Semi-automatic data transfer from Revit to BSim

- A rapid and consistent modeling for indoor simulations -

Master Thesis

Bogdan Alexandru Murariu

Aalborg University
Building Energy Design

This report it is written and edited using ShareLatex. The software used in the project are Revit, Dynamo, Excel and BSim. Figures and diagrams are created using online diagram creator LucidChart.

Civil Engineering
Aalborg University
Thomas Manns Vej 23
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Semi-automatic data transfer from Revit to BSim

Theme:
Scientific Theme

Project Period:
Fall Semester 2018

Project Group:

Participant(s):
Bogdan Alexandru Murariu

Supervisor(s):
Rasmus Lund Jensen
Kjeld Svidt
Torben Østergård

Copies: 1

Page Numbers: 51

Date of Completion:
January 10, 2019

Abstract:

The work on this project is based on the semi-automatic approach developed by MOE A/S and AAU to combine Revit with Danish Be18 software. The project is focused on the challenges of performing building energy simulations in the design phase while using BSim program. The report analyses the current practice in the industry with insights from company MOE and personal experience. With focus only on the geometry inputs, the BSim structure is described, and with use of Dynamo and Excel the semi-automation of the modelling is attempted. Dynamo scripts are designed to fulfill the task of extracting the required data from Revit and export to Excel spreadsheet. In Excel various methods are tested to export the information in BSim. The final goal is to successfully select spaces in Revit and export the geometry information via Dynamo script to Excel from where it can be exported to BSim, and reconstruct the model.

Contents

List of Figures

Preface

This Master Thesis has been written from September 2018 to January 2019 at the faculty of Engineering and Science at Aalborg University during the 4th semester of the Building Energy Design program.

Working in a BIM environment has been an interest of mine and I always enjoyed working with Computer-Aided Design (CAD) tools. So choosing a topic that offers the opportunity to learn a new program like Dynamo was seen as a chance to further develop my skills and get an introduction in to the programming environment.

Therefore, with guidance from the supervisors Rasmus Lund Jensen, Kjeld Svidt, and Torben Østergaard, it is decided to investigate the possibility of data transfer from Revit to BSim and improving the building energy simulation process.

A special gratitude goes to my supervisors Rasmus Lund Jensen, Kjeld Svidt, and Torben Østergaard for their guidance, helpful ideas and their overall inputs through out the semester. Moreover I would like to thank Kim Trangbæk Jønsson for sharing his raw data and advises. I would also like to thank my colleagues from the 2nd semester of Building Informatic for sharing their knowledge , and John Ellingsgaard for commenting on the work.

Aalborg University, January 10, 2019

Bogdan Alexandru Murariu
<bmurar17@student.aau.dk>

Chapter 1

Introduction

The focus on more energy efficient construction raises,and so does the need on more accurate energy analysis of the building in the early stages of the project [14].

The standard in modern building design is to use a model-based approach for designing, refining and optimizing building energy systems[8].

Traditionally the preparation for building performance simulation (BPS) starts only after architectural and HVAC design have progressed sufficiently to provide enough information to represent the building.

This means that by the time a simulation and analysis about the energy performance is performed, some fundamental design decisions have already been made. Decisions that might turn out to be critical for the energy performance of the future building[1].

The accuracy of simulation depends significantly on the correct representation of the building geometry. The geometry is commonly being generated by a Building Information Model (BIM) authoring tool according to an architectural perspective, in this project Revit is used. That geometry must be altered for energy performance simulation tasks and used in simulation software, and in this project BSim is being used.[2]. Further in this report any mentioning of simulation software will refer strictly to BSim.

An example of communication interface between sketching, BIM authoring tools and domain specific analysis tool can be seen in figure 1.1.

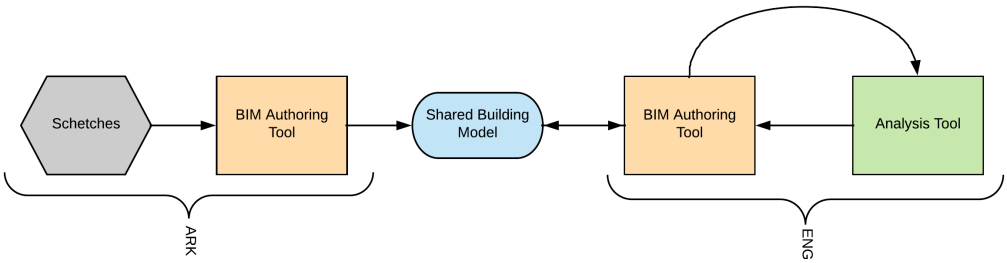


Figure 1.1: Communication process

Geometrical data extracted from BIM must be transformed and combined with material properties to be entered as inputs to energy simulation engines, a process which is time consuming and error prone [11].

It is this projects objective to use BIM authoring tool Dynamo and develop a methodology to extract the building geometry from Revit, and to be used for building simulations in BSim, figure 1.2.

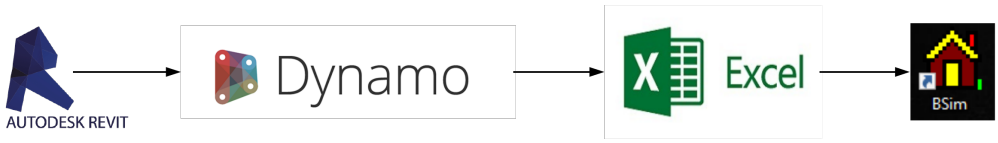


Figure 1.2: Process overview

1.1 Current Practice

Building energy simulation is currently being practiced as a combination of science and skill. The basis of science can be seen in the simulation algorithms, and in the level of building information detail required as input to these algorithms. The skill comes in to play in the current process of collecting building information from a variety of sources and manually transforming this information into specific input required by BSim [17].

This skill translates in to methods and rule-of-thumb developed over time by the specialist. But while based on professional expertise, the results can vary from one modeler to the next, even given the same initial building design information[1].

Under current common practice, a building is initially designed from an architectural perspective, producing a collection of building information defined from that perspective. Figure 1.3 represents the work-flow of an energy simulation specialist that must manually transform the architectural building information and add missing required information to create the BSim model required for energy simulation [16]. An example of Excel spreadsheet with building information added manually can be seen in appendix C figure C.4.

Figure 1.3: Manual process of creating a BSim Model

1.2 Problem Formulation

The importance of performing energy simulations in the early stages of design can not be neglected, but they require a significant amount of man hours. Due to manual extraction of data from the architect model and having to build the geometry and assign materials manually, engineers have to run simulations only on what they consider to be the most critical spaces in the building.

It is this projects goal to develop a method that will allow the semi-automatic export of data from Revit and use this information to build a model in BSim.This should improve the work-flow of the engineer giving the option of selecting mul-

tiple spaces and perform more precise simulations.

Problem formulation:

How can the correct definition of a model space boundaries improve the quality of data exported from Revit and allow the building energy simulation on a stochastic method?

Chapter 2

BSim

BSim provides user-friendly detailed hygrothermal simulation of buildings and constructions. This program is typically used to perform analyses on room level, but can be used for entire buildings if needed.

The high level of detail in the calculations places correspondingly greater demands on the degree of detail of the inputs to the model.

BSim has been used extensively over the past 20 years, previously under the name tsbi3.

Today BSim is the most commonly used tool in Denmark, and with increasing interest abroad, for energy design of buildings and moisture analysis[6].

2.1 BSim Structure

The program is composed of several modules:

- SimView - Graphic Editor
- tsbi5 - Building Simulation
- SimLight - Daylight
- XSun - Direct Sunlight and Shadowing
- SimPV - Photovoltaic power
- NatVent - Natural Ventilation
- SimDxf - Import from CAD

Because this project is focused on extracting the geometry information from Revit, only the SimView module will be described.

The modules in BSim are built around SimView which is the central program and the user interface.

In SimView a building model is displayed both in the form of hierarchical tree summary on the left of the screen and in the form of a graphical view on the right of the screen. The graphical view is divided up into a floor plan, two elevations and a spatial view, figure 2.1. North is displayed in the bottom right corner of the floor plan to show the rotation of the current building.[SBi- Statens Byggeforsknigsinstitut, 2016]

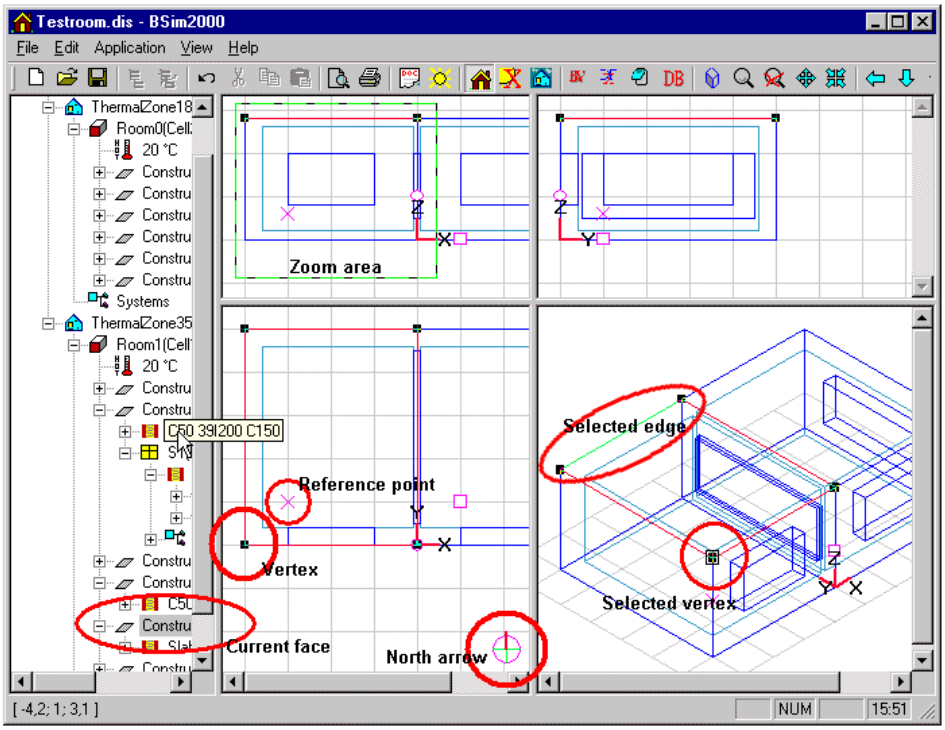


Figure 2.1: SimView user interface

A system of XYZ axis defines the model space coordinates. In this system X-axis is positive towards east, the Y-axis is positive towards north and the Z-axis is positive upwards.

When drawing a model in BSim it is important to remember that constructions facing Outdoor or Ground are drawn from the system line and inward while constructions between two rooms are drawn symmetric around the system line.

The type of file format that BSim accepts as input is XML which stands for eXtensible Markup Language. This format is easy to understand both by people and computers. It is very widely used in data exchange between various computer

software due to its excellent capability to contain information. An example of XML code from BSim can be seen in appendix A figure A.1.

Although as a limitation the format is only able to describe hierarchical relationships, appendix A figures A.2 and A.3.

2.2 Requirements for model geometry

A Revit model element contains a multitude of parameters, but not all of them are necessary to build a BSim model.To create an output file from BSim it is required a specific information for geometry of the building, and the type of construction. General information about the rooms such as room name and/or number are also necessary for easier identification. This geometry requirements refers to the building envelope.

The geometry of a BSim model is made up of 9 elements:

- Vector3D - is a point defined by xyz coordinates and it defines the location of the vertices.
- Vertex- it contains the Vector3D and defines all exported geometry.
- Edge - an element defined by two vertices.
- Face - an element defined by 3 or more edges. This elements include walls and storey partitions. The specific location of the object in relation to the building element depends on the type and location of the element.
- Windoor - an object defined by four edges and it includes windows and doors.
- Face-Side - is one of the two sides of the face. It can face outside or one of the Cell(inside).
- Cell - it is defined by several face-sides. Typically 4 walls, ceiling and floor.
- Room - it is represented by a Cell.
- Construction - it describes an element represented by a Face. It has an ID which is linked to a specific construction type from BSim database.

2.2.1 Geometry hierarchy

The BSim hierarchy system is defined as a parent-child relationship. The terms parent, child, and siblings are used to describe the relationships between elements in an XML document [7].

All of the elements that make up the BSim model are assigned a unique reference number called RID.

This attribute 'rid' has a central role in building BSim XML input as this is a unique identifier assigned to each parent elements so that BSim can distinguish the elements from each other and at the same time create the right relationships between elements, figure 2.2.

