Towards **automated BIM** using Natural Language Processing :

An Interdesciplinary Perspective

Pratul Nema Nagendra Kumar
MSc Construction Management and Engineering

Royal HaskoningDHV
Enhancing Society Together

TU Delft

Towards automated BIM design using natural language processing: An interdisciplinary perspective

Masters Thesis Report

P. Nema Nagendra Kumar

4748166

Faculty of Civil Engineering & Geosciences

Construction Management and Engineering

Colophon

AUTHOR:

Pratul Nema Nagendra Kumar
Student- 4748166
M.Sc. in Construction Management and Engineering
Faculty of Civil Engineering & Geosciences, TU Delft
E-mail: Pratul.Nema@gmail.com
Phone: +31-630544747

GRADUATION COMMITTEE:

Chair
Prof.dr.ir. A.R.M Wolfert
Professor of Engineering Assets Management
Department of Materials, Mechanics, Management & Design (3Md)
Faculty of Civil Engineering & Geosciences, TU Delft
TU Delft E-Mail: A.R.M.Wolfert@tudelft.nl

First mentor
Dr.ir. G.A van Nederveen
Assistant Professor at the section Integral Design and Management
Department of Materials, Mechanics, Management & Design (3Md)
Faculty of Civil Engineering & Geosciences, TU Delft
TU Delft E-Mail: G.A.vanNederveen@tudelft.nl

Second mentor
Dr M.T.J. Spaan
Associate Professor at the Algorithmics Group
Faculty of Engineering, Mathematics and Computer Science, TU Delft
TU Delft E-Mail: M.T.J.Spaan@tudelft.nl

Company mentor
R. van Lanen
Tech lead at advisory group Digital Services
Transport & Planning
Royal HaskoningDHV
E-Mail: Ronald.van.Lanen@rhdhv.com

Preface

Before you lies the master's thesis report for the topic 'Towards automated BIM design using natural language processing: An interdisciplinary perspective'. The report has been written to fulfil the graduation requirements for the master's program of Construction Management and Engineering. I was involved in performing this research and documenting the report from March to September 2019.

This research has been done as a collaboration between TU Delft and Royal HaskoningDHV. The formulation of the research question was done together with the entire graduation committee. Fortunately, the entire committee was always approachable for any clarification and queries I might have had.

I am thankful for my chair Rogier Wolfer, my committee Sander van Nederveen and Matthijs Spaan for their guidance throughout the entire research. I am equally grateful for my company supervisor Ronald van Lanen, for all his time and spot-on guidance. I am glad I had the opportunity to work with the entire digital services team especially Tom Moekotte who fulfilled the role of my daily supervisor in the company. Koen van Viegen has been a guiding figure without whom this research would never have begun.

I appreciate the valuable time and guidance of Lucy Lin form the University of Washington, without whom this project might not have ever been completed. A special thanks to Hans Hober for validating the research question and constant support with knowledge of the design process, throughout the research. Yorick Fredrix was extremally helpful with his knowledge in data science and the ability to understand the project.

I am thankful to Soumik Guha, Adithya Eswaran, Khyathi Rudraraju, Asmeeta Das Soni and Harsh Soni for thanking out time to help me conceptualise, validate, proofread and design graphics for my report.

Finally, I would like to thank my parents for making it possible for my study at this prestigious university.

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

Summary

This research stemmed from the curiosity question 'how can we automate tasks in the design phase of a construction project?' The design phase is chosen as it offers the most amount of control during the project. The design of an intervention for automating parts of the design phase can be carried out without much funding and resources as compared to automation in the construction phase.

The report begins with a brief introduction concerning the inefficiency in the construction industry and its inability to keep up with automation as compared to the automobile industry; where automation and production line practices have become an industry standard. The context for the research is set by defining Parametric and Generative Design. The key difference being the information they require and how they process the design. The context makes it clear that automation requires data and information and this information needs to be in an internal context. An internal context represents an environment that allows for the unrestricted and semantic comprehension of the information. In a more real sense, Information for any task (needed to be automated) comes from different sources, so an internal context speaks of the file format, the accessibility, and the relations the information has with various sources of information. Therefore internalising the information is the first task of automation of any task.

Before internalising the information, it is essential to have an understanding of the flow of construction information in the Netherlands. Since the construction industry in The Netherlands has embraced System Engineering, there are various sources of building information such as BIM repositories, requirements management systems and object libraries. Most of these systems are already connected in some form or the other by exchange formats such as IFC, government-issued wrappers such as COINS or privately developed software solutions like Neanex. These exchange formats help with internalising the information but, a recurring issue with all these exchange formats is that they as capable of re-encoding the information, but they do not add any smart functionalities or automate the selection of information that needs to be encoded. The area of focus of this research is the lack of automation in selecting the right information to be encoded together.

An investigation into the design process was carried out to identify the specific process that can be automated. The result of this investigation is a process map which explains the entire design process. The validity of the process map was reinforced by the fact that it reflected the design process from the Systems Engineering guidelines for Civil Engineering. A critical step in the design process is the interface between the Functional breakdown structure (FBS) and the systems breakdown structure (SBS). The interface manifests itself when function/ technical requirements in the FBS are allocated to an object in the object/ Systems breakdown structure before they can be designed.

At this point, the technological system that is at play and its drawbacks were understood so the research question was formulated as 'How can we automate the allocation of requirements to object libraries?' To answer the research question, several different technologies and innovations in the IT industry will be explored.

The first challenge was to find and internalise the information The exploration begins with identifying the requirements, which are stored in a Relatics workspace and the objects which were stored in IFC files. A Python package called PyRelatics was used to internalise this information, but this proved to be impossible as Relatics has relinquished support for some of the web API's that the package used. The level of detail on the IFC models also proved to be insufficient to produce an accurate project object list. The inability to internalise the information made it necessary to redefine the sources of information. It was decided that the requirements would be directly downloaded as an excel file form Relatics and the objects would be derived from the OTL of Rijkswaterstaat.

With a reliable new source of information, four criteria were developed for the proposed system; these criteria were derivative of the sub research questions. The essence of these criteria was that the proposed system should capture both the syntax and semantics of the information and use this information to allocate the requirements to the objects. The proposed system should also be transferable to other types of assets apart from what it will be tested against.

Natural Language processing is generally used in the IT industry to capture the syntax and semantics any textual information. The proposed system uses a Dependency Parsing to identify the parts of speech of the requirements and object descriptions. The dependency parser adds syntactic information to the requirements and object descriptions. Using the syntactic information, the proposed system performs an operation known as noun chunking. Noun chunking breaks the requirements and object descriptions into meaningful phrases which can now be semantically analysed.

The proposed system performs the semantic analysis with the help of a word embedding. Word embeddings are vector representations of words in a multi-dimensional vector space. The syntactic chunks of information about the requirements are plugged into the word embeddings. When the chunks are in a vector space, it is possible to measure their semantic similarity because words of similar meaning have similar word vectors. The measurement is done by finding the dot product of the vectors of the two chunks. This is also known as cosine similarity. Semantic cosine similarity measurements of the syntactic chunks proved to be the perfect blend information to allocate requirements to objects. The cosine similarity score was measured for all the combinations of requirement chunks and object chunk. The proposed system prioritises requirement chunks that are similar to the requirement title. With this, the proposed system was able to produce results that met all the criteria set out at the start.

Although this system met all the criteria for a proposed system, a group of specialists recommended that a classifier might be a better solution. After building a neural classifier to perform the same task it became clear a neural classifier or any type of classifier it is intrinsically unsuitable for the task of allocating requirements to objects. A neural classifier is incapable of handling the dynamic nature of a project object library, which can change several times a day. The dynamic nature implies that a classifier would be required to be retrained every time a new object is added to the project, which could result in an overtrained model.

Verification and validation were carried out to test the performance and relevance of the proposed system. The verification process takes a quantitative approach to answer the

question ‘did we build the system, right?’. Precision and recall were to measure the performance of the proposed system. Precision refers to the number of objects retrieved that were relevant to the requirement. Recall refers to the fraction of the relevant objects which have been retrieved. The results from the proposed system showed that the precision was at its highest of 60 to 80 per cent at a recall level of 60 to 80 per cent. It also showed that when the requirement is relevant to two vastly different objects, one of those objects would tend to be left out. This was because the system prioritises chunks that are similar to the title of the requirement. The results were then compared to those produced by a phrase matcher. The phrase matcher only managed to identify the right object for one of the twenty requirements that were verified.

The validation process took a qualitative approach to answer the question ‘did we build the right system?’. The validation was done by conducting interviews with contract managers, tech specialists/ developers and specialist engineers. The questions were of four main categories, the perceived value, the applicability of the system, the choices made during the design, future applications and prospects for the proposed system. The results of the validation show that the inability of the system to deal with Dutch data and the degree of detail of the verification are current barriers in full deployment. The validation also showed a plethora of uses for the proposed system from automating internal project management to selecting verification and validation procedures. The validation process acts as the basis for the recommendations.

The Verification and validation are followed by a discussion section where the interdisciplinary nature of the problem is discussed. It is argued that a pure data science approach to the problem should be avoided as this might lead to a situation where a classifier is deemed the perfect solution. The research proposes that future research should take place in interdisciplinary teams. The discussion section also speaks of some of the flaws of the proposed system and the research.

The research concludes that the proposed system meets all the research questions, but the degree of accuracy can be improved. The improvements are presented as recommendations for researchers and industry. The recommendations have been allocated either to academia or industry, but most of the recommendations require the combined effort of both. The recommendations are geared towards improving the proposed system and providing a direction for future research.

Contents

List of figures

List of Tables

List of abbreviations

AEC	Architecture Engineering and Contruction
API	Application Program Interface
BIM	Building Information Modelling
BM	Building Information Modeling
CNL	Controlled Natural language
DBB	Design Bid Build
EA	Enterprise Architect
GUI	Graphical User Interface
ICT	Information and Communications Technology
IDEF0	Icam Definition for Function Modelling
IFC	Industry Foundation Class
ISO	International Standard Organisation
NLP	National Language Processing
NLTK	Natural Language Toolkit
OTL	Object Type Library
OWL	Web Ontology Language
POS	Parts of Speech
RMS	Requirements Management Systems
RWS	Rijkswaterstaat
SBS	Systems Breakdown Structure
SE	Systems Engineering
SMART	Specific, Measurable, Achievable, Realistic, Timebound
SVM	Support Vectar Machine
XML	Full Extendable Mark-up Language

1 Introduction: AEC industry Vs. Others

The construction industry has been accused of not moving forward from its glory days in the 19th century (Winch, 2003). In the 19th century, the construction industry was in its prime, with projects like the Suez Canal and the transcontinental railway (Winch, 2003). Many say that the construction industry never moved into the 21st century while other industries were able to embrace the assembly line and later automation. For instance, the automobile industry has undergone massive modernisation technics and increased its efficiency many folds since its conception. Many authors have made bold, and somewhat hysterical statements about the construction sector such as 'where is the Henry Ford of future housing systems?' (Miles, 1996) Also, the 'industry God forgot'(cited in Lawrence and Dyer, 1983, p. 158). The allegation the construction industry did not 'innovate' has also been challenged on the manner of measurement rather than the fact that the industry has fallen behind(Winch, 2003). In most aspects, it is a widely accepted fact that the construction industry did not undergo the scale of automation that was observed in other industries, and some intervention is required.

The Farmer report (Farmer, 2016) discusses some of the major causes of the disruption in the construction industry. Farmer speaks of the following indicators:

- low productivity
- Low predictability
- Structural fragmentation
- Leadership fragmentation
- Low margins
- Adversarial pricing models & financial fragility
- A dysfunctional training funding & delivery model
- Workforce size & demographics
- Lack of collaboration & improvement culture
- Lack of R&D & investment in innovation
- Poor industry image

Therefore, in the past decade, the construction industry has started to see a change in attitude towards automation. There is also the idea that a substantial portion of the knowledge bearers in the construction industry is on the verge of retirement, and this could result in a major loss of knowledge. This notion stems from the fact that in countries like the UK, twenty-two per cent of the workforce in the construction sector is around fifty years old and fifteen per cent is more than sixty years(Woodhead, Stephenson, & Morrey, 2018). Irrespective of the causes, it is important to acknowledge that there are some positive steps in the right direction. The drive for efficiency and effectiveness in creating a paradigm shift in the entire industry. The United Kingdom calls it 'Industry 4.0' and the United States terms it as 'Industrial Internet'(Woodhead et al., 2018). In the Netherlands, companies such as Royal HaskoningDHV call it a 'digital transformation.'

2 Context: Automation in Engineering

In this section, we will look at what are the current trends in the AEC industry about automation.

This research attempts to help the construction industry to capitalise on the 'digital revolution' and move closer to a more efficient workflow by automating some of the laborious and time-consuming tasks in the design phase. The first step in this process would be to identify which aspects of the design process have the potential to be automated. An important parameter to take into account is the need for automation in a given task, tasks that are not time-consuming or repetitive need not be automated. The ultimate achievement of automation in the design process would have systems that would automatically generate designs given a set of inputs or parameters; this is the primary objective of parametric and generative design.

The primary difference between parametric design and generative design is that parametric design takes a rule-based approach to design, while generative design tries to learn parameters from existing designs. In parametric design, the developer tries to establish specific rules that would produce the result required (Bohnacker et al., 2009). In generative design, the principle is to learn parameters that can be varied based on an extensive collection of designs. Generative design tends to use machine learning algorithms; this is also the case in the Dreamcatchers(Kazi, Grossman, Cheong, Hashemi, & Fitzmaurice, 2017).

2.1 Parametric Design

Parametric design as a concept has been around for a long time but the maturity of modelling tools in end of the 20th century was insufficient to be able to augment artificial intelligence systems to modify 3D objects after they are created (Monedero, 2000). The principle challenge was that the designer needed to be able to go back to the design, and there would be constant changes that would occur during the design process. In the past few years, the parametric design has come into the limelight once again. There are several attempts to define a new way of thinking that is more supportive of parametric design(Bhooshan, 2017; Oxman, 2017). These new ways of thinking contribute to the idea of parametric design,

The first requirement to be able to conceptualise parametric design is parametric modelling, which what Monedero speaks about when he says, 'Interactable 3D models'. The advantages of parametric modelling were well known as the manufacturing industry had moved away from drafting and embraced 3D solid modelling. 3D modelling opened the gates for automation and development quality control applications (Sacks, Eastman, & Lee, 2004). Another critical requirement is the need for an open standard format to be able to exchange the information. An open standard is crucial as most 3D modelling software developers were developing their file format and data storage systems for their application(Thein, 2011). The above requirements formed the stepping stones for Building Information Modeling (BIM) and the development of Industry Foundation Class (IFC), an open standard to enable the interoperability of building information (Eastman, Teicholz, Sacks, & Liston, 2011). BIM and IFC allow the addition of various types of information to be associated with a specific 3D object. With BIM becoming an industry standard, the stage is now set to be to move towards parametric design.

2.2 Generative Design

The manufacturing industry has invested in 3D modelling a much longer time than the construction industry. Therefore manufacturers are several steps ahead in terms of generative design. There are prototypes such as the Dreamcatcher project at Autodesk Research, which can generate design alternatives based given set of parameters. The generative design capabilities are showcased for the body of a bicycle and the frame of a drone (Kazi et al., 2017).

One of the first steps of generative design is to be able to define a design problem to the system, and this is done in the Dreamcatcher system with the help of documents, goals, and constraints. The DreamCatcher system generates alternatives based on a knowledge base that primarily consists of objects that fulfil specific function or constraints. The functions and constraints are not only images or 3D models, but there is also the scope of using natural language(Kazi et al., 2017). How natural language is used is interesting because it opens a wide range of possibilities to capture the data required for the. The attempt is to convert some of the natural language into what is known as Controlled Natural language (CNL) design (Cheong, Li, Shu, Bradner, & Iorio, 2014). We will come back to such concepts later on in the report.

2.3 Data and Information

Looking at the advantages parametric and generative design in the manufacturing sector, there is a keen interest in the AEC industry to see the application of artificial intelligence and machine learning(Woodhead et al., 2018). Machine learning has seen a multitude of uses starting from detecting defects in masonry to predicting injuries on sits and even to enrich BIM data (Bloch & Sacks, 2018; Tixier, Hallowell, Rajagopalan, & Bowman, 2016; Valero et al., 2019). A significant conclusion in all these applications is the need for data. Data forms the central pillar of any application for any machine learning, artificial intelligence(Woodhead et al., 2018). This has led to an increased value for data resulting in the emergence of Big Data (Marr, 2015). Data is one part of the context; additionally having the data in the structured format that is interpretable by a computer (Information) is also a cha. Moreover, hence take the form of information. (Santos, Martinho, & Costa, 2017).

In light of these developments, there have been many attempts to automate some of the operations in the Civil Engineering domain. A rather prevalent domain is the automated checking of regulations. When developing algorithms that can generate parts of design automatically, there are two primary challenges; the first is to have the information in an internal context, i.e., having the information in a format that can act as input for an intelligent system. The second is to find a suitable AI technology that can help in making design decisions (Karan & Asadi, 2019). This research will attempt to assist the first challenge of internalising the information. To understand the nuances of the information that is available and

3 Problem Definition

In this chapter, we will delve deeper into the information management framework that is used in the Netherlands to identify the scope for automation. Next, the limitations of the existing framework which forms the bases for the problem statement.

3.1 Building Information Management in The Netherlands

In this section, the focus will be on the design process that the industry follows. That includes how it functions and who are the people involved. This section also forms the basis for the problem statement and the research question. A brainstorming session was carried out to understand the possible problems in the current industry practice. Figure 1 shows the outcome of the section where each knowledge sphere is explained below. It should be noted that this is one of many scenarios in which the given systems interact.

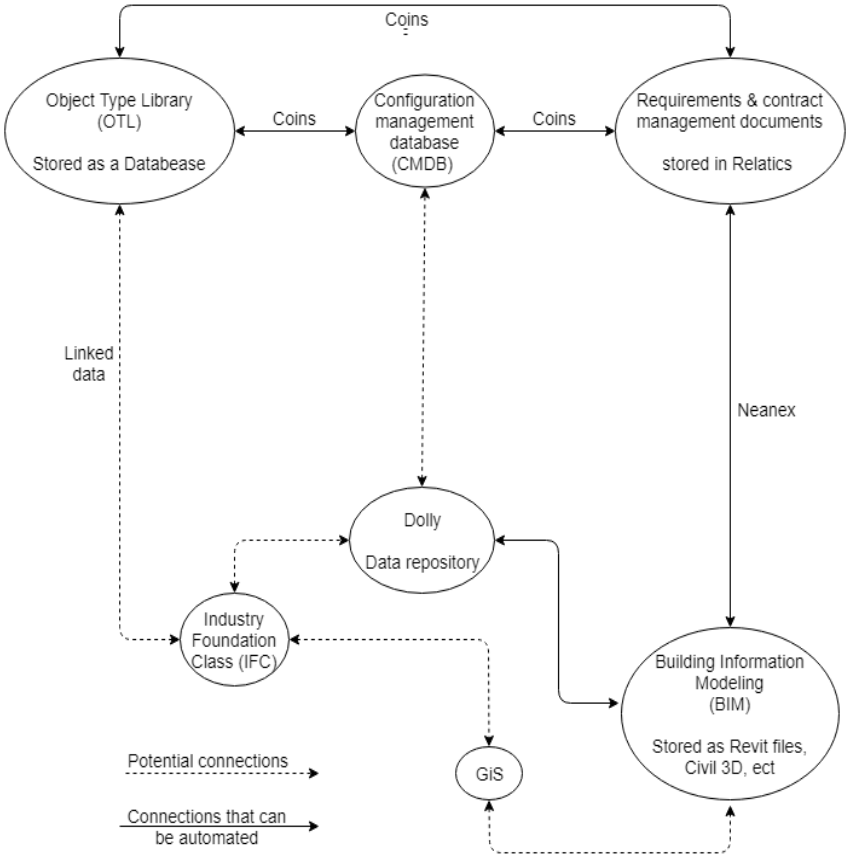

Figure 1: Industry practice for building information management

3.1.1 Object-type Library (OTL)

An OTL is a library of standard object-types along with their names and properties/ specifications. An OTL can virtually store any data from geometry data to metadata. Metadata is vital as each object has its own set of properties (O'Keeffe, Alsem, Corbally, & van Lanen, 2017). OTLs typically store data in a more dynamic way than the traditional (static) way they usually are captured in any relational database(Hoeber & Alsem, 2016). Rijkswaterstaat (RWS) is the public works and water management board of the Netherlands. RWS has an ambitious plan to document and stay up-to-date on all the data from all its assets. Asset data can be

both quantitative and qualitative, and an essential part of quantitative data is the decomposition of data. Decomposition is the breakdown of information into different levels. RWS has developed a reliable asset management platform for the based on a uniform Object-Type Library (Brous, Herder, & Janssen, 2015). Every time the term OTL is used in this research; it refers to OTL developed by RWS.

