SPACE PROGRAMMING REQUIREMENTS REPRESENTATION, ANALYSIS AND VISUALIZATION AT A LARGE SCALE ARCHITECTURAL FIRM

by

Helina Merkeb Gebru

B.Eng., Addis Ababa University, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Civil Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2018

© Helina Merkeb Gebru, 2018
Committee Members

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Space programming requirements representation, analysis and visualization at a large scale architectural firm.

Submitted by: Helina Merkeb Gebru in partial fulfillment of the requirements for the degree of Master of Applied Science in Civil Engineering.

Examining Committee:

Dr. Sheryl Staub-French, Civil Engineering- Project and Construction Management  
Supervisor

Terje Haukaas, Civil Engineering- Structural Engineering  
Additional Examiner
Abstract

Space programming is a primary task during the schematic design process, to produce a geometric configuration of a space layout that is in accordance with the project's requirements. By nature, space programming is an iterative process that evolves according to the client’s requirements. A critical challenge of space programming is the limitation in the link between the client’s requirements and design tools. The rigorous process of analyzing, structuring and extracting meaningful information often leads to requirements being overlooked or important requirements failing to be satisfied. Failure to meet the client’s space program requirements, could possibly lead to decline in the performance of the building, cost increase, client dissatisfaction and penalty fines charged by the client which are usually clearly stated in design contracts.

This study adopted an observation-based empirical research approach to investigate the current practices and challenges of space program requirements data management, and design workflow at a large scale international architectural/engineering firm. Following the case study and recording challenges, I developed a smart Microsoft Excel® template to structure and parse a client’s space programming requirements data. This was essential to extract significant information such as the name of the rooms that have a proximity relationship requirement. This data was used to develop a dashboard to visualize space programming information and to validate the compliance of a building project’s space programming requirements in conjunction with a visual computational tool through a visual floor plan overlay.
The developments were made to help designers extract space programming requirements in an automated manner and improve the iterative design process of space programming by automating visualizations to assess the compliance of space programs.
Lay Summary

During the space programming process there is a gap between client’s requirements documents and design tools that forces a designer to manually analyze requirements and use it to assess the compliance of a space program in a highly iterative, manual and tedious process.

This research will discuss the method that was used to develop a smart excel template to structure and parse the clients space programming requirements data. This was essential to extract significant information such as the name of the rooms that have a proximity relationship requirement. Next, it will discuss how this data was used to develop an interactive space program requirements data visualization dashboard. Finally, it discusses the development of a computational tool script used to validate the compliance of a building’s space program requirements through a floorplan overlay.
Preface

This study was conducted under the direct supervision of Dr. Sheryl Staub-French while working with Stantec Consultancy Ltd. The author is responsible for the data collected for this research and was provided by the architectural/engineering design firm Stantec Consultancy Ltd. All developments, figures, and images in this document were created by the author unless sited to the sources.
Table of Contents

Abstract .......................................................................................................................... iii
Lay Summary .................................................................................................................. v
Preface ........................................................................................................................... vi
Table of Contents ......................................................................................................... vii
List of Tables ................................................................................................................. x
List of Figures ................................................................................................................. xi
List of Abbreviations ..................................................................................................... xv
Acknowledgements ..................................................................................................... xvi

Chapter 1: Thesis Overview ..................................................................................... 1
  1.1 Introduction ........................................................................................................... 1
  1.2 Research Objectives ........................................................................................... 4
  1.3 Research Methodology ...................................................................................... 5
    1.3.1 Case Studies ................................................................................................. 8
    1.3.2 Research Approach ..................................................................................... 14
    1.3.3 Requirements Data Management .............................................................. 16
    1.3.4 Structuring and Visualizing Space Programming Requirements .............. 17
    1.3.5 Validation through Expert Reviews ........................................................... 17
  1.4 Thesis Overview .................................................................................................. 17

Chapter 2: Literature Review ................................................................................... 19
  2.1 Space Programming Requirements Representation ........................................... 19
  2.2 Visualization of Space Programming Requirements ........................................ 23
2.2.1 Space Programming Visualization Conclusion .......................................................... 28

2.3 Literature Review Conclusion ....................................................................................... 30

Chapter 3: Current Space Programming Process, Requirements and Data ....................... 31

3.1 Introduction ...................................................................................................................... 31

3.2 Tools Used for Requirements Representation, Design and Visualization ....................... 31

3.3 Current Practices: Space Programming Analysis and Representation ......................... 37

3.3.1 Design Requirements Analysis ................................................................................... 37

3.3.2 Design Requirements Representation ........................................................................ 40

3.3.3 Limitations of Current Space Programming Data Representation ............................ 50

3.3.4 Development of dRofus Database for Project B ....................................................... 56

3.4 Current Practices: Space Programming Process and Compliance Assessment ............. 62

3.4.1 Limitations of the Current Space Programming Compliance Assessment ................. 72

3.5 Conclusion ...................................................................................................................... 73

Chapter 4: A Computational Approach for the Visualization of Space Program Requirements ......................................................................................................................... 74

4.1 Approach ......................................................................................................................... 74

4.2 Development of Smart Excel Template to Parse Client’s Space Programming Requirements Data .................................................................................................................................. 75

4.3 Development of Space Program Requirements Dashboard Visualization ...................... 83

4.4 Development of Space Program Requirements Floor Plan Visualization ..................... 88

4.4.1 Daylight ...................................................................................................................... 89

4.4.2 Acoustics .................................................................................................................... 92

4.4.3 Location ..................................................................................................................... 98
4.4.4 Adjacency ........................................................................................................ 101
4.4.5 Access ........................................................................................................... 103
4.5 Challenges ...................................................................................................... 109
4.6 Suggestion ................................................................................................... 109
4.7 Evaluation of Space Program Visualizations ........................................ 109
4.7.1 Interview Results .................................................................................... 112

Chapter 5: Conclusions and Future Work ................................................. 113

5.1 Summary ........................................................................................................ 113
5.2 Limitations and Future Work ................................................................. 114

Bibliography ....................................................................................................... 116
List of Tables

Table 1-1  Projects description and data analyzed............................................................... 10
Table 1-2 Research activities carried out on each project .................................................. 14
Table 3-1 Challenges recorded from current practices of space programming and the methods
adopted to address them........................................................................................................... 73
Table 4-1 Words used in the formula to flag relationships between rooms ......................... 86
Table 4-2 Space program requirements data parsing efforts accuracy statistics .................... 88
List of Figures

Figure 1.1 Research roadmap …………………………………………………………………………………….. 6
Figure 1.2 Visual roadmap of space program requirements compliance assessment approach ……… 7
Figure 1.3 Project A rendered view …………………………………………………………………………… 11
Figure 1.4 Project B rendered view (Alberta Infrastructure, 2017)…………………………………… 12
Figure 1.5 Project D rendered view (SandraJansenMLA, 2017)…………………………………… 13
Figure 1.6 Empirical cycle according to A.D. de Groot (Wikipedia, n.d.)……………… 16
Figure 2.1 Shifting away from the goal (Kiviniemi, 2005) ………………………………………………… 23
Figure 2.2 Sample of layouts generated in the space layout panel (Yi, 2016) ……………………….. 25
Figure 2.3 Generated space plan layout options with design score for reference (Das & Haymaker, 2016)…………………………………………………………………………………………………… 26
Figure 2.4 Various programmatic elements for a hospital and the relationship with others. there are three different levels of importance indicated by tones (Boon et al., 2015).……………… 27
Figure 2.5 Most fit iteration generated using Galapagos script (Boon et al., 2015)……………….. 27
Figure 2.6 Process of generated multi-agent layout to optimized grid system (Guo & Li, 2016) 28
Figure 3.1 Sample dRofus® database (dRofus, 2018)…………………………………………………… 33
Figure 3.2 Sample Autodesk Revit® floor plan and 3D model rendering …………………………….. 34
Figure 3.3 Sample Dynamo® script (Autodesk, 2018) ……………………………………………………… 35
Figure 3.4 Sample PowerBI® dashboard (PowerBI, 2018) ………………………………………………… 36
Figure 3.5 Functional program of a department in Project A …………………………………………… 40
Figure 3.6 dRofus® plug-in within Autodesk Revit®……………………………………………………… 41
Figure 3.7 Current dRofus® requirements management process ……………………………………….. 42

xi
Figure 3.8 dRofus® Graphic User Interface (GUI) indicating checkboxes and drop-down menus used to avoid error ................................................................. 43
Figure 3.9 Illustration of data from a functional program being populated in their respective field in dRofus® ................................................................................. 44
Figure 3.10 Sample design requirements data received from an owner in an Excel spreadsheet 45
Figure 3.11 Sample design requirements data received from an owner in a scanned PDF ....... 45
Figure 3.12 dRofus interface to generate room data sheet................................................................. 46
Figure 3.13 Room data sheet of ‘Waiting Room’ in Project A ........................................................ 47
Figure 3.14 Sample scanned mark-up document ............................................................................. 48
Figure 3.15 Sample addendum issued ............................................................................................ 49
Figure 3.16 Redundant data in the dRofus® project database ......................................................... 51
Figure 3.17 Data missing in the respective field of the database...................................................... 53
Figure 3.18 External documents referenced in dRofus® ................................................................. 55
Figure 3.19 Project B dRofus database welcome screen ................................................................. 58
Figure 3.20 Project B dRofus database- room list .......................................................................... 59
Figure 3.21 Project B dRofus database- space programming requirements .................................... 59
Figure 3.22 Project B dRofus database- link to external document referenced in database....... 60
Figure 3.23 Project a location and adjacency diagram in functional program ................................. 62
Figure 3.24 Current space program workflow as identified in the architectural firm ................. 63
Figure 3.25 Designer’s meeting to deliberate space programming during early stages of design 64
Figure 3.26 Bubble diagram drawn for space programming .......................................................... 65
Figure 3.27 Space program compliance validation attempt by project team (i) ......................... 68
Figure 3.28 Space program compliance validation attempt by project team (ii) ......................... 69
Figure 3.29 Space program compliance validation attempt by project team (iii) ......................... 70
Figure 3.30 Space program compliance validation attempt by project team (iv) ..................... 71
Figure 4.1 Exporting excel spreadsheet from dRofus® ................................................................. 76
Figure 4.2 Space programming requirements data dumped into dRofus® fields ....................... 76
Figure 4.3 Generated excel spreadsheet from dRofus® ............................................................... 77
Figure 4.4 Excel spreadsheet structured for parsing ................................................................. 77
Figure 4.5 Formula used to flag room relationship requirements ............................................. 80
Figure 4.6 Formulae used to parse space program requirements ............................................. 81
Figure 4.7 Searchable dashboard for location relationship requirements (a) ......................... 84
Figure 4.8 Searchable dashboard for location relationship requirements (b) ......................... 85
Figure 4.9 Theatre space program requirement from SOR ....................................................... 87
Figure 4.10 ‘Location’ space program requirement accuracy statistics .................................... 87
Figure 4.11 Project A first-floor daylight requirements compliance test .............................. 90
Figure 4.12 Project A second-floor daylight requirements compliance test ....................... 90
Figure 4.13 Project A third-floor daylight requirements visualization ........................................ 91
Figure 4.14 Project A fourth-floor daylight requirements visualization .................................... 91
Figure 4.15 Project A first-floor acoustics requirements compliance test .............................. 94
Figure 4.16 Project A second-floor acoustics requirements compliance test ..................... 94
Figure 4.17 Project A third-floor acoustics requirements compliance test ............................. 95
Figure 4.18 Project A fourth-floor acoustics requirements compliance test ....................... 95
Figure 4.19 Dynamo® script for daylight and acoustics space program compliance validation . 97
Figure 4.20 Location space program requirement compliance validation - Project A first floor 100
Figure 4.21 Adjacency space program requirement compliance validation - Project A first, second and third floor ................................................................. 102

Figure 4.22 Access space program requirement compliance validation - Project A first floor ... 104

Figure 4.23 Access space program requirement compliance validation - Project A second floor .................................................................................................................. 105

Figure 4.24 Access space program requirement compliance................................................................................................................................. 105

Figure 4.25 Dynamo® script for location, adjacency and access space program compliance validation ........................................................................................................... 106

Figure 4.26 Smart excel template containing room relationship data ........................................ 108
List of Abbreviations

BIM: Building Information Model
BIM: Building Information Modelling
FLP: Facility Layout Problems
GSP: General Space Planner
GUI: Graphic User Interface
IMIT: Information Management and Information Technology
RDS: Room Data Sheet
SOR: Statement of Requirements
SPG: Space Plan Generator
Acknowledgements

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisor and mentor Dr. Sheryl Staub-French. Without her support and guidance, this study and milestone in my life would not have been possible. Thank you for supporting me on days I thought this wasn’t possible and lacked certainty in myself.

