MSc thesis
Master Construction Management and Engineering

Semantic enrichment of design requirements using Object Type Libraries for automated verification

Shreenidhi Raghavendra Rao
5000246

TUDelft

Royal HaskoningDHV
Enhancing Society Together

Semantic Enrichment of Design Requirements Using Object Type Libraries for Automated Verification

Shreenidhi Raghavendra Rao - 5000246

Thesis Committee:
Prof. dr. P.W. Chan
Dr. Tong Wang
Dr. Yan Liu
Ir. Hans Hoeber
Ir. Niels Willemsen

Project Duration: January 28, 2021 - August 26, 2021

Royal HaskoningDHV
Enhancing Society Together

TU Delft
Delft University of Technology

Acknowledgement

This report is a culmination of the last seven months of exciting research into the topic of Systems Engineering based requirements verification process. This report has been written to fulfil the graduation requirements for the masters program Construction Management and Engineering. The research was conducted from January to August of year 2021. The research was conducted in collaboration with Transport and Planning business line of Royal HaskoningDHV. This research stems from my motivation about automation in construction industry and it has been a roller coaster ride which was made possible with the support of a few people. I would like to take this space for expressing my deepest gratitude to all these people.

To begin with, I would like to thank my graduation committee for their valuable guidance and constant support throughout the process. I express my deepest gratitude to my first supervisor, Dr. Tong Wang, for helping me to gather my thoughts and refine the concepts of my thesis in order to give it a strong foundation. I would also like to thank my second supervisor Dr. Yan Liu for his valuable insights in developing this report. His meticulous review of every report I sent which ultimately had a huge impact on the overall quality of the final report. I would further like to thank my chair Prof. dr. P. W. Chan for helping with his different perspectives in this field of research and his additional support in setting up the thesis committee. I am grateful to Royal haskoningDHV for having me given the opportunity to conduct my research with them. I will always be indebted to my company supervisor, Ir. Hans Hoeber for his extreme efforts during this research. I am very thankful for his additional time spent on refining the research question, providing insights on Royal HaskoningDHVs practices and providing data whenever needed for the research. I would like to extend my gratitude to Ir. Niels Willemsen for his insights on research process and validation of the proposed framework. A special thanks to all the interviewees and team at Royal HaskoningDHV for their support during the research.

Finally, I would like to thank my parents for huge sacrifices in their lives and making it possible for me study at this prestigious university. I will always be grateful for my friends for their constant support and encouragement.

This research broadened my understanding of information management practices in construction industry and what is possible with the constantly innovating IT industry. I hope you have an informative experience in reading this report.

Shreenidhi Raghavendra Rao
Delft, August 2021

Executive Summary

Introduction

The current research subject stems from the motivation of increasing productivity in construction design process by automating tasks in design phase of the project. Infrastructure design process involves complying with broad range of requirements from building codes and standards, client needs, project needs and industry guidelines. The success of the project depends on the compliance to requirements amongst other things. To accommodate these requirements, Systems Engineering is being adopted as a industry wide practice in the Dutch construction sector. Inconsistency or lack of compliance to certain requirements is unavoidable over the construction phases and overtime it becomes difficult to keep track. The adoption of building information modelling (BIM) enabled information management processes for requirement management and verification claims to improve business performance by reducing the amount of work and errors. This could be achieved by standardising the work processes. However, the major obstacle in automating the verification process is the way requirements are currently written and stored. This thesis works towards providing more efficient workflows in architecture, engineering and construction (AEC) industry by proposing a framework for defining requirements in a machine and human understandable with the use of object type properties from object type libraries (OTL).

Research Design

The research is practice oriented. Research tries to reflect on the existing requirement verification practices and suggest changes and interventions. Double diamond research methodology proposed by Design Council (2019) is adopted in this study. First stage of double diamond process focuses on problem exploration and definition. Problems are identified based on the design process that the Dutch infrastructure sector follows. The problems identified are validated by finding relevant literature and by conducting interviews with practitioners from RoyalHaskoningDHV. Based on the problems identified and solution hypothesis made, a research question is defined to arrive at the final solution. The second phase of the process focuses on solution exploration and development. The process started with identifying several academic and commercial solutions through a rigorous literature search process. Based on various approaches available, a solution is suggested from existing or a potential combination of existing solutions. The solution is then tested on a small infrastructure project to identify feasibility and implementation challenges.

Problem Exploration and Definition

There now exists several problems in the current design and verification process. Few of the important problems identified are: requirements in textual form in Relatics becomes cumbersome and time consuming for designers to find, understand and visualise the needs. The perfect requirements should also exhibit SMART criteria and large number of requirements fall short of these. The various other problems identified are elaborated in Section 3.4. The automation of any data intensive tasks, requires a standardised information stream. Analysis of current design process in Royal HaskoningDHV immediately revealed that this is not the case in current workflow. Several problems with respect to requirements like ambiguous interpretation, lack of mapping with object data, dynamically changing requirements, proved that automating requirements verification process is difficult. Detailed analysis into this problem revealed that poor definition of requirements is the major cause for not being able to link requirements with object data there by, hindering the automation process. A research question was setup to focus on how requirements could be semantically mapped to object instances. The main research question defined was:

"How can we improve the mapping between requirements and BIM data for verification of design's compliance in infrastructure projects?"

The main research question was further sub-divided into four questions for answering the above research question. The formation of research question ends the first phase of the double diamond process.

Solution Exploration and Development

The second phase of the framework development process involved exploring existing solutions to solve the problem of requirements definition. More than 300 studies in the area of construction rule checking domain was identified. However, previous studies focused mainly on specific domains and not on the entire design process. Commercially available tools for verification, focus mainly on checking compliancy with building codes and completeness of BIM model. These tools are static in nature and do not consider the dynamicity of the project specific requirements. Thus, an approach for handling dynamic requirements is necessary. This can increase the value of verification process. Existing academic studies and commercial solution for automated requirement checking relies on expert knowledge of translating requirements into a machine understandable format. This makes the workflow closed since users require high level of expertise in programming, data structures and construction domain knowledge. The project team responsible for design and verification always has to rely on interpretation of the programmer. The study of existing solutions with different technologies revealed a common approach in arriving at a solution. In each of these approaches, requirements are first analysed for which object and property they belong to. Next, requirements are represented using only the object name and relevant property and value that needs to be checked. Additional noise in the form of natural language sentence structure such as prepositions et cetera are removed. Lastly, the rule is analysed for its complexity depending on logical operation that needs to be performed for verification and is classified accordingly. A framework was developed by providing an potential solution to these three steps.

Existing solution provided by Hjelseth and Nisbet (2011), Dimyadi, Solihin, and Eastman (2016) and Solihin and Eastman (2015) was adopted to answer these steps. These solutions, like mentioned earlier focused on building code verification but not on project specific requirements. However, they are promising approaches in rule checking domain and could be adopted to porject specific requirements with slight modifications. Study proposed by Hjelseth and Nisbet (2011) relied on data dictionary for building code as a main source for data classification. Use of domain ontology in the form of object type libraries makes it more flexible for classification. Semantic representation approach using conceptual graphs was proposed by Dimyadi et al. (2016) which recommended defining custom relations for use in domain specific processes. Two new relations are proposed before adapting the approach for semantic representation of project specific requirements. Four rule execution classes proposed by Solihin and Eastman (2015) is adopted directly for defining requirement rule classes. Using this framework, project specific requirements can be rewritten in a semantically understandable way.

Solution test implementation and validation

Verification and validation of the proposed framework was conducted in this step. Several existing applications used within the current workflow were reviewed for implementing the proposed framework. The implementation approaches identified are elaborated in Appendix G. Visual programming and natural language processing (NLP) table parsing approach were used for verifiying the framework because of ease of its use and availability of licenses. A Viaduct project was considered as a case study for verification. The project had 80 requirements. The requirements were first given an unique ID followed by classifying them into various types based on the proposed classification. The requirements are first analysed and was allocated a relevant object type property from OTL. The requirements are now semantically classified. Based on the logical relations, requirements are now represented as rules. These rules were used for constructing visual programming scripts using Dynamo for Autodesk Revit. Though visual programming approach verified the feasibility the framework, constructing Dynamo scripts for each requirements was a manual intensive task. Some requirements could not be easily represented in a visual programming way easily. This required creation of some custom nodes. NLP approach was tested next for its ease of use. BIM data from Revit was converted into tabular format and rules were passed on as queries on this data. The accuracy of the results was very poor and algorithm took longer processing time for simple requirements. This could be attributed to the lack of relevant training data.

The framework was further validated by expert assessment. The utility and applicability criteria suggested by March and Smith (1995) was adopted for validating the proposed framework. The interview panel was presented with the framework and results from case study with both approaches. The interviewees were asked questions about the feasibility of the process, ease of use, future prospects and other criteria. Based on the answers to these criteria, the framework was evaluated. Overall, a positive validation was obtained for the framework along with the recommendation for implementing in actual projects for identifying more practical challenges and obstacles. The interviewees also recommended further research on developing NLP approach with relevant training data.

Conclusion

In conclusion, research proposed and demonstrated a way of defining requirements with which users can communicate effectively using a common language from OTL when describing about an object, property or process. This makes understanding of requirements easier, unambiguous and less error prone. A relevant software application needs to be selected for efficient implementation of the rules. The research also identified several limitations in the study. Study was conducted predominantly using requirements in English language. The use of the framework in Dutch language needs to be researched. The framework is also very less efficient when constraints are written with several objects and properties. This requires more operators to be included when writing rules and requires longer processing time for machines to execute the rules. The research concludes with recommendations for academia and industry in this domain. The recommendation relates to implementation of the proposed framework and improvement of requirements management process. A recommendation is also made about developing company specific OTL.

Contents

List of Figures

List of Tables

Acronyms

Chapter 1
Introduction

1 Introduction

Recently, there has been a increasing trend in the usage of BIM in the infrastructure industry due to its potential benefits in improving productivity during the project process. This thesis works towards supporting the ongoing effort in providing more efficient workflows in AEC industry by proposing a framework for automating requirements verification processes during design phase.

The current chapter forms the starting point of this research. The research domain and subject is introduced in Section 1.1 and the research problem is defined in Section 1.3. Before defining the problem, it is required to have good understanding about the current design process which is provided in Section 1.2. The research gap from the perspective of academia and Royal HaskoningDHV is elaborated in Section 1.4. Section 1.6 discusses the goal, expected results and aim of the project. The problem definition, research gap and research questions are summarised in Section 1.8.

1.1 Subject introduction

Infrastructure design process involves complying with broad range of requirements from building codes and standards, client needs, project needs and industry guidelines. The success of the project depends on the compliance to these requirements amongst other things (Wheeler, 2003). To accommodate the needs and requirements during the life cycle, Systems Engineering is being adopted as a industry wide practice in the Dutch infrastructure sector with greater push from governmental organisations like Rijkswaterstaat[1]. Systems Engineering prioritises requirements from stakeholders and a design is proposed to accommodate the defined requirements. INCOSE (2015) defines Requirement as "A statement that identifies a system or product's characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability". ISO15288 (2015) defines requirement as "A statement that identifies a product or processes operational, functional, or design characteristic or constraint, which is unambiguous, testable, or measurable and necessary for product or process acceptability".

Inconsistency or lack of compliance to a certain requirement is unavoidable over the construction phases and it becomes difficult to measure the consequence of these lack of compliance. The adoption of BIM enabled information management processes for requirement management and verification claims to improve business performance by reducing the amount of work and errors. This could be achieved by standardising the work processes. Process standardisation tries to create a consistent workflow. Process standardisation is said to result in cheaper execution costs, better collaboration, fewer cost overruns, and higher quality Roy, Low, and Waller (2005). This thesis works towards developing the framework for efficient workflows in requirement definition. The framework provides a new method for interpreting and rewriting the requirements in a machine and human understandable way.

1.2 Design process in Dutch infrastructure sector

A deeper understanding of the current design process followed in Dutch infrastructure sector is needed before defining the problem statement. The workflow followed in Royal HaskoningDHV is considered for this analysis and later compared with other organisations to standardise the

[1] https://www.rijkswaterstaat.nl/over-ons

explanation provided. The BPMN model of the current design process is provided in Figure 1 and is further explained below.

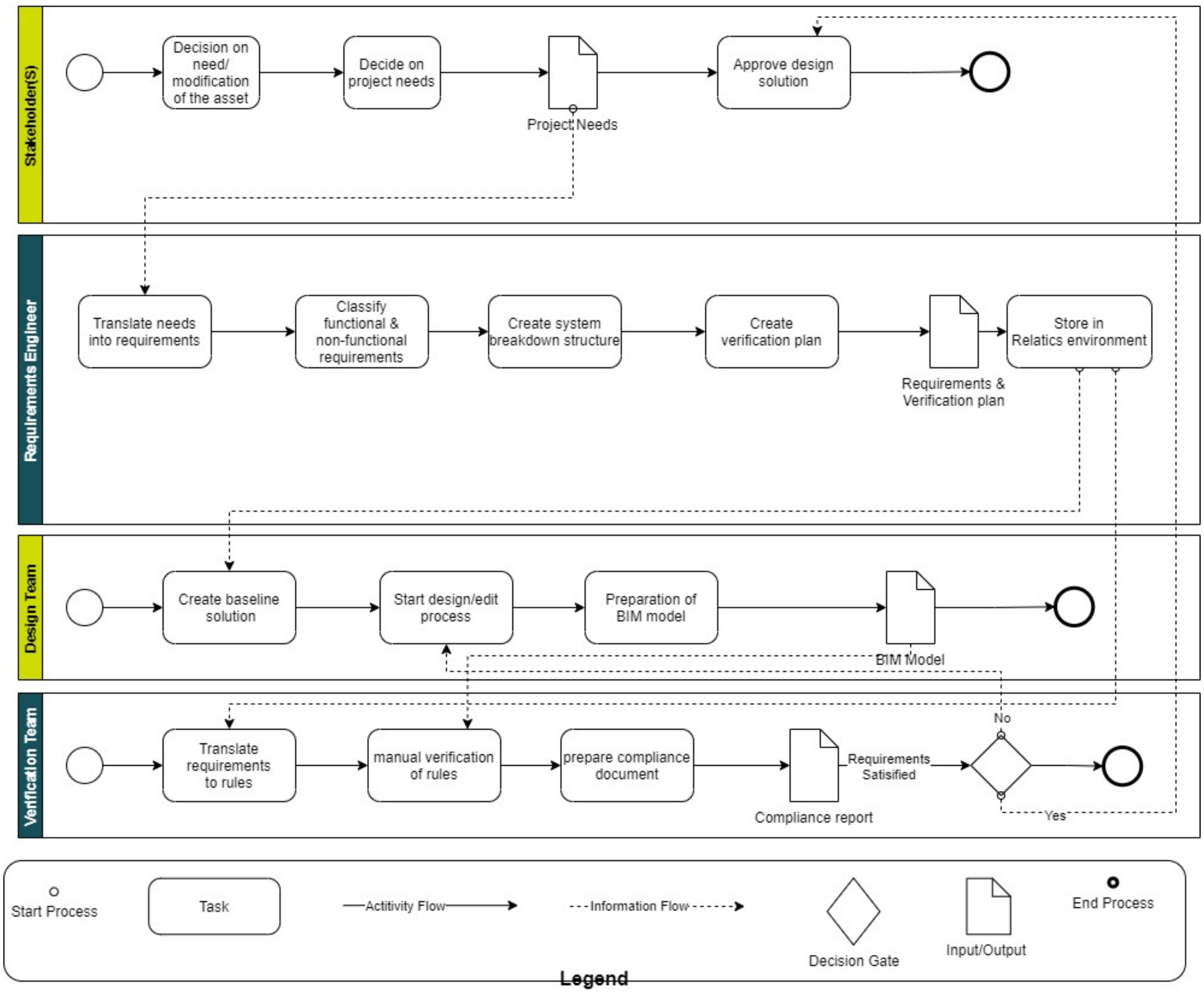


Figure 1: BPMN model of current design process

The Systems Engineering based design process in infrastructure industry begins with asset owners need for construction of new or modification of existing infrastructure asset. In the Netherlands, this is done by authorities such as Ministry of infrastructure and water management and is executed by organisations like Rijkswaterstaat. The authorities then along or, by collaborating with external contract management providers like Royal HaskoningDHV discusses the requirements and functions that needs to be satisfied by new/modified asset. The outcomes from these initial discussions forms top level requirements and needs of the asset. These needs are then handed over to contractors or constructing authorities based on agreed contract method. The contractor then analyses the top-level requirements and creates a detailed system breakdown structure based on initial breakdown structure provided by the asset owners during the contracting phase. The detailed requirements are drawn from top level requirements and is then classified into functional and non-functional requirements. The requirements structure from this process is stored in cloud-based requirement/document management systems such as Relatics [2]. Meanwhile, an object breakdown

[2]https://www.relatics.com/en/

structure is also created based on system design. The detailed description about objects and their properties can be derived from databases such as OTL. The OTLs are created and managed by Rijkswaterstaat or contractors developed themselves. The requirements stored in Relatics is assigned to the relevant object instances. A verification and validation plan is also developed to each of these requirements. These repository of requirements in Relatics environment forms the basis for the design of the project. The designers would look at the repositories and design a system to meet the given requirements. The verification team verifies the requirement according to the earlier defined verification plan. Though the theoretical process is simple, due to several operational complexities, the design and verification process gets difficult. The various problems faced during the process and root cause for these problems are identified in next section.

1.3 Problem definition

There now exists several problems in the current design and verification process. These obstacles needs to be rectified before moving towards automating the process. Requirements in textual form stored in Relatics becomes cumbersome and time consuming for designers to find, understand and visualise the needs (Kiviniemi, 2005; Malsane, Matthews, Lockley, Love, & Greenwood, 2015; Stancheva, 2017; Nagendrakumar, 2019; Beach, Hippolyte, & Rezgui, 2020). Requirements defined in natural language lacks a common vocabulary and may result in arbitrary interpretations causing overall delays in the project (Martins, Carvalho, & Almeida, 2016). Manual interpretations of requirements tends to be inconsistent due to personal experiences, bias and implicit regulations (Howell, Beach, & Rezgui, 2020). The perfect requirement should also exhibit SMART[3] criteria and large number of requirements fall short of these (Moonen, 2017). Lack of common vocabulary between various project members creates ambiguity in interpreting the needs (Alnaggar & Papadonikolaki, 2019b). Though Systems Engineering guidelines recommend formulating a standard project vocabulary, the definition and naming of the system components vary largely. Shifting design focus from conceptual to technical solutions also adds to the complexity of the information management. Several factors similar to mentioned above makes it virtually impossible for all stakeholders to know and remember all relevant requirements and their relationships with each other and to design solutions (Kiviniemi, 2005). These obstacles forms allocation of object to requirements a manual task. Requirement allocation becomes important part of design process since designers can design the system objects only after looking at requirements. The detailed problem analysis from literature and interviews based on requirements management, information management and requirement verification point of view is elaborated in subsection 3.4.

As found from literature study and interviews with industry practitioners, the root cause for several problems identified here originates from poor definition of requirements. Lack of common vocabulary to represent requirements is an major problem in mapping requirements with object instances for automated verification. Kiviniemi (2005) recommends that linking requirements with the design model helps in solving the interpretation and checking the problem. The current research considers this recommendation as a hypothesis and tries to provide solution for better definition of requirements with the use of object type properties from standard OTLs. This issue of mapping requirement with object instances for automating verification process is defined as a research question in Section1.5.

[3]SMART-Specific, Measurable, Attainable, Realisable, Time bounded

1.4 Research gap

1.4.1 Academic context

A design that complies with 100% of the requirement in the design phase, decreases the failure cost and improves project management processes (Alnaggar & Papadonikolaki, 2019b). However, along with problems mentioned in Section 1.3, the management of project specific requirements by clients now remains a manual process which requires initial investigation about information in the requirements and then use them in the design. This information about requirements are available within requirements management databases like Relatics. The usability and interaction with this information depends on the way it is written and stored. Research in this area to see how these requirements should be written, stored and linked with BIM models for usage in verification process is therefore necessary. Better information availability will aid in automating the verification process and improve the efficiency and quality of the design process (Ciribini, Bolpagni, & Oliveri, 2015).

The literature study on automated verification process identified more than 300 studies in this area. However, previous studies focus mainly on specific construction domains and not on the entire design process as a whole (Beach et al., 2020). The current available commercial tools for verification focus mainly on the compliancy with building codes and completeness of the BIM model. These tools are static in nature and do not take into account the dynamicity of project specific requirements. Thus, an approach for handling dynamic requirements is necessary. This can increase the value of the verification process. Lastly, the existing studies on automated requirement checking relies on expert knowledge of translating requirements into a machine understandable format. This makes the workflow closed since users require high level of expertise in programming, data structures and construction domain knowledge. The project team responsible for design and verification always has to rely on the interpretation of the programmer. An opportunity exists here in developing an automated requirement checking process that allow non-expert users to analyse requirements and convert them to rules. A framework for project specific requirements and representing them as rules by mapping them with object type properties from OTLs is proposed in this research.

1.4.2 Company context

This research was carried out with transport and planning business line of Royal HaskoningDHV. Royal HaskoningDHV is one of the largest engineering consultancy in the Netherlands, consisting of 5,800 employees and operating in 30 countries worldwide. During the past years, the company is focusing on how to boost its performance through digital programs and enhance profitability. With the digital ways of working (DWOW) program, Royal HaskoningDHV aims at standardising and automating their repetitive workflows thereby enhancing productivity. One of the main initiatives through DWOW program is integrated information modelling (IIM). Through IIM program, company is trying to identify activities that could be standardised and improved. The practitioners from Royal HaskoningDHV identify that the requirements verification process part of Systems Engineering process in large infrastructure projects is a repetitive activity and could be automated by standardising the workflows involved. The questions that however remains unsolved for company is that; to what extent the process should be automated, what are the obstacles for automation and what tools and technology is available or needs to be developed for this. This research tries to provide answer to this by creating and investigating the research questions defined in the following section.

1.5 Research question

From the problem analysis, four major questions are identified. The sub questions are then merged to form main research question. The reasons leading up to sub-questions are explained in this section and final research questions are given at the end of section.

Shifting design focus from conceptual to technical solutions adds to the complexity of information management. With several factors weighing into decision making, it becomes virtually impossible for all stakeholders to know and remember all relevant requirements and their relationships with each other and to design solutions. This issue is formulated as sub-question 1 on the way requirements is classified, communicated and documented. Lack of common vocabulary between various project members creates ambiguity in interpreting the needs (Alnaggar & Papadonikolaki, 2019b). This issue leads to sub-question 3 on level of detail needed for automated verification process. Poor requirements management without links to BIM models can often lead to ambiguity, inconsistency in assessments and delays in the overall construction process (Malsane et al., 2015). This issue is formulated as sub-research questions 2 and 4 on the process needed for mapping of requirements and thereby achieving automated compliance. Research questions that will be addressed in the current work are listed below. The answer to these research questions can be found in various chapters of this report. A guide to this is provided in Figure 2.

"How can we improve the mapping between requirements and BIM data for verification of design's compliance in infrastructure projects?"

To arrive at the final response for the main research question, it is further divided into sub-questions:

1. What are the different type of requirements in infrastructure design process and how are they managed and documented?
2. How can requirements be semantically mapped to object instances?
3. What level of detail is needed for allocating requirements data to object instances?
4. How can compliance of a particular object with its associated requirements be proven and how can this process be automated?

1.6 Research aim and expected results

This thesis project is dedicated to addressing the research gap described in Section 1.4. More precisely, it is to develop a general approach for writing requirements and mapping them with BIM data. From the problem definition stage, a clear overview of current practices in design process and information management will be given. Different classification of requirements in the infrastructure design, and the patterns in them will be identified. This answers the sub-research question 1 about the type of requirements and their classification. Literature review will provide a theoretical basis for mapping of requirements and object instances. Initial interviews with practitioners from Royal HaskoningDHV will confirm the needs in requirement management practices in construction industry and the need for linking of requirements and BIM model. The results from literature review will provide answers to sub-questions 2 and 3 on semantic mapping and level detail needed for object and requirement mapping. A test case will be evaluated using a set of requirements to validate the proposed framework and recommendation will be made towards developing a functional tool.Framework and validation of the framework will provide solution to sub-question 4 on compliance assessment. The research is aimed towards creating a framework for automated verification of client requirements hence every process and knowledge generated through the research helps in developing towards a final system.

In the longer run, research attempts to capitalise on these DWOW by automating some of the more manual, cost and time intensive process of requirement management and verification in the design phase. The ultimate aim of automated requirements verification is to automatically generate a set of design alternatives given the requirement parameters. This is done by parametric and generative design algorithms. The adoption of better requirement definition process creates an opportunity for application of parametric and generative models in design phases. The first step in use of these algorithms is to conceptualise the system to be designed as objects and parameters. However, the fundamental problem is how to make intuitive requirements in a way machine can interpret. The advent of BIM and open data standards is paving way for this.

1.7 Report structure

The report is organised into eight main chapters. This section will introduce the contents of each chapter of the thesis. The research questions answered in various chapters of this thesis is shown in Figure 2.

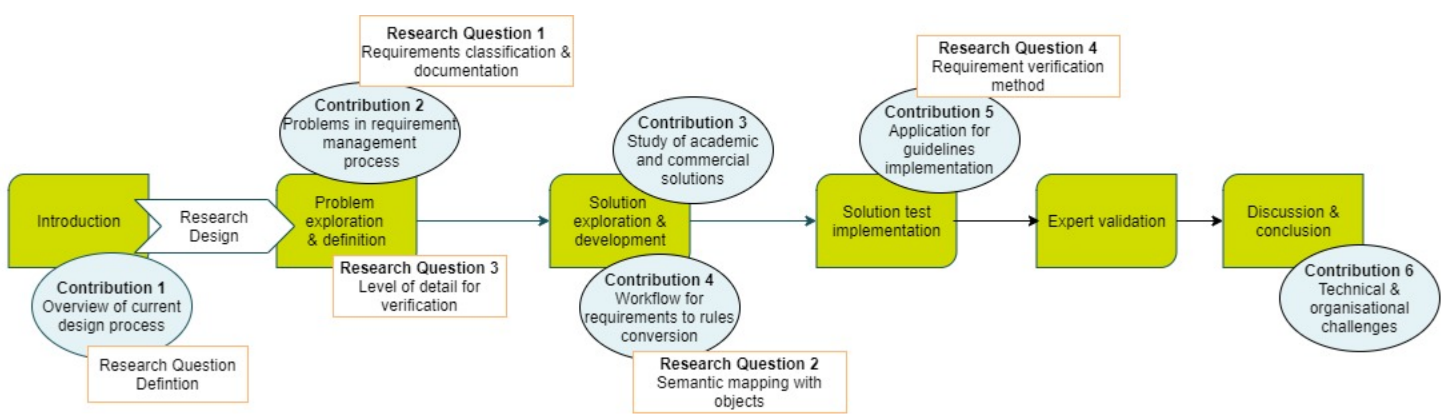


Figure 2: Structure of the report

Chapter 1 will present the subject and domain the research is related to, starting with the design process and problem definition. The chapter continues with defining the research context and defining the research questions based on the problems identified followed by providing a reading guide.

Chapter 2 introduces the reader to research design followed. The chapter starts with describing the research methodology and is followed by defining the research scope. Further, the aim and expected results from the research is introduced in the last section.

Chapter 3 provides the background knowledge required by introducing various concepts such as Systems Engineering, requirements management, BIM and requirement checking process. This chapter focuses on problem exploration and definition according to research design elaborated in

Chapter 2. The chapter ends with describing the results of interview conducted for problem verification and needs of the verification framework.

Chapter 4 focuses on developing framework. The chapter starts with study of various existing academic and commercial solutions and identifies the basic needs of the requirements checking process. A solution is proposed to satisfy the identified needs and is presented in the form of framework from the combination of existing solutions. The chapter ends with providing an BPMN model on how to integrate proposed framework in the current workflow.

Chapter 5 focuses on practical implementation of the proposed framework. Two approaches are proposed. Initially using NLP and then using Autodesk Dynamo [4]. The functionalities of each approach is analysed and explained. One of the approaches is further practically verified by implementing it on a small infrastructure project.

Chapter 6 focuses on the validation of the framework by expert assessment. The results from the case study is presented to the practitioners from Royal HaskoningDHV and are validated using criteria proposed in the chapter.