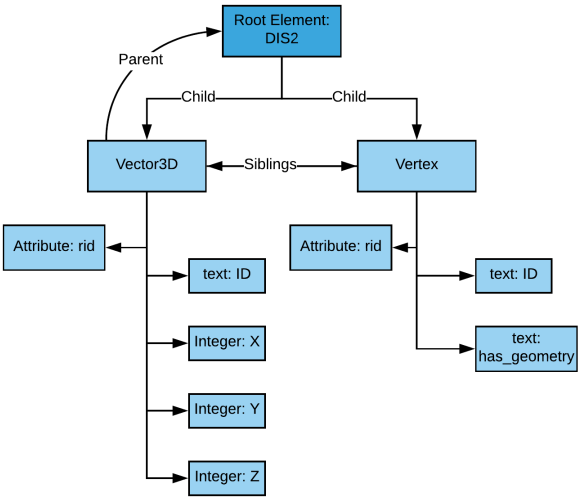


Figure 2.2: BSim parent-child relationship

In appendix A figure A.4 the diagram represents the relationship and how the elements depend on each other.

In other words, an edge is defined by two vertices and a face cannot exist without a minimum of 3 edges, typically 4, and a group of minimum 6 faces will define a cell or space, figure 2.3.

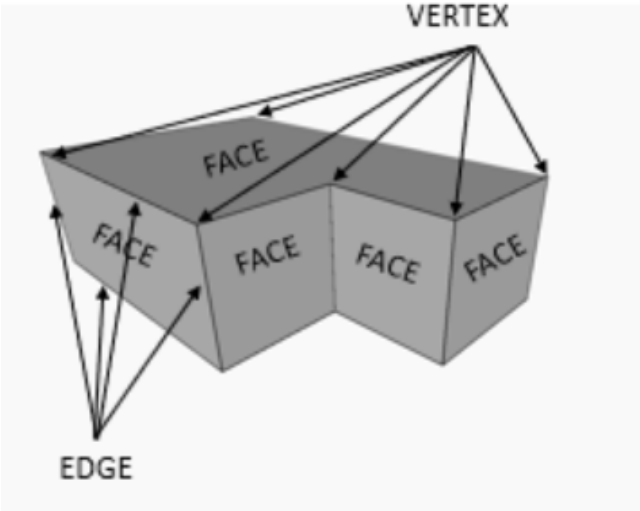


Figure 2.3: Visual representation of the main elements that define a Cell (space) in BSim

Chapter 3

Methodology

The work that has been done on this project is based on the Action Research Methodology.

Action research can be defined as an approach in which the action researcher and a client collaborate in the diagnosis of the problem and in the development of a solution based on the diagnosis[3]. In other words, one of the main characteristic traits of action research relates to collaboration between researcher and member of organization in order to solve organizational problems.

In this specific case, the researcher is represented by the student that is the author of this report, and the member of the organization will be the company representative Torben Østergaard, since he is representing the industry.

When using action research method, the following features need to be taken in to account:[9]

- It is applied in order to improve specific practices. Action research is based on action, evaluation and critical analysis of practices based on collected data in order to introduce improvements in relevant practices.
- This type of research is facilitated by participation and collaboration of a number of individuals with common purpose.
- Such research focuses on specific situations and their context.

Figure 3.1 represents the process of action study following self-reflective cycles:[9]

- Planning in order to initiate change
- Implementing the change (acting) and observing the process of implementation and consequences
- Reflecting on process of change and re-planning
- Acting and observing
- Reflecting

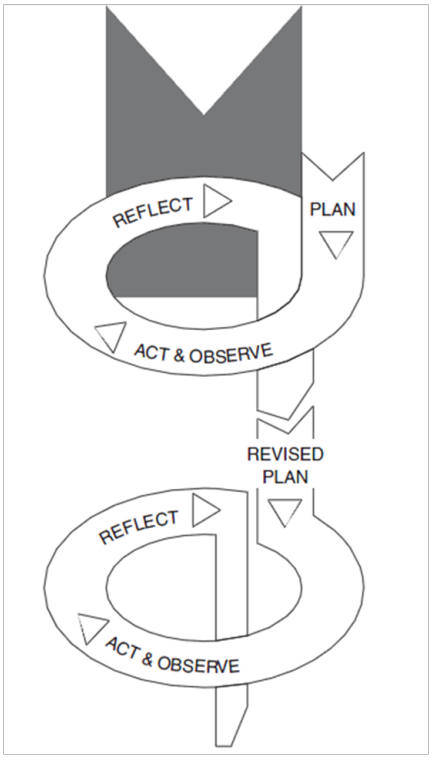


Figure 3.1: Kemmis and McTaggart Action Research Spiral (2000)

The steps that this project is following are presented in figure 3.2.

Figure 3.2: Action Research Cycle

In order to qualify as an action research method it is important that after observing the situation and planning the intervention a prediction on the result should be made.

This steps are further discussed in this chapter.

3.1 Observing the Situation

In this initial step it is identified what is needed in the industry when it comes to performing energy simulations.

As it was presented in Chapter 1, the current practice of performing simulations is very time consuming due to manual extraction of data and the manual process of rebuilding the geometry model in the simulation software. This results in having fewer simulations performed on a limited number of spaces, thus the analysis can not be very precise.

Since it is desired that the energy simulation to be available in the early stage of design, when decisions about materials and building layout are being made, it is necessary that the simulation and analysis time to be improved.

3.2 Planning the Intervention

One way of increasing the efficiency of building simulations is to develop a method that will automate building the model geometry for simulation software.

In collaboration with company MOE A/S, represented by Torben Østergaard, a plan is developed to semi-automate the extraction of geometry data from Revit and import it in BSim simulation software.

Although having a fully automated method of building models in BSim might sound like a better solution, it is desired that the specialist performing this tasks should have control over the input that goes in the simulation software. Achieving the goal of full system autonomy is quite difficult, and most systems will exist at some level of semi-autonomy for quite some time[10].

The more automation is added to a system, and the more reliable and robust that automation is, the less likely that human operators overseeing the automation will be aware of critical information and able to take over manual control when needed[10]. Examples of different levels of automation can be seen in figure 3.3. Where the highlighted section is representative for this project.

Level of Automation	Description	Role			
		Monitoring & Information Presentation	Generation of Options	Decision Making/ Selection of Course of Action	Implementation of Actions
Manual Control	Human performs all aspects of tasks	Gathers Info	All	All	All
Information Cueing	Computer aids in highlighting key information on screen or decluttering irrelevant information	Gathers Info & Highlights	All	All	All
SA Support	System gathers key information and integrates for level 2 & 3 SA	Gathers Info & Integrates	All	All	All
Action Support / Tele-operation	Computer aids in doing each action as instructed	Gathers Info	All	All	Single tasks
Batch Processing	Computer completely carries out singular or sets of tasks commanded by human	Gathers Info	All	All	Sets of tasks
Shared Control	Computer and human generate decision options, human decides and carries out with support	Gathers Info	Options (Both)	Decides	Single tasks
Decision Support	Computer generates recommended options, human decides (or input own choice) and system carries out	Gathers Info	Options (Both)	Decides	All
Blended Decision Making (Management by Consent	Computer generates recommended options and selects best, human must consent (or override) and system carries out	Gathers Info	Options (Both)	Decides Consent	All
Rigid System	Computer generates recommended options which human may select from (cannot override) and system carries out	Gathers Info	Options	Decides	All
Automated Decision Making	Computer generates recommended options along with human, system selects best and system carries out	Gathers Info	Options (Both)	Decides	All
Supervisory Control (Management by Exception)	Computer generates recommended options, selects best and system carries out. Human can intervene if desired	Monitors & May Intervene	All	All	All
Full Automation	Computer carries out all aspects of task with no human intervention possible	Gathers Info	All	All	All

Figure 3.3: Levels of automation formed from the combination of human and computer performance across four task stages. From Designing for Situation Awareness: An Approach to Human-Centered Design (2nd ed., p. 185), by M. R. Endsley and D. J. Jones, 2012, Boca Raton, FL: CRC Press. Copyright 2012 by Taylor and Francis Group

The work on this project is divided in two stages that will provide a better overview over the proposed work-flow.

The first stage of automating data extraction is presented in figure 3.4, and in figure 3.5 the second stage of automation for data export.

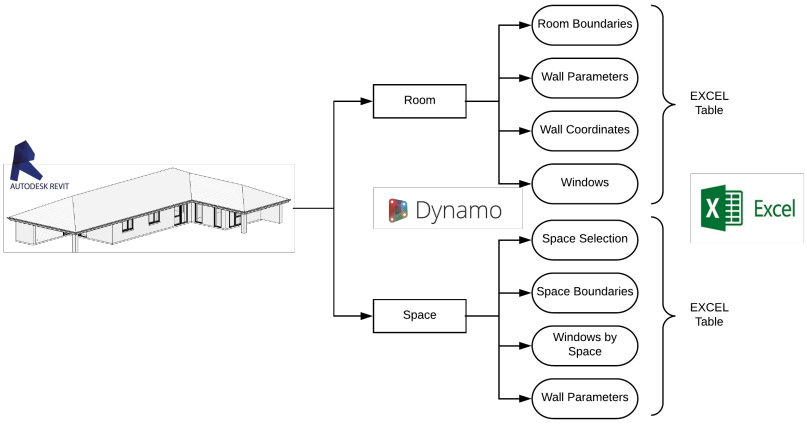


Figure 3.4: Road map of data extraction from Revit to Excel

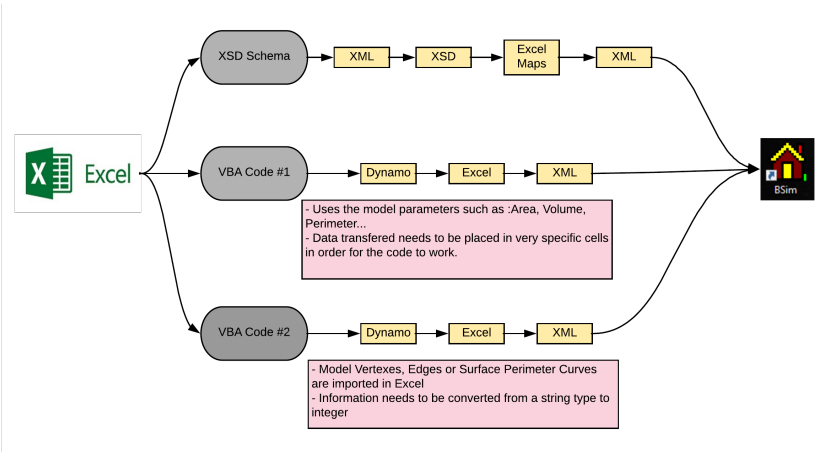


Figure 3.5: Road map of data input from Excel to BSim

3.3 Prediction of Result

It is expected that by automating part of the work-flow in a simulation process, the performance of engineers will improve in terms of time and precision. The repetitive task of collecting geometry data from an architect model and writing it in Excel spreadsheets will be performed by running a Dynamo script in Revit. Also the process of creating a model in BSim will be automated by exporting the necessary data from Excel in an XML format that can be opened in the simulation software. This improvements will allow the engineer to run simulations on multiple spaces and on more complex building layouts.

3.4 Intervene

To achieve the desired results, visual programming software Dynamo is used to extract geometry data from Revit and export it in Excel spreadsheet. In Excel the information is processed and exported in XML format so that it can be opened in BSim, figure 3.6.

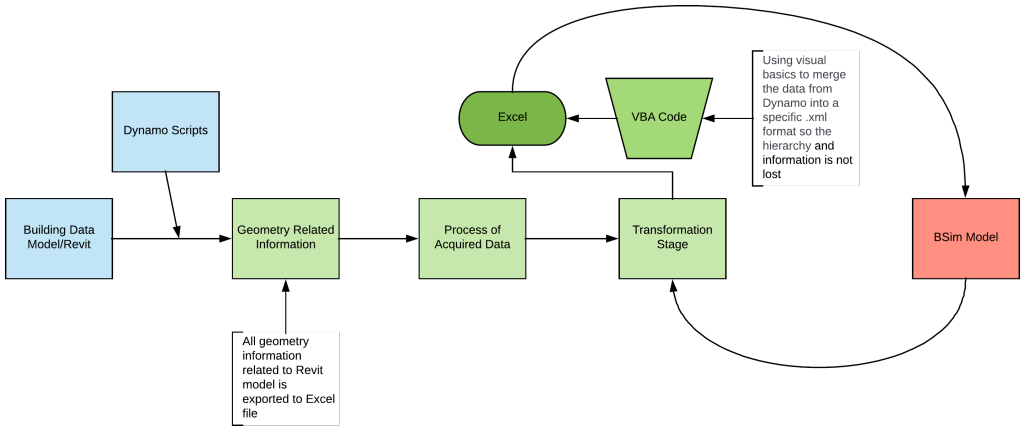


Figure 3.6: Work-flow of geometry extraction from Revit to BSim

Dynamo is chosen because it offers an easy approach to programming solution to extract data from Revit. Also the program is starting to be more commonly used to automate certain repetitive tasks in Revit and the fact that it is constantly improved by the user community makes it a very reliable tool in the BIM ecosystem. The process of data extraction is further described in chapter 4.

