TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

MASTERARBEIT

Native vs. Custom BIM Functions: An Enclosure Design Case Study

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom-Ingenieurs unter der Leitung

Ao.Univ.Prof. Dipl.-Arch. Dr.phil. Georg Suter

Institutsnummer 259.1

IEMAR Institute of Architectural Science

eingereicht an der Technischen Universität Wien

Fakultät für Architektur und Raumplanung

von

Fabio Palvelli
0928409
Hauffgasse 14/1/49
1110 Wien

Wien, am 10.01.2012

FABIO PALVELLI

Native vs. Custom BIM Functions:

An Enclosure Design Case Study

Abstract

Building Information Modeling (BIM) has become one of the major digital trends in AEC information technology and is also increasingly used by many architects. This thesis explores support for enclosure design in commercial BIM Design Tools. Computational requirements for enclosure design are defined first. These include iterative design, data integration, and performance analysis support. Next the Gasometer B in Vienna is described. It was used for the case study and was chosen because of the freeform shape of its enclosure. In the following, the Gasometer enclosure is modeled in two representative BIM Design tools, Revit and Rhino. While the former tool supports enclosure design with native, that is, built-in functions, the latter supports geometric design in general but may be customized for the purpose of enclosure design. Modeling tasks included massing design, schematic enclosure design, and detailed enclosure design. Based on the experience from the modeling exercise, the tools are evaluated with respect to the computational requirements. The thesis concludes with observations on how to improve computational support for enclosure design in BIM Design Tools.

Keywords: BIM, Geometric Design, Parametric Modeling.

Acknowledgment

I would like to express my thankfulness to my advisor Georg Suter for his consistent support and great help during this incredible journey.

I would also like to thanks my colleagues and teachers (too many to name them all) at the course of Building Science of the TU Vienna for helping me finding the motivation by being very challenging with their own work during these last 2 years of my life.

Last but not least I would like to thank my parents Massimo and Luisa for it is because of them that all of this was possible. They did always remind me that nothing comes out of nothing.

..."ex nihilo nihil fit".

Table of Contents

1 Introduction

1.1 What is BIM?

"Humankind has been interested in building construction for thousands of years. Construction projects, however, are typically too large for any one individual to accomplish alone, so from the very beginning humans have developed approaches to collaborating on such endeavors. Building often has a social context and benefits a number of persons, whose values it symbolizes.

These large-scale accomplishments necessarily require collaboration on the part of the participants. Various cultures create social events around such collaborative efforts that are required to build a facility for the community or for an individual of that community" (Kymmel, 2008).

The nomenclature of "Building Information Modeling", (BIM) is used to describe digital tools or activities and processes that facilitate all the operations related to a building's design (Eastman, 2011). Eventually BIM is finding only recently its popularity in the AEC field probably thanks to the sets of specific computer programs released in these last five to six years, labeled as BIM and dedicated specially to architecture and construction.

In 1986 Graphisoft introduced “Virtual Building Solution” nowadays known as Archicad. This new software made possible -for the very first time- the three dimensional (3d) representation of architecture project instead of the standard two dimensional one (2d) offered by the common Computer Aided Design (CAD) programs of the time. This was a revolution because architects and engineers were enabled to store large amounts of data sets within the digital building model. This data consisted of course of the geometry of the building, but also it represented consistent information that could be used for a more precise and controlled cost calculation of the building assembly (Laiserin.com, 2003), whereas designers that used standard CAD applications needed many specification sheets in order to convey and translate the information pertaining to the project.

Constructing a virtual building in a 3d digital modeling software -along with its associated real building data- is the main requisite for the practice of Building Information Modeling.

In BIM, parametric information is used for design decision making, production of high-quality construction documents, prediction of building performance, cost estimating and construction planning and lately (thanks to more contemporary advanced implementation of such building information) also for the management of the life cycle of a building once this is finally erected (Eastman, 2011).

Even if BIM is a relatively recent approach to computer aided architectural design, many of the concepts behind it were eventually already explored by earliest CAD applications such as Sketchpad (Sutherland, 2003) and they have always been considered when designing any general CAD software until now ("BIM Modeling Blogspot," 2010).

The structure of BIM software dedicated to architectural design is based on the parametric relationship that the set of building components (walls, floors, stair, etc.) have with each other, the geometric consistency and integrity of the building model is insured by the embedded automation of the software itself. Understanding the concept of these parametric objects is key to understanding what a building information model is (in Short "Building Model") and how it differs from traditional 2D design (Kymmel, 2008). A parametric object consists of a series of geometric definitions and their associated data and rules. In addition, these geometric definitions are integrated non-redundantly and do not allow for inconsistencies between the model and its associated data set. This means that any changes made directly to the model will result in an equal change to the data set associated with the model (i.e. plans, sections, elevations) (Kymmel, 2008).

Initial experiences with BIM indicate that the creation of a 3D model with associated information reduces errors of design, improves design quality, shortens construction time, and significantly reduces construction costs (Eastman, 2004). Due to these initial findings the popularity of BIM has grown tremendously in the past decade. In the U.S. for instance the A.I.A. (American Institute of Architects) is pushing the idea of making BIM software a legislative requirement for tenders of a project when designing a building, because these help defining the liability of all the professionals involved in the design and construction process.

Currently in the UK it is estimated by "Asite.com" that more than 75% of architects use 3d models as a presentation media only, concentrating all their efforts in working with 2d drawings from initial design proposal all the way to planning and construction. This means that the 3d model produced for presentation, represent only a set of plain geometries that carry no information apart from the texture applied to it for rendering/illustration purposes. Eventually part of these architects would need also physical carton models to study different design options and to communicate ideas to the clients. This way each technique used to design a building is not directly related but only referenced one to another. As one can easily assume this allows a large room for errors specially when having to translate geometrical information from a media to another. With the current availability of BIM software a designer can optimize this workflow of exchange of information by relying on the parametric interaction of the building elements used to construct a digital model of a building. Eventually all the 3d information is translated to 2d data and the drawings are also organized in the right schedule avoiding eventual mistakes due to revisions.

When this type of workflow is then extended to a larger collaborative environment outside the architect's office (i.e. Architects to Engineers to Product Manufacturers), the exchange of information becomes easier as well as keeping track of all the modifications or construction conflicts possible during the design process. We can convey that since the adoption of BIM in practice, architecture has changed noticeably (we just need to look at the "BIM made" architecture of Frank Gehry to understand the benefits and the advantages brought by such tools (Figure 1).

Figure 1. Gehry´s Disney Hall (http://www.gehrytechnologies.com/)

1.2 Current BIM solutions

At the moment there are on the market a lot of software solutions that are branded as BIM tools. These software are usually modeling packages, so to be precise, calling them BIM is just too generic and eventually misleading. BIM in-fact stands for a multitude of operations related to a building's design (Eastman, 2011) and there are many segments of this discipline for which we can use different specific BIM software. It is important to remind the reader that this work will concentrate on the 3d modeling aspect of BIM specifically of BIM Design Tools.

Currently there is no such a thing as a singular computer program that can be implemented and used for all the complex operations related to Building Information Modeling, therefore the abbreviation BIM for any software is usually misused in such context. There are however some design tools implemented in the BIM workflow that were created specifically for the purpose of architectural modeling that allow the implementation of some building information like construction details for instance. We refer to these tools as BIM Design Tools (Eastman, 2011). These tools have in common with each other, the fact that they can offer a lot of similar embedded functionalities; To name a few these are dynamic "3d to 2d" views, the possibility of drawing using, instead of simple lines, parametric building elements (like doors, windows, etc. which attributes can be controlled and replicate automatically throughout the project) and also the advantage of creating automated drawing schedules.

There are also more standard generic purpose CAD tools (mainly Geometric System tools) that are implemented and adapted to the modeling aspect of the BIM workflow because of their features, and general flexibility in facing these types of tasks. These software mentioned at last are usually known for their customizable functions, their modeling flexibility and more over because of their powerful data exchange possibilities that also allow the designers to move building information freely from a software to another.

To brief it up we can say that among all the BIM Design Tools, there are some specifically created for architectural modeling which provide the user with specific functions for designing architecture (we will refer to these functions as "Native Functions") and some tools that are created for generic purposes but that eventually can also be a valid option for architectural design when customizing their functions (we will refer to these functions as "Custom Functions") (Figure 2).

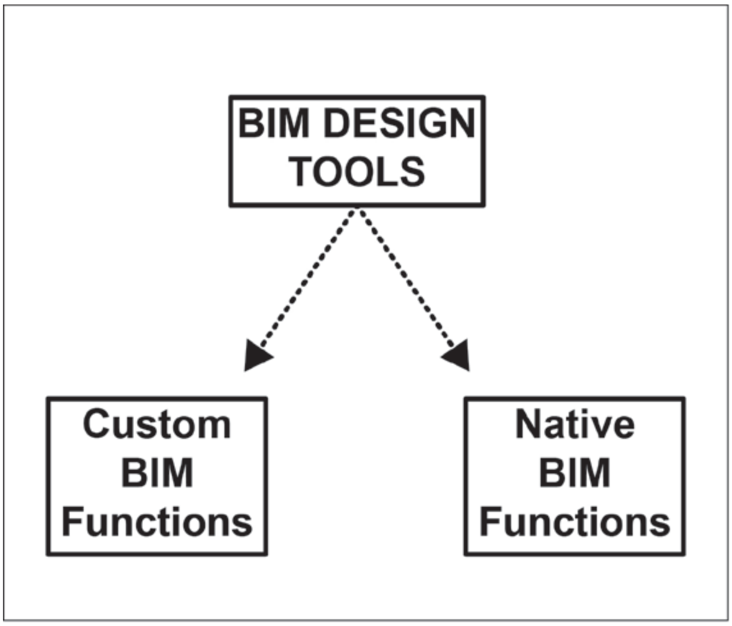

Figure 2. Diagram of BIM Design Tools (by the author)

The main producers of BIM Design tools natively dedicated to architecture are Autodesk with the Revit line and Autocad Architecture, Bentley with the Microstation V8 series, Nemetschek with

Allplan and Vectorworks and last but not least Graphisoft with Archicad.

Instead, as more generic purpose tools used for modeling architecture in BIM, we find McNeel with the Geometric System Rhinoceros 3d, Google with Sketchup and Dassault Systemes with Catia. As already mentioned, these solutions were not necessarily designed for architecture design. Catia for example does not fall at all in the AEC category as it is actually a solutions borrowed from the aeronautical industry and adapted to architecture. This program has the ability to carry a very large amount of building information thanks to a more advanced data base system that works differently from other tools. Late efforts carried out by "Gehry Technology" allowed also the development of plug-ins that enhance and at the same time ease the utilization of Catia in the architecture field.

There are many companies and individuals that have been trying to develop specific building information modeling plug-ins for BIM software such as Rhinoceros 3d. Very recent developments have brought us also "collaboration only" tools such as Naviswork from Autodesk, which is basically a digital container where the structural, the architectural and the mechanical 3d model of a building can be put together for review. Naviswork allows the 3d navigation of a project and collision check of the different model parts merged in it and also it allows the real time communication of the revisions needed, between different professionals involved in the same design project. Naviswork however falls in a category of BIM software which is in between the design tools and the analysis tools.

In the commercial realm there is really no point to show the main differences between these digital tools. Many people use one program over the other only because they got taught that way in school or at work (Ibrahim, 2007). So to be clear the advantages of one brand over another are always balanced by small additional features not present in other concurrent packages. This makes these products quite equivalent with each other.

1.3 Designing with BIM

In BIM the three tasks related to building projects/planning, design, and construction- are often considered together and in a collaborative effort, because they all occur in a relatively short time just before the occupancy of a facility (Kymmel, 2008).

When working with BIM tools the most important thing -when documenting a project in a collaborative way- is to create a very well detailed 3d digital model (Figure 3) where all the information of a building can convey from all sides.

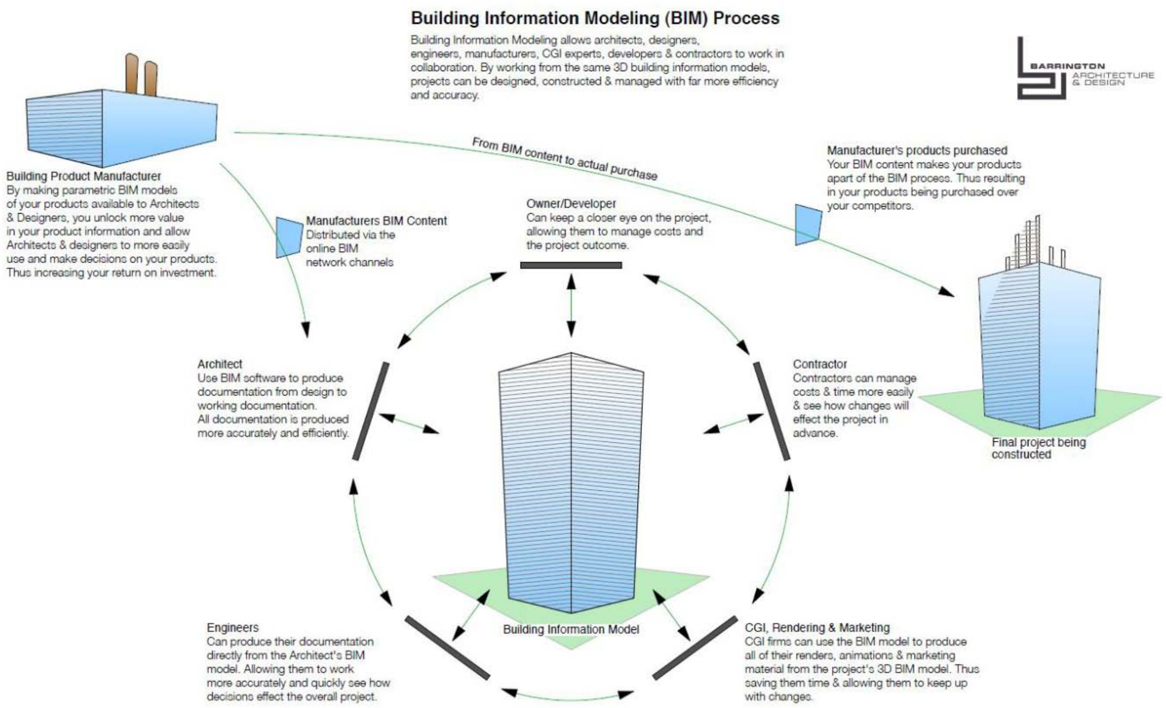

Figure 3. Barringnton guide to BIM (Barrington 2011)

The building assembly is usually responsibility of the architect. He or she has to make sure that all the data, produced from all the other professionals involved, fits inside the model without any mistake or collisions, therefore an important aspect of working with a BIM model is the communication done in 3d rather than via 2d plans or sections.

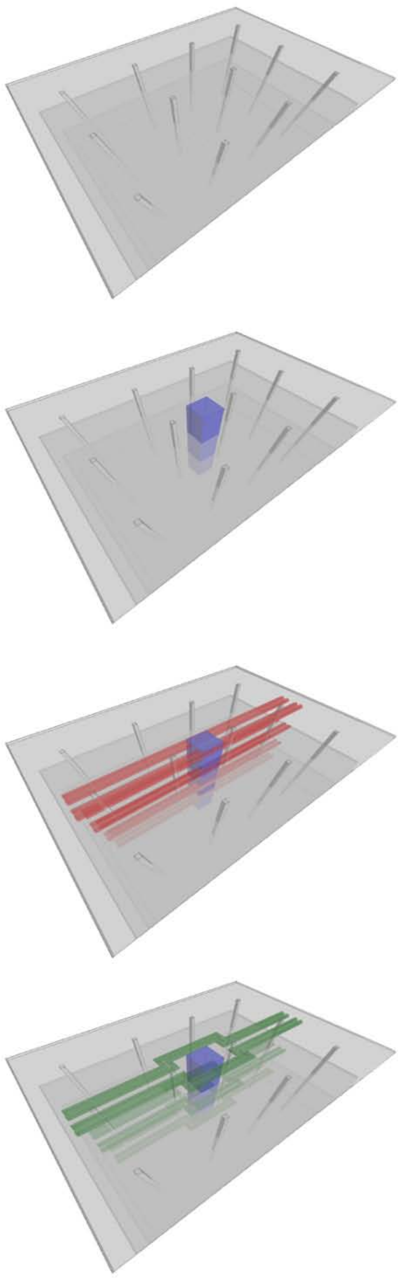

Figure 4. Detection of collision in BIM (by the author)

If an engineer needs to check where pipes and cables need to be wired in a building or see where these collide with other building elements, there is no better way to do this than in a 3D environment (Figure 4). It is therefore vital that the engineers have knowledge and access to the same type of software of the architects. In this case the architect should make sure that the engineer is able to work with this pipeline without any problems.