Chapter 7 answers the research questions defined in the first chapter. The limitations of the proposed framework and challenges faced are discussed. Future recommendations for academia and industry is also provided.

1.8 Summary

Success of the project depends on compliance to requirements amongst other things Wheeler (2003). This forms Requirements Management process an important step in the Systems Engineering life cycle. However, poor definition of requirements is an obstacle in automating the verification process. The existing solutions based on various approaches focus predominantly on building codes requirement verification and is not suitable for the dynamicity of client specific requirements. existing solutions also require users to have expertise on programming and construction domain knowledge. Hence, a general approach for rewriting client requirements in a machine understandable way is needed. The research focuses on developing this general approach using object and properties data from standard libraries such as OTLs.

[4] https://dynamobim.org/

Chapter 2
Research Design

2 Research design

The second chapter focuses on the research strategy and design of the thesis project. Section 2.1 describes the methodology and approach followed. Requirements verification and automated checking is a broad topic and hence, it is essential to limit the scope. This is defined in Section 2.2. The chapter is summarised in the Section 2.3.

2.1 Research methodology

The research is practice-oriented. The aim of the research is to reflect on the existing requirement verification practices and suggest changes and interventions. In this effort, obstacles in current system towards automation are identified and solution is suggested from reviewing existing academic and commercial solutions or a potential combination of several existing solutions. Double diamond research methodology proposed by Design Council (2019) is adopted in this study. Double diamond framework for innovation helps researchers in processing information in order to create solution (R. Costa, 2018). It describes the stages of divergent and convergent thinking. The original model proposed by Design Council can be seen in Figure 3.

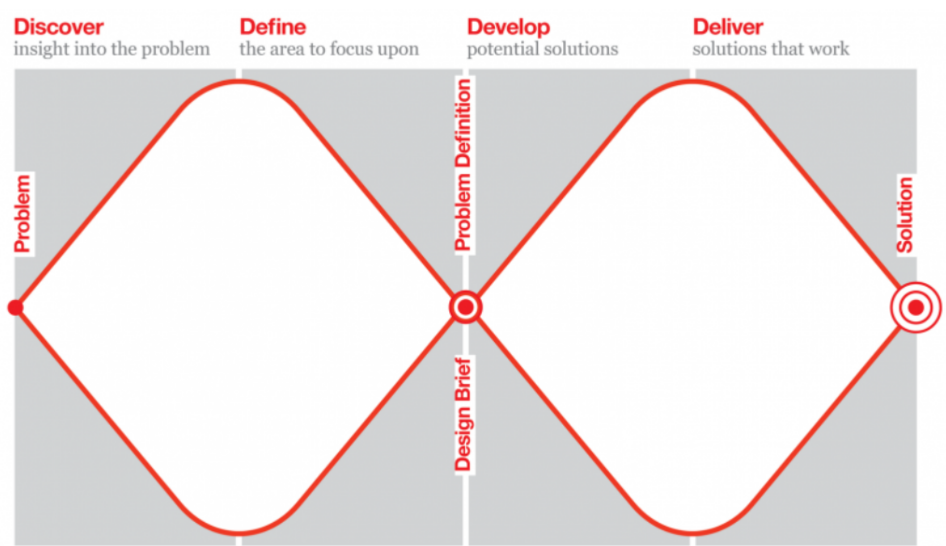


Figure 3: Double diamond design model (Design Council, 2019)

Design model involves four phases in defining a problem and arrive at a solution.

- **Discover:** Various problems faced in the process are explored. It entails conversing and spending time with those who are affected by the problems.
- **Define:** Knowledge gained during the discovery phase is used identify root cause of the problem that needs to be solved.
- **Develop:** Explore many solutions to the clearly specified problem, seeking inspiration from other sources and collaborating with a diverse group of people.
- **Deliver:** It entails experimenting with various ideas on a small scale, discarding those that will not work and refining those that will.

The above four stages are adapted to the current research study and various activities performed during various phases are explained below and is represented graphically in Figure 4.

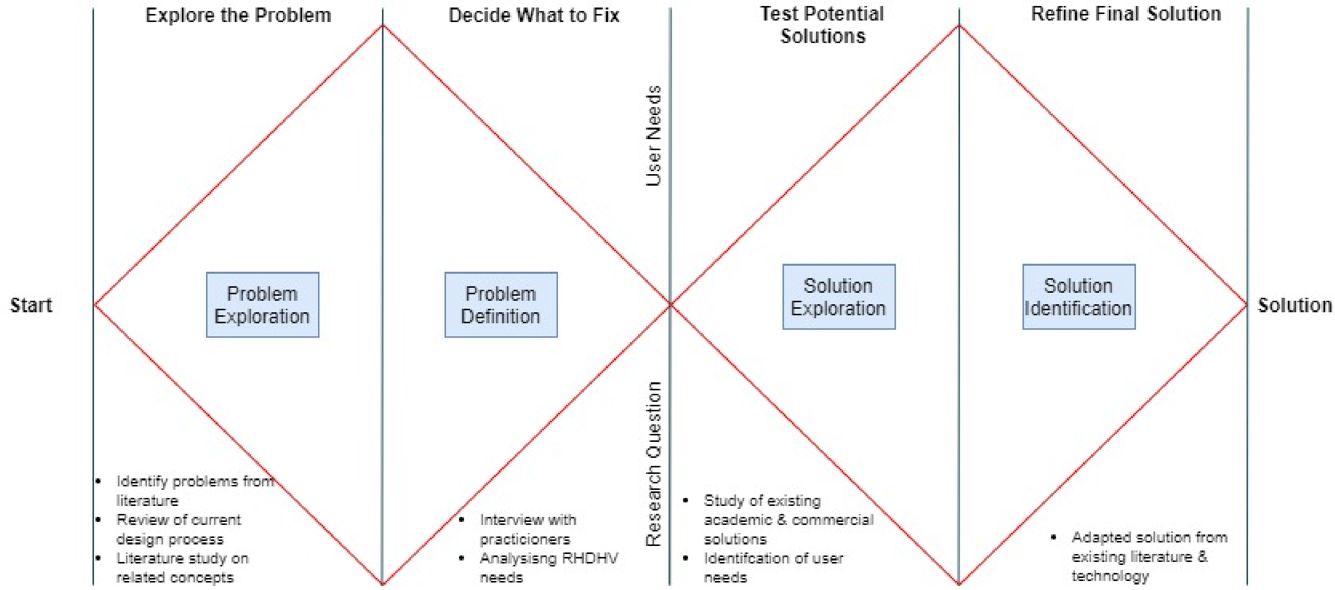


Figure 4: Double diamond model adapted to current research

The first stage of the double diamond process i.e. problem exploration and definition is elaborated in Section 3. The problem exploration stage forms the basis for this research. Problems are identified based on the design process that the Dutch infrastructure sector follows. Exploratory study is done to analyse both the design process and the information management practices of infrastructure projects in design stage. Identified problems are validated by finding relevant literature and studying their background. In the next stage, problems are further verified by conducting interviews with practitioners from Royal HaskoningDHV and other organisations. During this process, needs from user for an automated system are also identified. Based on the problems identified and solution hypothesis made, a research question is defined to arrive at the final solution.

After defining the research question, second phase of the process focuses on developing a framework that answers the defined research questions. The solution exploration and development part of the research is elaborated in Section 4. The process starts with identifying several academic and commercial solutions available through a rigorous literature search process. The literature collection process is explained in Section 4. Based on various approaches available, a solution is suggested from existing or a potential combination of several existing solutions. The solution is then tested as a use case on a small infrastructure project to identify feasibility and implementation challenges. At the end of the process, a potential user acceptable and feasible solution for requirements definition is proposed.

2.2 Research scope

To develop the framework for automated requirements verification, scope of the research needs to be defined since requirements verification is a broad topic with large boundaries. According to Eastman et al(2009), automated requirements verification process involves four important steps.

- Requirement interpretation

- Building model preparation
- Rule execution
- Reporting checking results

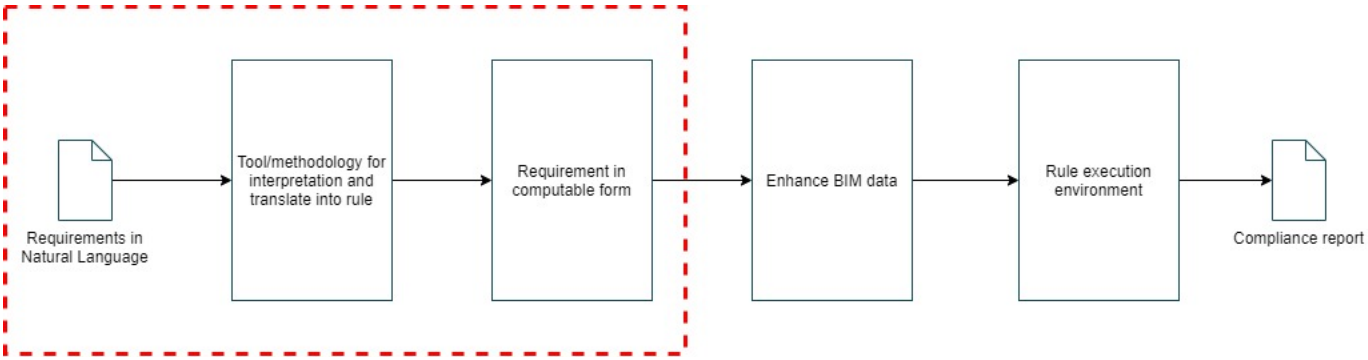


Figure 5: Requirements verification process

The scope of the current research in requirement verification process is highlighted in Figure 5. Nagendrakumar (2019) identifies that the requirements information stored in Relatics has not been harnessed for any automation because of the non-SMART definition of requirements. This forms a major barrier in knowledge transfer and a solution this could be having a guideline or manual that helps stakeholders in defining the requirement. However, several such guidelines published by INCOSE (2015) and Leidraad SE (2015) exists, without any improvement in the way requirement is written. Due to fragmented nature of construction industry, with several stakeholders from various organisations involved, strict enforcement of these guidelines proves difficult(Lenferink, Tillema, & Arts, 2013). In order to solve the above mentioned problem, this thesis project focus on developing a framework for interpretation of requirements and rewriting requirements in machine understandable form suitable for verification within the existing workflow without asking the involved stakeholders to work in a new manner. This is proposed by use of object type properties from OTLs. Though the highlighted part in Figure 5 forms the major scope of the research, enhancing BIM data with the help of OTLs and Dynamo scripting is also focused upon for verification and validation of the proposed framework.

2.3 Summary

The current research is more practice oriented trying to solve the requirements definition problem in Royal HaskoningDHVs business processes. Double diamond method proposed by British Design Council is adopted as a research methodology for the current project. The research is divided into two phases with initial phase focusing on problem exploration and defining research question. The second phase of the research focus on identifying a potential user satisfying solution from the existing or combination of existing solutions. Scope of the research is set on requirements analysis and rules creation in the entire checking process. However, BIM data enhancement and checking is focused upon briefly for verifiying and validating the proposed framework. As a goal, research tries to address the problems in client requirements definition and mapping. A new framework for writing the requirements, existing solutions review and requirement checking framework forms an important contribution of this research.

Chapter 3
Problem Exploration and Definition

3 Problem exploration and definition

This chapter gives an insight into Systems Engineering and its adoption in infrastructure industry, requirements engineering and management practices, BIM and requirements checking process. Conclusions from literature review forms the basis for interviews and also defines the needs for developing automated verification framework. This chapter answers the research question one and three about requirements classification, documentation and LOD needed for verification.

Section 3.1 provides introduction to Systems Engineering as a concept. The section is further divided into two. Section 3.1.1 focus on the various activities involved in the systems life cycle and Section 3.1.2 focus exclusively on the requirements management process. The chapter moves on to Section 3.2 which details about information management practices. Concepts of OTL and ontologies is explained here. Section 3.3 provides a general introduction to the existing studies on rule checking process and Section 3.4 analyses the problems and limitations of the current practice and existing technologies. This will supplement the problem definition from Section 1.3. The outcomes of the problem verification and needs analysis interview is explained in Section 3.5. Summary to this chapter is provided in Section 3.6.

3.1 Systems Engineering (SE)

Literature shows an increasing interest in adoption of Systems Engineering process in construction industry in the past decade (de Graaf, Vromen, & Boes, 2017). INCOSE (2015) defines SE as "An interdisciplinary approach for realisation of successful systems by considering both the business and technical needs of all stakeholders with goal of developing a final system that meets all the stakeholder needs". Systems Engineering is looked upon as a more standardised way of working in infrastructure industry (Leidraad SE, 2015). SE as a concept was first applied in 1940s in the telecommunication sector and in development of military systems. Later, SE was further developed in aerospace sector. All these sectors are known for their complexity and issues they face in implementation of the designed systems. Breaking down complex systems to its elements gave a better control in monitoring the issues and development of the system. International Council of Systems Engineering (INCOSE) was setup in 1990 to further develop Systems Engineering processes (INCOSE, 2015). In Netherlands, Werkgroep Leidraad SE was setup to gain and share knowledge in SE domain. INCOSE has also created a group for infrastructure to define SE guidelines for infrastructure sector. Prorail and Rijkswaterstaat, two biggest clients in The Netherlands are leading and specialised in application of SE process for their projects.

3.1.1 Systems Engineering life cycle

The SE methodology is based on systems thinking. A system can be defined as a set of interconnected components that work together to achieve a common goal (INCOSE, 2015). It is an iterative method that supports continuous improvement in the design process (Houdt & Vracken, 2013). Every individual component in the system must be considered as important and should be prioritised from the start of the design process (Leidraad SE, 2015). However, this is different from the current design practice (Section 1.2) followed in the Dutch infrastructure sector.

The goal of Systems Engineering process is to provide an user acceptable solution to a complex problem in dynamic environment. This could be achieved by having a life-cycle perspective from the start of the project. INCOSE (2015) describes systems engineering life cycle model with six phases. The various phases in systems engineering are shown in Figure 6.

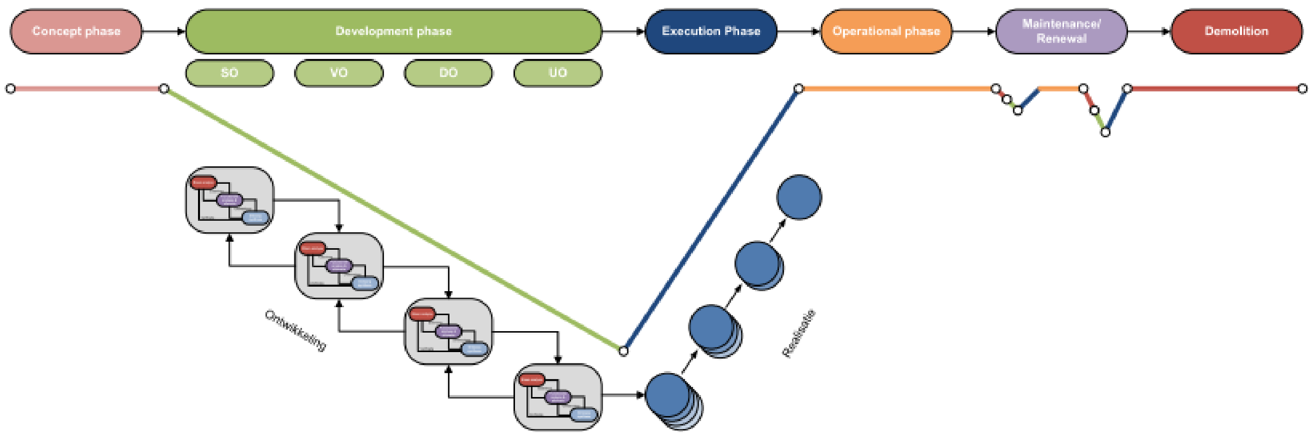


Figure 6: V-model and life-cycle phases (Huijnen, 2014)

- Concept phase
- Development phase
- Execution phase
- Operational phase
- Maintenance/renewal phase
- Demolition phase

These life cycle phases can correspondingly be related with the current design process explained in Section 1.2. Requirements verification forms an important part in concept and development phase. Besides different system life-cycle phases, Aslaksen et al. (2012) groups the various activities that are performed during system life-cycle into four process groups as shown in Figure 7.

In the context of this research, only technical process group is applicable. Technical process is concerned with technical and design decisions throughout the system life-cycle to fulfil stakeholder requirements. This is possible by making the design decisions explicit and by capturing the various decisions made at system component level so that the design meets the requirements for project (Albers, Scherer, Bursac, & Rachenkova, 2015). This is made possible with the use of requirement management applications like Relatics. Key elements during this process are requirements analysis and verification & validation.

3.1.2 Requirements management

When the design is complex or has a long realisation time, a formal Requirements Management process is of great importance. Requirements Management consists of definition, analysis, verification, and validation of requirements including communication and negotiations with the parties involved. The process of managing requirements leads to design and development of effective system. The process of requirement management should continue throughout the system life cycle,

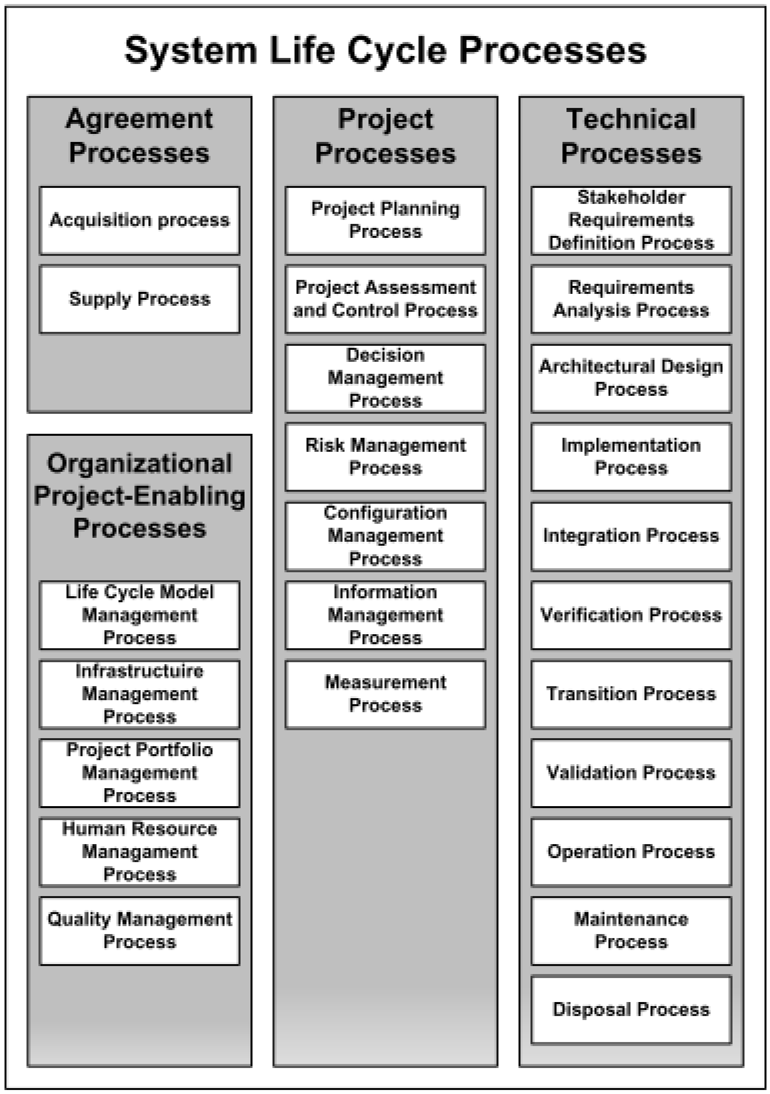


Figure 7: System life-cycle process (Aslaksen et al., 2012)

that is from identifying needs of the asset to demolition of the asset.

Requirements definition process: Defining requirements is part of the technical process. According to INCOSE (2015), the goal of requirements is to increase the likelihood that the right system will be built and when built, it meets the wishes and needs of the client at an acceptable level. According to ISO15288 (2015), various activities involved in Requirements Management process are described below:

- **Stakeholder identification:** Various stakeholders and classes who have interests in the project must be identified before the start of the process. This avoids problems during, and at the end of the design process since all stakeholders are part of the process and a mutually agreed solution could be designed (Ciribini et al., 2015).

- **Define stakeholder requirements:** The constraints resulting from existing agreements, technical complexities and management decisions should be clearly defined at the start of the process. Interaction between users and system should also be identified by specifying all the critical criteria such as health, safety, security and environment.

- **Analyse and maintain requirements:** Analyse and resolve stakeholder problems and contradicting requirements. It must be ensured that requirements captured adequately express the stakeholder needs. Requirement statements must be recorded using suitable digital application and ensured that requirements contain metadata that can be traceable to the stakeholder.

The analysis part in requirements definition process is different from activity "Requirements Analysis" presented in next paragraphs. Requirements definition process focuses more on formulation of requirements rather than analysing their characteristics. When defining requirements, highest goal must be stakeholder satisfaction (Van Swaay, 2011). According to Robertson et al.(2017), it is common for stakeholders to express needs in the form of solution. It is then the job of systems engineers to identify the exact needs from the solution and to decide whether if it is an ideal solution, or if it forms a constraint to design space. A good requirement specification must meet number of characteristics like completeness, up to date, clear and SMART according to Leidraad SE (2015). Responsibility of defining good requirements must be a collaborative effort of various stakeholders involved in the project. Dependency on a specific role or person leads to bias in the definition (Leidraad SE, 2015).

Requirements analysis process: Requirements analysis is a critical element in SE methodology. Purpose of requirements analysis process is to transform abstract requirements specifications defined into, a product, that satisfies end users expectations (ISO15288, 2015). Various activities in the requirements analysis process are:

- **Define system requirements:** Attributes, characteristics and functions expected from the system must be defined during this process. Functional constraints that need to be considered in the solution space should be identified. Verification and validation method that enable the assessment of fulfilment of requirements should be defined during this process.

- **Analyse and maintain system requirements:** It must be ensured that requirement set, and the vocabulary used is consistent and unambiguous. Captured requirements must be interpreted, and the interpretation should be cleared with responsible stakeholder for clear communication. Traceability of system requirements and stakeholders should be demonstrated.

Requirements should be captured along with their metadata. These metadata will link roles, methods, stakeholders and helps in clarifying design decisions and assumptions in further stages of the project. Verification method also should be captured along with the requirement. Various metadata that should be captured for a requirement are shown in Table 1.

Table 1: Requirements metadata

Attribute	Function
Unique ID	Unique identification system using numeric or alphanumeric strings for tracking and sorting.
Description	Additional information to clarify the assumptions and decisions made before capturing the requirement.
Role	Captures roles between parent requirements and other system components
Owner	Person responsible for managing and approving the requirement.
Verification Method	Captures the type and method of verification and defines how the verification is carried out and demonstrated.
Verification Responsibility	Person responsible for carrying out the verification process.
Verification Phase	Defines during which project phase the verification process is carried out.

Requirement classification is a necessary step in definition process. There are various classifications of requirement types proposed by guidelines like Leidraad SE (2015), INCOSE (2015), and several other academic studies. Requirement classification suggested by INCOSE (2015) are:

- Functional requirements
- Performance requirements
- Interface requirements
- Constraint requirements
- Operational requirements
- Regulatory requirements
- Reliability, Maintainability, Availability, Supportability aspect requirements
- Safety requirements
- Health hazard requirements
- Human performance interface requirements

This requirement types are for general classification of all requirements in the project. The machine cannot interpret the basis and functionalities behind these classifications. Since the current research focus on making requirements more machine understandable, a classification based on their function and rule type is introduced. These classifications are adapted from studies done by Moonen (2017) and Dimyadi et al. (2016). The classification type mentioned in Table 2 classifies requirements quantitatively depending on verification type without considering the functionality of requirement in the design process. Quantitative nature of this classification type helps machine and users understand whether if it is an automatable requirement and what type of check needs to be performed for compliance.

Table 2: Requirement types

Requirement Type	Operator	Description
Value requirement	=, > , <	Requirement that needs to be verified by checking values of certain parameters
Objective requirement	Document proofs	Requirements that forms the design principle which can be verified through complex simulations or human interventions
Relational requirement	Boolean (True or False)	Requirements that needs to verify the existence of a certain object, relation, or object property.

Requirements decomposition is part of requirement analysis process. Requirements from top-level are decomposed and is allocated to subsystems and various activities. The decomposition process forms an important step in requirement interpretation and development of verification of plans. The process is schematically shown in Figure 8. Further chapters discusses how requirements management process can be further improved with the use of new age digital applications like Relatics and how textual requirements could be used for semantic enrichment of BIM models.

Requirements Verification Process: It acts as a feedback loop for the design process. ISO15288 (2015) defines verification as compliance to the defined requirements. The results from verification process are worth as much as the data that is put in the process. Hence requirements should be clearly defined with relevant verification methods to carry out this process correctly(Moonen, 2017).

According to INCOSE (2015), verification process consists of two steps. They are:

- **Strategy preparation:** This involves preparing the verification methods according to costs and risks of the project. In this process, what should be verified is defined first and then followed by the method of verification (test, inspection, analysis).
- **Perform and manage verification:** Execute verification plan according to plan and analyse results. These results should be communicated with design team to evaluate what action needs to be done for system elements and requirements that are not complied.

3.2 Information management in infrastructure Sector

In this section, basic concepts on information management practices are defined with example. The concepts will be explained to required level for the understanding of this research. There exists

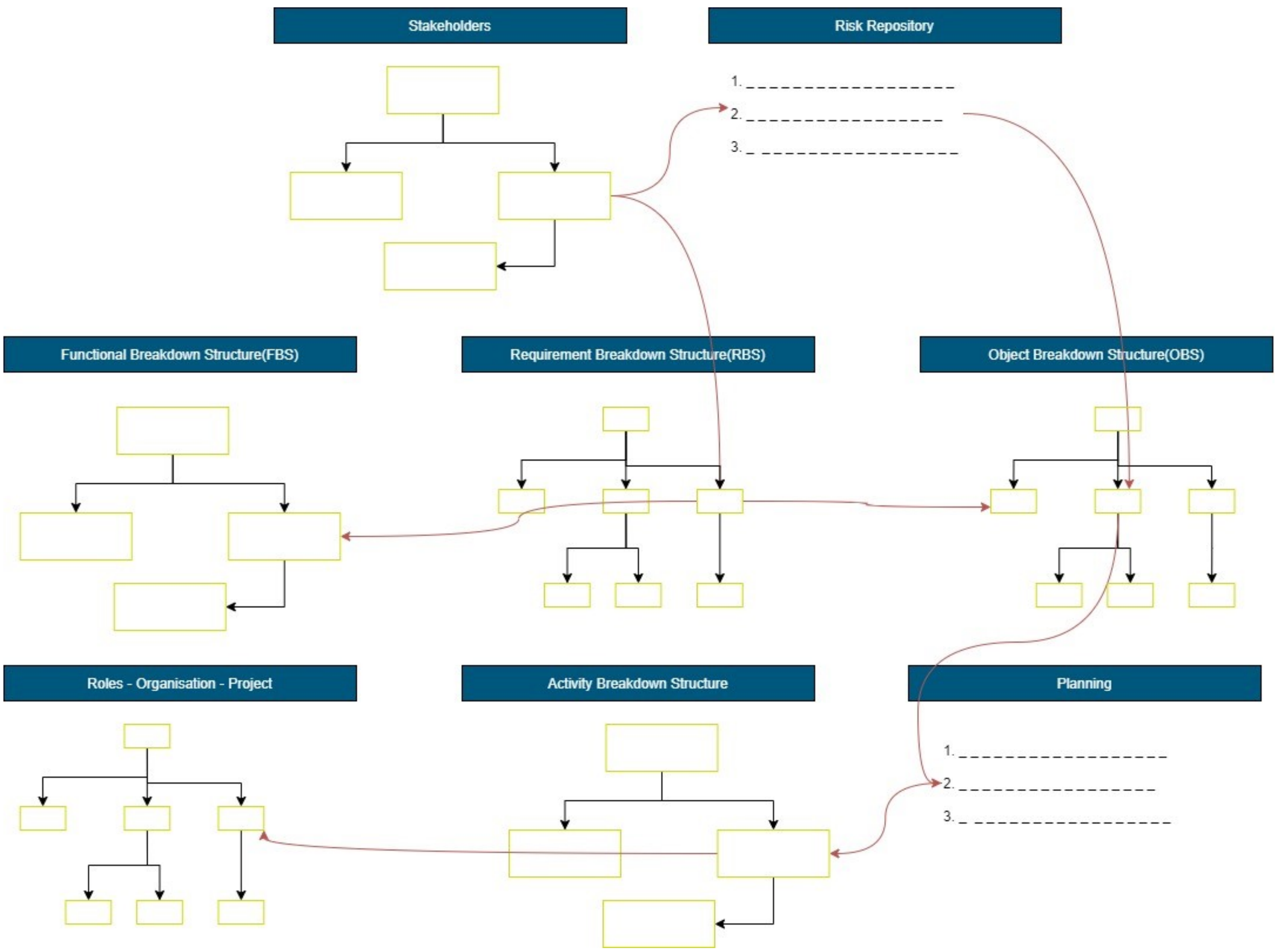


Figure 8: Requirements decomposition and allocation. Adapted from (Huijnen, 2014)

several definitions of what information management in infrastructure sector is and generally, the definition varies depending on the project stakeholder using it. According to ISO29481 (2016), BIM provides a concept for describing and displaying information required in the design, construction, and operation of constructed facilities. Before discussing about development of BIM models we first dive deep into the various concepts on object type library and semantic modelling.

