Eindhoven University of Technology

MASTER

Improving the management of structural engineering requirements in the design phase linking project requirements to BIM, based on the semantic web and linked data principles

Stancheva, M.

Award date:
2017

Link to publication

Improving the management of structural engineering requirements in the design phase

Linking project requirements to BIM, based on the Semantic Web and Linked Data principles

MIRYANA STANCHEVA
SEPTEMBER 27, 2017

CONSTRUCTION MANAGEMENT & ENGINEERING
EINDHOVEN UNIVERSITY OF TECHNOLOGY
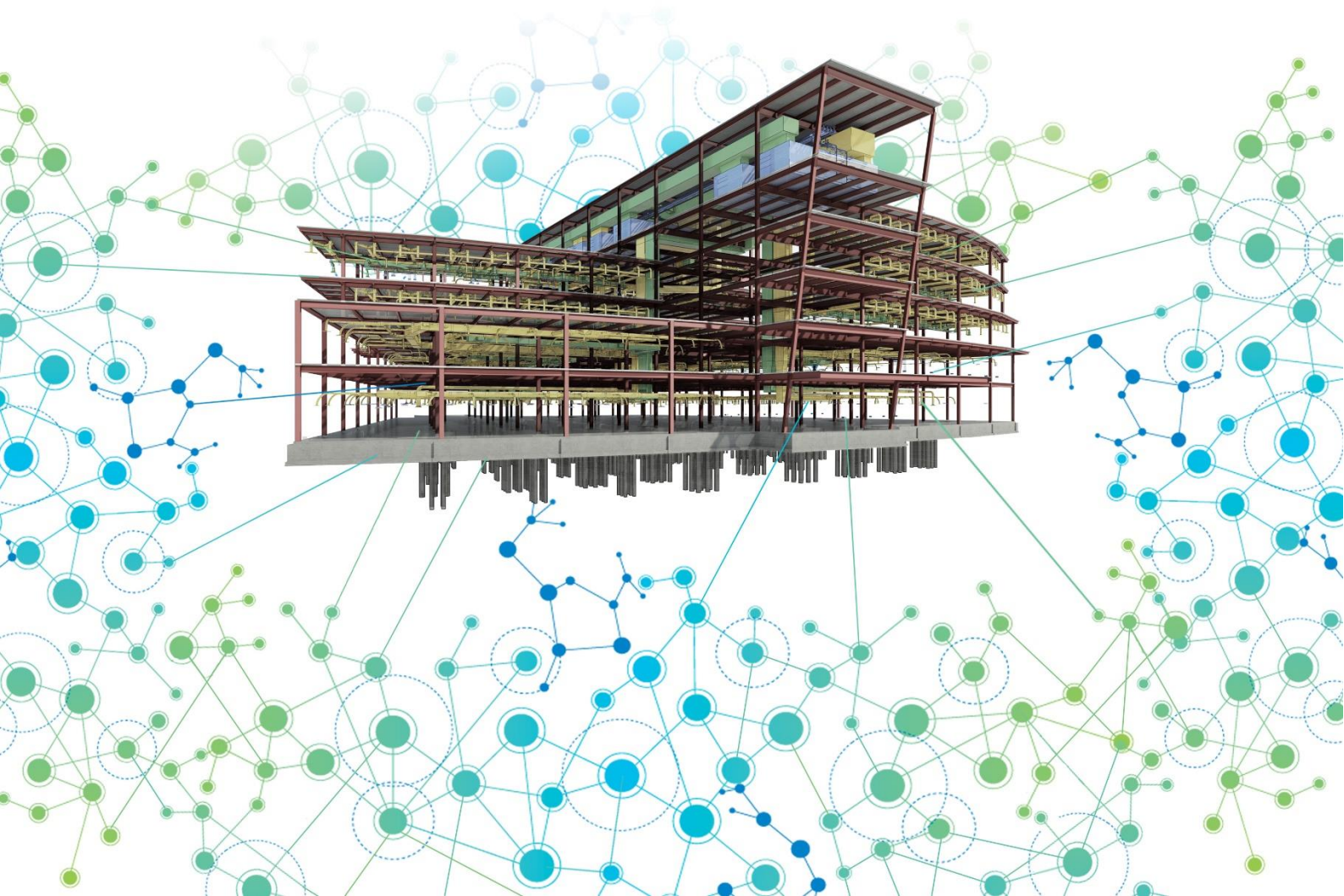
Improving the management of structural engineering requirements in the design phase

Linking project requirements to BIM, based on the Semantic Web and Linked Data principles

Author	Miryana Hristova Stancheva
E-mail	miryana.stancheva22@gmail.com
Student number	0978091
Date	24-09-2017
Date of presentation	27-09-2017

Institution:

University	Eindhoven University of Technology (TU/e)
Faculty	Faculty of the Built Environment
Master's track	Construction Management and Engineering (CME)

In collaboration with:

Company	Ingenieursbureau Verhoeven en Leenders B.V.