I would also like to extend my immense gratitude to Stantec Consultancy Ltd and the Practice Technology team, for the financial and motivational support for this research. Especially, Aubrey Tucker for providing me with a creative space, constant feedback and support on my work, this would not have been possible otherwise. I would also like to express my appreciation for the technical support and valuable feedback I have received from Achintya Bhat, Alyssa Haas and Ryan Wells throughout the development phase of this research, during my work at Stantec Consultancy Ltd.

I am most grateful for my family for their unconditional love and undying support throughout my life even when they were thousands of miles away. Last but not least I would like to thank my friends that I consider family for their constant support and encouragement throughout this journey.
To My Mother,

Who I Owe For Everything I Am Today.
Chapter 1: Thesis Overview

1.1 Introduction

Space programming is a critical task during the schematic design process that requires the geometric configuration of a space layout in accordance with the project’s requirements. By nature, space programming is an iterative process that evolves according to the client’s requirements (Kiviniemi, 2005 & Guo & Li, 2016). A space program frames the entirety of the building and has a major effect on whether the building will function well for the intended purpose. Designers are expected to comply to an owner’s requirement with a set of criteria and this can be a tough endeavor for designers as some building program requirements may be complex (Zifeng & Li, 2017). A designer constantly reiterates a space program until it is sufficiently compliant with the client’s requirement. This process generally entails a designer receiving space programming design information from clients in various formats and unstructured data sets during the early stages of design and manually analyzing it to make changes in the design software accordingly.

Space programming requirements can be significantly different depending on the type and function of the project being constructed. These requirements are commonly received from an owner in a standard document known as a Statement of Requirement (SOR). The SOR is a “rule book” that will contain key information about the functional program, such as room and department names, room areas, the function of each room, and the proximity between rooms. These requirements can be complex to analyze when developing a final space program solution that is compliant with the client’s requirements due to the interdependent relationship between
rooms, spaces and departments (Guo & Li, 2016). This demonstrates the importance of leveraging data to further improve the compliance of space programming in accordance with the client’s requirements.

A critical challenge of space programming is the limitation in the link between the client’s requirements and design tools (Chae, 2017 & Yi, 2016). The rigorous process of analyzing, structuring and extracting meaningful information often leads to requirements being overlooked or important requirements failing to be satisfied (Kiviniemi, 2005). Closing this gap could result in a designer approaching an overall design solution effectively by solving smaller requirements incompliance that are discovered though analysis and visualization. Failure to meet the client’s space program requirements, could possibly lead to decline in the performance of the building, cost increase, client dissatisfaction and penalty fines charged by the client which are usually clearly stated in design contracts. This is typically due to rooms that are not able to function for their intended purpose because of incompliance with proximity relationship requirements. A space program’s design compliance with the client’s requirements are important for two major reasons: (i) To allow for a building to carry its functions and high performance by using the space effectively and (ii) To avoid space program incompliance that may result in fines, client dissatisfaction, and the deterioration of the design firm’s ability to attain future projects (Touloupaki & Theodosiou, 2017, Derix, 2014, Deutsch, 2015, American, I. O. A, 2013 & Cherry & Petronis, 2016). For example, during the case study conducted at a large architectural firm, it was found that for large hospital projects the nonconformity of a space program could be quite costly, up to $200,000 CAD for the misplacement of a single room due to incompliance.

This research aims to investigate the extent to which computational tools in conjunction with BIM could be used to evaluate a building project’s space program requirements compliance
through semi-automated analysis at a large scale architectural firm. Specifically, I investigated the current practices of space program requirements data management, and design workflow within the firm and proposed a solution to bridge the gap between the manual analysis of the client’s space program requirements and the firm’s space programming workflow.

In order to understand the challenges and develop an optimization method, I first followed the firm’s current practices of design requirements data collection and management to populate the project database. By closely following project data managers and designers, I recorded their workflow and analyzed space programming requirements documents. Primarily, I focus on the structuring and analysis of the client’s requirements data for Mechanical, Electrical, Structural and Architectural disciplines for three building projects. Next, I populated the requirements information for all the disciplines into each of their dRofus® project databases. dRofus® is a data management and BIM collaboration tool server to maintain data for departments, rooms, room templates, finishes, items, systems, and components (dRofus, 2015). This allowed me to record the challenges and limitations of the workflow to structure qualitative data of space programming requirements. Shortcomings such as poor database management, data redundancy, and referencing external documents within the database, were revealed which are discussed later in this study. Observing the current highly iterative space programming design workflow and conducting informal interviews at the firm on other projects, revealed that space programming consumed about 10-20% of the total design time. The current practice of space programming did not benefit from the availability of requirements data to support its compliance with the client’s requirements.

Following the current practices of design data analysis and population at the architectural firm, I developed a semi-automated workflow that assesses the compliance of the space program
against the client’s requirements in a more efficient and automated manner. The shortcomings recorded motivated me to develop a smart Microsoft Excel® template to parse, analyze and structure space programming requirements. The focus of this research was narrowed to space program requirements of daylight, adjacency, access, visibility, daylight and acoustics. By leveraging the parsed requirements data from the smart Microsoft Excel® template I developed a Microsoft PowerBI® dashboard to visualize these space programming requirements. This provided designers an interactive, searchable dashboard visualization of room interrelationships. In addition, to assess the compliance of a building project’s space programming requirements I developed a visualization in conjunction with a visual computational tool. The evaluation and assessment of the space programming requirements was carried out by using Autodesk Revit® and Dynamo® to produce a visual overlay on the architectural model’s floor plan. This development was well received by the designers at the large scale architectural/engineering firm.

1.2 Research Objectives

After researching case studies and developments made in the field of space programming the following objectives were established for this research to contribute to space programming:

1. To examine the current practices and challenges of space programming and requirements management at a large scale architectural firm.

2. To develop an automated approach to analyze and visualize space programming compliance.

3. To develop a space programming information visualization dashboard.
In order to achieve the underlying objectives of this research project, the following research tasks (RT) were carried out:

**RT1:** Analyse and identify client’s design requirements.

**RT2:** Populate architectural, mechanical, electrical and structural design requirements in dRofus®.

**RT3:** Communicate with various design discipline teams to verify requirements database accuracy.

**RT4:** Attend space program design meetings and work shadow throughout the space program design process.

**RT5:** Develop smart excel template to parse space program requirements data.

**RT6:** Develop computational approach to visualize space program requirements as a floor plan overlay.

**RT7:** Develop Microsoft PowerBI® room relationships dashboard visualization.

### 1.3 Research Methodology

This section introduces the case study and research methodology carried out to accomplish the objectives and tasks of this research. Figure 1.1 shows the objectives, tasks, and outcomes (RO) that have resulted from this research. Under the tasks carried out are indicated the sub-section where a detailed discussion can be found of the specific task. Figure 1.2 shows a simplified visual roadmap for the visual assessment of space program requirements compliance.
<table>
<thead>
<tr>
<th>Objectives</th>
<th>Tasks</th>
<th>Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examine Current Practices of Space Programming Requirements Management and Design</td>
<td>Analyze and Identify Clients Design Requirements 3.4.1</td>
<td>Identify Requirements Management Process RO1</td>
</tr>
<tr>
<td></td>
<td>Populate Architectural, Mechanical, Electrical and Structural Design Requirements in dRoofs 3.4.2</td>
<td>Identify Space Programming Design Process RO2</td>
</tr>
<tr>
<td></td>
<td>Communicate with Various Discipline Design Teams to Verify Requirements Database Accuracy 3.4.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attend Space Program Design Meetings and Work Shadow Throughout Space Program Design Process 3.4</td>
<td></td>
</tr>
<tr>
<td>Develop Automated Approach to Analyze and Visualize Space Programming Compliance</td>
<td>Develop Smart Excel Template to Parse Space Program Requirements Data 4.2</td>
<td>Optimize the Manual Analysis of Clients Space Program Requirements Documents RO3</td>
</tr>
<tr>
<td></td>
<td>Develop Computational Approach to Visualize Space Program Requirements as a Floor Plan Overlay 4.4</td>
<td>Assess the Compliance of Space Program Requirements Through Visual Results RO4</td>
</tr>
<tr>
<td>Develop a Space Programming Information Visualization Dashboard</td>
<td>Develop Microsoft PowerBI Room Relationships Dashboard Visualization 4.3</td>
<td>Visualize Space Program Requirements RO5</td>
</tr>
</tbody>
</table>

**Figure 1.1 Research roadmap**
Figure 1.2 Visual roadmap of space program requirements compliance assessment approach
1.3.1 Case Studies

Case studies help a researcher gain insight on cases, by comprehending why decisions were made, how they were implemented and the results they deliver (Yin, 2008). The space programming design process in practice was investigated to develop a framework that could be implemented in future projects by analyzing the owner’s requirements. This was accomplished by implementing an observation-based empirical research approach.

This study was performed at large scale architectural and engineering firm. The firm is a significant international contender in this field with over 64 years of experience. The firm provides consultancy for architectural, structural, electrical, and mechanical disciplines. The study was performed in the Vancouver, British Columbia office over a course of one year.

The projects that were studied for this research along with their general information are depicted in Table 1-1. In an effort to populate the dRofus® databases for design purposes, different documents were analyzed for each project; these documents are identified in the ‘Data Analyzed’ column of Table 1-1. Each project’s datasets were examined to a different extent, and for different research objectives. The extent to which each projects dataset was analyzed is different depending on the hours spent on population and the density of data populated as indicated in Table 1-1.
<table>
<thead>
<tr>
<th>Project</th>
<th>Project Type</th>
<th>Location</th>
<th>Project Cost</th>
<th>Data Analyzed</th>
<th>Data Populated in dRofus®</th>
<th>Hours of Data Analysis</th>
<th>Type and Level of Interaction with Project Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>Institutional</td>
<td>New Westminster, BC</td>
<td>$106.5 Million</td>
<td>Statement of Requirements, Functional Program, Room Data Sheets, 3D Models, Addendums, Request for Proposal, Mark-ups, Space Programming Requirements Data</td>
<td>Mechanical, Electrical, Structural, and Architectural Design Requirements Data</td>
<td>120</td>
<td>Frequent emails with project design team to update dRofus® database</td>
</tr>
<tr>
<td>Project B</td>
<td>Institutional</td>
<td>Edmonton, Alberta</td>
<td>$260 Million</td>
<td>Statement of Requirements, Functional Program, Space Program List, 3D Models, Space Programming Requirements Data</td>
<td>Built dRofus® database with departments and zones with essential room data</td>
<td>21</td>
<td>None</td>
</tr>
<tr>
<td>Project C</td>
<td>Healthcare Facility</td>
<td>Owen Sound, Ontario</td>
<td>$25 Million</td>
<td>Room Data Sheet, Functional Program, Addendums, Mark-ups</td>
<td>Mechanical, Electrical, Structural, and Architectural Design Requirements Data</td>
<td>40</td>
<td>Limited emails with project design team to update dRofus® database</td>
</tr>
<tr>
<td>Project D</td>
<td>Healthcare Facility</td>
<td>Calgary, Alberta</td>
<td>$1.4 Billion</td>
<td>Room Data Sheet, Functional Program</td>
<td>Mechanical, Electrical, Structural,</td>
<td>12</td>
<td>Limited emails with project design</td>
</tr>
</tbody>
</table>
and Architectural Design Requirements Data

<table>
<thead>
<tr>
<th>Addendums</th>
<th>Few meetings with project data manager to discuss client design requirement document analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark-ups</td>
<td>team to update dRofus® database</td>
</tr>
</tbody>
</table>

Table 1-1 Projects description and data analyzed
Figure 1.3 illustrates a rendered view of Project A which was a secondary school replacement project located in New Westminster, BC. Although the procurement of Project A was not successful, the design and design database were highly detailed and were realistic enough for this research. The involvement with the project team was high due to the involvement with the project’s design database population of dRofus®. Constant feedback was received about the design data management from all disciplines. The documents containing this data were shared on the design firm’s server and Procore – a document management system. A final survey was conducted on Project A’s architectural design team to investigate the impact of the developed space programming framework; this will be discussed in Chapter 4.

