is then coupled with a height from a list of floor heights, allowing the distribution of the slabs along the different floors (see Figure 6, next page). As a result, the vertical positioning and number of slabs become dependent on both the height of each floor and the number of floors in the list. For example, by changing the number of floors in the list, we can rapidly change the number of slabs created.

It is during their placement that the rotations are applied to the slabs. To that effect, like with the heights, an angle of rotation from a list of angles is applied to each slab, ensuring the desired rotation for every floor. This list of angles is also provided as a parameter that can be changed, thus allowing the experimentation of different rotations for the towers. In order to model both AWTs, we created two specific lists of angles, corresponding to the actual rotation of each tower.

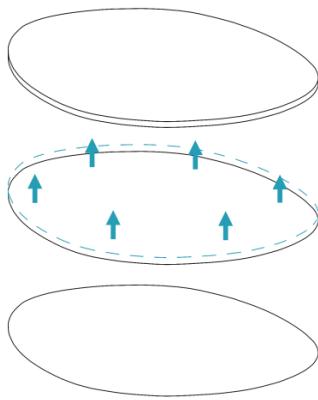


Figure 4 – Given a surface with the desired slab shape, the surface is extruded to create the slab.

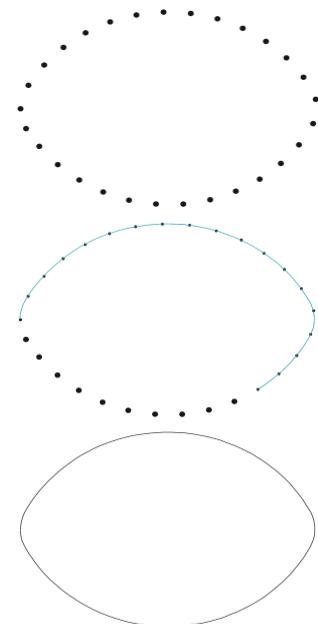


Figure 5 – The shape of the slab is defined by a list of points that outlines its boundary.

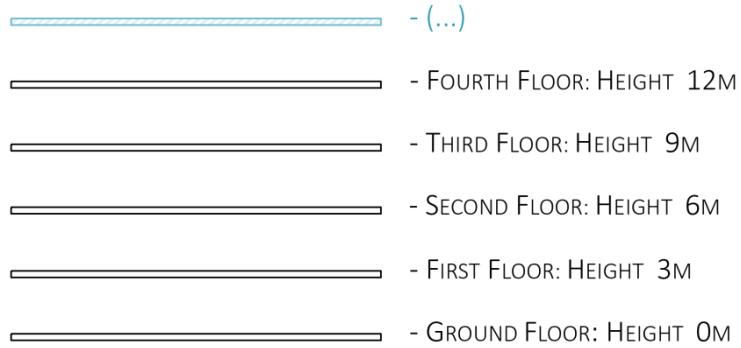


Figure 6 – The slabs are distributed along the floors by distributing them along the list of heights, thus establishing the number of slabs and their respective heights.

3.2. A-BIM

As mentioned before, BIM has libraries of pre-modelled, parametric building components that can be used and manipulated to fit a project's requirements. To take advantage of this, Rosetta provides pre-defined operations that allow the creation of these pre-modelled building components. As such, to create a slab, we used the pre-defined operation that creates slabs provided by Rosetta. This operation requires a shape for the slab, the level that the slab belongs to and the desired slab properties (e.g. material composition, thickness, etc...) as parameters.

Like with the former approach, the shape of the slab is defined by a list of points that, when connected, outlines its boundary though this time no surface is needed to create the slab. This list is also given as a parameter that can be easily changed in order to vary the form of the building.

Then, similarly to A-CAD, each slab is coupled with a level from a list of levels and an angle of rotation from a list of angles in order to distribute the slabs along the different floors and ensure the desired rotation for every slab.

3.3. Manual BIM

Before creating the slabs, we select the desired slab properties. By doing this at the beginning, we ensure that all subsequent slabs created will possess the selected properties.

To create the slab, we select the level in which we want to place it and, using the 'Floor' tool (see Figure 7), we draw the desired slab shape. In the case of the AWTs, the ellipsoidal drawing tool can be used to directly obtain the ellipsoidal shape. This creation process is then repeated – or the previously created slab is copied – for all the levels in order to create all the slabs.