Graduation Committee:

Chairman	Prof. Dr. ir. B. (Bauke) de Vries
1st Supervisor	dr. dipl.-ing J. (Jakob) Beetz
2nd Supervisor	T. (Thomas) Krijnen
3rd Supervisor	Ing. L.C.M. (Leon) Leenders

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Preface

I am happy to present my graduation thesis carried out in collaboration with the *Eindhoven University of Technology* and *Ingenieursbureau Verhoeven en Leenders*. After six months of hard work for completing my project, my Master's study in Construction Management & Engineering and my time being a student have come to an end.

By conducting a research in the fields of Building Information Modeling (BIM), requirements management, Linked Data and Semantic Web technologies, I chose a topic completely unrelated to the knowledge and academic background that I had before. For that reason, due to the fact that I chose to step out of my comfort zone and challenge myself, I recognize the great leverage that this research gave me from the perspective of learning and developing myself in the field of information management which will definitely be a great asset to my future career.

However, I could have never succeeded with my research without the help and guidance of several people who have supported me throughout the process. First of all, I would like to thank my main supervisor Jakob Beetz (TU/e) for giving me the interest and motivation to choose a BIM/Linked Data-related topic, for keeping me on track with my project, and for constantly helping me to improve my work. In addition, I want to thank my second supervisor - Thomas Krijnen (TU/e) for the technical support and for assisting me in resolving all programming issues that I had and which, as a complete programming newbie, I couldn't have managed on my own. I want to also thank my third supervisor - Leon Leenders (Verhoeven & Leenders) for the opportunity to graduate at his company, for the valuable discussions we had, and for the responsiveness and enthusiasm shown from his side throughout the process.

Last but not least, I want to thank all the employees at Verhoeven en Leenders for being quick to react each time I had questions or needed information and especially, Lucas Verhelst, who was also involved in the project and helped to keep our triplestore server up and running. I'm also grateful to all interviewees for taking the time to provide me with valuable insights into the current data management struggles within the AEC industry. To my fellow co-graduate and friend Francisco Bernal (a.k.a the G.G.), with whom we partially collaborated, I want to say thanks for having the patience (quite frankly, the bravery =) to work with me and for sharing all thesis struggles together. Last, but not least, I want to thank my family for the incredible support system they always are!

Miryana Stancheva
Eindhoven, September 2017

Table of Contents

Summary

The deep fragmentation between domains in the AEC industry has caused knowledge and information to become dispersed and also often untraceable. The unstructured nature of data management has caused knowledge gaps which currently are tackled through improved collaboration by the use of technologies such as BIM. Despite these efforts, however, the issue with information ambiguity and misinterpretation remains due to the variety of vendor-specific applications and the incompatibility between them (Harrison et al., 2003).

Although the development of the IFC open standard intended to eliminate the interoperability limitations arising from the use of different software packages by creating one neutral format for describing building information, it is far from solving the problem in its entirety. Due to the fact that almost always no exact mapping between the IFC description schema and the schema of the software application used is possible, complete data interoperability cannot be achieved (Pazlar and Turk 2008).

One of the issues resulting from the lack of complete interoperability within the AEC industry is the inadequacy in handling project requirements and the lack of integration between all requirement-relevant datasets. For the structural engineering practices and more specifically, for the engineering of building structures, the management of project requirements is conducted in an inconsistent manner. Interviews with experts were held in order to receive a more in-depth understanding concerning data integration processes within the industry and the problems resulting from the lack of them, confirming the still primarily paper-based management of information within most companies, as well as the inability of BIM to support data integration.

The information generated by the engineers in the design phase of a building structures project, which is also closely related to project requirements, has been kept in separate documents (primarily PDFs) without the existence of any proofs as to whether all instances of the designed structure satisfy the corresponding conditions. In addition to that, a link between the 3D model instances and the generated information (calculations, advisory reports, estimations etc.) does not exist, making the traceability of requirements inefficient and prone to errors. Therefore, the scattered and inconsistent nature of the requirements management practices in the engineering of building structures bears the risks of loss of valuable process-related information and of not complying with the initially defined project objectives. For that reason, the work of other parties who use the engineering information generated during the design phase as their main input can be also negatively impacted.

Therefore, the building structures project domain and more specifically, the final phase of the structural engineering design represent the main focus of this thesis project. From the data created by the engineers during the design phases, special attention is given to the required information for the validation of the structure's stability in the project handover to external parties such as sub-contractors, manufacturers and audit commissions.

