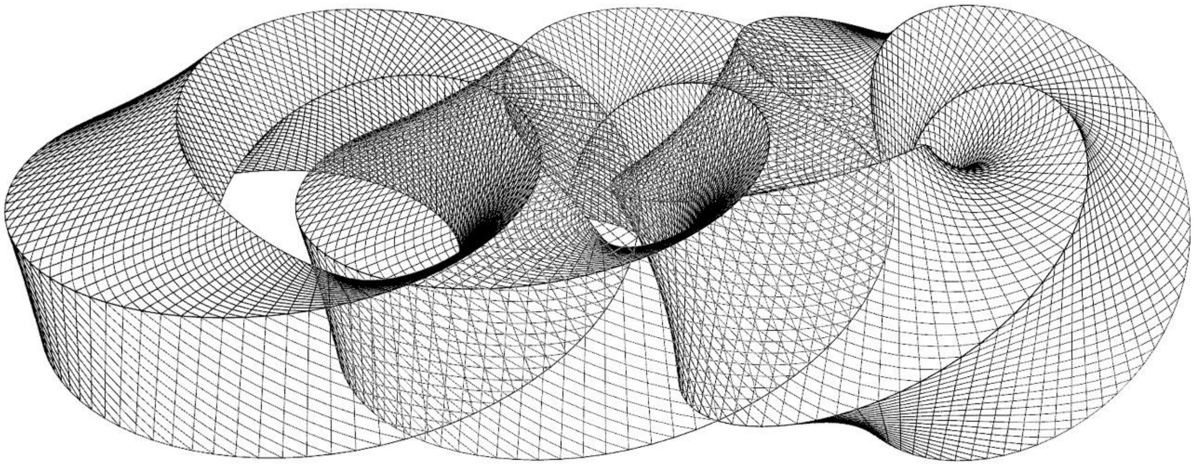


Integrated Algorithmic Design

Renata Alves Castelo Branco

Thesis to obtain the Master of Science Degree in

Architecture

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

Examination Committee

Chairperson: Prof. Dr. Ana Paula Filipe Tomé
Supervisor: Prof. Dr. António Paulo Teles de Menezes Correia Leitão
Member of the Committee: Prof. Dr. Francisco Manuel Caldeira Pinto Teixeira Bastos

May 2017

ACKNOWLEDGMENTS

My supervisor accuses me of being a poet. A wonderful quality to have if you're a writer, yet not so much if you aspire to be an investigator. With that said, this is the only page in this document that will not be evaluated. Hence, I chose to let the poetry in me take over. I wish to thank all those who helped me in this process, with all the adjectives and metaphors I held back throughout this document!

My deepest and greatest gratitude goes to Professor António Menezes Leitão, for exceeding his condition as supervisor, becoming more than a teacher, a role model and a constant source of inspiration. I thank him extensively, for the unconditional support, ridiculous availability and disposition to help, for his tireless creative spirit, his unbelievable work ethic and his unconditional dedication to investigation.

A warm acknowledgment to the members of ADA. This research group turned what might have been a lonesome task of a year, into a cooperative endeavour that improved each one of our works individually, and gave them all a sense of purpose, beyond the sole production of official documents.

Among the ADA team I must emphasize Guilherme Barreto. Not only would my work not have been possible without the one he developed, but I also would not have gotten as far, if not for the endless amount of time, he always seemed to have available to explain the smallest things to me. I must also state my great appreciation for Duarte Ramos. No longer a member of our team, yet he succoured me in those dreadful last days with a brilliant mixture of friendliness and professionalism.

Last, but not least, I thank my family and friends for their unlimited support and faith in me. For constantly reading and correcting documents they might not have fully liked or understood. For caring about me and my work and for making my existence a joyful event to live through. I am most grateful to Diogo for enduring all the stress and panic I conveyed during this process, giving back nothing but calm and solace.

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

ABSTRACT

Many great architectural endeavors today engage in a multi software approach, as each party develops their respective part of the project in a different software. Moreover, the architectural project itself covers many tasks, including 3D modeling, analysis, and rendering, which benefit from the use of different tools. Combining them in the same project involves the sharing and crossing of the various information systems, which is not always a successful process. A mechanism is needed that connects all the different tools used, in a more effective manner - a portability mechanism. Algorithmic Design (AD) presents itself as a potential solution.

AD is an algorithmic approach to architectural design that allows architects to transcend factory-set limitations of the currently used 3D software. As mathematical descriptions are oblivious to any software, the algorithmic descriptions of the designs become independent from the software that might be used to produce them.

This thesis aims to explore the advantages an algorithmic approach can bring to the design process, and investigate, at the same time, how to bridge the gap between the different tools with which architecture currently operates. We propose a methodology based on an algorithmic approach to design, where a single program can describe not only the intended model, but also additional tasks, such as model analysis. We call this approach, Integrated Algorithmic Design (IAD) and using it, the architect can take advantage of various CAD, BIM and analysis tools, with little effort when it comes to the transition between them.

Keywords: Algorithmic Design; CAD; BIM; Analysis tools

RESUMO

Grandes empreendimentos de arquitectura utilizam, actualmente, uma série de ferramentas no seu processo de desenvolvimento, uma vez que cada uma das partes envolvidas utiliza um software diferente para desenvolver a sua especialidade. Para além disto, o projecto de arquitectura em si engloba uma série de tarefas, como a modelação 3D, análises e produção de renders, para as quais são também necessárias diferentes ferramentas. A combinação de todas elas num projecto obriga à partilha e tradução dos vários sistemas de informação utilizados, processo que nem sempre é bem sucedido. É necessário um mecanismo capaz de fazer uma conexão mais eficaz entre as diversas ferramentas usadas. O design algorítmico apresenta-nos uma possível solução para este problema.

O design algorítmico é uma abordagem ao projecto que permite aos arquitectos ir além da manipulação manual de modelos tridimensionais e transcender as limitações impostas pelos softwares de modelação. Visto que descrições matemáticas são alheias a qualquer software, descrições algorítmicas de designs tornam-se também elas independentes do software usado para as produzir.

Esta tese tem como objetivo explorar as vantagens que uma abordagem algorítmica pode trazer ao processo de design e investigar, simultaneamente, como podemos ultrapassar as diferenças entre as diversas ferramentas com as quais os arquitectos trabalham. Propomos uma metodologia baseada no desenvolvimento de um único programa que, descreve não só o modelo pretendido, mas também outras tarefas adicionais como a análise do mesmo. Chamamos a este método design algorítmico integrado e, com dele, o arquitecto pode usufruir das ferramentas de CAD, BIM e análise, sendo o esforço necessário para transitar entre elas muito reduzido.

Palavras-Chave: Design Algorítmico; CAD; BIM; Ferramentas de Análise

CONTRIBUTIONS

During the development of this master thesis, three scientific articles were published:

- Castelo Branco, R. and Leitão, A. (2017) 'Integrated Algorithmic Design: A single-script approach for multiple design tasks', eCAADe (Education and Research in Computer Aided Architectural Design in Europe), Sapienza University, Rome.

- Castelo Branco, R. and Leitão, A. (2017) 'Translating Algorithmic Design from CAD to BIM', KineSIStem - From Nature to Architectural Matter, International Conference, ISCTE University, Lisbon, 181-190.

- Leitão, A., Castelo Branco, R. and Cardoso, C. (2017) 'Algorithmic-Based Analysis - Design and Analysis in a Multi Backend Generative Tool', P. Janssen, P. Loh, A. Raonic, M. A. Schnabel (eds.), Protocols, Flows, and Glitches - Proceedings of the 22nd CAADRIA Conference, Xi'an Jiaotong-Liverpool University, Suzhou, China, 137-146.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABREVIATIONS

2D – Two-dimensional

3D – Three-dimensional

AD – Algorithmic Design

AEC – Architecture, Engineering and Construction

ANL - Astana National Library

API - Application Programming Interface

BIM – Building Information Modeling

CAD – Computer-Aided Design

IAD – Integrated Algorithmic Design

GLOSSARY OF TERMS

Algorithm – Sets of rules, which translate instructions given by a human (to be performed by a computer, in this context), in order to solve a problem.

Algorithmic Design – The creation of architectural designs through algorithmic descriptions. Using AD, the architect does not build the digital model, but instead, builds the program that builds the digital model.

Parameters/Variables – A property of a program that when modified produces different design results.

Program – An unambiguous formal description of an algorithm, that is, a set of rigorous instructions, written in a way that the computer understands, i.e. a programming language, that tell the computer what specific steps to perform.

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

INTRODUCTION

The design process of an architectural creation has seen many changes over time, and more so over the past few decades. Representation methods are amongst the ones that shifted the most (Kalay, 2004). After centuries of producing precise technical drawings, perspectives, and models by hand, architects found these tasks facilitated by Computer-Aided Design (CAD) software. Designed to help creative users in an "interative performance of 'man-machine' problem solving engine" (Llach, 2013: 18), CAD tools support sketching, drafting, and image-altering, as well as 3D modeling and rendering (Brandon and McLain-Kark, 2001). Further along came the Building Information Modeling (BIM) paradigm, hailed as one of the most promising developments in architecture, engineering, and construction (AEC) industries, as it pledges to bring the three closer together in a more integrated design and construction process (Eastman et al., 2008). Parallel to these advances, an entirely different manner of conceiving architecture has been pressing forward: Algorithmic Design (AD). Algorithmic Design (Figure 0.1) emerges in an era led by technology and computation, and aspires to change the design paradigm.

Figure 0.1 - Generative and parametric modeling explorations from Formakers (source: http://www.formakers.eu/project-320-angel-quintana-parametric-architecture-and-design)

Focusing on the change from hand-drawing to CAD modeling, one might still consider it a smooth shift, as CAD tools maintained the architect's drawing board ideal, albeit replacing the pen and the ruler for more accurate digital tools. Following the same logic, one might relate 3D modeling in CAD to building scaled models by hand. Nevertheless, it is still up to the architect to cast his own design method: with CAD applications, he can design in 2D and decide to model a 3D for rendering purposes, for instance, or not model one at all; or he may start modeling in 3D from the start and produce 2D drawings from the model thereafter.

The BIM paradigm (Figure 0.2), however, greatly differs from the established CAD archetype and introduces a more substantial leap. These programs possess 2D modeling environments as well, but for a correct and full use of the paradigm, they require the user to model an accurate virtual 3D model of his design. Technical drawings like plans and sections can be automatically generated from the 3D model, which means they are updated whenever it undergoes changes. Moreover, BIM embeds the model with data needed to

Figure 0.2 - BIM scheme for a building's lifecycle (source: http://www.advancedsolutions.com/design/services/lifecycle-bim.html)

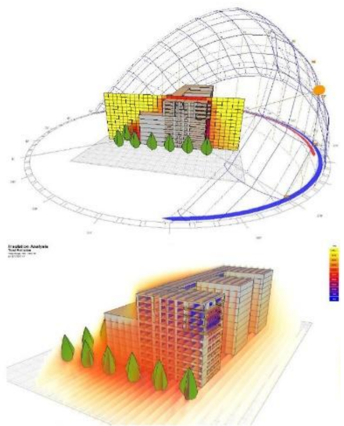


Figure 0.3 - Solar Radiation Analysis using ecotect (source: http://legacy.iaacblog.com/maa2011-2012-digitaltools/2011/12/ecotect-analysis/)

support construction, fabrication, and procurement activities (Eastman et al., 2008). Unlike a model built in CAD that contains only the modeled geometry, a BIM model contains information about the construction elements used, such as materials and the respective quantities and costs.

Despite the new methodologies introduced by these systems, architects still struggled to fully engage in a thoroughly digital process. A few decades ago, modeling tools were often incorporated only into later stages of the design process, where the production of precise and detailed construction drawings and specification documentation usually took place (Brandon and McLain-Kark, 2001). Mark Burry addressed the issue in a rather interesting manner: "I had thought naively that the 'D' in CAD stood for design, not drafting, which is how the software seemed to have been prioritized to me" (Burry, 2011: 28). Currently, the stakes are changing, partly due to Algorithmic Design. In order for modeling tools to become more than documental assistants and actually play a deeper part in the design process, architects must be allowed to utilize them with the same authority they might apply to a pen or a compass. AD offers such a possibility. With the use of Algorithmic Design the user is able to transcend the limitations the software might impose on him (Terzidis, 2006), and make use of the modeling tools in his own way.

Another important group of software that has set its ground in the design processes of the architectural agenda is the array of analysis tools (Figure 0.3). Performance analysis of buildings has long been part of the engineering discipline, although for decades it has been done through tiresome and error-prone manual calculations. Recently, these calculations have been implemented in specialized computational tools that perform them automatically using data extracted directly from 3D models, reducing not only the amount of time spent, but also possible human errors in the calculations. These tools brought forth a new concept: Performance-based design – a design process informed by a deeper analysis and understanding of the environmental context of the project (Oxman and Oxman, 2014).

MOTIVATION

Never before have there been so many or so diverse tools, techniques and methods for design. Architects are spoilt for choice as their practice becomes a liquid discipline, pouring into other domains like mathematics, computer science, robotics, manufacturing and more. Post-digital design has become a task of curious manipulation, speculation and experiment. This means the resulting product is, more than ever, influenced by how it is designed, that is, by the methods and tools used in the process.

The influence of the design tool on the design process, and the array of choices presented, leads the architect into a rather delicate choice. Different tools fit different purposes and the production of an architectural creation may, not only have several distinct objectives to start from, but is also likely to change them along the way. Furthermore, different stages of the design-to-production process present very different workflows and computational needs.

Amongst the more important software developments, we have highlighted CAD and BIM software and analysis tools. Each one of them possess a set of different advantages to the design process that cannot be left out. Combined, they may just cover the most important operations an architect needs, to see his design through to completion with the state-of-art methods. We believe architects should be allowed to take advantage of these tools in what they do best in their design process.

In a way, this is already possible. Current architectural studios are incorporating many of these tools in their design process. Nevertheless, the majority of those cases rely on an import-export system of communication between the tools. The solution is far from ideal, as this not only fails to be a continuous process, but some information may also get lost in the translation process. In complex models with rather great investments in them, this becomes a considerable problem.

A seamless incorporation of the different paradigms and workflows, on the other hand, would allow architects to take advantage of all their potential in one single working environment, with no need for imports or exports. In our opinion, an ideal architectural design process should imply the use of CAD, BIM and various analysis tools, in a workflow capable of incorporating and coordinating these different realities. Algorithmic Design offers the unique possibility to achieve such a desirable workflow. Designing algorithmically, the architect could, then, take advantage of each of these tools, using and combining their assets as he would see fit.

OBJECTIVES

The current architectural design process already makes use of different paradigms and tools, but suffering from portability issues along the way, with information getting lost in imports and exports. Our goal is to merge some of the most relevant paradigms and tools in a seamless workable process that architects can follow.

We propose an integrated algorithmic approach to design that aims to cover relevant aspects of a project. The thesis rests on the production of one single algorithmic description of the design - a program, changing and evolving through the phases of the design process and containing all required information, from design to production, presentation or construction. We call this, Integrated Algorithmic Design (IAD).

In order to cover all these stages, we propose to integrate three main categories of tools into the algorithmic modeling process: (1) CAD and (2) BIM, as modeling paradigms and visualization backgrounds, and (3) analysis tools, with particular emphasis on daylight analysis, as fundamental sources of data to use in the design process.

The proposed methodology allows architects to model and explore their design ideas, using a programming tool that supports portable Algorithmic Design, shifting from the CAD to the BIM paradigm as they see fit according to their own design workflow. Moreover, they can incorporate analysis of the design in any stage of the process as well. The IAD approach entails the liberty to explore the integrated tools, in an algorithmic process controlled by the designer. Furthermore, it leaves the door open for the integration of other tools or paradigms that architects may find, in time, relevant to the design process.

METHODOLOGY

The methodology we followed to achieve the appointed objectives is divided into four main phases: (1) literature review, (2) explanation of the Integrated Algorithmic Design approach, (3) application of the methodology to an existing case study, and (4) conclusions.

The first phase of the thesis, the literature review, summarizes an extensive research on the various topics that motivated the Integrated Algorithmic Design approach, and explains key concepts to allow its understanding. The historical context of design tools in architectural practice is outlined, and within it, the appearance of both CAD and BIM tools. The two paradigms are compared in the light of their impacts in the design process. Following a similar logic, the appearance of Algorithmic Design is explained and its relevance and potential to the design process is analyzed. The introduction of analysis tools to the design process is also scrutinized through an investigation of the recently rediscovered concepts of performative architecture and performance-based design, with emphasis on environmental performance. Finally, a review of the currently available algorithmic design tools that allow portability amongst different software is presented.

The second part of the document thoroughly explains IAD's methodology, an integrated algorithmic approach to design. We layout the methodology in separate phases and explain each one as steps architects can follow to apply it.

In the third part, we take an existing architectural project and follow the IAD methodology to model it from the beginning to the end, proposing some variations to the original design in the process. The application of the methodology implies the algorithmic modeling of the design for both CAD and BIM applications and the integration of analysis results in the design process. A thorough description is made to explain how the case study was modeled and the advantages found in the use of this methodology.

Lastly, in the fourth phase of this thesis, we review the methodology presented and summarize the evaluation made of the advantages and disadvantages it poses to the design process. We conclude our work with final considerations and the plantation of the seeds for future work.

STRUCTURE

This thesis is divided into two main parts: Background and Integrated Algorithmic Design. To these main chapters, the Introduction, Conclusion and Bibliography sections were added.

The first part – **Background** - is divided into 4 chapters:

1. Representation Methods | In this chapter, the history of representation methods in architecture is briefly reviewed, with emphasis on the appearance of Computer-Aided Design and Building Information Modeling. A short section is dedicated to other types of software tools used that were not intentionally developed for the use in architectural practice. A comparison is made between the two main paradigms mentioned before regarding their advantages to the design process.

2. Algorithmic Design | This chapter begins with an analysis of the untapped potential of the modeling tools presented in the previous chapter. Next, we define Algorithmic Design and the advantages of its use in the design process, when allied to those modeling tools. We explain core concepts that architects need to master in order to design algorithmically, such as programming, algorithms and parametric modeling. Finally, we evaluate the use of Algorithmic Design in current architectural practice.

3. Performative Architecture | In this chapter we take on the last group of tools that integrate the IAD approach, analysis tools, with primary focus on energy analysis tools. To this end we begin by defining the concept that motivated the development of these tools - performative architecture. We focus our research on environmental design and briefly reflect on how to properly achieve a performance-based design. We analyze modern architecture's use of these concepts by studying some exemplary buildings and we finish with a listing of the current energy analysis tools available in the market.

4. Portability Issues | This last chapter looks into the multiplicity of software available for the architectural practice and reinforces the need for an integrated approach capable of combining multiple tools and paradigms. A section is dedicated to algorithmic design tools and other current strategies that present valid solutions for the problem. Finally, some remarkable architectural projects that made use of CAD, BIM, and analysis tools in their design process are thoroughly studied.

The second part **– Integrated Algorithmic Design** - is divided into 6 chapters:

5. Design Methodology | In this chapter we explain our proposed methodology, an integrated algorithmic approach to design, in five stages. In (1) the 'Programming Environment' section we justify our choice regarding the programming tools. (2) 'Experimenting with form and concept' explains the primary modeling phase in CAD. The second phase is explained in (3) 'Transitioning to BIM'. The incorporation of analysis, that may occur during phase one or two, is fully explained in (4) 'Incorporating analysis in the design'. The final section, (5) 'Selling the product' occurs within the BIM paradigm, and is dedicated to a fine-detail modeling stage of the process where the goal is to produce renders.

6. Case Study: Astana Library | Here we present the case study chosen to test our methodology, Astana National Library from BIG architects. We justify our choice for the project and we summarily describe the main steps taken to model the design using the IAD approach.

7. Modeling for CAD | In this chapter we thoroughly explain the algorithmic description we developed to generate our case study in the CAD paradigm. In the process, we consider the advantages this paradigm brings to initial stages of the modeling process.

8. Transitioning to BIM | Next, we describe the transition our program underwent to convert the script from the CAD paradigm to the BIM paradigm, hence achieving portability between CAD and BIM. With this transition, we reflect on the different mindset needed to script a portable design.

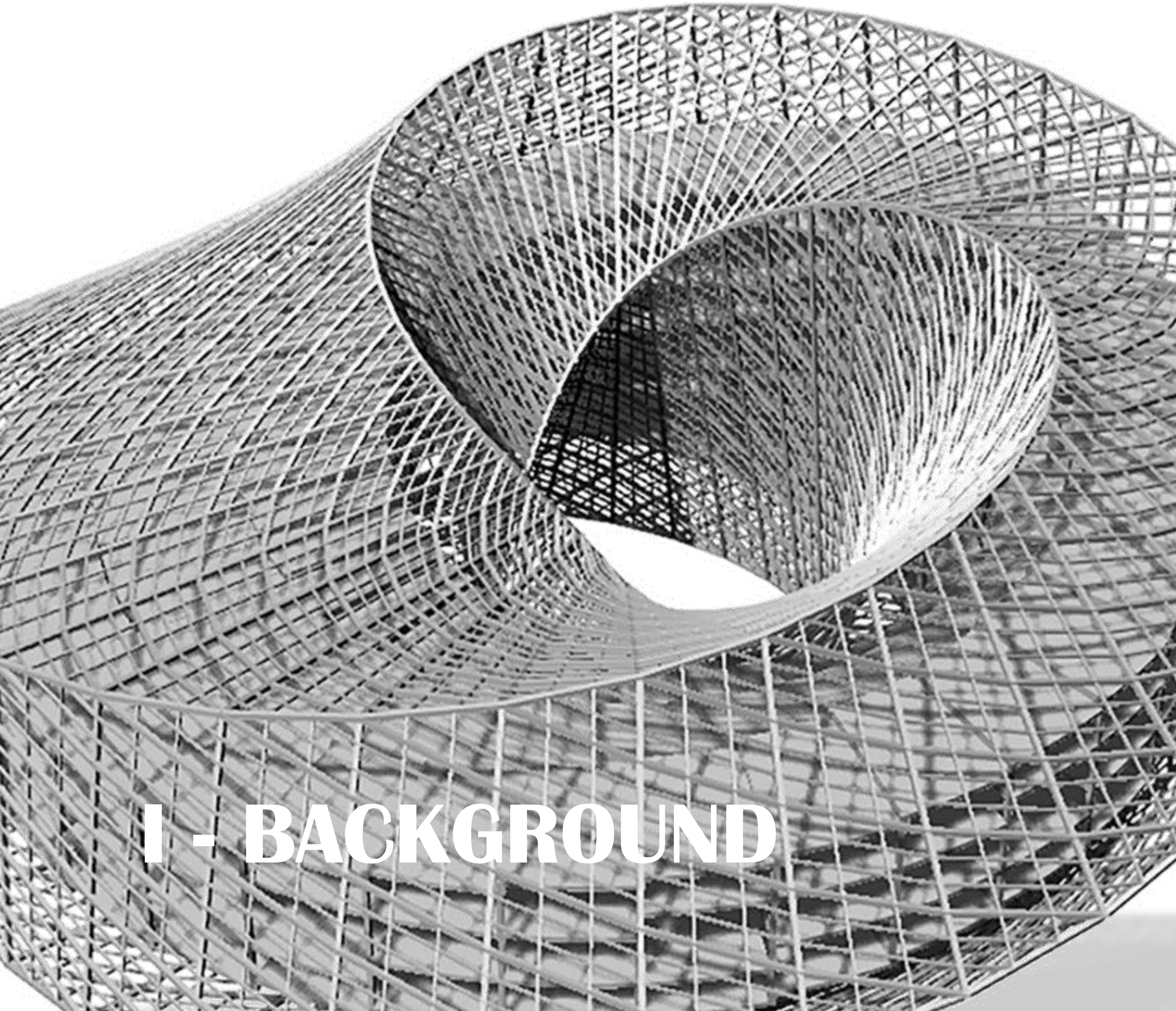
9. Going into Detail | We report the stage of the process where we had to relinquish the portability in order to take full advantage of the BIM paradigm's pre-modeled objects. Here we meditate on the advantages this paradigm poses to more evolved and detailed stages of the model.

10.Incorporating Analysis | In this chapter we address the inclusion of analysis information in the scripting of our design. We explain how the communication between the program and the analysis tool is made and how the results of the analysis are used in favor of the design.

11.Selling the Product | Here we ponder the importance of selling the design to a client, and the consequent need to produce realistically detailed images of the unbuilt reality. We reflect on the potential of CAD and BIM applications allied to AD, for such purposes. We describe how we scripted the elements required for this goal.

12.Evaluation | Lastly, we present an overview of some of the conclusion we could withdraw from our research, such as the advantages an algorithmic approach can bring to the design process, in comparison with the more traditional means of modeling. Some of the reflections are complemented with additional modeling tests we performed.

I - BACKGROUND



1 REPRESENTATION METHODS

Machines collaborating with human in design requires more than artificial intelligence - it requires artificial creativity (...). We know we have a manifestation of artificial creativity when a certain activity done by a machine would be considered creative if it were done by a human being. Within this rather circular definition, and provided we are not looking for miraculous manifestations, I would like to suggest that artificial creativity is already occurring, when we use tools that reinforce our creativity (...).

- Chris I. Yessios, 2003: 264

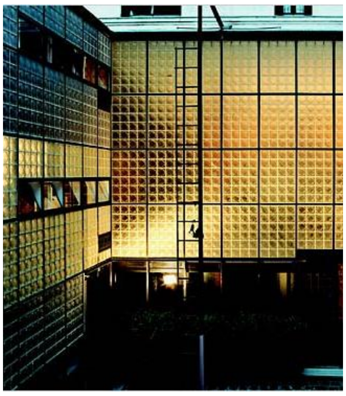


Figure 1.1 - Maison de Verre, Paris (1932) (source: http://architectuul.com/architecture/maison-de-verre)

Modern representation tools have changed more in the past few decades than in the sum of the last five hundred years (Sheil, 2008). Architectural drawings have for centuries been done by hand. Some remarkable figures have introduced new methods over time, like Pierre Chareau, who managed to design his Maison de Verre (Figure 1.1) in a very unconventional manner. In collaboration with Bernard Bijvoet and Louis Dalbet, the architect led his design to construction mainly through conversation and modeling. During the 20th century, many other pioneers have put forth efforts to change the paradigm and enlarge the spectrum of representation methods and tools used in architecture. Among them we find Antoni Gaudí, Richard Buckminster Fuller, Jean Prouvé, Cedric Price, Charles Eames, Ray Eames, and more. Despite their attempts to rethink the common practice, design tools remained mostly the same.

For most parts of the world the paradigm did not change until 1950 (Mitchell, 2004). The image of the architect to society was tightly woven with the pride and prestige they took from their graphic skills. In the last few decades, however, the architectural community has experienced a burst regarding digital tools that changed the pattern. Digital technologies are modifying the architectural practice at a rate that was unimaginable only decades ago (Kolarevic, 2003), leading the way to new architectonic possibilities.

Kolarevic compared the consequences of the digital information revolution in the building industry, to those the industrial revolution had in its time. The introduction of the digital in architectural representation methods is reconfiguring the relation between conception and production. New fabrication technologies such as computer numerically control (CNC) are allowing a direct connection from what the human mind can conceive to what the industry can build.

Challenging and exhilarating novelties are filling out practice. With such a scenario, it becomes even harder to predict the impact information technology will continue to have on the architectural practice. However, understanding how representation methods have gotten to where they are now, might help us discern a pattern for the cultural proliferation of changes through time, from which to extrapolate future possibilities.

1.1. HISTORY OF REPRESENTATION IN ARCHITECTURE

Architecture, as the professional practice that we know today, is a relatively recent phenomenon. Buildings prior to the *Renaissance* period were designed and planned in a very different way (Kalay, 2004). A master builder, commonly a mason, would develop a scheme of the building in his mind, following the traditional pattern of his era, which he likely learned from his master as an apprentice, and communicate his intention right on site to his craftsmen. This communication relied on rudimentary forms of design representations, rough sketches and physical models, as well as verbal instructions and demonstrations.

Kalay reported that only around 1450 were other means of representation used. Scale drawings allowed architects to distance themselves from the worksite as they could fully express their ideas on paper. Leon Battista Alberti was one of the first to pioneer this separation between conception and construction (Figure 1.2). The denomination "architect" was adopted at this time, and the craft became a profession. A new system arose to complement the communication between the newly divided arts: a language convention for plans, sections, and elevations.

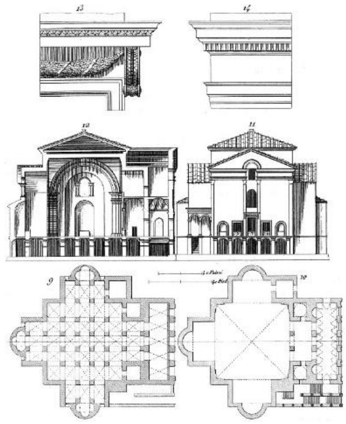


Figure 1.2 - Plans, sections and details of the architraves from Alberti's Church of St. Sebastian at Manuta, erected in 1460 (source: http://www.quondam.com/14/1459.htm)

The rift between architecture and construction widened even more during the mid-nineteenth century as these orthographic abstractions became contract documents (Kolarevic, 2003). With the twentieth century came new materials and technologies, which in turn led to increasing complexity in building design. World War II triggered many technological innovations (Kalay, 2004). Among them we have the radar and the atomic bomb, but also automatic data processing machines (computers).

Computers, later put to use in economics, politics, science and other areas, were originally invented to help the military, namely in deciphering enemy codes and calculating artillery firing tables. Architecture didn't take long to follow: the early success demonstrated in the resolution of mathematical problems with computers encouraged researchers to seek the integration of computational means to solve complex engineering problems.

1.2. COMPUTER-AIDED DESIGN

Ivan Sutherland pioneered the field in 1963, with the invention of Sketchpad, the world's first interactive graphic system (Llach, 2013). The Sketchpad (Figure 1.3) was part of his Ph.D dissertation at Massachusetts Institute of Technology, which was, between 1957 and 1967, funded by the US Air Force. The goal was for a group of engineers, researchers and students at MIT to develop a project that would reinvent design in the language of a machine. Hence, under the umbrella of cold war, Computer-Aided Design (CAD) was born and began redefining representation methods: computer generated graphics were not just new means of drawing a design, but were also computerized descriptions of that design. Llach asserts that Sutherlands' thoughts on design representation methods anticipated the contemporary culture of project description: one that integrates not only graphical outputs, but also material list, floor area calculations, building time estimates, and many other auxiliary outputs that help speed the production of documents and enable cost, heating, lighting, ventilation analysis and simulation.

Figure 1.3 - Sutherland drawing with Sketchpad (source: http://www.i-programmer.info/history/people/329-ivan-sutherland.html)

For many years, however, this technology was not accessible to public use. According to Kalay (2004), early CAD systems could only be used in large and powerful computers capable of calculation-intensive computing, and required specialized display hardware and input devices that were quite expensive. Only in the late 1980's, with the appearance of fast processors and growing storage capacity, did drafting software become available for a wider range of users. Companies like AutoDesk, VersaCad, Summgraphics and Microstation launched the first software tools specifically intended to support architectural design.

