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: Profª Ana Paula Filipe Tomé

May 2016

Acknowledgments

I would like to express my deepest gratitude to my Supervisor, Professor António Menezes Leitão, for all the support and guidance he provided me throughout, not only this work, but this entire journey. For his availability and patience, his ideas and advices. But above all, for his invaluable friendship.

To Guilherme Barreto and Bruno Ferreira for their help and contribution, without which this work would not have been possible.

To my family for their unwavering support and for always pushing me to better myself.

To Carmo Cardoso and Teresa Neto for their constant help and support throughout this journey.

To the remaining members of GAC, as well as all that were not mentioned but present.

Finally, this work was partially supported by national funds through the Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013, and by the Rosetta project under contract PTDC/ATP-AQI/5224/2012.

ABSTRACT

Keywords: Computer-Aided Design, Building Information Modelling, Algorithmic Design, Programming, Algorithmic-based Building Information Modelling.

Algorithmic Design (AD) is a programming-based approach to design where, instead of creating a model of the intended design, the designer creates a program that generates the model of the intended design. This approach provides many opportunities for innovation and improvement in the design process. However, despite having been extensively explored with geometry-based Computer-Aided Design (CAD) tools over the past years, AD has only recently started to be explored with the Building Information Modelling (BIM) methodology brought by BIM tools. BIM brings substancial gains to the design activity and, as a result, has started to become mandatory in architectural practices all over the world. By combining algorithmic processes with the BIM methodology, a new approach to design emerges, one that we designate Algorithmic-based Building Information Modelling (A-BIM).

In this thesis, we define, explore and evaluate A-BIM in the context of architectural design. Through a case study, we compare A-BIM to two other design approaches, namely an algorithmic approach to geometry-based CAD and a manual BIM approach, and we show that A-BIM can offer great benefits to architectural practices.

RESUMO

Palavras-chave: Design Assistido por Computadores, Building Information Modelling, Design Algorítmico, Programação, Algorithmic-based Building Information Modelling.

O Design Algorítmico (DA) é uma abordagem de design baseada em programação. Ao invés de ser criado o modelo do design pretendido, é criado o programa que gera o modelo do design pretendido. Esta abordagem potencia a inovação e a melhoria do processo de design. No entanto e apesar de ter vindo a ser amplamente explorada ao longo dos últimos anos com as ferramentas de Design Assistido por Computadores (DAC) baseadas em geometria, só recentemente é que o DA começou a ser explorado com a metodologia Building Information Modelling (BIM) trazida pelas ferramentas BIM. O BIM apresenta vantagens substanciais para a actividade de design e, como tal, começa a tornar-se obrigatório na prática arquitectónica em todo o mundo. Uma nova abordagem de design surge ao combinar processos algorítmicos com a metodologia BIM. A esta abordagem designamos Algorithmic-based Building Information Modelling (A-BIM).

Nesta tese, definimos, exploramos e avaliamos o A-BIM no contexto arquitectónico. Através de um caso de estudo, comparamos o A-BIM a duas outras abordagens, uma abordagem algorítmica às ferramentas DAC baseadas em geometria e uma abordagem BIM manual, e demonstramos que o A-BIM pode oferecer grandes vantagens à prática arquitectónica.

Table of Contents

LIST OF FIGURES

List of Tables

Abbreviations

2D – Two-dimensional

3D – Three-dimensional

A-BIM – Algorithmic-based Building Information Modelling

A-CAD – Algorithmic approach to geometry-based Computer-Aided Design

AWTs – Absolute World Towers

AD – Algorithmic Design

AEC – Arquitecture, Engineering and Construction

BIM – Building Information Modelling

CAD – Computer-Aided Design

Glossary of Terms

Algorithm – A step-by-step description of a solution that solves a given problem.

Algorithmic Design – An approach to design which allows the generation of forms and shapes using algorithms.

Algorithmic-based Building Information Modelling – An approach to Building Information Modelling which allows the generation of BIM models through algorithms.

Building Information Modelling – An approach to design and construction where a 3D virtual model of a building is constructed, containing all relevant data needed to digitally simulate the entire process of building prior to the actual construction.

Parameters/Variables – A property of a program that, for different values, produces different results.

Program – An unambiguous, well-defined, formal description of an algorithm. An algorithm written in a way that the computer understands, i.e. in a programming language, with specific and rigorous intructions that tell the computer what specific steps to perform.

Programming – The act of translating algorithms into instructions that can be understood by the computer, using a programming language.

Introduction

Along with the proliferation of affordable personal computers in the 1970s, a generation of Computer-Aided Design (CAD) systems became available to most architectural offices, consisting mainly of geometry-based drafting and modeling systems. Drafting systems in particular – in combination with improvements in communication and sharing among members of the design team brought about by the internet – improved the production of drawings in architecture and, consequently, the efficiency of the design activity which, nowadays, relies heavily on drawing documentation (Kalay, 2004).

However, where CAD systems proved to have the most appeal for the architectural culture was in their ability to support the design and – in combination with Computer-Aided Manufacturing technologies – construction of buildings with highly complex geometries in a timely and cost-effective manner (Mitchell, 1999), using continuous, digitally-driven design processes. The digitally-enabled, complex, highly curvilinear forms produced from the 1990s onwards (see Figure 1) allowed architects to break away from the norm and experiment with creative and novel forms in architecture (Kolarevic, 2003).

Figure 1 – Computer-Aided Design facilitated the modelling and construction of the Guggenheim Museum in Bilbao (source: http://www.guggenheim-bilbao.es/).

Despite the support provided by CAD systems for the creation of complex geometries, the manual exploration of these geometries can still be a challenging task. The introduction of programming in architecture allowed architects to conceive and efficiently explore complex geometries using algorithmic processes as active agents for form generation in the design process. 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. This relieved architects from tedious and error-prone work, allowing them to save a lot of time and effort during the design process.

To take advantage of AD, several tools and programming environments were introduced to design softwares commonly used by architects, as is the case of Grasshopper for Rhinoceros 3D, shown in Figure 2, or Visual LISP for AutoCAD. These tools enabled architects with basic programming skills to develop programs that generate models in CAD applications. Nowadays, various architectural projects have been successfully completed using processes that included algorithmic design phases, such as the National Aquatics Center (Water Cube) and the Beijing National Stadium, both presented in Figure 3.

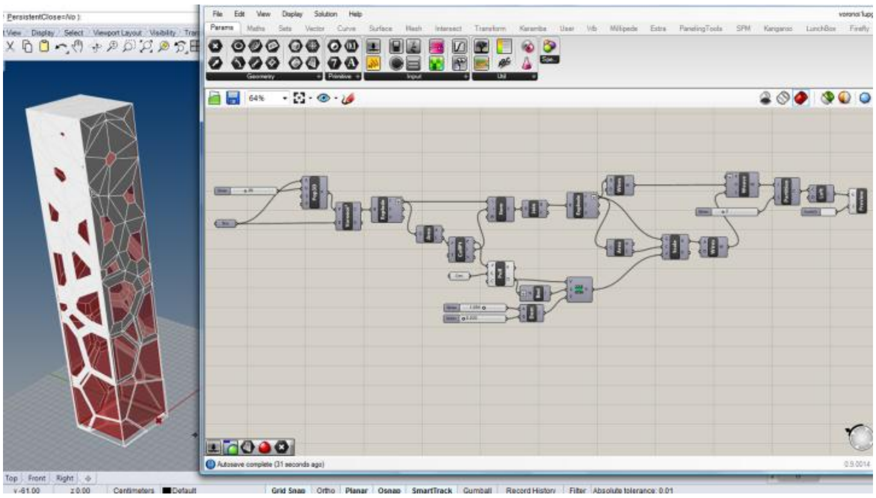


Figure 2 – Grasshopper for Rhinoceros 3D (source: http://www.grasshopper3d.com/).

Figure 3 – Examples of buildings designed through algorithmic processes. On the left: The National Aquatics Center. On the right: Beijing's National Stadium (source: http:/www.arup.com/).

Recently, however, Building Information Modelling (BIM) tools have been replacing former geometry-based CAD applications in architectural design. CAD and BIM tools are very different, and using the latter involves some significant shifts in design methodologies. For example, while CAD tools mostly deal with geometry, BIM tools produce digital representations of building components, containing both geometrical information and data attributes (Eastman et. al., 2008).

BIM has the potential to bring many improvements to the Architecture, Engineering and Construction (AEC) industries and, as a result, 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).

This approach can still benefit from AD and, for that reason, like with former CAD tools, several programming environments and tools have been recently made available to enable the use of AD with BIM applications (Feist et. al., 2016). The result is a new approach to design that combines AD with the BIM methodology and which we designate A-BIM, acronym for Algorithmic-based Building Information Modelling. Although some architects have already started to explore A-BIM in architectural practices over the recent years, this approach has yet to be properly addressed.

In this thesis, we propose A-BIM as an algorithmic approach to BIM that can offer great benefits to architectural design.

Due to the difference between BIM and former geometry-based CAD technology, A-BIM requires a different programming approach from the one needed for CAD. Thus, we will provide a programming methodology for A-BIM, adapted to the BIM paradigm.

Afterwards, in order to evaluate A-BIM, we selected an architectural case study which we modelled using three different but related approaches: (1) an Algorithmic approach to geometry-based CAD; (2) an Algorithmic-based Building Information Modelling approach with the programming methodology that we propose; and (3) a manual BIM approach. These three approaches are then analysed and compared in order to find the benefits and drawbacks of A-BIM.

Figure 4 – The Absolute World Towers, designed by MAD Architects (source: http://www.e-architect.co.uk/).

The case study in question is a pair of skyscrapers, more specifically the Absolute World Towers designed by MAD architects (see Figure 4), selected due to the benefits that they can extract from A-BIM.

OBJECTIVES

The aim of this thesis is to explore and evaluate A-BIM in the context of architectural design.

In this thesis, we identify the differences between the CAD and BIM paradigms and we propose a programming methodology fit for BIM.

As an algorithmic approach to design, A-BIM 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 always prove to be more efficient than simply producing the model manually. Thus, in order to evaluate A-BIM, we modelled our case study, the Absolute World Towers, using three different but related approaches: (1) an Algorithmic approach to geometry-based CAD; (2) an Algorithmic-based Building Information Modelling approach with the programming methodology that we propose; and (3) a manual BIM approach. This modelling process allowed us to:

1 Implement the proposed programming methodology;

2 Analyze and compare the three resulting modelling processes in order to find out the benefits and drawbacks of A-BIM when directly compared to the other two;

3 Evaluate if the initial investment can be recovered through the benefits extracted from an A-BIM approach.

METHODOLOGY

The methodology we followed is divided into four main phases: (1) Literature Review, (2) Introduction to A-BIM (3) Modelling Process of the Case Study, and (4) Evaluation and Conclusions.

The first phase of this thesis, the Literature Review, is based on the review of an extensive bibliography to discuss the impact of digital technologies in architecture over the recent years. In particular, we focus on two important developments brought by digital technologies for architectural design: Algorithmic Design and Building Information Modelling. Both these developments are explained and contextualized in architectural practices. Lastly, we examine how these developments are being combined in architectural practices, enabled by available AD tools for BIM.

In the second phase of this thesis, we introduce Algorithmic-based Building Information Modelling, and define a programming methodology for it based on the differences between programming for geometry-based CAD tools and programming for BIM tools.

The third phase starts with the introduction of the case study and the modelling process undertaken. The case study is then modelled in the three chosen approaches based on an analysis of publicly published architectural drawings, images, renderings, photos and textual descriptions of the existing buildings. Afterwards, a comparative study of the modelling process of the case study in the three chosen approaches is given in order to find the benefits and drawbacks of A-BIM.

Finally, in the last phase of this thesis, we evaluate the gains and losses obtained from using A-BIM in relation to the other two approaches analyzed. In the end, we conclude our work with final considerations and the expectations for future work.

STRUCTURE

This thesis is divided into two main parts:

I. Background;

II. Algorithmic-based Building Information Modelling.

The Introduction, Conclusion and Bibliography sections were added to these two parts.

The first part, the **Background**, is divided into 4 chapters:

1 Digital Technology in Architecture

In this chapter, we discuss the evolution of digital design tools and their impact in architecture.

2 Algorithmic Design

Here, Algorithmic Design is explained, while identifying the benefits and challenges that this approach poses to designers in architectural design. After that, we discuss how this approach is being used in recent architectural practices.

3 Building Information Modelling

In this chapter, the Building Information Modelling design methodology is explained and the advantages that this approach brings to architectural design are identified. Afterwards, we describe the evolution of BIM technology and later contextualize the state of BIM implementation in architectural practices.

4 Algorithmic Design for BIM

In the last chapter of the first part, we demonstrate how AD is being used with BIM in architectural practices with the help of two case studies – the Aviva Stadium and the Louvre Museum of Abu Dhabi – and introduce a few of the available tools responsible for enabling the use of AD with BIM in the first place.

The second part of this thesis, titled **Algorithmic-based Building Information Modelling**, is divided into 5 chapters:

5 Introduction

In the first chapter of the second part, we reintroduce the motivation of this thesis in order to contextualize our work in the second part.

6 Algorithmic-based Building Information Modelling

In this chapter, we introduce A-BIM and propose a programming methodology adapted to the BIM paradigm, while comparing it to the programming methodology needed for CAD.

7 Case Study: Absolute World Towers

Here, we give a brief overview of the case study – the Absolute World Towers – and explain the purpose and specifications of the modelling processes untaken.

8 Comparative Study: Modelling of the Absolute World Towers

In this chapter, we provide a comparative study of the modelling process of the Absolute World Towers in the three different approaches, divided by building components. For each building component, the three different modelling processes are explained and analyzed in order to find the benefits and drawbacks of A-BIM in relation to the other two.

9 Evaluation

Finally, in the last chapter of the second part of this thesis, we provide an analysis of the gains and losses obtained from using A-BIM and evaluate A-BIM's overall performance.

I

Background

1

Digital Technology in Architecture

"Digital technology is a force that has begun to profoundly affect design culture. Being digital has begun to affect the way we represent, present, communicate about, and materialize our designs by integrating media in the conceptualization, realization, communication, and production of designs. It has become ubiquitous in virtually all design disciplines, and even while it continues to develop and evolve rapidly, it is promoting a paradigm shift in the definition and practice of the design disciplines."

– Oxman, 2006a: 1-2

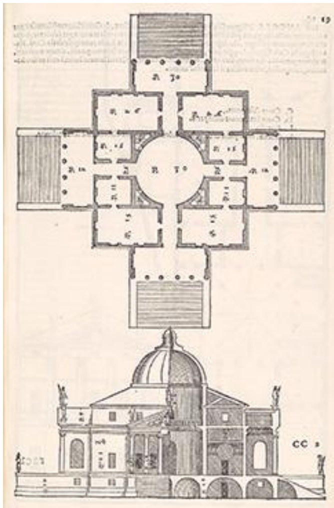


Figure 1.01 – Architectural drawings of Palladio's Villa Rotonda (source: http://www.metmuseum.org/)

Architecture as we know it today was born in the 1450s when Renaissance architect Leon Battista Alberti proposed the differentiation between the intellectual task of design and the craft of construction (Eastman, et. al., 2008). Whereas before buildings were erected with little to no prior planning and architects were essentially craftsmen, Alberti proposed that buildings should be planned in their entirety before their construction and architects should become the authors in charge of designing them. Consequently, drawings (see Figure 1.01) became important tools for design, allowing architects to plan and experiment with different design alternatives before committing them to stone, as well as for communicating their designs with builders and their clients (Kalay, 2004). Thus, Architecture as *"an art of design"* (Carpo, 2014: 8) was born.

With the establishment of architectural design as a profession, drawings became the predominant means of representation, communication and exploration in architectural design. As a result, the process of design became centred on the representational, where visual representations of the design, usually drawings, are manipulated by visual reasoning through a succession of stages, generally in the medium of sketching (Oxman, 2012). This notion of design as a process of *"reflection supported by representational processes"* (Oxman, 2008: 101) has remained unchallenged until the 1950s, i.e. until the advent of digital technologies.

Throughout history, architectural design has always been influenced by the evolution of Science and Technology, which serve as enablers for innovation and exploration in architecture and construction (Mitchell, 2007). For instance, in the Industrial Revolution, the emergence of new systems and materials, such as glass and steel, transcended previous constraints in construction, allowing larger, higher and more complex buildings with long-span structures as well as providing mechanically and electrically serviced interiors (Mitchell, 1999).

Digital technologies are no exception. The emergence of digital design tools and digitally-driven processes of fabrication have enabled architects to push past previous constraints and explore new formal territories in Architecture. But more than that, digital technologies have been challenging traditional conventions in architecture and stimulating the emergence of *new forms of designing*.

In the next sections, we discuss the evolution of digital design tools and their impact in architecture.

1.1 EVOLUTION OF DIGITAL DESIGN TOOLS

Computers were first used in the AEC industries around the 1960s as problem solving tools (Mitchell, 1995). Their ability to solve various intellectual problems, such as calculations of structural behaviour, proved useful for engineering analyses but offered a limited payoff for architecture.

The first time computers were used to help solve architectural design problems was in 1963 when Ivan Sutherland introduced the Sketchpad program – the first interactive design tool. Sketchpad (see Figure 1.02) demonstrated that computers could, in fact, be used for drafting and modelling as well as integrating the evolving design with analysis programs (Kalay, 2004). This tool was the first of many computer systems that followed, developed with the purpose of aiding design practices. These systems later became known as Computer-Aided Design (CAD) systems.

Figure 1.02 – The Sketchpad program, developed by Ivan Sutherland, allowed the user to draw on a screen of a especially modified computer system with a light pen and then copy master drawings into many duplicates (source: http://architizer.com/).

Yehuda Kalay (2004) distinguishes between three generations of CAD systems. The first generation of CAD systems was centred on architectural objects (e.g. doors, windows, slabs, walls ...) and was designed to help solve design problems from an intuitive, architectural point of view. OXSYS (see Figure 1.03) and CEDAR (1970) are two examples of this first generation of CAD systems, developed to aid in the design of hospitals and post offices, respectively, using industrialized building components and modular coordination. These first generation systems required large and powerful computers as well as specialized equipment to operate them, which made them expensive. As a result, their reach in the architectural community was limited.

Figure 1.03 – The OXSYS hospital design system. (source: Richens, 1997).

The **second** generation of CAD systems, according to Kalay, was introduced along with the personal computer in the late 1970s, which made CAD systems affordable to a larger architectural community. These computers, although cheaper and graphically better, were less powerful than their predecessors. As a result, CAD systems were also less capable than the ones of the former generation: they lost their analytical capabilities as well as the focus on building-specific objects. In fact, this generation was more concerned with aiding the development of representations of buildings instead of directly supporting the design activity. Though inferior, these systems provided a relatively easy and inexpensive practical introduction to CAD for many small architectural and engineering firms as well as schools of architecture (Mitchell, 1990), and marked the popular diffusion of CAD systems in architecture.

Figure 1.04 – A detail drawing produced with AutoCAD 1985. (source: Flanagan, 2015).

These systems consisted mainly of geometry-based drafting and modelling systems. Drafting systems were the first to be introduced and allowed designers to more easily edit drawings without the need of manually deleting and redrawing all representations of the design, thus making drawing production easier. In combination with improvements made in communication and sharing among members of the design team brought about by the internet later in the 1990s, these systems helped improve the efficiency of the architectural design activity. AutoCAD, released in 1982 by Autodesk, is a well-known example of a 2D drafting system, currently the most widely used (see Figure 1.04).

The modelling systems were introduced to personal computers in the late 1980s and, with them, the possibility of producing photo-realistic renderings (Fernandes, 2013). Interestingly, because these systems were based on abstract geometry, they could create highly complex 3D models with ease, therefore opening the possibilities of exploring with new complex shapes and forms that would have been difficult to visualize and represent with a bi-dimensional approach. In combination with Computer-Aided Manufacturing technology introduced to the building industry in the 1990s, the created 3D models could then be used to drive numerically controlled fabrication and assembly machinery, thus allowing these complex, curvilinear geometries to not only be visualized and accurately represented but also built in a timely and cost-effective manner (Mitchell, 1999), using continuous, digitally-driven design processes (Kolarevic, 2003).

The introduction of Splines and NURBS (Non-Uniform Rational B-Splines), i.e. mathematical definition of curves and curved surfaces, in these modelling systems was one important catalyst to the wild, curvilinear, “blobby” forms that started appearing in architecture in the latter half of the 1990s (Scheurer, 2014). The Villa NURBS, shown in Figure 1.05, is one example of a building designed with NURBS.

Figure 1.05 – Villa NURBS, designed by Enric Ruiz-Geli, (source: http://www.ruiz-geli.com/)

Another important step was the introduction of programming to these modelling systems in the beginning of 2000, as it popularized algorithmic approaches in architecture, such as Algorithmic Design, and allowed architects to efficiently explore and manipulate these complex geometries, something which would have been difficult to do using manual means. Algorithmic Design offered an entirely new way of designing which opened a new field of design exploration in architecture. This approach will be explained in further detail in chapter 2.

Finally, the third generation of CAD systems according to Kalay, like the first one, consists of object-oriented systems designed to support the design activity. Among these systems are Building Information Modelling (BIM) systems which prompted the development of an entirely new design methodology based on parametric, object-oriented 3D modelling and information databases. These tools and the resulting methodology will be explained in further detail in chapter 3.

2

ALGORITHMIC DESIGN

"Computer programming is becoming an increasingly valuable tool for [digital design specialists]. The design logic for a project can be used as the basis of a computer program that can rapidly generate many options and large numbers of elements."

– PETERS AND DEKESTELLIER, 2006: 12

Over the last half of a century, the AEC industries have seen many important developments, one of the most remarkable ones being Algorithmic Design.

Algorithmic Design (AD), also commonly referred to as Generative Design, is a programming-based approach to design, allowing the generation of forms and shapes using algorithms (Garber, 2014).

AD is a disruptive paradigm where, instead of creating a model of the intended design, the designer creates a program that generates the model of the intended design. By offering a new way of designing, this approach introduces a whole domain of design exploration in architecture (Kolarevic, 2003).

In the next sections, AD is explained, while identifying its possible benefits and challenges to designers, and its use in recent architectural practices is discussed.

2.1 DESIGNING ALGORITHMICALLY

In order to understand AD, it is useful to first understand what an algorithm is. Terzidis (2003: 67) defines an algorithm as *"a computational procedure for addressing a problem in a finite number of steps"*. In other words, it is a step-by-step description of a solution to address a given problem. Thus, an algorithmic approach to design is the encapsulation of design intent in procedural terms (Wojtkiewicz, 2014).

AD requires the use of algorithms, or algorithmic processes, as part of the design process which can pose a few new challenges for designers. Firstly, in order to execute algorithms in a computer, designers must learn how to program, i.e. translate algorithms into instructions that can be understood by the computer (Kalay, 2004). While designers are used to dealing with ambiguous and ill-defined design problems, designing algorithmically requires them to formulate an unambiguous, well-defined, formal description of the intended design solution and translate it into instructions that can be understood by the computer, using a programming language. This solution must be syntactically and semantically correct in the chosen programming language otherwise the program will not execute properly (Scheurer, 2014).

Secondly, they must understand mathematics, particularly geometry. Understanding the mathematics behind form creation and manipulation can offer new insight into design possibilities and increase the designer's control over the design. Mathematics can also be exploited as generative procedures.

Lastly, designing algorithmically requires one to think algorithmically. This requires designers to abstract themselves from the direct activity of design and the familiar visual, interactive representation of the design and focus on the logic that binds the design together and the textual instructions that describe the design (Woodbury, 2014). In other words, the designer no longer directly manipulates visual representations of the design in conventional tools but formulates an algorithmic description of the design.

Overall, designing algorithmically requires a different form of thinking than the one designers are used to, one based on intuition and ingenuity. This shift into algorithmic logic can be a barrier for most designers but once past this initial challenge, a new domain opens up for exploration.

2.1.1 Benefits of designing algorithmically

As mentioned before, AD allows the generation of forms and shapes through algorithms. In particular, a category of algorithms aimed at producing unpredictable results quickly triggered designers' interest, allowing them to explore new uncharted formal territories in architecture. Shape grammars, mathematical models (see also Figure 2.01), topological properties, genetic systems, mappings, and morphisms are a few examples of algorithmic processes explored for their unpredictability (Terzidis, 2003).

Figure 2.01 – A project developed by Michael Hansmeyer exploring subdivision processes to define and embellish Doric columns and obtain new column orders. Subdivision processes are also an example of algorithmic processes that can produce very unpredictable results. (source: http://www.michael-hansmeyer.com/)

Moreover, due to its algorithmic origin, the design generated with AD is usually parameterized so that different but related instances of the same design solution can be quickly generated by experimenting with different parameters values, thus allowing designers to visualize and explore a wide range of design possibilities (see Figure 2.02). Because the generated design is parametric – i.e. exploits associative geometry to describe relationships between objects, thus establishing interdependencies between them (Oxman, 2006b) – changes made to the algorithm that generates the design or its parameters are propagated so that the designer no longer has to manually update all aspects of the design.