This thesis project has two general objectives the first of which is to formalize the relationship between the structural design components and the general engineering requirements which relate to them. The second objective of the thesis is to explore the possibilities for mapping project requirements to object instances from the BIM model by relying upon the information generated from the first objective. Subsequently, the second objective also includes proving the compliance of the design with the requirements by linking them to the project's

documentation. In accordance with the set goals, the research is divided into two phases, the research on engineering requirements and the tool development phase.

Firstly, a research was conducted for the formalization of general engineering requirements fundamental in a building structures project which also constitute the core of the final engineering design handover to external parties. The findings from this initial research were represented in the form of a general requirements matrix. The matrix captures the general engineering requirements on structural elements which need to be proven at the end of the final design phase of a building structures project.

The method selected for achieving the second objective of the thesis project rests in the use of the Semantic Web technologies which especially in combination with open standards, have been deemed by researchers such as Abanda (2013) to present the opportunity of building a strong foundation for the efficient management of information and knowledge in the different domains of the built environment.

Therefore, as the main input for the second phase of the research, which focuses on the development of a tool enabling the mapping between engineering requirements and model components, the formalized requirements matrix was translated into an ontology based on the Semantic Web and Linked Data principles. The prototype interacts with a triplestore server where both the requirements ontology and the converted into RDF format IFC model are stored. It visualizes not only the IFC model loaded into the application, but also displays the RDF requirements data related to the individual design components by matching the GUIDs of the IFC file to the GUIDs of the converted geometry from the triplestore.

For confirming that a certain object instance from the model has been verified against a specific requirement, the options to semantically relate document proofs to the component and relate them to the requirement(s) that they prove for that component are facilitated by the tool. The newly created information is stored in an RDF format in the triplestore server. Once all elements in the model have been proved in relation to all requirements they must fulfill, the information necessary for the design handover to external parties can be considered complete. In this way, the three main information sources for the continuation of the project – geometry, documentation, and requirements are explicitly linked to each other.

When looking into the main conclusions that can be derived from the conducted research, they can be presented in twofold – the advantages which the semantic technologies can bring to the data handling practices within the AEC industry and the way in which requirements should be regarded and managed for achieving complete integration of information.

Several factors contribute to the added value of semantically describing a particular set of domain knowledge – the extensibility and reusability of data, the ability to retrieve search-specific sets of information and the software-independent nature of data handling. As semantics bring along the opportunity to link any concept of any knowledge domain to another concept, or a number of concepts from a different knowledge domain in an explicit and unambiguous way, the issue of combining different contents from the various project stakeholders can be tackled and thus, presents a prerequisite for solving the interoperability issues within the AEC industry.

In addition to the aforementioned, a semantic dataset can be easily extended by the addition of more concepts and relations while at the same time preventing repetition and redundancy of definitions. As the building industry is often described as data-intensive, the management and the retrieval of information become demanding and error-prone tasks. By storing data in an RDF format, however, the partial retrieval of only search-relevant information is enabled.

The management of requirements in the context of this research was considered from the limited perspective of the design processes in the building structures domain for the design handover to external project stakeholders. Therefore, project requirements were regarded in isolation from their predeceasing specifications in the form of soft client requirements from the elicitation phase and in the form of architectural/functional requirements from the design exploration phase. As the structural engineering requirements are a derivative from the aforementioned two, for the purposes of interoperability between all project stakeholders the bigger picture of requirements management practices needs to be considered when implementing it in an actual project so that all parties can have the overview of how their domain-specific requirements influence the requirements of the other project parties.

Abstract

In the building industry requirements are the basis of a project and they address general aspects such as overall goals, activities, and needs of the client, as well as very detailed characteristics of the product such as materials or special conditions. Therefore, the lack of established requirements management practices within a project bears many risks, amongst which the of loss of valuable information and the risk of not complying with the initially defined project objectives. When looking into the structural engineering field, and more specifically into the domain of building structures, the inefficiency in the design handover processes becomes apparent and it can be attributed to both the absence of practices related to requirements specification and record, as well as to the disjointedness between geometry, project documentation, and requirements. Therefore, this thesis firstly investigates which engineering requirements are fundamental in a building structures project and constitute the core of the final engineering design handover to external parties. Secondly, the thesis explores the possibilities of achieving a certain level of integration between requirements data, 3D modeling data, and project documentation for the validation of the design's conformity with the requirements. Due to the inability of the BIM/IFC approach to fully support the collaboration processes within a project and to provide explicit connections between the aforementioned information sources, a Linked Data approach for targeting the shortcomings of the current management practices has been implemented in the form of a desktop viewer. Based on a previously formalized requirements ontology, the tool enables the semantic mapping between engineering requirements and object instances from the 3D model. It provides the functionality of proving the compliance of model instances with the requirements by creating semantic connections to the calculation reports which justify the objects' conformity with the requirements. As the created solution represents an initial effort for requirements formalization and for interoperability of requirements-relevant data for the engineering design phase of a project, it also creates a prerequisite for future research into the possibilities for a collective approach to the management of requirements, involving all design parties in a building structures project.