Excel is very common among engineers and so it is chosen as the interface where the extracted data can be visualized and parameter variations can be per-

formed if needed. It is also the program that is used when building a BSim model the manual way. All the geometry information about the building such as walls and windows/doors dimensions or construction type is written down in spreadsheets. The convenience of creating an Excel file and setting up spreadsheet in Dynamo contribute to the decision of using Excel as a link between Revit and BSim.

3.5 Observe Results

This step will be discussed in chapter 6.

Chapter 4

Dynamo

When preparing for a building simulation the process is often defined by repetitive tasks that involve collecting of data and then using it to build a model. Programming allows for developing a system where this type of loop actions can be automatized.

Dynamo is a Visual Programming tool that works as an extension of Revit and has seen an impressive amount of popularity in the construction industry. It is used in design by architects, designers, engineers, and even contractors [12].

Rather than typing a code, in Dynamo the programs are created by manipulating graphic elements called nodes, figure 4.1.

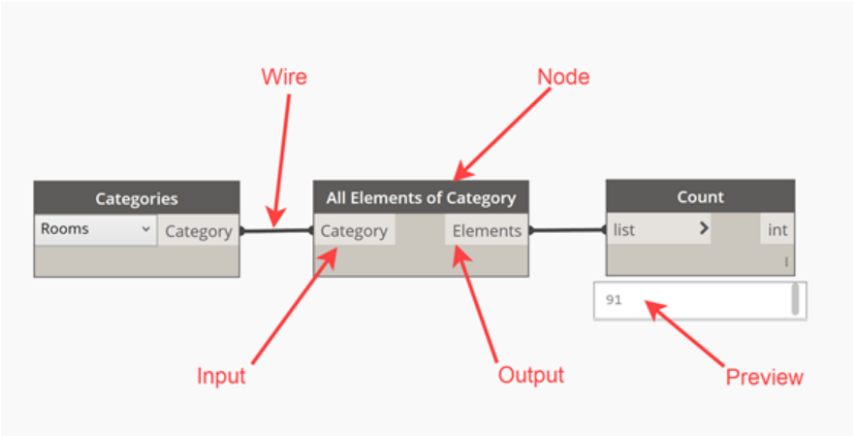


Figure 4.1: Example of visual programming (ArchSmarter)

In Dynamo, each node performs a specific task and they are connected by wires. The output from one node is connected to the input on another.

One of the most important benefit of Dynamo is the access to a library of nodes and packages. Thanks to a very active community this packages are updated all the time and they can help solve some very specific tasks.

4.1 Script Design

It is true that a lot of information can be exported from Revit but, looking at the BSim requirements for geometry, an analysis of what information to extract is needed and exporting only that information is critical to the speed and resource consumption of the script and to avoid crowding the output file with data that is not used later[14].

In figure 4.2 the hierarchy in Dynamo of abstract geometric types is presented and how this information is related with the BSim requirements. Seeing the similarity in hierarchy can help in deciding what type of nodes to choose for the script.

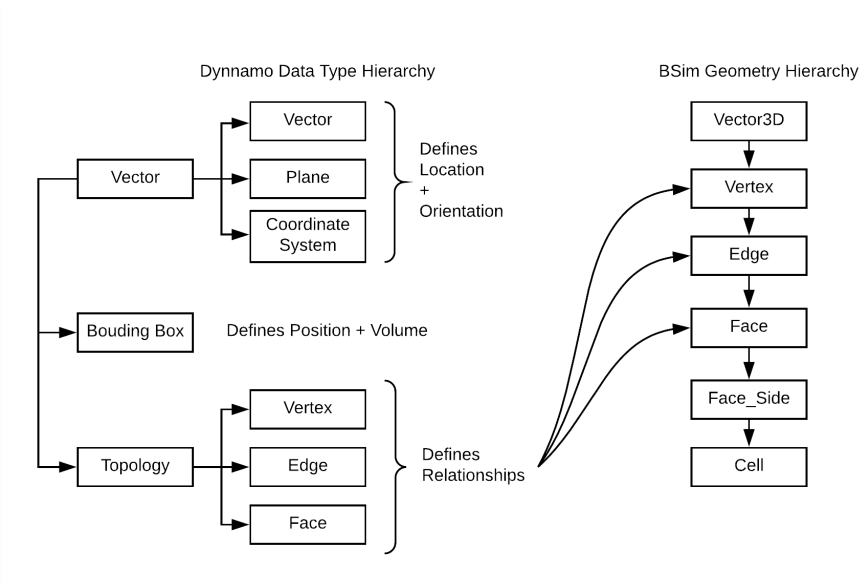
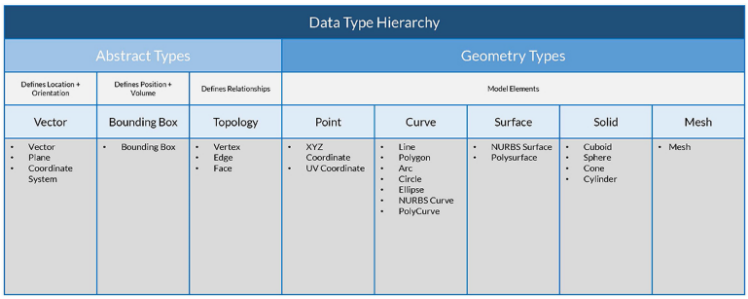


Figure 4.2: Data type hierarchy and relationships in Dynamo and BSim

A complete representation of Dynamo data type hierarchy can be seen in figure 4.3.



Data Type Hierarchy							
Abstract Types			Geometry Types				
Defines Location + Orientation	Defines Position + Volume	Defines Relationships	Model Elements				
Vector	Bounding Box	Topology	Point	Curve	Surface	Solid	Mesh
• Vector • Plane • Coordinate System	• Bounding Box	• Vertex • Edge • Face	• XYZ Coordinate • UV Coordinate	• Line • Polygon • Arc • Circle • Ellipse • NURBS Curve • PolyCurve	• NURBS Surface • Polysurface	• Cuboid • Sphere • Cone • Cylinder	• Mesh

Figure 4.3: Geometry and data type hierarchy in Dynamo (Dynamo Primer)

It is decided that using Topology nodes it is possible to extract 3 main components that will create the geometry in BSim.

Using Revit Application Programming Interface (API) LookUp add-on it is possible to investigate and navigate through the elements properties in the project database, figure 4.4.

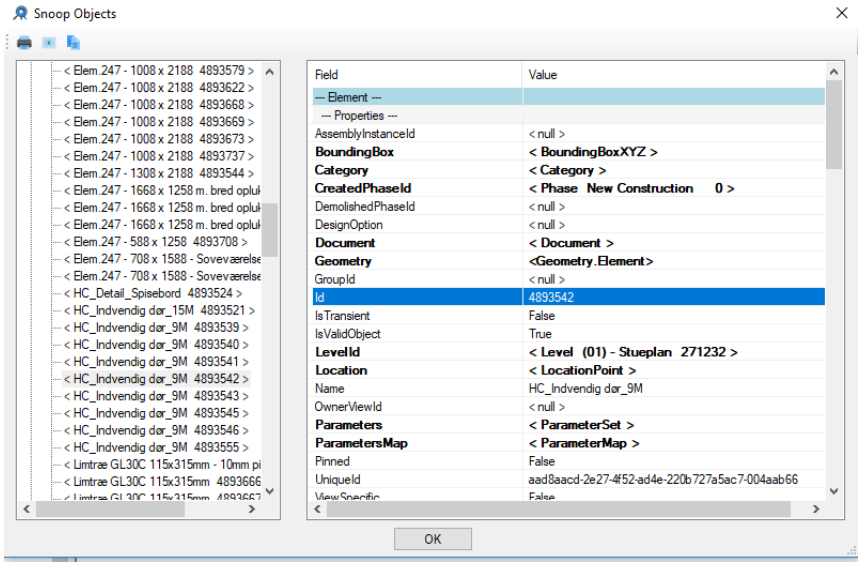


Figure 4.4: Element properties visualized by using LookUp add-on in Revit

This option can be used when creating a script. For example in figure 4.5 the relation between two elements can be seen in the properties menu. In this case a door is hosted by an interior wall. This information can help decide what type of node can be used in Dynamo in order to extract and export this data.

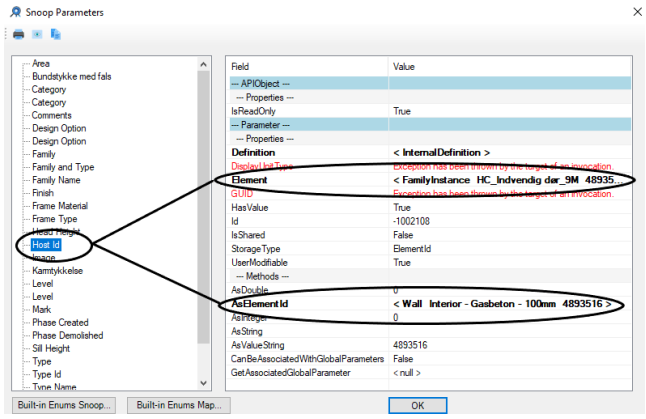


Figure 4.5: Relation between elements observed with Revit LookUp

4.2 Data extraction strategies

One of the strategies of collecting the required data is by creating a script that will reach in the BIM model and extract parameters such as room boundaries, wall openings and wall parameters. This method will be mentioned to from now as the Room Method. In figure 4.6 it is described the approach in Dynamo when using this method.

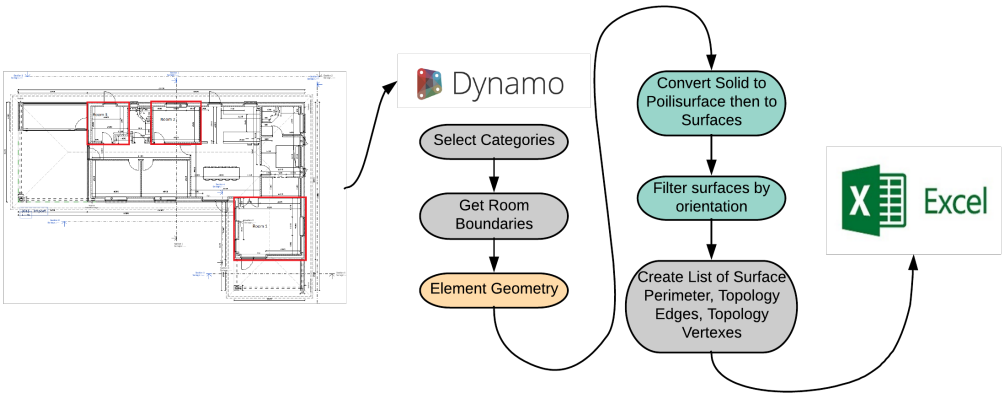


Figure 4.6: Road map describing the wrokflow for the Room Method

The use of this method is intended to select the walls that form the boundaries of a room, and get the walls geometry. In order to obtain the wall vertices and edges, Dynamo will convert the geometry into solids then to polysurfaces and surfaces. The entire process of the script can be seen in diagram B.1 and a more detailed representation in appendix B figure B.2 and B.3.

The process of extracting topology out of a Revit model is basically a reverse engineering on the geometry hierarchy, and can be seen in figure 4.7.

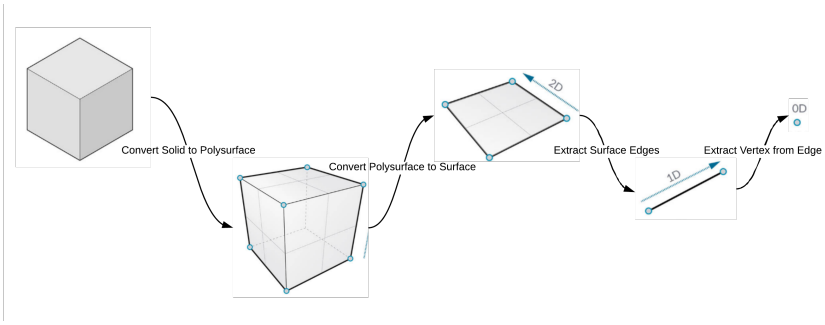


Figure 4.7: Process of collecting topology data in Dynamo script

The wall type and windows are extracted using scripts presented in appendix B figure B.4 and B.5. The diagrams are kept separated for a more clear view on how they are designed. But they are merged into the main script in the Room Method.