We can start to understand the dynamics of work that can be supported by BIM and we can think of unlimited different ways of interact with different professionals. This is also why some offices are more productive and stable than others when it comes to work management and risk assessment. However the real key to success of implementing "Building or Architecture Information Modeling" is to learn how to push the boundaries of these tools. In-fact with the software we have available nowadays, the main goal is to explore and "invent" new designs with the awareness of how to be able to build these structures in real life. If implementing BIM, This should not be a difficult task as the architect shall also be enabled to produce, with the same software, numerical information for digital manufacturing. In-fact CNC (computer numerical control) machines can read BIM data, making it possible for architects to have the highest degree of freedom when creating building components (of course only if the information created during design proves itself to be consistent). The right collaboration with engineers can then bring these ideas to real life and there are -from this point of view- very little limitations to creativity.

In a practice where there are more traditional methods implemented, a lead architect or a project manager would develop the first sketches, then they would draw plans and sections and continue this way creating no parametric relationships between all the documents produced. When all the 2d drawings are finally done, a facade will be composed. At the same time some CAD technicians and drafting assistants will be working as satellites going around different projects and making different digital 3d models (for rendering and illustration only) that carry no building information (Figure 5)

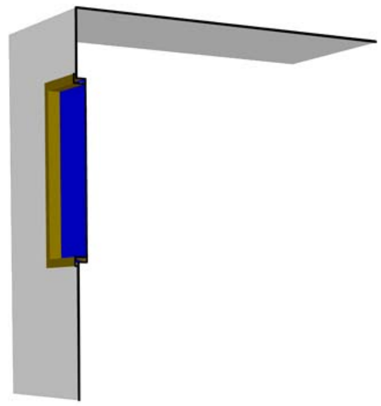

Figure 5. Model typology usually produced for rendering only (by the author)

as they represent only the data developed -in a single instances- by the act of checking against always changing plans, sections and elevations. This type of interaction is time consuming, it presents many gaps for human errors and requires much more effort than working with an integrated BIM environment.

Instead with a BIM based approach, a lead architect can also start a project with a plan or even directly with a massing study as the geometry created will always carry both 2d and 3d information needed for further documentation of the design

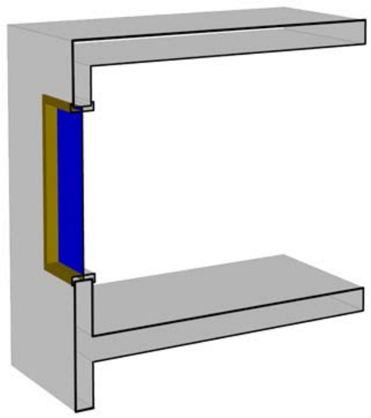

Figure 6. Model typology produced so to carry buildable information (by the author)

(Figure 6). At the same time if technicians are working as satellites on different tasks, the lead architect can keep an eye on all the modifications amended to the main project as this is composed of a single entity that is not spread across different folders like it happens with more traditional CAD based methods. These techniques of course vary depending on the software used and the scale of the project itself, however in both cases, file management is the main responsibility of a project manager. Even if BIM allows a better an easier supervision of the projects, it is still expected the project leader to be the responsible for any changes amended to a building during the design process (Kymmel, 2008).

BIM tools have automations scripted at the core of the program itself that allow a discrete file management compared to any other CAD application available to architects. The project files are organized in a structured way that cannot be modified without being noticed by collaborators in a team and many drafting operations are automated in such a way not to create conflicts specially in drawing schedules.

Many deduce from a deep observation of these facts that BIM is just a fancy word to indicate a type of software for "Architectural Project Management" (APM).

This assumption is quite legitimate as the functionalities of BIM over generic purpose CAD are specifically oriented towards project management rather than only 3d modeling.

In fact one has to consider a BIM model as a container where all the information produced in a collaborative way can be then be put together to check errors, retrieve building information, calculate costs and also plan ahead the life cycle management of a building (Doughty, 2011).

The digital "Building Model" (Figure 7) produced with BIM is used as an instrument where it is possible to combine the work of an engineer, an architect and a product manufacturer (i.e. electrical, AC, windows, doors and all types of other contractors).

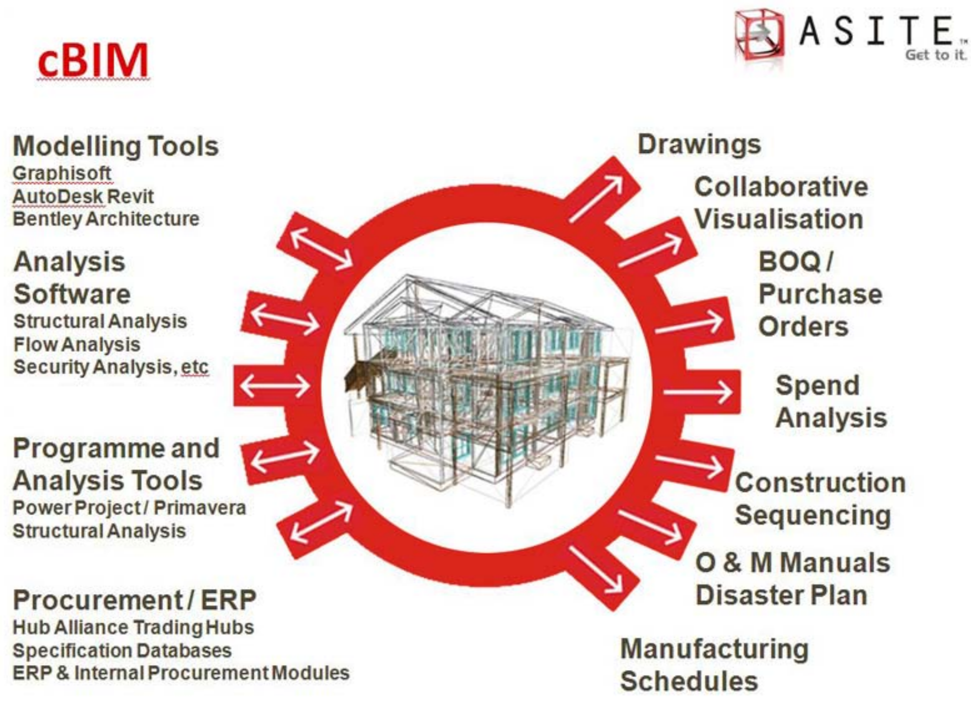

Figure 7. BIM Model interaction (asite.com 2011)

This does not mean that all the elements modeled in this single file, were produced with the same software. This is actually an important distinction that one has to do in order to understand that BIM is about information exchange and file management and not only about digital 3d modeling. A +BIM model becomes a a virtual place where a digital building can be checked for mistakes before going to site (Doughty, 2011).

1.4 Limitations of BIM

The main marketing strategy of most CAAD (Computer Aided Architectural Design) software available to the public, is to show that the functionality of their newest products is essential and vital to the development of innovative, cutting-edge architecture.

BIM is currently the most contemporary family of architectural software of this kind.

There is currently no software on the market that can provide an architect, an engineer or a designer, with all the functionality needed to produce a design from the initial stage all the way to the final presentation. This is not a problem related to BIM only, we could actually say the same thing about many other software categories. Some think this is a market strategy wanted by the manufacturers, some others believe that the current capabilities of the actual computer systems are quite limited to support all the data required by multiple applications. However it is strongly believed that architects would be the professionals that will get the most out of their BIM tools if these were rightly implemented (Figure 8).

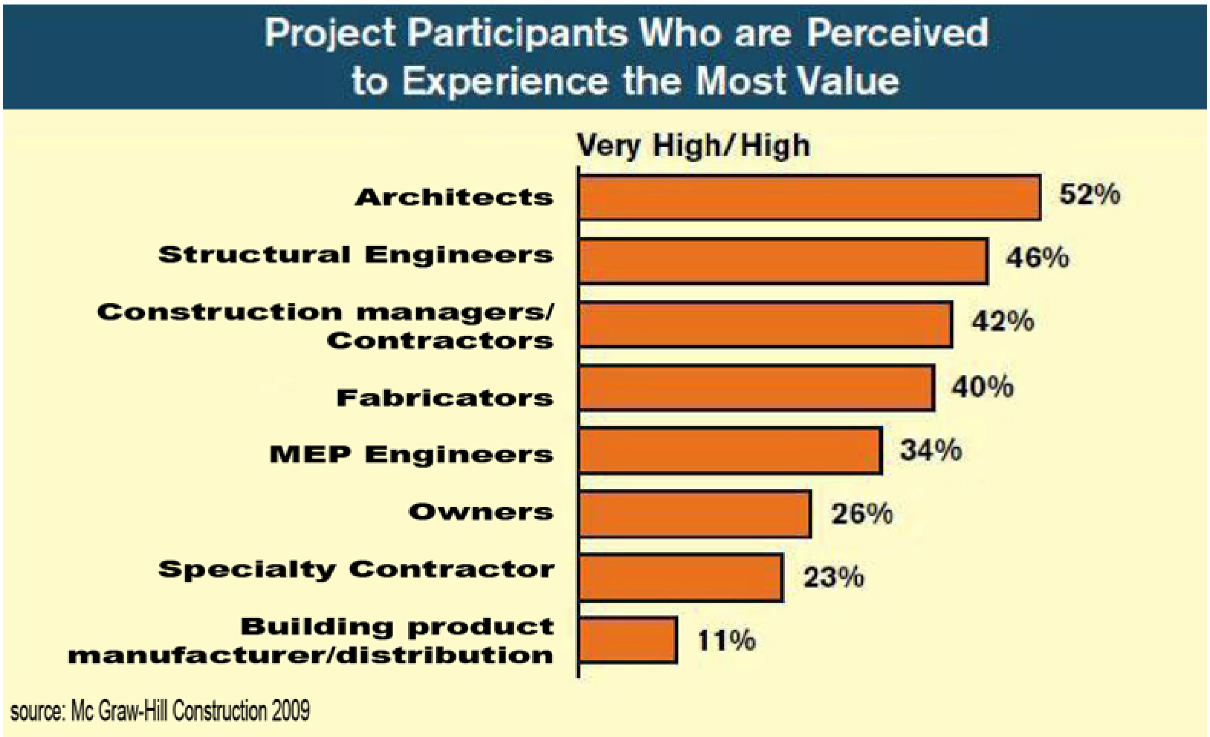

Figure 8. Expected rate of satisfaction of professionals using BIM (Mc Graw-Hill Construction 2009)

This again does not mean that architects will only need this one specific package to do their job, it is inevitable that a larger number of tools will be used sometimes contemporarily to get to the final goal of this design process. Said so it is vital to establish a solid workflow (file exchange, file exporting, file referencing and file storage) between all the professional -involved in the design of a building- in order to insure zero conflict between the different branches working on a single project (Krygiel, 2010).

Currently the main issues in this field is the shifting from the 2D media to the three dimensional one and also the training of different specialists in the use of specific 3d modeling tools. In fact most of the professionals involved in design are -as already mentioned earlier- working in 2d format and this does not allow them to take full advantage of the most contemporary BIM functionalities.

Structuring well the designer workflow, in such a way to make use of the building information modeling techniques available, it represents eventually phase one of starting working with BIM tools (Krygiel, 2010).

Many new features that have been introduced in these software, do facilitate a certain degree of collaboration among designers and also speed up the design process itself. This happens via new data-base systems and parametric building components that completely replace the most common drafting CAD tools. However many of these design features and more over modeling capabilities, are also limited by the constraints set by the same database functionality implemented in the software itself.

One of the aims of this Thesis is also to assess -in an objective way- the advantages of modeling architecture in a 3d digital environment regardless of the software used. In doing so we shall be able to evaluate the true advantages given by native BIM applications over custom ones.

We will now test native BIM modeling functions against custom ones. The tools in trial will be Revit (native BIM) and Rhinoceros 3d + Grasshopper (a customized combination of a multi-purpose Geometric Design system and its available plug-in).

With this test we should be able to understand and evaluate the advantages or disadvantages of working on architectural modeling for building enclosures using these two tools. Before continuing with our test we shall introduce first the basic requirements for enclosure design support.

2 Requirements for Enclosure Design Support

We will now look at the basic requirements for enclosure design support. As already explained we will aim at recreating the irregular facade of a building (The Gasometer B in Vienna) and its side connection so to see how these components can come together using the two different tools Revit and Rhino. We will aim at creating a digital model that will enable the user to iterate from the massing of the building through its schematic enclosure layout, all the way to the details of the enclosure components and back again to the original massing. Ideally for each single iterative step of this process the user shall be able also to implement different types of building analysis by integrating these inside the design tool or by exporting data to another application.

This kind of feature should enable the user to evaluate different design options also based on the feedback recevied from the analysis done. This should be possible at any stage of the project, keeping the parametric changes of each single design step (i.e. massing design, schematic enclosure design, detailed enclosure design) independent from one to another (Figure 9).

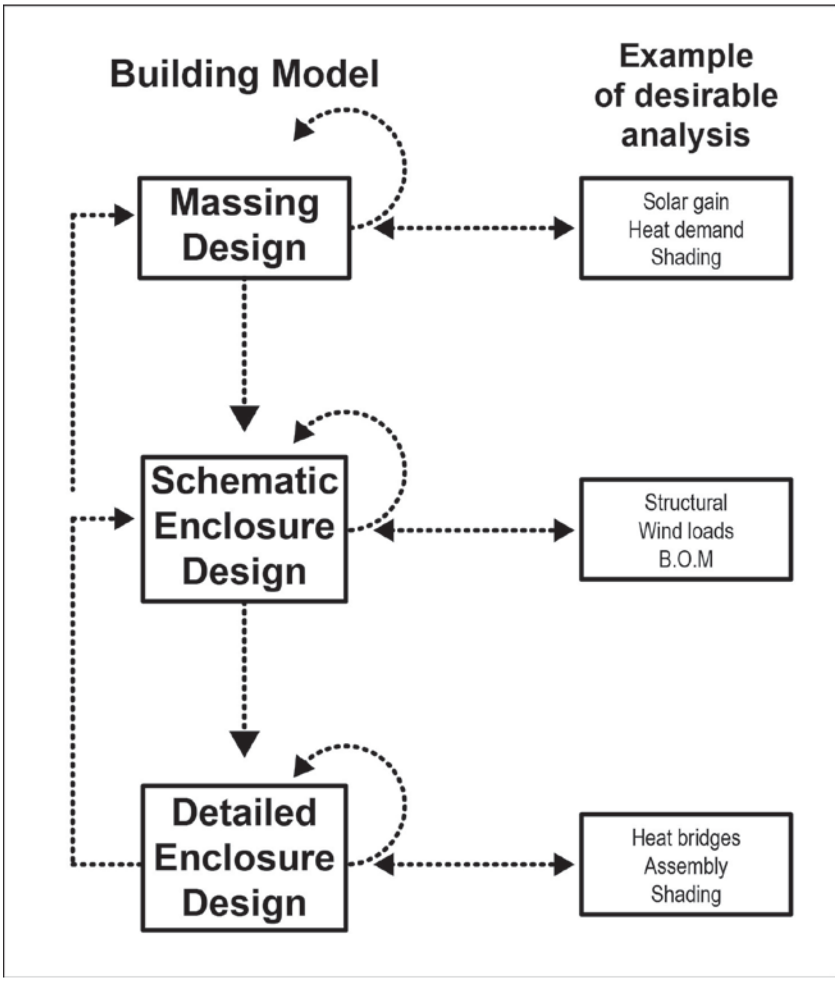

Figure 9. Iteration model (by the author)

The goal is to create a 3d BIM model so to asses, in a holistic way, the modeling functionalities of the two types of BIM Design Tools described in the very first chapter.

In the specific the capabilities of BIM, on which this study will focus on, are parametric design support (where we will look at the iteration achievable with such tools), performance analysis support (where we will look at the type of analysis that we can integrate within the design and how the user can iterate with these) and data integration exchange (where we will look at the quality and the amount of data that we can implement in the model and how this information interacts at the different stages of the design).