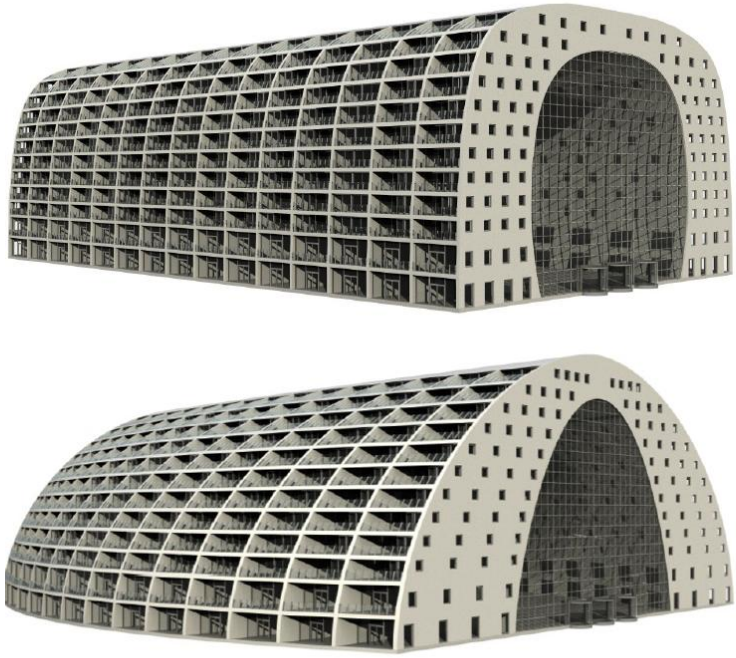


Figure 2.02 – The same algorithm can produce different instances of a design by attributing different values to its parameters. In this case, the same algorithm was used to generate different design alternatives for the Market Hall in Rotterdam, designed by MVRDV Architects (source: Fernandes, 2013).

Figure 2.03 – Designed by Foster & Partners, the London City Hall's round and slightly tilted form was obtained through an optimization process: the surface in contact with direct sunlight was minimized, resulting in reduced solar gains through the building's skin. (source: http://www.theneweconomy.com/)

By combining this flexibility of AD with analysis and simulation softwares, design alternatives can then be analyzed and compared with relative simplicity to select a solution that offers optimal performance (Kolarevic, 2003). This allows the designer to prioritize performance early in the design process, or even let it lead the process, as described with the case of the London City Hall in Figure 2.03, and presents a massive shift from traditional design methodologies, where performance evaluations are typically done at the end of the process, making it rarely a priority (Kalay, 2004). These optimization procedures are not only restricted to technical aspects of the design performance such as structure, thermal behaviour, acoustics, and aerodynamics; they can also include other aspects such as material usage, spatial distribution, among others.

Finally, AD also enables the automation of repetitive, time-consuming tasks that had to be manually executed before, such as repetitive modelling or fabrication processes. This relieves architects from tedious and error-prone work, allowing them to save a lot of time and effort during the design process.

2.2 Algorithmic Design in Architectural Practices

The use of algorithms in architecture is not a recent phenomenon and can, in fact, be traced back to long before computers existed, where algorithms were used to prescribe and externalize design processes or goals in order to achieve designs with certain attributes (Fernandes, 2013). However, the recent introduction of programming environments and languages to design softwares encouraged architects to start using algorithmic processes as active agents for form generation in the design process.

Mark Burry (2014) identifies two main motivators for architects to start scripting: productivity and design exploration. Productivity refers to the use of scripting as a productivity tool, i.e. as a mean to automate mundane, tedious and repetitive processes. On the other hand, design exploration refers to the use of scripting as a medium for research-based experimental design. According to a survey realized by Burry, both motivators were equally important in encouraging designers back in the beginning of 2000 to delve into the then unfamiliar territory of AD.

Nowadays, digital and programming knowledge have become basic skills required from young designers. Most universities already have programming and digital tools as a part of their architectural curricula and many have already created new specialization degrees of programming and computation specifically for architects (see example in Figure 2.04). The result is a new generation of digitally savvy professionals, both architects and programmers, which has been forming an interesting new area of expertise between the two fields. Scheible and Dimcic (2011: 7) describe this area as *"occupied by experts with the knowledge in design, programming and static analysis, that can solve the problems of structural design and fabrication arising from the complex geometry of the free form architectural design"*.

Figure 2.04 – The Master of Science programme Integrative Technologies & Architectural Design Research is an example of a specialization degree in the University of Stuttgart. (source: http://icd.uni-stuttgart.de/)

At the same time, in the labour market, employers are increasingly looking for professionals with multiple skills or fields of expertise. Specifically, some architectural firms have begun putting out job descriptions for architectural positions where programming/scripting skills (if not a degree in Computer Science) are required, or at least desired, along with the usual architectural qualifications. For example, Michael Mcinturf ARCHITECTS recently had a job position available for *Junior/Intermediate Architects* where, among others, a *"familiarity with basic Grasshopper or other scripting/programming languages"* was listed as a desired attribute. Another example is the Woods Bagot architectural firm which had been looking for *Architectural Designers with Coding Skills* for their SUPERSPACE design research group where a degree in Computation was required along with a degree in Architecture or Urban Design.

The latter constitutes an example of a growing trend in the AEC industries where many leading architectural and structural engineering firms have begun forming their own internal multidisciplinary research units to explore digital design (Oxman and Oxman, 2014). The main goal of these units is to carry out research and development in order to find unique and innovative design solutions, commonly in a project-driven environment, using digital technologies. While these internal units were mainly established to support the firm they are associated with, some of them also offer consultancy services to outside firms. Besides SUPERSPACE, a few examples of design research groups include the Specialist Modelling Group, Frank O' Gehry and Partners and BlackBox in the field of architecture and the Advanced Geometry Unit in the field of engineering (Santos et. al., 2012).

For example, the Specialist Modelling Group (SMG), established within the architectural firm Foster and Partners in 1998 by Hugh Whitehead, consists of a small group of professionals sharing a common architectural and engineering background but with diverse specialities and interests (Whitehead, 2003). Their expertise encompasses complex geometry, environmental simulation, parametric design, computer programming, and rapid prototyping (Peters and DeKestellier, 2006). The SMG have been involved in many successful projects, which they accompanied from conceptual design to fabrication. These include the Swiss Re Headquarters in London and the *Chesa Futura* in Switzerland (see Figure 2.05).

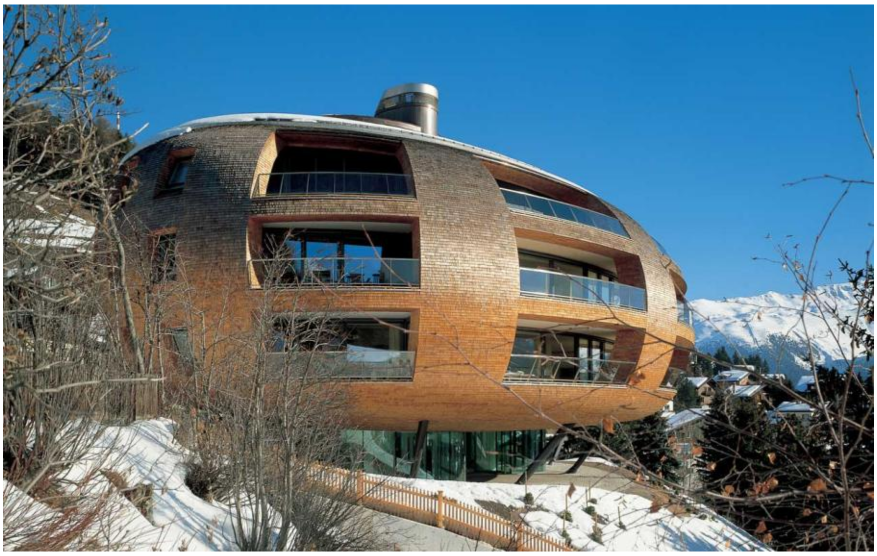


Figure 2.05 – The Chesa Futura apartment building designed by Foster & Partners with the support of the Specialist Modelling Group (source: http://www.fosterandpartners.com/)

Figure 2.06 – The CCTV Headquarters designed by Rem Koolhaas with the structural counselling of the Advanced Geometry Unit (source: https://aedesign.files.wordpress.com)

Similarly, the Advanced Geometry Unit, founded within the engineering company Arup in 2000 by Cecil Balmond, also consists of a small team of professionals comprised of engineers, architects, mathematicians, programmers, and artists. Their primary role is to research complex structural geometry to support architectural visions and solutions (Hudson, 2010) and they have also been responsible for carrying out the design and construction of many unique projects, including the 2002 Serpentine Pavilion, designed by Toyo Ito, and the CCTV Headquarters (see Figure 2.06), designed by Rem Koolhaas in Beijing, China.

Besides consultancy services offered by internal research units within larger companies, independent consultancy services are also a common model for putting programming skills into practice in Architecture and can come in the form of consultancy companies and independent consultants.

Programming Architecture, for example, is an independent company founded in 2006 by Milos Dimcic which offers consultancy services to help solve problems in the design and construction of complex architectural objects. These services involve formal and structural optimization of complex geometries as well as the automation of their fabrication processes (i.e. drawing generation and numerically controlled fabrication). Comprised of a small team of architects, engineers and programmers, Programming Architecture focuses on the development of custom software for specific projects (For more information, check the company's website at http://programmingarchitecture.com/). Among the projects developed by the company are the Bao'an International Airport in Shenzhen (see Figure 2.07) and the EXPO Axis in Shanghai.

Figure 2.07 – The Bao'an International Airport in Shenzhen designed by the architect Massimiliano Fuksas with consultancy services from Programming Architecture (source: http://structurae.net/).

Figure 2.08 – The Sagrada Família was left unfinished when the original architect Antoni Gaudí died in 1926. Since then, others have been attempting to finish it while being faithful to Gaudí's original style. (source: http://mcburry.net/).

Finally, Mark Burry is an example of an independent consultant who has been working on the unfinished design and construction of the *Sagrada Família* (see Figure 2.08) in Barcelona, as executive architect and researcher. His work involves using computational parametric tools to parameterize the geometric methods of Antoni Gaudí in order to find and explore solutions that fit Gaudí's original design intent for the *Sagrada Família* (Hudson, 2010).

3

BUILDING INFORMATION MODELLING

"We at the M.A. Mortenson Company think of [BIM] as 'an intelligent simulation of architecture.'"

– CAMPBELL, 2006: 1

Figure 3.01 – A BIM model containing all relevant data needed to digitally simulate the process of building. (source: http://national-cba.com/)

Another important and promising development which has been emerging in the AEC industries in the last half a century is Building Information Modelling (BIM). BIM is an intelligent approach to design and construction, rooted in the technological advances of digital modelling systems, where an accurate, 3D virtual model of a building (or a group of buildings) is constructed, containing geometric information as well as all relevant data needed to digitally simulate the entire process of building prior to the actual construction (Garber, 2014).

This approach has the potential to change and improve many practices of the AEC industries. It does so by offering a new design methodology which disrupts with traditional methods of representation and collaboration in architecture (Bergin, 2011), even as it enhances productivity. However, despite having been around for some time, it is still considered a recent development to the larger architectural community (Smith, 2014).

In the following sections, the BIM methodology is discussed, identifying its potential advantages, and its evolution in the building industry is explained.

3.1 THE BIM PROMISE

3.1.1 A BIM Methodology

One important, defining aspect of BIM is a shift from a 2D perspective to a 3D perspective. As mentioned before, BIM entails the construction of a virtual 3D model representing a building. However, as mentioned before, unlike former modelling systems which create only geometry (i.e. a graphic abstraction of the intended design), BIM produces intelligent, digital representations of building components containing both geometrical information and data attributes. These data attributes consist of information about what these building components are, what are they made of, how they should behave (structurally or environmentally), and how much they cost, among others (Eastman et. al., 2008).

The building components also have parametric rules that dictate their behaviour in the model. These rules establish requirements and associations between objects, such as a door that can only exist hosted in a wall, which help ensure that the digital building components behave more like their real counterparts. These associations also facilitate the designer’s job by propagating some changes in the model and all subsequent views of it, thus reducing time spent manually managing changes, as was the case with earlier CAD tools.

Using these building components, a 3D model linked to a database of building information can be created containing all data needed for the design, fabrication and construction. At any stage during the design process, this information can be extracted from the model and used for various purposes. This includes, for example, consistent and accurate 2D drawings which can be generated directly from the model, as drawings continue to be the primary legal and contractual source of building information in Architecture. However, drawings are no longer the primary means of representation and communication and drawing generation, which BIM intends to automate, no longer has such a huge weight or impact in the design process.

In essence, BIM involves a shift from a collection of 2D architectural drawings and other documents – which, only when put together, can provide an (often incomplete) perspective of the whole design solution (see Figure 3.02) – to a single parametric 3D model with an integrated database containing all the building information relevant to a project (see Figure 3.03).

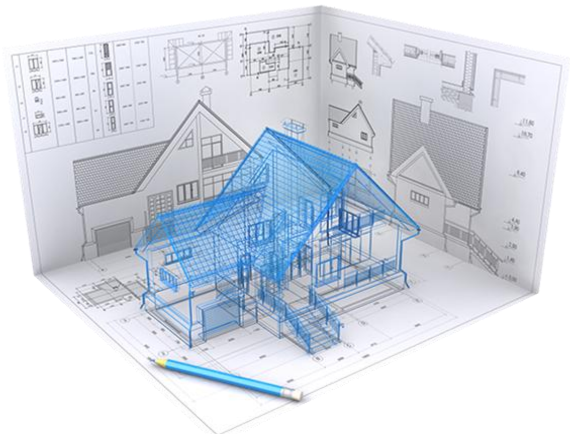


Figure 3.02 - Traditional approach (source: http://info.cadcam.org/*)*

Figure 3.03 - BIM approach (source: http://www.graphisoft.com/)

The BIM methodology is supported by a collaborative design process where all parties involved (from owners, to all members of the project team, to fabricators and subcontractors) can more easily access the digital database containing the building information – the BIM model. By working on a networked environment, information can be more easily and rapidly communicated and shared between all members of the project team, therefore ensuring that all parties are up-to-date on the latest changes to the design.

Furthermore, with changes being managed in a central shared location, errors and inconsistencies resulting from independent work from the different members of the project team can be identified and dealt with in a coordinated manner over the network.

This process relies heavily on open standards and workflows, which allows project members to participate regardless of the software tools they use (BuildingSMART, 2014). Because there are so many parties involved in a project, it is inevitable that the different design disciplines will use different, specialized softwares, tailored to their needs, which, in turn, will produce different digital file formats. As such, the existence of a common file format for sharing information is required and the Industry Foundation Classes (IFC) file format, which was developed in 1995 to allow the exchange of BIM data between different softwares, reduces these interoperability issues.

Figure 3.04 compares the traditional design process, where information circulates through the exchange of 2D drawings between individual parties, with the BIM process, centred on a shared BIM model using the IFC data model.

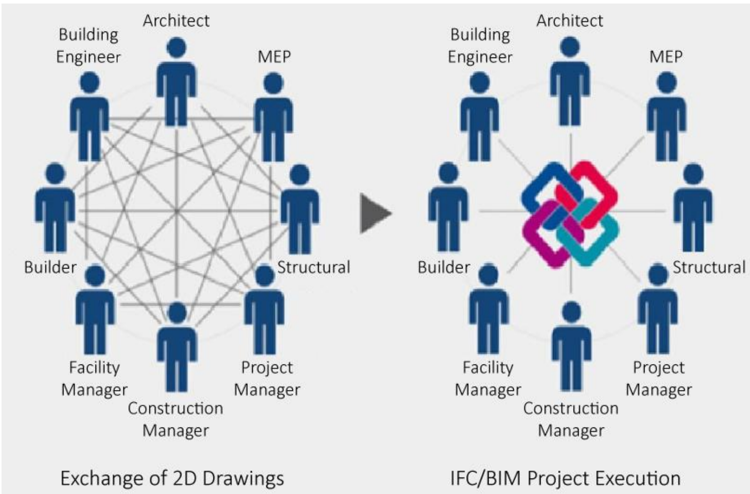


Figure 3.04 - BIM process (adapted from: http://ibimsolutions.lt/)

Finally, the use of BIM also encourages an earlier collaboration between different specialists in the design process. Because less time is spend on documentation and information is more easily shared, BIM presents an opportunity to involve other design disciplines earlier in the design process thus providing important insight into design problems. As a result, important design decisions are made at the beginning of the design process as opposed to later where changes to the design will result in added costs. Figure 3.05 illustrates the resulting workflow, as idealized by Patrick MacLeamy, and compares it to a traditional design workflow.

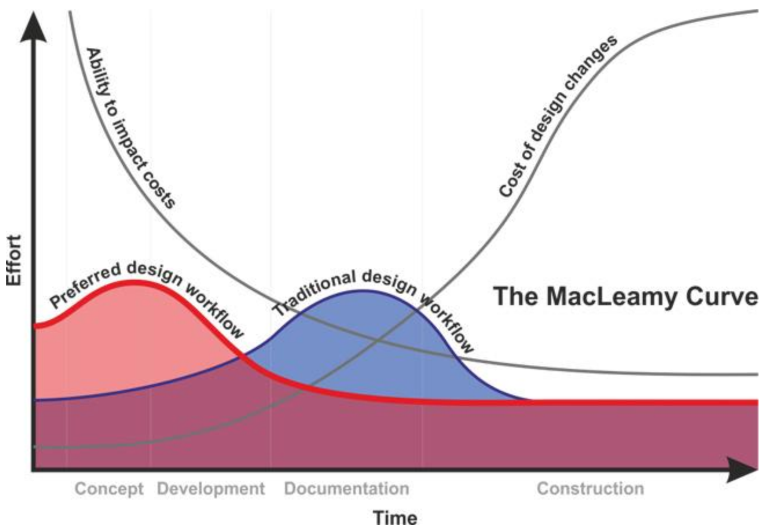


Figure 3.05 - The MacLeamy Curve of BIM design workflow (source: http://aecmag.com/)

3.1.2 Advantages of a BIM Implementation

Besides a new design methodology, BIM offers a wide spectrum of benefits ranging through the different phases of a building's lifecycle. These include the design and construction phases of a project as well as post-construction activities such as facilities management.

To begin with, there are the advantages of working with a 3D parametric model. A 3D model is inherently more comprehensive for all parties involved, offering a more unambiguous representation of the project compared to 2D drawings, which often form an incomplete perspective of the design and leave much to interpretation. Moreover, only one representation of the design is being developed at all times therefore eliminating the tedious, error-prone process of producing, modifying and updating multiple separate representations of the same design. Finally, because the model is parametric, some changes can be automatically propagated, thus reducing the need to manage changes.

Additional important advantages arise from the BIM model being infused with building information. This information can be extracted at any point during of the design process and used to produce additional information, including:

- consistent architectural drawings (e.g. plans, sections and elevations);
- renderings and animations for spatial visualizations;
- accurate representations of the building objects for fabrication and construction;
- material quantities for earlier procurement of materials from product vendors and subcontractors;
- spreadsheets of accurate bills of quantities for cost estimations;
- data for performance evaluations.

The last two are of particular importance because they allow designers to make informed decisions based on feedback in regards to costs and sustainability issues and, as such, can guide the design to a more desired or appealing solution.

Another advantage of the use of BIM is that, by working collaboratively around one or more BIM model, members of the project team can have a better access to up-to-date information, thus ensuring a better coordination between design disciplines. The resulting model can then be tested for clashes and conflicts between components, such as a pipe intersecting with a structural beam or the placement of a certain object causing a certain building regulation to be violated, and alert the participants to the problem. This allows design errors and inconsistencies to be identified before the construction phase, at a time where their correction does not yet entail significant costs or causes construction delays.

With BIM, it is also possible to plan and simulate the entire construction process before the actual construction. By doing so, the project team can visualize and manage the progress of construction activities before they occur, discovering sources of potential problems and opportunities for possible improvements (e.g. site logistics, crew and equipment, space conflicts, safety problems, among others) and adjust their plan accordingly (Eastman et. al., 2008). They will also be able to accurately schedule construction activities, ensuring just-in-time arrival of people, equipment, and materials, thus reducing costs and allowing a better collaboration at the job site.

Finally, once the building is constructed, the BIM model can then be used to support the management and operation of facilities. An updated building model provides an accurate source of information about the as-built spaces and systems and, as such, can be a useful starting point for managing and operating the building (Eastman et. al., 2008).

Overall, when implemented appropriately, BIM offers a boost in productivity during a project's execution and a reduction of costs (Taborda and Cachadinha, 2012).

3.2 Evolution of BIM Technology

Although only recently has it started to gain momentum in the architectural community worldwide, BIM is not a recent development. In fact, its concepts and ideas can be traced back to the earliest days of computing in the 1960s. However, early efforts to develop systems capitalizing on these concepts and ideas could not be realized until the introduction of a graphical interface, allowing the user to interact with the building model (Bergin, 2011).

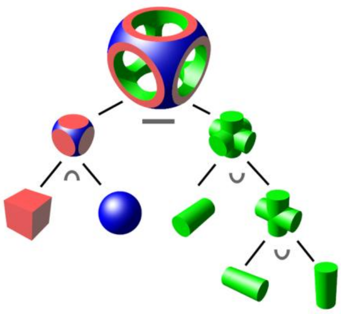


Figure 3.06 – CSG objects can be represented as a tree of operations, where leaves represent primitives, and nodes represent operations. In this figure, the nodes are labeled ∩ for intersection, ∪ for union, and — for difference. (source: https://en.wikipedia.org/)

The first generation of practical three-dimensional modelling systems, known as solid modelling, was developed in 1973 allowing the easy creation and editing of arbitrary 3D solid shapes. These systems utilized one of two different methods of displaying and recording shape information, which competed for supremacy: The Constructive Solid Geometry (CSG) method and the Boundary Representation (B-rep) method. The CSG method (see Figure 3.06) used a series of primitive shapes, representing either solids or voids, to create the appearance of more complex shapes through a combination of operations. The B-rep approach (see Figure 3.07, next page) also utilized a combination of operations to create shapes but the resulting volume was defined by the sum of its enclosing surfaces, i.e. the boundaries between solid and non-solid. Later, the merge of these two methods, which allowed the editing of shape parameters and the change of shapes, contributed to the development of modern parametric modelling (Eastman et. al., 2008).

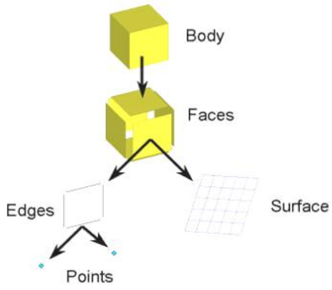


Figure 3.07 – In a B-rep approach, a cube can be defined by a collection of six identical square surfaces. Likewise, the square surfaces can be defined by four identical edges, which, in turn, can be defined as the connection between two points. (source: https://www.cadinterop.com/)

Solid modelling greatly contributed to the development of the first Building Modelling systems in the late 1970s and early 1980s. The first of such systems was the Building Description System, designed by Charles Eastman, which allowed designers to assemble building models with an integrated database, using a catalogue of elements stored in the program. This system also allowed the derivation of automatically consistent sections, plans, isometrics or perspectives from the model as well as the easy generation of cost estimates and material quantities (Eastman, 1975).

According to Michael Bergin (2011: 1), although Eastman's system saw little practical use, it served as *"an experiment (…) [to] identify some of the most fundamental problems to be tackled in architectural design over the next fifty years"*. And, in fact, he adds that *"Eastman's next project, GLIDE (Graphical Language for Interactive Design), created in 1977 at CMU, exhibited most of the characteristics of a modern BIM platform"*.

Following Eastman's example, many systems, such as GDS, EdCAAD, Cedar, RUCAPS, Sonata and Reflex, were created and developed using the same conceptual framework and even expanding on it. For instance, the Building Modelling system RUCAPS, developed in 1986 by GMW Computers, introduced the concept of temporal phasing of construction processes in BIM.

Unfortunately, these earlier systems required large and expensive hardware to operate and, when confronted with geometry-based CAD systems which saw their rise in popularity along with the personal computers in the 1980s, they fell into disuse (Johnson, 2014). In response to that, a new generation of Building Modelling systems was made available for the personal computer though these systems have yet to surpass the popularity of CAD systems.

The first of such systems was Radar CH (see Figure 3.08), now known as **ArchiCAD**, which according to Smith (2014) is viewed by many as the real beginning of BIM. ArchiCAD started being developed by Gábor Bojár in 1982 in Hungary and was released in 1984. Due to the limitations of the personal computer at the time, the software was unfit for large scale projects and, for a long time, was mainly used only in small scale projects such as family houses and small office buildings. Many improvements have been made since then and ArchiCAD is now the oldest continuously marketed BIM system and one of the major players in the market (Bergin, 2011).

Figure 3.08 – Interface of Radar CH (source: http://www.architectureresearchlab.com/)

Following ArchiCAD, **Revit** was also developed for the personal computer by Charles River Software, a company founded by Leonid Raiz and Irwin Jungreis toward the end of the millennia. The goal was also to create a parametric modelling software for architecture but one that could handle more complex projects than ArchiCAD.

The development of Revit led to a real shift towards effective BIM implementation in the building industry (Smith, 2014), partly due to the heavy promotion undertaken by Autodesk – which bought the Charles River Software company in 2002 – and partly due to improvements introduced by (and with) the software. For example, Revit reintroduced the concept of temporal phasing and scheduling as well as real-time cost estimations to BIM. Moreover, the availability of compatible softwares – both extensions of Revit released by Autodesk (e.g. Revit Structures, Revit MEP) and a great variety of specialization softwares related to BIM (e.g. analysis and simulation softwares) acquired by Autodesk – have led to various improvements in project collaboration and BIM implementation (Bergin, 2011). As a result, Revit is now the best known and current market leader for the use of BIM in architectural design (Eastman et. al., 2008).

As an example of less known and popular BIM systems, **Bentley's systems** also offer a wide range of related products from architecture to construction. These include Bentley Architecture, Bentley Structural, Bentley Building Mechanical Systems, Bentley Building Electrical Systems, Bentley Facilities, Bentley Power-Civil (for site planning), and Bentley GenerativeComponents. Although less popular, Bentley's systems have had an equally big impact in architecture. In particular, Bentley's GenerativeComponents, *"a design tool for exploratory architecture"* (Aish, 2003: 1) developed in 2003 by Robert Aish, offers an environment for complex and sculpted geometry to be manipulated and developed and has been used to promote and educate parametric modelling in practice (Qian, 2007).

