Department of Mechanical and Mechatronics Engineering

Using Dynamo to Perform Quality Assurance for Smoke Detector Design

A Report Prepared For:

The University of Waterloo

Prepared By:

Sophy Li

May 27th, 2019

116 University Ave W.
Waterloo, Ontario, N2L 3E2

May 27, 2019

Prof William Melek
Director, Mechatronics Engineering
Department of Mechanical and Mechatronics Engineering
University of Waterloo
200 University Ave W.
Waterloo, Ontario, N2L 3G1

Dear Professor Melek,

This report, entitled Using Dynamo to Perform Quality Assurance for Smoke Detector Design, was written to fulfil my 2B work term report requirements. This is my 1st work report submission.

The process of designing infrastructure requires engineers to conform to Clause 8.2.3.10 of the National Canadian Fire Alarm Code. This states that spot type ceiling detectors must be placed at least 450mm away from air diffusers in a building. A Dynamo script that performed quality assurance was developed and analyzed in this report. Smoke detectors that violated the Code were flagged in the model for the users to modify their placement. The script can be re-run, each time reflecting the most recent changes made by the user.

This report was written entirely by me, and has not received any previous academic credit at this or any other institution. I would like to acknowledge the help of Lance Desjardins, Electrical Engineer, who provided me with resources on learning Dynamo as well as defining the purpose of the project. I would also like to thank Steve Yilmaz, BIM coordinator and Akira Jones, Electrical Engineer, together who provided a sample Revit model and Revit families for me to test the Dynamo script. My role in the project was to create a Dynamo script that flags smoke detectors in the Revit model which violate the Canadian Fire Alarm Code. Learning how to use Dynamo, choosing a suitable topic, developing the script, and writing the report lasted 3 months. Mr. Desjardins can be contacted at lance.desjardins@hhangus.com.

Best Regards

Sophy Li
ID: 20703756
2B Mechatronics Engineering

Table of Contents

List of Figures

List of Tables

Summary

Quality assurance is critical for infrastructure projects, so there are no design clashes in a shared model. However, the QA process is often repetitive, tedious and requires time and effort to be invested by employees. The report introduces computer scripting as an alternative to traditional quality assurance processes. It investigates a problem that often needs to be checked in the design stage of infrastructure projects and analyzes the script that automates the checking process.

Clause 8.2.3.10 of the National Canadian Fire Alarm Code states: all spot-type smoke detectors must be placed at least 450mm away from air diffusers mounted to the ceiling of a building. Elements in level 1 of a sample Revit model is checked with a script written in Dynamo. The script calculates the distance between smoke detectors and air diffusers and flags smoke detectors that violate the Code. Flagging is achieved by changing the color of smoke detectors from black to red. After the user makes adjustments and re-runs the program, the script re-calculates distance values and updates the color of smoke detectors in the model.

Two iteration of scripts are developed in the process and are analyzed in the report. The first script draws boundary lines around elements and checks for intersecting geometries. Although the first script is successful in terms of functionality, the lines used for testing cannot be hidden in the model. A second script is developed to check if certain points along the boundary of an air diffuser overlap with the smoke detecting regions. The second script meets all criteria and aims specified in the report.

Further improvements are needed to accommodate for different sizes of air diffusers, reduce run time, check for more clauses specified in the Code, and reduce human intervention through algorithms and Python scripting.

1.0 Introduction

1.1 Background

1.1.1 History of Building Modelling

Architects, engineers, and consultants (AEC) in the building industry rely on drawings to effectively coordinate their designs. Traditionally, all designs were completed on paper and passed back and forth between parties. Even today, people favor this system due to its speed and ease of making updates in the preliminary design phase [1]. However, as computer technologies evolved, designers began to search for new ways that were more exact and consistent to model buildings. By the late 1900s, CAD (Computer Aided Design) was rapidly developing and becoming implemented in the building industry [2]. Popular CAD software enabled drawings to be quickly produced, conveniently shared, and easily stored. Substituting conventional practices with CAD also improved consistency and accuracy, so that communication between engineers and architects was greatly improved. The shift from hand-modelling to computer-modelling was the first step towards modern building design (Figure 1).

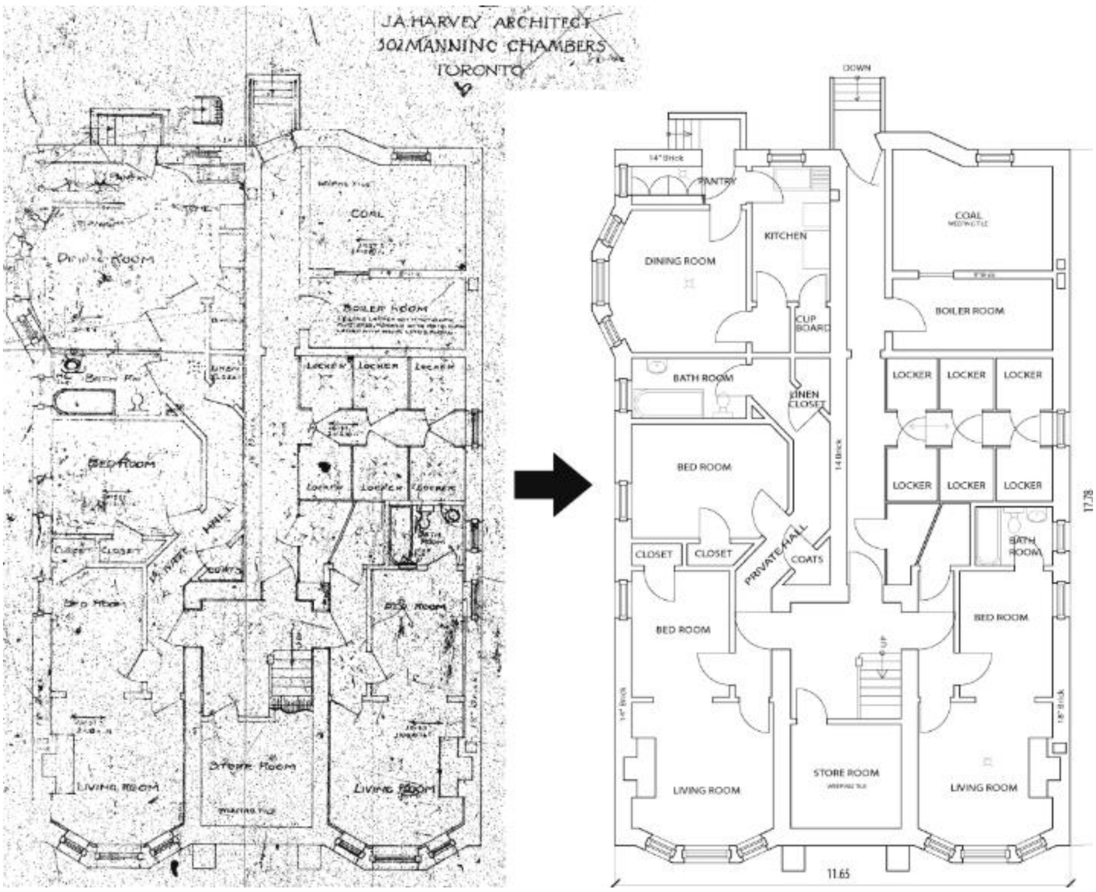

Figure 1 – Transition from hand-modelling to computer-modelling [3]

1.1.2 Building Information Modelling (BIM)

Although CAD software has its benefits, larger project scope and more complicated building design demanded for the creation of more robust modelling software. BIM, also known as “Building Information Modelling” is a popular and necessary approach used by AECs worldwide. BIM software creates 3D models of digital systems from building components that contain “living data”. Compared to drafting with lines and arcs in CAD, BIM models are built digitally as a database in the BIM software. Therefore, instead of having an archive of separate drawing schedules, cut sheets, and specifications, the relevant information is integrated into the intelligent object in the BIM model [4]. Consequently, elements in the model are able to adapt and change as designs are being modified. Revit (Figure 2) is a popular BIM software used by AECs in the building industry. It will be used as a modelling platform and information database for this report.