Thanks to the possibility of retrieving information from their database, most BIM design tools offer a higher degree of automation of architectural design related operations that is not available to generic purpose geometric systems or solid modeling software which however, as already mentioned before, are usually also implemented in the BIM workflow. One of these operations for example is of course wall and curtain wall design. The structure of most BIM design tools specific for architecture is very stiff and modifying its content and functionality requires the user to have real advanced programming skills.

Instead, geometric design systems can offer a higher customization degree, allowing the user to achieve more specific results, such as for instance flexible massing design and paneling automations.

This is possible thanks to lighter system structures and more over by the effort of more people related to different disciplines working with these programs. Including detailed project information in these types of environment though is more complicated due to volatile data that is not stored in any database apart from the project file itself ("BIM Modeling Blogspot," 2010).

In light of these statements, it is better to start with testing Revit first so to show the kind of modeling functionalities we can achieve with it and then see how the user can respond to that with a combination of Rhinocero 3d and its plug-in Grasshopper.

3 Case Study Description

3.1 Location

Our case study is the Gasometer B in Vienna. This building was designed by "Coop Himmel-blau" in 1999. Located in the 11th district south east of the Austrian capital, the Gasometer is part of a new development of the borough of Simmering.

3.2 Function

The building currently hosts apartments facilities, office spaces and communal areas with a bar and a sauna for the people living there. At the bottom of the building people can access also a concert space where very often bands play music.

3.3 Design Concept

The building is composed of a huge old brick cylinder (a gas container of the late 19th century) that has been renovated and connected to a new concrete structure that faces north (Figure 10).

Figure 10. North-east view of Gasometer
(http://www.nextroom.at/building.php?id=2616)

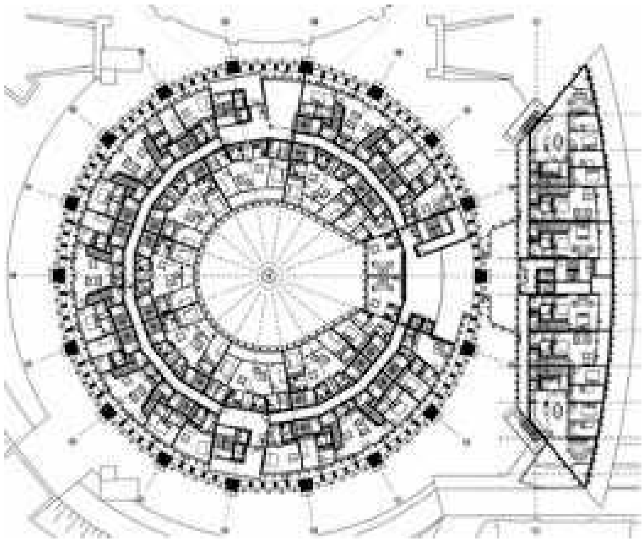

Figure 11. Gasometer plan
(http://www.nextroom.at/building.php?id=2616)

This building is a very good example of how new architecture can be incorporated with the old one and how this fusion can create new standards in the field of conservation and use of old structures. The new building has a very unconventional plan (Figure 11) with inefficient space at the corners. The two small facades at the east and west side, they are connected to the main facade via very tight angles creating very ambiguous spaces at their extremes.

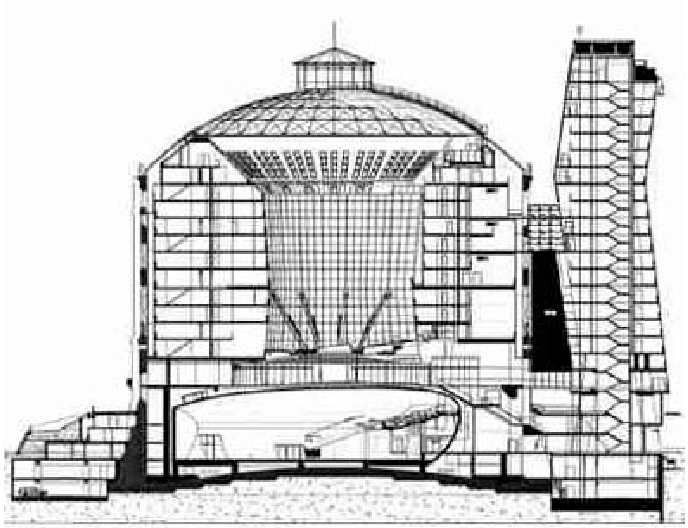

Figure 12. Gasometer section
(http://www.nextroom.at/building.php?id=2616)

The straight side is connected to the old structure (the round block) by a bridge located on the 9th and

10th floor (Figure 12) but also at ground level with a large atrium.

3.4 Massing

The massing of the building can be interpreted as two opposite intersecting cones one on top of the other trimmed at the edge or as a lofted surface generated from three consequential curves. There is a round and slanted side on the front, and two slanted surfaces at the sides, also the plan of the building opens up like a fan and all these added geometric features might not allow the full parameterization of the building volume/shape (Figure 12).

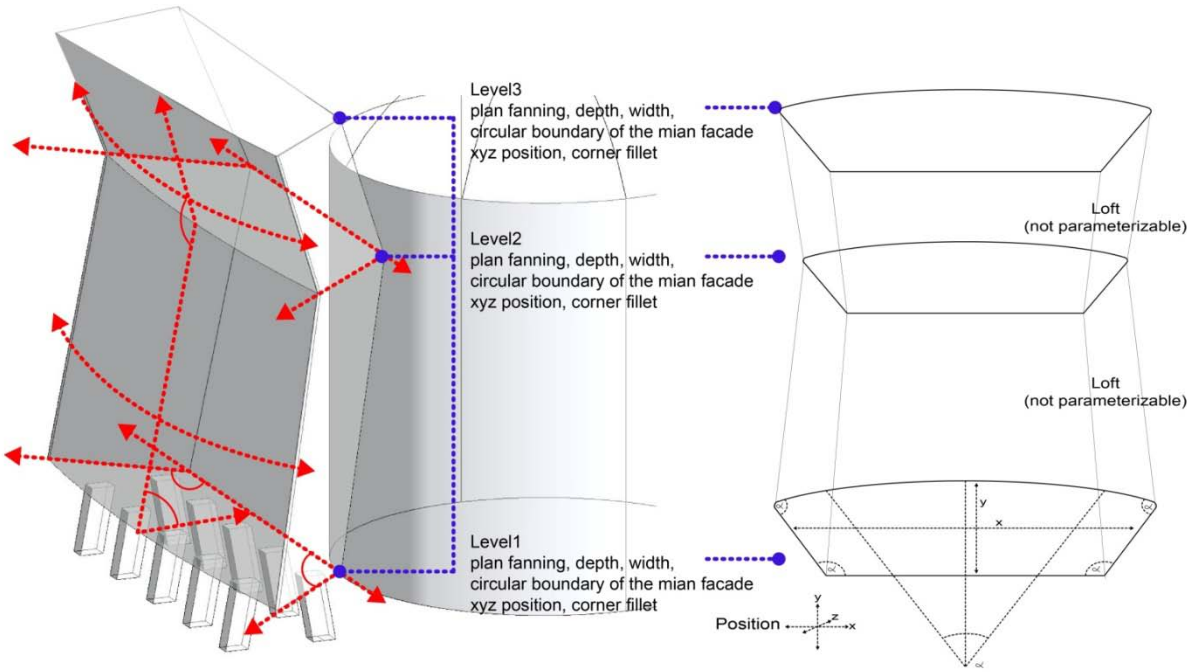

Figure 13. Massing of Gasometer (by the author)

From this point on, It is very important, when modeling in BIM, to define and categorize the building components we are dealing with beforehand, so to allow better planning and implement the best parametric relationships among the building components (Eastman, 2011). The Gasometer "B" is a very good architectural example containing plenty of different building components that, if related to the concept of building families in BIM, they are quite difficult to categorize.

3.5 Enclosure

The building facade of this building, even though it is curved and slanted, it has been subdivided in an orthogonal raster (Figure 14). This layout is quite simple to achieve: the designer only has to draw one straight vertical line and one horizontal, then array them in relationship to the center of the geometry. Afterward these lines can be projected on the surface and the grid is so created.

Figure 14. Gasometer main facade (by the author)

As we can see from the regular panelization of the facade, the panels dictate the visual rhythm with obscure and clear elements. Before continuing with the description of this facade it is important to say that no readable detail of its construction was found during the research process, therefore it was not possible to make further assumptions in regards of this matter.

If we take a closer look at our building we can see that the components we are dealing with can be categorized as:

- Walls
- Curtain walls
- Windows
- Curtain panels (subfamily of the curtain wall)
- Special structural wall elements (abstracted family)

The main Facade (Figure 15) is laid out in a 2 horizontal rows module. The top row has 5 horizontal main elements (solid/ window/2glass panel/window). The lower row has a large insulated panel (2 elements from the top row) and it has a rhythm of 2 elements at the time. The corner of the building has instead a round element divided also horizontally in 2 elements connecting one facade of the building with the other.

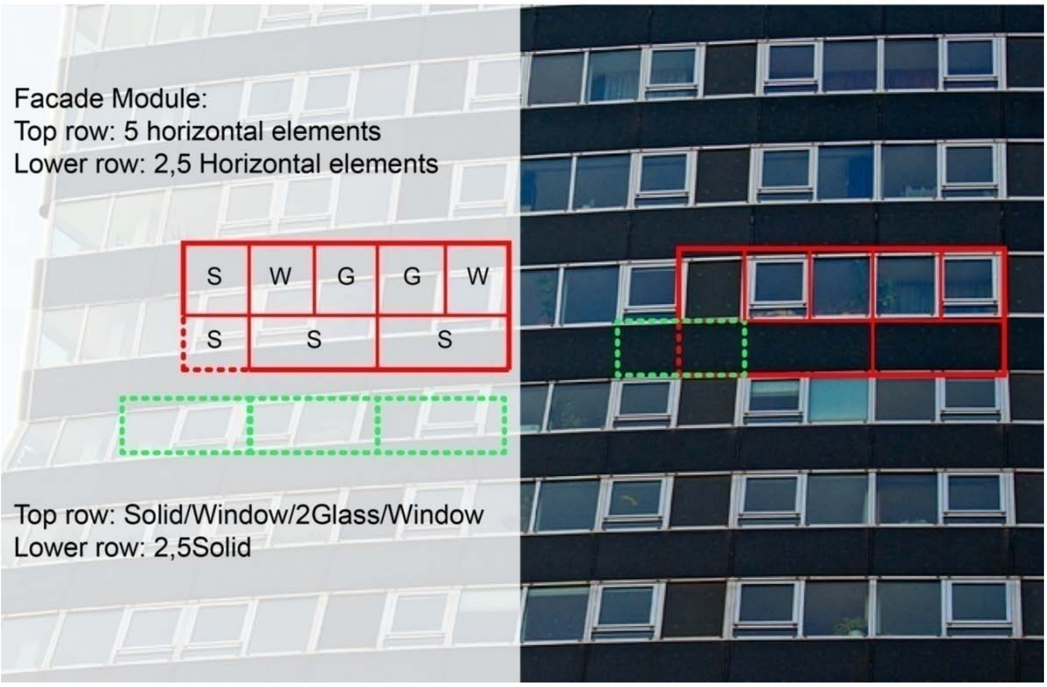

Figure 15. Main facade module of Gasometer (by the author)

The side facade is connected to the main one with a round element. The connection happens with a skewed glass/window/glass element, on what appears to be a simple wall cladded in zinc (Figure 15Figure 16).

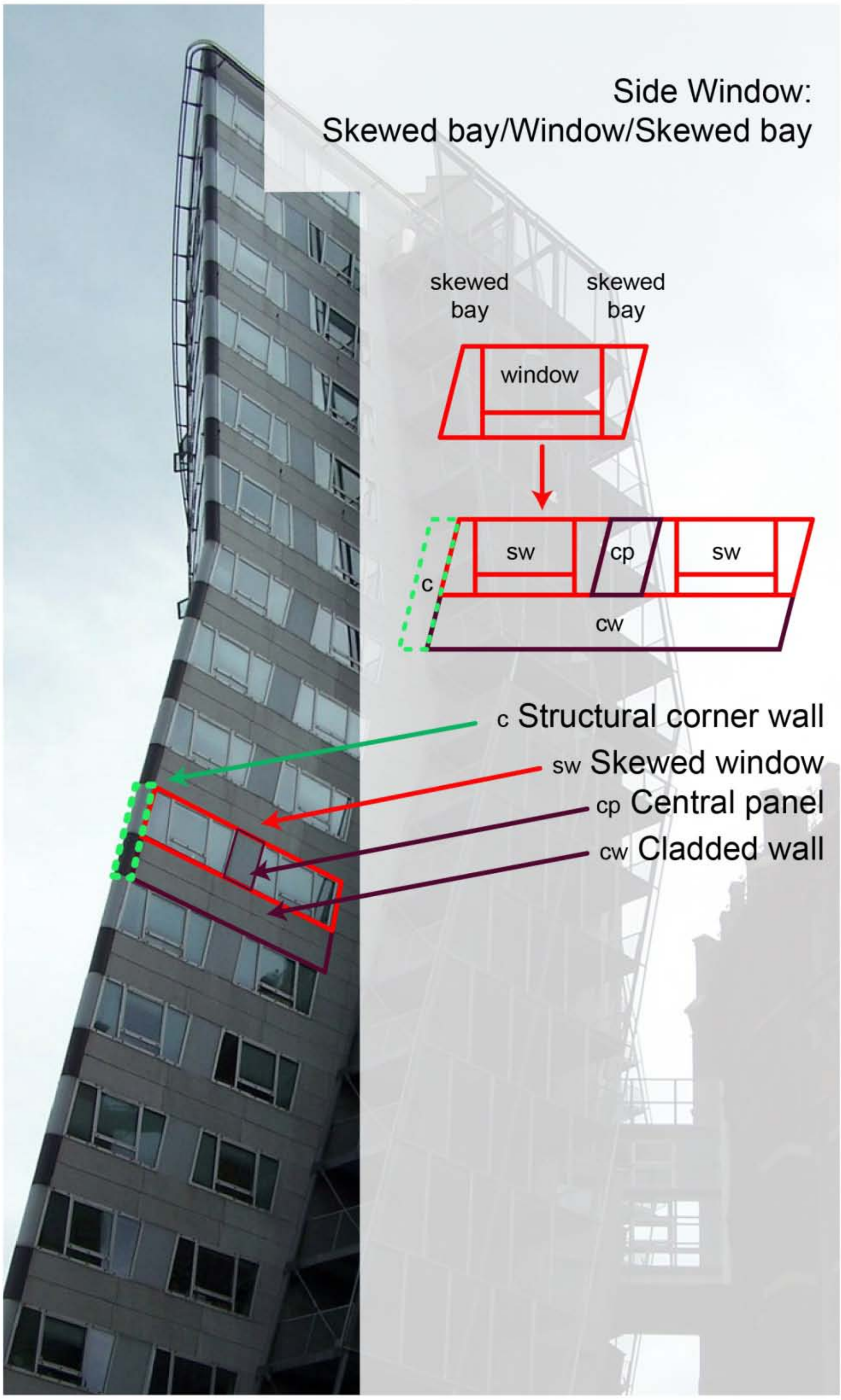

Figure 16. Side facade module of Gasometer (by the author)

In native BIM solutions, Inserting doors and openings in a model is very simple as long as we stay on a vertical surface. Slanted surfaces or organic round ones, they need a modification of the opening component (door or window) as usually these are hosted on the profile of the wall line that is then projected on the working plane.

Therefore in this case, by concentrating us on the facade of this structure, we will be able to see how much information can be parameterized so to create an iterative relationship between different design options of the same building. We will concentrate our effort specially on the corner of the building's envelop, where there are at least 5 different building components assembled together.

By working with windows, panels and the corner component we will work on modifying parameters and adjusting height, width, materials and all the attributes of the assembly. By creating an iterative model, we will understand the parameterization of such elements and the advantages of such approach.

This test will demand an advanced knowledge of these tools and specifically of the concept of parameterization of the building families. Key in implementing such tools -in a performative way- is to be ahead of the program itself understanding what is it behind its logic of function (BIMandintegrateddesign.com, 2011) so to tweak it and to make it respond to the real needs of the designer.