List of figures

List of tables

List of listings

List of abbreviations

2D	: two-dimensional
3D	: three-dimensional
AEC	: Architecture, Engineering, and Construction
BCF	: BIM Collaboration Format
BIM	: Building Information Model(ing)
CAD	: Computer Aided Design
COBie	: Construction Operations Building information exchange
COINS	: Construction Objects and the INtegration of Processes and Systems
GUID	: Globally Unique Identifier
IAI	: International Alliance for Interoperability
ICDD	: Information Container for Data Drop
IDM	: Information Delivery Manual
IFC	: Industry Foundation Classes
IFD	: International Framework for Dictionaries
ISO	: International Organization for Standardization
LOD	: Level Of Detail/Development
MVD	: Model View Definition
OWL	: Web Ontology Language
RBS	: Requirement Breakdown Structure
RDF	: Resource Description Framework
RDFS	: Resource Description Framework Schema
SBS	: System Breakdown Structure
SE	: Systems Engineering
SPARQL	: SPARQL Protocol And RDF Query Language
URI	: Unique Resource Identifier

PART A: PROBLEM DEFINITION AND RESEARCH METHODOLOGY

This section contains the general introduction to the thesis project. Chapter 1 provides information on the background and context of the research, while chapter 2 outlines the research problem, the research questions, the scope and the design of the project, the expected results and the scientific importance of the thesis. Furthermore, section 2.6 introduces the reader's guide.
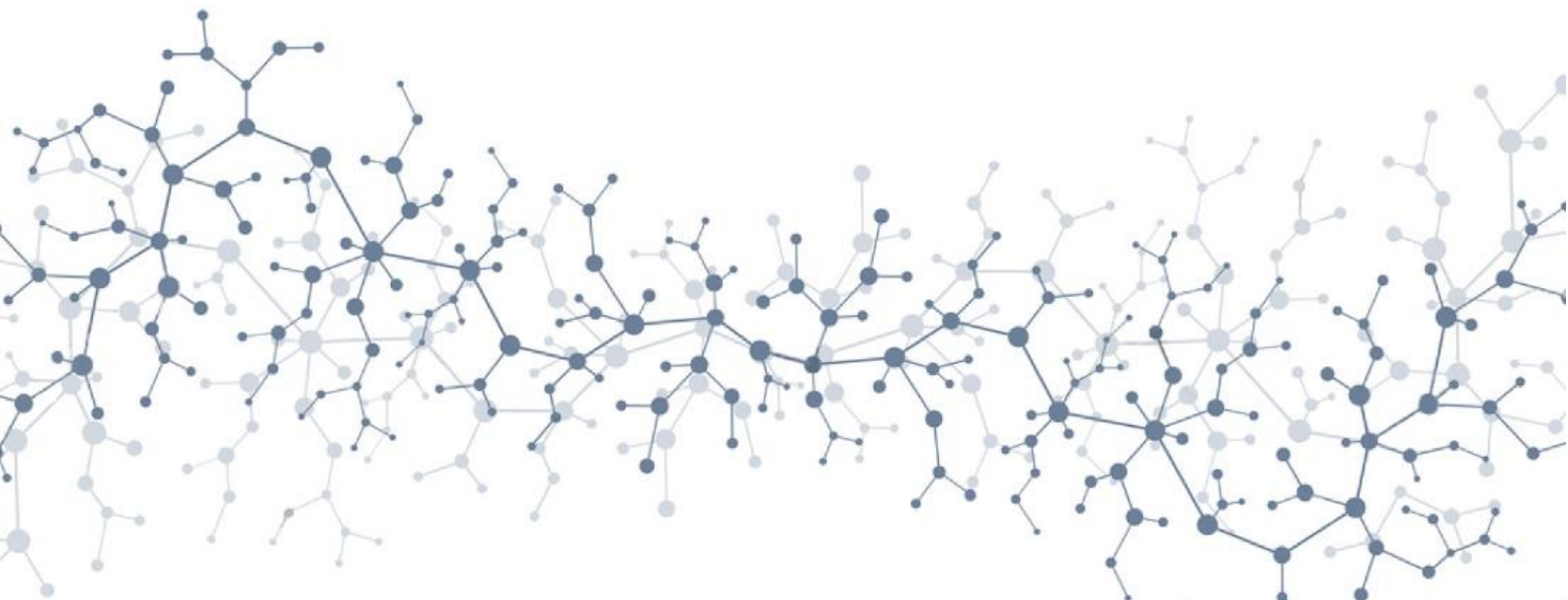
1   Introduction