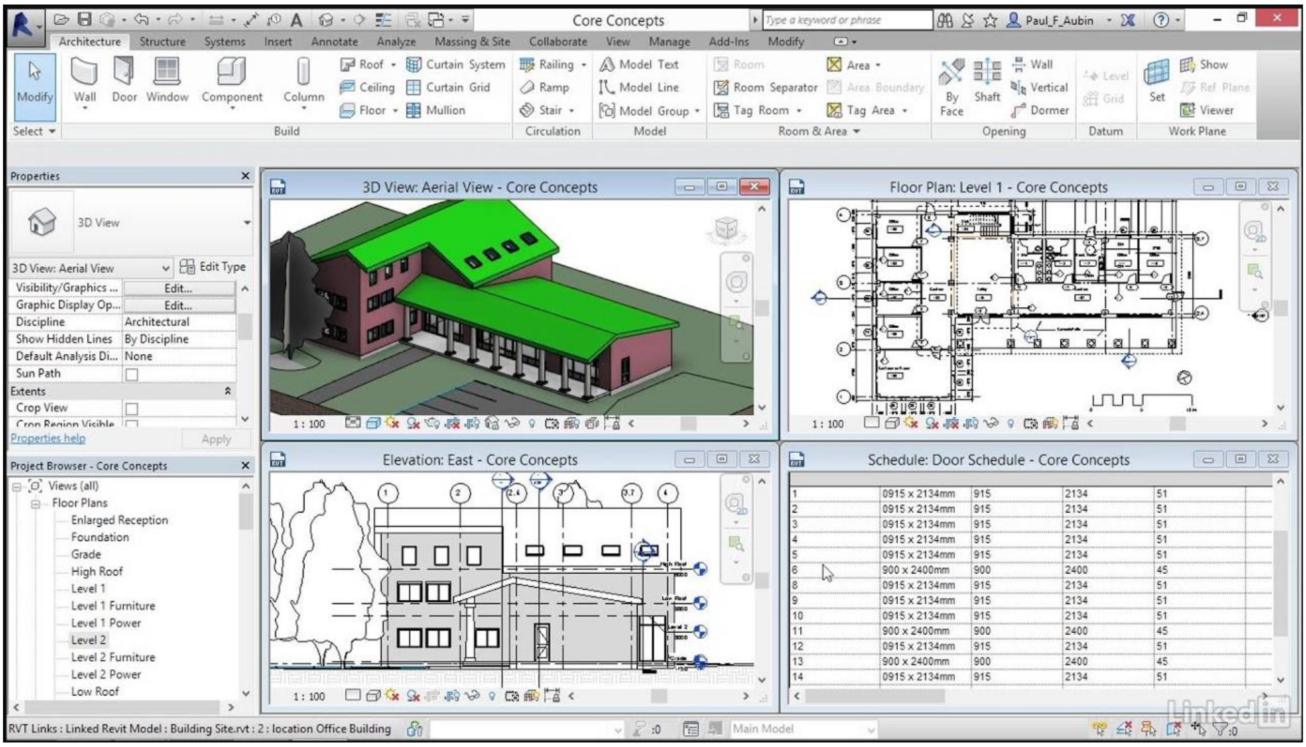

Figure 2 - 3D system in Revit [5]

1.1.3 Introduction to Dynamo

Being provided a digital archive of information, designers began building data-manipulating tools to automate conventional processes. One such tool that will be utilized in this report is a visual programming plug-in for Revit, called Dynamo. Dynamo is able to extract information stored in Revit components, sort and analyze data, and push changes back to the Revit model. This is accomplished by connecting node blocks that each performs a unique function. Users also have the choice to build, customize, and publish their own node packages with Python scripting. Dynamo's straightforward user interface and access to Revit building data makes it a powerful tool for building automation. Figure 3 below illustrates the connection of nodes in Dynamo. (Note: since Figure 3 depicts the overall layout of a Dynamo script, specific scripting details can be ignored in this diagram.)

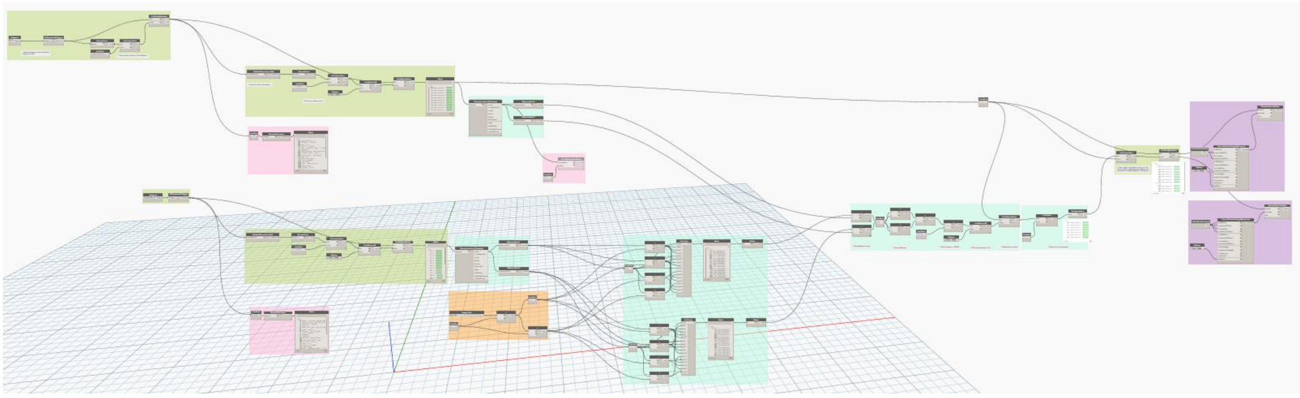

Figure 3 - Dynamo scripting example

Common applications of Dynamo scripting include but is not limited to quality assurance, creating views, renaming sheets, and printing large-scale projects. Re-assigning these repetitive and tedious tasks can potentially save companies millions of dollars as employees could have used the time to accomplish more innovative tasks [6].

1.2 Report Objectives

The main objective of this report is to show through an example, that Dynamo scripting can be integrated to automate quality assurance in building design. The report will tie in the National Canadian Fire Alarm Code and demonstrate Dynamo's ability to check for smoke detectors that do not follow Clause 8.2.3.10 of the Code.

2.0 Problem Definition

Clause 8.2.3.10 of the Code states:

"Spot type fire detectors shall not be located in a direct airflow or closer than 450mm from an air supply or exhaust outlet measured to the edge of the detector [7]." (Refer to Figure 4)

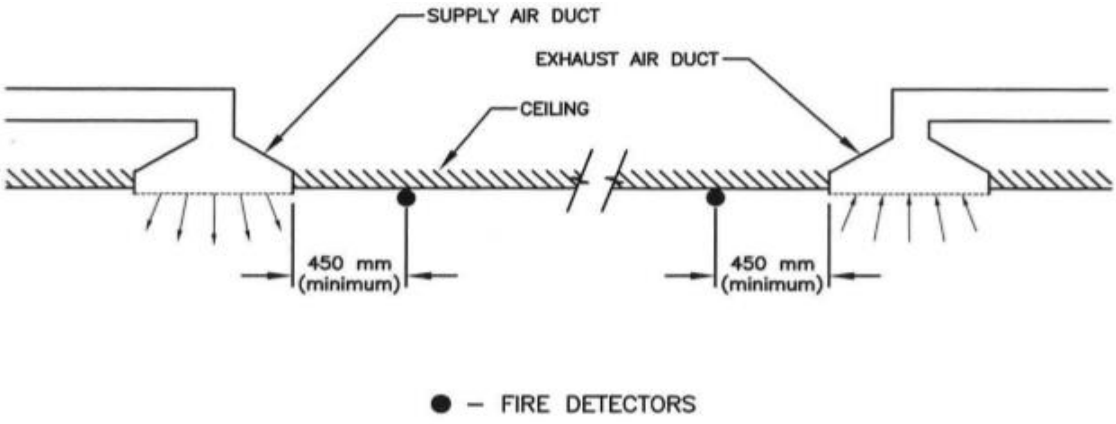

Figure 4 - Mounting of spot type smoke detectors [7]

In projects where there are hundreds of smoke detectors and air diffusers hosted to different levels in a building, it can be a tedious task for electrical and mechanical engineers to coordinate the location of equipment. It is estimated that on average, a company wastes 30% of its resources due to inefficient modelling solutions and use of building data [4]. A Dynamo script will be implemented to avoid having to manually check the distance between every smoke detector and air diffuser in the Revit model. The script will validate if each smoke detector is less than 450mm from an air diffuser and flag those that are not, so they can be shifted by the electrical engineer. The script can then be reused many times after modifications are made, to check and maintain elements in the model.

