POLITECNICO
MILANO 1863

School of Architecture Urban Planning and Construction Engineering Degree

program in Building and Architectural Engineering

MASTER THESIS

Generative building massing optimization in parametrical BIM environment: Evaluating different parametric BIM workflows from Grasshopper to Revit at conceptual design stage.

Professor: Mrs Giuliana Iannaccone

Obando Erik, 886335

May 2020

Acknowledgment

First of all, I would like to thank the Ecuadorian Government that through its institutions IFTH and SENECYT sponsored this master's studies and gave me the opportunity to be part of an elite academic environment in Politecnico di Milano.

To my thesis supervisor, Professor Giuliana Iannaccone, for her guidance and valuable supervision in these particular times. And a special thanks to assistant professor Matteo Pedrana whose advice was the kick-off for this investigation.

And to the visual programming community for having or been willing to find the right answer for every possible problem on the way of this research.

Table of Contents

List of figures

List of Tables

List of Annexes

1 Abstract

This new industrial revolution, known as Industry 4.0, has challenged the Architectural Engineering and Construction (AEC) industry by showing the potential of digitalization and interoperability on the construction field (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019). To face these challenges the industry has developed tools like Building Information Modeling (BIM) and Parametrical Design (PD). Nevertheless, there is a gap between digital practice and design theory (Ambrose, 2007), where BIM data exchange reflects insufficiently the thinking of parametric design (Wortmann & Tunçer, 2017). As a matter of fact, an improvement in the integration between this two approaches strength the design process (Boeykens, 2012). The simultaneous combination of design exploration and construction information in one model is referred as parametric BIM workflows (Janssen, 2014).

This research evaluates a methodology where PD and BIM modelling are coupled since the conceptual design stage. For this purpose, a residential building with a pixel/pop-out architectural concept is developed in its conceptual phase. The building massing is scripted and optimized in Grasshopper applying genetic solvers like Octopus and Galapagos, the parameters to be optimized are the pop outs distribution and the internal layout of the building. As a result, thanks to many iterations the designer can make a final data driven decisions.

After building massing is set, the parametrical elements are enriched and streamed to a BIM environment, in this research the BIM approach is done with Revit. The information is transfer by three Grasshopper and Revit extensions, exploring different parametric BIM workflow approaches and model coupling like: IFC information transfer through loosely coupled approach with GeometryGym, Revit API and CVS file for a tightly coupled approach in Hummingbird, and Cloud base workflows and Dynamo through tightly coupling via Speckle.

As conclusion of this research it appears that parametrical BIM workflow can support a data base design methodology that strength the automatization and interoperability. In the parametrical model it is possible to add an important amount of information from different construction fields and automated results thanks to numerous iterations. Furthermore, is possible to affirm that coupled models can be update at real time with the right set-up, therefore increasing the interoperability between PD and BIM.

Sommario

Questa nuova rivoluzione industriale, Industria 4.0, ha sfidato l'industria dell'ingegneria architettonica e delle costruzione (AEC) mostrando il potenziale della digitalizzazione e dell'interoperabilità nel campo dell'edilizia (Maskuriy, Selamat, Ali, Maresova e Krejcar, 2019). Per far fronte a queste sfide, l'industria ha sviluppato strumenti come Building Information Modeling (BIM) e Parametrical Design (PD). Tuttavia, esiste un divario tra la pratica digitale e la teoria del design (Ambrose, 2007), in cui lo scambio di dati BIM riflette in modo insufficiente il pensiero del design parametrico (Wortmann & Tunçer, 2017). In effetti, un miglioramento dell'integrazione tra questi due approcci rafforzano il processo di progettazione (Boeykens, 2012). La combinazione simultanea di esplorazione del progetto e informazioni sulla costruzione in un modello è definita flusso di lavoro parametrico BIM (Janssen, 2014).

Questa ricerca valuta una metodologia in cui la modellazione PD e BIM è accoppiata sin dalla fase di progettazione concettuale. A tale scopo, nella sua fase concettuale viene sviluppato un edificio residenziale con un concetto architettonico pixel/pop-out. La volumetrie dell'edificio è scritta e ottimizzata in Grasshopper applicando solutori genetici come Octopus e Galapagos, i parametri da ottimizzare sono la distribuzione dei pop-out e il layout interno dell'edificio. Di conseguenza, grazie a molte iterazioni, il progettista può prendere decisioni definitive basate sui dati risultanti.

Dopo aver impostato la creazione di massa, gli elementi parametrici vengono arricchiti e trasmessi in streaming a un ambiente BIM, in questa ricerca l'approccio BIM è fatto con Revit. Le informazioni vengono trasferite da tre estensioni Grasshopper e Revit, esplorando diversi approcci del flusso di lavoro BIM parametrico e accoppiamento del modello come: trasferimento di informazioni IFC attraverso un approccio vagamente accoppiato con GeometryGym, Revit API e file CVS per un approccio strettamente accoppiato in Hummingbird e flussi di lavoro basati su cloud e Dinamo tramite accoppiamento stretto con Speckle.

Come conclusione di questa ricerca sembra che il flusso di lavoro BIM parametrico possa supportare una metodologia di progettazione basata su dati che rafforzino l'automazione e l'interoperabilità. Nel modello parametrico è possibile aggiungere una quantità importante di informazioni da diversi campi di costruzione e risultati automatizzati grazie a numerose iterazione. Inoltre, è possibile affermare che i modelli accoppiati possono essere aggiornati in tempo reale con la giusta configurazione, aumentando quindi l'interoperabilità tra PD e BIM.

2 Introduction

2.1 Motivation

Digital revolution has transformed retailing, publishing, traveling, etc. And, more recently, it is changing the way we plan, design, build and maintain our social and economic infrastructure (Bew, 2015). This new industrial revolution is known as Industry 4.0, and it has challenged the Architectural Engineering and Construction (AEC) industry by showing the potential of digitalization and interoperability on the construction field. It has included in the industry the availability to automatically gather and process digital data and grant online digital access into the value chain of projects. (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019)

More and more sectors of industry are entering Industry 4.0, although regarding AEC industry, this is not the case. (Kovacs & Szoboszlai, 2019). As matter of fact, the industry has been criticized by its low margins and lack of innovation for years (Deloitte, 2018). Even if the potential of new technologies to transform many industries has been demonstrated, the building environment has been slow to adopt innovative processes and remains one of the most reluctant major industrial sectors to embrace new ways of working (Bew, 2015).

The discipline has not completed the precedent revolutionary cycle, and both, academics and practice around the world are still discovering the advantages of the digital era with a limited approach on the building industry (Arturo Tedeschi, 2014). Furthermore, different internal and external pending challenges can be responsible for this unimpressive evolution in the industry, such as: a constant fragmented industry, a lack of collaboration with suppliers and contractors, and a scarce transfer of knowledge between projects (World Economic Forum, 2016). To summarize, the industry presents as a pattern two major issues causing its technological development stagnation:

- Interoperability
- Innovative digitalization process

AEC industry is already tackling these issues and developing solutions to approach industry 4.0. There is an extensive potential to improve productivity and efficiency thanks to digitalization, innovative technologies, and new building techniques. Meaning a swift development on augmented reality, drones, 3D scanning and printing, Building Information Modelling (BIM), autonomous equipment (World Economic Forum, 2016). Among those, two tools stand out to solve interoperability and innovative digital process in this new industrial revolution:

- Building information modeling
- Parametric modeling.

Both lead to an intense and reinforced design process and include the essential objective of problem solving considering the integration of many function and factors. (Haliburton, et al., 2011). Long over-due changes in process from the analogue into the digital world are addresses in BIM, meaning the control, manage and interoperability of an unprecedent volume of digital data information (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018). This open the potential of innovation regarding parametrical design, where transforming digital data in design parameters can be a useful tool for BIM. In fact, the potential of algorithmic or parametric design is to generate and control design and information beyond human capabilities. As a matter of fact, it allows designers to develop new solutions and to manage data and information beyond traditional CAD software and 3D modelers limitation, leading to unpredictable results that answer to the collected parameters. (Arturo Tedeschi, 2014).

Nevertheless, there is a gap between digital practice and design theory (Ambrose, 2007), where BIM data exchange reflects insufficiently the thinking of parametric design (Wortmann & Tunçer, 2017). Design approach can be diametrical different between BIM and PD. While, BIM tools focus on design interrogation by adding detailed information, graph based system are set up to design exploration (Banihashemi, Tabadkani, & Hosseini, 2018). PD process is almost exclusively focus on modeling because it is mainly oriented on advance support for freeform geometry. Hence, BIM advantages, like structured model, embedded information, link to construction documentation are mostly missing in a parametric approach (Boeykens, Bridging Building information modeling and parametric design, 2012). As a matter of fact, contrary to the functionality of Parametric Design software, applications that apply BIM concepts presents a discontinuity of the design process, preventing the designer to step back and forth throughout the different design process or scale levels. (Boeykens, 2016).

Therefore, digital design and optimization must confront reality as early as possible such as the concept stage answering to performance and functional criteria, thus achieving tailormade solutions to the contemporary constrains of the building industry (Arturo Tedeschi, 2014). In fact, PD and BIM are linked since an early stage, where parametric modelling and data optimized analysis are at the beginning of BIM application value of chain. (World Economic Forum, 2016). There is a key point in the design where from minimal exploration modelling you must go to its maximal detail; this means going from conceptual modelling tools to BIM tools (Jassen, 2016). As a matter of fact, an improvement in the integration between this two approaches strength the design process (Boeykens, 2012). The simultaneous combination of design exploration and construction information in one model are in an increasing demand, where the couple approaches that link graph-based systems and BIM systems are preferable. This bridging approach is referred as parametric BIM workflows (Janssen, 2014).

Parametric BIM workflows creates a directly generative associative BIM model (Janssen, 2014). This means that the iterative behavior of Parametric Design adds to the BIM data structure with data matching algorithms that correctly interact with these data structures. Contrary to an exported compatible model which only generate explicit geometry, parametrical BIM workflows generates and associative model with its corresponding BIM information, thus allowing a more user friendly and streamlined BIM workflows (Janssen, 2014). The challenge is to keep a whole and simple intuitive process for designers through the balance between, advanced rules with intelligent interface, and intuitives and simple ways for the designer to override this intelligence. (Jassen, 2016)

Hence, stablishing a parametric BIM workflow since the conceptual design phase can represent an improvement in the design methodology and enhance the communication, coordination and data management between design, engineering, and construction. The aim of this research is to create and evaluate a generative building massing optimized by genetic solvers in PD environment, and asses the different possibilities of model coupling in the BIM environment. In this thesis, the BIM approach is represented by Revit, meanwhile parametric design is defined by Grasshopper.

2.2 Methodology

The first chapter of this research explain the relation of Parametric BIM workflow in the AEC industry, and why it is important to stablish coupled models of PD approach and BIM environment.

Then, the second chapter contains the state of art regarding: BIM, Parametrical Design, and parametric BIM workflow. Literature is reviewed to understand individually which are their roles in the AEC industry and which are the components and limitations of these tools.

Afterwards, the third chapter present a case study. First, a parametric model building massing is created to show the flexibility and the different types of information that is possible to parametrize. Then, the model is transfer through three different methods of model coupling in parametric BIM workflow to evaluate their characteristics and performances.

Finally, the fourth chapter displays the conclusion and further recommendations, regarding the entire capabilities and impact of integrating parametric design and BIM through a parametrical BIM workflow in the early stage for design.

2.3 Limitations

The study is limited by the design stage. At a conceptual stage, performance analysis for optimization can only assess limited information. Therefore, detailed analysis like envelope thermal transition, or energy performance are not possible at this level. However, once a parametrical BIM workflow is stablished is always possible to add more information according to the level of detail in the design stage.

Data transfer direction is another limitation. This study considered only one-way direction of data from Parametrical Design to Building Information Modeling. Nevertheless, parametric BIM workflow also considers a bi-directional transfer, from BIM to PD.

This analysis does not consider the time and knowledge to build up the visual programming script on the parametric modeler. Therefore, a comparison with a traditional design methodology regarding time efficiency has not been considered.

The study limits 2D manual drawing, all the geometry generated in this research is parametrized and scripted.

This study is limited to the existing tools in computational design to apply a parametric BIM workflow. It stablishes an evaluate a methodology of transfer but does not present a new software development.

3 Parametrical BIM workflow in the AEC industry

3.1 Context

3.1.1 Industrial 4.0 and the AEC industry

Digital revolution has transformed retailing, publishing, traveling, etc. and, more recently, digital technology is changing the way we plan, design, build and maintain our social and economic infrastructure (Bew, 2015). This new industrial revolution is known as Industry 4.0, and it has challenged the AEC industry by showing the potential of digitalization on the construction field. It has included in the industry the availability to automatically gather and process digital data and grant online digital access into the value chain of projects. (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019)

Digitalization represents a generational opportunity in the construction sector to redefine structural processes of the industry by (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018) :

- taking advantages from common practices of other industrial sector and engineering.
- including digital workflows.
- adding technology skills to shift to a higher level of performance.

Furthermore, the main technologies that identify Industry 4.0 for people are (Kovacs & Szoboszlai, 2019):

- Communication among machines
- Big data
- Artificial intelligence (AI)
- Cloud computing,
- Decision-supporting systems
- Personalized mass production

The construction sector is arriving to industry 4.0 through the digital transformation. Therefore, digitalization and technology are gradually becoming a regular practice in the AEC representing an economical benefit and the instrument of innovation in the industry. As a matter of fact large construction companies are increasing their interest on innovation and digitalization of their business models (Deloitte, 2018).

An increase in productivity by an efficient adoption of contemporary digital measures can have a major impact. For example, a 1% in the rise of productivity worldwide can represent an economy of $100 billion a year in the construction industry. (World Economic Forum, 2016)

Thus, business activities on top European construction companies are making a strong effort to include and understand the benefits of innovation and technology development. The next figure represents the connections between all clusters involved inside ConTech ecosystem in construction and how digitalization is shaping the industry (Deloitte, 2018).

ConTech ecosystem

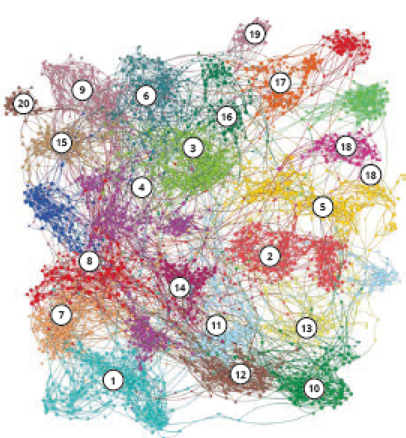


Figure 7 – Visualisation of the ConTech ecosystem (Sources: Quid, CapitalIQ, Crunchbase – Deloitte Analysis, data as at 03-12-2018)

Cluster		Share
1	Machinery, components & assembly	8%
2	General contracting	7%
3	Construction management software	7%
4	Robotics & 3D printing	6%
5	Building Information Modelling	6%
6	Consulting	6%
7	Construction materials and technologies	6%
8	Trading & investments	5%
9	Networks, cable & fiber	5%
10	Landscape construction, environment, waste & water	5%
11	Oil & gas	4%
12	Engineering & technology	4%
13	Road & railway, structural engineering	4%
14	Solar energy	3%
15	Artificial Intelligence, IoT, smart buildings	3%
16	Cad, surveying, laser	3%
17	Staffing employment & HR	3%
18	Plumbing & HVAC	2%
19	Drones & UAVs	1%
20	AR/VR/MR	1%

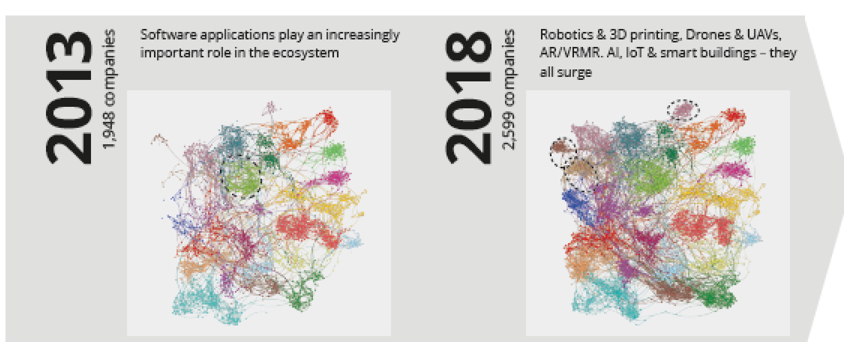


Figure 1: ConTech Ecosystem Evolution (Deloitte, 2018).

3.1.2 The AEC is behind in digital development

More and more sectors of industry are entering Industry 4.0, although regarding AEC industry this is not the case. (Kovacs & Szoboszlai, 2019). As matter of fact, the industry has been criticized by its low margins and lack of innovation for years (Deloitte, 2018). Even if the potential of new technologies has been demonstrated to transform many industries, the building environment has been slow to adopt innovative processes and remains one of the most reluctant major industrial sectors to embrace new ways of working (Bew, 2015). In addition, this slow and steady pace has a direct impact in the sector economical potential.

The discipline has not completed the precedent revolutionary cycle, both, academics and practice around the world are still discovering the advantages of the digital era with a limited approach on the building industry (Arturo Tedeschi, 2014). The industry needs to reproduce; in its diverse and heterogeneous value chain; tools and management philosophies from more develop sectors such as aeronautics or automotive industries (Grilo & Jardim-Goncalves, 2010).

The AEC sector has been dubious regarding technological opportunities and has not fully embrace them, as consequence its labor productivity has stagnate accordingly (World Economic Forum, 2016). Although efforts have been done in the industry, digitalization progress slowly. Considering ConTech ecosystem, the first 5 years (after its development around 2003) Building information Modeling became a structural part of the system, nevertheless its increase slow and even around 2013 growth did not accelerate (Deloitte, 2018). For example, an existing online survey carried out by the Lechner Knowledge Center in 2017 with 89 participants form the Hungarian AEC industry were asked what they use BIM for. As a result, 56% use it for documentation, and 45% for supporting the design process, these represents basic functions of Industry 3.0, meaning that industry is using a limited potential of technological tools (Kovacs & Szoboszlai, 2019).

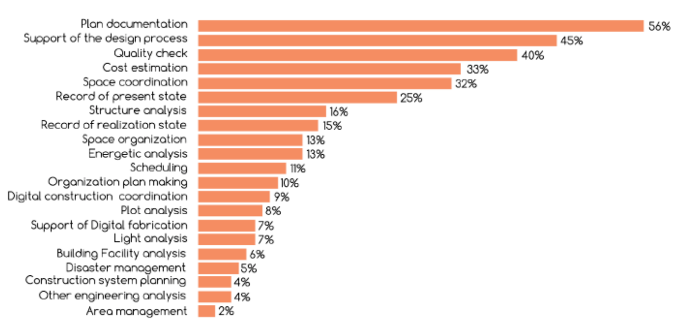


Figure 2: BIM use int he Hungarian AEC industry concerning function (Kovacs & Szoboszlai, 2019).

Slow pace of innovation is relevant, because of the great scale and impact of AEC industry. The industry accounts for about 6% of global GDP and is growing, in countries like India it can reach the 8% of GDP. In addition, the industry is the largest consumer of raw materials and other resources, it represents 3 billion tons of used raw materials and the 50% of global steel production. (World Economic Forum, 2016). Furthermore, These limitations produce unpredictable cost overruns, late delivery of public infrastructure and avoidable project changes, and, as a result poor value for public money and a higher financial risk. (EU BIM, 2017)

Different internal and external pending challenges can be responsible for this unimpressive evolution in the industry, such as: a constant fragmented industry, the lack of collaboration with suppliers and contractors, recruiting suited talented workforce, and a scarce transfer of knowledge between projects (World Economic Forum, 2016). To sum up, the industry presents as a pattern two major issues causing its technological development stagnation:

- Interoperability
- Innovative digitalization process

Because of its heterogeneous environment, interoperability is recognized as a problem in the AEC industry. The sector is defined as highly dynamic and adaptable with multiple applications and systems, and a large range of different players. As a result; and despite, standardized data models and services for main activities; a seamless global interoperability looks far from being achieve. To be specific, considering the information and communication technologies (ICT) a major obstacle that AEC companies are facing is the lack of interoperability of software applications to manage and coordinate projects (Grilo & Jardim-Goncalves, 2010)

On the other hand, innovation is arriving late in different construction fields. For example, construction automation still not fully employed, because technical aspects and technologies are still being investigated. BIM, cloud computing, mobile computing and modularization havent reach maturity (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019). To move forward, next generation of collaborative designer and contractors must move forward stablished processes and role stereotypes and embrace new working methods. (RIBA, 2012)

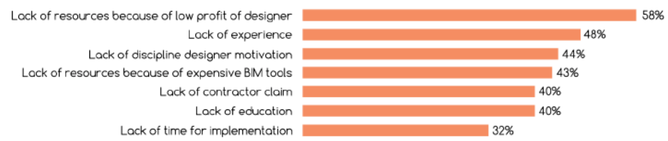


Figure 3: Top barriers of new Technology adoption in the Hungarian AEC Industry (Kovacs & Szoboszlai, 2019)

3.1.3 Interoperability and innovation

AEC industry is already tackling these issues and developing solutions to approach industry 4.0. There is an extensive potential to improve productivity and efficiency thanks to digitalization, innovative technologies, and new building techniques. Meaning a swift development on augmented reality, drones, 3D scanning and printing, Building Information Modelling (BIM), autonomous equipment (World Economic Forum, 2016). Two tools stand out to solve interoperability and innovative digital process in this new industrial revolution: Building information modeling and parametric modeling. Both lead to an intense and reinforced design process and include the essential objective of problem solving considering the integration of many function and factors. (Haliburton, et al., 2011). In addition, the growth of these innovations in the AEC industry shows that a solid foundation of software application and BIM is in place and allows an improvement of the industry (Deloitte, 2018).

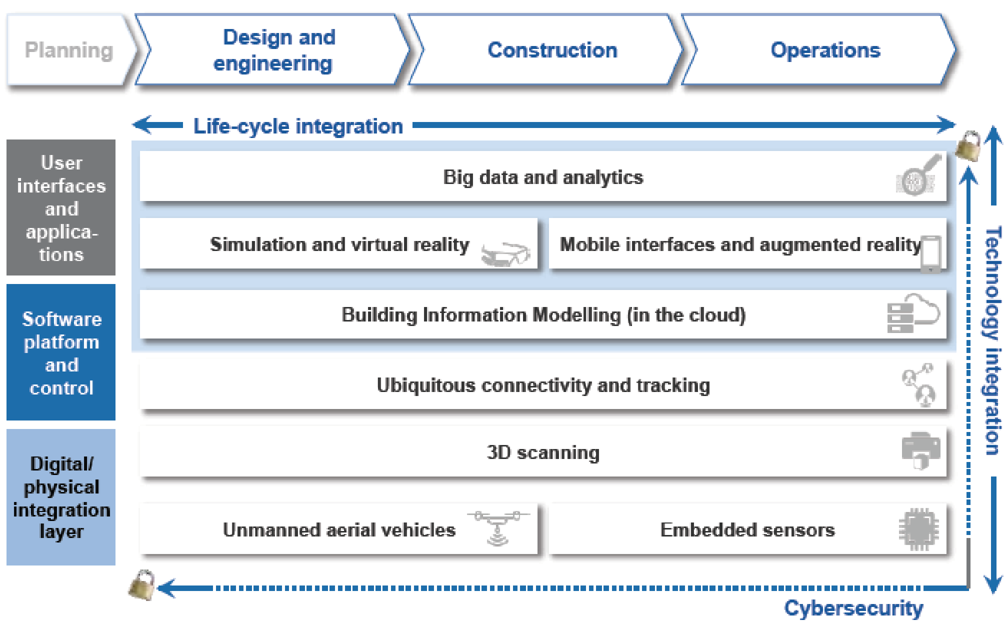


Figure 4: Digital Technologies Applied in the E&C Value Chain (World Economic Forum, 2016).

Technology and innovation are at the foundation of the AEC evolution in Industry 4.0 it had introduced new digital technologies, sensor systems, intelligent machines, and smart materials are stored, and BIM plays a central role in this cyber-physical system, around this system BIM functionalities improve the construction lifecycle. As a result, the fourth industrial revolution has made BIM a central repository for collecting digital information about a project. (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019)

Furthermore, the use of BIM has enable great productivity and workflow quality by reducing costs and time loads (Alfabuild ; Санджиев, et al., 2018). In fact, it has been implemented promptly as a strategic tool in different parts of the value chain for cost savings, productivity and operations efficiencies, improved infrastructure quality and better environmental performance. (Wortmann & Tunçer, 2017)

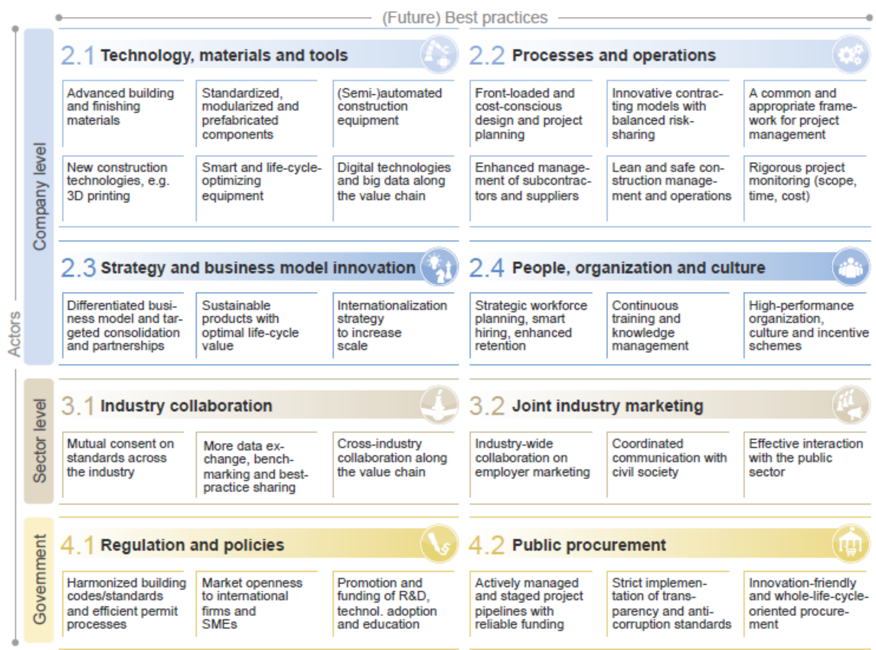


Figure 5: Industry Transformation Framework (World Economic Forum, 2016).

Top construction companies are focusing on digital innovation as a part of their strategy, and almost all of them include BIM in their projects (Deloitte, 2018). As a matter of fact, national public construction sector leading BIM interoperability programs has increased significantly, allowing the opportunity to expand sharing common practices (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018). For instances, Construction Best Practice Program (CBPP) Kingdom and an industry-led Movement for Innovation (M4I) at United Kingdom, action program

(SCADEC) stablished by Japanese Ministry of Construction, whose main objective is to develop a neutral CAD data exchange format based on STEP AP202 (Grilo & Jardim-Goncalves, 2010)

Long Over-due changes in process from the analogue into the digital world are also addresses in BIM, meaning the control and manage of an unprecedent volume of digital data information (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018). This open the potential of innovation regarding parametrical design, where transforming digital data in design parameters can be a useful tool for BIM. In fact, the potential of algorithmic or parametric design is to generate and control design and information beyond human capabilities. As a matter of fact, it allows designers to develop new solutions and to manage data and information beyond traditional CAD software and 3D modelers limitation, leading to unpredictable results that answer to the collected parameters. (Arturo Tedeschi, 2014). In addition, graphic/digital primitive attributes are normally fixed at any time while in a parametric composition they remain variable. They respond to a defined range of associative functions developing and in-built intelligence. (Schumacher, 2010)

3.2 Problematic

3.2.1 There is a gap between parametric design and BIM

The expansion of parametric design and BIM tools involves a larger number of participants in a design process including architects, consultants, and contactors, reinforcing the need for independent methods of data exchange between software and programming languages. Nevertheless, there is a gap between digital practice and design theory (Ambrose, 2007), where BIM data exchange reflects insufficiently the thinking of parametric design (Wortmann & Tunçer, 2017). As a matter of fact, contrary to the functionality of parametric design software, applications that apply BIM concepts presents a discontinuity of the design process, preventing the designer to step back and forth throughout the different design process or scale levels. (Boeykens, 2016).

In addition, workflow limitation have been identified regarding design phases in the AEC industry (Boeykens, 2016). Technology and standards for collaborative work between these tools are currently in an embryonic phase (Alfabuild ; Санджиев, et al., 2018). In fact, it is difficult to automate the materialization process to create a parametric model that; not being maximal in their information content; can be seamless imported into existing BIM tools, and be easily modify and further developed (Jassen, 2016).

3.2.2 Why this gap exists? Difference between PD and BIM

Parametrical design and Building Information Modelling are similar in the large amount of data collected, classified, and process in their models. In parametric modeling, objects are model and control by an overall logical script or scenario. Likewise, BIM models are extensively well structured, with a clear semantic information. Hence, in BIM, information about the project is created at the same time that objects are modeled, and more information is inserted in the properties items. Meanwhile, in parametric design, collected information define the project, and is gathered to create and modified the objects accordingly to the inserted data. So, if the concepts of these tools are similar, why are their applications so different? (Boeykens, Bridging Building information modeling and parametric design, 2012)

Design approach can be diametrical different between BIM and PD. While, BIM tools focus on design interrogation by adding detailed information, graph based system are set-up to design exploration (Banihashemi, Tabadkani, & Hosseini, 2018). PD process is almost exclusively focus on modeling because it is mainly oriented on advance support for freeform geometry. Hence, BIM advantages, like structured model, embedded information, link to construction documentation are mostly missing in a parametric approach (Boeykens, Bridging Building information modeling and parametric design, 2012).

In the AEC industry, what starts like a freeform geometry on PD once translated to BIM tools needs to be more precise and specific, requiring information like thickness, materials and other details to be defined for further construction (Jassen, 2016). While BIM software use internal algorithms and set up information about the construction domain to limit the amount of direct modeling, parametric modeling systems develops a recipe for a particular project, this can be assumed as an automated composition of geometric entities (Boeykens, 2012). Unfortunately, BIM Systems results in an associative modelling, which limits the ability to automate a model building process (Janssen, 2014). And, on the other hand while parametric design should involve BIM as a part of the project, its roles is generally limited to post design elaboration and construction documentation.

Integrated scripting and visual programming support or data flow modeling are other strong points that are missing or still very limited in BIM implementation, creating a separate communities of end users a practitioner (Boeykens, 2012). Most designers are already thinking parametrically, but nor having the inclination or the time to learn programming skills they do not have the knowledge to express or explore both tools simultaneously. This reinforces the need for software and programing language-independent methods of data exchange and workflow. For example: IFC classes; which lies at the heart of BIM; not necessarily ensure data exchange and likely insufficiently reflect the thinking of parametric designers. (Wortmann & Tunçer, 2017)

3.3 Solution

3.3.1 Parametric design in BIM universe

Although PD and BIM demand a larger base knowledge and higher skill set than regular design tools and procedures, they create an opportunity to overcome traditional architectural radically change architectural and building design process. Regarding the AEC industry both have projected design into the twenty first century, and show potential to bridge the future professional challenges (Haliburton, et al., 2011). The challenge is to size the opportunities presented when digitally driven design, process and production technologies are understand and seen more than just mere tools (Ambrose, 2007).

BIM and PD are closely related in the industry, since parametric modelling is one of the six functionalities components in BIM (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019). The articulation between algorithmic model and BIM consolidates a powerful tectonic process, where mechanisms for importing, exporting an interaction of geometric parameters, and construction information on different programs have been integrated (Maravilha De Azevedo, 2009). Furthermore, BIM plays a central role as enabler or facilitator for many other technologies where, for example, the building of a bridge can be aided by the combination of robotics and 3D printing via a parametrically design 3D model (World Economic Forum, 2016).

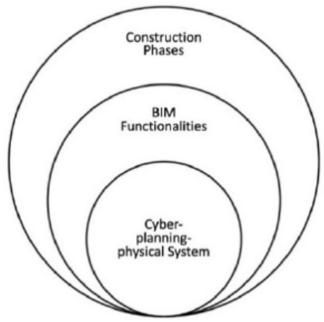


Figure 6: Relation between cyber-planning-physical system, BIM functionalities, and construction phases (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019).

Innovations of this kind enables new functionalities along the entire value of chain, from the early design phase to the end of its life cycle and demolition. (World Economic Forum, 2016). In order to achieve substantial improvements in construction productivity and operation and maintenance (O&M) costs reduction, companies need to ensure that the construction process, as well as the final operation phase, are in mind during the design and engineering phase (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018). Thus, digital design and optimization must confront reality as early as possible such as the concept stage answering to performance and functional criteria, hence achieving tailormade solutions to the contemporary constrains of the building industry (Arturo Tedeschi, 2014). In fact, PD and BIM are linked since an early stage, where parametric modelling and data optimized analysis are at the beginning of BIM application value of chain. (World Economic Forum, 2016).

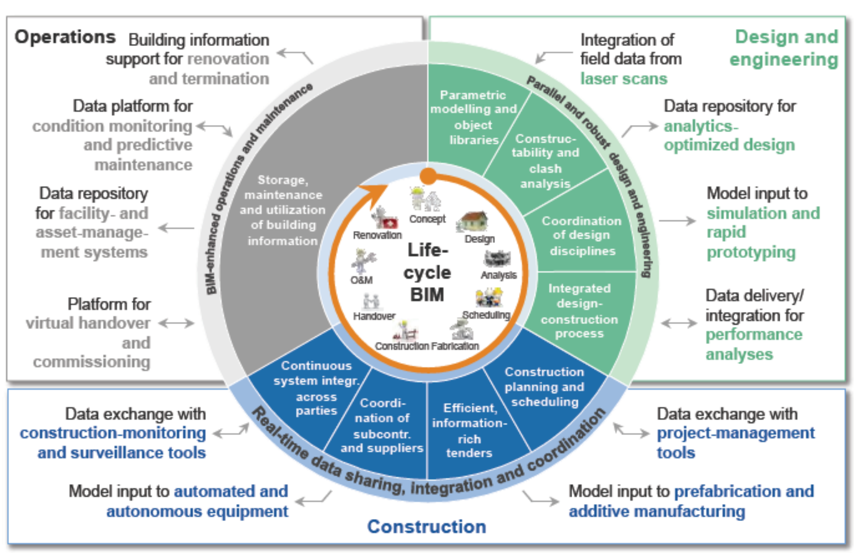


Figure 7: Applications of BIM along the E&C Value Chain (World Economic Forum, 2016).

Furthermore, early coordination and workflow regarding these tools have a great impact on construction savings. Regarding the total cost of ownership (TCO) perspective, construction cost of the total lifecycle cost of the project can be as high as 10-50%, while the O&M costs may account for 40-80%. Both are largely determined early on, during the design and engineering phase. Through early stages it is relatively easy and inexpensive to make changes, thus a significant value is created by making whole life cycle conscious decision and finding the right innovative and data driven solution for design and engineering (World Economic Forum, 2016).

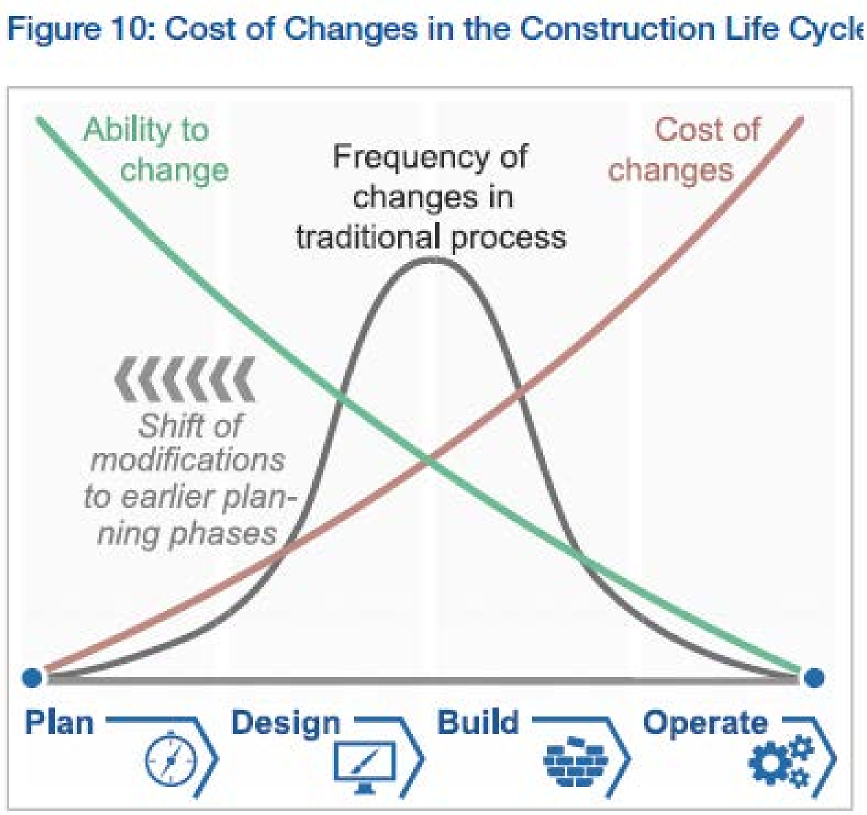


Figure 8: Cost of Changes in the Construction Life Cycle (World Economic Forum, 2016).

The ongoing transformation of the AEC industry will rely increasingly on BIM and other digital tools since there is a potential for coordinating all the stakeholders of the construction project and facilitate the construction process on-site. (World Economic Forum, 2016). Furthermore, at the planning stage, PD and BIM interaction enables designer, owners and user to work together producing and testing in computer before projects are built (Bew, 2015). Contractor, subcontractors, suppliers, and later owner and maintenance firm, and all the companies along the value chain should ideally incorporate their knowledge in an early phase. Benefits include: better coordination, faster production of accurate and reliable information, improvement in decision making, and quality of outputs (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018). As example, early project planning improvement with a minimal increase in upfront cost of about 2% supporting optimized design leads on average to a life cycle saving of 20% on total cost. (World Economic Forum, 2016).