The result of running this script although promising it does have some drawbacks.

The method is successful in:

- Selecting all the rooms in the model.
- Taking the room boundary walls and convert the solid geometry into surfaces.
- Can extract the inner surface of walls and write the vertex coordinates, edges and surface perimeter.

The method does fail in some categories:

- Inconsistent results.
- Surfaces extracted appear to be in a random selection.
- Can not select specific windows or doors.
- Can not define which rooms should be selected in larger models.

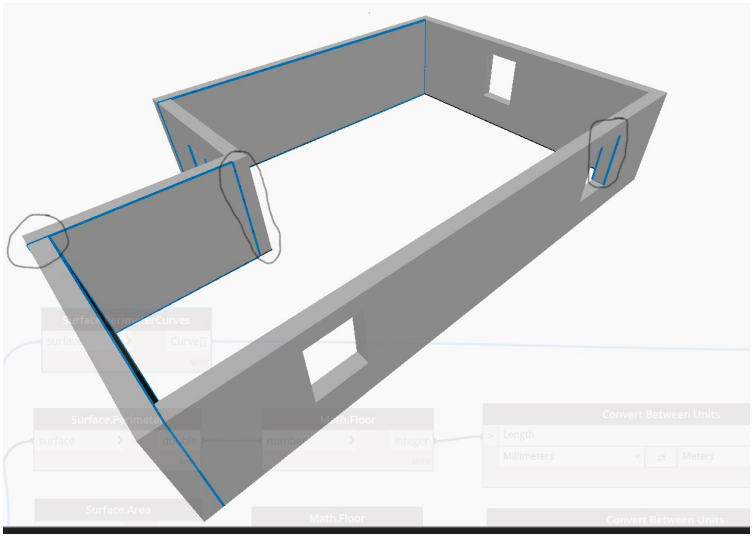


Figure 4.8: Geometrical representation of data extracted using the Room Method

In figure 4.8 the extracted surfaces are highlighted with blue and it is visible that the script does not take the vertices from corner to corner.

An explanation for this failed results might be the way that walls are connected in Revit. In figure 4.9 it is visible that the wall connect in a perpendicular way and the intersection creates a group of 3 vertices.

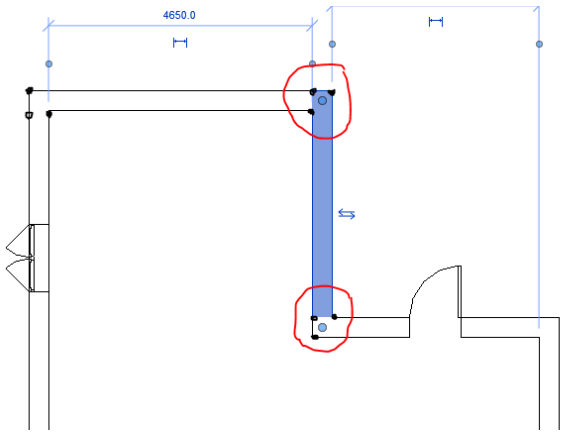


Figure 4.9: Wall connection in Revit

Since the walls form the boundary of a room and the connection between them in Revit is not suitable for extracting the coordinates, a different method of defining the room boundary must be designed.

In this second attempt the script is redesigned so that the spaces are selected instead of rooms and the method will be named the Space Method. The script workflow is described visually in figure 4.10.

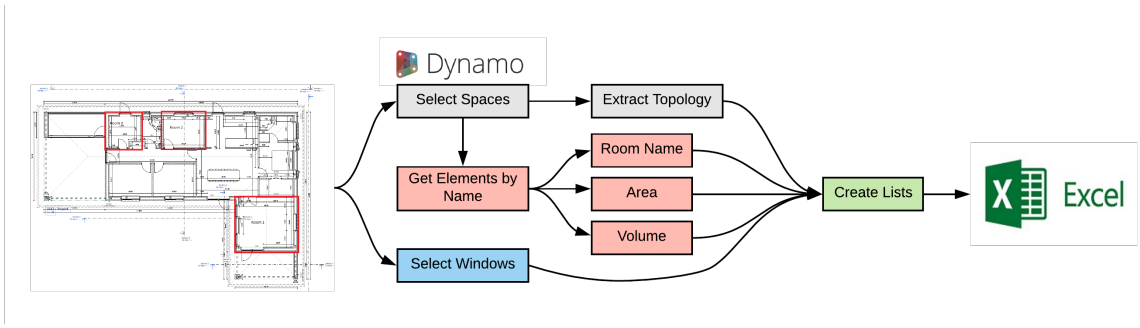


Figure 4.10: Road map of data extraction using the Space Method

Because the extraction of topology information was actually successful in Room Method, the same principle is kept also in this new script.

This time the script will take the volume of the room and converting it into solid. By doing so, the results after running the script will be formed out of coordinates of the inner surfaces from the selected spaces. The selection of spaces

it is not done in Dynamo. It is actually more convenient if the received architect model does not have spaces assigned, giving the opportunity to the specialist to select which of the building spaces he wants to simulate.

The windows will also have to be selected manually when running the script. This way only the windows that are located in the selected spaces will be exported. It is possible to do now, compared to the previous script, thanks to a new script design for window selection. This script is represented in figure 4.11.

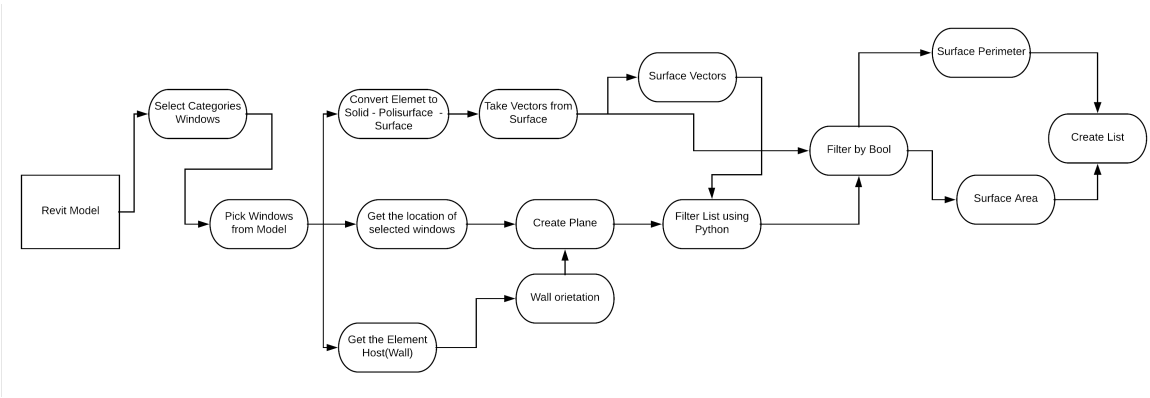


Figure 4.11: Diagram of selecting and extracting data for windows in the Space Method

Topology nodes are used again to get the windows coordinates. By using the node GetElement Host(Wall) it is possible to see where exactly each window is located.

A complete diagram of the script for the Space method can be seen in appendix B figure B.6.

When running the script, even on more complex models, the results are noticeably improved.

The script is successful in performing the following tasks:

- Select the model spaces.
- Extract the space inner wall surface.
- Export space parameters.
- Export inner surface vertex and edges.
- Extract windows according to each space selected.

Some of the drawbacks of the script are:

- It does not extract the wall thickness.
- Windows need to be manually selected for each space.

Looking at the 3D representation of the result from the Space Method in figure 4.12, and comparing with the 3D result from the Room Method, figure 4.8, it can be said that a more complete and clear result is achieved with the Space method.

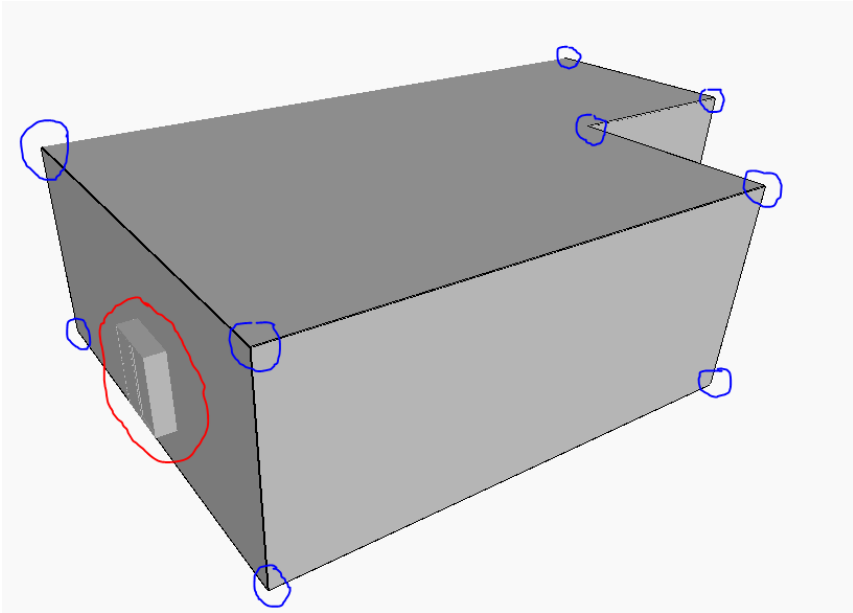


Figure 4.12: Geometry model in Dynamo as result of data extraction with the Space Method

The corners of the model represent the vertices of the inner surface from the space. It is also visible the window that appears to be floating, this is a result of the missing wall thickness. Although the wall thickness problem can be solved by adding an extra script to the method, one that will collect the data about the wall type. And based on this information it will be the engineer that will make the correction manually in Excel.

When comparing the success and failures of each method, it is the selection of Spaces that give the best result and it is considered for future work, despite the extra manual steps that need to be performed.

Chapter 5

Excel to BSim

After receiving the geometry data via Dynamo script in the Excel file, the next step is to export this data to BSim and reconstruct the model.

A method to transform the information received in to a format readable by BSim needs to be used.

Two methods will be tried in this project. Using an XSD schema and the Excel Visual Basic for Applications. Both of this methods can export data in an XML format.

5.1 Using the XSD schema

XSD stand for XML Schema Definition and it is used to describe in an abstract way the characteristics of an object and the relationship with other objects in an XML document[13].

It is chosen to work with XSD because it can generate XML documents and keep the desired hierarchy.

In order to work with XSD first an XML document needs to be converted. A BSim XML document it is used since it is desired that the schema to contain the same type of object and hierarchy as it is in BSim. The conversion can be done easily using online converters XML to XSD.

Next step is to import the schema in Excel using the Developer tab and upload via XML Maps. From this point the XSD needs to be populated with information about the model and just export in XML format.

In appendix C figure C.1 and C.2, examples of the imported XSD schema in Excel can be seen. And in figure 5.1, the schema is filled with information about the BSim model.

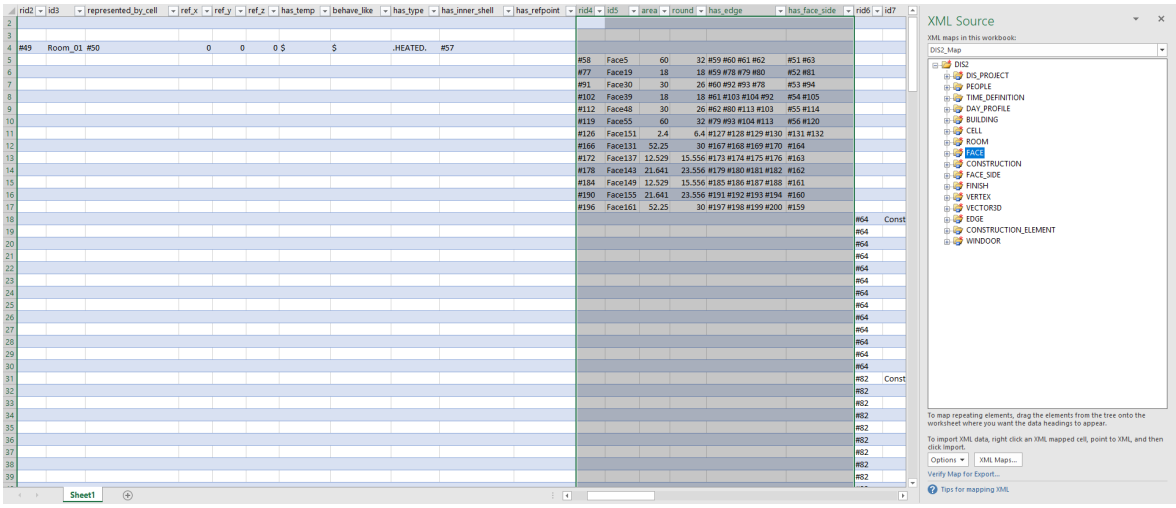


Figure 5.1: Example of XSD populated with information in Excel

Although this method seemed very promising in the beginning, it is abandoned pretty early.