2.1 Definitions

1. National Canadian Fire Alarm Code – "describes the requirements for the design and installation of equipment and devices required by a fire alarm system." The Code recognizes that "a fire alarm system may have features that differ from the requirements if the features seek to enhance the life safety of occupants in the building and/or the protection of the building or property [8]."
2. Spot type smoke detector – fire detecting devices that use either the ionization or photoelectric principle [9]. Note: the function of spot type fire detectors is not a focus in this report and therefore will not be explored further.
3. Air diffuser – "an air distribution outlet, usually located in the ceiling and consisting of deflecting vanes discharging supply air or taking in return air [10]."
4. Surface mounted – the object is fixed directly on top of the surface it is mounted onto [11].
5. Recess mounted – the object if fitted inside the wall or mounting surface [11].

2.2 Dynamo Conventions

When working on Dynamo scripts in a collaborative environment, it is important to have standard practices for everyone to follow. This will make it easier for designers to revisit their script after several years, and for future members who want to expand upon existing projects understand the existing work. The script that will be showcased in this report will follow the standards described below.

The first standard refers to the method of referencing any external packages used in the script. In addition to the core Revit nodes that come with installing Dynamo, designers are able to download third party packages to access nodes with more diverse functions. In these cases, the name of the package should be written in “[ ]” alongside the name of the node. Refer to Figure 5 below.

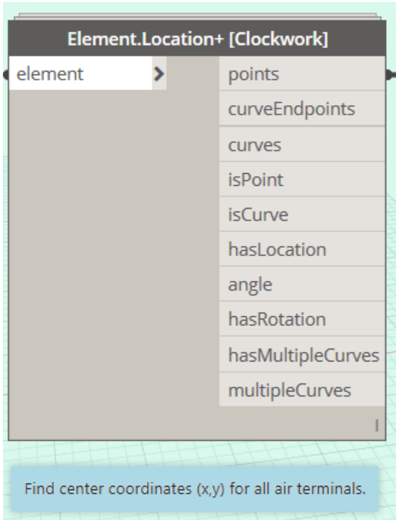

Figure 5 - Standard for referencing third party packages

The second standard for Dynamo scripting is arranging nodes that perform a similar function in groups. Each group should have a name that describes the main function it serves in the script as well as being associated with their respective color. This is shown in Table 1.

Table 1 - Color grouping of nodes used in Dynamo

Color	**Function**	**Color**	**Function**
Pink	Debugging	Blue	Data processing (general)
Purple	Override Changes	Orange	User input
Green	Filter/ group/ mask data		

Last but not least, user notes will be added to clarify or expand on sections of the script. (Refer to the user note under the node in Figure 5.)

2.3 Assumptions

1. It is assumed that a distance of 450mm is measured from the outer edge of a smoke detector to the outer edge of an air diffuser. Note: Figure 4 portrays smoke detectors as points. Realistically, smoke detectors have a 175mm radii geometry. Because the script extracts center points of elements, 625mm (450mm+175mm) will be used to verify the distance between smoke detectors and air diffusers.
2. Air diffusers are recess mounted on the ceiling whereas spot type smoke detectors are surface mounted on the ceiling.
3. It is assumed that the reference plane an element is hosted to matches the height of the ceiling at each level. Therefore, vertical distances between elements that are hosted at the same reference plane and the floor have the same z-coordinates.
4. It is assumed that the placement of all other elements (such as ducts and ceiling grids) has been previously verified in the Revit model. Therefore, clash detection will only be analyzed between smoke detectors and air diffusers.
5. The script will only be run on level 1 of the sample model to narrow the scope for testing purposes. It is assumed that if the script runs successfully and all other levels have similar geometries to level 1, then the code will perform consistently for other levels.

2.4 Constraints

1. After a user runs the Dynamo script, all smoke detectors that do not meet the Fire Code will turn red in the active view in Revit.
2. After a user has shifted the placement of smoke detectors that violates the Code and runs the script a second time, the active view will be refreshed to reflect the new changes.

2.5 Criteria

1. The common return diffusers used in this report (C1-C3) comes in 3 sizes: 600x600mm, 600x300mm, and 300x300mm. To simplify checking smoke detectors with 3 different perimeters, the largest diffuser size (600x600mm) will be used as a boundary for all return diffusers. Therefore, a tolerance of +150mm length and +150mm width will be used for 300x300mm C1 diffusers. And a tolerance of +150mm width will be used for 600x300mm

C2 diffusers. (Refer to Figure 6a and Figure 6b below) Using the largest diffuser size ensures that all smoke detectors are at least 450mm away from C1 and C2 diffusers, in accordance with the Code.

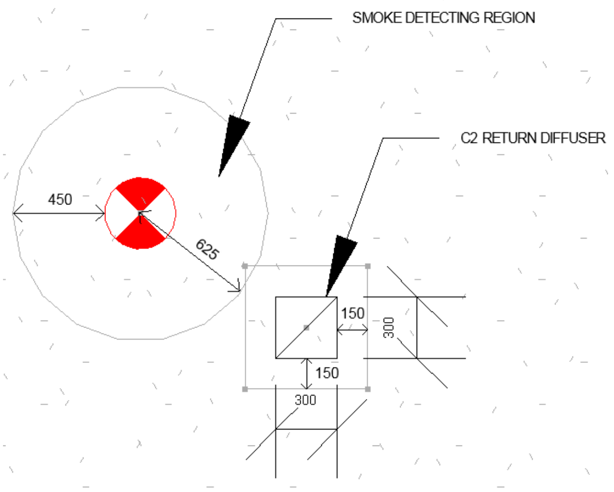

Figure 6a - C1 return diffuser

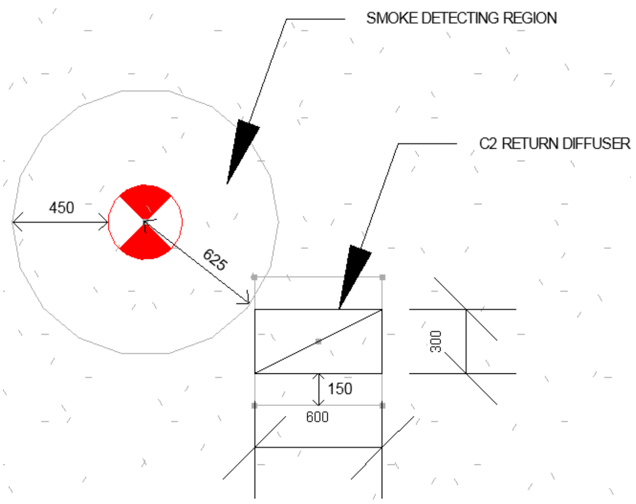

Figure 6b - C2 return diffuser

3.0 Technical Progress

3.1 Approach - Pseudo Code

Before any scripting was performed, pseudo code was written to find clashes between smoke detectors and air diffusers. The steps are shown below:

1. Extract air diffuser information.
2. Extract smoke detector information.
3. Filter for elements hosted at the same height (z-axis) on each floor.
4. Get the center point (x, y) of each air diffuser and smoke detector.
5. Create a 600x600mm square boundary around the center of each air diffuser.
6. Create a 625mm radii circular boundary (taking into account the 175mm radii of the smoke detector) around the center of each smoke detector.
7. Check if the boundaries intersect.
8. If the boundaries do intersect, change the smoke detector red.
9. If the boundaries do not intersect, do not make changes to the smoke detector or change it back to its original color.

3.2 Pseudo Code (Steps 1-4)

Figures 7, 8, and 9 show the script that matches steps 1-4 in the pseudo code.

Note: the figures only show the code for smoke detectors since the logic for air diffusers are very similar.