![Figure 1.3 Project A rendered view](image)

Project B is shown in Figure 1.4, it is a large public institutional building with an approximate budget of $260 million located in Edmonton, Alberta. It was procured from the Government of Alberta and was completed before this research had commenced. However, the availability of the design documents and design models allowed for this research to be conducted on the project. The study conducted on Project B consisted of analyzing its client’s requirements.
documents and design model to build a simple dRofus® database. I built the GUI to investigate the time consumed to build a dRofus® database and customize it to contain separate space programming fields to provide architectural designers with clarity. This simple database was built for research purposes alone and was not an official database.

Project C was a small-scale healthcare facility located in Owen Sound, Ontario. The project was not procured; however, multiple design documents were analyzed to populate the dRofus® database. Project D is a large healthcare facility under construction in Calgary, Alberta anticipated to be completed in 2023. The involvement with Project C and Project D for this research was limited to analyzing data and populating the database for design purposes. This helped to gain an understanding of the similarities and differences of design data requirements for different project types.
Data for this research had been collected from four projects but because Project A was in the space programming design phase throughout the duration of this research, the involvement and contribution were higher and as a result, it will be discussed in more depth. Table 1-2 illustrates the tasks carried out on each Project.

<table>
<thead>
<tr>
<th>Project</th>
<th>Project A</th>
<th>Project B</th>
<th>Project C</th>
<th>Project D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build dRofus® GUI</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze Design Documents</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Populate dRofus® Database</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Parse Space Programming Requirements Using Smart Excel Template</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Develop Space Programming</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1.3.2 Research Approach

An observation-based empirical research approach was implemented for this case study. This was conducted by following A.D. de Groot’s (Heitink, 1999)[1] empirical framework of:

1. **Observation**: “The observation of a phenomenon and inquiry concerning its causes.”
   (Wikipedia, n.d.)
   
   Primarily, an observation was made concerning the issues of the current practices of space programming and requirements management. Populating the dRofus database (RT2) and attending space programming meetings throughout the space programming design process (RT4) revealed that there was a poor representation of requirements and inefficient manual process of space program compliance assessment.


   The poor representation of requirements was assumed to arise from data dumping by design data managers as well as failing to populate data in the proper fields. In addition, the current space programming practice lacked a link between requirement documents and the design software. This was observed when I carried out RT2 and RT4.

3. **Deduction**: “The formulation of experiments that will test the hypotheses (i.e. confirm them if true, refute them if false).” (Wikipedia, n.d.)

   Identifying the requirements management process (RO1) revealed that the requirements documents received from the client were unstructured and time consuming to analyze for
the data manager and designers. In addition, identifying the space programming design workflow (RO2) revealed that assessing a space programs compliance was a highly iterative and time-consuming workflow.

4. **Testing:** “The procedures by which the hypotheses are tested, and data are collected.” (Wikipedia, n.d.)

I developed a smart Microsoft Excel® template to parse and structure space program requirements data (RT5). The template was tested on Project A and utilizing this template optimized the manual analysis of client’s requirements documents by significantly lowering the time consumed to analyze and populate space program requirements (RO3). Furthermore, a semi-automated visual script was developed in Dynamo® (RT6) to support designers visualize and assess space programming requirements (RO4). Finally, the parsed requirements data was used to develop a Microsoft PowerBI® dashboard (RT7) to visualize space programming requirements (RO5).

5. **Evaluation:** “The interpretation of the data and the formulation of a theory - an abductive argument that presents the results of the experiment as the most reasonable explanation for the phenomenon.” (Wikipedia, n.d.)

Designers were presented with the developments made in this study, such as the: (i) smart Microsoft Excel® template, (ii) visualization of space program requirements as a floor plan overlay, and (iii) visualization of room relationship dashboard, to evaluate and reflect the possibility of implementation in future projects. The developments were well received by the designers and the results of the evaluation are discussed in depth in Chapter 4.
1.3.3 Requirements Data Management

The database of three building projects were analyzed, structured and populated (RT1 and RT2); one secondary school and two healthcare facilities to identify the requirements management process at a large architectural/engineering firm (RO1). This was achieved by closely following the project’s data manager and recording her workflow. After learning the workflow, I structured and populated Project A, C and D’s requirements data into their respective dRofus database from SOR’s and addendums for designers and Project Managers to use (RT2). This was an iterative process of requirements data management that also included constantly communicating with the various design disciplines and constantly receiving feedback to verify the database’s accuracy (RT3).
1.3.4 Structuring and Visualizing Space Programming Requirements

I developed two visualizations, so designers can assess the compliance of space program requirements (RO4) and visualize space program requirements in a dashboard (RO5). The first visualization leveraged a visual programming tool to develop a script that visualizes space programming requirements in the form of a floor plan overlay using Autodesk Revit® and Dynamo® (RT6). The second visualization was a Microsoft PowerBI® dashboard that visualizes room interrelationship requirements such as location, adjacency, and access (RT7). The visualizations were developed so that designers would be able to easily identify space programming related data and avoid constantly filtering through the SOR.

1.3.5 Validation through Expert Reviews

An interview was conducted amongst the three architectural designers that had the highest influence on the space program design of Project A, to explore the value and likelihood of the architectural firm implementing the developments of this research on future projects. A presentation was made to the space programming design team discussing two visualizations that were developed during the course of this research. The developments were well received, and the results of the interview are discussed in depth in Chapter 4.

1.4 Thesis Overview

This thesis consists of five chapters. Chapter 1 provides a general overview of the research topic and discusses the research methodology that was implemented for this study. Chapter 2 discussed literature that has been reviewed and gains insight on the work that has been done by researchers in this topic. Chapter 3 will discuss the research conducted at the
architectural firm where the study was conducted. This will include the project backgrounds, the
current practices of space programming design and the current practices of requirements
management at the firm. Chapter 4 will discuss the approach implemented to support designers
visualize space programming requirements at the large scale architectural firm. Chapter 5 will
conclude the research and provide a conclusion as well as its limitations and scope of future
work.
Chapter 2: Literature Review

The Whole Building Design Guide (WBDG) defines space programming as “the research and decision-making process that identifies the scope of work to be designed”. Space programming has existed since the beginning of architecture as an exploration of decision making for design (Cherry & Petronis, 2016). Different researchers have developed algorithms to aid with solutions for space program issues since the 1960’s. However, the increasing complexity of design and construction in the last few decades and the significant reduction in time available for a project’s preliminary conceptual design phase has changed the space programming design activity (Donato, 2017). The constraint of space, decisions, information and specifications has served as a motivation for researchers to make developments towards space programming compliance (Yi, 2016). There are different constraints and requirement specifications that could be necessitated by the client depending on the complexity of the project such as the proximity between rooms, daylight requirements, and the noise level some rooms produce that may affect others.

2.1 Space Programming Requirements Representation

Any developments or process of space programming begins with the analysis of the space program requirements received from the client. The program document containing the critical information of the building requirements can be received in different forms and structures from a client (American, I. O. A, 2013). “Regardless of form, however, analyzing the building program and drawing out the critical information is an important first step” (American, I. O. A, 2013).
The content of this document may be referred to as Statement of Requirements (SOR), Owners Statement of Requirements (OSR) or Statement of Intent (SOI). However, the information required for the space program will be referenced as an external document referred to as either the ‘Functional Program’ or ‘Facility Program’. In this research, this document will be referred to as the ‘Functional Program’ in accordance to the case studies conducted- discussed later in this chapter. The functional program is a document that contains detailed information on the building scope, function, departments, rooms and spaces. Essentially information such as net area, lighting, temperature, sound level, and connections to other spaces can be found within this document (Kiviniemi, 2005). Large healthcare, institutional, and research facility projects have multiple functional requirements that are interdependent and complex. This interdependency of functional requirements requires different deliberations by the designer and may cause a strain when they are complex. A change in one of the functions can result in a chain reaction, affecting many other functions (Sang Min Park, 2004).

(Cherry, 2008) defines the process of space programming being successful when it is able to achieve the following:

1. The client’s requirements being met within the budget.
2. The design team being determined and focusing on meeting the criteria listed in the requirements.
3. A benchmark being established, for which any future changes can be compared against to charge the client for additional services when changes are made after the design phase has commenced.

Whether it is for a simple or complex project or application, (Pena & Parshall, 2001) establish the information required for a space program to be designed:
1. Function
   • People
   • Activities
   • Relationships

2. Form
   • Site
   • Environment
   • Quality

3. Economy
   • Initial Budget
   • Operating Costs
   • Life cycle costs

4. Time
   • Past
   • Present
   • Future

These requirements contain both quantitative and qualitative information. When dealing with quantitative data, such as room area or ceiling height, it is easier to extract or validate this data within the requirements. However, it may not be as easy when dealing with qualitative data as this data is possibly a description from an owner or future occupant (Marchant, 2015 & Pena
Limited research is available on how to automate the extraction of qualitative data for space programming.

The space programming process consists of designers relying mostly on their memory of the client’s requirements (Kiviniemi, 2005), this is because it is a highly iterative process and it would be protracting the limited time they have to refer to the SOR for the requirements of each space, and their interdependent relationships each time changes were made. As Kiviniemi (2005) explains and substantiated by (Guo & Li, 2016), it is impossible for a designer to remember all the critical information and the relationships between requirements for the following reasons:

- The amount and complexity of project information,
- The duration of projects,
- The need for designers to work simultaneously,
- Changing stakeholders in different project phases, and
- Shifting design focus, e.g., moving from overall problem solving to detailed technical solutions.

Kiviniemi (2005) argues that because “design tools do not support recording of client requirements or designer’s intent in the documents”, the aforementioned reasons would collectively result in what he calls a “shift away” from the goal, that being the client’s requirements. Figure 2.1 illustrates the drift that may result when designers do not access the requirements and make changes based on the previous design solution.
The challenges of space programming identified in the literature were a significant research motivation to develop a better workflow for space programming and to automate the extraction of critical data from design documents to produce visualizations of the compliance in the form of an overlay on the floor plan of the design software.

2.2 Visualization of Space Programming Requirements

Space programming requires significant analysis of requirements documentation. Although information could exist in a document or a database management system that is rich in data, it may be time consuming and challenging for a user to access if it is not in the suitable format or structure (Hu et al., 2016). When designers review design documents for information they encounter large data sets that is not organized in a way for them to quickly extract the information they need. Analyzing and visualizing this data would save the cumbersome time the designers spend on design documents. “Data visualization helps communicate complex information, insights, and abstractions to nonprofessionals, and makes data more accessible and
understandable to more people” (Deutsch, 2015). Utilizing data visualization could benefit in identifying relationships between various data to enhance decision making and improve the ability of the design team to analyze data (T Korde, 2005 & Russell et al., 2009). Visualizing space programming requirements allows for a designer to easily analyze and understand the requirements a client is conveying in a shorter period of time or to a greater depth (Chae, 2017 & Gallagher et al., 2008). In addition, an in-depth space programming requirements analysis and visualization could be accomplished to evaluate the compliance of the space program that has been designed by leveraging BIM tools. “In general, the benefits from design validation include greater opportunities to increase building performance, prevent miscues, and enhance project success” (American, I. O. A, 2013).

Recent researches focus on parametric modeling to generate and visualize space program requirements. This is due to a new generation of architects being accustomed to a powerful BIM digital process and BIM tools for design generation and representation, as well as the ability to address multiple requirements at the same time during various stages of design (Touloupaki & Theodosiou, 2017). Discussed below are a few approaches that have been developed to bridge the gap between space programming requirements analysis and the space programming process by developing visualizations of space programming requirements data.