For every slab created, we have to manually apply the desired angle of rotation. To create both towers, this process of rotating the slabs has to be executed twice in order to achieve both towers' rotations.

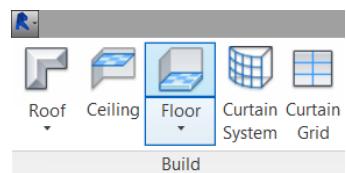


Figure 7 – The 'Floor' tool can be used to create slabs.

3.4. Analysis

By comparing the three modelling processes, we found that one advantage of using BIM (algorithmic or otherwise) over CAD is that BIM applications already know what a ‘slab’ is, both geometrically and semantically, thus facilitating the creation process and producing a building component semantically identified as a slab (or floor in the case of Revit). Furthermore, the created slab contains architectural properties and data, such as material composition, area covered, among others, while the slab created with A-CAD is a purely geometric entity.

At the same time, an algorithmic approach allows the variation of the shape of the slab, or the rotation of the tower, by simply experimenting with their respective parameter values, thus allowing the exploration of different design alternatives for the AWTs without having to redo or modify the algorithm that shapes them. As a result, we can quickly and almost effortlessly explore alternatives to the form of the building, while preserving the ability to faithfully reproduce the AWTs. For example, Figure 8 shows three different instances of the slabs of the AWTs generated by using different parameters for the number of floors, the shape of the slabs and the rotation of the tower.

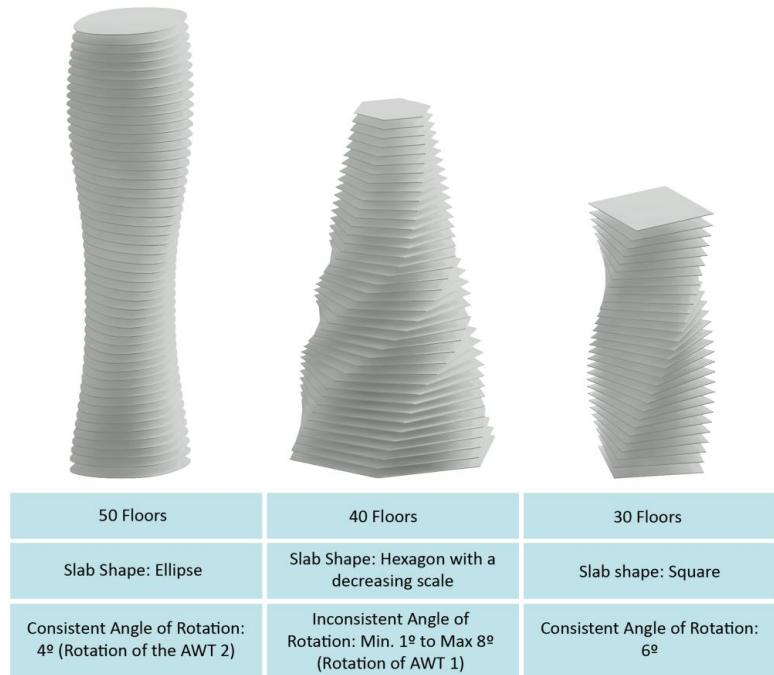


Figure 8 – Three instances of the slabs of the AWTs obtained by experimenting with the parameters of the number of floors, the shape of the slabs and the rotation of the tower.

With manual BIM, changing the slabs can mean either changing them all at once or redoing all slabs again. Due to their parametric capabilities, BIM applications have the ability to accommodate changes to a certain extent. For example, in Revit, by grouping all slabs together, we can change the shape of one slab and propagate that change to all the remaining slabs. On the other hand, changing the rotation of the tower may require manually updating all slabs to their new angle of rotation, resulting in a tedious and time-consuming process.

Finally, while with manual BIM slabs have to be created and rotated separately and manually, using an algorithmic approach affords us the ability to automate the creation of the slabs and the application of the respective rotation.