The reason being the fact that Excel can not export a parent/child relationship, also defined as a list-of-lists. This limitation of the program was discovered during the process and so it is decided to try a different method.

It is true that the method can indeed export from XSD to an XML format, but not the type required by BSim.

5.2 Visual Basic for Applications

Visual Basic for Application (VBA) is the programming tool in Excel. VBA programs are also known as macros and they are typically used to automate repetitive tasks. It can also be used to create custom applications with Excel or it can format, delete and rearrange data before outputting to a different document[15].

5.2.1 Using VBA coding with model measurements

In this method it is attempted to automate the process of writing down data about building geometry in an Excel spreadsheet.

The testing of this method is done on received Excel files from Kim Trangbæk Jønsson, where the author has manually input all the data and has created the VBA code used for exporting in BSIm. Examples from the Excel file can be seen in figure 5.2 and 5.3.

FLOOR ATTRIBUTES	Ground surface without walls: mÂ²	Volume: mÂ³	Ground Perimeter: m	Ceiling Perimeter: m	Walls with opening: mÂ²	Walls without opening: mÂ²	Ground surface with all walls: mÂ²	Ground surface with interior walls: mÂ²	Exterior Wall Th
Ground Floor	83.85	212	98.35	104.99	265.16	223.09	98.63	87.8	0.25 m
ROOM ATTRIBUTES	Ground Surface: : mÂ²	Volume: mÂ³	Ground Perimeter: m	Ceiling Perimeter: m	Walls with opening: mÂ²	Walls without opening: mÂ²	SurfaceOfDoors: mÂ²	SurfaceOfWindows: mÂ²	Ceiling Height
Ground Floor									
Living Room	19.06	48	18.07	19.09	48.21	30.81	10.24	7.16	2.53 m
Kitchen	8.74	22	12.28	12.28	31	28.02	1.63	1.35	2.53 m
Dining Room	22.37	57	19.43	19.43	49.07	44.5	1.63	2.94	2.53 m
Hall	1.52	4	2.81	5.11	12.91	6.58	6.33	0	2.53 m
Bathroom	2.31	6	5.38	6.09	15.38	13.93	1.46	0	2.53 m
Bathroom	5.37	14	8.55	9.33	23.56	21.97	1.59	0	2.53 m
Bedroom	12.34	31	13.28	14.08	35.57	32.64	1.65	1.28	2.53 m
Study	9.33	24	11.23	12.25	30.94	27.72	2.08	1.14	2.53 m
Closet	2.81	7	7.33	7.33	18.51	16.88	1.63	0	2.53 m
WALL ATTRIBUTES	Wall	Symbol	Surface: : mÂ²	Surface without openings: mÂ²	Width: : m	Height: m	Annotation	Type	Distance to Flo
Ground Floor									
Living Room	Wall 0	Wall	7.18	0.03	2.84	2.53		Wall	
Living Room	Wall 0	Sliding Window	7.16		2.84	2.52	3	Window	
Living Room	Wall 1	Wall	16.93	12.03	6.7	2.53		Wall	
Living Room	Wall 1	Hinged Door	1.63		0.81	2.03	1	Door	
Living Room	Wall 1	Hinged Door	1.63		0.81	2.03	1	Door	
Living Room	Wall 1	Hinged Door	1.63		0.81	2.02	2	Door	
Living Room	Wall 2	Wall	7.18	5.1	2.84	2.53		Wall	

Figure 5.2: Example of manually written information in measurements spreadsheet

Udvælg rum til BSim:	Living Room																	
Gulvareal	19.06																	
	Højde:	Bredde:	Vægtykkelse:	Vindueshøjde:	Vinduesbredde:	Brystningshøjde:	Ydervæg	Vindue Dist. til væg venstre	Vindue Dist. til væg højre					Koordinater				
	[m]	[m]	[mm]	[m]	[m]	[m]	[0/1]	[m]	[m]	Bund	Bund	Top	Top		x	y	z	
Væg 0	2.53	2.84	150	2.52	2.84	0.02	1	0	0	Vertex3	Vertex4	Vertex7	Vertex8	Vertex1	0	0	0	
Væg 1	2.53	6.7	150				0			Vertex4	Vertex2	Vertex8	Vertex6	Vertex2	2.99	0	0	
Væg 2	2.53	2.84	150				0			Vertex1	Vertex2	Vertex5	Vertex6	Vertex3	0	6.925	0	
Væg 3	2.53	6.7	150				0			Vertex3	Vertex1	Vertex7	Vertex5	Vertex4	2.99	6.925	0	
	Længde:	Bredde:	Kon. Tykkelse	Mod ude										Vertex5	0	0	2.68	
	[m]	[m]	[mm]	[0/1]										Vertex6	2.99	0	2.68	
Gulv	2.84	6.7	100	1										Vertex7	0	6.925	2.68	
Loft	2.84	6.7	100	0										Vertex8	2.99	6.925	2.68	
														Vertex9	0.075	6.925	0.12	
Rotation	90	°												Vertex10	2.915	6.925	0.12	
														Vertex11	0.075	6.925	2.64	
														Vertex12	2.915	6.925	2.64	

Figure 5.3: Data collected by the VBA code from the measurements spreadsheet

The necessary data for this VBA code consists of rooms, walls and windows/doors attributes. This attributes are: areas, volume, perimeter and dimensions of height, length and width. All of which can be obtained using the Dynamo scripts.

The challenge is to organize all of this data in list and export it from Dynamo. Since the VBA code collects the information from already defined cells it is important that the export to be done in those specific locations, otherwise the entire process will not work.

Although possible, figure 5.5, this process will require multiple scripts to run at the same time and each script to have a separate export to Excel, figure 5.4.

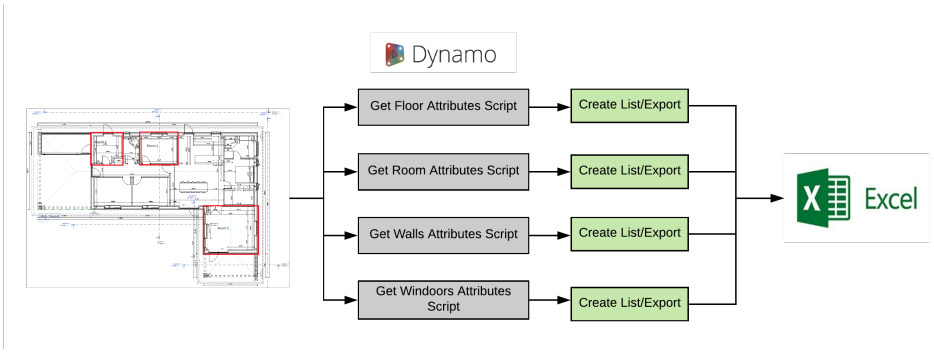


Figure 5.4: Workflow of data extraction to get required attributes

1	PLAN ATTRIBUTES				
2	Ground surface with all walls: m²				
3	Ground surface with interior walls: m²				
4	Ground surface without walls: m²				
5	Volume: m³				
6	Perimeter: m				
7	Floor				
8	Rooms				
9	Bedroom				
10	Bathrooms				
11	Windows				
12	Walls with opening: m²				
13	Walls without opening: m²				
14	Latitude				
15	Longitude				
16	Date				
17					
18	FLOOR ATTRIBUTES	Ground surface without walls: m²	Volume: m³	Ground Perimeter: m	C
19	Room				
20	Room				
21	Room				
22	Room				C
23	Room				
24	Room	12.08	29.455872	14.04	
25	Room	12.08	29.455872	14.04	
26	Room	11.078	27.0125952	13.44	
27	Room	7.819	19.0658496	11.66	
28	Room	11.613	28.3171392	20.06	
29	Room	5.317	12.9649728	9.62	
30	Room	40.334	98.3504256	26.34	
31	Room	7.5756	18.47234304	11.14	
32	Room	10.6251	25.90824384	13.04	
33	Room	5.0718	12.36707712	9.58	
34	Room	23.432	57.1365888	19.38	S
35	Ground Floor				

Figure 5.5: Example of data imported in Excel using Dynamo

The use of a multitude of scripts will make the process more vulnerable to software errors. It will also not be a reliable method in case it will be used on bigger models.

And BSim does not require this amount of information for modeling geometry, since it can calculate the areas, volumes and perimeters by itself.

Considering the effort of designing the scripts and the method not being viable for a variety of projects, a different way of exporting the data to BSim will be tried.

5.2.2 Using VBA coding with model coordinates

This method is designed to work with the main elements that stand at the core of building a geometry model in BSim. As it was presented in chapter 2 Bsim Hierarchy, this elements are the vertex, edge and face. It is considered that having only the edges exported to Excel, enough information will be provided to build the geometry in BSim. This is decided because a face is defined by edges and the edge will consist out of a set of vertices that define the start and end point of the edge.

When the information is imported in Excel, it will be presented in the form of a string. This means that a set up of transforming this type of data to integer type needs to be implemented. The workflow of this process can be seen in figure 5.6.

A string is a type of data used in programming and is used to represent text rather than numbers. It is composed from a set of characters that can also include spaces and numbers[5].

An integer is a commonly used data type in programming and it represents a whole number, not a fraction. Integers can be positive, negative or zero[4].

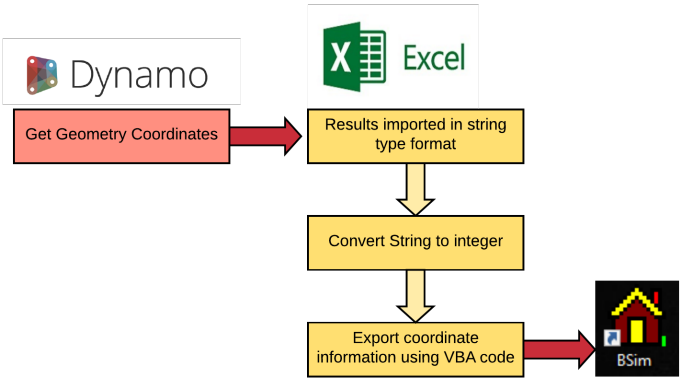
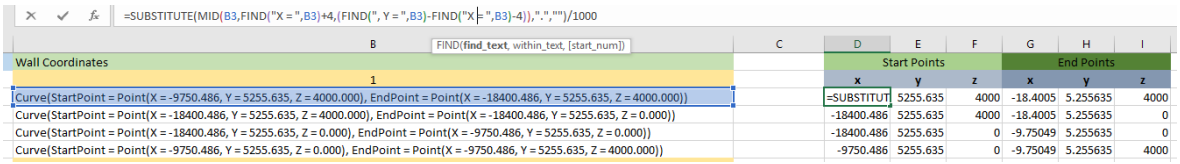


Figure 5.6: Workflow for the second method of exporting data from Excel to BSim

The conversion of string to integer is done by using formulas in Excel, figure 5.7. This process can also be automated by using VBA to loop through the cells.



=SUBSTITUTE(MID(B3,FIND("X = ",B3)+4,(FIND(", Y = ",B3)-FIND("X = ",B3)-4)),".","")/1000

FIND(find_text, within_text, [start_num])

	B	C	D	E	F	G	H	I
	Wall Coordinates		Start Points			End Points		
	1		x	y	z	x	y	z
	Curve(StartPoint = Point(X = -9750.486, Y = 5255.635, Z = 4000.000), EndPoint = Point(X = -18400.486, Y = 5255.635, Z = 4000.000))		=SUBSTITUT	5255.635	4000	-18.4005	5.255635	4000
	Curve(StartPoint = Point(X = -18400.486, Y = 5255.635, Z = 4000.000), EndPoint = Point(X = -18400.486, Y = 5255.635, Z = 0.000))		-18400.486	5255.635	4000	-18.4005	5.255635	0
	Curve(StartPoint = Point(X = -18400.486, Y = 5255.635, Z = 0.000), EndPoint = Point(X = -9750.486, Y = 5255.635, Z = 0.000))		-18400.486	5255.635	0	-9.75049	5.255635	0
	Curve(StartPoint = Point(X = -9750.486, Y = 5255.635, Z = 0.000), EndPoint = Point(X = -9750.486, Y = 5255.635, Z = 4000.000))		-9750.486	5255.635	0	-9.75049	5.255635	4000

Figure 5.7: Using Excel formula to extract numbers from a text

An overview of the Excel spreadsheet can be seen in appendix C figure C.3.

Since the coordinates in the spreadsheet are related to the inner surfaces of a space, a notation in Excel about the wall thickness should be made. Or alternatively, an export of wall type can be made in a separate sheet.