3.1.2 Requirements Management Systems (RMS)

Systems Engineering (SE) has been adopted in the construction sector in the Netherlands and has become an industry-wide practice. SE has made it possible to improve communication between all stakeholders in a project and encourages integrated work. The major players such as ProRail, Rijkswaterstaat, Bouwend Netherland, Vereniging van Waterbouwers, NLingenieurs and Uneto VNI have endorsed SE in all their projects and require those who collaborate with them to be capable of working with SE (Alsem et al., 2013).

The cardinal working principle of Systems Engineering is that the client's needs and the necessary functionality need to be internalised. The needs are then translated to requirements and documented in the conceptual phase of the project. Only after the documentation can the actual synthesis of the design can begin following which the validation of the design (Alsem et al., 2013).

The requirements management systems manage all the requirements in a systematic manner such that required information can be retrieved as and when necessary. In most of the organisations in The Netherlands, this is done using a cloud-based platform known as Relatics. All such repositories tend to have a large amount of structured and unstructured data as they are also used ad document storage systems from time to time. The Relatics environment is an ideal source of data for this research.

3.1.3 Building Information Modelling (BIM)

Although BIM has been introduced as a concept in the section on parametric design as one of the building blocks of parametric modelling, there is much more to be discussed when it comes to the context of The Netherlands.

BIM has moved towards an industry-wide practice, but there remains uncertainty with the implementation strategies used by different firms (Sebastian & van Berlo, 2010), so there is a lack of uniformity in the industry with regards to BIM standards. The lack of uniformity is due to the various file formats used by different software packages and different companies. Very often, information has to flow from one environment to another, and this creates a problem of interoperability of data when proprietary file formats are used. Sadly this is the industry practice where Autodesk Revit has its file format '.revit.' An open standard such as IFC is meant to solve this problem(Eastman et al., 2011), and it will be discussed in the following sections.

3.1.4 Industry Foundation Class (IFC)

IFC has its origins back in the 1990s but only came into significance after the widespread need for interoperability. The IFC Schema, along with it is native modules are used to transfer information between the various system that requires the same information (A H M Berlo, Beetz, Bos, Hendriks, & C J Tongeren, 2012). IFC solves the problem of interoperability to a great extent but the there are two principal factors that it depends on, the implementation of

IFC import-export functions from the schema in the application that wants to include IFC compatibility and the modelling of the assets itself (Geiger, Benner, & Haefele, 2015). These are the two factors that have prevented the widespread incorporation of IFC in the AEC industry. The applications that are used either fail to capture all the information into the IFC file or the modelling is not detailed enough to accept the incoming information. The wide array of information can be seen in the image of the IFC schema in. IFC has now come to be generally expected as a standard BIM data exchange format IFC in the Netherlands, and the need for an open standard is, for the most part, acknowledged in the Netherlands (Van Nederveen, Beheshti, & Willems, 2010). IFC forms an integral part of any central repository as it provides uniformity to an otherwise diverse environment, which will be seen in the next section on Dolly

3.1.5 Dolly Project

Dolly is a large data repository project that is has initiated by Royal HaskoningDHV. This repository is designed to store primarily BIM data in IFC format from various projects that are in progress or have been completed inside the organisation. The reasoning behind the development of this repository is built plugins into this environment that can use it as a knowledge base for parametric and generative design. The dolly system can potentially provide a large amount of data to this research.

3.1.6 COINS

COINS is an information exchange format that is based on Open- BIM concept, which was developed by a Dutch consortium (Van Nederveen et al., 2010). It stands for 'Constructive Objects and the Integration of processes and Systems'(Hoeber & Alsem, 2016). COINS use IFC and Web Ontology Language (OWL) standards to function as an Full Extendable Mark-up Language (XML) rapper for all the files. The importance of COINS lays in its use in the documentation in data repositories such as RWS's OTL. Information from the OTL such as object trees can be downloaded in a COINS package to be extracted.

3.1.7 Neanex

Neanex is a software application plugs into popular BIM tools like Autodesk Revit, Civil3D and Navisworks and establishes a connection to a Relatics workspace to enable the exchange of information.

3.2 Limitations of the current framework

The current system stores a large amount of data in a plethora of formats, but they are inherently flawed in some way or incorrectly implemented in the current industry framework. Below are some of them

COINS

The COINS system has no ground-breaking concept or development that was done as there were many stakeholders involved (Van Nederveen et al., 2010).

RMS

With its roots in Systems Engineering, RMS require all the stakeholders to define the requirements in a SMART[1] manner (Wasson, 2015), but this is never the case in the construction industry.

BIM

There is no denying the fact that BIM will be the future of the AEC industry, but the problems of interpretability and level of detail in the modelling are holding back the full potential (Van Nederveen et al., 2010).

3.3 Problem Statement

The data stored in the requirement management system is essential for the design process as it acts as guiding principles for the design. The current industry practice is such that the requirements are poorly defined. Hence how the requirements are defined prevents the information which is embedded in the requirements to flow to other systems in the framework.

[1] SMART – Specific, Measurable, Achievable, Realistic, Timebound

4 Research Design

With the given problem, we can formulate the research question and then design a research methodology to formulate a solution and evaluate it.

4.1 Research Objective

The non-SMART definition is why the information in the requirements have not been harnessed for any automation. Hence this problem acts as a barrier for the automated transfer of this knowledge to other systems in the framework.

The objective of this research is to intervene the design process to enable the automated transfer of requirements to objects in the other systems.

4.2 Research Scope

The apparent solution for the unsystematic definition of requirements is to have a system that provides a framework that will help the respective stakeholders to better define the requirements. The issue is that such frameworks already exist in the form of the 'Guidelines for Systems Engineering within the Civil Engineering sector'. Although Systems Engineering is embraced in the AEC sector in the Netherlands, not all the practices, have been followed, and strict enforcement of any guidelines is next to impossible.

For the above reason, this research hopes to present an intervention that does not ask the stakeholder to work in a new manner but tries to work with the existing system and industry practice. In order to design such an intervention, it is essential to have an overview of the design process that is in practice. The explanation of the design process will be in the next section, and it forms the context of the research.

4.3 Research Context

The research context section explains the design procedure that is followed in the construction industry. The design process is the steps that are followed when any design is to be made. An exploratory interview was conducted with an industry professional to identify the design process. The interview was conducted in the presence of one of the committee members in order to enrich the understanding of the design process. The Systems Engineering Guidelines were also used to give a contrasting view from theory to practice. The design process has been visualised in Figure 2 using the IDEF0[2] format. The basic IDEF0 principle can be seen in Appendix 3: IDEF0 .

The design process begins with the need for a specific asset or a modification such as an expansion or replacement. The contracting authority can identify this need themselves or it can be brought to their notice by an external party but what is essential is that the decision to construct a new asset/ modify the existing asset has already been taken. What is being discussed is which functions are required of the new/ modified asset. The client and the relevant stakeholders decide on what the new asset should satisfy. The result is a set of top-level requirements which are handed over to either a consultant or a contractor depending on

[2] IDEF0 - Icam Definition for Function Modelling

the procurement model that is being used. Let us consider a Design Bid Build (DBB) style contract for simplifying the explanation. The consultant would begin detailing the Top-Level requirements into functional and operational requirements. These requirements would follow a functional breakdown structure. The outcome of this entire process is a set of detailed requirements that are usually stored in a cloud-based system such as Relatics or Enterprise Architect.

The detailed requirements are allocated to specific objects that would either use or procured. The objects would be stored in object libraries such as OTL or manufacturers catalogues or a set of objects in a systems breakdown structure (SBS) that a contractor developed himself. These SBS's could make derivative of libraries such as the OTL or based on the experience of the contractor. There is a lot of room available to modify the definition of these objects. Therefore, a large and vastly different vocabulary is used to define the objects. The varied definitions happen even if though the Systems Engineering Guidelines require the formulation of a project vocabulary and a Glossary at each stage of the project. The complex language and unique vocabulary make the of allocating the requirements to objects a manual task that only an engineer with sufficient knowledge can do this task. A set of objects with their required specifications is essential for the design process. During the design of the product, the designer would look at the requirements and only then start designing the objects to ensure the design would meet the given requirements.

Now the research question can be formulated as there is a clear understanding of the design process.

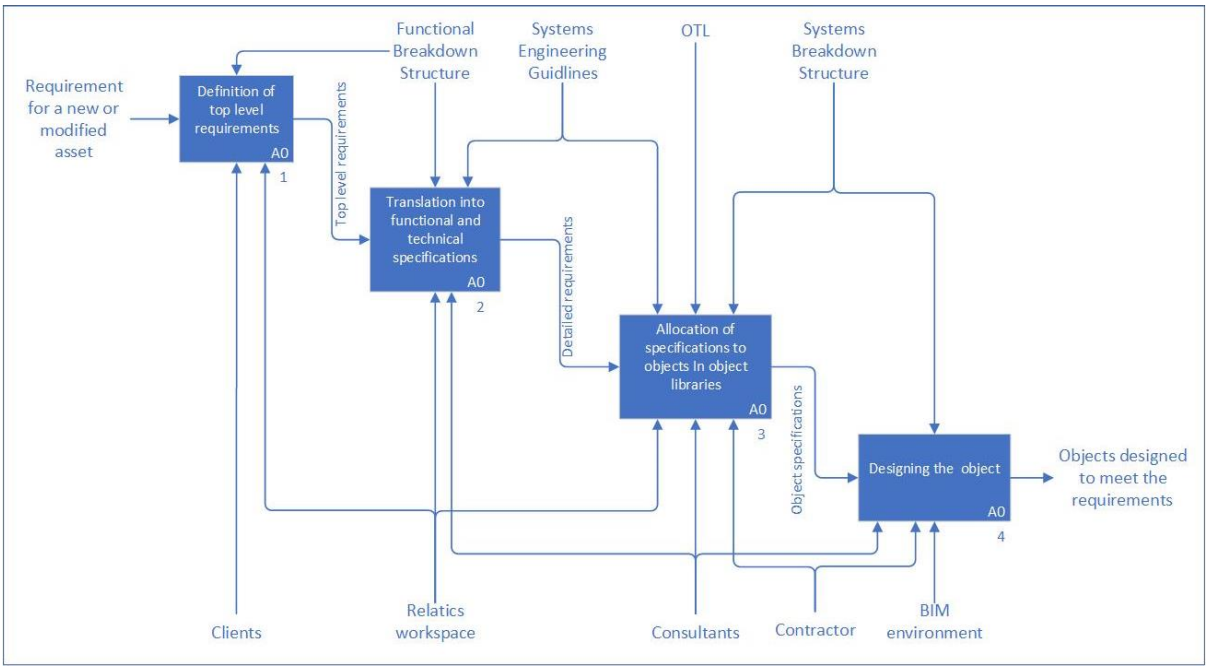

Figure 2: Asset design process overview

4.4 Research Question

The above should process works in tandem with the ISO 15288 and the Systems Engineering guidelines, but in reality, this does not happen as all the parties do not communicate in the same manner resulting in several delays as some objects are tagged with the incorrect

requirements. This also means that only a human (Engineer) can perform the task. The task of reading requirements is tedious and time-consuming. It is a task that is repetitive as new requirements enter a project throughout the design phase, and nobody finds this an intuitive task.

With the problem statement and the above flaws in the design process, we can formulate a research question.

"How can we automate the allocation of requirements to object libraries ?"

The research gives rise to the following sub-questions.

- How can we internalise the requirements and objects?
- How can we capture the syntactic and semantic structure of a requirement and object?
- How do we use this information to allocate a requirement to an object without human intervention while ensuring the designed system is transferable to all assets ?

4.5 Methodology

In this section, we will establish the theoretical framework for this research. The framework contains several feedback loops, so several steps were iteratively repeated. The repetitions were be followed by the practical application of the proposed framework and a diagram that represents the number of iterations that were carried out.

4.5.1 Theoretical framework

This research is practice-oriented research that aims to suggest a change or interventions in the design process within the AEC industry. The following steps are involved in this practice-oriented research approach.

- Problem analysis
- Diagnosis
- Design
- Change
- Evaluation

This procedure is based on the one described by Verschuren, Doorewaard, & Mellion, 2010. The above steps have been constructed into a design process with feedback loops at critical stages, as seen in Figure 3. There are five steps in this process, and each of them has a corresponding outcome indicated corresponding to it. The process spans from the analysis of the problem to the evaluation of the design. The steps involved are as follows:

1. Analysis: The precursor to this step is the problem statement. In this step, the problem at hand will be broken down into achievable tasks. The analysis would result in a set of criteria by which would be used in the evaluation of the design to ensure the design is performing the required task. These criteria are developed for the provisional design and not meant for validation in implementation in the industry.
2. Synthesis: Based on the required criteria to be met for the problem, a solution was suggested, and this would result in a provisional design that is meant to solve the problem.

3. Simulation: Based on the steps described in the provisional design, the task will be executed. The technology suggested in the synthesis will be used in this phase on the available data. At the end of the simulation, there will be a set of expected properties, which are mostly the results of the process. It is also possible that there are no results as the method suggested in the synthesis phase is not applicable or is flawed.
4. Evaluation: The results from the simulation will be checked if they meet the criteria that were formulated in after the analysis. After this step, we will be able to evaluate the value of the design to see if the problem is being solved. It is possible that some part of the problem is solved, and some parts require further refinement in the synthesis or even the analysis phase. There is also room for introspection on what could have gone wrong
5. Decision: Based on the value of the design that was proposed it be either redesigned in the synthesis phase or it is realised that the expected result is not possible. In that scenario, the problem needs to be realised and a fresh set if achievable criteria are defined

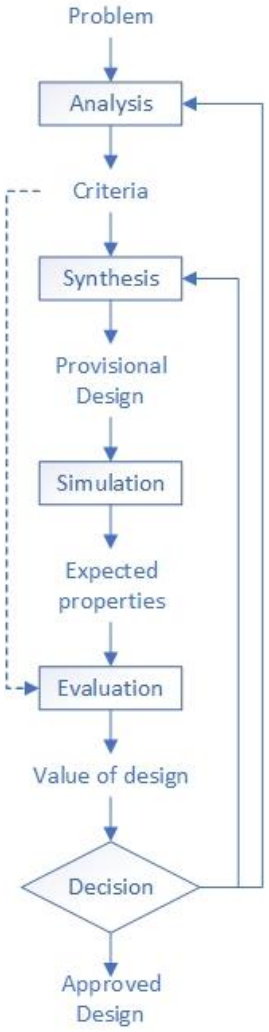

Figure 3: Design process template

4.5.2 Applied Methodology

When applying the framework see in Figure 4 for this research project, the feedback loops are essential for finding the right system to be able to answer the research question. Several techniques were explored to find a suitable solution given the data and time constraints available. The research took place in several Scrum Cycles. Each iteration of the framework also corresponds to a scrum cycle in order to be able to use the reasons for the failure in the evaluation stage. A visualisation of the process is in the image below following which we will look at which approach was taken in each iteration, what failed and what worked and will be taken into the next iteration. After a system that meets all the criteria is designed the system, verification and validation were carried out. The results from these form the bases for the discussion and recommendations.

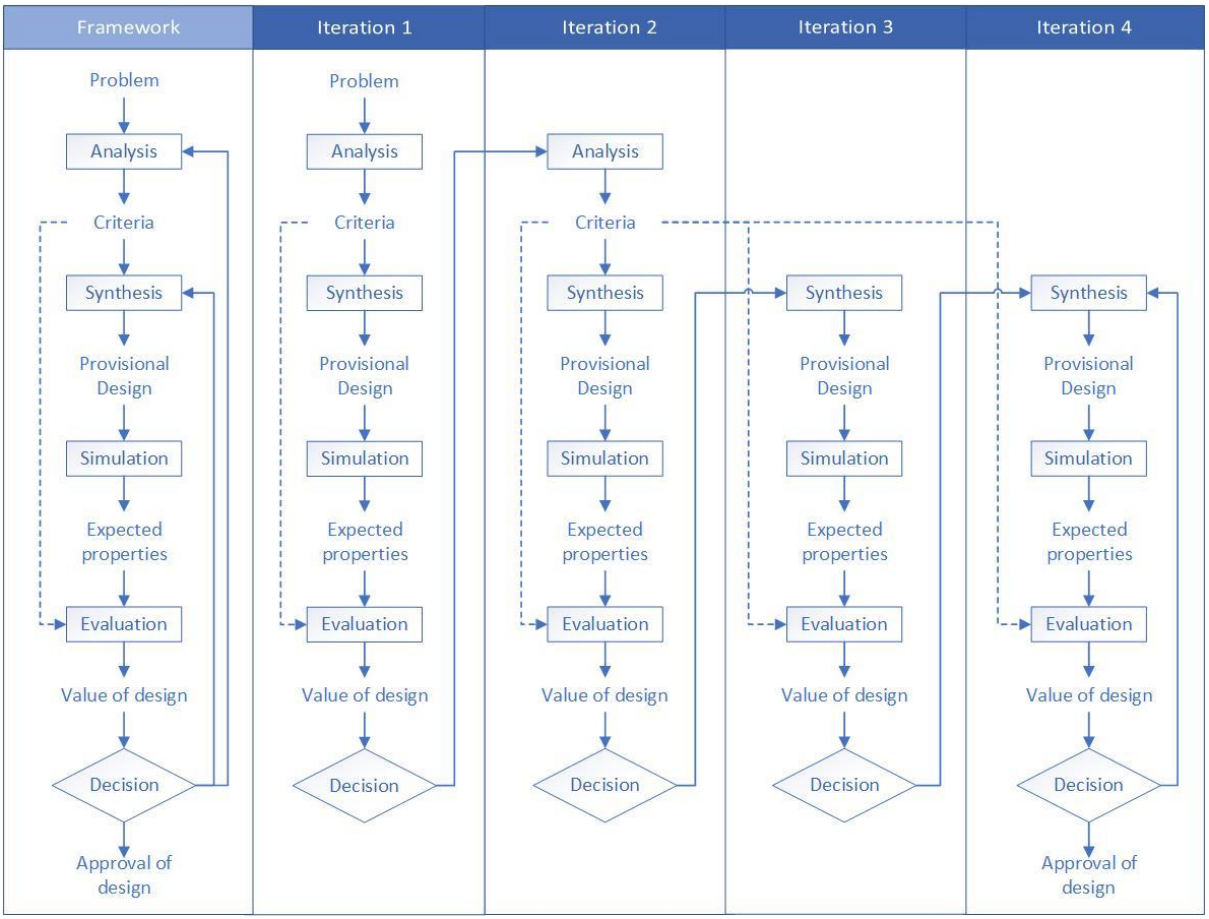

Figure 4: Applied research framework with iterations

5 Information internalisation: 3D BIM and Relatics to OTL

The Problem at hand is that the poor definition of requirements. It is not always clear which object the requirement is referring to. So the poor definition of requirements prevents them from being automatically allocated to their respective objects. In this chapter, the attempt is to identify information that relates to both requirements and corresponding objects. After identifying this information, the next step is to have this information in an internal context. Internalising refers to having the information in an environment where we can establish rules to analyse the data. In this research, any format that is readable in Python is an internal environment.