In a short period of time, CAD tools became indispensable to architectural practice. By the mid-1990s, designing a building without CAD tools had become unimaginable (Kalay, 2004). Only about three decades after the first working CAD technology was demonstrated, computers had assumed a fundamental role in the design process. With no intelligence or will of their own, these instruments could, however, augment the abilities of designers, as well as ease or automate specific tasks. Furthermore, computers could help architects see unbuilt realities.

1.3. HEISTING SOFTWARE FROM OTHER ARTS

Figure 1.4 - CATIA software for automobile design in 1988 (source: https://www.3ds.com/about-3ds/history/1981-1997/)

Architecture was not the only discipline that propelled the development of CAD systems. The first generation of CAD systems, as described by Kalay (2004), was also geared towards the automobile (Figure 1.4) and aerospace industries, namely the mechanical engineering applications. Over the past decades, the two routes have evolved in parallel, and new industries began developing software tools of their own as well, e.g. animation tools for the filming and gaming industries.

Unfortunately, CAD systems dedicated to the building industry do not always meet the needs of the profession and many architectural studios today resort to applications that are not specifically intended for architectural design (Aish, 2003). Aish divided the most common software types, used in the sample of projects and studios shown in the book *Architecture in the Digital Age: Design and Manufacturing*, in three: (1) general geometric foundation software, (2) solid modeling software for mechanical design, and (3) animation software.

In choosing to work with non-architectural specific software, the architectural design process suffers from what Kolatan (2003) described as "productive inadequacies". However, the software that is currently available for architectural practice has limitations, which justifies this search for different solutions. Kolatan illustrates the scenario felt at her studio with a vivid comparison between this inadequacy and the process of writing with a knife: one must rethink the act of writing through the logic of cutting in order to arrive at the notion of carving.

Ideally, the design medium the architect uses to produce the virtual model should restrict the modeling operations to reflect what is physically realizable, in order to achieve the final goal of architectural design: constructing the building. When using general CAD systems to produce architectural design, this does not occur.

1.4. COLLABORATIVE ENTERPRISE

From the two-dimensional or tridimensional model, developed by the architects to represent their design idea, to the full construction of the referred design, many other fields of expertise must be included. The project begins in the hand of the architects, in the form of ideas and sketches of those ideas, and overtime acquires enormous quantities of data, which may include schematic design options, alternatives sketches and analysis estimates (Pittman, 2003). The builders analyze this information and, in turn, produce even more data relevant to the construction process, such as cost estimates, construction sequence schemes, construction materials and systems, etc. Finally, the information arrives to the contractor, subcontractors and suppliers, each undergoing a similar process.

Pitmman stressed that, whereas the participants of the design process usually work together in a common studied workflow, within the construction process the players are typically a more loosely knit group. The groups involved may be geographically distributed, antagonist and even adversarial towards each other, and although all of them are trying to solve information problems, they usually work with different technology infrastructures and methods that are different to correlate.

Even today, the collaboration between the different teams of architecture, engineering and construction industries is mostly handled by shipping printed drawings around the world. Given the currently available technology that allows for instant information transfer digitally, this is a tremendous waste of time and resources. This procedure is not standard to all and some projects benefit from the current technology collaboration, which involves sharing the information through websites. While faster, these processes still do not offer a full collaboration capacity between the players of each industry.

1.5. BUILDING INFORMATION MODELING

Building Information Modeling (Figure 1.5) aims to provide an answer to this shortcoming by integrating in only one design environment multiple kinds of information, such as the building's form and structure, construction and assembly data, building components, product information sheets and more. When correctly implemented, BIM technology facilitates the integration of the various expertise that play a part in the building industry process, which results in better quality buildings, lower production costs and faster developments of the project (Eastman et al., 2008). It has been rapidly gaining acceptance among the architectural and engineering communities, as well as in building design and delivery professions, the construction and manufacturing industries, and building owners and managers too (Kensek and Noble, 2014).

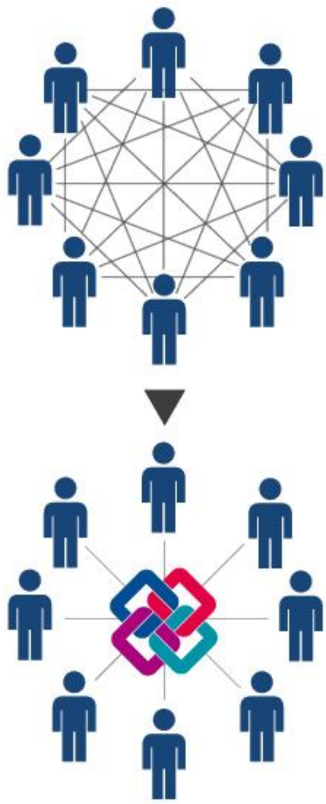


Figure 1.5 - Traditional Information Transfer Process on top, and optimized procedure in a BIM project at the bottom – edited (source: http://www.dds-cad.de/produkte/ihr-mehrwert/open-bim-und-ifc/)

Architecture is an ever-evolving discipline that seeks to exceed its barriers with every new creation. BIM appears in an era where architects are, out of sheer necessity, going back to being closely involved with the production of buildings (Kolarevic, 2003). The complexity of contemporary designs, namely highly curvilinear surfaces, brought forward new problems on how to construct them. The spatial and tectonic ramifications of such complex forms had to be rethought by the industry and constructability issues had to be introduced in earlier stages of the design process. With BIM technology, the cooperation between all industries involved in the process is optimized, which makes it possible to realize those complex forms in new ways, as well as construct them within reasonable budgets.

Figure 1.6 - BIM model (source: https://engenhariacivildiaria.com/2015/02/13/vantagens-e-desvantagens-do-bim/)

A BIM model (Figure 1.6) is an accurate virtual representation of the design that goes beyond form, to include rich semantic of other natures like performance and cost. When completed, it should contain all relevant data needed to support construction, fabrication, procurement and any other activities needed to execute the project (Eastman et al., 2008). The possibility of associating data with geometry allows architects to integrate technical and performance criteria at early stages of the design. The process is continuously informed by technological innovations, physically accurate digital materiality, intelligent database-enriched digital objects, etc. (Kensek and Noble, 2014). The exploration of alternative designs can be motivated by life-cycle assessment and other analysis or simulations that may offer accurate perspectives on the building's future.

According to Takim, Harris and Nawawi (2013), although hailed as a recent phenomenon, BIM technology can be traced back to 1982, when Gabor Bojar in Hungary developed the first BIM software for Graphisoft. Despite the industry's awareness of the advantages of BIM technology, construction organizations are still reluctant to fully embrace the paradigm (Takim, Harris and Nawawi, 2013). The reason for this slow progression ranges from the need of comprehensive training for designers that are used to CAD tools, to the legal regulations for construction in some countries that motivate studios to maintain the fragmented paper-based mode of communication.

1.6. TWO DIFFERENT PARADIGMS

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

In a CAD environment, one uses generic geometric objects to create almost every element of the model, from slabs and walls, to pillars and beams. Conventional CAD applications have very low semantic levels, they operate on lines, arcs, circles, etc (Aish, 2003). While this approach is very general and somehow similar to the workflow of drawings by hand, it captures little meaning. In the CAD paradigm, there are also no object libraries of families immediately available in the program. The user can either model the geometries from scratch or use pre-modeled blocks. CAD blocks, nevertheless, are not as flexible as BIM objects. While these present a set of parameters that allow several variations of the object to be created, CAD blocks must be modified by hand if more complex variations than for instance, scaling and moving, are desired.

Nevertheless, CAD tools, as free-form surface modeling tools, provide greater freedom in form creation (Zboinska, 2015). This may justify a preference in developing the early stages of the project, such as form and concept experimentation, in these applications.

Modeling with BIM semantics forces architects to model in a specific order and with pre-established conventions in order for the BIM logic to be maintained. With effect, further ahead in the process, this proves to be an advantage in resolving construction issues. Aish stressed that this last approach is extremely useful and productive when architects are working in close relationship with the engineering discipline, since, in fact, the vast majority of construction works this way, that is, with standard construction methods.

However, the idea of bringing architecture closer to the construction ideal, may somehow inhibit the creative potential of the designer. The obligation to model in a sequential manner that follows the constructive logic, may harm the creative workflow of the architect that would possibly rather model his design in a completely different order and fashion.

As each archetype better fits a specific stage of the design process, transitioning from CAD to BIM in the midst of a project is becoming more and more common. In order to take advantage of both approaches, some practitioners begin their design explorations in CAD environments, and when satisfied with the overall shape, they transition to BIM. However, the paradigms are so different, that transferring models from one to the other is hardly a simple task, and in most cases, architects end up having to rebuild the models from the start in order to avoid information loss. Some examples of projects that faced this issue are presented in section 4.

2 ALGORITHMIC DESIGN

"Just as log tables supplanted the abacus, the slide rule the log table, the calculator the slide rule, cold clumsy CAD finally exiled the drawing board and instrumental technical drawing tools from almost all offices worldwide. Once scripting becomes more accessible at least at a practical level, we will have new potential to share our design thinking across diverse design professions as well as with one another in fundamentally different conversations."

- Mark Burry, 2011: 247

Drawing is a mode of representation, as much as Computer-Aided Design. Programming, on the other hand, this is, describing design elements in logical loops of cause-effect relationships, is not only a mode of representation, but also a mode of generation (Shusta, 2006). The main difference lies in the way form is constructed. Using traditional representation methods, the architect channels an idea into a form, where with programming he channels a process into that same form.

Algorithmic Design (AD) defines the creation of architectural designs through algorithmic descriptions (Gerber and Ibañez, 2014). Using AD, the architect does not build the model directly, but instead, builds the program that builds the digital model. As such, AD can essentially be described as the production of architectural models from programs written by the architect. It is a design process that produces parametric models through a combination of geometric, as well as symbolic and mathematical representations of objects (Woodburry, 2010). The Algorithmic Design method is nowadays increasingly present in the architectural production process.

Mastering this process requires the architect to become part designer, part computer scientist, and part mathematician (Woodburry, 2010). Since an AD process involves the use of programs to generate forms, users must code their design intent, this is, translate the design into a program using scripting language. This is no simple task, since the user must take abstract ideas and turn them into precise instructions. Nevertheless, the benefit of using such an approach is great: AD allows designers to go beyond the mouse-based manipulation of models and transcend factory-set limitations of the currently used 3D software (Terzidis, 2006).

2.1. LIMITED USE OF SOFTWARE

In the previous chapter we concluded that CAD tools are more advantageous in initial stages of the design process, when compared to BIM tools, since they allow a greater freedom in modeling. Despite this fact, CAD technology is also not entirely successful in assisting architects in creative modeling (Brandon and McLain-Kark, 2001).

The initial goal for CAD tools was not only to free the designer from repetitive and time-consuming tasks, but also to allow him to explore design thinking beyond the traditional workflow of manual approaches (Terzidis, 2006). However, it is often only incorporated into latter stages of design. Brandon and McLain-Kark claim that tools such as AutoCAD are mostly used for producing construction drawings and other specification documents, that require precision detailing and high levels of accuracy.

Early stages of design focus on concept development, ergo in creative thinking. For imaginative phases of design, the architect needs tools that yield to his idiosyncrasies, in order to generate, communicate, and evaluate design ideas (Marshall, 1992), without being conditioned by a particular workflow or technique, other than his own.

The traditional means of conceptual design are hand-drawing techniques: sketching and drawing with pens, pencils, and markers, possibly the construction of scale models for further manipulation (Brandon and McLain-Kark, 2001). Most researchers and software developers are primarily concerned with the technicalities of converting design ideas into digital tools. Concerns regarding the use of these tools to actually design are left behind (Terzidis, 2006). CAD, if used in the conceptual design phase, only serves for the refinement of the already established concept. For a full incorporation of computational tools in the beginning of the design process, architects would need to adopt a different way of utilizing these tools.

2.2. SURPASSING THE LIMITATIONS

Others have addressed the issue of the limited use of software. Burry (2011) explains what drove him to transcend the limitations the software imposed on him as a designer. Instead of considering computers merely as productivity tools, he aspired to explore their potential to assist design explorations as well. But in order to use electronics instruments with the same authority we can apply to a pen, we must learn to control them. According to Burry, this is achieved through scripting.

Terzidis (2006) exposes the differences between computerization and computation (scripting). The first is the most commonly used in architecture today: the use of computer systems to represent, manipulate and store a concept that has already been developed in the designer's mind. The second concept is not so broadly used in our practice. It implies that the designer takes advantage of the computational power of the computer, transcending beyond the common and the predictable. According to Terzidis, the simple use of formally responsive computer applications cannot explore all possibilities that computational schemes can produce, nor can it bend to the designer's personality and will, beyond what the producers could predict.

To escape the influence we might experience, even if unconsciously, of using tools conceived by someone else, we must control the tools ourselves (Aish, 2011). In this way, designers can deflect the Whorfian effect (Terzidis, 2006), the risk of unknowingly becoming conditioned to the constraints of particular computer applications. Computation not only allows the user to have better control over the process and, as a consequence, the results, but it also facilitates the production of the specific documentation mentioned above. Delegating repetitive and boring tasks to the computer can accelerate the production process, as well as reduce the human error factor (Burry, 2011).

Designers often miss such opportunities for lack of understanding that scripting, or computing can be a part of the design process too. In the last two decades, however, according to Terzidis, architects have been changing the common practice from manually driven tool-based design to a computer-driven form-based design. Architects are starting to realize the inadequacy of industrial software, and some also begin programming their own in order to accomplish their design intents (Franken, 2003). The transformation from manual- to computer-driven approaches is nowhere near its full potential, yet the advancements of a few help us forecast the future of computational use in design practice.

2.3. THE PROGRAMMING APPROACH TO DESIGN

Programming, when applied to architecture, presents itself as a way for architects to go beyond the developed commercial applications, and mold them to their own way of thinking. The programming architect has the ability to extend and experiment with the rules and principles defined for traditional architectural processes (Terzidis, 2006).

Many fear that programming may have the potential to reduce the level of creative free-flow of the architect. The application of cold hard logic necessary to produce a program may in fact negatively affect the emergence of ideas, concepts and designs as we know them. However, while this logic is indispensable to distil the route to an answer to the problem, it is also natural that in digital design it should be applied in a way that avoids shoving the designer into a corner (Burry, 2011).

Instead, users should focus on the opportunities offered by this new medium of design. Programming has been hailed as an antidote to standardization, as it pushes architects into higher levels of form exploration. Far richer outcomes are possible with the same time investment, using programming instead of exploiting software merely at face value (Burry, 2011). The ability to extend software, to push their limits beyond what the manufacturers intended, provides a greater range of possibilities for creative speculation.

The overriding motivation for the designer, as a tool user, to become a toolmaker, lies on the quest for better designs (Burry, 2011). The search for a better design can be justified by a number of criteria: improved performance; the use of less quantities of materials more cleverly; the reduction in production timings and costs. All of which can only be achieved by wider spans of exploration possibilities (Burry, 2011).

2.4. ALGORITHMIC DESIGN

Designing with algorithms is a foreign concept to many. Architects tend to design with "an ethos of artistic sensibility and intuitive playfulness" (Terzidis, 2006: 57). The algorithmic logic then poses a problem as it is not perceived as a human creation, but rather something distant. What they fail to see is that by encapsulating simple problem-solving processes and leaving them to the metaphorical hands of the computer, one is allowed to leap into the new and unknown. Algorithms are then, not end products, but instead, vehicles for exploration, whose behavior is often unpredictable, and whose results often amaze even their creator (Terzidis, 2006).

Algorithms are sets of rules, which translate instructions given by a human to be performed by a computer, in order to solve a problem (Burry, 2011), in this case, a design problem. The problem, however can be totally or only partially known. Following the same logic, the coded solutions can be specific and determined or they can be unknown, vague, or even ill-defined (Terzidis, 2006). In these cases, lies the true power for the discovery of new ideas: unbounded and playful explorations in design that go beyond the imaginative capacity of the programmer (Woodburry, 2010).

Algorithms present the ability to compute using alternative and parallel logics to that employed by the human mind (Terzidis, 2006). This does not stand to say that algorithms are capable of replacing the architect's role in design exploration. On the contrary, they describe problems to be addressed and resolved as a human would do, as a human is coding them. The key factor here is the possibility given to the programmer of, not to simulate nor replace manual method of design, but to make the computer operate in ways similar to the human mind, as his counterpart. The combination of both systems, human and machine, is where the true power of algorithmic design discovery lies (Terzidis, 2006).

The specific problems that design algorithms solve are usually of geometrical nature. A programming architect must possess a sharp geometric knowledge in order to both understand and predict (to some extent) the effects of his algorithms (Woodburry, 2010). Algorithms are recipes that spell out practical tasks for the system to compute. Tasks of representing and manipulating design objects that the user must envision and comprehend if he is to write them in terms the computer can understand. For this, he must think geometrically, that is, to know when and how to apply such ideas as distance and angles, tangents and normal vectors, perpendicularity, etc (Woodburry, 2010).

2.5. DEALING WITH CHANGES IN THE DESIGN

Traditional design mediums include pencil, eraser and paper (Woodburry, 2010). One adds, the other subtracts, and the last one serves as stage where the design scene plays. The conventional design systems were quite straightforward in emulating this workflow. In these tools, the user creates the model by adding parts, relating them to each other through snaps and similar operations. Erasing is also an easy task since the parts are independent.

Making changes to these models, however can be a difficult task. Changing one dimension only may require the adaptation of other parts of the model, which has to be done manually (Burry, 2011). The more complex the model becomes, the more work is needed to manage changes. Algorithmic-based design, however, introduces a fundamental difference: hierarchical relations between parts of the design. This implies the user can no longer simply add and erase parts, as these actions may affect the whole. Instead he must relate and repair (Woodburry, 2010). The reason for this fundamental difference lies on parametric modeling.

Parametric modeling implies that the user, instead of creating the design solution by direct manipulation, creates a system of established relationships by which parts connect. The system is responsible for keeping the design consistent, and the building and editing of these relationships construct and change the model. This type of approach to modeling a design allows the user to explore a variety of different ideas without having to rework the model for every iteration (Woodburry, 2010). For instance, if the user changes one dimension that should affect various elements on the model, he needs only to modify that specific parameter and, since the entities in the design are logically connected, the ramifications of that change are automatically updated to the rest of the model (Burry, 2011).

Parametric modeling offers the user another type of play in the design process. The ability to support rapid change with very little effort at any stage of the design process is another great advantage. As projects advance in time and complexity, changes to the concept become costlier. This is partly due to the time consumed in changing the various affected aspects of the model by hand. With parametric modeling this issue is resolved. Hence the order in which modeling and design decisions are made can shift. This feature is both a deliberate strategy that characterizes parametric design, and a great financial argument for its use (Woodburry, 2010). Figure 2.1 presents the comparison amongst several variation tests for the shape of White Magnolia Tower, from KPF, possible due to the parametric characteristics of the digital model.

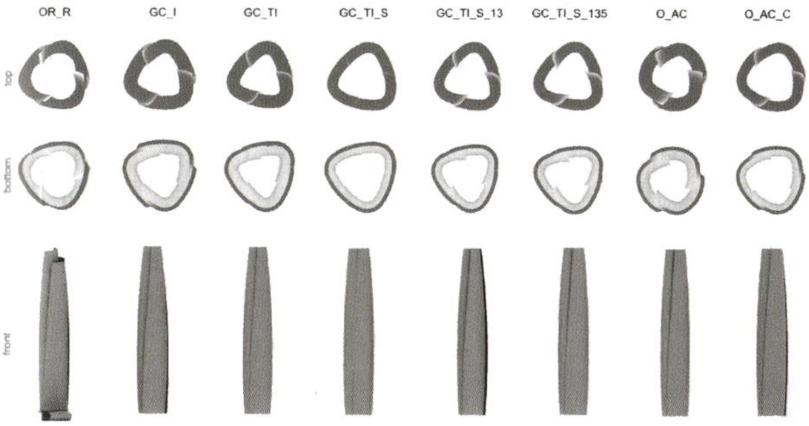


Figure 2.1 - Various shapes for White Magnolia Tower generated from the parametric model (Woodburry, 2010)

2.6. AD IN CURRENT ARCHITECTURAL PRACTICE

While some might still consider the need for programming skills a strange endeavor, we are, despite this fact, indisputably "moving from an era where architects use software to one where they create software" (Peters, 2013: 10). The structure of architectural studios is also changing to accommodate the work of the new computational designers. Peters highlighted four different inclusions of computer science in the today's architectural firms:

- **The internal specialist group** is the most common type. A separate unit works in parallel to the design team and can be integrated in the design process at different levels depending on the project's needs. This type can be found in practices such as Foster + Partners, Herzog & de Meuron, Grimshaw, Aedas|R&D, UNStudio, and Skidmore, Owings & Merrill (SOM).
- **The external specialist consultancy** is usually a technical and specialized practice, in engineering, software development, or other. They are hired by architectural firms to perform the computational aspects of the projects on their expertise. Examples of these sort of design consultants are Buro Happold SMART Solutions, Knippers Helbig Advanced Engineering and Gehry Technologies.
- **The computationally aware and integrated practice** includes the firms where design intent and computational technique are woven together to a point where computation is used in a natural or unconscious way. Among these we find MOS and Facit Homes.
- **The lone software developer/designer** presents an emerging style, a hybrid designer that is both architect and software engineer. This is most commonly found in small offices that have developed their own software. There are yet no cases of big architectural firms adopting this model, however, there are many small examples such as David Rutten (Grasshopper®/Galapagos), Daniel Piker (Kangaroo), Andrew O Payne and Jason Kelly Johnson (Firefly), Giulio Piacentino (WeaverBird), Thomas Grabner and Ursula Frick of [uto] (GECO™), Arthur van der Harten (Pachyderm Acoustical Simulation).

3 PERFORMATIVE ARCHITECTURE

*"Design is not just what it looks like and feels like.
Design is how it works."*

- Steve Jobs

Performative architecture refers to buildings whose design has as guiding principle its performance. In other words, what a building *is* should be defined by what the building *does* (Kolarevic and Malkawi, 2005). The logical progression of the design process has always involved the coordination of multiple parameters and the collaboration of multiple agencies. In performative architecture, however, the projected design undergoes analytical processes of evaluation in one or many of its agencies, in order to find the form that performs best.

The concept of performance is leading architecture into a form finding philosophy. Architects no longer stop at the definition of a design concept, but now use it as a foundation for a subsequent process – performance-based design. Through an informed search, an enhanced version of the building's shape can be found, that better responds to defined performance conditions. These conditions or parameters can be of multiple natures, such as structural, climatic, acoustic, economic, sustainable, etc. (Oxman and Oxman, 2014).

3.1. PERFORMANCE PIONEERS

This concept is not an entirely new thing. Take for instance the canonic examples of Antoni Gaudí, Frei Otto, Heinz Isler and others, whose performance techniques applied in design pioneered the form finding philosophy. Their experiments were conducted in a time where the digital was not yet such a big part of the equation. Yet with physical form-finding techniques they laid ground for the modern engineering techniques that have come to replace their methods.

Figure 3.1 - Gaudi's inverted model for the Sagrada Familia Church (source: https://cerebrovortex.com/2013/03/11/sagrada-familia/)

In the early 20th, the Spanish architect was using hanging chain models to determine the optimal form for load bearing structures (Kilian, 2004). Perhaps the best-known example is the inverted model for Gaudi's unfinished masterpiece, the Sagrada Familia Church, visible in Figure 3.1. The model's arches follow the catenary curve, the optimal shape for vaults purely under compression (The Works of Gaudi, 2016). The combination of this technique with the plaster models to guide the stone masons through the ruled surfaces, allowed for magisterial design solutions, that still leave researches today in awe of his ability to conceptualize this work. Figure 3.2 offers a view of the organic ceiling of the nave of the on-going project.

Figure 3.2 - Ceiling of the nave of the Sagrada Família (source: https://cerebrovortex.com/2013/03/11/sagrada-familia/)

Figure 3.3 - Umbrellas for Pink Floyd's 1977 concert tour of the United States (source: https://www.dezeen.com/2015/03/11/frei-otto-a-life-in-projects/)

Later on, the German architect Frei Otto, conducted experiments with lightweight structures, namely tents and soap films, suspended constructions, dome and grid shells, and branching structures (Ahmeti, 2007). A lightweight structure is, by definition, an object with very little mass capable of carrying loads. Conceiving such a system requires a rational use of the materials and a profound understanding of the forces acting in it. Otto's life-long research into lightweight and adaptable construction led him to a "natural construction" philosophy that focused on the relationship between architecture and nature (Nerdinger, 2005). One of his projects, from 1977, can be seen in Figure 3.3 - a temporary structure composed of multiple umbrella shapes, with 4,5 meters in diameter, made of white cotton (Architect, 2016).

In the same time frame, we find Heinz Isler, hailed as one of the pioneers of shell structures in the world (Kotnik and Schwartz, 2011). The Swiss engineer gained particular renown for the structures in thin-walled concrete. His form-finding endeavors were also a crossing between engineering and nature: the solution he found to technical problems that arose were inspired by nature. As reported by Kotnik and Schwartz, the capacity of this structures to resist to tear and break, having such minimal thicknesses, is justified by its natural honesty, meaning that everything that is unnecessary is left out and the minimum amount of material that stays must obey the laws of nature. Figure 3.4 shows one of the projects Isler built in Switzerland, a Tennis pavilion where the shell works both as supporting structure and space enclosure.

Figure 3.4 - Tennis Center from 1982 in Solothurn, Switzerland (source: http://www.baunetz.de/meldungen/Meldungen-Zum_Tod_von_Heinz_Isler_791805.html)

Using physical means to experiment with form, these three pioneers placed a performative emphasis in structural design. Nowadays, however digital technologies are assuming a preponderant role in the simulation industry with the display of currently available analysis tools. In fact, they have allowed for the redefinition of a term that has now become one of the techniques most frequently associated with performance – optimization (Oxman and Oxman, 2014).

3.2. OPTIMIZATION

Optimization, as explained by Nguyen, Reiter and Rigo (2014), defines the process of making something, i.e. a building's design, as functional or as effective as possible. A mathematical definition for the word may tell us it is the process of finding the best solution for a problem. However, for building performance simulation (BPS) a global solution is frequently impossible to achieve since the problem itself is most times composed of multiple natures combined. The simulation programs also pose some limitations, due to the amount of time they consume, as well as the computational power they require.

The criterion defining the optimization process is responsible for conducting the search within the limits of the building's design space – the collection of designs the building can or cannot become. A multi-criterion optimization process usually reaches a sub-optimal solution that responds to all or most of the criteria in a compromised manner. Optimization in BPS can then include interactive improvement processes, sensitive analysis, brute-force search and other methods, for so as long as the process is automated and entirely based on numerical simulation (Nguyen, Reiter and Rigo, 2014). To this end, simulation tools must be combined with algorithmic design tools to create a generate/evaluate/moderate cycle of digital design, capable of optimizing the building's shape (Oxman and Oxman, 2014).

The following section will focus solely on one of the possible natures of performative design, the environmental nature. We will focus on the understanding of how the building's physical context can inform the design process, a concept that has been explored long before digital technologies had a role in architecture.

3.3. ENVIRONMENTAL PERFORMANCE

Performative architecture influences buildings' design by blurring the distinction between geometry and analysis, between image and performance (Kolarevic and Malkawi, 2005). The performance evaluation should also take into account many aspects of the building's future life and use. Environmental performance places the emphasis on the relation between the designed building and its environment: local factors should generate different architectural responses. The building site and context must be analyzed at spatial, material, cultural, temporal level, and more if an optimal relation to the environment is meant to be found (Hensel, 2013).

Environmental performance has been recently revisited and is gaining ever more emphasis as climate change awareness grows, as cost reduction needs become more imposing, and numerous other factors. Adding to this, the availability of automated analysis tools has pushed architects to be more involved in analysis and simulation, resulting in the appearance of more ecofriendly energy- and cost-efficient buildings. However, the concept of buildings that physically adapt to the environment surrounding them is not an entirely 21st century idea (Burry, 2011). Take, for instance, Villa Girasole (visible in Figure 3.5), a rotational house that progressively follows the sun's movement during the day, built in the Italian province of Verona from 1929 to 1935 (Novaes, 2003).

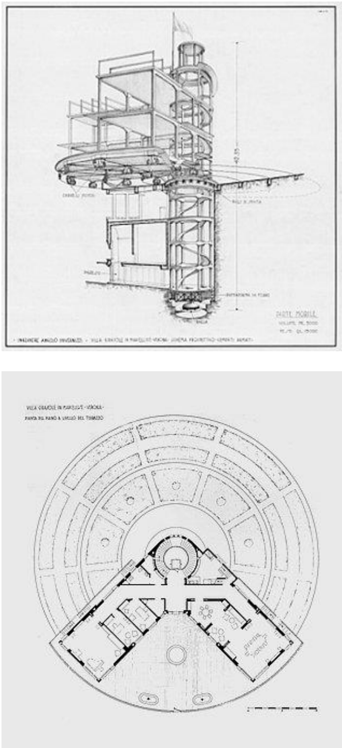


Figure 3.5 - Sectioned perspective and plan of Villa Girasole (source: http://obviousmag.org/en/archives/2011/01/rotational_villa_girasole.html)

According to Hensel (2013), if we trace the notion back to where it gained the popularity we will find ourselves in the mid-twentieth century, when the impact of several scientific fields on architecture grew stronger, biology in particular. The concept of performance began spreading in the humanities and social sciences, advancing afterwards to the arts and science in general. An intellectual movement known as the performative turn began taking shape during the 1940s, suggesting the notion of performance as a social and cultural element.

From this point onwards, many different attempts to theorize the concept of performance emerged, resulting in a multiplicity of approaches towards performance. According to Hensel (2013), David Leatherbarrow initiated the first integrated approach to performance that strived to join the various efforts that commenced at the beginning of the 21st Century. The integrated solution considered the relation between planned and unplanned performances, between the building's purpose and its location and context. The author argues, however, that the approach has not yet come to full fruition in architectural practice.

3.4. INTERIOR CLIMATE CONTROL

Around 1960, mechanical-electrical systems designed to modulate interior climates redefined interior spaces as quasi-hermetic boundaries. Modern buildings became universally conditioned by optimized technology, which reduced the need to deeply think, analyze and create forms that significantly influenced the building's performance (Frampton, 1983). The industrialization period contributed significantly to the climate control standardization trend. The attempt to devise closed ecological systems for spaceflight programs and the design of Cold War bunkers reached a peak in the 60s, and accelerated the development of interior climate control systems. These systems ensured homogeneous interior environments, as well as a firm division between exterior and interior.

Figure 3.6 - Palm House in Belfast Botanical Gardens, a 19th century cast iron and glass glasshouse (source: https://vagabondimages.in/tag/britain/)

The German philosopher Peter Sloterdijk (2005) placed another relevant benchmark in the glasshouses that emerged in Great Britain in the 19th century (Figure 3.6). These structures aimed to provision an interior environment, radically different from the local one, with the adequate conditions for the growth of plant species native to other climate zones.