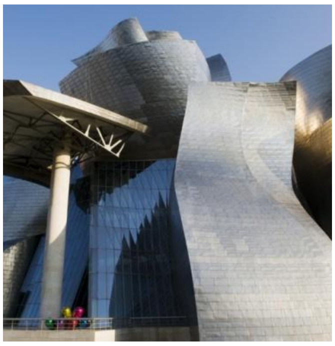


Figure 3.09 – Before it became Digital Project, CATIA was used by Frank Gehry to design and build the notorious Guggenheim Museum in Bilbao. (source: http://www.guggenheim-bilbao.es/)

Similarly, **Digital Project** – an architectural design software based on CATIA (a parametric modelling platform used in the aerospace and automotive industries), developed by Gehry Technologies around 2006 – also allows the exploration of parametric modelling and complex geometries and, together, both systems have revolutionized architectural design by enabling the design and construction of some very complex and sculptural architectural forms.

Tekla Structures is another example of a BIM system available nowadays for the personal computer. Formerly known as Xsteel, Tekla Structures was introduced in the mid-1990s by the Finnish company Tekla Corporation and quickly grew to be the most widely used steel detailing application throughout the world. Since then, the software has been extended to support the design, detailing and fabrication of other types of structures such as precast concrete, timber, and reinforced concrete structures, and structural analysis and engineering has been integrated into the software. In 2004, the software was renamed Tekla Structures to reflect this change (Eastman et. al., 2008).

Finally, other BIM tools available nowadays include AutoCAD-based applications and DProfiler. The **AutoCAD-based applications** include a series of 3D applications which were developed on top of AutoCAD before Autodesk acquired Revit and started developing that software instead. They are not parametric modellers with rules and constraints like the other BIM tools mentioned previously, and thus require changes to be manually propagated across drawing sets.

As for **DProfiler**, while technically not a general purpose BIM tool, it offers economic evaluations of construction projects (e.g. cost estimating and income forecasting) and is an important BIM tool for preliminary feasibility studies.

3.3 BIM Implementation

As mentioned before, BIM is not a recent development. However, in the past decades, its implementation has been relatively slow in the construction industry compared to other industries (e.g. manufacturing and engineering), until recently when a significant shift in momentum occurred (Smith, 2014). In fact, now that they have started to gain traction, BIM tools have been replacing former geometry-based CAD tools at a much faster rate than CAD tools took to replace hand drawings (see Figure 3.10).

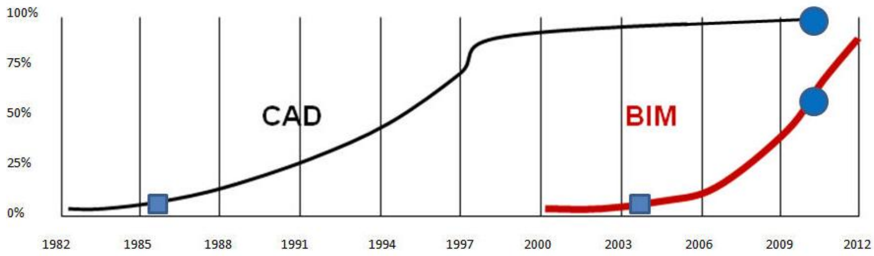


Figure 3.10 – A chart created by Dennis Neeley representing CAD versus BIM adoption. Blue squares indicate adoption rate at the time each chart was created. Blue circles represent the adoption rate in 2009. BIM has been adopted twice as quickly as CAD. (source: http://www.aia.org/)

One reason for this sudden acceleration of growth is improvements made in implementation issues (e.g. interoperability problems) as well as in BIM technology, which has finally started to gain the maturity and sophistication needed to efficiently support the design and construction processes. Moreover, according to a SmartReport published by the McGraw Hill Construction company (Bernstein et. al., 2014), major private and government owners who want to institutionalize BIM's benefits are increasingly becoming the driving force behind its adoption by mandating its use on their projects.

For instance, in the United States, The General Services Administration, the agency that manages all federal buildings, not only pioneered the implementation of BIM on public projects but also mandated its use for spatial program validation on all of its projects in 2007 (Smith, 2014). As a result, between 2007 and 2012, BIM adoption by contractor in North America skyrocketed from 28% to 71% (Bernstein et. al., 2014), with the United States being the world's biggest producer and consumer of BIM products and solutions as of 2011 (Wong, Wong and Nadeem, 2011).

Outside of the United States, the Scandinavian countries (i.e. Finland, Norway, and Denmark) are considered as the most active in BIM implementation (Wong, Wong and Nadeem, 2009). These countries adopted ArchiCAD (which originated from neighbouring country Hungary) early and therefore were among the earliest to adopt model-based design and advocate for interoperability and open standards in the AEC technology. A need for efficient prefabrication, which is facilitated by an accurate BIM model, also pushed for an early implementation (Khemlani, 2012).

Scandinavian governments have also had an important role in stimulating the development and implementation of BIM technologies (Smith, 2014). As of 2014, the Scandinavian countries, along with the United Kingdom and the Netherlands were the only European countries requiring the use of BIM for publicly funded building projects. Other countries from the European Union were urged to do the same in a recent directive issued by the European Parliament in January 2014 with the intention of spurring BIM adoption by 2016 (Autodesk, 2014).

On a national scale, Portugal is starting to take the first steps towards a BIM implementation. Over the recent years, several initiatives and work groups, such as the Portuguese Technical Committee for BIM Standardization (CT 197) and the GT BIM group (acronym for *Grupo de Trabalho* BIM), have been emerging all over the country to promote and encourage the migration to BIM so that Portugal may be able to compete with the international market in the AEC industries, where the BIM adoption is already underway (Venâncio, 2015).

In sum, on a global perspective, while some countries, such as the United States, Scandinavian countries, United Kingdom, Singapore, Canada, among others, have been implementing – and improving – BIM for several years, most countries are only now following their example and beginning to adopt BIM, even as they do so at an incredibly fast pace (Bernstein et. al., 2014).

4

ALGORITHMIC DESIGN FOR BIM

In chapter 2, we explained the impact of AD in architectural design and looked into how it is being used in architectural practices. We found that algorithmic and programming knowledge are becoming increasingly valuable skills in architecture and that many architects have already started to explore computational approaches in design practice. In fact, nowadays, various architectural projects have been successfully completed using processes that included algorithmic design phases.

The execution of these projects was possible mainly thanks to the availability of several programming environments and tools enabling the use of AD with geometry-based CAD tools which, for a long time, have been the predominant design tools in the architectural community.

However, as we also saw in chapter 3, BIM tools are now replacing former CAD tools. Given that fact, it becomes important to understand how AD can be used with the BIM methodology and how the capabilities of the two approaches can be combined. These are not entirely new questions as, in fact, as we'll show in this chapter, some architects have already started to use algorithmic processes with BIM.

Like with geometry-based CAD tools, this combined use of AD with BIM is supported by the recent availability of programming environments and tools which enable the exploration of AD with BIM applications.

In the next sections, we demonstrate how AD is being used with BIM in architectural practices using two architectural case studies and introduce a few of the available tools responsible for enabling the use of AD with BIM.

4.1 CASE STUDIES

To demontrate how AD is being used with BIM in architectural practices, two case studies were selected: the Aviva Stadium and the Louvre Museum of Abu Dhabi. In the following sections, the design process of both case studies will be explained, emphasising the implications of combining AD with BIM.

4.1.1 Aviva Stadium

Figure 4.01 – The AVIVA Stadium is a football and rugby stadium designed by Populous in 2005 and completed in 2010, located in Dublin, Ireland. (source: http://populous.com/).

Figure 4.02 – Inside of the Aviva Stadium (source: www.mercuryeng.com/).

The AVIVA Stadium (see Figure 4.01 and 4.02) was developed from start to finish using a parametric approach to BIM and was effectively delivered as a BIM building.

The project had an initial conceptual phase where studies were undertaken in Rhinoceros in order to explore the development and logic of the form's geometry (Shepherd et. al., 2011). Once the logic for the stadium's geometry defined, the parametric modelling framework developed for Rhinoceros was rebuilt in Bentley's GenerativeComponents (GC), a BIM system, to be further developed.

The development of the stadium benefited from a close collaboration between the design team (Populous) and the structural engineers (Buro Happold). To this end, a parametric GC script defining the stadium's geometry – in the form of a lofted surface with no thickness – along with a Microsoft Excel document containing the defining parameters of the GC script were developed to establish the geometric constraints that each firm (and later the subcontractors) had to work with. Through the sharing of these documents, both parties could simultaneously carry out their appointed tasks independently.

As each firm's work developed, it would be compared with the surface geometry articulated by the original script. Any deviation from the original form was discussed and, if deemed necessary, the GC script and Excel document could be updated and shared again (Garber, 2014).

Using the generated model as basis for design, both architects and engineers were able to collaborate to develop the stadium: the architects were responsible for the overall form of the building and the cladding layout, while the engineers were in charge of developing the structural members of the stadium. Figure 4.03 illustrates the collaborative design process undertaken for the development of the Aviva Stadium.

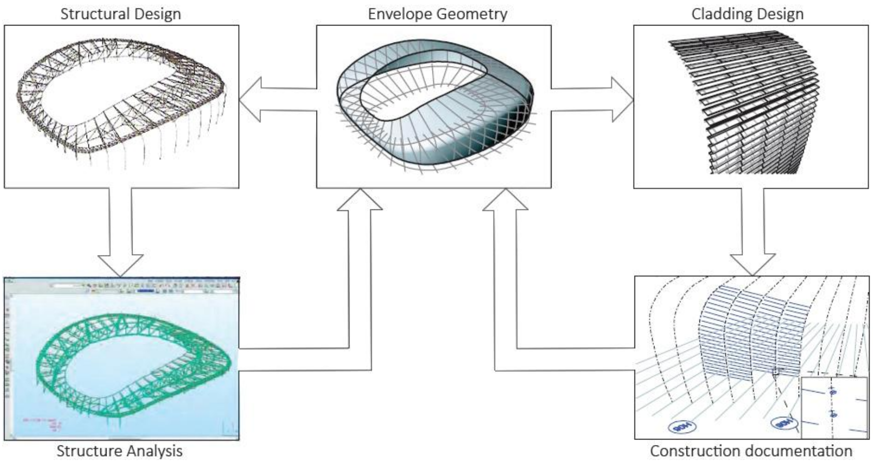


Figure 4.03 – Collaborative design process of the AVIVA Stadium (adapted from: Shepherd et. al., 2011)

For the engineers, this work involved extending the original GC script file to integrate structural analysis and simulation, namely through Autodesk's Robot Structural Analysis software. Doing so allowed them to optimize the stadium's structure as well as respond immediately to changes in the overall shape of the stadium without having to waste time rebuilding structural and analytical models to reflect the new geometry. Moreover, this allowed architects to quickly get feedback on the structural implications of their design decisions and a more optimal overall design was possible (Shepherd et. al., 2011).

As for the architects, defining the cladding layout involved developing an original script to simulate the entire cladding system in order to check if it would work correctly all around the stadium. The parametric modeling of the cladding system required the calculation of all parameters pertaining to the positioning and rotation of the individual panels of the cladding systems (see Figure 4.04). This information, along with a shared construction model, later became important information for the cladding subcontractor to produce the construction documentation necessary to create detailed fabrication models, and ultimately giving them the ability to manufacture the stadium's components parts.

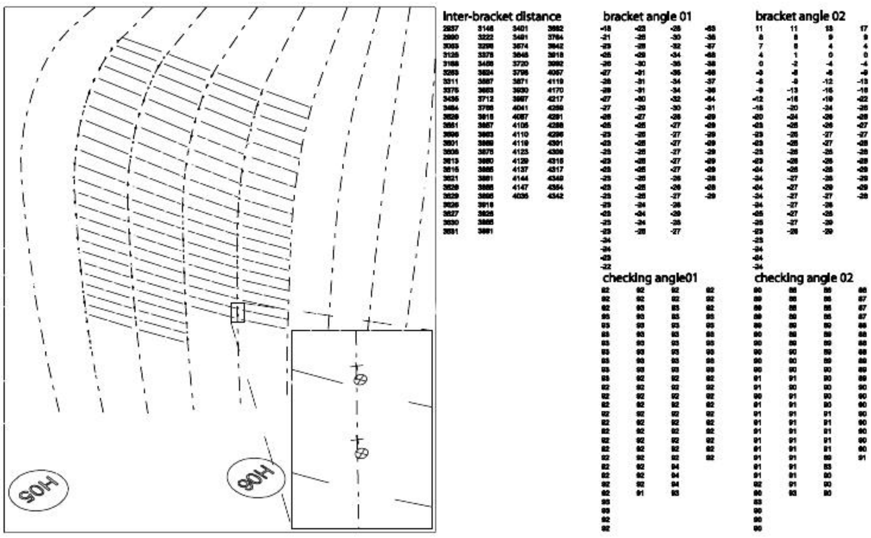


Figure 4.04 – Information issued for one facade bay (source: Shepherd et. al., 2011).

Like with the design process, the sharing of a single coordinated construction model – a wireframe model, based on the geometric model established by the architects, along with simple written criteria – determined the constraints that each subcontractor had to adhere to and enabled all parties to independently develop full construction models to the level of detail they require for production.

Figure 4.05 – Erection of the pre-fabricated steel work (source: Shepherd et. al., 2011).

The creation of fully detailed models enabled the fabrication of the stadium's component parts offsite, which were then shipped to the construction site. Much of the construction work afterwards was a sequence of connecting the pre-fabricated parts, erecting them and fixing them in their final position (see Figure 4.05).

In sum, the design process undertaken by the architects to build the AVIVA Stadium relied on a single parametric modelling framework which, in true BIM fashion, was developed and shared across the diferent design disciplines, ensuring the full coordination between teams. This process allowed an integrated workflow from conception to completion, with all information in the parametric framework being produced and updated via the scripting process itself (Garber, 2014). By using a parametric framework, responses to design changes were quicker and easier to manage since changes are propagated in the parametric model. Moreover, this approach allowed repetitive tasks to be automated, such as the calculation of all parameters regarding the cladding system and, later, its manufacture. Ultimately, this process placed the architect firmly in control of the project.

4.1.2 Louvre Museum of Abu Dhabi

Figure 4.06 – The Louvre Museum, located in the Saaydiyat Island in Abu Dhabi, was designed by Ateliers Jean Nouvel (architects), in association with Hala Wardé architecture, as well as Buro Happold (engineers), in association with TransSolar, between 2007 and 2012 (source: http://louvreabudhabi.ae/)

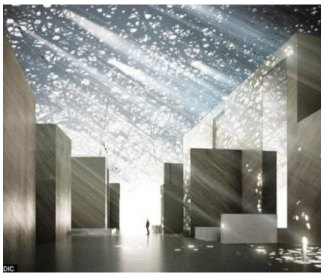


Figure 4.07 – Inside the museum, the rays of light that penetrate the dome resemble a 'rain of light', reminiscent to the light found inside oriental bazaars (source: http://louvreabudhabi.ae/)

At the heart of the design concept of the Louvre Museum of Abu Dhabi (see Figure 4.06 and 4.07) is a massive dome of approximately 180m of diameter, supported at only four points along its perimeter, thus seeming to float as a result. This dome displays a complex and seemingly chaotic pattern which creates dynamic light effects inside the museum.

For this concept to be realized, the project had to satisfy a series of multi-disciplinary requirements and constraints, including aesthetics, natural lightning, museography, environmental, structural and fabrication, all of which had a direct impact on the geometry and structure of the dome. To manage this complexity, the design team opted to use a concurrent design process where a central parametrically generated Digital Project model was shared among architects, engineers and consultants in order to ensure the proper coordination and collaboration between all parties. To facilitate this, a web-based model repository was put in place by Gehry Technologies, allowing each party to develop their respective work simultaneously on the same model (Imbert et. al., 2012).

At the base of this Digital Project model was a multi-constraint parametric framework where every project constraint could be translated into specific geometric rules for the design. This approach allowed design alternatives that did not satisfy the requirements to be automatically filtered out and design alternatives that did to be quickly tested and compared in order to guide the project to a consensual, optimal solution. As a result, the design of the dome evolved naturally along with the various constraints. Figure 4.08 illustrates the parametric workflow used in the design of the dome of the Louvre Museum of Abu Dhabi.

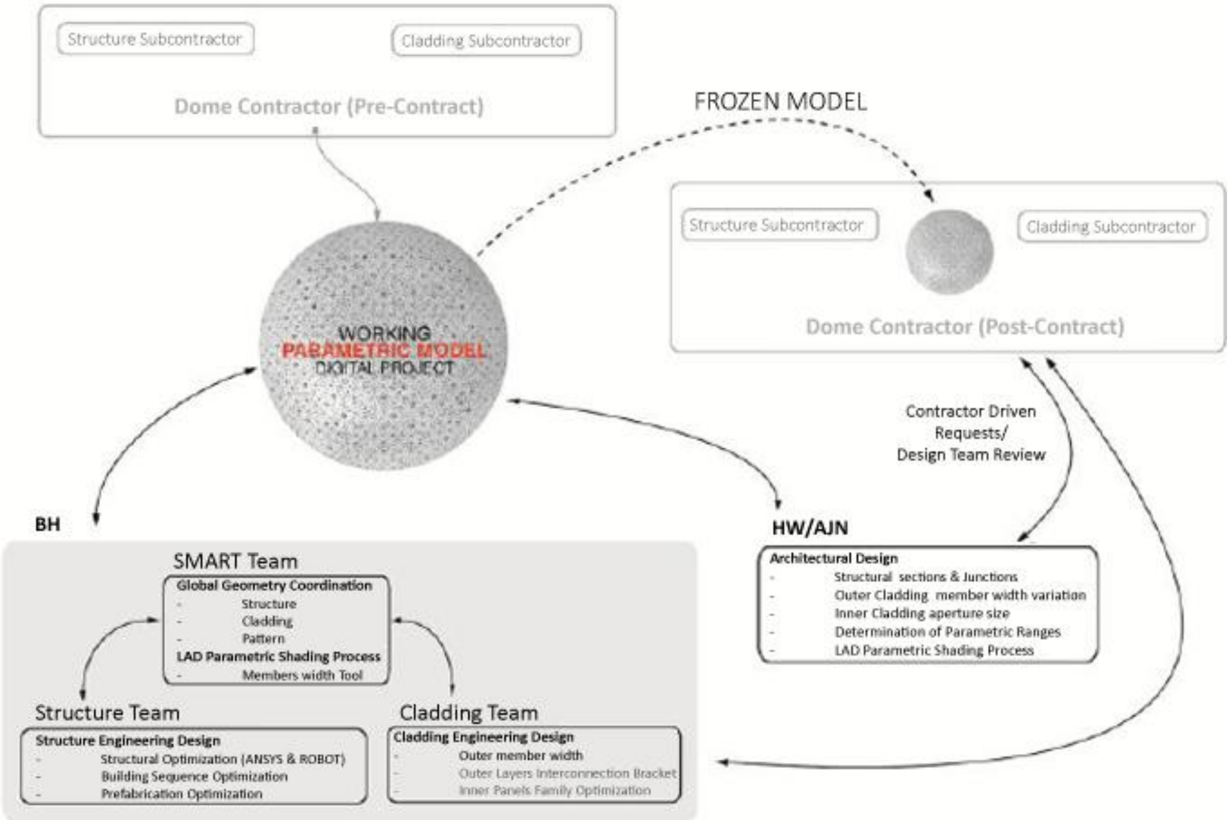


Figure 4.08 – Diagram of the parametric workflow, detailing how the various constraints of the project were folded into the central parametric model of the dome (source: Imbert et. al., 2012)

The dome itself ultimately adopted a double-layered steel grillage/shell structure, creating two cladding surfaces 6 meters apart (see Figure 4.09). These surfaces contain ten cladding layers: five layers on the inner surface of the dome and five layers on the outer surface. Shaping each cladding layer is a simple geometric pattern of an isosceles triangle repeated and rotated to form a system of squares and hexagons. Each cladding layer then has a distinct pattern scale and orientation in relation to the others as well as mass-customized member widths (see Figure 4.10, following page). By varying these values, it was possible to test and compare different levels of perforations for the dome and conse-

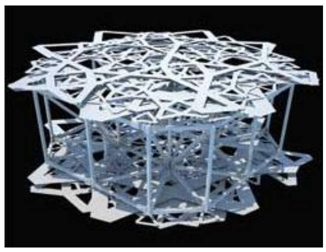


Figure 4.09 – Partial digital model of the dome structure and cladding. (source: www.bdonline.co.uk/)

quently control the lightning and achieve comfortable environmental conditions inside the dome.

Figure 4.10 – A picture of the dome under construction showing the pattern at the base of each layer, as well as the differences between layers (source http://www.grupo-sanjose.com/).

As the design of the dome developed, the detailing of the metal members of the cladding layers was also resolved collaboratively over the parametric model. Ultimately, the method of assigning member widths drove the cladding construction details. All construction documents were fully automated from these detail studies, producing both detail drawings and numeric quantities (e.g. members' length and width as well as vertex-wise angle defects) for the tendering of the dome (Imbert et. al., 2012).

In the end, Waagner-Biro was responsible for the construction of the dome. To that effect, they developed a program written in the F# programming language to represent and organize all cladding members of the dome. According to Goswin Rothenthal (n.d.: 1), *"this application enabled [them] to have an integrated workflow from the main geometry setout all the way down to the manufacturing data in a single parametric model".*

Overall, the dome of the Louvre Museum of Abu Dhabi was developed as a parametrically-driven BIM project. Through the sharing of a central, data-driven model over a web environment, full coordination and collaboration of all design disciplines was achieved. Driving this model was a multi-constraint parametric framework which guided the dome's design and enabled an optimal solution to be found. Finally, this approach offered a better control over the entire design process, managing the project's complexity and offering an integrated workflow from conception to manufacturing.

4.2 AD Tools for BIM

To enable the exploration of AD with BIM, several programming environments and tools have emerged, allowing the development of programs that generate models BIM applications. Many of these programming environments and tools have been tailored specifically for architects, which means that they allow the development of programs without requiring extensive knowledge of programming languages and computer science concepts.

In the following sections, we briefly introduce a few tools which allow the exploration of AD with BIM. These include tools that are already available in BIM applications as well as plug-ins developed for the production of AD programs.

4.2.1 *Lyrebird*

Lyrebird is a plug-in developed by LMN Architects as an interoperability tool between Revit and Grasshopper, a popular, easy-to-use visual programming language originally developed for Rhinoceros 3D. Lyrebird enables the usage of Grasshopper to structure the information needed to identify and instantiate the correct BIM objects in Revit (Logan, 2014).

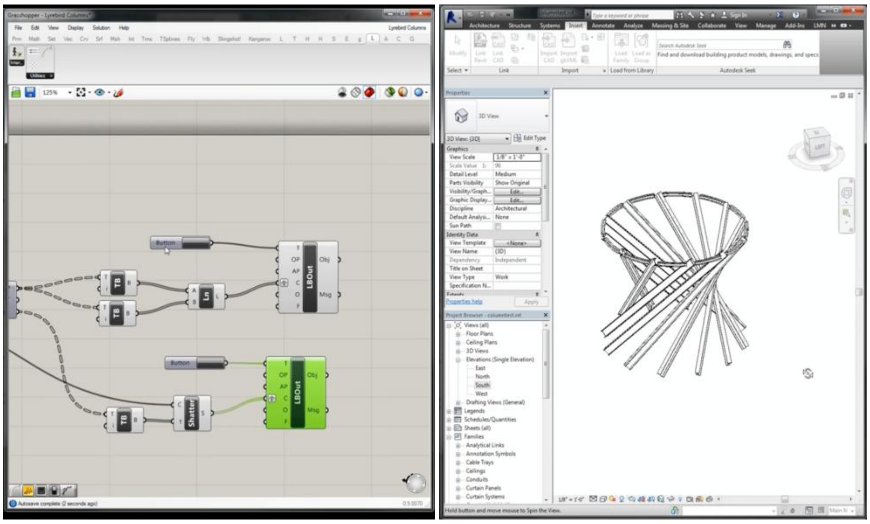


Figure 4.11 – Lyrebird's work environment and the corresponding model generated in Revit (source: www.pinterest.com)

4.2.2 *The Rhino-Grasshopper-ArchiCAD Connection*

The Rhino-Grasshopper-ArchiCAD connection is another plug-in based on Grasshopper but this time for ArchiCAD. Developed by Graphisoft, Rhino-Grasshopper-ArchiCAD makes use of geometrical information created in Rhino with Grasshopper to create BIM objects in ArchiCAD (Graphisoft, 2015).