Yi (2016) developed a space layout tool to generate space layout geometry to evaluate “daylight level and room shading” to help architects make design changes accordingly. “Design information, strategies, and functional requirements that are identified at this step outline an overall direction of the specified spatial organization within particular contexts and project objectives” (Yi, 2016). Figure 2.2 illustrates three variations of space layouts generated by Yi’s tool, each color block representing a room.
Das & Haymaker (2016) propose the use of “an emerging methodology and tool” that generates space plan designs known as Space Plan Generator (SPG) using Autodesk Dynamo® and Project Fractal®. They generate multiple designs using a hierarchical approach of placing departments first and then programs within the department, and finally circulation. As shown in Figure 2.3, this methodology uses a cell matrix approach that applies different weights, such as acoustic performance, daylighting, and site constraints to “cells”, which are departments, rooms or circulation. Using a program document containing information such as ID, name, department, quantity, area, program, preference value, and adjacency, an Autodesk Dynamo® script is used to analyze and allocate the elements.
Boon et al., (2015), used analytical tools Grasshopper and Galapagos to develop a script “to graphically represent and optimize the adjacency requirements in programmatic spaces”. Through client interviews and intuition, they first develop a relationship matrix for the programmatic elements, including a priority ranking with color tones (Figure 2.4). This workflow that developed the relationship matrix as shown in Figure 2.4 was developed manually by designers through meetings with clients. Next this relationship matrix is used to automate space program solutions for many rooms on multiple stories as shown in Figure 2.5. Figure 2.5 illustrates the final production after the requirements have been analyzed and the Galapagos script is run.
Figure 2.4 Various programmatic elements for a hospital and the relationship with others. There are three different levels of importance indicated by tones (Boon et al., 2015).

Figure 2.5 Most fit iteration generated using Galapagos script (Boon et al., 2015)
Guo & Li (2016) propose a method for “the automatic generation of a spatial architectural layout from a user-specified architectural program”. Although they do not mention the software or tools used to produce these generations, they adopt a model where rooms are represented by “bubble agents” and are controlled by “interaction rules” that are manually defined to move, push or swap rooms. The rules defined to generate the layout and place rooms are (i) topology (circulation, level of the room and the connection the room has with other rooms), (ii) shape, (iii) dimension, and (iv) aspect ratio and building shape. After, all these rules are manually defined by a user, the program swaps the position of rooms until the highest probability of compliance sought by the program is achieved. The 3D model shown in Figure 2.6 is then generated through an evolutionary process.

![Figure 2.6 Process of generated multi-agent layout to optimized grid system (Guo & Li, 2016)](image)

### 2.2.1 Space Programming Visualization Conclusion

The literature reviewed for this research adopt methodologies and tools for analysis, generation, and optimization of space programming. The methodologies discussed are among the commonly used methods to automate the space programming workflow using tools such as
Autodesk Revit®, Autodesk Dynamo®, Project Fractal®, Grasshopper®, and Rhino®. These methods are popular among designers to automate space programming workflow and generate thousands of solutions while taking multiple requirements such as acoustics, daylight, room adjacency and circulation, into account.

However, the solutions proposed by previous research have some limitations:

(i) They focus primarily on generating new layouts and do not evaluate the compliance of an existing space program.

(ii) They focus on fitting the entire space program into a floor plan without applying extensive consideration into the adjacencies between the elements.

(iii) They limit the relationship constraint to either adjacency or distance only; realistically, a room or space will have multiple or a combination of constraints that will require it to be placed near or further away from a room.

(iv) Lack of focus on the automation to parse space program requirements data.

Previous developments begin by manually inputting space program requirements into their respective tools to generate space programs. This creates a tedious workflow initially and also requires for these requirements to be manually changed if any iterations are made to the requirements.

Fallman (2003) and Kiviniemi (2005) have highlighted that a client’s requirements are too complex for a machine or algorithm to generate space programs that will be entirely compliant with the client’s requirements. The shortfall of these methods that use generative design as a solution to develop space programs is that by doing so you are overlooking the experience of the designers and depending on a computed technique of mathematical optimization. Guo & Li (2016) state that “the idea of introducing such a technique does not mean
replacing architects, but rather developing powerful tools to find fast solutions, verify ideas, and choose the best tool for furthering design” such a method helps designers assess their existing design through visualizations.

2.3 Literature Review Conclusion

Space programming is recognized for being an important first step in schematic design as it is critical in dictating the functionality of the building project. Recent developments are adopting methodologies and tools of computer science for analysis, generation, and optimization of design. Although previous researchers have made developments that clearly link space programming requirements to the design process by automating visualizations, their methodologies have an end product where the adopted workflow and tool produce space programs with no room for manipulation by the designer. This leaves out the biggest impact player out of the workflow- the designer.

My approach has made developments using the ideology behind Kiviniemi (2005) in focusing on the management of design documents and client’s requirements to produce design solutions. This would result in a data-driven design that supports designers by eliminating rigorous task such as repeatedly cross checking the floor plan to the space program requirements.
Chapter 3: Current Space Programming Process, Requirements and Data

3.1 Introduction

In this chapter, the research conducted at the large architectural firm is discussed. The case study analyzed the current practices of the architectural firm to manage the mechanical, electrical, and architectural requirements data for Project A, C, and D. This helped explore the current practices to gather and analyze data to develop a compliance assessment method for space programming design through requirements visualization.

The tools used as investigated during the case study for space programming requirements representation as well as the tools that were deemed fit and used during the development stage of this study are introduced below.

3.2 Tools Used for Requirements Representation, Design and Visualization

dRofus

dRofus® is a cloud-based data management and BIM collaboration tool server used to manage and represent data for departments, rooms, room templates, finishes, items, systems, and components (dRofus, 2015). dRofus® is a unique requirements database management software in that the graphic user interface (GUI) can be modified according to a project’s needs and user preference. Although the dRofus® GUI could be customized according to a design firm’s preference, it generally consists of a room database that contains the architectural, structural, and HVAC design information, while the equipment database consists of plumbing, mechanical, and electrical building item requirements that can be used for planning and cost estimation. dRofus®
is a BIM integrated tool that can be integrated with existing design tools, such as Revit®, ArchiCAD, Solibri, IFC compatible BIM solutions, and Microsoft Excel. In addition, it has advanced plug-ins for Revit® and ArchiCAD, which creates a seamless workflow for architects and engineers. The plug-in feature of dRofus® allows a user to validate building requirements by integrating or synchronizing design data from the project model to dRofus® or vice versa. Once the dRofus® database is populated with requirements data, it supports the publishing of reports and exports in PDF and Excel formats. In addition to tracking room and equipment requirements during the planning phase, dRofus® also has a procurement and maintenance module for the construction and maintenance stages of a project.

   dRofus® is used within the large architectural/engineering firm for project’s requirements data management and representation. Architects, Engineers, Data Managers and Project Managers within the firm use it to share project requirements and sync requirements data to the project model. I used dRofus® profoundly during the early stages of this study to structure and populate project requirements for different disciplines. This requirements data was later exported in a Microsoft Excel® format to develop the Microsoft PowerBI® dashboard and floor plan overlay visualization in Autodesk Revit® and Dynamo®.
Autodesk Revit

Revit® is design software that is used by architects, engineers, and contractors that supports 3D modelling, clash detection, generation of bill of quantities along with accurate quantity takeoff, and design change management. It is capable of creating one unified model with all building elements and parameters. This allows for all the data and information to be placed in a single project file within the Revit® server, that can also be used in future models if needed. In addition to the design phase, Revit® has a capability of shared parameters that can be used for facilities and operations management. This aids an owner and operator to maintain accurate data for scheduled maintenance, disaster planning, recovery and renovations.

Revit® was the BIM tool used within the firm to develop the models for the projects discussed in the next sub-topic. Revit® was used in this study for two major reasons: (a) to link
the requirements data of each project to the dRofus® database and (b) export the project space program to Dynamo® to produce the space program requirements as floor plan overlay visualization.

![Autodesk Revit® floor plan and 3D model rendering](image)

**Figure 3.2 Sample Autodesk Revit® floor plan and 3D model rendering**

**Autodesk Dynamo**

Dynamo® is a computational visual design tool that can be used as standalone software or as a plug-in for design software, such as Autodesk Revit® or Maya®. Dynamo® can be used as a visual programming tool to connect elements and define relationships and sequences. It bridges the gap between coding and composing algorithms for users that lack extensive knowledge of coding. The Dynamo® interface is shown in Figure 3.3 and consists of nodes that are attached to other nodes through wires by an input or output port. Each node is made up of a python script to perform an operation and when connected to other nodes constitute a workflow.
that carries out a task. Nodes exist in the Dynamo library by default, however custom node packages may be installed to carry out special or complex tasks.

Dynamo® was used in this study to develop a script for the visualization of space program requirements in conjunction with Autodesk Revit®. An in-depth discussion of this script can be found in Chapter 4.

Microsoft PowerBI

PowerBI® is a self-service solution for business analytics created by Microsoft. It is a platform that allows users to experience and monitor a customized data warehouse referred to as a dashboard. It allows users to transform data into visually appealing and interactive reports. PowerBI® allows users to connect with hundreds of essential data sources such as Excel, Azure,
and SharePoint. In design and construction, one of the ways PowerBI® can be used is to track the progress of design and design parameters, as well as assess project performance and status.

Using the space program requirements data identified by the smart Microsoft Excel® template, a Microsoft PowerBI® dashboard was developed during this study for space programming requirements. This was developed to provide designers with an interactive, searchable dashboard that will provide them with a quick room interrelationships visualization for space programming requirements.

![Figure 3.4 Sample PowerBI® dashboard (PowerBI, 2018)](image-url)
3.3 Current Practices: Space Programming Analysis and Representation

3.3.1 Design Requirements Analysis

This section will discuss the research conducted at the architectural firm on Project A, C, and D regarding findings on the design requirements received from clients and its analysis by designers and data managers. The three projects revealed that an SOR will include several sections ranging from Project Information, Architectural Design Principles, Engineering Design Principles, to Technical Specifications. In addition to an introduction, purpose, structure, and methodology of the functional program, data for the following requirements were included in the four projects studied for this research:

- Functional Area Description
  - Departments

- Space Program
  - Space List
    - Rooms
      - Zones
      - Net Area

- Access, Circulation, Location and Adjacency Diagrams or Description
  - Overall Access, Adjacency and Circulation
  - Access and Circulation - Handicapped
  - Access and Circulation - Vestibules

- Design Features
  - Temperature Requirement
  - Daylight requirement
- Hours of operation

- Room Data Sheets
  - Supplementary Information
  - Furniture, Fixtures and Equipment

- Architectural Requirements
  - Design Features
  - Volume Space and Massing
  - Codes, Regulations, and Building Standards
  - Materials and Finishes
    - Floors
    - Doors
    - Walls
    - Ceilings
  - Communication Systems
    - IT/Data Communications System
  - Security Systems
  - Fire Protection Systems
    - Fire Alarm System - Detection
    - Fire Protection System - Extinguisher
    - Sprinklers
    - Fire Department Access
  - Acoustic Systems
  - Vertical Circulation Systems
- Elevators and Escalators
- Elevator Life Safety and Security

  o Building Room Criteria
    - Doors and Openings
    - Ceiling Height
    - Windows

- General Remarks

  Although populating all these requirements could be beneficial for a design team, it was learned from the study conducted on Project A, C, and D that the completeness of the dRofus® project database is dependent on:

  - The stage of design the project is currently in,
  - The current practice of the firm and the awareness towards the usefulness of a comprehensive database,
  - The time available before a project is issued for tender, and
  - The quality of the functional program received from an owner.

  Figure 3.5 shows a sample functional requirement of the department ‘Grand Commons’ from Project A, pertaining to two rooms received from a client. This data is shown in Figure 3.8 and 3.9 populated in the project database.
3.3.2 Design Requirements Representation

dRofus® was the project database management platform of choice used by the firm to populate and manage Project A, C, and D’s design requirements data for Architectural, Electrical, Structural and Mechanical disciplines. The usage of dRofus® was encouraged in the firm because it promotes the validation of the database by linking the project database to the model using the dRofus® plug-in within Autodesk Revit® (see Figure 3.6). In addition, the
database can be leveraged for future projects of similar types, since the fields and typology are custom created for each project. After closely following the data manager and design team, I recorded the current practices of analyzing, populating and managing design requirements data for Project A, C, and D. The workflow was captured after spending 170+ hours populating data in dRofus® as well as constant interaction and feedback from the project design teams. Figure 3.7 illustrates the current practices of design data management as investigated in the large architectural firm.