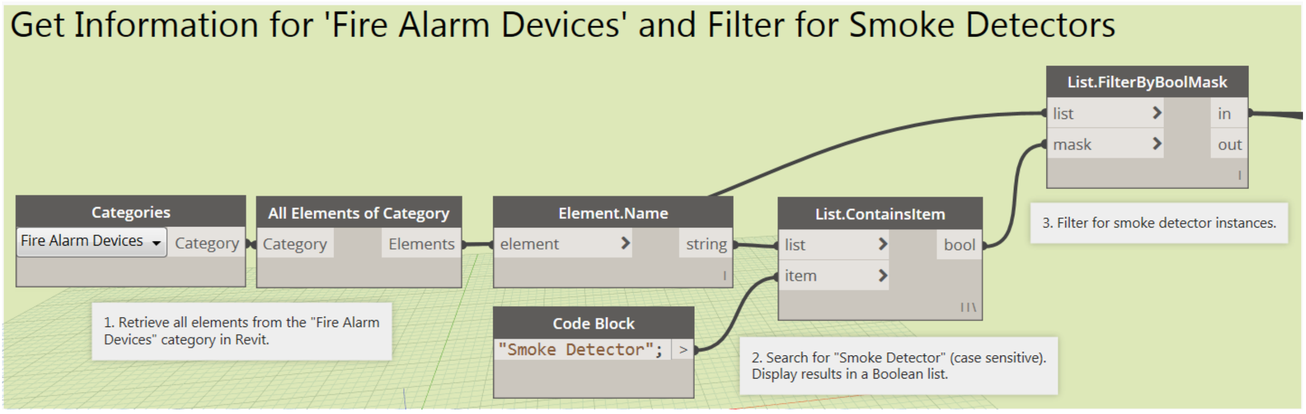

Figure 7 - Script that retrieves smoke detector information from the Revit model

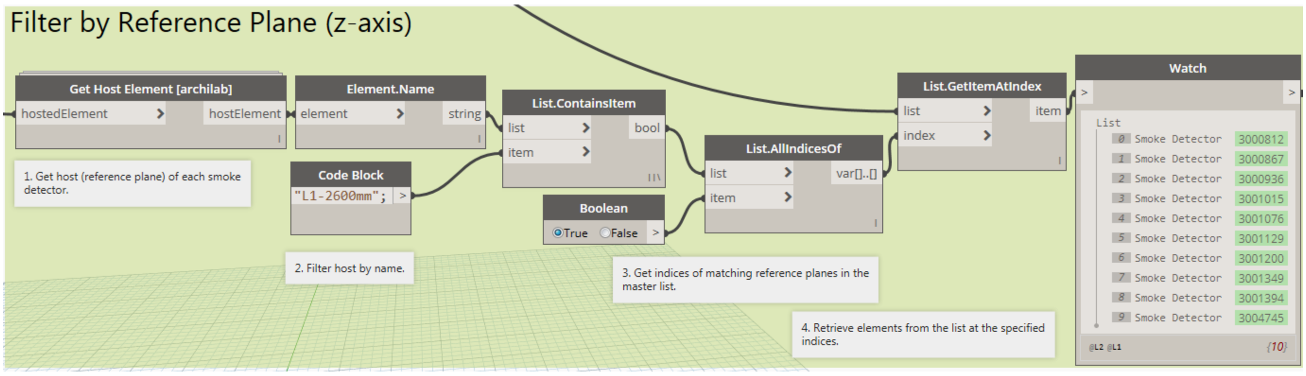

Figure 8 - Script that filters smoke detector instances by reference plane

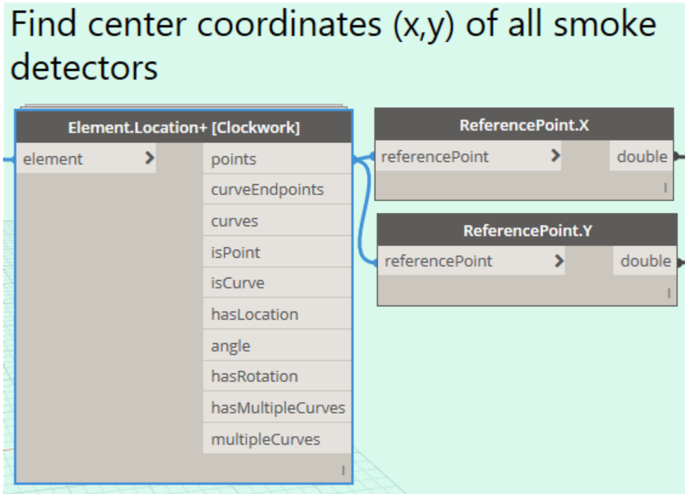

Figure 9 - Script that retrieves x and y center points of smoke detectors

3.3 Challenges and Solutions- Pseudo Code (Steps 5-9)

As the script was being developed, it became apparent that there were two major challenges to the script. The first challenge is finding a method to check for clash detection between the smoke detector and air diffusers. This refers to steps 5, 6, and 7 in the pseudo code. Elements can either align directly on top of one another, or the air diffuser can intersect the 450mm range of a smoke detector. (Refer to Figure 10)

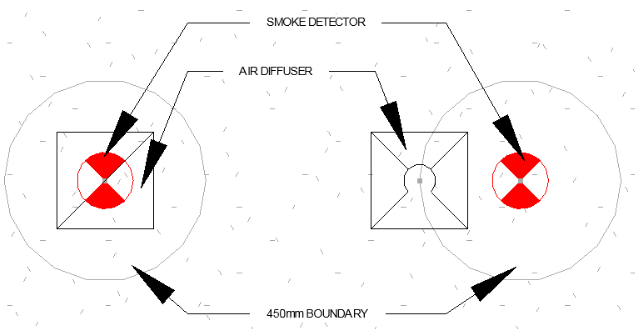

Figure 10 - Clash detection

The second challenge of this script is finding a user-friendly way to refresh Revit after the program has run. This refers to steps 8 and 9 in the pseudo code. This is important after a user has made modifications and would like the information in Revit to reflect the latest updates. This will also allow for continuous testing in the Revit model without interruptions.

3.3.1 Clash Detection - Script 1

To approach the first challenge of clash detection, two iteration of scripts were developed. The first script (Script 1) draws a geometrical boundary around each diffuser and smoke detector. The script then checks whether the geometries intersect. Figure 11 shows the script for drawing the boundary lines around air diffusers. Figure 12 shows the script for drawing the boundary lines around smoke detectors.