For Hensel (2013) another peak was reached with the development of the Bürolandschft (office landscapes - Figure 3.7), a 50's movement carried out by the Quickborner team for planning and organization. Bürolandschft were open plan spaces with cluster workstations, arranged according to the anticipated workflow of the employees that would occupy them. Their project intended to create a more "human office environment", however, the resulting homogenized interior is arguably flawed as the comfort requirements might not be the same, at all times and in all circumstances, for all human beings.

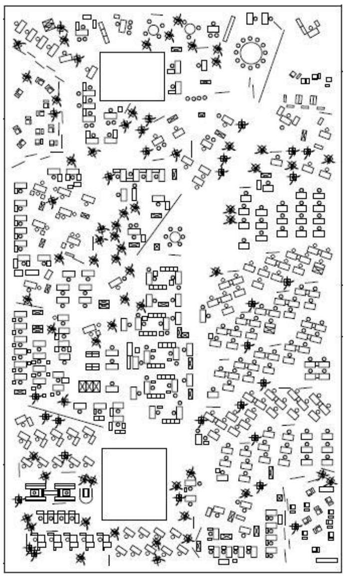


Figure 3.7 - A Bürolandschaft layout (source: http://workplaceinsight.net/organic-design-used-reflect-way-people-move-around-building/)

According to the same author, developments like the one mentioned above motivated the tight regulations for regular homogeneous interiors that were soon to follow. Interior climate control grew increasingly affordable from then on, becoming a status symbol as well. From the 1970 onwards, traditional means for environment modulation in architecture were practically obsolete throughout a large part of the world (Hensel, 2013).

Still today, the trend is to favor pre-calculated technical solutions that facilitate the approval of the building's plan and avoid costly research that would need to be done to acquire reliable data for a more environmentally informed architecture. The great majority of architects relies on specification changes and mechanical systems to improve energy performance, instead of considering an incorporation of the analysis data in early phases of the buildings' design (Anderson, 2014). Architecture is locked in a harsh dialect between the man-made and the natural, where no great relationship between the two is allowed to exist (Hensel, 2013).

3.5. PERFORMANCE-BASED DESIGN

In order for architecture to participate in the interlinked environmental and ecological processes that surround it, it cannot be limited by technologically facility exchanges. Architectural design must consider context- and time-specific exterior-to-interior relations, as well as interaction between the built structure and the dynamic environment surrounding it. If architecture is truly to perform, architects must somehow rely on the concept of "non-human agency", or a lack of intentionality in the agency (Hensel, 2013) that defines the form, which brings us back to form finding, to the detriment of the typical architectural approach of form concept definition.

Anderson (2014) declared that architects have, for too long, relied on engineers to understand and provide the climatic comfort conditions to the buildings they design. Energy modeling through cooling and heating equipment seem to be the common method of controlling interior environments. A question must be asked as to whether the physical characteristics of the building can effectively contribute to a larger extent to a better environmental performance and, in doing so, reduce the need to rely on devices for the same purpose. The making of informed decisions in the early stages of the design process, when the architects are still associating space, geometry and proportion in sketches, can affect energy use over the building's life for more than a hundred year. Whereas most technologies last ten to twenty years only, in comparison.

These decisions need to be meaningful if human societies are ever going to become sustainable. Reitan (2005) affirmed that our architecture should be as attuned as possible to the local environment. Buildings must interact with the local environment, modeling it and simultaneously receiving stimuli from it. However, Hensel (2013) admits that identifying which considerations should be included in the design to generate an appropriate architectural response, is far from straightforward. Many agencies can inform the design process, namely local communities (biotic factors and interactions), the local physical environment (abiotic processes and interactions), spatial and material organization, and more. The challenge is to understand which to exclude from the equation and how to relate the factors to consider.

In order to design with performance as guiding principle, this challenge needs to be carefully addressed. The development of new instruments and methods of predicting buildings' structural or environmental behavior is currently at its peak. However, while these new technological methods might contribute to an understanding of buildings' behavior, a full comprehension of how they are imagined, made and experienced cannot result from the development and deployment of new techniques alone (Kolarevic and Malkawi, 2005).

3.6. ENERGY EFFICIENT BUILDINGS

According to Hensel (2013), some promising approaches are beginning to appear in the domain of performative architecture. Free running buildings are one of them. These constructions are designed to need no heating or cooling in general or just during a particular season. This is, of course, easier to achieve in temperate climates, and usually during the summer months. Some examples of projects designed to be energy efficient constructions are presented next.

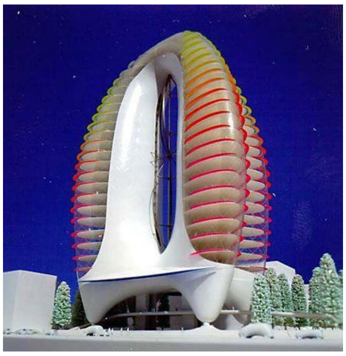


Figure 3.8 - Project ZED, a multiple-use building in London from 1995 (source: http://wind-energy.ucoz.com/)

Project ZED (Figure 3.8), was one of the first buildings designed for zero CO_2 emission (Stankovic, Campbell and Harries, 2009). It was meant to harness solar and wind energy, becoming a self-sufficient building in terms of energy use. The building's curved shape has a great influence on the wind recoil: the envelope's aerodynamic performance was enhanced through computational fluid dynamics analysis and is thus capable of channeling the wind that hits the building to the giant turbine placed in the center. To capture the sunlight, the façade is also loaded with photovoltaic panels, incorporated in the louvers (Kolarevic, 2014).

The London city hall (Figure 3.9), by Foster and Partners presented another example. The team wanted the building to be energy efficient, so they parametrically described the shape and Arup's engineer ran a series of analysis that defined many aspects of the projects (Figure 3.10). For instance, a solar study informed the decision for the cladding system and the acoustic analysis of the debating chamber was essential to determine the sound refection problems in the space and to discover the solution that resolved them – a spiral ramp wrapped around the building's flask-shaped atrium (Whitehead, 2014).

Figure 3.9 - London City Hall (source: http://www.fosterandpartners.com/projects/city-hall/)

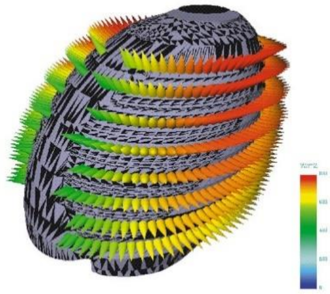


Figure 3.10 - London City Hall - Analysis model (source: http://www.detail-online.com/inspiration/city-hall-in-london-106486.html)

3.7. ADAPTIVE BUILDINGS

Hensel (2013) refers yet another approach that has been gaining some relevance: the adaptive architecture. Nicol and Pagliano (2007) explain the concept by observing the active behavior of the building's users towards the building itself. The authors claim that people have a tendency to adjust their surrounding environment in order to make it comfortable. This can be done by changing their clothing and activity or by interacting with movable or adjustable objects in the environment.

Architecture can also benefit from this ability to change in conformity to the conditions of the environment. An adaptive architecture, is then an architecture of movement, where some parts of the building either move, or let themselves be moved (manually, mechanically, electrically or digitally). The movement can be initiated by human or environmental agents, but in either case, with the aim to modify and mediate the environment inside the building (Kolarevic and Malkawi, 2005). Currently, as advancements in sensor and actuator technology are being associated to digital computation, real-time-responsive performance in architecture is emerging at a global scale (Burry, 2011). Architectural firms such as Ateliers Jean Nouvel and Aedas have provided us with good examples of this technological fusion.

The Arab World Institute, located in France, is an architectural project from Ateliers Jean Nouvel dating from 1987 (Ateliers Jean Nouvel, 2016). The building was meant to be a showcase for the Arab culture in Paris and its design features symbolic elements like the "moucharabiehs" (or *mashrabiya*), a natural ventilation system traditionally used in Arabic countries. These elements have particular relevance in the southern façade, where the tradition Arabic pattern is combined with advanced responsive metallic brise soleil (Winstanley, 2011). The façade (Figure 3.11) is equipped with light sensitive diaphragms and depending on the sunlight's intensity, the *mashrabiya* units either open or close. This adaptive mechanism not only shapes the light coming into in the interior, creating fluid motions of luminance, but also has an important role from an environmental control standpoint.

Figure 3.11 - Institut du Monde Arabe (source: http://www.akdn.org/architecture/project/institut-du-monde-arabe)

Figure 3.12 - Al Bahar Towers (AHR, 2016)

Al Bahar Towers in Abu Dhabi (Figure 3.12) are a more recent project from Aedas Architects, and it currently features the world's largest computerized dynamic façade (AHR, 2016). The design concept is a fusion of many aspects regarding the projects' environment, namely the history, culture and nature, developed in close relationship with state of the art technology. The geometry of the façade's shading layer mimics a flower, whose petals fold and unfold with the movement of the sun. This solution not only reduces the solar gains inside the building and, consequently, cooling demands, but also allows for diffuse light to penetrate into the interior, instead of direct light, which would be much less comfortable for the human eye.

3.8. ANALYSIS TOOLS

The software tools architects use today for building performance analysis are the result of years of research and validation (Anderson, 2014). Analysis methods and software accuracy have been improving in the past decades and architects today benefit from a plethora of different tools to choose from. Focusing on energy modeling, we will present some of the most used software in architecture, according to Kjell Anderson, architect at LMN Architects in Seattle (2014).

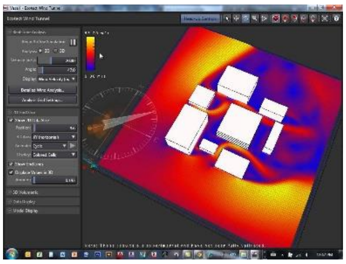


Figure 3.13 – Wind Tunnel analysis in Vasari (source: https://www.youtube.com/watch?v=CoZ2mp2xaZE)

Figure 3.14 – Diva for Rhino - logo (source: http://arc.prismatecs.com/courses/architectural-software/diva-for-rhino/)

- Autodesk provides the user with multiple analysis tools, namely several that are connected to the BIM tool Revit. Ecotect is, since 2008, one of them and allows daylight and energy modeling.
- Vasari also belongs to Autodesk, although it is not incorporated in Revit yet. It is capable of various simulations such as wind, climate, daylight and electric light analysis and solar studies. Figure 3.13 presents a wind tunnel simulation in Vasari.
- IES Virtual Environments is a costly software based on open-source ESP-r engine. It offers multiple levels of detail when choosing from the existing variety of modules. The models allow daylight, airflow, insolation, and other types of analysis.
- Diva for Rhino (Figure 3.14), on the other hand, is relatively inexpensive and is a very complete simulation tool. Operating within Rhinoceros it interacts with Radiance and Daysim, two other simulation engines for daylight analysis and daylight autonomy respectively. Diva can also connect to EnergyPlus for energy modeling and it allows the user to parametrically control the process through Grasshopper. The combination with other Grasshopper plug-ins like Ladybug or GECO also allows weather files to be used as inputs, as well as ecotect analysis results.
- With OpenStudio the user can produce energy simulation and daylight analysis for entire buildings. The software belongs to the National Renewable Energy labs and also includes energy performance, daylight and glare estimates simulation.
- Sefaira is a more recent tool with new features still being added monthly. In a very graphic interface the user can test and compare various design options, with energy and thermal simulations.

Despite the variety of tools available for architects, an issue still remains regarding software portability. Several analysis software require a particular geometric model of the building, different from the one used in the CAD or

BIM modeling tool, thus requiring a translation process or again, a rebuild (Aghemo et al., 2013).

Most of the tools mentioned above, besides their own native 3D modeling options, allow model imports from simpler modeling tools like SketchUp for instance. Some of them require the model to be built in the 3D modeling tool they are coupled with such as Revit or Rhino.

Having to limit the user's choice as to where the building should be modeled with base on the analysis tool choice is a great disadvantage. The issue becomes even worst when architects wish to perform multiple analysis of various different natures, thus requiring multiple analysis tools. In such cases, they must either transfer the model, which does not always work (Bazjanac, 2001), or completely rebuild it which is a time-consuming task.

Even when the user is modeling his design in a software that has a coupled analysis engine that suits him, most frequently he has to produce a simplified version of that model for analysis (Bazjanac and Kiviniemi, 2007). Diva for Rhino, for instance, requires all geometry to be mere planes, as it cannot process solid objects. Simplifying the model or exploding the geometry might seem a fairly simple task, however if the user desires to test multiple instances of the model, the process becomes repetitive and tiresome.

Analysis tools are more important now than ever before, as higher complexity levels are being achieved in building design, making them less predictable from the thermal, lighting, and acoustics point of view. However, the current inadequacy in the interoperability between 3D architectural models and the models required for energy modeling is still short of a truly efficient use of simulation tools in building design practice (Kensek and Noble, 2014).

4 PORTABILITY ISSUES

"An environment is missing that integrates representation and simulation-based approaches. It could, for example connect modeling with physics-based behavior, scripted elements and the generated structure could still communicate as whole to external environments."

- Martin Tamke, 2011: 65

Architecture, as an art and an industry, is currently competing in different fields - cultural and commercial for instance. According to Kolatan (2003), against branded products, advertising, the internet, and the music and film industries, it is not faring very well. A more suited scenario for Architecture is proposed by the author: a chimerical system where architecture could benefit from the different paradigms listed above, by adapting its models to all possible scales and contexts. According to Rosen (1995), a Chimera is the adaptive response of a system that finds its survival at stake due to environmental changes. This response is based on a cooperative behavior of the parts. Through this definition, we can then understand Kolatan's metaphor: architecture is but one system, organically interconnected with many others that both feed it and feed from it.

BIM has been driving us to the understanding of that notion as well as the necessity for close cooperation during the projects development. It is yet still far from achieving the full potential of the chimerical system. Deploying an architectural project requires a functional network of professionals and resources (Rahim, 2003). The more tightly woven this cooperation is, the better it alleviates redundancies in transmitting information between architects, engineers, clients, manufacturers, etc. Furthermore, they all require different forms of information, which means the network must be flexible and adaptable, capable of merging the needs of the different parties involved. In short, a Chimera.

These different types of information required by each party are most commonly translated in a multiplicity of software being simultaneously used to develop the same project at various levels. The cooperation process involves sharing and crossing the various information systems. However, due to the resulting collection of formats, much of the information is either lost in the translation processes, or has to be redone in other formats. While this phenomenon is currently unavoidable, as there is no software capable of performing all tasks required in the architectural creation process, the information spread can be controlled with portability mechanisms.

4.1. PORTABILITY MECHANISMS

The Industry Foundation Classes (IFC) was developed with the purpose of creating large data sets with consistent representations of building information that could be exchanged between different software applications (Eastman et al., 2008). IFC is the main building product data model for the AEC industries, namely for building planning, design, construction and management. These standards represent geometry, relations, processes and material, performance, fabrication and other properties, and may also be extended based on user needs. The IFC thus provides a mechanism for interoperability among applications with different internal formats.

Achieving portability between CAD or BIM models may be a relevant step towards a better cooperation of the different parts, however, it is not nearly enough to close the gap that separates the current reality from the utopian chimerical system. As mentioned by Kolatan (2003), the systems available to the architect today allow him to draw knowledge, inspiration and resources from other competing areas in the market. This fusion of crafts is what drives innovative designs and what allows those designs to come to a buildable reality. Hence, we find the need to merge different information approaches, not only between the different parties active in a project, but also within the architect's own workflow. With so many possibilities regarding software and modeling techniques available, the architect has the unique opportunity of joining the best of different approaches together in his own design process. He needs only a tool that allows him to do so.

A different option is offered by CORE studio. TTX is their inhouse interoperability software (Tomasetti, 2017). TTX establishes a database capable of storing updates and changes made to BIM models in a particular software, and automatically translate them to another. The platform currently supports Revit, Grasshopper, Tekla, SAP2000 and ETABS (the last three are software dedicated to structural analysis). Figure 4.1 presents the currently supported network of applications as well as some planned to be integrated soon. Flux.io (FLUX®, 2017) presents a similar system – a data exchange and collaboration platform that connects multiple design tools. Figure 4.2 shows the currently integrated tools.

Figure 4.1 - TTX currently supported programs and future work (Tomasetti, 2017)

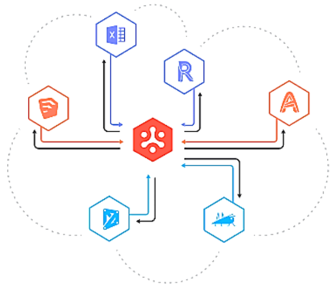


Figure 4.2 - Flux data network (FLUX®, 2017)

Algorithmic design, on the other hand, has the potential to become a portability mechanism on its own. The ability to produce an independent description of the design that can be generated in a software, other than the one it was modeled in, opens new possibilities. By programming a building's design, and using CAD, BIM tools or others as mere visualization backends, the user is free from these software's restrictions and limitations. It seems like a short step from this connection between algorithmic-modeling tool and visualization software, to an array of connections that could be made between the algorithmic description and other software. Matias del Campo (2011) expressed the perfect portability scenario as a possible universal scripting language capable of communicating with the entire array of software existent in an architectural studio.

4.2. ALGORITHMIC DESIGN TOOLS

Algorithmic Design, for all the limitations it vows to surpass in the use of modeling tools, is still coming short when it comes to the issue of portability. A large amount of programming tools is already available in the market, for both beginners and/or more experienced programmers, which connect to either CAD or BIM tools. Some are also capable of connecting to analysis tools as well. Nevertheless, programming tools still have to adapt to each software. Hence, it is still difficult to port AD programs between different applications (Leitão and Santos, 2011; Ferreira and Leitão, 2015).

As proof of this, we find numerous programming tools currently in the market designed to produce geometry in one specific modeling tool only. Take for instance Grasshopper (Davidson, 2017), a graphical algorithm editor, tightly integrated with Rhinoceros. For the same software, we also find, RhinoScripting and RhinoPython, both working with textual languages. AutoCAD offers the user the possibility to model in AutoLisp, F#, C#, C++, and others. Revit has various plug-ins as well. Dynamo is one of them, an application strongly influenced by visual programming languages. RevitPythonShell is another, designed as an alternative to access Revit's Application Programming Interface (API) using python. For Bentley's BIM tool, Microstation, a parametric and associative system was also developed – Generative Components.

4.3. PORTABLE ALGORITHMIC DESIGN TOOLS

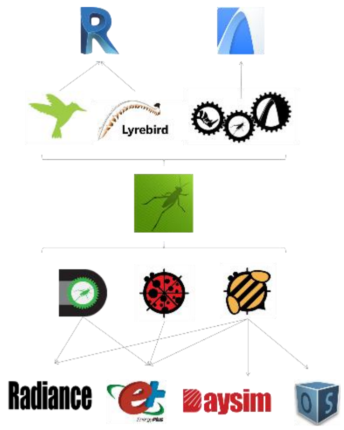


Figure 4.3 - Grasshopper plug-ins for BIM and Analysis tools

There are programming tools that allow portable AD. Grasshopper is one of them. This graphical algorithm also offers, within its programming environment, several plug-ins that connect the program to different BIM and analysis tools (see Figure 4.3). There is, for instance: Lyrebird, a plug-in that connects Grasshopper to Revit; Hummingbird, another plug-in that allows the creation of native Revit objects from Grasshopper; Rhino-Grasshopper-ArchiCAD, that makes the connection to Graphisoft's BIM tool; DIVA-Grasshopper, that allows the setting up of analysis in DIVA's plug-in for Rhinoceros in Grasshopper's programming environment; LadyBug, that allows the user to import Weather data files from EnergyPlus into Grasshopper; and HonneyBee, that connects Grasshopper to EnergyPlus, Radiance, Daysim and OpenStudio for building energy and daylighting simulation.

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

Following the portability lines, we have Rosetta, working within the textual language paradigm, however. Rosetta is capable of generating geometry in various CAD and BIM applications, such as SketchUp, Rhinoceros, AutoCAD, Revit and ArchiCAD, as well as in OpenGL for faster visualization (Leitão and Lopes, 2011; Feist et al., 2016). More recently it has also been extended to include Radiance, an analysis tool (Leitão, Castelo Branco and Cardoso, 2017).

These portable AD tools allow for a smoother transition between the CAD and BIM paradigms. Using the same programming environment, the user can model his design for both archetypes. However, this does not mean that the written descriptions are portable. In both tools mentioned above, the CAD and BIM operations used to generate the model differ, which means the user still has to adapt part of his description. Nevertheless, algorithmic-based approaches to design present great advantages still, for when properly implemented, the mathematics are oblivious to any software. Modeling operations, however, have to be adapted to the tools requirements, but this can be done in the same programming environment, hence using the same language and workflow.

The integration of analysis tools in the process also represents a great advantage to the modeling process, as it allows the architects to conceive their design based on a more informed research. Moreover, the connection of analysis tools to the algorithmic modeling tools opens doors to optimization processes, where the analysis tool and the modeling tool communicate in a loop process until a better performing shape is found.

4.4. PROJECT EXAMPLES OF INTEGRATED APPROACHES

As mentioned in the end of section 1, many great architectural endeavors today engage in a multi software approach, particularly when diverse companies participate on the project, such as engineers from different specialties, contractors, etc, each developing their respective part of the project in their work software. The following sub-sections present a series of buildings where a high coordination was needed to merge the various separate models of the same building, being developed simultaneously by all partners working on the projects.

Figure 4.4 – "Bubble": IAA 1999, Frankfurt, Germany (source: http://www.franken-architekten.de/)

4.4.1. BMW PAVILIONS

Bernhard Franken is as architect and an engineer, best known for the exhibition pavilions he designed for the BMW group in cooperation with ABB architekten (Figure 4.4 to Figure 4.7). His own architectural firm, Franken Architeckten, pursues the medieval concept of a coherent process from design to production, achieved through digital parametric design (Kolarevic, 2003).

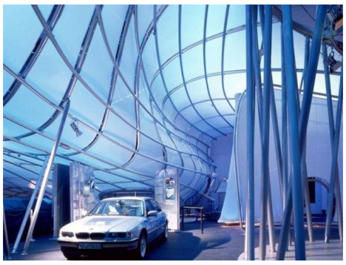


Figure 4.5 – "The Wave": BMW Pavilion at the Expo 2000, Munich, Germany (source: http://www.franken-architekten.de/)

The BMW pavilions are prototypical buildings whose main objective is to communicate the brand's chosen marketing value: innovation. The pavilions that the team designed had to embody the concept, not only in shape, but in the production process and finishing quality as well. To meet these goals, the architects used high-technology and were closely involved in the production process. However, In *Real as Data*, Franken (2003) confesses that to achieve the results his team intended for the pavilions, they had to invent new production methods of their own, as no existing technique in the building sector suited their needs. For a team of 75 architects, structural and mechanical engineers, communication experts, lighting designers, audio-visual specialists to work together and manage to produce high quality results in the tight schedules that characterize these types of projects, a finely-tuned production process must be conceived.

Figure 4.6 – "Dynaform": IAA 2001, Frankfurt, Germany (source: http://www.franken-architekten.de/)

The team used Maya, an animation software, for design development, and Ansys and R-Stab, special finite element programs, for structural calculations and tests. The load bearing structure was conceived using Rhinoceros and Mechanical Desktop, a mechanical engineering add-on for AutoCAD. CATIA also came in handy, as some of the structural elements could not have been worked anywhere else. Vectorworks was used by the interior designers, responsible for the communications, lighting and other constructions for the interior. The shop drawings required special programs like PK Stahl, running on workstation. For the CNC machines, additional programming was necessary in order to post process the separate data and convert it to a code that the machine could understand.

Figure 4.7 – "LightArc": BMW pavilion at the 2002 Auto Show in Geneva, Switzerland (source: http://www.franken-architekten.de/)

Finally, to coordinate all the files being produced in this variety of programs and operating systems, and facilitate their exchange among the team members, they used an interface format (IGES) with which the different programs could communicate, and a browser (Rhinoceros) where everyone could have access to the data.

4.4.2. 100 11TH AVENUE NEW YORK

Figure 4.8 - 100 11th Avenue Render (source: https://en.wikiarquitectura.com/building/100-11th-avenue-building/)

Located in Manhattan, Ateliers Jean Nouvel's 21-story residential condominium (Figure 4.8) completed in 2010, represented an innovative approach to the implementation of BIM (Eastman et al., 2008), that also required a variety of software interoperability. According to Eastman, et al., Font Inc, the façade consultants for the iconic curtain wall, used Digital Project (DP) for this building from the early stages of its design and from this platform they were able, not only to update all information regarding the products for the curtain wall, but also exchange it with the rest of the project members.

Parametric modeling was used to identify rational systematization systems that would not compromise the designs aesthetic value. The process required them to use several different software programs. They received from the architects a 3D model with polygonal representations of the tilted glass panels developed in Rhinoceros. They implemented the designer's intent in Digital Project, according to the Rhino model.

Beyer Blinder Belle (BBB), architect of record, produced the entire documentation for construction, excluding the steel and glass façade. The coordination between Font Inc and BBB for the production of the façade's detailed cross-section, required many iteration cycles of document transfer. Font Inc would deliver PDF documents, and BBB would send information back in AutoCAD files. The two-dimensional information from AutoCAD was then imported to DP.

Finally, for the fabricator in China, Font Inc. provided a 3D model in CATIA. Robot and Strand were also used along the way for structural analysis of the framing system. For that purpose, the DP file had to be exported in IGES format for the two simulators to read.

4.4.3. SHANGHAI TOWER

Figure 4.9 - Trio of towers in Lujiazui financial district (source: https://www.dezeen.com/2016/01/11/shanghai-tower-gensler-world-second-tallest-building/)

The Shanghai Tower, from Gensler and Tongji Architectural Design Institute, is a recent centerpiece to the Lujiazui commercial district. It is part of a tower-trio, visible in Figure 4.9, representing Shanghai's past, present and future. While drawing inspiration from Shanghai's traditional houses, the most recent tower (on the right), also manages to find meaning for its spiraling shape on China's emergence as a global financial power (Gensler, 2016).

The concept was initially generated in Rhinoceros, using Grasshopper, for parametric design of the mass and the skin of the building (Kensek and Noble, 2014). Further ahead, the architects turned to BIM, Autodesk Revit in this case, for performance-based design. By modeling various options and submitting them to wind-tunnel test, the design team was able to discover a combination of values for the twisting angle and the taper of the tower that reduced both wind-loads and material costs (Wujec, 2011). The result is a twisted stocking rotating 120 degrees along its height, while gradually narrowing to 55 percent of the initial dimension. The sustainable features go beyond the building shape, including rainwater collection, green roofs, wind turbines, water-efficient fixtures, lighting control, geothermal heating and cooling, and an intelligent skin.

4.4.4. HANGZHOU SPORTS PARK

In Hangzhou Sports Park (Figure 4.10), designed by NBBJ and CCDI, two iconic constructions are allocated: the Olympic Stadium and the Tennis Center. Their shapes are defined by repeated sculptural trussed geometries, which compose the exterior envelopes. Both projects were generated in Rhinoceros 3D, using Grasshopper, however the latter exceeded the main stadium in several capabilities. The algorithmic description used for the Tennis Center was expanded to include parametric geometry design, structural analysis, coordination in file exports, etc. (Miller, 2011).

Figure 4.10 - The Hangzhou Olympic Sports Park (Miller, 2011).

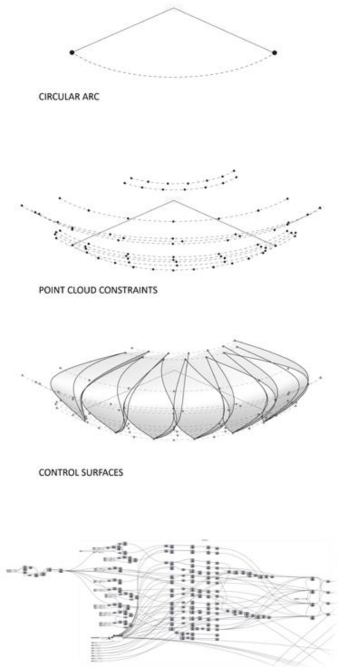


Figure 4.11 - The algorithm for defining the geometry of the exterior shell of the Hangzhou Tennis Center. A point cloud driven by circular arcs creates the control system for NURB control surfaces (Miller, 2011)

The base model of the truss modules, or "petals" was developed in Grasshopper (Figure 4.11) and this plug-in was, all through the project's development, instrumental for both design and documentation processes. Since multiple entities using different software collaborated on the project, file exports were necessary along the way.

The building's shape underwent structural analysis processes on the hands of CCDI structural engineering team. The team had to generate another alternative model. The analysis software required a structural centerline model only, so the team extended the stadium's algorithm to automate the generation of a wireframe model of the structure. The surface was also analyzed in order to visualize and quantify areas of curvature in the geometry, as each petal had to be subdivided into panels for fabrication.

Finally, another set of costume scripts had to be created to automate the export process of the model to 3D DWGs that could be then imported into Autodesk's BIM tool, Revit. Using Revit, the team was able to produce documentation sheets containing orthographic drawings of the exterior shell. However, this interoperability was not as successful as the team desired, since the model suffered considerable modification of data structure (e.g., from NURBS to specific meshes) and from a loss of parametric editing capability, when transferred to the BIM software (Kensek and Noble, 2014).

4.4.5. ELBPHILHARMONIE

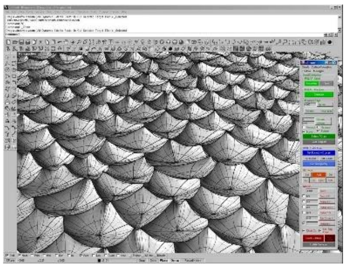


Figure 4.12 - One to One's Computational generation of one million sound-diffusing cells on Rhino (Architizer, 2016)

In Elbphilharmonie, Herzog and De Meuron's recently opened concert hall in Hamburg, a CAD-BIM cooperation was also necessary. The heart of the project, the Great Hall, has a unique sound-diffusing surface pattern, highly advanced and custom-developed in collaboration with acoustician Yasuhisa Toyota (Architizer, 2016). Comprised of 10,000 unique gypsum fiber panels resembling a "monochromatic coral reef", the room is a product of parametric design and fabrication techniques (Stinson, 2017). The architects contracted ONE TO ONE to create the digital data for the Great Hall's surface pattern that would convert it into a perfectly diffuse space where acoustic properties are the same in all possible location. One million sound-diffusing cells were, then, computational generated on Rhinoceros (Figure 4.12), and afterwards CNC-milled onto gypsum fiberboard panels (Architizer, 2016).

While the core was being developed in CAD software, a general model of the building was also created in BIM, by HOCHTIEF ViCon. From the first phases of the project, the 3D model was regularly updated, serving as base input for the analysis of geometrical conflicts and collisions. The BIM model also allowed a 4D construction sequence planning, through which conflicts and collisions could be identified and resolved (HOCHTIEF ViCon, 2015). A sectioned 3D model of the final project can be seen in Figure 4.13.