BIM authoring applications: These are tools that support 3D modelling of the design solutions for projects. BIM applications work on object-based modelling technique. Object based modelling means parameters, rules, geometric properties and constraints with surrounding elements define the objects (Sacks, Eastman, Lee, & Teicholz, 2018). According to Pauwels, Zhang, and Lee (2017), object properties has two major category. They are:

- Geometrically inherent and derived properties such as area, length, width, and height.
- Additional metadata such as manufacturers and product details, system classification code.

From the 3D BIM model, several 2D views of the system such as plan, section and elevations could be created. These 2D views could be annotated with several information retrieved from objects. The use of multiple views help in sharing the information consistently. Unlike 2D model, the objects in the BIM models are dynamic and if a property of one object changes then geometries associated with those objects changes accordingly. This leads to better information consistency across all the system elements (Pauwels, Zhang, & Lee, 2017). There are several commercial BIM authoring applications on the market like Revit, ArchiCAD, Tekla structures, BlenderBIM et cetera. Revit by software developer Autodesk is the most widely used software for authoring BIM models (Team-unifi, 2019).

BIM models use classification systems for more detailed specification of object types. Most BIM authoring tools come with generic classification systems built in. However, integrating with open standard classification systems such as object-type-libraries and Uniclass classification systems increases the usability of the system and more information could be leveraged from the models (Wu & Zhang, 2019). Use of standard and inter operable classification systems allows reuse of model data for more advanced applications like safety simulation, energy analysis et cetera (L. Ma, Sacks, Kattel, & Bloch, 2018).

Object type library: Luiten et al. (2018) defines OTL as 'A library with standardised object-types names (e.g. road, viaduct) and properties or specifications'. Data is unintelligent when it lacks context or meaning and becomes increasingly difficult to exchange between applications. This makes data unconnected, incoherent and bound to single databases (Daan, 2020). With the use of OTL, structured and meaningful standardised data sets of infrastructure objects could be created. In the context of this research, the purpose of OTL is to provide a common vocabulary for representing object information between various project stakeholders. How an open standard object type libraries with linked data technology could be created is elaborated in Appendix B. An object in OTL carries information about its geometry, properties, metadata and information about connected elements. The objects in the same domain are grouped to create ontologies (Huitzil, Molina-Solana, Gómez-Romero, & Bobillo, 2021). OTL offers several perspectives in terms of users, application, and data orientation point of view. Luiten et al. (2018) describes three perspectives as shown in Figure 9.

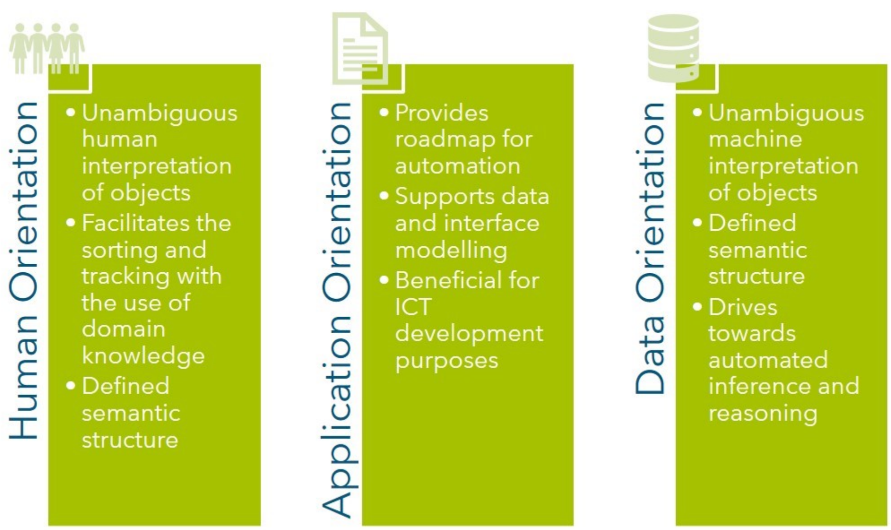


Figure 9: Perspectives on use of OTL. Adapted from (Luiten et al., 2018)

Ontology: Ontology is formal conceptualisation of the knowledge in certain domain by classifying them into concepts, elements and properties (J. Zhang & El-Diraby, 2012). Ontologies has paved a way for representing and reusing semantically rich domain information (Anumba, Issa, Pan, & Mutis, 2008). Several research studies demonstrate the advantages of BIM and ontology in rule/requirement checking process (Beach et al., 2020). Ontology specifies how individuals are grouped and linked within a hierarchy, as well as how they are divided into groups based on similarities and differences.

An ontology for a bridge can be decomposed by classifying the objects into load bearing elements, non-load bearing elements and maintenance types. These objects and object properties are also shared among OTL of other bridge types which provides greater level of details and can be created as a centralised repository or data dictionary (Luiten et al., 2018).

Though BIM can be potentially used for many analysis and simulation processes as discussed in earlier sections, there still exists a static link between requirements information and building objects. This means once the model or information changes, the information links have to be re-established again (Ding, Zhong, Wu, & Luo, 2016). For project specific requirements verification process, static links are not beneficial since updating links for every requirement change can be a costly and laborious process which is equivalent to manual verification process. Hence, information should be integrated within the BIM model in a dynamic and flexible way (Dimyadi et al., 2016).

Level of Detail (LOD): The LOD allows practitioners to accurately describe and express the content and information level of the BIM models at various construction stages(Safari & AzariJafari, 2021). LOD is an important concept in answering the research question about the amount of information needed for requirement verification process. The American institute of architects recommends a five level classification. They are LOD ranging from 100 to 500 in increments of 100. for information representation level in BIM models. For example, in LOD100, an object is graphically demonstrated in the model with a symbol (e.g., a line or a surface) or generically (i.e., dimensions and quantities are not defined), while in LOD200, the objects are shown as a generic object with approximate quantities, sizes, dimensions, position, and orientation. In LOD300, the model reaches to the point that it can define the overall function using a particular object that includes quantities, sizes, measurements, position, and orientation. The components at LOD400 have details, connections, installation, and setup in addition to the features of LOD300. Executive design plans and construction stage are identical at this phase. Model components are displayed in LOD500 in the exact order in which they were designed and constructed. The level of detailing available at various LOD levels is illustrated in Figure 10.

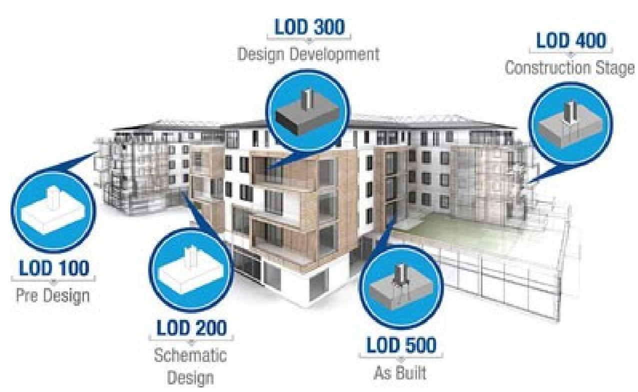


Figure 10: Representation of LOD (Bertin et al., 2020)

LOD also varies across the system life-cycle. Different objects have different LODs during the design process and also during different phases of the project (Solihin & Eastman, 2015). Identifying the LOD required for specific requirement is hence a necessary task in automated requirements verification process (Beach et al., 2020). For requirement compliance using BIM models at design phase, LOD300 is sufficient (Solihin & Eastman, 2015). However, requirement modelling should be evolved such that lowest level of LOD should be adequate for verification process.

3.3 Requirements/Rule checking process

Life-cycle of infrastructure construction is governed by varied regulations, requirements and standards (Nawari, 2019). Compliance checking process is growing as the complexity of criteria and number of requirements is also growing. Over last two decades, there have been several extensive research studies on automated compliance checking Dimyadi and Amor (2013). Even with extensive research, adoption of automated compliance checking as part of standard workflow is limited (Goh, 2008). Lack of acceptance in the past has been attributed to the lack of maturity of data sets generated during the design stages (Daan, 2020). However, as BIM models become matured and more widely adopted, automated compliance checking is becoming more feasible (Wu & Zhang, 2019).

Checking for conformity with standards is mostly a manual process that depends on expertise of those performing the task. As a result, perception bias is a big issue in this area (J. Zhang & El-Gohary, 2016). This makes interpretation of requirements, error prone. When interpretations are made standard with common project vocabulary using OTL, it reduces the need for interpretation by experts and thus reducing errors and time taken for evaluation (Dimyadi et al., 2016). Model checking is used as general term for several applications in the AEC industry. This involves requirement checking, code compliance checking and model validation among other applications. The checking is performed on the BIM model and on the information it contains (Hjelseth & Nisbet, 2011). The current research scope is on compliancy checking. The process for automated requirement verification is divided into four steps by Eastman et al. (2009). They are:

Step 1: Requirement interpretation and logical structuring into rules
Requirements are first defined by stakeholders in several formats in the form of natural language text, tables, equations, and pictures. These requirements should be interpreted and programmed to a machine understandable rule. The process depends on programmer's interpretation and the way of translating it into machine code. This process could be standardised with the help of common project vocabularies suggested by Leidraad SE (2015).

Step 2: BIM model preparation
In 2D drawings the objective was to create drawings that are visually correct and carry necessary labelled information. However, with object-based BIM models this is not the case. Objects should have type and properties. An example, in BIM models, treads in stairs could be marked as small slabs but the model checker will not interpret this as a stair unless it is defined as a stair with its properties tread, riser, run etc. Thus, information requirement from BIM models for automated checking is much higher. Eastman et al. (2009) suggests that if users are asked to explicitly enter complex properties such as volume of space et cetera then chances of errors are high. Hence, the preferred solution is to automatically derive data wherever possible.

Step 3: Rule execution phase
Rule execution brings rules interpreted from requirements and BIM models together. Eastman et al. (2009) suggests that syntactic checking and data completeness checking (Appendix A) should

be done before execution of the rules. Because of several complexities involved in the design process and since requirements vary around life cycle of the system, the verification system should be a mixture of automated and manual checking process. Based on verification method, requirement type and type of rules, Dimyadi et al. (2016) has provided a classification which is described in Table 2.

Step 4: Reporting of checking results
The last step in requirements verification is reporting of the results. Requirement will have a pass, fail, or error as a result (Moonen, 2017). The results should be linked with original requirement to make it traceable and to ensure documentation and reasoning. This makes documentation easier for systems engineer during verification and validation process (Leidraad SE, 2015). Due to the broad nature and scope of requirements verification process, there are several difficulties in creating the requirement. Limitation and difficulties in requirements checking is elaborated in Section 3.4.

3.4 Limitations in current practice and existing technologies

Literature review and study of related works in the previous sections and analysis of current design process also exposed several limitations and difficulties in them. In the below section, limitations of various concepts discussed above are further elaborated.

Limitations in requirements management practices: AEC industry is often described as fragmented and data intensive. AEC industry now uses traditional and integrated procurement methods for construction projects. There exists a challenge in requirements traceability and continuity, causing each actors to have their own domain specific interpretation (Kiviniemi, 2005). The use of requirement/document management applications like Relatics has solved the problem of requirements traceability and continuity by providing features like version control and change identification. However, data is still stored in isolated data silos without connection to the BIM models which makes querying and automation of data difficult (Nagendrakumar, 2019).

Limitation in BIM processes and workflow: According to Soman et al.(2020), the current use of BIM in design has two major limitations. The first limitation is fact that significant amounts of information and knowledge aboust design and planning are not represented in BIM model. Information such as trade off matrices, design decisions, requirement derivations et cetera by not being represented in BIM models, is lost during the design process. Second limitation in (Soman et al., 2020) describes the problem associated with BIM in collaboration processes. Dimyadi et al. (2016) describes the problem is due to low level of semantic information parameters. Interoperability is also a major issue among software applications used in AEC industry. In the current process, the information (graphical and non-graphical) modelled is enclosed in native formats and is difficult to link to external data sets dynamically. Incapability to connect requirement data sets and complexity in sharing the information among different applications would present a potential risk of not complying with specified requirements. Thus, necessary support required for this task needs to be built in BIM workflows.

Limitations in Requirement checking: The vast academic research and several commercial applications developed in AEC industry has identified many technical, organisational and political challenges in the requirement checking domain (Beach et al., 2020). As defined earlier, the main problem lies in the way requirements is written and linked to various system elements. Poor definition of requirement leads to wrong assumptions and interpretations. The interpretation also varies with background and experience of the stakeholders interpreting it Eastman et al. (2009). According to Dimyadi and Amor (2013), the complexity of requirement verification process lies in the complexity of the built up rule. Language structure and domain knowledge needs to be standardised to make the rule built up process consistent and reusable. This data needs to be made available quickly and easily, for a more efficient process. If a neutral exchange format is not achieved then the model checking can be incomplete and inconsistent (Pauwels & Terkaj, 2016). The most important difficulties in the execution of rule checking and reporting is the maintenance of database for rule execution (Moonen, 2017).

Leidraad SE (2015) and INCOSE (2015) recommend that defining requirements must be an iterative process. Expectations from the requirements itself may change overtime based on needs of project. Hence, rule execution platform must have a possibility to dynamically interpret and update it as rules. The costs associated with updating is high since entire system design cycle must be carried out and verification methods should be validated again.

3.5 Interview and discussion

The motive of this interview process is to investigate problems in requirements engineering process and define the needs for automated verification workflow. For designing the framework, a clear understanding of requirements management, design and verification process is needed. The findings from review of existing studies, limitations and commercial solutions are discussed with industry professionals. During the interview, design and verification process explained in Section 1.2 was presented to the participants to give an overview of the current situation. This gives opportunity to discuss the process and prerequisites for automation workflow. Open interview structure is adopted to allow new ideas and discuss their possibilities. This gives the advantage of interactively collecting data by asking in depth questions, clarifications and thus provide direction to the conversation.

Interview setup: The interview panel included professionals from the domains of systems engineering, BIM coordination, design and verification process and software developers for Systems Engineering tools. The interviews will try to gather knowledge and experiences of professionals who use requirements during the entire construction life-cycle. The knowledge gained from the interviews is used to answer some of research questions and gather requirements for the development of workflow. Similar topics were discussed with different experts to get their respective perspectives on the current process. A total of seven interviews was conducted over a period of 3 weeks. The composition of interview panel, interview questions and expected outcome from those questions are elaborated in Appendix C.

Interview outcomes & discussion: The discussions and findings from interviews are classified into 3 sections. They are:

Requirements management: Requirements management in the Dutch infrastructure is carried out using applications like Relatics with occasional use of other applications like Briefbuilder[5] and Laces[6]. All the participants of the interview predominantly used Relatics in their existing

[5]https://www.briefbuilder.com/
[6]https://laceshub.com/nl/producten

workflow. Current requirements management practice is described in Section 3.1.2. In case of integrated contracts, though requirements analysis is done before the submission of tender, detailed requirements analysis is done only after the contract is won due to investment reasons. Due to this, complete design is submitted to tender without detailed analysis. This leads to several change management issues. The interviewees identified that an automated solution will be an added value in such situations. The second major problem faced during requirements management is the allocation of requirements to specific object instances. System structure needs to be designed in the early design stages when all the relevant information is not available. Major problems occur later during the design and verification process due to missing or wrongly allocated links.

The problem of interpretation was also mentioned here. Ambiguity in interpreting the requirements leads to problems in object allocation adding to the complexity of requirements management process. A solution for this problem can be by adopting standard system guidelines for carrying out System Engineering process and standardised way of creating object and system breakdown structures. Though creating guidelines and workflows is simple, the participants argue that it is very difficult to implement these new workflows since it requires a huge organisational change in their way of working. However, several system engineers gave the opinion that requirements management process could be improved when there is an option of re-usability from standard libraries and older projects, organisations can invest more in creating standard object libraries and specification libraries. Several participants also argued that a new open standard system apart from Relatics is also needed for innovation in the requirements management field.

Design Process: The detailed design process followed in the entire building life-cycle is described in Section 1. In an ideal design process, every requirement from the clients will be specifically and unambiguously defined and no changes are made during the design process. This is not the practice in the existing workflow. Requirements are mostly defined by the client in contractual language and designers use several methods and techniques to translate this into quantifiable requirements. This takes huge amount of time during the design process because of back and forth communication with clients to interpret the needs. The right interpretation is necessary to achieve the design that client envisioned. In some cases efforts are made to translate the requirements into design brief before the start of the process by system engineers. This gives a better clarity to start the process. However, in larger projects where many systems engineers involved, each of them analyses these requirements in their own way which again leads to multiple interpretations, ambiguity and then followed by sending request for information to the client.

Various stakeholders involved in design process mentioned that design process could be improved and be made more efficient by providing them with unambiguous, specific, translated requirements in a visual way in the form of object or requirements graph rather than the current textual way. This requires good allocation of requirements with the objects and spaces. The interviewees also indicated that apart from translation of requirements and linking to objects, several other information in phases such as design or construction, verification and validation plan, design domain (structural, architectural, MEP), responsible stakeholder and level of information needed in the BIM model for the verification also needs to be allocated .

Information Modelling and Management: Information modelling is governed by several guidelines issued by Dutch government, ISO standards and guidelines set by organisations for their internal workflow. However, due to inconsistent information modelling in the current process, using software applications like Relatics is a constraint in moving towards automation of verification process. Though a object and system breakdown structure is created and objects are linked to requirements, there is a lack of data about relationship between objects and their properties. There is also a lack of direct reference between BIM model and calculations, documents, and reports.

The participants in the interview implement BIM in their workflows mainly for design visualisation, clash detection between different design disciplines and to keep track of projects progress. The documents produced during these processes are again stored sometimes in Relatics document database and sometimes in common data environments like BIM360 or onedrive. Current system lacks connection with these documents. Creating a BIM model with all the relevant information is an intensive process and requires effort from all stakeholders. This can be achieved by making agreements during early stages on what information needs to be integrated in the model and this needs to be consistent during the entire project life-cycle. For simple automation workflows, the use of basic information models is sufficient. but for complex functionalities higher degree of information structure is needed.

3.6 Summary

The Systems Engineering process of dividing total system into layers of subsystem gives more control on individual parts of the project there by leading to an efficient design that meets the user needs when compared to conventional project management processes. The various activities spread over the SE life cycle ensures that the needs of the client stays in the priority by focusing on requirements of the project. Requirements Management process involves definition, analysis, validation and verification of requirement statements. Requirements classification approach followed in the current process is solely based on functionality and their intended use in the design, than, classifying it in a machine understandable way. Several problems in the management process such as ambiguous interpretation, lack of connection with BIM data is mainly due to poor definition of requirements.

Most of the research in relation to Systems Engineering in AEC industry, requirements management and automated requirements verification propose a theoretical solution rather than an actual implementation. Some of the research done by Eastman et al. (2009); Wu and Zhang (2019); Moonen (2017) have developed a practical implementation of the solution. However, these solutions are domain specific and is not scalable for the complete infrastructure design process (Beach et al., 2020). Underestimation of verification process is risky and can result in expensive mistakes (Leidraad SE, 2015). It can be concluded from several scientific publications, Dutch standards for information modelling, and from existing commercial software's that, semantic technologies and interoperable platforms can provide a potential solution for automation of requirement verification process. This depends on standardisation of domain knowledge and changing the way of storing requirements from current textual model to semantic model.

It can be concluded from interview results that the industry is moving towards a standardised way of working with the use of information modelling. To accelerate this shift, handling of requirements in a structured and consistent manner is necessary. Lack of standardised and consistent requirements management practices can have negative impact on success of the project. The negative impact could potentially be avoided by better integration of requirements, documentation, and design model. This solution is recognised by industry stakeholders and scientific works. Kiviniemi (2005) is one of the first researchers in this domain who recommended that linking requirements with data libraries and design representations (BIM model) should increase the traceability and transparency in the design phases of the project. Object wise connection between requirements and their relationship is preferred rather than systems wise connection. Consistency of information depends on the consistency of OTL and requirements data. Overall, this chapter helped in understanding concepts like Systems Engineering process, BIM enabled information management practises and was able to define problems and limitations of current requirement management process. The recommendation provided by Kiviniemi (2005) can be considered as an hypothesis for the project based on which research questions are defined in Section 1.5. This narrows the scope for solution identification. End of this chapter marks the end of first phase in the Double Diamond process of research methodology as explained in Section 2.1. Next chapters focus on the second phase that is solution exploration and development.

Chapter 4
Solution Exploration and Development

4 Solution exploration and development

This chapter focuses on solution development i.e. second phase of the double diamond process. Based on research question defined (Section 1.5), a review of existing solutions is made. Solution development patterns in various studies are identified and a framework from combination of existing solution is arrived.

The chapter starts with Section 4.1 reviewing currently available studies and applications in this domain. The section is divided into two, Section 4.1.1 academic research and Section 4.1.2 commercial applications. At the end, Section 4.1.3 gives an analysis about the available studies and identifies next steps in arriving at a solution for the problem at hand. Section 4.2 works on developing the solution based on the steps defined in the previous section. Proposed solution and new workflow needed for adopting the current process is provided in Section 4.2. The chapter is summarised in Section 4.3.

4.1 Review of existing solutions

A rigorous literature search process was conducted in finding relevant studies and platforms. Research studies from the last decade was considered because of fast moving technologies in this area. Initially it was conducted by searching through keywords like 'Automated Code Compliance', 'BIM checking', 'model validation', 'E-permit checking', 'infrastructure systems engineering' on various platforms such as google search engine, Google scholar, Researchgate, TU Delft library, TU Delft research repository, scopus et cetera. BuildingSMART[7] has a committee known as 'Regulatory Room' for guiding research on automated compliance verification. Committee reports about feasibility and business cases were studied and research papers through references mentioned there was also obtained. Since there are several sub-domains within this area that cannot be searched through keywords, manually went through each issue of the last five years of online journals like 'Automation in construction', 'Advanced engineering informatics', 'ASCE chapters' et cetera to obtain papers and reports. Last process in gathering articles was through the use of an online web source known as ConnectedPapers[8]. If a DOI or a title of an article is given then a graph with all the links to cited articles are generated. Though most papers were repetitive, some new research papers were obtained. This also validated the fact that most of the information obtained through the previous process is connected and relevant.

4.1.1 Academic studies

A total of more than three hundred studies conducted in automated requirement verification domain was identified. The studies included articles, conference proceedings, journal publications, master and PhD dissertation studies. Identified papers were classified based on related construction domain, technology being used and phase of the project the proposed solution is best suited to. Based on relevancy of the domain to infrastructure industry, 84 articles were selected for detailed study and of that, solutions from 33 studies which focused solely on the design phase were considered. The studies were catalogued in the form of table by noting down their checking methodology, unique features, input and output data and limitations of the study. The detailed review table can be found in Appendix D. The various previous studies conducted in this domain can be classified into four types. They are:

[7] https://www.buildingsmart.org/about/
[8] https://www.connectedpapers.com/

- **Review and consolidation articles**
  Over the years several academic studies have proposed solutions using various approaches. Researchers have consolidated these articles by identifying their common observations, reviewed the feasibility of the solution and provided road map to future research. These articles also identify common obstacles that authors have faced during the study period. This type of review studies are grouped under this category. (Beach et al., 2020) and (Pauwels, de Farias, et al., 2017) are example of studies in this category.

- **Predicate logic based approach**
  This approach uses semantic web technologies for the implementation of requirement interpretation process. Requirements/rules are represented in the form of triples using subject-predicate-object logic. Use of first order logic and description logic also fall under this domain. Description logic is the formal representation of knowledge using ontologies. It is used to describe and reason about relevant concepts in the knowledge domain. (Dimyadi et al., 2016) and (Soliman-Junior, Formoso, & Tzortzopoulos, 2020) are examples of studies in this category.

- **Object oriented approach**
  Object oriented approach is a knowledge organisation technique similar to description logic approach but information is represented by breaking them into smallest object level. In this approach, building data is established, interpreted and classified before modelling requirements/rules. Parametric and generative design uses this technique for design generation. (Sydora & Stroulia, 2020) and (Martins et al., 2016) are examples of studies in this category.

- **AI and ML based requirement inferencing approach**
  In this approach, machine learning techniques like natural language processing (NLP), convolutional neural network (CNN) and word classification are used for semantic enrichment of BIM data and for requirements to rules conversion. (Bloch & Sacks, 2018) and (Nagendrakumar, 2019) are example for studies in this category.

Each of these approaches was explored to accommodate various criteria required for developing an automated system and to identify pitfalls and limitations. Firstly, source and documentation of data was studied followed by study about the way data is represented. The last step in this process was to evaluate how the solution processes rules and produces output. Though each of these approaches used different technologies, a pattern can be identified in how requirement is prepared for checking. This forms the basis for developing requirements definition framework in Section 4.1.3.

4.1.2 Commercial applications

As reviewed in the previous section, there has been significant research in the area of requirement checking domain with various technologies. However, practical implementations of these technologies is very limited. The research gap mentioned in Section 1.4 also addresses this issue. Around ten practical implementations of automated requirement checking was identified, of that only two of them are in commercial phase and the rest remain in development phase. Various commercial implementations and their features are shown in Section E.

As mentioned in the research gap (Section 1.4), most solutions focus on specific building codes requirement rather than client specific requirement. Applications such as Upcodes AI[9] are built

[9]https://up.codes/?s

by hard coding requirements into application with very less flexibility for modification. These applications were not reviewed based on their actual use in projects since most applications are still in their development phase and due to requirement of licenses in accessing them. BriefBuilder and Solibri model checker was reviewed based on their actual use. Remaining applications working were reviewed based on publicly available information on their websites. In conclusion, commercial applications available offers very limited and general solutions for building industry. These applications requires high level of customisation before integrating them into current workflow.

4.1.3 Solutions analysis

Design model and rules are two critical elements for the model based verification process (Pauwels, de Farias, et al., 2017). Design model is prepared through several commercially available and open source BIM authoring applications. As identified from various studies in Section 4.1.1, converting requirements to rules involves a three step process. They are:

- Semantic classification of requirement data
- Semantic representation of requirements into rules
- Rule execution operations

The various approaches with different technologies, use the same three stop process for converting requirement to rules and then executing rules on BIM data. Requirements written by clients are written in natural languages like English, Dutch et cetera. The requirement author creates statement based on the final intended use of the system. The author sometimes refers to relevant object directly or might use a common name or use any other reference. It is part of systems engineer's workflow to identify which object the requirement belongs to and allocate it to those objects. Semantic classification is necessary here to identify the relevant objects and then connect it with properties of the object by eliminating the noise in the requirement statement.

The requirements classified semantically from the previous steps needs to be represented as rules. This can be represented as a sentence constraint, graph et cetera based on the technology used. The represented rules must now be applied on the BIM database or models to verify and check rules. Rules need to be classified based on its complexity and how the check needs to be performed. This allows the user to understand what kind of operation is required on a specific requirement and if it can be automated or not.