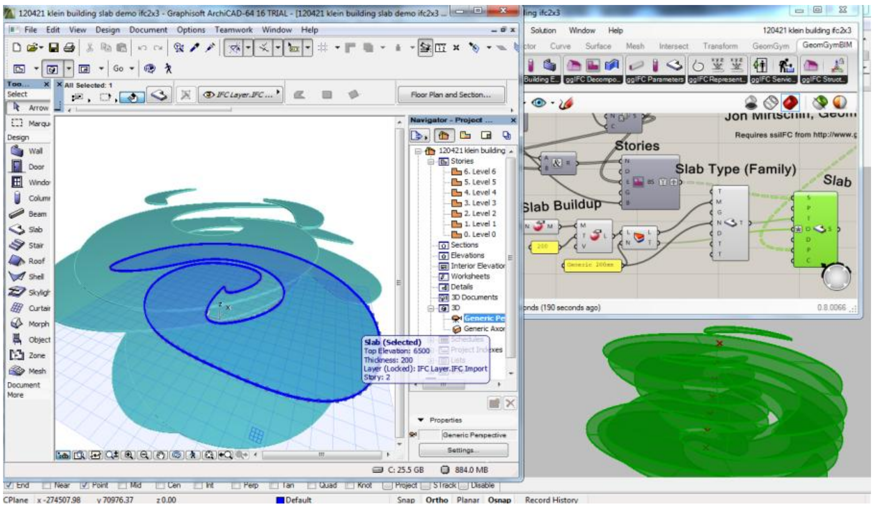


Figure 4.12 – The Rhino-Grasshopper-ArchiCAD connection's work environment and the corresponding models generated in both Rhino and ArchiCAD (source: http://ssi.wdfiles.com/).

4.2.3 Dynamo

Dynamo is a plug-in for Revit that is strongly influenced by visual programming languages. Like Grasshopper, users create a workflow of nodes and connections to generate BIM objects in Revit. (Autodesk, 2015).

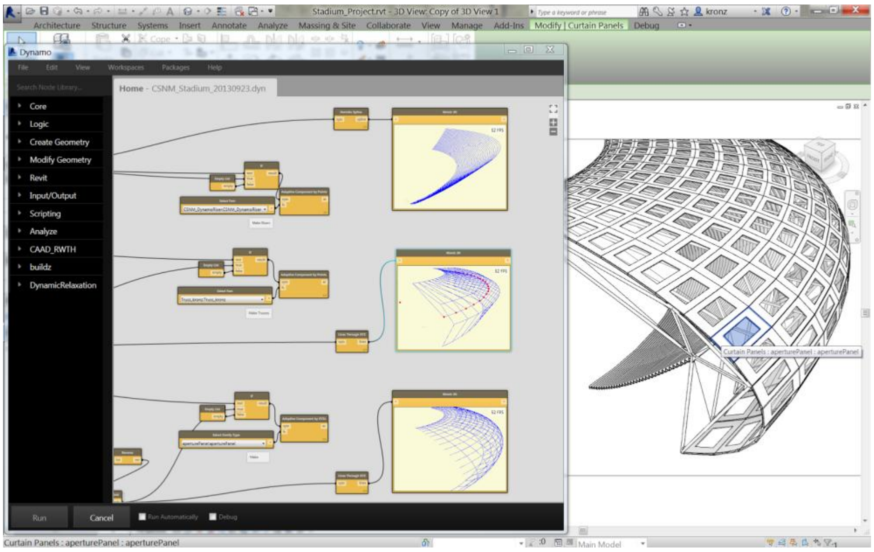


Figure 4.13 – Dynamo's work environment and the corresponding model generated in Revit (source: http://2.bp.blogspot.com/)

4.2.4 GenerativeComponents

GenerativeComponents (GC) is a parametric and associative system developed for Bentley's Microstation. GC has three ways of programmatic interaction: (1) by creating a workflow of nodes and connections in the Symbolic Diagram, not unlike Grasshopper; (2) by defining relationships among objects with simple scripts in GCScript; and (3) by writing more complex programs using the C# programming language, allowing the definition of complex algorithms (Aish and Woodbury, 2005).

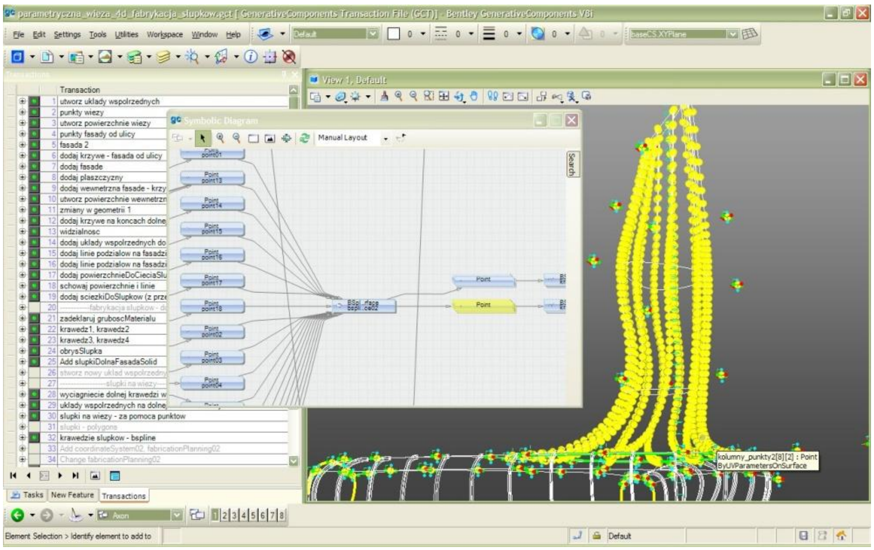


Figure 4.14 – GenerativeComponents' work environment. The model in the image was generated using the Symbolic Diagram within GenerativeComponents (source: http://www.projektowanieparametryczne.pl/).

4.2.5 RevitPythonShell

RevitPythonShell is a plug-in developed for Revit that allows users to take advantage of the Revit's Application Programming Interface (API) but using Python. This tool was developed to simplify the workflow needed in order to use the Revit's API (Thomas, 2009).

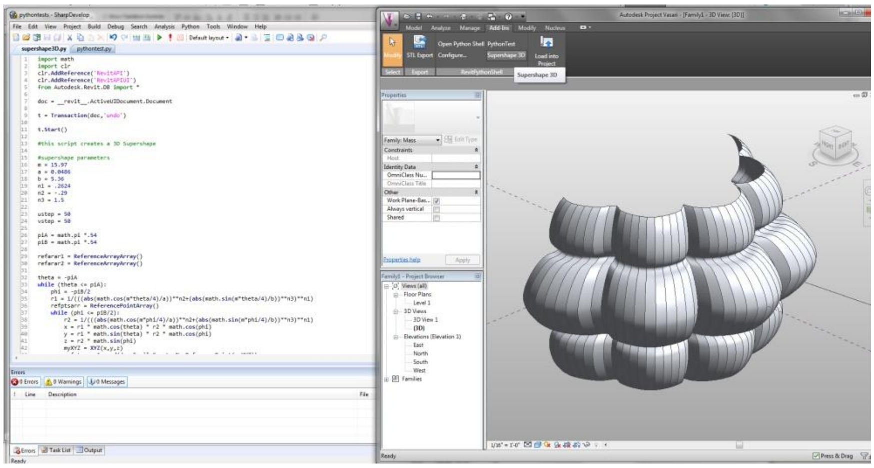


Figure 4.15 – RevitPythonShell's work environment and the generated model (source: theproving-ground.wdfiles.com/).

4.2.6 Rosetta

Finally, Rosetta is a programming environment tool that offers portable AD for both geometry-based CAD and BIM applications. While originally developed as an AD tool for CAD (Lopes and Leitão, 2011), Rosetta was recently extended to support BIM applications, namely Revit and ArchiCAD. With Rosetta, scripts can be created in different programming languages and models can be generated in the different supported CAD and BIM tools.

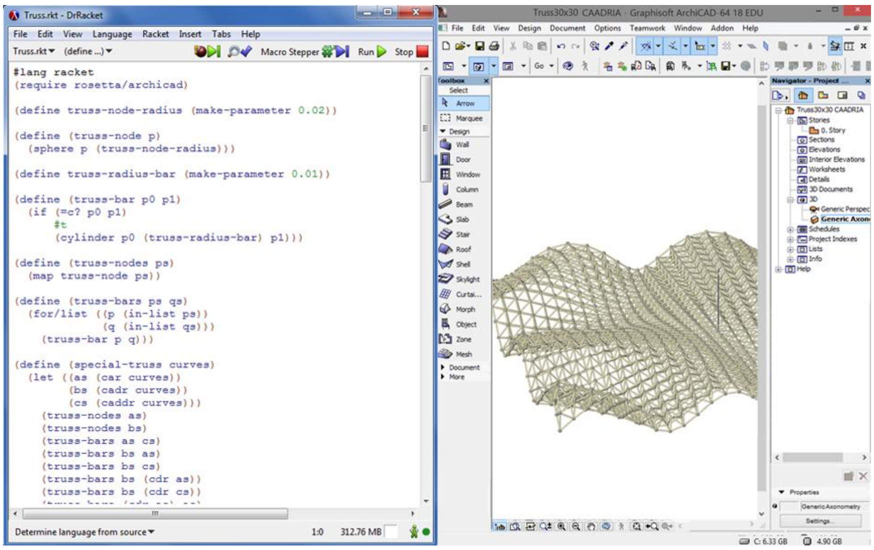


Figure 4.16 – Rosetta's work environment, DrRacket, and the corresponding model generated in ArchiCAD.

4.2.7 Conclusion

In the previous sections, we presented some of the most representative tools that support AD for BIM. Of these tools, Rosetta is the only one that supports portability, i.e. the ability to work with more than just one specific BIM tool. For example, while Dynamo only works with Revit and GenerativeComponents with Microstation, Rosetta works and produces equivalent results in both Revit and ArchiCAD. Similarly, while Lyrebird and the Rhino-Grasshopper-ArchiCAD connection both use the Grasshopper environment and language to generate objects in Revit and ArchiCAD respectively, both plug-ins are specifically tailored for the BIM tool they were designed for and are, thus, not portable.

This portability makes Rosetta ideal to test algorithmic processes with BIM tools as it offers a more generalized use of AD for BIM, i.e. not entirely dependent on the workings of one specific BIM tool. For this reason, in the following sections, we will be using Rosetta to experiment and evaluate our ideas regarding AD for BIM.

II

Algorithmic-based Building Information Modelling

5

INTRODUCTION

As shown in the previous chapters, digital technologies have been changing architectural design over the past years. In particular, Algorithmic Design (AD) and Building Information Modelling (BIM), both highly promising developments in architecture, have been affecting and improving how architects design.

AD offers a challenging but flexible new way of designing, one rooted in algorithmic logic. This approach introduced a whole new field of design exploration in architecture, allowing architects to efficiently conceive and explore complex geometries, and helped improve the design process by providing a way of automating repetitive, time-consuming tasks that had to be manually executed before. Realizing the usefulness of AD, many architects have been exploring computational approaches in design practice over the past years and, nowadays, algorithmic and programming knowledge are becoming increasingly valuable skills in architectural design.

Meanwhile, BIM has been establishing itself as new design methodology with the potential to bring many improvements to architectural design, since boosting projects' productivity to reducing the overall costs. Due to its growing list of advantages, BIM has been quickly gaining the interest of the architectural community over the past decade and a half, with the adoption of BIM tools growing at a fast pace and replacing former geometry-based CAD tools in architectural practices. In fact, the gains are so substantial that 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 AEC industries.

Given this context, we want to continue exploring algorithmic processes in architecture but adapted to the new BIM methodology which has started to replace former CAD-based design practices. The result is a new approach to design, one that we call A-BIM, acronym for Algorithmic-based Building Information Modelling, an approach that has only recently started to be explored in architecture due to the recent availability of software that enables the use of algorithmic processes with BIM tools.

In the second part of this thesis, we propose A-BIM as a design approach that can offer great benefits to architectural design. To do so, we start by introducing A-BIM and explain how this approach is different from an algorithmic approach to geometry-based CAD. Due to the differences between CAD and BIM, the corresponding programming methodologies will also differ. Thus, we also propose a programming methodology for A-BIM, adapted to the BIM paradigm (Chapter 6).

Afterwards, in order to evaluate the capabilities of A-BIM, we selected an architectural case study – the Absolute World Towers designed by MAD Architects (see Figure 5.01) – which we modelled using three different approaches: (1) an algorithmic CAD approach, (2) the A-BIM approach that we propose, and (3) a manual BIM approach.

Figure 5.01 – The Absolute World Towers, designed by MAD Architects (source: http://www.e-architect.co.uk/).

After introducing the case study and the modelling process (Chapter 7), we explain and analyze the modelling of the case study in the three chosen approaches (Chapter 8) in order to find the benefits and drawbacks of A-BIM in relation to the other two. In the end, we provide an analysis of the gains and losses obtained from using A-BIM (Chapter 9).

6

Algorithmic-based Building Information Modelling

In order to allow AD in the BIM context, we propose Algorithmic-based Building Information Modelling (A-BIM), an algorithmic approach to BIM which we define as the generation of BIM models through algorithms.

Similarly to AD, A-BIM creates a paradigm shift in the design process, since the designer, instead of developing the model directly in the BIM application, develops the algorithm which generates the model in the BIM application. At the same time, similarly to BIM, the generated model consists of an intelligent 3D representation of a building containing relevant information for design and construction, and constrained by parametric and associative rules. However, unlike BIM, the source of all information is the parametric framework at the base of the generated model, which can still be shared and developed concurrently between members of the design team and offering a more flexible, controllable, and integrated way of managing the project's data, as we saw with the AVIVA Stadium and the Louvre Museum of Abu Dhabi.

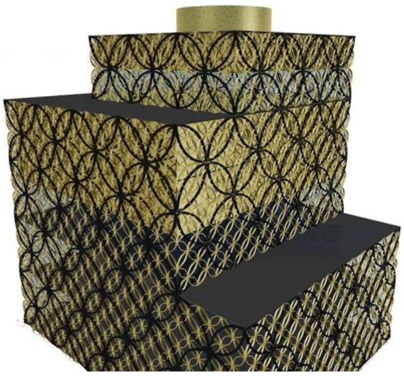


Figure 6.01 – A parametric model of a facade (source: Leitão, Caetano and Correia, 2016).

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 façade (see Figure 6.01), of entire buildings (see Figure 6.02), or even a whole city (see Figure 6.03). However, using this approach also requires programming knowledge and an initial investment of time and effort to produce the algorithm which, in the end, might not always prove to be more efficient than simply producing the model, or models, manually.

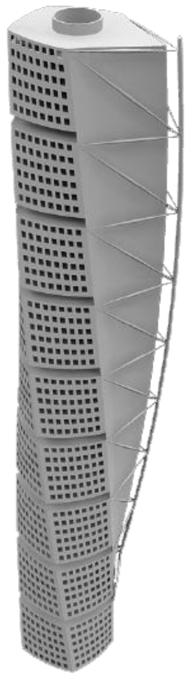


Figure 6.02 – A parametric model of a building (source: Leitão 2012).

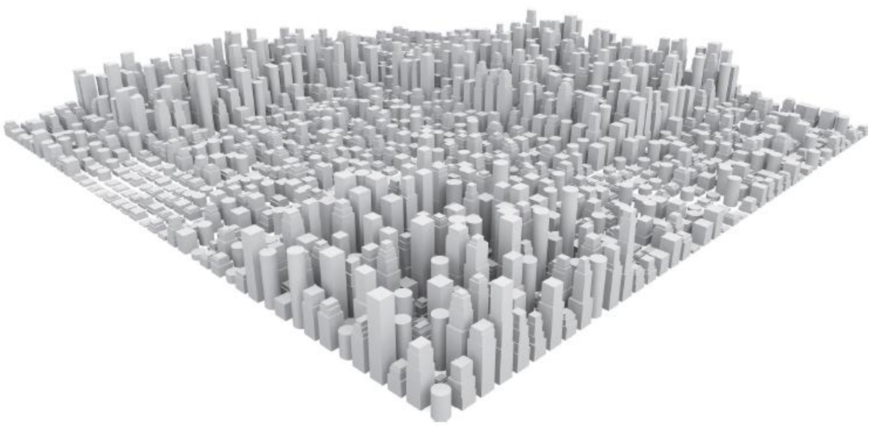


Figure 6.03 – A parametric model of a city (source: Leitão, 2012).

On that subject, we found that an algorithmic approach is particularly efficient on projects with a large degree of repetition, e.g. skyscrapers, or in projects where extensive exploration is required, e.g. buildings with complex geometries. Due to its algorithmic nature, repetitive modelling processes can be automated and the resulting algorithm can be highly parametric, allowing a wide range of design possibilities to be quickly explored and evaluated by simply experimenting with different parameter values. Moreover, because the generated objects are parametrically interdependent, changes to these parameters can be propagated to the entire model.

6.1 Programming for BIM

As mentioned before, an algorithmic approach 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 write the algorithm in a language that the computer understands so that it can be executed. These languages are known as programming languages.

This was already the case with programming for CAD and these concepts also apply to A-BIM. However, from a programming standpoint, A-BIM requires a different approach from the one needed for geometry-based CAD. This is a consequence of the differences between working with CAD and BIM tools. Due to these differences, the corresponding programming approaches will also differ. Thus, a programming methodology fit for BIM is required.

In the next section, we describe a set of guidelines for the programming methodology that we found appropriate for BIM, while comparing it to the programming methodology needed for CAD. Our programming methodology is especially adapted to textual programming languages because we found them to be more flexible than visual programming languages. In fact, studies (Leitão and Santos, 2012) have shown than visual programming languages, while easy to learn and use, do not perform well with more complex programs, such as the ones needed to model entire buildings, making those programs harder to read, change and reuse.

6.1.1 From CAD to BIM

One major difference between working with geometry-based CAD and with BIM is that, as stated before, 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 6.02): 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.

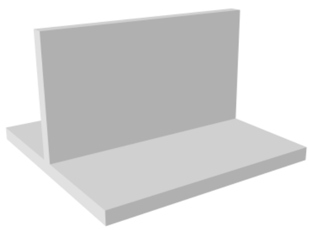


Figure 6.04 – A generic slab and wall.

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 meaning. 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 of pre-modelled shapes or forms which can facilitate the design process of complex geometry. However, typically, these objects are not as parametric as the ones available for BIM tools, thus restricting the designer’s ability to manipulate their geometry as they see fit.

7

CASE STUDY: ABSOLUTE WORLD TOWERS

In order to evaluate the capabilities of A-BIM, we selected an architectural case study which we modelled using three different approaches, including A-BIM. The case study in question is the Absolute World Towers, two residential twin towers designed by MAD Architects and located in Mississauga, Canada.

In the following sections, a brief overview of the Absolute World Towers is given and the purpose and specifications of the modelling processes undertaken here are explained.

7.1 THE ABSOLUTE WORLD TOWERS: CONTEXTUALIZATION

The Absolute World Towers (AWTs) are part of a larger project of a master-planned community of five residential condominium towers, spread out throughout a total area of more than 158 000 m^2, located in Mississauga city centre of the Great Toronto Area, Canada. After the first three towers – Absolute City Center 1 & 2 and Absolute Vision – were completed in 2008, site developers Fernbrook Homes and Cityzen Development Group decided to break away from the norm and sponsored an open international design competition to build a new iconic building (which later became two) that would be a landmark in the city. This project was carried out with a public partnership with the Mayor and the City showing considerable interest (and even participating) in the development of the new tower.

The competition was won by Chinese architecture firm MAD Architects with their unique design of a skyscraper which sought to create a marker in Mississauga skyline with its flowing, organic form that recreated the fluidity of natural lines (Lagendijk et. al, 2012). Its curvaceous figure, reminiscent of a feminine human figure, earned the first tower its nickname of "Marilyn Monroe Tower" by local residents (see Figure 7.01).

While the competition only requested one tower, the design of the first tower became so popular amongst Mississauga local residents that a second, complementary, more *"masculine"* tower was eventually built right next to the first (Lagendijk et. al, 2012). Together, the two towers stand out from their surroundings, and quickly became iconic landmarks in the city (see Figure 7.02).

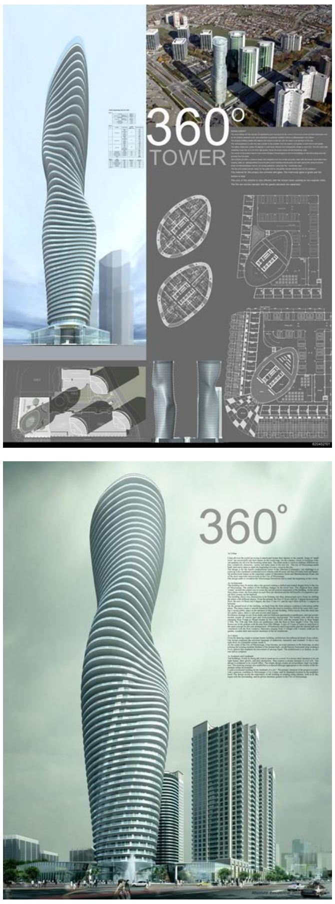


Figure 7.01 – Different design proposals for the first Absolute World Tower for the competition. (source: http://www.skyscrapercity.com/)

Figure 7.02 – Absolute World Towers, designed by MAD Architects and located in Mississauga, Canada (source: http://www.domusweb.it/)

Some project data of the Absolute World Towers is presented below:

ABSOLUTE WORLD TOWERS PROJECT DATA (adapted from: Lagendijk et. al, 2012, p.17)

Location: Hurontario St. & Burnhamthorpe Rd.
Building Function: Residential

Absolute World 56
Height to Architectural Top: 176m
Stories: 56
Total Area: 45,000 m^2
Total Cumulative Floor Plate Rotation: 209º, min.1 to a max. of 8º rotation per floor

Absolute World 50
Height to Architectural Top: 158m
Stories: 50
Total Area: 40,000 m^2
Total Cumulative Floor Plate Rotation: 200º (consistent 4º rotation per floor)

Owners/Developers: Fernbrook Homes & Cityzen Development Group
Design Architect: MAD
Architect of Record: Burka Architects
Structural Engineer: Sigmund Soudack & Associates Ltd.
MEP Engineer: ECE Group Ltd.; Stantec
Concrete Engineer: Coffrey Geotechnics
Contractor: Dominus Construction Group
Forming Contractor: Premform
Material Suppliers: Innocon; Gilbert Steel Ltd.
Landscape Architect: NAK Design
Interior Designer: ESQAPE Design

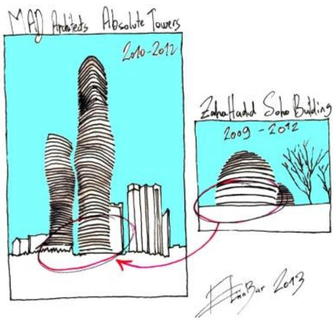


Figure 7.03 – A couple of sketches by Eliinbar emphasizing the strong horizontality of the towers and comparing it to the Soho Building by Zaha Hadid Architects (source: https://archidialog.com/).

Both AWTs share the same design concept, although with a few differences in their development: a vertical building with a marked horizontality (see Figure 7.03) and an organic, sculptural overall form. This concept manifests in the form of a sequence of prominent rotating slabs supported by a grid of structural walls which are repeated in every floor, as can be seen in Figure 7.04.

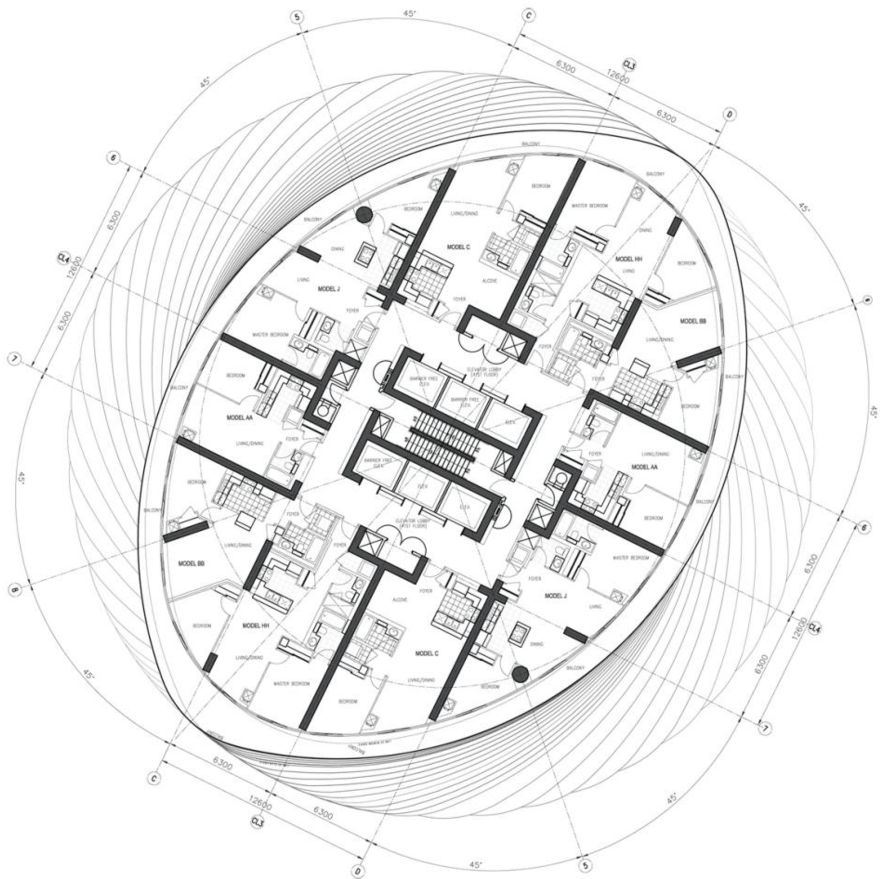


Figure 7.04 – Absolute World Towers: plan of a typical floor (source: http://www.dezeen.com/)

Besides the two residential towers, the final project of the ATWs also included six underground parking floors and a two-floor recreational area around the base of Tower 2, as can be seen in Figure 7.05. However, only the two towers were included in our modelling process.