Figure 3.6 dRofus® plug-in within Autodesk Revit®
Figure 3.7 Current dRofus® requirements management process
Investigating the current practices of design data management, revealed the following steps were taken to manage, administer and populate a project’s database:

1. The SOR is analyzed by the data manager expert, to recognize the client’s requirements and build the graphic user interface (GUI) for the database. The GUI is then customized to the projects needs by altering the fields and the typology of structure the data will take. Figure 3.8 shows filters such as drop-down menus, checkboxes, alpha or numeric restricted fields applied to expedite the population process and reduce errors. These filters could be applied for requirements such as temperature, humidity, window frame type, or finish, where there are a few possibilities of data entry.

![Figure 3.8 dRofus® Graphic User Interface (GUI) indicating checkboxes and drop-down menus used to avoid error](image)

2. Documents containing design requirements such as the SOR and functional program are carefully analyzed by a project data administrator to populate data that that is presumed
critical into the dRofus® database (Figure 3.9). The format of these documents varies significantly depending on the project. These documents may be a standard pdf, scanned pdf, interactive pdf, or excel spreadsheet; each instance affecting the entirety of the workflow considerably. As shown in Figure 3.10 and 3.11, the type of files received from owners for this project were in the form of Excel spreadsheets and scanned PDFs.

Figure 3.9 Illustration of data from a functional program being populated in their respective field in dRofus®
Figure 3.10 Sample design requirements data received from an owner in an Excel spreadsheet

Figure 3.11 Sample design requirements data received from an owner in a scanned PDF
3. A pdf report referred to as Room Data Sheet (RDS) can be generated with multiple options on what is to be included in the document as shown in Figure 3.12. Figure 3.13 shows the document that is generated and sent to the various disciplines to confirm the accuracy of the information.

Figure 3.12 dRofus interface to generate room data sheet
<table>
<thead>
<tr>
<th>Room Design Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupancy</td>
</tr>
<tr>
<td>Patients</td>
</tr>
<tr>
<td>Staff</td>
</tr>
<tr>
<td>Visitors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Room Design Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
</tr>
<tr>
<td>Notes</td>
</tr>
<tr>
<td>Furniture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Room Design - Door &amp; Window Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Exterior Window Required</td>
</tr>
<tr>
<td>Window Coverings</td>
</tr>
<tr>
<td>Window Shades</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Room Finishes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floor Finishes</td>
</tr>
<tr>
<td>Ceiling Finishes</td>
</tr>
<tr>
<td>Wall Finishes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plumbing Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water supply</td>
</tr>
<tr>
<td>Water Temperature</td>
</tr>
<tr>
<td>Water supply - hot water</td>
</tr>
<tr>
<td>Water supply - cold water</td>
</tr>
<tr>
<td>Fire Protection</td>
</tr>
<tr>
<td>Wet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HVAC Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heating and cooling</td>
</tr>
<tr>
<td>Temperature Range</td>
</tr>
<tr>
<td>Ventilation</td>
</tr>
<tr>
<td>OSA 2017.02 Class</td>
</tr>
<tr>
<td>Controls</td>
</tr>
<tr>
<td>Relative Humidity</td>
</tr>
<tr>
<td>Relative Pressure</td>
</tr>
<tr>
<td>Exhaust</td>
</tr>
<tr>
<td>Notes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Electrical Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
</tr>
<tr>
<td>Regular Power</td>
</tr>
<tr>
<td>Other supply</td>
</tr>
<tr>
<td>Lighting</td>
</tr>
<tr>
<td>Lighting Type</td>
</tr>
<tr>
<td>Moderate</td>
</tr>
<tr>
<td>Zone divided lighting</td>
</tr>
<tr>
<td>Special lighting</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IMIT System Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications</td>
</tr>
<tr>
<td>Data</td>
</tr>
<tr>
<td>TV</td>
</tr>
<tr>
<td>Overhead Paging</td>
</tr>
</tbody>
</table>

---

**Figure 3.13 Room data sheet of ‘Waiting Room’ in Project A**

4. Mark-ups are issued by the disciplines, after manually reading the report and detecting errors. In addition, markups are made due to multiple changes being made throughout the
primary stages of design. These markups are handwritten and scanned before they are sent to the data manager. Figure 3.14 illustrates a sample mark-up.

Figure 3.14 Sample scanned mark-up document
5. The project data administrator makes changes to the database according to the mark-ups received. This activity is reiterated until all the team disciplines find the database sufficient. In addition, the data manager is notified each time a new addendum is issued so changes are made to the database accordingly. Figure 3.15 illustrates a sample addendum.

2. MODIFICATIONS TO APPENDIX I (INITIAL DRAFT DESIGN-BUILD AGREEMENT)

(a) REPLACE the Statement of Requirements dated August 2, 2017 with the version dated August 16, 2017 and placed in the Data Room.

(b) REPLACE the Functional Program dated August 2, 2017 with the version dated August 15, 2017 and placed in the Data Room.

(c) REPLACE Appendix 1E [Equipment List] dated August 2, 2017 with the version dated August 15, 2017 and placed in the Data Room.

2.1 APPENDIX 1B [ROOM DATA SHEETS]

(a) DELETE the A/V checkbox requirement for the following rooms:
   (1) 1.02 Survery
   (2) 1.14 Book Room
   (3) 2.08 Gender Neutral Make-up/ Dressing Rm
   (4) 2.34 Graphic Arts
   (5) 3.03 P.E/Health Classroom
   (6) 3.05 PHE Office
   (7) 4.01 Wood Shop
   (8) 4.06 Engineering Lab
   (9) 4.10 CTE Classroom
   (10) 4.11 Discovery Shop
   (11) 4.31 Home Economics Classroom
   (12) 4.42 Culinary Arts Classroom
   (13) 6.01 General Classrooms
   (14) 6.02 General Classrooms/Computer Lab
   (15) 6.03 Inclusive ED Learning Support CR (in LNs)
   (16) 7.11 Inclusive ED Classroom
   (17) 8.07 Small Maintenance/IT Break Room
   (18) 8.11 Meeting Room

(b) DELETE the 180m² in room 4.41 Teaching Kitchen and REPLACE with 160 m².

Figure 3.15 Sample addendum issued
6. The dRofus® database is synced to the Revit® model by using the dRofus® plug-in as a final step for completion and validating consistency.

3.3.3 Limitations of Current Space Programming Data Representation

Although effort had been made to establish a comprehensive database to help designers access data easily, there were limitations to the current workflow of representing data. There were four significant limitations:


During the initial five months of the study, 170+ hours was spent populating Project A’s requirements data. The recorded workflow revealed that the largest portion of that time was spent manually analyzing the client’s requirements document and manually searching for requirements information that were presumed critical to populate in dRofus®. This was an onerous process, where I searched for requirements in scanned pdfs, interactive pdfs, and excel spreadsheets, and then typed or copied them into the database.

2. Data Dumping

A data dump is the transfer of a large amount of data from one location to another. In the case of the studied projects, data was dumped from SOR’s, addendums, and functional programs to the dRofus® databases. In order to indicate the critical problems that arose from data dumping, the database of the ‘Drama Classroom’ from Project A will be used:

(i) Redundant Data

Redundant data made for a poor database that was cluttered with information. There were numerous instances of data repetition throughout the database. For instance, as illustrated in Figure 3.16 the location information of a room is duplicated. This may cause ambiguity as well as information being missed as it is unappealing to a designer to read.
Figure 3.16 Redundant data in the dRofus® project database
(ii) *Populating Data in the Wrong Fields*

Data failing to be populated in the proper fields was a critical issue. As shown in Figure 3.17, the ‘Sound’ requirement of the ‘Drama Classroom’ is unavailable. However, in the special requirements fields along with other unorganized data there is a specification referring to an external document. In addition, this room has multiple room access requirements as indicated in the ‘Special Requirements’ field of Figure 3.17. However, it is similarly in the wrong field and not populated in its respective ‘Room Access’ field. This would result in either the designer assuming there are no access and sound criterions that restrict the room from being located anywhere during space programming, or the designer going through the entire database looking for space programming information if he/she is aware of the database’s limitations affecting the time and/or accuracy.
Figure 3.17 Data missing in the respective field of the database
(iii) Referencing External Documents

As shown in Figure 3.18, external references were common in the database. There are several limitations to this, such as: (i) the designer may not have access to this document, (ii) it causes a disruption to the space programming workflow until this information is received from the data manager, and (iii) if not received immediately, it may result in the designer overlooking this requirement. Furthermore, a database such as dRofus® should be independent and refrain from requiring a user to look elsewhere for information if it is to be used as a pillar of information exchange in a firm.
Figure 3.18 External documents referenced in dRofus®
3. Mark-ups

Although mark-ups are common for a project in the preliminary design stage, the mark-ups received by the data manager were highly unorganized, and unclear resulting in a significant amount of time to make changes to the database. Project C alone had 840 scanned pages of handwritten mark-ups. Apart from handwritten information being flawed for a form of communication, the data administrator would be required to go through the entire report to find where the mark-ups are located since there was no guide that showed where changes were made.

4. Validating the Database

Once the project is ready for tender, the database must be validated to ensure it matches the design requirements received from the client. Due to the natural evolution of changes during the design process and different designs being proposed, the client’s requirements evolve although the requirements documentation is not updated. These changes are recorded “in the memory of the participants, and in the best case in meeting or personal notes” (Kiviniemi, 2005), breaking a flow of information to the data manager since an addendum or mark-up is not produced. Since the data manager is aware of this possibility, a final validation of data is accomplished by manually checking each field of each room in the project database to the requirements documents. This process took me nearly half the time as the initial population of the database.

3.3.4 Development of dRofus Database for Project B

Although Project B had already been completed, in order to investigate the time and process of developing a dRofus graphic user interface (GUI) from scratch, I built a simple
dRofus requirements database consisting of basic information for Project B, by analyzing the following documents:

- Statement of Requirements
- Functional Program
- Space Program List
- 3D Models
- Space Programming Requirements Data

These documents were used to create a database consisting of the following information:

- Departments (8)
- Zones (5)
- Rooms (310)
  - Designed Area
  - Programmed Area
  - Perimeter
  - Ceiling Height
  - Occupancy
  - Proximity/Adjacency Requirements
  - Location/Access Requirements
  - Heating and Cooling
  - Temperature Summer/Winter
  - Relative Humidity
  - Moisture
  - Filtration
- Surveillance
- Link to External Documents

**Figure 3.19** Project B dRofus database welcome screen
Figure 3.20 Project B dRofus database - room list

Figure 3.21 Project B dRofus database - space programming requirements
The purpose behind creating this database was to mainly investigate the effort required to improve the limitations that were raised with the current practice of space programming requirements representation observed in Project A, C and D. Issues such as room duplication, data redundancy, population of data in the wrong fields and referencing external documents were avoided by initiating the following steps:

- Creating respective fields for critical information such as location, adjacency, and access requirements. This eliminates from these requirements being dumped into a general field called “Special Requirements” and allows for a designer to easily find the specific requirement they are looking for.

- Leveraging the ‘documents’ option in dRofus to add documents that have been referenced in the database, so a user can easily access it within the tool without having to look for the document externally (See Figure 3.22).
• Linking the project’s dRofus database to the Revit® model to avoid any room duplication or deficit. When this feature is used throughout the design process, a warning notification is shown if any of the rooms that exist in the Revit model do not exist in the dRofus database or vice versa.