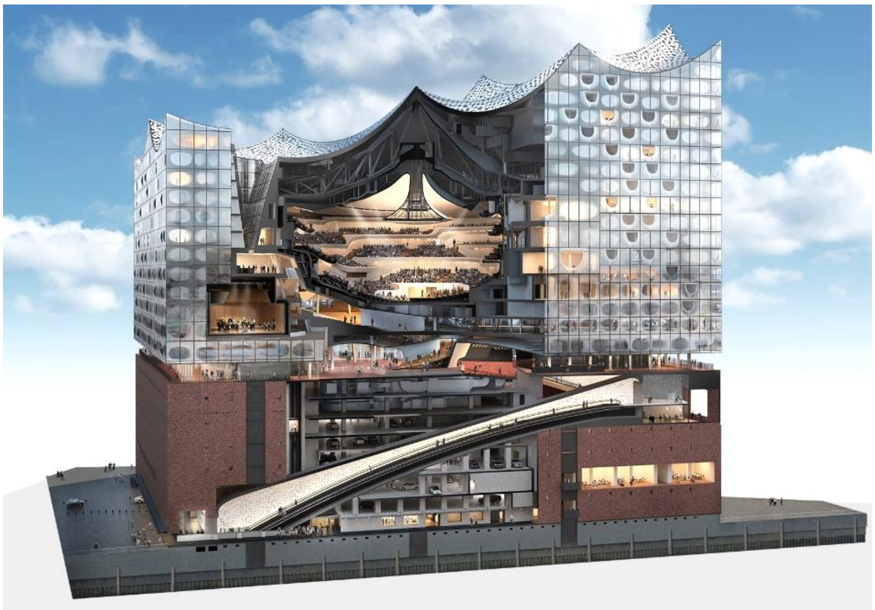
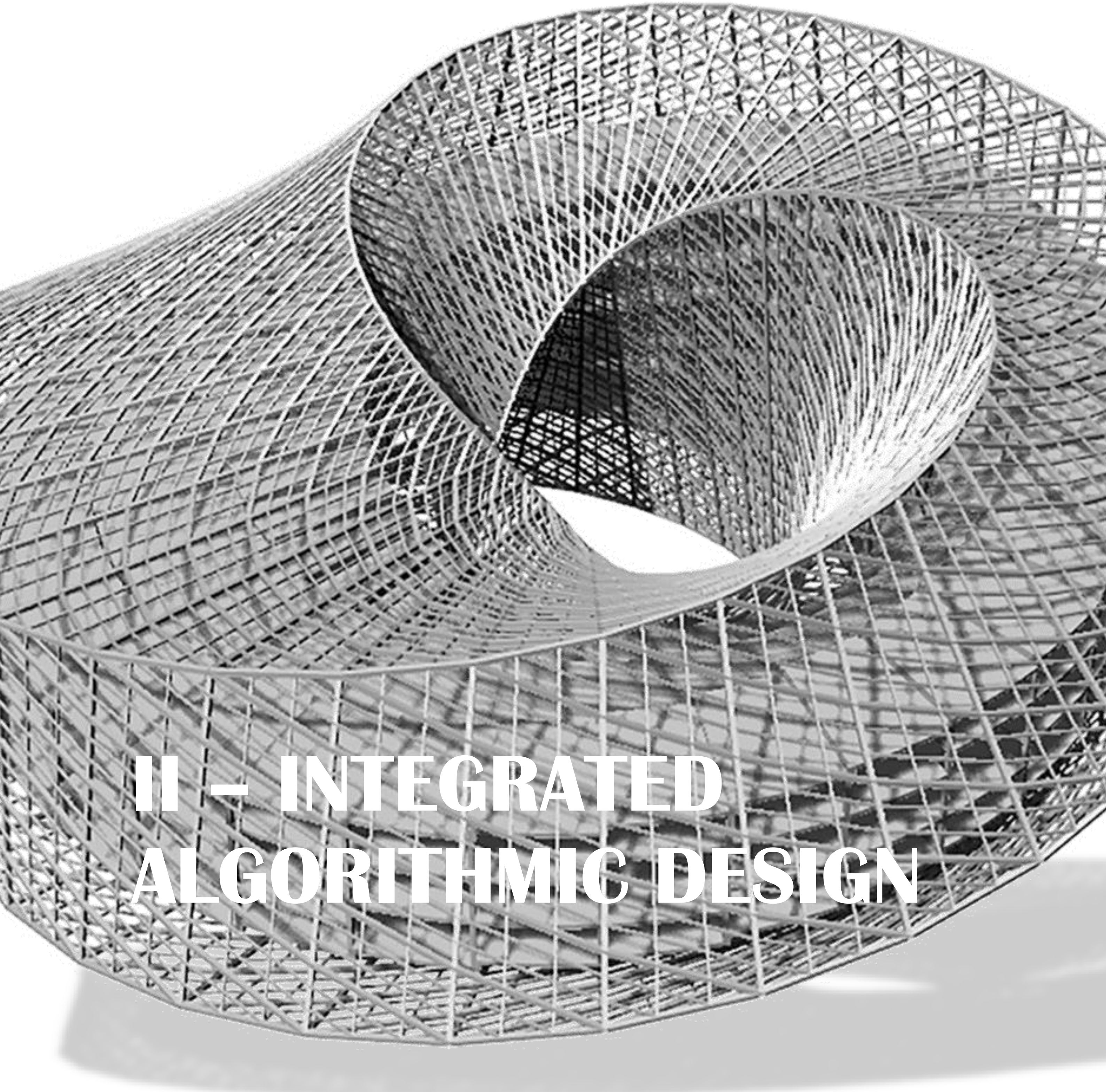


Figure 4.13 - Rendered Section of Elbphilharmonie (source: https://www.elbphilharmonie.de/en/elbphilharmonie)



II – INTEGRATED ALGORITHMIC DESIGN

5 DESIGN METHODOLOGY

Computation in architecture is but an infant, with a promising future in front of it. Meredith (2008) stated that architecture's current use of parametrics is still falling short from its true potential. Architects have made a superficial appropriation of technology, failing to understand the possibilities AD can offer in correlating multivalent processes or typological transformations, parallel meanings, complex functional requirements, site-specific problems, collaborative networks and more. In *Scripting Cultures*, Frazer (2011) reminded us that the introduction of computation in architectural practice did not have as purpose the mere replacement of drawing with typing. Algorithmic Design seeks something more. In the words of del Campo (2011: 55), the main challenge now set for the generation of programming designers is to move from "inventive articulated patterns" and "small-scale installations" to the full architectural scale. Only then can AD finally unleash the array of opportunities it has to offer this practice.

Different schools approach computation in different manners, for instance, parametric modeling can be a great design exploration method, although some might think, merely for the creation of senseless complex shapes. Others explore it as a mechanism for faster and more cost-effective fabrication, perhaps even an antidote to standardization (Burry, 2011), at the same time that repetitive and time-consuming tasks are avoided. Nevertheless, and with no disregard to all advancements made in exploring all the many advantages computation brings to architecture, the field is still lacking an integration process in order for the architect to profit from all the gains. Taking from the general philosophy of trendy BIM paradigm, parametric design has much to gain from an integrated communicative approach.

Our main goal with the development of this thesis is to explore the advantages an algorithmic approach can bring to the design process, and investigate, at the same time, how we can bring into that approach, the different paradigms within which architecture currently operates. We propose a methodology based on the development of a single computer program that describes not only the intended model, but also additional tasks, such as the required analysis. Moreover, it takes advantage of CAD, BIM and analysis tools, with little effort when it comes to the transition between them. We call this, Integrated Algorithmic Design (IAD).

The proposed method aims to cover the relevant phases of the design process, exploiting the features of the different tools, without bending to their imposed workflow. Hence, the manner in which the designer seizes the tools becomes part of his personal creative approach. The IAD methodology has the potential to encompass many paradigms and tools that architects may find relevant to the design process, for so long as they benefit from an algorithmic approach. However, for the evaluation of the methodology we restricted our investigation to the tools analyzed in the first part of this thesis. We considered them to be the main focus and trend of today's architectural practice. In this second part, a possible application of the IAD approach is outlined in three main stages, explained in the following order:

- CAD modeling for an initial exploratory form and concept stage (1st phase)
- BIM modeling for a more detailed stage in the process (2nd phase)
- Analysis integration (may occur during the 1st and/or 2nd phase)

A scheme of this practical application of the methodology can be seen in Figure 5.1. Using a programming environment as modeling tool, the user begins modeling his design intent within the CAD paradigm, and can visualize the result in a CAD tool. He may wish to include performance data in early stages of the design, hence, while still in the "CAD phase" of the process. The programming tool connects to the analysis software and exchanges the necessary data from and to the scripted model. In a more detailed phase, the user shifts to the BIM paradigm, visualizing the modeled geometry in a BIM tool. The performance analysis may instead be called upon on this stage, for which the process is identical. The detail modeled in BIM, can not only satisfy construction purposes, but may also include additional decorative elements in order to sell the project's image to a possible client.

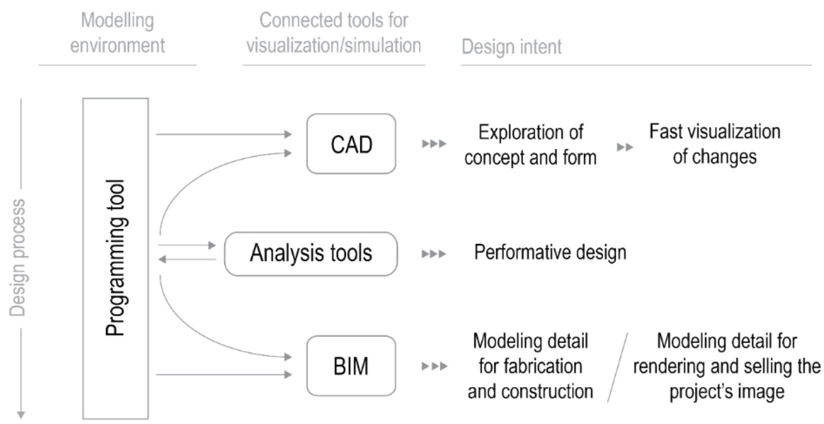


Figure 5.1 - Design process scheme with the application of the IAD approach

5.1. PROGRAMMING ENVIRONMENT

The modeling tool we propose is Rosetta (Leitão and Lopes, 2011), a programming environment that supports portable AD, and thus allowing the user to attempt an integrated algorithmic design process. Rosetta supports scripts written in various programming languages and allows the generation of their respective results in a series of CAD and BIM applications (see Figure 5.2). This is possible due to a front-end/backend architecture: the IDE connects programming languages (front-ends) suitable for beginners, such as Racket, Python, and Processing, to CAD or BIM applications (backends), as well as OpenGL for fast visualization. Currently, AD programs written in Rosetta can be generated in SketchUP, Rhinoceros, AutoCAD, Revit and ArchiCAD.

Figure 5.2 - Rosetta's backends and front-ends

Rosetta's portable nature is guaranteed by an abstraction layer containing the common functionalities amongst the modeling tools, like shape constructors and transformations. The abstraction layer is translated into the requirements of each backend, meaning that the same script is interpreted differently by each application. Despite this fact, similar geometry is produced in different backends if the modeling operations used are common to all. Most CAD applications accommodate these general operations and thus, the user can create portable AD programs between them. Only recently were BIM tools integrated in Rosetta (Feist et al., 2016) and, consequently, new operations incorporated. There are operations available for modeling the parts of a BIM model, including slabs, walls, columns, and beams, among others, and these operations also work for CAD tools. Nevertheless, many operations are only available in specific tools. With this in mind, Rosetta also provides the user with the possibility to work with specific functionalities of each tool, thus relinquishing the portability of his programs. However, this makes Rosetta available to a broader audience of Designers. We call this functionality tool-specific operations, and each backend has, in this case, a unique read of the code.

Just like Rosetta integrated different CAD/BIM tools, several analysis tools are currently going through a similar process. The tool is presently developing additional backends specialized for the generation of models for analysis, namely in Radiance and EnergyPlus (Leitão, Castelo Branco and Cardoso, 2017). Most analysis tools require simplified versions of the 3D models, hence, from the same script that produces a complexly detailed model in a CAD or BIM application, Rosetta's simulation backend generates only the simplified version of the essential elements needed for the analysis. For instance, depending on the analysis backend, slabs, beams, and columns might be interpreted as mere lines, planes, and surfaces. This automation process spares the user from the tiresome work of adapting or reconstructing the model for analysis purposes.

5.2. EXPERIMENTING WITH FORM AND CONCEPT

CAD tools are more advantageous in an initial stage of the model as they present a better performance when compared to BIM tools. Not only do they allow for the generation of more complex geometries that some BIMs just cannot process, but they also allow for a constraint free modeling workflow, where no sequences or precedencies are imposed. Furthermore, since they do not deal with the semantics inherent to BIM objects, CAD programs can generate geometry faster. For this reason, an architect can test a wider range of solutions for their design in a shorter time span.

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

Furthermore, the abstraction layer also knows how to translate BIM object functions into operations that CAD backends can understand. For example, a beam operation with two given points as parameters is understood by Rosetta/AutoCAD as a cuboid placed in space from one point to the other. Much like the beam command, similar abstractions exist for slabs, columns, walls, etc. This means that the architect may choose to shift parts of his program to the BIM paradigm while still modeling for a CAD backend. This will grant a smoother transition to BIM further ahead in the design process, while still allowing for form experimentation of the elements in a CAD environment.

5.3. TRANSITIONING TO BIM

Modeling in BIM, on the other hand, implies some significant loss of freedom, as some interdependencies are required and the order in which the elements are generated matters as well, unlike in CAD. The speed of the generation is also affected, since BIM applications must load and process more information in the creation of each object, as well as detect collisions and intersections between the elements and attempt to correct them automatically in each generation. Nevertheless, shifting to the BIM paradigm, as the project evolves into greater detail, has proven to greatly reduce time and effort spent on the modeling process, as well as on the production of documentation. In a BIM environment the architect can take advantage of all the information available in the families or libraries, saving a lot of time, as he does not need to model every single geometric element, as he would in CAD. In fact, in CAD, architects would not likely reach the level of detail needed for construction nor for detailed renderings, for example. These tasks are most commonly left to construction experts, in the first case, and rendering plus image editing software, in the second. However, BIM tools bring forth the possibility to reach higher levels of detail, in the same model and with less effort than ever before for the architect.

Transitioning to BIM, the designer can take advantage of all the semantics embedded in his program that, in CAD, was not as useful. Due to all the information included in Rosetta's abstraction layer, with the same operation that in a CAD backend produce only geometry, the user obtains, in a BIM backend, more data, such as default materials and all information associated to it like weight, density, cost, etc.

5.4. INCORPORATING ANALYSIS IN THE DESIGN

Performance-based design entails buildings to be conceived with performance as a guiding principle. This means that simulations and analysis of the building's performance must be included in the design process, in order to find the form that best fits the requirements. The performance criteria can be of structural, environmental, economic, ecological, spatial, cultural, technological, or other nature. Within these categories, the requirements can also be of numerous different natures according to the architect's design intent.

Performative architecture encompasses the architect's wish to create a building whose design is adapted to a specific set of conditions, thus focusing on finding the design whose analysis satisfies these conditions. The design intent also influences the way through which analysis data is incorporated in the process. For form finding explorations, optimizations methods are commonly used. This process entails an established communication channel between the analysis tools and the generative ones, where the later provides the former with a sequence of variations of the original model, produced according to the results obtained by the analysis. The cycle is repeated until the optimization criterion is met (see Figure 5.3, left). This is usually a rather time-consuming process and a computationally intense one. Additionally, it is often difficult to understand if the result obtained represents the global optimum, or simply a slightly better solution (Nguyen, Reiter and Rigo, 2014).

For a more controlled enhancement of the building's performance, architects can opt for a direct approach. When considering few criteria, the design solution that performs best can often be deduced without an optimization algorithm (Figure 5.3, right). As a practical example, one can envision an architect wishing to base the design of the façade elements on the building's solar exposition to guarantee the best thermal or lightning conditions in the interior space. In such a case, he would only need to perform the analysis once in order to collect the solar exposition data, and input it into his algorithmic description of the design. This process, while not so adequate for multi-criteria optimization, is simpler and faster than optimization methods and grants the user full control over the process and the final result.

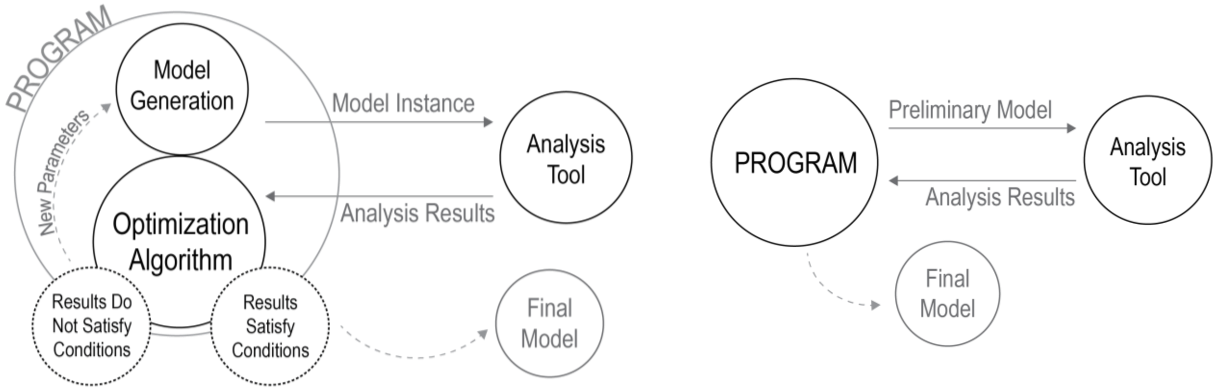


Figure 5.3 - Optimization loop scheme (on the left) and a scheme of a direct use of analysis results to produce the final model (on the right)

Using Rosetta, just as one single program can be interpreted by CAD and BIM applications to generate similar geometries in each one, the same is now possible with analysis tools, like Radiance and DAYSIM. Furthermore, these tools give back information retrieved from the analysis they perform. Using an algorithmic based approach with Rosetta, the results are simultaneously transferred to our program, where they can be used in the modeling process, or further evaluated in an optimization process. In the last case, there is a continuous loop of information transfer between the model and the analysis tool until a defined condition is met.

5.5. SELLING THE PRODUCT

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

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

What, in a manual approach, is normally the last task to be executed, and one the architect is usually unwilling to repeat many times, given the time it consumes, is now, using an algorithmic approach, an exercise that, after the initial burden, can be repeated countless times with no effort, for every variation the model may have.

6 CASE STUDY: ASTANA LIBRARY

In order to evaluate the validity of our proposed approach, we selected an architectural case study to model. The project in question is Astana National Library (ANL) for Kazakhstan, designed by BIG architects (visible on Figure 6.1). The chosen case belongs within the category of buildings that benefit from an algorithmic-based approach to design. ANL, along with most endeavors from BIG studios, is characterized by complex and somewhat repetitive geometry. Designing buildings of this nature is faster and easier through algorithms, and considerably painful via the manual modeling approach. The same can be said for the use of BIM tools in automating the construction process, as higher levels of building complexity cannot be dealt with using traditional construction methods.

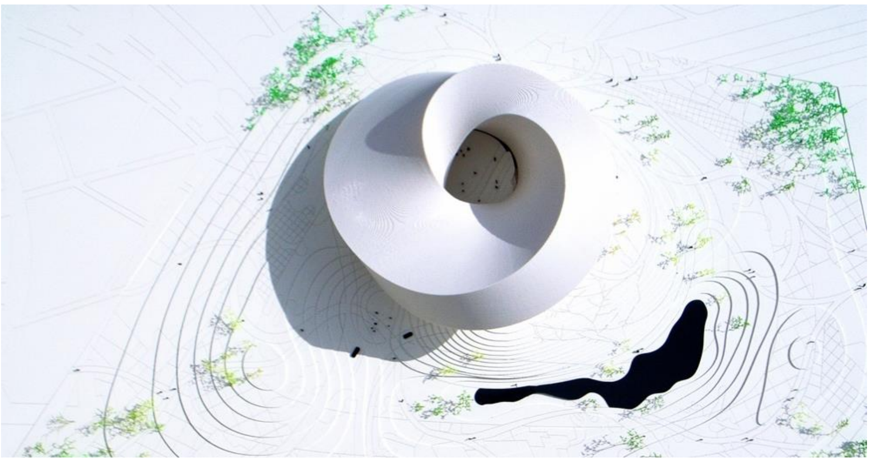


Figure 6.1 - Astana National Library model (source: http://www.archdaily.com/33238/national-library-in-astana-kazakhstan-big)

6.1. BJARKE INGELS GROUP

Figure 6.2 - Bjarke Ingels, founder of BIG (source: http://www.designerwerktag.de/wp-content/uploads/2010/02/Bjarke_Ingels_yesismore-590x951.jpg)

BIG is a group of architects, designers, builders, and thinkers operating within the fields of architecture, urbanism, interior design, landscape design, product design, research and development. Currently based in Copenhagen and New York, Bjarke Ingels Group was founded in 2005 by Danish architect Bjarke Ingels (in Figure 6.2). With more than 40 projects awarded with prizes, the Studio is nowadays led by twelve partners. Among them we find Thomas Christoffersen who has worked on every notable project since the VM Houses, and has led one of the most global ones, the Astana Library (BIG, 2017).

Named in 2016 one of the 100 Most Influential People in the World by TIME Magazine, and having received numerous awards and honors before, Ingels has developed a reputation for designing innovative buildings, both programmatically and technically, that are always cost and resource conscious.

In point of fact, as part of their design process, Bjarke Ingels group has created an internal technology-driven special projects unit, BIG ideas, that hopes to expand the scope of the architect from its traditional dimension. Exploring within the digital and material realms, they are increasingly relying on technical simulations that would traditionally be in the engineering scope. As such, daylight, thermal exposure, airflow, turbulence, wind, traffic flow, and other factors can now be simulated and controlled, enabling the architects to make designs that are literally shaped by the forces surrounding them (BIG, 2017). Bjarke Ingels describes his work as pragmatic utopian architecture that emerges out of a careful analysis of how contemporary life constantly evolves and changes. In short, they look at the "BIG picture" (World-architects, 2015).

6.2. ASTANA NATIONAL LIBRARY

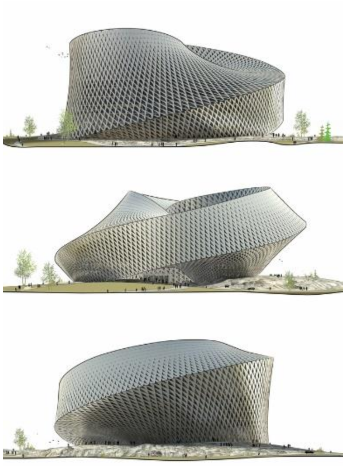


Figure 6.3 - ANL elevations (source: http://www.archdaily.com/33238/national-library-in-astana-kazakhstan-big)

The design of Astana National Library went beyond a mere architectural challenge, as the building intended to represent one of the future cornerstones of Kazakh nation identity. Following the country's separation from the Soviet Union in 1991, the government moved the capital in 1997 from Almaty to Astana. The library, located in the new political and cultural center, should then not only accumulate history but also provide a foundation for the nation's new future (Fairs, 2009).

BIG's project (see Figure 6.3) was, in 2008, awarded first prize in the open international design competition, which included nineteen entrants, including, Norman Foster and Zaha Hadid. Later named after the first President of the Republic of Kazakhstan, Nursultan Nazarbayev, the projected building has also received the Future Community category award at Cityscape Dubai, on October 2009 (World Architecture News, 2009).

With a size of 45000m², the edifice was hailed as being both modern and rational, and anchored in a classical vocabulary of traditional libraries. In the article by Fairs (2009) at Dezeen website, Bjarke Ingels tells us that "the design of the National Library combines four universal archetypes across space and time into a new national symbol: the circle, the rotunda, the arch and the yurt[1] are merged into the form of a Möbius strip" (see Figure 6.4). In Christoffersen words, "the envelope of The National Library transcends the traditional architectural categories such as wall and roof. Like a yurt, the wall becomes the roof, which becomes floor, which becomes the wall again" (Fairs, 2009).

The authors' description of their creation leaves no doubt as to the impossible task modeling ANL manually would be. Naturally, the architects had to use programming to generate their concept in 3D. According to Ramboll UK's engineers (Morgado, n.d.), BIG used the Grasshopper plug-in for Rhino parametric modeling software to generate their design.

The current status of the project is still merely ideal though. As reportd by Ferro (2014), its completion was scheduled for 2012 but, having encountered the bribery and corruption involved, the architects pulled out.

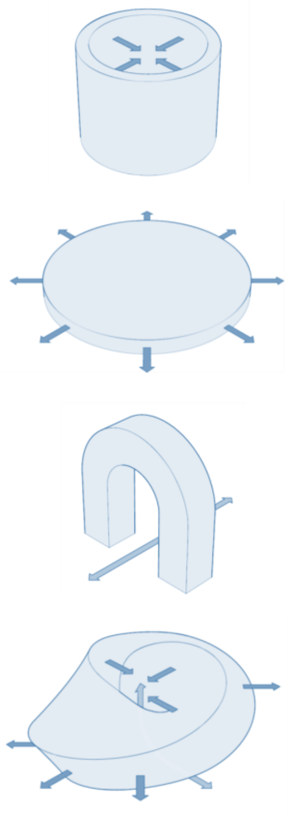


Figure 6.4 - Diagram of the elements that make ANL's shape concept – edited (source: http://www.big.dk/#projects-anl)

6.3. MODELING THE CASE STUDY

At the initial stage, all elements were modelled for CAD tools, as they present a better performance when compared to BIM tools. Since they do not deal with the semantics inherent to BIM objects, CAD programs can generate geometry faster. For this reason, an architect can test a wider range of solutions for their design in a shorter time span. In conclusion, for an initial phase of exploration these tools are more advantageous. Section 7 explains the initial phase of modeling of Astana.

As the project evolved into detail, we shifted our approach to a BIM paradigm where we could take advantage of all the information available in the families or libraries. Modeling in BIM implies some loss of freedom, as some interdependencies are required and the order in which the elements are generated matter as well, unlike in CAD. The speed of the generation is affected as well, since BIM applications must load and process more information in the creation of each object as well as detect collisions and intersections between the elements and correct them automatically. Section 8 describes some of the advantages we encountered in this transition, such as a general simplification of the code, and some of the issues we had to overcome as well.

Afterwards, in section 9 we will focus solely on detail modeling. The BIM paradigm allows for an easier and faster development of the final modeling stages. Taking advantage of already existing library parts or families, the architect can save a lot of time, as he needs not to model every single geometric element, as he would in CAD. In fact, in CAD no architect would likely reach the level of detail needed for construction, nor for detailed renderings, for example. These tasks are most commonly left to construction experts and their software, in the first case, and rendering plus image editing software, in the second. However, BIM tools bring forth the possibility to reach higher levels of detail in the same model, and using our proposed methodology, with just one written program. Nonetheless, reaching higher

[1] A yurt is a circular tent of felt or skins on a collapsible framework, used by nomads in Mongolia, Siberia, and Turkey. It is part of Kazakh history and heritage (Ramboll, 2010)

levels of detail implies the loss of portability. As we began using specific BIM objects, we had to narrow down our range to only one modeling tool, in this case, we chose ArchiCAD.

In section 10 we also describe how we incorporated performative analysis in the building's design, as the architects originally envisioned it. However, we performed this task using an alternative approach, in accordance to our proposed methodology. Algorithmic-based analysis was made possible by the extension of Rosetta to new analysis backends. Just as the same script can be interpreted by CAD and BIM applications to generate similar geometries in each one, the same is possible with analysis tools, like Radiance and DAYSIM. Furthermore, these tools give back information retrieved from the analysis they perform. Using an algorithmic-based approach with Rosetta, the results are simultaneously transferred to our program and can immediately be used in the modeling process.

Finally, we finished Astana with the modeling of detailed elements for rendering purposes. Selling the product has always been an important part of the business for the architects, but as technology advances, new possibilities emerge for ever more realistic models. Section 11 expands this discussion topic and describes the modeling process of these specific type of model detail, in our case study.

---

7 MODELING FOR CAD

The modeling processes used for our case study can be divided in two main sections: interior blocks and exterior façade. These two elements are relatively autonomous from each other and could be developed in parallel. These elements were conceived in phases, each one increasing the detail of the model and respecting the order of interdependencies that characterize construction, and therefore, BIM modeling.

This sequential arrangement of phases not only ensures a correct placement of elements throughout the project, but also ensures the automatic propagation of changes through the model when the program is modified. CAD applications do not necessarily oblige the user to have this sort of concerns, however, good programming practices dictate that we should not only consider the order in which the elements are created, but also the use of intermediate abstractions that organize the code in a logic and understandable way for other users.

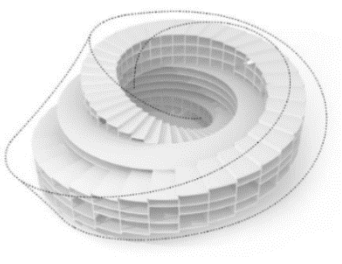


Figure 7.1 - ANL Internal structure (source: http://www.big.dk/#projects-anl)

The interior space (see Figure 7.1) can be described as an interlocking of two structures: a ring-shaped volume accommodating the library's archive and a spiral volume, orbiting around the first one, intersecting it inside and out, containing additional functions, such as reading and study rooms, auditoriums, museum and administration. Together, the two structures, create a building that evolves from a horizontal organization where all functions are placed next to each other, to a vertical organization where they are stacked on top of each other, as can be seen in Figure 7.2. In between the two, the building holds a diagonal organization combining vertical hierarchy, horizontal connectivity and diagonal view lines, as Figure 7.3 illustrates (Fairs, 2009).

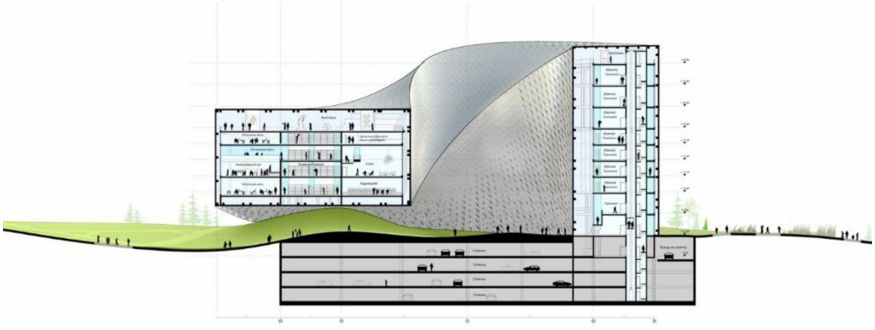


Figure 7.2 - ANL section (source: http://www.big.dk/#projects-anl)

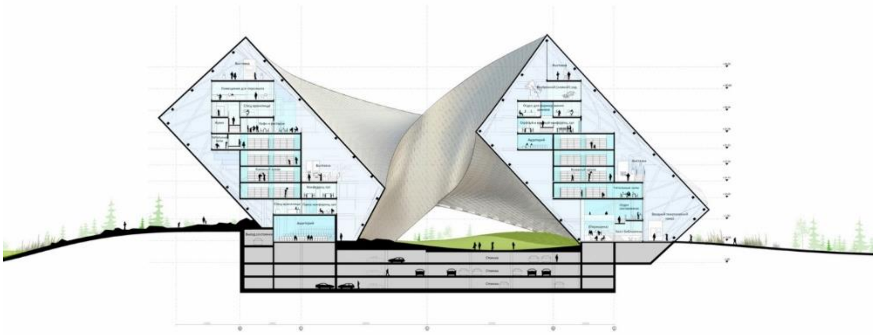


Figure 7.3 - ANL section (source: http://www.big.dk/#projects-anl)

7.1. SLABS

We began by modeling the slabs. As this building portrays a rather complex arrangement of floors, these elements are key to a correct understanding of the spatial composition. The remaining elements were allocated in accordance to the defined slab positioning.