The data from a viaduct construction project will be used as the testing ground to perform this research.

5.1 Relatics and BIM data

The problem in hand is twofold, the requirements are in a database in the Relatics environment, and the objects are in the BIM environment. Let us look at internalising the requirements and then the objects.

5.1.1 Importance of internalising requirements in the Relatics workspace

Most of the requirements of a given project in the Netherlands is stored in Relatics. The Relatics environment in design is like that of a relational database. The database like environment means that all the information stored in Relatics is connected through relations. The critical factor that differentiates Relatics from a regular relational database is that the various data silos are connected with relations to each other with a relation. All the information regarding a given project that uses Relatics is accessible via a Relatics Workspace. The Relatics Workspace is representative of a Relatics Type Design. A typical Type Design in Relatics consists of a graphical representation of the various types of information that needs to be stored. Relations connect each of the different types of information. Appendix 5: Typical Relatics Type Design contains a Type Design for the reconstruction of a bridge.

5.1.2 Why internalising object data from the BIM environment?

The main reason BIM is required for this research is that the BIM models of the projects have a set of objects that were modelled. We need to extract these objects and lists that form an object hierarchy or systems breakdown structure (SBS). This SBS would be used to categorise the requirements. The BIM environment is also host to a vast library of material-specific detailing that would otherwise only be in manually drafted drawings.

5.1.3 How to internalise requirements in the Relatics workspace?

Relatics allows a user to download any data that is stored in it in excel format. However, this requires someone to download each section of the workspace manually. It is worth noting that a list of relations can also is downloadable in the same manner.

Relatics also offers web services and has a wide variety of API's that can be used to access any workspace. There is also a Python package called PyRelatics[3] that uses these web services to interact with a workspace. It simplifies the process of writing large amounts of XML code which is required to use the API's.

PyRelatics makes it possible to quickly internalise the requirements, the additional Relatics data and their relations.

5.1.4 How to internalise objects from BIM models?

As discussed in the problem definition, chapter IFC is an easily readable format. If there are BIM Models than they can be exported to an IFC format, the information in IFC can be internalised using a Python package IFCOpenShell[4] developed by TNO. IFCOpenShell can open any IFC file to extract and create any object in the file. With this tool, we can extract all the objects that exist in the IFC file along with the details such as their material, properties and any other information that is associated with an object.

5.2 Internalisation execution

The desired information was not generated after the execution of the above procedure, as some of the steps were not possible to perform while others were poor sources of information. Below are the expected properties for the Relatics and BIM data.

5.2.1 Results from Relatics data

The Relatics web services changed its authentication procedure for their web services from a username and password system to OAuth 2.0[5] , which is standard practice for web services. This meant that PyRelatics could no longer be used to get data from a Relatics workspace.

Only the manual download in the form of an excel file was possible and downloading an entire Relatics workspace was not a practical proposition.

5.2.2 Results from the BIM environment

In the IFC files that were available for a chosen project, the level of detail on the modelling was enough to identify which objects they were. However, the way the information about each object was stored was not following the standard protocol. The information such as material, dimensions and everything else was concatenated along with the name of the object. This means that all the properties of the object were not separated into specific sections but where all in one column.

The poor detailing practices of the BIM models meant that the list of objects could be extracted, but the other details that should be in a proper BIM model are either missing or poorly formatted.

[3] https://pythonhosted.org/pyrelatics/manual.html
[4] http://ifcopenshell.org/python.html
[5] https://oauth.net/2/

5.3 Conclusion: Choices and changes to information internalisation.

Internalising the information from the above sources was not possible but this information was still available. It was decided that the requirements from a project would be downloaded as an excel file. Data from the IFC files cannot be used as the BIM data was not well modelled to extract any information that would contribute towards the development of an SBS. An SBS is required as the objective is to allocate requirements to objects in an SBS. The OTL developed by RWS offers an extensive collection of objects which can be used as a substitute for objects from a BIM model. Figure 5 shows how the system that will be developed would interact with the requirements, the OTL and the design of objects.

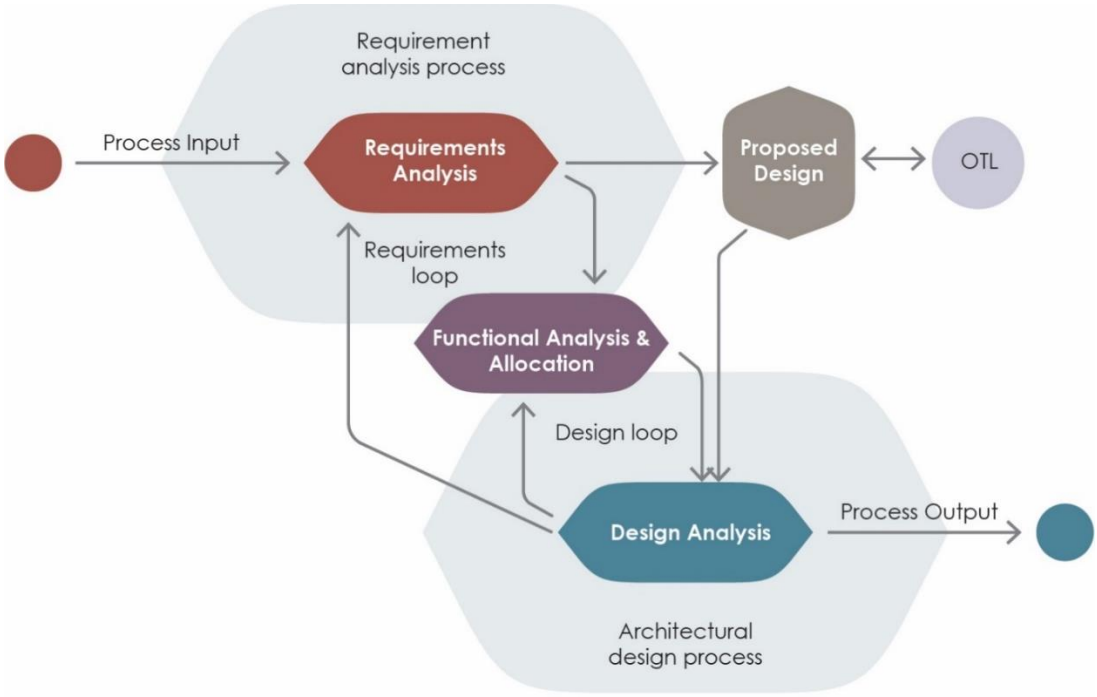

Figure 5:Position of the proposed system in the design process

6 Solution formulation: Processing of internalised information

The proposed solution was identified by using the iterative process described in the research methodology in section 4.5. A set of criteria were established to ensure that the formulated solution is capable of automating the task at hand. A wide variety of solutions were considered throughout the research and examined if they could meet the criteria.

6.1 Introduction to the solution: OTL, Metadata and NLP in construction

There is no longer a problem of internalising the data, but this also means that that we have less information to work with to identify which requirement belongs to which object. The OTL offers its own set of relations into objects and offers a tree diagram to visualise these relations. The tree diagram of the OTL for a viaduct can be seen in Appendix 2: OTL tree diagram of viaducts (Dutch). For a given object like a viaduct, the OTL has a set of standard objects that constitute a typical viaduct. These set of typical objects is what we are interested in this iteration. Each object in the OTL has metadata which stores a description of that object. Sometimes the metadata can also contain some specifications, safety standards, typical images and details of how to construct that object. The metadata offers a vast amount of data, but its availability for all objects is questionable.

Most of the information in the OTL and the requirements from Relatics is in the Dutch language. The language of the data is an essential factor to keep in mind as an NLP library dependent on the language used.

6.1.1 NLP in the construction sector

This research is not first attempting to use National Language Processing (NLP) to capture the information in documents from the construction sector. As mentioned earlier in the context chapter, the field of automatic checking of regulations is very popular. The regulations documents are fairly structured and provide a large pool information to derive rules to extract the semantics. NLP and artificial intelligence have shown promising results in this process (Ghannad, Lee, Dimyadi, & Solihin, 2019). NLP has also been used for extracting clauses from contracts and perform a primitive contractual risk review (Lee, Solihin, & Eastman, 2019). Developing mechanisms to convert regulatory information is also a field of research in itself and has seen great success in preliminary testing (Zhang & El-Gohary Nora, 2015). In this iteration, the attempt is to try using the technology used in the above research.

6.1.2 Definition of criteria for a proposed system

The objective is to use natural language processing to determine if a given requirement belongs to a specific object. The following can be possible criteria to check if the system is functional or not.

- The requirements and OTL objects along with the metadata can be formulated to be a suitable input for an NLP system.
- The NLP system can capture the information and understand the semantics.
- At least some of the requirements can be categorised to the correct object.
- The task performed should be transferable to objects other than viaducts.

6.2 Designing the system

Several alternatives were explored to meet the various criteria. The entire process has been visualised in Figure 6. This figure includes the process of internalisation of the information to the final solution which is marked in green. The various pitfalls are marked in with an X on top. The figure can be divided into five sections. The first is technology; this forms the underlying knowledge base for all the solutions that were explored. The second is a data source which is usually an instance of the technology which is being used as a source of information. The third is the internal environment where the information manifests itself in various forms. The information in the internal environment is a subset of the information in the data source. The penultimate step is the processing or operations that are performed in the various data sets that have been internalised. The last stage is evaluating the output against the criteria that were set up in section 6.1.2. Each component of the proposed system is explained in the subsequent chapters.

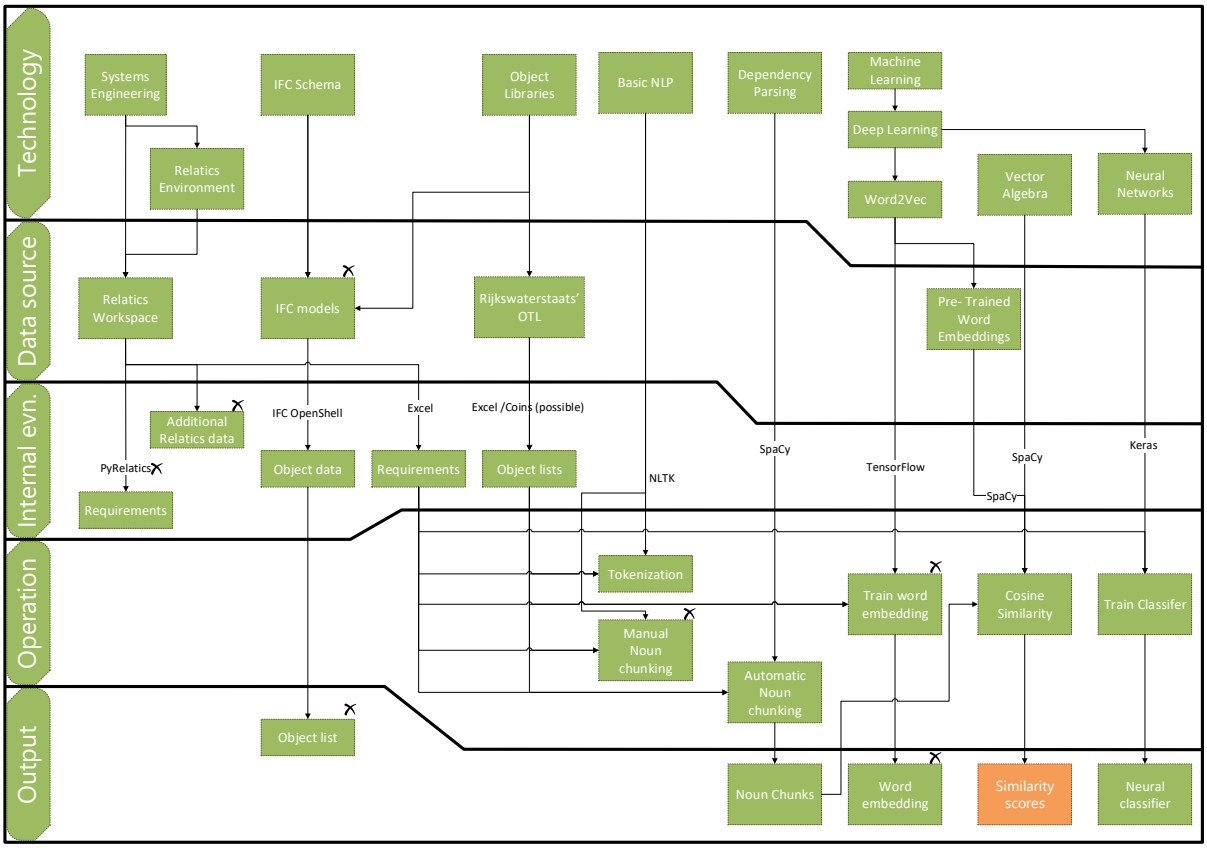

Figure 6: Solution system identification process

7 Proposed System: Dependency parsed requirements and objects compared in a vector space

The two main components of the proposed system are the dependency parser and the word embeddings. The dependency parser modifies the requirements and object descriptions from the OTL into meaning chunks. The second is the similarity measurement using word embeddings. In order to understand the working of the proposed system, it is helpful to have some background knowledge of NLP and machine learning. Other NLP operations are explained in Appendix 9: NLP in Python using NLTK with example.

7.1 Dependency parsing in Python

Natural language processing operation requires the incoming text to be in a specific format that is interpretable by the NLP package that is being used. Dependency parsing is the process of developing a general grammatical structure for a given sentence(Choi, Tetreault, & Stent, 2015). Dependency parsing is similar to writing Chunk Grams[6] (explained in detail in Appendix 9: NLP in Python using NLTK with example) using regular expression[7] and POS tags[8], but the advantage is that a parser does the tagging and classification automatically. Dependency parsing led to the complete elimination of manually written chunk grams. Which means it is possible to have different semantic structures, Figure 7 shows the relations that are developed by a dependency parser.

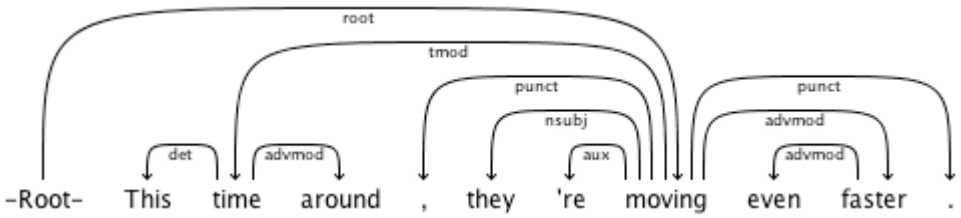

Figure 7: Visualization of dependency parsing in a sentence

Dependency parsing has been available for a long time, but recent development in the field of deep learning and NLP have led some breakthroughs (Choi et al., 2015). There are several dependency parsers in the market, but most of them are stand-alone solutions, whereas SpaCy is a complete Python package with all other NLP functions built into it. SpaCy has proper documentation and an interactive website to learn how to use it. SpaCy has support for the Dutch language as well.

[6] A chunk gram is set of rules formulated using a the regular expressions and POS tags.
[7] Regular expressions represent a pattern of data and in our case words that would appear in a sentence that is required to be identified. All the regular expressions in python are listed in
[8] POS tags refer to parts of speech tags such as noun, pro noun, verb, adjective and adverb etc.

7.1.1 Noun chunking the requirements and object description.

As a feature of dependency parsing, SpaCy lets the user Chunk tokens based on the presence of nouns. The noun chunker is easy to use and can extract the meaningful phrases of requirement or object. The noun chunker uses a pre-loaded NLP pipeline[9] , so there is no need for a tokenisation or any pre-processing of the data. The noun chunker performs implicitly typecasts[10] to a 'Span' which is datatype of SpaCy, so all other Spacy functions are callable.

It was possible to tokenise and tag the data with parts of speech but chunking the data into meaningful bits requires an extensive understanding of the grammatical rules of both English and Dutch. This means that the first criterion from the list of criteria in section 6.1.2 is met.

7.2 Similarity with word embeddings

In order to measure the similarity between the requirements and the object descriptions in a manner which does not use a database of synonyms requires the numerical quantification of the semantics.

7.2.1 Why words in vector spaces?

The reason why it makes sense to represent words in a vector space is that words tend to have different meanings, and they are used in different contexts. In a multidimensional vector space, words can be close to different words in many different directions, which ensures a relation with all associated words. This can be explained through an example. Take 'Nokia' and 'Samsung' as two words in a two-dimensional cartesian space. Both the words would be close together as it is very likely they were both used in a similar context. However, one would associate the word 'Nokia' to be close to the word 'Finland' for a totally different reason for Nokia being a Finnish company. Following that line of thought, in a two-dimensional space, the words 'Samsung' and 'Finland would inevitably be associated with each other, resulting in a false relation. In a multidimensional vector space, a wide range of relations can be maintained without associating incorrect relations.

7.2.2 Word vectorisation: n-gram or bag-of-words

Textual information can be understood either as a sequence of letters or sequence of words. There are no patterns in the way individual characters of requirements and object descriptions are formulated. So the design can be limited to word vectorisations rather than character vectorisation. (Chollet, 2018)

A rather primitive way of vectorisation is to use an n-gram or a bag-of-words. In an n-gram, 'n' number of words from a sentence are grouped. Here is a 2-gram for a simple sentence.

Sentence: 'the man sat on a bench.'

2 gram: 'the', 'the man', 'man', 'man sat', 'sat', 'sat on', 'on', 'on a', 'a', 'a bench', 'bench.'

An n-gram loses the order in which the words occur in a sentence and instead puts all the grams into one "bag" and hence the name "bag-of-words". The use of n-grams is primarily for

[9] An NLP pipeline is the system of tokenizing and POS tagging to prepare any data for analysis.
[10] Typecasting is the process of changing the datatype of any given data.

feature[11] engineering[12]. On the other hand, deep learning does not require such a rigid set of features as it uses a hierarchical feature learning. Deep learning is capable of learning such patterns without being explicitly asked (Chollet, 2018). This system is not suitable for our application as it is necessary to maintain the structure of the Noun Chunks from the previous iteration. A vectorisation system that is compatible and takes advantage of deep learning algorithms can prove to be useful.