4.2 Solution formulation

The solution is developed based on object oriented approach (Section 4.1.1). Since the problem being addressed is about poor definition of requirements and finding a solution for mapping them with object data, no single solution that satisfies all the above mentioned steps correctly could be identified. However, solutions offered by few individual studies did satisfy few steps mentioned above. The proposed framework is a combination of approaches by individual studies of Hjelseth and Nisbet (2011), Dimyadi et al. (2016), and Solihin and Eastman (2015). Predicate logic based approach also provides an potential solution for mapping of requirements data but this requires users to adopt a completely new technologies like semantic web that is not very prevalent in industry practice. Existing solutions by companies like Semmtech[10] and Neanex[11] is working in this

[10]https://semmtech.com/
[11]https://www.neanex.com/

regard to make these solutions more user friendly. Hence, object oriented approach was selected for developing the framework.

Semantic classification of requirement statement:
Hjelseth and Nisbet (2011) proposed a RASE[12] approach for semantic classification of data from requirement text. The application uses hyper text markup language for reading text and tags each word as R, A, S or E based on the standard dictionary. For example, requirement text ”Minimum clear height shall be 2.25 m for the full width of the defined walking zone of the entire access route including crossing points” will be marked by RASE markup as shown in Figure 11. The above mentioned requirement is presented as example in the study (Hjelseth & Nisbet, 2011) and same example is used in next steps of defining the framework.

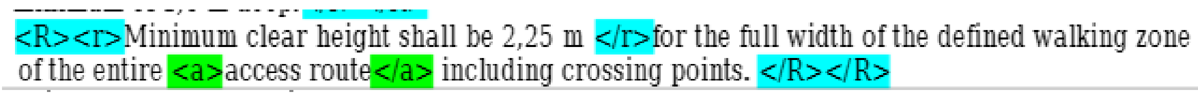

<R><r>Minimum clear height shall be 2,25 m </r>for the full width of the defined walking zone of the entire <a>access route including crossing points. </R></R>

Figure 11: RASE requirement tagging (Hjelseth, 2011)

This tagging is possible only if all the words used in the requirement text are present in a data dictionary with their meanings. The dictionary should also contain words belonging to different parts of speech like pronouns and prepositions. Creating this kind of dictionary is cumbersome. This method is also more suitable for building codes since most of the codes are defined in a standard language unlike project specific requirements. This is also the limitation identified by the author in the study. However, the method proposed by Hjelseth and Nisbet (2011) is suitable to reduce noise in the requirement text, and mark-up relevant objects and their properties. The data dictionaries can be replaced with OTL and as a result, this allows system engineers to look at object type properties from the OTL and mark-up the object, property, required comparator and target value. For the above mentioned requirement, from the OTL of bridge it can be identified that the requirement belongs to object ‘pedestrian path’, ‘crossing points’ and to their property ‘clearing height’. The comparator ’Equal to(=)’ should be used and the value is ‘2.25m’.

Semantic representation of requirement statement:
Dimyadi et al. (2016) proposed a conceptual graph model for representing requirements in the form of constraints unlike RASE approach where constraints were represented as marked-up sentences. Conceptual graph identifies each object as a node and are connected with other objects and properties using relations. Multi line requirements could be written individually and then can be connected using logical functions like AND, OR et cetera. For example, the requirement text ”Minimum clear height shall be 2.25m for the full width of the defined walking zone of the entire access route including crossing points” is shown in conceptual graph and is represented in Figure 12.

The solution recommends defining dedicated semantic relations based on the construction domain to connect various objects and properties. In the current case, two semantic relations ‘HAS’ and ‘EXCEPT’ is introduced. Relation ‘HAS’ is used to query the linked objects and properties of that instance. Relation ‘EXCEPT’ removes certain objects from the query. Unlike, the general way of semantic classification proposed by Dimyadi et al. (2016), the combination with RASE approach provides more standard way of marking-up and representing requirements with a common

[12]R-Requirement, A-Applicability, S-Selection,E-Exception

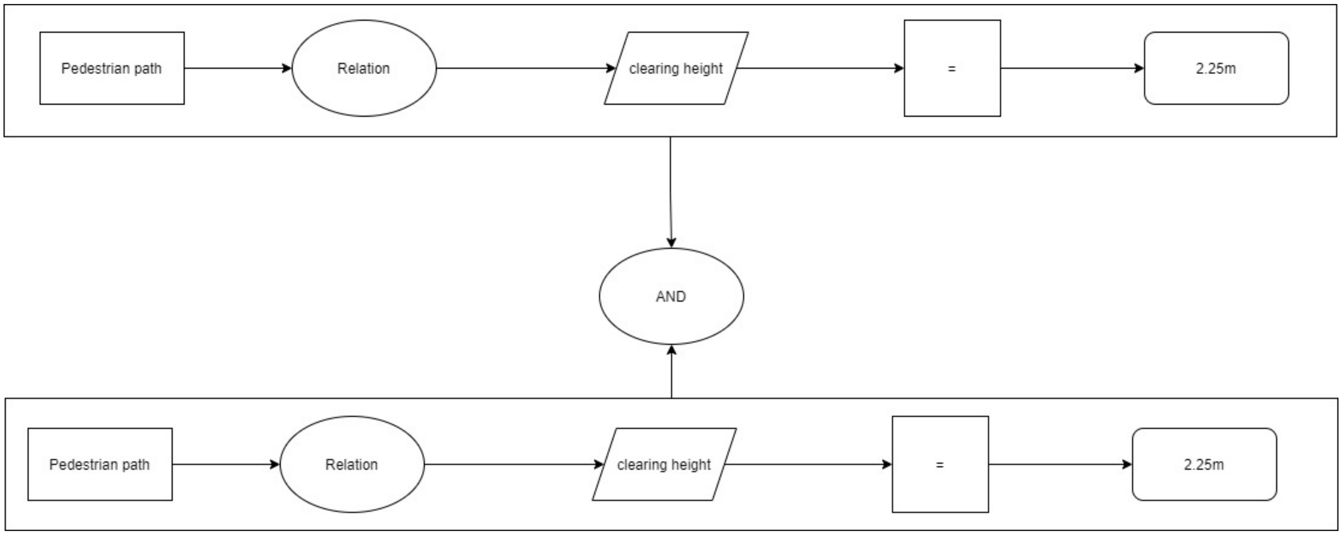


Figure 12: Semantic representation of requirement

vocabulary from OTL. With semantic classification and representation using new relations, the above requirement could be written as a constraint as shown in the Figure 13.

object:Pedestrian path HAS property:clear height Operator:= value:2.25m

AND

object:Crossing section HAS property:clear height Operator:= value:2.25m

Figure 13: Requirement represented as a constraint

Rule execution operations:

Requirements that are semantically classified and represented must be now identified based on required operations to be performed, complexity and possibility of being automated. A rule classification method is needed and this is adopted from the classification study done by Solihin and Eastman (2015). Four types of rule classes are introduced based on the data required for the check and the complexity involved. The four rule classes are shown in Table 3.

Theses rule classes are further explained below with examples:

Class 1: Rules that require a single or small number of explicit data

This class of rules checks explicit attributes and object references that exist inside the BIM model. For this level, information is explicitly available from the model objects and its associated properties. The data can be accessed with simple querying of data and relations.
Example: A bridge deck should be 250mm thick

Class 2: Rules that require simple derived attribute values

These rules deal with checks that are based on single values or datasets. New data need not be generated but multiple arithmetic or trigonometric calculations are needed to arrive at new values from existing dataset. Implicit relationships should be identified in this class of rules.

Table 3: Rule types

Rule class	Description
Class 1	Requirements that require single or small number of explicit data.
Class 2	Requirements that requires calculations from the existing data.
Class 3	Requirements that requires the creating of new data from existing data.
Class 4	Requirements that cannot be verified but requires a proof of solution.

Example: When the construction project is situated next to residential neighborhood, the protecting walls of project site should have a noise resistance level of 45dB

Class 3: Rules that require extended data structure
These rules require calculation from extended data. This means new data needs to be generated to execute the rules. The complexity of the requirement requires data to be generated in various ways. The execution of these rules relies on geometry engines, simulation engines and other complex algorithms and calculations.
Example: Emergency telephones and fire exits should be present in the tunnel at every 500 m interval.

Class 4: Rules that require a 'proof of solution'
These rules require proof of compliance instead of just complying with it. The complexity with these rules is limited.
Example: Smoke detectors used in tunnels should be according to NFPA 502 Section 4.3.1.

Above mentioned classification and representation of requirement in Figure 13 involves use of AND gate for verification and hence it can be classified as 'Class 2' type of rule. Semantically represented rules can be applied on BIM models using various applications such as visual programming, machine learning and other approaches. Two such methods are proposed in Section 5.

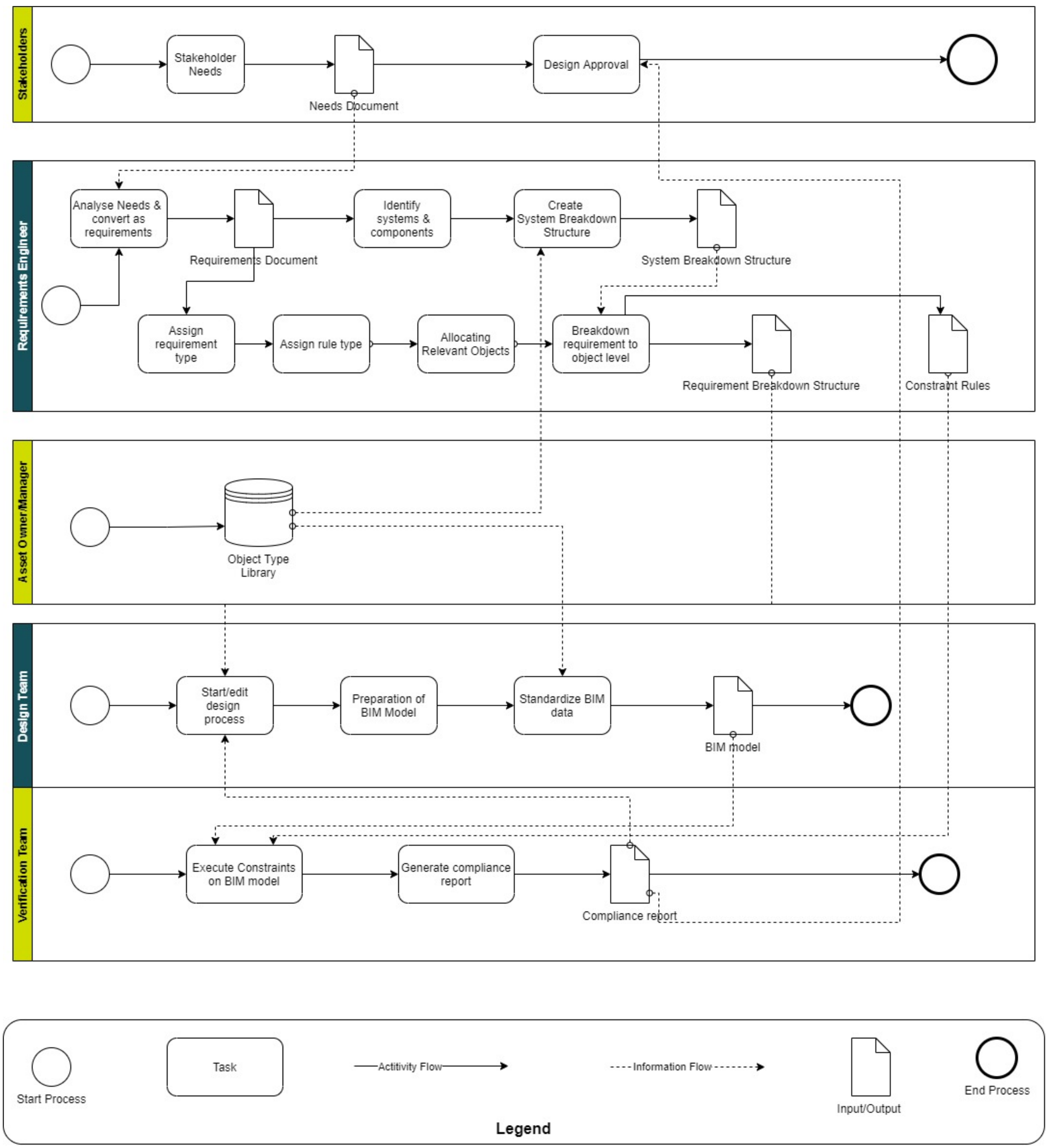


Figure 14: BPMN model for integrating proposed solution in the existing workflow

Proposed Workflow

In authors opinion, the currently proposed object oriented approach using OTL offers cost and time effective system for requirement definition. The proposed approach is more about working with current used tools in existing process than use of sophisticated new tools. The integration of the above proposed solution within the existing design process workflow is shown as a BPMN model in Figure 14. The existing workflow and proposed workflow are placed to next to each other

Appendix F to view the changes made.

Step-by-step process of the new workflow presented in BPMN model is enumerated below:

1. Stakeholders formulate the needs from new infrastructure and create a needs document.

2. The requirements engineer analyses the needs document and converts them to a requirement. From the requirements, system and components of the system that needs to be built are identified.

3. With object names from OTL as a reference, a system breakdown structure is created. Requirements are assigned a requirement type depending on whether its a value, objective or existence type checking along with the existing functional classification. Then, based on the complexity of verification, rule class(Table 3) is assigned.

4. Using object types from OTL, requirements are semantically classified (Section 4.2) and represented. Requirements are broken down to object level and constraints/rules are created.

5. The requirements is now passed to design team for the start of design process. The design team creates BIM model and assigns object type properties from OTL. Same object properties from OTL needs to be used for model and requirement mapping to achieve uniformity and consistency.

6. The BIM model is now checked by verification team using semantically represented requirements. The checking approach using commercial solutions shown in Section 5 could be adopted.

7. Compliance report generated is sent to stakeholders for design approval to move on to the next phase of construction.

The proposed workflow is implemented on a case study and is validated through expert assessment. Various issues that could be faced in implementing the workflow is identified in section 7.

4.3 Summary

Though, the current available solution uses different technologies, it uses common steps in arriving at the solution. Requirements are first semantically classified, represented and then checking is performed using rule classes. Existing academic and commercial solutions focus more on client building codes requirement than on the client specific requirement as mentioned in research gap. A workflow using object oriented approach from the combination of existing solutions is proposed. The workflow uses approaches proposed by Hjelseth and Nisbet (2011) for semantic classification by modification of the data source, by Dimyadi et al. (2016) for semantic representation by adding two more relation classes and by Solihin and Eastman (2015) for rule classification. With this combination, requirements can be represented as constraints. A common vocabulary across the requirements management process and BIM model is achieved with the use of object type properties from standard OTL.

Chapter 5
Framework Test Implementation

5 Solution test implementation

This chapter discusses the use of commercially available tools for mapping between requirements and object components based on previously proposed classification approach. The main use of the tool is to create a connection between the requirements text and BIM model for monitoring the process of requirement verification. The implementation shows proof of concept of the proposed solution for verification process.

Proof of concept is shown using two methods which is discussed in Section 5.1.1 and Section 5.1.2. A case study is presented in Section 5.2. The proposed workflow is applied on this case study to check the feasibility and workability of the solution. Though the solution using NLP is more practical and easy to use, because of certain limitations within the study it cannot be adopted for a case study. A visual programming approach is applied on case study.

5.1 Implementation approach

In Dutch infrastructure sector, requirements of a given project are stored mostly in Relatics application. Relatics web services offer API for accessing the work space. Relatics also allows users to download requirement and other connected data in excel format. Requirements data required for the current research is downloaded manually in excel format.

BIM model data can be obtained in an inter-operable way using industry foundation class (IFC) format. However, IFC format for infrastructure is still in development and complete details about the project cannot be exported. The poor details from open standard means that the data from BIM authoring applications are dealt with directly. The object data and properties for requirement allocation are downloaded directly from Rijkswaterstaat's extensive collection[13] of objects.

There are several applications through which the proposed framework could be implemented. Some of the implementation approaches other than the one's proposed in next section is elaborated in Appendix G. Visual programming approach using Dynamo was chosen since Royal HaskoningDHV as part of their daily work process uses Dynamo for several automation tasks. As mentioned above BIM data is dealt with directly and Dynamo as an integral part of Revit application gives an added advantage to access and query data. This makes, use of Dynamo more relevant in testing the framework. The use of visual programming also comes with certain limitations. Hence, a different to directly execute the rules was tried. NLP using Table Parsing (TAPAS) approach was chosen since with this method rules can be executed directly on the BIM database without the need for programming individual rules.

5.1.1 Use of visual programming

Visual programming is a programming interface that allows users to create machine executable programs through graphical elements. Use of graphical elements makes interpretation simpler and quicker for users. In recent years, several visual programming applications are developed dedicatedly for AEC industry. Grasshopper for Rhinoceros3D[14], Dynamo for Autodesk Revit, Marionette Vectorworks are few of the applications. The primary feature of these applications is to support parametric modelling. However, several efforts are made in using these applications for other analysis. Visual programming scripts contains a network of nodes. Each node has a input

[13] https://otl.rws.nl

[14] https://www.rhino3d.com/

and output and performs a dedicated defined function. The semantically classified and represented requirement can be represented as visual programming nodes as shown in Figure 15.

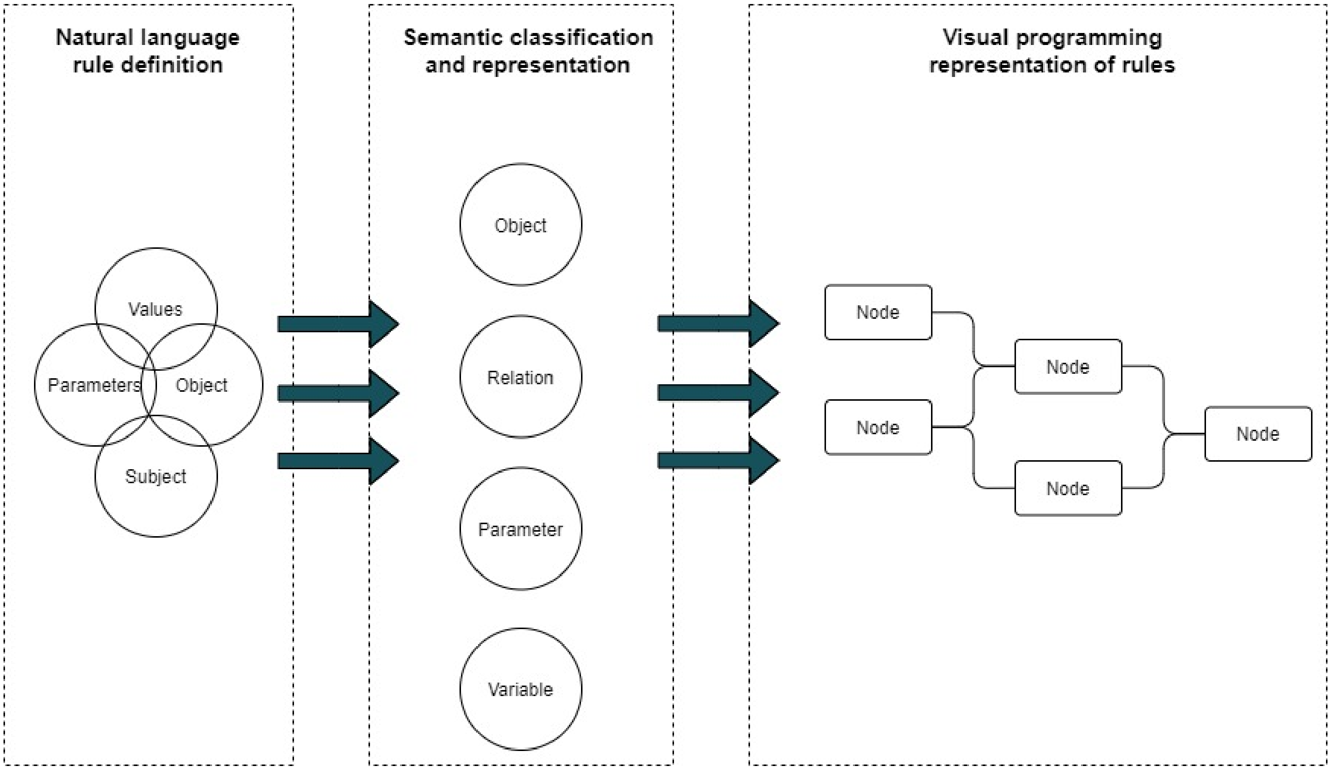


Figure 15: Requirement to visual program representation process. Adapted from (Ghannad et al., 2019)

Dynamo for Autodesk Revit has been used in implementing the solution because of following reasons. Firstly, Dynamo is available within Revit application which is a majorly used BIM authoring application within Royal HaskoningDHV. Secondly, no programming language knowledge is needed for creating Dynamo scripts. However, the programming level of thinking helps in creating efficient network of nodes. Lastly, the scripts can be easily created by anyone who has basic understanding of the Revit object structure of the model.

The first task in creating rule script is to select building elements from the model. For example, in case of the rule in Figure 13, the object pedestrian path and crossing section needs to be selected. No currently available nodes in Dynamo is able to do this in a simple way by taking name of element in the form of a string. Hence, a custom node was created to select elements from their names. The custom node is shown in Figure 16.

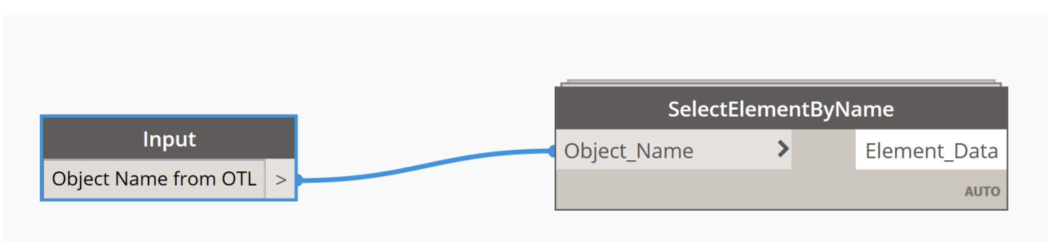


Figure 16: Custom node for object selection by name

Next step is to query property information of the object. Related information of the object can be obtained by 'ParameterByName' node. Output data from this node is considered as input for checking. Logical relations required to implement the rules are provided by standard logical nodes such as AND, OR, IF_THEN et cetera within Dynamo application. For example, in Figure 13 the parameter clearing height is obtained by providing it as a text as shown in Figure 17.

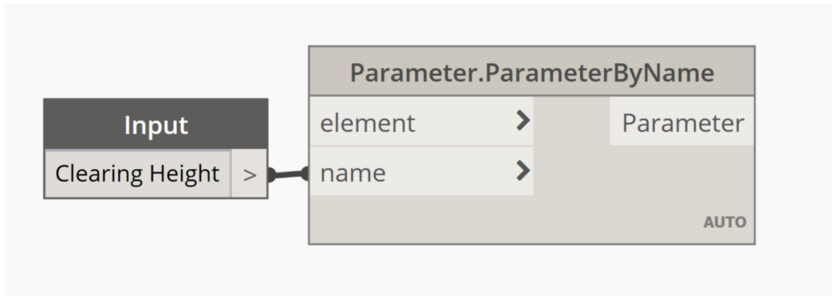


Figure 17: Node for parameter selection of the object

The logical operator 'Equal to(==)' is used to compare it with the value in requirement text. Logical operator 'AND' can be used to connect both the requirements. These predefined nodes help users in creating simple automation scripts and provides users the flexibility in the rule writing approach. The complete Dynamo script for the example is shown is Figure 18.

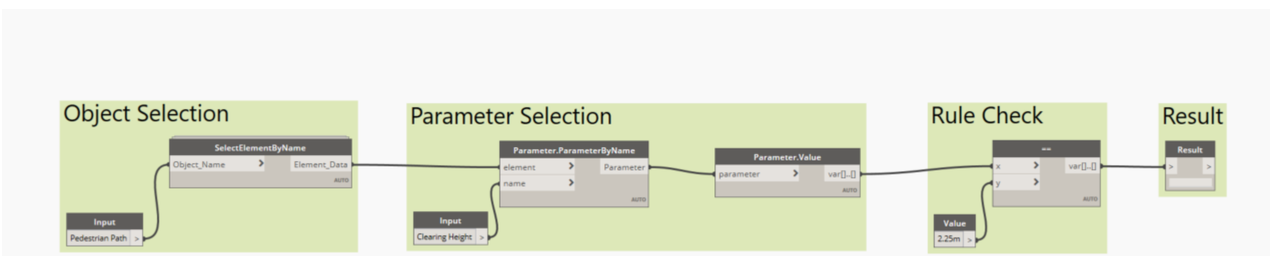


Figure 18: Dynamo representation of rules

All the executable Dynamo scripts and BIM model referred in the thesis are made available in the Github repository of the author which can be accessed through the link provided in this footnote[15]. Detailed implementation of Dynamo scripts for various rule classes is done on a case study in Section 5.2. Through the use of interface like Dynamo player for executing Dynamo scripts, verification team need not go into programming mode for running the scripts. A sample Dynamo player interface is shown in Figure 19.

[15] https://github.com/Shreenidhi95/MasterThesisShreenidhi

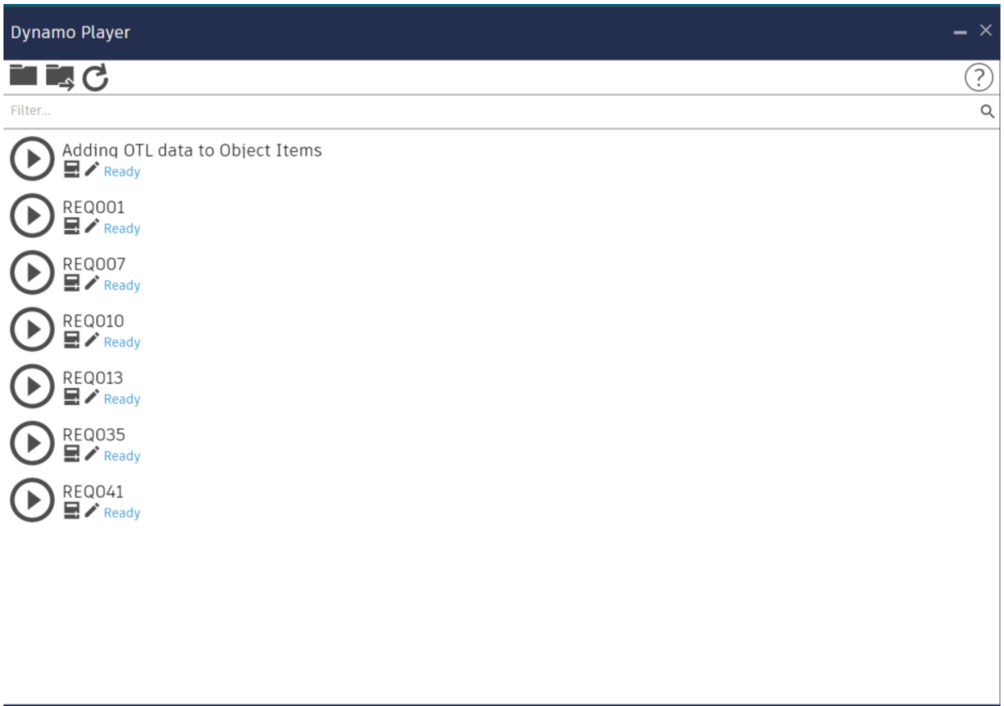


Figure 19: Dynamo player interface

The visual programming method of encoding texts into rules require domain experts in the process and not an ideal replacement for manual based checking but they provide transparency, flexibility and portability (Ghannad, Lee, Dimyadi, & Solihin, 2019). However, it validates the fact that based on the framework semantically represented requirement using data from OTL can be verified programmatically.

5.1.2 Use of natural language processing(NLP)

There are several studies using NLP to analyse requirements and information extraction. In the current research, information extraction is not the goal since it is already semantically classified and represented in the way of constraints. Since BIM model could be extracted as a database in the form of table, it can be parsed using natural language to obtain answers for the constraints.

The table parsing (TAPAS) algorithm proposed by Herzig et al. (2020) is used for developing a proof of concept in this research. The TAPAS is a collaborative effort from researchers at Google research[16] and Tel-Aviv university. TAPAS is an question answering approach over tabular data. Instead of conventional NLP approach of using ideas and emotions from the text, the TAPAS method tries to reason over the tabular data by literal or primary meaning of words. A sample data used in case study was exported from Autodesk Revit into an excel format and is then processed to adapt it to the TAPAS algorithm. The table and query made in the TAPAS programming interface is shown in Figure 20.