For the purpose of this test we will use Revit from Autodesk and for Rhinoceros 3d from Mcneel, with its node based scripting platform Grasshopper (to be more specific we will use an hybrid combination of Rhino and Grasshopper).

4 Enclosure Design using Native BIM Functions

4.1 Massing Design

With Revit it is possible to model buildings for three dimensional surveys, but it is difficult to implement real construction data in doing so, unless this is provided by the component manufacturer himself. Much of this information can be however replaced by plain 3d models and then implemented in 2d. For the purpose of this experiment we do not have real survey data available, we will therefore limit this exercise to the understanding of the building relationships created among the construction families that we are going to use for our project, so to also implement a certain iterative behavior of the model itself.

As a first instance we will look at the building envelop and then at the main facade, so to understand the way we could lay it out.

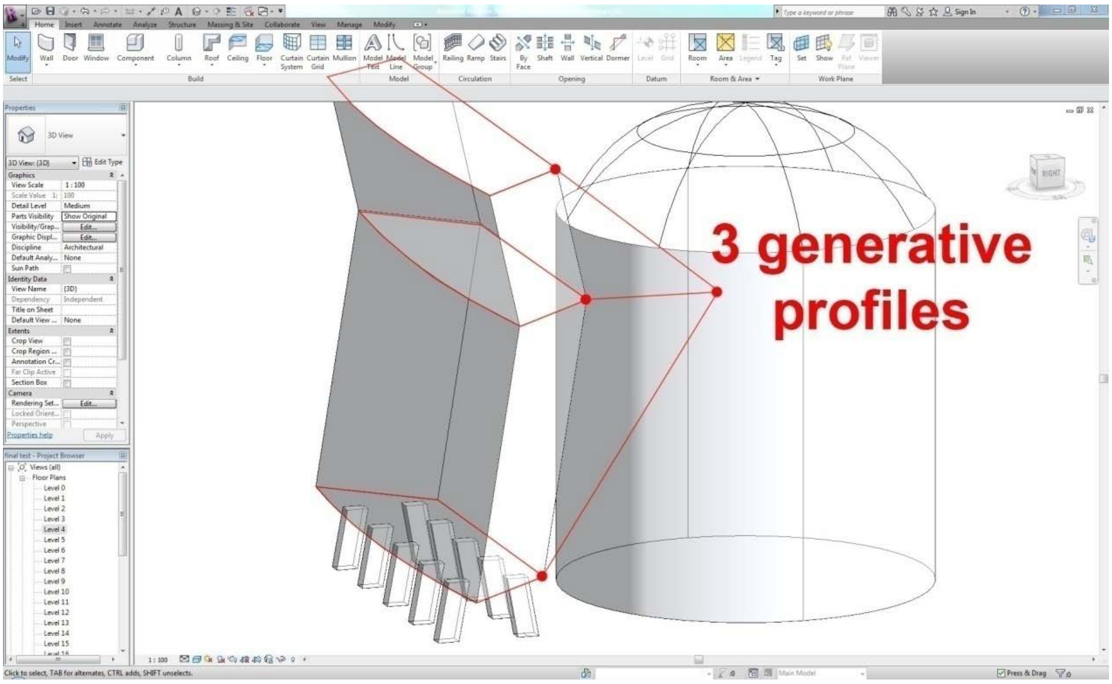

Figure 17. Massing from 3 profiles (by the author)

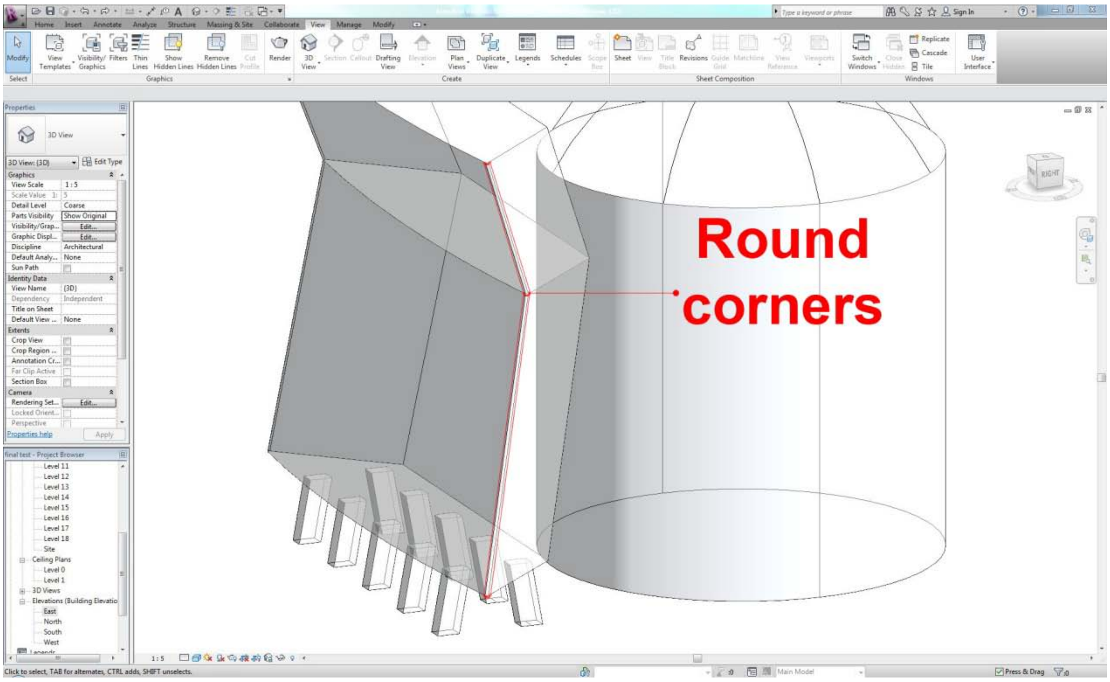

Figure 18. Massing with round corners (by the author)

By generating the main body of the building with 3 main lines (Figure 17) laid on 3 different reference planes, we had to deal already with the issue of the round corners at the side of the structure. Later on this might be difficult to correct as walls are generated as a projection of the "normals" of the envelop itself meaning that a change to the corner means having to redraw all the walls again. When the massing is ready this can be exported for a variety of analysis such as solar gain, heating demands, shading studies of the massing over the site and so on.

4.2 Schematic Enclosure Design

We can now look at ways of creating a grid for the panels across the facade. The easiest option would be to generate a curtain wall and then divide it into the window module. As we discussed already at the introduction of this case study, the issue that we could encounter when modeling the building components of the facade is its actual layout.

Current architectural design software solutions in general would allow an automation of the paneling system using the UV coordinates of the main surface hosting the panels, instead of using the system implemented for Gasometer (Figure 19). This is a natural optimization system.

Creating such a window layout nowadays with current software would need a thorough break down of the geometry of the facade so to planarize the elements that compose it.

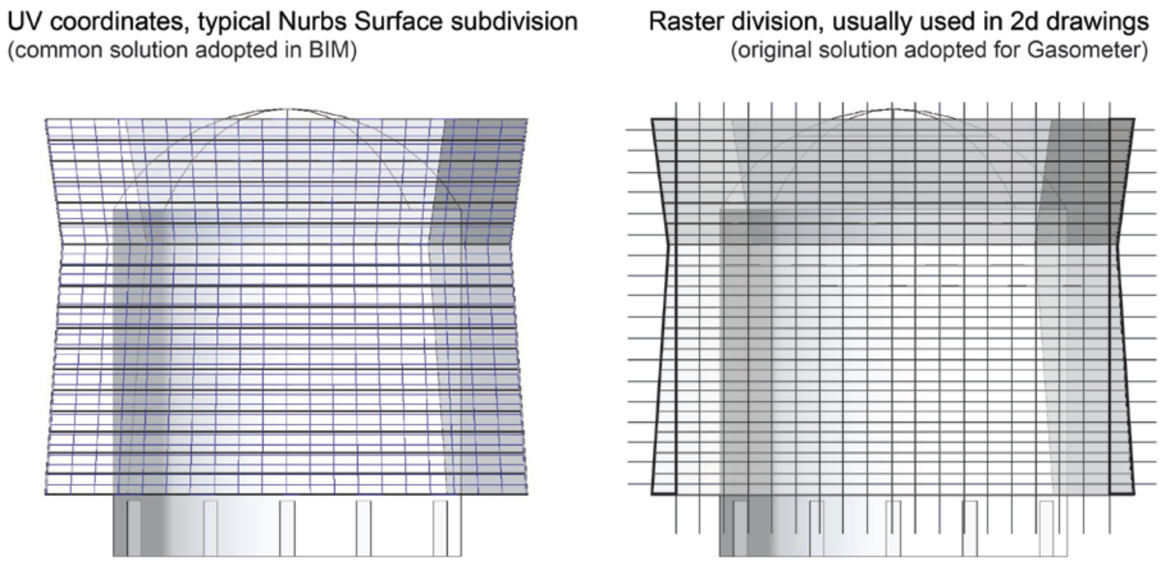

Figure 19. UV subdivision (left) and orthogonal grid subdivision (right) of a surface (by the author)

To proceed in this way we will have to define first all the floors to create a parametric relationship between these and the future apertures across the facade (Figure 20) and then the number of UV subdivisions.

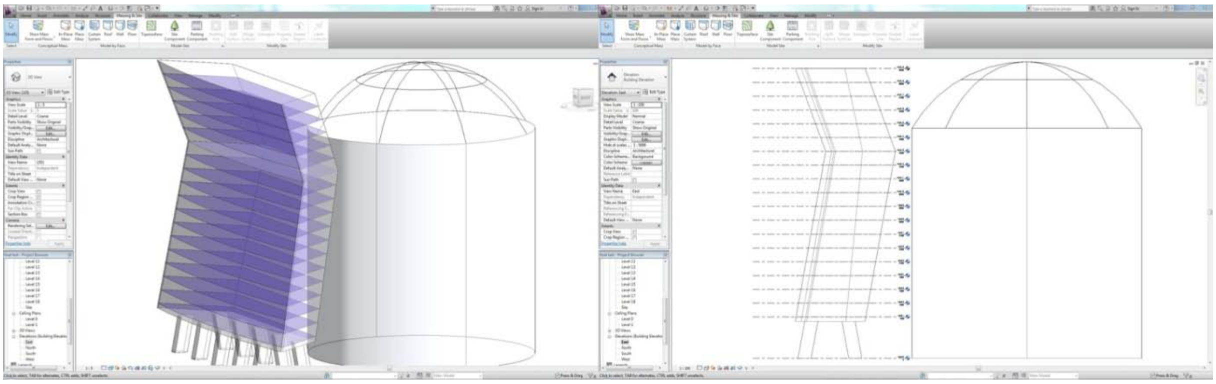

Figure 20. Generating floors based on the massing of the structure (by the author)

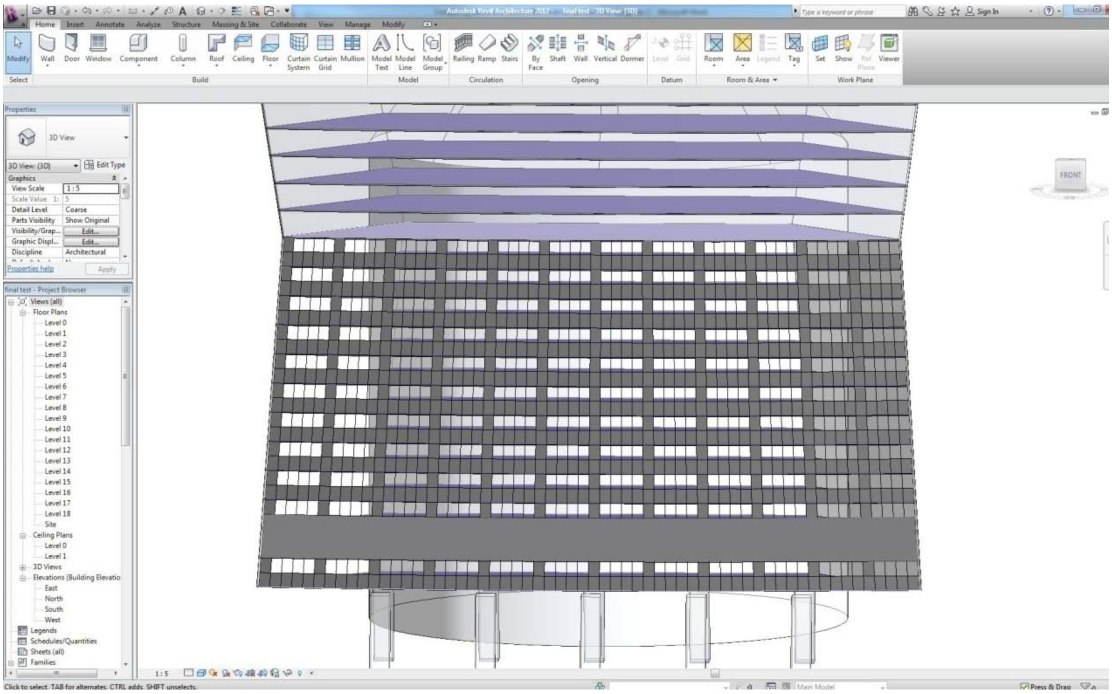

Figure 21. UV subdivisions (by the author)

The subdivision exercise unfortunately leads us to failure as the windows get squashed inward by the UV coordinates of the building envelop. We would have to go and slice the facade one floor at the time to achieve more precise coordinates. This however means that the automation provided by Revit would break creating a non iterative relationship between all the windows and the panels. In a case like this the implementation of the paneling is limited also to the single panel element. This means the designer should go to each single panel and control its properties independently.

At this point we could take a single floor at the time and try out if this separation can give us a better subdivision system. To do so we will have to create a new host for the panel grid (Figure 22).

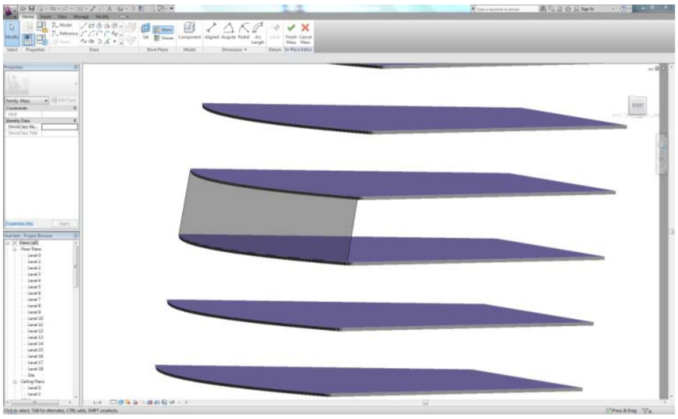

Figure 22. Subdivision of the enclosure in floors (by the author)

This way we could fake the curvature of the window but we would still lack the continuity of the grid. Also the processes used until now have very little parametric properties because of the unrepeatability of its element and also because of the fact that we are operating on an organic surface, which it makes it very complicated for the software to deal with placing specific subcomponents. Once the layout is created this component should be exportable for analysis like structural for instance. The bill of material can be executed within Revit.

4.3 Detailed Enclosure Design

We can now start replacing the grid panels (Figure 23) with glazing and windows.

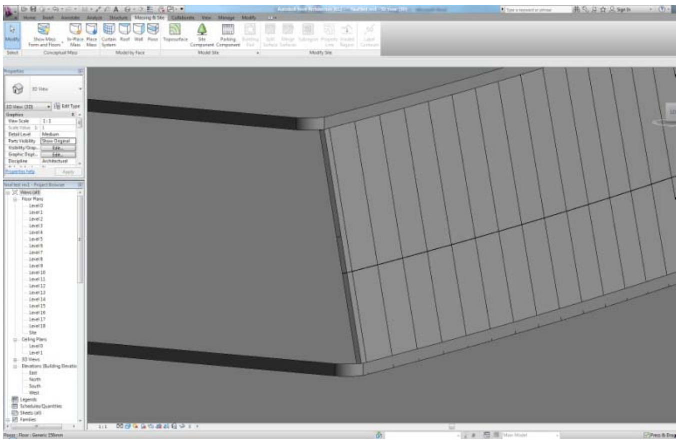

Figure 23. Simple paneling applied (by the author)

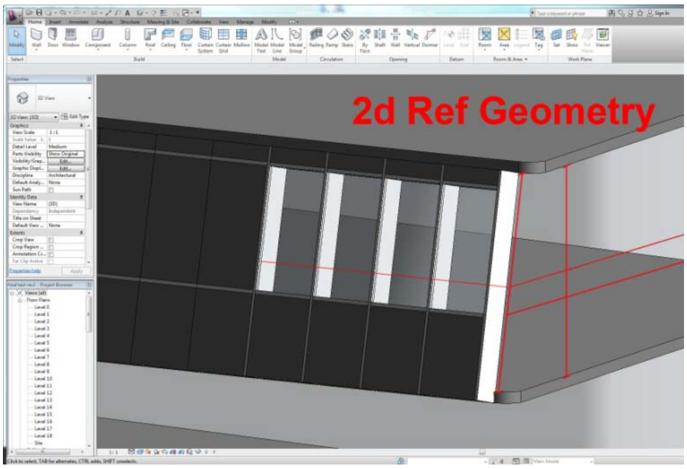

Figure 24. Paneling replacement (by the author)

Since the placement plane of the mullions is slanted, we can only use reference geometry to understand more or less the height for the insertion of the building components (Figure 24). However the overall precision of this process it is insured by the specific information carried out by the properties of the curtain wall itself.