7.2.3 Vectorisation with one-hot encoding

One-hot encoding is one of the most common vectorisations technique that is available. One-hot encoding works by assigning a unique integer to each word and then converting it to a binary vector whose size is the same as the number of words. Table 1 shows this process for a simple sentence where each column of numbers below a word is the one-hot vector for that word. One-hot vectors are as long as the size of the vocabulary, so they usually tend to be in the ballpark range of twenty thousand dimensions or more. Most of the vector is just 0s, so this generates a lot of redundant data. Although one-hot vectors are numeric representations of words that are technically suitable for deep learning algorithms, they are computationally expensive.

Problems	Turning	Into	Banking	crises	as
1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1

Table 1:One-hot encoding

7.2.4 Vectorisation with Word2vec

Word2Vec models are commonly used to reduce the number of dimensions in a one-hot embedding (Chollet, 2018). The word2vec model uses a neural network single hidden layer to perform for an unsupervised feature learning. The word2vec model does not perform the task of allocating requirements to objects. The goal of the model is to learn the patterns in which words in the training data repeat. Learning the patterns givens, the word vectors that capture the semantic information. The model trains itself to find the probability of a word in a sentence given certain words appeared in its context (Mikolov, Chen, Corrado, & Dean, 2013)

For example, if the trained model is given the word ‘Dutch’, it would give a higher probability to words like ‘Netherlands’ and ‘language’ than random words like ‘window’ or ‘paper’. In the next section, let us look at how this is done.

[11] A feature is any one aspect for any given data that points to the characteristics of that data.
[12] Feature engineering is the process of identifying and separating the specific data point from the data

7.2.5 Neural networks for training word embeddings

A Gradient Descent[13] is used to train the word vector over multiple iterations. The values of the word vector are altered gradually. The words which are used in a similar context are seen to move gradually towards each other. This is based on the number of times they appear in the same context. Although a large amount of data is not needed, it is vital that the data is repetitive. The same object needs to appear in different contexts multiple times. Generally, the learning rate is low (to ensure that the model is not overtrained). So when a word appears only once, it does not train the word vector in any manner. To be able to cover the vast vocabulary that is present in our data set of object descriptions and requirements, it is required that we have data from at least fifty to sixty viaducts. This is required in order to train a consistent word vector. This was verified by training a word embedding using the requirements of four projects, and a two-dimensional compressed visualisation is Figure 8.

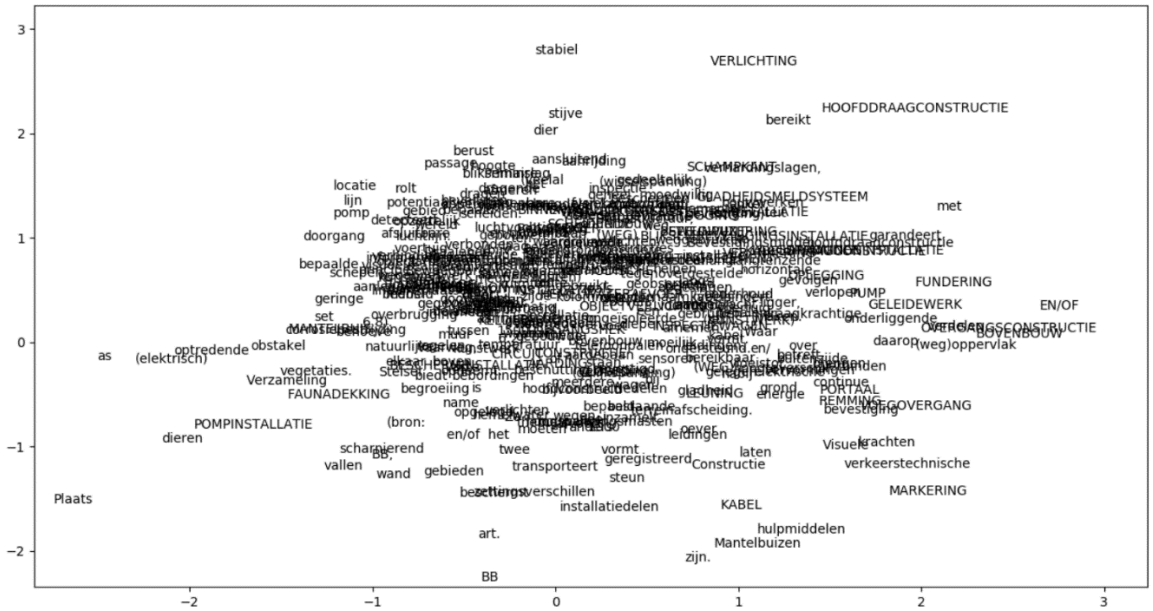

Figure 8:Word embedding generated from requirements and objects in Dutch

These observations led to the conclusion that with the data at hand, it is not possible to train a new word embedding. The underlying reason is that most of the words only appeared once. It was still possible to use word embeddings in this research as there are several pre-trained word embeddings available that can be used.

7.2.6 Pre-trained word embeddings

When there is not sufficient data to train a word embedding, a solution can be to use a pre-trained word embedding. A pre-trained word embedding is in fact what the name suggests; it is a word embedding with word vectors which have already been trained, generally on extensive data sets. For example, GloVe, an unsupervised algorithm to produce word vector representations, is trained on 840 billion tokens and has a vocabulary of 2.2 million different

[13] Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function.

words(Pennington, Socher, & Manning, 2014). Another popular word embedding is fastText, and there are also several word embeddings within SpaCy, which can also be used.

Dutch word embeddings

There are several word embeddings in the Dutch language. The Dutch word embedding was developed as a linguistic resource to compare unsupervised methods with regular dictionaries (Tulkens, Emmery, & Daelemans, 2016).

7.2.7 Semantic similarity using cosine similarity

In the K nearest neighbour method, the distance between the various data points is measured in order to find the closest category (which is used to classify the data point). In a vector space, this distance is called the Euclidian distance. An alternative to finding similarity is to find the cosine of the dot product of the two vectors whose similarity is in question(S. Perone, 2013). This calculation is shown in Figure 9

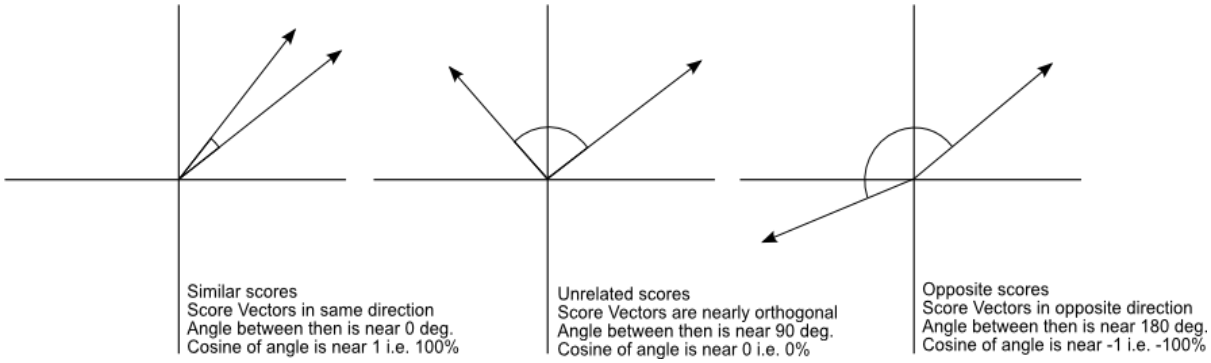

Figure 9: The Cosine Similarity values for different documents, 1 (same direction), 0 (90 deg.), -1 (opposite directions) (S. Perone, 2013).

7.2.8 Word embeddings in SpaCy

SpaCy is the same Python package that was used for Noun Chunking. SpaCy also has its own set of word embeddings, with vectors of up to three hundred dimensions. The SpaCy documentation has instructions to convert any word embedding in any format into a SpaCy compatible word embedding. Internalising the external word embeddings means that it is possible to import the Dutch word embeddings to check for similarity.

7.3 Final System

Looking back at all the criteria defined in section 6.1.2. The first three criteria are achieved as the data has been converted into a format that is suitable for analysis. Pre-trained word embeddings already have the semantics of the language so when the noun chunks can be compared using cosine similarity. This system has been visualised in Figure 10

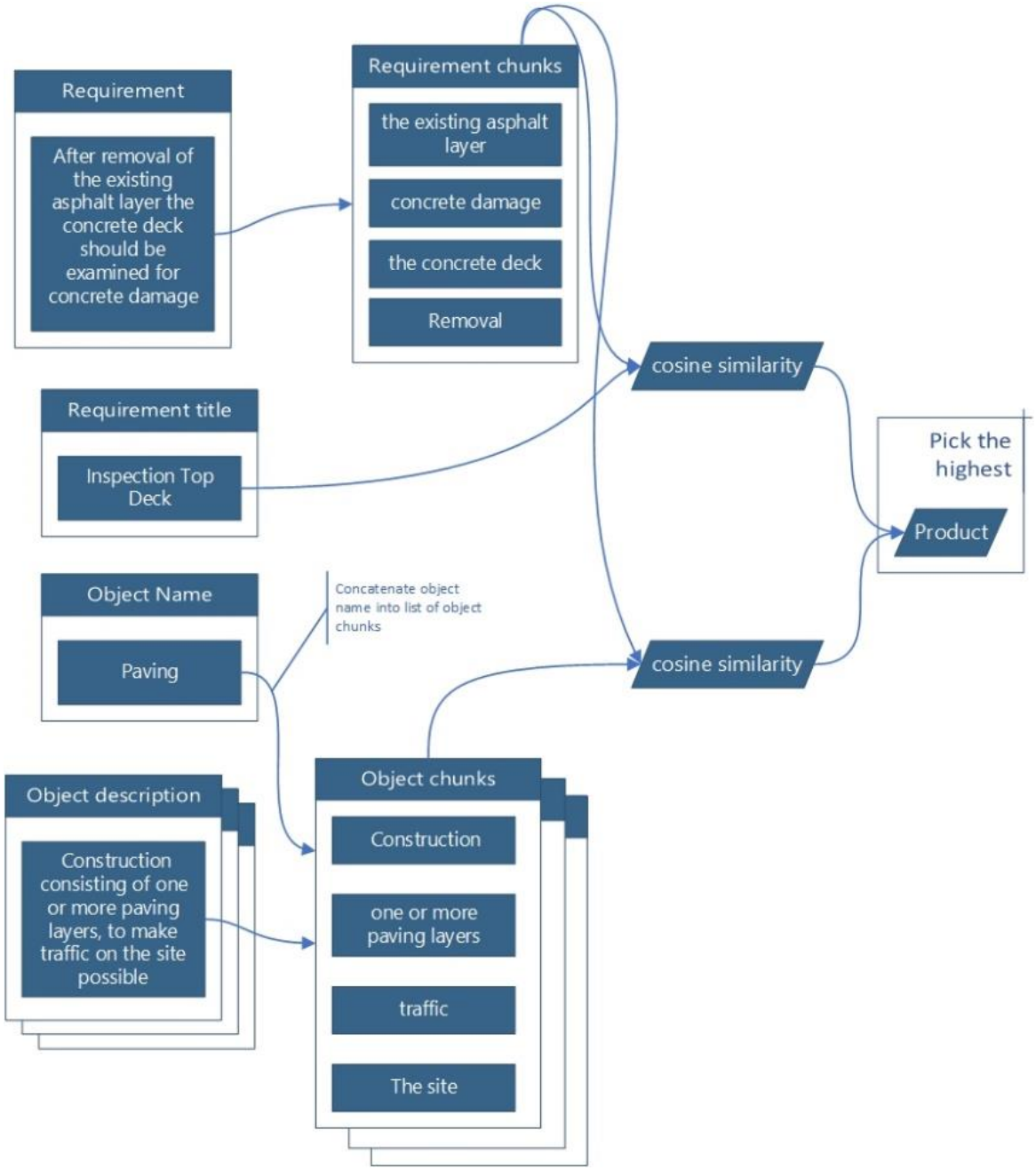

Figure 10: Proposed system

In the system proposed dependency parser of SpaCy to identify noun chunks; this is a syntactic process where nouns are identified and concatenated into one phrase. Each requirement has a title; the system finds the similarity between the title and various chunks of the requirement. The chunks with higher similarity to the title would have higher importance when the chunk compared to a chunk from the object description. Each chunk from the requirements and the object descriptions is checked for cosine similarity. The requirement will be allocated to the object chunk that has the highest similarity with a requirement chunk.

7.4 Results: Similarity score

The proposed system is capable of producing similarities between the chunks requirements and object descriptions. Theses scores are in descending order to to find the chunk pairs that

have the highest similarity. A sample result is shown in Table 2: Sample results of allocation of a requirement to objects for the requirement 'The utility must be provided with an anti-graffiti coating on all visible surfaces of Concrete Steel -Masonry; -Plastic/glass.'

OTL object	OTL words	Requirement words	Similarity
Anti-vandalism provision	Anti-vandalism provision	an anti-graffiti coating	0.648428
Protective coating	Protective coating	an anti-graffiti coating	0.594731
MAIN WEAR CONSTRUCTION BRIDGING	a Primary load-bearing constructi	an anti-graffiti coating	0.55714
Paving	one or more paving layers	an anti-graffiti coating	0.545949
Impact bar	A beam-shaped collision protecto	an anti-graffiti coating	0.545877
Access gate	a surface	an anti-graffiti coating	0.544227
Capital	a bearing surface	an anti-graffiti coating	0.543904
Beam	a Load-bearing horizontal structu	an anti-graffiti coating	0.533011
Pier	a free-standing support	an anti-graffiti coating	0.529684
Pylon (road)	A construction element	an anti-graffiti coating	0.519233

Table 2: Sample results of allocation of a requirement to objects

7.4.1 Visualisation of the design in a graphical user interface

The entire process of inputting a requirement, processing and the output happen in Python this might be difficult for users to see, use and comprehend. For this reason, a graphical user interface was designed to interact with the code. It can be seen in Figure 11:GUI for the allocation system

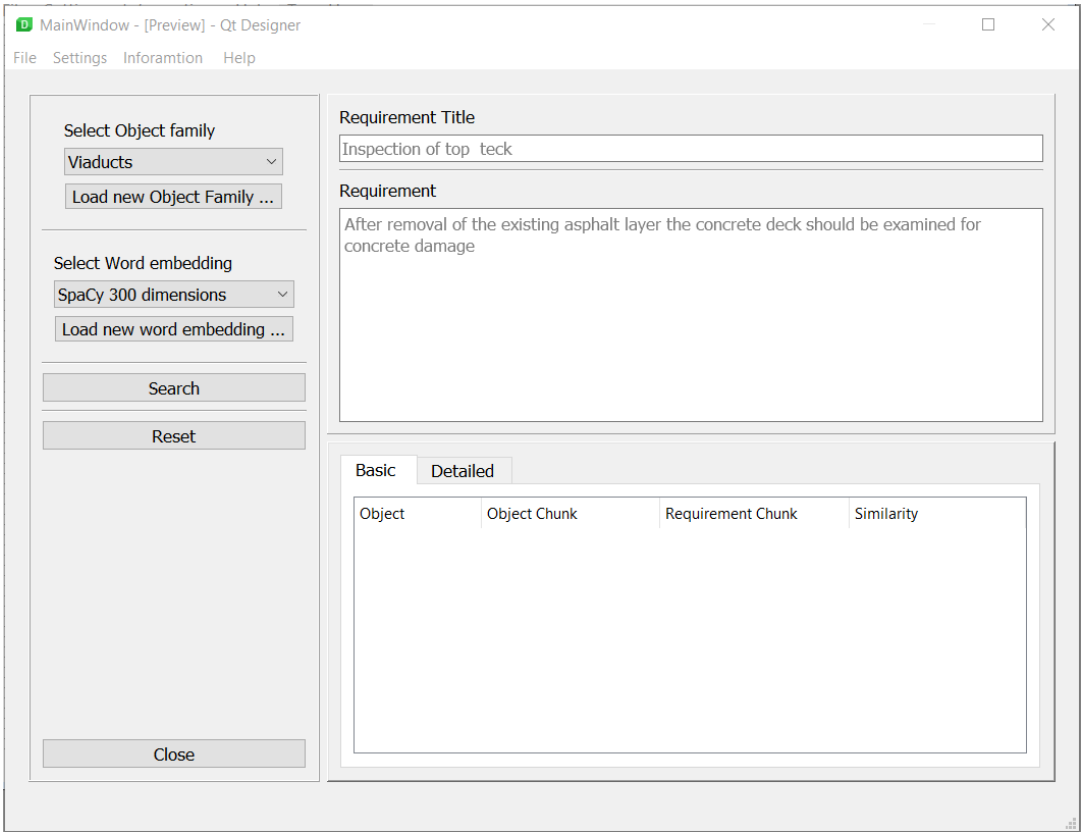

Figure 11:GUI for the allocation system

8 Alternatives: Neural classifier and more

A wide variety of solutions were considered throughout the research. Each one has its drawbacks; the first solution proposed to use NLTK analysing the requirements and object descriptions. The NLTK system used legacy functions that require extensive linguistic knowledge before any meaningful results can be achieved. The SpaCy package has inbuilt dependency parsers which would automatically capture the syntax of a sentence but a parser incapable of capturing the semantics.

The word2vec model had shown promising results in understanding the semantic relationship between text, the output of the word2vec model is a word embedding containing words in a multi-dimensional vector space. A word embedding contains words of similar meaning close to each other and vice-versa. Training a word embedding requires data in which words repeat multiple times in different contexts, but the data from the requirements and objects are unique, so it was not possible to generate a consistent word embedding.

It is still possible to use word embeddings as there is a vast array of pre-trained word embeddings available. The inbuilt word embeddings in SpaCy were used to check plugin noun chunks from requirements and objects to measure the angle between them and find their cosine similarity. This similarity is a good indication of which requirement belongs to which objects as was explained in the previous chapter.

In the last iteration, a neural classifier was built to classify the requirements into object categories. This system has been explained in detail as it is the only system that could theoretically satisfy all the criteria explained in section 6.1.2. A comparison of the various solutions is presented in Table 3 and showcases where they fail (red) or succeed (green) or can succeed with more data (yellow).

Features	Syntactic NLP using NLTK	Synthetic NLP for phrase matching	Semantic NLP with machine learning	Semantic NLP using pre-trained word embedding	Neural classifier
Data Internalisation					
Capture syntax					
Capture semantics					
categorise requirements to objects					
Transfer to other object catagories					
Discrete output					

Table 3: Comparison of the technologies considered

8.1 Requirement classification to objects with a classifier

The first step in the design synthesis would be to find the various types of classifiers available, what inputs they can accept only then can the choice of a classifier be made. Next is choosing a suitable Python package for that classifier.

8.1.1 Classifiers

A classifier is a rendering of supervised learning from machine learning; in fact, K nearest neighbour and SVM are also classifiers. The other classifiers that were considered are decision trees and neural classifiers. Experts recommended that when dealing with text input for classifiers, it is better to have a neural architecture as others have not shown promising results in the past.

8.1.2 Neural classifier for text classification using Keras

As was seen in iteration four, the data still needs to be given to the classifier in a format that is acceptable for the algorithm. The vectorise the data is an essential factor for classification, but this can now be done using a pre-trained word embedding. The word embedding functions as the first layers of a neural network. Another critical operation is to tokenise the entire input. Both of these operations, including the definition of the hidden layers, can be done using a python package called Keras. Keras is a high-level neural network package for Python. A noteworthy point here is that labelled data is required to train the neural classifier, and we have a large number of requirements with their allotted objects.