Although the final step in the process is not automated, it is possible to design a VBA code that will collect the information and export it in an XML document that can be opened in BSim.

Chapter 6

Conclusion

From the experience of 3 semesters working with BSim on each semester project, and with insight from the industry, it can be said that the modeling part in the program has proven to be a real challenge and time consuming process.

Considering the importance of simulations and the benefits of having them in a very early stage of the project, the need of a method that can improve this process it is more and more necessary.

Because BSim is such a common tool for simulations, in this project it was tried to automate parts of the simulation process, and thus creating a semi-automatic workflow.

Dynamo and Excel are used in attempt of achieving this goal, and in each program more than one methods are tried.

In Dynamo, two methods are designed for collecting data in different ways. The Room method is focused on extracting the geometric coordinates of a room boundary, which are the walls.

Using the Space method, the coordinates of the inner surfaces from a room are extracted. This points define the volume of the room otherwise named a Cell in BSim.

The Space method is considered to work better when compared to the Room. This is due to the ability of selecting what window should be exported, consistency in results, and the amount of data extracted is easier to work with.

In Excel three methods are tried, although the first option of using the XSD schema is proved to be unusable from an early stage.

The second option of using VBA code and measurements of the model it is disconsidered due to high amount of data required and the difficulty of organizing this data in Excel spreadsheets.

The third and last method is to use only the vertex start and end points of edges and to export this coordinates using VBA code. Although the export it is not automated, it is considered to be possible with the right coding in VBA.

In conclusion, it is the Space method from Dynamo and the export of vertices by VBA coding from Excel that will be recommended for the semi-automatic workflow presented in figure 6.1.

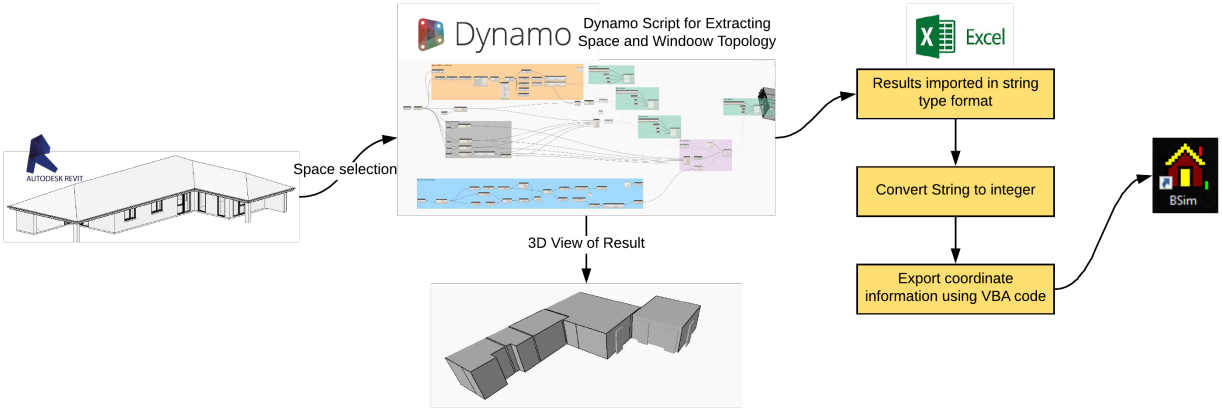


Figure 6.1: Workflow for proposed solution

Bibliography

[1] Vladimir Bazjanac. *IFC BIM-Based methodology for semi-automated building energy performance simulation*. Paper presented at International Conference of Information technology in Construction, Santiago, Chile 2008. 2008.

[2] Vladimir Bazjanac. *Space boundary requirements for modeling of building geometry for energy and other performance simulation*. V. 2010. 2010.

[3] A. Bryman and E. Bell. *Business Research Methods*. 3rd Edition. Oxford University Press, 2011.

[4] P. Christensson. *Integer Definition*. https://techterms.com. 2006.

[5] P. Christensson. *String Definition.* https://techterms.com. 2006.

[6] Jon W. Hand Michael Kummert Crawley Dury B. and howpublished = Brent T. Griffith title = Contrasting the Capabilities of Building Energy Performance Simulation Programs year = 2005.

[7] Refsnes Data. *XML Tree*. https://www.w3schools.com/xml/xml$_{t}$*ree.asp*.

[8] A.R.Florita G.P.Henze D.Jacob S.Burhhene. *Optimizing building energy simulation models in the face of uncertainty*. `https://www.researchgate.net/publication/228822516_Optimizing_building_energy_simulation_models_in_the_face_of_uncertainty`. 2010.

[9] John Dudovskiy. *The Ultimate Guide to Writing a Disertation in Business Studies: A Step by Step Assistance*. january 2018 Edition. E-Book, 2018.

[10] Peter Nørkjær Gade. *Practical Dynamo - Lecture*. IT System Development Lecture. 2018.

[11] C.Valaseda G.I.Giannakis M.A.Garcia. *A methodology to automatically generate geometry inputs for energy performance simulation from IFC BIM models*. `https://www.researchgate.net/publication/286452825_A_methodology_to_automatically_generate_geometry_inputs_for_Energy_Performance_Simulation_from_IFC_BIM_models`. 2015.

[12] Michael Kilkelly. *What is Dynamo and 5 reasons you should be using it*. https://archsmarter.com/what-is-dynamo-revit/. 2018.

[13] George Lawton Margaret Rouse. *XSD (XML Schema Definition)*. https://searchmicroservices.te XML-Schema-Definition. 2015.

[14] Pavel Pavlov. *Automation of information flow freom Revit to BSim using Dynamo*. `https://projekter.aau.dk/projekter/files/231026964/Dynamo_Project.pdf`. 2015.

[15] Website i programmer.info. *Getting started with Excel VBA*. https://www.i-programmer.info/ebooks/automating-excel/1264-getting-started.html. 2018.

[16] Arto Kiviniemi Vladimir Bazjanac. *Reduction, simplification, translation and interpretation in the exchange of model data*. `https://www.researchgate.net/publication/228963955_Reduction_simplification_translation_and_interpretation_in_the_exchange_of_model_data`. 2007.

[17] C. Wilkins and Kiviniemi A. *Engineering- Centric BIM*. ASHRAE Journal, December 2008. 2008.

Appendix A

BSim


```xml
<DIS2>
<DIS_PROJECT rid="#1"><id>BS7122-255</id><description></description><database>F:\BUILDING
    ENERGY DESIGN\MASTER_PROJECT\BSim_Models\TEST_SimpleRomm_001.mdb</database><tstep>0
    </tstep><options>0</options><scale>109.85</scale><grid>1</grid><layer_thick>0
    </layer_thick><start_time></start_time><end_time></end_time><has_design_parm>$
    </has_design_parm></DIS_PROJECT>
<PEOPLE rid="#2"><id>PersonNormal</id><person_heat>0.072</person_heat><person_moist>0.044
    </person_moist><protect>1</protect></PEOPLE>
<PEOPLE rid="#3"><id>Standard</id><person_heat>0.1</person_heat><person_moist>0.06
    </person_moist><protect>1</protect></PEOPLE>
<PEOPLE rid="#4"><id>MediumActivity</id><person_heat>0.12</person_heat><person_moist>0.123
    </person_moist><protect>1</protect></PEOPLE>
<PEOPLE rid="#5"><id>HighActivity</id><person_heat>0.17</person_heat><person_moist>0.198
    </person_moist><protect>1</protect></PEOPLE>
<TIME_DEFINITION rid="#6"><id>Always</id><hour>1-24</hour><day>1-7</day><week></week><month>1
    -12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#7"><id>WeekDays 1-24</id><hour>1-24</hour><day>1-5</day><week></week
    ><month>1-12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#8"><id>WeekDays 9-16</id><hour>9-16</hour><day>1-5</day><week></week
    ><month>1-12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#9"><id>WeekDays 9-17</id><hour>9-17</hour><day>1-5</day><week></week
    ><month>1-12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#10"><id>Week-ends 1-24</id><hour>1-24</hour><day>6-7</day><week></week
    ><month>1-12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#11"><id>Oct-Apr</id><hour>1-24</hour><day>1-7</day><week></week><month
    >1-4 10-12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#12"><id>May-Sept</id><hour>1-24</hour><day>1-7</day><week></week
    ><month>5-9</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#13"><id>Spring</id><hour>1-24</hour><day>1-7</day><week></week><month
    >3-5</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#14"><id>Summer</id><hour>1-24</hour><day>1-7</day><week></week><month
    >6-8</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#15"><id>Autumn, Fall</id><hour>1-24</hour><day>1-7</day><week></week
    ><month>9-11</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#16"><id>Winter</id><hour>1-24</hour><day>1-7</day><week></week><month
    >1-2 12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#17"><id>HeatingSeason</id><hour>1-24</hour><day>1-7</day><week>1-19 39
    -53</week><month></month><tariff_class>0</tariff_class><protect>1</protect
    ></TIME_DEFINITION>
<TIME_DEFINITION rid="#18"><id>Aftensmad</id><hour>19</hour><day>1-7</day><week></week><month
    >1-12</month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION>
<TIME_DEFINITION rid="#19"><id>SchoolHolidays</id><hour>1-24</hour><day>1-7</day><week>26-32
    </week><month></month><tariff_class>0</tariff_class><protect>1</protect></TIME_DEFINITION
    >
```


Figure A.1: BSim XML text

```
▼ object {1}
  ▼ DIS2 {16}
    ▶ DIS_PROJECT {12}
    ▶ PEOPLE [4]
    ▶ TIME_DEFINITION [24]
    ▶ DAY_PROFILE [18]
    ▶ BUILDING {8}
    ▶ CELL [2]
    ▶ ROOM {11}
    ▶ FACE [13]
    ▶ CONSTRUCTION [6]
    ▶ FACE_SIDE [20]
    ▶ FINISH [14]
    ▶ VERTEX [20]
    ▶ VECTOR3D [32]
    ▶ EDGE [40]
    ▶ CONSTRUCTION_ELEMENT [3]
    ▶ WINDOOR {28}
```

Figure A.2: BSim XML tree structure

```
▼ BUILDING {8}
    id  : TEST_001
    rotation : 0
    current : 0
    height : 0
    composed_of : #49
    located_on_site : $
    has_thermal_zones : value
    _rid : #48
▼ CELL [2]
  ▼ 0 {4}
      id  : Cell2
      volume : -180
      bounded_by : #51 #52 #53 #54 #55 #56
      _rid : #50
  ▶ 1 {4}
▶ ROOM {11}
▶ FACE [13]
▶ CONSTRUCTION [6]
▶ FACE_SIDE [20]
▶ FINISH [14]
▶ VERTEX [20]
▶ VECTOR3D [32]
```

Figure A.3: BSim XML tree hierarchy

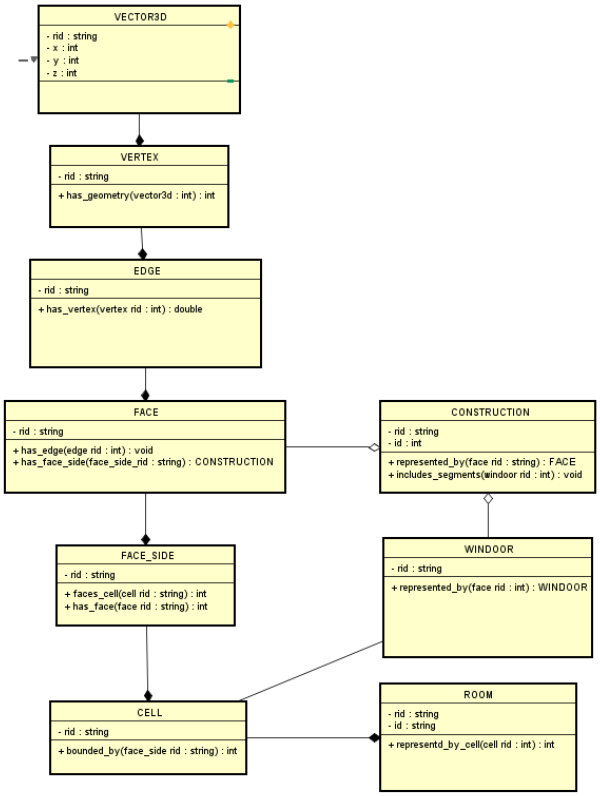


Figure A.4: BSim parent-child hierarchy diagram

EDGE (4)

<>FACE_SIDE

<>FACE

EDGE (4)

	@rid	<>id	<>edge_length	<>has_vertex
<>	#194	Edge156	9.5	#157 #151
<>	#193	Edge157	2.278	#157 #147
<>	#192	Edge158	9.5	#143 #147
<>	#191	Edge159	2.278	#143 #151