Curtain panels are a quick way to create repetitive systems across a building facade. However in most cases it is better to create a specific window bay component and place it manually. In our case we were forced to create a curtain wall also because the placement of straight elements across a curved surface would have been very difficult. We can now add the horizontal mullions (Figure 25).

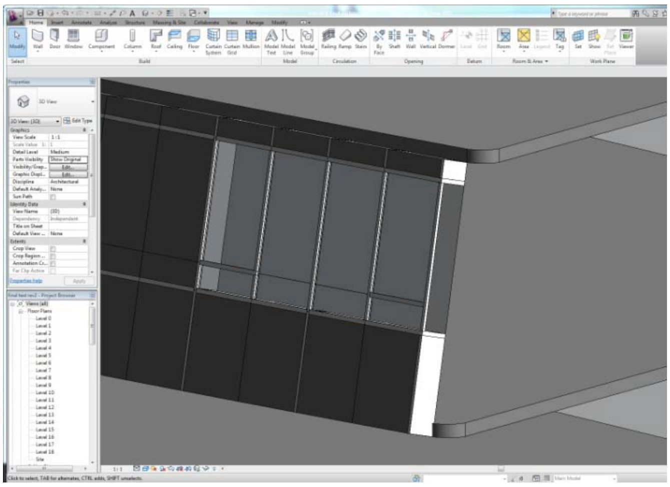

Figure 25. Mullion placement (by the author)

As representation media, this is however a truly valid option also because at this point to evaluate a real construction solution we should consult with a window manufacturer in order to understand their assembly method (Eastman, 2011).

We can now work on the corner component. As we already specified it before, this element can be replaced by a simple blend shape between the two slabs. The reason for that is that in our case, we are dealing with a complex building part that is not classified in standard BIM families. The designer meets at this point one of the limitations of these tools, where native BIM modeling cannot provide him or her with enough functionality.

Figure 26. North west corner of Gasometer (by the author)

If we look at the corner of the "Gasometer" (Figure 26) we see that this component is really particular in its shape because of its geometrical properties but also because it cannot be used generically in other more conventional buildings. Recently many

Figure 27. Detail of the building corner (by the author)

BIM tools have been enabled with massing capabilities that can overcome such issues, this means that the shape is achievable without having to leave the native BIM environment. If we look at the corner of such structure (Figure 27), we can convey that the building component that connects the two facades is not a wall (Figure 28, Figure 29), not a window and not anything else available from the catalogue of traditional building components of the software.

Sometimes since the construction information is hard to implement in such specific building components, the designer has to specify it with extra "two dimensional" drawings attached to the 3d model (Figure 31).

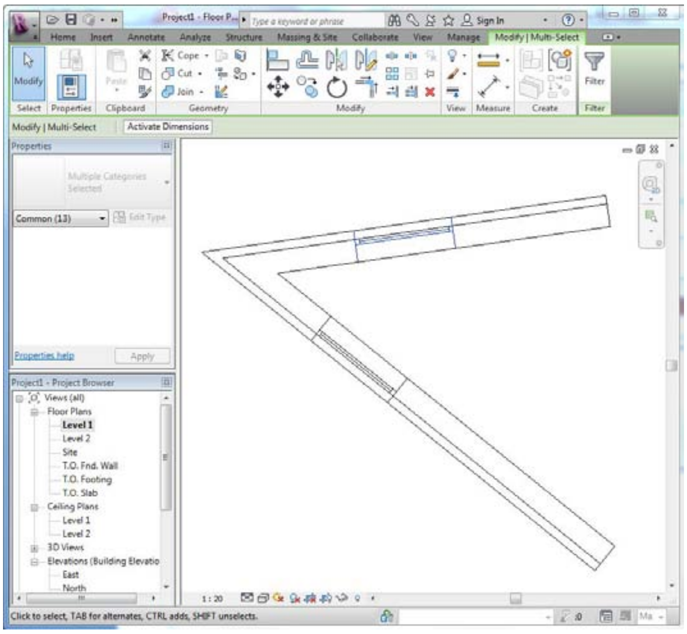

Figure 28. Corner created automatically in Revit (by the author)

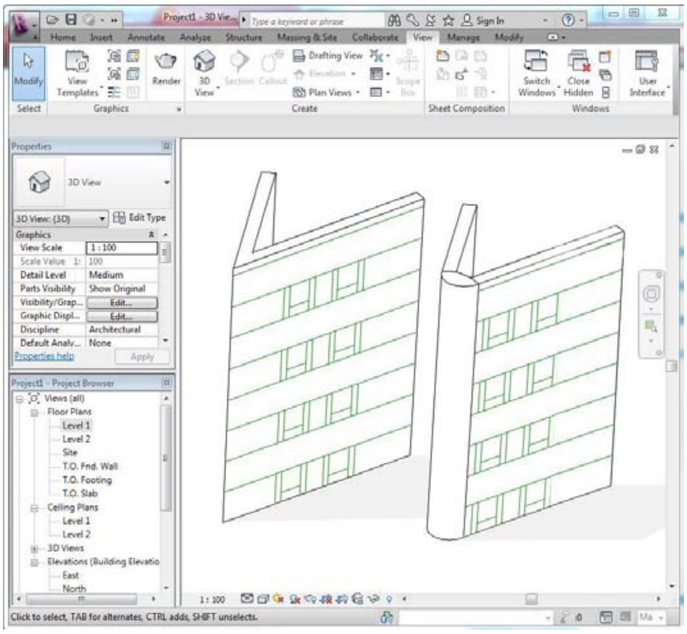

Figure 29. Difference between automatic created corner (left) and the corner with round solution (right) (by the author)

It is very important when working with such components to identify the way these parts will be assembled on site in reality. Therefore the 3d modeling can be a simplification of such parts with the integration of 2d information (Figure 30, Figure 31).

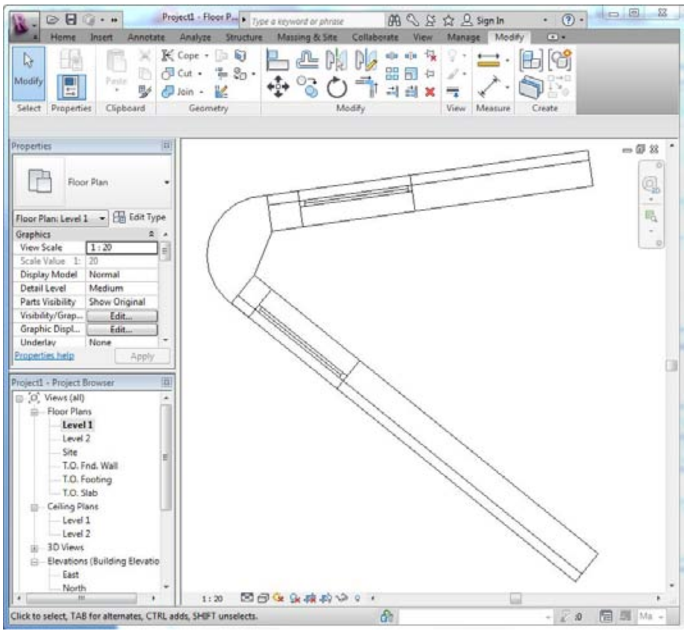

Figure 30. Special building component (by the author)

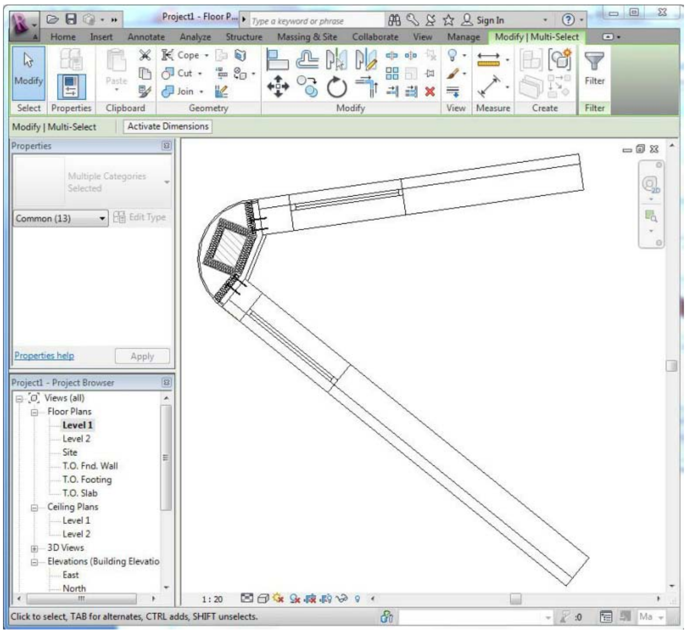

Figure 31. Assumed construction detail of the special building component (by the author)

It is also the attention to such detail that contributes to create a beautiful building. In this process, working on custom parts with BIM tools (Figure 32), is very time consuming and it requires a vast knowledge of the software and of the construction methods used in architecture and engineering. Also creating custom parts means most of the time having to break down the iterative process limiting most of the embedded automations belonging to software.

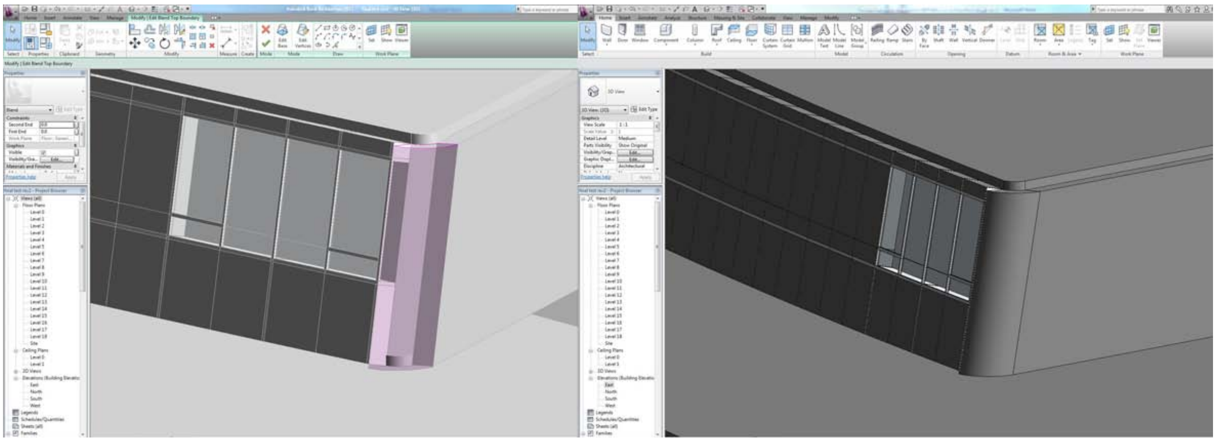

Figure 32. Round corner as custom component (by the author)

The wall at the right of the main facade, has no curvature in its geometry, but still it has a slope. Slopes are also very difficult to manage because of the whole idea of reference planes with which Revit operates. Fortunately, Revit has powerful families that can adapt to this kind of issue, by using as a reference the slope itself. However, on this type of situations is very difficult to manage measurements, therefore it is vital to create the right floor levels to use them also as references for vertical placements.

From now on we can work on the windows by creating them directly in their family editor (Figure 33). This is a quite advanced feature of Revit. A deep knowledge of the software behavior is needed to create parametric relationships within such families (Figure 34).

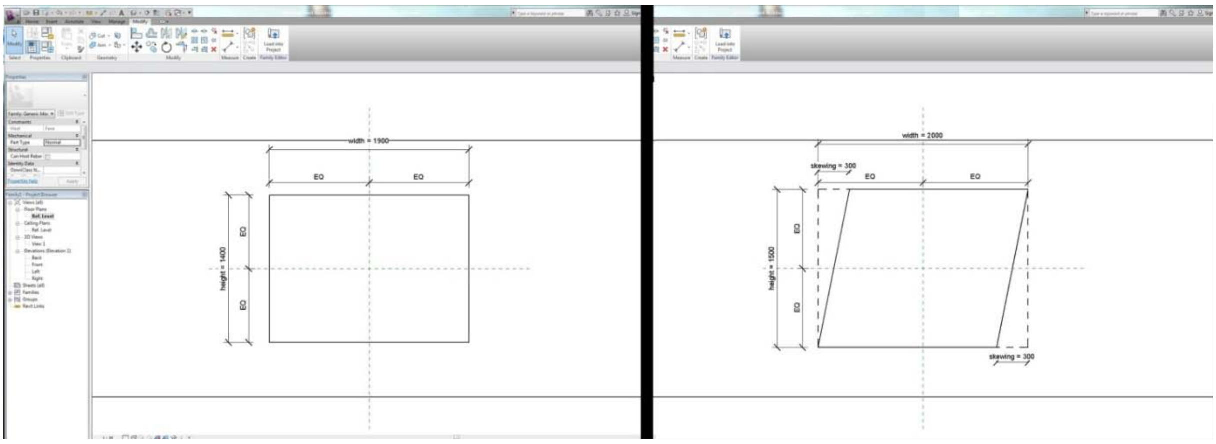

Figure 33. Window reference lines for parameters (by the author)

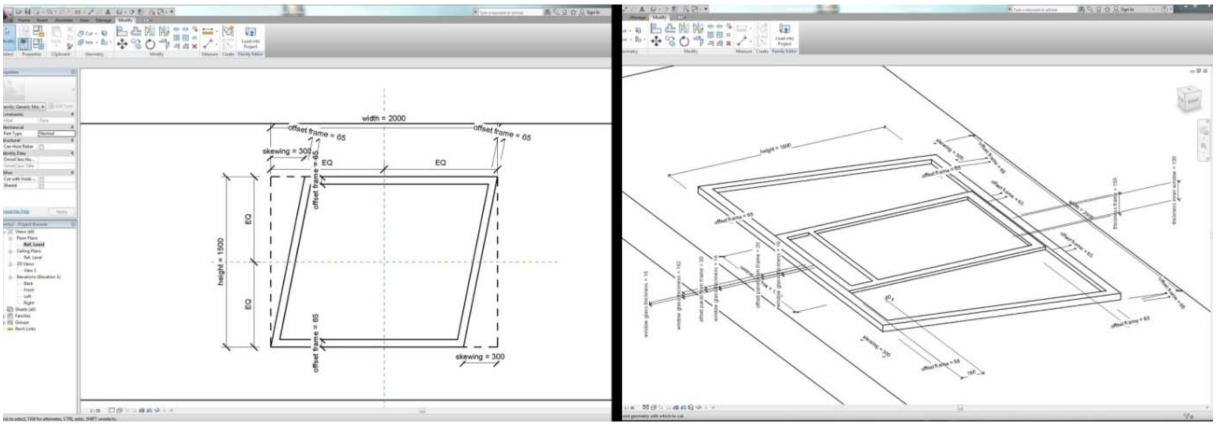

Figure 34. Window parameters assigned (by the author)

The parameterization of a window itself is not really necessary specially when working with regular shapes. It can be useful but it is not vital. In our case though, being the placement of the window itself very difficult, having parameters to tweak and change will play a fundamental role.

In fact the slanted wall will require additional reference geometry for the placement of the window and the tuning of its slope with the corner of the building.