8.1.3 Provisional design

- Format the requirements data into a long python list.
- Prepare a list of the objects, number them and change the labels of the requirements to numbers.
- Sperate the data into training and testing data.
- Tokenise the requirements while still maintaining a link to their respective labels.
- Define the shape of the input layer as the length of the longest requirement.
- Add an embedding layer and chose any pre-trained word embedding to vectorise the tokens.
- Define a set of hidden layers
- Train the model

8.2 Simulation

In order to get the data from excel to a format, the tokeniser would accept was challenging for a novice programmer. During the second step, it becomes evident that the object lists like the one in appendix 8 do not retain any syntactic or semantic information. The objects get replaced with numbers. Keras has inbuilt tools to separate the data, tokenise and freeze the word embedding as the first layer of the neural network. When the training began, the results were quite poor as 700 requirements are not sufficient to train a neural classifier. This could be overcome by simulating a positive trend if there was more data. The percentage of training data would be increased, and the training would begin again, and if the results are better than the original split, it is possible to say that more data would solve the problem. However, other inherent characteristics of a neural classifier made them unsuitable for classifying requirements to objects.

8.2.1 Expected properties

The neural classifier can be expected to categorise to the given set of objects but if another object family (Apart from viaducts) the entire model would have to be retrained with requirements from that object. Other drawbacks which are not related to the output but the chartists of a neural network will be explained in the evaluation.

8.3 Evaluation

The neural classifier was able to achieve the additional criteria of providing a deterministic output as there is a SoftMax activation function at the end of the training, but the neural classifier did not able to achieve the initial criteria defined in iteration two.

As mentioned in the expected properties, the neural classifier fails to capture any information from object description. This means that the OTL object failed to give any input to the system and hence, the first and second criteria are not satisfied.

The third criterion is partially satisfied but as it is possible to simulate the categorisation, but it is a far cry from a functional system because of the last criterion which speaks about transferability of the system where the neural classifier fails because the system needs to be separately trained for each type of asset. , renders neural classifiers completely untransferable.

8.3.1 Value of design

When we look at the applicability of neural classifier in actual practice another significant flow is evident. The number of objects in the system is fixed, and if a new object is added the model is no longer applicable and needs to be retrained. During the design phase of a project, the number of objects that are part of the project can change on a daily bases which means that the model needs to be trained every day on new requirements to be able to cope with the project dynamics. Training the data every day could result in a situation where the model becomes overtrained. This limitation significantly reduced the value of the design.

8.4 Decision

After careful consideration of the above design, it seemed less likely that a neural classifier would be able to classify the requirements to objects for projects other than viaducts or even a different viaduct project. The rigid structure of the output layer and the need for extensive amounts of labelled data make them difficult to use and adapt to other projects.

The use of word embeddings as the first layer in a neural classifier seemed like a logical progression to the classification of requirements into objects, but one key factor was overlooked; by tokenising the data before entering the neural network the minimal syntactic relationships that exist in the requirements is lost because there is no operation like noun chunking. It would be possible to provide a set of noun chunks as features to the neural network which might help in capture some syntax, but all of this would be in vain as almost half of the semantics is lost when the object descriptions are not used at all. For this reason, it makes no sense to further develop on a neural classifier for the problem of allocation of requirements to objects.

9 Verification: Did we build the system, right?

In the spirit of Systems Engineering, verification and validation were carried out to measure the efficiency and effectiveness of the proposed system. In the verification process, the attempt is to check 'if we built the system right?' while the validation process refers to 'did we build the right system?'. This process is handled in the two next two chapters.

The first chapter will first present a quantitative analysis of the results from the proposed system which is accompanied by an analysis of how to interpret the results, in which scenarios the system works and when it does not. This is followed by a qualitative validation to check if the proposed tool is usable in the industry and to what extent is it usable.

The verification process involves the measurement of metrics to determine the accuracy of the proposed system. The proposed system has been evaluated as an information retrieval system. The metrics that are generally measured for an information retrieval system are Precision, Recall, F-score and accuracy. Amongst these Precision and recall have been calculated in this research, F-Score is a derivative of Precision and Recall so it can be calculated later if required(Baeza-Yates & Ribeiro-Neto, 1999). The measurement of accuracy requires the measurement of both true negatives and true positives, but the proposed system only measures true positives; therefore, it is not possible to measure accuracy as a metric. The calculation of Precision and Recall has been explained in Figure 12.

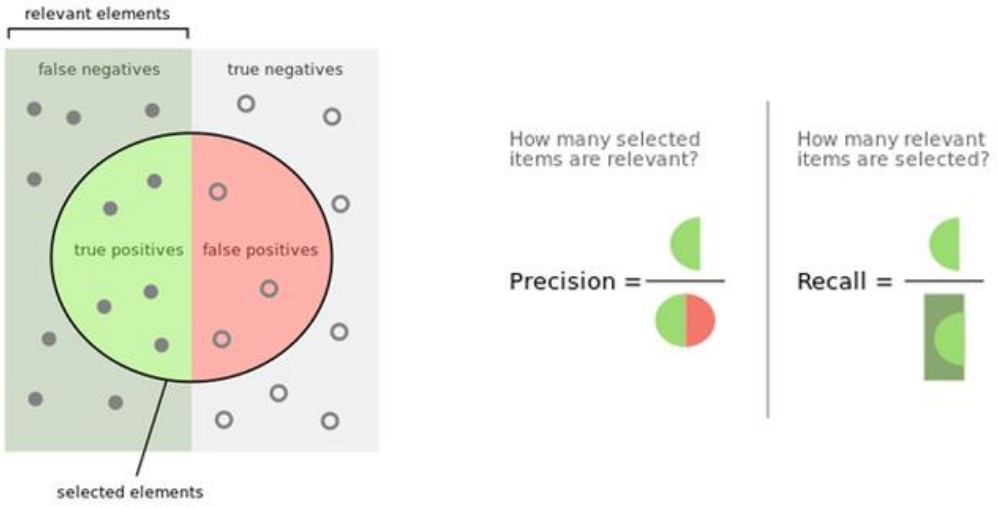

Figure 12: precision and recall calculations (Walber (Own work) [CC-BY-SA-4.0], via Wikimedia Commons)

9.1 Methodology: Procedure for validation

The measurement of Precision and Recall was carried out for 20 requirements from a catalogue of requirements from a viaduct renovation project. The requirements were defined in Dutch, so they had to be translated into English, which was done using Microsoft Translate Services. Of all the requirements a set of 20 requirements were selected based on the quality of translation. These requirements were manually checked against each of the OTL objects for a viaduct, and a perfect answer set was prepared. This set of objects will be the answer key against which the proposed systems will be evaluated. The top ten results from the proposed system are collected, duplicated are removed, and the right answers marked. The below equations are used to calculate the Precision and Recall for each requirement and a graph of

precision versus recall is plotted. These plots are unique for each requirement, so an averaged precision at standard values of recall is calculated using equation number 3. The results are analysed to and presented in the form of a reflection and comments on the observations made during the process of verification.

Where

9.2 Results

The results are shown in two sections, the first is the individual values of precision and recall for a few characteristic patterns of precision vs recall. Second, average precision and recall are calculated and plotted.

9.2.1 Possible outcomes of precision and recall

Based on the performance of the proposed system for different requirement a few characteristic patterns of precision vs recall were observed. Below are a set of tables each corresponding to requirement at which is mentioned at the top of each table. These are followed by a graph for their precision vs recall. Three typical result types are shown in the figures and tables below.

Type 1 is an answer set that has a pattern of precision vs recall when there are multiple objects to which a given requirement can belong to, and these objects appear in the search slowly with some wrong object in between. The data and plot are found in Table 4 and Figure 13

Type 2 represents a pattern of precision vs recall when there are multiple correct answers, and all of them are found by the proposed system one after the other. The plot and data are found Figure 14 and Table 5

Type 3 represents a pattern of precision vs recall when there is only one correct object to which the requirement can belong to and if it is found in the first position. There were other instances where the answer is not found in the first recall, but they also produce a similar precision vs recall graph, but they start with a lower level of precision. The data and plot are found Table 6 Figure 15

All other patterns of precision vs recall are combinations of these three types of results. The precision and recall graphs of twenty requirements are in the appendix.

The tensioning of the longitudinal pre-tension must be done from two sides. The Cros pretension can be stretched from one side.					Correct answers	VM15,VM26, V12, VM4	
		OTL object	OTL words	Requirement words	Similarity	Recall	final sim
1	VM7	Arch	a Tension structure	the longitudinal pre-tension	0.534063264	0	0
2	VM4	Beam	a Load-bearing horizonta structural element	the longitudinal pre-tension	0.521650774	25	50
3	V34	Supporting point	the main load-bearing structure	the longitudinal pre-tension	0.518794473	25	33.33333
4	VM26	Plate field	the main load-bearing structure	the longitudinal pre-tension	0.518794473	50	50
5	V12	MAIN WEAR CONSTRUCTION BRIDGING	a Primary load-bearing construction	the longitudinal pre-tension	0.502170472	75	60
6	VM15	Cross beam (structure)	the longitudinal beams	the longitudinal pre-tension	0.492683823	100	66.66667
7	VM30	Guy line	tension forces	the longitudinal pre-tension	0.466119981	100	57.14286
8	VM1	Anchor plate	a Flat structural element	the longitudinal pre-tension	0.461346478	100	50
9	VM31	Kerb	a Low-height construction	the longitudinal pre-tension	0.45968152	100	44.44444
10	VM34	Wind brace	the surface	the longitudinal pre-tension	0.458349419	100	40

Table 4:Pattern 1 sample results of the proposed system

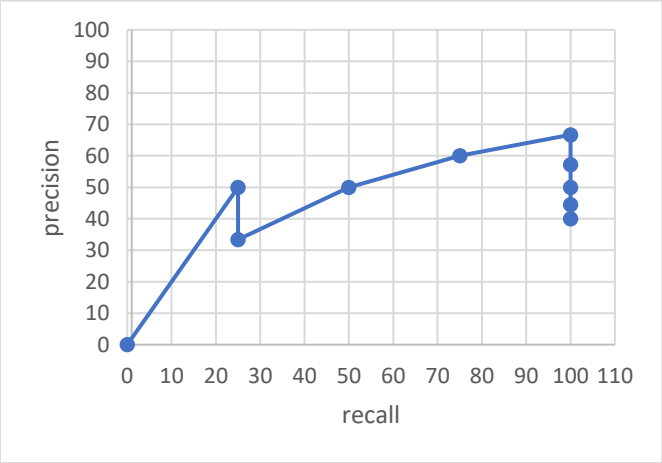

Figure 13:Precision Vs. Recall graph for type 1 results

The utility must be provided with an anti-graffiti coating on all visible surfaces of: Concrete Steel -Masonry; -Plastic/glass.					Correct answers	V11, V21, VM10	
		OTL object	OTL words	Requirement words	Similarity	Recall	final sim
1	V3	Anti-vandalism provision	Anti-vandalism provision	an anti-graffiti coating	0.64842752	50	100
2	VM5	Protective coating	Protective coating	an anti-graffiti coating	0.594731372	100	100
3	V12	MAIN WEAR CONSTRUCTION BRIDGING	a Primary load-bearing construction	an anti-graffiti coating	0.557139785	100	66.66667
4	V6	Paving	one or more paving layers	an anti-graffiti coating	0.54594934	100	50
5	V1	Impact bar	A beam-shaped collision protector	an anti-graffiti coating	0.545876579	100	40
6	V36	Access gate	a surface	an anti-graffiti coating	0.544227239	100	33.33333
7	VM19	Capital	a bearing surface	an anti-graffiti coating	0.543903597	100	28.57143
8	VM4	Beam	a Load-bearing horizontal structural element	an anti-graffiti coating	0.533011215	100	25
9	VM25	Pier	a free-standing support	an anti-graffiti coating	0.529683964	100	22.22222
10	VM34	Pylon (road)	A construction element	an anti-graffiti coating	0.519233155	100	20

Table 5: Pattern 2 sample results of the proposed system

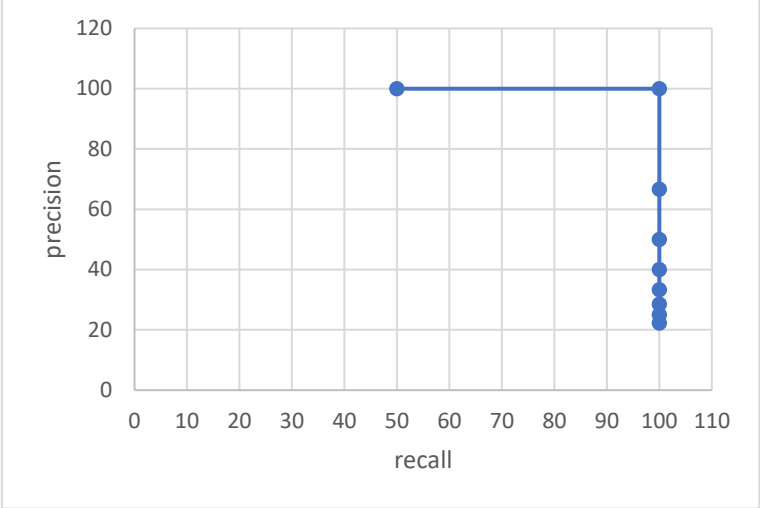

Figure 14:Prescion Vs Recall for type 2 results

Fasteners in concrete must comply with the NVN-CEN/TS 1992-4 range. Retrofitting of fasteners a reinforcing steel (further anchors) in hardened concrete shall be carried out by a holder of a KOMO process certificate based on BRL 0509. Anchors without CE marking must be tested according to following SKIRT provision at NVN-CEN/TS 1992-4-5. Anchors with CE marking must also be tested						Correct answersVM6	
		OTL object	OTL words	Requirement words	Similarity	Recall	final sim
1	VM6	Fastener (construction)	Fastener	Fasteners	0.843209624	100	100
2	V38	Anchoring	a Fastener	Fasteners	0.666636169	100	50
3	VM17	Hanger (arch bridge)	Hanger	Fasteners	0.483775765	100	33.33333
4	VM26	Plate field	steel or concrete plates	Fasteners	0.461905658	100	25
5	VM1	Anchor plate	Anchor plate	Fasteners	0.434666097	100	20
6	VM5	Protective coating	Protective coating	Fasteners	0.41851452	100	16.66667
7	VM1	Anchor plate	the anchor rod	Fasteners	0.418314487	100	14.28571
8	VM29	Groove superstructure	a Steel superstructure	Fasteners	0.411956191	100	12.5
9	VM11	Corbel	a Protruding structural element	Fasteners	0.406427294	100	11.11111
10	V39	Joint transition	seal	Fasteners	0.401991963	100	10

Table 6:Pattern 3 sample results of the proposed system

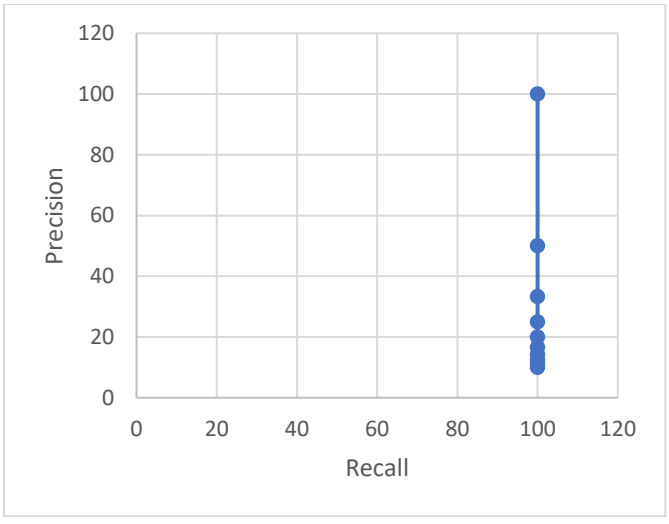

Figure 15: Precision Vs Recall for type 3 results

9.2.2 *Averaged precision vs recall*

The precision for all the requirement at specific recall levels was calculated using equation 3. Three different lines were plotted for different answer sets. The different answer sets are for the top 10 objects the system retrieved, the top 5 objects the system retrieved and the to 3 objects the system retrieved. In situations where there was no data available for a given recall level, it was interpolated from the precision values of the surrounding recall levels(Baeza-Yates & Ribeiro-Neto, 1999). The data and plot are found in Table 10 and Figure 16

Recall	Precision		
	top 10	Top 5	top 3
0	0	0	0
16.666667	29	32.08	41.66
20	33.97	45.6	61.11
25	28.34	35.28	41.66
33.3	35.08	45.6	54.1
40	22.222222	47.4	55.7
50	32.3	49.2	57.3
60	30	66.3	70.35
66.6	34.3	83.4	83.4
75	60	60	72.5
100	34.2	50.4	61.6

Table 7:precision vs recall for various portions of answer set

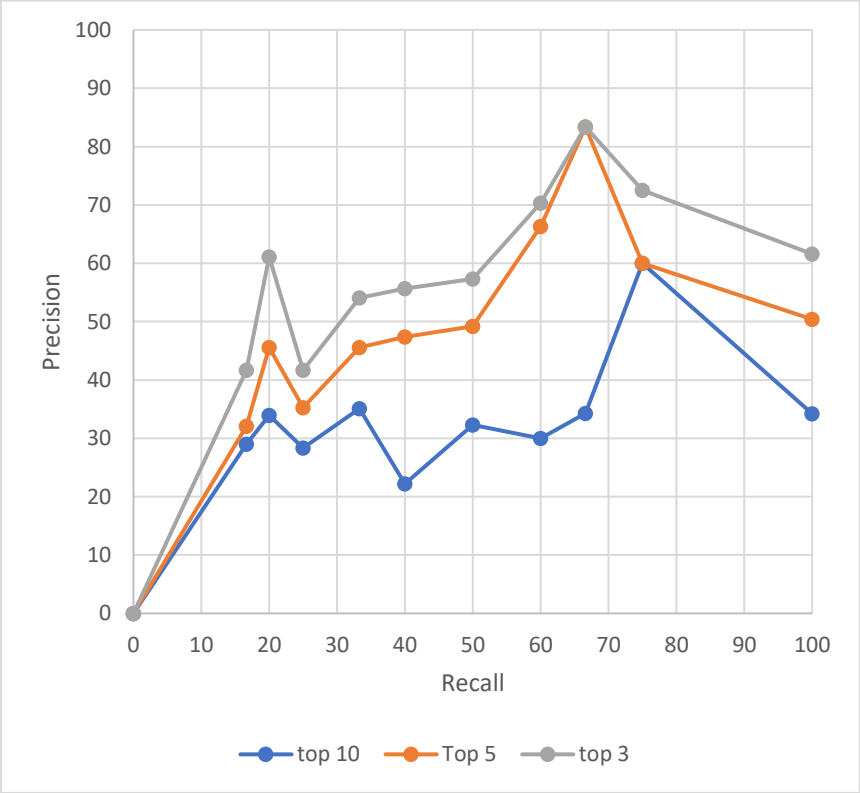

Figure 16:Precision Vs. Recall at different answer sets

9.2.3 Comparison with Phrase Matcher

A phrase matcher is a readily available tool in any programming platform. The phrase matcher returns a Boolean after checking if a given phrase is present in a sentence or not. The same chunks that were used to measure the similarity with the word embedding were used as input for the phrase matcher.