<>FACE_SIDE

@rid	#159
<>id	FaceSide160
<>locked	0
<>has_normal	#195
<>faces_cell	#57
<>has_face	#196

<>FACE

@rid	#196
<>id	Face161
<>area	52.25
<>round	30
<>has_edge	#197 #198 #199 #200
<>has_face_side	#159

EDGE (4)

	@rid	<>id	<>edge_length	<>has_vertex
<>	#200	Edge162	5.5	#147 #149
<>	#199	Edge163	9.5	#149 #145
<>	#198	Edge164	5.5	#145 #143
<>	#197	Edge165	9.5	#143 #147

VECTOR3D (6)

	@rid	<>id	<>x	<>y	<>z
<>	#195	FaceSide160's Normal	0	0	-1
<>	#189	FaceSide154's Normal	1	0	0
<>	#183	FaceSide148's Normal	0	-1	0
<>	#177	FaceSide142's Normal	-1	0	0
<>	#171	FaceSide136's Normal	0	1	0
<>	#165	FaceSide130's Normal	0	0	1

Figure A.5: Graphic representation of XML parent-child relation

Appendix B

Dynamo

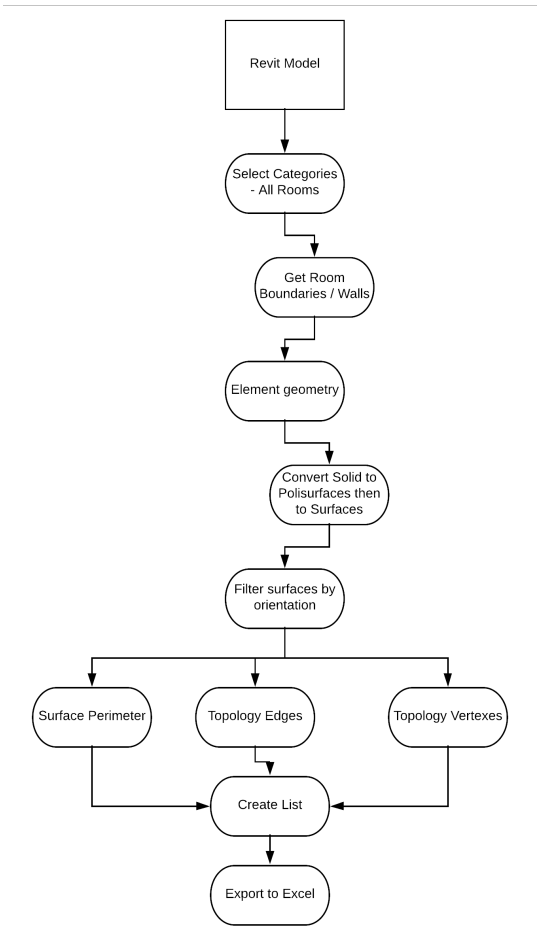


Figure B.1: Script diagram of export room coordinates from Revit model

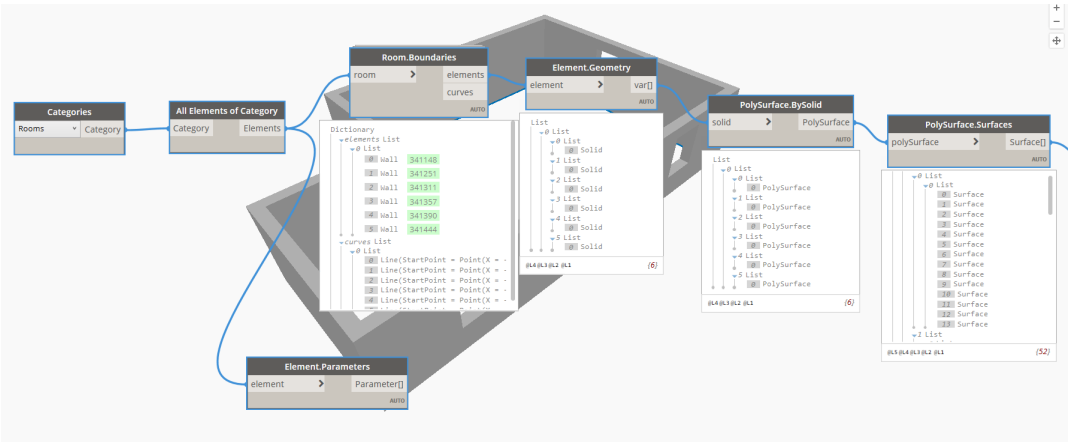


Figure B.2: Getting room geometry from Revit using Room Method part1

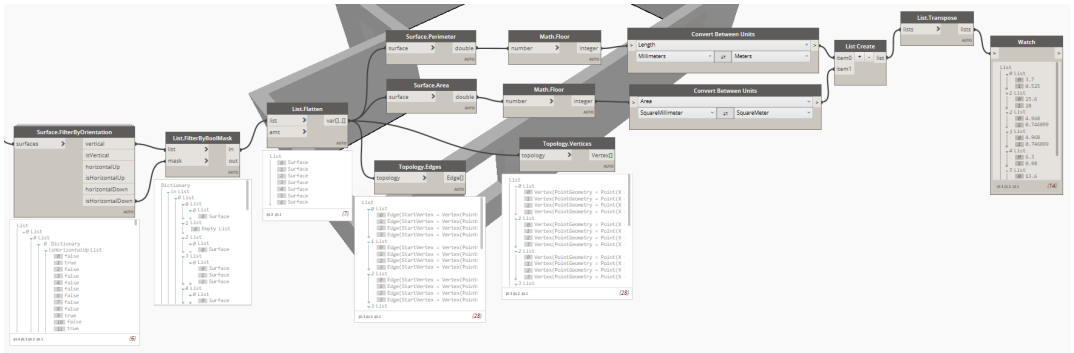


Figure B.3: Getting room geometry from Revit using Room Method part2

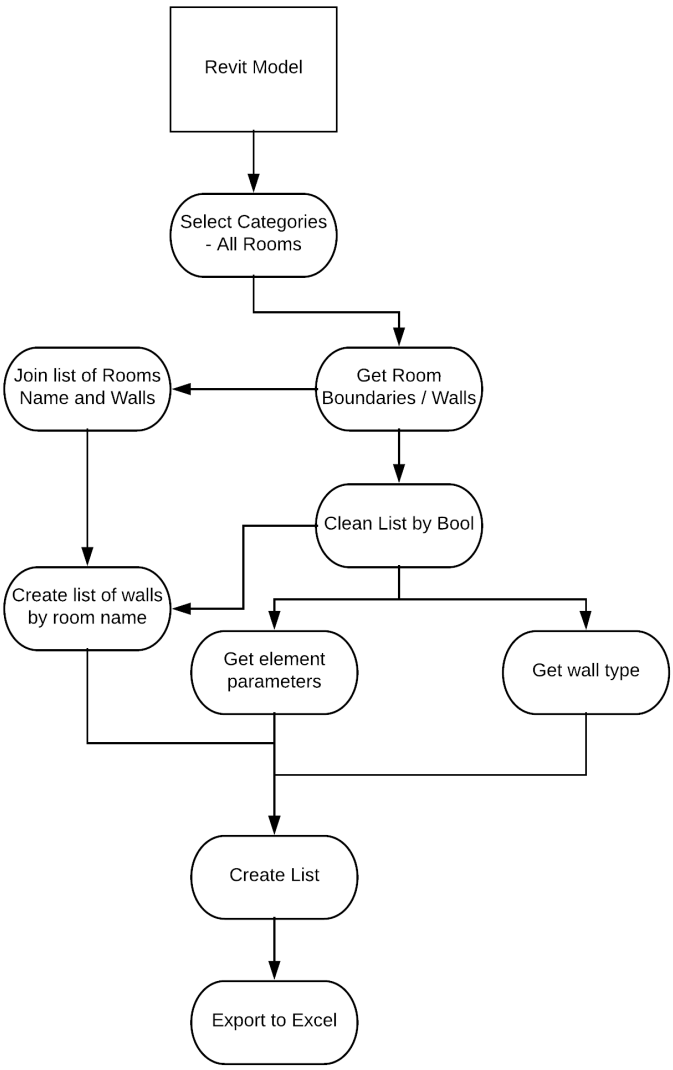


Figure B.4: Getting the wall type from Revit to Excel

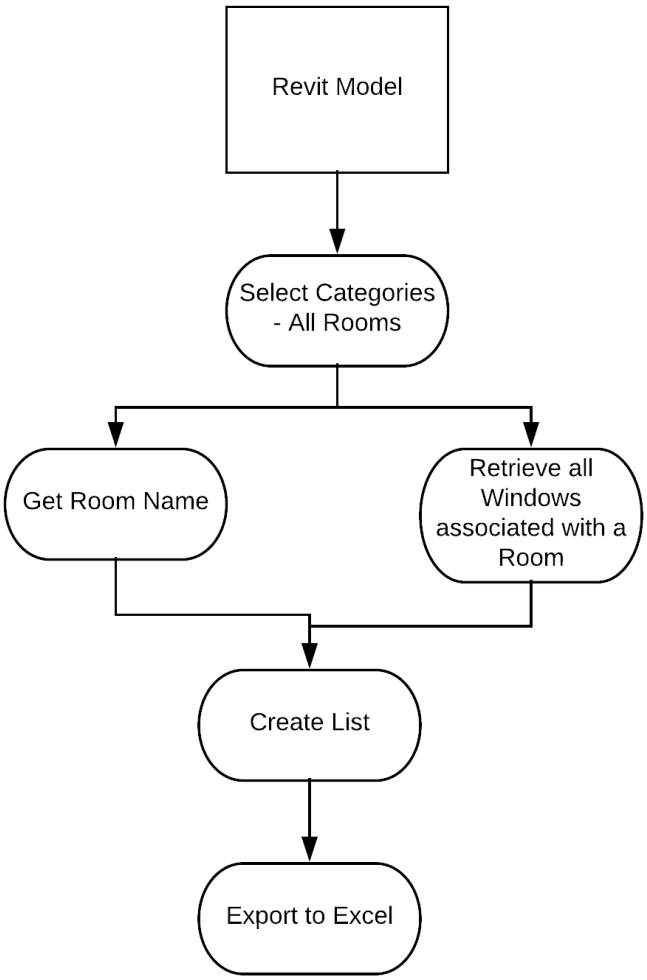


Figure B.5: Getting the windows from Revit to Excel

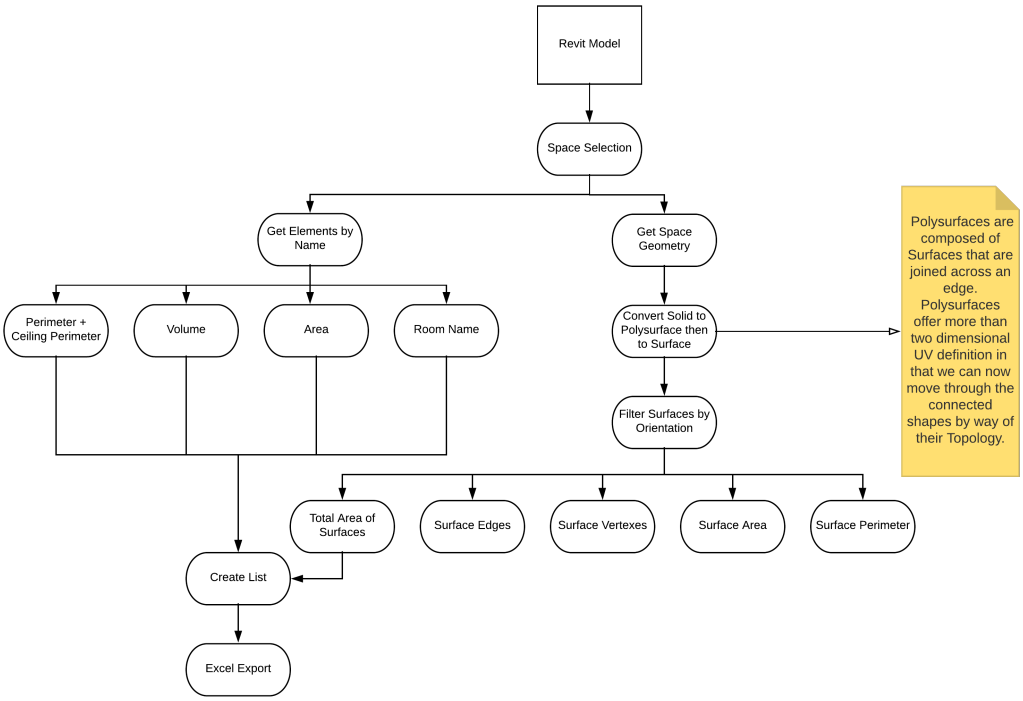


Figure B.6: Script diagram for the Space Method

Appendix C

Excel

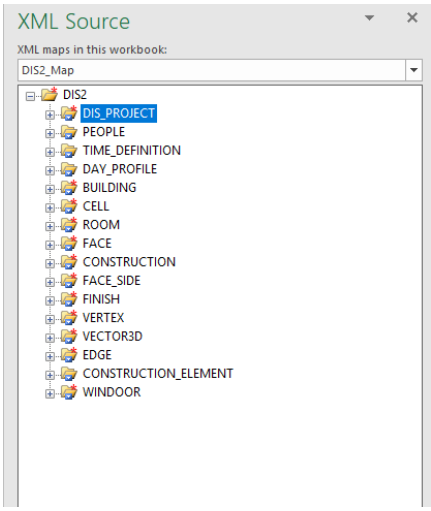


Figure C.1: XSD structure tree in Excel

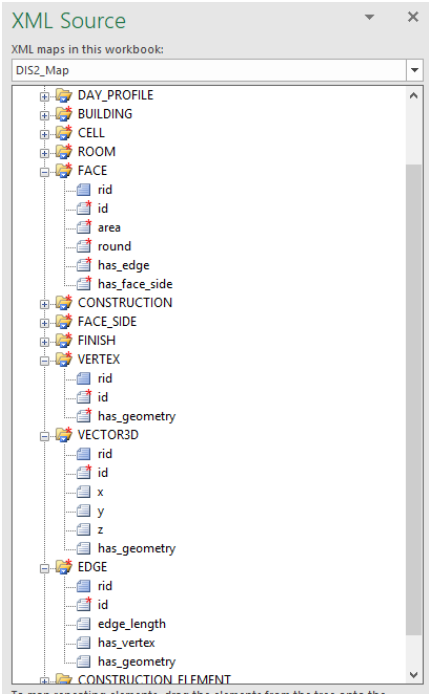
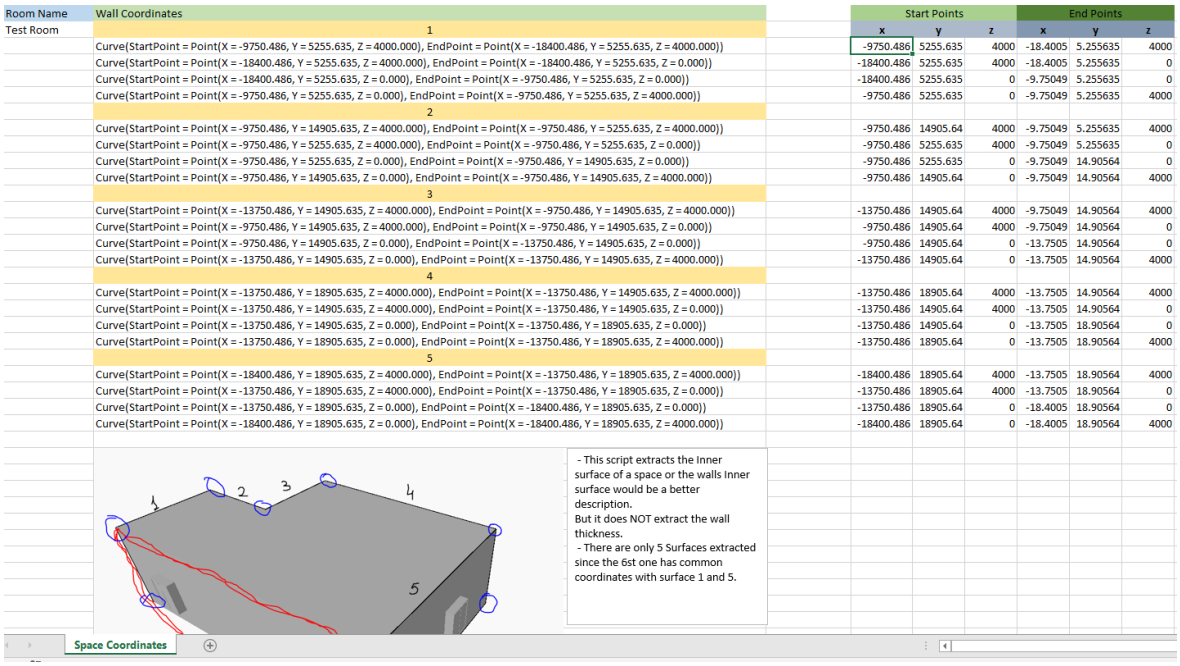


Figure C.2: XSD details of elements



Room Name	Wall Coordinates	Start Points			End Points		
Test Room	1	x	y	z	x	y	z
	Curve(StartPoint = Point(X = -9750.486, Y = 5255.635, Z = 4000.000), EndPoint = Point(X = -18400.486, Y = 5255.635, Z = 4000.000))	-9750.486	5255.635	4000	-18.4005	5.255635	4000
	Curve(StartPoint = Point(X = -18400.486, Y = 5255.635, Z = 4000.000), EndPoint = Point(X = -18400.486, Y = 5255.635, Z = 0.000))	-18400.486	5255.635	4000	-18.4005	5.255635	0
	Curve(StartPoint = Point(X = -18400.486, Y = 5255.635, Z = 0.000), EndPoint = Point(X = -9750.486, Y = 5255.635, Z = 0.000))	-18400.486	5255.635	0	-9.75049	5.255635	0
	Curve(StartPoint = Point(X = -9750.486, Y = 5255.635, Z = 0.000), EndPoint = Point(X = -9750.486, Y = 5255.635, Z = 4000.000))	-9750.486	5255.635	0	-9.75049	5.255635	4000
	2						
	Curve(StartPoint = Point(X = -9750.486, Y = 14905.635, Z = 4000.000), EndPoint = Point(X = -9750.486, Y = 5255.635, Z = 4000.000))	-9750.486	14905.64	4000	-9.75049	5.255635	4000
	Curve(StartPoint = Point(X = -9750.486, Y = 5255.635, Z = 4000.000), EndPoint = Point(X = -9750.486, Y = 5255.635, Z = 0.000))	-9750.486	5255.635	4000	-9.75049	5.255635	0
	Curve(StartPoint = Point(X = -9750.486, Y = 5255.635, Z = 0.000), EndPoint = Point(X = -9750.486, Y = 14905.635, Z = 0.000))	-9750.486	5255.635	0	-9.75049	14.90564	0
	Curve(StartPoint = Point(X = -9750.486, Y = 14905.635, Z = 0.000), EndPoint = Point(X = -9750.486, Y = 14905.635, Z = 4000.000))	-9750.486	14905.64	0	-9.75049	14.90564	4000
	3						
	Curve(StartPoint = Point(X = -13750.486, Y = 14905.635, Z = 4000.000), EndPoint = Point(X = -9750.486, Y = 14905.635, Z = 4000.000))	-13750.486	14905.64	4000	-9.75049	14.90564	4000
	Curve(StartPoint = Point(X = -9750.486, Y = 14905.635, Z = 4000.000), EndPoint = Point(X = -9750.486, Y = 14905.635, Z = 0.000))	-9750.486	14905.64	4000	-9.75049	14.90564	0
	Curve(StartPoint = Point(X = -9750.486, Y = 14905.635, Z = 0.000), EndPoint = Point(X = -13750.486, Y = 14905.635, Z = 0.000))	-9750.486	14905.64	0	-13.7505	14.90564	0