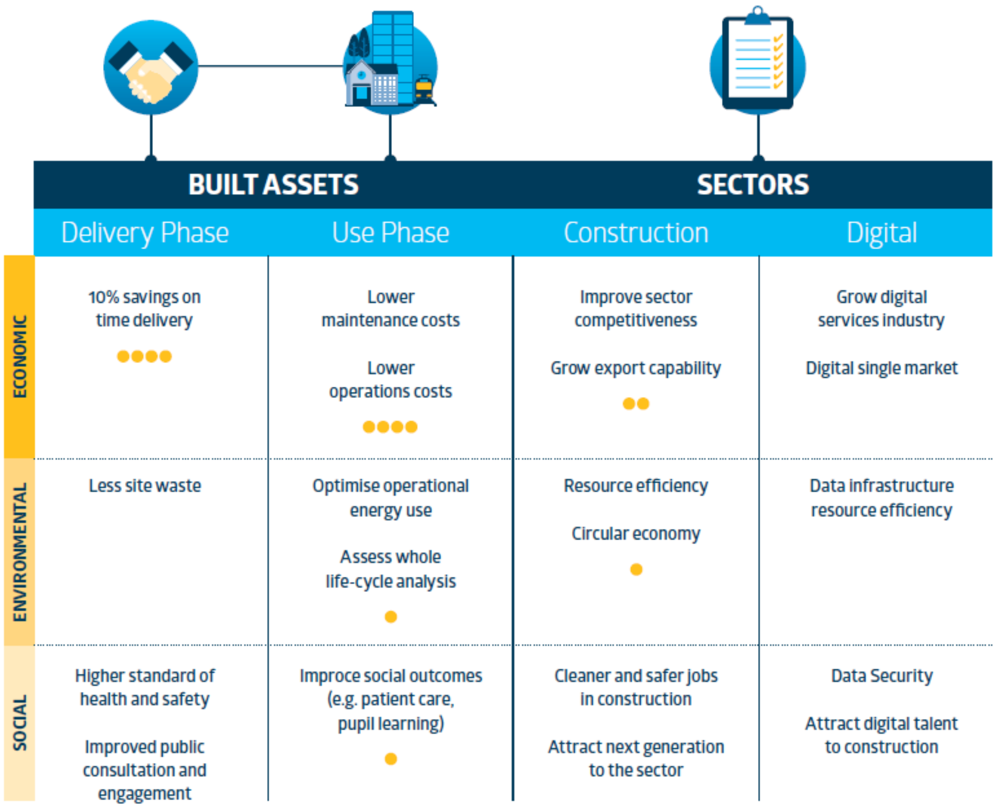


Figure 9: Advantages of BIM implementation (EU BIM, 2017)

Also, professional areas of activity have reduced their boundaries because of the characteristic of a generative model, which depends on constructive parameters in the early stage of conception (Maravilha De Azevedo, 2009). The conceptual and practical advantages and consequences of BIM provide the frame for a critical analysis of architectural design and design process on how they are fundamentally conceived, on the other hand, due to PD premises for design processes, fabrication and construction are increasingly changing the relationship between architecture and its means of production (Ambrose, 2007). Therefore, many problems, especially at early design stages, are due to the difficulty of collaborative effort to develop parametric models inside a BIM environment (Chaszar & Joyce, 2016).

3.3.2 Parametric BIM workflow as an approach

In recent years, architectural practice has had a great influence by computational design. Large and small design practices are investing in new computational capabilities in order to customize their process and follow new and innovative design agendas (Autodesk, 2019). Considering, that a human design activity is modification of goals during exploration, meaning weights of performance, trade off opportunity, and multidisciplinary/ multi-objective optimizations (Chaszar & Joyce, 2016). A successful design assistant provides the designer valuable information and acknowledges the nature of the design process. Therefore, it is important for developers to consider a tool that considers flexibility regarding the preliminary concept design phase (PCD) (Hugo & Charles, 2011). The simultaneous combination of design exploration and construction information in one model are in an increasing demand, where the couple approaches that link graph-based systems and BIM systems is preferable. This bridging approach is referred as parametric BIM workflows (Janssen, 2014).

Computational optimization and design exploration processes are outstanding benefits of this workflow approach thanks to its iteration through countless design variations by digitally updating the parameters models. Is about a hierarchy process of geometrical and mathematical relations which create a model that can be manipulated by changing parameters, where process is automatize and repeat with no possibilities of human error (Nezamaldin, 2019)**.** For more complex non-repetitive configurations, remodeling becomes time consuming and error prone, nevertheless this does not require a fully automated process, but a workflow that minimize the effort required to remodel the design, regardless of the design stage and software (Jassen, 2016). For instances, the integration of performance aspects into the design faster and collaboration between architects, consultant and contractors is achieved through generative master models rapid iterations (Wortmann & Tunçer, 2017).

4 State of Art

4.1 BIM:

4.1.1 Definition

Is mistaken to consider Building Information Modeling as a software, a 3D model or a system, this confusion and divergence in the AEC industry is a barrier for its successful implementation (Van Beusekom, Sarwarzadeh, Sinke, Sturm, & Zegger, 2018). Is far more powerful to position BIM as a way of thinking and not limiting it as a tool (Ambrose, 2007). BIM methodology present a more valuable approach where its advantage is to design coherence and productivity. It is a complete digital building model from the core database, where 3D models, drawings, sections and also quantity estimations and simulations are derived (Boeykens, 2016). Therefore, in the BIM environment a 3D model is an advance computer technology and not merely a static representation of a project. It can manage information for (Barazzetti, 2016):

- Automatic generation of drawings (sections, plans, etc.) and reports
- Design analysis
- Schedule simulation
- Thermal and structural simulation
- Facilities management

The US National Building Information Model Standard Project Committee on its 2018 standard describe BIM definition as the following:

“Building Information Modelling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition.”

BIM begins with the virtual construction (simulation) of the whole, for afterwards been viewed as a series synthetic assemblies of constituent components. This design process prioritize the contextual construction of a formal/spatial systemic intelligent simulation instead of an abstract representation or fragmented conventions of communication (Ambrose, 2007).

4.1.2 Context

4.1.2.1 History

Since the orthographic and perspective projection in the fifteenth and sixteenth centuries, nothing has intended to shift the design process and reshape the production of architectural ideas and objects like BIM (Ambrose, 2007). Although the concept of BIM appears at the end of the 1970's, the term Building Information Modelling did not become popular until the begging of the XXI century. It evolves from "Building Model" in the mid 1980 published by Simon Ruffle, to further on be called "Building Information Model" by GA van Nederveen and F. P. Tolman in 1992. (Nezamaldin, 2019).

During mid-1990s, the AEC sector started with the arrival of sophisticated CAD systems because of the new wave of ICT developments. At the time, it was possible to enrich the 3D models of buildings and structure adding, not just vector data but complementary information such as physical characteristics, unit costs, quantity take offs, starting the methodology know nowadays as BIM. Although, it was embraced in academia since then, the appearance of BIM in real world projects began in some pilot projects and lately in some major projects only after the year 2000. Nowadays, it remains a rare approach in projects (Grilo & Jardim-Goncalves, 2010).

4.1.2.2 Industry 4.0

Although, Industry 4.0 is a well-known term in AEC industry and academia, its research is relatively a new topic. The name Industry 4.0 comes from the fourth industrial revolution which is being leaded by the Internet of Things (IoT) and the Internet of Services integrated with manufacturing. It looks forward to global connectivity and intelligent control of machinery, factories, and warehousing facilities for all industrial businesses through cyber-physical systems sharing information that triggers actions. (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019)

Strategies of implementation has been proposed since it create value that transform the overall business strategy in the construction industry. Industry 4.0 is expected to improve the quality and productivity of construction and attract more investors thanks to its capability to automate both design and manufacturing processes and the possibility of managing a heterogenous and substantial amount of information (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019).

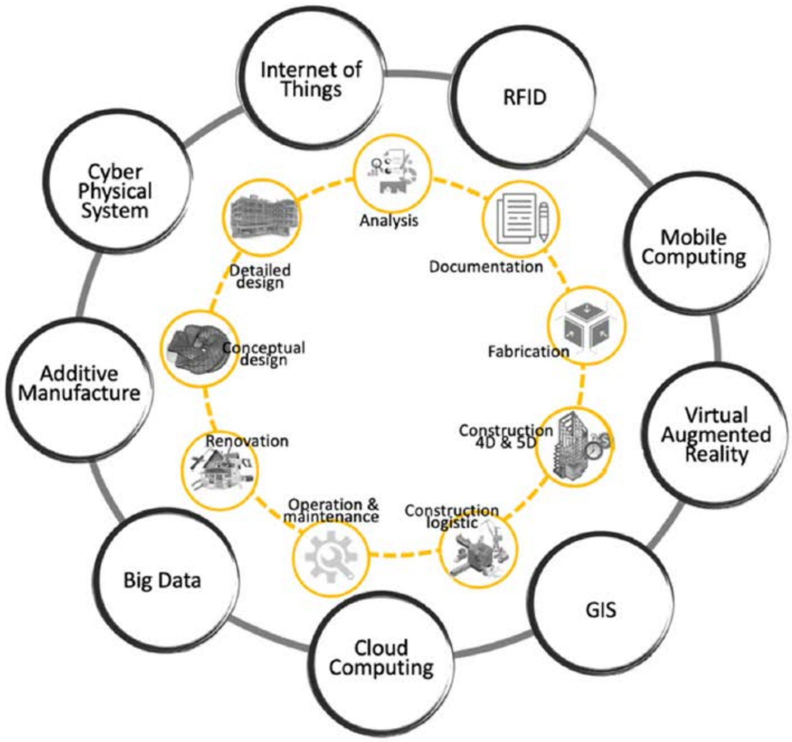


Figure 10: Concept fo technologies in Industry 4.0 with BIM as its core strcuture with collaboration and an autonomous synchonization system (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019).

4.1.2.3 Role of BIM in the industry 4.0

Digitalization moment in the construction sector is introduce by Building Information Modelling (BIM), in fact, wider use of technology, digital processes, automation and higher skilled workers are contributing to the industry's economic, social and environmental future. As a digital form of construction and asset operations, BIM coupled technology, process improvements and digital information to radically improve client and project outcomes and asset operations. It plays a strategical role on decision making improvements for both buildings and public infrastructure across the whole lifecycle. And, it can also be applied to new built projects, and renovation, refurbishment and maintenance of the built environment. (EU BIM, 2017)

Furthermore, the integration of BIM into the Information Technologies (IT) environment support the transition from a current "react to event" practice to "predict the event" practic. A well modeled BIM is a virtual model working as closer as possible to reality by containing all the necessary information for the construction process. For example, Robotic technologies have been integrated in the construction industry as construction automation technologies to create elements of buildings, building components and building furniture (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019).

4.1.2.4 BIM in the AEC industry

The application of BIM in the AEC industry represents a large price. Regarding Europe, a wider adoption of BIM can deliver 10% saving to the to the construction sector, meaning €130 billion would be saved from an €1.3 trillion industry market, accelerating the growth and competitiveness of the construction sector, especially its small and medium enterprises (SMEs). Furthermore, economic impact seems small compared with the potential social and environmental benefits. (EU BIM, 2017)

Indeed, BIM appear to be an efficient methodology which combines conception, coordination, simulation, planification and operation for building and infrastructure of any size and complexity. This virtual environment in the AEC industry enables designers, engineers, contractors, and suppliers to integrate complex components, cutting waste and reducing the risk of errors. Furthermore, during operation it provides customer with real-time information about available services and maintainers with accurate assessments of the conditions of assets (Bew, 2015).

In addition, BIM six functionalities component are considered as: (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019):

- Team communication and integration
- Parametric modelling and visualization
- Building performance analysis and simulation
- Automatic document generation
- Improved building lifecycle management
- Software interoperability with other applications

4.1.2.5 BIM by countries

AEC industry has adopted worldwide BIM as a strategy to develop the construction sector, having different actions and results in many countries:

- USA: Industry-wide adoption of BIM surged from 28% in 2007 to 71% in 2012, emphasizing the importance and the speed of the BIM's spreading effect. (McGraw-Hill Construction, 2012)
- China: Chinese construction industry promote BIM and other new technologies as a long-term strategy towards digital construction. A largely increase in the transaction volume of technology enterprises is expected. Research and development of intelligent and green buildings and building automation systems is the focus of this technical upgrading. Building information Modeling/Management is developing at an intense pace, hence a new leap in technology, management, system and mechanism is expected. Furthermore, regarding long term strategy towards digital construction, internal digital construction technologies are being develop by the top tier construction companies. Nevertheless, companies are partnering with tech companies to develop these new technologies (Deloitte, 2018).
- Brazil: Is one of the leading countries regarding its application, Building Information Modelling is a promising market. (Deloitte, 2018)
- Portugal: Digitalization in the AEC industry sector is driven by designers adoption of technologies and not necessarily acquisitions. The sector aims to develop its technological maturity by adopting strategic technologies, (e.g. BIM, 3D printing), digitalizing the supply chain and procurement channels (use of collaborative tools such as IoT), and promoting innovation management and R&D, especially in SME's (Deloitte, 2018)

4.1.3 Dimensions

BIM presents as one of its features the option to expand design beyond the 3D world. The first added dimension is time management as 4D, with schedules organization and project phasing. Followed by 5D as cost estimation, achieving real time modeling and cost planning, engineering optimization, and prefabricated solution. 6D is the sustainability dimension of a project, where BIM allows to integrate environmental analysis and optimization to achieve a better building performance. Finally, the 7D regards Facility management applications, collecting operation and maintenance information to improve life cycle BIM strategies.

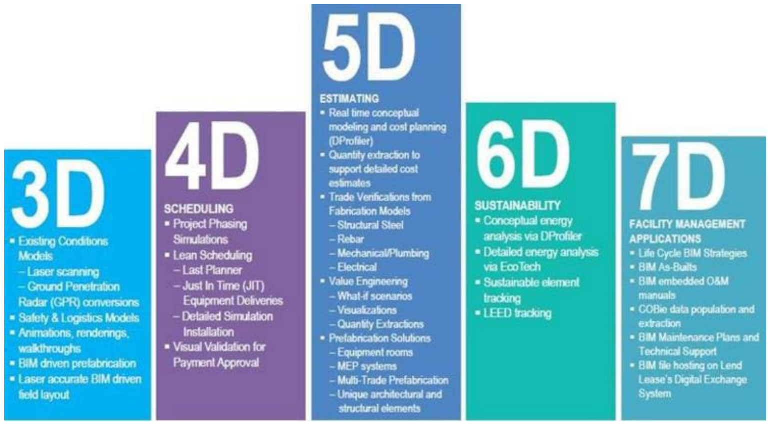


Figure 11: Multidimensional possibilities of BIM (BIMestimate, 2020)

4.1.4 Levels of information in BIM

The management of diverse and extensible amount of data is a quality of BIM methodology. Since information is added gradually during the design phases of a project, not all projects require the same amount of information or detail. As a result, countries have stablished different work plans and standards as road maps to organized information and design deliverables stages during the project development. These standards define the collaborative methods for transfer information, the stage of development of the project, or the information available of the objects in the model.

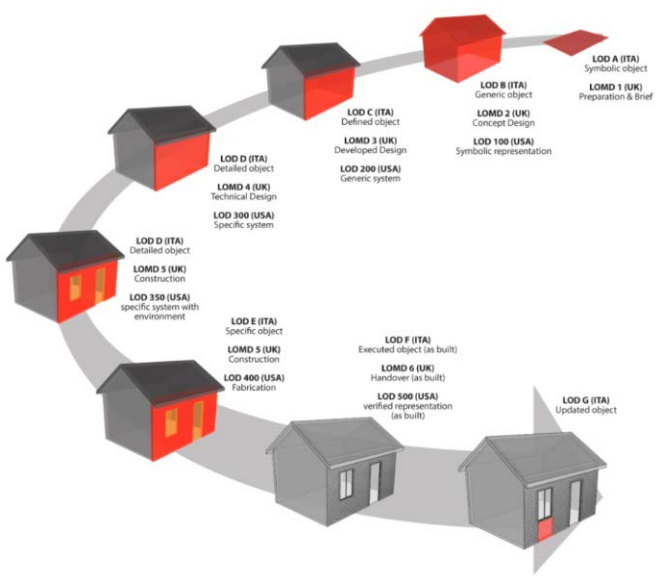


Figure 12: BIM levels of information (Rebim, 2020)

4.1.4.1 *BIM maturity levels*

BIM maturity levels acknowledge the impact of both data and process management. It defines the collaboration of BIM models with the project and with all the asset information, documentation, and data. (RIBA, 2012).

Level 0: BIM is limited to the use of 2D CAD files for production information. Unfortunately, most design practices have used it like this for many years.

Level 1: Recognize the increasing use of both 2D and 3D information on projects. It can be considered as 'Lonely BIM' since, in the case of architects, 3D software has been used just as a conceptual design tool during the early project stages and a visualization for client presentations once the project is finished.

Level 2: Production of 3D information models by all key members of the Integrated Team. Unfortunately, its interoperability is limited because, these models does not co-exist in a single model.

Level 3: Challenging the harnessing of information in a single model is even more important than just coordinating a collaborative use of information in one model.

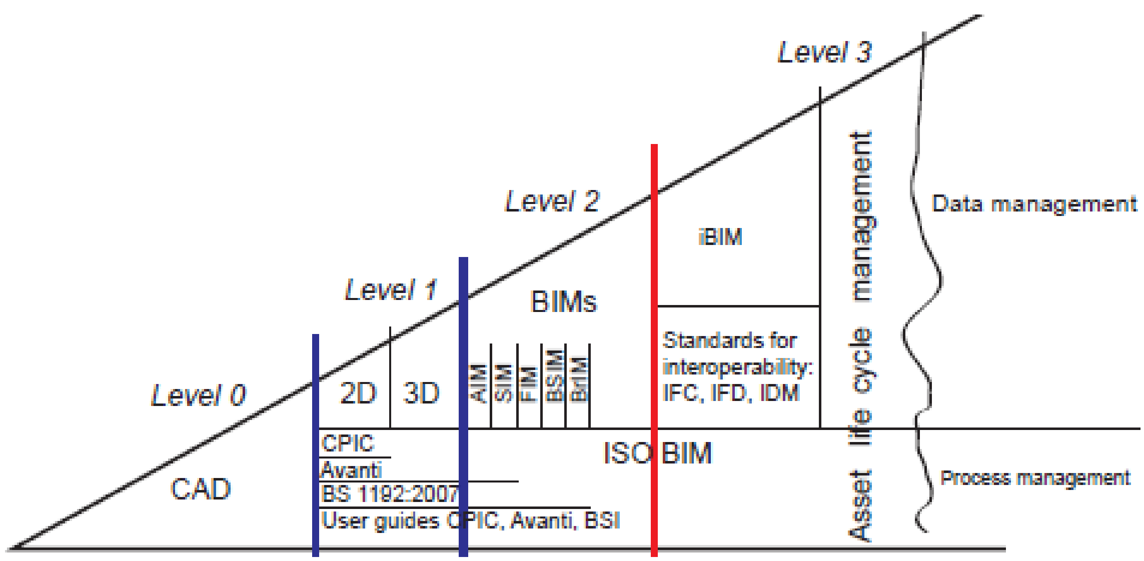


Figure 13: BIM maturity diagram (Bew, 2015).

4.1.4.2 RIBA Plan of work

A plan of work provides the design project team a road map, defining goals by several stages. It supports the design approach and builds consistency throughout the design development in a highly complex environment and largely diverse data processes. In addition, these stages provide essential guidance to clients that faces their first building project. (RIBA, 2020)

Stage 0: Finds a solution to fulfill client's requirements. An appropriate solution may not be building, so an open mind is required.

Stage 1: Develops the brief for the design process and making sure everything needed is in place for the following stage. This includes adjusting the brief on the site.

Stage 2: Arrives to the right design concept making sure the building feels and looks like the client's vision, brief and budget. It is important to ensure that tasks to undertake this mission are clear. Although going to much in detail can swift the design team's effort away from setting the best strategy, if details are insufficient next stage becomes inefficient.

Stage 3: Coordinates spatially the design before the focus turns into detailing the information for manufacturing and constructing the building. At the end of this stage, the information needs to be sufficiently coordinated to avoid iteration further on, and that planning application has the best possible information.

Stage 4: Develops the information and details required to manufacture and construct the building. Information is required by the design teams and the specialist subcontractors employed by the contractor.

Stage 5: Building manufacturing and construction.

Stage 6: Completing the Building Contract, once the building has been built this stage focus on closing out any defects and completing the tasks required.

Stage 7: Period when the building is used util the building reaches the end of its life cycle.

4.1.4.3 Levels of development LOD (UNI 111337)

Levels of development have largely been accepted in European standards to describe object's amount of information inside BIM to incorporate the multidisciplinary skills of designers and engineers involved in the construction process. Concerning the Italian Standard UNI 11337:2017 LOD are defined by the object geometry (LOG) and the level of information (LOI). They are organized in alphabetical order, with A as the less develop category and G as the final one.

LOD A: Symbolic object, line composed object represented in 2D.

LOD B: Generic object, approximative volume on 3D.

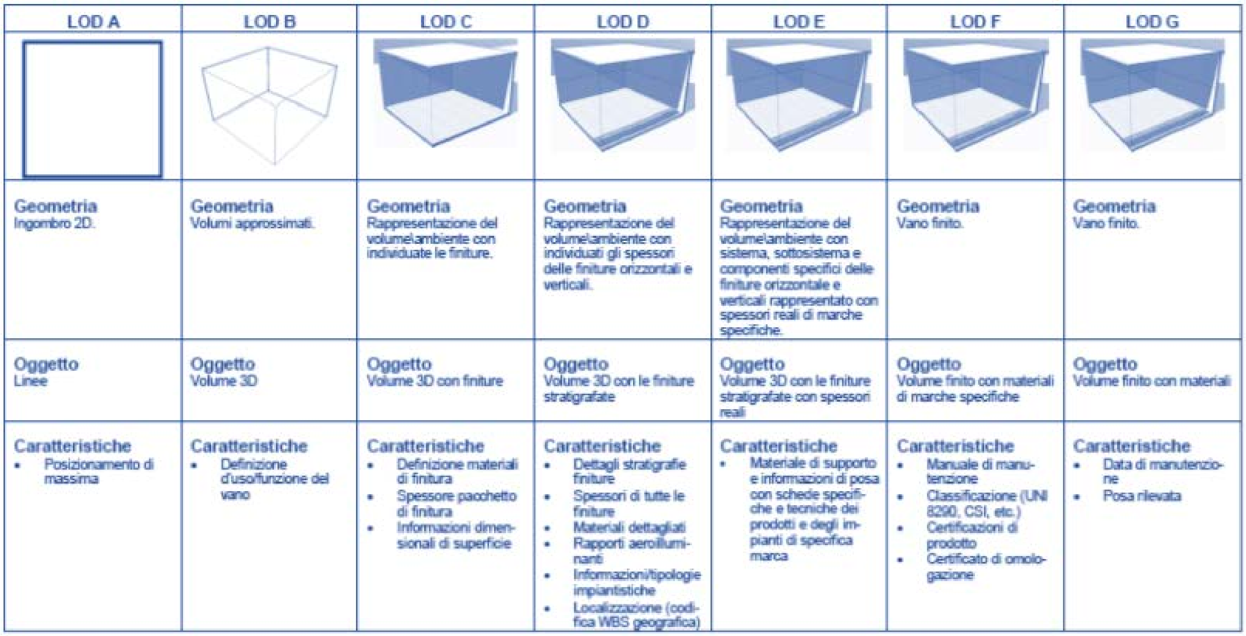
LOD C: Defined object, finishing is added to 3D objects.

LOD D: Detailed object, stratigraphy is defined in 3D objects.

LOD E: Specific object, layer specific information is added in the stratigraphy definition.

LOD F: Executive object, final volume with material suppliers for construction.

LOD G: Updated object, final object.



LOD A	LOD B	LOD C	LOD D	LOD E	LOD F	LOD G
Geometria Ingombro 2D.	**Geometria** Volumi approssimati.	**Geometria** Rappresentazione del volume\ambiente con individuate le finiture.	**Geometria** Rappresentazione del volume\ambiente con individuati gli spessori delle finiture orizzontali e verticali.	**Geometria** Rappresentazione del volume\ambiente con sistema, sottosistema e componenti specifici delle finiture orizzontale e verticali rappresentato con spessori reali di marche specifiche.	**Geometria** Vano finito.	**Geometria** Vano finito.
Oggetto Linee	**Oggetto** Volume 3D	**Oggetto** Volume 3D con finiture	**Oggetto** Volume 3D con le finiture stratigrafate	**Oggetto** Volume 3D con le finiture stratigrafate con spessori reali	**Oggetto** Volume finito con materiali di marche specifiche	**Oggetto** Volume finito con materiali
Caratteristiche • Posizionamento di massima	**Caratteristiche** • Definizione d'uso/funzione del vano	**Caratteristiche** • Definizione materiali di finitura • Spessore pacchetto di finitura • Informazioni dimensionali di superficie	**Caratteristiche** • Dettagli stratigrafie finiture • Spessori di tutte le finiture • Materiali dettagliati • Rapporti aeroilluminanti • Informazioni/tipologie impiantistiche • Localizzazione (codifica WBS geografica)	**Caratteristiche** • Materiale di supporto e informazioni di posa con schede specifiche e tecniche dei prodotti e degli impianti di specifica marca	**Caratteristiche** • Manuale di manutenzione • Classificazione (UNI 8290, CSI, etc.) • Certificazioni di prodotto • Certificato di omologazione	**Caratteristiche** • Data di manutenzione • Posa rilevata

Figure 14: BIM Levels of detail (UNI 11337:2017)

4.1.5 Interoperability

One of the main benefits of BIM at level 3 maturity is the possibility to import other designers 3D BIM files into their own analysis software, and obtain the information for costing, programming or other purposes (RIBA, 2012), once the direct flow between the actors involve in the design process the BIM models present significant improvement by minimized time-consuming manual operations reducing possible errors, and Rationalization the way information is transferred from the consultant to the contractor (Hansen Gøran & Bjørn Smith, 2017).

Unfortunately, if the output model of one software package is not compatible with another, these opportunities are limited (RIBA, 2012). Therefore, in order to achieve interoperability BIM environment have achieve two different approaches:

- Create exportable standard elements with consistent bases that minimize or prevent information losses, where IFC and XML languages have been defined to allow data transcript across programs (Maravilha De Azevedo, 2009).
- Extend its base functionality. Nowadays, most software's provides various possibilities to customize and extend its base functionality with visual scripting environment, macros or application programming interface (API) (Schwerdtfeger & Zaha, 2018).

4.1.5.1 IFC

One of BIM existing problems lies in the exchange of information between agents involve in the construction project. To simplify work process management in BIM an effort to make collaboration easier and flexible is developing with many standards and methodologies being created. One of the solutions for an open exchange information between applications is Industry Foundation Classes (IFC) standard by building SMART International. It presents for open BIM the best open standard an neutral alternative used to facilitate this exchange (Alfabuild ; Санджиев, et al., 2018). In fact, among all the construction-related software tools is meant to be the universal mean of data sharing (Wojciech Adamus, 2013). Using shared and open specifications allows IFC to not be controlled by a single software vendor. Furthermore, since software development is shared and not restricted greater interoperability between software platforms is achieve (RIBA, 2012).

Advantages for model in IFC format are (Alfabuild ; Санджиев, et al., 2018):

- Open interoperability criteria.
- intellectual property is preserved in the native model information.
- Simplicity to integrate and combine the returned model with other models for the constructor, regardless of the program (ex: Revit, Navisworks, etc.)

IFC is an open data schema; which within the digital building model; describes all features of the building in its whole lifecycle. IFC specification is written using the EXPRESS data definition language (ISO 10303-11). And it includes detailed information about building geometry, structure, materials, products, processes, actors involved, etc. Regarding standards, Version 2x4 of IFC is to be adopted as international standard ISO 16793 (Wojciech Adamus, 2013).

IFC structure is complex and contain a large variety type of objects, which difficult the implementation of it in the software applications. Some initiatives have been taken to expand its use. For example, the model view definitions MVDs which are parts of the whole IFC model data, this includes specific sets of data to perform a particular action, like structural or thermal analysis. Nevertheless, IFC model ensures flexibility of object's property definition thanks to its oriented and hierarchical structure. Hence, building's element property type inherits the attributes of the IfcObject class by assigning any available property set (IfcPropertySet object)

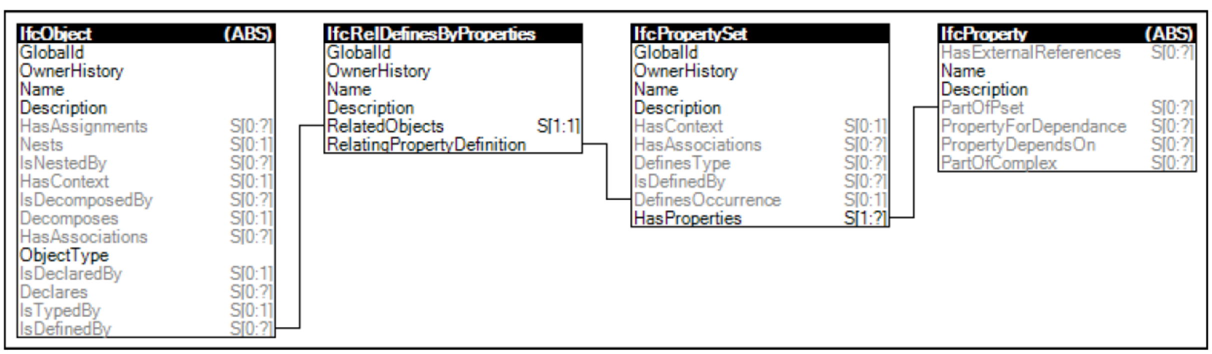


Figure 15: Schema of binding a property ti the objects in IFC (Wojciech Adamus, 2013).

Additionally, property set goes beyond describing items for construction and can also declare the environmental impact of and item. IFC 2x4 version specification includes property sets especially developed for environmental analysis (consistent with ISO EN 15804), like, environmental impact indicators describing values for a given "functional unit" of the element, and environmental impact values taking environmental impact values of element as an entity. Currently IFC is expanding its support in a wide variety of BIM based design applications, including architecture, structural engineering, mechanics, etc.

4.1.5.2 gbMXL

Green Building XML (gbXML) open data was first released in 2000, and its main purpose is transferring information of building from CAD application to environmental or energy analysis software. Its schema includes mainly building geometry with data for operational energy consumption analysis, HVAC installations, and thermal properties of construction materials.

- (Wojciech Adamus, 2013)

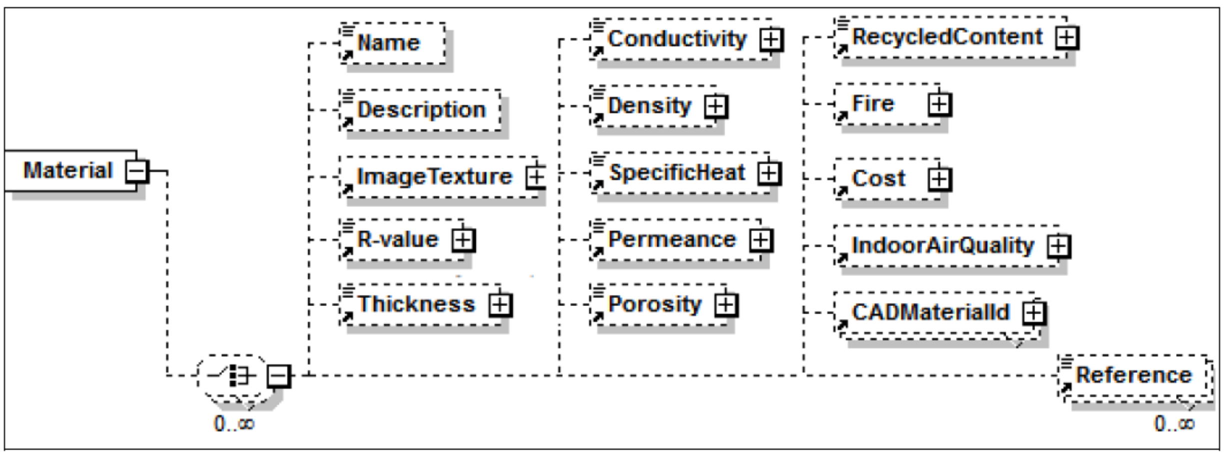


Figure 16: gbXML material properties schema (Wojciech ADAMUS, 2013).

gbXML is based on a simple and widely known language, simple XML, and it is considered as relatively easy software to implement. Nowadays gbXML is commonly used to transfer data in BIM-based energy simulation software like Ecotect, Green Building Studio or IES Virtual Environment.

On the other hand, the limited information about construction products that influence the operational energy use and the lack of specific data about construction materials necessary in the context of environmental analysis to perform the building‘s life-cycle assessment (LCA) are the main limitation of gbXML schema. In addition, gbXML is not present as an official standard. This allows the data schema to have a rapid evolution but limits the participation of various market representatives in the process of the schema development (Wojciech Adamus, 2013).

4.1.5.3 API

Application Programming Interface (API) is based on high level interface by programming source code, it includes the specific combination for programming languages routines, data structure, classes and variables. Moreover, API specifies the different interaction of software components between each other involving access to database, hard drive, disc drive, video card, etc. Concerning visual languages as C#, API helps to specify the interactions and behavior of objects class definitions. In addition, API is also important in web development, since it can dynamically share contents and data between communities and applications using an open architecture in web programming.

ICT research on API applications are not new, nevertheless new contributions can still be made by introducing suitable methodologies upgrading the research tasks. One of the major challenges of the current internet evolution of cloud computing data is the lack of standardized APIs, currently been discussed to explore distributed synchronous and asynchronous exchange/management of BIM. Data exchange standards or API level customization for interoperability supports the early academic research on data integration and management represented by BIM. API can be used to expand existing programs or modify the semantic relation of data structures at runtime. According to research, this is particularly useful to use existing BIM platforms with similar API code patching capabilities as proving grounds for prototyping purposes. Moreover, BIM extensions are communicating with models in existing BIM platforms relying to API programming technology. BIM extensions are considered the new software systems that add functionality to the already in use BIM tool through API based add on applications, indeed API implementation can capture attributes, geometry and spatial information of elements feature on BIM to improve the interface of future activities in construction management (Oti, Tizani, Abanda, Jaly-Zada, & Tah, 2016).

4.1.6 Revit

Founded in 1997 like Charles River Software and rename Revit Technology Corporation in 2000 was the first Revit software to line up with the BIM requirements. In 2002 it has acquired by the Autodesk group (Nezamaldin, 2019).

One of the main features of Revit is the ability to transfer parameters between the different sublevels of information of each element. All objects use the same logic, they are divided in four parts with the same hierarchy elements. Elements are organized in categories then distributed in families, which has at least one type defined, finally this type has a set of properties called instances, these final values are the same for all identical model objects. Finally, in order to label the model elements certain information has to be carried from one sublevel to another (Nezamaldin, 2019).

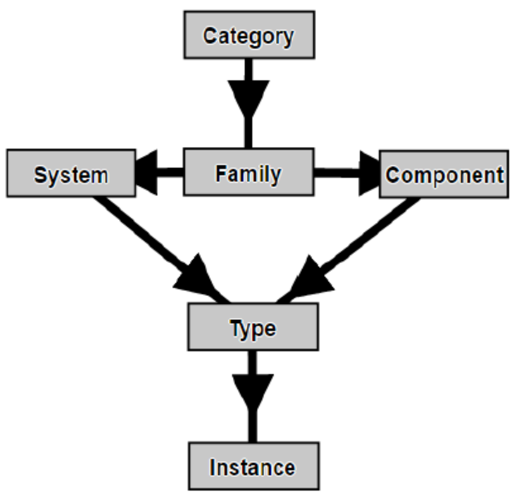


Figure 17: The Revit hierarchy (Wojciech ADAMUS, 2013).

Category: They are already defined and cannot be created or deleted, all elements belong to a category.

Family: All elements inside a category belong to a family. Families of Revit are divided into three:

- System Families: Predefined inside a Revit project. Ex: walls, floors, roofs and ceilings.
- Loadable Families: Built separately from the project. Ex: desks, parking space and trees.
- In-Place Families: Built inside the project.

Type: A specific element of the family that is defined by parameters, like size and different measures.

Instance: Physical element created in the project**.** It is the lowest of the hierarchy and therefore exists in various forms.

Another important characteristic of Revit is its object-oriented programming, meaning different objects with different attributes. The four principles of object-oriented programming area:

Encapsulation: In the case of walls, floor and roof objects communicate with each other when connected. Nevertheless, they keep the object form inside its families, meaning that other objects do not have access to this element form. For example, a wall property can only be change from inside the wall category itself and not for a roof.

Abstraction: Works as an extension of encapsulation since complex scripts and codes are simplified, the user only needs to see the easy way of modifying parameters. For example, even if the programming script can be complex a wall height modification is straight forward for a user.

Inheritance: Including some slight changes for different objects, the same main script can be used to modify an element's parameter with the same attributes as height, area, length. The hierarchy between instances, types, families and category, turns the highest one in the parent class and the lowest one the children class of the parents.

Polymorphism: Defines the way to use a family like a parent class keeping for each child its own methods. Is possible to use the same method for different objects with the same access to its properties.

4.2 Parametric Design

4.2.1 Definition

In few words, parametric design is creating a geometry where its parameter depends on each other (Nezamaldin, 2019). Parametric design translate design as a constrained collection of schemata; therefore, designer works is the definition of specialized schema and its constraints. A parametric model emphasis in representation that creates a model as an infinite set of instances, each model's independent variable is determined by a particular definition of values (Aish & Woodbury, 2005).

Normally, the objective of an initial parametric model is to compilate all the known parameters and associations of the proposed design solution and setup a model, where in order to explore different options the geometry and resultant behavior may be control by changing variables (Nazim & Joyce).

Parametrical design is closely related to computational design since it is link to innovative problem solving to deliver design solutions by means of a powerful and novel computational algorithms to automate, simulate, script, parameterize, and generate design solutions (Autodesk, 2019). An algorithm is the procedure to perform a particular task through a list of basic and well-defined inputs. They find the solution to a problem, splitting it into a set of simple steps that can be easily computed. (Arturo Tedeschi, 2014). In fact, algorithms can define any geometry, by writing a rough mathematical definition, imputing variables as parameters and translating it into programming language. Parameters normally include, numbers, distances, angles or functions like, equal, if, then, etc.

One powerful feature of parametric modeling methodology through the design process is creation and edition of geometry with an automatic update on its final shape when parameters are change. Users of parametric systems develops relationship between objects, coding them into a node or graph, this gives the flexibility to develop new relationships when the system does not support them since the beginning (Aish & Woodbury, 2005). Parametric modelling can be divided into four main types depending on how theses modelling types support iteration (Janssen, 2014):

- **Object modelling:** does not allow for any iteration.
- **Associate modelling:** allows for single-operation iteration.
- **Dataflow modelling:** allows for implicit multi-operation iteration.
- **Procedural modelling:** allows for explicit multi-operation iteration.

4.2.1.1 Parametric-associative modelling

Associative design is based on parametric design methodology where associative geometry is used. Relationship between objects are clearly described and established an interdependence between various objects, therefore variations can easily be managed using these attributes. While maintaining conditions of the topological relationship the assignment of different values can generate multiple variations (Oxman, 2006). Different from conventional CAD system, where geometric elements are manipulated individually, parametric models can be modify by parametric associations by changing the geometry generating components, input parameter or association between components (Nazim & Joyce). Regarding a building design parametrically, parameters have limits or boundaries, when these boundaries change the parameters assigned to connected elements allows them to be automatically adjusted and changed. For example, if a classroom is design with a parameterized furniture layout regarding the classroom size, then the parametric design would automatically adjust the seating layout based on the parameters assigned to the seats and the surface of the class. (RIBA, 2012)

Parametric associative modelling characteristics are (Chaszar & Joyce, 2016):

- Discrete units of logic, often geometric, encapsulated in functions components or nodes.
- Definitions of base input data primarily geometric definitions from external sources or parameters, usually with a specific range and number set such as real or integer.
- Associations between nodes whereby the outputs of one or more nodes can be fed into the inputs of the subsequent nodes.
- Feed forward only, thus no cyclical dependencies.
- Duplication results in repeated function calls and repeated output data, often called replication.
- System regeneration based on changes to nodes and propagation and re-evaluation downstream of the graph.

4.2.1.2 Genetic programming in Generative Design

Generative design assessment is base in the implementation of smart building, by including knowledge in the generative end of design. Here, rules are applied and evaluated or inform while allowing the system to evolve into a goal-oriented design model. This reduces the need of post design assessment or data migration to external applications or data structures outside the first BIM application (Hugo & Charles, 2011).

There are several performance criteria that must be evaluated during the design process, such as cost estimation, accessibility, energy, structure, durability, acoustics, transport, planning and many others. Nevertheless, the interpretation of these results depends of the expertise of engineer or designers and decisions are often postpone until late in the design process (Jassen, 2016). Traditional practices and workflows have seen architects and design consultants generating and maintaining independent models of the same design project. Although, to build the basis for the next loop of design in the AEC industry an iterative process of assessing and evaluation is necessary (Mirtschin, 2010).