4 EVALUATION

A-BIM has a vast applicability in architectural design: it can be used to develop parametric models of parts of a building, such as a building's facade, of entire buildings, or even a whole city. However, using this approach also requires programming knowledge and an initial investment of time and effort to formulate the algorithm that generates the model which, in the end, might not be recovered. Thus, before choosing which approach to use in a project, it is essential that designers first establish their design priorities for the project at hand and evaluate if A-BIM would be beneficial for the design process of that specific project. To do that, they should consider all the potential gains and losses from working with A-BIM before deciding if these are likely to hinder or help them achieve their objectives.

Next, we analyze the gains and losses obtained from using A-BIM.

4.1. Automation of Repetitive Tasks

A-BIM, due to its algorithmic origin, enables the implementation of procedures designed to automate tedious, repetitive tasks that would have had to be manually executed otherwise, thus consuming a lot of time and effort that could be spent on more important activities. As demonstrated with the AWTs, this is very useful for buildings with a repetitive nature and, therefore, constitutes one important gain for architectural design.

4.2. Propagation of Changes

Because objects generated with A-BIM are parametrically interdependent, changes can be propagated to the entire model. While this ability to propagate changes is not exclusive to A-BIM as BIM tools themselves also offer this ability, A-BIM's ability to propagate changes is much more flexible than the one typically available in BIM tools, as explained with the slabs in section 3.4.

4.3. Exploration of a Wide Range of Design Alternatives

By experimenting with the parameters that control the parametric model of the AWTs, a wide range of design alternatives can be quickly explored and visualized without having to redo or modify the algorithm that generates them. However, this greater flexibility of design solutions also requires a greater initial effort to make the program flexible which, in turn, requires more time.

In order to test this initial effort, we simulated a design process where we explored a series of design alternatives for the AWTs in order to measure the impact that this exploration has on the project, i.e. the time and effort required for the implementation of these design alternatives for both A-BIM and the manual BIM approach. A-BIM was implemented by us with Rosetta while the manual BIM approach was implemented in Revit by an expert user.

As an example, let us consider the model of the second AWT, as seen in Figure 9. The modelling times of the tower in both approaches are presented in Table 1.

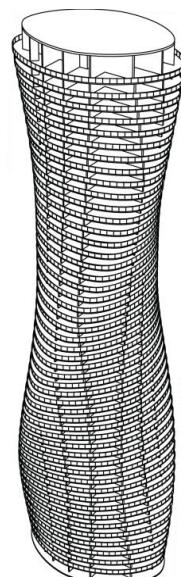


Figure 9 – Model of the second AWT.

A-BIM	MANUAL APPROACH
14h	2h 14min

Table 1 – Time required to produce the model of the second AWT for both A-BIM and the manual approach.

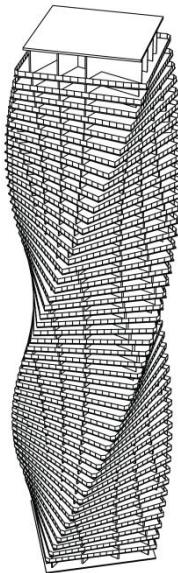


Figure 10 – Model of the tower with rectangular floors

A-BIM	MANUAL APPROACH
2min 08s	1h 25min

Table 2 – Time required to change all floors to a rectangular shape for both A-BIM and a manual approach.

In this thesis, we explore several other scenarios where changes are applied to the initial model of the AWT. We show that, as the initial model starts to evolve and frequent changes become a requirement, the manual handling of changes become a cumbersome, time-consuming task that can discourage further changes. In this regard, the initial investment required for A-BIM can be quickly recovered when extensive exploration is a requirement and/or frequent changes are anticipated. In these cases, A-BIM becomes an important paradigm that can significantly improve the design process.

4.4. Building Information

Another advantage of A-BIM is that, as a BIM model, the generated model is infused with building information and data. This building information can be consulted and extracted at any point during the design process and used to produce additional information, such as data for performance evaluations.

By combining the flexibility of A-BIM with this ability to extract information from the model, the multitude of design alternatives that can be generated and visualized can also be analyzed and compared through this additional information. For example, Table 3 shows some of the results of an energy simulation

analysis executed with Revit, for both the initial model and the variation of the tower in the scenario explained in section 4.3.