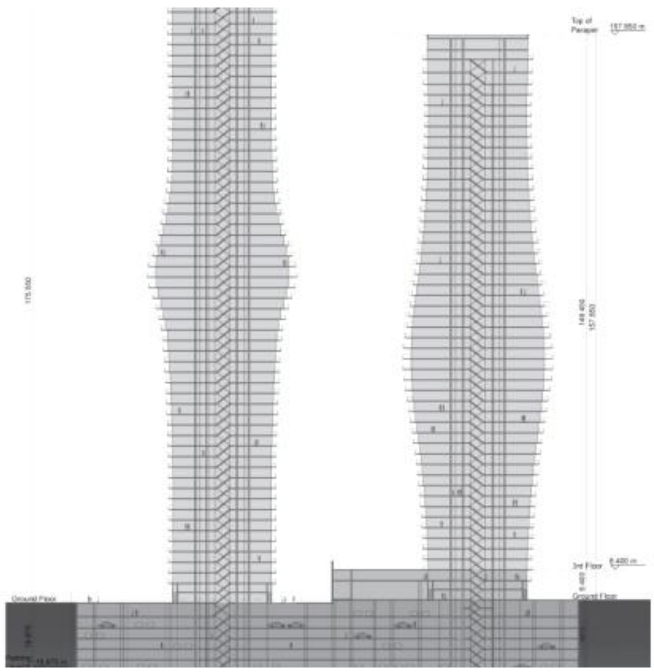


Figure 7.05 – Absolute World Towers: section (source: http://phaidonatlas.com/)

7.2 MODELLING OF THE ABSOLUTE WORLD TOWERS

To evaluate the capabilities of A-BIM, we modelled the AWTs using three different but related design approaches, namely (1) an Algorithmic approach to geometry-based CAD (A-CAD); (2) an Algorithmic-based Building Information Modelling approach (A-BIM); and (3) a manual BIM approach. We chose the AWTs for this evaluation because, as skyscrapers, they can benefit immensely from an A-BIM approach due to their repetitive nature and formal homogeneity, as we will show in the following chapters.

The aim of the modelling process undertaken here 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 intent is not to faithfully reproduce every aspect and detail of the AWTs but rather to provide a comparative study of the three approaches.

For both algorithmic approaches, i.e. A-CAD and A-BIM, the modelling process entailed capturing the design intent and ideas behind the modelling of the towers, and translating them into a programming language. Since both towers share the same design concept, they can be both generated with the same parametric program. In the end, we expect to be able to not only reproduce both towers, but also to generate several different design alternatives for the AWTs using the same parametric framework.

Although the concept of BIM presupposes the creation of complete BIM models containing objects pertaining to all design disciplines, the scope of this thesis is only that of the architectural project. As a result, the resulting BIM model will not include, e.g., structural analyses or facilities.

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.

Finally, the manual BIM approach was modelled in Revit while both A-CAD and A-BIM were implemented using Rosetta which, as mentioned before, 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, as illustrated in Figure 7.06 (following page), thus providing a more generalized implementation of AD for both CAD and BIM.

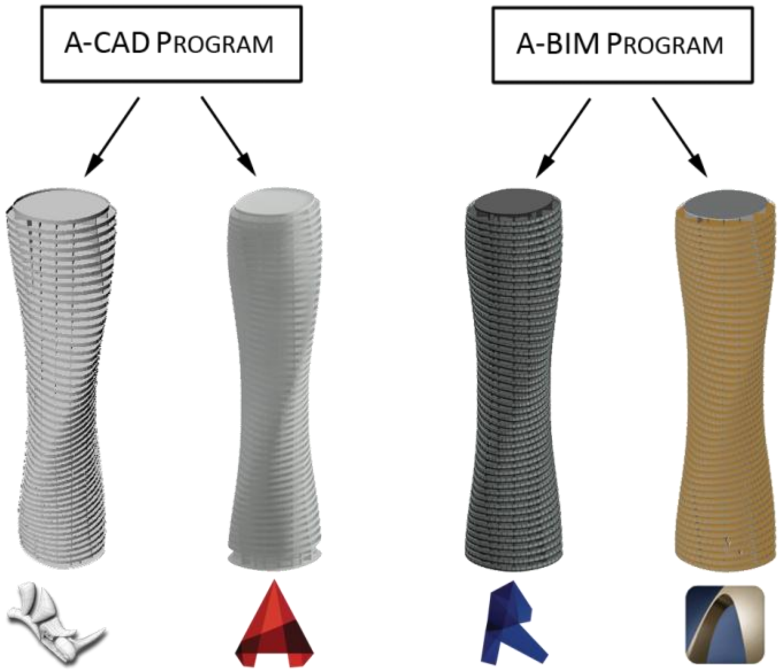


Figure 7.06 – The A-CAD program was implemented for both Rhino and AutoCAD while the A-BIM program was implemented with Revit and ArchiCAD.

However, there are limits to this portability achieved with Rosetta, especially for A-BIM. For instance, some of the building components contain features that are too different between Revit and ArchiCAD to be made portable. As an example, the creation of stairs in Revit requires several parameters that ArchiCAD does not need thus making it difficult to find one normalized way of creating stairs that works for both Revit and ArchiCAD. In these cases, we are still able to utilize these building components but had the choice of one of two different ways to create them: the Revit way or the ArchiCAD way. In the end, we chose to take advantage of Rosetta's portability whenever possible, thus implementing our modelling process in both Revit and ArchiCAD, and, when that was not possible (namely for the modelling of the stairs), we chose one of the two ways.

8

Comparative Study: Modelling of the Absolute World Towers

In this chapter, we explain and compare the modelling processes of the AWTs in the three aforementioned approaches, divided by building components.

8.1 Modelling Process of the Levels

The first step taken in the modelling of the AWTs was to create levels corresponding to the floors' heights, thus establishing the number of floors as well as their respective heights.

8.1.1. A-CAD

Levels are a BIM concept that is non-existent in CAD applications. However, in the A-CAD program, we still implemented a method to create the floors and establish their heights. It consists of creating a list of heights corresponding to the sum of every floor's height. This list is generated using a procedure that, given a total height for the building and a desired number of floors, computes a list of numeric values corresponding to each floor's height.

8.1.2. A-BIM

While a non-existent concept in CAD applications, levels are important 3D elements in BIM, allowing BIM applications to attribute objects to a given height as well as establish associations between heights.

To create the levels, we used the same procedure as A-CAD to generate the list of floor heights. This list is then used to mechanically generate all the levels: for each numeric value corresponding to a floor height in the list, a level is created, placed at the corresponding height (see Figure 8.01).

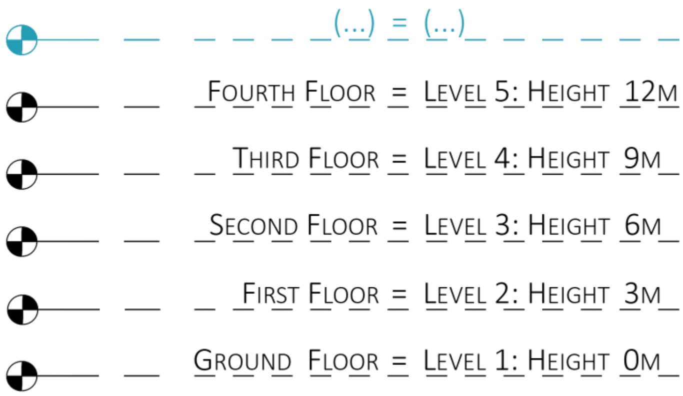


Figure 8.01 – Each level represents a floor and has a corresponding height.

8.1.3. *Manual BIM*

A level can be created by using the 'Level' tool (see Figure 8.02) available in the Revit toolbar and by writing the numeric value corresponding to the height of that level. This process is then repeated – or the formerly created levels are copied – to create the remaining levels.

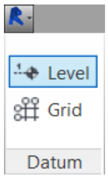


Figure 8.02 – The 'Level' tool available in Revit's toolbar can be used to create levels.

8.1.4. *Analysis*

The creation of levels consists of an important first step for BIM, allowing BIM applications to attribute objects to a given level and height as well as establish associations between heights. However, for A-CAD, this step is merely an intermediate abstraction in the program, i.e. unnecessary as the CAD applications themselves have no use for this information but useful to differentiate and organize different types of information and making it available for latter reuse.

8.2 Modelling Process of the Slabs

Once the levels created, we started by modelling the building's slabs. Due to the prominence of the slabs in the building, the overall form of the AWTs greatly depends on both the shape of the slabs as well as their development along the building's shaft.

Both AWTs have ellipsoidal slab shapes that rotate from floor to floor along the building's vertical axis. This rotation is different for both towers: the first tower has an inconsistent rotation, with varying angles of rotation per floor (1º minimum to 8º maximum), while the second tower has a consistent rotation of 4º per floor.

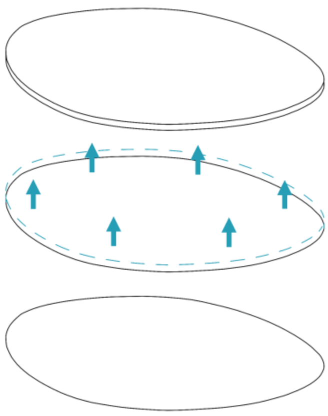


Figure 8.03 – In A-CAD, given a surface with the desired slab shape, the surface is extruded to create the slab.

8.2.1. *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 8.03).

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

8.04). 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.

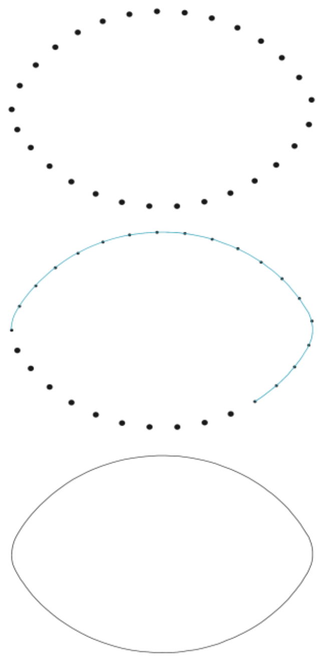


Figure 8.04 – The shape of the slab is defined by a list of points that outlines its boundary.

Each slab is then coupled with a height from the previously created list of heights, allowing the distribution of the slabs along the different floors (see Figure 8.05). 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.

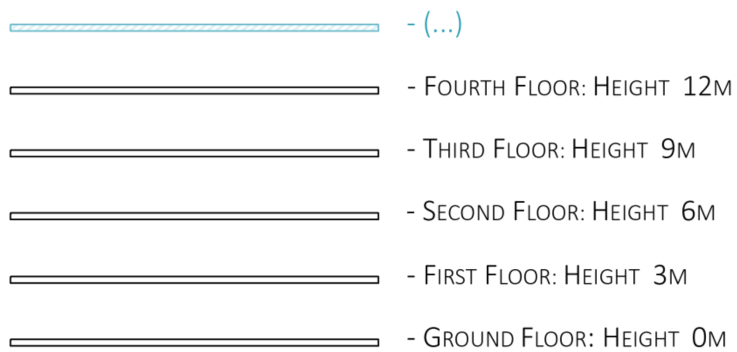


Figure 8.05 – The slabs are distributed along the floors by distributing them along the list of heights previously created, thus establishing the number of slabs and their respective heights.

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 tower. In order to model both AWTs, we created two specific lists of angles, corresponding to the actual rotation of each tower.

8.2.2. A-BIM

As mentioned in chapter 6, 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 thus facilitating their creation process. 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 the previously created list of levels and an angle of rotation from the same list of angles created for A-CAD in order to distribute the slabs along the different floors and ensure the desired rotation for every slab.

8.2.3. 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 a slab, we select the level in which we want to place it and, using the 'Floor' tool (see Figure 8.06), 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.

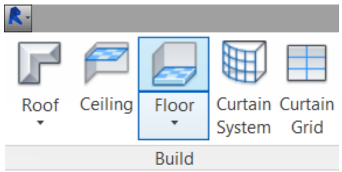
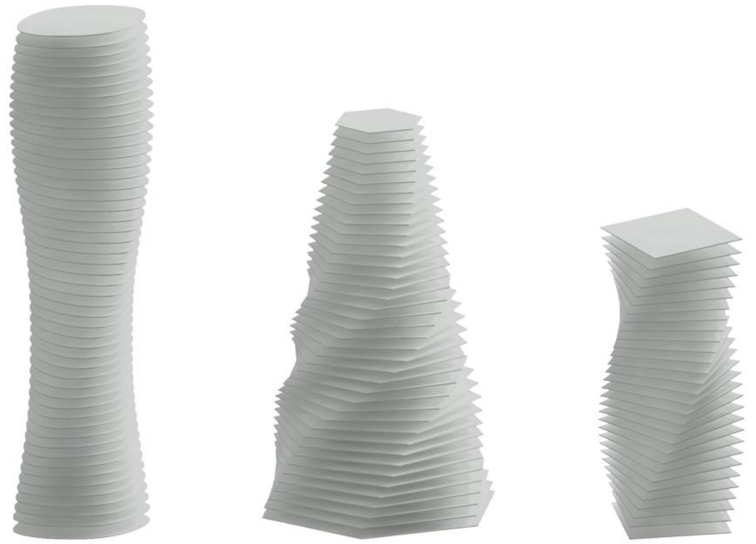


Figure 8.06 – The 'Floor' tool can be used to create slabs.

8.2.4. 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 8.07 (following page) 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.



50 Floors	40 Floors	30 Floors
Slab Shape: Ellipse	Slab Shape: Hexagon with a decreasing scale	Slab shape: Square
Consistent Angle of Rotation: 4º (Rotation of the AWT 2)	Inconsistent Angle of Rotation: Min. 1º to Max 8º (Rotation of AWT 1)	Consistent Angle of Rotation: 6º

Figure 8.07 – 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. The first instance was generated with the parameters of the AWT 2.

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.

8.3. Modelling Process of the Openings in the Slabs

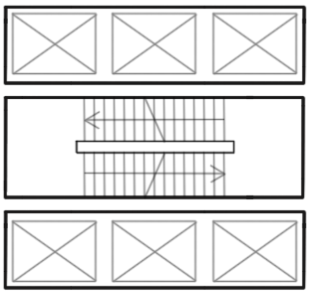


Figure 8.08 – Shape and configuration of the openings of the AWTs.

After creating the slabs, we were able to produce the openings in the slabs corresponding to the elevators and stair shafts, which are located in the center of the buildings. In the case of the AWTs, there are three openings: one larger rectangular opening right in the centre of the building corresponding to the stair shaft and two smaller, identical rectangular openings (one on each side of the larger one) corresponding to the elevator shafts which accommodate three elevators each (see Figure 8.08).

8.3.1. A-CAD

To create an opening in a slab, we have to model the volume corresponding to the void created by the openings which then has to be subtracted from the slab.

In the case of the AWTs, we modelled three volumes corresponding to the three openings and subtracted the same three volumes from all the created slabs. Figure 8.09 illustrates the sequence of actions taken to create the openings.

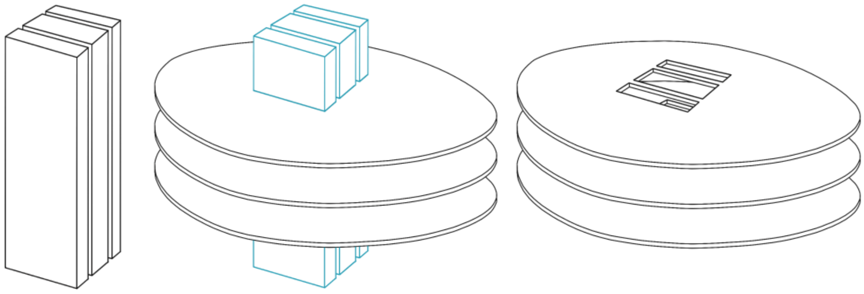


Figure 8.09 – To create the openings, we first modelled the volumes corresponding to the voids created by the openings, placed them in the correct location, and subtracted them from the slabs.

8.3.2. A-BIM

To create an opening, just like with the slabs, we used the pre-defined operation provided by Rosetta that creates openings in slabs. This operation requires a slab in which to place the opening as well as a shape for the opening, defined by a list of points.

Once the shapes of the openings defined, we apply this operation thrice, one time for each opening (see Figure 8.10), to all of the created slabs, using an automated process.

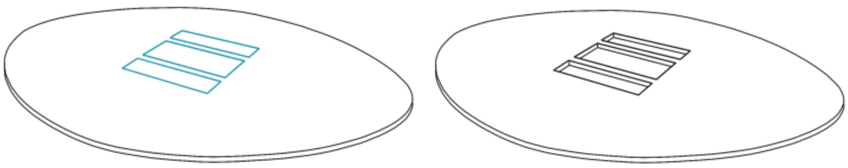


Figure 8.10 – The process of creating all three openings is the same for every slab and can be automated to all the created slabs.

8.3.3. Manual BIM

Using the 'Shaft' opening tool (see Figure 8.11), similarly to A-CAD, we can model the volumes corresponding to the voids created by the openings. However, this time no subtraction is necessary: once created, these volumes automatically affect all slabs that they intersect and create all opening at once.

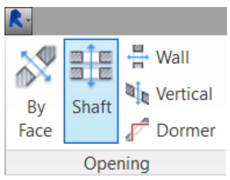


Figure 8.11 – The openings were created using the 'Shaft' opening tool.

8.3.4. Analysis

By comparing a CAD approach with a BIM approach (algorithmic or otherwise), we note that while A-CAD requires an operation of subtraction to create the openings, both manual BIM and A-BIM require no subtraction to do the same. That is because, for BIM, the necessary operation of subtraction is already implicit in the creation of the openings: an opening in a slab is, by definition, a void in the slab. Therefore, by modelling (or shaping) the opening we are, in fact, modelling the void that will be subtracted from the slab. A BIM application knows this and handles the subtraction internally, thus requiring no external input from the user.

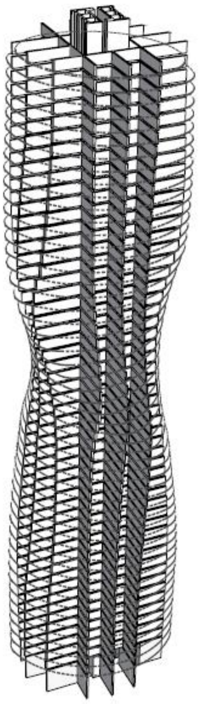


Figure 8.12 – Structural Diagram of the AWTs. The walls adapt themselves to the rotation of the tower in each floor. (source: Lagendijk et. al, 2012)

On a different note, while A-CAD presents a major advantage over a manual CAD approach by automating the creation of the openings in all the slabs, A-BIM doesn't hold the same advantage over manual BIM as some BIM tools, such as Revit, already allow the creation of all openings at once. Thus, although all three approaches handle the creation of the openings differently, all three facilitate the modelling process by creating all openings at once.

8.4 Modelling Process of the Walls

After finishing modelling the slabs, we moved on to the building's structural walls. Both towers have a reinforced concrete structure, composed of the slabs and a grid of load-bearing walls which is repeated in every floor to ensure vertical structural continuity. These walls, because of the sectional fluctuation created by the rotation between floors, lengthen and shorten in order to adapt themselves to the form of the building, as shown in Figure 8.12.

8.4.1. A-CAD

To create a wall, we had to model its geometry. To do that, we traced the line of the wall's axis and used it as a sweep path to create the wall (see Figure 8.13).

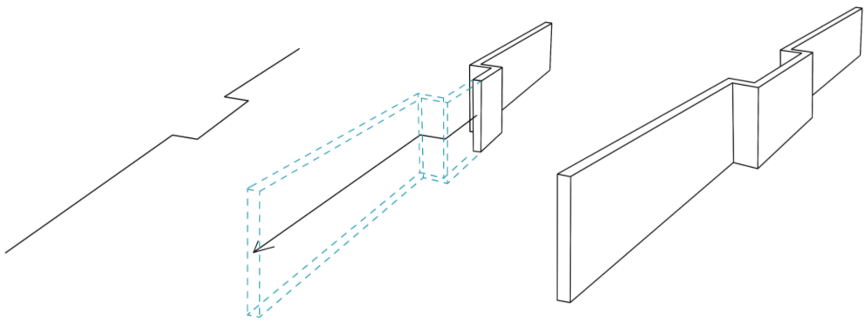


Figure 8.13 – The walls are created by sweeping the profile of the walls along the lines of the walls' axes.

Using this method, we created a grid of walls, placed according to the plan configuration of the AWTs, which is the same for almost every floor (for a reminder of what the plan of a typical AWT floor looks like, refer to Figure 7.04, in page 45). Like with the slabs, this grid of walls was then distributed to all the floors by distributing it along the list of floor heights.

This list of heights is also used to determine the height of the walls in each floor. These are calculated by a procedure that, for any given floor, computes the height of that floor's walls as the distance between the floor in which the walls are placed on and the floor immediately above (see Figure 8.14). This way, if any change is made to the heights of the floors, the heights of the walls immediately adjust to that change.

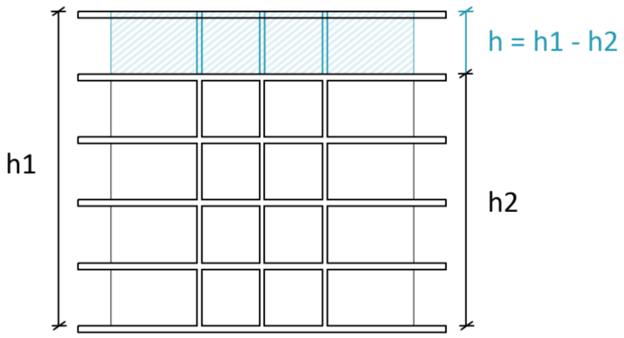


Figure 8.14 – For any given floor, the height of the walls corresponds to the distance between consecutive floors.

In order to simplify the modelling process of the walls, we considered a grid of walls with identical, extended wall lengths for every floor, completely disregarding the sectional fluctuations caused by the rotation of the tower. Because of that, we still had to take into account this fluctuation of the walls and contain them within the interior space of the building. This was achieved with an intersection between the previously created grid of walls and an auxiliary abstract volume we modelled corresponding to the interior space of the building in any given floor – because of the rotation of the tower, this volume is rotated differently in every floor, thus resulting in different intersection results and different wall lengths in every floor. Figure 8.15 shows the process iteratively applied to the walls in each floor in order to contain them within the interior space of the building and adapt the length of the walls to the rotation of the slabs.

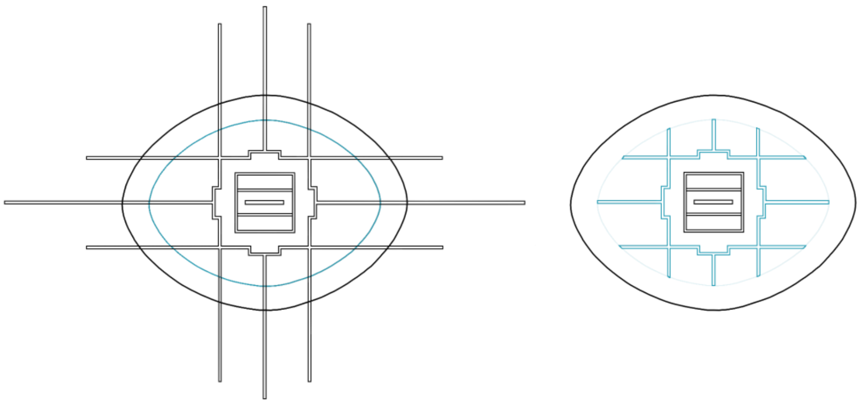


Figure 8.15 – For each floor, the lengths of the walls are adjusted by an intersection between the grid of walls created and an abstract volume corresponding to the interior space of the building.

Finally, to further generalize the form of the building, like with the shape of the slab, the grid of walls that is distributed along the floors and the shape corresponding to the interior space of the building are also given as parameters in order to offer a greater flexibility over the forms that can be obtained.

8.4.2. A-BIM

A wall can be generated using the operation that creates pre-modelled walls provided by Rosetta. This operation requires a list of points – which, when connected in pairs, represent the wall's axis (see Figure 8.16) – to determine the wall's location as well as additional information pertaining to the wall's

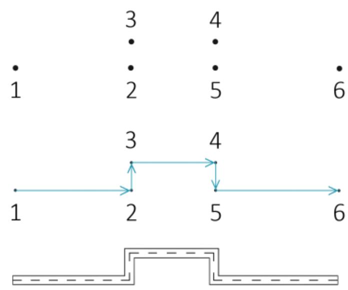


Figure 8.16 – For A-BIM, the path that each wall takes is defined through a list of points that, when connected in pairs, represent the walls' axes.

properties such as thickness, material, angle of inclination of the wall, among others.

Moreover, this operation also requires either a level in which to place the wall and a height or two levels, a bottom level and a top level, in order to define the height of the wall. In the case of the AWTs, we decided on the later: the walls were modelled to be parametrically contained between consecutive levels which means that the height of the walls corresponds to the distance between consecutive levels.

Then, using this operation, a grid of walls was created, placed according to the AWTs plan configuration. This grid was distributed to all the floors by coupling it with the levels: the bottom level of the walls was coupled with a level from the previously created list of levels and the top level of the walls was coupled with the level immediately above the previous one (see Figure 8.17).

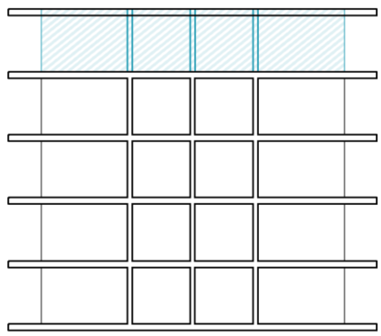


Figure 8.17 – The walls are parametrically contained between consecutive levels, thus establishing their heights and associating them with the levels.