The ring-shaped volume is divided in four floors, defined by circular slabs with circular holes in the middle (see Figure 7.4).

Figure 7.4 - Ring volume slabs

The spiral volume is formed by small slab pieces, each covering 10º of the total ring in the original model, positioned at ascending or descending heights and with increasing or decreasing radiuses (Figure 7.5). The blocks that form this volume follow the frame grid of the façade. The 10º result from the division of one complete loop (360º) by the thirty-six frames of the façade. Although the total height of these blocks is identical to the ring volume, they have only three floors placed at variating heights from block to block (see Figure 7.6's scheme). Figure 7.8 shows the spiral volume with all its composing slabs.

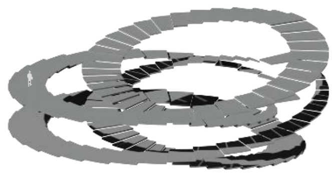


Figure 7.5 - Spiral volume top and bottom slabs

Figure 7.6 - Middle floor scheme of the spiral block

The spiral can be divided in two loops, where the radius and height of the blocks evolve, according to sinusoidal functions, along the angle of the loop. As explained in Figure 7.7, in the first section of the loop, the blocks begin with the same radius as the ring and one "block-height" below. The radius decreases as the angle varies from 0 to π, and increases again from π to 2π, returning to the original value. The height increases all through the loop (from 0 to 2π) until the final block is on top of the ring (two times the "block-height"). In the second loop the radius suffers an increase as the angle varies from 2π to 3π, and decreases again to meet the ring radius at 4π. The height begins its descent at 2π, landing with the initial value from the first loop at 4π.

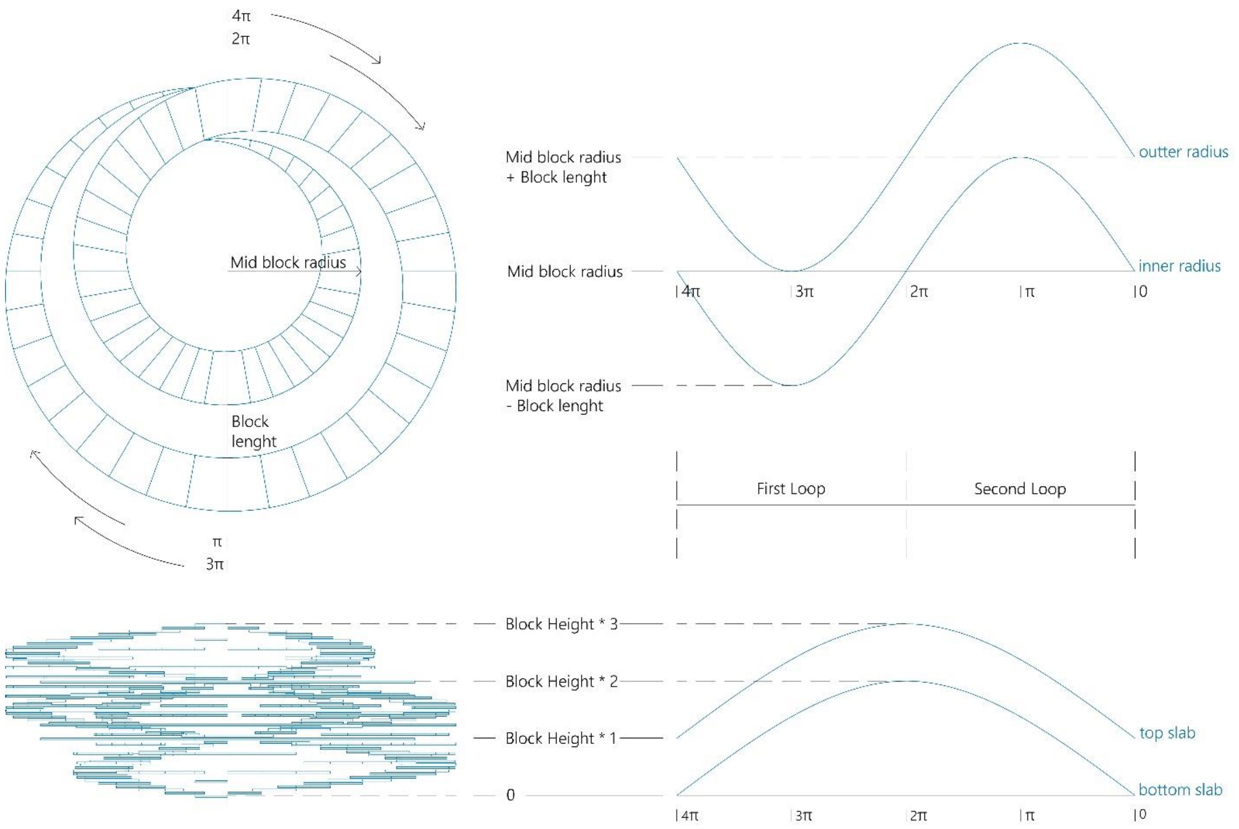


Figure 7.7 - Plan and section of the slabs on the left. Radius and height variation along the loops on the right

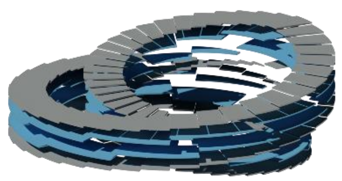


Figure 7.8 - Spiral volume with interior slab-pieces highlighted in blue

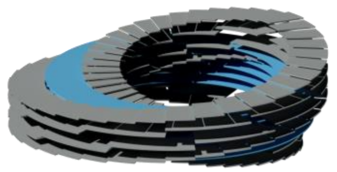


Figure 7.9 - Spiral volume slabs sectioned to avoid collisions with the middle volume (highlighted in blue)

To model the ring-shaped slabs in CAD applications, we defined five surface circles at the appropriate heights, with the outer radius of the block, and extruded them according to the desired slab-thickness. Simultaneously, five other surface circles were created, with the inner radius of the block coinciding with the previous. The second set was then subtracted from the first.

For the slab pieces, two list of points were calculated for each, containing the inner and outer contour respectively, with exception to the heights where they collide with the middle volume. In such cases the points are recalculated to create only part of the slab to avoid the juxtaposition (see Figure 7.9). From these lists, we were then able to produce surface polygons that were afterwards extruded to the correct slab-thickness.

7.2. COLUMNS

The columns act as support of the floating blocks. They connect the slabs to one another receiving their loads and redistributing them downwards. They also receive some weight from the façade structure redirected through crossbeams. Two different sets of columns were identified, the ones supporting the ring volume slabs (Figure 7.10) and the ones supporting the floating blocks of the spiral volume. The columns belonging to the first set are all equally sized, whereas the ones from the second set have different heights, in accordance with the intersections of the spiral volume with the ring (Figure 7.11). Despite the various heights, the section is also common to all, a 0.5x0.5m square, and following the slab positioning, they are all aligned with the frame grid in plan (see comparison between the frame grid and the columns positioning in Figure 7.12).

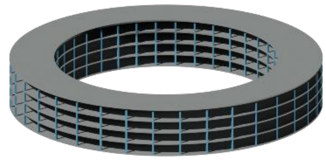


Figure 7.10 - Columns on the ring volume highlighted in blue

Since the building's geometry is not conventional, the structural organization of floors is a peculiar one and the vertical elements connecting the blocks to the façade take structural precedence over the horizontal elements. Hence, giving priority to the pillars over the slabs, we opted for continuous columns piercing the block-slabs from top to bottom, holding the structure together. For each block, two sequences of columns were needed, one for the interior perimeter and one for the exterior. Considering the first set of columns, each sequence has the exact same number of elements as the structural frames of the façade. In the second set, this number duplicates as the spiral volume circles the ring twice.

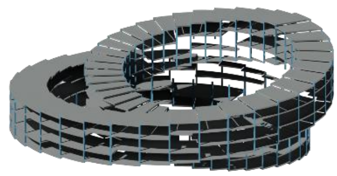


Figure 7.11 - Columns on the spiral volume highlighted in blue

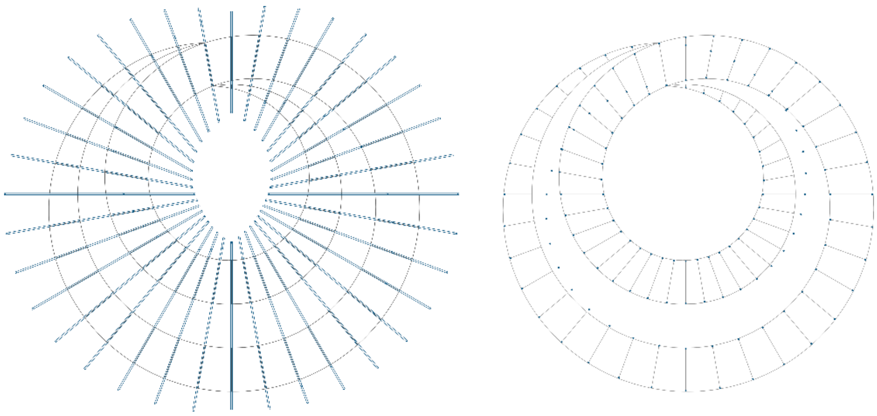


Figure 7.12 - Slabs in plan with frame grid on the left and with columns positioning on the right

On our experimental CAD approach, we conceived the columns as extrusions of squared surface. All columns are generated in their correct location and rotated around their middle axis to guarantee their correct facing towards the middle. The extrusion covers the total height of the block in the ring volume, crossing the four floors. In the spiral volume, the bottom and top heights are calculated for each column separately to guarantee the proper intersection of each block with the ring.

7.3. BEAMS

Three different series of beams were modelled for the interior: the beams directly underneath the slab pieces, supporting them and assuring the connections between them; the crossbeams of the ring volume, setting a diagonal grid of weight distribution alongside the pillars; and the connection beams between block pillars and façade frames.

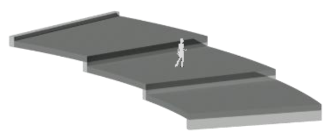


Figure 7.13 - Spiral slabs with beams in between

The first set of beams (Figure 7.13) were modelled sweeping a rectangular surface along a path. The rectangle width is invariable and equal to the columns' dimension (0.5m). The height of the beam is automatically calculated to fill in the gap between slab pieces. The path defined for each sweep is the linear side contour of each slab piece.

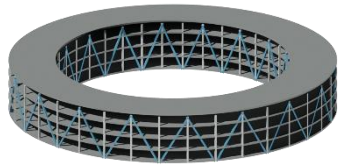


Figure 7.14 - Cross beams in the ring volume

The cross beams for the ring volume (Figure 7.14) were also generated through sweeps, although in this case their section is identical to the columns, 0.5x0.5m. The paths along which the squares were swept consist of lines drawn in space between the middle points of the columns top and bottom bases.

The connection between block pillars and façade frames is made through beams with a similar section - 0.5x0.5m. Each frame is supported by eight beams distributed around the blocks, and the place where these beams intersect it is equal for all frames. This means the frames are alike in all instances of the rotation to facilitate construction issues. Figure 7.15 explains exactly how the beams are arranged: primarily, when the blocks align vertically, and secondly, when they start twisting, the block corners are connected to the same frame points. The third section shows the point where the corners change reference points, all shifting to the next designated point in the frame, in preparation for the final instance, when the blocks align horizontally. The exact same process occurs backwards for the second part of the rotation, until the block reach the vertical alignment again.

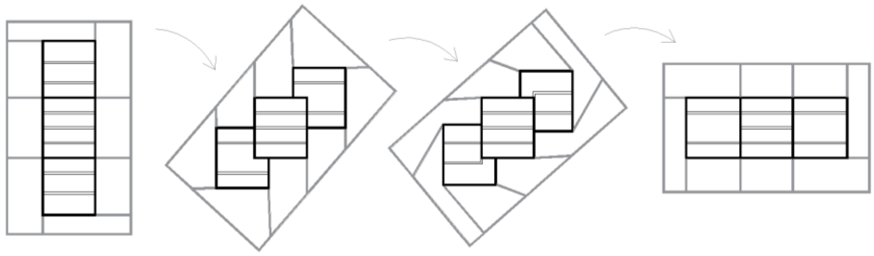


Figure 7.15 - Connections beams scheme along the rotation

7.4. WALLS

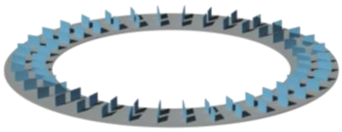


Figure 7.16 - Library book walls

The inner circular core, which we have been referring to as ring volume, was meant to contain the Presidential Library's collection. To fulfill this purpose the architects imagined it as a continuous space, yet simultaneously divided by the frame grid that defines the rhythm of the whole structure. With every new frame, a double book wall is placed separating the spaces, yet allowing the communication between them (see Figure 7.16).

In our initial approach, the walls were built from vertically swept rectangles (intended to be the wall profile) along virtual lines created in plan from the beginning to the end location of each wall. On each floor of the middle ring, and with the same spacing as the frames, groups of two walls are placed perpendicular to the slab contour, leaving three openings for passing: two at the sides and one in the middle.

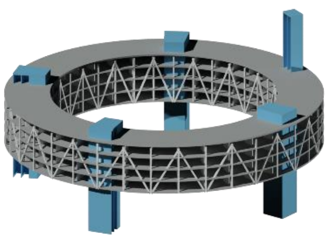


Figure 7.17 - Cores placed in the middle ring

The central volume has still another wall set: "five concrete cores (Figure 7.17), spaced evenly around the spiral, provide lateral stability and reduce the length of the cantilevers" (Archello, 2010). These elements are present in all floors of the bloc, and, besides functioning as a structural element, they also cluster circulation and sanitary areas. The core supporting the building at the point where the blocks align vertically is different from the others. The normal core configuration features two elevators and an emergency staircase, female and male restrooms, and two structural voids through which lighting, plumbing and ventilation systems can be distributed vertically. The different core configuration has two thirds of the original size as it does not incorporate restrooms. Plans of both can be found in Figure 7.18.

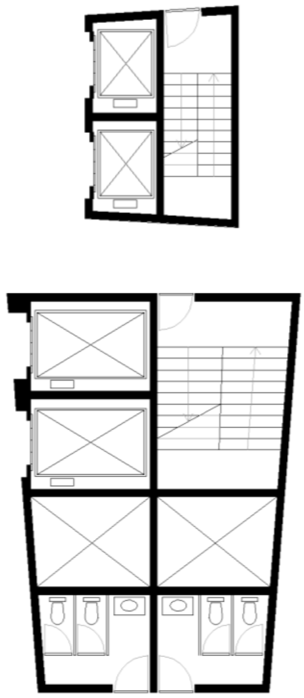


Figure 7.18 - Normal core plan on the bottom and exceptional core plan on top

The cores are distributed along the middle ring, taking the place of some of the library walls, or more accurately, incorporating them in their design (see plan in Figure 7.19). As structural elements, they carry the weight of the structure, delivering it to the ground. Since the architecture project had a parking lot designed below the ground, the cores also extend the vertical assess (elevators and emergency staircases) all the way to the underground. The different core set rises to the highest point of the building, since it is placed in the exact point where the structure aligns vertically, it could rise up to three “block heights”. For CAD applications, these walls were modelled using a similar process to the one described for the library walls.

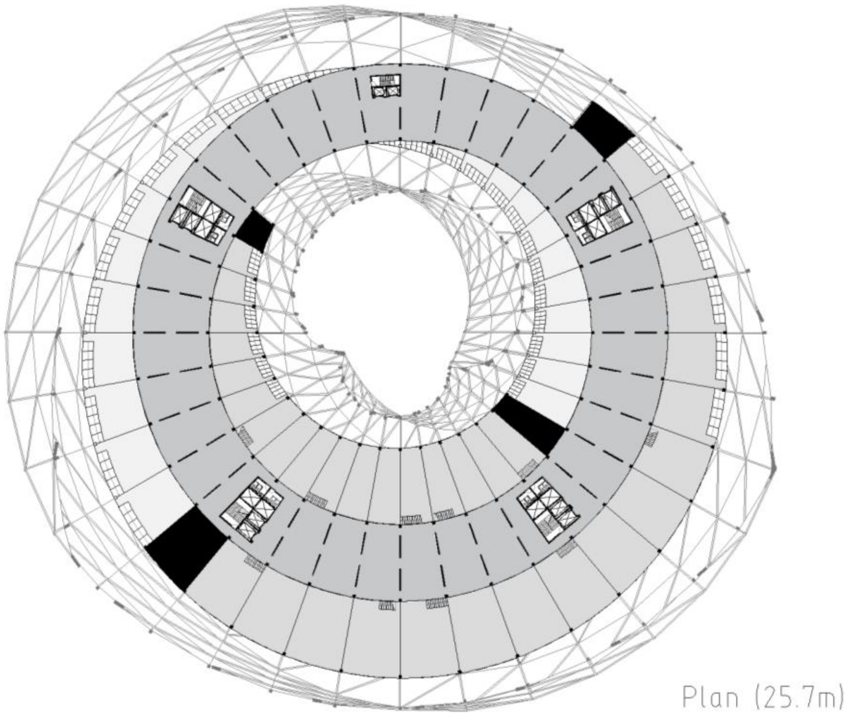


Figure 7.19 - Building plan at 27,5m (3rd library floor)

7.5. SPIRAL STAIRS

The spiral volume orbiting around the middle ring is formed by slab blocks of decreasing or increasing height. The result, on the top and bottom floors of this set, is a long public pathway that circles the inside and outside of the ring, where people can walk through the building in two continuous loops, moving up and down as they go. Figure 7.20 shows, in plan, where the stairs are placed. The gap from slab to slab is not significant, however, it is still too big for a human to climb. For this reason, spiral staircases are introduced on the sides, accompanying the slabs’ rotation. Since the height to cover between blocks is modest, these stairs are made of very low and very long steps.

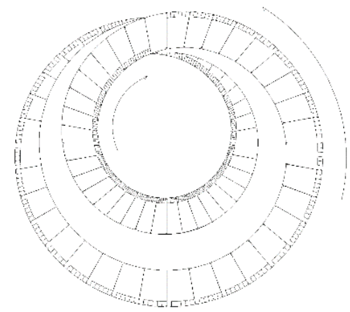


Figure 7.20 - Spiral stairs plan scheme

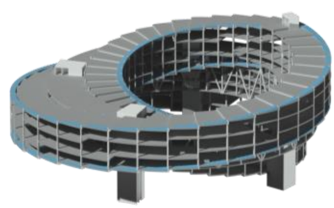


Figure 7.21 - Spiral staircases placed on the spiral volume

Each block was assigned a set of six steps. Each of these steps was generated through a loft between two surface-polygons. Each of the polygons was generated from a list of 8 points per step, four points for each side of the step, to assure an approximate curve. The two surfaces were needed, because on each block, the step height rises gradually but the base must remain the same. Finally, four sets of spiral steps with different radii create four different staircases circling all around the building in the inner and outer part, on top and on the bottom of the blocks circling the center. The result can be seen in Figure 7.21.

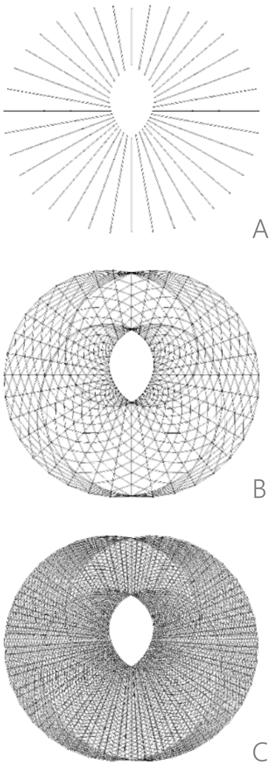


Figure 7.22 - Plan view of A: Frame grid; B: Frames and bays; C: Frames, bays and secondary structure

7.6. FAÇADE GRID

The façade wrapping around the entire structure is conceived as a double Möbius strip. As explained by Ramboll UK's engineers, the envelope structure is a steel frame truss flowing from a ten-story "tower" to a four-story horizontal volume, leaning at 45º along the way. It is supported by the ground at just four points, with the main load being taken by the central building, which acts as a torsion ring, suspending and cantilevering the rest (Ramboll, 2010).

The rectangular frames, 15.5m in height and 14.5m in length, are arranged radially around the intertwining volumes and linked to every block corner by longitudinal beams, as explained in the previous section. Thirty-six frames stand around the center equally spaced, by 10º angles, and with an increasing rotation factor for the section. Each frame suffers a 5º increment to complete a full 180º spin along the circle. Four sets of beams connect the corners of all frames in four continuous lines. The frames are also linked to each other by a skewed net of bays creating a firm structural web. From frame to frame four beams on the short side of the rectangle, and seven on the long side are placed with a slight deviation, to provide shear strength.

For CAD applications, we began modeling the beams that form the frames sweeping 0.5.x0.5m squares along rectangular polylines, virtually generated through four calculated locations for each frame (Figure 7.22 and 7.23-A). The frames were then composed of four continuously sweeped beams. The four point lists were also used to create the contour beams, reorganized to unite the corners of the frames. For the bays, a matrix of points needed to be created, containing a list for each frame, with more locations than the four points previously used. These locations were then used to produce virtual lines along which the same squares were sweeped (Figure 7.22 and 7.23-B).

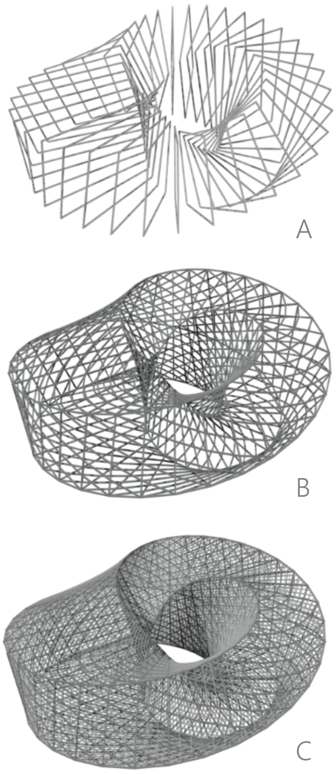


Figure 7.23 - Perspective view of A: Frame grid; B: Frames and bays; C: Frames, bays and secondary structure

Within the main grid, a secondary one was placed to set the metrics for the photovoltaic panels and to serve as support for them as well (Figure 7.22 and 7.23-C). This secondary structure is composed of steel bars of a smaller diameter - half of the main bars (0.25x0.25m). For this secondary grid two new bars are placed between the steel beams, which means the divisions made by the original grid are triplicated. The modeling process of these beams, for CAD application, was analogous to the one performed for the main grid, only with a smaller section.

7.7. PANELS

The existing façade grid not only serves as partial structural support for the whole building, along with the cores inside, but it also holds the glass panels that separate interior from exterior space. The glass panels are triangular shaped in order to accommodate themselves to the revolving structure in which they are placed. For each grid section, there are two glass panels. For CAD, they were modelled via surface-triangles.

Over the façade glass layer, we find yet another set of triangular panels of smaller dimensions, the photovoltaic lattices. The design of the pattern composed by the photovoltaic panels will be further discussed in section 10. For the initial placement experiments, we considered different scenarios for distribution of the variety of sizes the panels could have. Figure 7.24 shows a random assignment of the sizes on the left and a gradient one on the right.

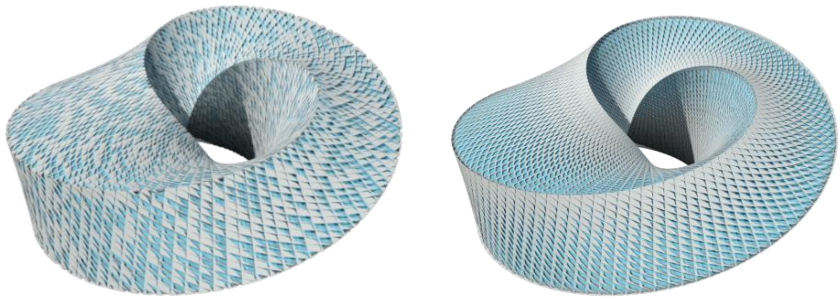


Figure 7.24 - Façade panels distribution: random (left) and gradient (right)

8 TRANSITIONING TO BIM

After a phase where we took advantage of the CAD tools capabilities to quickly experiment different alternatives, as the programming approach stabilized, we entered a phase where it was important to include additional detail, which motivated the use of BIM tools.

As we transitioned, we verified an overall simplification of our code thanks to the abstractions already programed in Rosetta, that allow the CAD applications to recognize BIM operations, and convert them into basic geometry elements. For example, a beam command, with two given points as parameters, is understood by CAD applications as a cuboid placed in space from one point to the other. Much like the beam command, similar abstractions exist for slabs, columns, walls and panels. This meant we could transform our functions to pair the BIM paradigm, simplifying the operations implemented, while maintaining the portability of our code for CAD applications.

With this shift, more information is automatically instilled in our model, consequence of the BIM methodology. Such information is, nonetheless, inexistent when the model is produced in CAD, as these programs deal only with geometry and do not support the extra load of data that BIM implants in its objects. Nevertheless, the portability faculty is a great advantage, since it is quite common, in the development of an architectural project, for the designer to still want to experiment small or big changes in the shape or size of the building, in advanced stages of the modeling process. Being able to change backends in later stages of the project as well, means he can still experiment with CAD tools without losing all the information already instilled in the model.

8.1. SLABS

As we migrated to BIM applications, the code was simplified with slab operations, already defined in Rosetta. The slab operation takes place over the extrusion of a surface polygon, mentioned in section 7.1.

The ring-shaped slabs were defined by arcs from π to π with a circular slab-opening. For the slab pieces, the same lists of points were used, now inserted in the slab operation.

The BIM operations used in our code are also available for CAD applications through Rosetta, which means that our modified operations for BIM are still capable of producing the same results in CAD backends. Nonetheless, several changes were introduced to Rosetta to correct some discrepancies. For instance, BIM applications generate slabs by default under the level to which they were assigned, while the slab operation for CAD was extruding them upwards.

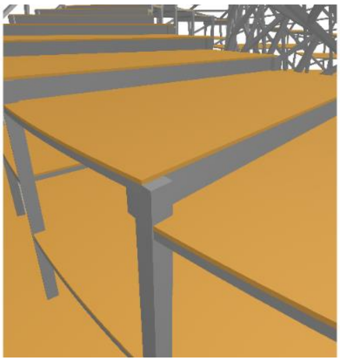


Figure 8.1 - General slab family with defined thickness and coating

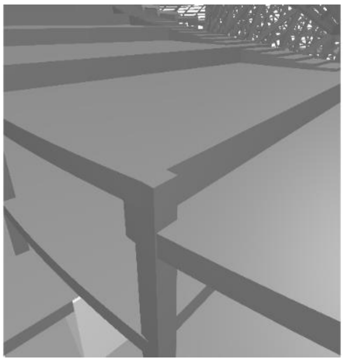


Figure 8.2 - Modified slab family with no coating and changeable thickness

8.2. COLUMNS

Switching to BIM the column function takes place over the extrusion. Width, height and angle were converted into specific parameters of the object, but no calculations needed to be altered. However, our default slab function in Rosetta did. The thickness of the slabs and their positioning in the levels created varies according to the family chosen. This makes it difficult to achieve a perfect alignment of the columns with the slabs. Our solution was to redefine the default slab family in the backend, electing one that allowed us to control its positioning and thickness. In the case of the BIM tool ArchiCAD, a generic structural family with no coating served the purpose. Figure 8.1 and Figure 8.2 show both these cases.

8.3. BEAMS

The slab beams, for a BIM backend, resorted to the beam operation that needs only two points and dimensions as input. The beams were then, defined using the same two points at the end of each slab piece, that were being previously used to generate the path line. The same occurred with the connection beams, visible in Figure 8.3.

The transition of the code correspondent to the cross beams required some changes in the implemented function of Rosetta, since we needed these beams to have a particular skill only columns possess. The cross beams of the ring volume should be sectioned by the top and bottom slab, just like they would in real construction, instead of prolonging their edges into the slab. To surpass this issue, we implemented a trim parameter that, when activated within the beam operation, produces a column with the desired inclination instead. Figure 8.4 presents the problematic case in the CAD model. Figure 8.5 shows the correct solution in the BIM model.

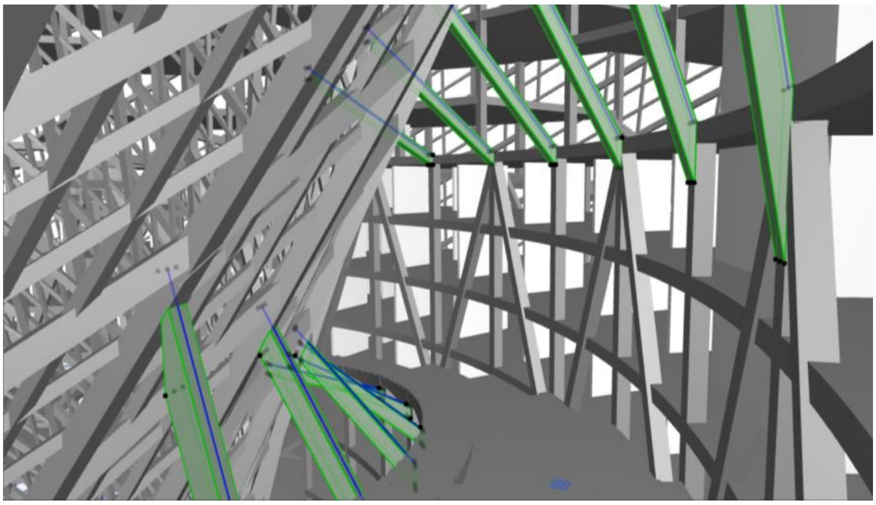


Figure 8.3 - Connections beams selected in the BIM model

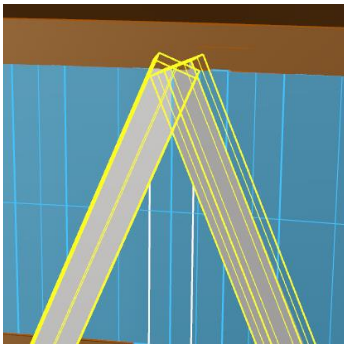


Figure 8.4 - Beam command result in the CAD model

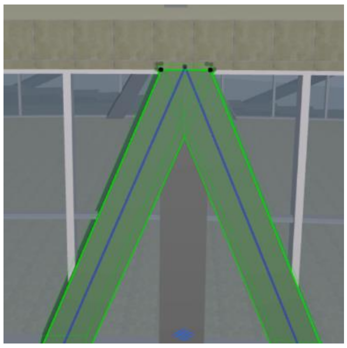


Figure 8.5 - Correctly sectioned beam in the BIM model

8.4. WALLS

For a BIM application, there is no need to model the wall profile, as we did for CAD (section 7), since the program assumes the family values by default, such as thickness and material for instance. The wall command required only two points, the beginning and ending location of each, and the heights of the current level and the one above. The code designed to distribute the two wall sets along the frame's spacing and the four floors was reused, just like the code producing the cores. Comparing the necessary function used to model the objects in CAD and BIM, we can, once more, verify the simplification obtained in our BIM program, where a sweep of a surface rectangle along a line was converted into a mere "wall".