The processes in the AEC (Architecture, Engineering, and Construction) industry are a complex endeavor, which involve not only a wide variety of stakeholders, but also many diverse activities. Regardless of the size of the project, the collaboration between a large number of organizations including clients, architects, engineers, advisors, quality surveyors, contractors, and subcontractors is inevitable and therefore, leads to the fragmentation of design, engineering and construction practices (Steel, Drogemuller, and Toth 2010).

The AEC industry is fragmented not only due to the various parties involved in a project and their diverse professional domains but also due to the different stages a project goes through until its completion. The different project delivery phases usually involve different stakeholders who, in many cases, don't directly interact with one another but nevertheless, need to build upon the results of the previous party. Therefore, the information management and data sharing practices are of great importance not only for the efficiency of the processes but also for the mitigation of project-related risks. Some of the negative outcomes due to inadequate information management practices could entail technical issues with the structure and the design, cost and time overruns, material waste due to unnecessary rework, etc.

In order to overcome some of the setbacks related to the fragmented and data-intensive nature of the AEC sector, the use of BIM has been implemented not only as a tool in the design process but also as an interface for the exchange of information in later project phases. In comparison with the traditional data exchange practices in the form of drawings and documents, the shift towards digitalization presents the opportunity of improving the communication between the actors involved in a project. However, the use of BIM has brought with itself also some interoperability challenges emerging from the broad variety of vendor-specific tools used by the different teams. The incompatibility of these software platforms often presents an obstacle for model exchange (Steel, Drogemuller, and Toth 2010).

The Industry Foundation Classes (IFC), a vendor-neutral standard for the exchange of building models developed by the buildingSMART alliance, was created as a response to the interoperability issues of the BIM tools. While the standard's schema is an object-oriented data model which incorporates concepts, such as object and attributes, relationships, property sets and quantity definitions, it does not support the management of project requirements. The internal structure of the existing design software also bears the same limitation (Kiviniemi 2005).

As structures are designed to fulfill demands of users, clients, and society, many of these demands are expressed in the form of requirements through building codes, regulations, standards, and client specifications. The requirements of a project, therefore, create the basis for the decisions taken during the design processes, which, on their part, determine the final design outcome. When the management of requirements lacks transparency, it can lead to nonconformity of the design to the project's requirements, resulting in design iterations and rework (Jansson et al.,2013). The lack of a proper requirements management framework, however, also bears the risk of miscommunication during the information handover to external parties between the design and construction phases. Quite often some of these parties (such as subcontractors), who take over from the design teams, are not involved in the project from its initiation. Therefore, in order to prevent miscommunication and failures during construction, these stakeholders need to be able to trace back the relationships between design and requirements in a clear and unambiguous manner.

2   Research approach

In order to provide an overview of the project, this chapter firstly introduces the problem definition, the research questions and the research design that have directed the research process. Secondly, the chapter elaborates upon the expected results of the research and its scientific importance.

2.1   Problem definition and research gap

In the current building design processes, requirements are not documented in a coherent manner and are quite often stored either in the meeting minutes or only in the minds of the team as tacit and implicit knowledge. As requirements evolve over time, the corresponding changes are difficult to find due to the lack of proper documentation and the lack of ways to unambiguously trace the relevance of a requirement to the different design components. This situation, in addition to aspects such as long project durations, changes of personnel, and the human inability to remember details, leads to significant loss of information and thus, raises the risk of nonconformity to requirements in later project stages (Kiviniemi 2005).

The structural engineering domain can be conceptually divided into two branches, namely, a civil branch and a building structures branch. While the former one follows the governmentally-imposed information management practices such as Systems Engineering (SE), the latter one barely implements any type of information management system for the traceability of data and processes. The main reason for this division rests on the fact that in the most of the cases the civil projects are initiated by governmental institutions such as municipalities. The building structures branch, however, deals primarily with a wide variety of private clients where standardization of project practices often does not exist. The general project workflow and information management systems are determined either by the client or by the leading stakeholder in the project, who is determined by the type of contract.

Many tasks associated with information management and SE such as clash detection, verification, issue control, and requirements management are, in fact, already being performed by the engineering teams also in building structure projects. However, all of the aforementioned are not executed in a logical and targeted manner and in addition to that, they are also poorly documented. Project requirements for building structure projects and their evolution throughout the project's lifecycle are not recorded in any shape or form, making it difficult to trace back issues occurring in later project phases.