Results from algorithm had very less accuracy and precision. Algorithm also takes longer processing time for generating results. For a simple query showed in Figure 20, algorithm took 4m 30s for processing. This could be attributed to poor domain specific training data. The TAPAS algorithm is pre-trained using general training data available in the public domain[17]. A training

[16] https://research.google/
[17] https://github.com/google-research/tapas

```
result = predict(
    """ Properties           | Bridge foundation | Bridge columns | Bridge girder
        Assembly code        | VD001             | VD002          | VD003
        Anti graffiti coating | TRUE             | TRUE           | TRUE
        Beam visibility      | FALSE             | TRUE           | TRUE
        Bottom length        | 0                 | 0              | 140
        Bottom lighting      | FALSE             | TRUE           | TRUE
        Bottom paint         | FALSE             | FALSE          | TRUE
        Brick type           | NA                | Groningen red  | NA
        Cable ducts          | FALSE             | FALSE          | TRUE
        Chloride protection  | TRUE              | TRUE           | TRUE
        Column space material | NA               | Basalt split   | NA
       """,
       "Bridge foundation has Anti graffiti coating")
```

```
is_built_with_cuda: True
is_gpu_available: False
GPUs: []
Training or predicting ...
Evaluation finished after training step 0.
Completion time: 4m 30s
TRUE
```

Figure 20: Query and results from TAPAS interface

data specific for construction domain is needed for improving the accuracy, precision and efficiency of the algorithm. This approach was not further considered in this research due to limited scope and time constraints of the study. Table parsing using NLP will be more feasible feature unlike visual programming approach where manual work is again needed to create Dynamo scripts. It is one of the recommendations of this research to explore the feasibility and implementation of this approach in future studies which is discussed in Section 7.1.

5.2 Case study and results

The verification of the proposed framework is carried out using data from an infrastructure project. A quantitative analysis of the data is initially carried out to determine the type of requirements and rules. A set of requirements is then selected to present the requirements in the form of rules and then a Dynamo script is created for verifying it through a BIM model. This is followed by qualitative expert validation in the section 6 to check the applicability of the framework.

As a case study, new viaduct from Aanpak ring zuid construction project in Groningen is used. The viaduct is constructed over the proposed new highway to connect the existing cycling path and pedestrian path. The project in total had 86 requirements ranging from design to handover phase. The design model of the structure is shown in Figure 21.

Based on the proposed framework, to semantically classify and represent requirements, it was first assigned unique IDs to identify and track. Requirements from the project are available in Appendix H. Requirements were assigned a requirement type based on whether its a value, object or existence type and further, based on verification plan, the requirement was assigned one of the four rule classes. The requirements are spread in three project phases. They are:

- **Design phase:** It is when a lot of alternative solutions are analysed and narrowed down to find the most effective and efficient technique for construction.
- **Construction phase:** The building and project scope are completed during this phase.

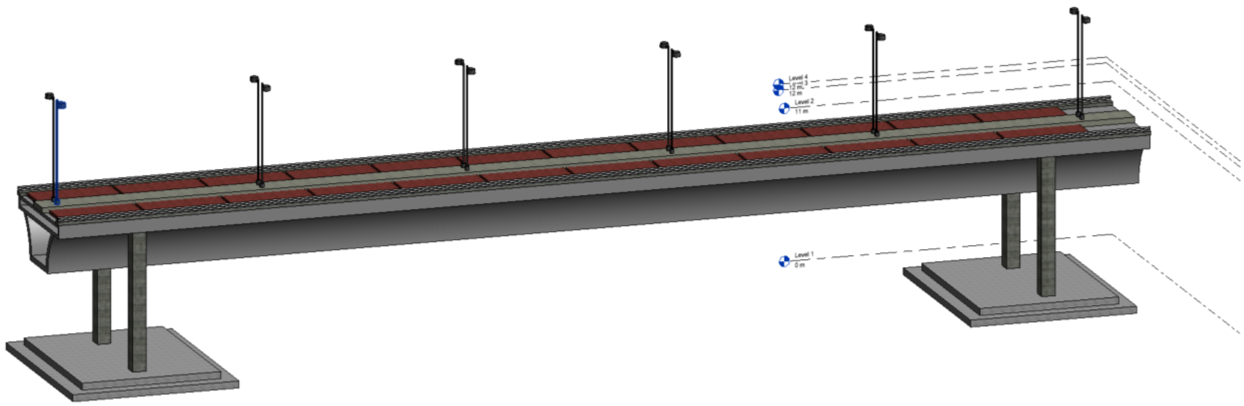


Figure 21: 3D model of Viaduct based on requirements from Aanpak ring Zuid project

- **Handover phase:** Handover (or Turnover) is a responsibility transfer procedure in which the project owner is given care, custody, and control (CCC) at the end of the project after it has been built, inspected, and tested.

The number of requirements in various phases of the project are presented in table 4.

Table 4: Number of requirements in different project phases

Verification Phase	
Design phase	78
Construction phase	6
Handover phase	2

Each requirement has a verification plan. There are four verification methods identified for this project. They are:

- **Document inspection:** Demonstrate the requirement verification on design drawing, maintenance and management plan.
- **Calculation:** Demonstrate the design calculations determined using methods mentioned.
- **Analyse:** Analysis by means of design, drawings or computer files.
- **Review:** Demonstrate the requirement by reviewing the design drawings in detail

The number of requirements with above mentioned verification plan is shown in table 5.

As defined in the proposed framework, requirements are given a requirement type and a rule class is assigned. Various requirement and rule classes in the current project are shown in Table 6 & Table 7.

Table 5: Number of requirements with different verification methods

Verification Methods	
Document inspection	28
Calculation	11
Analyze	18
Review	29

Table 6: Number of requirements with different requirement types

Requirements Type	
Value requirement (A value of certain parameter needs to be checked)	18
Objective requirement (Requirement relating to design principles and guidelines)	40
Existence requirement (Requirement to check if certain objects and properties exists or not)	28

Table 7: Number of requirements with different rule type

Rule Type	
Rule 1: Requirements that require single or explicit data	30
Rule 2: Requirements that require derived values	13
Rule 3: Requirements that require extended data structure	17
Rule 4: Requirements that require proof of solution	26

The next step of framework involves identifying system components and allocation of requirements. After identifying system components, a system breakdown structure is created from the existing OTL of Rijkswaterstaat. The detailed requirements analysis performed is available in Table 16. Using system breakdown structure as baseline, requirements are now rewritten as constraints as mentioned in proposed framework. Few requirements belonging to various rule classes are presented below. Dynamo scripts are written for these constraints and is executed using Dynamo player on the BIM model. The executable Dynamo scripts can be found here[18].

RULE 1: Requirements that require single or explicit data

Requirement ID: REQ002

Text: Structure shall have a design life of at least 100 years in the case of new non-replaceable structural components.

[18] https://github.com/Shreenidhi95/MasterThesisShreenidhi

Description: The requirement expects that new permanent components of the structure shall have a property design life with 100 years as value.

Compliance result from Dynamo player

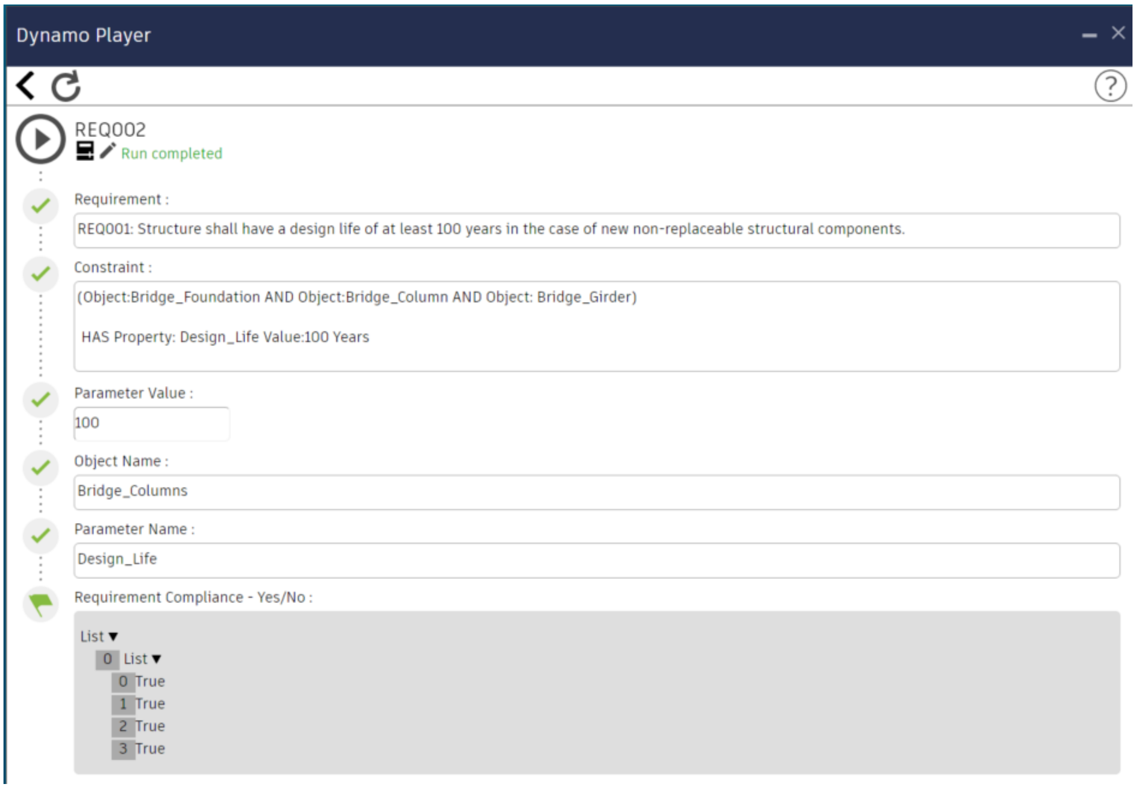


Figure 22: Compliance result of REQ002

Requirement ID: REQ007

Text: Structure shall provide space for an inspection path of at least 500 mm wide behind vehicle barriers and along the side verges with 2,200 mm in height.

Description: The requirement expects that structure shall have an pedestrian path behind cycling path of 500mm width and also shall contain a side barrier with 2200mm height.

Compliance result from Dynamo player

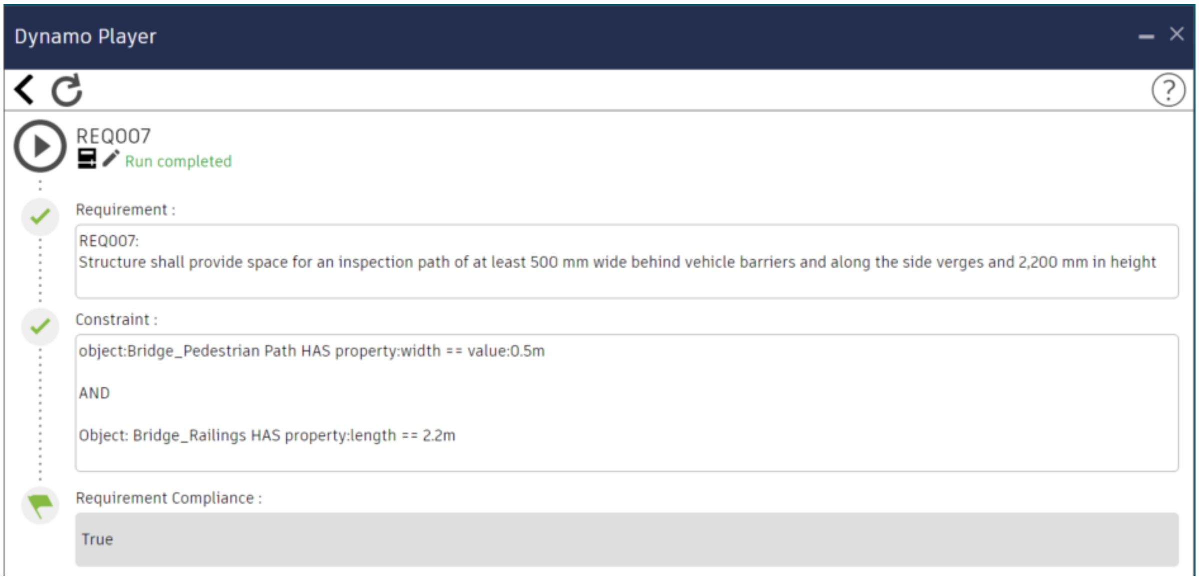


Figure 23: Compliance result of REQ007

RULE 2: Requirements that require derived values

Requirement ID: REQ013

Text: The bike path and pedestrian path are at the same height and / or in sight of each other.

Description: The requirement expects that bike path and pedestrian path to be at the same level from surface and both objects shall be visible to one another.

Compliance result from Dynamo player

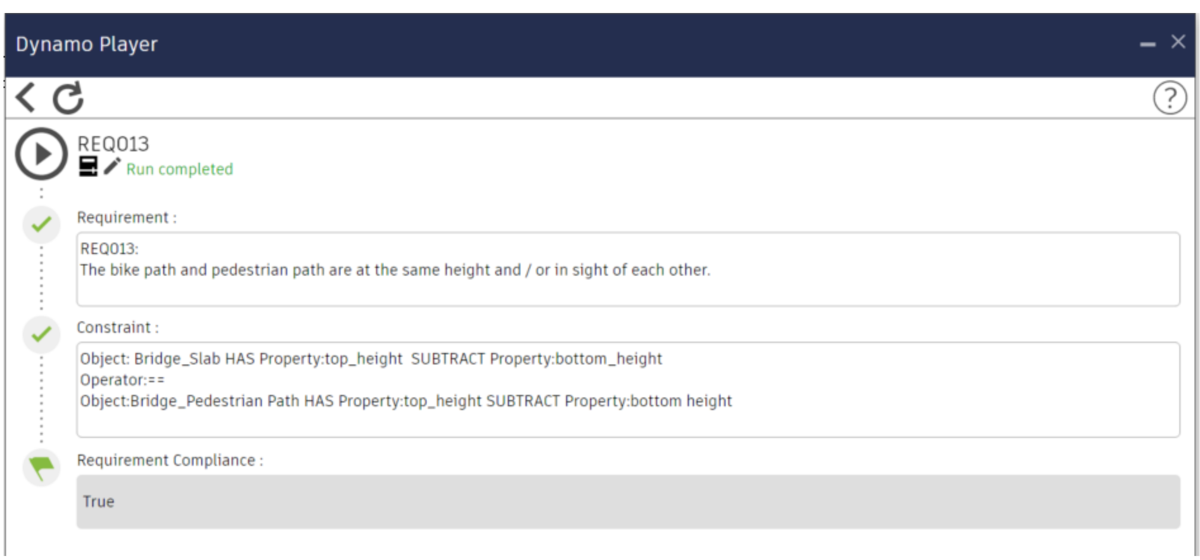


Figure 24: Compliance result of REQ013

Requirement ID: REQ060

Text: Structures shall not have a narrow width and height profile.

Description: The requirement expects that the horizontal and vertical width of the structure should be equal and does not taper or get narrowed at the end.

Compliance result from Dynamo player

Figure 25: Compliance result of REQ060

RULE 3: Requirements that require extended data structure

The rule 3 types of requirements require more complex simulations and logical calculations for verification. Requirements of this type requires checking platform to have a simulation and logical capability of high order. Automation of this requirements can be focused depending on the cost and effort involved. Integrating simulation platforms into checking environment requires a custom software development which involves high costs. If there are limited requirements of this type then the development effort is not justifiable. A study could be done here on what level of simulation capability can be adopted in the checking environment. Few such requirement examples are given below.

Requirement ID: REQ014

Text: The structure shall be resistant to vandalism, in the sense that it contains no loose parts or parts that are removable with simple hand tools.

Description: The requirements expects that structure shall be vandalism proof and shall contain only fixed parts that are not easily removable.

Requirement ID: REQ074

Text: Structure should be designed in such a way that rainwater shall first flow in the transverse direction, is discharged and then in the longitudinal direction.

Description: The requirements provides constraint on drainage flow in the system.

RULE 4: Requirements that require proof of solution

Requirement of rule 4 type cannot be automated and requires a documentation support. This could be done by providing an option for document upload.

Requirement ID: REQ010

Text: Structure shall be inspectable constructively using non destructive methods.

Description: The requirements expects that structure shall be inspected only using non destructive methods.

Compliance result from Dynamo player

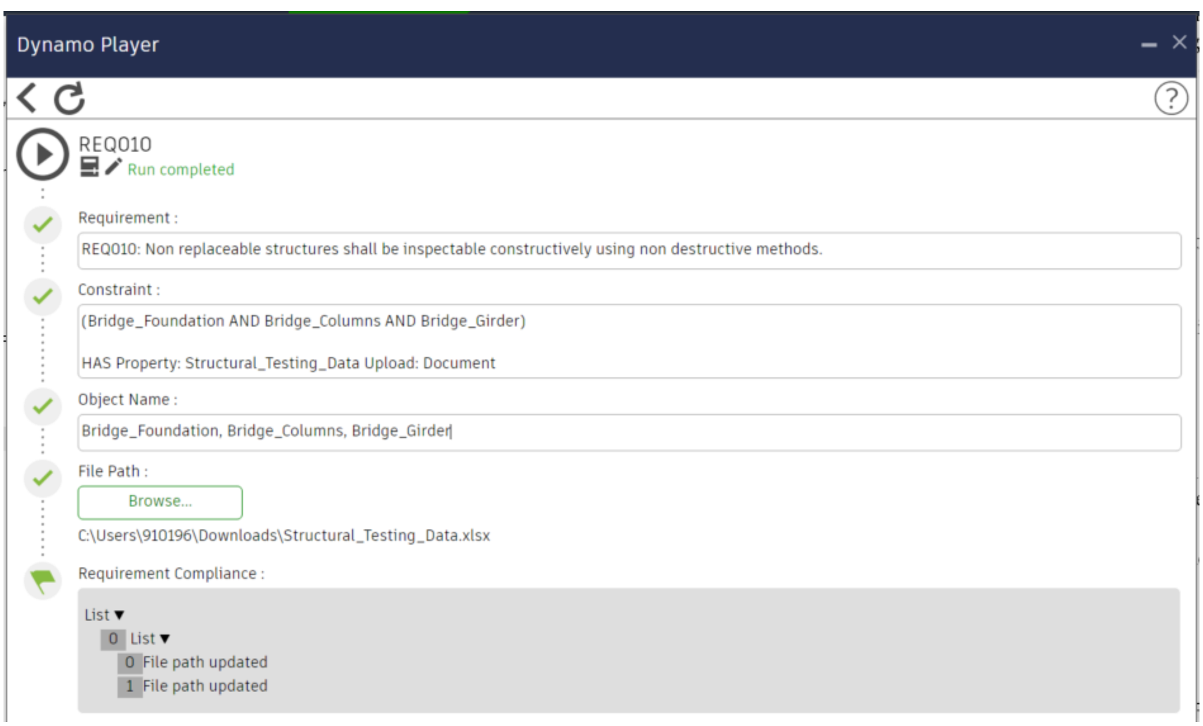


Figure 26: Compliance result of REQ010

5.3 Summary

Main goal of the implementation proposed here is to verify that proposed framework for semantic classification and representation is usable for requirement verification. Results from test implementations forms main data for validation of the research. It can be observed from solution approaches that with simplified requirement, more precise object allocation is possible. Constraint checking implementation was proposed using two approaches. The first approach is through visual programming and second through use of NLP and table parsing. The visual programming scripts were created using Dynamo for Autodesk Revit. Clear object and parameter definitions along with custom nodes made it easier to write Dynamo scripts which was also evident from the case study. However, some level of expertise in programming and domain knowledge was needed in creating these scripts. With the use of Dynamo player, verification team can execute requirement constraints without accessing Dynamo scripts. The ideal approach for verification using constraints is use of NLP table parsing. With relevant pre-training data, algorithm can be trained for more accurate results. This approach could be used by all project team members without requiring to have specific knowledge of algorithms. Both approaches presented in this chapter are very limited in its development and functionalities explored. Future research is required for identifying their complete potential.

Chapter 6
Expert Validation

6 Expert validation

In this chapter, the framework proposed in Section 4 and proof of concept presented in Section 5 are validated through experts by interview approach. The goal of validation interview is to understand the practicality and workability from the perspective of practitioners in the industry. Validation methodology adopted and outcomes from the interview are discussed in Section 6.1. The summary of validation exercise done is provided in Section 6.2.

6.1 Interview and discussion

Interviews were carried out with series of practitioners from Royal HaskoningDHV and a software developer for AEC sector based in the Netherlands. The practitioners from Royal HaskoningDHV included system/requirement engineer, project manager and BIM manager for the projects. System engineer was chosen for validation since they are the users of the framework. Project manager and BIM manager are involved early in the project phases. From the project perspective they should be aware of requirements and their right interpretations. The software developers mainly validate framework in terms of feasibility to develop it as a practically applicable solution in the existing Systems Engineering workflow.

The framework was assessed based on utility and applicability criteria suggested by March and Smith (1995) and adopted in a similar framework development study conducted by Soliman-Junior et al. (2020). This method proved effective in validation of the research for tool development without a fully functional demonstration. The criteria are set to analyse current and future usability and applicability of the study. The interview panel was presented with the proposed framework, test implementation and results from the implementation. An open interview structure was adopted during the interview process. The interviewees were asked to provide their views on the framework and implementation based on the criterion's provided below one by one. The highlights of this discussion was considered as qualitative validation and is explained in the next section. The interviewees list and details is provided in Appendix C.

- Use of geometry and semantic information
- Consistency of the requirement modelling process
- Information traceability and transparency
- Ease of use for stakeholders
- Added value to the current process and future prospects

Use of geometry and semantic information

All four practitioners agreed that use of OTLs for representing requirements makes it more explicit in terms of explaining the expectations of requirement and what object and properties of the requirement belongs to. This is currently a manual process and requirement engineers spend hours on discussing the requirement and object allocation. Requirement engineers identified that the allocation discussion could be now traced back with use of OTL. The project manager and BIM manager identified the fact that, with the implementation of framework, all the stakeholders use a common vocabulary when referring to any object, instance or parameters and this makes communication consistent and unambiguous. Software developers felt that the reliability of geometric information is high since this data will be originating from BIM authoring applications but semantic information needs to be entered by modellers and reliability and precision of the data might be less here. This problem gets bigger when two or more organisations with different workflow collaborate with each other. The task of getting everyone on common platform using same OTL will be difficult.

Consistency of the requirement modelling process

Project manager and BIM manager identified from the proof of concept and case study that with use of framework, requirements tend to become standard and uniform when compared to the existing process. However, requirements engineer felt that some requirements are difficult to model in this way in the initial design stage because of lack of information. Requirements Engineer is also of the opinion that converting regulatory requirements in this way would be more time consuming. However, it was quickly argued by other panel members that converting regulatory requirements is a one time task and represented rules are reusable in other projects. Member from IIM group of Royal HaskoningDHV saw the proposed framework as a business case in standardising requirements interpretation and writing, but also mentioned that wide scale implementation and testing is needed before the business case could be justified. Practitioners identified that the framework brings a systematic and structured approach for writing requirements. This makes requirements interpretation less ambiguous for all project team members involved. Because of a structured approach, the checking process should be more efficient and less prone to error. Requirements engineer identify that framework becomes more useful only when large OTL with detailed properties become available.

Information traceability and transparency

The framework makes implicit data in the requirements visible through it's systematic semantic classification approach. The information's transparency improves as a result of constant effort required to deal with semantic data. The practitioners pointed that more detail about information traceability could not be identified at this stage and implementation in practice with varying kind of projects may bring out more obstacles.

Ease of use for stakeholders

All four practitioners agreed that the proposed framework is valid and is applicable to infrastructure and other building projects within Royal HaskoningDHV. There was also a positive response about integrating it within existing workflow without need for additional work resource. Project manager felt that the proposed framework would apply to all of their projects but however the inability of framework to deal with top level requirements prevents complete implementation. Feasibility of the framework when requirements are in Dutch was also discussed. System engineers felt that the framework would be applicable even with manual check since all the requirements needs to be checked before verification.Software developer was of the opinion that more studies on different scales of the project needs to be conducted to ensure credibility.

Added value to the current process and future prospects

The interviewees stated that the proposed framework has followed a logical approach in developing it and is effective for overcoming the current limitations proposed in subsection 1.3. Project managers felt that the proposed framework could help in building towards a more efficient verification and validation procedure. System engineers is of the opinion that the framework could help in building more detailed OTL at a faster rate. Software developer though identified the potential in developing it as a stand alone system, more efficient approach would be building it as a add-in to the BIM authoring applications. This could potentially beneficial in developing it as a virtual design assistant.

6.2 Summary

The chapter starts with explaining the proposed validation method, criteria and how the expert panel was setup for validation. Validation is based on usability and applicability of the framework and proof of concept. The criterion suggested by March and Smith (1995) and implemented in a similar study done by Soliman-Junior et al. (2020) is used. Each criteria point was taken as a discussion point to understand the interviewees perspective. Each practitioner saw the value of the framework in a perspective from their current workflow. A more qualitative analysis of the framework needs to be performed by actual implementations. Overall, positive response was received on the workability and applicability of the framework within the existing workflow. Different practitioners identified the practical future applications and prospects of the solution for moving towards a automated system. More research on NLP based checking was recommended.

Chapter 7
Discussion, Conclusion and Recommendations

7 Discussion, conclusion and recommendations

The chapter starts with discussions on the proposed workflow, implementation approaches and results from the case study. This is elaborated in Section 7.1. The section along with discussion of above points also highlights the limitations of the current research. The next Section 7.2 answers the research question and provides conclusion to the research. Section 7.3 provides recommendations for Royal HaskoningDHV and future research in this area.

7.1 Discussion

The aim of the research was to identify a way for automating the repetitive task of Systems Engineering based requirements verification process in infrastructure projects. However, the initial literature studies and current process analysis identified that a standardised machine and human understandable way of requirements definition is needed for automating the process. The research is practice oriented and goal was to identify the existing solution or combination of existing solutions to solve the problem without the need for complex and sophisticated new tools.

The research started with collecting academic solutions available from the last decade and finding commercially available solutions. The identified solutions were classified and catalogued based on their usage, features and limitations. The thesis proposes an framework and workflow for defining requirements using object type properties from Object Type Libraries. The framework was developed based on combining three existing research solutions. The framework provides a standardised functions and operators through which requirements could be described along with object type properties from OTL. After analysis, requirements are rewritten by semantically classifying them on which object and property it belongs to and is represented as constraints or using conceptual graphs. The framework further provides a requirement and rule classification approach based on which requirements could be classified based on the verification type and complexity. The implementation of this framework in the existing workflow reduces the need for all stakeholders involved to understand entire system breakdown structure. The final users can also communicate more effectively using a common language of OTL when describing about an object, property or process. This makes understanding of requirements easier, unambiguous and less error prone.

The requirements definition framework was verified by implementing on a infrastructure case study using available solutions. Several implementation approaches as discussed in Section 5.1 and Appendix G. Visual programming approach using Dynamo was initially chosen since Royal HaskoningDHV has part of their workflows uses it dedicatedly. The BIM managers were able to easily able to convert the given semantically represented requirements into a programming script unlike when given actual requirement sentences where they have to analyse what the requirement means, study verification plan, identify objects and then create a script. The drawback with this approach was that a programming script is needed to write for each requirement which is an equivalent effort as manual verification process. The process of creating scripts could be justified if the same requirements needs to be verified multiple times during the phases of the project. Not all constraint rules could be easily expressed as a Dynamo scripts. Obtaining required data in a correct format using existing nodes is sometimes complex and requires additional operations like data transpose to be performed. To solve this several custom nodes were created during the case study implementation. There are several researches now in the domain of automated node generation based on the core data of the model which can make the Dynamo scripting more easy in the future. For the current working practices, Royal HaskoningDHV can conduct dedicated study on what type of custom nodes is needed and to create them for adopting in their workflow. Another option to explore in the direction of Visual programming is automated generation of nodes using Revit APIs. Direct use of textual constraints can makes the process easier. With this hypothesis, NLP Table parsing approach was implemented using Google's BERT NLP approach. Essentially it is an algorithm for querying data in tabular format. To test this approach, the BIM data from Revit application was exported in tabular format using Dynamo script. The data in table was provided to NLP algorithm and rules were executed on this data. The results from the algorithm was very poor. This can be attributed to the fact that NLP program was not trained using construction domain data. The execution times for the program was also very high. The precision of the program could be further improved by training with relevant data but creation of this training data requires significant effort. The labelled data models for general AI Models are established now for several years but domain specific labelled data models are not available. The algorithm analyses the semantic relationship between the words and tries to answer the question however a method needs to be developed for identifying logical relationships and logical order of checking. For requirements verification, the translation of requirements to rule forms a critical phase. The proposed framework tackles this issue. More case studies with various technologies is needed for identify a suitable implementation approach for its implementation in the workflow. The major contributions from this research are highlighted below:

- A clear evaluation of current design process and information management in infrastructure domain.
- Study of prominent existing academic and commercial solutions to provide theoretical basis for mapping requirements to object instances using various approaches.
- An expressive guidelines for representing requirements as machine understandable rules/constraints.
- A proof of concept using available applications for executing constraints developed on BIM model.
- A recommendation for implementing the proposed framework using software tool kits.
- A list of technical and organisational challenges for enabling towards automated process.