8.5. SPIRAL STAIRS

Shifting to BIM applications, the natural way of producing stairs would be recurring to the object's libraries available. Unfortunately, stairs with a non-constant radius, as is the case of ANL, are impossible to produce using the stairs objects provided by ArchiCAD. A sequence of stair objects composed by one step only was attempted, each one having its own radius. However, the orientation of the steps was still incorrect, as each step needed to have an initial and a final angle, and the object did not provide such parameters. Ultimately, we used the slab command to design the spiral steps that configure the public pathway. As these spiral steps seemed more of a scaling of the original slab staircase to a human walkable dimension than an isolated staircase on its own, we considered the slab command to be equally appropriate. The steps are then extensions of the blocks' slabs. The code from our CAD approach was reused, six steps with ascending level heights were created for each block, with six increasing thicknesses to assure the bases touched the block slab. The process is repeated four times to cover all four paths.

8.6. FAÇADE GRID

In our BIM approach, we simply used the beam operation in replacement of the sweep of squares along lines. The beams were defined by a beginning and ending location, with the same width and height - 0.5m, predefined as the default family's values. The lists of 4 locations per frame generated for CAD were reused, just like the matrix for the bays. A problem arose though, since BIM applications are incapable of producing vertical beams, nor horizontal columns for that matter. As the frames complete a total spin of 180º, the beams are generated in almost all possible spatial positions, from horizontal to vertical. The solution we implemented required, once more, changes in Rosetta. A transformation within the beam operation was carried out, converting it into a smarter function, now capable of evaluating if the two given points align to a vertical position. In such case, it produces a column instead of a beam.

However, problems were encountered at every intersection point, as we could not get the beams to join correctly (Figure 8.6-A). This happens because ArchiCAD considers the beam axis in the centre of the upper face, which makes sense in constructive terms and was quite useful to us when placing the slab beams. Nevertheless, when modeling a structure that positions the beams according to a matrix of points, the joints can only be correct if the beam axis considered is in the center of the beam, just as it happens with columns. Unfortunately, columns could not be used in the presented case, as their top and bottom ends are sectioned by their defined top and bottom level respectively, jeopardizing their intersection with the remaining structure (Figure 8.6-B). The solution we encountered was the use of profiles for the beams. When defining our own profiles, we may choose where the axis is located, in this case the beam's center.

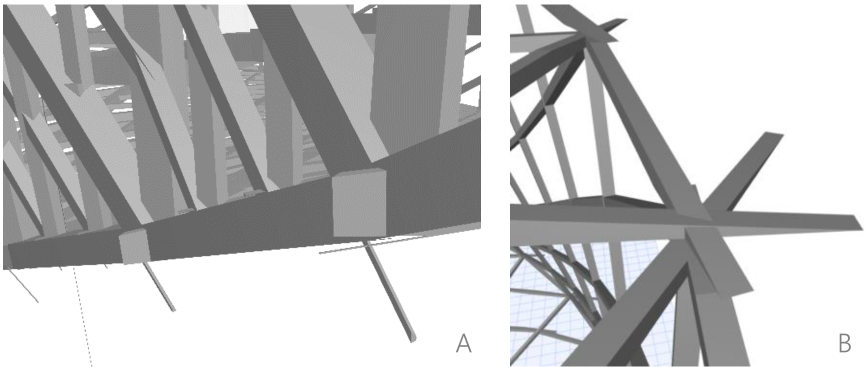


Figure 8.6 - Failed intersections of the beams in the façade grid

Another improvement to the joints accuracy was accomplished by rotating the bars. Both CAD and BIM programs generating a beam from Rosetta's beam function will place it with the down face parallel to the ground plane. When the beam is generated between two points at different heights, the geometry is rotated around one of the points to meet the second, yet unless another angle is specified, it will remain partially parallel to the ground plane. In our case study this fact resulted in distorted intersections of the structure, as the beams' angle did not follow the façade's surface normals. We then introduced a new parameter to the BIM function for ArchiCAD capable of rotating a beam around its center axis, and by rotating each bay beam correctly, we obtained more accurate joints.

Another relevant fact to note is the lack of accuracy found in intersection groups in ArchiCAD (see Figure 8.7, top). This functionality is supposed to perform trims when objects from the same group intersect. In the case of our complex beam arrangement it did not function properly and was highly dependent on the order by which the elements were generated. When generating the bays before the frame bars, beams were cut in wrong places, leaving out needle points in the joints. When generating the frames before the bays we obtained a much more accurate representation of the intersections (see Figure 8.7, bottom).

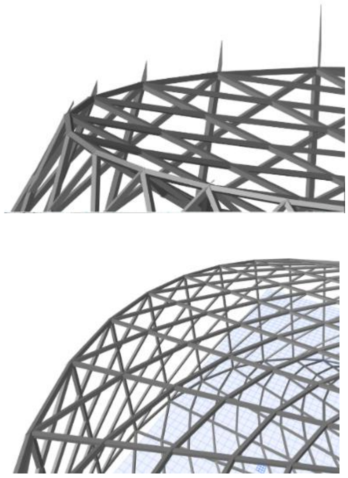


Figure 8.7 – On top: lack of accuracy in the beams intersection due to the generation order. On the bottom: the correct solution that resulted from a different generation sequence

8.7. PANELS

The transition for BIM regarding the glass panels was done using Rosetta's panel operation. The panel function, when executed in CAD backends, generates a polygon with any given number of coplanar points extruded to a defined glass-panel thickness. When running in the ArchiCAD backend the function generates the same geometry using a morph semantic. This temporary solved our problem, however it is not a perfect solution. Using construction BIM objects would be preferred to using morphs.

The same transition was applied to the photovoltaic panels, only a different material was attributed to the morphs. A render of Astana's façade panels generated in ArchiCAD can be seen in Figure 8.8.

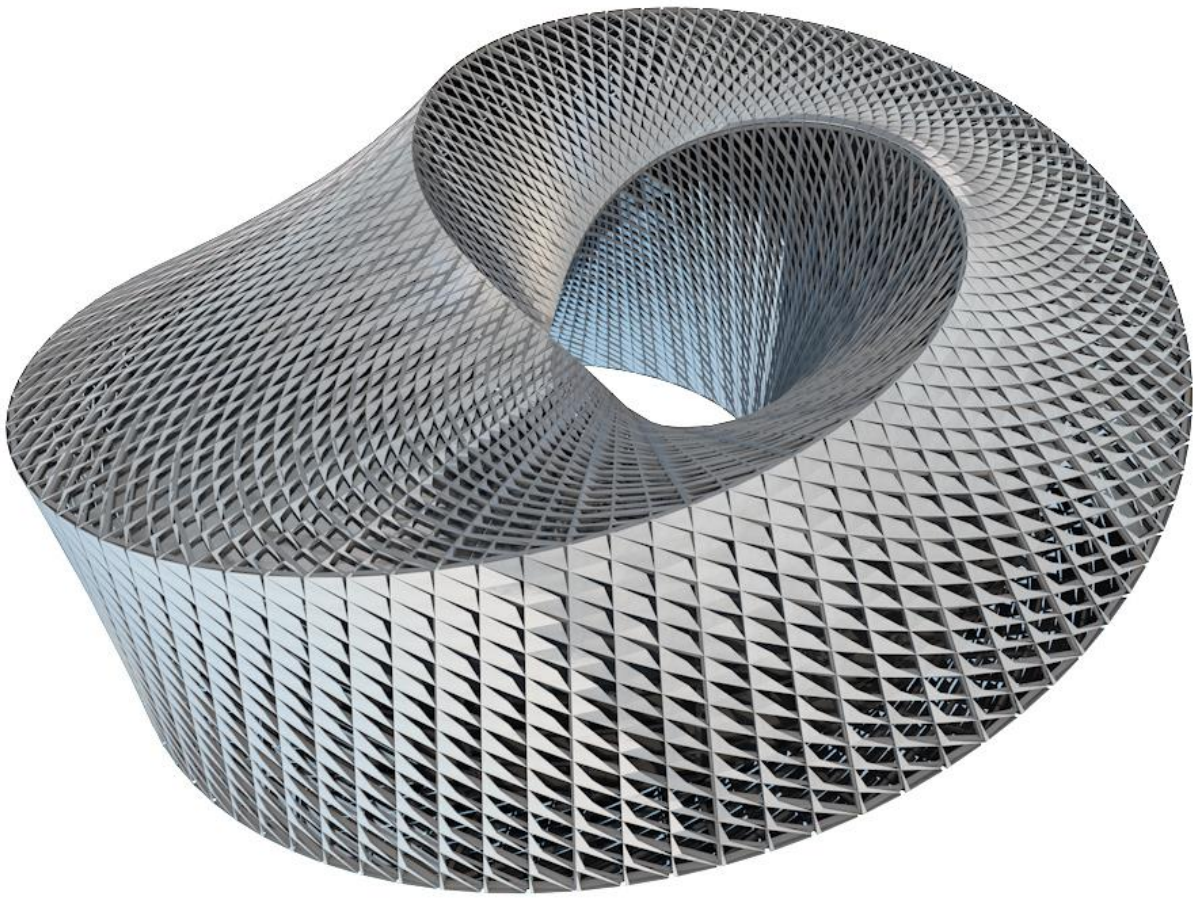


Figure 8.8 - Render of Astana's façade panels

9 GOING INTO DETAIL

Halfway through our model we began entering the realm of detail, meaning we found it more beneficial to start modeling almost exclusively in BIM, as these applications provide the user with more information by default. Designing elements such as glass walls or curtain walls, stairs, railings, doors, etc, is considerably faster and easier within the BIM paradigm as these elements are already embedded in the program with all its details, whereas in CAD we would have to model each single object from scratch.

The elements described in this section sealed a turning point in our approach: the final transition from CAD exploratory modeling to BIM detail design. The integration of specific objects from the application libraries in our code implied a loss of portability. Our CAD backends are unable to recognize the specific objects from the BIM libraries. In fact, some of the objects are not even common among BIM applications, which forced us into choosing only one backend to program for. Due to our own experience in this software, we decided to choose ArchiCAD. Some additional features had to be added to the backend in order to model this case study. Its development did, in the end, help create and test new features for Rosetta's backends.

Some of the objects mentioned above were still programmed for CAD for the sake of comparison and some more exploratory positioning, but only a coarse approximation was done, with no further detail, than a mere representation. Modeling objects like doors, elevators, curtain walls, etc, with all the detail BIM provides automatically, would be a largely time-consuming task with little benefit for all the work that would need to be fulfilled. In opposition to these rough simplifications for CAD stand the selected BIM objects with much more accessories and components changeable through the parameters.

9.1. INTERIOR STAIRCASES

Figure 9.1 - Staircases in between slab sets of the spiral volume, highlighted in blue

Inside the spiraling blocks, another type of staircases lie, overcoming the height difference between the slabs (see Figure 9.1). The interior floors are organized in sets of three joint slab pieces, fulfilling 30º of the loop at each height. At each transition, another spiral staircase must be placed. The spiraling angle of these sets are considerably smaller than the one correspondent to the spiral steps mentioned before. The staircases follow the standard proportion of Blondel's formula[2], and are placed alongside the middle volume with a constant radius.

The rude approximation of the form we modelled for CAD consisted of extrusions of surface-polygons (Figure 9.2-A). These polygons are rectangular and each one corresponded to a step. The overall look of the staircase feels wrong though, as the base of the stairs ends up being a mirror of the steps instead of a smooth planar surface. In BIM applications, the

[2] 2 x step run + step rise = +/- 64 cm (comfortable human stride)

staircases were generated using the stairs command, requesting the spiral-staircase object (Figure 9.2-B). This object was then molded to our desire through its regular and additional parameters. The manipulation of the offered parameters allowed us to easily obtain a fully finished staircase, with guard-rails and step-coating, not having to model any of this detail (Figure 9.2-C). If we were to accomplish the same result in a CAD environment, we would have had to model each detail from scratch.

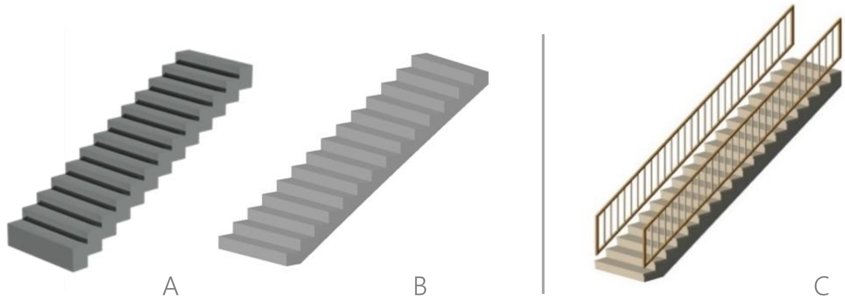


Figure 9.2 - A: Original staircases modeled in CAD; B: Original staircases in BIM; C: A different type of staircases obtained in BIM by changing the object's family

9.2. GLASS WALL

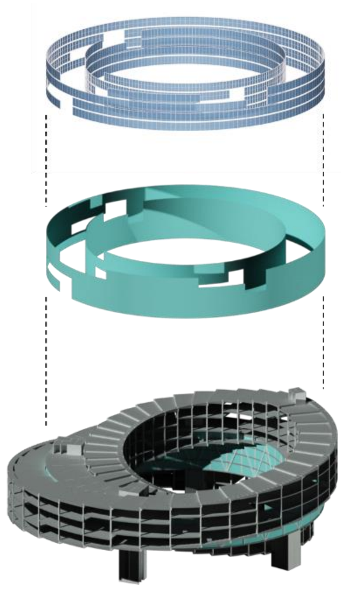


Figure 9.3 - Glass wall for CAD (on the bottom) and for BIM (on top)

The library's archive, located inside the ring-shaped volume, is offered some privacy from the remaining building by sets of frost glass panels placed amidst the columns. The panels are placed all along the four floors, except in modules where the spiral volume slabs level up with the middle floors and a passage from one to another can be achieved. A scheme of the placement of the glass wall in the building can be seen in Figure 9.3, as well as the CAD and BIM differences that will be explained in the next paragraphs.

Our inexact approximation in CAD consisted of extrusions of line sequences virtually placed at the bottom of each floor, until the top height of that same floor (Figure 9.4, top). The process is repeated for all four stories, while the intersection of the volumes is calculated. Where the connection is verified, no extrusion is performed and a void remains in its place.

The BIM object used to produce the referred panels was the curtain wall (Figure 9.4, bottom). This object allows for the creation of not only panels of various materials, but also their support structure. In our case, metal frames were generated between every glass panel and at the borders of each set. In our BIM backend, and with a simpler code than the one produced for CAD, we were able to achieve a far greater level of detail, as the object used already contained the information we needed.

Figure 9.4 - Detail of the glass wall modeled for CAD (on top) and of the curtain wall in BIM (on the bottom)

9.3. RAILLINGS

Focusing on the spatial organization of the building, one may notice that most of the spaces inside the spiraling volume are supported only by pillars. Some recline against the middle volume also, but in most cases, no walls confine the slab pieces. This gives the visitor direct contact with the wrapping façade along the looping path. Yet another sort of confinement is then required, to guarantee safety measures in a composition like this one.

Glass railings were the protections defined by the architects and we can find them placed around all slab pieces (see scheme on Figure 9.5). Two types of railings can be distinguished, depending on the side of the slab pieces they belong to: curvilinear railings avoid falls to the existing space between the path and the façade wrap, and rectilinear railings grant protection between the level differences of the public pathway around the loops. For the first, calculations were made to subtract the places where staircases were placed, and for the second, intersections between the two intertwining volumes were taken in consideration. For the same reason, no railings are necessary when the blocks contact with the middle ring.

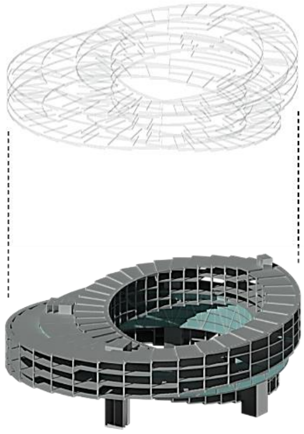


Figure 9.5 - Railings' positioning on the building

We began modeling the railings in CAD via extrusions of lines but rapidly shifted to the BIM panel operation also available for CADs. The panel operation worked for both applications, but was not ideal for BIM since it generated planar morphs. For an all-BIM approach, a specific railing object would be preferred. Unfortunately, rail objects' parameters are quite limited and the ones we attempted to manipulate just could not fit Astana's requirements. In addition to this, no object available in ArchiCAD's library presented the exact specification of the architect's design. This sealed our adoption of the morph as the BIM element for the railings in this project.

Nevertheless, the differences between our CAD and BIM models were starting to be apparent regarding their levels of detail. Figure 9.6 shows a comparison between the two.

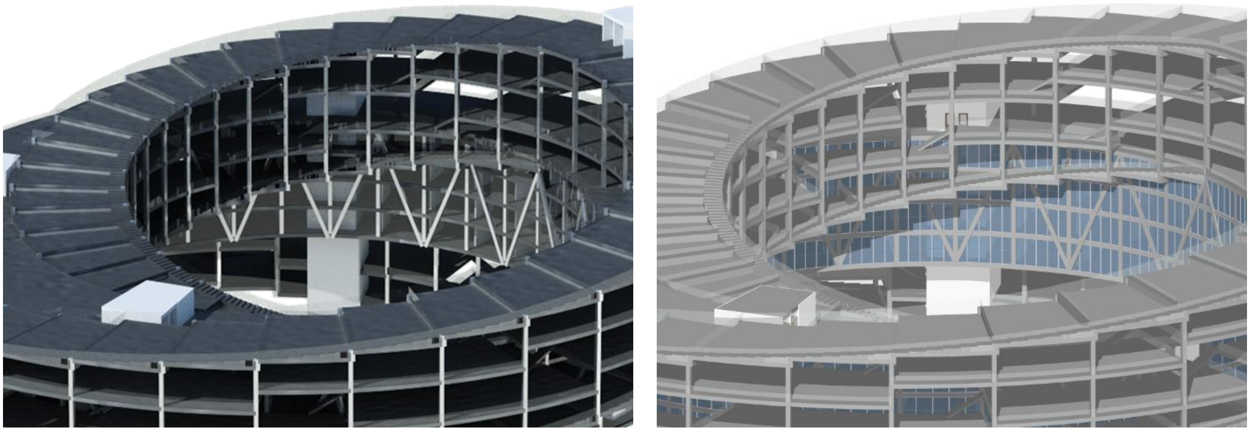


Figure 9.6 - Render of the CAD (on the left) and BIM model (on the right)

9.4. DOORS

In the core volumes, we find access doors to the restrooms and emergency staircases. We did not attempt to model them in CAD. In BIM, however, we could take advantage of the existing libraries with pre-modelled door models. Furthermore, while in CAD we would have to perform Boolean subtractions in order to create the hole for the door, BIM applications do this automatically. Placing a door in a wall implies the automatic creation of a corresponding hole in the wall.

BIM doors require a host wall, they cannot be randomly placed in the model. Hence, the function available in Rosetta that generates doors also demands that the user provides the wall for the door to be placed. Furthermore, the location where the door is placed must be defined in a specific reference system, located at the beginning of the host wall. This is meant to prevent the creation of doors outside the wall's limits, as it makes very little sense in constructive terms. In a BIM paradigm, one cannot supply the door function with a spatial location contained outside the host wall.

As mentioned before, in the BIM paradigm we find several pre-modeled door types so there is no need to model them. Hence, if the user wishes to change the door type, he needs only to specify the type, in contrast to the CAD paradigm where he would have to model a new design from scratch. Figure 9.7 presents two examples of door types changed with little effort in the BIM model.

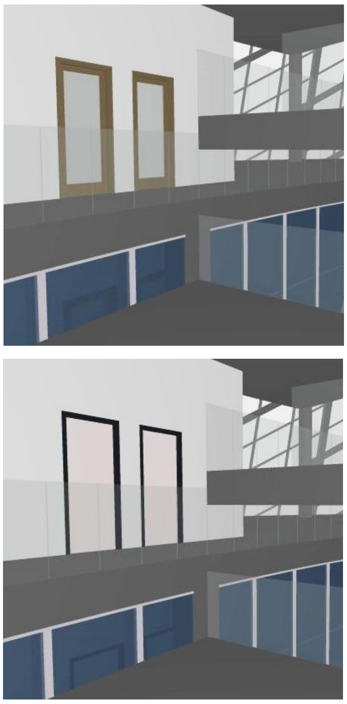


Figure 9.7 - Astana's cores with two possible door types

9.5. CORE ELEVATORS

Taking advantage of the pre-modelled objects available in ArchiCAD's library, we managed to insert other detailed elements in our program with very little effort, namely, core elevators. In Figure 9.8 we can see the doors and elevators of the core volumes of the middle ring generated in the backend.

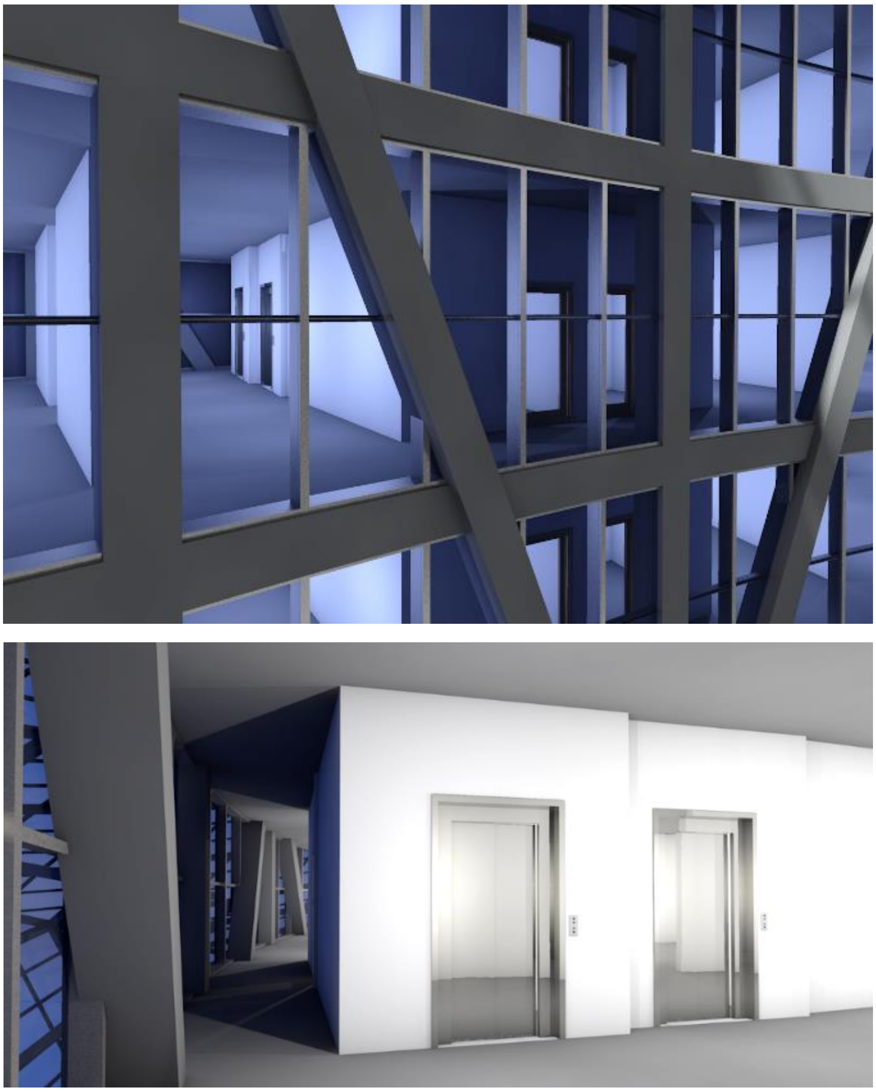


Figure 9.8 - Renders of the interior space, namely the ring volume with the doors and elevators of the core volumes

10 INCORPORATING ANALYSIS

In a time where architects excessively rely on pre-calculated technical solutions and mechanical systems to improve energy performance and control the interior climate of their creations, analysis tools bring back the stimuli they need to experiment with a more environmentally informed architecture. Considering an incorporation of the analysis data in early phases of the buildings' design grants a better relationship between the man-made and the natural. Nevertheless, working with analysis tools and dealing with its requirements can also pose quite some challenges.

Preparing a model for analysis is usually a task that consumes quite some time. The limited geometric detail most analysis tools allow, in order for the analysis to be computationally feasible, means elements like mullions, window frames, wall thicknesses, and others, must often be abstracted or eliminated. Adding to this, some additional tasks like the setting of a ground plane and the attribution of materials might also be necessary if these are not already part of the model.

When the user wishes to test a variety of solutions the scenario worsens. Not only does he need to model the variations of the building to test, but this preparation work might also need to be repeated for each variation. Given that the time and effort spent in the preparation for analysis is not negligible, its repetition makes the designer less open to do more than just a few analyses. When designing with an algorithmic approach, the first phase is not an issue. Parametric relations allow for the automatic generation of variations through the manipulation of the parameters. The second part, however, was yet to be automated.

The IAD methodology offers a set of algorithmic operations for preparing a model for analysis so that the preparation can be done concurrently with the generation of the 3D model, freeing the user from the effort needed to prepare each variation for analysis. The set of algorithms also includes the integration of the analysis results into the program. The following subsection thoroughly explain the workflow of the incorporation of analysis into the algorithmic design process, as well as the practical application we did with Astana.

10.1. PERFORMATIVE FAÇADE

Photovoltaic tiles on Astana's façade wrap absorb energy from the sun, while also providing passive shading. With the solution envisioned by the architects, the air would naturally ventilate between the inner structure's interior space and the atrium (Modern Green Structures and Architecture, n.d.). The façade acts as a thermal buffer for the mechanically ventilated interior, and a high degree of latent heat recovery is incorporated into the services design to minimize humidification loads (Ramboll, 2010). At the same time, the interior space benefits from natural light coming through the openings in the exterior shell.

To produce this result, the architects operated with "advanced computer modeling to calculate the thermal exposure on the building's envelope", states Bridgette Meinhold in Inhabitat (Meinhold, 2009). Due to the wrapping and twisting of the façade geometry, the thermal imprint has a wide range of intensities along the Möbius strip. As we can see in Figure 10.1, a thermal map was constructed revealing which zones receive more light (the range from yellow to blue shows witch areas need more to less shading respectively). This climatic information was converted into data that, in turn, regulated the façade pattern. In the same image, we can see the variating openness created, forming an "ecological ornament that regulates the solar impact according to the thermal requirements" (Archello, 2010). The façade shading design is thusly a product of artistic expression ruled by ambient conditions.

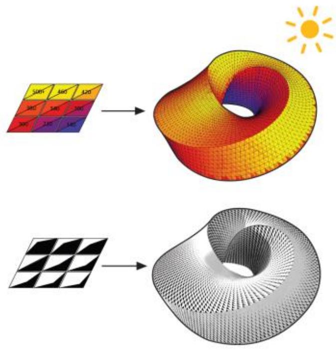


Figure 10.1 - ANL Ecotect analysis (source: http://www.archello.com/en/project/anl-astana-national-library-0)

The photovoltaic tiles have triangular shapes with different dimensions in consonance with the facade's sunlight exposure. The panels are arranged according to the grid that supports them, the finer network defined by the secondary structure of the beams composing the façade. Each section of the grid holds two lattices, mirroring one another. These lattices can have one of nine possible dimensions, creating in the overall view nine distinct openings distributed along the façade.

10.2. ALGORITHMIC ANALYSIS WORKFLOW

Our chosen programming tool to apply the methodology, Rosetta, processes a set of analysis backends. For the analysis of our case study we used the Radiance backend, since Radiance is as analysis engine suited for lighting simulation (Radiance, 2017).

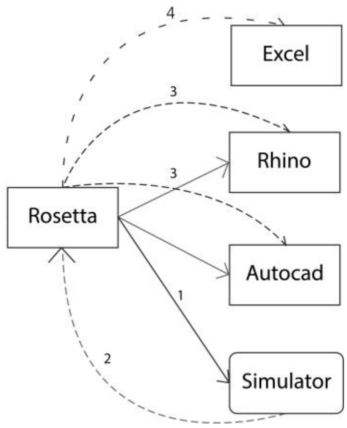


Figure 10.2 - Rosetta's Radiance backend workflow

Most of the steps needed to prepare a model for analysis are automated in this backend. The geometry of the model is generated in the way required by the analysis tool. The practical result is that, despite the user's program being the same, the model given to the analysis backend is independent and different from the model produced in the visualization backend. The model generated by Rosetta contains all required elements, and/or simplifications, that the user would otherwise have to do manually. The level of detail, or features, of the algorithmically generated building are produced according to the analysis needs.

Figure 10.2 illustrates this process: the user programs his design in Rosetta, with no concerns regarding the analysis tools requirements. (1) Rosetta then sends only the necessary information to Radiance for the analysis. After the analysis is concluded, (2) the results are retrieved and (3) displayed in the 3D modeling backend, e.g., Rhino or AutoCAD, or (4) they are exported for further processing, for example, in Excel.

10.3. ANALYSIS CONDITIONS

A Point-in-time (PIT) analysis is ran for a single moment in time. A time-step analysis, on the other hand, is ran over a time period, such as a year, for instance (Anderson, 2014). While the former is fairly quick to run, and provides the user with more accurate results, the latter is capable of analyzing trends through time. In order to evaluate a building behavior throughout the whole year with PIT's, many analysis need to be ran. However, in a time-step process, while the software performs a sequence of analysis over time, it must also connect them based on a series of assumptions to form an overall assessment, giving back less accurate results.

For Astana's façade radiation analysis, we set our analysis in a PIT system to the shortest day of the year, the Winter solstice. Since we are calculating the amount of photovoltaic panel area that will cover the façade, we believe the best choice for a PIT analysis is the worst-case scenario for light reception.

Boundary conditions are the edges of a simulation. According to Anderson (2014), boundary conditions are usually assigned when studying a portion of a building. The goal for their use is to limit the geometric scope and run time of the simulation, allowing the user to obtain faster results. In Astana's case, the façade was the boundary for the analysis of the radiation each panel would receive. The interior space did not need to be considered for the analysis. Limiting the boundary to the façade wrap considerably reduced the time required to run the analysis.

10.4. ANALYSIS SETUP

In the case of Radiance, the analysis tool we used, all elements in the model need to be mere surfaces, poly-surfaces, or meshes. Fortunately, the algorithmic approach automates this process by generating the elements in the model according to the requirements of the analysis tool. The user may program slabs, beams, glass, etc., as he normally would and the Radiance backend turns them to surfaces, poly-surfaces, and meshes. The Radiance backend does not generate solid geometry and this is why it differs from the CAD/BIM backends.