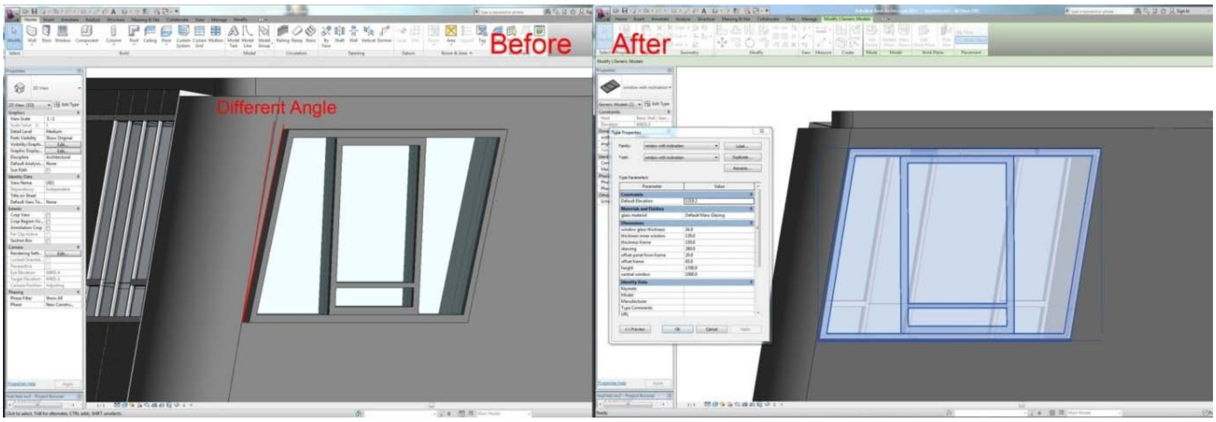

Figure 35. Parametric window working (by the author)

Once we created this window we will have at least one specific system that will work quite well with a lot of consistency specially from the parametric point of view that we were trying to achieve (Figure 35).

With this very first exercise we implemented the most solid modeling possibilities of Revit through a building example that forces the software to use very little parametricism. Eventually this structure was planned a long time before these software were available (1999). Probably this is the reason why windows are laid out in a straight grid and thin cladding panels were used to cover up building imperfections. Once these components are ready, they shall be exportable for a detailed analysis such as

5 Enclosure Design using Custom BIM Functions

5.1 Massing Design

Rhino is a very stable Geometric Design, NURBS based application that allows a huge degree of freedom when designing anything in general. Files related to the same project, in Rhino, have though very little interaction with each other, unless of course we are using blocks. Blocks are eventually not dynamic, and they do not update unless specified by the user. It is probably because of these reasons, that a lot of work has been done from external developers in order to improve the software's file management capabilities. Rhino has no scheduling features whatsoever. Being though a very complete application, it has mathematical functions embedded in it that allow the user to retrieve useful information from the geometry produced. Also with the Grasshopper plug-in (www.grasshopper.com) this information can be extracted and also organized in a functional way.

It is with Grasshopper that the user can structure the design of the massing of the building and allow a real time three dimensional feedback that Revit lacks.

To create our main mass we will generate 3 curves just like it would be done in drawing the building's plan (Figure 36).

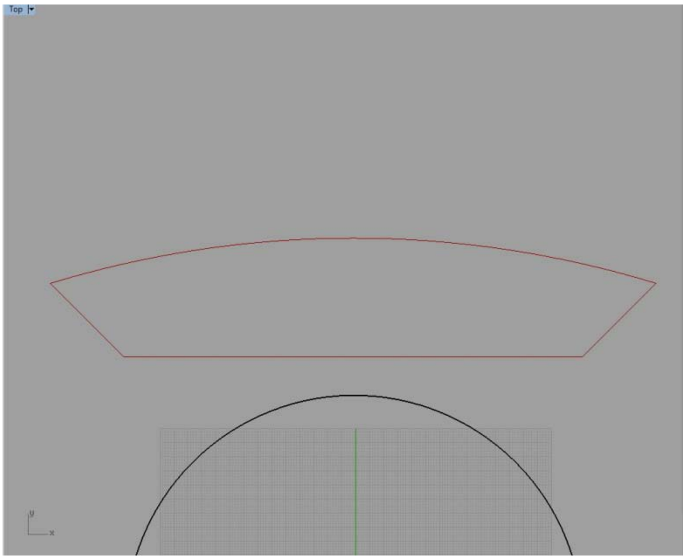

Figure 36. Plan in Rhino (by the author)

Then we can generate the 2 other curves giving to each of them XYZ coordinates (Figure 37).

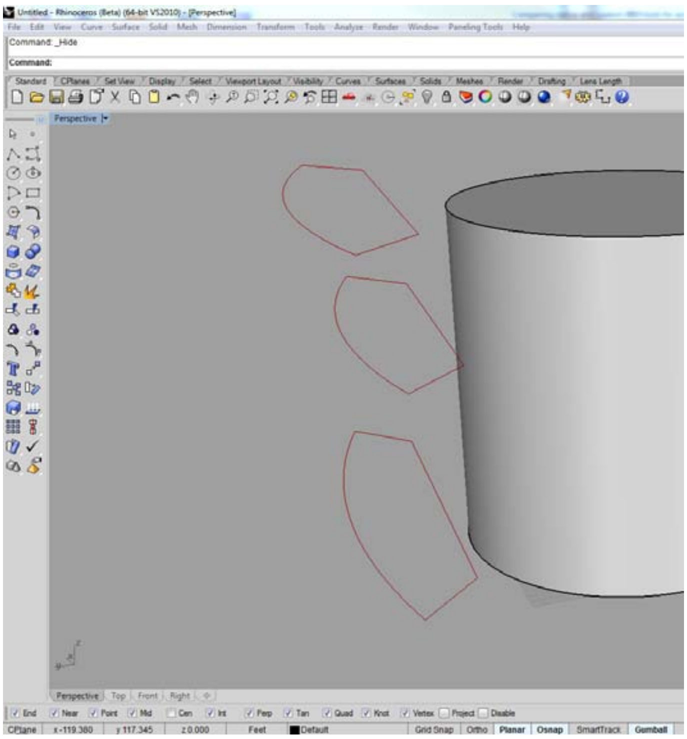

Figure 37. 3 Generating curves for massing (by the author)

These curves being movable in three dimensional space, they can generate a set of iterative designs that can be modified in real time giving a quick visual feedback to the designer (Figure 38, Figure 39, Figure 40).

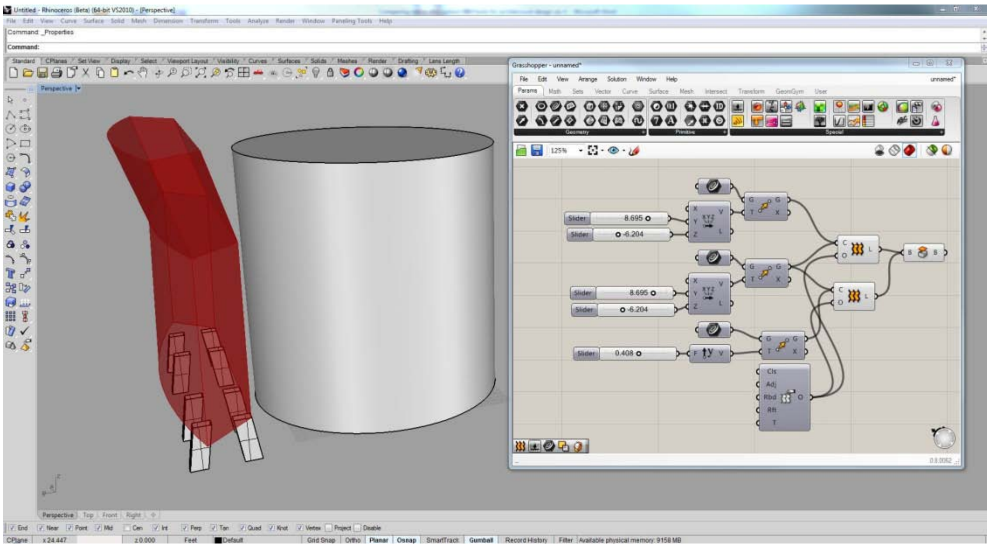

Figure 38. Massing option 1 (by the author)

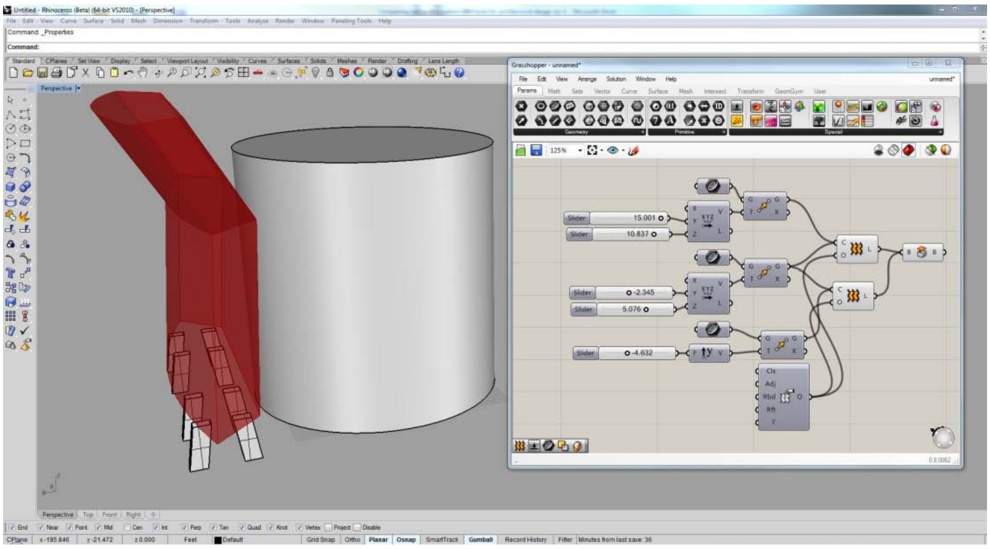

Figure 39. Massing option 2 (by the author)

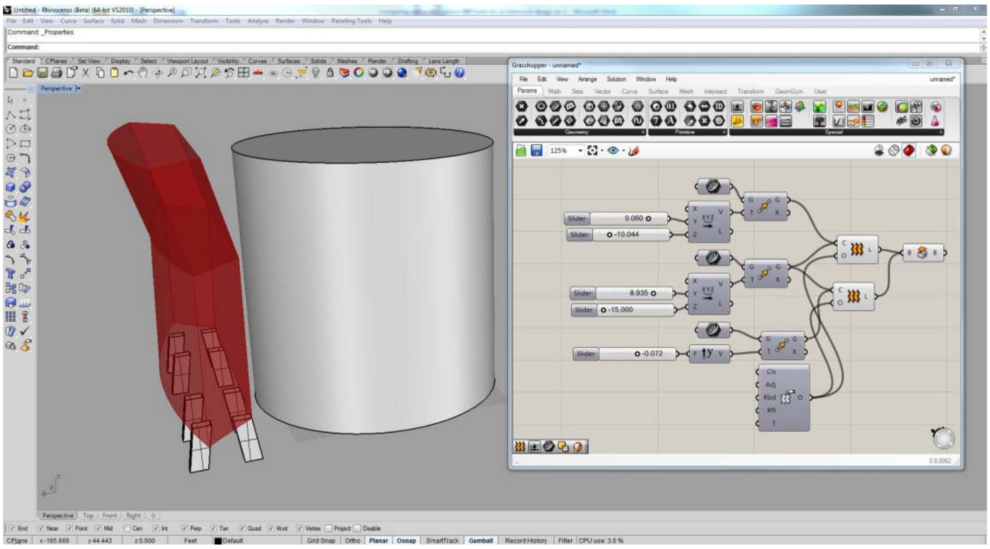

Figure 40. Massing option 3 (by the author)

One other difference with Revit is that we do not need to care about the round corners as we will be able to implement the special component described in the chapter before, independently from the massing. Also in Rhino, when the massing is ready this can be exported for a variety of analysis such as solar gain, heating demands, shading studies of the massing over the site and so on.

5.2 Schematic Enclosure Design

At this point we will have to work on the main facade so to see if there is a practical way to reconstruct the orthogonally divided facade of this building. Unfortunately, the automations of Grasshopper and Rhino will make it difficult for us to control precisely the UV coordinates of this large side of the Gasometer. We can now also concentrate on the floor management as well. We will instruct grasshopper to take the geometry generated parametrically and to slice it accordingly to the amount of floors required. We will be able also to control the offset from the ground and the thickness of the slab (Figure 41), just like Revit does.

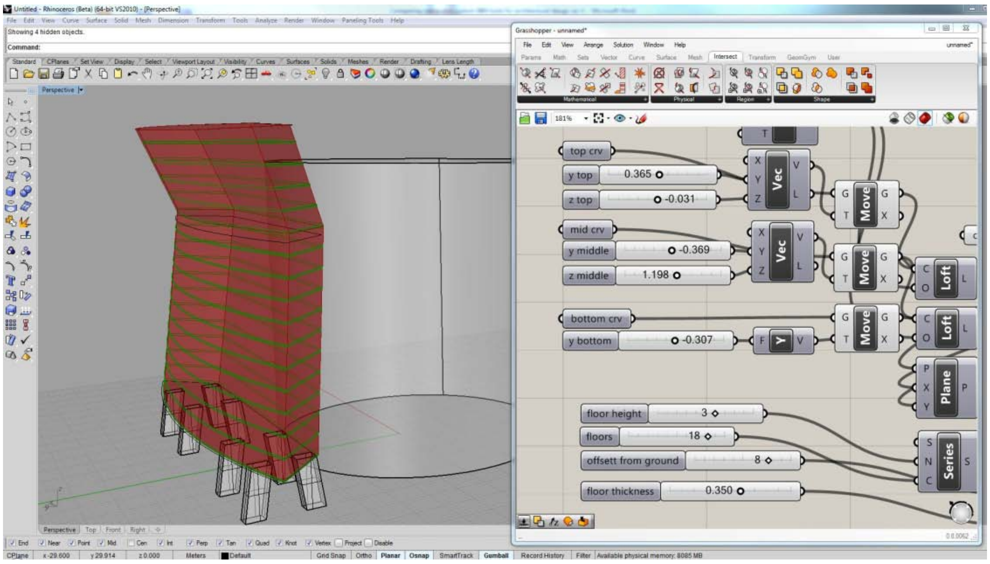

Figure 41. Division of the massing in floors (by the author)

We will create first a window component that thanks to its fixed layout won't need to be a parametric object. However because of the modeling capabilities of Rhino this instance will be modifiable also upon completion of the curtain wall giving a full iterative quality to the entire facade.

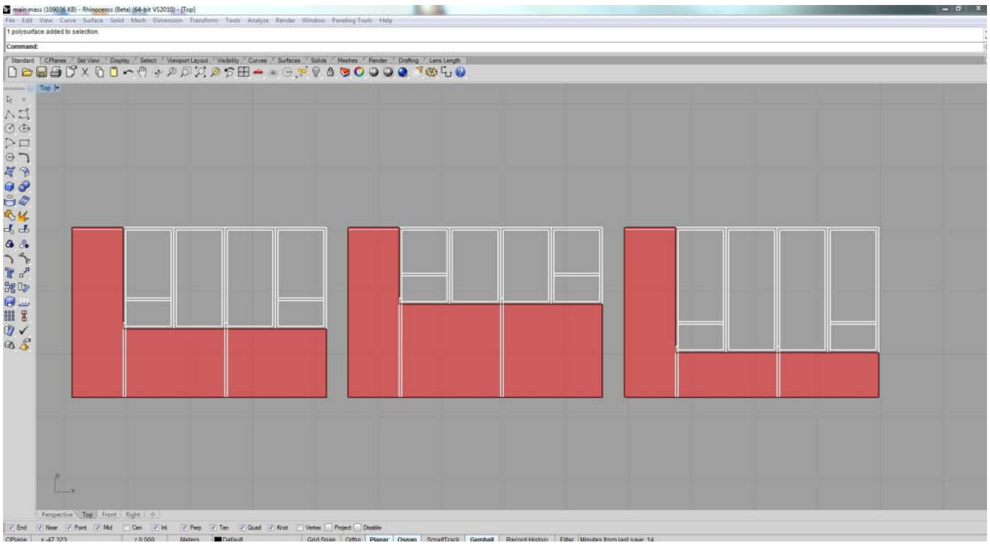

Figure 42. Window Component (by the author)

Some of the aspects of the window as we discussed already can be changed because of the geometry type which is very basic and has very little parametric constraints.
A this point we will leave the UV automation of the facade surface as default to see what happens when populating it with the window bays.