Inspired by the former approach, we also decided to simplify the modelling process of the walls by creating them identical for every floor and then adapting them to each floor’s shape and rotation via intersection. However, while an intersection is a perfectly natural operation for CAD applications, BIM applications do not provide intersections. Nevertheless, we were still able to simulate the effect of an intersection by implementing a procedure that, given a wall and a slab, computes the length of the wall in relation to the perimeter of that slab, not letting it transcend its boundary. Thus, instead of intersecting abstract geometry, we calculate the lengths of the walls in relation to an auxiliary slab corresponding to the interior space of the building in a given floor and adjust the walls accordingly. This process is repeated iteratively for every floor. Afterwards, the created auxiliary slabs are erased.

Finally, like with A-CAD, the grid of walls and the shape corresponding to the interior space of the building are given as parameters in order to make the program more flexible.

8.4.3. Manual BIM

Like with the slabs, we start by selecting the desired wall properties in order to ensure that all subsequent walls created will possess the selected properties.

Then, to create the walls, we chose a level in which to place them (their bottom level), set their top constraint to the level immediately above the former (their top level) and, using the 'Wall' tool (see Figure 8.18), we place the walls according to the desired plan configuration. This process is repeated – or copied – for all the levels in order to create all the walls.

Because the lengths of the walls vary from floor to floor due to the tower's rotation, all the walls had to be manually adapted to each floor's rotation.

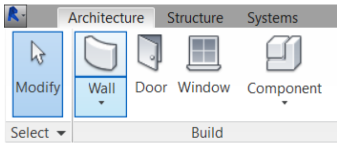


Figure 8.18 – The 'Wall' tool can be used to create walls.

8.4.4. Analysis

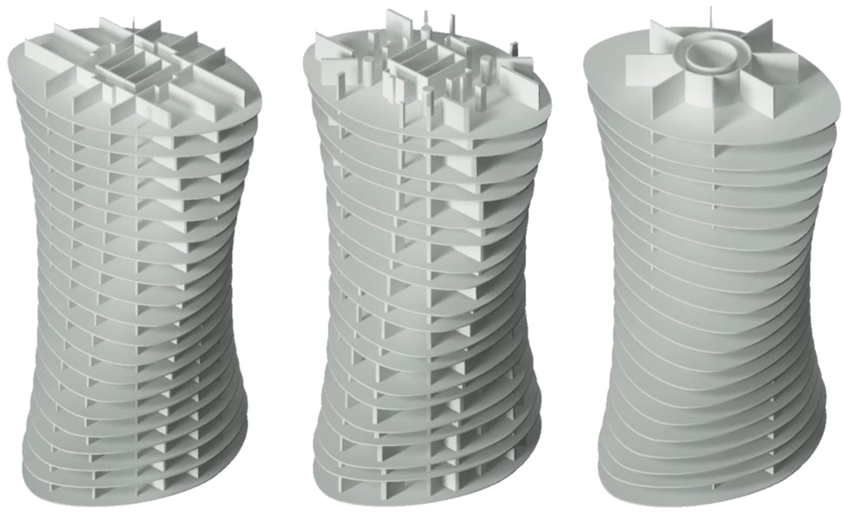
For all three approaches, the walls were created parametrically contained between consecutive floors (which were established through the levels) thus establishing the heights of the walls as the distances between floors. As a result, all three accommodate changes to the heights of the floors and propagate those changes to the walls.

For A-BIM, this association between walls and floors is transferred to the BIM application through the levels which, in this case, serve as parametric constraints to the walls. However, for A-CAD, as mentioned before CAD applications have no concept of levels and thus, when generating the model, this association is lost. In other words, while for A-BIM this association is present in both the program and the generated model, in A-CAD, this association only exists in the program.

Then, like with the slabs, both BIM approaches produce 'walls' instead of abstract geometric entities and these walls contain data properties related to them, such as material composition, thermal resistance, among others.

At the same time, while with manual BIM all walls had to be created and manually adapted to each floor, using an algorithmic approach, because the configuration of the walls is the same for almost every floor, the creation of the walls and their adaptation to the tower's rotation can be automated with a procedure that distributes a grid of walls to all the floors and adapts them to that floor's shape and rotation via an intersection with the shape of the interior space of the building.

Both the grid of walls and the interior shape are given as parameters, thus allowing the quick exploration of different design alternatives for the AWTs and facilitating the process of propagating changes to those parameters. For example, by experimenting with different wall grids, we can change the walls that are distributed to the entire building and automatically adapt them to the form of the building. Likewise, by experimenting with different interior shapes, we can change the shape that adjusts the length of the walls and all walls will be adapted accordingly. Figure 8.19 (following page) shows three different instances of the AWTs that can be obtained by exploring with different parameter values for the walls.



Interior Shape: Ellipse	Interior Shape: Ellipse	Interior Shape: Circle
Grid of Walls: AWTs Grid	Grid of Walls: Alternative Grid 1	Grid of Walls: Alternative Grid 2 - Radial Grid
Consistent Floor Heights: 3,75m per floor.	Alternated Floor Heights: Alternating between 3m and 5m per floor.	Consistent Floor Heights (3,6m) with the exception of the first and last two floors

Figure 8.19 – Three instances of the AWTs that can be generated by exploring different parameters for the interior shape of the building, the grid of walls, and the heights of the levels.

Using a manual BIM approach, we can also propagate some changes to the walls throughout the building though not to the same extent as an algorithmic approach. For instance, by grouping all walls together in Revit, we can change the configuration of the walls in one floor and propagate that change to all remaining floors. However, this propagation would not take into account the association between the lengths of the walls and the form of the building and would still have required all walls to be manually updated to adapt them to each floor's shape and rotation.

Finally, as mentioned in chapter 6, BIM tools have several limitations regarding geometric modelling operations, among which intersections, due to them no longer dealing with abstract geometry. Nonetheless, for A-BIM, we still found this operation to be useful to facilitate the process of adapting the walls to the tower's shape and rotation. Therefore, with regular intersections ruled out, we had to implement a procedure that simulates the effect of an intersection between slabs and walls in a BIM tool. The result is similar to that of a regular intersection but instead modelling abstract geometry, we adjust the length of the walls in relation to the perimeter of an auxiliary slab.

8.5 Modelling Process of the Roof Slab

Once the walls modelled, we covered the building with a roof slab.

8.5.1. A-CAD

To create the roof slab, we used the exact same method used to model regular slabs, i.e. by defining the shape of the roof slab and extruding the resulting surface with a given thickness. The only differences in the implementation of this method were that the roof slab required a greater thickness than a regular slab and that the shape of the roof slab had to be the same as the one that shapes the interior space in the last floor of building in order to appropriately cover the last floor (see Figure 8.20).

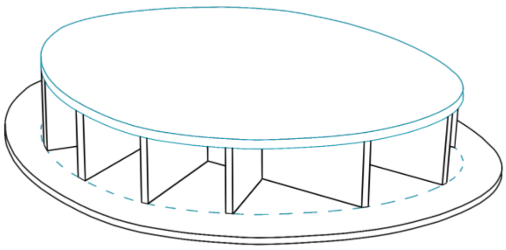


Figure 8.20 – The shape of the roof slab is the same as the one that shapes the interior space, and consequently the walls, of the last floor of the building.

For the AWTs, this translates into a smaller, thicker ellipsoidal slab. On a more generalized perspective, this translates into a dependency between the walls and the roof slab. By creating this dependency, we ensure that the designer can change the interior shape of the building and both the walls and the roof slab will adapt to that change.

8.5.2. A-BIM

To create the roof slab, we used the operation provided by Rosetta that creates roofs which requires a shape for the roof slab, a level in which to place it and the desired properties. In this case, the level corresponds to the last created level in the list of levels and the shape of the slab corresponds to the interior shape of the building.

8.5.3. Manual BIM

To create the roof slab, we selected the last level corresponding to the topmost height of the building and, using the 'Roof' tool (see Figure 8.21), the desired roof shape was drawn. Like with the slabs, we can use the ellipsoidal drawing tool to directly obtain the ellipsoidal shape of the AWTs. Unlike the slabs, on the other hand, we don't necessarily need to choose the roof slab properties before constructing it since we are only creating a single roof slab. Here, we can select the roof slab properties either before or after creating it.

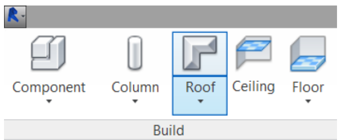


Figure 8.21 – The 'Roof' tool can be used to create the roof slab.

8.5.4. Analysis

When comparing the modelling process of the roof slab with a regular slab (for a quick reminder, refer to sections 8.2) for BIM (algorithmic or manual), we note that both different building components, despite their similar geometry, require different modelling operations to create them. This is a consequence of the fact that these building components are semantically different and possess different properties: for example, a roof should contain several layers of insulation to protect the building from environmental conditions. On the other hand, a slab does not usually require the same level of protection.

By comparing the same modelling process but this time for A-CAD, we note that not only do we use the exact same method to model entirely different building components but also that the resulting objects are virtually identical: abstract geometric entities with similar geometry.

Finally, with an algorithmic approach (either CAD or BIM) it is possible to create a dependency between the shape of the roof slab and the lengths of the walls through the interior shape, thus ensuring that, if any change is applied to this shape, both the roof slab and the walls will adjust to the new shape. With a manual BIM approach, no such dependency exists and a change to the walls would require the roof slab to be manually updated.

8.6 Modelling Process of the Stairs

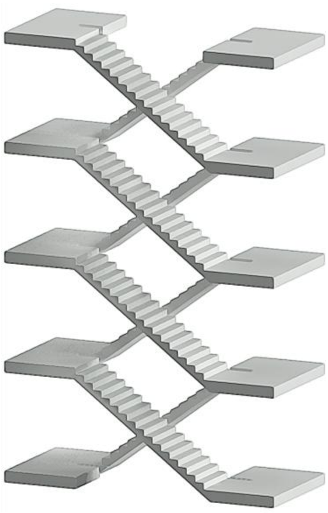


Figure 8.22 – The AWTs have two stairways with alternated flights.

The AWTs have two stairways that are located at the center of the building and have alternated flights, as illustrated in Figure 8.22.

8.6.1. A-CAD

Once again, to create the stairs, we first had to model their geometry. We started by creating the profile of a parametrically constrained flight of stairs, that automatically adjusts itself to each floor's height, containing the top and bottom landing. The created fligtht profile was then converted into a surface and the resulting shape extruded to the desired stair width in order to create a singular flight of stairs with half of the bottom and top landing (see Figure 8.23).

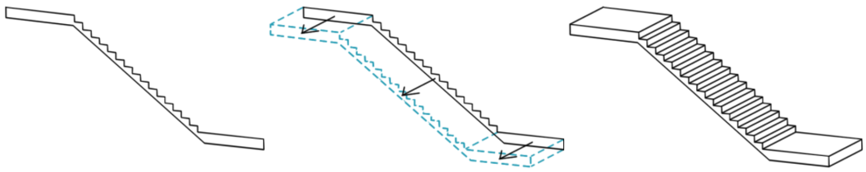


Figure 8.23 – The Stair profile is extruded to create a flight of stairs with half of the bottom and top landing.

Once created, this flight of stairs is distributed iteratively along the heights from the list of heights, while alternating the side and direction of the flight, as illustrated in Figure 8.24.

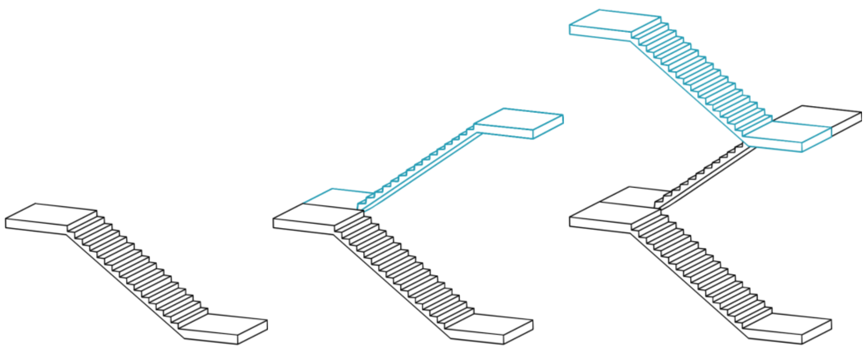


Figure 8.24 – The flights are distributed iteratively along the list of heights, while alternating the side and direction of the flight.

Once the first stairway finished, a second, identical stairway is placed right next to the first one but rotated 180º along the stair's vertical axis (see Figure 8.25). To finish, the four missing halves of the landings in the bottom and top of the stairs (consisting of a small rectangular parallelepiped) are placed in the corresponding location. This process is demonstrated in Figure 8.26.

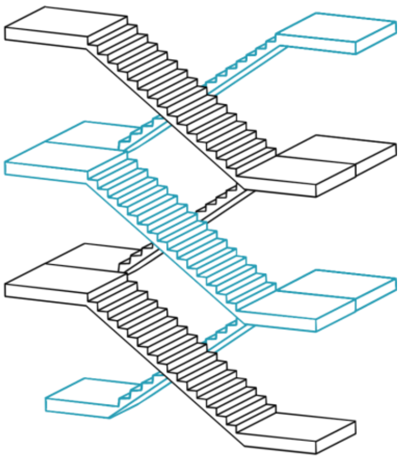


Figure 8.25 – After placing the first stairway, a second stairway is placed next to the first but rotated 180º.

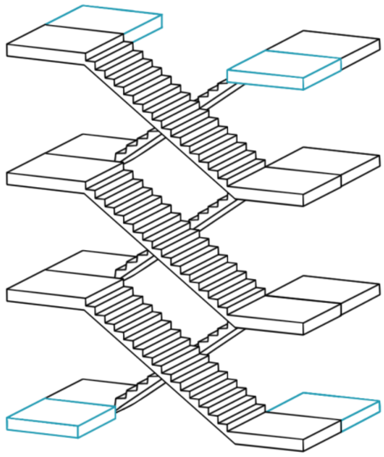
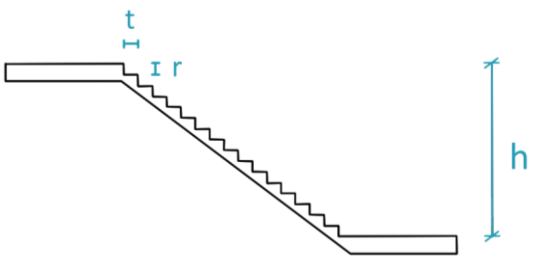


Figure 8.26 – Once the stairs placed, the four missing halves of the landings are placed.

These stairs were designed to automatically adjust to changes to the heights of the floors: for any given height between floors, the program computes the number and dimensions of the steps in the flight of stairs. To do this, we had to implement several parametric rules to restrict the possible outcomes. For example, according to the Portuguese Building Regulation code (i.e. *Regulamento Geral das Edificações Urbanas* (Portugal – Ministério das Obras Públicas, 1951)), a riser should not be smaller that 14cm but also should not be bigger than 19,3cm. Likewise, a tread should not be smaller than 25 cm. In addition, stairs typically follow ergonomic standards, such as the *Blondel Formula*, that dictate the proportions between tread and riser in order to make comfortable stairs. All these rules had to be implemented in order to filter the outcomes and create safe and ergonomic stairs that automatically adjust to the heights of the floors (See Figure 8.27).



PARAMETRIC RESTRICTIONS:

14 cm < Riser (r) < 19,3 cm

25 cm < Tread (t)

Height (h) = r x nº risers

BLONDEL FORMULA:

62 cm < 2 x r + t < 64 cm

Figure 8.27 – Parametric restrictions applied to the stairs in order to create safe and ergonomic stairs that automatically adjust to the heights of the floors.

In the end, by doing this, we only have to provide the floor heights as parameters and the flights of stairs automatically adjust themselves to each floor height.

8.6.2. A-BIM (Implemented in Revit)

Like we mentioned previously, there are some building components that contain features that are too different between Revit and ArchiCAD to be made portable with Rosetta. As we also mentioned, stairs are one of these building components as the creation of stairs in Revit requires several parameters that ArchiCAD does not need thus making it difficult to find one normalized way of creating stairs that works for both Revit and ArchiCAD. For that reason, we had the choice of creating the stairs the Revit way or the ArchiCAD way. In the end, we chose to implement the Revit way so that we could more directly compare A-BIM with the manual BIM approach which was also implemented with Revit.

To create stairs in Revit, we used the pre-defined operation provided by Rosetta that creates flights of stairs in Revit. This operation requires as parameters two points to define the position and direction of the flight of stairs, two levels (the bottom and top level) to establish the height where the flight of stairs starts and ends respectively as well as additional information regarding the type of stairs required, e.g. the family of the stairs, width, among others (see Figure 8.28).

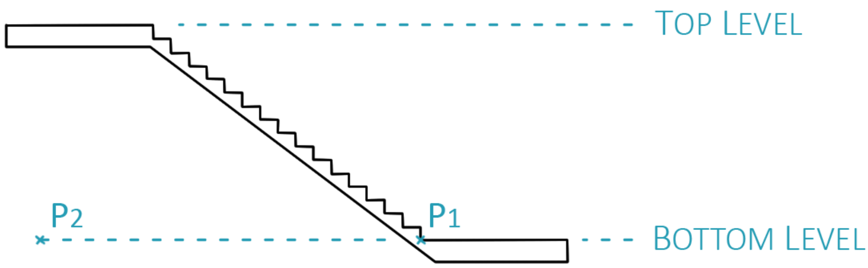


Figure 8.28 – Required parameters to create a flight of stairs with Rosetta.

By associating these flights with consecutive levels from the list of levels, we can distribute them iteratively to all the floors while ensuring that, like with the former approach, the flights automatically adjust to changes to the heights of the levels. However, unlike the former approach, we had no need to implement the parametric rules that dictate the dimensions and proportions of the steps ourselves as these rules are already built-in into the stair component created with the BIM tool.

Finally, when placing several flights of stairs one after another, the landings are created automatically, connecting these flights.

8.6.3. Manual BIM

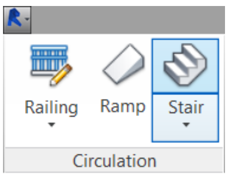


Figure 8.29 – The 'Stair' tool can be used to create the stairs.

Using the 'Stair' tool (see Figure 8.29), the stairs can be created, one flight at a time. Like with the former approach, we can select the bottom and top levels between which we want to put the flight of stairs and place the two points that define the position and direction of the flight respectively. Again, by associating the flights with the levels, we ensure that these stairs automatically adjust to changes to the heights of the levels. Moreover, the landings connecting the flights are also created automatically upon creating the flights.

8.6.4. Analysis

Once again, BIM greatly facilitates the creation process by offering pre-modelled, parametric stairs that can be manipulated to fit the requirements of the project (though this manipulation depends on the type of stairs selected).

However, this time BIM goes further than that. Whereas for CAD we had to implement the parametric restrictions that ensure the appropriate dimensions and proportions for the steps, for BIM, these restrictions are already built-in, thus allowing us to save a lot of time and work in their implementation.

These parametric restrictions also allow us to create a dependency between the flights of stairs and the heights of the floors. Because of that, if any change is made to the heights of the floors, the stairs would automatically adjust to that change. As an example, Figure 8.30 shows three different variations of the stairs, depending on the floor heights provided.

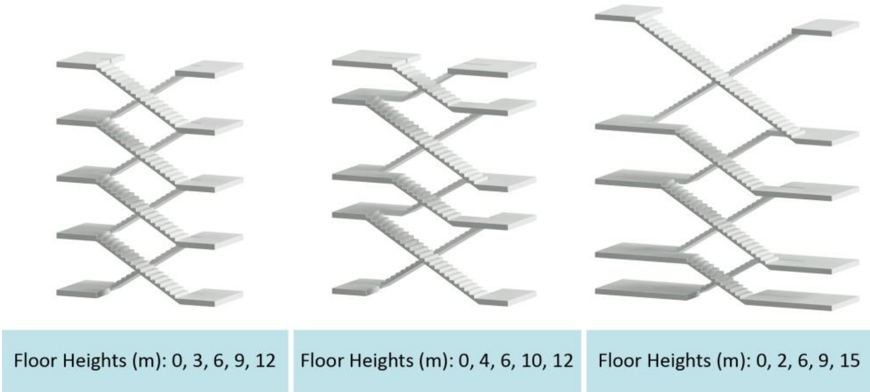


Figure 8.30 – Three instances of the stairs obtained by experimenting with different floor heights.

8.7 Modelling Process of the Doors

8.7.1. A-CAD

To create a door, all of its geometry and subcomponents (e.g. casing, door, handle, …) were modelled separately and from scratch (See Figure 8.31, next page). Depending on the desired type of door and degree of detail, the result can be a lengthy code and take some time to produce.

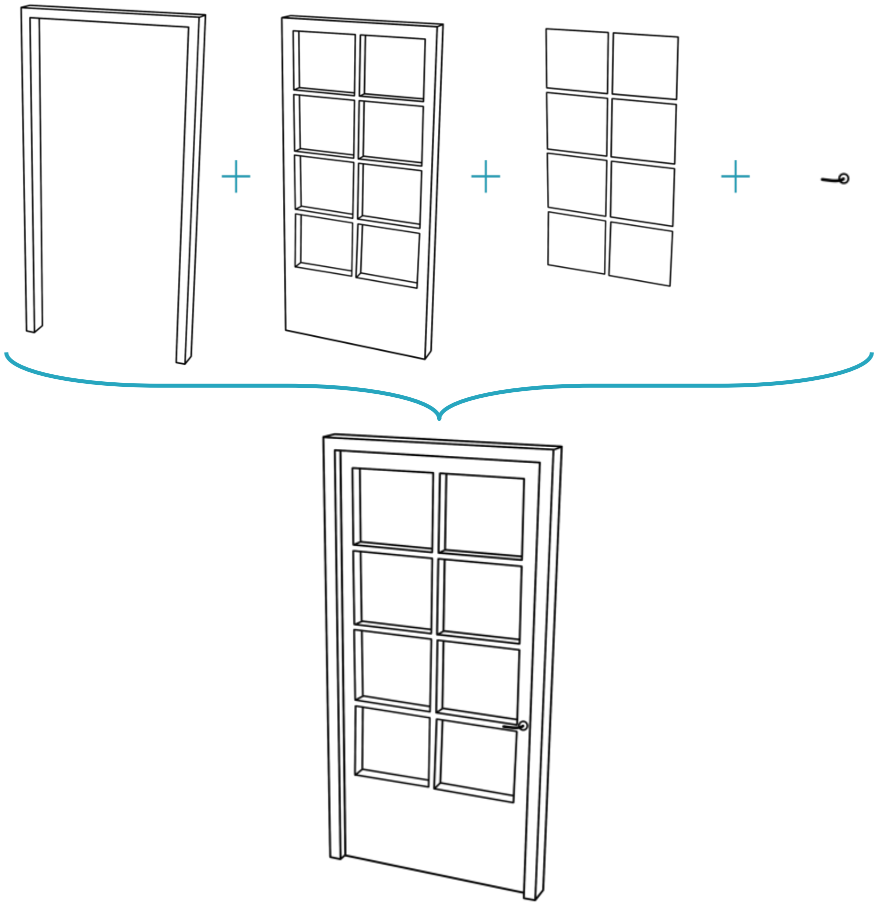


Figure 8.31 – Depending on the type of door wanted, the door can have a lot of subcomponents that need to be separately modelled.

At the same time, the volume corresponding to the opening in the wall for the door also has to be modelled. This volume is then subtracted from the selected wall in order to create the opening for the door and only then can we place the door, as illustrated in Figure 8.32.

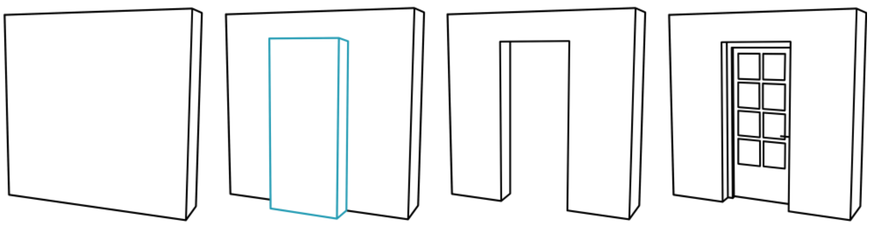


Figure 8.32 – In order to create a door, we must first model the volume corresponding to the opening for the door which we then subtract from the wall. Only then can we place the door.

8.7.2. A-BIM

As stated in chapter 6, when modelling for BIM, doors can only exist hosted in walls. This is reflected in the code as the operation that creates doors requires a host wall as a parameter, along with a location along the wall for the door and the desired type of door. The door can be chosen from the library of pre-modelled doors available in either BIM application and its parameters adjusted to fit the requirements of the project.

Upon creation, the subtraction necessary to create the opening is automatically applied to the wall (see Figure 8.33).

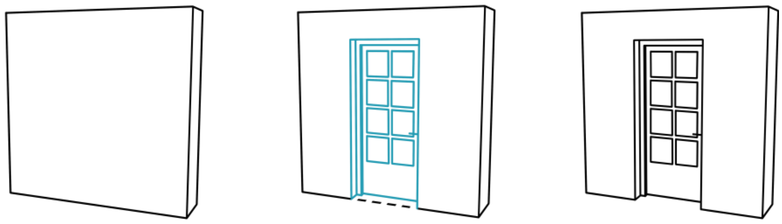


Figure 8.33 – The opening in the wall for the door is automatically created when placing the door in the wall.

8.7.3. Manual BIM

Using the 'Door' tool (see Figure 8.34), we can load the door that we want to use from Revit's library of pre-modelled doors, adjust its parameters to meet the desired requirements, and place it in the chosen wall, in the desired location. Like with the former approach, the necessary subtraction is applied automatically.