Once the model has been parametrized and all the information for the design goal is fully compiled, "generativity" is introduced by second order logic where changes are deliberated and have some degree of goal-orientation. This is call genetic programming method, and its characteristics are (Chaszar & Joyce, 2016):

- Target system to evolve (typically a programming language or mathematical schema).
- Higher level abstraction of the language into a simple but generic form.
- Ability to encode the abstraction into a single string or definition.
- Methods to combine and partially modify these definition strings whilst still being valid with respect to the system.
- Formulation of target behavior for the system as method to quantitatively measure definitions.
- Typically, the use of Genetic Algorithm to evolve the system by stochastically trialing automatically generated functions.
- Stopping criteria after which best performing definition is returned.

4.2.2 Context

4.2.2.1 History

The faculty of architects to predict design outcomes has been the organization of ideas, resources and spaces on drawings. As methods of representation evolve, new styles arrive from perspective in Renaissance to projective geometry in Modernism. Nevertheless, the traditional drawing has always been an additive process, achieving complexity through the addition and overlap of independent signs on paper. Drawing is not a "smart" medium, but a code base on standards a convention, similar to traditional Computer Assisted Design (CAD). CAD method is the translation of the additive logic within the digital realm, designers determine the overall consistency by adding digital signs or geometric primitives on digital space and controlling CAD layers. On the other hand, in Parametrical Design every creative act is represented by a geometric alphabet built on actions that link ideas to digital constructions (Arturo Tedeschi, 2014).

In 1939, the Italian architect Luigi Moretti invented the definite for "Parametric Architecture". Viewing angles and economic feasibility were the input design parameters in Moretti's design, hence the final shape was generated by calculating pseudo isocurves, in order to optimize views from every position in the stadium. Furthermore, Moretti's research was a collaborative work with Bruno De Finetti with whom he founded the Institute for Mathematical Research in Architecture (I.R.M.O.U.).

In Moretti's words:

"The parameters and their interrelationships become [...] the code of the new architectural language, the "structure" in the original sense of the word [...]. The setting of parameters and their relation must be supported by the techniques and tools offered by the most current sciences, in particular by logics, mathematics [...] and computers. Computers give the possibility to express parameters and their relations through a set of (self-correcting) routines". NOTE 1 F. Bucci and M. Mulazzani, Luigi Moretti opere e scritti (Milano: Electa, 2006), 204-208. In (Arturo Tedeschi, 2014)

After Moretti, in 1953 Ivan Sutherland, an American computer scientist, Ivan Sutherland, created the first interactive Computer Aided Design (CAD) called Sketchpad, it has defined as a "Machine Graphical Communication System". Sketchpad is considered as one of the most influential computer programs, it was designed to draw basic primitives such as point lines and arcs using a light-pen input. Moreover, it was based on an advance associative logic called atomic constraint allowing combination between objects to generate relationship, overcoming the limits of the additive logic of traditional drawings. Nevertheless, the associative capabilities introduced by Sketchpad were not integrated in the commercially successful Autocad (1982).

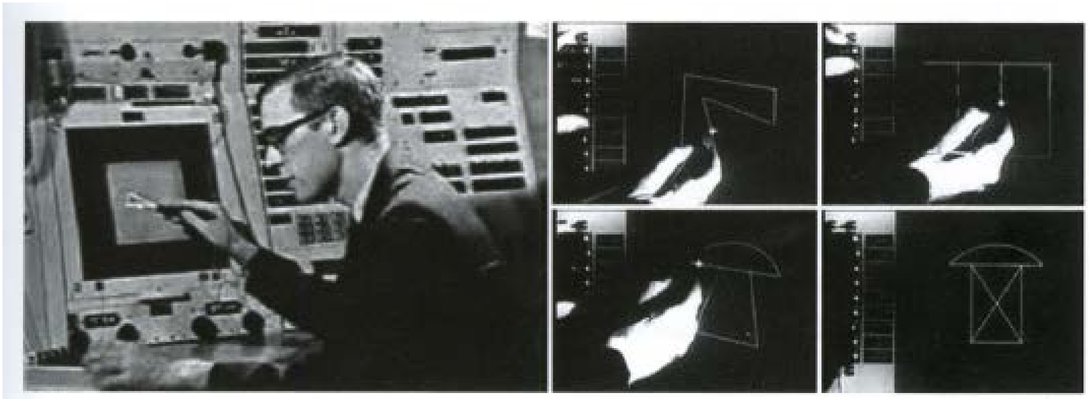


Figure 18: Ivan Suntherland on MIT Lincoln Labs' TX-2 computer (1963) (Arturo Tedeschi, 2014).

The next step into parametric design will come from avant-garde architecture in the 60’s where designers push drawing limits using several methods of representation driving them to a generative process. For example, Eisenman's diagram for House IV impressed with an entire sequence of geometric operations that led to the final object.

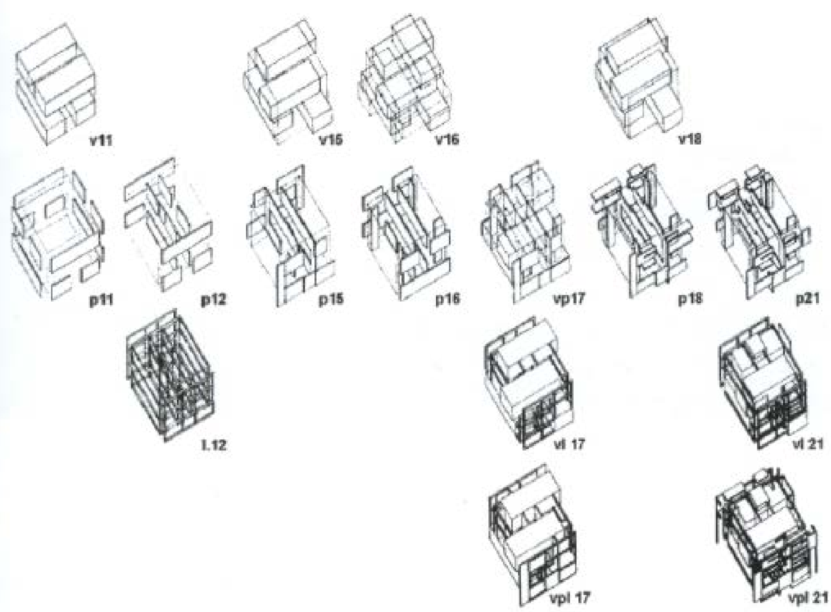


FIGURE 0.4
Peter Eisenman, House IV, Falls Village, Connecticut, 1971.

Figure 19: Peter Eisenman, House IV, Falls Village, Connecticut, 1971 (Arturo Tedeschi, 2014).

Furthermore, physical form finding replaced drawing as a medium to investigate structural optimization, devices demonstrated how dynamic forces shaped new self-optimized architectural forms. Structural optimization through physical modelling was mono-parametric since it was; above all; gravity based, but pointed the road for a multiparametric form finding interacting with heterogeneous data like geometry, dynamic forces, environment, social data.

Figure 20: Forces and Forms are correlated - Heinz Isler, Service Station in Deitingen (Arturo Tedeschi, 2014).

Nowadays, in multiple design fields including architecture, an imagistic innovations by the dramatic form-generative potential of parametric design has been the driving force in the expansion of a digital design culture (Oxman, 2006).

4.2.2.2 Importance in AEC industry

In real construction industry 3D modeling of projects is mainly a manual procedure meaning it is time consuming and labor intensive (Barazzetti, 2016). Therefore, industry attention is developing systems where the design object is represented parametrically allowing the rapid representation of changes at design and structure (Aish & Woodbury, 2005). In order to let a model highlight the singularity and adaptability of the constituent elements of the building, parametrization or modeling based on programming scripts is used (Maravilha De Azevedo, 2009). Through PD objects are generated, represented and fabricated with a greater level of detail and singularity (Wortmann & Tunçer, 2017). The appearance of this new technology and interest enhance the exploration in a new design space where architecture and its supporting technologies of design and fabrication experience a co-development and rapid change (Aish & Woodbury, 2005). This result in the reexamination inside the AEC industry of current design theories and methodologies to explain and guide future research on representation, generation, and interaction. (Oxman, 2006)

Digital means are now assisting the design, documentation, fabrication, and assembly of buildings. Withing this new digitally mediated framework, architectural design in the digital era is characterized by high levels of complexity; this allows the introduction of sensitive response into the design palette to contextual demands such as site, program, and expressive intention (Mitchell, 2005). As a matter of fact, the evolution on digital fabrication and digital architecture tools makes complex constructions more economical and buildable. (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019)

Parametric design also have a huge potential for cost effective design and has become an essential part of early stage design process, and most recently into later design stages (Hansen Gøran & Bjørn Smith, 2017). Contrary to non-parametric design where activities are relatively linear and all changes are time consuming, in PD simple parametric changes are fast and synchronized. The benefits of this interface is a powerful tool for designers in computational optimization and exploration process by iteration through countless design variation by digitally updating parameter (Nazim & Joyce).

4.2.2.3 Role in the conception and design of a building project.

Is impossible to underestimate or under evaluate the change in the work structure created by computational design. Nevertheless, nothing can be created in a parametric system if the designer has not clearly specified the relevant conceptual and constructive structures (Aish & Woodbury, 2005). Parametric Design concept is not debating the central role of human interaction in the design process, but it highlights the relevance and centrality of designer in the digital model's design.

This means that, as complex as it may be, the control of digital processes is based upon interaction and reflection of the designer (Oxman, 2006). In fact, optimal design freedom comes from an appropriate balance of automation and human interface in Parametric-associative modelling and genetic-programming methods of generative design. (Chaszar & Joyce, 2016)

Dynamic concepts are driving digital design formalisms to redefine the role of model representation, since new bases for design thinking are being stablished thanks to advance digital techniques (Oxman, 2006). Parameterizations increase the complexity of designer's tasks and multidisciplinary knowledge since the model is not only a design tool, but a conceptual and digital structure of the design process. Indeed, parametric modeling task is simultaneous with the task of creating a design, at the end the result is a concrete design represented by a graph structure and instances. This model symbolizes the chosen relationship decisions and delivers computing precise values or structures that depend on the relationship (Aish & Woodbury, 2005).

Nowadays, designers need a rigorous introduction to computation (Kirschner, 2015). New digital architecture creations require architects perception and cognitive abilities to build digital geometric forms in computer programs, this requires the fundamental of computational foundation regarding its action and reaction rules (Maskuriy, Selamat, Ali, Maresova, & Krejcar, 2019). Parametric associative modeling has become relevant in architecture, because inside a visual programming environment it allows designer to fit design logic in graph-based definitions of geometric relationships and variables (Nazim & Joyce). The rapid development of generative tools including analysis and simulations has encouraged inspiring projects and proposals. The arrival of many new tools is imminent as the beginning for exciting times in architecture design. (Mirtschin, 2010).

4.2.3 Visual programming

Visual programming is at the core of parametric design development since it makes programming and computing more accessible to visual thinkers. Visual programming tools aim to connect a difficult and complex, even esoteric activity like programming, with a new audience (Kirschner, 2015). It connects functional blocks into a sequence of actions keeping a simple syntax where the blocks receive an appropriate and well structure data type according to the delivered a desired result. This characteristic release the user of learning a new programming language creating a paradigm in programing, where the user manipulates logic elements graphically instead of textually (Mode Lab, 2015).

Sutherland’s Sketchpad in the 50’s understood that associative rules can be expressed by a graphical method base on node diagrams. The tree of dependencies was able to be visualized like a flow chart and manipulated with instant effects on the on the drawing. (Arturo Tedeschi, 2014)

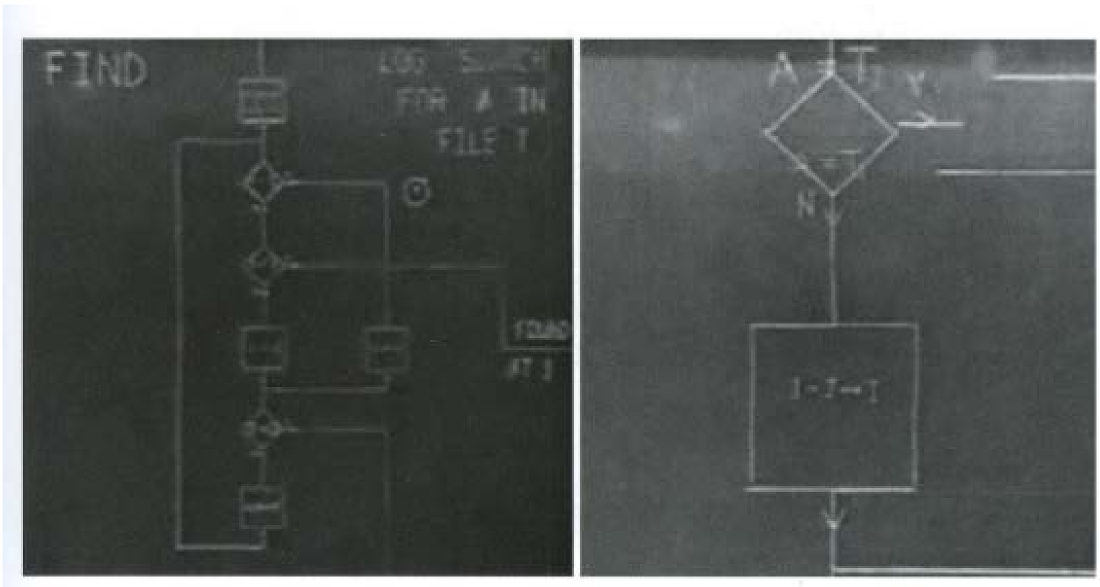


Figure 21: The Sketchpad (1963) flow chart (Arturo Tedeschi, 2014).

Moreover, in 1975 David Canfield Smith’s wrote a turning point in visual programming with his PhD dissertation “Pygmalion: A Creative Programming Environment”. Pygmalion proposed an icon-based programming paradigm where the user created, modified, and linked “icons” with defined properties to perform computations (Kirschner, 2015). Nowadays, many software firms have developed visual tools to make scripting accessible to users with reduced or any programming skills. (Arturo Tedeschi, 2014)

Coding is a tool to express logic, method and function wrote and intended to be read by humans. Therefore, one of the goals of visual programming is to improve readability and expression of complex interactions in code with languages composed of tangible objects (Kirschner, 2015). Therefore, visual programming language provides designers the tools for constructing programmatic relationship by a graphical user interface, instead of starting to write a code from scratch the designer assembly nodes with a predefined relationship to create a custom algorithm (Autodesk, 2019).

4.2.3.1 Classes

For Boshernitsan and Downes in (Kirschner, 2015), visual language classes are defined by the following characteristics:

Data Flow Languages: These languages do not usually expose the concept of execution order or control flow to the user, instead the entire program can be thought of as a single large expression that is evaluated as needed. Nodes execute when their inputs have been executed. They lend themselves to a functional programming style.

Spreadsheet or Form Based Languages: These languages use the idea of a cell or spreadsheet of 2D tabular data to build up composite objects. Applications like Microsoft Excel or Apple Numbers might fit partially into this classification. More powerful general programming languages that use this technique allow nesting of different groupings of 2D data into complex abstractions that carry data around with them.

Purely Visual Languages: These languages attempt to avoid any textual output or input where possible instead relying on visualizations for almost all precision and communication of program function.

Hybrid Text/Visual Languages: These languages combine both textual input/output and visual representation. Many of these languages also allow input from other textual languages. For instance, both Autodesk's Dynamo and McNeel's Grasshopper both allow for building of nodes within c# and python textual programming languages. These are arguably the most powerful classification of languages and have greater appeal to larger groups of user types.

Explicit Execution Languages (Control Flow): These languages either have some mechanism for explicitly ordering the execution of nodes or expose it as a central visual concept like data flow in data flow languages. This is the analog of constructs like line numbers, function calls, go to statements, and loops in a text-based language. The language described in this thesis is an explicit execution language, it is necessary for the user to understand the concept of sequential execution to use this language.

Constraint Languages: These languages allow the creation of constraints of various types between objects in the language. Ivan Sutherland's PhD Sketchpad CAD drawing tool is one of these languages. The user was able to create constraints between various geometrical entities.

Constraints in other languages might not relate directly to objects in the world, but the output value of nodes for instance.

By Example Languages: These languages allow the user to interact with a set of objects and possible interaction methods to 'teach' a system about a program that should be generated. Pygmalion, Apple Automator, and Photoshop Actions are all examples of this kind of programming environment.

4.2.4 Elements

4.2.4.1 Working environments

Parametric design visual scripting in based in two main working environments: a visuals editor (A) and a 3D modeling environment (B). Each environment represents two different outputs, a node diagram also called parametric diagram or visual algorithm, and a parametric diagram constitute by 2D or 3D geometry. (Arturo Tedeschi, 2014)

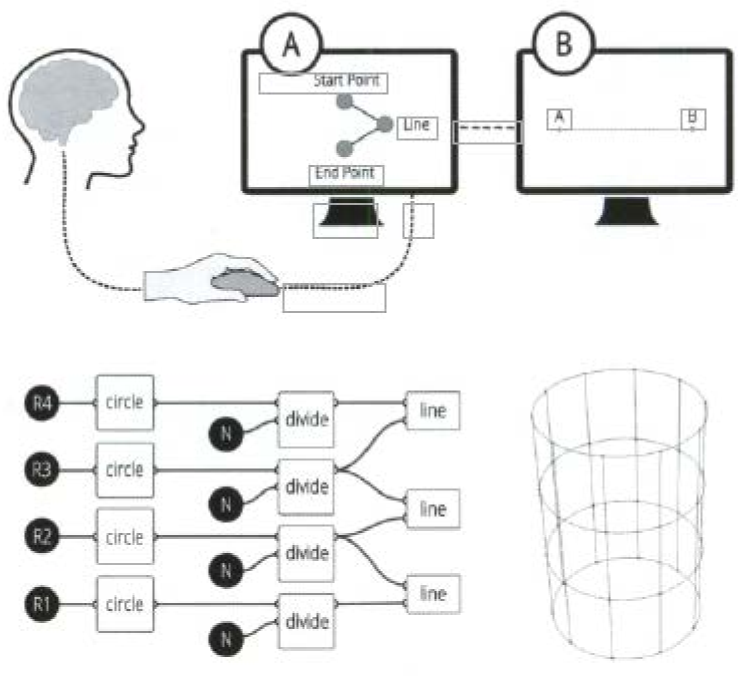


Figure 22: Working environment in PD (Arturo Tedeschi, 2014).

4.2.4.2 Components

The visual programming components can be divided as nodes, ports and wires, and normally they are organized in libraries.

Node / Icon: Icon objects that perform an operation, they can be connected by each other to build and visual algorithm. Nodes can be described as blocks with an intrinsic function or object.

Port / Pin: Part of the node and defines the input variable needed in the node to execute its function. It and can be further classified into in data going in or out the node as input or output ports.

Edge / Wire /Arc: Represents data flow from one node to another indicating the relationship between them.

Library: Is a collection of function or computational procedures. They can be built in primitives, imported and use from other programs, or constructed by the user and added to this repository.

4.2.4.3 Lists and Data trees

Lists and Data trees are the core structure of visual programing in parametrical design. A data tree is a hierarchical structure which organized and store data in lists. Output information from a component delivered a structured data organized in a list or set of sub-lists. Therefore, multiple lists of data as an output is a possibility, but there is a need to identify each individual list. In fact, they work as a folder structure in a regular computer organization, acceding indexed items involve going through paths that subsequently collect parents lists and their own sub-index. (Kirschner, 2015).

Furthermore, since most nodes return new data structures lists and data tree structure allow users to manage all possible transformations states of data. In fact, users can inspect the results from one node to the next and understand the transformation that the node performs. Unfortunately the granularity is usually low since the nature of nodes is to deliver from a multiple evaluation a list of results (Kirschner, 2015).

4.2.5 Grasshopper

Grasshopper (McNeel, Rutten) is a visual programming plugin for Rhinoceros 3D (McNeel, Rutten), which provides a visual programing interface to the large Rhino geometry library and built in tools for lists and nested list manipulations (Kirschner, 2015). Grasshopper allows a parametric control over models in a higher level programming logic platform, therefore is has the ability to explore generative design workflows in an intuitive graphical interface (Mode Lab, 2015). In addition, thanks to an ease model preparation Grasshopper increase productivity of model exchange from PD environment with commercial solvers capable of form finding against geometric constraints, pre-stress, orthotropic materials and imposed loading (Mirtschin, 2010).

Back in 2008, Grasshopper origin is related to “Record History” feature on Rhino3d Version 4. This built-in function stores in the background the modeling procedure in real time. For example, the action of lofted curves recorded with record history preserved the loft action even if the curves were modified updating the resultant surface geometry for the modified curves. Further on, in the pursuit of more explicit control over history function the precursor of Grasshopper was created, “Explicit History”. This new function allow detailed editing and encourage users to develop logical sequence by giving access to the history tree of a model (Mode Lab, 2015).

One of the limitations in Grasshopper is that it does not supports recursion. In fact, Grasshopper manages iteration in a semi-automatic way, like a form of map or auto mapping procedure in functional languages. It looks to match analogous sets of data types for any functions in the argument that is handle to that function. Input information will automatically call a node multiple times over list of inputs if those inputs can be passed to the function (Kirschner, 2015).

4.2.6 Dynamo

Dynamo (Autodesk, Ian Keough) is a hybrid textual visual programming language built on Autodesk design script language running on .NET. It depends on functional constructs to perform iteration in most instances. Its language is built around the interoperability with geometry kernel and BIM platform, Revit (Autodesk) (Kirschner, 2015). Through its node-based Visual Programming interface Dynamo enables the designer a custom computational and automation processes, there is no limitation in complexity therefore users with different skill level can use the tool to be productive at their own pace. Dynamo allows a sophisticated data manipulation, relational structures and geometry control which are not possible using a traditional 3D modeling interface. Dynamo access Revit's elements and directly manipulates their parameters. The same parameter can belong to different elements complicating its manipulation just from Revit interface. (Nezamaldin, 2019).

The main focus of Dynamo is to build parametric geometry objects or to inspect and organize Revit primitives in BIM (Kirschner, 2015). Like other software, Revit can be manipulated by programming. Dynamo use code block nodes or Python scripts nodes to write programing codes which access directly into Revit's data structure and manage information directly from it (Nezamaldin, 2019). In fact, Dynamo is built to amplify the parametric capabilities of Revit through the logic of a graphical algorithm. As a result, users have the ability to (Alfabuild ; Санджиев, et al., 2018):

- Connect the workflows with different software.
- Access the Revit API.
- Automate processes.

Connectivity with other software is one of Dynamo greatest advantages in Revit, for example Excel (Microsoft). This connection enhances the interoperability between Revit and other BIM software (Nezamaldin, 2019). In addition, Dynamo also have packages available to download directly from its interface. Every package includes new nodes develop by other users extending Dynamo initial built in library. Furthermore, Dynamo can be used as a tool to correct imported IFC models, since it gives Revit a greater degree of flexibility to manage the import data structure. This supports the use of the IFC format as a more viable way to exchange information with an efficient coordination and verification (Alfabuild ; Санджиев, et al., 2018).

4.3 Parametric BIM Workflow:

4.3.1 Definition

Parametric BIM workflow is a method to link graph-base systems to BIM systems, therefore creating directly generative associative BIM models (Janssen, 2014). This means that the iterative behavior of parametric design adds to the BIM data structure with data matching algorithms that correctly interact with these data structures. As a result, the digital model can still rely on the parametric design while external information streams are feed from a BIM environment. Contrary to an exported compatible model which only generate explicit geometry, parametrical BIM workflows generates and associative model with its corresponding BIM information, thus allowing a more user friendly and streamlined BIM workflows to be created (Janssen, 2014).

As a matter of fact, when dealing with complex geometry and high degrees of variation a traditional BIM modelling may not be a suitable option. Hence, Parametrical BI M workflow deal with these constrains turning Building information Modelling into Building information Generation (Wortmann & Tunçer, 2017). The challenge is to keep a whole and simple intuitive process for designers through the balance between, advanced rules with intelligent interface, and an intuitive and simple ways for the designer to override this intelligence (Jassen, 2016).

4.3.2 Context

4.3.2.1 Compatibility of PD and BIM

Regardless of BIM built-in parametrical functionality it is mostly used on an object level, creating an assembly of rather independent objects. Moreover, even if the model itself contains a high amount of project related information, sometimes its use is limited to traditional drawings and 3D views. On the other hand the interest of dataflow modeling or visual programing present in PD depends precisely on the presence and use of all the information available in diagrammatic logical sequence, acting as the main interface of the project. (Boeykens, Bridging Building information modeling and parametric design, 2012).

There is a key point in the design where from minimal exploration modelling you must go to its maximal detail; this means going from conceptual modelling tools to BIM tools (Jassen, 2016). As a matter of fact, an improvement in the integration between this two approaches strength the design process (Boeykens, Bridging Building information modeling and parametric design, 2012). BIM models are the basis for a structured analysis of the project but is possible to improve the way they are created. BIM have enhanced rule based generative design or parametric modeling. In fact, as Parametric Design, Building Information Modeling is a collection of objects with their own parameters like geometry, attributes and relations (Hugo & Charles, 2011). Furthermore, BIM models are created progressively adding information and design parameters from design consultant and specialists into the project.

4.3.2.2 Advantages in the AEC industry

Is undoubtable that Parametric Design and Building Information Modeling have helped in the evolution of the AEC industry, and that the two concepts are coming closer to be a dissociable methodology. This coupling through parametrical BIM workflow represents a back and forth inter-scalar coordination and data accurate decision making from design to construction, and the ability to add a multiples objectives parametric approach into BIM constructive universe. In fact, Parametrical BIM workflow profs that an effective cross application workflow from design to BIM can use content created in a third-party software and to prevent duplicate work. In addition, as much as possible parametrization and automatization of BIM enable an accelerated creation of design iterations and the implementations of modifications saving time on repetitive tasks (Schwerdtfeger & Zaha, 2018). BIM and Parametric modeling allow macro and micro design decisions through a better integration of both, structural and mechanical systems (Haliburton, et al., 2011). Furthermore, genetic and parametric creation of BIM models in a controlled manner opens the possibility of leveraging Artificial Intelligence in the construction industry (Nazim & Joyce).

Between the Sketch Design Phase (basic design), Preliminary Design Phase (design coordination), and the Construction Design Phase (executive design) the project is constantly adapted, refined and improved thanks to previous design experiences, and project characteristics feedback from different parties. Furthermore, concerning the level of detail inside each phase is possible to distinguish between three scale levels, from a master, to block and finally space level. Parametric BIM workflow stablish a design environment which support this back and forth iteration throughout the whole design phase. (Boeykens, Improving Design Workflow in Architectural Design Applications, 2016)

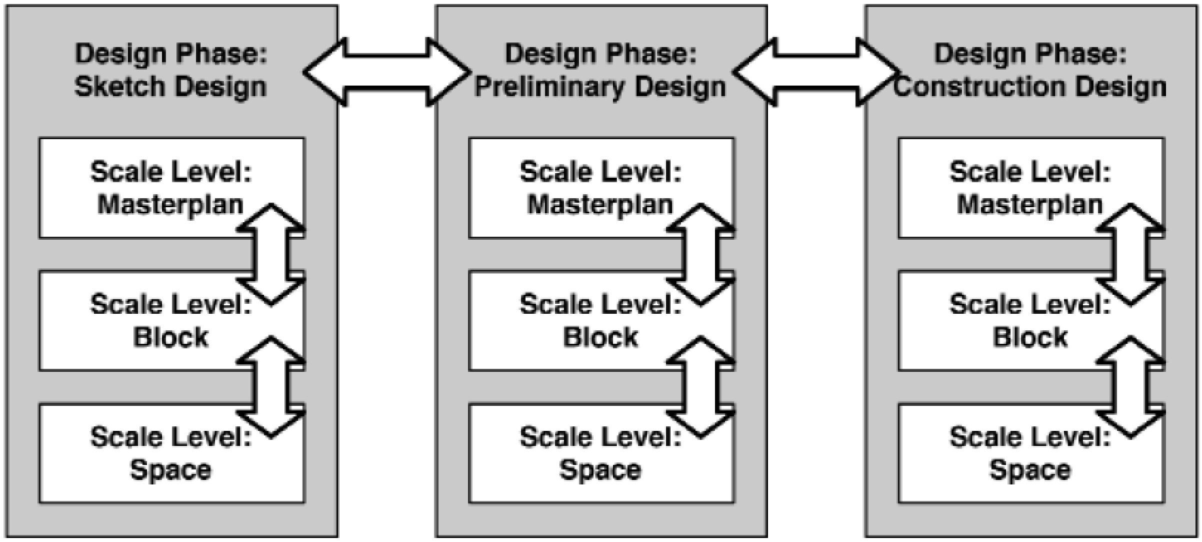


Figure 23: Inter-scalar design (Boeykens, 2016)

But, to integrate, process and assess many alternatives in a rapid succession and maximize the benefits of a generative BIM model manual coordination of independent models should be minimized (Mirtschin, 2010) Iterations are mandatory in a quality design solution during the design process. Each design phase iteration is distinguished by its detailed level. Efficiency is obtained through an BIM model approach for a shared level of detail and information regardless of the design phase. (Hansen Gøran & Bjørn Smith, 2017). Parametrical BIM automated process can deal with the generation of components at various scales, from whole building models to complex non-standards structural or faced details. (Janssen, 2014).

Complex BIM models are time consuming and prompt to human mistakes and inaccuracies (Janssen, 2014). Therefore, parametric modeling has increased in popularity worldwide since its demand has expanded in Building Information Modeling construction projects (Barazzetti, 2016). In fact, the implementation of solvers like Galapagos in Grasshopper; where genetic algorithms have a goal-oriented task through multiple objective optimization; can replaced a manually conducted optimization (Mirtschin, 2010). For example: A large engineering and design company like Atkins has implemented an advance parametric design techniques for detailed design “optioneering” in the water infrastructure industry, which made possible to provide 22 design options in one day, a 95% time improvement on traditional design methods for similar results. (World Economic Forum, 2016). Nevertheless, for a designer to evaluate the output results of the processed script and to succeed in the current market demands it is necessary to go beyond its personal spatial vision and dominate tools and processes to achieve a higher level of coordination within the design (Arturo Tedeschi, 2014).

4.3.2.3 Participation on the early stage of design

Considering a traditional human-computer interaction, conceptualization and the early stages of design the activity is best perform with a higher emphasis on the human, while the computational participation starts later on and increases as moving forward in the detail production and evaluation (Chaszar & Joyce, 2016). It seems to be common agreement that designer structure the design problem in different representation diagrams in order to solve them, and then develop strategies to overcome these problems including design constraints and technical requirements. Therefore automating this process has the potential to include digitalization in the early design stage to ease the inconstancy associated to a manual checking and delays (Hugo & Charles, 2011). Parametric model captures all the known parameters and associations, and generates a design solutions that can be explored and controlled by modifying its variables (Nazim & Joyce), but once the switch is made to BIM tools, the objective is to define the building elements as precisely as possible with its thickness, materials and other details, and seems to limit back and forth iterations between conception and definition (Jassen, 2016). Nowadays, BIM has a limited support for the early design stages, specially to transfer the building model throughout the different phases of the design process and does not allow iterative and recursive exploration. This is of the outmost importance in the early design stages where architects must make design decision with a large range impact and must explore over different scales going from the generic to the detail and vice-versa (Boeykens, Improving Design Workflow in Architectural Design Applications, 2016).

Design as a practice has been a “tricky” problem since model definitions may face many changes and even fully redefinition depending on the needs and exploration interests (Nazim & Joyce). Therefore, even if the real potential of BIM is attained if its extend throughout all phases of the design process, architects are reluctant to start using a BIM approach in the early stages of the design and rely more in generic modeling and drawing techniques (Boeykens, Bridging Building information modeling and parametric design, 2012). As a matter of fact, designer’s transfer from parametric models into BIM happens when the design is almost finalized and the generated geometry is mostly static, therefore there is no way back once the model is transfer. Furthermore, as design evolves over different iterations and design phases the model should not be started all over again. This shortcoming interaction of the current BIM approaches can be further improve an evolve towards and Architectural Information modeling or a Generative Building information Model to better cope with incomplete or late design information and not explored early optimization during the preliminary stages of design (Boeykens, Bridging Building information modeling and parametric design, 2012).

4.3.2.4 Limitations

Parametrization required an supplementary effort from traditional approach, since it increase the complexity of every design decision and increase the items to be analyzed to fulfil its task (Aish & Woodbury, 2005). This is an issue intrinsic to architecture, where the complexity of a building is the result of a timeframe, cost and socio-economic and political context (Arturo Tedeschi, 2014). As a matter of fact, relationships that designers need to analyze to complete a design are idiosyncratic and considerably large to sometimes set as node types in a parametrical modeling environment. Designers involve in parametric design must be able to develop specific and simplify relationships according to the task at hand (Aish & Woodbury, 2005).

The complexity of representation and interface is another challenge to face in parametric model that restrains it use in a BIM workflow. At a base level designer new set of skills to understand new concepts such as itself, node compilation, intentionality and a set of mathematical ideas related to descriptive geometry and linear algebra. The interface looks to be the principal technical problem regarding its expiation in the practice. Although algorithmically simple, there is complex sophisticated language-level and users interface needed. In fact, the structure of design process using parametric programming remains poorly understood. Designers must simultaneously face a design problematic and capture it in a conceptual mathematical structure (Aish & Woodbury, 2005). Nowadays designers are proficient with two-dimensional or three-dimensional modelling, but most are not comfortable with programming/scripting environments, they face a challenge moving away from their stablish domain into computer programming space. Yet, the objective is not to be a professional programmer but to write a code that support the delivered design task (Kirschner, 2015).

Contrary to Graph-based systems focus on design exploration where models are responsive and rapid to update, because they are centered in based light-weight minimal models. BIM systems focus on the design interrogation, therefore they require maximal models that can incorporate as detailed information as possible (Janssen, 2014). Building information modeling only moves forward between the different phases accumulating as much information available, which makes the model extensively structured and with a clear semantic information. Though out the modeling, created object embedded information inside which are organized as property values of the item. The objective is that all entities have a clear meaning and function since the culmination of a BIM structure are the Industry Foundation Classes. IFC describe all possible and foreseeable building elements in over 800 entities, 350 property sets and over 100 data types (Boeykens, Bridging Building information modeling and parametric design, 2012).

In conclusion, the fundamental limitation for a parametrical BIM workflow is that PD and BIM use are highly different since creating a single BIM system that support both exploration and building information may not be viable. There are two reason (Janssen, 2014):

- Nowadays, BIM systems have very complex user interfaces and adding advanced dataflow and procedural modelling capabilities may result in a user-interface that is far too complex resulting in a steep learning curve for novice users.
- BIM models are already a very nature large complex dataset. Hence, including parametrically exploration for users to such models may severely reduce the performance and robustness of the system. For Example: when making parametric constraint-based changes to large models in Revit the interface can become slow and may often result in errors.

4.3.3 Approaches

4.3.3.1 BIM modelling tools that support geometric import

One approach to Parametric BIM workflow is to import just the geometric model directly into a BIM tool, and then convert the geometric entities to BIM native elements (Jassen, 2016). Inside BIM during early design modeling it is possible to use a mass modeling approach, where basic primitive volumes and voids are created and then sliced in multiple geometric shapes that afterwards will be assign floor levels, spaces, and enclosing elements. Nevertheless, this is still a primarily unidirectional workflow, since the mass model is only used at the beginning of the process and is abandoned once the building elements are generated, it does not support an iterative design process flowing back and forth between scale levels and design phases. (Boeykens, 2012)

For example, in a software like Revit, the import of geometric solids is allowed as massing models. Once imported the faces of the conceptual mass can be converted to BIM native elements such as slabs and walls. Unfortunately, most of the BIM elements cannot be created in this way (Jassen, 2016). Nevertheless, working a model in a native format can be an considerable solution when all participant are required to do so, simplifying the complexity involved in exchanging the model (Alfabuild ; Санджиев, et al., 2018).

4.3.3.2 Geometric modelling tools that support IFC export

Considered as an embedded approach, it implies that parametric system will support a BIM compatible data format export to be imported into a BIM software. It means transferring semantically richer models between both methodologies (Janssen, 2014). The use of a graph base system that supports IFC exporting is the most direct to approach a Parametric BIM workflow. Nevertheless, exchanging files always expose serious limitations regarding the type of information or the geometry that can be processed. In fact, there are two down sides to this process (Jassen, 2016):

- The 3D model must be definitive, thus there is any advantage over modelling it directly in a BIM software
- Created IFC element can no longer be easily modified once imported in the BIM environment since it is defined as a boundary representation,

Even after writing valid IFC export models, a typical BIM implementation does not support all possible geometric elements coming from a parametric modeler. The best results are achieved for boundary representation (BREPS), which are straight extrusion and with static geometry. (Mirtschin, 2010). In a model defined in IFC format, current deficiencies in the process of importing lead to undesired results that need to be remodeled from incomplete models or with unnecessary parameters that have failed to program the correct restrictions, this reorganization of information within the import program imply an investment of a lot of hours. For example: Inside

Revit, once a model in IFC is imported all elements that are originally of the same type are not recognized like one. Instead, different types are created for each copy, which obviously differs from the parametric base purpose. As a Result, the type information must be assign manually one by one, increasing the risk of misplaced information due to human mistakes (Alfabuild ; Санджиев, et al., 2018).

4.3.3.3 Graph-based parametric BIM modelling systems.

A different approach is a coupled approach, where dedicated graph-based systems are coupled to BIM systems, thus allowing graph-based systems to generate models and BIM systems to manage the data. The assembled parametric objects used in the BIM environments are further connected and related to each other. Through a parametric modeler, object creation in BIM are constrained by rules and recipes, rather than by manual modeling and static positioning. Parametric modelers define and customize the rules for the materialization process into BIM (Jassen, 2016). BIM systems have been extended to support dataflow or procedural modelling these approaches could be used to generate models or parts of models (Janssen, 2014). In fact, visual programming framework allows the user to create a unique system of relationships expanding BIM universe which can be used to drive design ideation (Autodesk, 2019). Moreover, recent develops include tools that make possible to designer to assign BIM attributes to generative models and measure performance including costing and functionality, nowadays is possible to include in a generative model performance characteristics, cost, floor usage, egress, and project quantities (Mirtschin, 2010).

Numerous workflows and plugins have been developed for generating BIM models using directly parametric modelling tools. Two approaches have been stablished (Jassen, 2016):

- Loosely coupled
- Tightly coupled

An important range of tools are starting to emerge supporting these two approaches of parametric BIM workflow.