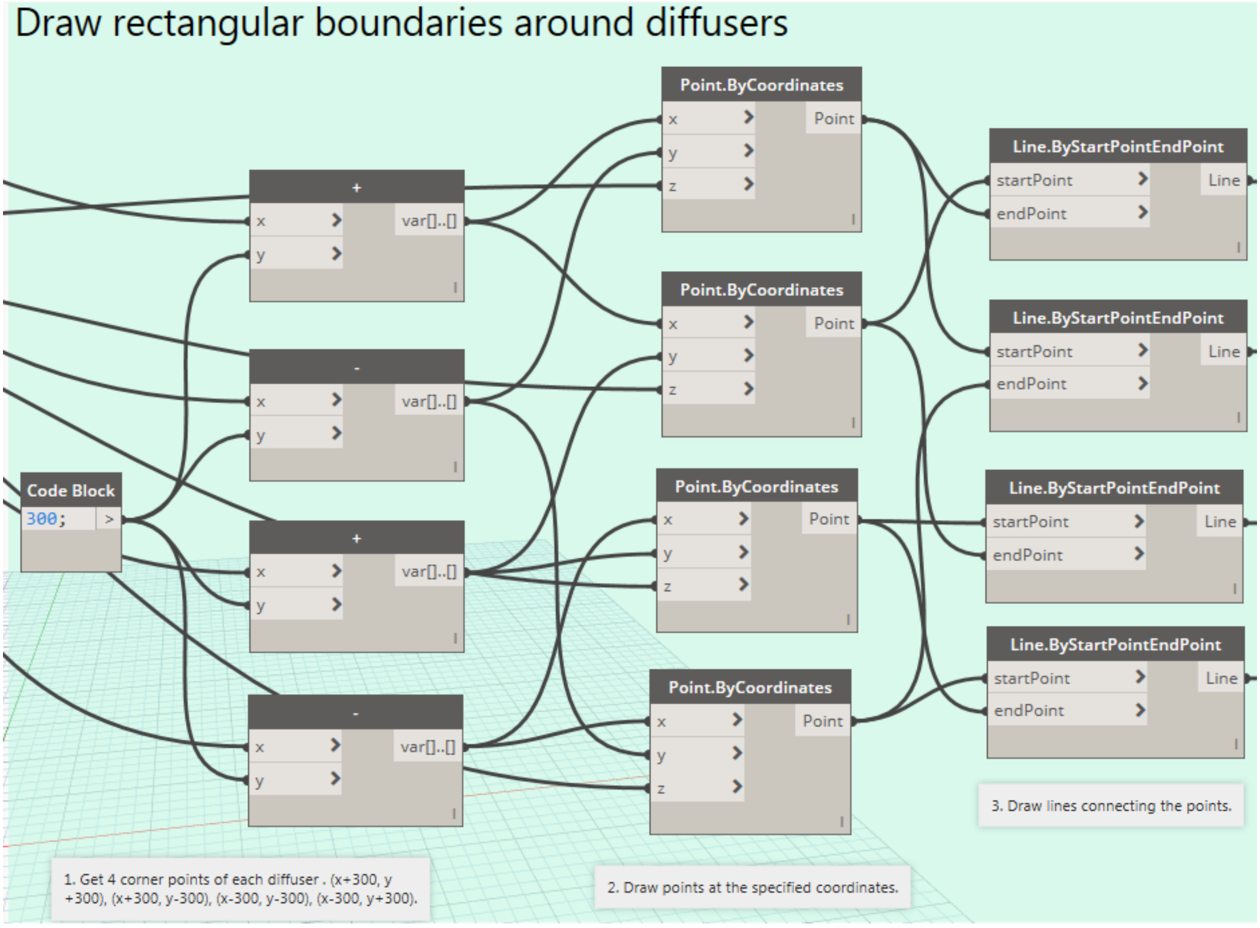

Figure 11 - Script that creates 600x600mm boundaries around air diffusers

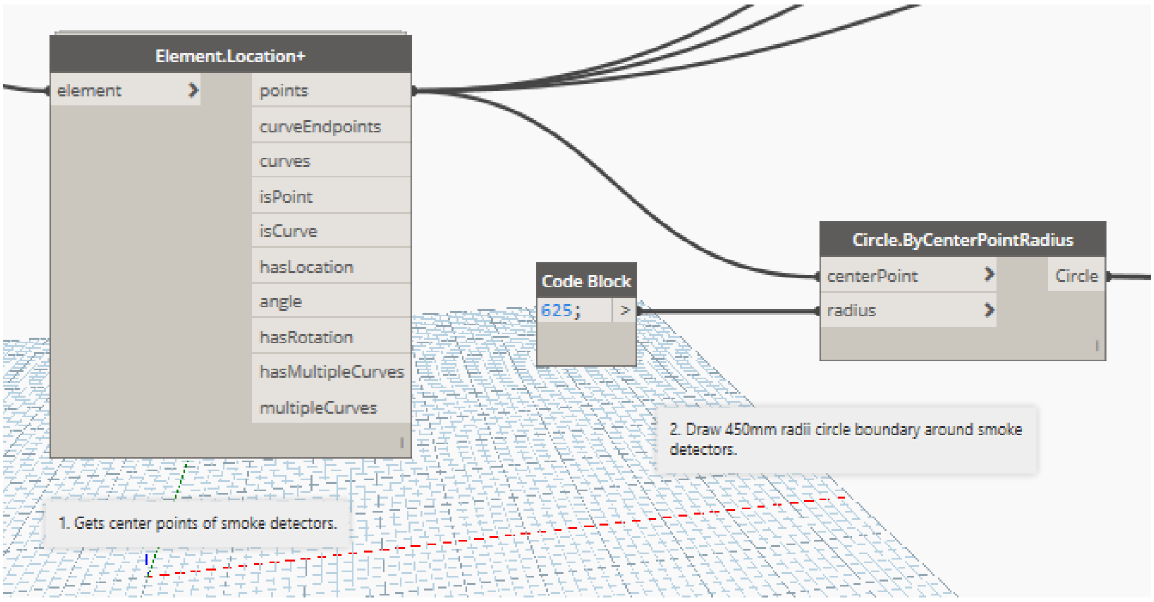

Figure 12 - Script that creates 625mm radii boundaries around smoke detectors

Figure 13 checks for intersecting geometries and Figure 14 illustrates the result after running Script 1 in Revit.

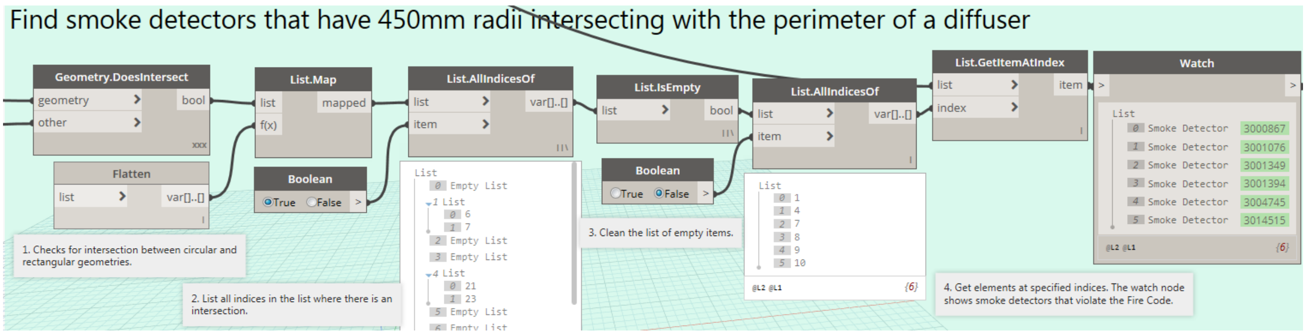

Figure 13 - Script that checks for intersecting boundaries

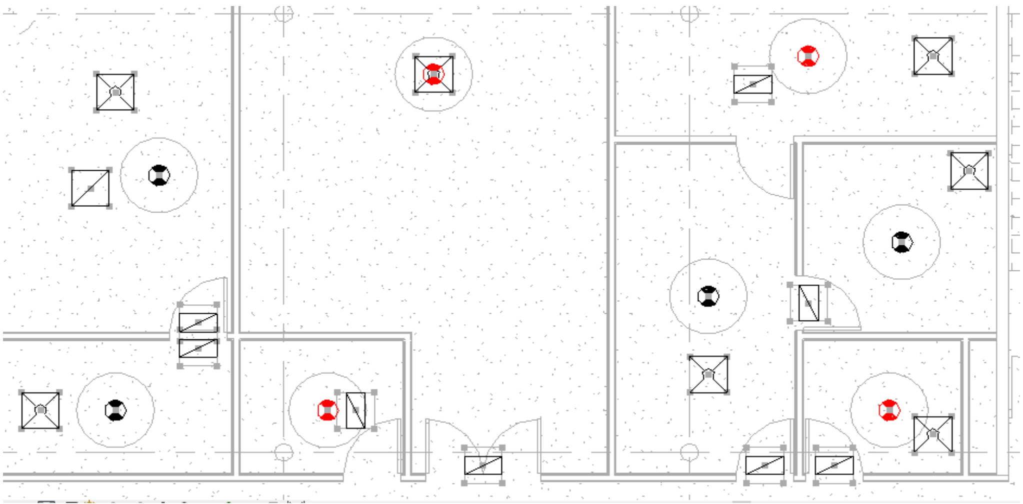

Figure 14 - View in Revit after Script 1 was run

As can be seen in Figure 14, the boundary lines used for clash detection appear in the active view in Revit. For users who only want to see smoke detectors and air diffusers on the sheet, these lines can be confusing. One possible solution is to hide these lines by making them transparent. However, because the lines are geometrical shapes created by Dynamo, they are not considered elements in Revit and therefore their properties cannot be changed. As a result, a new approach is needed to find clash detection between smoke detectors and air diffusers without drawing and checking for intersecting boundary lines.

3.3.2 Clash Detection - Script 2

Script 2 focuses on testing a handful of points along the perimeter of the diffuser that are close enough together, so if an intersection with a smoke detector does occur, at least one of the points lies within the 450mm radii boundary. The points are represented with variables in Figure 15. The x and y coordinates of the numbered points are shown in Table 2.