The below sections discusses few of the limitations of the research:

- The proposed framework was developed based on analysing requirements from several infrastructure case studies of Royal HaskoningDHV. Though a broad range of projects were chosen for this case studies, an effort is required for implementing it in other organisation or in other kind of projects.
- Greenfield projects were taken into consideration when developing the framework. Requirements from renovation, operation or maintenance phase were not considered.
- Requirements from project which was in Dutch language initially was translated to English for use in the project. The proposed framework is created based on the assumption that project requirements are in English. The feasibility and modification to the framework needed for implementation of this framework in other languages needs to be explored.

- Systems Engineering process identifies and allocates the requirement at various project levels. If requirements are in high level i.e. at system level them rewriting them as suggested in the proposed framework proved to be difficult. This is because with use of OTL focus is at object level.

- Arity and combinatorial issues: It issues occurs from the fact that requirements are hardly complied by only one answer and multiple rules needs to be applied on one element. The ability to answer a requirement should be considered properly. Ignoring this issues may lead to false negatives with inconsistent results.

- It was observed that, the proposed framework seems to be more efficient when constraints are written for one specific object. With two or more objects, Rule 2 type of calculation needs to be invoked even for simple explicit data checking. If two objects in turn have several sub-components then constraints become more complex.

- When similar object types are involved for multiple type of requirements, same object needs to be retrieved several times. This hampers efficiency of checking system and may also return in false results. However, this issue could be solved by prioritising requirements during the checking phase.

- The framework and results from the case study executed by authors were presented to industry practitioners for validation. A dedicated workshop needs to be carried out with industry practitioners to identify more complex requirements that could not be written. This also helps to identify any hidden patterns in the requirement definition.

- Due to limitations of time and resources, not all solutions could be reviewed by testing it individually. This can also in the future helps us to identify a more feasible existing solution suitable for the Royal HaskoningDHVs needs.

7.2 Conclusion

Research started with the aim of automating Systems Engineering based requirements verification process. Before automating any process, standardised representation of information stream is required. This was a major obstacle that was immediately identified in the current requirements management process. The major obstacle for automation was poor definition of requirements which becomes a limitation in linking requirements with BIM model data. A main research question was formulated about mapping requirements with object data and verification compliance. Four sub-research questions were formulated to arrive at an answer to the main research question. The answer to sub-questions based on research findings are discussed below:

What are the different types of requirements in infrastructure design process and how are they managed and documented?

This research question can be answered by studying the current design process and investigating on how documentation and classification is done. The design process model (Figure 1) was arrived based on the workflow adopted by Royal HaskoningDHV. It was further standardised by interviewing project team members from other organisations. In majority of projects in the Netherlands, requirements and related documentation is stored in Relatics application. The data structure of Relatics is similar to that of a relational database. No changes were suggested on the current documentation process. However, in the current process, some documents about projects are stored in Relatics and documents about BIM model are stored in common data environment applications like Autodesk 360. Interoperablity of data being biggest problem with native applications, it becomes difficult to connect the data across various applications. Project teams should try to limit the number of applications being used for data storage and try to find platforms that could connected with each other. Royal HaskoningDHV is making an effort in this area with the use of Speckle[19] application which helps in efficient transfer of data between various applications.

Based on research conducted, in infrastructure domain, requirements are classified using several approaches. INCOSE (2015), Leidraad SE (2015) proposes different types of classification depending on the function to be realised. Physical, functional and non-functional requirements, RAMS[20] are some of the currently used classifications. However, these classifications are based on expectations of requirements rather than representing them in a machine understandable way. This is a major drawback in the current system since applications cannot distinguish between quantitative differences among requirements. To mitigate this, a requirement classification based on requirement type and rule class is introduced in Table 2 and Table 3. These classifications provided the information which applications needed for verifying the BIM data. The new classification along with the existing functional requirements classification helps system engineers in identifying which requirements are automatable along with their intended use.

What LOD is needed for allocating requirements data to object instances?

LOD formalises the amount and precision of information available in various project phases. LOD as elaborated in Section 3.2 forms an important concept required for automated verification process. Different research studies recommend LOD300 as an ideal level for requirement verification. However, LOD300 is difficult to achieve in the initial design phase. Since, most of the requirements need to be verified in design phase, they cannot be automated without the required information.

The use of OTL in requirements and BIM data tries to solve this issue. Standard OTL for system types contains data that can be used in any phase of the project. This provides minimum required information about objects, spaces and properties from the start of the project. The use of OTL also standardises the geometric and semantic information that modeller needs to provide. Overall, though research identified LOD300 as a required level, experts indicated that it depends on the working style, requirement type and interoperability of data between various applications. In one of the validation interview with a software developer, it was recommended that, an effort of making all the information available through OTL is a right approach to improve information level. Rijkswaterstaat in Netherlands is trying to do this at a national level by creating OTL for all public infrastructure types. The use of OTL though does not solve the issue of information available it provides information to the users on what data is missing and needs to be filled. With the parameter present from the start, the systems engineer can already map the requirements with BIM object. However, the requirements will be shown as not complaint until the required data becomes available.

How can requirements be semantically mapped to object instances?

The answer to this research question forms an important step in answering the main research

[19]https://speckle.systems/

[20]R-Reliability, M-Maintainability, A-Availability, S-Safety

question. In the current process, requirements are linked to objects with pre-existing object breakdown structure within Relatics application. The linking is done for the complete textual requirement. No information about the link between object and its related property is available. There are some applications like Briefbuilder who have provided API services to connect requirement with BIM models. However, similar to Relatics, entire text is mapped to one element in BIM model as a parameter. This renders the process not useful for automated verification.

The solution to this was arrived by reviewing various existing academic and commercial solutions. Different approaches were identified for mapping of requirement data at object level using different technologies. A common pattern was identified among them which is elaborated in subsection 4.2. The requirements are first semantically classified as their objects and properties and allocated to their relevant objects. They are then represented in a constraint manner based on technologies like visual programming, conceptual graph et cetera. A solution for mapping is proposed by classifying and representing data based on combination of approaches proposed by Hjelseth and Nisbet (2011) and Dimyadi et al. (2016). This forms the basis for the proposed framework. By use of OTL for classification and representation, machine readable requirement statements are created.

How can compliance of a particular object with its associated requirements be proven and how can this process be automated?

In the current design process, requirements are provided with a verification plan. Various verification plan types identified from case study are provided in Table 5. The detailed verification plans are documented in reports and spreadsheets. Machine readable requirement data created using proposed framework and written in a constraint way, should be in line with verification plan. Based on the complexity of this verification plan, four rule classes were suggested by Solihin and Eastman (2015). This classification helps system engineers in identifying the effort needed in developing automated compliance for the the given requirement.

For implementation of the framework, two approaches were proposed using commercially available solutions. The first approach is through visual programming. Dynamo for Autodesk Revit was used for writing automation scripts. Although simple, visual programming requires an effort in writing Dynamo scripts for each of the requirement. This is again a time consuming process. A different approach was proposed using NLP for table parsing. The BIM data is exported as a database in tabular format on which semantic represented requirements are run as a query. The model had less precision due to lack of training data. A research on NLP approach with extensive training data may lead to a more efficient solution. This was also recommended by practitioners during validation.

The main research question is answered now, after answering the sub-questions:

How can we improve the mapping between requirements and BIM data for verification of design's compliance in infrastructure projects?

Properly defined requirements and detailed object information are two necessities in mapping requirement and BIM data. Use of OTL forms a bridge in connecting requirements with BIM model data.

Requirements are first analysed and semantically classified using classification approach proposed in subsection 4.2. The classified requirements are allocated to object instances of BIM model. Requirements are now written as constraints using semantic representation approach proposed in section 4 using defined relations, concepts and properties from OTL. The rewritten requirements can now be constructed as scripts using visual programming approach or as a query using table parsing approach, for compliance verification. The direct link between object data and requirement text allows design and verification team members to find out necessary data without any misunderstandings. In addition, the project team can now handle the project with increased level of information when compared to current process with the use of OTL.

7.3 Recommendations

The recommendations provided in this chapter is based on observations and conclusions made during design process analysis, case study verification and validation of the framework. Recommendations provided here focus more on further development of the framework which can potentially contribute towards automation of the process.

First recommendation relates to implementation of the proposed framework. The workflow is quantitatively validated now and needs a validation on the practicality of implementation, technology to be used and obstacles that will be faced during the process. This would also pave for future research on complete automation and gives a clear insight on which disciplines should be involved and interfaces that needs to be developed among them.

Second recommendation is about improvement of requirements management process. Academia and industry should lead a combined effort on identifying the root causes for why SE guidelines proposed by INCOSE (2015) and Leidraad SE (2015) is not followed. A further exploratory study is required on why achieving a common vocabulary with different stakeholders is not possible. This study can provide insights to the problem in SE implementation and its tools being used. The targeted study may also help in getting recommendations about the kind of digital tools needed for further development.

Third recommendation is about standardisation of project vocabulary. Different project stakeholders use varying terms while describing about a element or its properties. This is one of the major problems for poor definition and interpretation of requirements. Royal HaskoningDHV should work on implementation and improvement of object type library. The usage of open standards in developing OTLs makes the library flexible and easy to extend.OTLs developed on open standard also results in a connected database which opens up data exchange.

Last recommendation relates to the future of automated requirements allocation and verification. Automated verification has a significant potential in generative design concept which is still in a nascent stage in the construction industry. The main barrier in this domain is about internalising the information. Practical implementation of writing requirements in a machine understandable way is one step closer to it being an input for generative design algorithms.

References

Albers, A., Scherer, H., Bursac, N., & Rachenkova, G. (2015). Model based systems engineering in construction kit development - Two case studies. In *Procedia cirp* (Vol. 36, pp. 129–134). Elsevier B.V. doi: 10.1016/j.procir.2015.01.044

Alnaggar, A., & Papadonikolaki, E. (2019a). A Proposed Model for Digital Transformation of Requirements Management in the Design of Healthcare Facilities. *Proceedings of the 36th CIB W78 2019 Adavances in ICT in Design, Construction and Management in Architecture, Engineering, Construction and Operations (AECO)*(September).

Alnaggar, A., & Papadonikolaki, E. (2019b, sep). A Proposed Model for Digital Transformation of Requirements Management in the Design of Healthcare Facilities. In *In: Proceedings of 36th cib w78 2019 conference. international council for research and innovation (cib): Newcastle, united kingdom. (2019).* International Council for Research and Innovation (CIB). Retrieved from http://cibw78.northumbria-eee.co.uk/

Anumba, C., Issa, R., Pan, J., & Mutis, I. (2008, 07). Ontology-based information and knowledge management in construction. *Construction Innovation: Information, Process, Management*, *8*, 218-239. doi: 10.1108/14714170810888976

Aslaksen, E. W., DeLamare, M., Fehon, K., Godau, R., Knott, A., Kouassi, A., & de Liefde, J. (2012). *Guide for the Application of Systems Engineering in Large Infrastructure Projects* (Tech. Rep. No. June). San Diego, California: International Council on Systems Engineering.

Astafjevas, G. (2019). *Improving Standard Compliance Through Linked Data: A case of product life-cycle management in maritime sector* (Master Thesis). University of Amsterdam.

Baldauf, J. P., Formoso, C. T., Tzortzopoulos, P., Miron, L. I., & Soliman, J. (2020). Using building information modelling to manage client requirements in social housing projects. *Sustainability (Switzerland)*, *12*(7), 1–21. doi: 10.3390/su12072804

Beach, T. H., Hippolyte, J. L., & Rezgui, Y. (2020, oct). Towards the adoption of automated regulatory compliance checking in the built environment. *Automation in Construction*, *118*, 103285. doi: 10.1016/j.autcon.2020.103285

Bloch, T., & Sacks, R. (2018). Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Automation in Construction*, *91*(March), 256–272. Retrieved from https://doi.org/10.1016/j.autcon.2018.03.018 doi: 10.1016/j.autcon.2018.03.018

buildingSMART". (2019, 06). *Industry Foundation Classes (IFC).* Retrieved from https://technical.buildingsmart.org/standards/ifc/

Bultman, R. (2019). *Specifying abstract requirements of buildings using Knowledge-Based Systems Towards automated requirement verification using Open BIM* (master thesis). Eindhoven university of technology.

Ciribini, A. L. C., Bolpagni, M., & Oliveri, E. (2015, jan). An Innovative Approach to e-public Tendering Based on Model Checking. *Procedia Economics and Finance*, *21*, 32–39. doi: 10.1016/s2212-5671(15)00147-1

Costa, G. (2017). *Integration of building product data with BIM modelling: a semantic-based product catalogue and rule checking system* (PhD Thesis, Universitat Ramon Llull). doi: 10.13140/RG.2.2.11123.63522

Costa, R. (2018, 11). *The Double Diamond model: what is it and should you use it?* Retrieved from https://www.justinmind.com/blog/double-diamond-model-what-is-should-you-use/

Daan, O. (2020). *Specification Libraries.* Retrieved 2021-01-17, from https://semmtech.com/use-case/specification-libraries/

Dawood, H., Siddle, J., & Dawood, N. (2019). Integrating IFC and NLP for automating change request validations. *Journal of Information Technology in Construction (ITcon)*, *24*(Special issue), 540–552. Retrieved from https://www.itcon.org/2019/30 doi: 10.36680/j.itcon.2019.030

de Graaf, R. S. R., Vromen, R. M. R., & Boes, J. H. (2017, apr). Applying systems engineering in the civil engineering industry: an analysis of systems engineering projects of a Dutch water board. *Civil Engineering and Environmental Systems*, *34*(2), 144–161. Retrieved from https://www.tandfonline.com/doi/full/10.1080/10286608.2017.1362399 doi: 10.1080/10286608.2017.1362399

Design Council. (2019, 09). *What is the framework for innovation? Design Council's evolved Double Diamond.* Retrieved from https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond

Dimyadi, J., & Amor, R. (2013). Automated Building Code Compliance Checking. Where is it at? *The c*(380), 172–185. doi: 10.13140/2.1.4920.4161

Dimyadi, J., Solihin, W., & Eastman, C. (2016). A knowledge representation approach in BIM rule requirement analysis using the conceptual graph. *Journal of Information Technology in Construction*, *21*(November), 370–402.

Ding, L., Zhong, B., Wu, S., & Luo, H. (2016). Construction risk knowledge management in bim using ontology and semantic web technology. *Safety Science*, *87*, 202-213. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925753516300492 doi: https://doi.org/10.1016/j.ssci.2016.04.008

Eastman, C., min Lee, J., suk Jeong, Y., & kook Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033. Retrieved from http://dx.doi.org/10.1016/j.autcon.2009.07.002 doi: 10.1016/j.autcon.2009.07.002

El-Diraby, T. E., & Osman, H. (2011). A domain ontology for construction concepts in urban infrastructure products. *Automation in Construction*, *20*(8), 1120–1132. Retrieved from http://dx.doi.org/10.1016/j.autcon.2011.04.014 doi: 10.1016/j.autcon.2011.04.014

Fan, S. L., Chi, H. L., & Pan, P. Q. (2019, sep). Rule checking Interface development between building information model and end user. *Automation in Construction*, *105*, 102842. doi: 10.1016/j.autcon.2019.102842

Francisco, L., Lagare, T., Jain, A., Chaudhary, S., Kulkarni, M., Sardana, D., ... Franzon, P. (2020, nov). Design Rule Checking with a CNN Based Feature Extractor. In *Mlcad '20: Proceedings of the 2020 acm/ieee workshop on machine learning for cad* (pp. 9–14). Association for Computing Machinery (ACM). Retrieved from https://dl-acm-org.tudelft.idm.oclc.org/doi/10.1145/3380446.3430625 doi: 10.1145/3380446.3430625

Ghannad, P., Lee, Y. C., Dimyadi, J., & Solihin, W. (2019). Automated BIM data validation integrating open-standard schema with visual programming language. *Advanced Engineering Informatics*, *40*(September 2018), 14–28. Retrieved from https://doi.org/10.1016/j.aei.2019.01.006 doi: 10.1016/j.aei.2019.01.006

Goh, B. H. (2008). E-government for construction: The case of Singapore's CORENET project. *IFIP Advances in Information and Communication Technology*, *254*, 327–336.

Heinen, J. (2021, 07). *Data exchange: BIM and OTL go hand in hand* (Tech. Rep.). Retrieved from https://blog.bim-connected.com/q1-21-c2/bim/4-essenti%C3%ABle-stappen-voor-het-opzetten-van-een-object-type-library

Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., & Eisenschlos, J. M. (2020). *Tapas: Weakly supervised table parsing via pre-training.*

Hjelseth, E., & Nisbet, N. N. (2011). Capturing normative constraints by use of the semantic mark-up rase methodology..

Houdt, S. V. D., & Vracken, J. L. M. (2013). *Rolling out Systems Engineering in the Dutch Civil Construction Industry* (scientific paper). Delft university of technology.

Howell, S., Beach, T., & Rezgui, Y. (2020, jul). Robust requirements gathering for ontologies in smart water systems. *Requirements Engineering*, *26*(1), 97–114. Retrieved from https://doi.org/10.1007/s00766-020-00335-z doi: 10.1007/s00766-020-00335-z

Huitzil, I., Molina-Solana, M., Gómez-Romero, J., & Bobillo, F. (2021). Minimalistic fuzzy ontology reasoning: An application to building information modeling. *Applied Soft Computing*, *103*, 107158. Retrieved from https://www.sciencedirect.com/science/article/pii/S1568494621000818 doi: https://doi.org/10.1016/j.asoc.2021.107158

INCOSE. (2015). *Systems engineering handbook: A guide for system life cycle processes and activities* (4th ed.; International Council on systems engineering, Ed.). San Digeo, CA: John Wiley and Sons Inc.

Ismail, A., Strug, B., & Ślusarczyk, G. (2018, 01). Building knowledge extraction from bim/ifc data for analysis in graph databases. In (p. 652-664). doi: 10.1007/978-3-319-91262-2_57

ISO15288. (2015, 05). *Systems and software engineering — System life cycle processes* (Tech. Rep.). Retrieved from https://www.iso.org/standard/63711.html

ISO29481. (2016). *Building information models — Information delivery manual — Part 1: Methodology and format* (Tech. Rep.). International standards organisation.

Jiang, L. (2015). *A Constructability Review Ontology To Support Automated Rule Checking Leveraging Building Information Models* (PhD Dissertation, The Pennsylvania State University). Retrieved from https://www.researchgate.net/publication/303856442AConstructabilityReviewOntologyToSupportAutomatedRuleCheckingLeveragingBuildi

Jiang, S., Wang, N., & Wu, J. (2018). Combining BIM and Ontology to Facilitate Intelligent Green Building Evaluation. *Journal of Computing in Civil Engineering*, *32*(5), 04018039. doi: 10.1061/(asce)cp.1943-5487.0000786

Kim, H., Lee, J. K., Shin, J., & Choi, J. (2019). Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking. *Journal of Computational Design and Engineering*, *6*(2), 143–148. Retrieved from https://doi.org/10.1016/j.jcde.2018.08.002 doi: 10.1016/j.jcde.2018.08.002

Kiviniemi, A. (2005). Requirements management interface to building product models. *VTT Publications*(572), 1–12.

Lee, P. C., Lo, T. P., Tian, M. Y., & Long, D. (2019). An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning. *KSCE Journal of Civil Engineering*, *23*(5), 1952–1962. doi: 10.1007/s12205-019-1750-2

Lee, Y. C., Ghannad, P., Dimyadi, J., Lee, J. K., Solihin, W., & Zhang, J. (2020). A Comparative Analysis of Five Rule-Based Model Checking Platforms. In *Construction research congress 2020: Computer applications - selected papers from the construction research congress 2020* (pp. 1127–1136). American Society of Civil Engineers (ASCE). doi: 10.1061/9780784482865.119

Leidraad SE. (2015, 06). *Leidraad Systems Engineering* (Tech. Rep.). Retrieved from https://www.leidraadse.nl/

Lenferink, S., Tillema, T., & Arts, J. (2013, may). Towards sustainable infrastructure development through integrated contracts: Experiences with inclusiveness in Dutch infrastructure projects. *International Journal of Project Management*, *31*(4), 615–627. doi: 10.1016/j.ijproman.2012.09.014

Luiten, B., Bohms, M., Alsem, D., & O'keeffe, A. (2018). Asset information management for European roads using linked data. In B. Luiten (Ed.), *Proceedings of 7th transport research arena tra 2018* (pp. 16–29). Vienna, Austria.

Ma, L., Sacks, R., Kattel, U., & Bloch, T. (2018). 3d object classification using geometric features and pairwise relationships. *Comput. Aided Civ. Infrastructure Eng.*, *33*, 152-164.

Ma, Z., & Liu, Z. (2018, jun). Ontology- and freeware-based platform for rapid development of BIM applications with reasoning support. *Automation in Construction*, *90*(August), 1–8. Retrieved from `https://doi.org/10.1016/j.autcon.2018.02.004` doi: 10.1016/j.autcon.2018.02.004

Malsane, S., Matthews, J., Lockley, S., Love, P. E., & Greenwood, D. (2015, jan). Development of an object model for automated compliance checking. *Automation in Construction*, *49*(1), 51–58. doi: 10.1016/j.autcon.2014.10.004

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251-266. Retrieved from `https://www.sciencedirect.com/science/article/pii/0167923694000412` doi: https://doi.org/10.1016/0167-9236(94)00041-2

Marchant, D. (2016). The design brief: requirements and compliance. *Journal of Information Technology in Construction (ITcon)*, *21*(Special issue), 337–353. Retrieved from `http://www.itcon.org/2016/22`

Martins, J. P., Carvalho, B., & Almeida, V. A. (2016). Automated rule-checking – a tool for design development. In *International association for housing science - 41st world congress on housing - sustainability and innovation for the future* (pp. 1–8). Retrieved from `https://core.ac.uk/download/pdf/143403202.pdf.`

Moonen, L. T. (2017). *The implications of automated verification of client specific requirements using semantic web standards and rule checking techniques* (Master Thesis). Eindhoven univeristy of technology.

Nagendrakumar, P. N. (2019). *Towards automated bim using natural language processing: An interdisciplinary perspective* (Unpublished doctoral dissertation). Delft University of Technology, Delft, Netherlands.

Nawari, N. O. (2019). Automating Code Compliance Checking. *MDPI - Buildings*, *9*(4), 86. Retrieved from `https://www.mdpi.com/2075-5309/9/4/86` doi: doi.org/10.3390/buildings9040086

Panah, R., & Kioumarsi, M. (2021, 01). Application of building information modelling (bim) in the health monitoring and maintenance process: A systematic review. *Sensors*, *21*, 837. doi: 10.3390/s21030837

Park, S., Lee, Y.-C., & Lee, J.-K. (2016). DEFINITION OF A DOMAIN-SPECIFIC LANGUAGE FOR KOREAN BUILDING ACT SENTENCES AS AN EXPLICIT COMPUTABLE FORM. *www.itcon.org-Journal of Information Technology in Construction*, *21*(Special issue), 422–433. Retrieved from `www.w3.org:`

Pauwels, P., De Meyer, R., & Van Campenhout, J. (2010). Visualisation of semantic architectural information within a game engine environment. In *Proceedings of the 10th international conference on construction applications of virtual reality* (pp. 219–228). Retrieved from `http://biblio.ugent.be/record/942253`

Pauwels, P., de Farias, T. M., Zhang, C., Roxin, A., Beetz, J., De Roo, J., & Nicolle, C. (2017, aug). A performance benchmark over semantic rule checking approaches in construction industry. *Advanced Engineering Informatics*, *33*, 68–88. doi: 10.1016/j.aei.2017.05.001

Pauwels, P., & Terkaj, W. (2016). Express to owl for construction industry: Towards a recommendable and usable ifcowl ontology. *Automation in Construction*, *63*, 100-133. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580515002435 doi: https://doi.org/10.1016/j.autcon.2015.12.003

Pauwels, P., Zhang, S., & Lee, Y. C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, *73*, 145–165. Retrieved from http://dx.doi.org/10.1016/j.autcon.2016.10.003 doi: 10.1016/j.autcon.2016.10.003

Robertson, S., & Robertson, J. (2017). *Mastering the Requirements Process: Getting Things Right* (3rd ed.; P. Network, Ed.). Drexel hill: Pearson Education Limited. Retrieved from https://www-proquest-com.tudelft.idm.oclc.org/docview/1968021079/fulltext

Roy, R., Low, M., & Waller, J. (2005). Documentation, standardization and improvement of the construction process in house building. *Construction Management and Economics*, *23*(1), 57-67. Retrieved from https://doi.org/10.1080/0144619042000287787 doi: 10.1080/0144619042000287787

Sacks, R., Eastman, C., Lee, G., & Teicholz, T. (2018). *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers* (3rd ed.). Wiley & Sons Inc.

Safari, K., & AzariJafari, H. (2021). Challenges and opportunities for integrating bim and lca: Methodological choices and framework development. *Sustainable Cities and Society*, *67*, 102728. Retrieved from https://www.sciencedirect.com/science/article/pii/S2210670721000238 doi: https://doi.org/10.1016/j.scs.2021.102728

Santos, R., Costa, A. A., & Grilo, A. (2017). Bibliometric analysis and review of building information modelling literature published between 2005 and 2015. *Automation in Construction*, *80*, 118-136. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580517302297 doi: https://doi.org/10.1016/j.autcon.2017.03.005

Senthilvel, M., & Beetz, J. (2020). A Visual Programming Approach for Validating Linked Building Data. In *Eg-ice 2020 workshop on intelligent computing in engineering* (pp. 403–411). Berlin, Germany: RWTH Aachen University. Retrieved from https://www.researchgate.net/publication/345458791AVisualProgrammingApproachforValidatingLinkedBuildingData

Solihin, W., Dimyadi, J., & Lee, Y. (2019). In search of open and practical language driven BIM based automated rule checking systems. *Advances in Informatics and Computing in Civil and Construction Engineering*(January), 577–585. doi: 10.1007/978-3-030-00220-6

Solihin, W., Dimyadi, J., Lee, Y.-C., Eastman, C., & Amor, R. (2017). The Critical Role of Accessible Data for BIM-Based Automated Rule Checking Systems. In *Lc3 2017: Volume 1 - proceedings of the joint conference on computing in construction (jc3)* (pp. 53–60). Heraklion, Greece. doi: 10.24928/jc3-2017/0161

Solihin, W., Dimyadi, J., Lee, Y.-C., Eastman, C., & Amor, R. (2020). Simplified schema queries for supporting BIM based rule checking applications. *Automation in Construction*, *117*, 103248. doi: htrps://doi.org/10.1016/j.autocon.2020.103248

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*(May 2016), 69–82. doi: 10.1016/j.autcon.2015.03.003

Soliman-Junior, J., Formoso, C. T., & Tzortzopoulos, P. (2020). A semantic-based framework for automated rule checking in healthcare construction projects. *Canadian Journal of Civil*

Engineering, *47*(2), 202–214. doi: 10.1139/cjce-2018-0460

Soman, R. K., Molina-Solana, M., & Whyte, J. K. (2020). Linked-Data based Constraint-Checking (LDCC) to support look-ahead planning in construction. *Automation in Construction*, *120*(July), 103369. Retrieved from https://doi.org/10.1016/j.autcon.2020.103369 doi: 10.1016/j.autcon.2020.103369

Song, J., Kim, J., & Lee, J.-K. (2018). NLP and Deep Learning-based Analysis of Building Regulations to support Automated Rule Checking System. In *35th international symposium on automation and robotics in construction.*

Stancheva, M. (2017). *Improving the management of structural engineering requirements in the design phase.* (Master Thesis). Eindhoven Univerisity of technology.