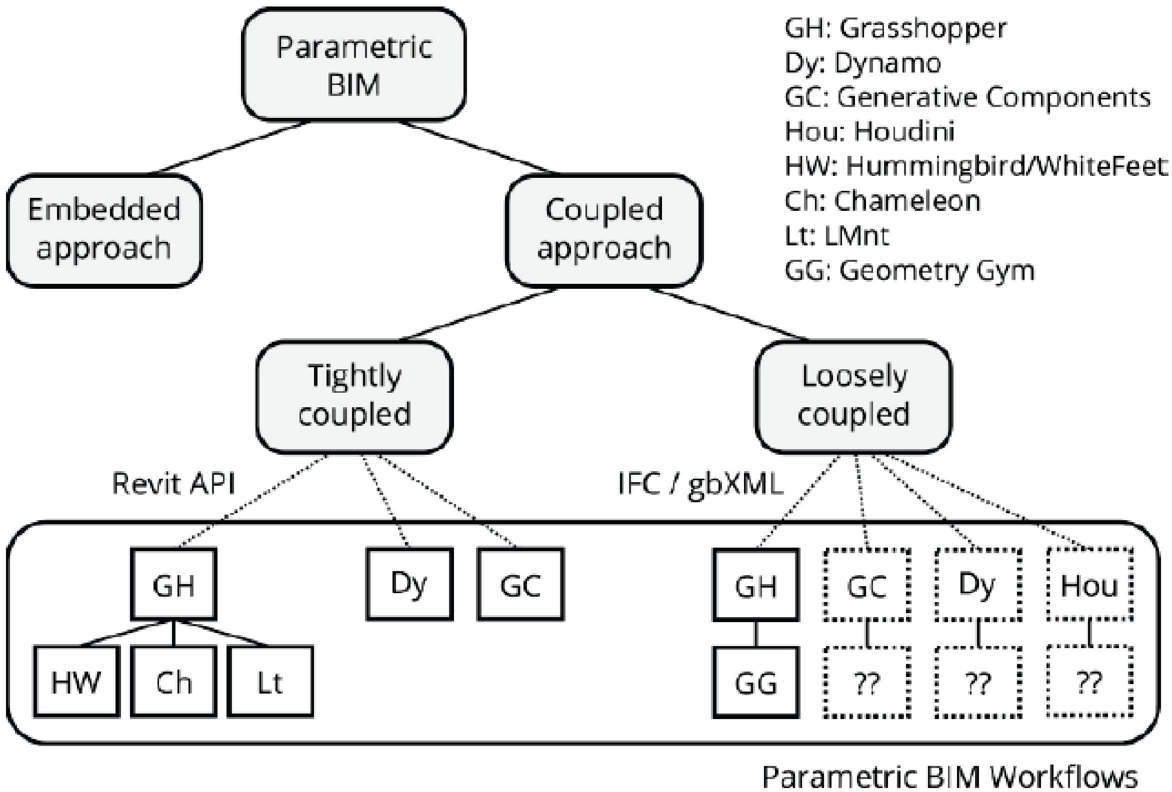


Figure 24: Parametric BIM workflows (Jassen, 2016).

4.3.4 Loosely coupled

In a loosely coupled approach, systems (PD and BIM) are coupled through model exchange. Meaning that a parametric graph-based system generates data directly in a standard file format (IFC or gbXML) that can be directly imported into the BIM system. (Janssen, 2014). The use of an open standardized file format exchange is one of loosely coupled advantage, it allows user to link diverse tools and systems that support forms of open collaboration and exchange. Although this approach still evolving some examples are Grasshopper/Geometry Gym for IFC exchange format, and Grasshopper/Chameleon for gbXML (Janssen, 2014). In the case of Geometry Gym output file, since the result is and open standard IFC file it can be import to any BIM application that can support an IFC interoperability. (Janssen, 2014).

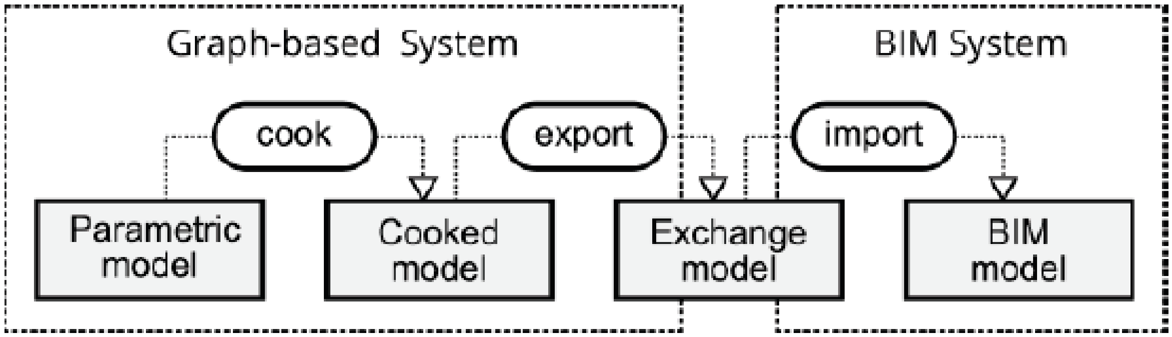


Figure 25: Loose coupling between graph-base systems and BIM systems (Janssen, 2014).

Contrary to an embedded approach. that creates and impractical strategy for a parametric BIM workflow creating a cooked model that only contains explicit geometry from the graph base system. Loosely coupled approach develops a more sophisticated exporter since it must transform the cook model into an associative exchange model to achieve a fully parametric BIM workflow. These approaches are still evolving (Jassen, 2016).

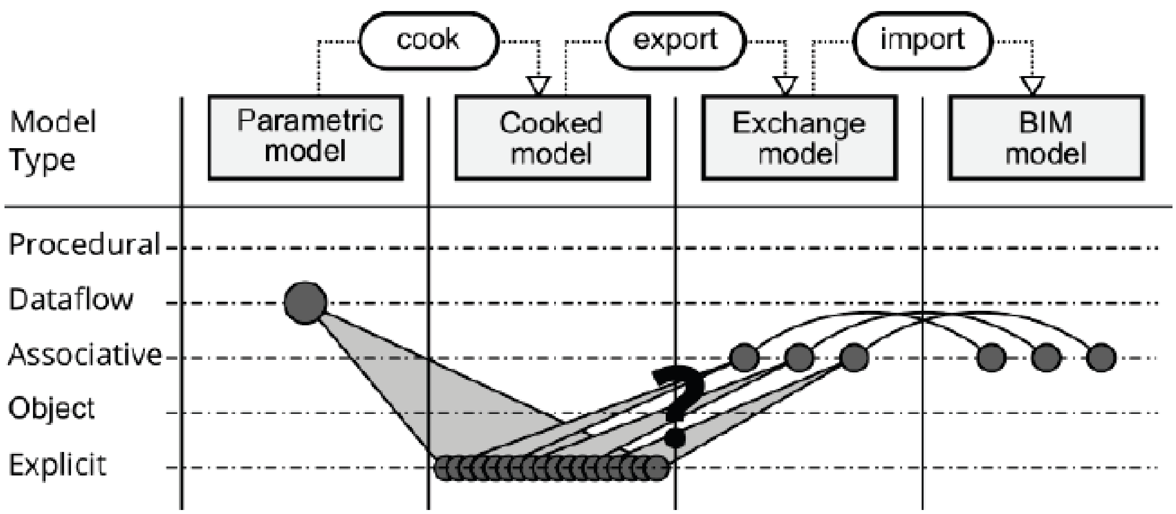


Figure 26: An impractical strategy for parametric BIM workflow (Jassen, 2016)

4.3.4.1 Geometry gym

Created by Jonathan Mirtschin, Geometry Gym's approach to the problem taken is to use the dataflow model to directly generate the exchange model, avoiding the need for an exporter altogether and therefore a cook model. Unfortunately, the dataflow model becomes saturated with extra nodes need by the exporter but have a very marginal relevance to the parametric modelling task. This increase the complexity of the dataflow graph (Janssen, 2014).

Geometry Gym plug in functions are being used to automate modeling procedures in a cost-effective manner. This plug in has already release third party software tools to assign relevant attributes to BIM data, allowing the exchange with these models using BIM standards files. In fact, IFC specification also includes data management to generate and edit attributes for building services and MEP. Furthermore, NURBS geometry representations are also incorporated in IFC, this allows a more accurate model representation of freeform geometry architecture commonly model in graph base systems (Mirtschin, 2010).

4.3.5 Tightly coupled

Tightly coupled approach is a system where the parametric modeler communicates with the BIM system through the BIM Application Programming Interface (API). Thus, directly updating geometry in the BIM model each time the graph-based model is executed and creating a native BIM geometry by exchanging a process instead of the geometry itself. For example, Open Source graph phase system ANAR+ system (Labelle) generates scripts from a parametric model in Processing (Reas & Fry) which includes an option to define native model definitions for a BIM system, using the same compatible scripting that ArchiCAD API. This approach is not exchanging geometry, but the underlying recipe to re-create this geometry in the receiving application. (Jassen, 2016)

So far, the most common native approach for parametric modeling in a BIM environment is through directly programming or scripting of associative objects in the same BIM platform (Boeykens, Bridging Building information modeling and parametric design, 2012). Dynamo provides the parametric functionality of Grasshopper directly into the BIM software, therefore creating a computational design workflows within the context of a BIM environment. (Autodesk, 2019). Nevertheless, even if it relies in an in-built BIM visual programming environment, a tightly couple approach comes from the idea of a message passing system from even an external parametric graph base software. Inspired by multimedia systems like VJ and audio performance, this approach communicates in real-time with different software tools and hardware devices using simple messages. As an advantage, incompatible systems avoid the limitations of operating

Systems and running on the same hardware, they can talk between each other, over some network (Jassen, 2016).

The strategy is to create a parametric BIM workflow that avoids explicit geometry and maintains a clear separation between parametric model creation and an exchange file generation. Hence, the cooked dataflow model does not result in an explicit model but instead generates a set of associative objects. (Janssen, 2014).

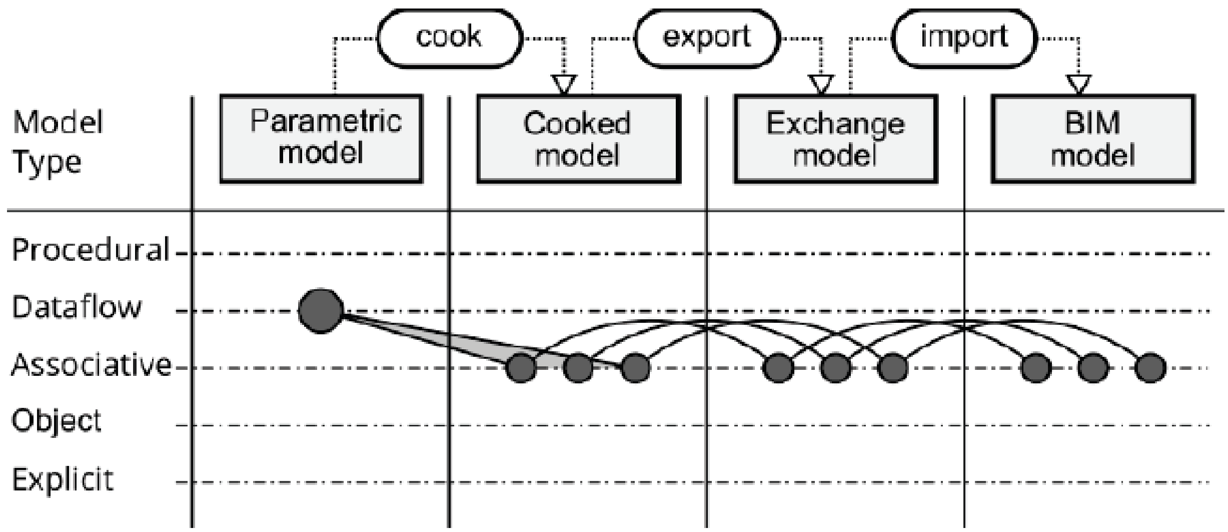


Figure 27: The proposed strategy for parametric BIM workflows (Jassen, 2016).

A tightly couple approach in not required to exchange the full set of information of the project at once. This means, not broadcasting the full building information, but optimizing a subset of messages used by the receiving application like a small set of commands. It works like a creation message which store a source ID, and once the parametric system is updated the objects built from the same source ID are modified, while preserving all other objects. Furthermore, is not necessary that each system supports the same characteristics or to has a fully geometry exchanges, for example, in a structural design software only node positions and node connectivity are necessary to recreate beams, columns and plates position (Janssen, 2014).

The explicit data representation stores a set of well-defined geometric entities in a simple data-structure list, meanwhile an associative representation show the associative data for each operation, thus giving access to its parameter (Janssen, 2014).

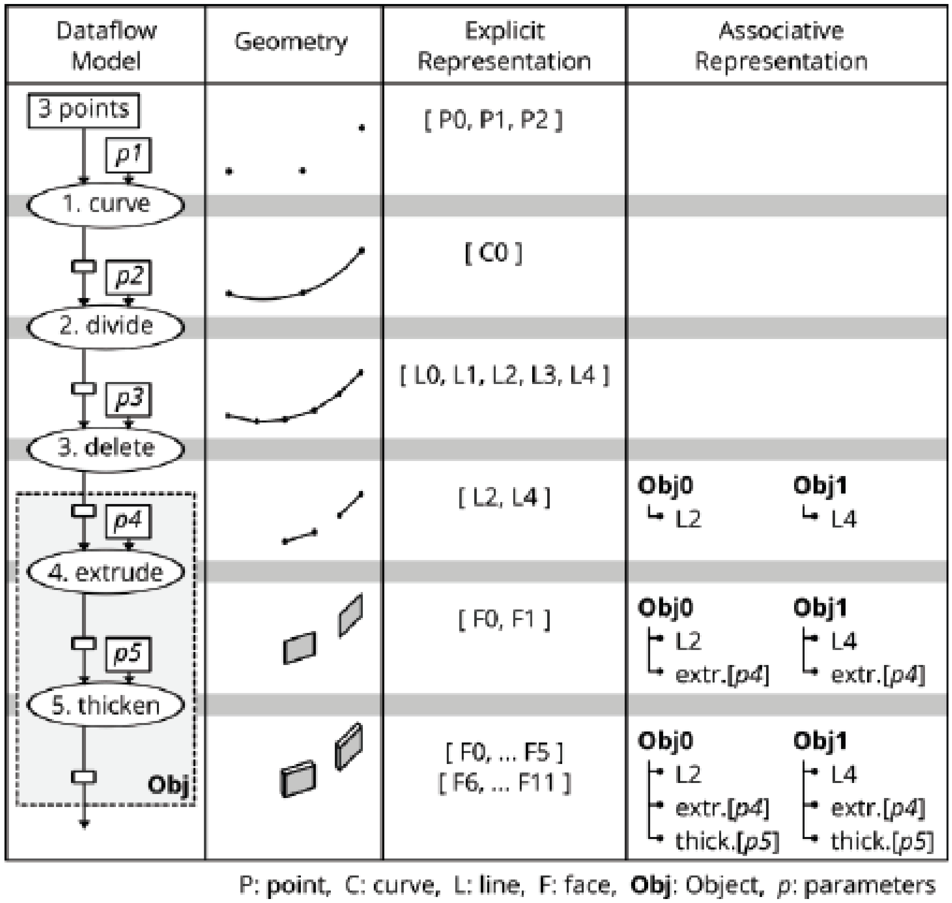


Figure 28: A comparison of the explicit representation versus the associative representation (Jassen, 2016).

Object-based representation advantage is that each object can be interrogated into its construction history. This allows a much more straight-forward BIM model exporter. For instances, in the case two walls are exported using an standard representation the centerline, width and height parameters could easily be extracted or manage, and modify (Janssen, 2014).

4.3.5.1 Hummingbird

Hummingbird plugin was developed by Mario Guttman and is a set of Grasshopper library of nodes that transfer information directly to Revit in order to create a BIM native geometry. The process transforms parametrical geometry inputs into Comer Separated Values (CVS) text file, then the data is import and rebuild in Revit thanks to a similar plug in included in the download package and installed simultaneously in the BIM platform called ModelBuilder. (Parametricmonkey, 2020)

At the graph based parametric modeler, the input basic data that constitutes a geometric object, like curves or points, is transform into a simplify geometrical description through point coordinates, or lengths. Then, the data is written an organized in a CVS text file thanks to a C# DLL library programming language. And finally, is transmitted through the Revit API into the BIM modeler with the ModelBuilder package, which translates the CVS text file information in native BIM geometry.

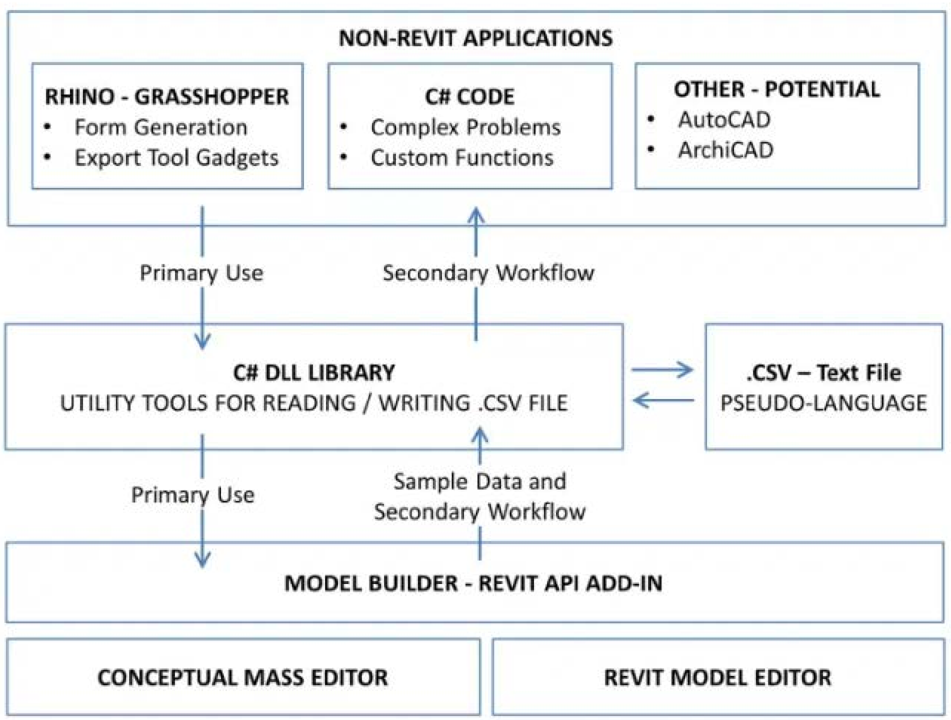


Figure 29: Hummingbird data flow (Project Hummingbird, 2020).

Furthermore, Hummingbird has recently added a bi-directional data exchange, which includes a component for reading .CSV files and using Revit modeler data information to build geometry in Rhino-Grasshopper.

4.3.5.2 Speckle

Speckle is cloud base workflow showing a real-time project connectivity among graph base parametric software and BIM software, it also includes immediate updates for large teams (Rhino to Revit and Back, 2020). Speckle started in November 2015 thanks to Dimitri Stefanescu as an instrument for digital design communication. Now days it is evolving as a management platform for digital data to face the problem of interoperability in the AEC industry. It is an open source data platform that provides a method of free sharing data form one platform to another in a rapid, organized and efficient way (Speckle: Data Platform for AEC, 2020).

Speckle remarkable characteristics are:

- Open Source project, it is not tied to any organization.
- Cloud base structure, it allows transfer from one software to another not necessary in the local machine but thought a various web platform.
- Fully single controlled data stream, it doesn't use one centralized server.

Speckle considers as "clients" the platforms responsible to send and/or receive the data stream. A connector which sends and receives data from an application common type of client. So far, connectors have been developed for Grasshopper, Dynamo, and Rhino. As clients keep growing the flexibility of Speckle increase for a designer since more software are included in the workflow (Speckle: Data Platform for AEC, 2020).

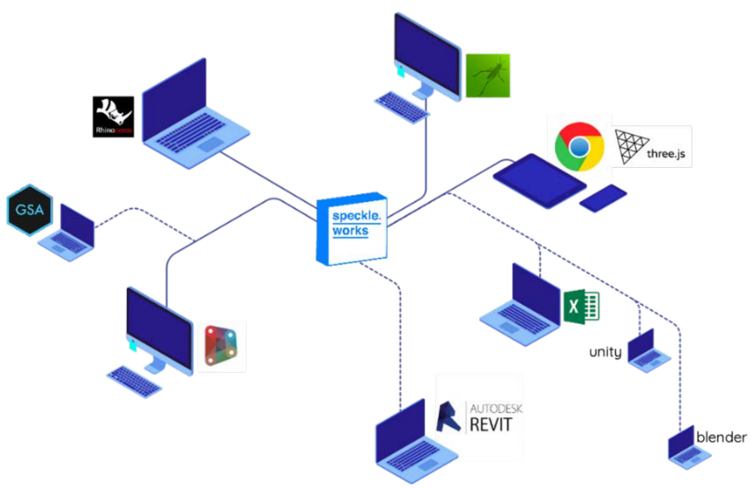


Figure 30: Speckle client's network (Speckle: Data Platform for AEC, 2020).

Nevertheless, the role of a client can also extend to the form of scripts and web applications. In fact, Speckle's own management interface is also a client. For example, ArupCarbon is a web application-based client tool which calculates the embodied carbon of a model sent from Revit.

Speckle can be used to:

- Platforms interface by extracting and providing data among them.

- Information Management assigning permission, sharing and organizing data.
- Streaming data of the same project between different users.
- Platform extensions in order to create a custom third-party applications and workflows.

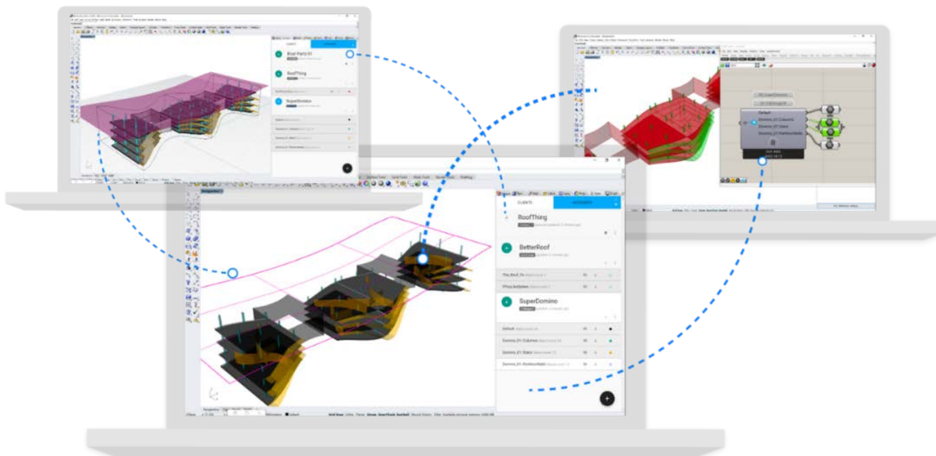


Figure 31: Speckle interface (Speckle: Data Platform for AEC, 2020).

Speckle currently available kits can be dividend in the interface of geometry and elements among clients. The possible interoperability is:

- Core geometry is used for Rhino, Grasshopper, Dynamo, Revit, and three.js (webviewer).

SpeckleGeometry	Github	Nuget							
Classes	**Rhino/Gh**		**Dynamo**		**Revit**		**Gsa**		**Threejs**
	ToNative	***ToSpeckle***	***ToNative***	***ToSpeckle***	***ToNative***	***ToSpeckle***	***ToNative***	***ToSpeckle***	**ToSpeckle***
Number	YES	YES	YES	YES	n/a	n/a	n/a	n/a	n/a
	YES	YES	YES	YES	n/a	n/a	n/a	n/a	n/a
String	YES	YES	YES	YES	n/a	n/a	n/a	n/a	n/a
Interval	YES	YES	NO	NO	n/a	n/a	n/a	n/a	n/a
Vector	YES	YES	YES	YES	YES	n/a	n/a	n/a	NO
Plane	YES	YES	YES	YES	YES	YES	n/a	n/a	?
Interval2D	YES	YES	NO	NO	n/a	n/a	n/a	n/a	n/a
Point	YES	YES	YES	YES	YES	YES	n/a	n/a	YES
Line	YES	YES	YES	YES	YES	YES	n/a	n/a	YES
Polyline	YES	YES	YES	YES	NO	NO	n/a	n/a	YES
Circle	YES	YES	YES	YES	NO	NO	n/a	n/a	YES
Arc	YES	YES	YES	YES	YES	YES	n/a	n/a	YES
Ellipse	YES	YES	YES	YES	NO	NO	n/a	n/a	?
Nurbs Curve	YES	YES	YES	YES	YES	NO	n/a	n/a	YES
Polycurve	YES	YES	YES	YES	NO	NO	n/a	n/a	YES
Mesh	YES	YES	YES	YES	NO	NO	n/a	n/a	YES
Brep	YES	YES	as mesh	n/a	NO	NO	n/a	n/a	YES
Box	YES	YES	NO	NO	NO	NO	n/a	n/a	YES
Extrusion	YES	YES	as mesh	NO	NO	NO	n/a	n/a	YES

Table 1: Speckle geometry interoperability (Speckle: Data Platform for AEC, 2020).

- Elements: Speckle Elements are used in Revit, and in Grasshopper via the Schema Builder component.

SpeckleElements	Github	Nuget		
Classes	**Revit Out**	**Revit to Revit**	**Revit In**	**Gh Out**
				(Creation via schema builder)
GridLine	YES	yes, buggy	yes, buggy	YES
Level	YES	yes, buggy	yes, buggy	YES
Floor	YES	yes, buggy	yes, buggy	YES
Column	YES	yes, buggy	yes, buggy	YES
Wall	YES	yes, buggy	yes, buggy	YES
Beam	YES	yes, buggy	yes, buggy	YES
Shaft	YES	yes, buggy	yes, buggy	YES
Room	YES	NO	yes, buggy	YES
Topography	YES	yes, buggy	yes, buggy	YES
FamilyInstance	YES	NO	NO	YES
GenericElement	YES	NO	NO	YES

Table 2: Speckle elements interoperability (Speckle: Data Platform for AEC, 2020).

5 Case study

The aim of this chapter is to design a building as case study in its generic conceptual phase (Stage 2), which will be defined by a generative algorithm, and directly stream from a parametric design software into a BIM environment. The objective is to obtain, measure and compare different and complete parametric workflows during the preliminary concept design, where the model is able to be modify parametrically, and, at the same time, be ready for further contribution at development design phase (RIBA Stage 3) from engineers and client through a BIM platform as REVIT.

5.1 Definition

5.1.1 Context

5.1.1.1 Preliminary Concept Design to Design Development

Inside the design of a project there are different phases or levels to go through, each one adding more definition and level of detail in order to build the project. The Conceptual Design phase in known as stage 2 in the RIBA plan of work is considered to be where a “robust” Architectural concept has to be produced (RIBA, 2020).

The minimum of preliminary concept content represent (Hugo & Charles, 2011):

- Building massing
- Program spaces
- Circulation
- Wall partitions

This is the first step into de volumetrically development of the project and intentions, in some cases according to RIBA the information required can be achieved by the designers empirical knowledge in other cases it can involves a series of analysis to be integrated.

5.1.1.2 Parametric BIM Workflow benefits in the early design

Parametric BIM workflow benefits the early design stage coordination since the process can be redefined and easily represented, includes design and technical analysis in a strategic a seamless digital workflow, and it allows adjustments to happened effortlessly between conceptual and detail design development phase.

Non digital design approaches are frequently considered limited regarding its speeds and accuracy in the design development and design variations. This is the results of the amount of time required to be produce (Chaszar & Joyce, 2016). A project design follows numerous redefinitions during its design phase, from abstract concept, to detail design (Hugo & Charles, 2011). One of the benefits of parametric modelling is that scripts are usually created bottom-up (Jassen, 2016). This means that it is assembly from smaller entities up to the entire definition. Hence, changing a small parameter can redefined a part or completely the design result. This modification results simultaneously in a graphic visual representation for every new definition of the project.

Designers in BIM and parametrical design are looking into a seamless digital workflow (Jassen, 2016), since interaction between redefinition and representation in a parametrical BIM environment allows to integrate and evaluate performance criteria and analysis. This compilation of information during the conceptual stage may require a back and forth exploration and changes in the design project, therefore digital workflow between models (parametrical and BIM) must follow along this exploration. (Boeykens, 2016)

Models in a parametrical BIM workflow decrease the workload regarding changes that involve different design stages. Evaluation of performance or analysis results during an advanced detail phase can lead to modifications on the conceptual design or the preliminary phases (Boeykens, 2016)**.** Conventional models are considered static, consequently modifications during the detail phase can be very limited. On the other hand parametric BIM models minimizes the effort of changes in a constant loop feedback (Schwerdtfeger & Zaha, 2018)**.**

5.1.1.3 Why in a residential building?

The AEC industry growth depends in a great part on its residential market. There is a flourishing market in many European countries, in 2017 there were a considerable increase of the market, doubling the average of deals targeting a residential builder in the past four years. (Deloitte, 2018) Furthermore, if we compare the construction volume worldwide the industry Residential housing accounts 38%, meanwhile energy and water infrastructure 32%, followed by institutional and commercial buildings 18%; and finally, industrial sites (from cement to automotive manufacturing) 13%. (World Economic Forum, 2016)

Moreover, is important to consider that residential demand is going to be the focus of further urban development considering that the population of the world's urban areas increasing by 200,000 people per day. This represents a great pressure on future planification to find the affordable housing as well as social, transportation and utility infrastructure. (World Economic Forum, 2016)

5.1.1.4 Selection of Grasshopper and Revit as the case study software

In the design and AEC industry computational aid has been a break thorough, pushing the limits of building and design, nowadays the future of the industry is based in the development of building information models and their definition and interaction by different disciplines involve in the construction field. Software tools have been developed to follow this evolution, regarding its function the more commonly use are Grasshopper (McNeel, Rutten) for conceptual and parametric design and Autodesk Revit for BIM modelling (Jassen, 2016).

As a matter of fact, Grasshopper (McNeel, Rutten) visual language, is now days a design education common language to share for designers as they build complex artifacts, and is at the base of a new way of thinking about design and computation (Kirschner, 2015). Also, it has modify the workflows of professionals across multiple industries and created an active global community of users (Mode Lab, 2015). For example, it is already been use in more fields of the industry than design, where structural consultants use Grasshopper (McNeel, Rutten) as a visual dataflow modeler, it reduces the time laps to generate different variations and modifications by the design consultant (Poirriez, Wortmann, Hudson, & Bouzida, 2016).

On the other hand, for Finances Online in 2018, Revit was considered the best Building Information Model software solution in (Nezamaldin, 2019)

1. Revit
2. Navisworks
3. Autodesk BIM 360
4. SketchUp
5. Tekla BIMsight
6. Procore
7. Dassault Systèmes BIM
8. Trimble Connect
9. AECOsim Building Designer
10. Hevacomp
11. BIMobject
12. BIMx
13. Archicad
14. Vectorworks Architect
15. PriMus IFC
16. Edificius
17. midas Gen
18. Allplan Architecture
19. Buildertrend
20. BricsCAD BIM

Figure 32: List of BIM software (Finances Online,2018)

In addition, Autodesk Revit has the capability to represent multiple views from the centralized building data base, therefore any change in one view is reflected in all of them. Furthermore, what distinguish this software from other architectural design applications is the parametric constraints than be define for the native Revit elements. (Boeykens, 2016)

Nowadays, Autodesk Revit and Grasshopper (McNeel, Rutten) are powerful tools in architectural exploration and had an impact in designers' final decisions (Mirtschin, 2010). Working together with the creativity and diversification offered by Grasshopper enhanced with the database-enriched digital objects provided by Revit are making BIM parametrical workflows applied across the construction industry (Banihashemi, Tabadkani, & Hosseini, 2018)

5.1.2 Limits

The case study will be defined as a residential building with a recurrent architectural contemporary concept and measure the benefits of a generative model. The model is defined as a conceptual model mass in a parametric graph base software as Grasshopper. The parametrical model definition considers as the final objective the following parameters:

- Stakeholder: surface (m^2) and typological distribution objective.
- Architecture: Architectural concept.
- Engineering: solar radiation optimization during summer and winter period (kWh/m2)

The case study objective is to show practical benefits of data integration on the early stage of design and to evaluate the integration of PD in a BIM environment through parametrical BIM workflows. Therefore, the model is defined as a conceptual mass with no surrounding but with a developed information of its components, like thickness and materiality of wall layers and slabs. The model keeps since an early stage an interoperability between design and construction, thus the maturity of coordination for this conceptual model is the highest since the begging. Following the standard, the model is defined as:

- RIBA design stage: Stage 2 – Concept Design.
- Level of definition UNI 11337:2017: LOD C - Generic object.
- BIM maturity level: Level 3: Interoperable data

Further, more the elements created and transfer in this case study are:

- Levels
- Slabs
- Shafts (vertical and horizontal circulation, technical rooms)
- External, internal, and core walls
- Rooms

Finally, the conceptual building mass will be transfer to a BIM platform as Revit. Three different methods will be analyzed to create a comparative matrix of parametrical BIM workflow:

- IFC export (Loosely coupled model)
- CVS Data Transfer (Tightly coupled model)
- Cloud base (Tightly coupled model)

5.1.3 Description

5.1.3.1 Architectural Concept

- Pop out or Pixeled

The Architectural concept chosen is a Pop out or Pixeled concept. A starting a regular volume base is virtually dividend in a regular grid, where cantilever boxes are attached to the façade in an apparently randomly distribution. The distribution of this boxes extends the floor surface and create a higher spaces flexibility. Nevertheless, this concept increases the floor plan distribution complexity since floor plans among floors are different between each other and repetitive typologies are hard to find. This concept has been used in both, residential and office building.

Figure 33: Architectural concept reference - The wedge Office building / A- Lab; 2222 Jackson / ODA New York; Carabanchel Housing / dosmasuno arquitectos; WoZoCo / MVRDV.

5.1.3.2 Base Building massing

The base building model is a square regular prism that can store 17 floors.

- Footprint = 18m x 18m
- Height = 51m
- Floor height = 3m
- Number of floors = 17 floors
- Shaft dimension = 5.5m x 5.5m

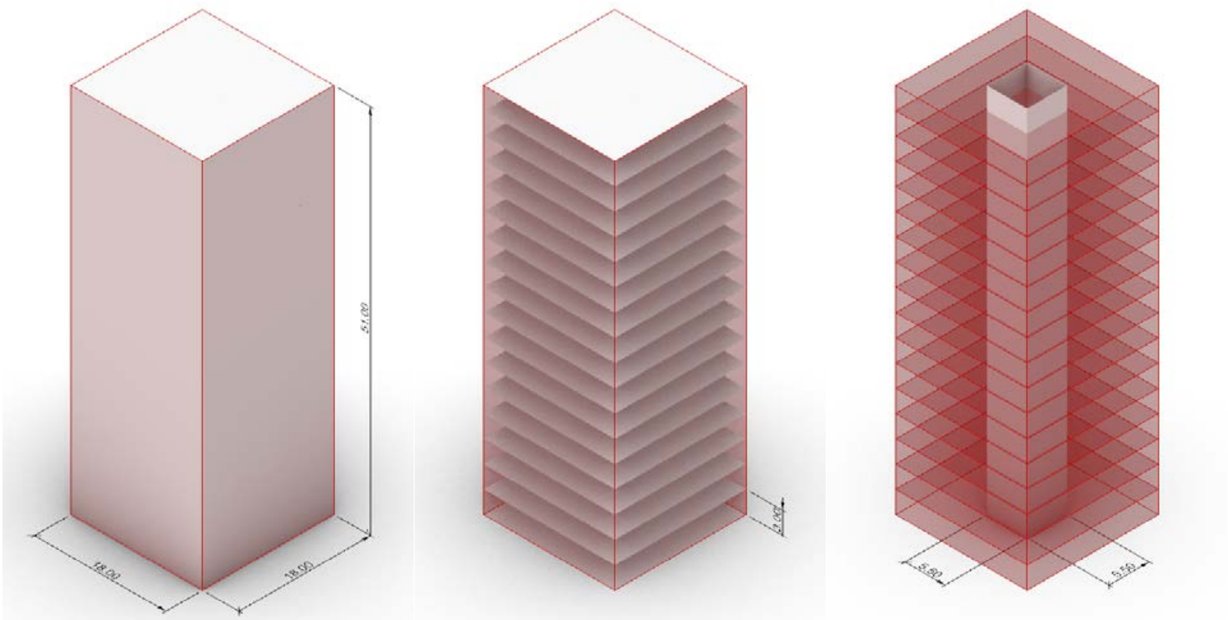


Figure 34: Base model

5.1.3.3 Location

Location is one of the main parameters to generate the conceptual mass since it defines the pop out generation through a genetic solver. To the purpose of the research the location is defined as a generic place in the city of Milan, Italy. Nevertheless, the script is defined so it is possible to maintain the accuracy in the results of the generative algorithm, even if a different location is chosen. The conceptual mass does not present any surrounding buildings since the purpose of the research is base in the procedure and do not size all the constraint possible in a city emplacement like neighbors' buildings. However, the script is built as the surrounding can be added in further stage of the research.

Milano-Linate_ITA:

- Latitude 45.43
- Longitude 9.28
- Time Zone 1.0
- Elevation 103.0

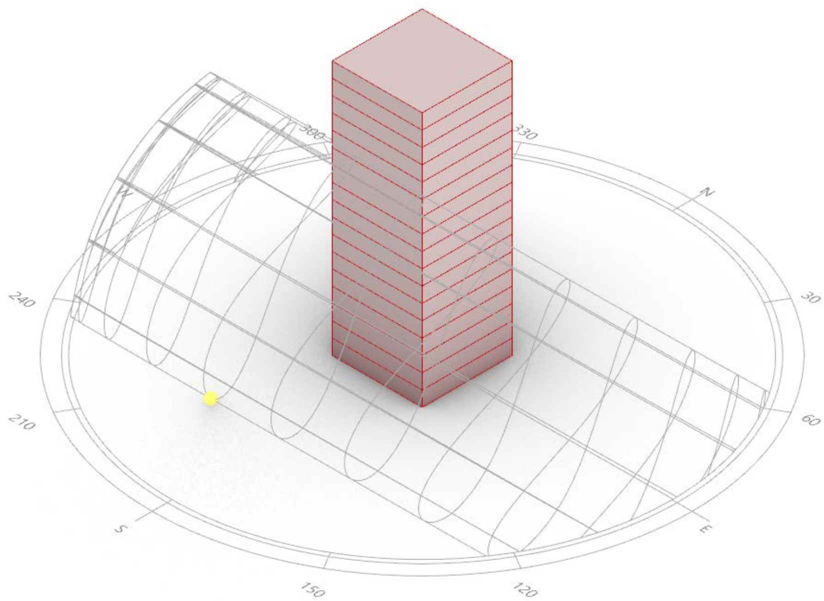


Figure 35: Sun Path and base model

5.2 Generative building massing optimization

In the first part of this case study, the model tries to considered the following characteristics from PD in architectural practice (Wortmann & Tunçer, 2017):

- Translation of design ideas into parametric models.
- Rationalization of designs into buildable shapes and components.
- Control and setting out of architectural forms.
- Generation and testing of design variants based on various criteria and specialist input, i.e., efficiency-focused design exploration or optimization.

- Capture of design knowledge from different stakeholders.
- Sharing of information.

At the core base of PD is data management and optimization of data results. Using genetic solvers, the script focus on the generation of data conscious decision for pop out number and apparently random distribution considering the location. Meanwhile the typology optimization responds to an stablish number of m^2 divided in different living typologies, in addition every typology should have at least the minimum required façade length according to its surface and number of rooms.