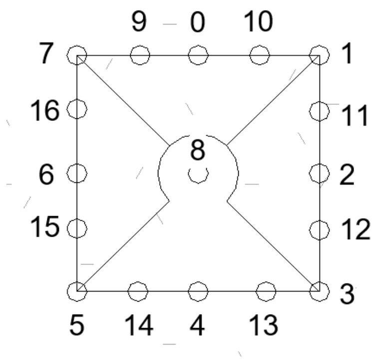

Figure 15 - Points 0-16 along the perimeter of an air diffuser

Table 2 - x and y coordinates of points 0-16

Value	**x**	**y**	**Value**	**x**	**y**
0	x	y+300	**9**	x-150	y+300
1	x+300	y+300	**10**	x+150	y+300
2	x+300	y	**11**	x+300	y+150
3	x+300	y-300	**12**	x+300	y-150
4	x	y-300	**13**	x+150	y-300
5	x-300	y-300	**14**	x-150	y-300
6	x-300	y	**15**	x-300	y-150
7	x-300	y+300	**16**	x-300	y+150
8	x	y			

Note: Manipulating the x and y coordinates of the diffusers to get new points in Table 2 is similar in terms of scripting as Figure 11 above, and therefore will not be illustrated in this report.

Unlike Script 1, Script 2 does not involve the creation of geometrical lines. Therefore the "Geometry.DoesIntersect" node used in Figure 13 will not be applicable. Instead, the Pythagorean Theorem is used to cross check whether a point on a diffuser lies within a 625mm radii of the center point of a smoke detector.

$$(x_a - x_s)^2 + (y_a - y_s)^2 = d^2 \qquad (1)$$

Where:

x_a = x coordinate of an air diffuser
y_a =y coordinate of an air diffuser
x_s = x coordinate of a smoke detector
y_s = y coordinate of a smoke detector
d = distance of separation

If $d^2 < 625^2$, the point along the perimeter of the diffuser lies within the smoke detector boundary. This rule is used to check every distance between an air diffuser and smoke detector. This logic is used in the script, as shown in Figure 16 and Figure 17 below.

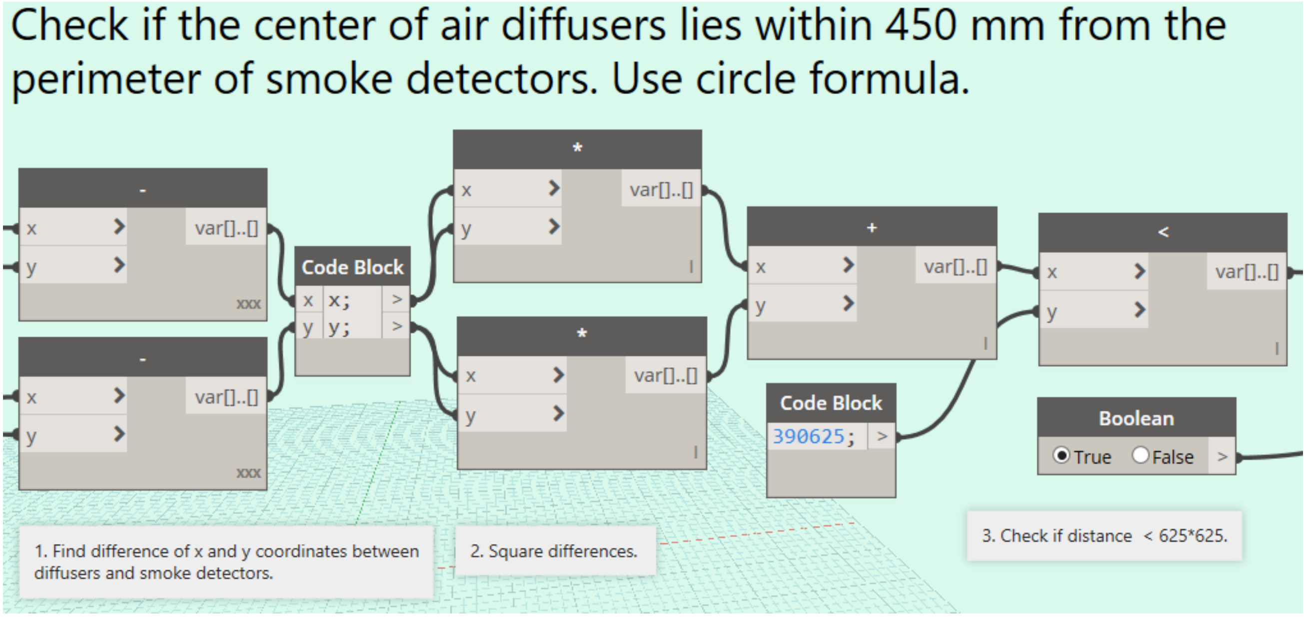

Figure 16 - Script that checks if points 0-16 on an air diffuser lies within the 450 mm boundary of smoke detectors

Note: filtering the list of smoke detectors (so only a set of unique items are obtained) is a necessary step because of the overlapping results after cross-checking the distance between every smoke detector and air diffuser.

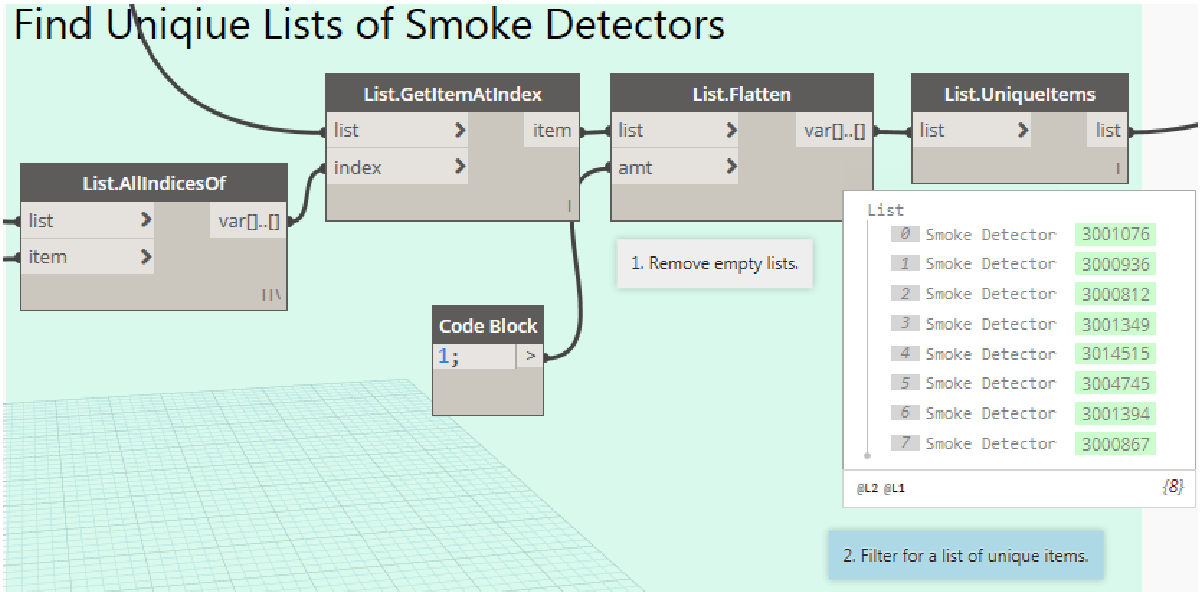

Figure 17 - Filter list for a set of unique elements

The result from Script 2 is shown in Figure 18 below, this time without any boundary lines around the elements.