To conclude, although Project B’s dRofus database was not as complex as the other databases of Project A, C, and D, the population process was more streamlined and efficient as there were not many questions that were raised on where to populate certain data requirements. The entire process took 21 hours including the manual analysis of the requirements documents and population in dRofus. However, the same amount of time spent on Project A, C, and D could not produce the same result due to the ‘data-dumping’ approach. Moreover, Project B’s database is more user-friendly making it more efficient and less time consuming for a designer to extract information from.
3.4 Current Practices: Space Programming Process and Compliance

Assessment

For this research, focus was placed on the location, access, adjacency, visibility, daylight and acoustics data, to leverage it for space programming. Some functional programs may include a location and adjacency diagram (Figure 3.23), as did Project A. However, this does not entirely capture the requirements written in text and is not sufficient enough to be relied on for space programming. Figure 3.24 illustrated the workflow I recorded by interviewing and shadowing the designers during the space programming stage of Project A.

Figure 3.23 Project a location and adjacency diagram in functional program
Figure 3.24 Current space program workflow as identified in the architectural firm
**Iteration 1**

The first set of iterations of the space programming process begins with the designers reading the SOR so they are familiar with the client’s requirements documentation and producing bubble diagrams. These bubble diagrams are drawn based on Zones and Departments, and the Rooms that are confined in them. Figure 3.25 and Figure 3.26 shows an early meeting that took place between designers to brainstorm the approximate location of departments and zones.

![Designer’s meeting to deliberate space programming during early stages of design](image)

*Figure 3.25 Designer’s meeting to deliberate space programming during early stages of design*
Iteration 2

In the second stage of iterations, the designers read the SOR again, this time analyzing the requirements to prioritize the critical ones. Once the designers sense that they have a good grasp on these criteria’s, the rooms are designed based on the function of the room, in addition to requirements such as daylight and acoustics (sound privacy). Daylight affects where a room is located and its orientation, depending on whether it requires direct or indirect sunlight. For instance, in Project A this indicated that a room requiring direct daylight such as the Gymnasium could not be placed in the center of the project site where other buildings may shelter it from daylight. In the case of acoustics, this meant that rooms such as the Metal Fabrication Shop, and
Discovery Shop where sound producing activities, such as carpentry and welding take place, could not be located near classrooms or the library.

*Iteration 3*

During this stage, configurations are made to the space program, until it is in accordance with the project’s room interrelationship requirements. This is accomplished by constantly referring to the SOR or based on what is recalled from reviewing the SOR previously and rearranging the layout until the designer feels they are compliant with the requirements. The four criterions considered during this process are:

1. **Location**: A requirement for a room to be located in the proximity of another room limited by a certain distance.
2. **Adjacency**: A requirement for a room to either share a common wall, or vertex with another room.
3. **Access**: A requirement for a room to have direct access to another room; i.e. Share a door for access
4. **Visibility**: The terminology of this requirement is vague and may indicate the requirement of a window between the two rooms for supervision purposes or may indicate a criterion of positioning a room, so it is in visible sight from another room without any obstructions.
Iteration 4

Throughout this process the space program is not compliant with all the requirements, since conflicts between requirements are common (Kiviniemi, 2005). Conflict in requirements will result in a chain reaction where, aiming to comply with one requirement will affect the non-compliance of many other functions (Sang Min Park, 2004). This requires the project team to prioritize the critical requirements and make trade-offs based on previous experience. To aid with the decision-making, the project team develops multiple pros and cons list of the current space program making rearrangements until it is as close to compliant as possible with the least amount of ‘cons’. Figure 3.27, 3.28, 3.29 and 3.30 show the team making visualizations by posting project floor plans and pages from the SOR on a wall. The time spent on this process for Project A was approximately 2 days. If the team finds the space program compliant this is the last phase of iterations and the space program process is complete. However, if the project team does not find the space program compliant, the iterations are repeated beginning with the second stage of iterations.
Figure 3.27 Space program compliance validation attempt by project team (i)
Figure 3.28 Space program compliance validation attempt by project team (ii)
Figure 3.29 Space program compliance validation attempt by project team (iii)
Figure 3.30 Space program compliance validation attempt by project team (iv)
3.4.1 Limitations of the Current Space Programming Compliance Assessment

The following space programming limitations were identified during the case study that was performed on Project A:

1. Due to the time constraint, the designers rely on their memory of the SOR to avoid repeatedly checking the requirements. However, this jeopardizes the accuracy of their decisions and the compliance of the design that is produced.

2. Validation of the compliance in the final stage of design before submitting it for tender is determined by the senior designers and their experience. This is a limitation as each project is unique and may have critical requirements that change the design drastically.

3. Multiple iterations are performed to get the space program to comply with all the requirements. However, this is a tedious process and heavily dependent on a trial and error approach. Meaning that rearranging the space program for the compliance of one requirement may result in two or more requirements that were previously compliant to be incompliant. This is because a change in one function affects many more similar to a chain reaction, as this is the nature of the trial and error approach.

4. In interviews that will be discussed in Chapter 4, the designers that were interviewed said on average 10-20% of the design time is consumed by space programming. This will make a large contribution towards the project cost, resulting in not only time inefficiency but high cost as well.
3.5 Conclusion

In this chapter, the case study that was performed on Project A, Project B, Project C, and Project D were discussed, in addition to their project information. The current practices of design data management and space programming were discussed to give detailed background before introducing the optimizations that were developed to improve their limitations in the next chapter.

The next chapter will attempt to address the challenges mentioned in Table 3-1 of the current space programming compliance assessment.

<table>
<thead>
<tr>
<th>Challenges Recorded from Current Practice</th>
<th>Approach to Address Challenges</th>
<th>Sub-Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designers relying on their memory of the SOR during space programming</td>
<td>Development of Smart Excel Template to Parse Client’s Space Programming Requirements Data</td>
<td>4.2</td>
</tr>
<tr>
<td>Assessing space program compliance by seeking advice from senior designer’s experience</td>
<td>Development of Space Program Requirements Floor Plan Visualization</td>
<td>4.4</td>
</tr>
<tr>
<td>Inefficient manual process of space program compliance assessment</td>
<td>Development of Space Program Requirements Floor Plan Visualization</td>
<td>4.4</td>
</tr>
<tr>
<td>Minimizing project cost by reducing significant time contribution towards space program requirements analysis and assessment</td>
<td>Development of Space Program Requirements Dashboard Visualization</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Table 3-1 Challenges recorded from current practices of space programming and the methods adopted to address them
Chapter 4: A Computational Approach for the Visualization of Space Program Requirements

4.1 Approach

This chapter will discuss the developments that were made in an effort to support the compliance of a building project’s space program with the client’s requirements. Following the research conducted at the architectural firm, it was evident that the current workflow required enhancements to make the space programming process less tedious and to avoid the trial and error driven approach for designers. In this chapter, I will discuss three developments that will optimize the current space programming workflow by leveraging the clients design requirements.

Primarily, I will discuss the method that was used to develop a smart excel template to structure and parse the clients space programming requirements data. This is essential to extract significant information such as the name of the rooms that have a proximity relationship requirement. Next, I will discuss how this data was used to develop an interactive space program requirements data visualization dashboard. Finally, I discuss the development of a computational tool script used to validate the compliance of a building’s space program requirements through a floorplan overlay.
4.2 Development of Smart Excel Template to Parse Client’s Space Programming Requirements Data

In order to automate the inefficient manual process of analyzing space programming requirements for the evaluation of space program compliance, I developed a smart Microsoft Excel® template that parses and structures space program requirements.

As a first step in this automation, space programming information needed to be exported from dRofus® (Figure 3.1) by generating an Excel spreadsheet. The generated excel spreadsheet contains the room name, department, ‘Location and Adjacency’ and ‘Special Requirements’ field in dRofus® (Figure 4.1). The ‘Location and Adjacency’ and ‘Special Requirements’ fields were where the space programming requirements data would be dumped along with other uncategorized requirements (Figure 4.2). Extracting significant space programming requirements from these fields were essential to this research. A formula was devised for a smart excel template through trial and error, to parse the space program requirements data. This would help a designer identify the space programming requirements they are looking for, without having to read fields that would commonly contain about 150 words, as shown in Figure 4.2. This data also served as the significant piece to develop the space programming dashboard and floor plan overlay.
Figure 4.1 Exporting excel spreadsheet from dRofus®

Figure 4.2 Space programming requirements data dumped into dRofus® fields
### Figure 4.3 Generated excel spreadsheet from dRofus

<table>
<thead>
<tr>
<th>Room Name</th>
<th>Space Requirements (Multi Box edit)</th>
<th>Location</th>
<th>Room Relationship 1</th>
<th>Room Relationship 2</th>
<th>Room Relationship 3</th>
<th>Accuracy</th>
<th>Reason For Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theatre Ho</td>
<td>Choir, Theatre,Receiving</td>
<td></td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.5.2: Stan and Chair: to be located adjacent to the back of the house</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Office, House</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Buddy Room 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dance Sto</td>
<td>Dance Studio</td>
<td></td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Theatre</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Office and Studio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General Sto</td>
<td>General Storage</td>
<td></td>
<td>TRUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>calming 4 to 600 students</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Requirements data exported from dRofus

### Figure 4.4 Excel spreadsheet structured for parsing

- Columns containing formula, that extract room relationships
Figure 4.3 illustrates the Microsoft Excel® spreadsheet generated from dRofus® and Figure 4.4 illustrates the same spreadsheet transformed into a smart-excel template to structure and parse space programming requirements. This spreadsheet contains 4 workbooks each assigned to Location, Adjacency, Access, and Visibility. This spreadsheet was developed to automate the parsing effort of the requirements by developing excel formulas for column E, F, G, and H. The spreadsheet was prepared with each column representing the following components:

- Column A and B are a duplicate of column C which is the ‘Master Room List’ exported from dRofus® and is synced to the architectural Revit® model nomenclature. Column A and B were created because column F, G, and H required different columns to conduct the parsing on, since a room may have up to three relationships with other rooms.

- Column D contains the merged data from ‘Location and Adjacency’ and ‘Special Requirements’ fields in dRofus®. They were merged since they both contain space program requirements data.

- Column E represents the type of space program requirement being parsed (location, adjacency, access or visibility). This is the only column whose formula varies in each workbook for the specific space program requirements. Figure 4.5 shows that the formula for column E varies with each space program requirement. These variances of ‘flag words’ were identified due to familiarity with the project’s SOR and identifying a pattern of words used to describe each requirement based on the consistency and repetition. This formula works by checking whether the flag words are contained in Column C. If these ‘flag words’ are contained in Column C then a ‘True’ value is returned and if not, it will return a ‘False’ value. Consequently, the drop-down filter
applied to the column will be used to filter the results that have a ‘True’ value as shown in Figure 4.4. The design languages or ‘flag words’ used for the space requirements were:

- **Location**: ‘located’, ‘locate’, ‘near’
- **Adjacency**: ‘adjacent’, ‘adjacency’
- **Access**: ‘direct access’, ‘within’, ‘connected’
- **Visibility**: ‘visible’, ‘visibility’

- Column F, G, and H are the significant columns that indicate the room names that the room specified in column C has a relationship with. Each cell in Column F, G, and H analyzes the space requirements in column D and then matches them to the room names in either column A, B, and C correspondingly, returning the matched values as a result (Figure 4.6). These columns were limited to three for this case study since analyzing the SOR revealed that there was no room that had more than 3 relationships with other rooms.

- Column I was manually populated by manually checking the accuracy of the spreadsheet against the SOR and checking whether the spreadsheet returned all the room relationships accurately. If any rooms were not identified by this tool the value ‘False’ was populated to indicate it was not accurate and that there is a missing requirement.