A ground plane also needs to be set, and materials attributed. Due to its CAD-BIM portably, all models produced in Rosetta benefit from the BIM approach to material information. Even if the geometry is meant to be generated in a CAD tool, like Rhino, the user can take advantage of default BIM families. As such, Rosetta contains all material data of the model needed for the analysis, despite generating only simple geometric elements in the chosen CAD application. For the analyses, the list of materials is extracted automatically from the generated model, using the actual materials that the designer selected for each element or that were assigned by default.

In order to analyze the radiation received by Astana's façade, we used the Radiation Map metric. This analysis required some additional setup, namely the positioning of sensor-nodes. A grid of points had to be created all over the façade surface. Each point represents a sensor node where the light levels are calculated. Astana's façade, however, is a particularly complicated one as the surface curves over itself. Each node positioned over this surface is controlled by a differently oriented normal vector. Once more, in an attempt to free the user from this complicated setup requirements, in Rosetta's Radiance backend, each non-planar surface is automatically meshed according to the node separation. Moreover, for each surface, the corresponding vector field is computed, and the correct location and orientation of the sensors is provided to the analysis software. In Figure 10.3 it is possible to see the nodes placed with the same offset from the given surface.

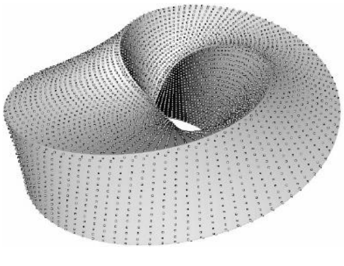


Figure 10.3 - Astana's sensor nodes places over the façade surface

10.5. INTERPRETATION OF RESULTS

For a visual interpretation of the analysis outcome, the Radiance backend for Rosetta allows the user to see the results in Rosetta's visualization backends. For this representation, the numerical results correspondent to radiation values obtained in each sensor node are translated into a color-scale. The meshed surface becomes a set of color-coded rectangles that intuitively conveys the analysis results. In Figure 10.4 we can see the analysis results in both Rhino and AutoCAD backends. We can intuitively perceive that the red color represents the most heated areas and the blue, the cooler ones.

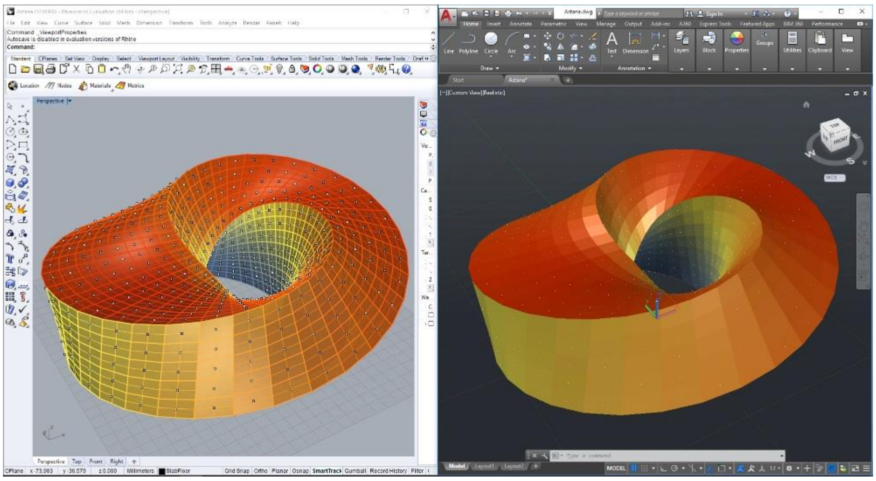


Figure 10.4 - Analysis results generated in AutoCAD and Rhino

Some experts call this color-schemes, "false colors" (Anderson, 2014) as they are not the true color of the building, but instead a limited color range provided by the software to help users interpret the results. They are used to graphically convey levels of solar energy, or other results, usually converting the span of numerical values into a color palette.

However, as useful as these representations are to help architects interpret the numbers, the color ranges cannot be used for further processing. In order to generate the panels as intended by the architects we must use the radiation values that were found.

10.6. INCORPORATION OF THE RESULTS INTO THE DESIGN

In order to incorporate the performative intent into our generation of the façade panels, we connected Astana's program to one of Rosetta's analysis backends, Radiance. The backend subdivides the façade surface into triangular and quadrangular areas, corresponding to the zones affected by each panel, and assigns a sensor node to the center of each zone. Using a Radiation Map metric, Radiance performs a light simulation for all the given nodes, and gives back the radiation values for each one. The values are stored in a dat file that is then used as input to the Astana Program.

The radiation values for each panel, exported by Radiance come in a scale of zero to one hundred. We import them to the program and feed them to the function that creates the panels, in the correct generation order so that the value for each zone is correctly attributed to the corresponding panels.

In a first phase, we computed the panel dimension using a linear map between the radiation values and the panel sizes: the minimum and maximum panel sizes correspond to the minimum and maximum radiation values, and an intermediate radiation value will cause the creation of a panel with an intermediate size. In other words, each different value defines a panel of a unique size, as the values are not discretized. Figure 10.5 presents an scheme of this conversion.

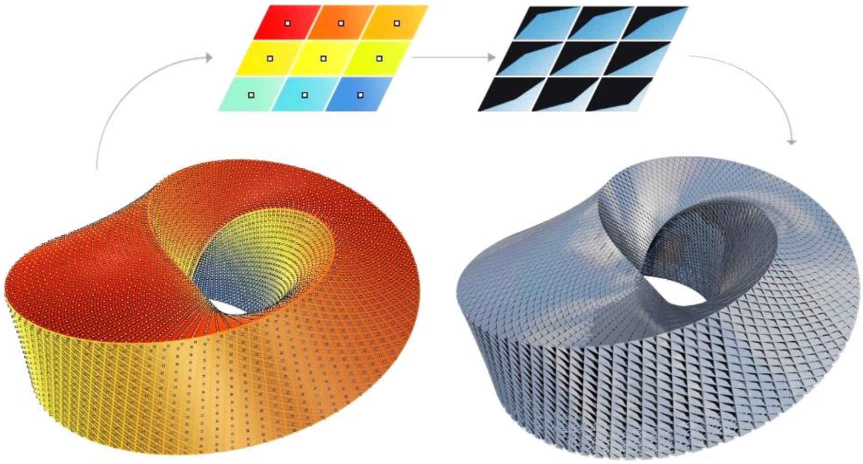


Figure 10.5 - Scheme of radiation values converted to panel sizes in Rosetta

However, given the almost continuous variation among radiation levels, this would also cause a continuous variation among panel sizes, which in our case, would imply the creation of 8856 different panels, which would be prohibitively expensive. To reduce manufacturing cost, the panel sizes would have to be discretized. Following the architect's original intent, we subdivided the possible panel categories into only nine. Table 10.1 shows the resulting subdivision of the panels.

Table 10.1 - Number of panel of each size. The radiation values are divided in nine classes

CLASSES OF RADIATION VALUES	PANEL TYPES BY SIZE	Nº PANELS OF EACH TYPE
0 – 11,11	A	2376
11,11 - 22,22	B	949
22,22 - 33,33	C	707
33,33 - 44,44	D	614
44,44 - 55,56	E	584
55,56 - 66,67	F	589
66,67 - 77,78	G	648
77,78 - 88,89	H	786
88,89 - 100	I	1603
TOTAL Nº PANELS		**8856**

The first column presents the radiation values divided in nine categories, while the second names the nine types of panels, and the third column has the number of panels within each category, with A being the smallest type of panels, corresponding to the lowest radiation values and I being the bigger ones, corresponding to the highest levels of radiation.

Figure 10.6 presents the final result of the first and second approach respectively. On the left we see that the façade is made of panel of different sizes corresponding to the radiation values. On the right, we verify that the transition between panel sizes is more abrupt as there are only nine different shapes.

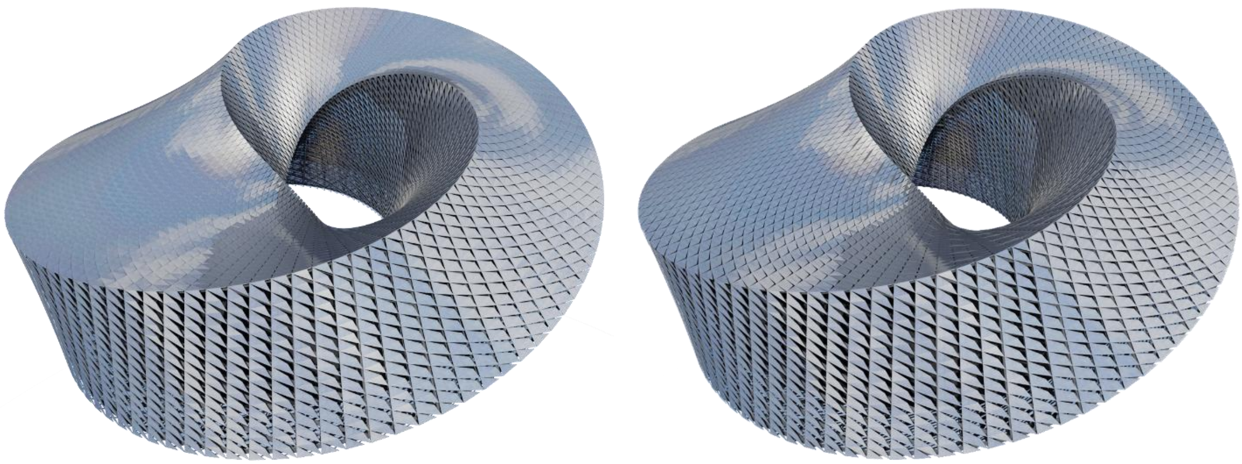


Figure 10.6 - Render of Astana's façade panels with non-discretized number of sizes (on the left), and with only nine different sizes (on the right)

11 SELLING THE PRODUCT

Architects traditionally design with drawings and physical models, as means of representation. The notion of drawings covers a vast range, from rough sketches to fully detailed plans, sections, perspectives or other views. Physical models follow a similar logic: an architect may begin with coarse sculptures of concept and end with a detailed model at any given scale. Any of these methods serves not only as a mean to place their ideas in paper, so to speak, but also to communicate them to others. Architects understand each other like so, and the same happens with engineers.

These standard representation methods, however, require a degree of interpretation that may require training and are, hence, not immediately understood by every element involved in the project. Most professionals working in the industry will know how to read technical drawings, but most layman outside the AEC business, and this can include the clients and maybe contractors also, will not have the immediate tridimensional understanding of the project by reading a plan, as does a trained architect, for example. Even if they do, they may feel unsure enough to be uncomfortable assuming it as the sole representation method. Furthermore, traditional orthographic representation methods are not unequivocal, and often prove insufficient to convey certain types of data. Some projects require more than basic technical planning. For instance, working the ambience, mood, branding, materials and textures, lighting, etc. is impossible through plans and sections exclusively. This is often a grey area between architecture and decoration that is frequently requested in business and public space projects.

11.1. THREE-DIMENSIONAL MODEL

Tridimensional physical models are a good alternative to the methods presented above. With one additional dimension, our understanding of the unbuilt environment naturally increases. Gaudi used his hanging chain model, previously mentioned in section 3.1 as a structural analysis model, for this purpose too. He would take pictures of the model, inside and outside, invert them and present them to his clients, in order to show them how the cathedral would look like.

However, physical models are costly and time consuming. Architects cannot produce a new physical model for every change the client or the engineer proposes. Currently, a second alternative has been gaining popularity: tridimensional digital models. These are, not only less expensive as there is no physical material involved, but they are also easier to change. Just as it is now considerably faster to change digital plans and sections, the same occurs with digital 3D models. Furthermore, they allow a level of detail that physical models are very unlikely to reach, due to the usual small scales they represent.

11.2. RENDER IMAGE

In a digital model, scale is not an issue, it is a question of zoom. Digitally generated images allow the viewers to truly experience the ambiances inside the project's spaces. The common denomination for these images is Render. A rendered image allows, not only a better perception of how the space will be when built, but also gives the viewer an idea of the ambiance it will have. The ambience is created by many factors: the lighting, both natural and artificial, indoors or outdoors, coloring and temperature of said light, materials, textures, reflectivity and roughness of such surfaces; the furnishing and decoration of the space; possible users of the space taking on activities that denote the space's proposed use; etc. All this detail, present in an image that intends to be almost a photograph of an inexistent reality, can show the viewer the unbuilt project with a realism that was not achievable with the previously available means.

Clients are now ever more eager for this representation methods, and many architects are specializing in this task alone. A couple of years ago only projects of considerable dimensions would request such work, or clients with a reasonable budget, at least. Currently, however, the ability to fully visualize the possible ambience of the unbuilt space has gained such an importance, that even small projects do not spare the work. In fact many changes are triggered by the clients' evaluation of the simulated ambient.

11.3. PRODUCTION RENDER

Renders images are not a fast endeavor, however, particularly if a photographic realism is intended. The more detail the architect wishes to input in the model, the more time he will need, both to model and render it. Traditionally, the production of presentation oriented 3D models used to be employed only in the later stages of development when a higher degree of certainty about the project was already achieved between the involved parties. This was mostly due to the cost and the inherent difficulties to the development of such 3D models.

In recent times however, with the trivialization of these processes and the evolution of the tools and software involved, they are becoming more of a conception and work tool, rather than exclusively a presentation medium, used in the final stages for illustrative or promotional ends only. This poses a big change in the way render tools are used. While for presentation purposes there is a clear goal to achieve, using them as work tools requires an iterative approach, with many changes in a rinse-and-repeat progression.

In earlier stages of a project, however, full photo-realism may not be desired by the architects for a multitude of reasons. Extremely detailed renders may result in an unintentional commitment regarding the look of the final project. This look is, in fact, never certain in early stages, since the final products may not yet have been decided, or for technical, financial, or management reasons, they are likely to be changed still. In such cases, realistic renders produced in early stages may also create expectations that may not be met in the end, potentially looking as failures in the eyes of the client.

Conceptual, non-photorealistic, stylized rendering may be used instead, as a representation method. If the architect wishes to transmit only a notion of the space's possible use and volumetric capacity, generic objects can be placed in the model and a simplistic render can be produced in a fairly reduced amount of time. Even so, rendering (simple production renders or final detailed ones) is still not a process that the architect is willing to undergo many times in the course of the project modification phases. However, the introduction of an algorithmic approach to the process may offer the possibility to change this premise.

11.4. ALGORITHMIC RENDER

The ability to propagate changes through the model is extensible to rendering detail. Just as we algorithmically relate construction detail in the program to allow the parametric model to automatically adapt when parameters are changed, the same can apply to detail for rendering purposes. Since a render is always a speculative scenario of what the ambience could be, a mathematical distribution of furniture and people can be envisioned, that generates these elements automatically in accordance to the shape of the various spaces in the building.

In Astana's case, we decided to take advantage of ArchiCAD's pre-modeled objects and we programed their distribution inside the library, according to the buildings variable parameters. This means, that if any changes to the model are made, such as the heights of the levels or the distribution of the

Figure 11.1 - ArchiCAD standard book-shelf

Figure 11.2 - ArchiCAD standard book cluster

Figure 11.3 - ArchiCAD standard square table

walls, the placement of these elements will change accordingly. Moreover, we also programmed the production of renders so that, for each set of changes to the program, a new series of images can be automatically produced, as views, lighting, sky effects and many other image sets were mechanized in the program.

For this task, we focused on the central volume and we filled the space with objects that we felt would naturally decorate a library space. We used several of ArchiCAD's pre-modeled objects, particularly, human figures, bookshelves, tables, and chairs.

Figure 11.1 shows the bookshelf object that we used. However, for Astana we modified its parameters in order to better suit the library's environment, namely its length, height, number and size of shelves and materials. We defined the bookshelves height in accordance to the floors' heights and their lengths in accordance to the library walls. We then distributed them radially adjacent to these same walls as can be seen in Figure 11.4.

The bookshelves were filled with book clusters. The standard object can be seen in Figure 11.2. The only parameter we changed in it was the size. We programmed a function that would randomly provide variable lengths, within a defined range, and we placed these book clusters on the existing shelves with a random factor of 70%, in order to leave some empty spaces.

Next, we turned the library rooms into working spaces by allocating tables and chairs. We began using a standard dining table set from ArchiCAD's library, visible in Figure 11.3, and modified some of its parameters in the program, namely the materials and chair arrangements. Tending to the rooms shape we distributed the tables also radially, as visible in Figure 11.4.

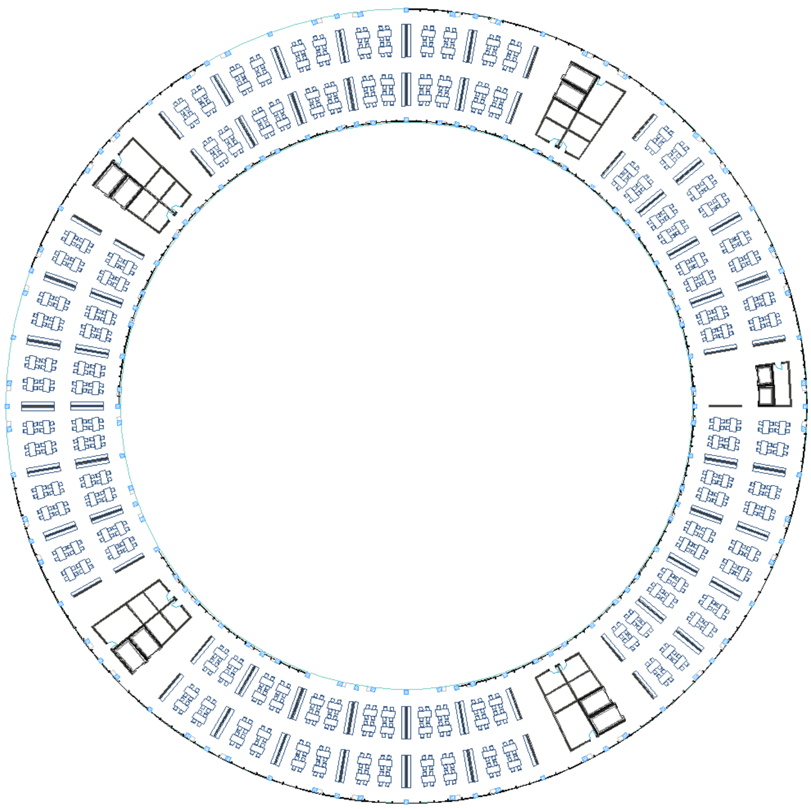


Figure 11.4 - Middle volume plan, 2nd floor

Finally, we focused on the people. ArchiCAD provides a series of pre-modeled objects representing individuals performing various tasks and each of them has numerous configurable parameters, such as clothing type and color, skin and hair color. Figure 11.5 presents some of these objects. In order to obtain a varied mix, we introduced functions capable of randomly assigning clothing types and complexions for our individuals, and we programmed their distribution along the three main corridors of the middle volume. As not to crowd the space we defined a 30% occupancy rate for the possible positions we had defined.

Figure 11.5 - Selection of people objects from ArchiCAD's library

Lastly, we came across the need to implant lights inside the volume as well. For this we used the light function for ArchiCAD. We placed the light spots radially along the ceilings in all floors. Once more, we stress that since the model is parametric, if the number of floors, or their dimensions are altered, all the mentioned elements will adjust accordingly. Figure 11.6 presents a view of the furnished space, from the spiral volume.

As the whole process is automated, changing the color or any other parameters of the programed objects is a rather simple task. The changes are propagated to the entire model and the program can automatically produce a new set of renders. Our case study went through this process a couple of times as we attempted not only some material variations to the furniture, but also some modifications in its placement and some replacements to the originally chosen set of objects.

Figure 11.6 - Render of the library's interior space furnished. View of the spiral volume

We invested some effort in a broader research for more esthetically pleasing objects than the standard ones we had originally elected, since these proved to be inapt to deliver a proper ambience for a library. Particular focus was given to the set of tables and chair. Figure 11.7 and 11.8 present some of the objects we experimented with and the scenarios we obtained.

Nevertheless, it is important to stress that the goal of an algorithmic automation of rendering production using BIM's pre-modeled generic objects is, primarily, to produce conceptual images that can provide the viewer with a notion of volume and scale. Hence, the time and effort worth spending in improving the generic scenario, or in producing different variations to that scenario must be determined by the architect. Typically, higher investments in space planning studies are more appropriate in latter stages of the project's development.

Figure 11.7 - Render of the library's interior space furnished. View from the middle volume

Figure 11.8 - Render of the library's interior space furnished. View from the middle volume

12 EVALUATION

There are many ways to represent an architectural design: two- or three-dimensional, manually or digitally, or yet algorithmically. Building a tridimensional digital model in a CAD or BIM tool surely differs from the most traditional method of all, drawing by hand. However, either imply a direct manipulation of the modeled design. Designing algorithmically, on the other hand, is not only a mode of representation, but also a mode of generation (Shusta, 2006). The main difference lies in the way form is constructed. Using traditional representation methods, the architect channels an idea into a form, whereas with programming he channels a process into that same form.

An algorithmic approach to design implies that the user, instead of creating the design solution by direct manipulation, creates a system of established relationships that represent the design. AD allows the user to go beyond the developed commercial applications he might want to use, and mold them to his own way of thinking, as the mathematics of his description are independent from those applications. However, designing with algorithms usually requires an additional effort from the user in initial stages. The return of this investment, nevertheless, compensates the effort.

In this chapter, we present an overview of the conclusion we could withdraw from our research, particularly the advantages an algorithmic approach can bring to the design process, in comparison with the more traditional means of modeling. Some of the reflections are complemented with additional modeling tests we performed.

12.1. AD OVER MANUAL APROACH

While direct manipulation of the design allows the user to immediately see results as soon as he starts interacting with the system, an algorithmic approach is more intellectually taxing. In order to correctly implement the relations between the model elements, that will allow an appropriate propagation of changes further on, the user must first reflect on the logic of his design. He must deconstruct the problem and find the logic that binds his design together, the important parameters that define it and that may change in time, and translate these thoughts into an algorithmic language that the computer can understand.

This process is, initially, more time consuming and, generally, more intellectually challenging than traditional approaches. Nevertheless, the gains found in later stages are considerable. The ability to handle change in any stage of the design process greatly supplants that of manual approaches. Furthermore, some designs are so complex that navigating through the model becomes a problem. This problem can occur in simpler designs as well, when greater levels of detail are achieved. It is common to find big projects being split so the parts can be handled separately. Big models are not only hard to navigate through, but the amount of information they contain also slows down the software, delaying each modeling task.

Big and complex models, however, are not the only kinds of models that might benefit from an AD approach. Repetitive geometries and mathematical shapes have even more to gain with algorithmic descriptions. Repeating a task manually, any task, is not a problem if done once or twice. However, if an operation is to be repeated over and over, the time lost in the process might be substantial. Considering most designs are full of repetitions, such as placing columns, beams, doors, rotating these elements, or diminishing all their lengths or heights over a change in the floors height or length, mechanizing these tasks would greatly reduce the time-waste.

Figure 12.1 - Santiago Calatrava's Ysios Bodegas (source: http://buildipedia.com/aec-pros/featured-architecture/santiago-calatravas-ysios-bodegas)

The concept of a mathematical shape is somewhat redundant. Virtually any shape has a correspondent mathematical description. Hence, they could all be considered mathematical. However, not all designs are the result of explorations made through the manipulation of mathematical expressions. The term "mathematical shape" used above refers to the cases where the design is a clear result of a mathematical phenomenon. Any building with a simple geometrical shape is easy to describe: a cube, a sphere, an oval shape or a parabolic one, etc., just as any variation made to these shapes, such rotations, torsions, reflections, or other. Less obvious geometries can, nevertheless, be a result of mathematical manipulations as well.

Figure 12.2 - 30 St Mary Axe, London - Swiss Re (source: http://www.fosterandpartners.com/projects/30-st-mary-axe/)

Astana National Library, the case study we chose, is a clear example. Its shape follows that of a Möbius strip, which has a known mathematical expression. Given different parameters, numerous geometric results can come out of that expression. Many other examples can be found. For instance, a sinusoidal defines the roof of Santiago Calatrava's Bodegas Ysios (Figure 12.1), while Swiss Re (Figure 12.2) from Foster and Partners is but a circle rotating in plan as the tower rises in height, while its perimeter simultaneously widens as it rises and tapers towards the top. Even Guidi's seemingly naturalistic shapes at Sagrada Familia are all defined by ruled surfaces (Burry, 2011).

Contemporary architecture, however, honors us with many "non-mathematical" designs. Part of the beauty of design comes from the architect's ability to freely draw the building's shape, independent from numerical restrictions. As mentioned above, any shape can be mathematically described. However, if that same shape was not produced through mathematical manipulations, inverting the process to find its expression is no simple task. Hence, odd shapes, straight out the architects mind, are sometimes hard to describe algorithmically. In such cases, the time and effort it would take to find their mathematical correspondence would hardly compensate, despite all the advantages AD brings to the table. But not wanting to miss out on them, some studios have found ways to complement both approaches.

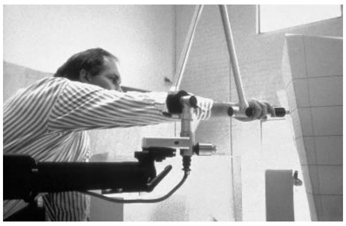


Figure 12.3 - Rick Smith digitizing Frank Gehry's model of the Walt Disney Concert Hall with FaroArm, 1991 (source: http://www.cca.qc.ca/en/issues/4/origins-of-the-digital/39920/a-fish-is-kind-of-aerodynamic)

Frank Gehry's office, pioneer in the use of BIM models (Kolarevic, 2003), is an example of this. The architect comes to his design building tridimensional models. When satisfied with the result the model is scanned with a digitizing wand (Novitski, 1992) and imported to Digital Project (software developed by Gehry Technologies, customized from Dassault's CATIA for the aerospace industry). Figure 12.3 show this process being applied to the Walt Disney Concert Hall model. The coordinates imported to the digital model are then resolved into parametric surfaces that the designers can digitally manipulate to optimize for design and constructability, etc.

12.2. HANDLING MULTIPLE TOOLS

Another issue that manual approaches face is the fact that they are usually limited to one modelling tool only. Using a manual approach, the user is forced to choose a tool in which to model his design from beginning to end before he starts the whole process. If he decides midways the tool does not suit his needs, he may try transferring the work done so far to another one, yet some information might get lost in the process.

IAD, as an independent modelling process, offers the user the possibility to describe his design regardless of the tool it might be generated in. Furthermore, as we have extensively discussed in this thesis, the process of an architectural creation has much to gain in utilizing as many tools as possible. Since they all present different features, each different set of tools can bring different advantages to the design process.

Hence the IAD methodology tried to encompass a series of tools that our research found to be the most commonly used tools in the design process. Draining from AD's potential as a portability tool, we attempted to integrate the two main modelling paradigms of our time – CAD and BIM - and analysis tools – a set that is becoming ever more present in the process as well. We explored the advantages of each one, assessing which tasks they perform best, and at which stages each is more advantageous. Figure 12.4 shows the Astana model generated in the various backends we managed to include in our practical application of the methodology.

12.3. BETWEEN PARADIGMS

Integrating CAD allowed us to perform some exploratory modeling. Although we were working with an already existing design, for the sake of our research we attempted some modification to its shape (Figure 12.5). Using CAD operations and the CAD tools as visualization backends the whole process is considerably faster than when doing the same in BIM. Hence, conducting experimental changes proved much more advantageous in this paradigm.

Using the BIM paradigm, we were able to model some of the building's detail much faster than we could have in CAD. We used pre-modeled objects for the BIM paradigm, while in CAD we had to model the geometry of each element from scratch. It would then be safe to assume that at this stage we terminally transitioned to this paradigm. However, as we mentioned in section 9, we kept going back and forth between the two paradigms, conducting placement experiments in CAD and only placing the final objects in BIM. The IAD methodology allows for this kind of exchange and alternation, that enrichens the design process with the best qualities of both paradigms.

Within the BIM paradigm we found many other advantages besides the pre-modeled objects, however. The automatic production of construction documents was one of them. In CAD, plans and sections would either be done separately from the 3D model, or, if extracted from the 3D model, they would have to be worked on. The CAD tool is only capable of slicing the geometry, not understanding its structural meaning, hence, the

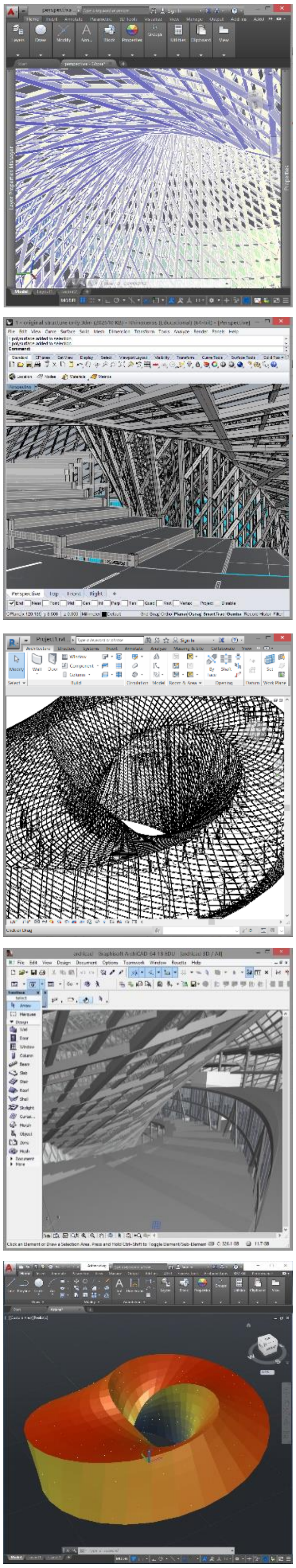


Figure 12.4 - Astana's 3D model generated in AutoCAD, Rhinoceros, Revit and ArchiCAD and analysis results from Radiance shown in Autocad

representation comes out wrong. This was the case of the plan found in page 49, a worked section of Astana's AutoCAD model. In BIM, not only are these elements extracted from the 3D model automatically, but they require no further manipulation, unless the studio needs a specific type of representation system that the program cannot conceive. The plan visible in page 68 was extracted from the Astana's BIM model in ArchiCAD. Other documents such as sections and quantity charts could have produced just as easily.

Another potential advantage of the IAD approach we explored in the BIM paradigm was the rendering production. The pre-modeled objects available in the libraries allowed us not only to detail our model in term of construction elements, but also in decoration basics for production renders. BIM tools have sufficiently good rendering engines that allow the user to produce quality images with very little set up effort. The possibility to occupy the space with standard objects such as tables, chairs, sofas, people, etc, not having to model any of it, grants the user a very fast solution for simple rendering. In non-final stages of the project, where the point of rendering is selling the environment to the client without its final details, BIM tools prove to be greatly advantageous as well.