5.2.1 Pop out generation

5.2.1.1 Script Definition

The Pop out script is built to extrude all the boxes following the façade grid from the Level 1 to Level 16, and afterward select in an apparently random distribution the boxes that remains to keep the architectural concept of pop outs or pixels. The dimensions are the following:

- Pop Out footprint: 6m x 6 m
- Facade grid for pop out modules: 6m x 3m

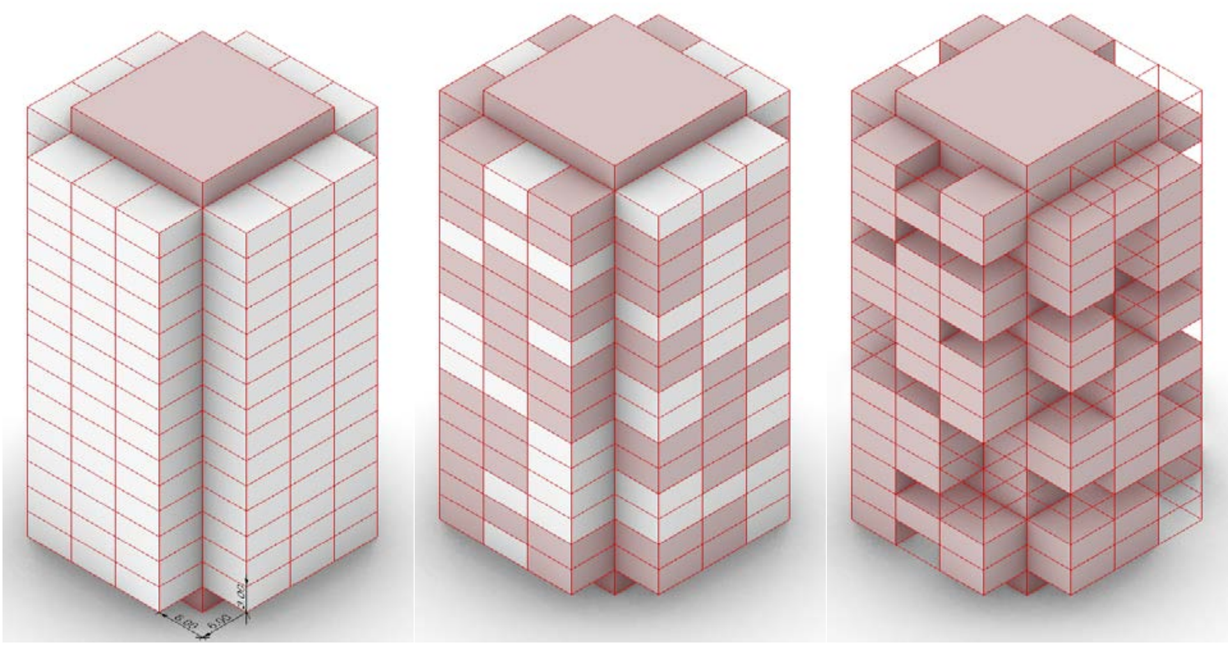


Figure 36: Pop-out generation.

Nevertheless, the position and the number of pop outs need to be optimized by a genetic solver. The case study consider that the optimization of a conceptual building massing is the solar energy over its façade. Solar radiation is one of the principal thermal gains in a building, therefore it has a great impact in further detailed studies like energy demand for heating and cooling. Hence, is important to weigh its influence during the different season of the year since it can naturally warm up the building during winter but may cause overheating in a summer period. The pop outs directly influence this parameter since they increase the total façade surface but can also act like a shading device for the building. Therefore, their position and number are inputs to a balance by a genetic solver due to the large amount of possible solutions and computing required.

5.2.1.2 Octopus solver

The Octopus solver is multi-objective evolutionary optimizer. It introduces the Pareto-Principle for multiple goals based on David Rutten's Galapagos solver and ETH Zurich SPEA-2 and HypE algorithm. Therefor it allows to search for a range of optimized trade off solution between the extremes of each goals or targets (McNeel, 2020).

The input variables to be analyzed to find an equilibrium between solar gains and shading production are:

- Pop outs number.
- Pop outs location.

In order to optimized conceptual mass of the project the octopus solver proccess the information of the total radiacion during the summer and winter period for the specific location of Milan. This means that two objective values are defiend by:

- Maximal solar radiation on façades during winter period from 1st November to 28th of February (kWh/m^2).
- Minimal solar radiation on façades during summer period from 1st June to 30th September (kWh/m^2).

From the Octopus solver is possible to make data conscious decisions of shape and performance. Octopus generates cartesian graph assigning to every possible variation in the pop out generation the total solar radiation over the façade for winter period in the y axis and for summer period in the x axis. Every result has and associated graphical representation of its final geometry in Grasshopper/Rhinoceros visual interface, therefore is possible to realize the impact in design and performance for each possible solution.

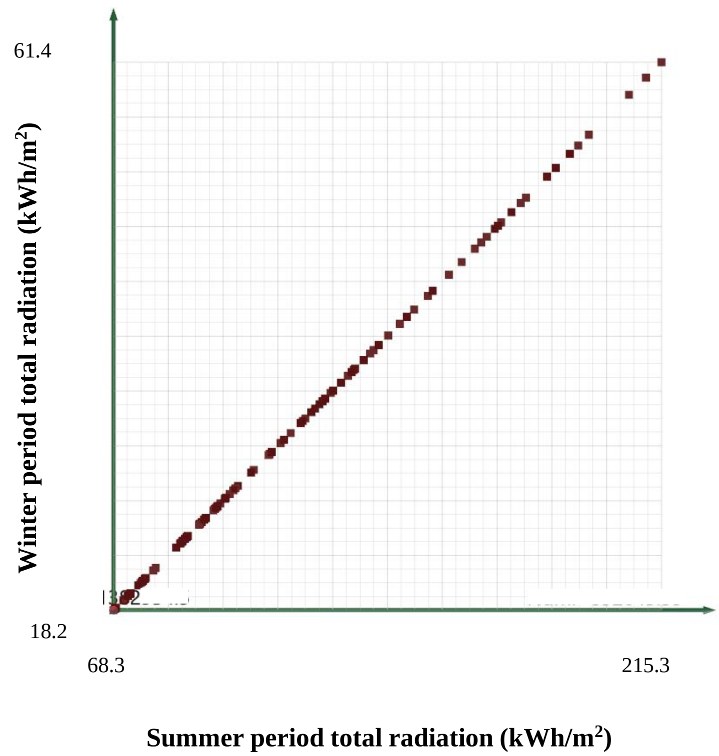


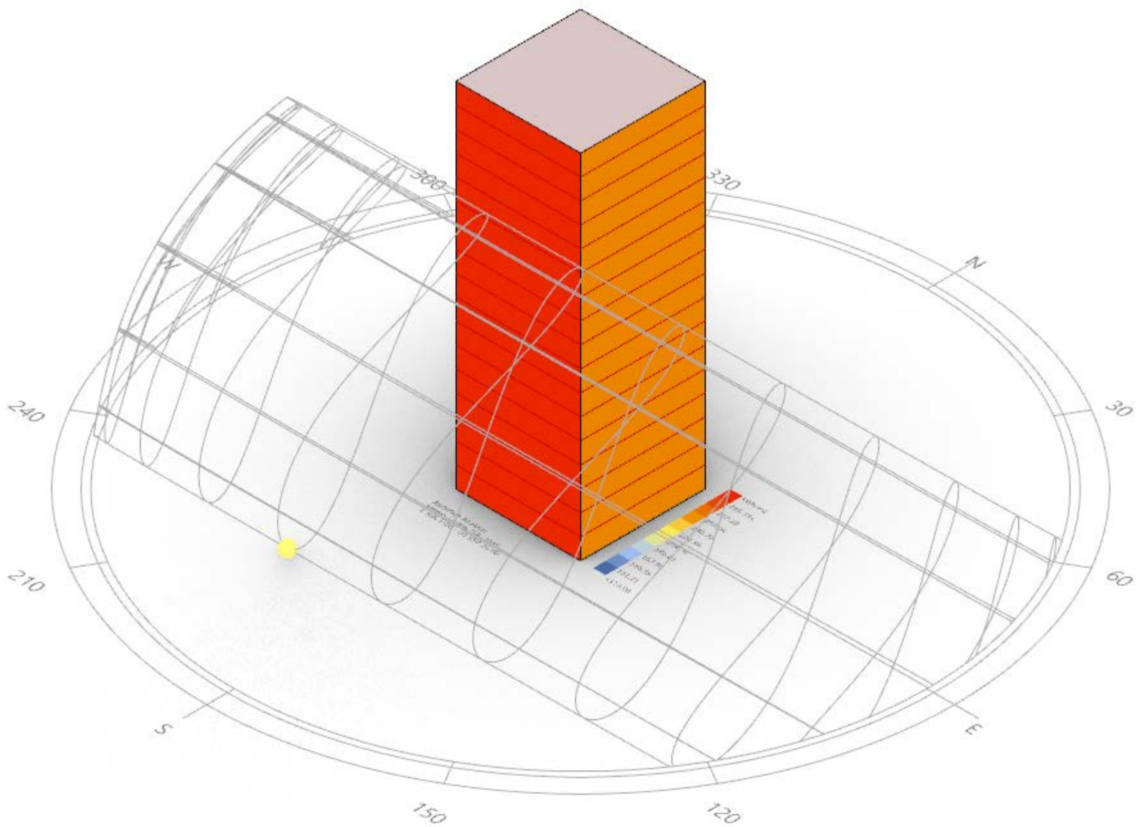
Table 3: Genetic solver solutions for total solar radiation over the façade surface .

The graph shows a proportional distribution between extreme goals due to the position and number of pop outs. Therefore, a higher total radiation over its facades during winter period also means a higher result during a summer period and vice versa. The difference in the unit of magnitude among the x and y axis answer to the amount of the solar irradiance during winter and summer.

5.2.1.3 Result comparison

To arrive to an optimized solution is necessary to contrast the generated results with each other, therefore the base model is compared with 3 different solutions with a different objective result.

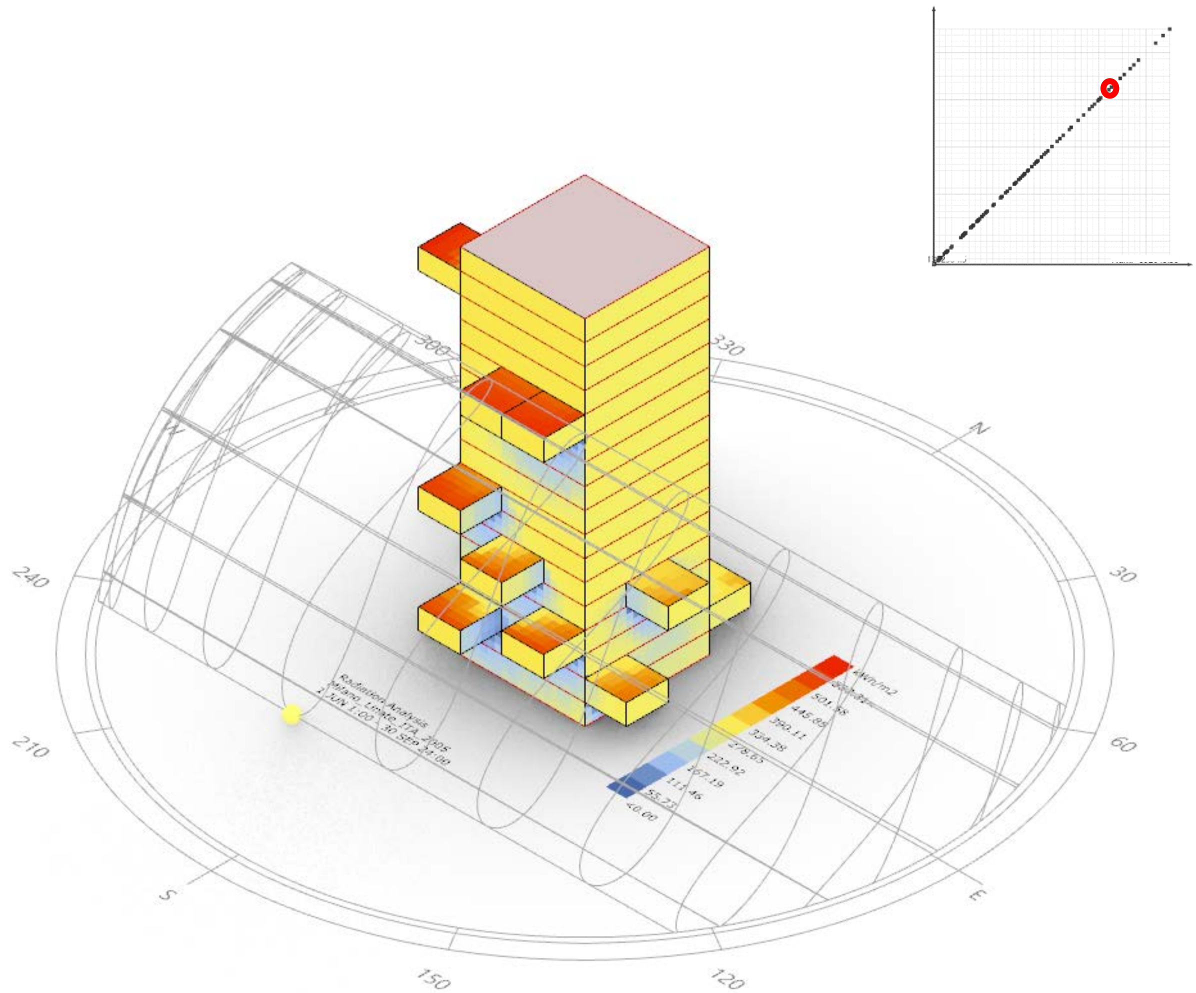
Base model:



Number of pop outs (#)	Winter period radiation over façade per unit area (kWh/m²)	Summer period radiation over façade per unit area (kWh/m²)
0	**65.5**	**226.8**

Table 4: Base model analysis results.

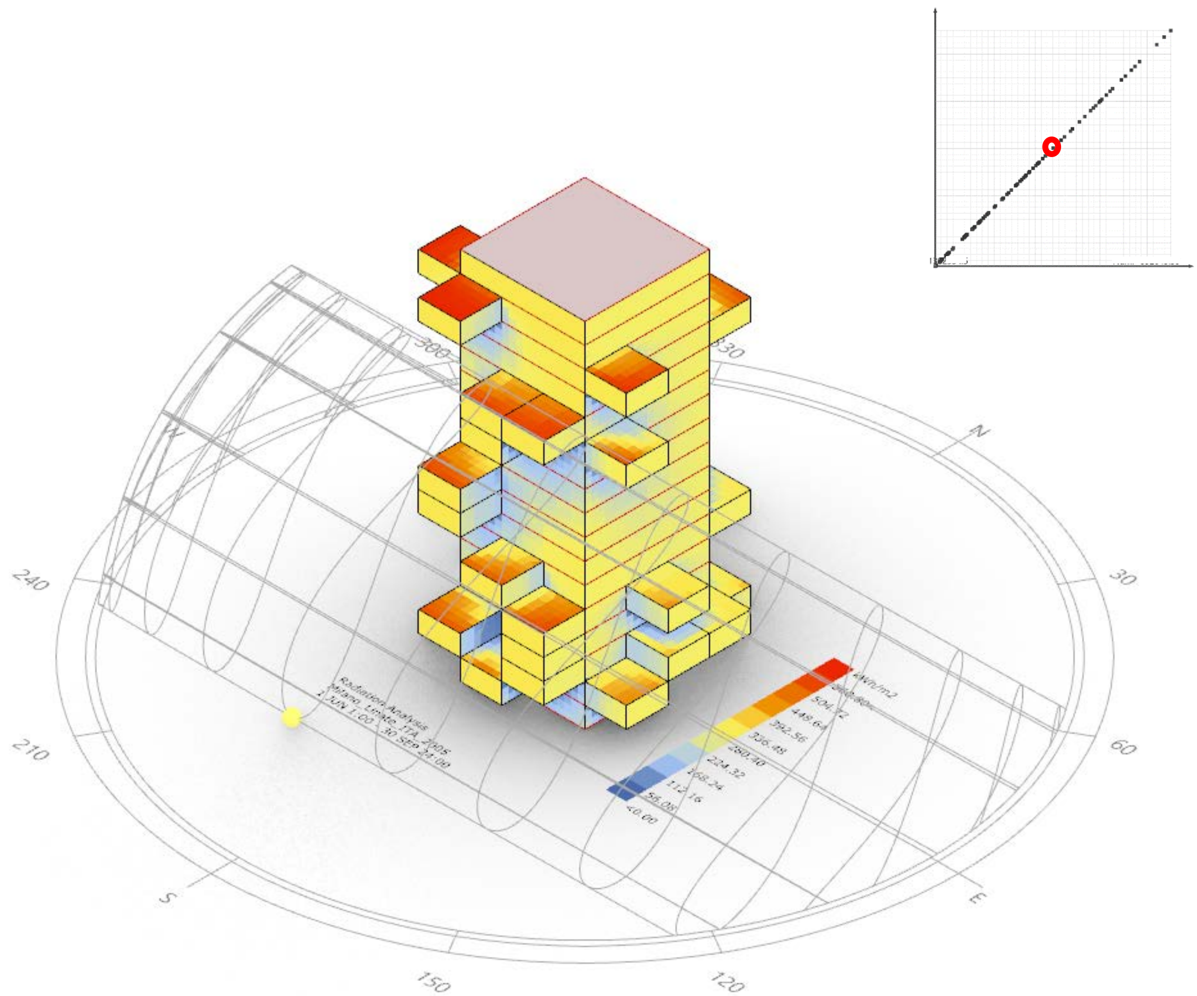
Option 1: The second one considers a lower radiation result for maximal radiation in winter period and the lower radiation result for maximal radiation in summer period



Number of pop outs (#)	Winter period radiation over façade per unit area (kWh/m²)	Summer period radiation over façade per unit area (kWh/m²)
13	**48.5**	**177.4**

Table 5: Option 1 analysis results.

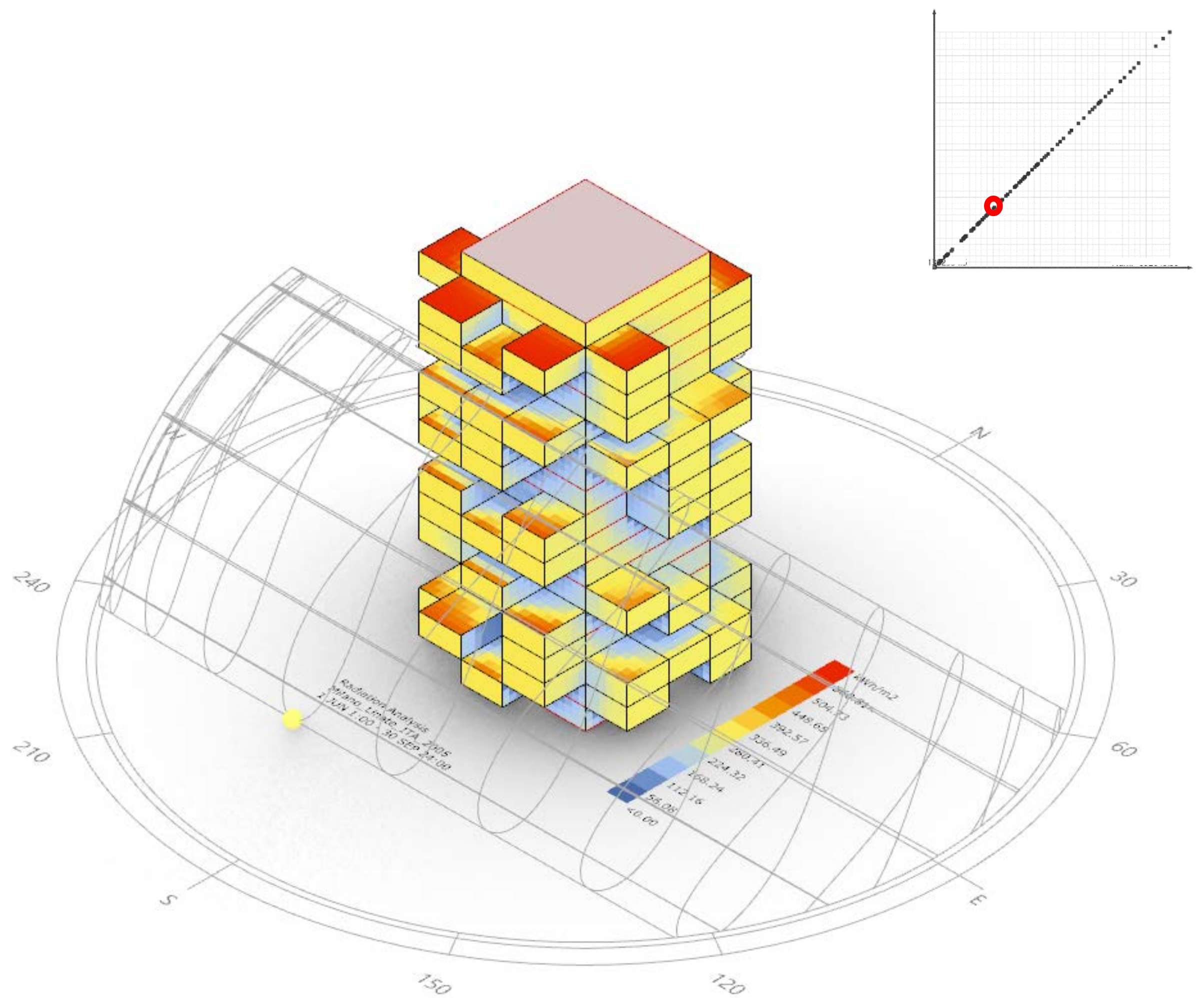
Option 2: The first one is the closest to a proportional result between maximal radiation in winter period and minimal radiation during summer period.



Number of pop outs (#)	Winter period radiation over façade per unit area (kWh/m²)	Summer period radiation over façade per unit area (kWh/m²)
34	**36.9**	**138.4**

Table 6: Option 2 analysis results.

Option 3: Finally, the third option analyze a configuration that supports a higher total radiation during summer period and higher total radiation during winter period.



Number of pop outs (#)	Winter period radiation over façade per unit area (kWh/m²)	Summer period radiation over façade per unit area (kWh/m²)
99	24.4	91.1

Table 7: Option 3 analysis results.

The base model presents the maximum radiation results during winter and summer periods due to the uniformity of the solar incidence over the façade. The façade of the base model does not present any geometry overcasting shadows over the building therefore, the solar exposure will be maximum at any period of the year. This result even if beneficial to maximize solar gains is not optimal during summer season since the objective is to reduce external and internal thermal gains. Considering this logic, every option studied by adding pop outs reduce the performance of solar gains during winter and summer. The objective then is to weigh the reduction of solar gains with the shading generated by the pop out number and disposition. Compared to the base model:

Solar Radiation over façades in kWh/m2 compared to base model			
Period	**Option 1**	**Option 2**	**Option 3**
Number of Pop Outs	13	34	99
Winter	74%	56%	37%
Summer	78%	61%	37%

Table 8: Comparison table.

From this chart is possible to conclude that a higher amount of pop outs represents a higher reduction of solar radiation over façades compared to base model. Pop outs, as they are defined, work as an efficient shading for summer but its dimensions are not suitable for winter since there is also a high radiation reduction during winter that can be useful for the cold period. A further step should be adding as a variable parameter the pop out footprint dimension and measure its impact.

5.2.1.4 Final volume

To support that final decision of building massing and considering the data acquired through the generative parametric process a final assumption is assumed:

- In a residential building, thermal loads that helped balance winter conditions are a naturally produced by the solar irradiation and the intrinsic occupancy of the building, while on the other hand, subtraction of heat from a space during summer is not a natural process if external conditions are warmer than inside.

Therefore, the final building massing favored a configuration that reduce the impact of solar gains during a summer period, instead of a high solar gain reduction during winter. The final building massing to continue this case study is a negotiation between these two constraints and try to find the biggest gap possible between this parameter without a drastic reduction of solar gains during winter period. Therefore, the option chosen to continue the case study is:

- **Option 2**

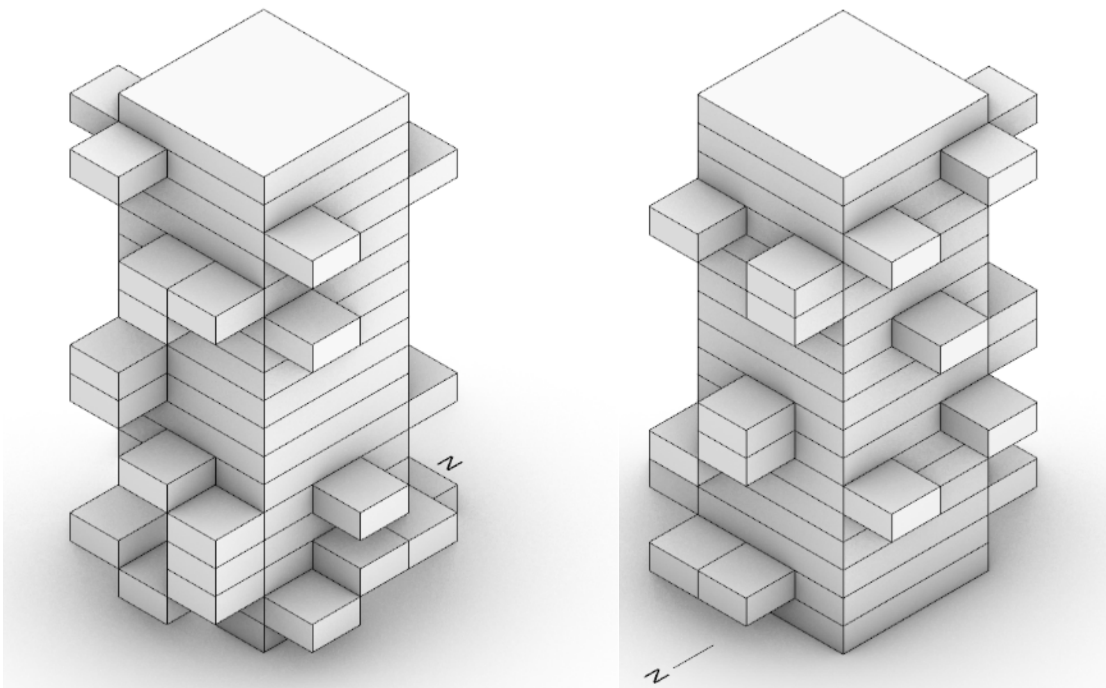


Figure 37: Final building massing.

Thanks to the genetic process is possible to have a conscious decision of the percentage of solar gains losses during winter that the designer is ready to compromise and to the final aesthetic result of the volume. In addition, iteration between all possible solutions generated by the solver still available, expanding the options for a possible new solution to be discuss between the client and the designer.

5.2.2 Typology distribution generation

Furthermore, this case study adds the internal typology generation into the building massing with the following purpose:

- Recreate a stakeholder demand of distribution of the typologies into a residential building.
- Diversify the elements to be included into a parametric BIM workflow with the addition of rooms elements.

The typology distribution is the internal division of each floor in the final volume. In a residential building every floor is divided in different apartments and its typology is defined by the number of rooms. Every typology has a minimum surface to be a functional space. From experience, the case study considers typologies and areas as:

- T1: studio - 25m^2
- T2: 2 rooms - 45m^2
- T3: 3 rooms - 65m^2
- T4: 4 rooms - 85m^2
- T5: 5 rooms - 105m^2

For the generation of the typologies distribution the next assumption is defined:

- The commercial success of a residential project is related to the diversity of its typology since it reaches a wider spectrum of users. From individuals to short and large families.

Therefore, the typology distribution must achieve a great diversity of apartments and respect, as close as possible, the surfaces assigned to each typology. The case study recreates a construction industry reality, where small areas are an uninhabitable space, and large areas are not rentable for a Stakeholder. Since, it is a conceptual phase, these areas consider internal partitions and small shafts, therefore it is important to notice that further deductions should be done to find the real livable area.

Once the distribution is achieved by the genetic solver a second parameter is considered to evaluate the efficiency of the genetic solution. Each typology should consider a length of façade to assure that its livable spaces can be naturally lighted. From experience, the minimum length for every typology to be verified are:

- T1: studio - 3m
- T2: 2 rooms - 6m
- T3: 3 rooms - 9m

- T4: 4 rooms - 12m
- T5: 5 rooms - 15m

Due to the complexity of the project the genetic solver may not able to generate all the typologies with a suitable area and facade length. Nevertheless, is a tool that simplifies the early conceptual tasks and decision. Designer's intervention is to evaluate and refine the solver's results and complete the design process where the programming script produces any flaw in the desired final solution.

5.2.2.1 Script definition

The objective of the script and genetic solver is to arrange the maximal number of different typologies in the entire project and verify its inhabitability and rentability. The division of every typology is defined by the internal walls position aver the entire floor surface. In addition, the internal wall keeps a vertical continuity so they can be useful, if needed, for a structural purpose.

Considering the difference of floor plans between each other because of the pop out's apparently random distribution and due to the internal partition vertical alignment, the typologies distribution from one floor to the next one is different. To find the suitable answer of the are distribution the following assumption is make:

- The core of the building connects all levels and all the apartments directly, therefore all the internal walls are connected directly to the core walls.

As a result, the efficiency of the script is based on the right position of the walls around the core walls to obtain the maximum number of typologies with a suitable area.

5.2.2.2 Galapagos Solver

Galapagos is generic solver developed by David Rutten in 2012. In combination with Grasshopper it can replace a manually conducted optimization, since genetic algorithms have a goal task through multiple objective optimization (Mirtschin, 2010). A generic solver founds a solution to a problem that can be express mathematically, although these answers may not be exact, they are a considerable useful approximation to a final solution. (Rutten, 2013)

Galapagos works with input genes or variable parameters which automatic combination pursue a final goal objective. Therefore, is necessary to set the objective and recognize the input genes and translate the goal into a mathematical logic.

In this case, the goal is to have as maximum typologies possible, thus, the following objective goal is set:

- Every floor plan should have at least one of the five typologies or more when the surface of the floor plan area is bigger.

To achieve this the genes are set as the following:

- The position of the walls over the total length of the core perimeter.

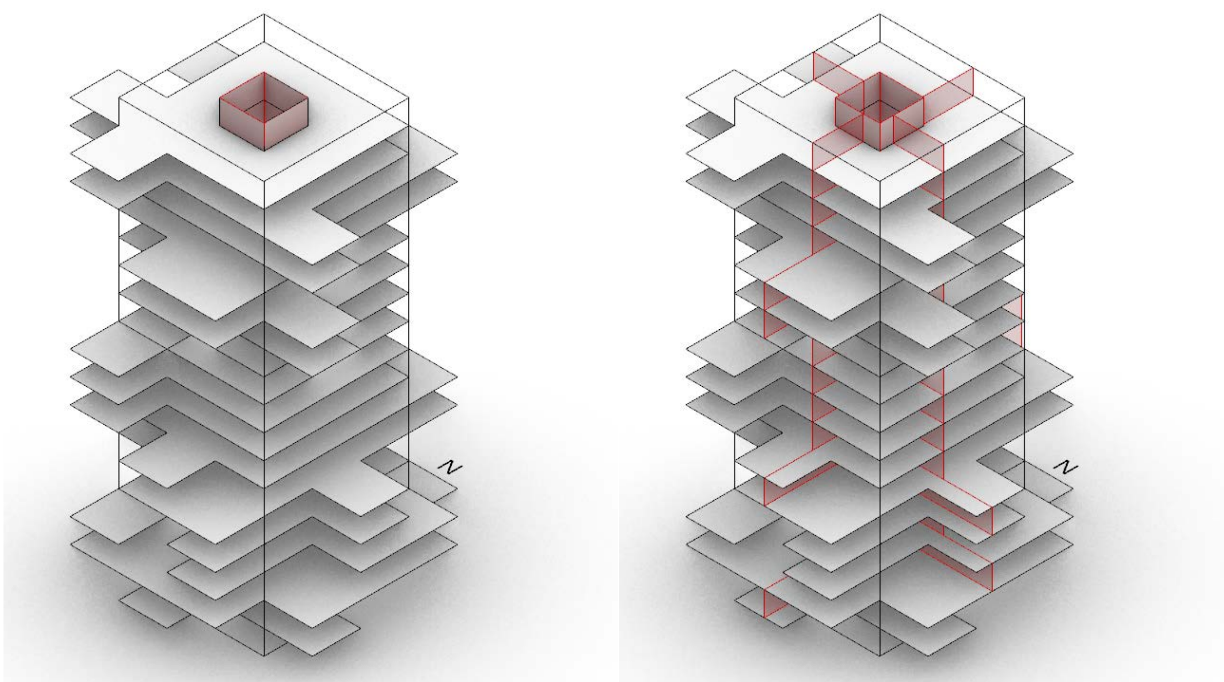


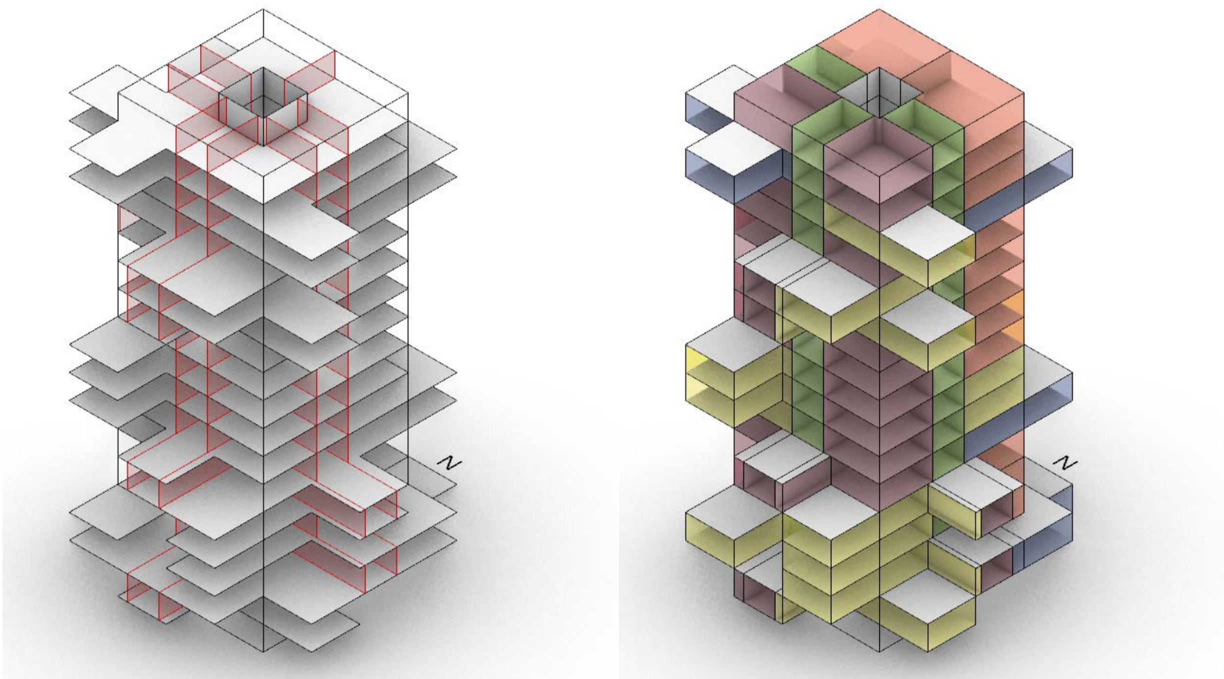
Figure 38: Gene elements considered for the genetic solver.

Finally, the mathematical relation of the problem is translated as:

- The difference of the considered area for each typology in this case study (25m^2, 45m^2, 65m^2, 85 m^2, 105 m^2) and the area of each surface generated by the position of the internal walls around the core of each floor. Closer this difference goes to 0, more accurate the areas generated by the displacement of the internal walls is to the objective.

The Galapagos solver will iterate among all the possible combination of the internal wall positions around the core always trying to minimize the goal objective and get as close to 0 as possible.

Galapagos optimized genetic solution is:



Typology	**T1**	**T2**	**T3**	**T4**	**T5**	**Total**
Number	41	27	18	17	9	**112**
Area accuracy	100%	95%	100%	97%	100%	**98%**

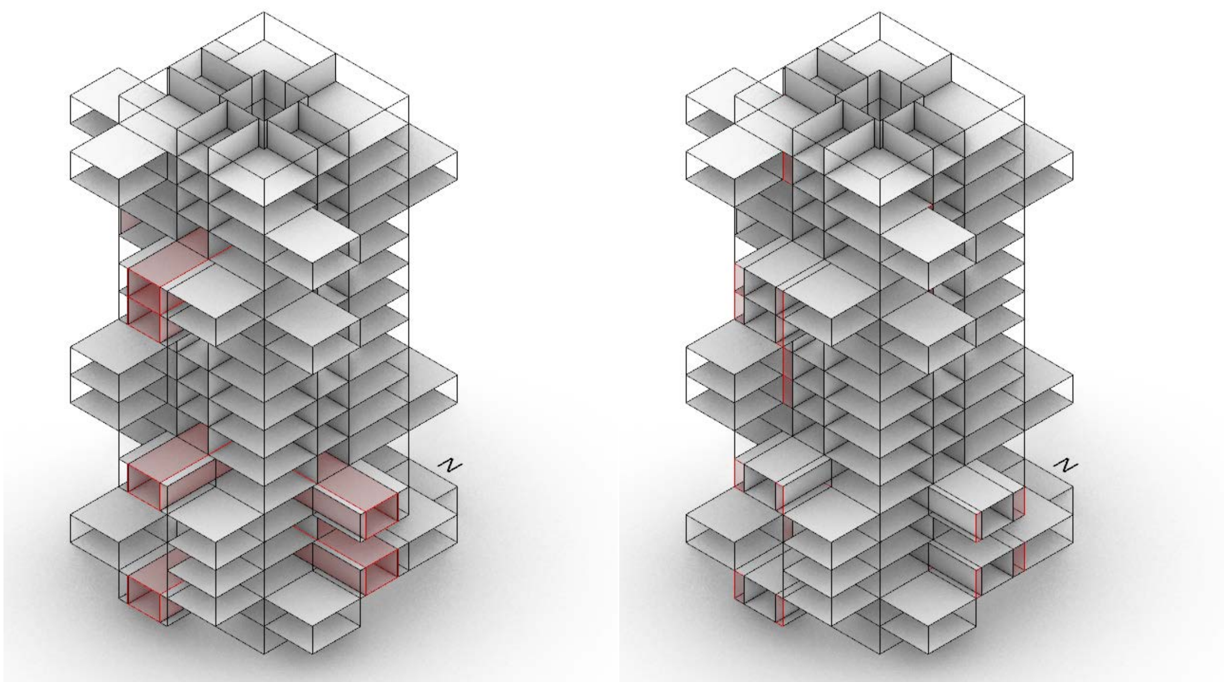
Table 9: Genetic solver final distribution.

5.2.2.3 Facade verification

Furthermore, once the typologies are set by the genetic solver besides its surface the script evaluates the façade length in ordered to assure the inhabitability of the apartments generated. To this purpose two measurements are considered:

- Façade length for each typology should not be less than the minimum length stablished for this case study according to the number of rooms (3m, 6m, 9m, 12m, 15m).
- The minimum length to be considered for a segment of façade in each typology should not be less than 2.5m. From experience, a room with a shorter width is difficult to arrange or uninhabitable.

Regarding this condition we can verify which typologies fulfill the requirement and assess any future modification. To calculate the efficiency of the scrip and the genetic solver solution a comparison of the invalid typologies and faces with the totality of elements produced has been done.