- Column J indicates why the tool was unable to identify the relationship if column I has a ‘False’ value to record the reason for error. This statistic was later used in Microsoft PowerBI® dashboard to present to the potential users of this smart excel template the limitation and why it had occurred (see Figure 4.10).
Figure 4.5 Formula used to flag room relationship requirements

<table>
<thead>
<tr>
<th>Room Name</th>
<th>Space Requirements (Multi line edit)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Band Room</td>
<td>Choral, Theatre, Receiving 2.1.6.2 Band and Choir rooms shall be located adjacent to each other to accommodate shared use of Music Office, Music Storage, Practice Rooms, and Recording Studio.</td>
<td></td>
</tr>
<tr>
<td>Accessible Toilet &amp; Shower</td>
<td>Accessible washroom with barrier-free shower, folding changing table. Restricted sightlines for privacy (e.g. off secondary corridor with blank wall opposite). Shower and toilet privacy curtains.</td>
<td></td>
</tr>
</tbody>
</table>

=SUMPRODUCT(COUNTIF(E4, "**adjacent**", **adjacency**))>0

<table>
<thead>
<tr>
<th>Room Name</th>
<th>Space Requirements (Multi line edit)</th>
<th>Adjacency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Toilet &amp; Shower</td>
<td>Accessible washroom with barrier-free shower, folding changing table. Restricted sightlines for privacy (e.g. off secondary corridor with blank wall opposite). Shower and toilet privacy curtains.</td>
<td>Access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=SUMPRODUCT(COUNTIF(E4, &quot;<strong>adjacent</strong>&quot;, <strong>adjacency</strong>))&gt;0</td>
</tr>
</tbody>
</table>

=SUMPRODUCT(COUNTIF(E4, "**direct access**", **within**", **connected**")>0

<table>
<thead>
<tr>
<th>Room Name</th>
<th>Space Requirements (Multi line edit)</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Toilet &amp; Shower</td>
<td>Accessible washroom with barrier-free shower, folding changing table. Restricted sightlines for privacy (e.g. off secondary corridor with blank wall opposite). Shower and toilet privacy curtains.</td>
<td>Access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=SUMPRODUCT(COUNTIF(E4, &quot;<strong>direct access</strong>&quot;, <strong>within</strong>&quot;, <strong>connected</strong>&quot;)&gt;0</td>
</tr>
</tbody>
</table>

=SUMPRODUCT(COUNTIF(E4, "**visible**", **visibility**")>0

<table>
<thead>
<tr>
<th>Room Name</th>
<th>Space Requirements (Multi line edit)</th>
<th>Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Toilet &amp; Shower</td>
<td>Accessible washroom with barrier-free shower, folding changing table. Restricted sightlines for privacy (e.g. off secondary corridor with blank wall opposite). Shower and toilet privacy curtains.</td>
<td>Visibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>=SUMPRODUCT(COUNTIF(E4, &quot;<strong>visible</strong>&quot;, **visibility&quot;)&gt;0</td>
</tr>
</tbody>
</table>
Figure 4.6 Formulae used to parse space program requirements
Although this smart Microsoft Excel® template was developed specifically for Project A, it was proposed for generalized use on future projects at the firm-requiring only minimal manipulation. To use this template for future projects, only two variations would need to be made:

- The requirements data exported from dRofus® (column C and D) would be replaced with requirements from the project requiring parsing resulting in new results.

- All formulae used within the spreadsheet would stay consistent, other than the one used in column E, which contains the words used to flag proximity requirements between rooms such as ‘located, near, connected to’ etc. It is highly likely that these words may not be changed as they are common words used to frame proximity requirements. However, to increase accuracy these words may be updated with simplicity if alternative flag words are identified following familiarity with the project’s SOR.

However, it is important to note that not all projects may require the analysis provided by this smart-excel template. The parsing of space programming requirements may only be beneficial for projects with complex space program requirements, typically large hospital, institutional and laboratory projects.
4.3 Development of Space Program Requirements Dashboard Visualization

Once the space program requirements data were identified by the smart-excel template, a dashboard was developed for the ‘Location’, ‘Adjacency’, ‘Access’ and ‘Visibility’ requirements using Microsoft PowerBI®. This was developed to provide designers with an interactive, searchable dashboard that will provide them with a quick room interrelationships visualization for space programming requirements. Figure 4.7 shows the ‘Location’ space program requirements, visualized by the ‘Force-Directed Graph’ in conjunction with a ‘Text Filter’ feature in PowerBI®. When a user enters a search word for query, the graph filters down to all the rooms containing that search word along with the relationships it has with other rooms indicating a proximity requirement. The proximity requirement between two rooms are indicated when they are connected by a line as shown in Figure 4.8.
Figure 4.7 Searchable dashboard for location relationship requirements (a)
Figure 4.8 Searchable dashboard for location relationship requirements (b)
The dashboard was also used to indicate the accuracy of the parsing implemented in the smart excel template. The spreadsheet was analyzed manually to develop Column I and J (see Figure 4.4), which are the ‘Accuracy’ and ‘Reason for Error’ column correspondingly.

Developing these two columns were critical to understand the limitations of the parsing efforts and avoid mistakes by relying on the dashboard. Figure 4.10 shows the dashboard reflecting the number and percentage of correctly and incorrectly identified requirements. In addition, out of the percentage that were not identified, a donut chart is used to indicate the reason why these requirements were not identified. The spreadsheet failed to recognize requirements due to three reasons:

i) Inconsistency of room name nomenclature; a room existing under different names such as “Gymnasium” and “Gym” or misspelling a room name in the SOR.

ii) Unidentified flag words such as “close to” for ‘Location’ requirements and “passive supervision” for ‘Visibility’ requirements were not included when developing the formula initially. This was later added to reflect a higher accuracy.

<table>
<thead>
<tr>
<th>Location</th>
<th>Adjacency</th>
<th>Access</th>
<th>Visibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate</td>
<td>Adjacent</td>
<td>Direct Access</td>
<td>Visible</td>
</tr>
<tr>
<td>Located</td>
<td>Adjacency</td>
<td>Within</td>
<td>Visibility</td>
</tr>
<tr>
<td>Near</td>
<td>Positioned</td>
<td>Connected</td>
<td>Positioned for Supervision</td>
</tr>
<tr>
<td>Close to</td>
<td></td>
<td>Integrated into</td>
<td>Passive Supervision</td>
</tr>
</tbody>
</table>

Table 4-1 Words used in the formula to flag relationships between rooms
iii) Mistaking the mention of a room name as a relationship requirement. In the SOR, a “light trap vestibule” was required for ‘Theatre’ room. However, the project contained rooms under the same name, resulting in the spreadsheet mistaking this as a relationship between the ‘Theatre’ and ‘Light trap vestibule’.

Figure 4.9 Theatre space program requirement from SOR

Figure 4.10 ‘Location’ space program requirement accuracy statistics
The overall accuracy statistics of all the requirements are shown in Table 4-2. Once these incongruities were identified the spreadsheet was revised by fixing these errors manually to reflect 100% accuracy.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Correctly Identified</th>
<th>Unidentified Or Mistaken</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>366</td>
<td>20</td>
<td>94.8</td>
</tr>
<tr>
<td>Adjacency</td>
<td>376</td>
<td>10</td>
<td>97.4</td>
</tr>
<tr>
<td>Access</td>
<td>382</td>
<td>4</td>
<td>98.9</td>
</tr>
<tr>
<td>Visibility</td>
<td>386</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4-2 Space program requirements data parsing efforts accuracy statistics

4.4 Development of Space Program Requirements Floor Plan Visualization

This section aims to investigate the extent to which a computational tool in conjunction with BIM, could be used to analyze the compliance of a space program’s requirements with a semi-automatic workflow. This was implemented using the BIM design tool Autodesk Revit® along with Autodesk Dynamo® a visual computational tool. Two visualization methods were developed for the space program requirements, according to the type of requirements. The ‘Daylight’ and ‘Acoustics’ requirements were visualized using color schemes to represent the different requirements. Consequently, the requirements such as ‘Location’, ‘Adjacency’, and ‘Access’ which indicate proximity relationships between rooms, were identified by drawing lines between room centroids.
4.4.1 Daylight

Daylight affects where a room is located and its orientation, depending on whether it requires direct or indirect sunlight. For instance, in Project A, this indicated that a room requiring direct daylight, such as the Gymnasium, could not be placed in the center of the project site where other buildings may be an obstruction from direct daylight. The daylight space program requirement has three possibilities as follows:

i. **Direct Daylight**: There are no obstructions such as a room or building structure shadowing a room from natural daylight. This requirement is indicated in yellow.

ii. **Indirect Daylight**: There are no preferences for the placement of the room and can be placed anywhere according to daylight requirements. This requirement is indicated in orange.

iii. **No Daylight Requirements**: No requirement was found for this room in the SOR. This requirement is indicated in grey.

Figure 4.11-4.14 show the Daylight requirements visualizations as on overlay on the Revit® architectural floor plan produced by the Dynamo® script for Project A.
Figure 4.11 Project A first-floor daylight requirements compliance test

Figure 4.12 Project A second-floor daylight requirements compliance test
Figure 4.13 Project A third-floor daylight requirements visualization

Figure 4.14 Project A fourth-floor daylight requirements visualization
According to this compliance validation method, non-compliant rooms can be seen in the third and fourth floor of Figure 4.13 and 4.14. This is because they are placed in the center of the project site where they are obstructed from direct daylight. This provides a simple visualization for the designer to evaluate the design and make changes. On the contrary, there may be a compromise being made and changes may not be made. This is when there is a requirement clash that requires for a room to have direct daylight and simultaneously requires for it to be near or accessible to one or multiple rooms. Usually the compromise made by designers for such instances is the latter, where the interrelationship between rooms outweighs daylight preferences.

### 4.4.2 Acoustics

The acoustics space program requirement refers to a room placement according to sound and noise ratings. This means that rooms in Project A, such as the Metal Fabrication Shop, and Discovery Shop that produce high noise due to activities such as carpentry and welding cannot be located near rooms such as classrooms or the library.

The acoustics space program requirement has four possibilities as follows:

i. **Normal Speech Privacy**: ‘Normal’ refers to rooms that require moderate noise privacy such as classrooms and offices. This requirement is indicated with blue shading.

ii. **Speech Privacy**: Rooms such as wood shops, band rooms, and metal shops that produce loud noises and cannot be placed near other rooms that require low acoustics such as libraries and classrooms. This requirement is indicated with orange shading.

iii. **No Privacy**: Rooms such as gymnasiums and cafeterias that do not require any speech privacy. This requirement is indicated with green shading.
iv. **No Requirements**: No requirement was found for this room in the SOR. This requirement is indicated with grey shading.

Figure 4.15-4.18 show the Acoustic requirements visualizations as on overlay on the Revit® architectural floor plan produced by the Dynamo® script for Project A.
Figure 4.15 Project A first-floor acoustics requirements compliance test

Figure 4.16 Project A second-floor acoustics requirements compliance test
Figure 4.17 Project A third-floor acoustics requirements compliance test

Figure 4.18 Project A fourth-floor acoustics requirements compliance test
According to this compliance validation method, there are two non-compliant rooms for acoustics requirements on the first and fourth floor of Project A (see Figure 4.15 and Figure 4.18). The rooms indicated in orange shading exert noise that the rooms indicated in blue would be undesirably affected by. As mentioned earlier, there may be a compromise being made and changes may not be made due to a requirement clash forcing the client to compromise.

Figure 4.19 illustrates the script developed in Dynamo® to validate the space program requirements for Daylight and Acoustics space program requirements.
Figure 4.19 Dynamo® script for daylight and acoustics space program compliance validation
The script is divided into three major sections:

1. The architectural Revit® model geometry is exported into the Dynamo® workspace. Next, the room geometry and room names are filtered by level, so the designer can evaluate the space program visualization one level at a time.

2. Data such as room names and location/adjacency requirement are imported from the Excel Spreadsheet and sorted to match the indices of the room name list imported from Revit®. This substantiates that correct requirement is applied to the corresponding room.

3. Lastly, each color group in the third set represent each one of the three daylight requirements. For adjacency requirement there was an additional group since adjacency had four possible requirements. Each group filters one requirement such as direct, indirect, or no requirement and applies a color to the room geometry.