The phrase matcher required a 100% match to give a positive result and the only chunks that were a direct match where chunks like "a construction" and "construction". The exception being the chunk "fastener" from the requirement "Fasteners in concrete must comply with the NVN-CEN/TS 1992-4 range……". All other requirements did not find a match with any object.

If and when there are multiple matches from the phrase matcher, it does not provide any prioritisation of the objects. The output would be an unordered list of objects.

9.3 Observations

The following are some of the observations on the type of objects that were retrieved.

- The system performs sufficiently when the requirements are specific to a single object. In all the twenty requirements, the proposed system was successful on all requirements, which had only one object. The object was not the first object to be found, but it appears in the top 10 objects.
- When there were two or more objects to be identified the system performs satisfactorily if the objects are similar to each other. If the objects are speaking of vastly different objects then the system fails to identify both. This has been explained in the example below.
    - Requirement: Drain pipes outside the work of art must be carried out in HDPE. All the relevant objects (In bold) were identified because the objects were also similar to each other.

OTL object	OTL words	Requirement words	Similarity
Conduit	pipes	Drain pipes	0.567617
Rainwater drainage	a Rainwater drainage	Drain pipes	0.436933
Inspection car	a pipe	Drain pipes	0.436813
Tube	a Hollow cylindrical pipe	Drain pipes	0.393988
Plate field	steel or concrete plates	Drain pipes	0.372272
Jetty	the water	Drain pipes	0.350793
Conduit	HDPE tubes	Drain pipes	0.337807
Pump system	a pump	Drain pipes	0.325031
Guy line	tube	Drain pipes	0.324222

Table 8:Results from the drain pipes requirement

- When the objects are not similar, then the system fails to identify one of them. This is because the system prioritises chunks of the requirement that are similar to the title of the requirements. This is visible in the example below
    - Requirement: The rainwater drainage must be vandalism resistant.
      The relevant objects where rainwater drainage and Anti-vandalism provision but the system only picks the first object because the system protected chunks that were similar to 'drainage system'.

OTL object	OTL words	Requirement words	Similarity
Rainwater drainage	a Rainwater drainage	The rainwater drainag	0.73648
Jetty	the water	The rainwater drainage	0.557719
Signage	the paving	The rainwater drainage	0.489449
Marking	the paving	The rainwater drainage	0.489449
Joint-free transition	the paving	The rainwater drainage	0.489449
Anchor	a Soil mechanical construction ele	The rainwater drainage	0.463332
Joint transition	the road surface	The rainwater drainage	0.460394
Transition construction (road)	the embankment	The rainwater drainage	0.447499
Wind brace	the surface	The rainwater drainage	0.441557
Foundation (structure)	the underlying ground	The rainwater drainage	0.423277

Table 9: Results from the drainage and vandalism resistance

- The system has a poor performance if the requirements are a top-level requirement which applies to the entire project. This is mostly because the system was not designed for the top-level requirement. For example
    - Requirement: Viaducts and bridges must be free from fencing. Fall protection should be incorporated into the brick parapet.

      This requirement is applicable for the entire viaduct because the closest match is an object named 'railing' which did appear in the retrieved set, but the industry experts explained that there would never be an object 'railing' because the requirement itself specifies that there should not be any fencing.

The following are observations on the Precision and Recall levels

- In most scenarios (requirement and object combinations), the majority of the results appear early on if they do appear. So the last few results usually are wrong and hence reduce the precision. This explains why the precision for the top 5 is far better than the top 10.
- The Precision seems to be the highest around a recall level between 60 and 80, but this could also be a statistical anomaly, and further testing of the system would be required to get a verified result.
- The Phrase Matcher proved to be useful only if there is a 100 match which is not likely to happen as different people write the requirements and object descriptions.

10 Validation: Did we build the right system?

The validation is a qualitative process where the principal question is whether the proposed system is a correct system for the task of automating requirement allocation. The series of interviews were carried out to gain insights into the validity of the system. The observations made during these interviews are to validate the choices that were made during the system design phase and to validate the future potential of the research. The interviews were designed such that both these objectives could be met.

10.1 Methodology: Procedure for validation

There were three main groups of people that the validation was carried out with. The first is the contract managers, the second is tech developers, and tech leads. The last was the specialist civil engineers

The contract managers were chosen for the validation because they are the ones who use Systems Engineering as a tool for writing requirements associating them with particular objects. Contract managers would be the end-users of the proposed system. The tech developers and the tech leads were chosen because they are the ones that would develop and lead the development of tools like the proposed system. As a control or point of comparison, regular engineers who had no prior experience with the definition of requirements or tool development were interviewed after giving a clear explanation of the proposed system.

The validation was conducted with one tech developer, two tech leads, ten systems engineers/ contract managers and two civil engineers.

10.1.1 the structure of interviews

The interview questions were of four main categories.

- Added Value: These questions were to ensure that the proposed system is understood by the interviewee and determine what they perceive to be the added values along with the advantages and disadvantages.
- Applicability: These questions were focused on validating the applicability of the project to the projects that they deal with on a day to day bases. Some of the questions from this section were only applicable to the contract managers.
- System choices: These questions were intended to validate the choices that were made during the research in terms of the selection of technology and technics. There were some questions in this section that was exclusively for the tech developers and leads.
- Future prospects: These questions were intended to find other possible applications and identify specific features for future tools. The interviewee later presented certain propositions which were debatable. The propositions shed light on the way each person vision for automation.

The answers to all the questions have been summarised in the results section. The section is divided based on the four categories mentioned above and the role they have in a project team. A reflection on the results is presented in the observations section.

10.2 Results

10.2.1 Value of research

All the interviewees were able to identify some value in the research. The value manifested itself in different forms for the various actors. The opinion of each category is summarized in Table 10

Contract managers	Tech specialists	Specialist engineers
The value addition was predominantly in the in terms of time saved in performing the task and effort required as manual allocation required an extensive amount of clicking in small menus in Relatics.	The tech specialists saw a business case in the proposed system as it could be sold as a service to external parties. However, they also mentioned that more work and testing is needed before a business case could be justified.	The specialist engineers acknowledge the value and said the proposed system is useful when large OTL's are available. They also mentioned that OTL's are not always present, and a human allocator does not need an OTL to do his job.

Table 10: Value of research from different perspectives

10.2.2 Applicability

A majority of the questions in this section are intended for the contract managers as they were the intended users of the proposed system. The opinion of each category is summarized in Table 11

Contract managers	Tech specialists	Specialist engineers
They felt that the proposed system would apply to all their projects, but the inability of the system to deal with top-level requirements prevents the deployment. The inability to work with Dutch requirements was also mentioned	The tech specialists say that despite the enthusiasm everyone has, the system needs to have a higher accuracy to ensure the credibility of the proposed system. In that sense, we should perform more such studies that which improve the process further.	The specialists felt the system would be applicable even if a manual check is needed because a designer will always check the requirements before designing.

Table 11: Applicability of the proposed system from different perspectives

10.2.3 System choices

The system choices refer to the decisions that were made when the various solutions were compared, and specific tools like a neural classifier were deemed unworthy. The main reasons the neural classifier could not be used was that the classifier could not work with dynamic object categories. The classifier cannot classify a single requirement to multiple objects. These aspects were validated with the questions in this section. The answers have been summarized in Table 12

Contract managers	Tech specialists	Specialist engineers
They were indifferent to whether the system considered the semantics of the object description or not. For contract mangers, the functioning of the system was most important. The objects in a project do change sometimes but not on a daily bases as assumed.	They were critical at these questions and concurred with the decisions not using a neural network but also pointed out that training a word embedding requires a neural network. So they emphasised that the classifier might be useful in other aspects but just not for classifying requirements to objects. A crucial comment at this point is that specific metadata might not be available at all times and in those situations, the neural classifier will still be able to function, but the proposed system might not.	They looked at these questions from a broader perspective and said that the semantics are essential, but they do not have an opinion on how these systems should function. They looked at changing objects as change requests and said that such a system should be able to handle changes in the design requirements.

Table 12: Different perspectives on the choices made during the research

10.2.4 Future applications and prospects

Listed in this section are some of the applications of such a tool outside of the intended use case of this research. The second row refers to how what prospects they see for the technology in terms of investment in new technology (should future investments go towards NLP or training to write better requirements). The answers have been summarized in Table 13

Contract managers	Tech specialists	Specialist engineers
They felt that the proposed system could be used for selecting the correct verification & validation procedure. A more ambitious use case was to automate the selection of standard answers to questions asked by contractors.	They proposed that the system could be used to build OTL's for clients like RWS at a much faster rate and hence provide better services. Apart from this, the tech specialists took a critical look at the other potential applications and concurred that tools like a vocabulary management tool and neural chunk picker are valid applications of the technology behind the proposed system.	The specialists were more interested in using the proposed system for their internal project management, where tasks need to be allocated to specific designers.
They were all new to NLP as a technology but said we should invest in NLP because not all	They were reluctant to point out any choice of investment in technology because more they expected more factual information	They pointed out that investment in NLP will lead to better requirements.

requirements are handled by Royal HaskoningDHV so there will always be poorly defined requirements.	concerning return on investment in technology. For example, X euros gets us a system that is Y % faster.	

Table 13: Different perspectives on future applications and prospects for NLP

10.3 Observations

Many of the results such criticism on accuracy were expected, but this section focuses on the results which were not expected from the validation process. Many of the observations lay the groundwork for the discussion and recommendation section.

- Each actor saw the value only in terms of their perspective. This statement is based on the difference in the perceived value and the wide range of future applications they each saw.
- The wide range of applications that were envisioned requires a word embedding that is better at distinguishing specific terms in civil engineering.
- The process in which the word embedding and dependency parser are used could be improved to achieve better results and classifiers could be playing a role for a better process.
- Unless a system is not able to categories all the types of requirements to a sufficiently accurate range it will be challenging to deploy and get users to adopt the system.
- More research is required to determine investment strategies for NLP application in Civil Engineering.

11 Discussion

The objective of this chapter is to reflect on the behaviour of the proposed system, the choices made during the design of the system, and the verification and validation to stimulate discussions about the proposed system and problems in the domain of information management.

The first section will look at the change in the style of thinking in individuals and team compositions that would nurture the development of tools like the one developed in this research. The second section will look at limitations that were encountered during development and shortcomings of the system in its usability in industry.

11.1 Moving from Mono-disciplinary to an interdisciplinary approach

The comparison between the results and basic principles of the last two iterations is vital for the conclusion and understanding the implications for this research. The neural classifier represents a typical solution that would be given to the problem of allocation of requirements to objects when the problem is seen from a monodisciplinary data science perspective. This is mostly because the obstacle given to a data scientist is to build a system to allocate a requirement to an object and this is a classification problem, and the most straight forward answer to this is to build a classifier. The inherent flaws would just be presented as limitations of the system.

It takes an interdisciplinary approach to realise that although a word embedding and similarity scores produce multiple answers (non-discrete), it is much more capable of handling the dynamic nature of object lists in the project. An interdisciplinary approach to this problem is to look at it as an engineering knowledge capturing problem. Where the style of decision making and thought process that an engineer would use, needs to be replicated by the system. A broad problem definition, such as knowledge capturing in civil engineering is inconceivable from a pure data science perspective. From a pure data science perspective, the problem would be quickly broken down into deliverables, resulting in a classification problem and the development of a classifier once again. For this reason, an interdisciplinary approach is necessary to develop an ideal solution.

The interdisciplinary nature of the problem was validated by tech specialists during the validation process. Each of the three disciplines used to see the problem only from their perspective, so it is not expectable for them to see the problem for another perspective. A data scientist is generally not aware of the role of a systems engineer, but once the drawbacks of specific systems like the neural classifier were explained they do comprehend the problem, it is just that they can not foresee the problems of a given system. The same is applicable for the contract managers who saw many applications for NLP without the insight on the data which is required to train such systems, but when this is explained they help find multiple sources of training information.

To develop the proposed system further or other similar systems, an interdisciplinary team is needed. An interdisciplinary team would ensure early identification of barriers and the development of robust systems.

11.2 Limitations in development and deployment

11.2.1 Representative object libraries

The proposed system requires an object description that is meaningful and contains the right words to describe the object. Most of the objects in the OTL contain a concise and representative definition of that object for a given asset. However, the number of objects in the OTL are limited and not all the requirements might find a suitable object. The project object lists that were available suffered from inadequate definitions that were not suitable for either syntactic or semantic analysis. Object libraries that are more than just documentation would help achieve the full potential of the proposed system.

The level of detail of the metadata in the object libraries is a limitation for the accuracy of the proposed system. This aspect was mentioned in the validation processes, and a potential solution could be to use a classifier which does not require metadata.

11.2.2 Language problems

The proposed system works only for requirements and object descriptions that are in English. However, most of the projects in the Netherlands use the Dutch language for the definition of the requirements. This is an unavoidable limitation of the system. This limitation was also discussed during the validation. The OTL, on the other hand, does have a few object families that are in English. This is it is a step in the right direction in terms of the usability of information systems for automated and intelligent systems. This limitation of language compatibility is because there is no dependency parser for the Dutch language and the lack of Dutch word embedding with Civil Engineering terms in their vocabulary.

Both of these barriers can be overcome to a great extent if the recommendations are implemented. However, for now, they remain as limitations of the proposed system.

11.2.3 Validity of verification

The verification was only carried out for twenty requirements from two projects. To get a statistically significant result more tests need to be carried out. This was mention by the tech specialists during the validation meeting but is also recognisable from the results of the verification process. The reasons and solutions for this have already been discussed in the observations in section 9.3

11.2.4 Process improvements

The proposed system uses a basic process of dependency parsing and checking the cosine similarity can be improved with a better algorithm. The current algorithm has no asset-specific filters to ignore or pprioritize specific requirement chunk. This was done to ensure the proposed system is transferable to other types of assets but in the future if specific words need to be prioritized or ignored they need to be added into the NLP pipeline.

11.2.5 Industry readiness

The proposed system is a provisional design that is written in an elementary manner. It was coded to show a proof of concept and not to be deployed into the industry in the current form. There needs to be a redevelopment from scratch and additional functionalities like exception handling capabilities.

12 Conclusion

This research explores the first steps required to move towards automated building information management by attempting to automate the functional analysis and allocation of requirements to objects. The first barrier to any automation is the internalisation of information. In the context of this research, the information is the requirements and objects. The internal environment any file format that a Python script can access. The initial strategy was to interact with a Relatics workspace through API's and IFC files to extract the information. However, this proved to be challenging, and it was decided to directly download the requirements from the Relatics workspace and use a standard Object Type Library for the list of objects, these objects also have a description in the form of metadata.

Once the information was internalised, the next barrier was to understand the syntax and semantics of the requirements and object descriptions. Several approaches were possible, but this research focuses on using natural language processing. The proposed system uses a dependency parser to add linguistic features to the requirements and object descriptions. The parsed information now has a syntax, and it is now possible to perform syntactic operations on the information. The proposed system then performs a task known as Noun Chunking, where nouns and the words around them that make a meaningful phrase.

The noun chunks represent some semantic information about the requirements and object description. The next barrier is to use a semantically measure the similarity between the chunks of the requirements and objects. To measure the similarity, the proposed system uses a word embedding. A word embedding is a vector representation of words that have a similar meaning and used in a similar context. The proposed system plugs in the chunks of the requirement and object description to assign a vector for each of the chunks. Then the system measures perform a dot product on the two vectors. The dot product is also known as a cosine similarity check, which is a proven measure to find the semantic similarity between two-word vectors.

The semantic comparison of the various word vectors is useful, but it is not sufficient to base decisions of allocation purely based on these similarities. The proposed system takes into account the title of the requirement and prioritises chunks that are relevant to the title of the requirement. This ensures hot chunks that are not relevant to the given requirement are not checked for similarity with object chunks. This process is not specific to any a single type of infrastructure or asset, so there are no sound barriers to apply the proposed system to any construction project.

Verification and validation were performed to measure the performance of the proposed system. The verification process measured precision and recall for a given set of requirements. The verification process showed that the proposed system had the highest precision (Between 60 and 85 per cent) at a recall level between 60 and 80 per cent. The validation process, amongst other things, showed that this precision should be higher. The validation process also shed light on how to improve the proposed system and alternative uses for the proposed system which will be discussed in the next chapter.

13 Recommendations: Filling gaps in technology for automation

The recommendations presented within this chapter follows from the observations made during the design of the proposed system and the verification and validation process. There are two main recommendations which have implications not only to the current system but also open doors for further development of new systems which can potentially contribute towards further automation. The recommendations have been allocated either to academia or industry, but most of the recommendations require the combined effort of both.

13.1 Recommendations for academia

13.1.1 Building a Civil Engineering word embedding

Iteration 4 was about training a word embedding, but our attempt was unsuccessful because the data needs to contain words that repeat several times in the entire training corpus. If there were more requirements from other projects which also speak of the same objects and similar requirements, then they can be used to train a word embedding. Having a word embedding that was trained on Civil Engineering knowledge means that the similarity scores would be much more accurate and based on the right semantics. A Civil engineering word embedding would be essential if such systems are to work for Dutch data. Incorporation of Dutch data was not possible because the pre-trained word embeddings in Dutch did not include the plethora of industry-specific word that was present in the requirements and object description.

The two approaches to the development of a civil engineering word embedding are described as in appendix 10

13.1.2 Generating data for generative design

The concept of generative design has been covered in section 2.2 Generative Design. Studies by Autodesk Research into generative design have primarily used object linked to requirements that they fulfil. Generative design as a concept in Civil Engineering is still in its infancy, but one of the challenges is having the data in an internal context (Karan & Asadi, 2019).

The solution for requirement allocation to objects addresses the problem of internalising the requirements to a given set of objects. So this research is one step closer to generating large amounts of meaningful Big-Data which can be then used as an input for generative algorithms. The only missing ingredient is 3D BIM models, and that can only be incorporated after the modelling practices are improved. The 3D BIM data represents the main challenge in this case. This represents the collection of engineering knowledge in the form of big data from construction documents using unsupervised learning.

13.1.3 Root cause analysis for non-smart requirements

The first two recommendations looked at implementing various technologies, but academia should also look into why standards such as the system engineering guidelines are not followed in practice. This research will help focus on the areas in which smart tools should be built.

13.2 Recommendations for the industry.

13.2.1 Project vocabulary management tools

One of the biggest problems in the construction sector is that various actors and stakeholders use different terminology while speaking about the same object or concept. It has already been mentioned that the systems engineering guidelines ask all the parties involved to develop a common vocabulary of words. However, even after developing a list of words, it is challenging to ensure everyone follows it and enforcement is not an option.

Word embedding that was specially trained on the terms used in civil engineering data can be beneficial in this situation. The ability to check for the similarity between words inside this space will allow for the development of tools that recognise the use of words outside the project vocabulary and then notify the writer that; this project uses XYZ word rather than XXY.

Such tools will help in ensuring a uniform project vocabulary with simple plugins into any software that is used by any of the actors. This would also make the documents much more uniform, making it much more suitable for training future word embeddings and other intelligent systems. The documentation from projects that use these systems would be the perfect data set for future research. This notion sets the stage for the next potential application.