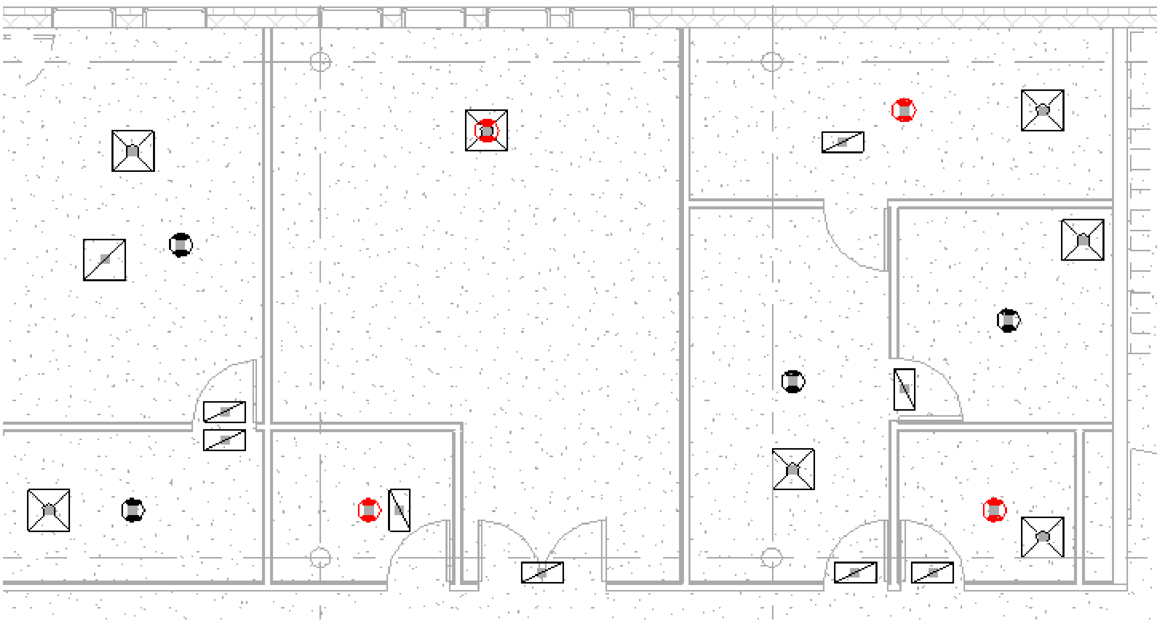

Figure 18 - View in Revit after Script 2 was run

3.3.3 Refresh Revit Script

The second challenge referred to earlier is to display the most recent smoke detector information in the active view after a user has modified its placement. The solution used in this report is to implement a Boolean mask that takes the original list of smoke detectors and filters it by the smoke detectors that violate the Code. Smoke detectors in the original list that match, have the color override as red, while smoke detectors that are more than 450mm from an air diffuser, have their color reverted back to black. Any changes will take effect immediately once the user has pressed "Run" in Dynamo. This script can be run an infinite number of times and will always update the model to reflect the newest information. The script is shown below in Figure 19a and Figure 19b.

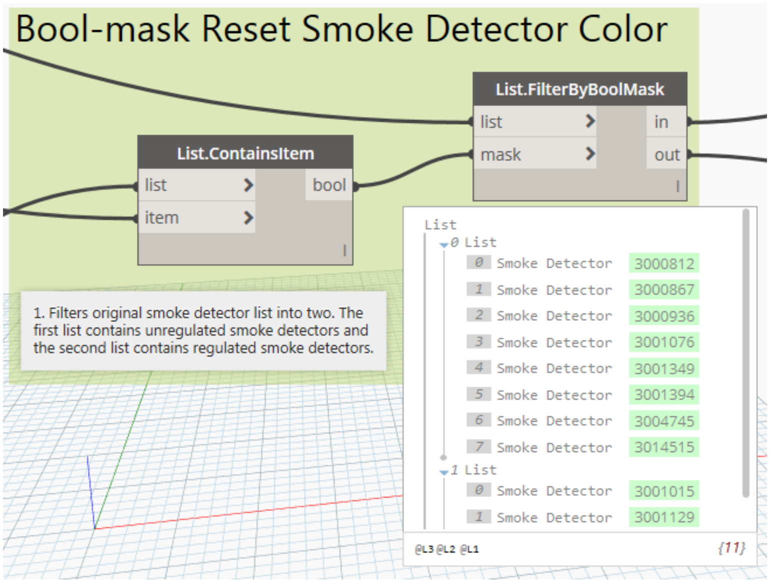

Figure 19a - Color override Script

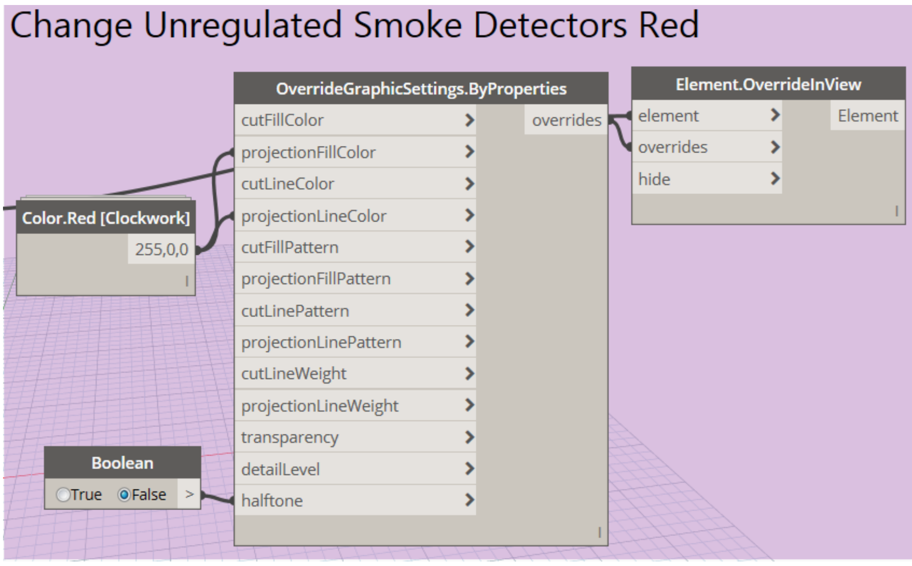

Figure 19b - Color override Script

Note: The script for changing the color of smoke detectors to black is similar to the script in Figure 19b and will not be shown in the report.

Figure 20 and Figure 21 show a comparison of the Revit model before and after the user has made changes and refreshed.

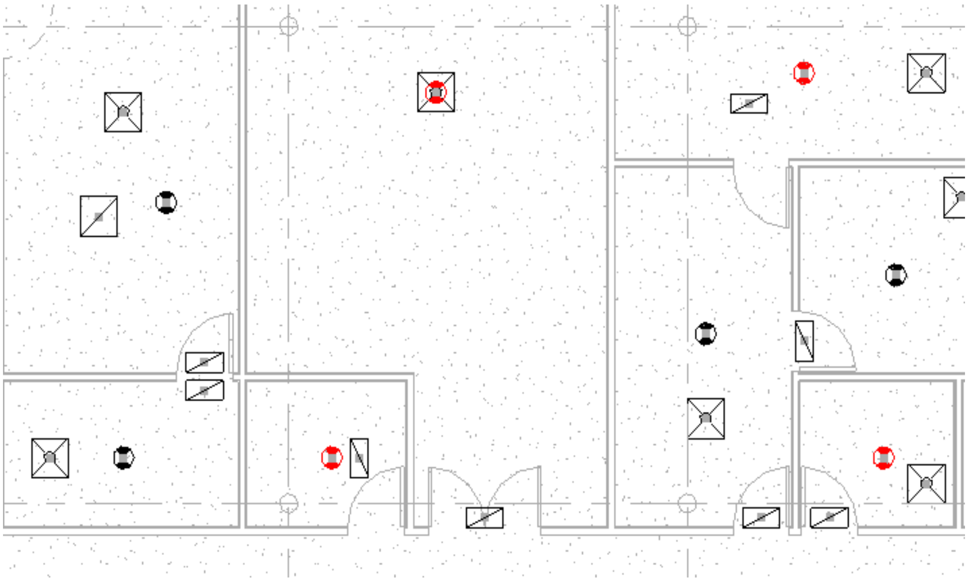

Figure 20 - Revit model after one run of Script 2

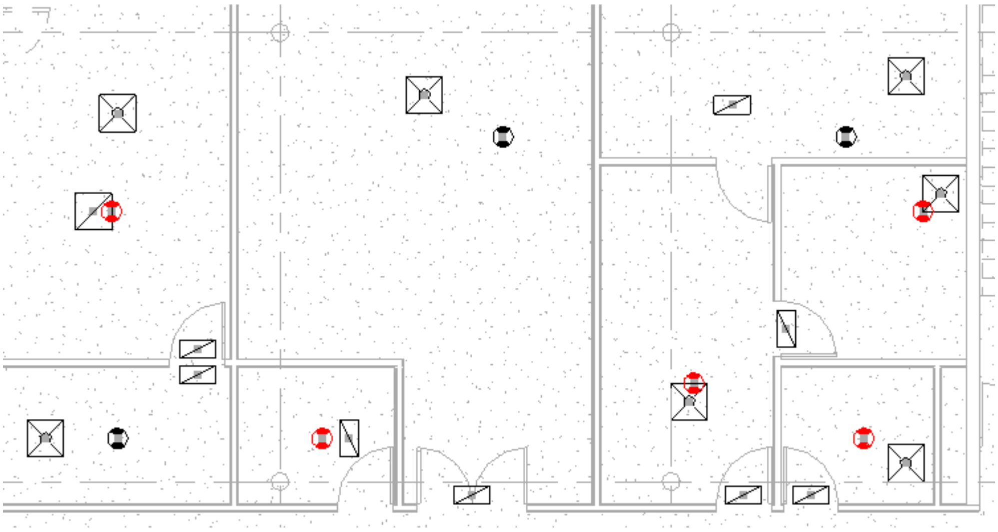

Figure 21 - Revit model after smoke detectors have been shifted and Script 2 was re-run

3.4 Progress to Date

The Dynamo script was successful in meeting both criteria stated in the early portions of the report. After the user has pressed “Run”, smoke detectors with a distance less than or equal to 450mm from an air diffuser will turn red (Figure 18). It was also demonstrated in Figures 20 and 21 that by pressing “Run” a second time, the active view will be refreshed to reflect the newest changes.