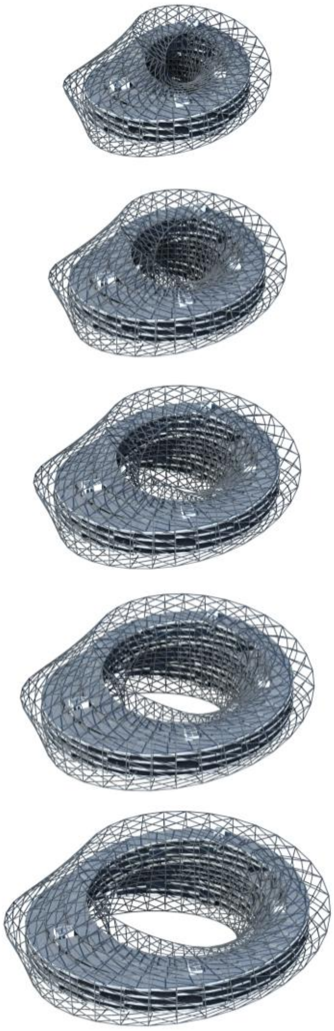


Figure 12.5 - Astana shape tests: changing parameters of the Möbius strip

12.4. ANALYSIS AUTOMATION

The integration of analysis tools seems like a logical step as well. Model analysis can be quite useful when coupled to a manual approach, but mostly as a proofing mechanism. If an architect conducts an analysis in his hand-made digital model, and the results are not satisfactory, he may apply some changes to his model in order to obtain better results. He may then test the design again, but the process is likely to be repeated no more than a few times, as both the analysis set up, and the modifications to the model are tiresome and time-consuming tasks, when done by hand.

Coupling simulation with an algorithmic approach, however, presents a much broader scenario of opportunities. AD bring to the equation the ability to analyze the data automatically, as well as the capacity to connect the tool directly to the program. This means that not only can the information resultant from the simulation be used in the design process directly, but also that the process can also be automated in order to let that information shape the design. The following paragraphs build up on our approach to this potential.

Since the Astana model was built parametrically, it is possible to change many aspects of the design, including the number and size of the panels. This means that for every different iteration of the model a new analysis must be done to acquire the correct radiation values to produce the panels.

Using an integrated algorithmic approach, in comparison to a manual approach, already speeds up the process by freeing the user from tiresome tasks he would have to perform to set up each analysis manually. The usual preparation involves introducing additional objects, adapting, simplifying or rebuilding the geometry of the building elements and restructuring the arrangement of layers. Given that the time and effort spent in the preparation for analysis are not negligible, performing repetitive analyses inhibits the designer's desire to perform more than just a few of them.

To overcome this inhibition, Rosetta's analysis backend automates all these tasks. This approach makes it easier for the architect to introduce changes to the design program, which are then automatically translated according to the analysis tools' requirements. Therefore, as change becomes effortless, the architect is more willing to try variations of his model.

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

Figure 12.6 - Variation A: structure (top), analysis model (middle), and render of the final panels (bottom)

12.5. VARIATION ANALYSIS

Although we did not take advantage of the optimization possibilities of our approach in this case study, we did take advantage of the parametric feature to analyze a couple of variants to the original model, obtained by changing only a few parameters that define the shape of the Möbius strip and the number of panels in the façade.

In the first experiment presented, a variation of the panel sizes and number was tested, as well as an enlargement of the Möbius strips' radius from the original 35 m to 60m, creating a bigger square in the center of the project (Figure 12.6). The second experiment has a double twisting of the Möbius strip, meaning that the façade rectangle that is repeated along the building rotation completes a full 360º rotation instead of the original 180º. Also for the façade not to collide with the interior space in its rotation, the rectangle was replaced by a square shape (Figure 12.7).

As both the total façade area and the number of panels and their respective sizes vary, from one variation to another, there is little sense in comparing the total radiation detected by each node. We can, however, understand from the given values some advantages of the new attempted designs. Table 12.1 presents a comparison between the average radiation values obtained in the analysis of the original design, and the ones obtained with the variations.

Table 12.1 - Comparison of radiation values between the original design, variation A and B

	ORIGINAL	VARIATION A	VARIATION B
NUMBER OF PANELS	8856	6912	9504
AVERAGE RAD VALUE	35,22	34,45	34,92
STANDARD DEVIATION	45,16	47,64	41,66

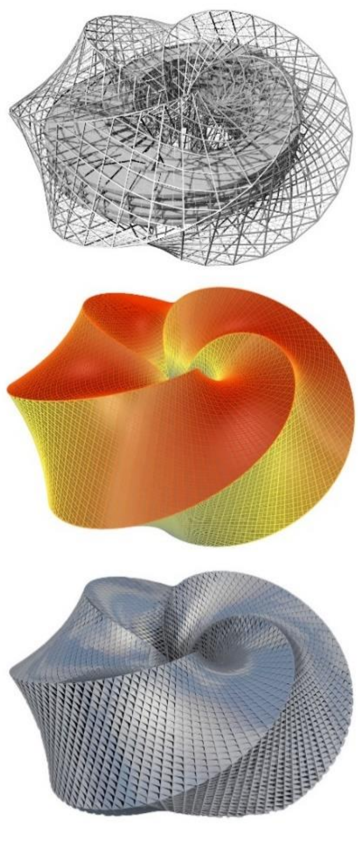


Figure 12.7 - Variation B: structure (top), analysis model (middle), and render of the final panels (bottom)

The average radiation value (on a scale of 0 to 100) experienced on the façade is 35,22. Both variations present smaller numbers, meaning that more parts of the building are in the shade, or less area is directly facing the sun path.

Either way, the results indicate that the interior space of both designs would be cooler than the original, although only by a short margin.

Finally, a second comparison can be made regarding the discrepancies felt in the more or less exposed areas. For this purpose, the standard deviation value can give us an idea of the existing differences. The original design presents a standard deviation of 45,16, where variation A has 47,64, and variation B, 41,66. This means the third design is the one where a smaller span of differences is felt. Variation A on the contrary, presents a wide range of radiation values which means some areas of the building will be significantly hotter or cooler than others. This is not necessarily a problem, only a fact to be considered when projecting the cooling systems for the interior spaces.

12.6. IMAGE PRODUCTION

The IAD approach presented yet another advantage, in regard to the production of rendering images: accelerate the process by automating most tasks. The render itself will always take time to generate. However, setting it up also consumes some time, and that part can be automated.

If we intend to make a sequence of renders, the situation is even worse, as we are relatively chained to the machine since we need to wait for one render to finish to set up the next, and so on. If the whole process is automated, we can leave the computer working overnight, or go work somewhere else during the day while the process goes on uninterrupted. The user only needs to start the program and come back hours later to collect the end product.

We applied these principles for the rendering sequence of our case study. In this case we were attempting to create an animated GIF containing the creation by phases of all the main constructive elements of the project. Figure 12.8 present the images that composed this sequence.

12.7. PRODUCTION RENDERS

The render production presented in the previous chapter, serves its purpose well in automating the architects' task in non-final stages of the project where the ambience required is a generic one. Moreover, the pre-modeled objects offered by the BIM tool are rather limited and quite simplistic. Naturally a wider research could be conducted for more detailed objects, or new ones could be created, but this task would only add to the amount of time spent in the process, which we are trying to shorten.

Taking advantage of the previously modeled BIM elements for construction purpose, for rendering purposes as well, proved to be a valid way of accelerating the process. Since the modeling task of most elements is required for both scenarios, it makes sense not to duplicate the work. However, for later stages of the project, designing wall coatings, flooring, ceiling paint, light spots and lamp choices, specifying furniture models, etc, the approach does not seem as fit. Some BIM tools have good rendering engines, but none compare to specialized rendering software. For this type of detail modeling, integrating rendering or animation software in the algorithmic process seems a more appropriate choice.

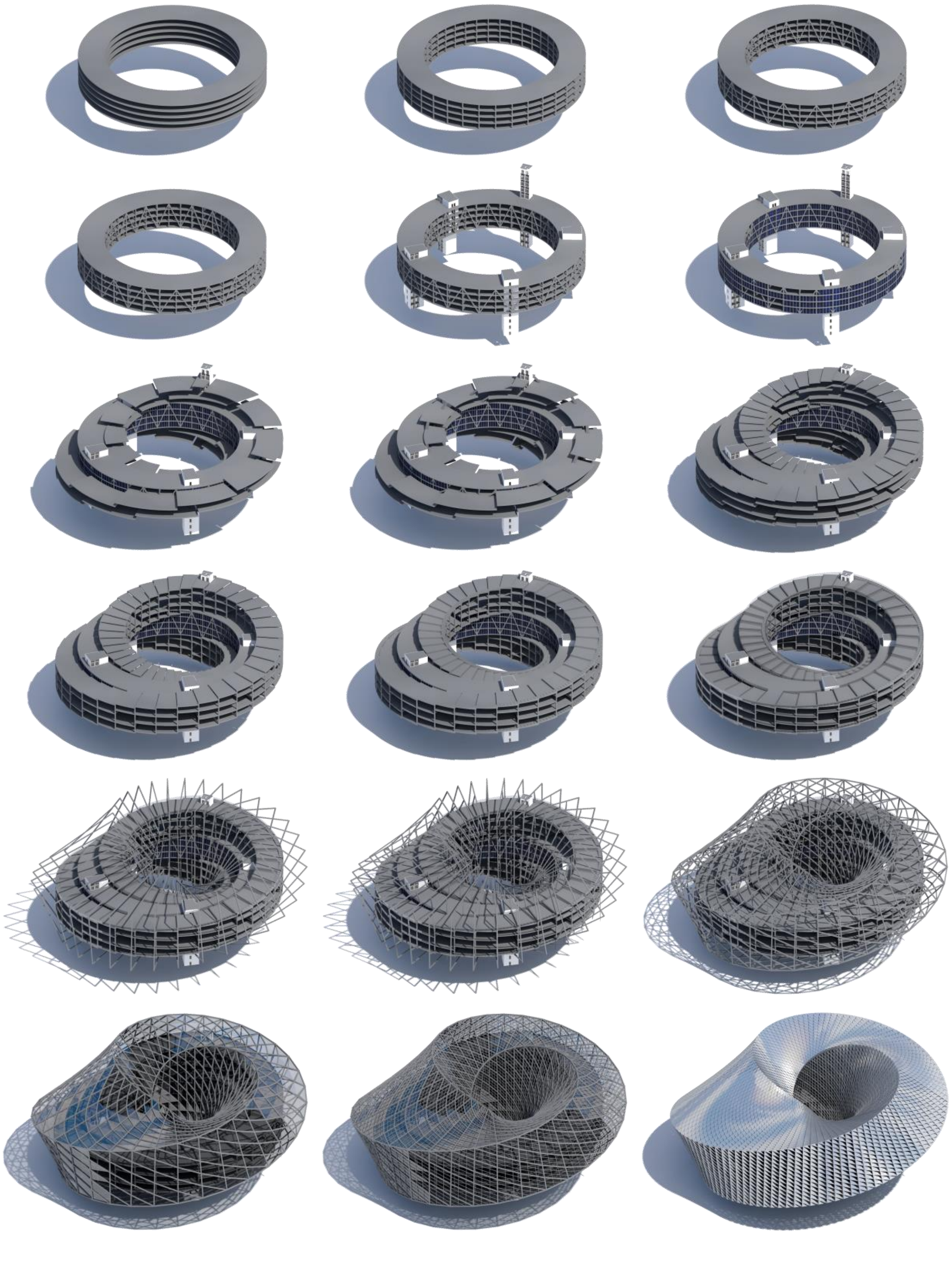


Figure 12.8 - Astana generation sequence rendered

CONCLUSION

The design process of an architectural creation has seen many changes over time, and more so over the past few decades. Architecture is currently a more interactive and collaborative process than ever before. Deploying an architectural project requires a functional network of professionals and resources, and many great architectural studios today engage in a multi tool approach, as each party involved develops their respective specialty in a different tool.

Never before have there been so many or so diverse tools, techniques and methods for design. Architects are spoilt for choice as their practice becomes a liquid discipline, pouring into other domains. Not only must they operate with the tools used by other participants in the process, but the design task itself is currently divided by all the tools offered in the market. Amongst the more important software developments, we have highlighted CAD and BIM software and analysis tools. Each one possesses a set of different advantages to the design process that cannot be left out.

INTEGRATED ALGORITHMIC DESIGN

In this thesis, we proposed a solution that integrates some of the most currently used paradigms and tools in the design process, in a seamless way with no effort when it comes to the translation process between them. For this, we focused on Algorithmic Design, given its capability to transcend factory-set limitations of the currently available digital tools, which allows the user to mold all those tools to his own will and workflow.

We have presented a design methodology that explores an integrated approach to design, where the user is able to produce a single program of his design, which includes not only the model's description but also the necessary steps for its analysis. Using the proposed methodology, the architect may choose to model his design using the CAD or BIM paradigms at any given stage of the design process. Moreover, he can perform analysis of his design directly from the program and incorporate the results in the algorithmic description of the design.

This integrated algorithmic approach to design mitigates the current CAD / BIM / analysis tools portability issues that many practitioners face today. By using the same algorithmic description to generate the models according to the CAD paradigm, the BIM paradigm or the requirements of the analysis tools, we avoid the migration problems typically found when exporting models between different tools.

We have not only described the theoretical frame of the methodology, but also applied it to a case study - Astana National Library - that, in our opinion, belongs within the category of buildings that benefit from an algorithmic-based approach to design. We thoroughly explained the algorithmic modeling of the design for both CAD and BIM applications and the integration of analysis results in the design process. Our investigation was, however, restricted to the backends and modeling operations offered by the chosen portable AD tool. We focused in the available CAD and BIM tools, and on the lighting simulation engine.

The methodology, nevertheless, can be extended to other tools and paradigms that architects may see fit to include in their own design process. The industry is always producing innovative new tools, and architects should be able to keep up with the latest technology without having to change their work methods in order to incorporate it in their process. AD allows for the integration of more tools, paradigms, and workflows at any time, as each user can customize their own process, that is, their program.

Other paths lie ahead unexplored. The following section takes a look at some of the possibilities left to explore in the sequence of the work here developed.

FUTURE WORK

Three main topics for future work are highlighted in this section. The first - "project-specific detail" - is a problem we feel our methodology can still evolve to solve. The second - "space decoration logics" - is a complementary research that, if done, would certainly enrichen the logics behind the methodology. The third – "integrate more backends" - represents what seems, to us, the natural follow up to the work we have developed. In fact the integration of some of the proposed tools is already undergoing, in what regards Rosetta, the portable AD tool we used to explore the IAD approach in this thesis.

PROJECT-SPECIFIC DETAIL

Section 9 narrates the last phase in the detail modeling of our case study but we restricted ourselves to model just some of the buildings detail. However, in order to reach the same level of detail in the whole model, we would have to model our own BIM objects. The use of morph objects to represent the glass railings and the façade glass panels are examples of this limitation.

ArchiCAD's library, as extensive as it is, cannot possibly hold all imaginable variables and parameters to change their objects. Yet architecture must be free itself from the existing object constraints: the fact that one can use pre-modelled objects should not limit the designer's choice and/or imagination. Therefore, the IAD process should also allow the user to model new objects in their program, that are specific to the project in hands, non-existent in the current libraries or existing but with limited parameters/variations.

For this purpose, and considering the use of Rosetta as a tool to apply the IAD approach, we envision some possible improvements to this tool. Namely, a converter from the chosen front-end language (in this case we were using Racket) to the language understood by the BIM backend (GDL, in the case of ArchiCAD, for the modeling of library parts). This would allow the user to model the objects in the same program, with the same programming language used for the whole building, while Rosetta translated them to the BIM backend, creating new objects or families.

SPACE DECORATION LOGICS

The same logic of the specifically modeled objects can apply to detail elements for rendering purposes. This is, in our opinion, a less important investment. Modeling a new BIM object takes time. We consider more important engaging in this endeavor when the project really needs specific pieces in order to be built. A render will always be a mere possible scenario. The furniture placed in the space may not be the one used in the project in the end, whereas, construction elements must be followed as the project documents dictate.

An important investment to make, regarding the automation of the rendering job, however, would be to study logical and generally accepted theories of object placements in the space. Defined standards could be found for object disposition in space, for production renders only, where the main goal is to give the client a generic notion of the space, and not so much, how the space's ambience will eventually be like. As the space's final decoration is not in question at this stage, the architect should not have to concern himself with such details. Standards for measures or distances of the furniture, etc, would facilitate the automation of much of the process.

INTEGRATE MORE BACKENDS

We believe it would also be interesting to extend the integrated tools to include more analysis backends and, possibly, some animation or rendering software.

In this thesis we have focused our analysis integration in solar radiation analysis. However, many other types of analysis can be conducted, such as structural, energetic, acoustic, cost, wind, etc. All of them provide important inputs to the design process and many architectural projects are already making use of various analysis engines simultaneously. A natural step to follow would then be the inclusion of more analysis backends to the methodology that would, not only allow a more informed design in various areas, but also open the door for multi-criteria optimization processes.

The render production we presented proved to be a helpful method for non-final stages of the project. In later stages, however, where the ambience required is of a more detailed nature, the approach does not seem as fit. Some BIM tools have good rendering engines, but none compare to specialized rendering software. For more specialized work it would be interesting to include in the IAD methodology other tools specific for this purpose, such as rendering or animation software. The production of specialized renders can still benefit from the geometry modeled for other purposes, as well as the automation potential offered by AD. Hence, the integration of this additional step into the IAD methodology would likely bring some advantages to the development of this task as well.

BIBLIOGRAPHY

Aghemo, C., Blaso, L., Dalmasso, D., Erba, D., Del Giudice, M., Osello, A., Fracastoro, G., Pellegrino, A. and Ruffino, P. (2013) 'Interoperability between building information models and software for lighting analysis', Building Simulation Applications BSA 2013 1th IBSA Italy Conference, Bolzano, 69-78.

Ahmeti, F. (2007) *Efficiency of Lightweight Structural Forms: The Case of Tree-like Structures - A comparative Structural Analysis*, Vienna: Technische Universität Wien.

AHR (2016) *Al Bahr Towers*, [Online], Available: http://www.ahr-global.com/Al-Bahr-Towers [20 January 2017].

Aish, R. (2003) 'Extensible Computational Design Tools For Exploratory Architecture', in Kolarevic, B. (ed.) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Aish, R. (2011) 'Cultural defence', in Burry, M. (ed.) *Scripting Cultures*, John Wiley & Sons Ltd.

Anderson, K. (2014) *Design Energy Simulation for Architects: Guide to 3D Graphics*, New York and Abingdon: Routledge.

Archello (2010) *ANL - Astana National Library*, 22 July, [Online], Available: http://www.archello.com/en/project/anl-astana-national-library-0 [6 December 2016].

Architect (2016) *Umbrellas for Pink Floyd's Concert Tour of the United States*, [Online], Available: http://www.architectmagazine.com/project-gallery/umbrellas-for-pink-floyds-concert-tour-of-the-united-states-6722 [15 March 2017].

Architizer (2016) *Architectural Details: Herzog & de Meuron's Spectacular Elbphilharmonie*, [Online], Available: http://architizer.com/blog/architectural-details-herzog-de-meuron-elbphilharmonie/ [18 January 2017].

Ateliers Jean Nouvel (2016) *Arab World Institute*, [Online], Available: http://www.jeannouvel.com/en/desktop/home/#/en/desktop/projet/paris-france-arab-world-institut1 [16 January 2017].

Bazjanac, V. (2001) 'Acquisition of Building Geometry in the Simulation of Energy Performance', Seventh International IBPSA Conference, Rio de Janeiro, 305–311.

Bazjanac, V. and Kiviniemi, A. (2007) 'Reduction, Simplification, Translation and Interpretation in the Exchange of Model Data', CIB W78: 25th International Conference on Information Technology in Construction, Santiago, 163–168.

BIG (2017) *BIG*, [Online], Available: http://big.dk/#about [20 January 2017].

Brandon, L. and McLain-Kark, J. (2001) 'Effects of Hand-Drawing and CAD Techniques on Design Development: A Comparison of Design Merit Ratings', *Journal of Interior Design*, vol. 27, no. 2, pp. 26–34.

Burry, M. (2011) *Scripting Cultures*, United Kingdom: John Wiley & Sons Ltd.

Davidson, S. (2017) *Grasshopper: Algorithmic Modeling for Rhino*, [Online], Available: http://www.grasshopper3d.com/ [22 January 2017].

del Campo, M. (2011) 'Cultural Defence', in Burry, M. (ed.) *Scripting Cultures*, John Wiley & Sons Ltd.

Eastman, C., Teicholz, P., Sacks, R. and Liston, K. (2008) *BIM Handbook: A Guide to building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, Hoboken, New Jersey: John Wiley & Sons, Inc.

Fairs, M. (2009) *Astana national Library by BIG*, 25 August, [Online], Available: http://www.dezeen.com/2009/08/25/astana-national-library-by-big/ [27 June 2016].

Feist, S., Barreto, G., Ferreira, B. and Leitão, A. (2016) 'Portable Generative Design for Building Information Modelling', Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA 2016, Hong Kong, 147–156.

Ferreira, B. and Leitão, A. (2015) 'Generative Design for Building Information Modeling', eCAADe (Education and Research in Computer Aided Architectural Design in Europe) and Faculty of Architecture and Urban Planning, TU Wien, Vienna, 635-644.

Ferro, S. (2014) *Weapons, Bribes, And Dictators: Where Architects Draw The Line*, 30 April, [Online], Available: http://www.fastcodesign.com/3028892/slicker-city/weapons-bribes-and-dictators-where-architects-draw-the-line [2 July 2016].

FLUX® (2017) *Flux.io*, [Online], Available: https://flux.io/ [23 April 2017].

Frampton, K. (1983) 'Towards a Critical Regionalism: Six Points for an Architecture of Resistance', in Foster, H. (ed.) *The Anti-Aesthetic: Essays on Postmodern Culture*, Port Townsend, Washington: Bay Press.

Franken, B. (2003) 'Real as Data', in Kolarevic, B. (ed.) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Frazer, J. (2011) 'Cultural defence', in Burry, M. (ed.) *Scripting Cultures*, John Wiley & Sons Ltd.

Gensler (2016) *Shanghai Rising*, [Online], Available: http://du.gensler.com/vol6/shanghai-tower/#/shanghai-rising [6 March 2017].

Gerber, D. and Ibañez, M. (ed.) (2014) *Paradigms in Computing: Making, Machines, and Models for Design Agency in Architecture*, Los Angeles: eVolo Press.

Hensel, M. (2013) *Performance-Oriented Architecture: Rethinking Architectural Design and the Built Environment*, Wiley.

HOCHTIEF ViCon (2015) *Elbe Philharmonic Hall*, [Online], Available: http://www.hochtief-vicon.com/vicon_en/19.jhtml [18 January 2017].

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

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

Kilian, A. (2004) 'Linking Digital Hanging Chain Models to Fabrication', Proceedings of the 23rd Annual Conference of the Association for Computer Aided Design in Architecture and the 2004 Conference of the AIA Technology in Architectural Practice Knowledge Community, Cambridge and Toronto, Ontario, Canada, 110-125.

Kolarevic, B. (2003) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Kolarevic, B. (2014) 'Computing the Performative', in Oxman, R. and Oxman, R. (ed.) *Theories of the Digital in Architecture*, Abingdon and New York: Routledge.

Kolarevic, B. and Malkawi, A. (ed.) (2005) *Performative Architecture: Beyond Instrumentality*, New York: Spon Press.

Kolatan, S. (2003) 'Generative Convergences', in Kolarevic, B. (ed.) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Kotnik, T. and Schwartz, J. (2011) 'Tha Architecture of Heinz Isler', *Journal of the International Association for Shell and Spatial Structures: J. IASS*, vol. 52, no. 3, pp. 185-190.

Leitão, A., Castelo Branco, R. and Cardoso, C. (2017) 'Algorithmic-Based Analysis - Design and Analysis in a Multi Back-end Generative Tool',. Janssen, P. Loh, A. Raonic, M. A. Schnabel (eds.), Protocols, Flows, and Glitches - Proceedings of the 22nd CAADRIA Conference, Xi'an Jiaotong-Liverpool University, Suzhou, China, 137-146.

Leitão, A. and Lopes, J. (2011) 'Portable Generative Design for CAD Applications', ACADIA 2011: Integration through Computation: Proceedings of the 31st annual conference of the Association for Computer Aided Design in Architecture (ACADIA), Banff, 196-203.

Leitão, A. and Santos, L. (2011) 'Programming Languages for Generative Design: Visual or Textual?', eCAADe (Education and Research in Computer Aided Architectural Design in Europe), UNI Ljubljana, Ljubljana, 549-557.

Llach, D. (2013) 'Algorithmic Tectonics: How Cold War Era Reaserch Shaped Our magination of Design', *Architectural Design*, vol. 222, no. Computation Works: The Buildings of Algorithmic Thought, March/April, pp. 16-21.

Marshall, T.B. (1992) 'The Computer as a Graphic Medium in Conceptual Design', Mission - Method - Madness: ACADIA Conference Proceedings, Charleston, South Carolina, 39-47.

Meinhold, B. (2009) *Inhabitat - BIG Architects Unveil Massive Mobius Strip Library for Kazakhstan*, 27 August, [Online], Available: http://inhabitat.com/big-unveil-massive-mobius-strip-library-for-kazakhstan/ [6 December 2016].

Meredith, M. (2008) 'Never Enough (transform, repeat ad nausea)', in Sakamoto, T. and Ferré, A. (ed.) *From Control to Design: Parametric/Algorithmic Architecture*, Barcelona, New York: Actar-D.

Miller, N. (2011) 'The Hangzhou Tennis Center: A Case Study in Integrated Parametric Design', ACADIA Regional 2011: Parametricism, Lincoln, Nebraska, USA, 141-148.

Mitchell, W.J. (2004) 'Foreword', in Kalay, Y.E. (ed.) *Architecture's New Media: Principles, Theories, and Methods of Computer-Aided Design*, Massachusetts Institute of Technology.

Modern Green Structures and Architecture *Kazakhstan's New National Library in Astana*, [Online], Available: http://www.mgsarchitecture.in/projects/500-kazakhstan-s-new-national-library-in-astana.html [27 June 2016].

Morgado, N. *The Presidential Library, Astana*, [Online], Available: http://www.ramboll.com/projects/ruk/the%20presidential%20library%20%20astana?projectid=&text=3 [27 June 2016].

Nerdinger, W. (ed.) (2005) *Frei Otto Complete Works: Lightweight Construction Natural Design*, 2nd edition, Birkhäuser Architecture.

Nguyen, A.-T., Reiter, S. and Rigo, P. (2014) 'A review on simulation-based optimization methods applied to building performance analysis', *Applied Energy*, no. 113, pp. 1043-1058.

Nicol, F. and Pagliano, L. (2007) 'Allowing for Thermal Comfort in Free-running Buildings in the New European Standard EN15251', 2nd PALENC Conference and 28th AIVC Conference on Building Low Energy Cooling and Advanced Ventilation Technologies in the 21st Century, Crete island, Greece, 708-11.

Novaes, R. (2003) *Rotational Villa Girasole*, [Online], Available: http://obviousmag.org/en/archives/2011/01/rotational_villa_girasole.html [16 January 2017].

Novitski, B.J. (1992) 'Gehry Forges New Computer Links', *Architecture - Incorporating Architectural Technology*, August, pp. 105-110.

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

Peters, B. (2013) 'Computation Works: The Building of Algorithmic Tought', *Architectural Magazine*, vol. Computation Works: The Building of Algorithmic Tought, no. 222, pp. 8-16.

Pittman, J.H. (2003) 'Building Information Modeling: Current Challenges and Future Directions', in Kolarevic, B. (ed.) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Radiance (2017) *RADSITE*, [Online], Available: https://www.radiance-online.org/ [05 May 2017].

Rahim, A. (2003) 'Designing and Manufacturing Performative Architecture', in Kolarevic, B. (ed.) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Ramboll (2010) *The Presidential Library, Astana — Close Working Archives Buildadility*, [Online], Available: http://www.ramboll.no/news/ruk/presidentiallibraryastana [27 June 2016].

Reitan, P.H. (2005) 'Sustainability science - and what's needed beyond science', *Sustainability: Science, Practice, & Policy*, vol. 1, no. 1, pp. 70-80.

Rosen, R. (1995) 'Cooperation and Chimera', in Casti, J. and Karlqvist, A. (ed.) *Cooperation and Conflict in General Evolutionary Processes*, John Wiley & Sons, Inc.

Sheil, B. (2008) 'Protoarchitecture: Between the Analogue and the Digital', *Architectural Design - Protoarchitecture: Analogue and Digital Hybrids*, vol. 78, no. 4, pp. 6-11.

Shusta, C. (2006) 'Epi(multi)logue', in KostasTerzidis (ed.) *Algorithmic Architecture*, Oxon and New York: Architectural Press.

Sloterdijk, P. (2005) 'Atmospheric politics', in Latour, B. and Weibel, P. (ed.) *Making Things Public*, MIT Press.

Stankovic, S., Campbell, N. and Harries, A. (2009) *Urban Wind Energy*, London: Earthscan.

Stinson, L. (2017) *What Happens When Algorithms Design a Concert Hall? The Stunning Elbphilharmonie*, 01 December, [Online], Available: https://www.wired.com/2017/01/happens-algorithms-design-concert-hall-stunning-elbphilharmonie/#slide-1 [6 March 2017].

Takim, R., Harris, M. and Nawawi, A.H. (2013) 'Building Information Modeling (BIM): A new paradigm', *Procedia - Social and Behavioral Sciences*, vol. 101, pp. 23-32.

Terzidis, K. (2006) *Algorithmic Architecture*, Oxon and New York: Architectural Press.

The Works of Gaudi (2016), [Online], Available: http://personal.strath.ac.uk/j.wood/Biomimetics/Inspirational%20people/Antoni%20Gaudi_files/Antoni%20Gaudi.htm [15 March 2017].

Tomasetti, T. (2017) *CORE studio | Thornton Tomasetti*, [Online], Available: http://core.thorntontomasetti.com/tag/ttx/ [23 April 2017].

Whitehead, H. (2014) 'laws of Form', in Oxman, R. and Oxman, R. (ed.) *Theories of the Digital in Architecture*, Abingdon and New York: Routledge.

Winstanley, T. (2011) *AD Classics: Institut du Monde Arabe / Jean Nouvel, Architecture-Studio, Pierre Soria and Gilbert Lezenes*, [Online], Available:

http://www.archdaily.com/162101/ad-classics-institut-du-monde-arabe-jean-nouvel [17 March 2017].

Woodburry, R. (2010) *Elements of Parametric Design* , Oxon: Routledge.

World Architecture News (2009) *BIG wins Cityscape Dubai Architectural Award*, 22 October, [Online], Available: http://www.worldarchitecturenews.com/in-brief/356/big-wins-cityscape-dubai-architectural-award.html [27 June 2016].

World-architects (2015) *BIG – Bjarke Ingels Group*, [Online], Available: http://www.world-architects.com/en/big/ [16 September 2016].

Wujec, T. (2011) *Imagine, Design, Create*, Autodesk.

Yessios, C.I. (2003) 'Is There More To Come?', in Kolarevic, B. (ed.) *Architecture in the Digital Age: Design and Manufacturing*, Spon Press – Taylor & Francis Group.

Zboinska, M. (2015) 'Hybrid CAD/E platform supporting exploratory architectural design', *Computer-Aided Design*, no. 59, pp. 64–84.