Sydora, C., & Stroulia, E. (2020). Rule-based compliance checking and generative design for building interiors using BIM. *Automation in Construction*, *120*(July), 103368. Retrieved from https://doi.org/10.1016/j.autcon.2020.103368 doi: 10.1016/j.autcon.2020.103368

Team-unifi. (2019). *BIM Software: Which is the Most Popular?* Retrieved from https://unifilabs.com/BIM-software

Van Swaay, T. G. H. (2011). *Requirements and configuration management development from the specifications for civil constructions* (Master thesis). University of Twente.

Warren, D. (2019). Parametric Modelling in Construction: Investigating the Quality of Rule-Based Checking. *Building Information Modelling (BIM) in Design, Construction and Operations III*, *1*(December), 57–68. doi: 10.2495/bim190061

Wheeler, W. (2003, apr). System Requirements and project success. In W. Wheeler (Ed.), *Integrating wireless technology in the enterprise* (chap. 6). Boston: Digital Press. Retrieved from https://www.elsevier.com/books/integrating-wireless-technology-in-the-enterprise/wheeler/978-1-55558-295-1

Wu, J., & Zhang, J. (2019). New automated bim object classification method to support bim interoperability. *Journal of Computing in Civil Engineering*, *33*(5), 04019033. Retrieved from https://ascelibrary.org/doi/abs/10.1061/28ASCE29CP.1943-5487.0000858 doi: 10.1061/(ASCE)CP.1943-5487.0000858

Zhang, D., Zhang, J., Guo, J., & Xiong, H. (2019). A semantic and social approach for real-time green building rating in BIM-based design. *Sustainability (Switzerland)*, *11*(14), 1–16. doi: 10.3390/su11143973

Zhang, J., & El-Diraby, T. E. (2012). Social semantic approach to support communication in aec. *Journal of Computing in Civil Engineering*, *26*(1), 90-104. Retrieved from https://ascelibrary.org/doi/abs/10.1061/28ASCE29CP.1943-5487.0000130 doi: 10.1061/(ASCE)CP.1943-5487.0000130

Zhang, J., & El-Gohary, N. M. (2016, mar). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering*, *30*(2), 04015014. doi: 10.1061/(asce)cp.1943-5487.0000346

A Model checking concepts and categories

The various model checking concepts as described by Hjelseth and Nisbet (2011) is identified in Table 8.

Table 8: Types of model checks

Model Check	Description
Geometry based checking	Check based upon topological relationships and Boolean algebra to define if geometry of design is complied with given rules. Clash detection is an example of model checking.
Compliancy checking	Check if the design is according to the building codes and requirements which applies to the project.

Solihin and Eastman (2015) has defined that the technique of model checking be classified into seven categories. They are:

- **Syntactic model checking:** Checks if the model is according to the modelling agreements, employer information requirements and standards.
- **Regulatory code checking:** Checks if the model is complied with all the applicable building codes.
- **Client requirements checking:** Checks if the model is complied with specified client requirements.
- **Constructability validation:** Checks if the model can be as built in construction site without constraints.
- **Construction safety validation:** Validates via simulations and checks for any potential danger during the construction execution.
- **Warranty approval checking:** Checks the post construction model for issues that affect the product warranties and cost to maintain.
- **BIM data completeness checking:** Checks the data completeness of BIM models according to initial agreed BIM execution plans before the handover of structure and as built information to asset owners.

B Linked data and IFC technologies

IFC standards

BIM is becoming a new approach for displaying and describing information in construction sector. BIM is able to combine different information about design, construction, operation processes into single environment thus reducing the documentation in PDF[21] and document form. However, in order to get benefits of this information consolidation , the quality and communication of this data exchange needs to be assured. BuildingSMART consortium working on improving data exchange formats in AEC industry has developed several standards for this use. Some of the data exchange standards proposed by BuildingSMART are described in

Table 9: openBIM standards (BuildingSMART. 2015)

Name	Functionality
IDM (Information delivery manual)	Process description
IFC (Industry foundation classes)	Information exchange
BCF (BIM collaboration format)	Change management
IFD (International framework for dictionaries)	Data mapping
MVD (Model view definition)	Process description

In the current research, the BIM data from Revit application is dealt with directly due to the limitations of IFC infrastructure type. However, further developments in IFC file types provides the advantage of using open-standards in the workflow. Next section focuses on understanding about IFC file type and its structure. IFC is an platform neutral format and it contains model of a building, including spatial elements, materials and shapes. Unlike, OTLs the file includes both geometry and data. IFC schema is constantly evolving. The latest schema is IFC4.3. The schema contains concepts such as classes, attributes, relationships, property sets, quantity definitions. IFC can be generally described as an object-oriented data model composed of class definitions describing elements and processes (buildingSMART", 2019). Currently , IFC is used to exchange information from one party to another for a specific business transaction. IFC can also be used as a method for archiving project information, whether incrementally during the design, procurement, and construction phases, or as an "as-built" collection of information for long-term preservation and operations purposes. The IFC data can be encoded in various formats, such as XML, JSON, and STEP and transmitted over web services, imported/exported in files, or managed in centralised or linked databases. BIM authoring applications will provide interfaces to end users to export, import, and transmit data in some IFC format. It is up to users to decide what they want to share from their tools via IFC.

IFC follows an object based inheritance hierarchy in its file structure. There are two main entity types they are rooted and non-rooted entities. Rooted entities are named with globally unique ID, name and description. Rooted entities derived from IFCroot can be created independently while non rooted entities could be created only if a reference exists from rooted instance directly or indirectly. There are there abstract subcategories to IFCroot. They are IfcObjectDefinition which describes object occurrences and types. IfcRelationship that captures the relationship between

[21] https://en.wikipedia.org/wiki/PDF

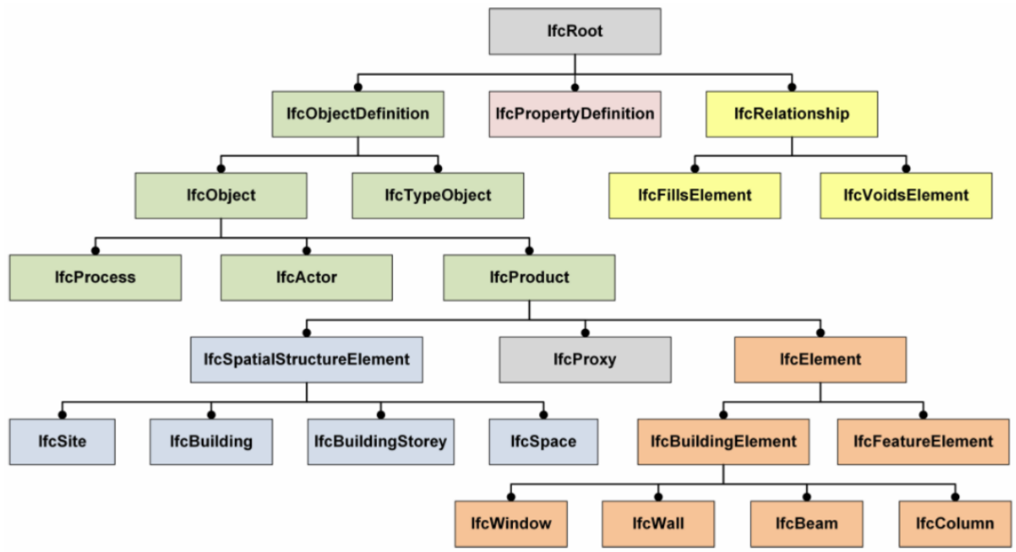


Figure 27: IFC data architecture (Borrmann et al. 2015)

them and IfcPropertyDefinition represents the properties and attributes of objects. Figure 27 gives an overview of the IFC architecture. The current research proposes a method for how this BIM data could be standardised in terms of names and properties by assigning the object type properties from OTLs. The next section elaborates about linked data and OTL technologies and what are strategies that could be adopted when creating a OTL.

Linked Data and OTL

Semantic web technologies and ontologies are among more recent research topics that are gaining momentum in the field of BIM(Santos, Costa, & Grilo, 2017). With these technologies it is possible to add more machine understandable meaning to information. This section provides the necessary foundation for understanding the linked data concepts required in creating OTLs.

The idea of semantic web was first proposed by Tim Berners Lee in the year 2001 (Pauwels, Zhang, & Lee, 2017). The vision was to upgrade from web of document to web of data. It is converting human readable documents into machine understandable data. The traditional web contains information by means of natural language, pictures, videos where the meaning of information is implicit and hidden in the context which is understood only by humans. For this to convert to a machine understandable data, the information needs to be additionally described explicitly without leaving it for multiple interpretation. For example, the word python can also mean an animal or a programming language. Humans can interpret the meaning and information trying to be conveyed depending on the context however for this to be made machine understandable, the word python needs to be defined with its properties and attributes. In the context of current research, enriching the objects with semantic descriptions suggests that data model describes the meaning of its instances (Stancheva, 2017). According to Sacks et al. (2018), domain knowledge can be made explicit by formal and standardised representation using ontologies. In this way, machines can automatically understand the meaning of data and facilitates the automated deduction of implicit or hidden data from existing data. This is particularly required in the requirements verification process since one of the requirement rule type involves creation of new data from existing data using logical interpretations. Linked data is the core technology for the creation of semantic web (Sacks et al., 2018). Linked data is method for publishing data in a structured way, in which it can be intertwined by the use of semantics in a meaningful and structured manner, where different data instance have one to one and one to many relationship with one another. The main goal of the Linked data is to connect different concepts and individual elements and maximise the value and usefulness by interconnecting different data sets (Panah & Kioumarsi, 2021).

In the current way of working, OTLs are used by asset owners to describe information for the maintenance and management process. In the proposed workflow in Figure 14, asset owner provides the OTL and delivered data (Native and IFC models, 2D drawings, documents) must contain same data(Heinen, 2021). Setting up and OTL has many advantages:

- OTL helps in speaking one language across the organisation and sector when done at national level
- It describes what information should be recorded for a particular object and how it should be done.
- With OTL development, object details has to be captured only once and reused multiple times
- OTL makes information computer interpretable which can further help in processes like model checks.

Heinen (2021) describes four steps in setting up of OTL and provides two ways or variants based on which data could be interpreted and stored. The four steps defined are:

- **Formulate the purpose of OTL:** Developing OTL should start with formulating the goal. It should be ensured that OTL will be the solution to the problem it is being designed for. The goal determines the technical architecture, functions and features of the OTL. During setting up, there should be continuous co-ordination with decision makers and users.
- **Use of open standards:** The use of open standards makes the exchange of data easier and faster. Open standards ensures correct information exchange can happen between organisations and other partners committed to open standards.
- **Provide good architecture:** The data architecture forms the backbone of OTL and ensures object data can be expressed properly and queried. Without correct architecture, OTL becomes unstructured data, which is difficult to interpret and use for communication with parties.
- **Describe information need:** Defining information need for each object is essence of OTL. What is correct definition of, for example, a viaduct and which properties are important to know and must therefore be filled in. This steps makes clear and unambiguous for everyone what the need is. With this it is possible for design team to ask the correct information. Company wide agreements must be made about information needs and how it should be documented.

The above four steps ensures the development of an efficient organisation specific OTL. (Heinen, 2021) proposes two way of recording data to deliver BIM models. In the first variant, the data and geometry is separated and in second variant data and geometry is combined. In variant 1, the BIM model is degraded to 3D model and geometry data is recorded. Remaining data (all alphanumeric values) that says something about an object is recorded seperately. The recording of these two data separately should be done by means of specially designed database. The data will have to meet the requirements as laid down in the OTL of the client. In variant 2, object data and geometry data is combined. The openBIM standard IFC is suitable for combining geometry with data. The IFC schema supports the recording of metadata around documents and the relationship to objects. This makes it possible to supply all data, and the relationships to documents and external sources, in one file format.

C Interview setup and questions

Topics discussed in the interviews were:

- Systems engineering adoption in AEC industry
- Model based systems engineering
- Use of systems engineering tools (E.g., Relatics, Briefbuilder etc.)
- Requirement's interpretation and verification process
- Information exchange and requirements of BIM model
- Information databases
- Automated rule checking

The same topics were discussed with different experts to get different perspectives of the problem existing in the current process. A total of 7 interviews was conducted over a period of 3 weeks. The details pf various domain experts interviewed are represented in the Table 10.

Table 10: Problem verification interviewees list

Index	Name	Role	Company
1	Niels Willemsen	Project manager	Royal HaskoningDHV
2	Gijben Hornes	Software developer	ICOP
3	Daan Alsem	Systems engineer	Royal HaskoningDHV
4	Daan Oostinga	Software developer	Semmtech B V
5	Ben Visser	Systems engineer	Oxand
6	Hein Wilman	BIM Manager	Royal HaskoningDHV
7	Tsoefiet van Beuningen	Data modeller	Royal HaskoningDHV

The following questions were formulated after the review of existing literature and design process. The questions were discussed further using sub-questions. The knowledge expectation from these questions is described below.

Q1. What are the conventional systems engineering process followed in infrastructure industry?

- What is the standard process in your organisation/projects for creating system breakdown structure?
- What is the good method to create system breakdown structure in relation to link between various objects and functions?

Expectations: The expected information from these questions is to evaluate how BIM and systems engineering is integrated in the design process. This questions also help in reflecting on the problem in the current system design process and identify the standard way to create system and object breakdown structure.

Q2. What is the information exchange plan (BIM plan) followed in infrastructure industry?

- How are requirements documentation handled currently and how much of the process is still in paper or document-based format?
- How do you ensure that the requirements remain up to date?

Expectations: These question aims to capture the ambiguities in information transfer using BIM models. The current practices of requirements transfer from client to contractors and the process of change management is also discussed here.

Q3. What is the standard process in your organisation/projects for designing the structure and verification?

- Do you face any problems with current verification process? What are they?
- At which stage of design and construction do you experience problems with respect to requirements management and verification.
- What are the most important parts in design verification process?

Expectations: These questions mainly focus on the design process and tries to identify the consequences of ambiguities raised during the design process. The influence of requirements management process on the entire building life-cycle is evaluated here.

Q4. Which type of requirements are difficult or cumbersome to verify?

- Is there a pattern or typology that you have found in requirement types and their verification?
- Do you use any software tools for requirements management and has the use of it improved the verification process?
- Literature suggests that lack of common interpreting language between various stakeholders is one of the significant issues in the requirements management system. Have you experienced this problem? An example
- How do you cope with this issue? How has the use of software tools helped in this process?
- Automating which type of requirements would improve the verification process.

Expectations: The questions here focus on how requirements are documented, typology of requirements and software applications used by construction companies in Netherlands for the management of requirements. These questions also highlight on the important issue of unambiguous interpretation of requirement for carrying out design. Effort is also made to capture the benefits of the existing software applications and their unique features. Identifying automatable requirements is also forms the scope of this set of questions.

Q5. Do you use any model checkers currently in the verification process? If so, what are they and for what type of requirements it is used.

- Would a common dictionary or knowledge graph of requirements can help you?
- How do you imagine that a translation of text requirements into rules should be done?
- Do you use any model checkers currently in the verification process? If so, what are they and for what type of requirements it is used.
- What features would you expect from an automated requirements verification tool?
- Do you think that linking requirements and their documentation with BIM Models could have potential benefits? If so, what are the benefits that you expect?
- In your opinion, do you think that complete automation of requirements verification is possible? If not, what are the bottlenecks that you see

Expectations: Formulating the pre-requisites for improving requirements management and verification is the goal of these questions. The discussion is initially focused on type of digital model checkers being used in the existing workflow and their benefits. The discussion is then lead towards benefits of standardised knowledge graphs and expert systems. The discussion is concluded by participants opinion on moving towards automated verification process and the problems that is expected by them to adopt the workflow.

Framework validation interviewees list

Table 11: Framework validation interviewees list

Index	Name	Role	Company
1	Niels Willemsen	Project manager	Royal HaskoningDHV
2	Phuong Nguyen	BIM manager	Royal HaskoningDHV
3	Daan Alsem	Systems engineer	Royal HaskoningDHV
4	Daan Oostinga	Software developer	Semmtech B V

D Detailed review of academic studies

Table 12: Review of academic solutions

Citation	Checking Domain	Methodology	Input	Output	Features	Verification Phase	Limitations
(Y. C. Lee et al., 2020)	NA	This paper reviews various rule checking platforms and studies the highlights the major similarties and differences among them. Rule-based checking approaches such as IfcDoc, KBim, ePlanCheck, ACABIM, and SNACC is reviewed in this article.	NA	NA	Approach for translating requirements into machine understandable rules	Design phase	Focus more on building regulations and not on client requirements

(Soliman-Junior et al., 2020)	healthcare projects	This article classifies requirments into qualitative, quantitative & ambiguous and are then converted to logical rules using previously created semantic information.	NA	NA	Semantic enrichment of regulations data	Design phase	High level of detail needed for healthcare regulations. Continously changing situations and updated requirements requires simulation efforts

(Sydora & Stroulia, 2020)	building interiors, space modelling	The requirements are converted to rules using FOR and IF-THEN loops. Standard syntaxes are created for writing rules.	Requirements interpreted using rule editor. IFC models	- Unity interface - PDF report	Use of logical operators to model complex rules. Simulation capability within the platform. Generative design for decision making	Design phase	Constraints are predefined. Hardcoding new requirements requires development effort

(Solihin, Dimyadi, Lee, Eastman, & Amor, 2020)	data accessibility	This article proposes a unique approach for transforming BIM data into a simplified schema for making BIM data easily accessible.	IFC data is uploaded to custom developed BIM RL server.	Processed BIM database can be downloaded from BIMRL server in the required format.	Data available for analysis in the efficient and queryable manner	Design phase, Handover phase, Construction phase	Framework needs to be implemented for large scale projects for identifying obstacles

(Francisco et al., 2020)	precase slabs	A convolutional neural network algorith is developed to check if the design satisifies a set of layout rules and achieve high yield when manifactured.	2D CAD images of set sizes according to pixels are provided to the CNN model for rule checking.	A compliance report is generated after rule verification.	Requirements to design conversion is not clear	Design phase	2D images are limited in use in practice

(Baldauf, Formoso, Tzortzopoulos, Miron, & Soliman, 2020)	residential construction, social housing	A requirement classification approach is presented and the requirements are stored according to that. Requirements are then stored in BIM object data as parameters.	Requirements are stored in dRofus application which is then connected to BIM models and requirements are stored as parameters. The model vertification is done through hard coding the rules and verifiying through solibri model checking.	Visual report is made availale through solibri model checking environment.	Use of existing environments for compliance verification	Design phase, Handover phase, Construction phase	Hardcoding new requirements requires development effort

(Senthilvel & Beetz, 2020)	BIM data creation	This paper proposes a method to create an inteface for creating constraints in SHACL using open source API.	NA	NA	NA	Design phase, Handover phase, Construction phase	SHACL constraint creation needs to be checked for infrastructure domain
(Solihin, Dimyadi, & Lee, 2019)	rule check languages	Review of various rule languages available	NA	NA	NA	NA	NA
(Alnaggar & Papadonikolaki, 2019a)	healthcare projects	Provides an idea on which type of rules to be checked in at what stages of construction and what level of detail of BIM model is required for the checking.	NA	NA	Identifies various domain specific requirements and its complexity in healthcare projects	Design phase, Handover phase, Construction phase	The implementation needs to be practically implemented for validation of the framework

(Fan, Chi, & Pan, 2019)	Interface development	A two layered logic based framework is proposed for automatic checking of requirements. This involves first layer of designing the rules and establishing its association with BIM elements. Performing checking at the second layer.	The requirements are modelled using specifically built visual programming interface. The rules modelled are linked to revit model and is executed in the developed interface environment.	Compliance of rules can be obtained as a report in the interface environment.	Visual programming allows non programming experts for rule creation	Design phase, Handover phase	More operators needs to be introduced for modelling of complex rules

(D. Zhang, Zhang, Guo, & Xiong, 2019)	green building, LEED requirements	An ontology is created that converts green building knowledge into machine understandable requirements. The rules are then created using SWRL engine. The rules are linked to Revit model using Autodesk Forge as a interface.	Requirements and rules are available in linked data format. Revit file format is used as a BIM model.	Compliance of rules can be visualized in Autodesk forge interface.	Integration of collaboration tool in the checking interface	Handover phase, Maintenance phase	Approach is suitable for fixed regulations and not for dynamic client requirements

(Dawood, Siddle, & Dawood, 2019)	change management requests	A web bases system called ’Architrack’ is developed which manages and tracks changes to the BIM model and maintains the version of the document.	The change input is predominantly in the form of text via email, documents or direct entry of text. BIM models are made available in IFC format.	The model changes can be visualized within the web based system developed.	Useful for version management of design and requirements	Design phase, Handover phase, Construction phase	NA
(Kim, Lee, Shin, & Choi, 2019)	buidling codes and regulations	The building regulations are hard coded as rules using KBIM code visual language. These rules coded in visual language are eexcuted on BIM model for permit application processing.	Requirements are one time hard coded into rules. BIM models are made available in IFC format.	NA	NA	Design phase	Hardcoding new requirements requires development effort

(Warren, 2019)	commercial checking applications	Review of commercial checking implementations	NA	NA	Identified obstacles for complete implementation of the automation process	NA	NA
(P. C. Lee, Lo, Tian, & Long, 2019)	virtual design assistant	An ontology of the building design is built and ifcOWL is used to describe the semantic information of the model. Rules are created using SWRL engine and is executed on the model. Several solutions generated are ranked using analytic hierarchy process to provide design recommendations.	Requirements are modelled in Protégé environment. IFC format is used for BIM model.	The compliance report is generated after the execution of rules.	Several design options are generated and evaluated using AHP process	Design phase	Use of framework in non standard structures needs to be evaluated

(Ghannad et al., 2019)	visual programme language	The solution proposed focused on creating schema for requirement to rule translation and execution. The framework includes 3 phases (1) Unstructured rule definition as natural language (2) formalized rule definition (3) VPL based rule representation	The requirements are provided in natural language is formalized using XML based application RuleML and LegalRuleML. The formalized rules are represented in visual way using marioentette vectorworks application.	Parameterized rules for execution on BIM models	Visual programming allows non programming experts for rule creation	NA	Complex requirements needs more efficient operators for visual modelling

(Astafjevas, 2019)	maritime rules and regulations	Various standards rules and specifications in maritime sector are stored in the form of RDF triples and is stored in linked data platform/interface created. The BIM data is also converted to RDF triples and is stored in common linked data platform. Rule compliance is executed in linked data platform.	Requirements are input in the natural text and is broken down to RDF triples. BIM data is input in the IFC format.	Report	Specification library for storing and reusing requirements	Design phase, Handover phase, Construction phase	NA

(S. Jiang, Wang, & Wu, 2018)	green building, LEED requirements	The information required for greeen building rules is created as a ontology to arrive at the rules. The rules extracted from ontology is then executed on BIM data to arrive at the compliance report.	The ontology is in .ttl format. SWRL engine is used to create rules. The BIM data is converted to OWL ontology. The BIM data is extracted from JESS rule engine and is executed with green building rules.	The output from each rule is given in the form of score from which a total score is calculated to arrive at final green building rating.	Efficient solution for permit applications and rating systems	Handover phase, Maintenance phase	Approach is suitable for fixed regulations and not for dynamic client requirements

(Bloch & Sacks, 2018)	semantic enrichment	Semantic enrichment using AI and ML	Requirements in text format	Semantically enriched BIM data model	Rule inferencing approach using AI and ML methods	Design phase	The requirement in Dutch language needs to be checked for efficiency interpretation
(Z. Ma & Liu, 2018)	architectural design data	The framework for transforming BIM data to ontology data using xBIM toolkit is proposed.	The rules are modelled using previously created domain ontology. The BIM data required for certain rule is queried using MVD from whole BIM data.	Rule compliance can be visualised in the interface developed.	Framework for domain ontology creation	Design phase, construction phase	Domain ontology needs to be extensive for better interpretation

(Song, Kim, & Lee, 2018)	NLP & ML based approach	The framework for understanding semantic meaning of building regulations to automate requirement interpretation process is described using word embedding technique.	Requirements in natural language was input into developed program interface.	The relevant BIM objects for the particular requirement	Automated object identification and allocation	Design phase	The requirement in Dutch language needs to be checked for efficiency interpretation
(Pauwels, Zhang, & Lee, 2017)	performance benchmark review	NA	NA	NA	Provides critical points for analysing semantic rule checking framework	NA	NA

(Dimyadi et al., 2016)	building codes and regulations	The requirements are interprted into four class of rules and is represented in machine understandable manner using conceptual graph.	Conceptual graphs can be created using any graph database or linked database platforms such GraphDB, Protégé etc.	NA	Rule calssification based on the calculation complexity	Design phase, Handover phase, Construction phase	Rule design becomes more complex with higher rule types

(L. Jiang, 2015)	constructability review	The rules are modelled using IF-THEN framework. A standard vocabulary is achieved by creating domain ontology. The 3D model and rule sets adhered to vocabulary of ontology. The rules are created based on previously defined type of constraints. The rules are then coded into solibri model checking platform.	Rules are coded into Solibri platform, 3D model is provided in IFC format	Rule execution can be visualised on Solibri platform. Report are available in PDF and HTML formats.	Rules are coded into existing platform	Design phase	Complex requirements are difficukt to model using If-Then framework

(El-Diraby & Osman, 2011)	domain ontology	Several modalities is identified to represent the exising ontology in a better hierarchial framework.	The framework proposed can be applied in creating using any database creation platforms.	BIM data	BIM data for efficient querying	Handover phase, Maintenance phase	NA
(Pauwels, De Meyer, & Van Campenhout, 2010)	VR applications	3D building model is visualised in a game engine and is connected to semantic web servers for 3D visualisation of semantic information. Semantic query interface is built where queries and rules can be constructed and run on the game engine environment. This results in a specifically requested pass/fail results.	Direct input of 3D model from Revit through API connection, Semantic data through SPARQL endpoint sent through HTTP server	Visualisation of results on game engine environment	Unreal engine supports integration of various platforms through their API service	Handover phase, Maintenance phase	Framework adaptation for non heritage building needs to be developed

(G. Costa, 2017)	Product data	Provides methodology for integrating external data with BIM solutions	Data input through semantic web technologies	IFC; COBIe	Integrate product catalogue BIM data	Design phase, Handover phase, Construction phase	Checked only on structural safety regulations
(Solihin, Dimyadi, Lee, Eastman, & Amor, 2017)	Data accessibility	Provides a concept for accessible BIM database that supports efficient queries for requirement checking	SQL and No-SQL databases	IFC	Efficent queries for complex rules	Design phase, Handover phase, Construction phase	Focus more on building regulations
(Stancheva, 2017)	Structural engineering domains	The study provides an tool semantic web based platform for linking requirements with BIM data	Textual requirements. BIM data in IFC format	Visualisation of requirements on developed tool	Easy linking of requirements with BIM model	Design phase, Handover phase, Construction phase	Focus on quantitative regulations . Process needs to be simplified more for more complex requirements

(Moonen, 2017)	Space and room requirements	The study proposes an semantic web platform for writing requirements as constraints for checking on BIM model	Requirements in linked data format	Boolean yes or no for compliance of requirements	Requirements are defined in the form of custom rule templates	Design phase	The framework needs to be validated for other domains of requirements. Classification proposed is not applicable in infrastructure domain
(Bultman, 2019)	Maritime rules and regulations	Methodology for converting abstract top level requirements into detailed specifications	Requirements in text format	Specifications in text	Focuses on maritime dictionary	Design phase	The data dictionary needs to be at large level for applicable on wide variety of requirements

(Park, Lee, & Lee, 2016)	Korean building regulations	Language driven approach for representing building regulations as rules	Building codes from published regulations	Visualisation in the developed tool	Strategy for lexical and syntactic analysis of regulations	Design phase	Syntax for design guidelines, design proposals and client requirements needs to be developed
(Marchant, 2016)	Architectural design brief	The study investigates various method of converting design brief data to rules	Design brief in natural language texts	Individual reports based on method used	Proposes extensions to the current IFC data structure	Design phase	NA