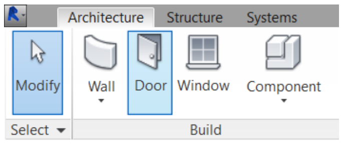


Figure 8.34 – The 'Door' tool available in Revit's toolbar allows the creation of doors.

8.7.4. Analysis

By comparing the three approaches above, we note that one of the greatest advantages of using a BIM approach over a CAD approach in the creation of doors is that we can take advantage of a vast library of pre-modelled doors available with BIM tools from which to choose, thus greatly facilitating their modelling process. These doors are parametric which means that they can accommodate several variations to their parameters (e.g. dimentions, materials, among others) to adapt them to the project's requirements and they can also be modified to display the desired degree of detail in the visualizations.

This library can be further expanded by the availability of several online libraries of free downloadable BIM objects which can be added to the BIM library. Many of the pre-modelled doors available in these libraries correspond to real, commercialized products with an associated manufacturer which means that, at the end of the design process, when all is said and done, they can be ordered and used in the construction of the building.

Finally, in the case that we want to use a door not available in any of these libraries, a new customized door can always be created, either by manipulating the geometry of an existing door or by modelling a new door from scratch, and added to the library for future reuse.

With A-CAD, as mentioned before, we could import external blocks of pre-modelled shapes, also available in online libraries of objects, but these blocks are not usually parametric as the ones available for BIM tools and, in most cases, they would still have to be manually modified to fit the project's requirements. As a result, it is sometimes easier to just create the doors from scratch and make them parametric while adapting them to the degree of details needed.

In the end, we chose to model the door for A-CAD from scratch resulting in substantial differences in the modelling of a door for both algorithmic approaches. As a concrete example, the door on the left in Figure 8.35 required 39 lines of code to be created with A-CAD, as all of its subcomponents had to be modelled, and a single line of code to be created with A-BIM, as we can take advantage of the BIM libraries to produce it. In addition, although the door created with A-CAD is parametric and can accommodate several changes to its dimensions and proportions, changing from the door on the left to the one on the right requires profound changes in the script. With A-BIM, we only need to change the parameter that specifies the type of door.

Figure 8.35 – Two doors that can be selected from BIM libraries with A-BIM but require different algorithms to be created with A-CAD.

As a final point, another advantage of A-BIM over A-CAD stems from the associative rules involved in the creation of a door in a BIM tool. Like with the openings in the slabs, the subtraction necessary to create the opening in the wall for the door is implicit in the creation of the door. As a result, while with A-CAD we had to model the volume corresponding to the opening and apply the subtraction to create it, with A-BIM the subtraction is handled internally and the opening is created automatically upon placing the door.

8.8 Conclusion

In this chapter, we have explained and analised several operations we used in the modelling of the AWTs for our three appproaches. Although there are still many more operations that we could have explained and analyzed, the operations we covered here already provide us with a sufficiently thorough understanding of the differences between the three processes.

9

Evaluation

Algorithmic approaches, such as A-BIM, have a vast applicability in architectural design regarding their use. Although in this thesis we mainly showed the applicability for the design of buildings such as skyscrapers, other studies have explored algoritmic approaches for the exploration of facade design (Caetano, 2015), housing design (Correia, 2013), and urban design (Beirão, 2012), among others.

However, as mentioned before, using A-BIM 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.

In the following sections, we analyze the gains and losses obtained from using A-BIM.

9.1 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, creative 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.

One example is the adaptation of the walls to the form of the building. While using a manual BIM approach we had to manually adjust the length of each wall in each floor in order to adapt them to that floor's shape and rotation, using A-BIM, we were able to implement a procedure that, for each floor, computes the length of the walls and models them automatically adapted to that floor's shape and rotation. As a result, we were able to automate the process of adapting the walls to the building.

This advantage become even more significant when we consider that the designer might want (or need, as is often the case in architecture) to change the design of the tower or the placement of the walls after finishing modelling the walls in the entire building, as we will explain in the following section.

9.2 Propagation of Changes

The model generated with A-BIM is inherently parametric, i.e. it is defined by design rules and relationships between objects. Because the generated objects are parametrically interdependent, changes can be propagated to the entire model.

This ability to propagate changes is not exclusive to A-BIM; BIM tools themselves also offer the ability to propagate changes to a certain extent. However, A-BIM's ability to propagate changes is much more flexible than the one typically available in BIM tools and include, e.g., the ability to generate additional BIM elements.

Remembering the former example, let us consider that the designer just finished modelling the walls in the entire building. Once done, they might not be entirely satisfied with the final result or further interior space planning could require them to relocate the walls to adjust them to the new spatial distribution in the plan.

Whatever the case, using A-BIM, we can simply modify the necessary parameters and generate the tower anew, thus automatically updating the entire model to adjust the new walls to the building. However, with a manual approach to BIM, as seen in section 8.4.4, while we can update the grid of walls in all the floors, adjusting the new walls to the form of the building would still have to be executed manually. In fact, even the smallest alteration to the walls might require manually updating all 50+ floors to adapt them to the form of the building. At that point, the process of handling changes becomes a tedious, repetitive and time-consuming chore that can actually dissuade further changes and discourage design exploration.

Furthermore, this process of manually handling changes is also an error-prone process. This becomes a problem when considering that the complete BIM model must be accurate in order to produce the correct information for construction. An error left unnoticed in the model might be detected much later, during construction and, consequently, resulting in additional corrective costs. For example, an oversight in the updating of the length of the walls would lead to an incorrect bill of quantities extracted from the model and the procurement of the incorrect quantity of materials from product vendors to construct those walls. The late detection of this mistake would then lead to waste or to additional costs to order the remaining materials on short notice.

9.3 Exploration of a Wide Range of Design Alternatives

Once the algorithm formulated and the program properly parametrized, we can easily experiment with the parameters that control the parametric model in order to generate different instances of the design of the towers, including the actual AWTs. This allows a wide range of design alternatives to be quickly explored and visualized without having to redo or modify the algorithm that generates them. However, this greater flexibility of design solutions requires a greater initial effort to make the program flexible which, in turn, requires more time.

In order to test this initial effort, we made an experiment: 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. More specifically, we measured the time and effort required for the implementation of these design alternatives compared to a manual approach.

This simulated process started with the initial modelling of the tower and followed with a series of scenarios where various changes were applied to the initial model of the tower in order to explore different design alternatives. This process was implemented for both A-BIM and the manual BIM approach, thus allowing us to directly compare the times required for the implementation of these changes. Finally, while A-BIM was implemented by us with Rosetta, the manual BIM approach was implemented in Revit by a BIM expert, Naim Korqa.

In the next sections, we describe and evaluate each of these scenarios.

9.3.1. Initial Modelling

We started our experiment by modelling the second AWT, i.e. the one with 50 floors and a consistent rotation. The created model contains the levels, the slabs, the openings in the slabs, the walls, the roof slab and the guardrails, as can be seen in Figure 9.01. The modelling times of the tower in both approaches are presented in Table 9.01.

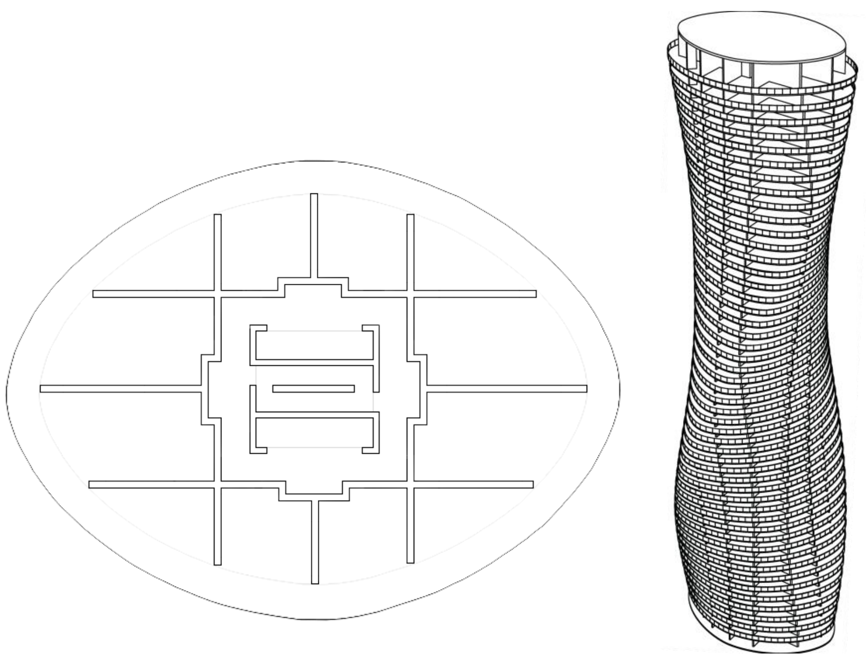


Figure 9.01 – Plan and model of the second AWT.

A-BIM	Manual Approach
14h	2h 14min

Table 9.01 – Time required to produce the model of the second AWT for both A-BIM and the manual approach.

By comparing the times in the table, we note that A-BIM required more time to produce the initial model. That is because it required a greater intellectual effort from the designer: they have to focus on the logic that binds the design together as well as the relationships between parameters and translate these into instructions that can be understood by a computer in a programming language so that they can be executed.

If we were satisfied with this initial model and stopped the exploration here, A-BIM would be less efficient than a manual approach. However, if we take into account future changes and further exploration of design alternatives, this situation starts to change considerably, as we will see in the following sections.

9.3.2. Scenario 1: Changing all floors to a rectangular shape

Taking our initial model, we changed the shape of the floors to a rectangular shape, as can be seen in Figure 9.02. This change required all slabs (including the roof slab), all guardrails and all walls to be updated to the new shape of the building.

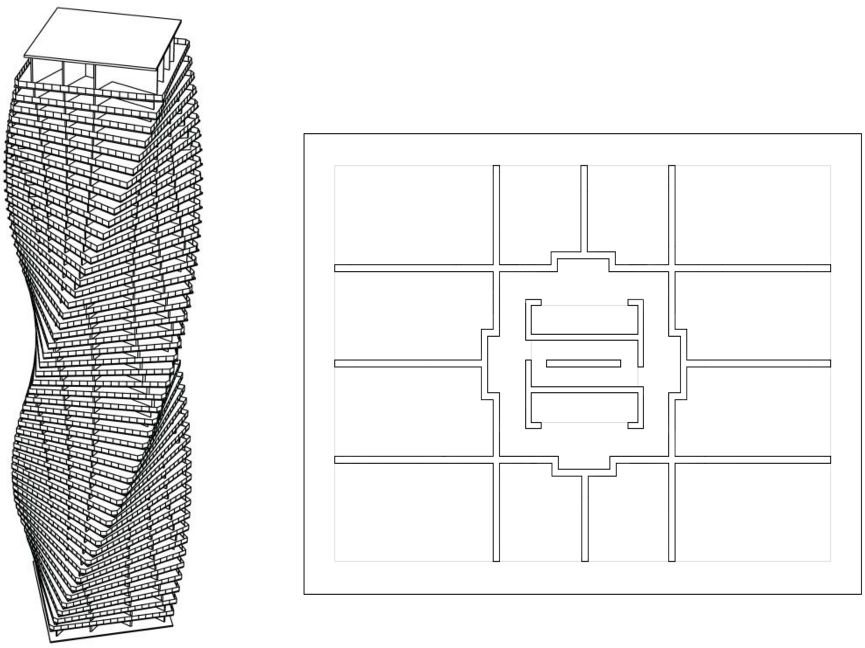


Figure 9.02 – Model and plan of the tower with rectangular floors.

Because of the dependencies created by the parametric model, for A-BIM, changing the shape of the floors implies simply modifying the parameter of the shape of the slabs as well as the parameter of the interior shape of the building. These modifications are then propagated to all objects dependent on these parameters, including the walls and the guardrails.

As for manual BIM, the same change required all slabs and walls to be manually updated to the new shape. Despite what we said before about BIM tools being able to propagate changes to the shape of the slabs by grouping them together, in this particular case, we had to ungroup the slabs in the initial model in order to create the opening in the slabs thus losing the ability to propagate that change. As a result, we had to recreate and rotate the slabs all over again, including the roof slab. Once the slabs created anew, we were able to group them together again and create all guardrails at once. In the end, we had to ungroup the new slabs again in order to create the new openings in the slabs. As for the walls, they had to be all manually adjusted to the new shape, for all floors of the building.

For both approaches, the times needed to apply these changes are displayed in Table 9.02. For A-BIM, between deciding that we want to apply a change to the model and obtaining the updated model, we have to modify the necessary parameters and generate the model anew in order to visualize the final result. Both these tasks take time to be executed. However, in these scenarios, we are not taking into account the generation time of the model for two reasons: (1) while the model is being generated, the computer requires no input from the designer and the latter is free to work on other tasks during that time; and (2) the generation time can change drastically depending on several factors, many of which unpredictable and/or liable to change (e.g. processing power of the computer, application where the model is being generated on, among others). Thus, the following table only displays the time needed to make the modifications to the parameters:

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

Table 9.02 – Time required to change all floors to a rectangular shape for both A-BIM and a manual approach.

In a reverse of the situation of the initial modelling, a manual approach requires more time to apply the changes. That is because, despite the parametric capabilities of BIM, many changes still had to be manually applied to the entire model, which takes time. On the other hand, for A-BIM, we only had to change the necessary parameters and, because the model is dictated by parametric rules and dependencies, this caused that change to be propagated to the entire model.

9.3.3. Scenario 2: Increasing the scale of the floors along the building

Again, taking the inital model as a starting point, we increased the scale of the floors along the building, as shown in Figure 9.03. This change required updating the slabs, the roof slab, the guardrails and the walls.

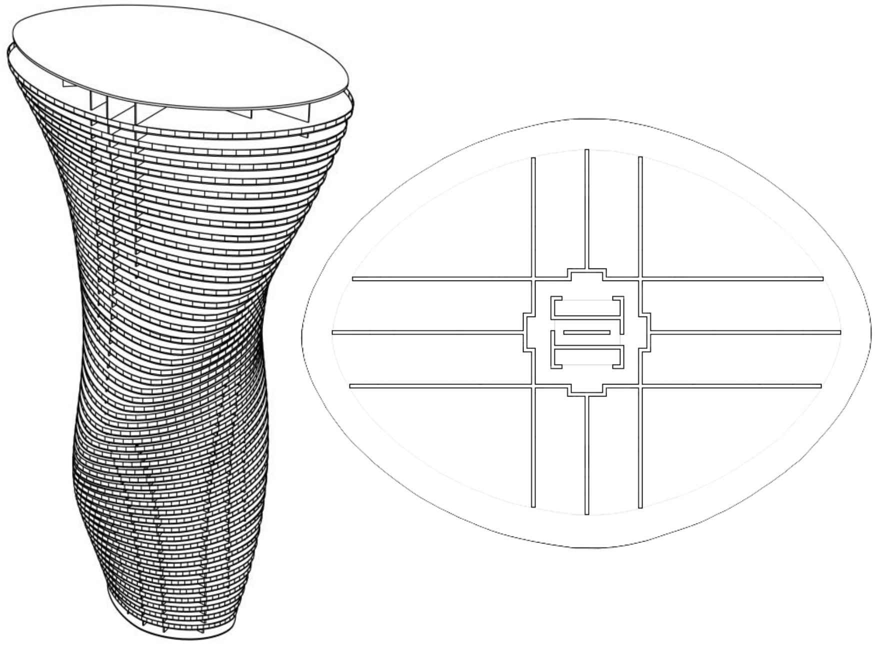


Figure 9.03 – Model and plan of the tower with an increasing floor scale.

For A-BIM, this change required applying an increasing factor to the radiuses of the ellipsoidal shapes of both the slabs and the interior shape of the building. Again, these changes were then propagated to the affected objects.

With a manual approach, like the former scenario, we had to redo, rotate and apply a scaling factor to all slabs individually, including the roof slab. Then, because scaling is not a change that can be propagated with groups, all guardrails had to be redrawn. Additionally, all walls had to be manually adjusted to the new interior shape.

The resulting times are as follows:

A-BIM	MANUAL APPROACH
1min 20s	1h 52min

Table 9.03 – Time required to increase the scale of all the floors for both A-BIM and a manual approach.

Again, this change takes a lot longer to be applied using a manual approach than a programming approach. In fact, for a manual approach, this change takes almost as long as recreating the model from the start. That is because many objects had to be recreated from scratch, while others had to be manually updated to the new change, resulting in a repetitive and time-consuming process.

9.3.4. Scenario 3: Changing all floors to an hexagonal shape and the rotation of the tower

In this scenario, we changed the shape of the floors to an hexagonal shape as well as the rotation of the tower to a decreasing angle of rotation along the building, as is illustrated in Figure 9.04. Once again, this change required updating the slabs, the roof slab, the guardrails and the walls

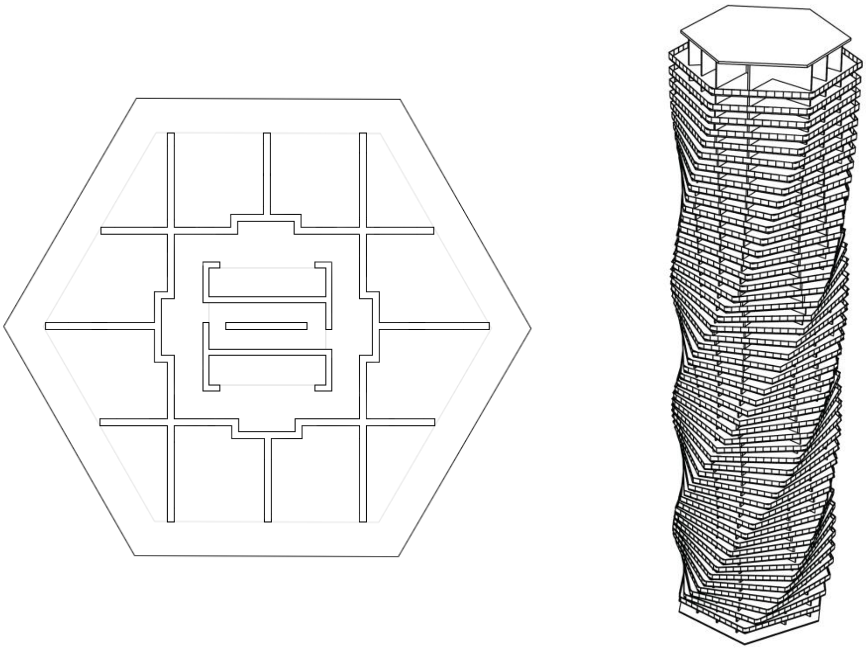


Figure 9.04 – Plan and model of the tower with hexagonal floors and a decreasing angle of rotation.

To apply this change with A-BIM, we had to modify the parameters that define the shape of the slabs and the shape of the interior of the building as well as the list of angles that defines rotation of the tower. These changes were then propagated to the rest of the model.

As for a manual approach, we had to recreate the slabs and the roof slab with the new shape and rotate them individually with their new angle of rotation. Once the slabs created, we were once again able to use groups to create all guardrails at once but had to undo the group again in order to create the openings. Again, all walls had to be manually adjusted to both the new shape and rotation of the tower.

The following table shows the times required to execute these changes:

A-BIM	Manual Approach
2min 52s	1h 37min

Table 9.04 – Time required to change all floors to an hexagonal shape and decrease the angle of rotation of the tower for both A-BIM and a manual approach.

Once again, A-BIM ability to propagate changes greatly facilitates the exploration of a different design solution, as they only have to modify the necessary parameters and generate the model anew. With a manual approach, most of the changes required to explore this design alternative had to be applied manually, thus consuming time and effort from the designer.

9.3.5. Scenario 4: Changing the grid of walls and the angle of rotation of the tower

Finally, in this last scenario, we changed the grid of walls to a radial grid as well as the angle of rotation of the tower from 4º to 6º of rotation per floor, as shown in Figure 9.05. This change required updating the slabs, the opening in the slabs, the roof slab, the guardrails and the walls.

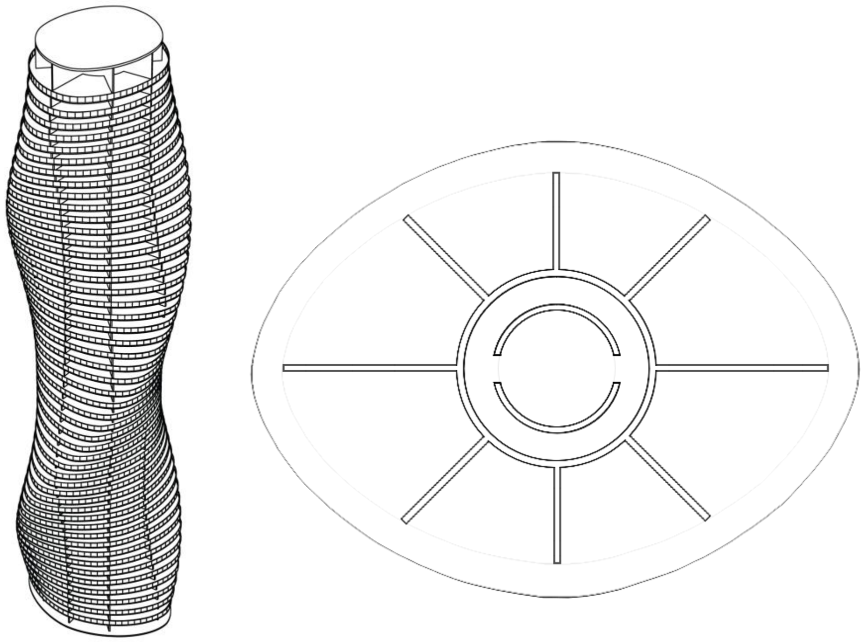


Figure 9.05 – Model and plan of the tower with a radial grid as well as a consistent 6º of rotation per floor.

For A-BIM, this change required changing the parameters of the grid of walls, and the shape of the opening as well as the list of angles that defines rotation of the tower. The grid of walls was the parameter that took longer to change as a new grid had to be defined from scratch. These changes were then propagated to the rest of the model.

For the manual approach, we also had to redraw the new grid of walls and, and manually update them to all the floors. Then, we had to apply the new rotation to all slabs and to the roof slab and, by grouping all slabs together, we were able to recreate all guardrails at once, with the new tower rotation. As for the openings in the slabs, we had to ungroup the slabs and remodel the volume corresponding to the void created by the openings and this volume allowed us to create all openings at once.

Table 9.05 shows the times required to apply these changes:

A-BIM	Manual Approach
7min 33s	**2h 46min**

Table 9.05 – Time required to change the grid of walls and the angle of rotation of the tower in both A-BIM and a manual approach.

Even with changes that take longer to be applied as new geometry has to be created from scratch, A-BIM is still more efficient than a manual approach when it comes to the time and effort spent on applying changes and exploring different design alternatives. This is mostly due to A-BIM's ability to both automate parts of the modelling process and propagate changes in the model.

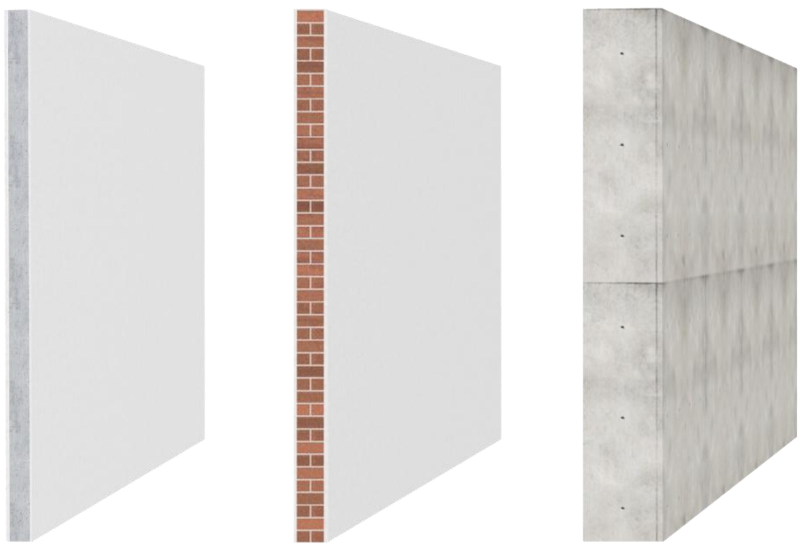
9.3.6. Conclusion

As stated before, if we had a very clear idea from the start of what we wanted to model, the means to efficiently do it manually, and were satisfied with the initial model, then A-BIM would not be an efficient approach to use. However, the design activity is rarely that straightforward and, typically, the design is constantly evolving. Frequent changes to the model are usually required and manually handling these changes is a cumbersome task that often discourages further changes. This process is only aggravated as the model grows in complexity, increasing the number of objects to be updated with these changes and the time and effort needed to do it. In these cases, A-BIM becomes an important paradigm that can significantly improve the design process.

A-BIM allows a wide range of design alternatives to be quickly explored by varying the parameters that control the model. This, in turn, encourages change and greatly promotes design exploration during the design process thus making A-BIM ideal for projects where extensive exploration is a requirement and/or frequent changes are anticipated.

9.4 Building Information

Another advantage of A-BIM is that, as a BIM model, the generated model is infused with building information and data. Each different building component in the model is appropriately identified and possesses the architectural properties and data related to that component. For instance, Figure 9.06 shows a wall containing different analytical properties depending on its material composition.