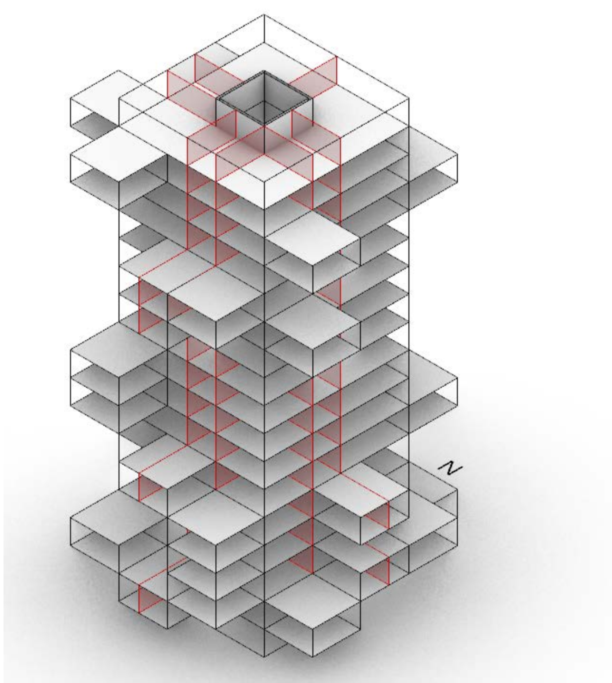
Typology < min façade	%	Typology façade side < 2.5m	%
7	**6.25**	28	**10.37**

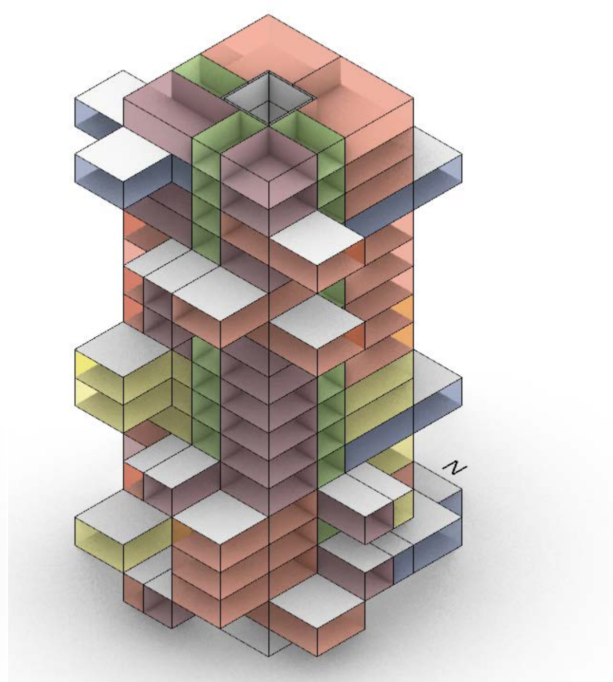
Table 10: Façade length analysis.

Overall, the typologies compromise by its façade length is a small percentage of the total typologies created by the genetic solver, the same as the façade sides that are smaller than 2.5m. Nevertheless, this parameter constraints in a great measure the quality of the space, therefore the parameters concerning these typologies should be verified by the designer and modify weighing the impact in globality of the project.

5.2.2.4 Final Distribution

As said by David Rutten a genetic solver gives a close answer but may not find an exact one. In a complex building a genetic solver will answer to a logic mathematical algorithm but this may not include all the inhabitable parameters. Therefore, the genetic solution result parameters have been adjusted by the designer criteria. The capability of a parametrical script created for this purpose allows an intervention of the designer to solve punctual issues in the internal typological distribution. Therefore, the final distribution shows some small modification to increase the efficiency of inhabitability already calculated by the genetic solver. A few walls have been parametrically moved to increase the façade length to the minimal required and adjust better to the area required for each typology.





Typology	T1	T2	T3	T4	T5	Total
Number	41	23	28	11	9	**112**
Area accuracy	84%	94%	93%	98%	95%	**93%**

Table 11: Final typologies distribution.

Typology < min façade	%	Typology façade side < 2.5m	%
1	**0.89**	0	**0**

Table 12: Final facade verification.

Compared to the genetic solver solution there is a reduction on the area accuracy. The genetic solver focuses strictly in approaching the desired area as the only objective, meanwhile the wall modification obeys more to the habitability of the space regarding its possible daylight exposure and inhabitability compromising the surface accuracy set at the beginning. This decrease in accuracy represents a reduction or increase of the actual apartments area in the current model. Further verification will be the final room surface schedule which is expected to be computed by the BIM software, because of the parametric BIM workflow.

5.2.3 Conclusions

Parametrical design and genetic solvers are capable to enhance automatization at a conceptual design stage. Thanks to these two approaches it is possible to generate a building massing and iterate through a considerable amount of geometrical variations without a 2D manually participation of the designer.

The architectural concept is closely link to the performance of the building. Thanks to the analysis of the genetic results is possible to assess that the pop-out concept didn't have a considerable impact in the target results. It was not possible to achieve a big difference between total radiation over façade surface during summer and winter, both results where closely related. Furthermore, the pop-out solutions increase the volume to surface radio since they are considered internal spaces. Therefore, it is important to consider the thermal transmittance of this new elements to balance its role in the energy consumption performance of the building.

Computational design can process a big amount of information, but the designer must evaluate the results. Design involved idiosyncratic and abstract decisions that requires a human intervention (Aish & Woodbury, 2005). Therefore, even if assisted by the computational calculation, the designer needs to take the final decision. In this case study, the genetic solver produced a high accuracy for surface layout and distribution with 98%. Nevertheless, some of those spaces were not inhabitable, the human intervention to make those typologies livable spaces reduced the accuracy to 93%. The designer role is to find the compromise between computational generation and human needs and find alternative solutions for the surfaces that are out of range.

5.3 Parametric BIM Workflow

5.3.1 BIM model definition

This chapter evaluates the different ways to consolidate a parametric BIM workflow from the object base parametric software (Grasshopper) towards a BIM platform (Revit). The model defined in the previous chapter need to be stream directly into a BIM universe to manage further interoperability with engineers, client, and designers. The objective is to measure the accuracy and capacity of different workflow methods available now days.

Different ways of communication between parametric modeler and the BIM platform are studied separately and define a different degree of coupling between the PD model and BIM model. The grasshopper extensions use in this case study are:

- Geometry Gym for IFC export (Loosely coupled model)
- Hummingbird for CVS Data Transfer (Tightly coupled model)
- Speckle for Cloud base (Tightly coupled model)

They work as standard grasshopper nodes added in the base library, parameters are input and are modified at the end of the process. Every extension pushes the same amount of information from Grasshopper into the BIM environment and the same result of definitions and detail is achieve after every single transfer. At the end of the workflow, graphic base geometry introduced is automatically transformed into Revit native elements. Basic geometry information like points coordinates, line lengths, curves shapes are assigned a construction element specification, therefore on the other side of the stream this element will have information like function, thickness, material layers, thermal properties, etc. Additionally, some information must be added into the Revit model for a correct interface and visualization of the model in the BIM platform like levels and tags. Finally, schedules are created to verify the quantities and measures of the information transfer from one platform to another.

From Grasshopper, the final generative building massing is the following:

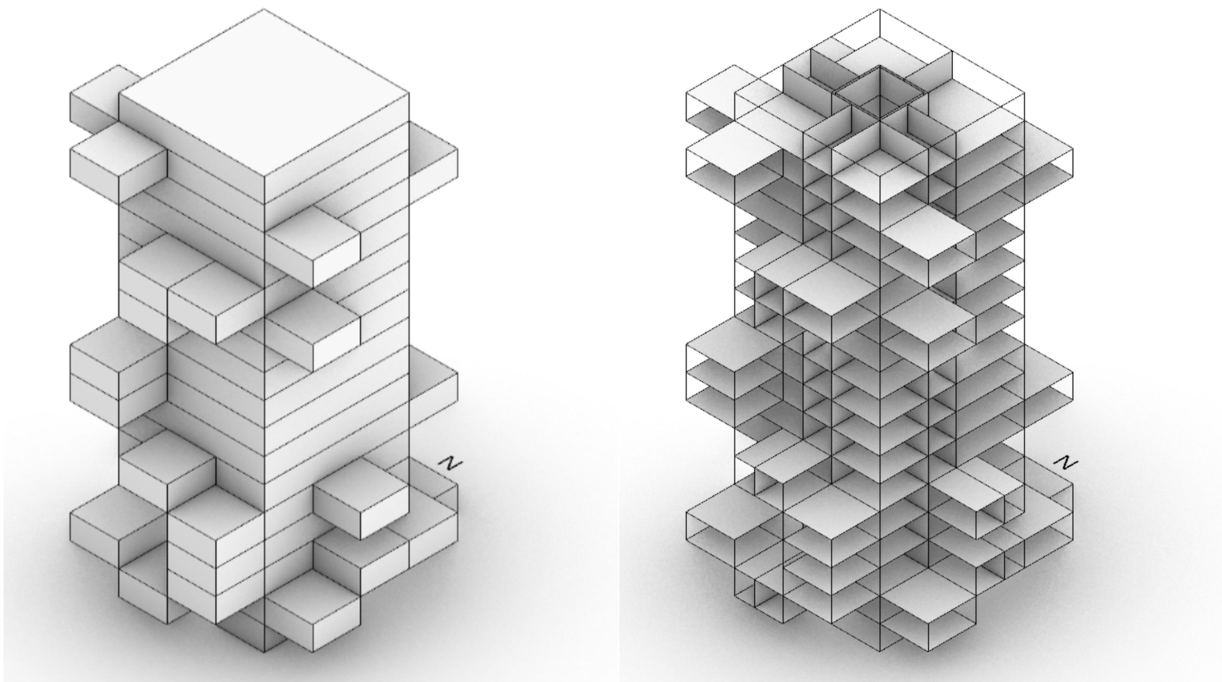


Figure 39: Model to be coupled.

Once the model geometry has been set, the amount of definition of its elements needs to increase. The objective is to build a conceptual model that can be easily push back and forth from PD to BIM with a certain definition so it can be process and easily move forward form the stage 2 of design (Conceptual) to stage 3 (Development).

To increase the levels of detail of every geometry defined in Grasshopper a Revit family is assigned according to its function with a specific type that defines its parameters. Types are selected from the Revit internal library. Since a Parametric BIM workflow is stablished, initial types can be easily change at the beginning, once the coupling models are being defined, or, once the native elements are created, directly in Revit. The native elements create in Revit are:

- Slabs: 160mm Concrete With 50mm Metal Deck

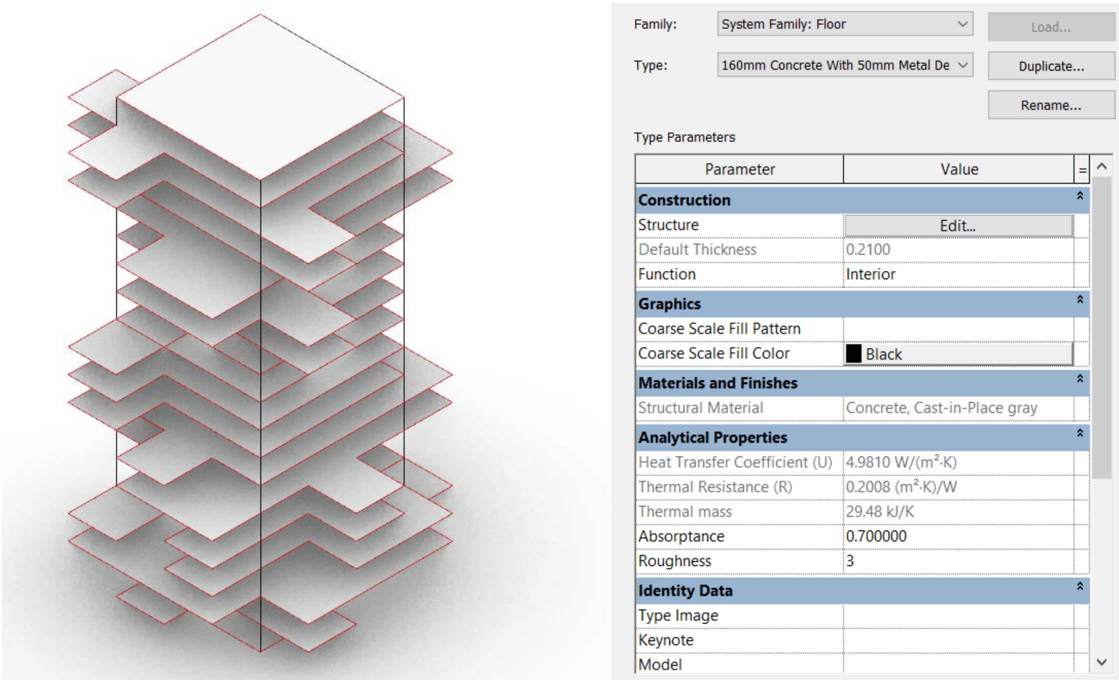


Figure 40: Slabs definition.

- Core walls: Core - 250mm Concrete

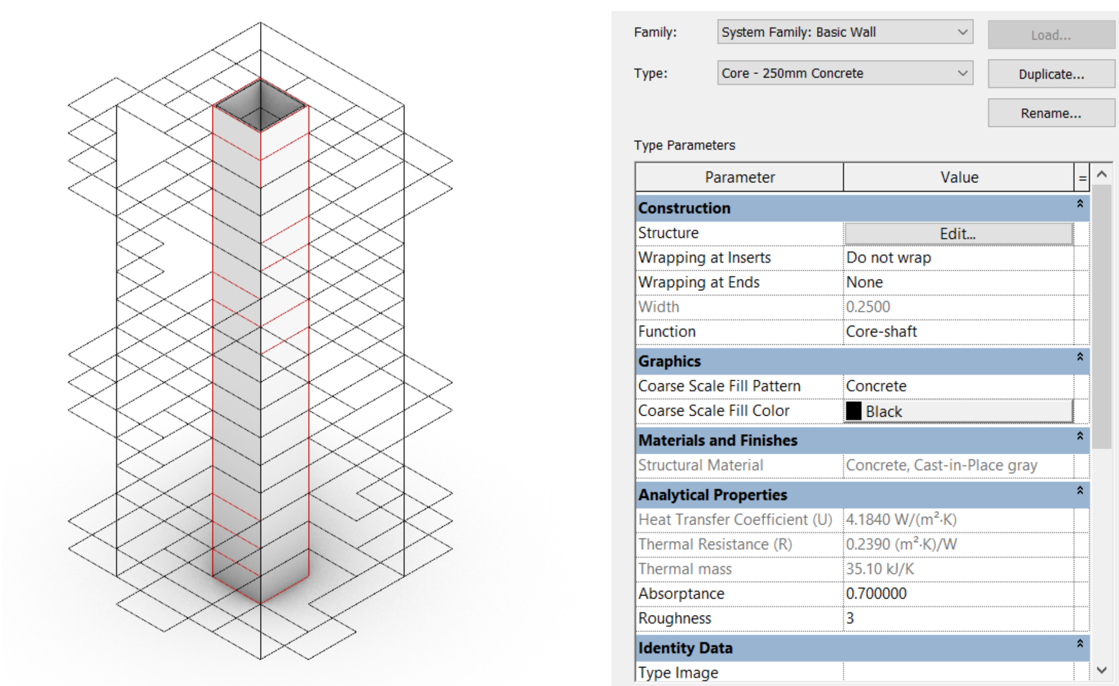


Figure 41: Core walls definition.

- Internal walls: Generic - 150mm Masonry

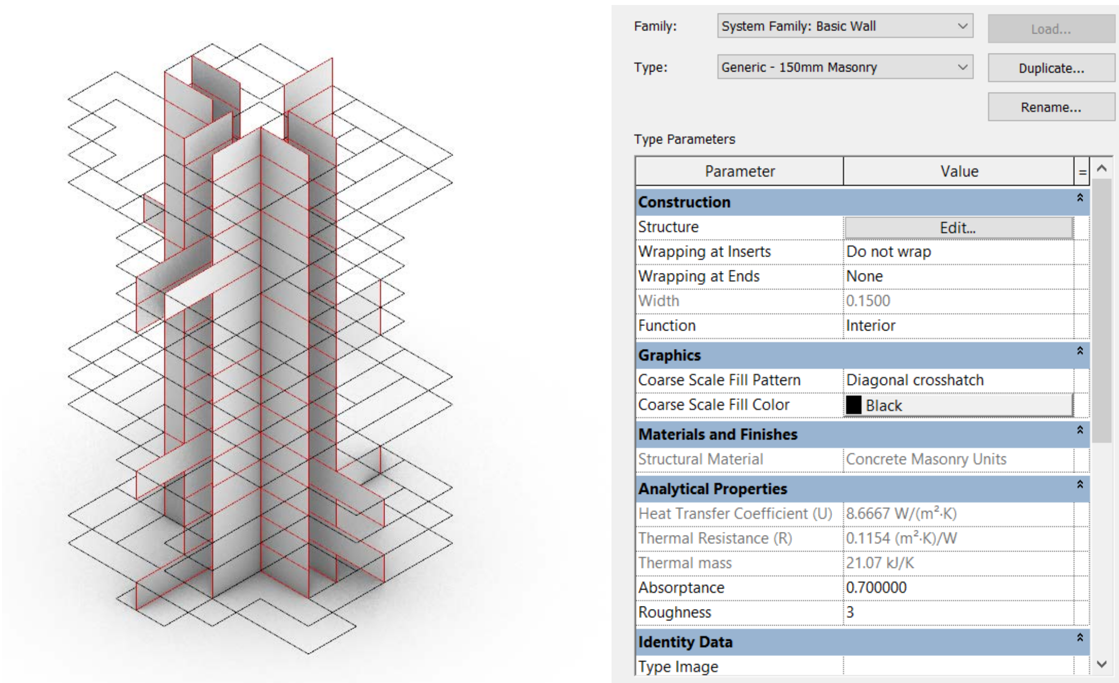


Figure 42: Internal walls definition.

- External walls: Exterior - Brick on Metallic Stud

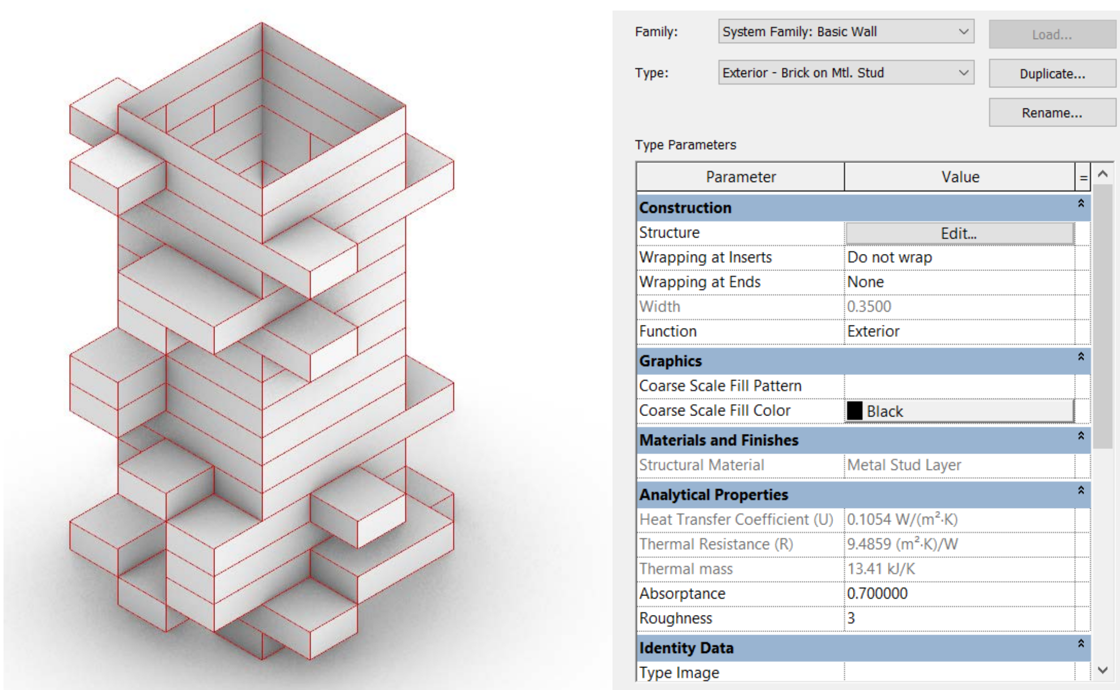


Figure 43: External wall definition.

Once the parametrical BIM workflow is stablished every extension is expected to consolidate a BIM conceptual model that can easily be manage and update in Revit and from Grasshopper.

The final 3D visualization of coupled models is:

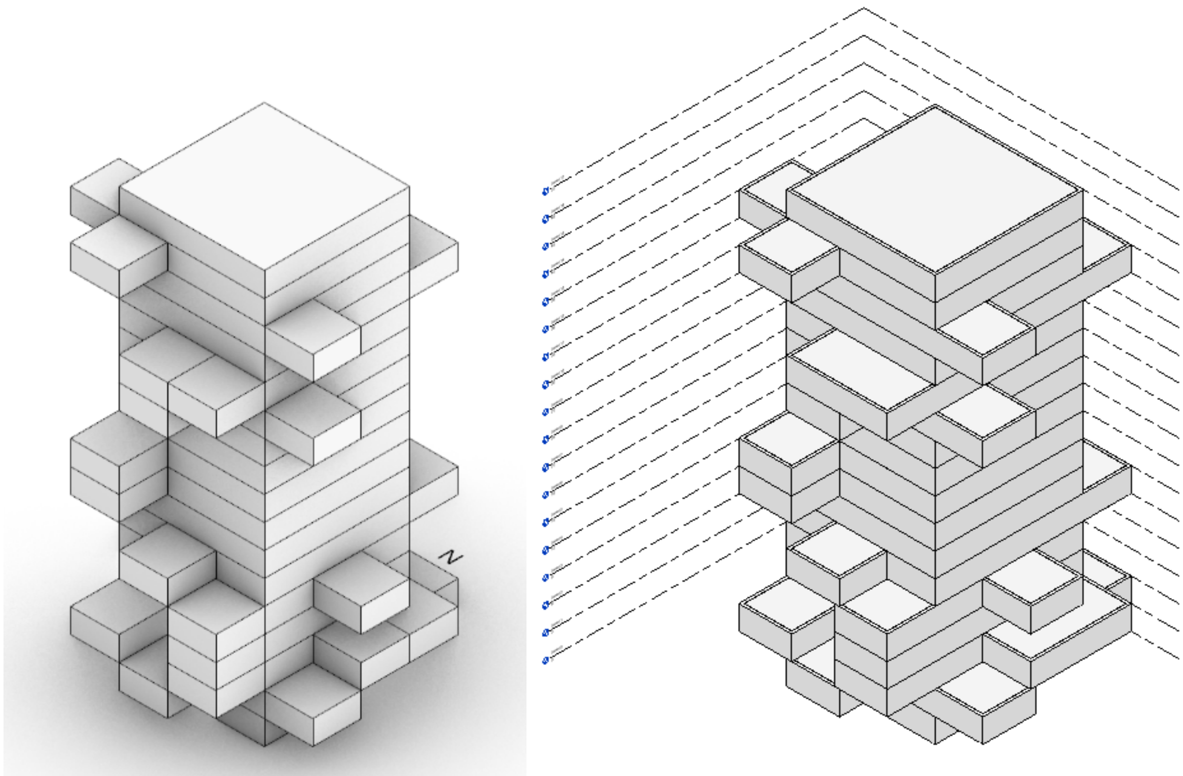


Figure 44: Parametric and BIM models

Furthermore, a parametric BIM workflow considers more elements than solid geometry. From the parametric environment is possible to transfer basic parameters that are translated into a BIM as support elements for the proper organizations of the model or basic information for the visual representation of the model. This case study adds the following information to be properly transfer and defined into the parametric BIM workflow:

- Levels: Basic information since most of geometric elements are associated to this parameter. It contains the height of each different level of the project.

- Shaft: Void geometry that is used to trim a solid element in the model. This parameter is represented by contains a regular prism.
- Rooms: Virtual box that is bounded by the model elements. It provides information of the space definition like area, volume, name, use, etc. In this case study the information transfer for this element is a point inside every space created by the internal walls and its typology name.

All this infromation added to the conceptual model defitnion supports BIM features of visualization and can easily be organized and represented on level plan view. Neverthless, not all representation elements can be transfer from PD. For example, to achive this presenation level Tags have been added directly into Revit.

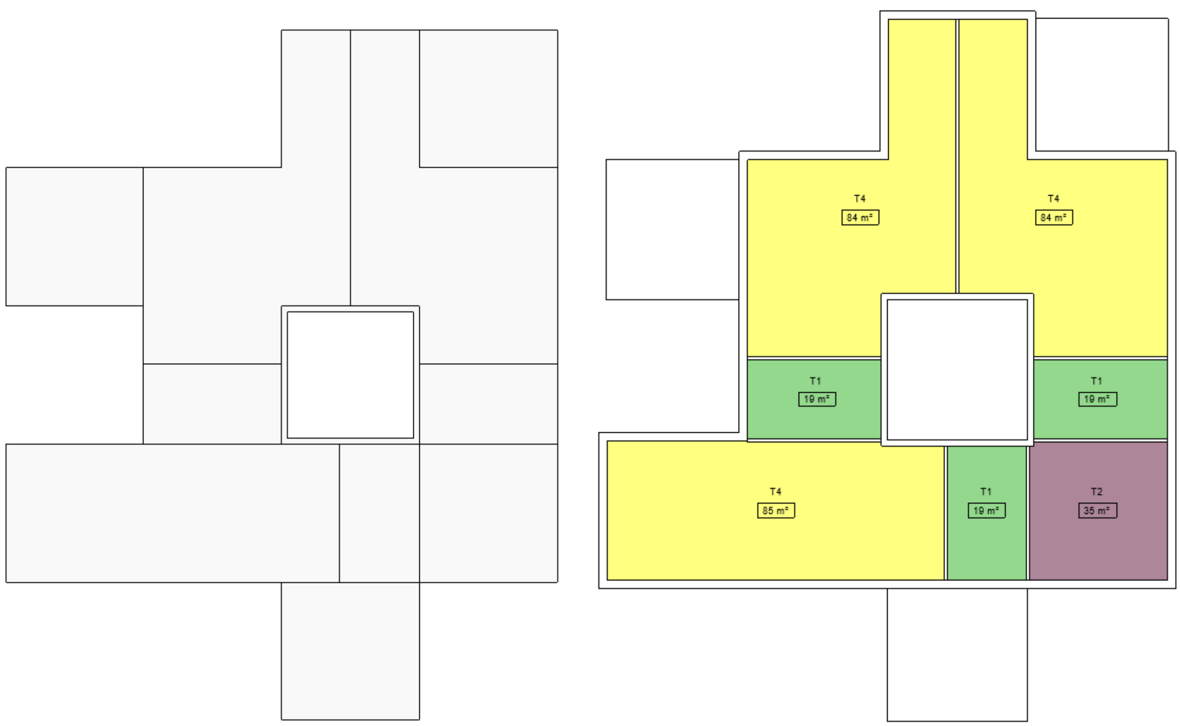


Figure 45: Parametric and BIM plan view, Level 6

Finally, once the parametrical BIM workflow is completed of all the elements definitions are directly stream into Revit. Thus, all elements contain valuable information that is possible to organize, group, measure, count and categorized though schedules in the BIM environment. In this case study schedules supports a rapid assessment and the evaluation for the typologies created by the genetic solver:

- Different typologies per level:

Level 6	
T1	19 m²
T4	84 m²
T1	19 m²
T1	19 m²
T2	35 m²
T4	84 m²
T4	85 m²
Level 6: 7	344 m²

Level 7	
T1	19 m²
T4	84 m²
T1	19 m²
T1	19 m²
T2	35 m²
T4	84 m²
T4	85 m²
Level 7: 7	344 m²

Table 13: Typologies per level

- Total typologies distribution:

T4	
Level 1	84 m²
Level 2	85 m²
Level 4	81 m²
Level 6	84 m²
Level 6	84 m²
Level 6	85 m²
Level 7	84 m²
Level 7	84 m²
Level 7	85 m²
Level 8	85 m²
Level 10	81 m²
T4: 11	919 m²

T5	
Level 1	120 m²
Level 2	117 m²
Level 5	102 m²
Level 5	102 m²
Level 12	102 m²
Level 13	102 m²
Level 13	102 m²
Level 14	102 m²
Level 15	121 m²
T5: 9	969 m²
Grand total: 112	5481 m²

Table 14: Typologies distribution

Even if the result of each parametrical BIM workflow must arrive to same definition from into Revit, all these levels of information are stream in a different way by each Grasshopper extensions. Every extension has a different level of complexity, therefore is important to stablish the amount of information that is going to be transfer from the PD environment to BIM. For this case study this information and detailed level is considered the minimum to have a clear view of the potential of a parametrical BIM workflow at conceptual stage of the project. It is important to also consider the way models are coupled (Tightly or Loosely) since this defines the flexibility of the interaction between the two design approaches.

5.3.2 IFC Information transfer: GeometryGym

GeometryGym is a Grasshopper and Revit extension that is automatically installed in both software for a loosely coupled parametric BIM workflow. At the parametric graphic base entry, it enriches the simple geometry elements information to consolidate an IFC file. Once the required data is introduced in its node the information needs to be “backed” to create the transfer file.

Table 15: GeometryGym’s bake command

Once backed, at the other end of the stream it translates and decompress the IFC file to create a native geometry in Revit with all the information already assigned in the parametric modeler. The benefit to have its own add in on the Revit software is that the information is read and rewritten with highest accuracy than just by a simple export and import action. Therefore, all information added in the parametric model is correctly represented and assigned in the BIM environment, reducing time consuming mistakes and rearrangements, usually present, after traditional export of an Industry Foundation Class file.

- Grasshopper Interface

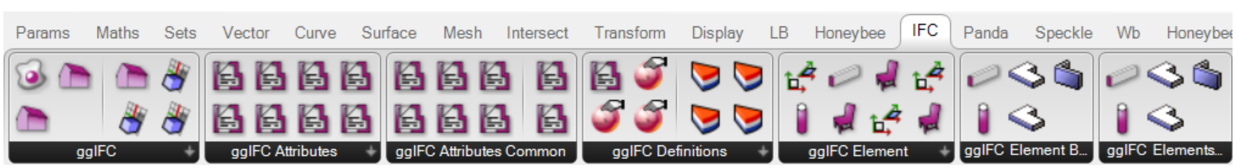


Figure 46: GeometryGym’s Grasshopper library.

- Revit Interface

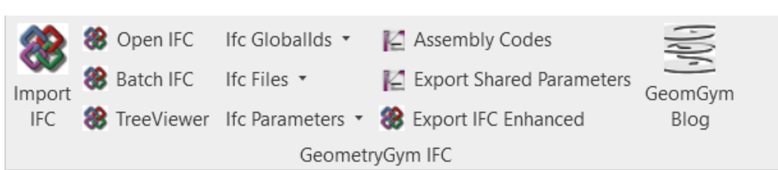


Figure 47:GeometryGym’s Revit commands.

Furthermore, an advantage of GeometryGym and a loosely coupled approach is that since the exported file is already in an exchangeable format in the BIM environment it can also be used in other software compatible with IFC standards.

5.3.2.1 Model set up

Project information:

In the case of GeometryGym, since it creates an IFC file some details should be added even before the geometry of the project. To create an IFC project information concerning the building description should be added like the Global ID and project name. This information is the host of other transfer elements. These information needs to be plug in the nodes called:

- IFC Building
- IFC Project

Levels:

Levels are created by the node IFCbuilidngStorey and the data input is:

- IFC Building: Data from previous IFC Building node.
- Name: Simple flatten text list with the name of each level.
- Elevation: Simple list of numbers whit a series of number corresponding the height of each. This list length and structure should match with the Name list.

Slabs:

Slabs are created by the node IFC Slab Standard Case and the data input is:

- IFC Host: Simplify and grafted data list from previous node IfcBuildingStorey node.
- IfcSlabType: Contains all the information that defines the type of slab. This data is collected in the following series of nodes:
  - IfcMaterial
  - IfcMaterial Layer
  - IfcSlabType

- Planar Slab Perimeter Cure or Face: Simplify and graft list of polyline curves defining the slab shape.

Shafts:

Slabs are created by the node IFC Opening. No compliant IFC will be created for this node, therefore the information generated is only recognize by GeomoetryGym extension in Revit. The data input is:

- IFC Container: Data from previous IFC Building node.
- IFC Representation Item: Contains the geometric information of the shape volume. This data is collected in the node:
    - IFC Extruded Area Solid.

Core Walls:

Core walls are created by the node IFC Wall Standard Case and the data input is:

- IFC Host: Simplify and grafted data list from previous node IfcBuildingStorey node.
- IfcWallType: Contains all the information that defines the type of slab. This data is collected in the following series of nodes:
    - IfcMaterial
    - IfcMaterial Layer
    - IfcWallType
- PathOrPerimeter: Simplify data tree. Every path contains the core walls center line per floor. Each path represents one floor and the number of paths should be the same that the number of paths of the IFC Host.
- Wall Height: Floating number.

Internal Walls:

Internal walls are created by the node IFC Wall Standard Case and the data input is:

- IFC Host: Simplify and grafted data list from previous node IfcBuildingStorey node.
- IfcWallType: Contains all the information that defines the type of slab. This data is collected in the following series of nodes:

    - IfcMaterial
    - IfcMaterial Layer
    - IfcWallType
- PathOrPerimeter: Simplify data tree. Every path contains the Interior walls center line per floor. Each path represents one floor and the number of paths should be the same that the number of paths of the IFC Host.
- Wall Height: Floating number.

External walls:

External walls are created by the node IFC Wall Standard Case and the data input is:

- IFC Host: Simplify and grafted data list from previous node IfcBuildingStorey node.
- IfcWallType: Contains all the information that defines the type of slab. This data is collected in the following series of nodes:
    - IfcMaterial
    - IfcMaterial Layer
    - IfcWallType
- PathOrPerimeter: Simplify data tree. Every path contains the Interior walls center line per floor. Each path represents one floor and the number of paths should be the same that the number of paths of the IFC Host.
- Wall Height: Floating number.

Rooms:

External walls are created by the node IFC Space and the data input is:

- IFC Container: Simplify and grafted data list from previous node IfcBuildingStorey node.
- Long Name: Simplify data tree containing the name of each room in the project. Each path represents one floor and contains all the room names for that floor.
- IfcSpaceType: Contains the name of the type of space, in this case “Room”. This data is collected by in the following node:
    - IFC Space Type

- IfcRepItem: Simplify data tree containing the geometric information of the room volume. This data is collected in the node:
  - IFC Extruded Area Solid.

5.3.2.2 Model Transfer

Once the information is baked in Grasshopper the file needs to be imported through the GeometryGym plug-in in Revit. Once the extension is open in the Revit environment and the file is chosen, the only command to continue with the transaction is Proceed. The process is straight forward and does not need any additional work from the user, all the information defined in the parametrical environment is transferred together in the same IFC file.

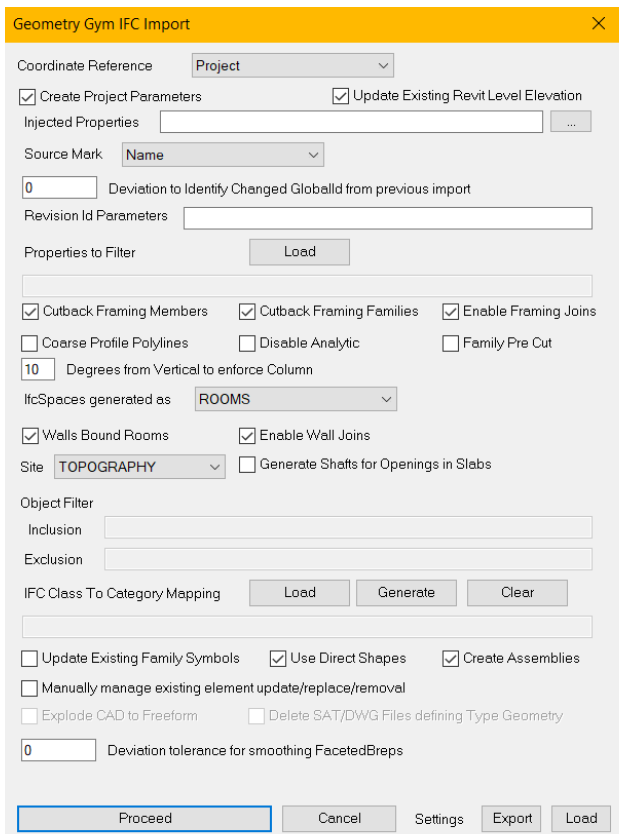


Figure 48: GeometryGym's import window.

The model transfer from the parametric environment fulfill the desired definition and there is no need of additional work. The process is simple and reduce the human error factor during the transfer. Nevertheless, because of the loosely coupled models, every time a modification is made in the parametrical environment a new file needs to be created and retransfer into the BIM environment. There is no simultaneous update between the two models.

5.3.3 Revit API and CVS file: Hummingbird

Hummingbird is a Grasshopper and Revit extension that is automatically installed in both software for a Tightly coupled parametric BIM workflow. The extension works directly in the Revit API since it automatically draws Revit native objects from data in a CVS file. The plug-in translates the information from the parametric modeler as start and end points for Revit commands. At the parametric graphic base entry, it captures and organized basic geometric definition and enriches the simple geometry elements so they can have minimum of necessary information to create an object in a BIM environment. Once the required data is introduced an individual CVS file needs to be written and save for every node. All nodes can share the same saving path and button action for this process to have a centralized "backed" command. Some exceptions may apply to rooms due to the limitations of the plug-in.

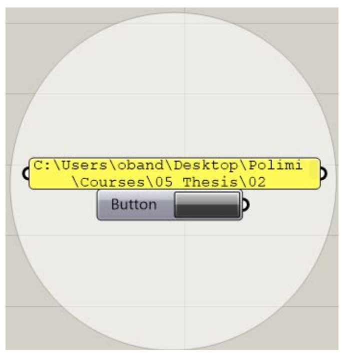


Figure 49: Hummingbird's write and path nodes inputs.

Once backed, at the other end of the stream in BIM software the plug in works in the program API to recreate modeling actions with the coordinates acquiree from the parametric modeler though the CVS files. Once modeled following the instructions created at the parametric modeler the result is the creation of a native geometry in Revit.

- Grasshopper Interface

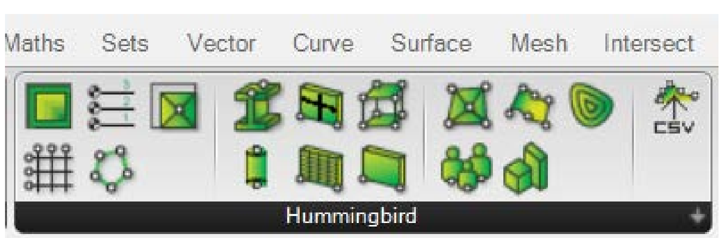


Figure 50: Hummingbird's Grasshopper library.

- Revit Interface

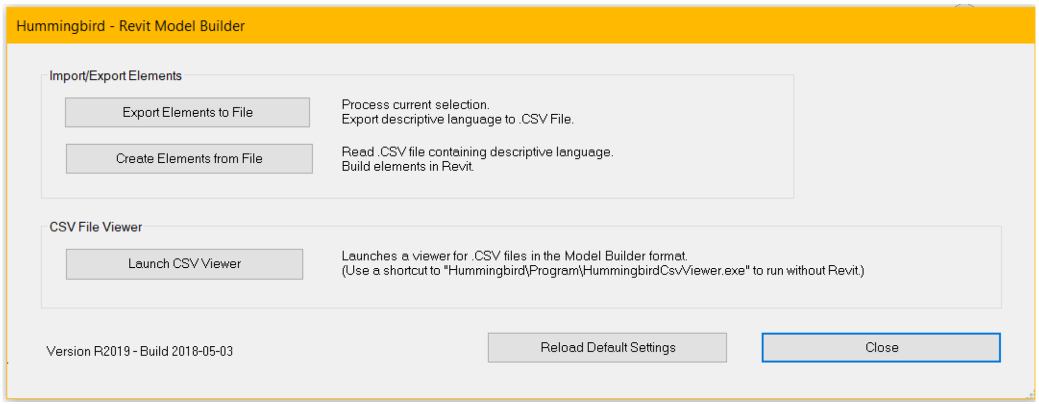


Figure 51: Hummingbird's Revit commands

Furthermore, an advantage of Hummingbird interface is that it enables the user to surf over the CVS file thought it owns CVS Viewer before creating the elements, this allows the designer to understand the structure of the data inserted to fulfil the Revit commands. It can ease the comprehension of the information required to plug in at the parametric graphic base modeler.