	Curve(StartPoint = Point(X = -13750.486, Y = 14905.635, Z = 0.000), EndPoint = Point(X = -13750.486, Y = 14905.635, Z = 4000.000))	-13750.486	14905.64	0	-13.7505	14.90564	4000
	4						
	Curve(StartPoint = Point(X = -13750.486, Y = 18905.635, Z = 4000.000), EndPoint = Point(X = -13750.486, Y = 14905.635, Z = 4000.000))	-13750.486	18905.64	4000	-13.7505	14.90564	4000
	Curve(StartPoint = Point(X = -13750.486, Y = 14905.635, Z = 4000.000), EndPoint = Point(X = -13750.486, Y = 14905.635, Z = 0.000))	-13750.486	14905.64	4000	-13.7505	14.90564	0
	Curve(StartPoint = Point(X = -13750.486, Y = 14905.635, Z = 0.000), EndPoint = Point(X = -13750.486, Y = 18905.635, Z = 0.000))	-13750.486	14905.64	0	-13.7505	18.90564	0
	Curve(StartPoint = Point(X = -13750.486, Y = 18905.635, Z = 0.000), EndPoint = Point(X = -13750.486, Y = 18905.635, Z = 4000.000))	-13750.486	18905.64	0	-13.7505	18.90564	4000
	5						
	Curve(StartPoint = Point(X = -18400.486, Y = 18905.635, Z = 4000.000), EndPoint = Point(X = -13750.486, Y = 18905.635, Z = 4000.000))	-18400.486	18905.64	4000	-13.7505	18.90564	4000
	Curve(StartPoint = Point(X = -13750.486, Y = 18905.635, Z = 4000.000), EndPoint = Point(X = -13750.486, Y = 18905.635, Z = 0.000))	-13750.486	18905.64	4000	-13.7505	18.90564	0
	Curve(StartPoint = Point(X = -13750.486, Y = 18905.635, Z = 0.000), EndPoint = Point(X = -18400.486, Y = 18905.635, Z = 0.000))	-13750.486	18905.64	0	-18.4005	18.90564	0
	Curve(StartPoint = Point(X = -18400.486, Y = 18905.635, Z = 0.000), EndPoint = Point(X = -18400.486, Y = 18905.635, Z = 4000.000))	-18400.486	18905.64	0	-18.4005	18.90564	4000

Figure C.3: Excel spreadsheet with geometry coordinates and conversion of string type data to integer

1	i»¿PLAN ATTRIBUTES									
2	Ground surface with all walls: mÂ²	98.63								
3	Ground surface with interior walls: mÂ²	87.8								
4	Ground surface without walls: mÂ²	83.85								
5	Volume: mÂ³	212								
6	Perimeter: m	44.33								
7	Floor	1								
8	Rooms	9								
9	Bedroom	1								
10	Bathrooms	2								
11	Windows	5								
12	Walls with opening: mÂ²	265.16								
13	Walls without opening: mÂ²	223.09								
14	Latitude	57.027								
15	Longitude	10.005								
16	Date	8/30/2018								
17										
18	FLOOR ATTRIBUTES	Ground surface without walls: mÂ²	Volume: mÂ³	Ground Perimeter: m	Ceiling Perimeter: m	Walls with opening: mÂ²	Walls without opening: mÂ²	Ground surface with all walls: mÂ²	Ground surface with interior walls: mÂ²	Exterior Wall Th
19	Ground Floor	83.85	212	98.35	104.99	265.16	223.09	98.63	87.8	0.25 m
20										
21										
22	ROOM ATTRIBUTES	Ground Surface: : mÂ²	Volume: mÂ³	Ground Perimeter: m	Ceiling Perimeter: m	Walls with opening: mÂ²	Walls without opening: mÂ²	SurfaceOfDoors: mÂ²	SurfaceOfWindows: mÂ²	Ceiling Height
23	Ground Floor									
24	Living Room	19.06	48	18.07	19.09	48.21	30.81	10.24	7.16	2.53 m
25	Kitchen	8.74	22	12.28	12.28	31	28.02	1.63	1.35	2.53 m
26	Dining Room	22.37	57	19.43	19.43	49.07	44.5	1.63	2.94	2.53 m
27	Hall	1.52	4	2.81	5.11	12.91	6.58	6.33	0	2.53 m
28	Bathroom	2.31	6	5.38	6.09	15.38	13.93	1.46	0	2.53 m
29	Bathroom	5.37	14	8.55	9.33	23.56	21.97	1.59	0	2.53 m
30	Bedroom	12.34	31	13.28	14.08	35.57	32.64	1.65	1.28	2.53 m
31	Study	9.33	24	11.23	12.25	30.94	27.72	2.08	1.14	2.53 m
32	Closet	2.81	7	7.33	7.33	18.51	16.88	1.63	0	2.53 m
33										
34	WALL ATTRIBUTES	Wall	Symbol	Surface: : mÂ²	Surface without openings: mÂ²	Width: : m	Height: m	Annotation	Type	Distance to Floo
35	Ground Floor									
36	Living Room	Wall 0	Wall	7.18	0.03	2.84	2.53		Wall	
37	Living Room	Wall 0	Sliding Window	7.16		2.84	2.52	3	Window	
38	Living Room	Wall 1	Wall	16.93	12.03	6.7	2.53		Wall	

Figure C.4: Building information manually added in Excel (Kim Trangbæk Jønsson)