For Location, Adjacency, and Access space program requirements a different approach was used to visualize the requirements on the Revit® floor plan. Since, all three of the requirements are proximity relationships between rooms, projection lines were used to validate their space program compliance

4.4.3 Location

This requirement conforms to the location of a room in proximity to another, limited by a certain amount of distance. This distance however is not defined and often referred to by words such as “close to” and “near”. Figure 4.20 shows the first-floor layout for Project A- this was the only floor that had location space program requirements according to the SOR. The black dotted
lines projected onto the floorplan are produced to validate the space program requirements by running the Dynamo® script that was developed, within the Autodesk Revit® architectural model of Project A. Although these lines illustrate the relationships between rooms, it is still up to the designer to decide whether the layout is compliant with the space program or not unlike the adjacency and access requirements. This is because the location space program requirement does not have strict distances provided as a guideline. According to this method of validation, indicated in red is only room that is not compliant with the client’s requirements, as the rooms are clearly far away from one another.
Figure 4.20 Location space program requirement compliance validation - Project A first floor
4.4.4 Adjacency

The adjacency space program requirement entails for a room to either share a common wall, or vertex with another room. Figure 4.21 illustrates the floor plan of Project A with the adjacency space program requirements projected onto it. According to this method, there are two instances of incompliance with the SOR space program requirements. The script was developed to draw the validation lines from one room centroid to another, however the lines circled in red indicate that one side of the line is not attached to a room centroid. This indicates that the rooms with an adjacency requirement have been placed on different floors. These lines will be projected on both floors where these rooms are located and will have one end floating on both floors indicating the incompliance.
Figure 4.21 Adjacency space program requirement compliance validation - Project A first, second and third floor
4.4.5 Access

The space program’s access requirement, means that a room must have direct access to another room, usually conformed by the two rooms sharing a mutual door. Access had the most instances of space program requirements in the SOR and was deemed the most important by the designers. According to the compliance validation workflow developed, there was only one instance where Project A’s floor plan was not compliant with the client’s access space program requirements (Figure 4.22).
Figure 4.22 Access space program requirement compliance validation - Project A first floor
Figure 4.23 Access space program requirement compliance validation - Project A second floor

Figure 4.24 Access space program requirement compliance validation – Project A fourth floor
Figure 4.25 Dynamo® script for location, adjacency and access space program compliance validation
Figure 4.25 illustrates the Dynamo® script used to visualize location, adjacency, and access space program requirements. This script is divided into 11 main components as divided in Figure 4.25:

1. Deletes any previously placed lines if the script has been run before. This constrains the script from producing duplicate lines resulting in ambiguity.
2. The parsed space program requirements data are imported, and the inessential columns are removed.
3. The geometry of all the rooms in the project are identified in the form of a bounding box. Consequently, the bounding box is used to identify the centroid of each room. Next, room parameters such as room names, and levels are identified and sorted to assign the correct room name and level to each centroid that was identified. This step develops the ‘Master Room List’, ‘Master Point List’, and ‘Master Level List’ to serve as the correct indices of rooms with their subsequent centroids and levels as changing their order of indices would create drastic error.
4. The room relationship columns are extracted from the excel spreadsheet. This would be column F, G, and H (see Figure 4.26 below)
5. The null values in the room relationship column are removed.
6. The room relationships containing a value are then matched to the indices of the rooms in Step 3 and assigned the subsequent centroid and levels.
7. Step 6 is repeated for column C (see Figure 4.26 below).
8. The centroids of the rooms in column C are connected to the centroids of the rooms they have a relationship with; which are the names indicated in column F, G and H.
The lines drawn at this stage will only show as a projection on top of the floor plan and aren’t not interactive like model or detail lines.

9. This step checks whether the rooms that indicate a relationship are on the same level. If the script returns a ‘False’ result, then the lines are drawn on both levels where each room is located.

10. The lines produced in Step 8 are filtered according to the floor the rooms exist on, otherwise the lines will be visible on each floor causing ambiguity. There are four groups in this step since each group represents one of the four floors in Project A. To avoid designers mistaking the requirement compliance lines with design lines the script places them on a separate sheet created specifically for each requirement of Location, Adjacency and Access. Next, the lines are converted from projection lines into detail lines, so they are interactive, and their color and line weight can be adjusted.

11. Finally, the line color, weight, and type (dashed, dotted etc.) are changed so they can be distinguished from design detail lines.

<table>
<thead>
<tr>
<th>Location</th>
<th>Room Relationship 1</th>
<th>Room Relationship 2</th>
<th>Room Relationship 3</th>
<th>Line Color</th>
<th>Line Weight</th>
<th>Line Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Office</td>
<td>Recreational</td>
<td>Recreational</td>
<td>Recreational</td>
<td>TRUE</td>
<td>Plain</td>
<td>Plain</td>
</tr>
<tr>
<td>Vasco Hall</td>
<td>Vasco Hall</td>
<td>Vasco Hall</td>
<td>Vasco Hall</td>
<td>FALSE</td>
<td>No-Plain</td>
<td>No-Plain</td>
</tr>
<tr>
<td>MainEntry</td>
<td>MainEntry</td>
<td>MainEntry</td>
<td>MainEntry</td>
<td>TRUE</td>
<td>Plain</td>
<td>Plain</td>
</tr>
<tr>
<td>Director's Office</td>
<td>Director's Office</td>
<td>Director's Office</td>
<td>Director's Office</td>
<td>TRUE</td>
<td>Plain</td>
<td>Plain</td>
</tr>
</tbody>
</table>

**Figure 4.26 Smart excel template containing room relationship data**
4.5 Challenges

Manually populating the dRofus® database for Project A, C, and D consumed many hours, and multiple iterations due to the inconsistent formats of design data documents of lower quality such as handwritten and scanned documents. In addition, the deadlines for the project tender resulted in a high work load, this in conjunction with different disciplines in the firm requiring similar data to be structured and formed differently was time consuming.

4.6 Suggestion

The challenges of populating the dRofus® database manually could be overcome by implementing the usage of fillable PDF documents or Microsoft Excel® spreadsheets. The dRofus® interface allows for a smooth import of data from Excel spreadsheets, while fillable PDF documents could be easily converted to excel spreadsheets with ease, for a similar import of data into dRofus®.

4.7 Evaluation of Space Program Visualizations

To explore the likelihood of the architectural firm implementing the developments of this research, an interview was conducted amongst the three architectural designers that had the highest influence on the space program design of Project A. The interviewees vary in design experience ranging from 4-10 years, all of which have worked on highly demanding and complex projects at this large reputable firm. The questions constructed for the interview were:

1. What is the average time spent on space programming during the design phase?

   Interviewee A: “About 10-20% of the total architectural design time”
**Interviewee B:** “Roughly about 10% of the design phase would go to space programming, however due to the iterative nature of space programming the layout is usually configured even towards the end when certain requirements are discovered, or clashes are identified, so perhaps more than 10%”

**Interviewee C:** “It depends on the project, during the schematic design space programming is 50% of the work and that’s about 10-15% of the overall architectural design.”

2. What is the estimated amount of time this workflow could reduce during the preliminary stages of space programming?

**Interviewee A:** “Generally, the space programming requirements you have visualized are missed. This would be an additional quality control measure saving us at least 10-20% of time spent on space programming”

**Interviewee B:** “It’s tough to estimate time that could be saved but depending on the project and requirements complexity, it would surely reduce a portion of the design time”

**Interviewee C:** “It could save a lot of time on project’s where the client’s requirements are complex. For healthcare projects with complex space program requirements it could save about 25% of the time dedicated to design”

3. Could this workflow reduce project cost by reducing the time spent on space programming?

**Interviewee A:** “The project cost wouldn’t be reduced as it would just be utilized for something else, even though we would be spending less time coming up with space layout options and checking the requirements tediously”
**Interviewee B:** “Yes, not only space programming design cost but also the validation of information ahead of time prevents costly design changes in the construction phase. The MacLeamy Curve shows the cost of making design changes during construction is at its peak. These visualizations would help us avoid that.”

**Interviewee C:** “It could reduce project cost if you are diligent about running the visualizations that way you would reduce the risk of errors being brought beyond schematic design where it is costly to make changes.”

4. Sometimes changes cannot be made to a floor plan even if they are incompliant because of requirements clashes in the client’s requirements. Given that, is it likely that you as a designer would make changes to the floor plan of an architectural model due to an incompliance in space program requirements detected by the visualizations?

**Interviewee A:** “Yes, very likely. It’s not just a designer that would benefit from the visualizations but the client as well. When there are clashes with the client’s requirements we would draw layouts and then show them the requirements to show them that it wouldn’t work. Sometimes we would draw sketches in meetings to show them how a certain requirement affects the space layout, but now we can just use these visualizations.”

**Interviewee B:** “Absolutely. These visuals flag and identify the wrong placement of rooms in a floor plan, so design changes would be made based on that.”

**Interviewee C:** “Yes, definitely I would rethink the whole layout. Even if I can’t make changes I would use it to communicate with the client.”

5. What is the expected likelihood of implementing this workflow to support the compliance on future projects?
Interviewee A: “It depends on the project complexity, this could be used on labs, schools and hospitals where the space program requirements are complex. Otherwise it depends on whether the requirements data is available and if it’s worth the effort of populating this data in dRofus® or Excel.”

Interviewee B: “This depends on the project we’re working on however; this entire workflow of structuring space programming requirements data can now be used from a different angle. I have an upcoming project where this firm is developing space programming requirements for the client, for another indicative design firm to comply by. Now that I understand the importance of well-structured space programming data I could follow this template to develop requirements documents that are well-structured for an ease of use.”

Interviewee C: “The likelihood is high depending on the project timeline, the quality of information from the client and the fees to implement the visualization like the time it would take to structure the data. This could really help the client visualize their requirements.”

4.7.1 Interview Results

The interview responses conclude that the developments of this research received a positive response from the designers interviewed. It is evident that not all building projects have complex space programming requirements. The time and cost required to structure the database and manipulate the Dynamo® script could be significant and outweigh the value of the space programming visualizations developed. This con that the developments made during this research must be used on projects with complex space programming requirements in order to gain optimal benefits from the efforts required.
Chapter 5: Conclusions and Future Work

5.1 Summary

In this research, a case study was conducted to investigate the current practices of space program requirements data management, and design workflow in a large architectural firm. By closely following project data managers and designers, I recorded their workflow and analyzed space programming requirements documents. This was implemented to investigate the extent to which computational tools could be used to validate the compliance of a building project’s space program requirements. In order to understand the challenges and develop an optimization method, I first followed the firm’s current practices of design requirements data collection and management to populate the project database. This allowed me to record the challenges and limitations of the workflow to structure qualitative data of space programming requirements. I extracted, analyzed and structured space programming requirements data using a smart excel template I developed in Microsoft Excel®. This data was used to develop a dashboard to visualize space programming requirements and to validate the compliance of a building project’s space programming requirements in conjunction with a visual computational tool. The evaluation and assessment of the space programming requirements was carried out by using Autodesk Revit® and Dynamo® to produce a visual overlay on the architectural model’s floor plan. This development was well received by the designers that this work was presented to at the large scale architectural/engineering firm. It is clear that BIM technology and related tools can play a profound role in the entire process of assessing space programs according to a client’s needs and requirements. The visualizations were produced under the presumption that they would:
• Allow for designers to evaluate project space programs without having to manually analyze client’s requirements.

• Save cumbersome space program design time by eliminating the trial and error process as recorded in this study.

• Eliminate the need for designers to rely on their memory of the space program requirements and jeopardize accuracy and instead run the Dynamo® script for evaluation.

• Save the firm penalty fines that arise from space program incompliance.

5.2 Limitations and Future Work

During this research, datasets from four projects were studied to understand the patterns in space programming requirements and investigate the space programming requirements data management in the firm. However, the space programming script was only implemented on one of the projects due to the time constraints of this study. This makes it difficult to validate the success of the script when evaluating the compliance of space programming requirements on other projects. In addition, there were limitations observed with the developments I made for this research; however, the time constraint of this study did not allow for further optimizations to be made. The following are limitations that that were observed and serve as motivation for future analysis and research:

• The formulas developed for the parsing of the space programming requirements data is limited to three relationships per room, and the extracted information must be manually validated.
• The formulae developed for the spreadsheet is not a universal tool that can be applicable on any project. Manipulations must be made to the formula by analyzing the design language of the project’s unique SOR.

• In this research it was observed that there was inconsistency in room name nomenclature. This will result in the spreadsheet failing to detect a relationship between rooms unless the room names are identical.

• If there are multiple rooms under the same name, the script developed will not be able to identify a relationship for all of them rather for the room that comes first in the indices of the ‘Master Room List’

This research approach attempted to develop a framework that could be implemented to validate the compliance of a building project’s space programming requirements data using unstructured data. For future work, the developments made in this research could benefit from generalizability that would allow for them to be applied in a broad range of projects without extensive manipulation.
Bibliography


Doulgerakis, A. (n.d.).