5.3.3.1 Model set up

Levels:

Levels are created by the node Levels and the data input is:

- Write: Boolean (true/false) data.
- Path: Saving folder path.
- File: Saving file name.
- Elev: Simple list of numbers whit a series of number corresponding the height of each. This list length and structure should match with the Name list.
- Name: Simple flatten text list with the name of each level. This list length and structure should match with the Name list.

Slabs:

Slabs are created by the node Floors and the data input is:

- Write: Boolean (true/false) data.

- Path: Saving folder path.
- File: Saving file name.
- Type: Text value with a Revit type name. In the case the type name is not present in Revit the element will be created with the last type for this model element.
- Curves: Simplify and graft list of polyline curves.

Shafts:

In hummingbird there is no node for shaft creation, but the slab voids are created when internal curves are drawn inside the slab perimeter. Therefore, for each floor the shaft curve is added inside the path of the slab creation curve data tree structure.

Core Walls:

Core walls are created by the node Walls and the data input is:

- Write: Boolean (true/false) data.
- Path: Saving folder path.
- File: Saving file name.
- Curves: Simplify and flatten list of polyline curves representing the core walls
- Type: Text value with a Revit type name. In the case the type name is not present in Revit the element will be created with the last type for this model element.
- Height: Floating number.

Internal Walls:

Internal walls are created by the node Walls and the data input is:

- Write: Boolean (true/false) data.
- Path: Saving folder path.
- File: Saving file name.
- Curves: Simplify and flatten list of lines representing the internal walls
- Type: Text value with a Revit type name. In the case the type name is not present in Revit the element will be created with the last type for this model element.
- Height: Floating number.

External walls:

External walls are created by the node Walls and the data input is:

- Write: Boolean (true/false) data.
- Path: Saving folder path.
- File: Saving file name.
- Curves: Simplify and flatten list of polyline curves representing the external walls
- Type: Text value with a Revit type name. In the case the type name is not present in Revit the element will be created with the last type for this model element.
- Height: Floating number.

Rooms:

Rooms are created by the node Rooms/Area. This node's information is transfer differently than the other nodes. The "bake" process is constrained due to the plug-in limitation in the Revit API. All information in this node is bake in the current floor view plan in Revit, therefore an individual file should be written per floor and then created in the corresponding floor view plan in Revit.

- Write: Boolean (true/false) data. Need to be reuse for every floor.
- Path: Saving folder path.
- File: Saving file name.
- Points: Simplify and grafted list of points coordinates. The point must be inside every typology. Information is baked per floor.
- Params: Text value with a room parameter to be modified. In this case study the parameter to be modified is Name according to its typology. This list length and structure should match with the point list. Information is baked per floor.
- Values: Text value with a room parameter to be specified. In this case study the values correspond to the typology name. This list length and structure should match with the point list. Information is baked per floor.

5.3.3.2 *Model Transfer*

Once the data is transformed into the CVS file in grasshopper, the files needs to be open through the Hummingbird plug-in inn Revit call Model Builder. Once the extension is open in the Revit environment the files are chosen one by one. All transfers can be done from Revit 3D view except for rooms, which needs to be place floor per floor in the floor view plan.

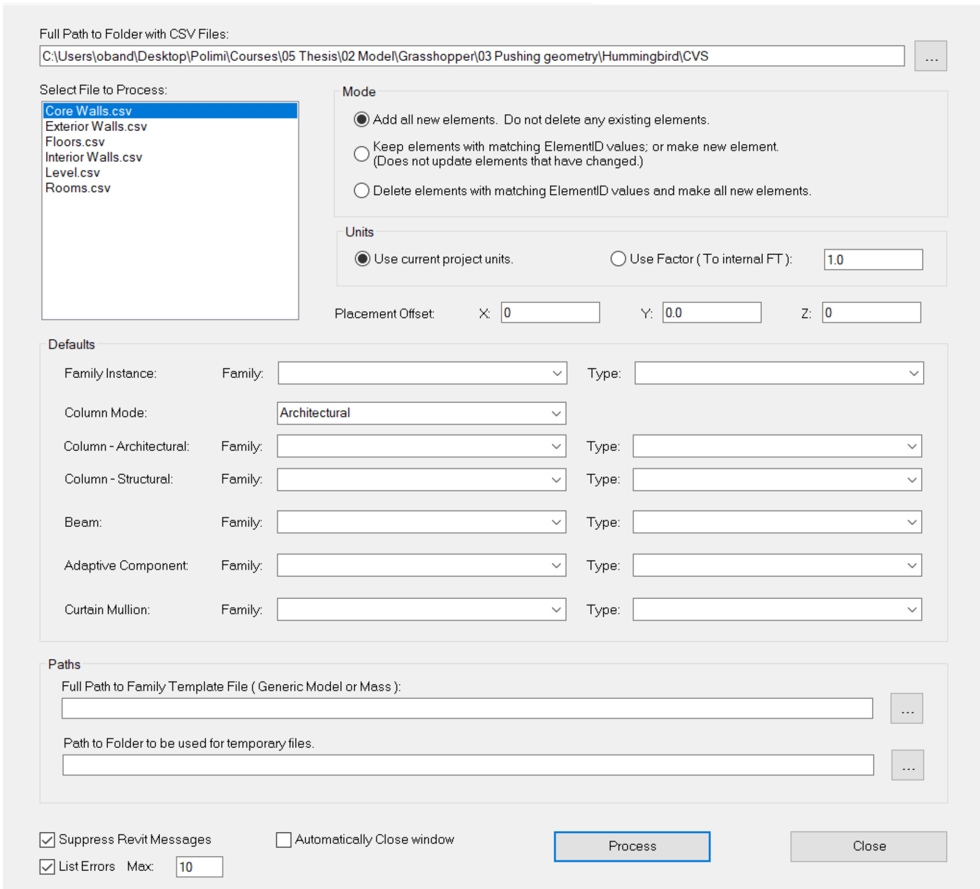


Figure 52: Hummingbird's import window.

The BIM conceptual model reach the desired visualization and the types are correctly assigned. Nevertheless, due to API limitations in the plug in the levels information for every element is different. All geometrical elements are placed over the level 0 and offset vertically until it reaches the required height. This fact reduces the quality of information complied at the end of the data stream in BIM since the real level of elements is lost in the process, to fix this data the elements should be selected directly on Revit and reassign the proper level manually. Added to the rooms interface this represents a high human intervention during this parametrical BIM workflow.

5.3.4 Cloud base workflows and Dynamo: Speckle

Speckle is a Grasshopper and Revit extension that is automatically installed in both software for a tightly coupled parametric BIM workflow. For this case study, the extension works directly in the BIM software API through Dynamo, which is the visual programming interface for Revit. Once installed, Speckle is added directly into Dynamo packages. This parametric BIM workflow operates different that the other approaches studied, since it does not require an intermediate file, and therefore a "baked" component. In this case, the parametric graphic base entry collects and stream just the basic geometric definition or description of the elements to be recreated in the BIM environment. The information is sent to a virtual network cloud already defined by the speckle developer, and with direct access from the user.

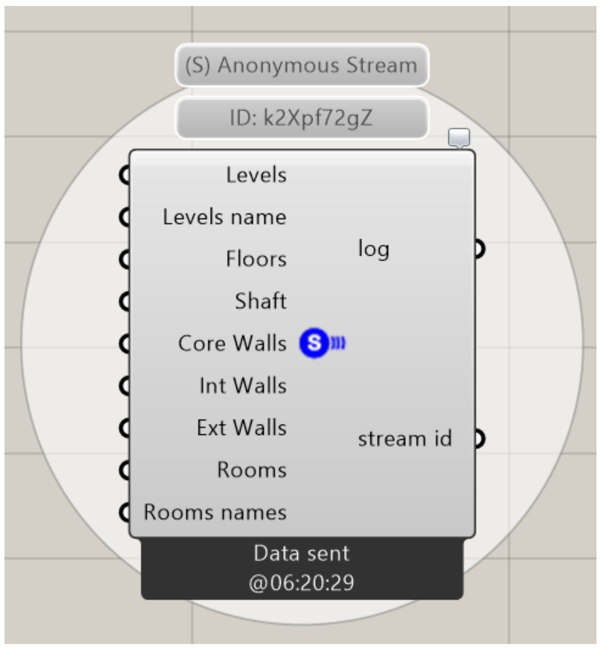


Figure 53: Speckle sender node

At the other end of the stream in the BIM software, the plug-in node in Dynamo works as a receiver. Different from the other parametric BIM workflows, the BIM platform receives raw data from the parametric designer and not full modeled objects with enriched information to be recognized as a BIM object. In this case, the information is organized and translated by nodes in a different visual programming interface. The principles of visual programming remain the same on both software, but this cloud base approach requires a higher expertise on coding. Nodes and data structure management for list and trees are different from Grasshopper to Dynamo.

- Grasshopper Interface

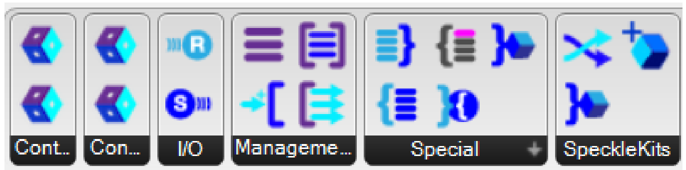


Figure 54: Speckle's Grasshopper library.

- Dynamo Interface

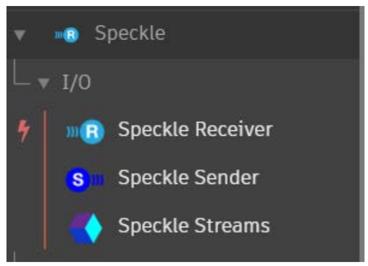


Figure 55: Speckle's Dynamo library.

Depending in the internet connection, this transfer of data can be a live streaming to multiple receivers, and to other software than just Dynamo. This means that the model can be instantaneously updated from Grasshopper into Dynamo and Revit. Another advantage of this parametric BIM workflow is that Dynamo can added an automated production of visualities elements that were manually introduced in the other methods like Floor plans views and tags.

5.3.4.1 Model Preparation

In this method the data send will be defined as the entry data at Grasshopper sender and the data input as the information require at Dynamo's node to generate Revit native element.

Levels:

Levels sent data input in grasshopper is:

- Levels: Simple list of numbers whit a series of number corresponding the height of each.
- Levels name: Simple flatten text list with the name of each level. This list length and structure should match with the Name list.

Levels are created by the Dynamo node Level.ByElevationAndName and the data input is:

- elevation: Levels data from receiver
- name: Levels name data from receiver

Slabs:

Slabs sent data input in grasshopper is:

- Floors: Simplify and flatten list of polyline curves.

Slabs are created by the Dynamo node Floors.ByOutlineTypeAndLevel and the data input is:

- outline: Floors data from receiver.
- floorType: Data from Floor Types node.
- level: Data from Level.ByElevationAndName node.

Shafts:

Shaft sent data input in grasshopper is:

- Shaft: Simplify polyline curve.

Shafts are created by the Dynamo node Opening.ByPathTypeAndLevel and the data input is:

- path: Shaft data from receiver.
- bottomLevel: Data from Level.ByElevationAndName node.
- topLevel: Data from Level.ByElevationAndName node.

Core Walls:

Core walls sent data input in grasshopper is:

- Core Wall: Simplify and grafted list of polyline curves representing the core walls. Every Path contains the information of one floor.

Core walls are created by the node Wall.ByCurveAndHeight and the data input is:

- curve: Core Wall data from receiver transform into a poly curve.

- height: Double number.
- level: Data from Level.ByElevationAndName node.
- wallType: Contains all the information that defines the type of wall. This data is collected in the following node:
    - Wall Types

Internal Walls:

Internal walls sent data input in grasshopper is:

- Internal Wall: Simplify and grafted list of lines representing the internal walls. Every Path contains the information of one floor.

Internal walls are created by the node Wall.ByCurveAndHeight and the data input is:

- curve: Internal Wall data from receiver transform into a poly curve.
- height: Double number.
- level: Data from Level.ByElevationAndName node.
- wallType: Contains all the information that defines the type of wall. This data is collected in the following node:
    - Wall Types

External Walls:

External walls sent data input in grasshopper is:

- Internal Wall: Simplify and grafted list of polyline curves representing the external walls. Every Path contains the information of one floor.

External walls are created by the node Wall.ByCurveAndHeight and the data input is:

- curve: Internal Wall data from receiver transform into a poly curve.
- height: Double number.
- level: Data from Level.ByElevationAndName node.
- wallType: Contains all the information that defines the type of wall. This data is collected in the following node:
    - Wall Types

Rooms:

Rooms sent data input in grasshopper is:

- Rooms: Data tree structure of points coordinates. Every Path contains the information of one floor.
- Rooms names: Data tree structure of text values with the room names coordinates. Every Path contains the information of one floor. Data tree structure should match with Room data input

Levels are created by the Dynamo node Room.ByLocation and the data input is:

- level: Data from Level.ByElevationAndName node.
- location: Rooms data from receiver.
- name: Rooms name data from receiver.

5.3.4.2 Model set up

Once the data is stream from the sender into the receiver, Dynamo collect the information and organize it through its nodes. The data is received with the same structure it was sent. Set up coding process is longer and complex than in other parametrical BIM workflow, since the visual programming is different with different nodes adapted to its BIM environment. Furthermore, the raw data there is still a certain among of information that needs to be added to consolidate, the BIM model definition. Thus, a great number of nodes needs to be added even after the information is received in Dynamo.

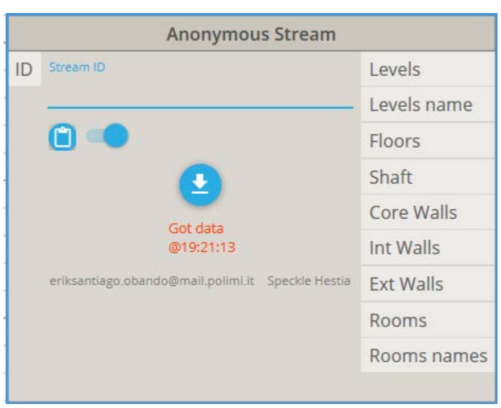


Figure 56: Speckle’s Dynamo receiver node.

On the other hand, once the model is set up is the most effective parametric BIM workflow, because it allows a full interoperability between PD and BIM. There is no intermediate model or file between the different environments, which reduces the human error factor in future modifications since there is no supplementary arrangements once the nodes are set up at both ends. Furthermore, modifications to the conceptual mass from the parametric modeler are simultaneously done in the BIM model. Any update to the design decision has an immediate impact in the BIM environment, and any detail definition of elements is done directly in the Dynamo interface.

5.3.5 Comparative Matrix.

In this chapter the three parametric BIM workflow are analyzed considering five different parameter that measure their performance in different characteristics of the workflow before and after the coupling of the models. The objective is to show that even if the same result can be achieved at the end of the process in the BIM environment, every method has its limitations and strengths. The role of the designer is to weigh this characteristic according to its needs and future adaptability of the model and chose the correct parametric BIM workflow for each case.

The objective of an efficient parametric BIM workflow is to enables the semi-automated generation linking of graph-based systems and BIM systems. This requires an approach that develops a graph-based system that use an object-based associative representation. Therefore, having an asociative model from parametrical model to the BIM model (Janssen, 2014). Once the workflow is stablished, to evaluate the efficency of the paramtrical BIM workflow method the recreation of associative characteristics of the final BIM model are compared with those of the starting generative model. The characteristics of a generative model defined by (Wortmann & Tunçer, 2017) are group and name as:

- Assembly: The rationalization of designs into buildable shapes and components.
- Accuracy: The control and setting out of architectural forms.
- Flexibility: The generation and testing of design variants based on various criteria and specialist input.

- Extensibility: The control and setting out of architectural forms, the translation of design ideas into parametric models.
- Interoperability: The sharing of information and the capture of design knowledge from specialist inputs.

Assembly:

It measures the complexity to stablish the workflow. It considers the first steps and the knowledge required to couple the Parametrical design objects to the BIM environment. The parameters considered are:

- Time of preparation to set up the workflow: Calculated by the number of nodes and commands needed in each end of the process to generate the final BIM model.
- Software knowledge: Consider the different types of list and data trees, for example: flatten lists, grafted lists, data trees. Also, the understanding and organization of these types of different data, its structure and definition.

Accuracy

This parameter asses the capability of the workflow to deliver a detail and complete BIM model. It measures the amount of correct organization of the information transfer from the PD at the BIM environment and the further manual adaptations that should be done to the model to arrive the desired definition. The measures for this purpose are:

- Human interaction once exported: Once in the BIM environment calculate the number of actions executed manually. More human interaction to finally set up the model increase the human error decreasing the accuracy of the model.
- Levels of definition that can be added: Is calculated by the inputs possible in the nodes that consolidates the workflow. Without adding any new nodes; therefore, not adding more

complexity to the workflow; the ability of the already stablished configuration to increase its level of definition.

Flexibility

This characteristic measures the interaction between the PD model and the BIM model. It defines the flexibility for the stablished workflow to modify the parameter it at the graph base modeler and update this modification in the BIM environment. The following parameter are considered:

- Actions on the other end to update model: Consider number of interactions with the BIM interface once a modification is done in the parametrical environment.
- Time to transfer and execute an update: establish a relation between the time of export of every parametrical BIM workflow transaction. Time depends a lot in the hardware use to model, hence this parameter calculated based in the relation of time between the fastest as and the slowest workflow to transfer the information.

Extensibility

This feature considers the amount of information that can be added to the parametric BIM workflow in further processes. The possibility to extend the amount of information and the different type of elements stream from the parametric model to the BIM environment. The parameters considered are:

- Amount of data and definition that can still be added: Is calculated through the total number of nodes added to the visual programming library by the plug in.

Interoperability

This point considered the capability of the workflow to extends its interoperability capacity more BIM software further than Revit. Once the interoperability between parametric model and BIM is stablished it considers the potential of a wider spectrum of interaction of the workflow with structure, management, and MEP modelers. The parameters measured are:

- The ability of the workflow to produce and open standard file: ability of the parametric BIM workflow translates the file directly to an IFC standard
- The direct transfer of raw data to different software: Measure the possibility to export in other BIM or design platforms.

5.3.6 Conclusions

A comparative radar is established to compare and evaluate every methodology. Every approach presents a different degree of transfer definition and human interaction. It is possible to assess that a higher accuracy and flexibility requires a higher complexity regarding the model assembly. And that not all method seems to have the same degree of flexibility and interaction between the parametrical modeler and the BIM environment, this means that model updates are more straight forward and intuitive than others. Even if all methods reach the same BIM definition and visualization, there is not an unify way and result regarding Parametric BIM workflow.

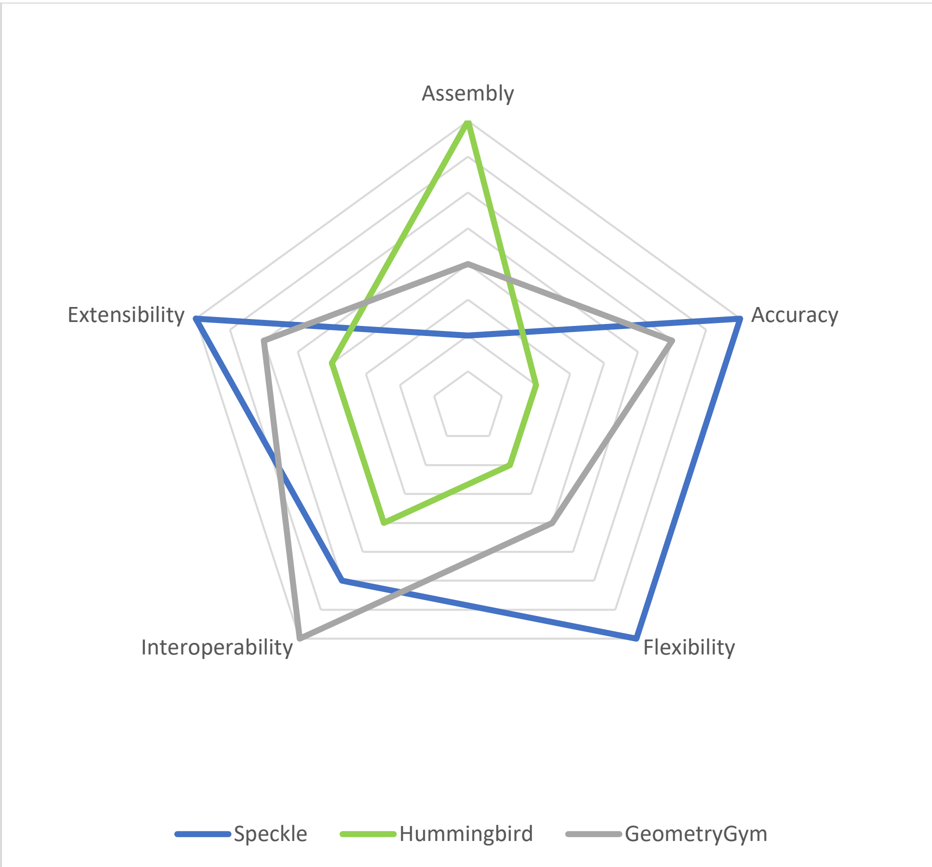


Figure 57: Comparative radar of parametrical BIM workflows

Parametrical BIM workflows methods were not able to stablish one unique model. All methods studied still work with two different models, one in the parametric environment and one at the BIM software. The levels of transfers vary from every method and some demand further human intervention to set up the models.

Nevertheless, once set up, it is possible to stablish a method that updates automatically both models. Speckle through a cloud base workflow and Dynamo interface manage to lively update the BIM model if any modification is done in the parametrical modeler. Therefore, the update and inclusion of information is automatically stream from PD to BIM, allowing a higher interoperability. Although this is the most efficient way of transfer is the most complicated one to set up since the designer must be proficient in both Dynamo and Grasshopper.

It is early to assess the real interoperability of a cloud bas transfer. The only Parametrical BIM workflow IFC standards are directly achieved directly by GeometryGym, therefore its interoperability is the highest, since ,for now, IFC lies at the heart of BIM interaction (Wortmann & Tunçer, 2017). Nevertheless, a cloud base approach stream raw information that can be translated directly into the receiver analysis software. If more “clients” decide to join a cloud base transfer, this can open a new possibility of associative interoperability.

6 Final Conclusions and Recommendations

Parametrical Design and Building information Modeling are diametrical different: PD for design questioning and BIM for the design interrogation (Banihashemi, Tabadkani, & Hosseini, 2018). Therefore, far from creating one model, the parametrical BIM method studied link their models by and additive workflow. In the three cases more nodes are added to transform basic geometry into BIM elements.

The methods in this research were not able to establish a unique model for PD and BIM, but it is possible to establish a command model in PD that, once coupled, automatically update in the model in the BIM environment. This allows to create a generative BIM model directly in Revit. This is possible thanks to Revit visual programming interface Dynamo.

Regarding innovation, is possible to consider that parametric BIM workflow manage information in an innovative way adding automatization and flexibility to the design process. In this study, once the models are set-up and coupled human intervention is reduced to its minimum. Geometry generation and BIM definition follows at every modification or inclusion of new parameter.

Furthermore, is possible to say that parametrical BIM workflow extends interoperability in the industry. Parametrical BIM workflows generates and associative model with its corresponding BIM information, thus allowing a more user friendly and straightforward information transfer. Parametrical modeler can add, manage, and optimize designer's, engineer's, and stakeholder's information, to directly stream it in BIM environment. Furthermore, this research shows that is possible to generate directly a IFC standard file if need, or transfer information thorough a cloud base approach to other software.

Further developments in parametrical BIM workflow should consider achieving a unique model creation. As a matter of fact, to prove the industry interest in the matter from 2019, for the first time, Rhino developer McNeel has develop its own extension Rhino.Inside.Revit to works directly in Revit using graphic interface. Supplementary research should consider this new approach.

Finally, the extension of parametrical BIM workflow should consider the cloud base approach as a new way of interoperability, more software should directly include the integration of raw data to generates its own native geometry. This development will approach the industry to IoT objective of Industry 4.0.

7 Bibliography

Aish, R., & Woodbury, R. (2005). *Multi-level Interaction in Parametric Design.*

Alfabuild ; Санджиев, Н., Лалин, В., Савченко, А., Сердюков, Д., Lalin, V., Savchenko, A., & Serdiukov, D. (2018). *Dynamo platform for automation Revit/ Sandzhiev N.*

Ambrose, M. (2007). *Bim And Integrated Practice As Provocateurs Of Design Education.*

Arturo Tedeschi. (2014). *AAD: Algorithms Aided Design.*

Autodesk. (2019, 12 20). *Dynamo: Visual Programming for Design Contents.* Retrieved from http://help.autodesk.com

Banihashemi, S., Tabadkani, A., & Hosseini, M. (2018, 4 1). Integration of parametric design into modular coordination: A construction waste reduction workflow. *Automation in Construction, 88,* 1-12.

Barazzetti, L. (2016, 8 1). Parametric as-built model generation of complex shapes from point clouds. *Advanced Engineering Informatics, 30*(3), 298-311.

Bew, M. (2015). *Digital Built Britain – Level 3 Strategy.*

BIMestimate. (2020, 03 16). Retrieved from https://bimestimate.eu/en/the-theory-of-evolution-bim-3d-7d/

Boeykens, S. (2012). Bridging Building information modeling and parametric design.

Boeykens, S. (2016). Improving Design Workflow in Architectural Design Applications.

Chaszar, A., & Joyce, S. (2016, 6 1). Generating freedom: Questions of flexibility in digital design and architectural computation. *International Journal of Architectural Computing, 14*(2), 167-181.

Deloitte. (2018). Global Construction Monitor 2018-2019.

EU BIM. (2017). *Handbook for the introduction of Building Information Modelling by the European Public Sector.*

Grilo, A., & Jardim-Goncalves, R. (2010, 8). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction, 19*(5), 522-530.

Haliburton, J., Ap, A., Clayton, M., Ozener, O., Farias, F., & Jeong, W. (2011). *Parametric Modeling and BIM: Innovative Design Education for Integrated Building Practices.*

Hansen Gøran, A., & Bjørn Smith, H. (2017). *From parametric design to full BIM: Eliminating the need for 2D drawings.*

Hugo, S., & Charles, E. (2011). *Preliminary Concept Design (PCD) Tools for Laboratory Buildings, Automated Design Optimization and Assessment Embedded in Building Information Modeling (BIM) Tools.*

Janssen, P. (2014). Parametric BIM Workflows.

Jassen, P. (2016). Automated Generationn of BIM Models Strict.

Kirschner, M. (2015). *Visual Programming in Three Dimensions: visual representations of computational mental models Associate Professor Of Architecture and Design Computation Chair, Department of Architecture Committee on Graduate Students 1.*

Kovacs, A., & Szoboszlai, M. (2019). *Key for Entering Industry 4.0 in the AEC Sector: BIM Organisation Development Building Information.*

Maravilha De Azevedo, O. (2009). *Metodologia BIM - Building Information Modeling na Direcção Técnica de Obras.*

Maskuriy, R., Selamat, A., Ali, K., Maresova, P., & Krejcar, O. (2019, 7 15). Industry 4.0 for the Construction Industry—How Ready Is the Industry? *Applied Sciences, 9*(14), 2819.

McGraw-Hill Construction. (2012). *The Business Value of BIM in North America: Multi-Year Trend Analysis and User Ratings (2007–2012).*

McNeel. (2020, 04 19). Retrieved from food4Rhino: https://www.food4rhino.com/app/octopus

Mirtschin, J. (2010). Engaging Generative BIM Workflows.

Mode Lab. (2015). *The Grasshopper Primer.*

Nazim, I., & Joyce, S. (n.d.). *User Directed Meta Parametric Design for Option Exploration.*

Nezamaldin, D. (2019). *Parametric design with Visual Programming in Dynamo with Revit.*

Oti, A., Tizani, W., Abanda, F., Jaly-Zada, A., & Tah, J. (2016). Structural sustainability appraisal in BIM. *Automation in Construction*.

Oxman, R. (2006, 5). Theory and design in the first digital age. *Design Studies, 27*(3), 229-265.

Parametricmonkey. (2020, 01 27). Retrieved from https://parametricmonkey.com/2015/06/24/hummingbird/

Project Hummingbird. (2020, 02 24). Retrieved from https://ghhummingbird.wordpress.com/

Rebim. (2020). Retrieved from https://rebim.io/level-of-detail-or-development-lod-in-bim/

Rhino to Revit and Back. (2020, 03 16). Retrieved from McNeel Wiki: https://wiki.mcneel.com/rhino/architecture/bim/rhino-to-revit

RIBA. (2012). *BIM Overlay to the RIBA Outline Plan of Work.*

RIBA. (2020). *BIM Overlay to the RIBA Outline Plan of Work.*

Rutten, D. (2013). Galapagos: On the Logic and Limitations of Generic Solvers.

Schumacher, P. (2010). Parametric Diagrammes.

Schwerdtfeger, E., & Zaha, |. (2018). *Custom Computational Workflows for BIM Design Implementation.*

Speckle: Data Platform for AEC. (2020, 02 30). Retrieved from https://speckle.systems/

Van Beusekom, C., Sarwarzadeh, M., Sinke, G., Sturm, C., & Zegger, L. (2018). *European Construction Monitor 2017-2018: A looming new construction crisis?*

Wojciech Adamus, L. (2013). *BIM: Interoperability for sustainability analysis in construction.*

Wojciech ADAMUS, L. (2013). *BIM: Interoperability for sustainability analysis in construction.*

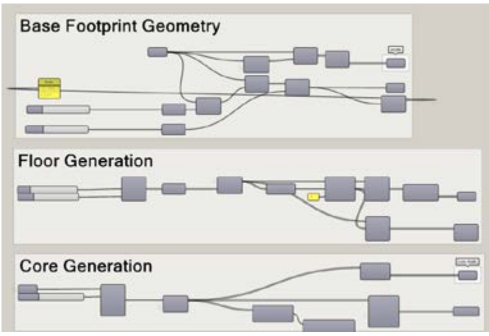
World Economic Forum. (2016). *Industry Agenda Shaping the Future of Construction A Breakthrough in Mindset and Technology Prepared in collaboration with The Boston Consulting Group.*

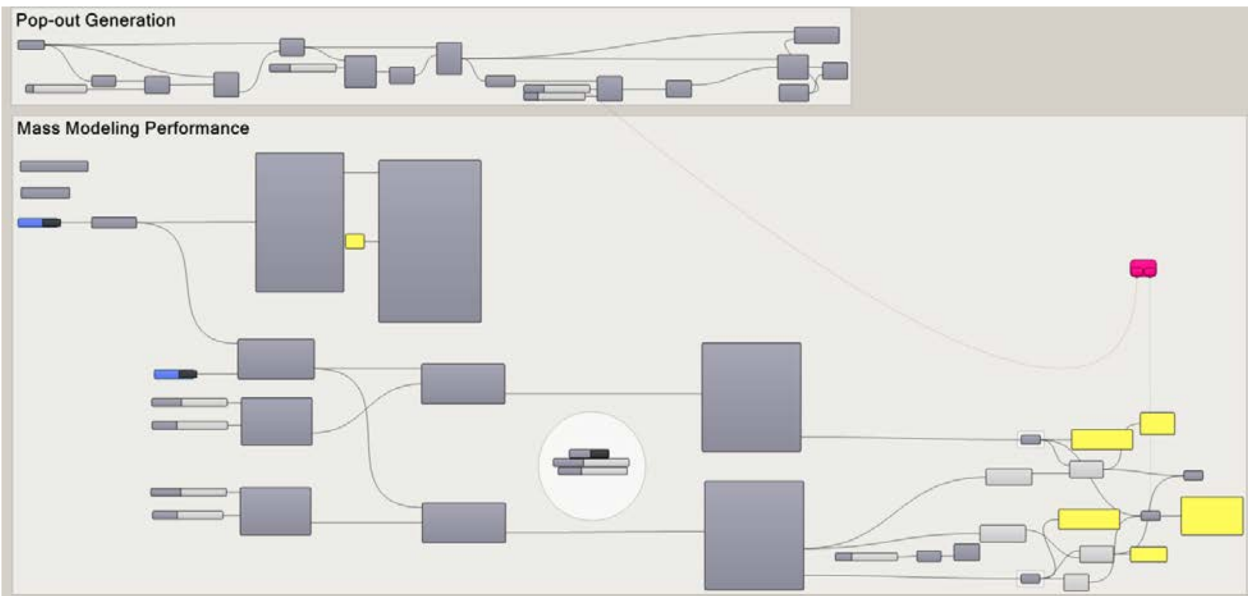
Wortmann, T., & Tunçer, B. (2017, 9 1). Differentiating parametric design: Digital workflows in contemporary architecture and construction. *Design Studies, 52*, 173-197.

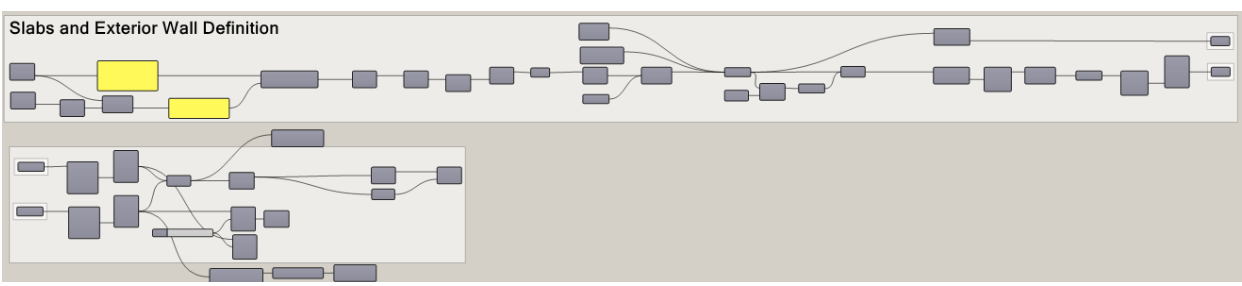
8 Annexes

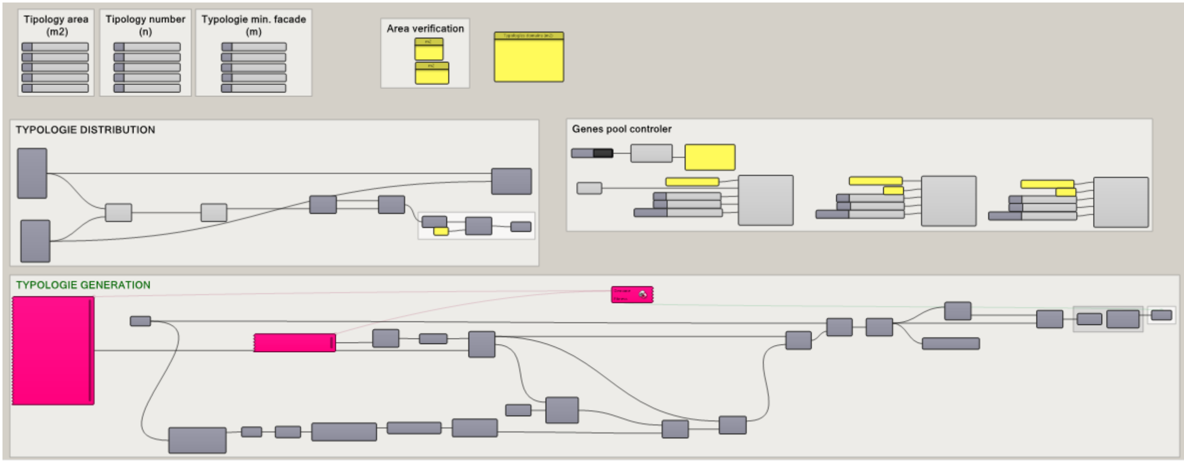
8.1 Script definition

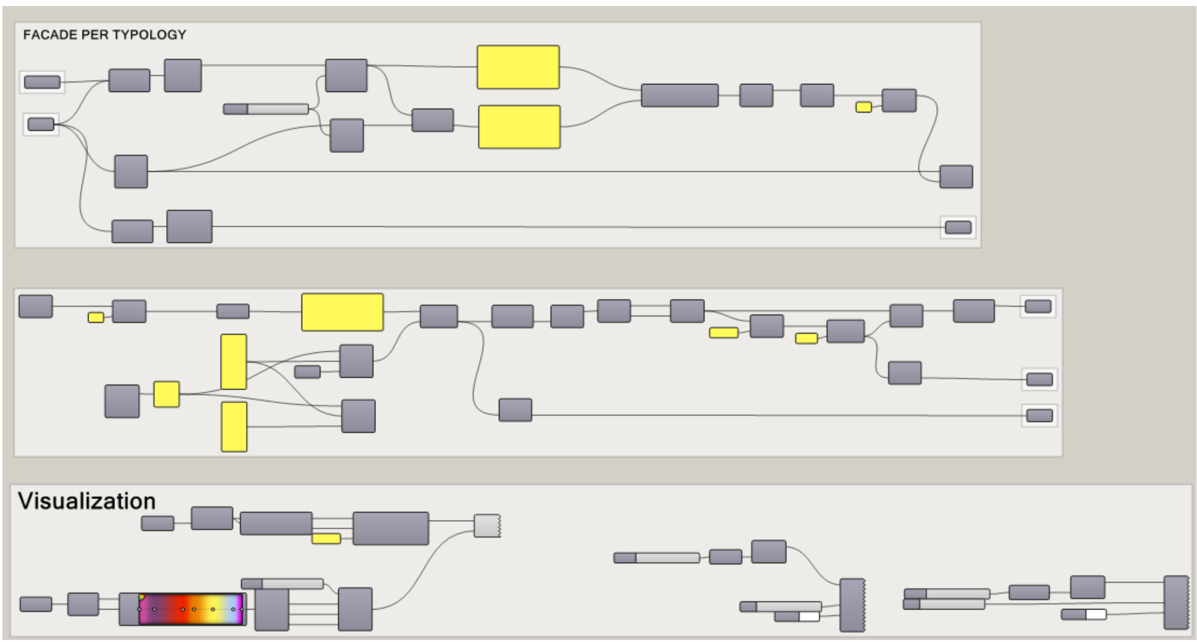
Annex 1: Building massing Script



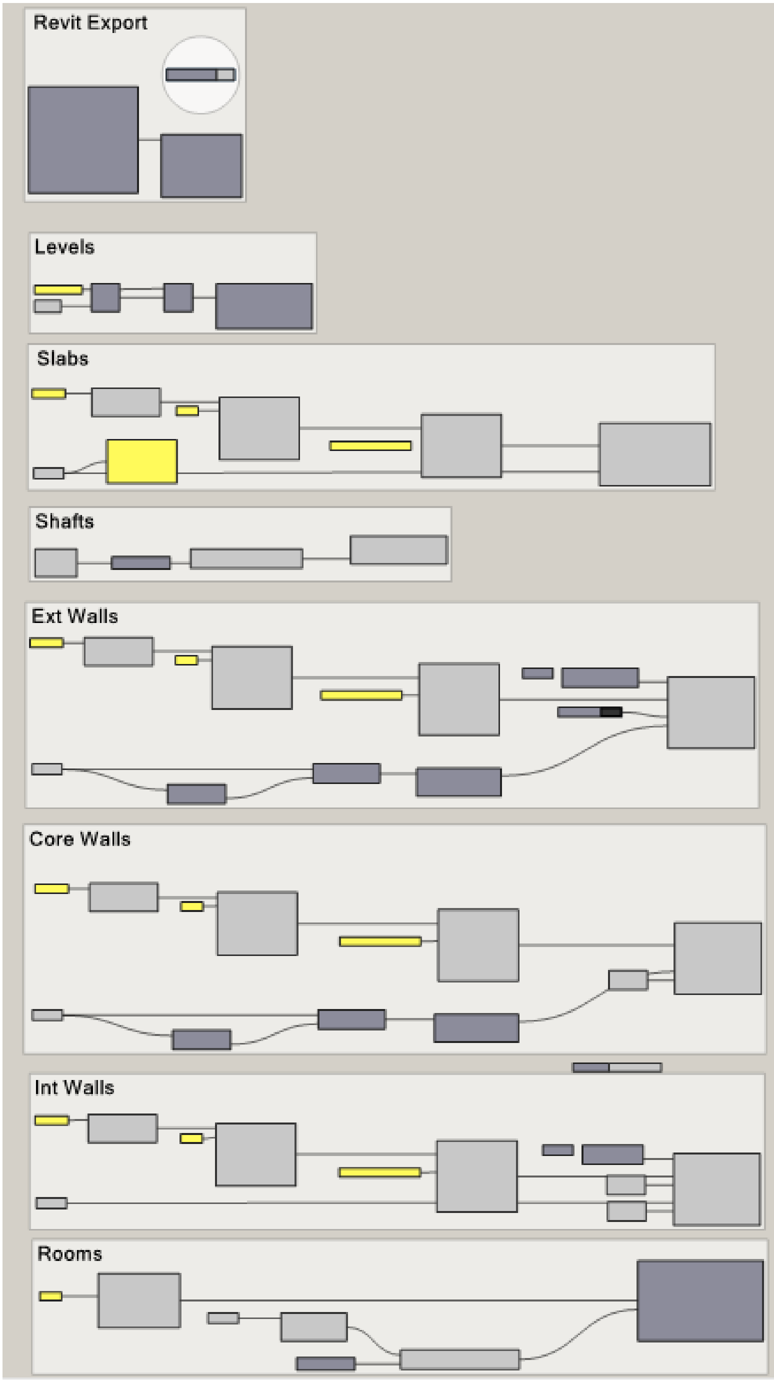




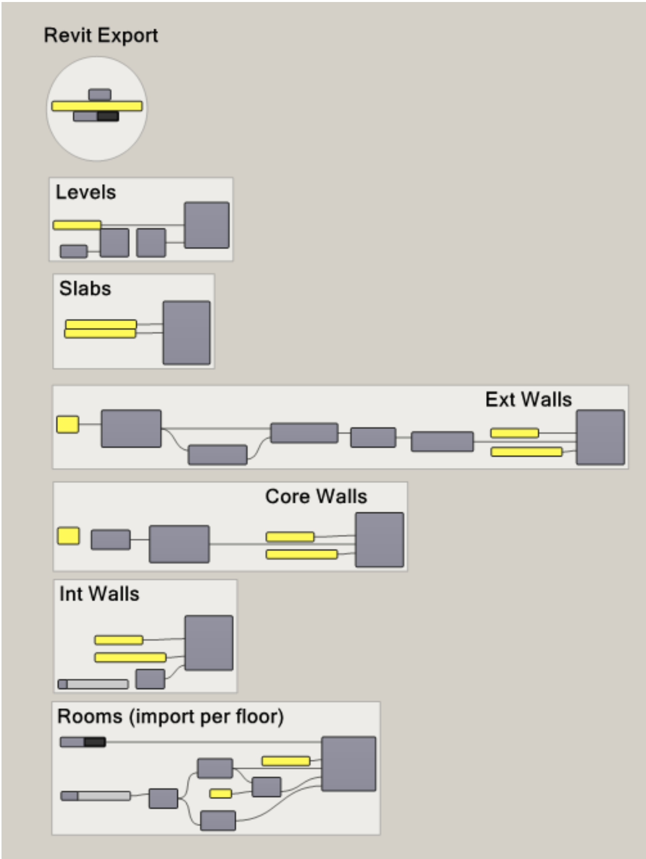

Tipology area (m2)
Tipology number (n)
Typologie min. facade (m)
Area verification
TYPOLOGIE DISTRIBUTION
Genes pool controler
TYPOLOGIE GENERATION