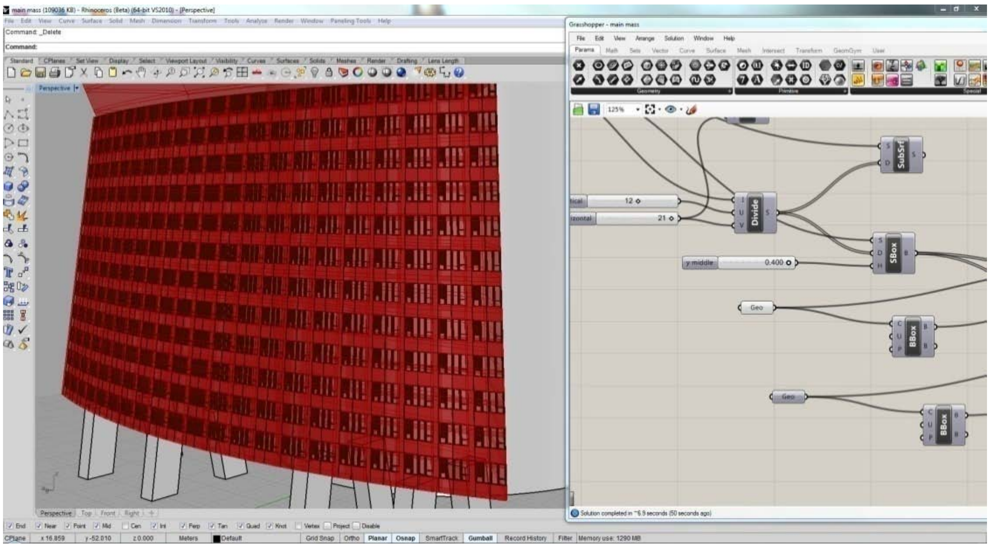

Figure 43. Populated Facade (by the author)

We can have already with this model an iterative relationship between the massing, the main facade and the window detail.

However, for the sake of our argument we will try to see if there is a better way to implement the orthogonal grid rather than the UV automated one provided as default tool from Rhino and Grasshopper.

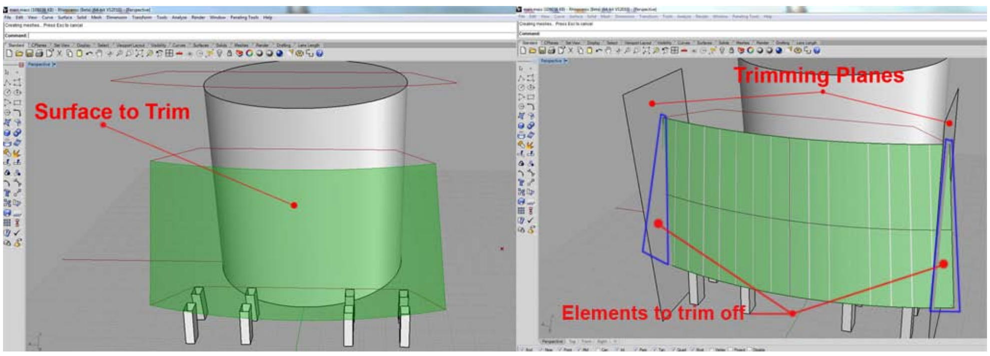

Figure 44. Base surface for Orthogonal trimming (by the author)

The task here is to generate a surface with orthogonal UV coordinates. To do so we will firstly trim a surface that will give us the base for the curves that then will be lofted so to create the final new surface with the exact coordinates (Figure 44).

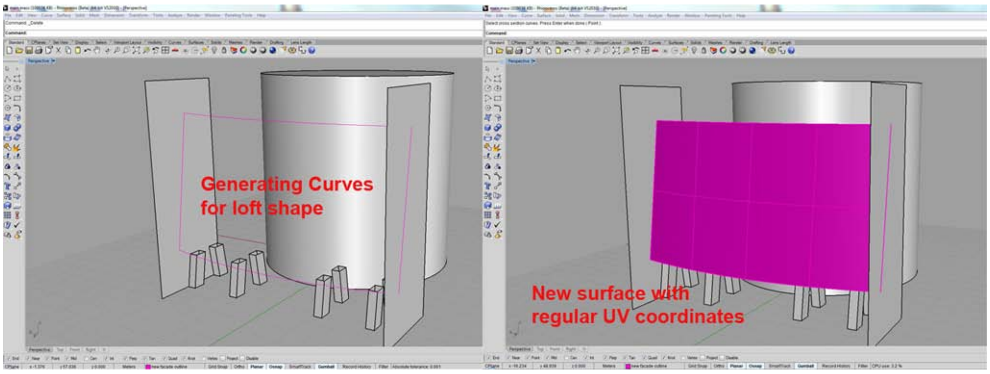

Figure 45. Regular base surface for paneling (by the author)

Once we have the regular surface (Figure 45) we can use the same bay window element created before to re-tile the facade.

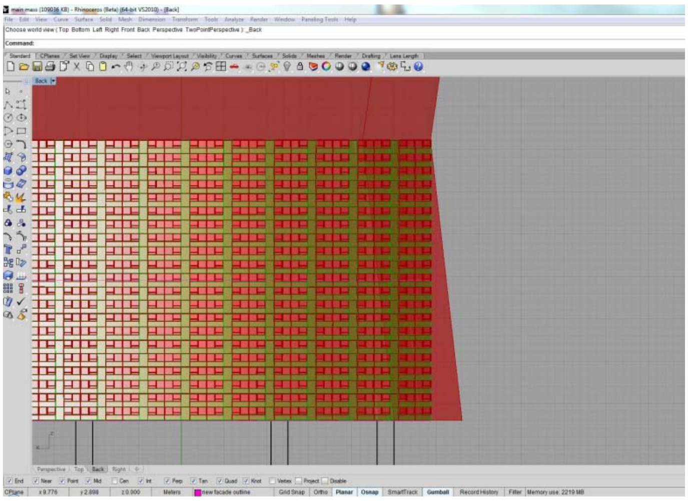

Figure 46. Orthogonal Grid on the Facade (by the author)

As expected the surface modification brought us to the desired result in a much closer way to the original building (Figure 46). For the corner we shall implement the same method, bearing in mind that the geometry that we will create, will have to be trimmed off at the end of the process breaking the iteration feature implemented so far (Figure 47).

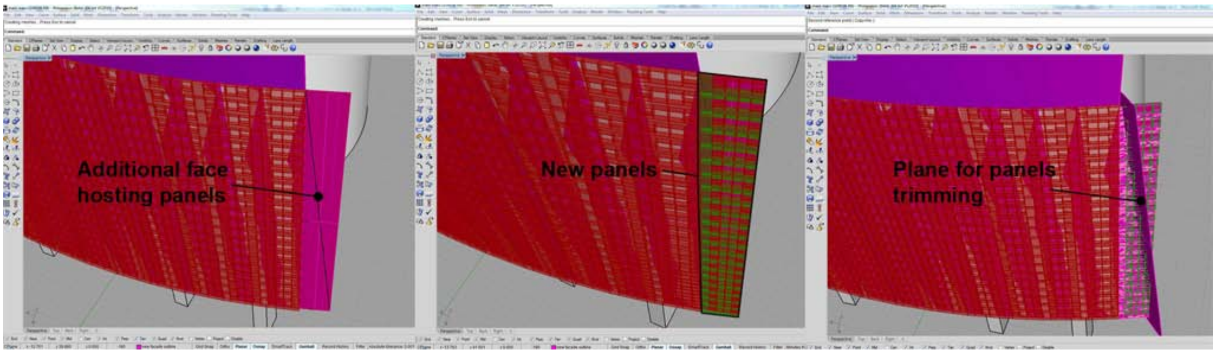

Figure 47. Trimming new Panels (by the author)

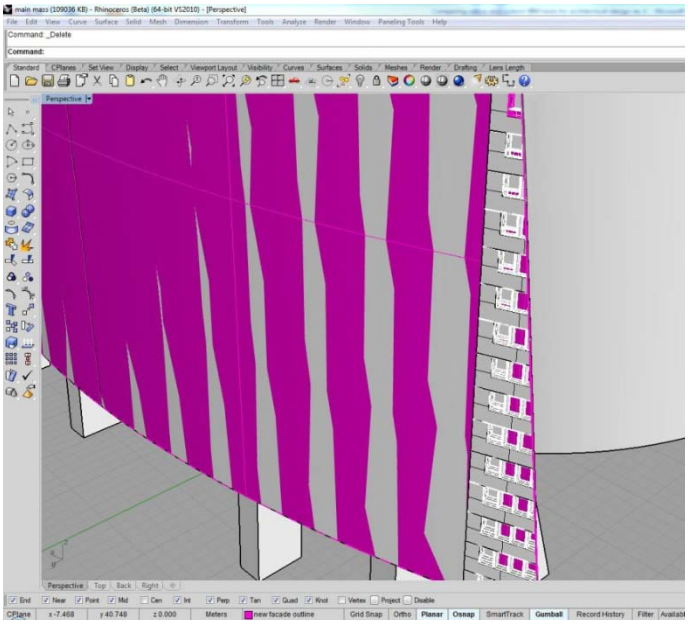

Figure 48. Trimmed Window Bays (by the author)

For convenience the geometry created for the corner (Figure 48) could be grouped so that in case changes are needed, one can replace the entire component with a new one.

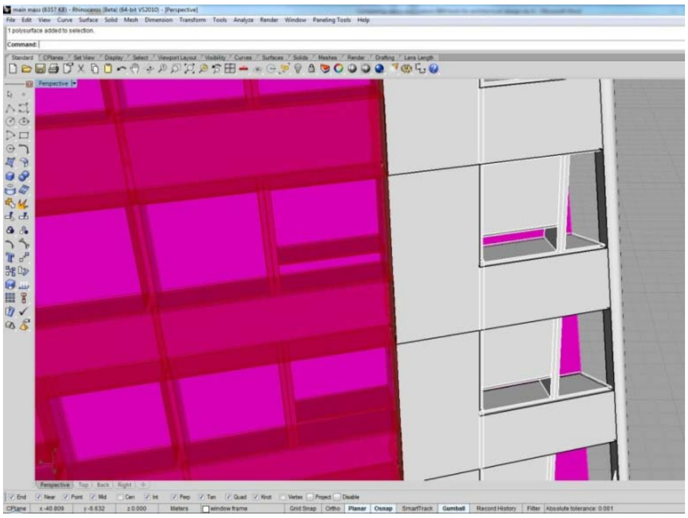

Figure 49. Corner Element (by the author)

Now as the corner geometry for the windows (Figure 49) is completed, we can concentrate on the round wall joint from the previous chapter. To do that, we can simply define the profile of the component and then detect the curve of the mass to which this should be lofted along.

The process is also in the case of Rhinoceros a destructive one, meaning that the geometry created cannot be recomputed and updated once the design of the building changes.

The wall on the other side around the building's corner will be a simple off-set geometry that might get trimmed once the details of the building have been decided. Once the layout is created this component should be exportable for structural analysis for instance. Grasshopper provides in this sense also this functionality. The bill of material can also be executed within Grasshopper.

5.3 Detailed Enclosure Design

We can now create the side window with all the connected parameters. Eventually this is not a very simple task in Grasshopper because most of the automation embedded in Revit are typical of that program only. Implementing the same parameters in Rhino will take a bit longer, but once the component is created, it can be used as a block inside any other Rhino project. The Grasshopper plug-in will also allow the gathering of information needed to create a bill of materials as well as giving us dynamic information on its size.

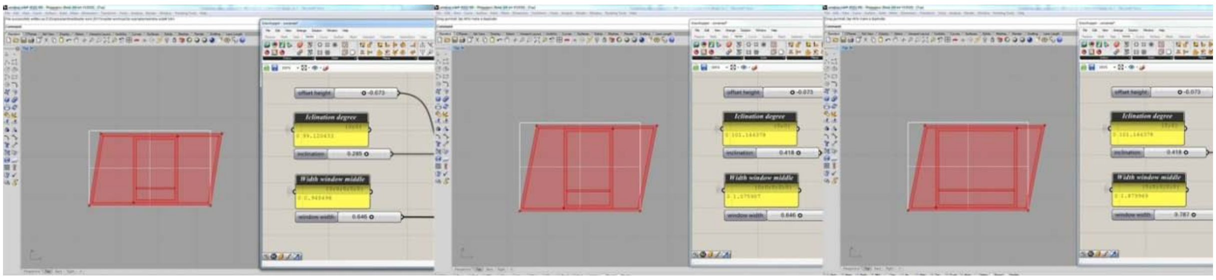

Figure 50. Parametric window in Grasshopper (by the author)

Now that the window is ready and working, one way we have to implement it, is to create a block instance. In this way, after the placement of the window inside the project, we can still change its attributes parametrically in its native file.

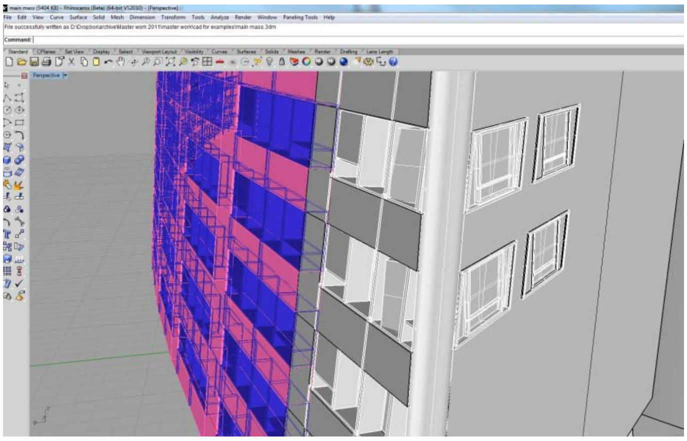

Figure 51. Windows matching the corner component (by the author)

The window is inserted (Figure 51) and placed along the facade using the slanted side and the floor levels as reference (Figure 52).

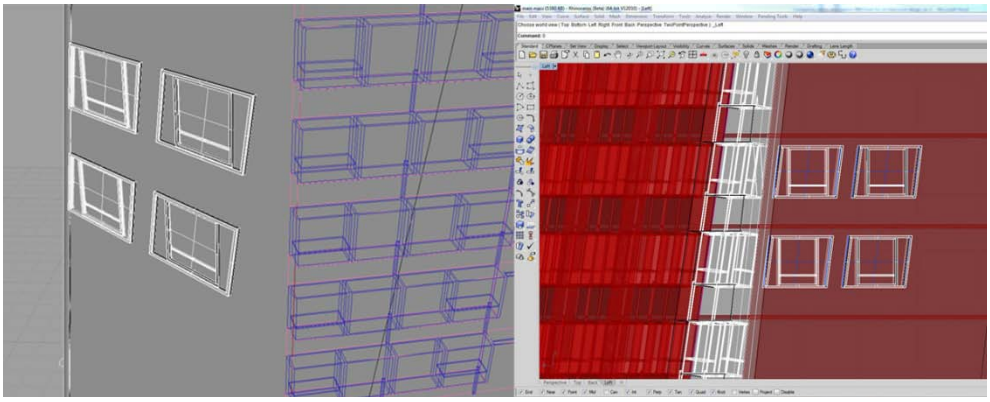

Figure 52. Windows placement on the side facade (by the author)

The windows can now trim the wall so to create the opening across the facade. This is a destructive process. Eventually the window can be changed dynamically in its own file. If changes should happen, the designer would have to be forced to re-trim the wall again to create the opening. This might not be a problem though, because eventually details like these ones are added once the building is ready to be constructed on site.