Titanium Stud Partition Wall	Brick Wall with Plaster Finish	Concrete Wall
Materials: Gypsum Plasterboard (thickness: 20 mm) Titanium Frame with air space (thickness: 80 mm) Gypsum Plasterboard (thickness: 20 mm)	Materials: Gypsum Plaster Finish (thickness: 20 mm) Brick (thickness: 110 mm) Gypsum Plaster Finish (thickness: 20 mm)	Materials: Concrete (thickness: 300 mm)
Analytical properties: Heat Transfer Coefficient (U): 0.2731 W/(m_2*K) Thermal Resistance (R): 3.6615 (m_2*K)/W Thermal Mass: 3.44 kJ/K	Analytical properties: Heat Transfer Coefficient (U): 3.7701 W/(m_2*K) Thermal Resistance (R): 0.2652 (m_2*K)/W Thermal Mass: 16.74 kJ/K	Analytical properties: Heat Transfer Coefficient (U): 3.4867 W/(m_2*K) Thermal Resistance (R): 0.2868 (m_2*K)/W Thermal Mass: 42.12 kJ/K

Figure 9.06 – Material and analytical properties contained in three different variations of the same wall.

This building information can also be consulted and extracted at any point during the design process and used to produce additional information, including:

- Consistent two-dimensional architectural drawings, namely plans for all 50+ floors of the towers, sections and elevations;
- Renderings (see Figure 9.07);
- Spreadsheets of material quantities;
- Data for performance evaluations.

Figure 9.07 – Rendering of the Absolute World Towers, produced with ArchiCAD.

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, considering two of the five variations of the tower explored in the previous section, we can use the generated models to produce thermal analyses of both design solutions. In Figure 9.08 (next page), we have the results of a thermal radiation analysis produced with DIVA, an environmental performance analysis plug-in developed for Rhino (Grynberg, 1989), for both the initial model and the variation of the tower in the first scenario. Alternatively, in table 9.08 (also next page), we have the results of an energy simulation analysis executed with Revit also for both design solutions.

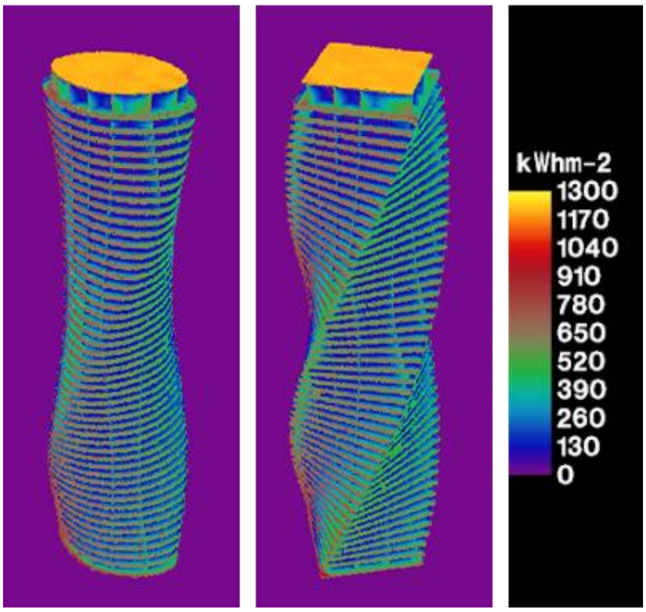


Figure 9.08 – Results of a thermal radiation analysis produced with DIVA. The image shows the variations of the thermal radiation in the initial model (left) and in the variation of the first scenario (right).

	Initial Model	Scenario 1
Energy, Carbon and Cost Summary		
Annual Energy Cost	16,371 $	4,991 $
Lifecycle Cost	222,970 $	67,975 $
Annual CO_2 Emissions		
Electric	42.1 Mg	11.6 Mg
Onsite Fuel	1.2 MG	2.5 MG
Large SUV Equivalent	4.3 SUVs/ Year	1.4 SUVs/ Year
Annual Energy		
Energy Use Intensity	222,239 MJ/m^2/Year	6,329 MJ/m^2/Year
Electric	112,081 kWh	30,953 kWh
Fuel	24,143 MJ	49,496 MJ
Annual Peak Demand	43.9 kW	11.4 kW
Lifecycle Energy		
Electric	3,362,418 kW	928,576 kW
Fuel	724,292 MJ	1,484,871 MJ

Table 9.06 – Results of an energy simulation analysis executed with Revit for both the initial model and the variation of the first scenario.

9.5 Optimization

By analyzing and comparing several design alternatives through the information produced with the models, we can guide the design to a solution that better fits the established requirements. In fact, by taking advantage of A-BIM's algorithmic capabilities, this process of analyzing and comparing design alternatives can then be automated in order to let the computer find even better solutions.

For example, we could let the computer test and compare different design alternatives in order to find out what shape and rotation the slabs should have to optimize the buildings aerodynamic performance. In the future, we plan to further explore this topic.

9.6 Parametric and Associative Capabilities

With A-BIM, by using an algorithmic approach with BIM tools, the algorithm and the BIM application 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 a greater solution space to be explored throughout the design process, as discussed in section 9.3.

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.

9.7 Libraries of Building Components

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

These libraries can be expanded through online libraries that offer free downloadable BIM objects to be used (As an example, the National BIM library at http://www.nationalbimlibrary.com provides a vast range of free downloadable products, from floor construction solutions to sanitary accessories). Many of these products are freely provided by product manufacturers, resulting in accurate BIM representations of existing, commercialized products. The manufacturers themselves have good reasons to provide their goods as BIM objects to these libraries and keeping them updated: in most cases, the objects the designers use in their BIM model end up being the ones they order for the construction of the real building.

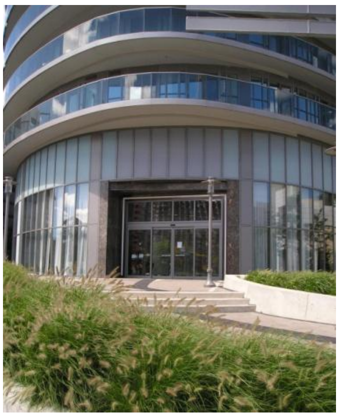


Figure 9.09 – Curved curtain wall in the ground floor of the AWTs (source: www.randyselzer.com/)

On a more negative note, these libraries of pre-modelled building components can also restrict what can be built in BIM tools. The reason for this is that these building components mostly correspond to standard building components used in standard projects, i.e. frequently used design and construction solutions. An atypical construction solution, such as, for example, the curved curtain wall for the window frames of the AWTs (see Figure 9.09), 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 be able to add the new building component to the library thus enabling the reuse of the newly created customized object in future projects.

9.8 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. This is a result of the fact that BIM tools mostly deal with building components instead of just geometry. Although BIM tools are still capable of producing simple geometric objects like former CAD tools, doing so diminishes the advantages of using the building components, e.g. the building information.

One reason for this restricted manipulation of the geometry of building components is that BIM tools themselves are usually more sensitive to what can be built. However, this also greatly depends on the limitation of the BIM tool itself. The BIM tools we used to test A-BIM in this thesis, Revit and ArchiCAD, are more focused on standard construction solutions and, thus, have more limitations in what can be built. Other tools, such as Microstation, do not rely exclusively on a pre-defined library of components, thus making it easier for users to create their own unique components (Aish, 2003).

Another reason for this limitation is that many of the previously needed modelling operations have been made mostly redundant by the building components which already handle some of these operations internally, as explained in section 8.3.4 and 8.7.4 with the openings in the slabs and the doors.

Despite these limitations, with A-BIM we are still able to implement procedures that can simulate the effect of geometric modelling operations, as was the case with the intersection that we simulated to adapt the walls to the form of the building (refer to section 8.4.4).

CONCLUSION

Digital technology has been affecting the design culture over the past years. In particular, Algorithmic Design (AD) and Building Information Modelling (BIM), both important developments brought by digital technologies, have been affecting and improving how architects design, thus gaining significant interest from the architectural community in recent years.

In this thesis, we propose a new approach to design, one that combines AD with the BIM methodology and that can offer important benefits to architectural design. We call this approach A-BIM, acronym for Algorithmic-based Building Information Modelling.

A-BIM offers a challenging but flexible new way of designing, one based on algorithmic logic. Due to its algorithmic origin, repetitive modelling processes can be automated and the resulting algorithm can be highly parametric, allowing a wide range of design alternatives to be quickly explored and evaluated by experimenting with different parameter values. These changes to the parameters are 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, created by assembling building components infused with building information and constrained by parametric and associative rules. These building components 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. That is because A-BIM requires not only a priori knowledge of programming but also an initial investment to formulate the algorithm that generates the model which, for some projects, might not be entirely recoverable.

As demonstrated in this thesis with the case study of the AWTs, the design of highly repetitive buildings such as skyscrapers can benefit from A-BIM. For the AWTs, the initial investment can be quickly recovered as the project evolves and frequent changes become a necessity.

Due to its flexible nature, A-BIM is also an approach that greatly promotes design exploration. In architecture, the design activity thrives on exploration and thus can greatly benefit from this approach.

FUTURE WORK

In the future, we will expand our research to further explore the capabilities of A-BIM for the Architecture, Engineering, and Construction industries. Below, we present some topics we plan to explore with A-BIM:

1 **GOING BEYOND THE ARCHITECTURAL PROJECT |** The scope of this thesis only covers part of the architectural project. In the future, we want to continue modelling the AWTs through the following stages of the design process, including going beyond the architectural project and exploring other design disciplines, e.g. structural analysis.

2 **EVALUATE A-BIM IN A PROJECT-DRIVEN ENVIRONMENT |** In this work, we modelled an existing building as a case study to help us evaluate A-BIM in a simulated design process. However, in a real project, the conditions are very different as there are multiple participants involved from different design disciplines and the design is constantly evolving. Therefore, it would be useful to evaluate A-BIM in a project-driven environment where different members of the design team develop the program that generates the model concurrently and the design evolves as it goes.

3 **FIND PROJECT MANAGEMENT STRATEGIES |** Following the former suggestion, if A-BIM is ever evaluated in a project-driven environment, project management strategies would be needed to manage the design process. As one possible example, we want to test the efficacy of version control tools (e.g. Tichy, 1985) as a mean to aid the coordination between the multiple participants involved in the project and manage the program's increasing complexity.

4 **TESTING A-BIM WITH OTHER BIM TOOLS |** In this thesis, we tested A-BIM with two BIM tools commonly used in the architectural community: Revit and ArchiCAD. However, there are other existing BIM tools, e.g. Digital Project and Microstation, and these tools might deal with BIM differently. As a result, they might also respond to A-BIM differently. Thus, it would be interesting to test A-BIM with other BIM tools.

5 **EXPLORE OPTIMIZATION PROCESSES |** While in this thesis we gave a couple of suggestions on possible optimization processes that we could execute with the model of the tower, it would be interesting to actually be able to execute them in order to evaluate how they perform with A-BIM.

CONTRIBUTIONS

This thesis proves the usefulness of A-BIM in architectural practices, specifically in the design of highly repetitive buildings such as skyscapers and in projects where a high level of exploration is required. In both cases, the initial investment required by an algorithmic approach is recovered when the possibility of experimenting with multiple different design solutions is not overburden by the need to handle changes in the whole project.

In order to use A-BIM, a programming approach different from the one needed for geometry-based CAD is needed. Thus, in this thesis we offer a set of guidelines for a programming methodology fit to BIM.

This thesis also provides a comparative study of three different but related design approaches in order to find out the benefits and drawbacks of each one: (1) an Algorithmic approach to geometry-based CAD; (2) an Algorithmic-based Building Information Modelling approach with the programming methodology that we propose; and (3) a manual BIM approach.

Finally, some of the results produced by this thesis were published in a paper titled *Portable Generative Design for Building Information Modelling* (Feist et. al., 2016), presented at the CAADRIA 2016 conference, the 21st international conference on Computer-Aided Architectural Design Research In Asia.

Aish, R. (2003) *Bentley's GenerativeComponents: a Design Tool for Exploratory Architecture* [Online], Bentley Systems Inc. Available at: http://195.250.185.245:8080/MyWeb/get/Bentley_training/GenerativeComponents.pdf [Accessed in January 2015].

Aish, R. and Woodbury, R. (2005) Multi-level interaction in parametric design, In: *International Symposium on Smart Graphics*, Berlin: Springer Berlin Heidelberg, pp. 151-162.

Alfaris, A. (2009) *Emergence Through conflict the Multi-disciplinary Design system*, PhD, Massachusetts Institute of Technology.

Autodesk (2014) *European Parliament Directive to Spur BIM Adoption in 28 EU Countries,* [Online] Available at: http://inthefold.autodesk.com/in_the_fold/2014/01/european-parliament-directive-to-spur-bim-adoption-in-28-eu-countries.html [Accessed in October 2015].

Autodesk (2015) *The Dynamo Primer,* [Online] Available at: http://www.dynamoprimer.com/ [Accessed in November 2015].

Beirão, J. N. (2012) *CItyMaker: Designing Grammars for Urban Design*, PhD, Delft University of Technology.

Bergin, M. S. (2011) *A brief history of BIM,* [Online] Available at: http://www.architectureresearchlab.com/arl/2011/08/21/bim-history/ [Accessed in July 2015].

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.

Building Smart (2014) *Technical Vision,* [Online] Available at: http://www.buildingsmart.org/standards/technical-vision/ [Accessed in October 2015].

Buro Happold Engineering, (2016) *Louvre Abu Dhabi – Structural Excellence,* [Online] Available at: http://www.burohappold.com/technical/louvre-abu-dhabi-structural-excellence/ [Accessed in January 2016].

Burry, M. (2014) "Cultural Defense" in Oman, R., Oxman, R. (Eds.) *Theories of the Digital in Architecture,* New York: Routledge, Cap. 26, pp. 383-406.

Caetano, I. (2015) *DrAFT: An algorithmic Framework for Façade Design*, M.Sc, Instituto Superior Técnico.

Campbell, D. A. (2006) *Modeling Rules,* [Online] Available at: http://www.architectureweek.com/2006/1115/tools_2-1.html [Accessed in June 2015].

Carpo, M. (2014) "Foreword" in Garber, R., *BIM Design: Realising the Creative Potential of Building Information Modelling*, (Vol. 2), Chichester, UK; John Wiley & Sons, Foreword, pp. 8-13.

Correia, R. C. (2013) *Designa – A Shape Grammar Interpreter*, M.Sc, Instituto Superior Técnico.

Das, J., Leng, L. E., Lee, P., Kiat, C. (2011) *Build Smart: The BIM Issue*, Singapore; Building and Construction Authority.

Eastman, C. M. (1975) "The Use of Computers instead of Drawings in Building Design", *Journal of the American Institute of Architects*, March, pp. 46-50.

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.

Fasoulaki, E. (2008) *Integrated Design: A generative Multi-Performative Design approach*, M.Sc, Massachusetts Institute of Technology.

Feist, S., Barreto, G., Ferreira, B., Leitão, A. (2016) Portable Generative Design for Building Information Modeling, In: *CAADRIA 2016 conference* [Online], Melbourne: The University of Melbourne, pp. 147-156. Available at: http://papers.cumincad.org/data/works/att/caadria2016_147.pdf [Accessed in May 2016].

Fernandes, R. (2013) *Generative Design: a New Stage in the Design Process*, M.Sc, Instituto Superior Técnico.

Ferreira, B., Leitão, A. (2015) Generative Design for Building Information Modeling, In: *eCAADe Conference* [Online], Vienna: Vienna University of Technology, pp. 635-644. Available at: http://cumincad.scix.net/data/works/att/ecaade2015_118.content.pdf [Accessed in May 2016].

Flanagan, J. (2015) *Technical Report: An Exploration and Examination of 2D & 3D detailing in Virtual Building Software*, B.Sc, Mayo Institute of Technology.

Garber, R. (2014) *BIM Design: Realising the Creative Potential of Building Information Modelling,* (Vol. 2), Chichester, UK; John Wiley & Sons.

Graphisoft (2015) *GRAPHISOFT introduces Rhino–Grasshopper–ARCHICAD connection,* [Online] Available at: http://www.graphisoft.com/info/news/press_releases/graphisoft-introduces-rhino-grasshopper-archicad-connection.html [Accessed in November 2015].

Grynberg, A. (1989) "Validation of Radiance", Lawrence Berkeley Laboratory, Berkeley LBID, 1575.

Hudson, R. (2010) *Strategies for Parametric Design in Architecture; An application of practice led research*, PhD, University of Bath.

Imbert, F., Frost, K. S., Fisher, A., Witt, A., Tourre, V., Koren, B. (2012) Concurrent Geometric, Structural and Environmental Design: Louvre Abu Dhabi, In: *Advances in Architectural Geometry 2012* [Online], Wien: Springer-Verlag, pp. 77–90. Available at: http://atelier-imbert.fr/resources/LAD_paper+V7.pdf [Accessed in February 2015].

Johnson, B. R. (2014) "One BIM to Rule Them All" in Kensek, K., Noble, D. (Eds.) *Building Information Modeling: BIM in Current and Future Practice,* New Jersey; John Wiley & Sons, Cap. 13, 175-184.

Kalay, Y. (2004) *Architecture's New Media: Principles, Theories and Methods of Computer-Aided Design*, Massachusetts; The MIT Press.

Kalay, Y. (2006) "The impact of information technology on design methods, products and practices", *Design studies*, Vol. 27, no. 3, pp. 357-380.

Kensek, K., Noble, D. (2014) *Building Information Modeling: BIM in Current and Future Practice,* New Jersey; John Wiley & Sons.

Khemlani, L. (2012) *Around the World with BIM,* [Online] Available at: http://www.aecbytes.com/feature/2012/Global-BIM.html/ [Accessed in August 2015].

Kolarevic, B. (2003) *Architecture in the Digital Age: Design and Manufacturing*, New York; Spon Press.

Kolarevic, B. (2003) Computing the Performative in Architecture, In: *eCAADe Conference,* Graz: Graz University of Technology, pp. 457-464.

Korqa, N. (2015) *Generative Design for BIM: Its influence in the design process*, M.Sc, Instituto Superior Técnico.

Kotnik, T. (2010) "Digital Architectural Design as Exploration of Computable Functions", *International Journal of Architectural Computing*, Volume 08, no. 1, pp. 1-16.

Lagendijk, B., Vadlotto, S, Pignetti, A. (2012) "Case Study: Absolute World Towers, Mississauga", *CTBUH Journal*, Issue IV, pp. 12-17.

Leitão, A. (2012) *Programação para Arquitectura*, Lisboa; Instituto Superior Técnico.

Leitao, A. and Santos, J., (2012) "Programming Languages for Generative Design: a Comparative Study", *International Journal of Architectural Computing,* Volume 10, no. 1, pp. 139-162.

Leitão, A., Caetano, I., Correia, H. (2016) "Processing Architecture", *International Journal of Architectural Computing,* 1478077116638927.

Logan, T. (2014) *Superb Lyrebird,* [Online] Available at: http://lmnarchitects.com/tech-studio/bim/superb-lyrebird/ [Accessed in November 2015].

Lopes, J. (2012) *Modern Programming for Generative Design*, M.Sc, Instituto Superior Técnico.

Lopes, J. and Leitão, A. (2011) Portable generative design for CAD applications, In: *ACADIA 2011* [Online], Banff: ACADIA 2011 196–203. Available at: http://cumincad.scix.net/data/works/att/acadia11_196.content.pdf [Accessed in April 2016].

McCormack, J., Dorin, A. and Innocent, T. (2004) *Generative design: a paradigm for design research*, In: DRS Melbourne 2004 Conference [Online], Melbourne: Design Research Society. Available at: http://www.csse.monash.edu.au/~jonmc/research/Papers/genDesignFG04.pdf [Accessed in October 2015].

Mitchell, W. (1990) *The Logic of Architecture: Design, Computation, and Cognition*, Massachusetts; MIT press.

Mitchell, W. (1990) Afterword: The design studio of the future, In: *CAAD futures 1989* [Online], Boston: The MIT Press, pp. 479-494. Available at: http://cumincad.scix.net/data/works/att/435a.content.pdf [Accessed in July 2015].

Mitchell, W. (1995) CAD as a Social Process, In: CAAD futures 1995 [Online], Singapore: National University of Singapore, pp. 7-9. Available at: http://papers.cumincad.org/data/works/att/f766.content.pdf [Accessed in December 2015].

Mitchell, W. (1999) "A Tale of Two Cities: Architecture and the Digital Revolution", *Science*, Vol. 285, pp. 839-841.

Mitchell, W. (2004) "Constructing Complexity in the Digital age", *Science*, Vol. 303, pp. 1472-1473.

Mitchell, W. (2007) "Built from Scratch", *MIT Technology review*, December, pp. 13.

Oxman, R. (2006a), "Editorial: Special Issue of Design Studies on Digital Design", *Design Studies*, Vol. 27, pp. 225-227.

Oxman, R. (2006b), "Theory and Design in the First Digital Age", *Design Studies*, Vol. 27, pp. 229-265.

Oxman, R. (2008), "Digital architecture as the challenge for Design Pedagogy", *Design Studies*, Vol. 29, pp. 99-120.

Oxman, R. (2012) "Novel Concepts in Digital Design" in Gu, N., Wang, X. (Eds.) *Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education*, United States; Information Science reference, Cap. 2, pp. 18-33.

Oxman, R., Oxman, R. (2014) *Theories of the Digital in Architecture*, New York; Routledge.

Peters, B., & De Kestelier, X. (2006) The Work of Foster and Partners Specialist Modelling Group. In: *Bridges London Conference* [Online], London: University of London, pp. 9-

12. Available at: http://archive.bridgesmathart.org/2006/bridges2006-9.pdf [Accessed in February 2016].

Portugal – Ministério das Obras Públicas (1951) *Regulamento Geral das Edificações Urbanas*, Diário do Governo nº 166/1951, 1º Suplemento, Série I de 1951-08-07, [Online] Available at: https://dre.pt/application/file/289115 [Accessed in May 2016].

Qian, C. (2007) Design patterns: Augmenting user intention in parametric design systems, In: *Conference on Creativity & Cognition*, Washington: Association for Computing, pp. 295.

Richens, P. (1997) "OXSYS-BDS Building Design Systems", *Bulletin of Computer-Aided Architectural Design*, Vol. 25, pp. 20-44.

Rothenthal, G. n.d., *Testimonials and Quotes,* [Online] Available at: http://fsharp.org/testimonials/#goswin-1 [Accessed in February 2016].

Santos, L., Lopes, J., Leitão, A. (2012) Collaborative Digital Design: When the architects meets the software engineer, In:, *eCAADe Conference* [Online], Prague: Technical University in Prague, pp. 87-96. Available at: http://cumincad.scix.net/data/works/att/ecaade2012_225.content.pdf [Accessed in May 2016].

Scheible, F., & Dimcic, M. (2011) Parametric Engineering: Everything is Possible, In: *2011 IASS Annual Symposium* [Online], London: IABSE-IASS Symposium. Available at: http://www.programmingarchitecture.com/publications/ScheibleDimcic_IASS_2011.pdf [Accessed in January 2016].

Scheurer, F. (2014) "Materializing Complexity" in Oxman, R., Oxman, R. (Eds.) *Theories of the Digital in Architecture,* New York: Routledge, Cap. 18, pp. 287-291.

Shepherd, P., Hudson, R. and Hines, D (2011) "Aviva Stadium: A Case Study in Integrated Parametric Design", *International Journal of Architectural Computing*, Volume 09, no. 2, pp. 187- 204.

Singh, V., & Gu, N. (2012) "Towards an integrated generative design framework", *Design Studies*, Vol. 33, no. 2, pp. 185-207.

Smith, P. (2014) "BIM Implementation – Global Strategies", *Procedia Engineering*, Vol. 85, pp. 482-492.

Taborda, P., Cachadinha, N. (2012) "BIM nas Obras Públicas em Portugal: Condicionantes para uma Implementação com Sucesso", In: *Congresso Construção 2012* [Online], Coimbra: *ITeCONS,* pp. 1-14. Available at: http://run.unl.pt/bitstream/10362/10012/1/ID21.pdf [Accessed in January 2015].

Terzidis, K. (2003) *Expressive Form: A Conceptual Approach to Computational Design*, London; Spon Press.

Thomas, D. (2009) *Introducing RevitPythonShell,* [Online] Available at: http://www.darenatwork.blogspot.pt/2009/12/introducing-revitpythonshell.html [Accessed in November 2015].

Tichy, W. F. (1985) "RCS—a system for version control", *Software: Practice and Experience,* Vol. 15, no. 7, pp. 637-654.

Venâncio, M. J. (2015) *Avaliação da Implementação de BIM – Building Information Modeling em Portugal*, MSc, Universidade do Porto.

Whitehead, H. (2003) "Laws of Form" in Kolarevic, B. (Ed.) *Architecture in the Digital Age: Design and Manufacturing*, New York; Spon Press, Cap. 7, pp. 82-100.

Wojtkiewicz, S. (2014), Generative Systems in architecture design, In: XVII Generative Art Conference (GA2014) [Online], Milan: Generative Design Lab, pp. 334-346. Available at: http://www.generativeart.com/ [Accessed in September 2015].

Wong, A. K. D., Wong, F. K., & Nadeem, A. (2009) Comparative roles of major stakeholders for the implementation of BIM in various countries [Online], Hong Kong: Polytechnic University. Available at:

https://www.academia.edu/671734/Comparative_Roles_of_Major_Stakeholders_for_the_Implementation_of_BIM_in_Various_Countries [Accessed in September 2015].

Wong, A. K., Wong, F. K., & Nadeem, A. (2011) "Government roles in implementing building information modelling systems: Comparison between Hong Kong and the United States", *Construction Innovation*, Vol. 11, no. 1, pp. 61-76.

Woodbury, R. (2014) "How Designers Use Parameters" in Oman, R., Oxman, R. (Eds.) *Theories of the Digital in Architecture,* New York: Routledge, Cap. 10, pp. 153-170.