main scientific competencies that is required to be known by you BEFORE starting the project

This project explores the definitions, working principals, fundamentals, dynamics, and the theory of construction of S-graphs.

Having some knowledge on S-graphs is helpful prior to the project as the topic will be explored in its entirety and will be the primary focus of the scientific research.

To achieve the task of exploring S-graphs in depth, certain tools and environments are necessary.

BIM models is the most fundamental aspect of creating S-graphs, in the methodology followed by me and the team behind this project. An understanding of BIM models, their uses and basic functionalities is important.

To complement S-graphs, knowing about the working fundamentals of robots and sensors is recommended, it gives insight on why certain strategies are adopted when forming the S-graphs.

main technical competencies that is required to be known by you BEFORE starting the project

In my previous BSP, I used digital construction plans in a BIM format to extract specific data using Dynamo for Revit.

In this BSP, which is a continuation of my previous BSP, the reverse premise will be done. Standardized input data will be utilized to alter an existing BIM or create a new BIM building plan.

The process will be automated and generalized, such that the tool can generate a BIM from any input data so long as it follows certain restrictions or alter an already existing BIM.

To do this task, certain tools are used and consequently basic working knowledge of the uses of these tools is necessary.

The most fundamental tool is REVIT, which accommodates BIM models. However, most of the manipulation will be done in Dynamo for Revi, it is a tool for Revit, which allows for manipulation of BIM models.

Dynamo is Python compatible, so Python scripts can be run inside dynamo scripts to do more specific tasks.

Finally BimVision, a visualising tool for BIM models, is helpful. It allows the user and client to see the changes in real time that the dynamo script actuates on the existing BIM.

A Scientific Deliverable

In my last BSP-2, my scientific question was as follows:” How can the information available in BIM models be used by robots for their autonomous navigation?”.

I will be continuing to explore this question, focusing more on the information available and its usage on robots, S-graphs.  
Previously, in the Scientific part, I worked with BIM (Building Information Modeling) and explored what data was relevant from BIMs, methods, and techniques of extracting said data and briefly introduced S-graphs. In this BSP-3, on the Scientific part, I want to further explore this scientific question. The scientific research explores BIMs with an emphasis on S-graphs, what they are, how are they generated, their purpose, etc.

o-----o

Information available in BIM models is vast, in this project the aim is to explore S-graphs.

S-graphs contain topological and metric-semantic information about a building. This information can be extrapolated from BIM models, and, from direct and odometrical robot data.

Me, personally, will not be working on the robot, however, scientific background on the working principles of how robot data is used will be explored. So, Robotics, and Sensor Input/Output and Computer Vision will be explored.

To complement the robot data, BIM models, which contain geometrical, semantic, and topological data from a building is explored. This means exploring BIM as a digital data format.

Furthermore, data within BIM must be handled. This permits us to access specific data, altering data if needed, overall, allowing making available data useful. So, Data Handling and Data Processing is a key component.

Finally, to utilize all available data, Python and Dynamo for Revit are used, so Programming is necessary. To conceptualize S-graphs, Computer Vision and Graphs are necessary.

There are a lot of topics to cover, but the goal is not to explore them all deeply, but to cover some topics, just enough, to be able to explore S-graph in depth. By the end of the scientific part, certain sub-questions should be answered.

o---o

Some sub-questions, necessary to understand S-graphs are proposed:

How to construct an S-graph? why make an S-graph? What does an S-graph provide to the robot? What information in an S-graph can the robot use? How can the robot use that information? Etc.

These are not set in stone; however, they serve as a guideline for providing a path for exploring and understanding S-graphs.

To answer the S-graph questions, some other questions arise, such as: why use BIM as a format?

This is quite a relevant question, because, answering this question by focusing on the proper aspects, gives understanding behind the reasoning on certain technical decisions on making S-graphs.

Other questions relating to robots and sensor I/O will be addressed as well. These will be explored in much less detail, however.

Finally, after all the necessary information is introduced, S-graphs will be explored.

S-graphs are the topic of the research, so they will be explored in depth. S-graphs will be introduced, definitions will be explained, what components makes an S-graph, how to obtain these S-graphs components to make an S-graph, explain and define all these components, how to conglomerate the necessary components to make up an S-graph as well as any relevant information related to S-graphs within the scope of the scientific question.

o---o

S-graphs are quite complex data. They comprise topological and metric-semantic data. These terms can be quite vague, so I will first give some meaning to these terms by defining them in the context of S-graphs and buildings.

Further on, I will explore which data is available from the accessible sources, BIM models and Robot Outputs.

As such, I will first try to explain what data is available, resulting in exploring geometric, semantic, topological data from BIMs. Also exploring odometry and how sensors work in a basic sense in from robots. How to obtain this data from BIM models and from Robot Outputs is also relevant.

As such, BIM as a digital format and basics robotics and robot sensor IO will be explored.