E Detailed review of commercial applications

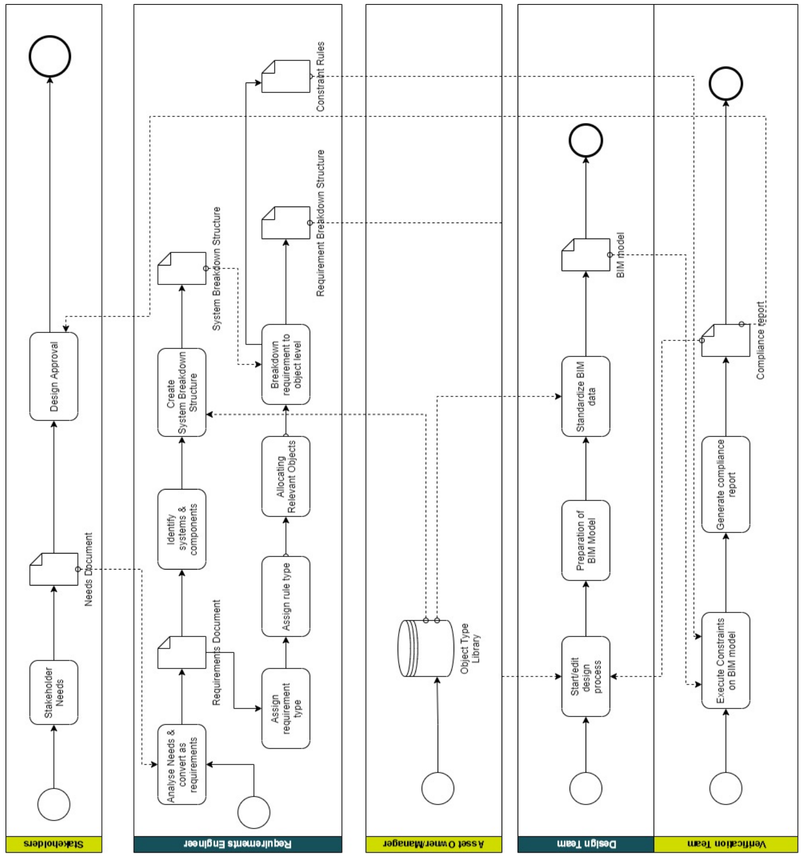
Table 13: Review of commercial solutions

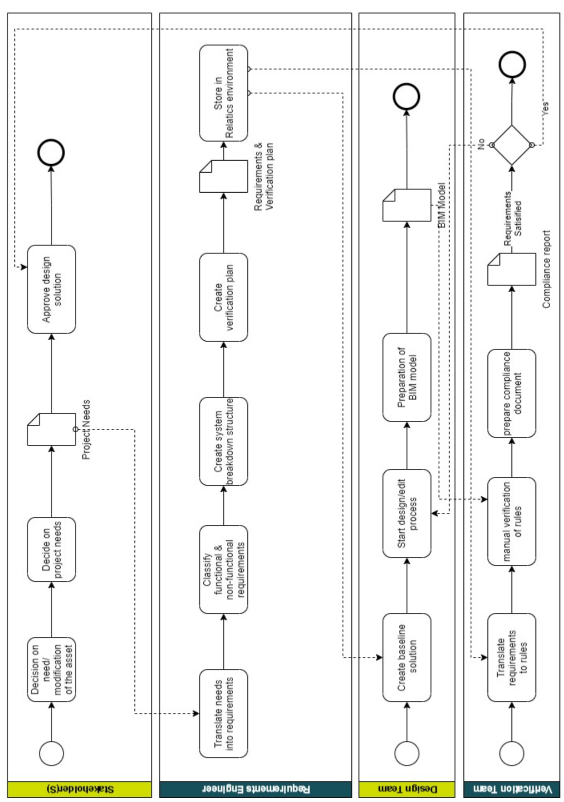
Index	Name	Domain	Methodology	Input	Output	Features	Verification Phase	Limitations
1	AEC3 Require1	Building regulations	Requirements are converted to rules using RASE markup	IFC	Textual reports, XML and IFC	- ISO12006 compatible solution - Text could be exported to semantic statements, BPMN workflow - Supports multi lingual and multi functional rules	Design phase; construction phase	- The software is ideal for general checking process but not compatible with system engineering processes
2	Autodesk Model Checker	Building regulations; client requirements	Hardcoding of requirements to rules	Revit	Report	- Compatible with current Revit eco system of common data environment	Design phase; construction phase; handover and maintenance phase	- Requirements needs to be hardcoded into the platform - Requires additional development for including more requirements

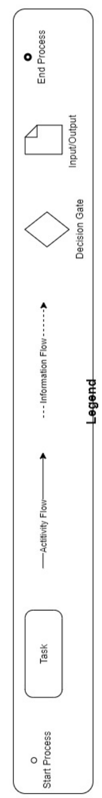
3	BriefBuilder	Client Requirements	Innovative GUI for capturing requirements at object level	IFC + Revit	Report	- GUI makes non programming experts to define object level requirements	Design phase; construction phase; handover and maintenance phase	- Requires external applications and API connectivity with Revit for checking\
4	CARS	Road and bridge design	Using word processing applications rules are structured and conncted to BIM tools via APIs	Not specified	Not specified	- Standardisation of contractual terms - Needs and document are maintained at a central place	Design phase	Software not available for checking
5	GliderBIM	Custom Rulesets	Innovation GUI for rule editing	IFC	Report, RFI	- supports system engineering process - Lifecycle data can be managed in same platform	Handover phase	- Needs development support in adding custom requirements
6	Xinaps	Custom Rulesets	Hardcoding of requirements to rules	Revit	Visual report	- Realtime model checking	Handover phase	- Requires standaridized data dictionary

7	UpCodes AI	US building codes	Predefined building codes as rules	Revit	Report	- Specifications at a single place	Design phase; construction phase	- Available only for US based building codes temporarily
8	SMART review	Building codes	Rule editor	Revit	Report in HTML format	- Workflow is more applicable for muncipalities and permit approval bodies	Design phase	- Used predominantly for design review and permitting
9	Jotne EDM-modelchecker	Building regulations; client requirements	Rules modelling in EXPRESS schema	IFC	Report in HTML format	- Better solution for product data management	Design phase; construction phase; handover and maintenance phase	- Not specific to AEC industry
10	Solibri Site or Enterprise Versions	Building regulations; client requirements	Rule editing via predefined rule templates	IFC	Report	- Widely used existing platform - Supports broad data formats	Design phase; construction phase; handover and maintenance phase	- Requirements needs to be hard coded into the platform - Requires additional development for including more requirements

F Workflow comparison


Stakeholders
Requirements Engineer
Asset Owner/Manager
Design Team
Verification Team
Stakeholder Needs
Needs Document
Design Approval
Analyse Needs & convert as requirements
Requirements Document
Identify systems & components
Create System Breakdown Structure
System Breakdown Structure
Assign requirement type
Assign rule type
Allocating Relevant Objects
Breakdown requirement to object level
Requirement Breakdown Structure
Constraint Rules
Object Type Library
Start/edit design process
Preparation of BIM Model
Standardize BIM data
BIM model
Execute Constraints on BIM model
Generate compliance report
Compliance report


Stakeholder(S)
Requirements Engineer
Design Team
Verification Team
Decision on need/ modification of the asset
Decide on project needs
Project Needs
Approve design solution
Translate needs into requirements
Classify functional & non-functional requirements
Create system breakdown structure
Create verification plan
Requirements & Verification plan
Store in Relatics environment
Create baseline solution
Start design/edit process
Preparation of BIM model
BIM Model
Translate requirements to rules
manual verification of rules
prepare compliance document
Compliance report
Requirements Satisfied
No
Yes


Legend
Start Process
Task
Activity Flow
Information Flow
Decision Gate
Input/Output
End Process

G Framework implementation approaches

Chapter 5 presents two approaches for testing the proposed framework using visual programming and NLP techniques. This two approaches were chosen based on its feasibility and easy access of software license for verification of framework. However, there are several other combination approaches through which the propose framework could be implemented in the project workflow. Several of these approaches requires some custom software development before adopting them in the workflow. Few such approaches using commercial solutions and open source solutions is elaborated below:

IFC data querying using SPARQL queries

There have been several researches carried out using this method for automated compliance checking. In this method, the requirements or specifications is formalised and is stored in the form of ontology. The BIM model in native or IFC format is converted to a linked data file format for example in a 'ttl' file format. The rules are created from the ontology using semantic web rule language(SWRL). The SWRL allows us to create queries that helps us to do the reasoning on the ontology and BIM data. Many ad-hoc applications in the domain of building codes checking, green building rating and evaluation, permit application automation is carried out using this technologies. However, as explained in the research gap of the thesis report, this method is not suitable for dynamically changing project specific requirements. These requirements are provided by stakeholders and is sometimes not written in a standardised way as available in building codes or green building standards. More research is required on how this technology can be used to write dynamically updating rules is necessary.

BIM data querying using graph databases

A graph data model can be used to represent, extract, and analyse relationships among objects. Based on the graph structure of the model several queries as proposed in the framework can be executed. No software application i.e academic or commercial solution is currently available for transforming native BIM models into graph structure. However, An automatic workflow for transformation of IFC models into a graph based model using the property graph database Neo4J as a graph database framework is recently developed(Ismail, Strug, & Ślusarczyk, 2018). The presented graph model is useful for data management as it allows to explore, check and analyse the complex relationships inside BIM models, and run complex queries for information retrieval. Since IFC format for infrastructure is still under development, conversion of those into graph databases needs to be researched upon.

Specification retrieval using Linked data technologies

This technology is used to tag or identify requirements from the specifications. Only a small number of researches have been conducted on using Linked data principles. Four steps are executed in the process: exposing data by a collaborative interface, publishing current contextual data as RDF triples, consuming and exploring combined knowledge as linked data, and providing semantic support. The rules can then be constructed as a SPARQL queries using collaborative interface. The SPARQL queries is then executed on the BIM models as explained in above section.

BriefBuilder + BIM Bridge

The Dutch company BIM connected has developed a platform called BIM-bridge to exchange and compare building data between different applications. BriefBuilder is an requirements management that provides similar functionalities as Relatics application. The combination of BriefBuilder and BIM-bridge application makes an effective application to quickly create, compare and verify BIM data. However, the application is still in a development stage and the current version is able to do model checks only spaces. The format supports only IFC models.

Upcodes.AI

Upcodes AI is an application that works as a design aid for architects and engineers that is available for native Revit models. The application helps in reviewing and flagging building code violation and other issues that might go unchecked in manual verification. Application provides a high level summary of the code violation and highlights the requirement or specification in violation. Upcodes AI analyses without any additional setup or input. This can be integrated into the current workflow efficiently without any changes. The major problem with application is that the requirements/specifications are hard coded and cannot be changed easily. More efficient way of specifying requirements using open standard technologies needs to be researched upon.

AEC3 Require1

AEC3 Require1 is an application for capturing regulations and requirements and makes them computer understandable. The application is based on the RASE methodology that was proposed by (Hjelseth & Nisbet, 2011). The application allows domain experts to mark-up the normative text using four colors based on the function. The marked text is used to construct rules. The rules are then applied on the BIM model as a constraints. The markup of the text happens using the data dictionary. The application however is not suitable for Systems Engineering process as described by Hjelseth and Nisbet (2011). Similar to other applications more research is needed on using the application for project specific requirements.

H Case Study - requirement classification and objects allocation

Table 15: Requirements list from project stakeholders

ID	Requirement Text	Verification Phase	Verification Method	Verification Plan
REQ001	For each structure placed on the substructure bearings, the bearings shall be replaceable. The bearings on the existing highway shall be able to be jacked independently without hindrance for the traffic.	Design Phase	Document inspection	Demonstrate on design drawing
REQ002	Structure shall have a design life of at least 100 years in the case of new non-replaceable structural components. Structure with a design life of less than 100 years shall be easy to inspect, maintain and replace without affecting the overall civil structure.	Design Phase	Document inspection	Demonstrate in maintenance and management plan
REQ003	The structure shall prevent height differences, buckling and cracks in the superstructure in accordance with [ROK 1.3 - Guidelines Design structures], including attachments.	Design Phase	Document inspection	Demonstrate on design drawing
REQ004	Concrete parts of the structure that could be affected by de-icing salts shall be protected against chloride penetration in accordance with [ROK 1.3 - Guidelines Design structures], including attachments.	Construction Phase	Document inspection	Demonstrate in maintenance and management plan

REQ005	The new structure shall be able to carry all loads in accordance with [ROK 1.3 - Guidelines Design structures] in such a way that no obstacles or dangers occur for the road traffic and the environment.	Design Phase	Document inspection	Demonstrate on design drawing
REQ006	Replaceable structural parts with maintenance requirement have a technical life span of atleast: - Expansion joints: in accordance with [ROK 1.3 - Guidelines Design structures (Annex B)], RTD 1007; - Bearings: 50 years; - Handrails: 50 years; - Masonry: 50 years; - Transparent windows (noise reduction measures): 30 years.	Design Phase	Calculation	Demonstrate in maintenance and management plan
REQ007	Structure shall provide space for an inspection path of at least 500 mm wide behind vehicle barriers and along the side verges and 2,200 mm in height	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ008	Structure shall feature silent (low noise production) expansion joints	Design Phase	Document inspection	Demonstrate on design drawing
REQ009	Curb sides shall provide facilties for cable ducts and associated components	Design Phase	Document inspection	Demonstrate on design drawing
REQ010	Structure shall be inspectable constructively using non destructive methods.	Construction Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.

REQ011	Structure shall be fire safe in accordance with - Guide Design Engineering Works of art - The guidance basic repressive fire services - User's guide extinction water supply and accessibility(NVBR).	Design Phase	Calculation	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ012	Structure shall be socially safe in accordance with manual secured by design, public space section	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ013	The bike path, sidewalk and roadway are at the same height and / or in sight of each other.	Design Phase	Document inspection	Demonstrate on design drawing
REQ014	The structure shall be resistant to vandalism, in the sense that it contains no loose parts or parts that are removably with simple hand tools.	Design Phase	Review	Demonstrate by reviewing the design drawings
REQ015	Structure shall, at height differences according to the building code, provide safety measures (railing) in accordance with [ROK 1.3 - Guidelines Design structures (Annex A)]	Design Phase	Document inspection	Demonstrate on design drawing
REQ016	Structure shall be able to withstand standard (environmentally friendly) cleaning products and methods.	Construction Phase	Review	Demonstrate by reviewing the design drawings
REQ017	Structure shall have a constructive reliability according consequence class CC3 [EN 1990].	Design Phase	Calculation	Demonstrate in the design calculation that the reliability is determined using parameters in accordance with a result Class CC3

REQ018	Structure shall prevent traffic hindrance due to snow, ice, and large quantities of rainwater.	Design Phase	Analyse	Demonstrate on drainage designs using existing standards and Directives. Use missing parts that could lead to decreasing snow and / or ice.
REQ019	The structure shall discharge rainwater in such a way that it causes no risk for traffic safety.	Design Phase	Analyse	Demonstrate on design drawing
REQ020	Rain water from structure shall flow along intersecting road discharging to the drainage system of the above along intersecting road.	Design Phase	Analyse	Analysis that design of a drainage system fits into the drainage system of the road.
REQ021	Structure shall be provided with masonry front walls and noise barriers of mixed red burned brick with a rough surface with joints recessed in dark color. Flat and / or monochrome brick are not allowed.	Construction Phase	Review	Analysis by means of drawings to show that required texture and color is achieved
REQ022	Structural shall contain masonry without filled vertical joints and has to be provided with 20 mm wide recessed dark horizontal joints.	Design Phase	Document inspection	Demonstrate on design drawing
REQ023	Masonry work shall be carried out in alternate application of Waal-, thick-, and double-Waal format.	Design Phase	Document inspection	Demonstrate on design drawing
REQ024	The stones used in structure shall be composed of variety of surface finishes with orange and red predominant colors.	Design Phase	Review	Analysis by means of drawings to show that required texture and color is achieved

REQ025	Expansion joints used in masonry shall coincide with mortar joints.	Design Phase	Document inspection	Demonstrate on design drawing
REQ026	Masonry work in structure shall be carried out in accordance to CUR recommendation 71 and CUR recommendation 82.	Design Phase	Review	Demonstrate on design drawing
REQ027	Structure shall be provided with masonry front walls and noise barriers in a vertically-oriented half-brick bond.	Design Phase	Document inspection	Demonstrate on design drawing
REQ028	Structure should not contain edges or ridges which birds can settle. Beads or ridges having a width of not more than 3 cm are allowed. Requirement does not apply to transparent noise barriers of urban art.	Design Phase	Analyse	Review on design drawings
REQ029	Viaduct shall be structurally safe in accordance with: - [Building Act 2012]; - [ARZ - starting note BBB]; - [ROK 1.3 - Addendum Earthquakes ARZ (including attachments.)]; - [ROK 1.3 (including attachments.)]; - [RBK (existing works of art); - [Euro Codes]; - [listing requirements of ProRail, i.a. OVS00030].	Design Phase	Document inspection	Demonstrate in the design that standards are applied in the design
REQ030	The alignment of the structure is in horizontal and vertical sense, to be connected to the alignment of the adjacent roads.	Design Phase	Analyse	Demonstrate on design drawing

REQ031	Structure should feature brickwork with specifications: - Durability Class MX-4 according to [NEN-EN 1996]; - Clinker bricks of quality; - Jointing: add hardness class VH 45 according to [CUR recommendation 61].	Design Phase	Calculation	Demonstrate in the design calculation that the brickwork is determined using mentioned specifications
REQ032	Modifications to existing structures shall comply with [ROK 1.3 - Guidelines Design structures], including attachments.	Design Phase	Calculation	Demonstrate in the design calculation that the brickwork is determined using mentioned specifications
REQ033	Structure shall be able to carry traffic loads resulting from: [ROK 1.3 - Guidelines Design structures (ROK)], including attachments.	Design Phase	Calculation	Demonstrate in the design calculation that the brickwork is determined using mentioned specifications
REQ034	Structure shall not use walls as a structural support system.	Design Phase	Analyse	Demonstrate on design drawing
REQ035	Columns for structures shall be positioned in an equal and continuous line of the existing work.	Design Phase	Review	Demonstrate on design drawing
REQ036	Structural components are required to follow the curves of the structure, in such a way that there are no visible facets.	Design Phase	Analyse	Review the drawings from design booklet
REQ037	Structure shall be provided with flat new structural components (Bridge deck edges, walls, covering stones, lintels, viewing windows, etc.) with a maximum width of 2 m and / or angle of 2.5 degrees.	Design Phase	Document inspection	Demonstrate on design drawing

REQ038	Structure shall provide a headroom for housing by going slow movement of at least 3.00 meters.	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ039	The structural framing slope front walls should be of the brand type Klosterman greenston or equivalent.	Construction Phase	Review	Demonstrate on design drawing
REQ040	Structure shall be equipped with an asphalt pavement with specifications in accordance with ROK 1.3 - structures Design Directive (Annex B), RTD 1009: 2012 Guidelines for the design of asphalt pavements on concrete and steel bridge	Design Phase	Document inspection	Demonstrate on design drawing
REQ041	The pedestrian and roadway shall be so constructed and arranged that cyclists and pedestrians are clearly visible to occupants of vehicles.	Design Phase	Calculation	Demonstrate based on lighting plan simulation
REQ042	Structure including all visible edges and connections, connect in such a way to be detailed that no discoloration occurs due to rainwater, moisture and dirt accumulation.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ043	Main works the underneath crossing, shall be suitable for securing objects of the Traffic Management System (signage, matrix boards, etc.) in accordance with [Directive CROW publication Signs 2014] and [Guidelines Design structures ROK] incl. Attachments.	Design Phase	Document inspection	Demonstrate on design drawing
REQ044	Edges of the roadway shall be vertical	Design Phase	Document inspection	Demonstrate on design drawing

REQ045	Edges of the roadway should be at least as high as the guide structure behind it.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ046	Noise reducing devices shall be part of the structure in accordance with VTW167	Design Phase	Document inspection	Demonstrate on design drawing
REQ047	Sound barriers used in the structure shall not be transparent to prevent bird casualties by markings on transparent sound barriers.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ048	Noise reducing devices shall be positioned at right angled connection Tapered and receding noise barriers are not allowed	Design Phase	Document inspection	Demonstrate on design drawing
REQ049	Structure shall serve as an integral unit including the edge beams, front walls and possible noise barriers.	Construction Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ050	Structure shall not have visible bearings.	Design Phase	Document inspection	Demonstrate on design drawing
REQ051	Abutments no more than 1,0 meters in height are visible	Design Phase	Document inspection	Demonstrate on design drawing
REQ052	Cables, pipes and rainwater drainage are mounted out of sight for users.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ053	There shall be no loose parts in the underside of bridges	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ054	The retaining walls of the structures shall be clad with brick work or finished with a similar quality	Design Phase	Review	Demonstrate by reviewing the design drawings in detail

REQ055	Masonry of structures shall be constructed with special masonry relationships	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ056	Structures shall have a rounder corner solutions.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ057	Urban structures shall have a refined detailing in concrete like casing stones, pillars and pillars.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ058	The noise barriers provided should seamlessly integrate with masonry walls.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ059	Sound barriers should integrate with masonry on roadway edges.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ060	Structures shall not have a narrow width and height profile	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ061	A stack of supporting beam and column visibly is not allowed.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ062	Columns, abutments and intermediate supports shall be expandable for future expansions of the structure.	Design Phase	Document inspection	Demonstrate on design drawing
REQ063	The cycle path of the structure should have a slope not exceeding 1:1	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.

REQ064	Embankment shall not exceed the slope of 1:2	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ065	Embankments under structures shall be provided with a coating similar to basalt columns.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ066	The space between the concrete columns be filled with dark gray basalt split	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ067	Viaducts and bridges shall be free of fences Fall protection shall be incorporated in to the brick parapet	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ068	Structures shall be free of siding, portals and advertising objects	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ069	Structure shall be painted with its own color material on the bottom	Design Phase	Document inspection	Demonstrate on design drawing
REQ070	Functional lighting shall be placed under the structure so that faces of pedestrians and cyclists can be identified at any time	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.

REQ071	Strret lights shall be placed on the structure so that faces of pedestrians and cyclists can be identified at any time	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ072	Modifications to the objects surrounding the structures shall be in accordance with ROK 1.3 - Guidelines Design structures (Annex A), Standard Details of concrete bridges.	Design Phase	Analyse	Demonstrate on design drawing
REQ073	Structure shall be provided with an anti-graffiti coating in all exposed surfaces	Design Phase	Document inspection	Demonstrate on design drawing
REQ074	Structure should be designed in such a way that rainwater shall first flow in the transverse direction, is discharged and then in the longitudinal direction	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ075	Structures shall be designed to carry loads due to induced earthquakes in accordance with ARZ - starting note BBB and ROK 1.3 - Addendum Earthquakes ARZ (including attachments).	Design Phase	Calculation	Calculation by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ076	Auxiliary construction and structural parts that are non functional shall be removed	Handover phase	Review	Demonstrate by reviewing the design drawings in detail

REQ077	Auxiliary constructions and / or of the structure parts that are non-functional, the whole of which removal is technically impossible, then auxiliary structure to at least 2.0 m below the surface level is completely removed.	Handover phase	Review	Demonstrate by reviewing the design drawings in detail
REQ078	The existing structures should be demolished in such a way that they should be reusable as a building material	Design Phase	Analyse	Analysis by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ079	Screen barriers shall only be applied to the structure if the height thereof is not less than 0,50m	Design Phase	Calculation	Calculation by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ080	The sound barriers and concrete parts of new structures shall be in accordance with Appendix 5 - ARZ CUR 100 Recommendation Clean concrete	Design Phase	Document inspection	Demonstrate on design drawing
REQ081	The fall protection works shall be designed as edge of bridge deck	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ082	The non-availability of the structure shall have same values as non-availability of the intersecting infrastructures.	Design Phase	Document inspection	Demonstrate on design drawing

REQ083	Structure shall have a free height between the lower and upper may be atleast 100 mm plus 25mm set space	Design Phase	Calculation	Calculation by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ084	The road way shall be within workplane with a maximum deviation in the x-y-direction of 4 m and in the z-direction of 2 meters.	Design Phase	Calculation	Calculation by means of design, drawings or computer files, with which it is shown that compliance with the requirement.
REQ085	Entrances to the structure shall be provided with red brick of the red clay specific Groningen.	Design Phase	Review	Demonstrate by reviewing the design drawings in detail
REQ086	Structures should be uniformly finished with standard elements	Design Phase	Document inspection	Demonstrate on design drawing

Table 16: Requirements classification and allocation

ID	Requirement Type	Rule Type	ID	Requirement Type	Rule
REQ002	Objective Requirement	Rule 1	REQ003	Objective Requirement	Rule 3
REQ006	Objective Requirement	Rule 1	REQ014	Objective Requirement	Rule 3
REQ007	Value Requirement	Rule 1	REQ019	Objective Requirement	Rule 3
REQ008	Existence Requirement	Rule 1	REQ020	Objective Requirement	Rule 3
REQ017	Value Requirement	Rule 1	REQ030	Value Requirement	Rule 3
REQ021	Existence Requirement	Rule 1	REQ034	Objective Requirement	Rule 3
REQ024	Existence Requirement	Rule 1	REQ036	Objective Requirement	Rule 3
REQ025	Existence Requirement	Rule 1	REQ049	Objective Requirement	Rule 3
REQ027	Existence Requirement	Rule 1	REQ052	Existence Requirement	Rule 3
REQ028	Existence Requirement	Rule 1	REQ056	Value Requirement	Rule 3
REQ039	Existence Requirement	Rule 1	REQ059	Objective Requirement	Rule 3
REQ044	Existence Requirement	Rule 1	REQ074	Objective Requirement	Rule 3
REQ046	Existence Requirement	Rule 1	REQ075	Existence Requirement	Rule 3
REQ047	Value Requirement	Rule 1	REQ076	Existence Requirement	Rule 3
REQ048	Value Requirement	Rule 1	REQ077	Value Requirement	Rule 3
REQ051	Value Requirement	Rule 1	REQ081	Existence Requirement	Rule 3
REQ054	Existence Requirement	Rule 1	REQ004	Objective Requirement	Rule 4
REQ063	Value Requirement	Rule 1	REQ005	Objective Requirement	Rule 4
REQ064	Value Requirement	Rule 1	REQ010	Objective Requirement	Rule 4
REQ065	Objective Requirement	Rule 1	REQ011	Objective Requirement	Rule 4
REQ066	Objective Requirement	Rule 1	REQ012	Objective Requirement	Rule 4
REQ067	Existence Requirement	Rule 1	REQ015	Objective Requirement	Rule 4
REQ068	Existence Requirement	Rule 1	REQ016	Objective Requirement	Rule 4
REQ069	Existence Requirement	Rule 1	REQ018	Objective Requirement	Rule 4
REQ070	Objective Requirement	Rule 1	REQ023	Objective Requirement	Rule 4
REQ071	Existence Requirement	Rule 1	REQ026	Objective Requirement	Rule 4
REQ073	Existence Requirement	Rule 1	REQ029	Objective Requirement	Rule 4
REQ082	Value Requirement	Rule 1	REQ031	Objective Requirement	Rule 4
REQ083	Value Requirement	Rule 1	REQ032	Objective Requirement	Rule 4
REQ085	Existence Requirement	Rule 1	REQ033	Objective Requirement	Rule 4

REQ009	Existence Requirement	Rule 1	REQ040	Objective Requirement	Rule 4
REQ013	Value Requirement	Rule 2	REQ042	Objective Requirement	Rule 4
REQ022	Existence Requirement	Rule 2	REQ043	Objective Requirement	Rule 4
REQ035	Existence Requirement	Rule 2	REQ053	Objective Requirement	Rule 4
REQ037	Existence Requirement	Rule 2	REQ055	Existence Requirement	Rule 4
REQ038	Value Requirement	Rule 2	REQ057	Existence Requirement	Rule 4
REQ041	Objective Requirement	Rule 2	REQ058	Objective Requirement	Rule 4
REQ045	Value Requirement	Rule 2	REQ062	Objective Requirement	Rule 4
REQ050	Existence Requirement	Rule 2	REQ072	Objective Requirement	Rule 4
REQ060	Value Requirement	Rule 2	REQ078	Objective Requirement	Rule 4
REQ061	Existence Requirement	Rule 2	REQ080	Objective Requirement	Rule 4
REQ079	Value Requirement	Rule 2	REQ086	Objective Requirement	Rule 4
REQ084	Value Requirement	Rule 2			
REQ001	Objective Requirement	Rule 3			

TU Delft

Royal HaskoningDHV
Enhancing Society Together