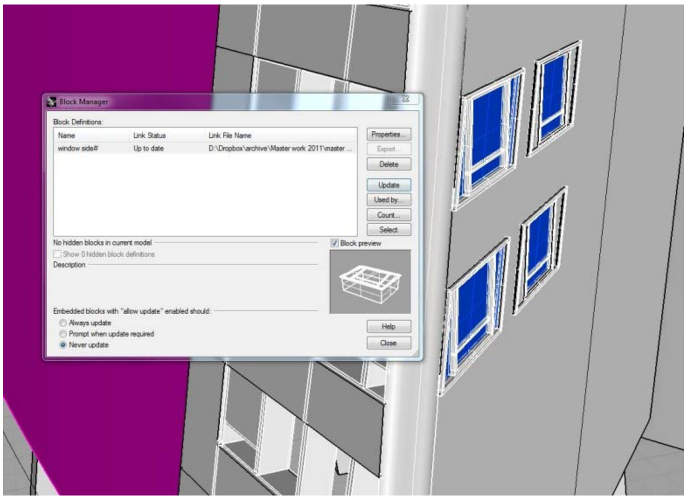

Figure 53. Window saved as a block (by the author)

Finally, just to show the procedure, the file is re-worked on its own as a block and the changes made in it, are replicated also inside the main project (Figure 53). The process is quite automatic, but it needs good supervision specially when creating the different parts in different files and folders (Figure 54).

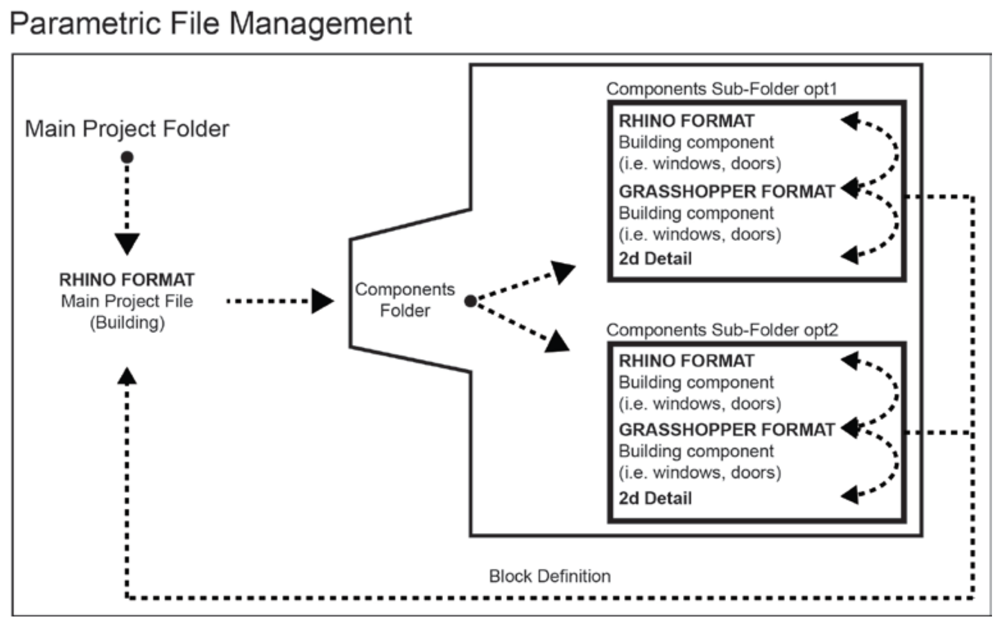

Figure 54. Parametric File Management in Rhino (by the author)

If any building detail should be included inside the project, this should be allocated also as a block inside the component folder.

6 Evaluation

In this last exercises with the two different tools Revit and Rhinoceros, we were focusing our attention on three main qualities of BIM: Iterative Design Support, Data Integration Support and Performance Analysis Support. These tasks were then translated across our modeling exercise.

The results of our test are related to the performance of a software over the other.

As hard as it can be to evaluate a tool of this kind in an objective way, the test has showed us areas where one program can perform better than the other. One interesting point is for sure the comparison of the iterative design support with which the tool can iterate between the different design stages (Figure 55).

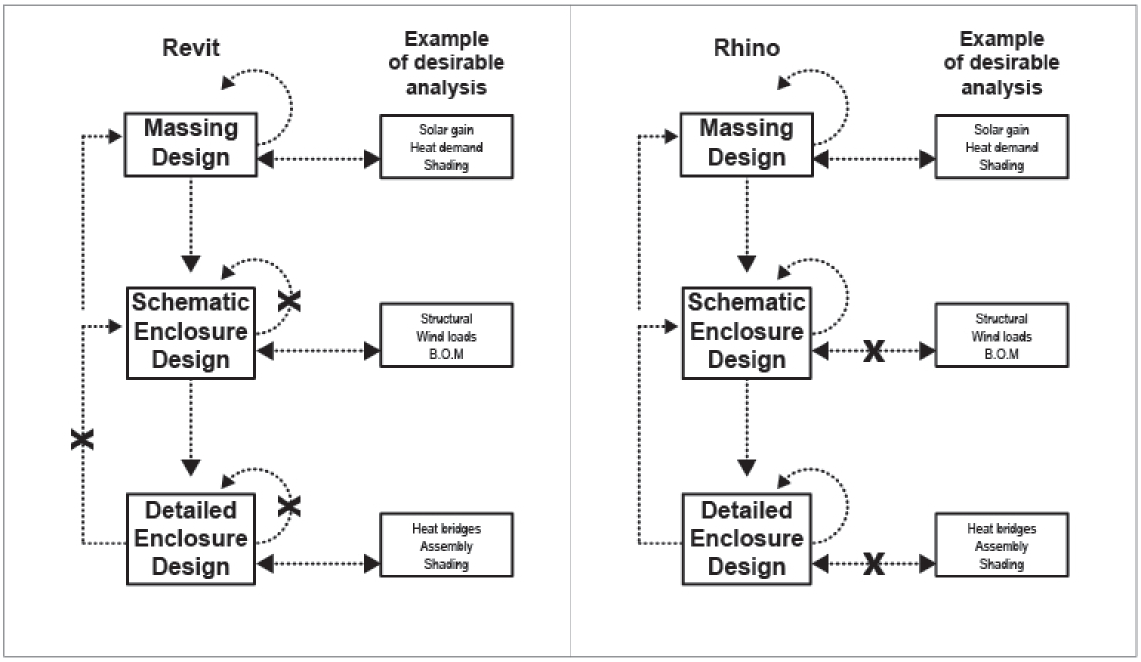

Figure 55. Iteration model comparison (by the author)

Revit showed some limitations in keeping an iterative relationship between the three different stages of the enclosure design specially when dealing with the detailing stage (Figure 55). The black cross indicates in fact when the iterative relationship between the various stages or a single stage itself is damaged or broken.

Eventually in the BIM workflow it is expected to introduce buildable data at the very end of the design process when the building is ready to be go on site (Eastman, 2011). Therefore even if the test in Revit hasn't shown exciting results it does not mean necessarily that the software performs badly. On the other side Rhinoceros due probably to the lighter data that it uses for displaying the model data, it was much more reliable, quick and consistent in providing the user with a visual feedback. The main downside of Rhino was that it was not possible to implement any building information within the model at any stage.

We evaluated our experiment basing us on the same criteria developed during the design process of the enclosure of the "Gasometer B". We took a closer look at the following:

- Massing Design
- Schematic Envelope design
- Detailed Envelope design

6.1 Massing Design

In this exercise the building automations for which Revit is known for, were limited by the actual building constraints of the structure itself i.e. the double curvature of the facade, the orthogonal grid layout of the windows and so on. Rhinoceros showed immediately more flexible modeling capabilities than Revit. Starting from the massing the process was much more fluid in Rhino. In the first instance the parametric task of keeping an iterative relationship between the massing and the building envelop was satisfied. To improve the outcome of the main facade and to make it look more like the original one, we forced Rhino to create a specific component that could have hosted the curtain wall. Doing the same operation in Revit was not possible because we could not use the same degree of precision when trimming the curves that were generating the needed surface. The outcome of this operation was therefore much more successful in Rhino than in Revit.

6.2 Schematic Enclosure Design

The process in Revit was limited by the native function of dividing the hosting surface in UV coordinates. Eventually if the same design were to be carried out nowadays instead, with more contemporary 3d tools, we could speculate that the building might have looked different for many aspects. The designer might have taken full advantage of the modeling capabilities of the tool in trial, using for instance the available UV automation when dividing the main facade for creating the curtain wall . This task was also about reproducing the existing structure in the way it currently presents itself. In doing so, Revit has shown us quite few flaws in its modeling flexibility. Rhino also had this type of limitation due to the embedded UV surface calculation. However with Rhino we could control more precisely the loft surface on which the curtain wall was going to be laid off. The result was an orthogonal grid that was more precise than the Revit one. The process was parametric for both Rhino and Revit, with the advantage in Rhino to have a higher speed in terms of creating geometry. Sometimes in fact Revit would fail partially if the massing changes where too large. The host for the orthogonal grid was not parametric meaning that every time the mass changes, a new host has to be custom built.

6.3 Detailed Enclosure Design

Rhino carries by default no 2d detail in its models. This does not mean that the model cannot give building information to the user. We can extract in fact 2d line drawings from the model in a manual way which in Rhino is very powerful. Revit has the ability of carrying detail, but this process is difficult and many times, when geometric errors of the subcomponent happen (in our case the curtain panel) the whole curtain wall will return an error message. In a very complex structure like the one we were dealing with, it is suggested to insert detail information only when the design is final. In Rhino the paneling operation could be done in real time, so eventually if there was an error of the sub component, the user would have been able to identify it straight away. Rhino over Revit presented a much more complex procedural approach in creating the curtain wall, but the result was -in its complexity- better than the one achieved in Revit.

6.4 Additional assessments

We shall now shortly talk about performances related to the time used to create the model, hardware requirement and last but not least the quality of the 3d model itself.

The two software performed quite well until the main curtain wall data was introduced. Afterward, Revit seemed much slower than Rhino with computational times of the main facade of up to twelve minutes. Rhino and Grasshopper were giving a quick feedback with computational times never longer than a minute. This connects us to the hardware issue. The test was ran on a rendering machine with a Xeon quad-core processor, 12GB of Ram memory and a 2gb Nvidia Quadro Graphic card. When the same test was ran on a laptop with a dual core processor, 8GB of Ram and 512MB of graphic memory, the programs had more issues. Revit did not manage to compute and crashed after twenty minutes of calculating the curtain panel, Rhino instead had troubles with the graphics when panning the model around but it did not crash. Finally, the quality of the 3d model was much better in Rhino. In fact Revit had a lot of gaps and inconsistencies due to the slanted surfaces that did not allow too much snapping. All the transformations related to the facade and to the curtain panel of the Revit model were much more imprecise compared to the Rhino ones. Rhino has proved with its flexibility to work better and more precisely on freeform surfaces than Revit. Also the test has shown that Rhino, even though it is not designed as a specific architecture software, it is able to carry as much 3d information just as good -if not better- than Revit.

7 Conclusions

From our first introduction and after the thorough comparison made on these tools, we can all convey that the effort done in the development of BIM Design Tools such as Revit has brought designers mainly a better file management software, with features that are not possible otherwise to the likes of Rhinoceros. These features though, they prove itself not to be too useful when dealing with complex geometry. In cases like this the building information that is usually automated by Revit, has to be introduced by the designer externally from the software environment and manually (so without using any of the native automations), meaning consequentially a partial loss of utility of the file management system of the software.

On the other side instead, tools like Rhino allow a very high degree of manipulation of the software itself, making it possible to adapt with great flexibility to more tasks than Revit. Rhinoceros 3d over Revit gives a greater level of freedom when creating any type of form. The parametric modeling characteristic of Revit can be implemented to a certain degree also in Grasshopper. The solid modeling possibilities of Rhino though are much more advanced than the Revit ones.

Said so, even if BIM Design Tools do bring a lot of automations when designing a building, the flexibility and advanced modeling capabilities of software such as Rhinoceros, will be also always needed, in order to higher the standards and push the boundaries of the work produced in a firm.

As far as we can speculate on the future of BIM Design Tools, real improvements need to be made on the embedded modeling capabilities of such programs, excluding of course open scripting platform as this would make the choice of migrating towards these more expensive BIM solutions, even more difficult. By improving the modeling capabilities of BIM Design Tools, we would reduce the need for using additional external software, reducing also the room for human errors, implementing a better and faster iteration between design options and more over we would speed up the entire design process reducing also the need to leave often our main design platform. If this optimization in BIM might not happen, there could be a shifting of this software family towards a specific building management category of tools that will probably cross the boundaries between architectural design and project management. BIM Design Tools will then be used only as a method to integrate check and manage designs coming from a plurality of other more powerful modeling packages.

On the other hand, even if the flexibility of digital 3d modeling offered by generic Geometric Systems like Rhinoceros 3D is still not available to tools such as Revit, we cannot rely on having to split a project in many different files as shown in our previous example when we tested Rhino and Grasshopper. Improvements regarding these types of software should be done in their file management system. In the case of Rhinoceros 3D there are various plug-ins that have been developed specifically to simulate some of the most common native BIM functionalities, still main modifications should be executed at the core of the program itself so to allow more files to be managed within one single instance and in an automated way, enabling also the possibility to check these changes in real time and to allow more people to work on the same project and at the same time.

At last, architects and designers should always remember that due to the sets of rules (in the case of architecture we have aesthetical rules as well as structural, functional and budgetary) that define the project itself, each field of design expects the designer himself to have good management skills (Eastman, 1991). Even if BIM tools are created to simplify the handling of design information and facilitate design automations, the images of the project manager just like the one of an architect or a designer or a structural engineer are just irreplaceable and vital to the life of a project. Actually it is this professional collaboration between these people that -together with experience, creativity and eventually great knowledge of these software- can bring exceptional and successful buildings to life.

Glossary

BIM (Building Information Modeling). This nomenclature is used to indicate both processes and technologies related to a building's design and facilitate from information exchangeable via and between computers (Eastman, 2011).

Building Model. This is usually the 3d model of a building that works also as the database of all the information related to the construction details and eventually also performance of a building (Eastman, 2011).

Building objects. We define building objects all the elements that together construct a building. Building objects are a subset of the building model. Element or component is used as a synonym for object, in Revit we also use the word Family (Eastman, 2011).

Parametric Objects. Objects which shape can be edited via modification of their parameters (Eastman, 2011)

CAD (Computer Aided Design). A type of software that allows the user to draw technical drawings in 2d.

CAAD (Computer Aided Architectural Design). A specific CAD tool enabled to produce also 3d models specifically for the use in architecture.

GM (Geometric Modeling). this is a branch of computer aided design that focuses on the production of 3d models as well as 2d. The implementation of such systems can be wider than the implementation of CAAD tools. Rhinoceros can be defined as a Geometric Modeling System.

SM (Solid Modeling). Like geometric modeling but focusing exclusively on 3d objects that enclose a volume (Eastman, 2011).

Workflow. The combination of consequential operations among different collaborators, used to accomplish a determined result.

List of figures

References

BIM Modeling Blogspot. (2010). Retrieved December 26, 2011, from http://bim-modeling.blogspot.com/2010/12/history-of-bim-and-success-story-till.html

BIMandintegrateddesign.com. (2011). BIM and integrated design. Retrieved October 8, 2011, from http://bimandintegrateddesign.com/2011/10/08/the-case-for-case/

Doughty, N. (2011). *Asite-cBIM-standard-deck* (pp. 2-11). London: ASITE. Retrieved from www.asite.com

Eastman, C. (1991). Use of Data Modeling in the Conceptual structuring of Design Problems. Zurich.

Eastman, C. (2004). New Methods of Architecture and Building New Methods of Architecture and Building. In ACADIA (Ed.), *Architecture* (pp. 20-27). Toronto CA.

Eastman, C. Teicholz, P. Sacks, R. Liston, K. (2011). *BIM Handbook* (Second Edi., pp. 4-9, 31-51, 77-94, 585-589). Hoboken, New Jersey: John Wiley & Sons, INC.

Ibrahim, M. M. (2007). Teaching BIM, what is missing? *ASCAAD*, (3rd international ASCAAD), 652-659.

Krygiel E. , Read P., V. J. (2010). *Mastering Autodesk Revit Architecture 2011*. (P. Gaughan, Ed.)*Imprint* (pp. 17-62, 139-168, 255-344). Indianapolis: Wiley Inc.

Kymmel, W. (2008). *Building Information Modeling, planning and managing construction projects with 4d CAD and simulation*. (I. the McGraw-Hill Companies, Ed.)*Construction* (pp. 1-23). New York: the McGraw-Hill Companies, Inc.

Laiserin.com. (2003). laiserin. www.laiserin.com. Retrieved November 22, 2011, from http://www.laiserin.com/features/issue19/feature01.php

Sutherland I. E. (2003, June). *Sketchpad: A man-machine graphical communication System*. MIT. Retrieved from www.cl.cam.ac.uk