13.2.2 Using similarity scores as features for Neural classification

The use of neural classifiers for the classification of requirements to objects has already been discussed extensively and argued that neural classifiers are not the right choice. However, neural classifiers can have potential application in picking out the right noun chunks to be plugged into a word embedding. A neural classifier, as usual, requires labelled data; in this case, labelled data would constitute noun chunks that are labelled as good chunks and bad chunks. Good and bad merely refer to whether the chunks are representative of the meaning of the chunk.

The challenge here is that the labelled data would have to be generated by asking engineers a questionnaire. This is where companies like Royal HaskoningDHV can harness their large amounts of engineering knowledge that is present in the form of the thousands of engineers that are employed. With an ageing workforce, it makes more sense to try to capture their knowledge before it is lost forever. This data collection operation does not only mean that the proposed system gets more accurate. However, in terms of research, it represents the collection of engineering knowledge for supervised learning right from the engineers and having something useful to do with this research.

13.2.3 Other allocation tools

Various other tasks that are similar to allocations of requirements to objects can also be automated with minor changes to the system proposed in this research. Tasks such as the selection of verification and validation methods for a given design can be automated.

13.2.4 interdisciplinary teamwork

The research showed that a momo disciplinary approach to automation would lead to a sub-optimal result that might not be able to handle the dynamic nature of the industry. Therefore it would be better to do further research in interdisciplinary teams.

References

A H M Berlo, L., Beetz, J., Bos, P., Hendriks, H., & C J Tongeren, R. (2012). *Collaborative engineering with IFC : new insights and technology*.

Alsem, D., Kamerman, J., van Leeuwen, C., van Ruijven, L., den Toom, T., & Vos, M. (2013). *The Guideline for Systems Engineering within the Civil Engineering sector*. Retrieved from Netherlands: https://www.leidraadse.nl/assets/files/downloads/LeidraadSE/V3/Leidraad_V3_SE_web.pdf

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463): ACM press New York.

Bhooshan, S. (2017). Parametric design thinking: A case-study of practice-embedded architectural research. *Design Studies, 52*, 115-143. doi:https://doi.org/10.1016/j.destud.2017.05.003

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python* [1 online resource (xx, 479 pages) : illustrations].

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

Bohnacker, H., Groß, B., Laub, J., Lazzeroni, C., Gross, B., Laub, J., & Lazzeroni, C. (2009). *Generative Gestaltung: entwerfen, programmieren, visualisieren*: Schmidt Mainz.

Brous, P., Herder, P., & Janssen, M. (2015). Towards Modelling Data Infrastructures in the Asset Management Domain. *Procedia Computer Science, 61*, 274-280. doi:https://doi.org/10.1016/j.procs.2015.09.215

Cheong, H., Li, W., Shu, L., Bradner, E., & Iorio, F. (2014). *Investigating the use of controlled natural language as problem definition input for computer-aided design.* Paper presented at the Proceedings of the 2014 International Conference on Innovative Design and Manufacturing (ICIDM).

Choi, J. D., Tetreault, J., & Stent, A. (2015). *It depends: Dependency parser comparison using a web-based evaluation tool.* Paper presented at the Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).

Chollet, F. o. (2018). *Deep learning with Python*. Shelter Island, NY: Manning Publications Co.

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*: John Wiley & Sons.

Farmer, M. (2016). The farmer review of the UK construction labour model. *Construction Leadership Council*.

Geiger, A., Benner, J., & Haefele, K. H. (2015). Generalization of 3D IFC building models. In *3D Geoinformation science* (pp. 19-35): Springer.

Ghannad, P., Lee, Y.-C., Dimyadi, J., & Solihin, W. (2019). Automated BIM data validation integrating open-standard schema with visual programming language. *Journal Advanced Engineering Informatics, 40*, 14-28.

Hoeber, H., & Alsem, D. (2016). Life-cycle information management using open-standard BIM. *Engineering, Construction and Architectural Management, 23*(6), 696-708. doi:10.1108/ECAM-01-2016-0023

Karan, E., & Asadi, S. (2019). Intelligent designer: A computational approach to automating design of windows in buildings. *Automation in Construction, 102*, 160-169. doi:https://doi.org/10.1016/j.autcon.2019.02.019

Kazi, R. H., Grossman, T., Cheong, H., Hashemi, A., & Fitzmaurice, G. W. (2017). *DreamSketch: Early Stage 3D Design Explorations with Sketching and Generative Design.* Paper presented at the UIST.

Kinsley, H. (2015). Chunking with NLTK. *Natural Language Processing tutorial.* Retrieved from https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/

Lee, Y.-C., Solihin, W., & Eastman, C. M. J. A. i. C. (2019). The Mechanism and Challenges of Validating a Building Information Model regarding data exchange standards. *100*, 118-128.

Marr, B. (2015). *Big Data: Using SMART big data, analytics and metrics to make better decisions and improve performance*: John Wiley & Sons.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miles, J. (1996). *Where is the Henry Ford of future housing systems?* : Royal Academy of Engineering London.

Monedero, J. (2000). Parametric design: a review and some experiences. *Automation in Construction, 9*(4), 369-377.

Information Management for European Road Infrastructure using Linked Data | Investigating the Requirements. Retrieved from https://roadotl.geosolutions.nl/static/media/INTERLINK_D2_D3_WPA_WPB_Report_Final_Issue.pdf

Oxman, R. (2017). Parametric design thinking. *Design Studies, 52*, 1-3. doi:https://doi.org/10.1016/j.destud.2017.07.001

Pennington, J., Socher, R., & Manning, C. (2014). *Glove: Global vectors for word representation.* Paper presented at the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

S. Perone, C. (2013). Machine Learning :: Cosine Similarity for Vector Space Models (Part III). *Machine Learning.* Retrieved from http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

Sacks, R., Eastman, C. M., & Lee, G. (2004). Parametric 3D modeling in building construction with examples from precast concrete. *Automation in Construction, 13*(3), 291-312.

Santos, M. Y., Martinho, B., & Costa, C. (2017). Modelling and implementing big data warehouses for decision support. *Journal of Management Analytics, 4*(2), 111-129. doi:10.1080/23270012.2017.1304292

Sebastian, R., & van Berlo, L. (2010). Tool for Benchmarking BIM Performance of Design, Engineering and Construction Firms in The Netherlands. *Architectural Engineering and Design Management, 6*(4), 254-263. doi:10.3763/aedm.2010.IDDS3

Thein, V. (2011). Industry foundation classes (IFC). *BIM interoperability through a vendor-independent file format*.

Tixier, A. J. P., Hallowell, M. R., Rajagopalan, B., & Bowman, D. (2016). Application of machine learning to construction injury prediction. *Automation in Construction, 69*, 102-114. doi:https://doi.org/10.1016/j.autcon.2016.05.016

Tulkens, S., Emmery, C., & Daelemans, W. (2016). Evaluating unsupervised Dutch word embeddings as a linguistic resource. *arXiv preprint arXiv:1607.00225*.

Valero, E., Forster, A., Bosché, F., Hyslop, E., Wilson, L., & Turmel, A. (2019). Automated defect detection and classification in ashlar masonry walls using machine learning. *Automation in Construction, 106*, 102846. doi:https://doi.org/10.1016/j.autcon.2019.102846

Van Nederveen, S., Beheshti, R., & Willems, P. (2010). *Building information modelling in the Netherlands: a status report.* Paper presented at the W078-Special Track 18th CIB World Building Congress May 2010 Salford, United Kingdom.

Wasson, C. S. (2015). *System engineering analysis, design, and development: Concepts, principles, and practices*: John Wiley & Sons.

Winch, G. M. (2003). How innovative is construction? Comparing aggregated data on construction innovation and other sectors a case of apples and pears. *Construction Management and Economics, 21*(6), 651-654.

Woodhead, R., Stephenson, P., & Morrey, D. (2018). Digital construction: From point solutions to IoT ecosystem. *Automation in Construction, 93*, 35-46.

Zhang, J., & El-Gohary Nora, M. (2015). Automated Information Transformation for Automated Regulatory Compliance Checking in Construction. *Journal of Computing in Civil Engineering, 29*(4), B4015001. doi:10.1061/(ASCE)CP.1943-5487.0000427

Appendix 1: IFC 2x3 Schema

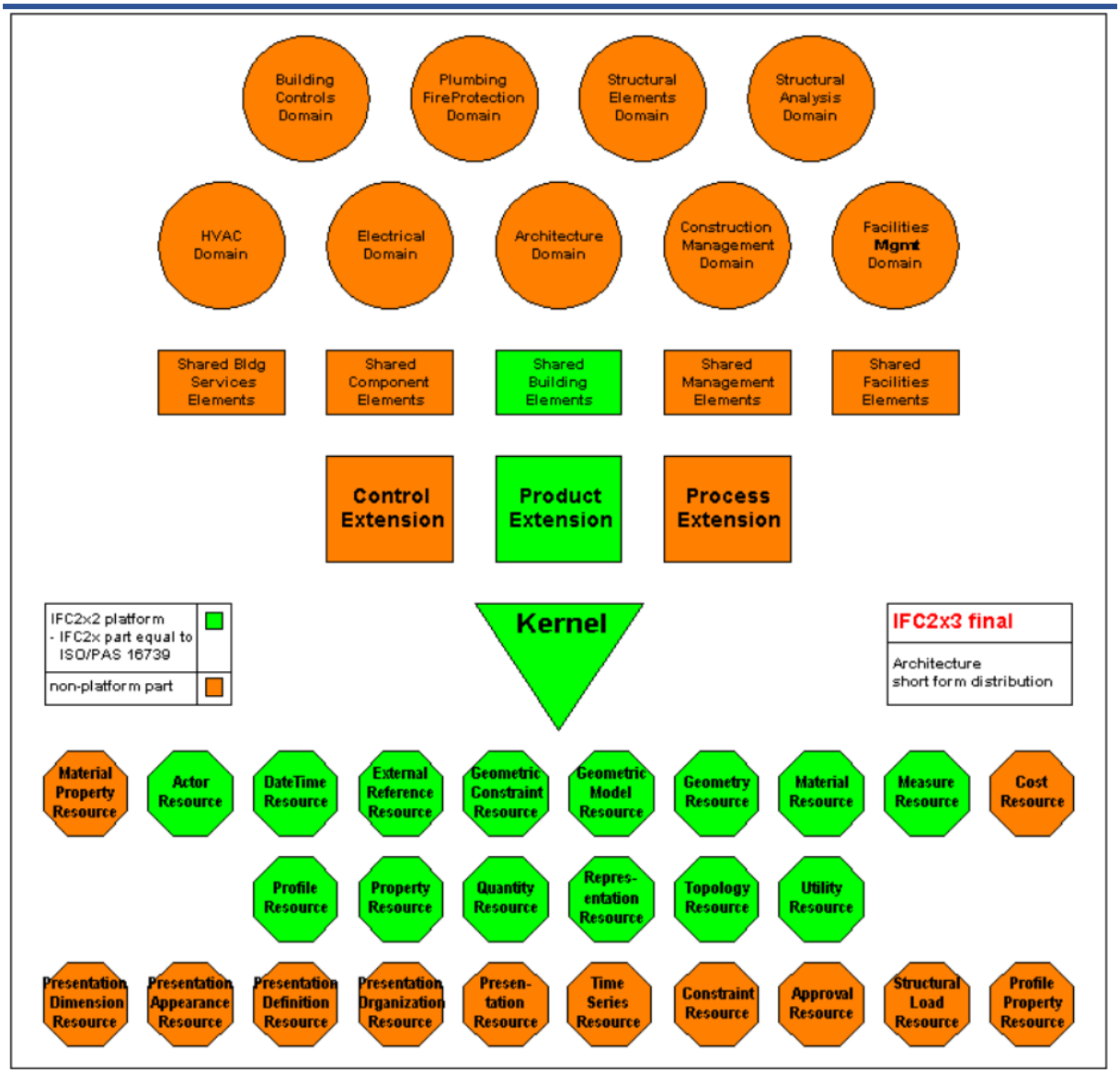

Figure 17: IFC 2X3 Schema

Appendix 2: OTL tree diagram of viaducts (Dutch)

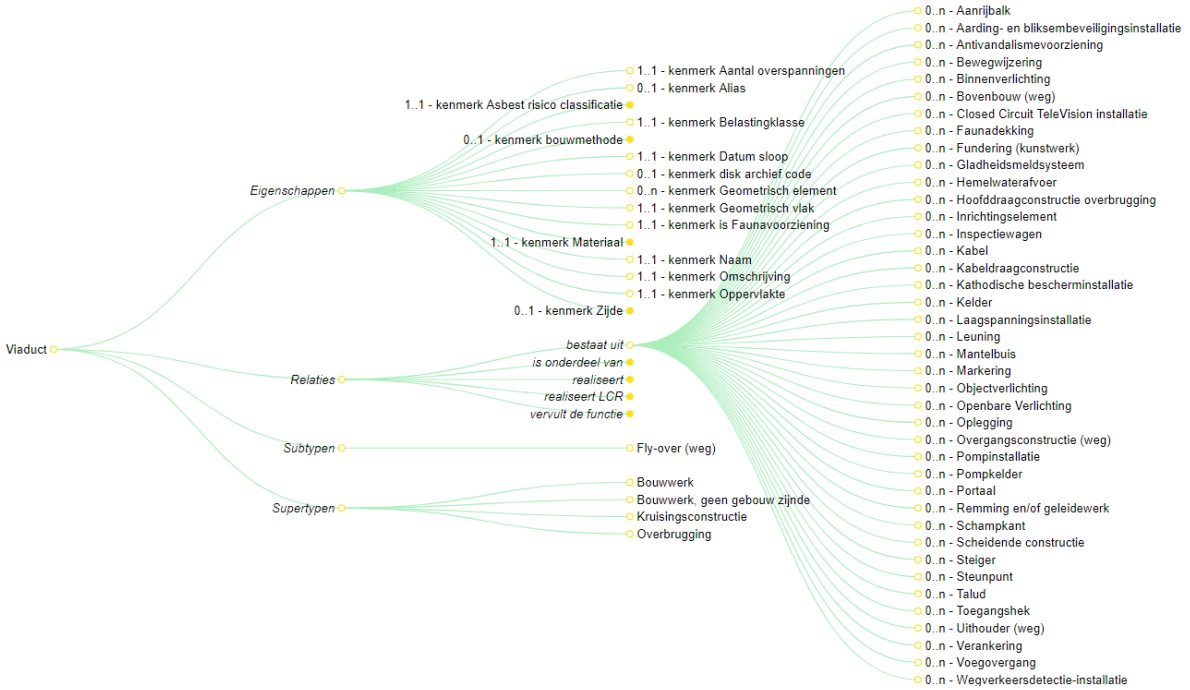

Figure 18: Tree diagram for viaducts from Rijkswaterstaats' OTL

Appendix 3: IDEF0 Template

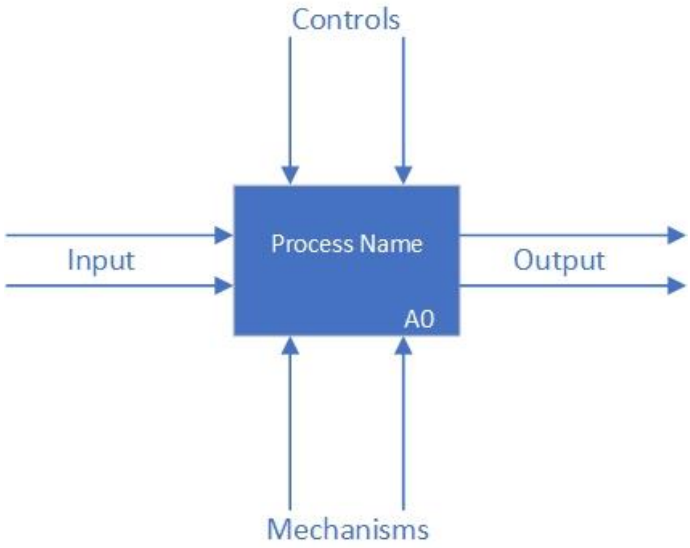

Figure 19: Basic IDEF0 Template

Appendix 4: Requirements analysis in ISO 15288 and V- model

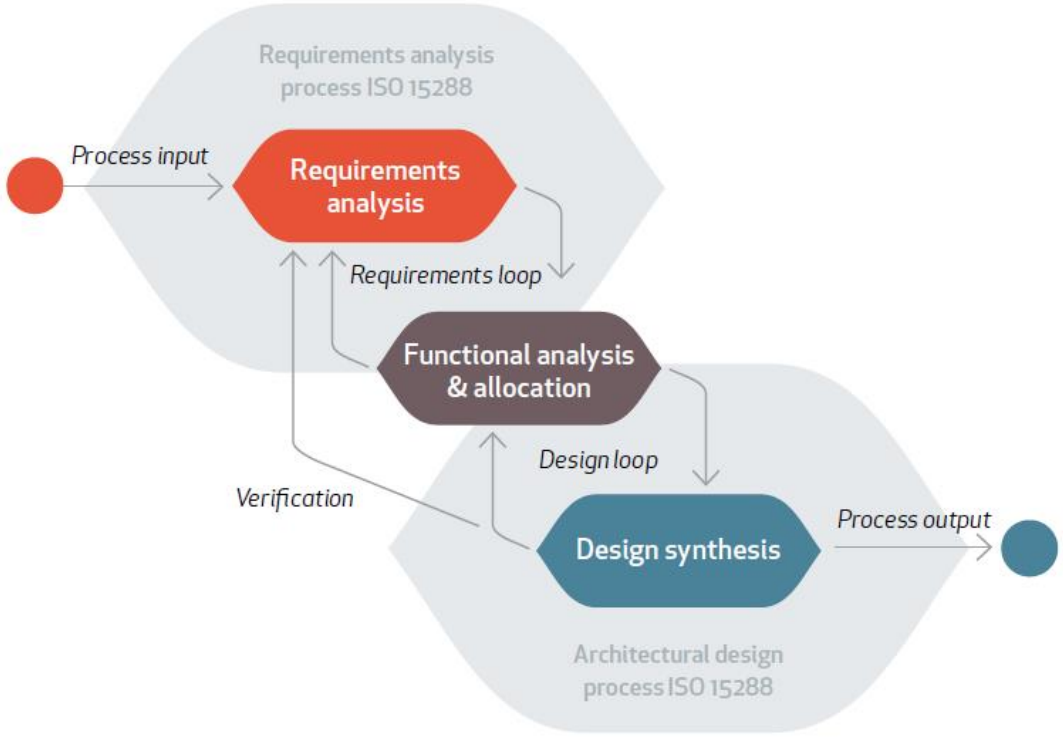

Figure 20: Iterative nature of requirements and design Source:(Alsem et al., 2013)