	INITIAL MODEL	TOWER VARIATION
ENERGY, CARBON AND COST SUMMARY		
Annual Energy Cost	16,371 \$	4,991 \$
Lifecycle Cost	222,970 \$	67,975 \$

Table 3 – Results of an energy simulation analysis executed with Revit for both the initial model (left) and the variation of the tower introduced in section 4.3 (right).

4.5. Optimization

By taking advantage of A-BIM's algorithmic capabilities, the process of analyzing and comparing several design alternatives can be automated in order to let the computer guide the design to better solutions. This is a topic we plan on further exploring in the future.

4.6. Parametric and Associative Capabilities

With A-BIM, the algorithm that generates the model and the BIM application where the model is generated can both benefit mutually from the inherent parametric and associative capabilities of each other. On one hand, a parameterized A-BIM program can be very flexible and accommodate the exploration of a greater solution space, as discussed in section 4.3.

On the other hand, the generated objects have to follow the established associative rules dictated by BIM tools. As an example, the BIM associative rule that dictates that a door can only exist hosted in a wall is implicitly ensured with A-BIM: the operation that creates a door requires a host wall as a mandatory parameter.

4.7. Libraries of Building Components

With A-BIM, we can also take advantage of several libraries of pre-modelled, parameterized building components available for BIM tools. Doing this greatly facilitates the modelling process of the model, especially when modelling building components containing a lot of sub-components (e.g. doors).

Unfortunately, these libraries of pre-modelled building components can also restrict what can be built in BIM tools as they mostly offer libraries of standard building components used in standard projects, i.e. frequently used design and construction solutions. An atypical construction solution (e.g. see Figure 11), might not be available in these pre-modelled libraries, thus restricting our ability to construct it. In this case, we would have to create a new building component from scratch in order to use it, thus losing the advantage of the pre-modelled libraries but regaining the ability to construct exactly what we want. Once created, we would then be able to add the new building component to the library thus enabling the reuse of the newly created customized object in future projects.



Figure 11 – Curved curtain wall in the ground floor of the AWTs (source: www.randyseitzer.com/)

4.8. Geometric Modelling

As mentioned previously, one drawback of working with A-BIM is that BIM tools are more restrictive in the manipulation of geometry compared to former geometry-based CAD tools, due to the fact that they mostly deal with building components instead of just geometry.

Despite this limitation, with A-BIM we are still able to implement procedures that can simulate the effect of geometric modelling operations.

5 CONCLUSION

A-BIM combines AD with the BIM methodology. On one hand, as an algorithmic approach to design, it offers a challenging but flexible new way of designing for architects, one that allows repetitive modelling processes to be automated and a wide range of design alternatives to be easily and quickly explored and evaluated by experimenting with different parameter values. Changes to these parameters are also propagated to the entire model, thus reducing the need to manually handle changes as is usually the case in a manual approach.

On the other hand, the model generated with A-BIM is a BIM model infused with building information and constrained by parametric and associative rules. The building components that compose the model are provided through libraries of pre-modelled, parametric components available with the BIM tools, something that can both greatly facilitate the modelling process of the model and restrict what can be built, depending on what we intend to design and what tools we use to do so.

Although A-BIM has a vast applicability in architectural design, the use of this approach might not always be the most appropriate or beneficial for a given project due to the initial investment required to formulate the algorithm that generates the model which, for some projects, might not be entirely recoverable. In this thesis, we show that the initial investment can be quickly recovered as the project evolves and frequent changes become a necessity. In this regard, A-BIM promotes design exploration and can, therefore, greatly benefit the design activity which thrives on exploration.

In the future, we will further explore the capabilities of A-BIM for the building industry, namely by evaluating this approach in a project-driven environment or by exploring optimization processes with A-BIM.

REFERENCES

- Bernstein, H. M., Jones, S. A., and Russo, M. A.** (2014) *The business value of BIM for construction in Major Global Markets: How contractors around the world are driving Innovation with Building Information Modelling*, Bedford; McGraw-Hill Construction.
- Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K.** (2008) *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, New Jersey; John Wiley & Sons, Inc.
- Kolarevic, B.** (2003) *Architecture in the Digital Age: Design and Manufacturing*, New York; Spon Press.
- Terzidis, K.** (2003) *Expressive Form: A Conceptual Approach to Computational Design*, London; Spon Press.