3.5 Future Improvements

While Script 2 successfully addressed the problem stated in the report, there is room for improvement. For instance, it was noted in the constraints section of the report that a 150mm tolerance was used to calculate the distance between smoke detectors and return air diffusers. This simplification was used to accommodate for varying diffuser sizes. However, it is possible to make the script more accurate by having the user input the dimensions of the return diffuser they are using in the model (with an integer slider) and have Dynamo calculate perimeter points based on user input.

The next iteration of the script can be made more efficient by incorporating test cases rather than checking all 17 points along the perimeter of the air diffuser (Figure 15). For example, if the distance between the centers of a smoke detector and an air diffuser was less than 450mm, the smoke detector would violate the Code without having to check all the other points along the diffuser. This saves space, memory, and computer run-time, which are valuable resources in scripts for large-scale projects.

Furthermore, future scripts could also check if smoke detectors placed on walls are regulated by the Code. In these cases, the Code states: “spot type fire detectors may be installed on the wall between 100mm and 300mm from the ceilings”. Dynamo could check the ceiling height of the room a wall-type smoke detector is placed in and make sure the vertical distance between the smoke detector and the ceiling is in the range specified by the Code.

The script can further be improved by introducing algorithms and Python code to automate smoke detector placement. Instead of simply flagging unregulated smoke detectors, Dynamo can rearrange the smoke detectors in a new position without the need to change its color and have the user modify its location. This method will require additional rules if smoke detectors were to be automatically arranged. For example, the user should specify in the Dynamo script that smoke detectors should not collide with ceiling grids or light fixtures, etc. An alternative that is more user-controlled can be to have Dynamo provide directions on where to move a smoke detector and by how much, instead of automatically making changes to the Revit model.

4.0 Conclusions

4.1 Conclusion

The objectives of the report were to create a script that can check the placement of smoke detectors in a Revit model, so they meet the National Canadian Fire Alarm Code, and demonstrate Dynamo's powerful features that can be extended and customized by automation.

The first objective is met because the script is able to check if every smoke detector is at least 450mm away from the perimeter of an air diffuser and flag smoke detectors that do not meet this criteria. By automating this process, users do not need to manually check the distance between every smoke detector and air diffuser in the model.

The second objective is also met since every smoke detector that is less than 450mm from an air diffuser is successfully flagged in Revit. Furthermore, the script is able to be re-used to reflect continuous changes made in the model.

4.2 Future of BIM – Design Automation

As programming becomes more popular in the building industry, scripts will focus on building design, rather than automating the process of checking and maintaining Revit models. Given a set of rules, computers are already generating hundreds of possible design iterations for the user to choose from. This is known as "generative design." The rules fed into the computer include building codes that must be followed, but it can also include user input with slide bars. Generative design was illustrated by Nate, a designer from Holland who used Grasshopper (a programming software similar to Dynamo) and algorithms to find the optimal location for real estate. Some of the criteria included: price, retail, view of the ocean, etc. [6] (Figure 22) Furthermore, the program was able to layout floor plans that optimized bedroom space, based on layout requirements per space.

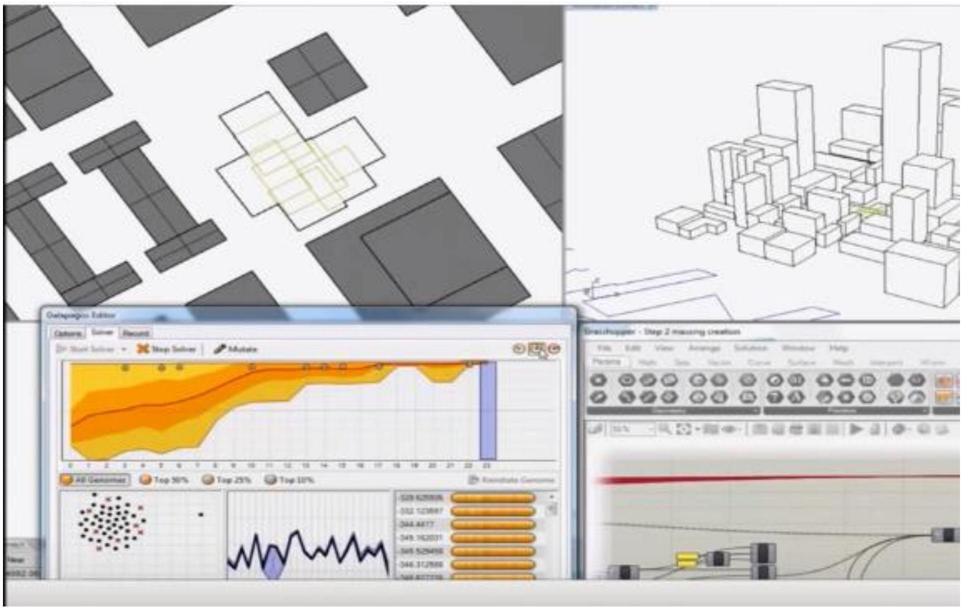

Figure 22 – Dynamo generative design for real estate location [6]

Rather than manually updating information, the computer is able to create optimal solutions based on "rules" fed to the computer. Equipped with more knowledge of design possibilities, building modelling can become more efficient and creative in the future.

5.0 References

1. Kensek, K. and Noble, D. (2014). *Building Information Modeling: BIM in Current and Future Practice*. John Wiley & Sons, Inc., Hoboken, New Jersey.
2. En.wikiversity.org. (2019). *Computer-aided design/History, Present and Future - Wikiversity*. [online] Available at: https://en.wikiversity.org/wiki/Computer-aided_design/History,_Present_and_Future [Accessed 26 Apr. 2019].
3. JRX (n.d.). *Professional AutoCAD Drafting, Engineering & Technical Graphics*. [image] Available at: https://jrgraphix.net/CAD/ [Accessed 26 Apr. 2019].
4. MEPCore, Inc (2010). *The Power of BIM - An Introduction to Building Information Modeling*. [video] Available at: https://www.youtube.com/watch?v=EajEb2Busfs [Accessed 26 Apr. 2019].
5. LinkedIn Learning (2016). *Introducing building information modeling (BIM): Revit Architecture 2016 Essential Training*. [image] Available at: https://www.youtube.com/watch?v=ErGFP-2p6K0 [Accessed 26 Apr. 2019].
6. Autodesk University (2016). *The Future of BIM Will Not Be BIM—and It's Coming Faster than You Think*. [video] Available at: https://www.youtube.com/watch?v=xq6yKyauu-o [Accessed 26 Apr. 2019].
7. CAN/ULC-S524:2014-AMD1. (2014). ULC Standards, p.29.
8. Orderline.com. (2019). *CAN/ULC-S524-14 Standard for the Installation of Fire Alarm Systems Sixth Edition*. [online] Available at: https://www.orderline.com/can-ulc-s524-14-amd1-standard-for-the-installation-of-fire-alarm-systems-sixth-edition-including-amendment-1-11 [Accessed 26 Apr. 2019].
9. Community.nfpa.org. (2019). *Spot Type Smoke Detector | NFPA Xchange*. [online] Available at: https://community.nfpa.org/thread/2008 [Accessed 26 Apr. 2019].
10. TheFreeDictionary.com. (2019). *air diffuser*. [online] Available at: https://encyclopedia2.thefreedictionary.com/air+diffuser [Accessed 26 Apr. 2019].
11. Warren, J. (2019). *What's the difference between a surface, flush, and recessed…*. [online] 1800Doorbell. Available at: https://www.1800doorbell.com/faq/whats-difference-surface-flush-recessed-mounted-wired-doorbell-button [Accessed 26 Apr. 2019].