The data generated by the engineers in the design phase, which is also closely related to project requirements, has been kept in separate documents (primarily PDFs) without the existence of any proofs as to whether all instances of the designed structure satisfy the corresponding conditions. In addition to that, a link between the 3D model instances and the generated information (calculations, advisory reports, estimations etc.) does not exist, making the traceability of requirements inefficient and prone to errors. Quite often there is no register kept, neither on the specifics of a check or model iteration nor of the decision-making process that has led to a certain result. The tradeoff matrix, where it is proven that due to a specific reason, a certain solution should be reevaluated and the documentation related to it - updated, is also currently not been documented.

The lack of traceability impacts not only the efficiency of the engineering team but also the effective communication with external parties. In the information handover process at the end of the design phase, all of the produced information needs to be delivered to various sub-

contractors. They need to unambiguously identify the conditions to which each component they are responsible for delivering or manufacturing should comply with. In addition, the engineers have the responsibility to validate in front of other parties, such as the client and the external audit, that all physical components of the structure have been considered and all aspects associated with them – calculated and thus, proving that the structural engineering requirements have been satisfied. Therefore, without the use of a formalized approach to the management of engineering requirements, these processes become susceptible to miscommunication and errors.

To conclude, the scattered and inconsistent nature of the requirements management practices in the engineering of building structures bears the risks of loss of valuable process-related information and of not complying with the initially defined project objectives. The inefficiency in the design and handover processes also indicate the need for achieving a certain level of integration between requirements data, 3D modeling data and project documentation in order to improve the traceability of information.

2.1.1 Research scope

Considering the broadness of the previously introduced problem, defining the scope of the research is necessary. As presented in *Figure 1* below, the research focuses on the Building Structures project domain and more specifically, on the design phase of the structural engineering process. In order to identify the relations to other stakeholders (such as architects, contractors, and the client) and their influence on the engineering information, the research will entail evaluating the design process from a broader perspective. From the data created by the engineers during the design phases, special attention will be given to the required information for the validation of the structure's stability in the project handover to external parties.
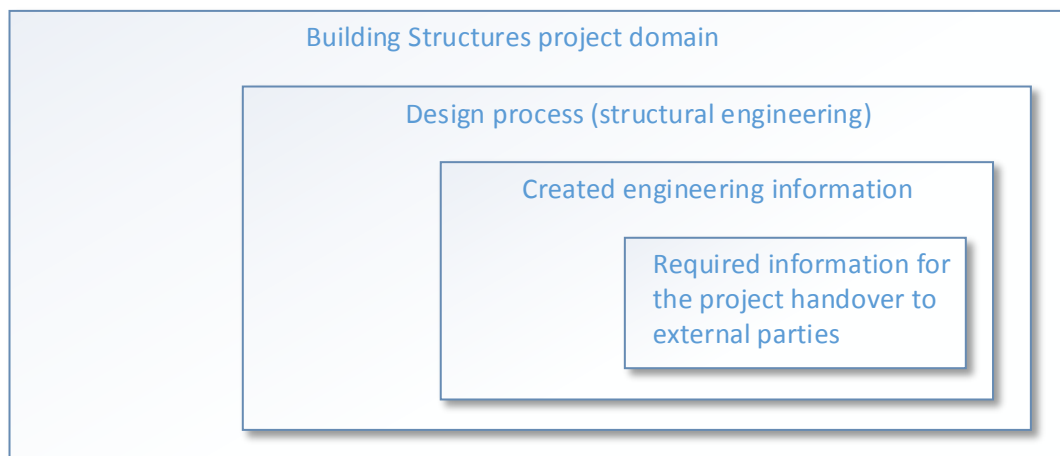
Figure 1: Research scope

2.2 Research questions

The main research question of this project, which has been derived from the problem definition presented in *section 2.1* and from the research scope presented in *section 2.1.1*, is the following:

How can a mapping between project requirements, design and documentation be created for the validation of the design's conformity with the requirements and for improving the traceability of information?

In order to support the main research question, several sub-questions have been identified:

1. Which engineering requirements are fundamental in a building structures project (general requirements vs. project-specific requirements) and constitute the core of the final engineering design handover to external parties?

2. What are the levels of detail/development and on which LOD is it most beneficial to connect engineering requirements data with the BIM model (e.g. building storeys, spaces, elements, systems of elements)?

3. How can structural engineering requirements be mapped to all object instances from the geometry model to which they relate to?

4. How can the conformity of a particular model instance with its associated requirements be proven and the information reused also in later project delivery phases?