Additionally, what data from these available data sources is relevant for the purposes of S-graphs.

Finally, everything S-graph within a delimited scope will be explored. If possible, a simpler and basic S-graph will be constructed as well.

A Technical Deliverable 1

The technical deliverable of this project is a tool, named “BIM bluepring manager “, upon which a standardized set of data is transformed into a BIM construction plan. This tool can be used to either create an entirely new BIM model, or, to alter a pre-existing BIM model with the desired properties mentioned in the input of the user.

o---o

BIM blueprint manager or BIMbm, is a tool that, as the name implies, manages blueprints. These blueprints refer to, but not exclusively, construction plans.

These construction plans are digitalized into a BIM format. BIM is widely used, it is a digitalized blueprints format, and in the context of this project, BIM construction blueprints are handled.

Various tools and technologies are compatible with BIM models.

Revit, is the tool of choice to handle BIM for this project. Revit renders BIM models, allows for manipulation, visualization, and many others. Revit is a broad software, and it has many sub tools which are compatible with it, one of them being Dynamo for Revit.

Dynamo for Revit, or Dynamo for short, is a plugin for Revit and a library for dynamo Nodes \*|1 .

Unlike Revit, which to my knowledge, only allows for in place modification of BIM models, Dynamo allows for the creation of scripts that can modify a BIM upon being executed.

Other worth mentioning tools, that, although not fundamental, do help in the completion of this task. Python 2 and Python 3 are compatible with Dynamo and will be used whenever some functionality is required that is not already provided by Dynamo’s default dynamo Nodes.

Python 3 will be for certain more complex data handling and processing tasks, and Python 2 might be used to access certain older tools which were developed in Python 2 such as the library IronPython which was used in my previous BSP-2, in Dynamo, also in the context of S-graphs.

Lastly, BIM vision, a visualization tool for BIM models is to be used throughout the making of BIMbm. Although Revit offers some basic visualization features of BIMs, BIM vision is a tool specifically designed for that and has more functionalities, so, it is preferred over Revit. It allows the user and developer to see the changes made by BIMbm in real time clearer.

\*|1 https://github.com/DynamoDS/DynamoRevit

o---o

The goal for BIMbm is to allow the user to alter or create BIMs.

Dynamo is the main tool used for coding BIMbm because of the utility of the scripts that can be run to achieve a specific desired task.

Standardized Input :

The script, must receive a standardized input that contains the data to be modified.

A good example of standardized input data that is compatible with the desired final version of BIMbm is my BSP2 tool.

BSP2’s technical tool, extracts structural data from a construction’s BIM and outputs it into an excel file.

This data is combined in one single excel filed (.xlxs). The data is structured and organized into excel sheets. Each excel sheet refers to a single component type withing the structure, e.g. walls. In each sheet, columns refer to different properties of the components e.g. Length. As can be seen in the attachment below.

A screenshot of a computer

Description automatically generated

BSPbm’s Requirements:

The BSPbm tool would alter values in BIM models according to the user’s instructions.

The values to be changed will be input in an interface.

The user, when changing values, will not need to keep in mind inter relationships of walls. Meaning that, if the user changes the data pertaining to an arbitrary Wall A into Wall A’; Consequently, Wall B that is originally connected to Wall A would then automatically have its starting point in the new end point of wall A’. Resulting in a conflict free and more convenient way to change data.

The tool will house similar features for other BIM components: walls, rooms, windows, and doors.

It will then generate a new BIM model with the altered data.

The tool can alter or create entirely new models.

o---o

The first functionality, and the only one to be implemented until now, is to create BIM files from available data. Suppose the goal is to create a simple 2D planar view of a building, with a fence around it, such as in the picture below.

A diagram of a house

Description automatically generated

A screenshot of a computer

Description automatically generatedThe premise is to input the information relevant to the creation of the building in some sort of storage that can handle the data. For the earlier builds of BIMbm, the storage of input data is excell .xlxs, later on it most likely will be .CSV format.

The data represents points with x and y coordinates in the A and B columns respectively.

An earlier version of the tool BIMbm creates a simple BIM from an excel sheet filled with data.

A drawing of a diagram

Description automatically generated

This is the output from the excel sheet of data.

This method can be applied to bigger and more complex BIMs. Furthermore, this concept can be expanded upon. For instance, we can give these connected dots, wall properties.

A blue cube with a square top

Description automatically generated

The final product is supposed to allow the end user to change any existing BIM file. Walls can be changed without disrupting the file, meaning that the integrity of the building is preserved, and subsidiary changes are made to accommodate for a changed wall. Doors and windows are also planned to be modifiable. A group of walls can be used to create a room, and existing rooms will be retained as a room when the walls are modified. All these functionalities are expected to be in a plugin dropdown list, whereby the user can click a plugin and it will run certain functionalities. Here is an abstract example of this concept.

A screenshot of a computer

Description automatically generated