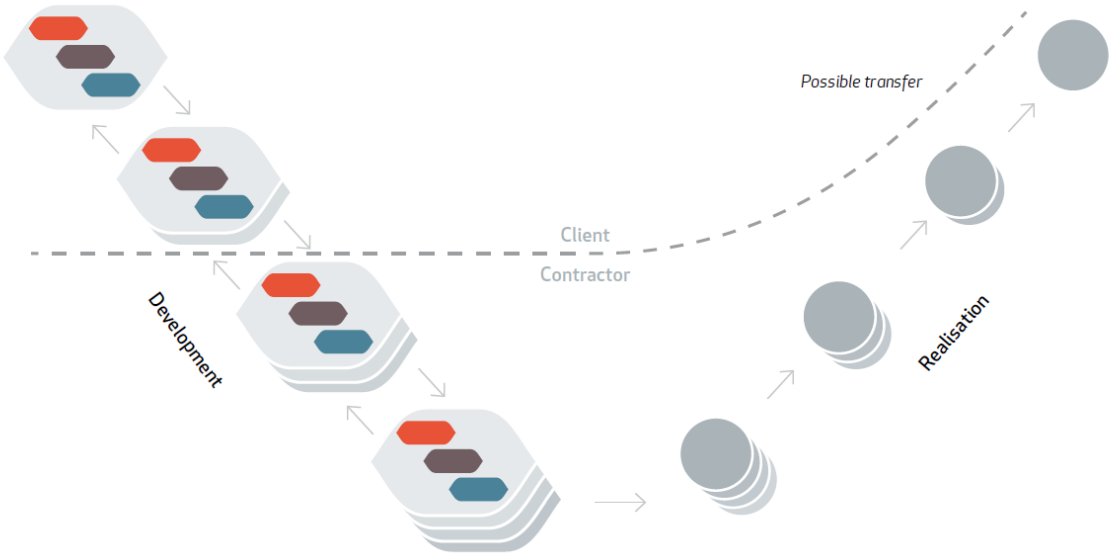

Figure 21: V- model Source: (Alsem et al., 2013)

Appendix 5: Typical Relatics Type Design

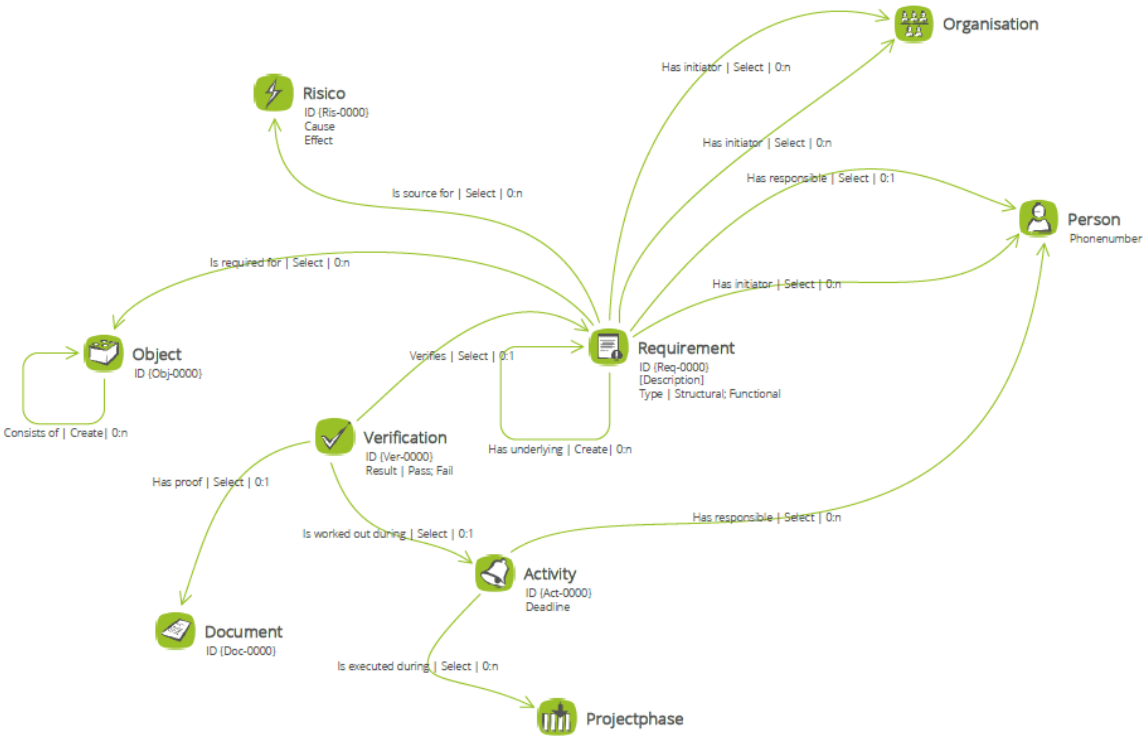

Figure 22: Relatics Type Design for a requirements management system in construction project

Appendix 6: regular expressions in Python

(Kinsley, 2015)

Identifiers:

- \d = any number
- \D = anything but a number
- \s = space
- \S = anything but a space
- \w = any letter
- \W = anything but a letter
- . = any character, except for a new line
- \b = space around whole words
- \. = period. must use backslash, because . normally means any character.

Modifiers:

- }1-3{ = for digits, u expect 1-3 counts of digits, or "places"
- + = match 1 or more
- ? = match 0 or 1 repetitions.
- * = match 0 or MORE repetitions
- $ = matches at the end of string
- ^ = matches start of a string
- | = matches either/or. Example x|y = will match either x or y
- [] = range, or "variance"
- }x{ = expect to see this amount of the preceding code.
- }x,y{ = expect to see this x-y amounts of the preceding code

White Space Charts:

- \n = new line
- \s = space
- \t = tab
- \e = escape
- \f = form feed
- \r = carriage return

Appendix 7: POS Tags in NLTK

(Kinsley, 2015)

```
POS tag list:

CC     coordinating conjunction
CD     cardinal digit
DT     determiner
EX     existential there (like: "there is" ... think of it like
"there exists")
FW     foreign word
IN     preposition/subordinating conjunction
JJ     adjective    'big'
JJR    adjective, comparative     'bigger'
JJS    adjective, superlative     'biggest'
LS     list marker  1)
MD     modal  could, will
NN     noun, singular 'desk'
NNS    noun plural  'desks'
NNP    proper noun, singular      'Harrison'
NNPS   proper noun, plural 'Americans'
PDT    predeterminer'all the kids'
POS    possessive ending   parent\'s
PRP    personal pronoun    I, he, she
PRP$   possessive pronoun  my, his, hers
RB     adverb very, silently,
RBR    adverb, comparative better
RBS    adverb, superlative best
RP     particle     give up
TO     to     go 'to' the store.
UH     interjection errrrrrrrm
VB     verb, base form     take
VBD    verb, past tense    took
VBG    verb, gerund/present participle  taking
VBN    verb, past participle      taken
VBP    verb, sing. present, non-3d      take
VBZ    verb, 3rd person sing. present   takes
WDT    wh-determinerwhich
WP     wh-pronoun   who, what
WP$    possessive wh-pronoun      whose
WRB    wh-abverb    where, when
```

Appendix 8: OTL objects and descriptions in English

Code	EN name	relation	EN Discription
V1	Impact bar	is	A beam-shaped collision protector over the road to protect structures against impact by a vehicle.
V2	Earthing and lightning protection system	is a	System to protect buildings and structures against the effects of lightning strikes and to protect people and animals in case of earth current faults
V3	Anti-vandalism provision	is	A construction that prevents or impedes the deliberate and/or malicious destruction of physical objects.
V4	Signage	is a	Visual aid that is installed on, along or above the paving to guide, control, inform and warn the traffic and to help road users to determine their route.
V5	Interior lighting	is a	Installation that serves to illuminate spaces inside a building (source: BB art. 6.2).
V6	Paving	is a	Construction consisting of one or more paving layers, to make traffic on the site possible
V7	Closed Circuit Television installation	is a	System with which a certain area can be visually observed and recorded remotely.
V8	Fauna cover	is a	Place that provides shelter to animals consisting of vegetation or natural plants.
V9	Foundation (structure)	is a	Construction to distribute forces over the underlying ground or transfer them to a deeper supporting layer
V10	Slipperiness warning system	is	A system that uses sensors in or beside the road to measure variables such as temperature and humidity and sends these data to a control room where the chance of ice is determined.
V11	Rainwater drainage	is a	Rainwater drainage is a system that collects, removes and transports rainwater

V12	MAIN WEAR CONSTRUCTION BRIDGING	is a	Primary load-bearing construction of a bridge that absorbs occurring loads.
V13	Planning element	is a	Collection of furniture and objects that are used in buildings and in public space.
V14	Inspection car	is a	Movable vehicle part of an object that can be transported along a pipe for maintenance and inspection to places in an object that are otherwise difficult to reach or inaccessible.
V15	Cable	is	A conductor that is provided with a mantle and is intended for transport of energy or data.
V16	Cable support construction	is a	Cable support construction
V17	Cathodic protection system	is a	System for corrosion prevention that relies on the principle of reducing the potential of the object to be protected.
V18	Cellar	is a	Architectural construction or space under the ground floor
V19	Low voltage installation	is a	Installation for receiving and using electrical energy below 1000 volts (alternating current) or 1500 volts (direct current) (source: BB, art. 6.8).
V20	Railing	is a	Construction installed on a wall or partition that provides support and protects against falling at a height difference
V21	Conduit	is a	Tube for protection of cables, pipes and HDPE tubes
V22	Marking	is a	Visual aids that are applied on or in the paving to guide, inform, alert and regulate traffic.
V23	Object lighting	is a	Installation that serves to illuminate the exterior of buildings and other constructions or objects
V24	Public lighting	is a	Installation that serves to light public infrastructure, especially roads.
V25	Support	is a	Construction that takes up forces and deformations from the superstructure of a built structure and fully or partially transfers these forces to the substructure.

V26	Transition construction (road)	is a	Construction that is installed in the road embankment and is installed with a hinge on a structure to enable settlement differences between the embankment and structure to occur evenly
V27	Pump system	is a	System that can displace a liquid or gas using a pump
V28	Pump cellar	is a	Building that is fully or partially installed under the ground for a pump system
V29	Portal	is a	Construction for fastening traffic installation elements or signs consisting of two vertical columns with a horizontal beam across the top that forms a rigid connection with the columns.
V30	Piled fendering and/or guide wall	is a	Construction along or next to a navigation channel, adjoining a built structure intended to provide guidance to ships and/or to moor ships.
V31	Kerb	is a	Low-height construction built along the road to prevent the traffic leaving the road.
V32	Partition construction	is a	Usually linear construction intended as an obstacle to separate two areas from each other.
V33	Jetty	is a	(Often wooden) construction built above the water or floating for landing and mooring boats and on which the shore can be reached by foot
V34	Supporting point	is	A construction connected to the immovable world on which the main load-bearing structure is supported.
V35	Slope	is a	Inclined side of an embankment.
V36	Access gate	is a	Pivoting construction that (electrically) mechanically swings around a vertical axis or rolls over a surface to create a closable passage in a wall or enclosure.
V37	Pylon (road)	is	A construction element that serves to support other construction elements.
V38	Anchoring	is a	Fastener to connect construction element with an anchor and to fix them to the substrate for a stable structure.

V39	Joint transition	is a	Construction with seal that forms a continuous (road) surface between adjoining main structural elements (e.g. bridge sections/land abutments) and that guarantee the continuity of the road surface.
V40	Road traffic detection system	is a	Installation that detects the presence or passage of a vehicle at a certain location in a road network.
VM1	Anchor plate	is a	Flat structural element to support the anchor rod
VM2	Anchor	is a	Soil mechanical construction element for geotechnical stabilization or retaining structures.
VM3	Counterweight boom	is a	Swinging part of a Dutch draw bridge with the link arm at the front end and the counterweight at the rear.
VM4	Beam	is a	Load-bearing horizontal structural element or which the length is many times longer than the width or the height in the cross section
VM5	Protective coating	is a	Layer of material that is applied to protect a surface against external effects
VM6	Fastener (construction)	is a	Component for fixing construction elements and keeping them in place.
VM7	Arch	is a	Tension structure in a curved form in which the forces are transferred through the curve to two support points.
VM8	Platform	is a	Horizontal construction with hardened surface at, or along a wall that provides a place to stand or passage for people.
VM9	Bridge deck	is a	Construction element of the supporting structure or an engineering structure about which traffic can drive.
VM10	Tube	is a	Hollow cylindrical pipe for carrying liquids, gases or capsules, or to protect cables
VM11	Corbel	is a	Protruding structural element on a construction such as a wall that serves as a support for another structural element such as a mast, beam or slab
VM12	Counterweight	is a	Component of a movable bridge that serves as a counterweight for the bridge deck.

VM13	Robbery	is a	Exterior partition structure with a predominantly horizontal character
VM14	Box float	is a	Mooring point, part of a floating bollard, that ensures that the bollard is carried up and down with the ship along the locks between rails in the lock wall.
VM15	Cross beam (structure)	is a	Beam in a construction that connects the main beams together for a more transfer of loads and that can reduce the length of the longitudinal beams.
VM16	Cantilever column	is a	Column of a Dutch draw bridge on which the counterweight boom rests
VM17	Hanger (arch bridge)	is a	Structure in the main support structure of an arch bridge, that makes the connection between the arch and the road deck
VM18	Link arm	is a	Rod in a movable bridge that connects the bridge deck and the counterweight boom
VM19	Capital	is a	Top piece of a column that can transfer the force of a bearing surface
VM20	Junction (construction)	is a	Place where staves meet and provide a connection.
VM21	Column	is a	Vertical construction that can serve as a support.
VM22	Ladder	is a	Construction that consists of two rails with a series of rungs in between to span height differences.
VM23	Spar (structure)	is a	Beam between the cross beams, or from support to support, to carry the deck of the structure
VM24	Transition construction (road)	is a	Construction that is installed in the road embankment and is installed with a hinge on a structure to enable settlement differences between the embankment and structure to occur evenly
VM25	Pier	is a	Supporting structure of a bridge, viaduct and similar built structures that function as a free-standing support.
VM26	Plate field	is a	Part of the main load-bearing structure formed by a combination of steel or concrete plates fastened to each other.
VM27	Railing	is a	A construction built as a fence that protects people against the risk of falling.

VM28	Grate	is a	Horizontal framework or parallel bars, or framework or crossing bars.
VM29	Groove superstructure	is a	Steel superstructure on the main beam or a bascule bridge
VM30	Guy line	is a	Cable or tube that takes up tension forces
VM31	Truss	is a	A framework, typically consisting of rafters, posts, and struts, supporting a structure.
VM32	Joint-free transition	is a	Joint construction in a built structure that is finished on the top side of the paving and therefore no longer visible from the exterior
VM33	Wall	is a	Partition structure with a predominantly vertical character
VM34	Wind brace	is a	Stability brace that can be installed in the surface of a construction for reinforcement. This often involves a diagonal brace in a rectangular frame.

Appendix 9: NLP in Python using NLTK with example

Traditional NLP tries to use the syntax of human language to capture the meaning of a given sentence. There are several functions in NLP which can be used to analyse any text. A detailed study was conducted to develop a system that uses these functions; This process started with identifying all the various functions available in NLP packages

NLTK stands for Natural Language Tool Kit is a python tool kit that offers most of the functionalities of NLP in the python environment. Many functions were considered, but only the ones that were relevant for analysing requirements and objects are explained below.

Tokenisation: Tokenisation is a process by which vast bodies of text are broken down into smaller sections. There are word tokenisers that can break up sentences into words (word tokenisers) and paragraphs into sentences (sentence tokenisers). Tokenisation necessary as the requirements and object descriptions needs to be broken into smaller words to compare(Bird, Klein, & Loper, 2009).

Lexicon: A lexicon is a variation in the meaning of a word based on the situation in which it is used(Bird et al., 2009). For example, when someone says "the employee was fired", it does not mean he/ she has been on fire, it means the person was no longer employed. To avoid such confusions its necessary to have a good collection of lexicons in a corpus[14]. Lexicons are essential in the context of requirements, as many of the terms used are context-dependent.

[14] A corpus is a large body of text which functions as the database for NLP tasks.

Chunking: Chunking is the process by which tokens of a given sentence are grouped to make meaningful chunks of words(Bird et al., 2009). Chunking in NKTL uses regular expressions[15] and POS tags[16] in specific patterns to identify the tokens that need to be chunked together.

Named entity recognition: In any given sentence, there are several entities that can be identified(Bird et al., 2009). Named entity recognisers are used to automatically extract entities such as people, places, locations, organisations etc.

WordNet: WordNet is a digital dictionary for the English language. In NLTK it behaves as a lexical database with the meaning of words, synonyms and antonyms. The University of Princeton developed this corpus.

Example

The above procedure was tested on one object description to make it easier to analyse if an individual step is not functioning as it was supposed to. The same object and its description were used in Dutch.

Text in English: "Construction to distribute forces over the underlying ground or transfer them to a deeper supporting layer."

Text in Dutch: "Constructie om krachten te verdelen over de onderliggende grond of over te brengen naar een dieper gelegen draagkrachtige laag."

13.2.5 Tokenizing and POS Tagging

The above statements are the description for the foundation (Dutch: fundering) Both of these were retrieved from the OTL of a Viaduct. Different functions had to be called for simple tasks such as tokenisation in both languages. Therefore it became very clear from the start that different tools had to be used to achieve the same function for both the Dutch and English data.

POS tagging was very straight-forward for both English and Dutch; however, the POS tagging accuracy for the Dutch object descriptions could not be verified due to the lack of language expertise.

13.2.6 Chunking using Regular Expressions

The next step was to use regular expressions in conjuncture to the POS tags that were added in the previous step. This proved to be challenging as writing Chunk Grams[17] is a linguistic skill. An Example of Chunk Gram is <RB.?>*<VB.?>*<NPP+<NN>?. This example looks for one or more adverbs (RB) which are followed by a verb (VB) which is followed one or more proper nouns (NPP) which are followed by an optional singular noun (NN). It was possible to develop a similar chunk gram to identify the given object description, but it was next to impossible to formulate a chunk gram that applied to all object description even though all the descriptions were retrieved from the OTL. It seems quite farfetched to be able to systematically reduce the requirements into chunks in the above-described manner as the requirements were poorly

[15] Regular expressions represent a pattern of data and in our case words that would appear in a sentence that is required to be identified. All the regular expressions in python are listed in

[16] POS tags refer to parts of speech tags such as noun, pro noun, verb, adjective and adverb etc.

[17] A chunk gram is set of rules formulated using a the regular expressions and POS tags.

defined. The requirements would need to be translated into English, so there is also the possibility that some errors are made there as well.

Appendix 10: Training word embeddings

Retrain a pre-trained word embedding

Retraining a word embedding is recommended when the size of the sample data is not extensive. There are several Python API's that are capable of retraining a word embedding, and the procedure is commonplace, so there is no need to go further into how to train.

It is worthwhile to discuss some other points that should be kept in mind before retraining a word embedding. It is generally accepted that pre-trained word embeddings are well trained, so moving words around is not something that should be done on one-off data sets. The style of writing of the training data should be representative of the testing data. Only if such data is not available, then general data from the field of civil engineering should be used. This might result in all the vocabulary being covered, but similarity might not match the lexicons from civil engineering.

In the most basic sense, one should retrain the word embedding only if they have copious amounts of the data they plan to test on and train it for a specific function. In this case, that would be requirements and objects. Therefore collecting this data is the barrier that needs to be achieved.

Training new word embedding

When the objective of the word embedding is to cover a wide variety of knowledge that is applicable in several domains, it is better to train a word embedding from scratch. A large amount of data is necessary to train such a word embedding that cover a vast range of topics but developing such a word embedding can be quite useful in automating several tasks. The biggest challenge for training a word embedding from scratch is the amount data that is required; the pre-trained word embeddings are trained on massive corpora containing millions of new articles or reviews or any other form of text. Replicating these would be a large scale data collection project.