5. What is the added value of semantically linking requirements and design and can this adjustment lead to a higher efficiency in the design process due to the mitigation of risks associated with design non-conformity with requirements?

2.3   Research design

This section provides an outline of the research model, graphically displayed in *Figure 2* below, which will be used as a framework for the thesis project. In the model, the overall process is divided into three general stages, namely, the preparatory stage, the development stage, and the reporting stage.

The preparatory stage incorporates two main tasks - the literature research and the research on the in-house practices of the company. Each of these tasks is divided into sub-tasks, which follow no particular order. The literature research focuses on topics such as requirements management, BIM and 3D modeling, Semantic Web and Linked Data, as well as the current data integration practices in the AEC industry. At the same time, the information management and 3D modeling practices of *Verhoeven en Leenders* will be also explored in order to gain a better insight into the processes of the company. Special attention will be paid to assessing the modeling workflow in the building structures division and the importance of different project requirements, as they will be the major input for the development stage of the thesis.

Once the preparatory stage is completed and the conclusions are drawn, both the qualitative research and the prototype development can take place. Each one of these activities is divided into sub-tasks as well, with the difference that this time they follow a particular order. The qualitative research consists of conducting interviews with domain experts and the subsequent processing and analysis of the data. The prototype development firstly involves setting a feasible scope of the functionalities it can provide. Secondly, a system for mapping requirements data and a system for querying the mapped data should be designed. Thirdly, an approach validation, based on a use case scenario will be carried out. Conclusions from both the qualitative research and the prototype development are drawn and the last stage of the research model, namely the reporting stage, will outline all processes, results, and findings.
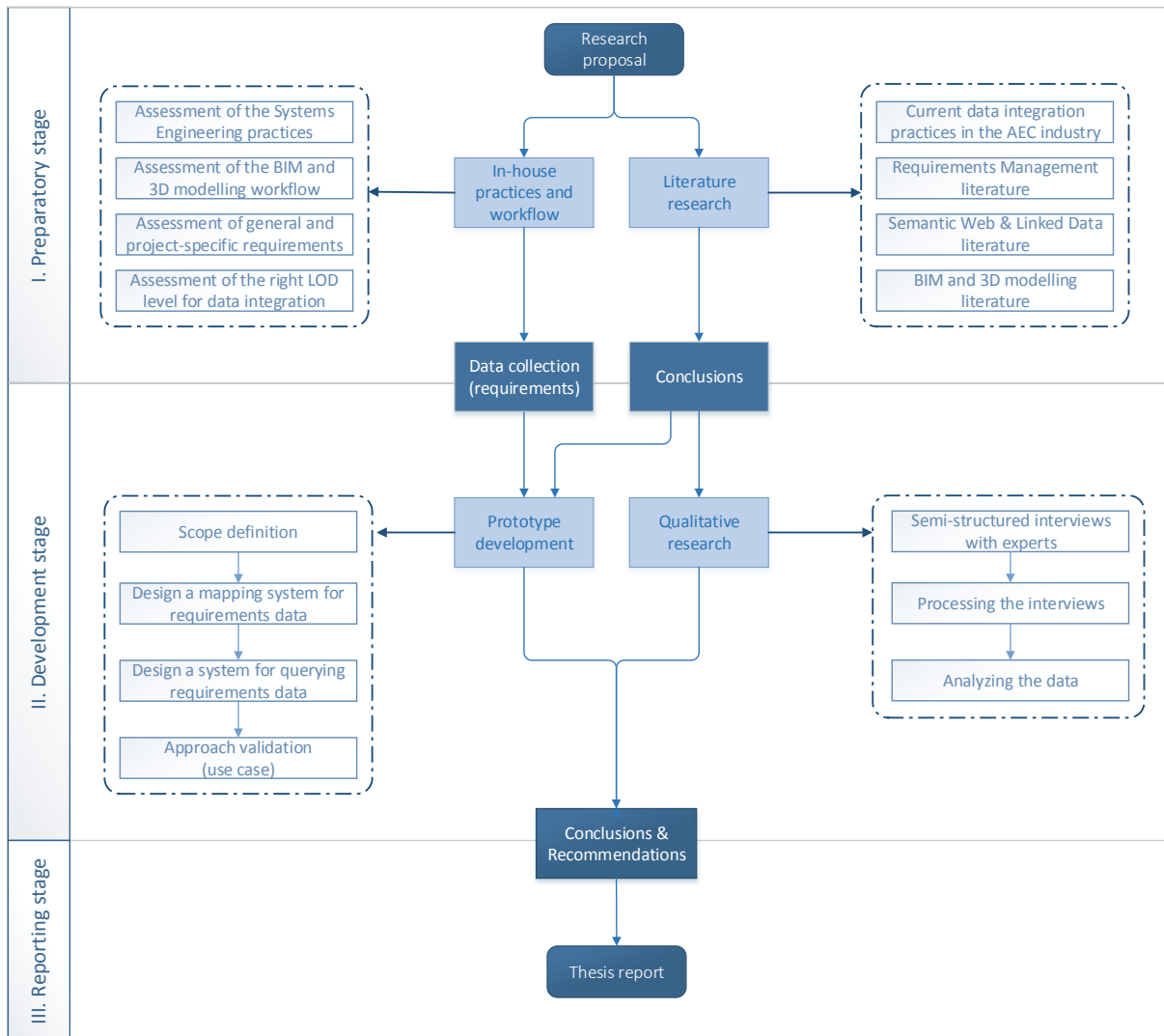
Figure 2: Research model