FACADE PER TYPOLOGY
Visualization

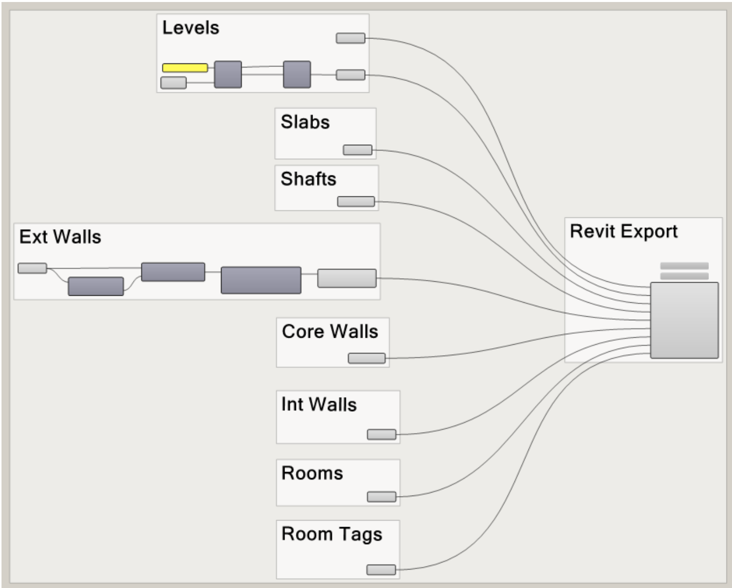
Annex 2: GeometryGym set-up script



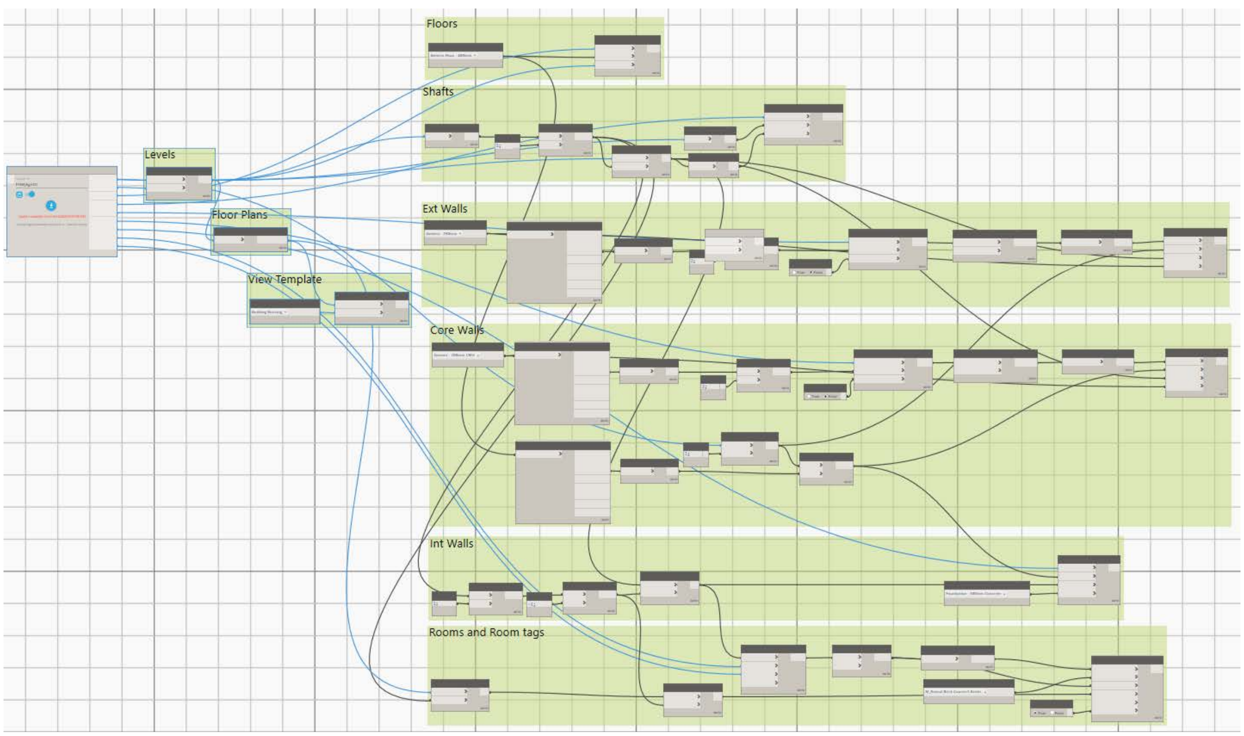
Annex 3: Hummingbird set-up script



Annex 4: Speckle set-up script in Grasshopper

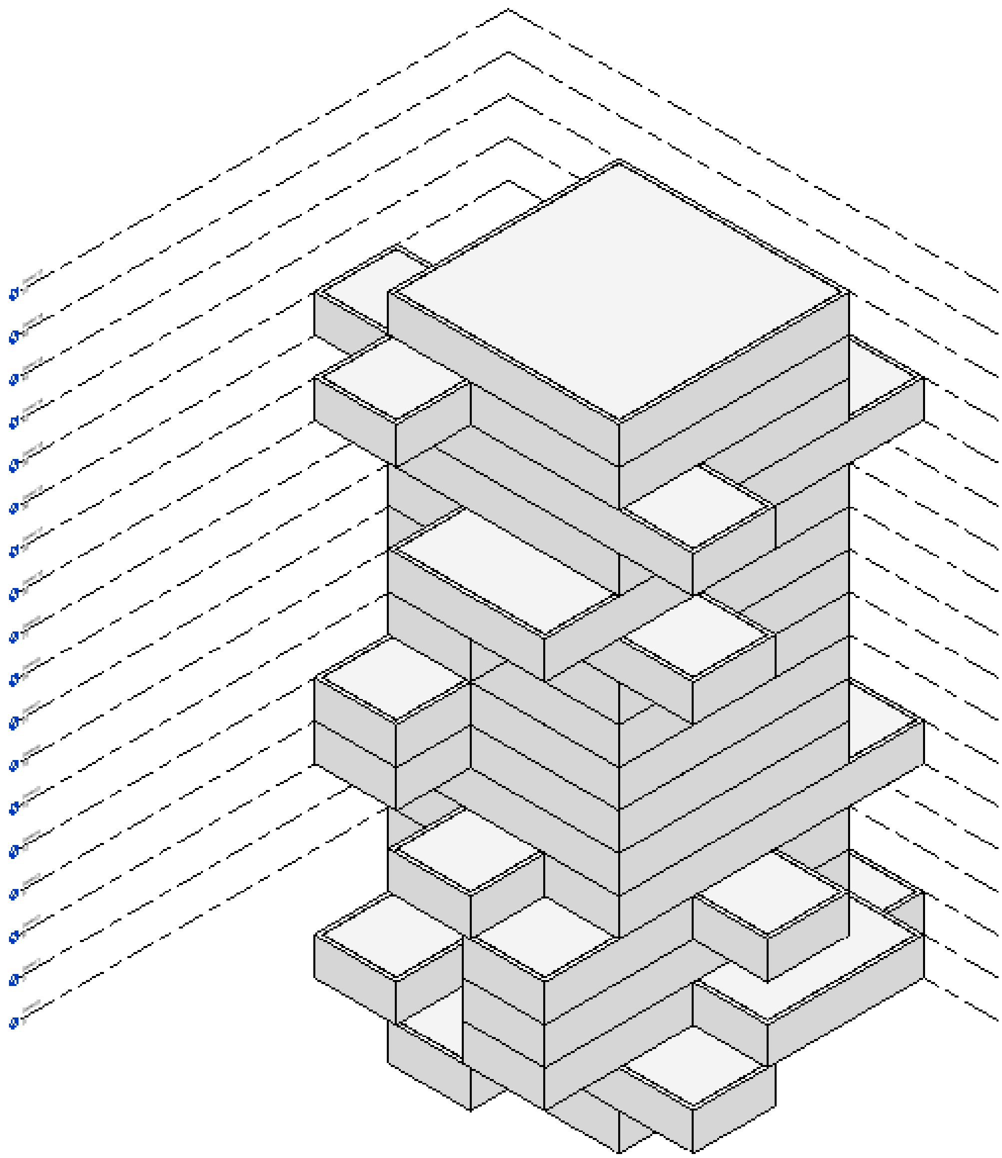


Annex 5: Speckle set-up in Dynamo

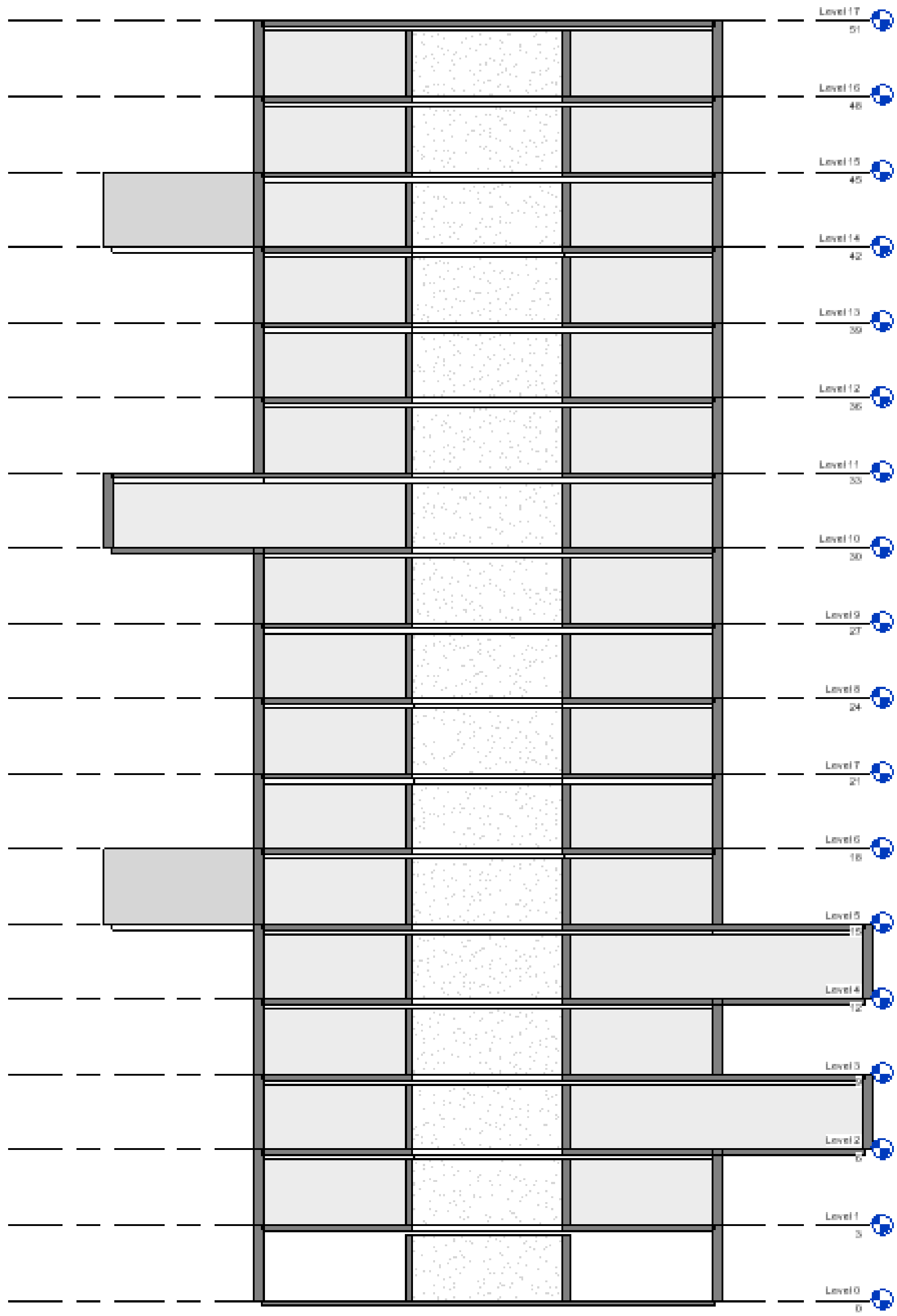


8.2 Revit Visualization

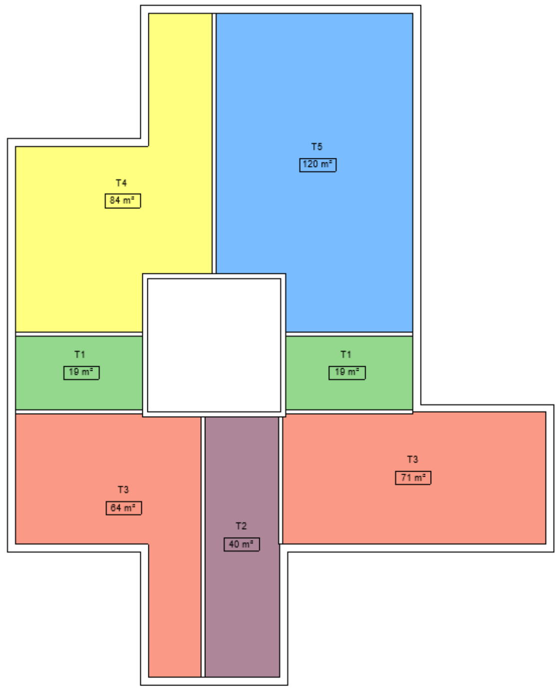
Annex 6: 3D view



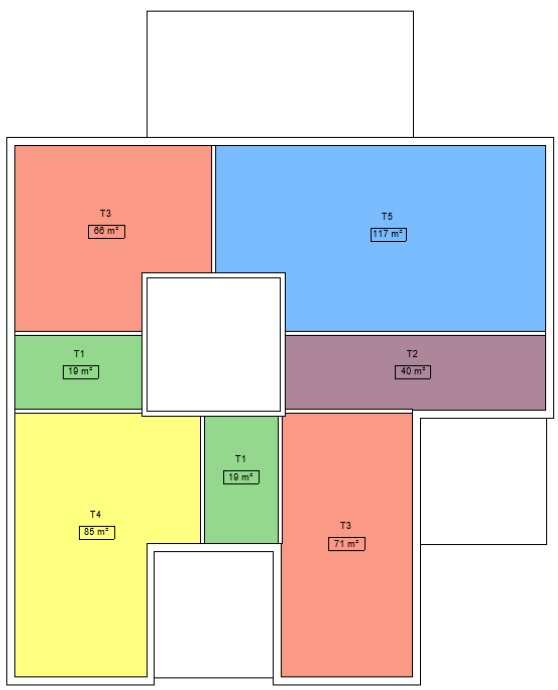
Annex 7: Section



Annex 8: Floor Plan Level 1

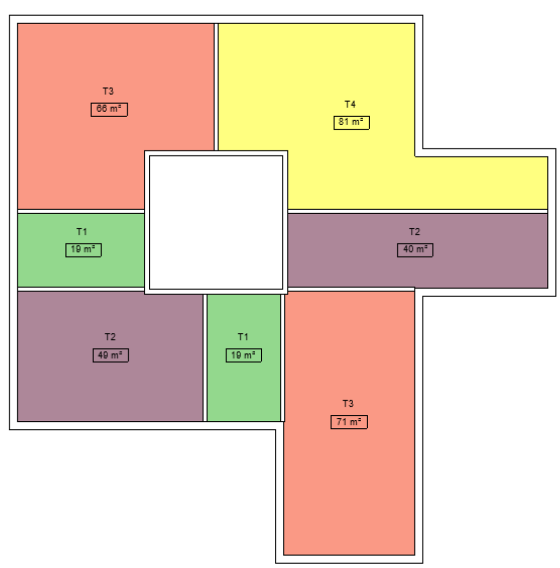


Annex 9: Floor Plan Level 2

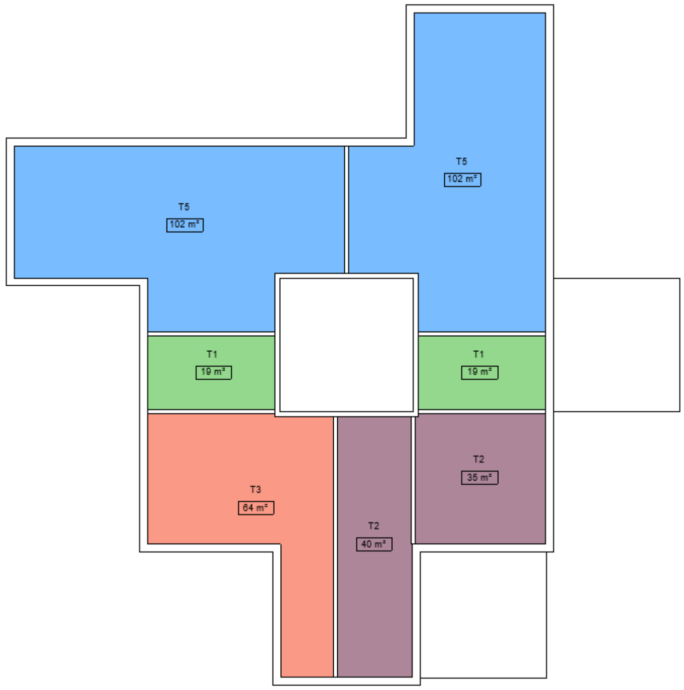


Annex 10: Floor Plan Level 3

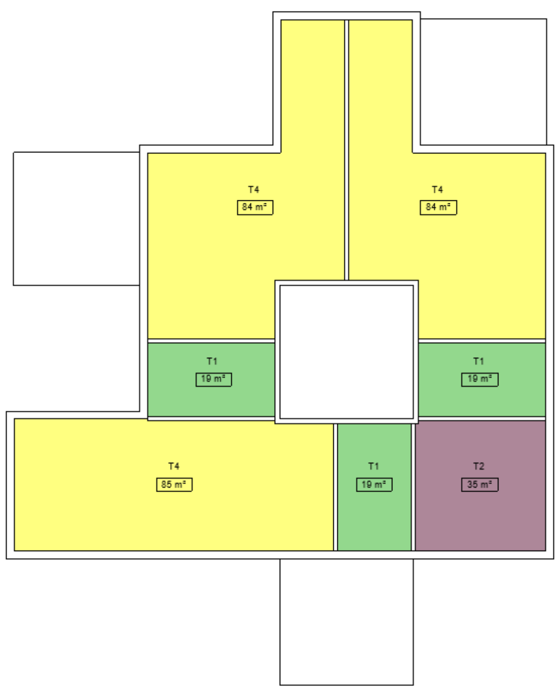
Annex 11: Floor Plan Level 4



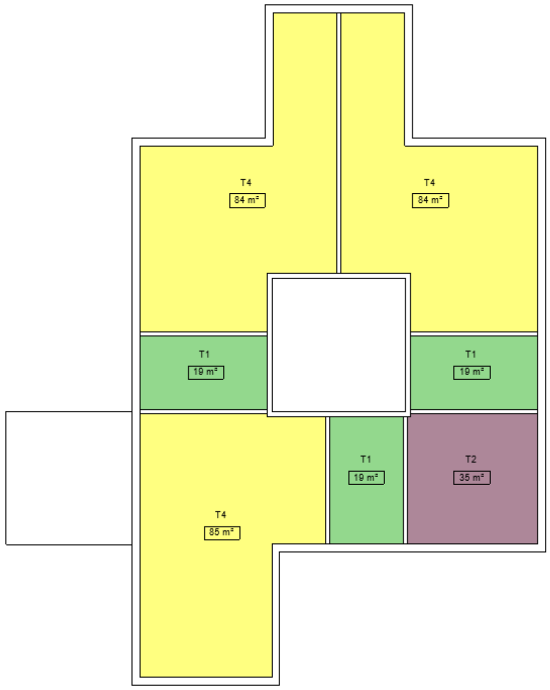
Annex 12: Floor Plan Level 5



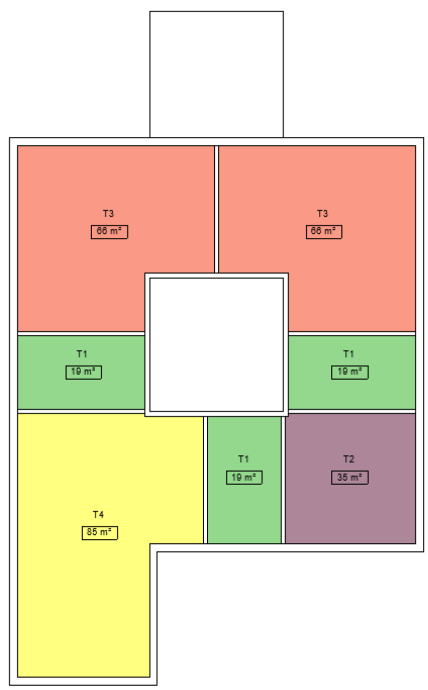
Annex 13: Floor Plan Level 6



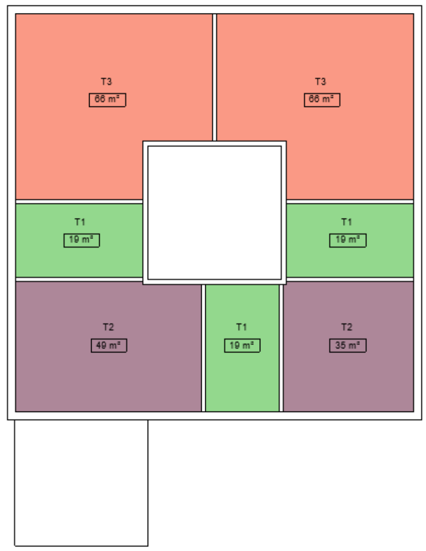
Annex 14: Floor Plan Level 7



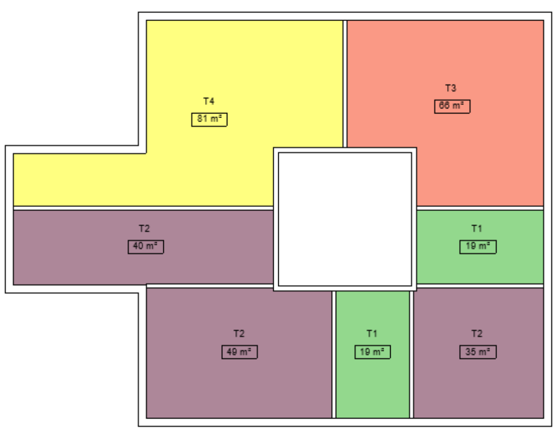
Annex 15: Floor Plan Level 8



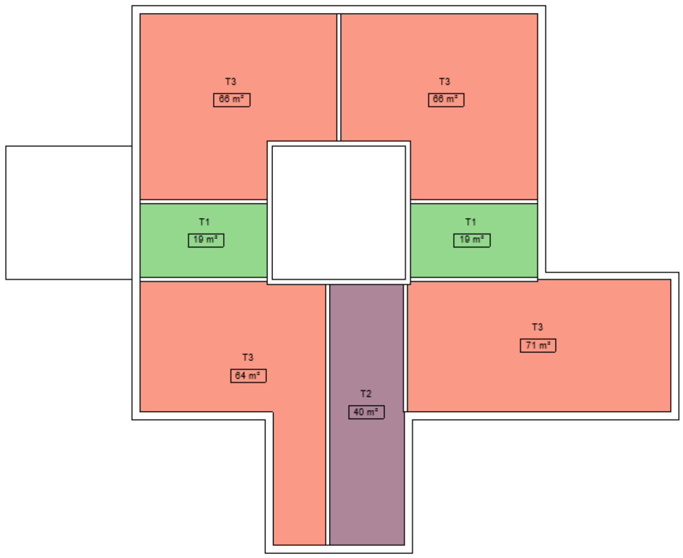
Annex 16: Floor Plan Level 9



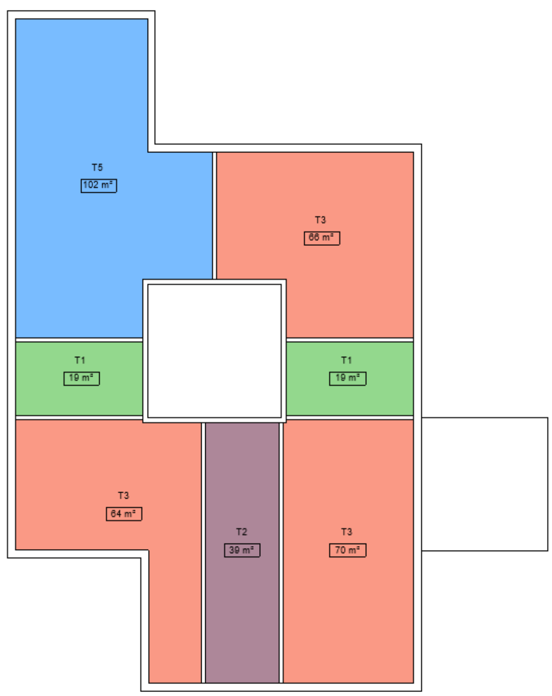
Annex 17: Floor Plan Level 10



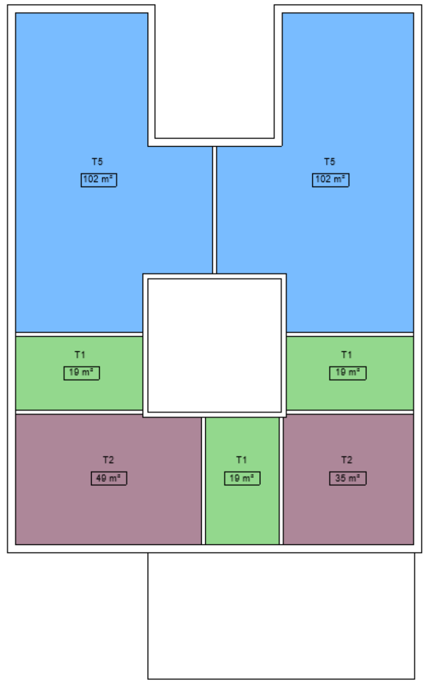
Annex 18: Floor Plan Level 11



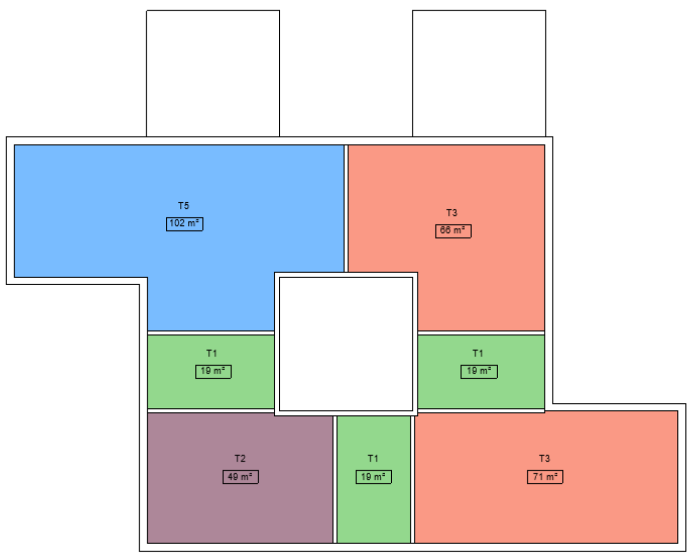
Annex 19: Floor Plan Level 12



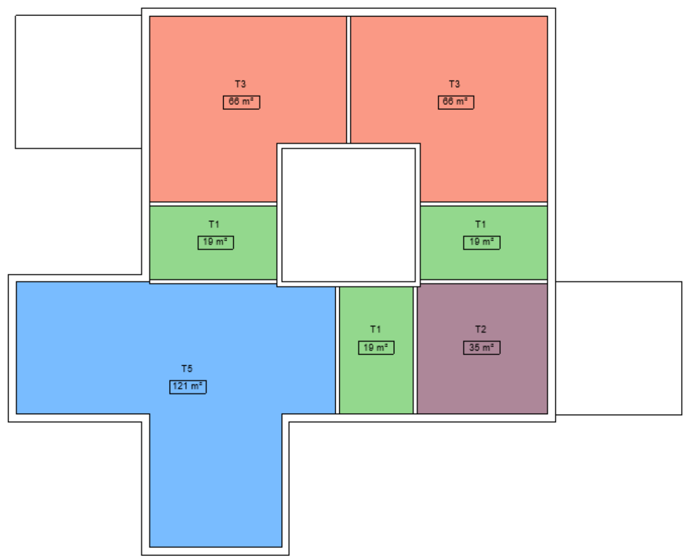
Annex 20: Floor Plan Level 13



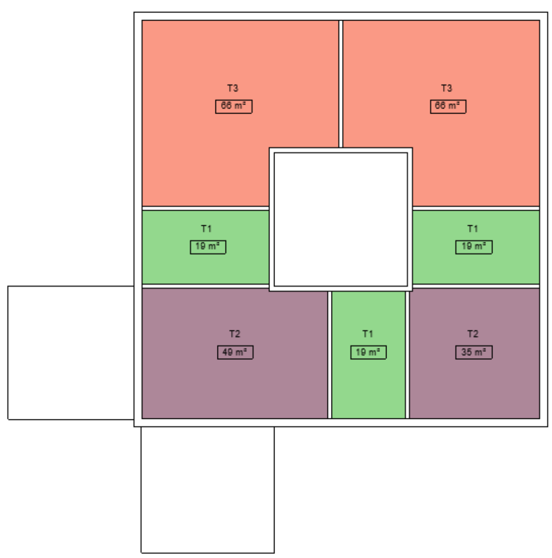
Annex 21: Floor Plan Level 14



Annex 22: Floor Plan Level 15



Annex 23: Floor Plan Level 16



Annex 24: Typology per level

Level 1

T1	19 m²
T4	84 m²
T2	40 m²
T1	19 m²
T3	71 m²
T5	120 m²
T3	64 m²
Level 1: 7	415 m²

Level 2

T2	40 m²
T3	66 m²
T1	19 m²
T1	19 m²
T3	71 m²
T5	117 m²
T4	85 m²
Level 2: 7	416 m²

Level 3

T1	19 m²
T3	66 m²
T1	19 m²
T1	19 m²
T3	71 m²
T3	66 m²
T2	49 m²
Level 3: 7	310 m²

Level 4

T2	40 m²
T3	66 m²
T1	19 m²
T1	19 m²
T3	71 m²
T4	81 m²
T2	49 m²
Level 4: 7	344 m²

Level 5

T1	19 m²
T5	102 m²
T2	40 m²
T1	19 m²
T2	35 m²
T5	102 m²
T3	64 m²
Level 5: 7	380 m²

Level 6

T1	19 m²
T4	84 m²
T1	19 m²
T1	19 m²
T2	35 m²
T4	84 m²
T4	85 m²
Level 6: 7	344 m²

Level 7

T1	19 m²
T4	84 m²
T1	19 m²
T1	19 m²
T2	35 m²
T4	84 m²
T4	85 m²
Level 7: 7	344 m²

Level 8

T1	19 m²
T3	66 m²
T1	19 m²
T1	19 m²
T2	35 m²
T3	66 m²
T4	85 m²
Level 8: 7	310 m²

Level 9

T1	19 m²
T3	66 m²
T1	19 m²
T1	19 m²
T2	35 m²
T3	66 m²
T2	49 m²
Level 9: 7	274 m²

Level 10

T1	19 m²
T4	81 m²
T1	19 m²
T2	40 m²
T2	35 m²
T3	66 m²
T2	49 m²
Level 10: 7	309 m²

Level 11

T1	19 m²
T3	66 m²
T2	40 m²
T1	19 m²
T3	71 m²
T3	66 m²
T3	64 m²
Level 11: 7	344 m²

Level 12

T1	19 m²
T5	102 m²
T2	39 m²
T1	19 m²
T3	70 m²
T3	66 m²
T3	64 m²
Level 12: 7	380 m²

Level 13

T1	19 m²
T5	102 m²
T1	19 m²
T1	19 m²
T2	35 m²
T5	102 m²
T2	49 m²
Level 13: 7	345 m²

Level 14

T1	19 m²
T5	102 m²
T1	19 m²
T1	19 m²
T3	71 m²
T3	66 m²
T2	49 m²
Level 14: 7	345 m²

Level 15

T1	19 m²
T3	66 m²
T1	19 m²
T1	19 m²
T2	35 m²
T3	66 m²
T5	121 m²
Level 15: 7	345 m²

Level 16

T1	19 m²
T3	66 m²
T1	19 m²
T1	19 m²
T2	35 m²
T3	66 m²
T2	49 m²
Level 16: 7	274 m²
Grand total: 112	5481 m²

Annex 25: Typology distribution

T1

Level 1	19 m²
Level 1	19 m²
Level 2	19 m²
Level 2	19 m²
Level 3	19 m²
Level 3	19 m²
Level 3	19 m²
Level 4	19 m²
Level 4	19 m²
Level 5	19 m²
Level 5	19 m²
Level 6	19 m²
Level 6	19 m²
Level 6	19 m²
Level 7	19 m²
Level 7	19 m²
Level 7	19 m²
Level 8	19 m²
Level 8	19 m²
Level 8	19 m²
Level 9	19 m²
Level 9	19 m²
Level 9	19 m²
Level 10	19 m²
Level 10	19 m²
Level 11	19 m²
Level 11	19 m²
Level 12	19 m²
Level 12	19 m²
Level 13	19 m²
Level 13	19 m²
Level 13	19 m²
Level 14	19 m²
Level 14	19 m²
Level 14	19 m²
Level 15	19 m²
Level 15	19 m²
Level 15	19 m²
Level 16	19 m²
Level 16	19 m²
Level 16	19 m²
T1: 41	786 m²

T2

Level 1	40 m²
Level 2	40 m²
Level 3	49 m²
Level 4	40 m²
Level 4	49 m²
Level 5	40 m²
Level 5	35 m²
Level 6	35 m²
Level 7	35 m²
Level 8	35 m²
Level 9	35 m²
Level 9	49 m²
Level 10	40 m²
Level 10	35 m²
Level 10	49 m²
Level 11	40 m²
Level 12	39 m²
Level 13	35 m²
Level 13	49 m²
Level 14	49 m²
Level 15	35 m²
Level 16	35 m²
Level 16	49 m²
T2: 23	934 m²

T3

Level 1	71 m²
Level 1	64 m²
Level 2	66 m²
Level 2	71 m²
Level 3	66 m²
Level 3	71 m²
Level 3	66 m²
Level 4	66 m²
Level 4	71 m²
Level 5	64 m²
Level 8	66 m²
Level 8	66 m²
Level 9	66 m²
Level 9	66 m²
Level 10	66 m²
Level 11	66 m²
Level 11	71 m²
Level 11	66 m²
Level 11	64 m²
Level 12	70 m²
Level 12	66 m²
Level 12	64 m²
Level 14	71 m²
Level 14	66 m²
Level 15	66 m²
Level 15	66 m²
Level 16	66 m²
Level 16	66 m²
T3: 28	1874 m²

T4

Level 1	84 m²
Level 2	85 m²
Level 4	81 m²
Level 6	84 m²
Level 6	84 m²
Level 6	85 m²
Level 7	84 m²
Level 7	84 m²
Level 7	85 m²
Level 8	85 m²
Level 10	81 m²
T4: 11	919 m²

T5

Level 1	120 m²
Level 2	117 m²
Level 5	102 m²
Level 5	102 m²
Level 12	102 m²
Level 13	102 m²
Level 13	102 m²
Level 14	102 m²
Level 15	121 m²
T5: 9	969 m²
Grand total: 112	5481 m²