2.4   Expected results

This thesis project has two general objectives the first of which is to formalize the relationship between the structural design components and the general engineering requirements which relate to them. The second objective of the thesis is to explore the possibilities for semantically connecting project requirements and object instances of the BIM depending on the input information from the first objective and subsequently, proving their validity.

Based on the aforementioned objectives, this research also aims at providing an answer to the central research question of the thesis, as well as to the sub-questions derived from it. Therefore, the expected results come in threefold and namely, conclusions from both the literature review and from the investigation of the company's in-house practices, conclusions from the interviews with professionals in the AEC field, and a prototypical tool for mapping requirements to object instances in a BIM model.

2.4.1   Expected results objective 1

From the preparatory phase, schematically displayed in the research model in *Figure 2*, a clear evaluation of the current practices of information management within the company must be achieved. In addition, the different categories of requirements, as well as the level of detail at

which they must be integrated within the model must be estimated. For that, the literature review will provide a good theoretical basis on topics such as BIM, IFC, Semantic Technologies and requirements management. The in-house research should result in a formalized schema presenting the general engineering requirements from the design phase of a building structures project. This schema will contain the input information for the second phase of this research.

Afterward, as indicated in the research model, the interviews with experts will be held in order to receive a more in-depth information concerning data integration practices and the need of direct linking between project requirements and building information models. In addition, the interviews must provide an expert opinion on the expected impact of such linking on the data management practices, on the efficiency of design processes and on the mitigation of conformity-related risks. Thus, the interviews are useful for gaining a better insight into the problems resulting from the lack of proper data integration. They can, therefore, confirm or deny certain claims and even bring out some new aspects to the topic.

2.4.2 Expected results objective 2

The third part of the expected results has to do with the development of a tool, which is able to link engineering requirements data with model instances by the means of adding semantic definitions to different objects in the model. The mapping between requirements and object instances must be based on one-to-one and one-to-many relations and must also be reusable and generic at least within the context of the company's practices.

Additionally, the tool must allow the model to be queried in a bidirectional way, based on the input from the mapping. This would suggest that the model can be queried either for a specific requirement with the result of retrieving all object instances related to it, or a specific object instance can be queried for the requirements it must conform to. Besides that, the tool should provide the possibility for attaching document proofs to a specific element (or groups of elements) in relation to the requirements the document proves for that object instance. With the aim of obtaining practically relevant results and validating the undertaken approach for data integration a case study will be conducted.

2.5 Relevance of the research

The importance of this research has been evaluated based on both its social (practical) relevance and its scientific relevance.

2.5.1 Social relevance

From a social perspective, this graduation project contributes to improving the information management practices of *Verhoeven en Leenders B.V.* as it creates a formalized overview of the must-have information for a project handover to external parties. The developed prototype has the potential of being implemented in the management of engineering requirements allowing for the traceability of requirement proofs and facilitating the interrelation between documentation and BIM object instances.

In addition, both the formalization of requirement data and the tool development present a way of official requirement validation which has the potential to become mandatory in the Netherlands from the beginning of 2018 due to a new law. The law currently awaits approval from the second chamber of the Dutch parliament and it is named 'Kwaliteitsborging voor het bouwen' (English: Construction quality assurance). In the case of an approval, the law would

demand from the engineering teams in a project to present proofs that the design and its components comply with the engineering requirements listed in the building codes.

Therefore, from a social perspective, the thesis contributes not only in terms of improving the efficiency within the company but also helps to find a practical solution for the integration of design and engineering data for quality assurance in the future.

2.5.2 Scientific relevance

From an academic perspective, this graduation project contributes to the existing literature through the concrete focus on requirements management and formalization for the engineering design processes of building structures. Furthermore, the thesis explores the concept of semantically linking requirement's data with objects from the building information model for the purposes of information traceability and proof of requirement compliance.
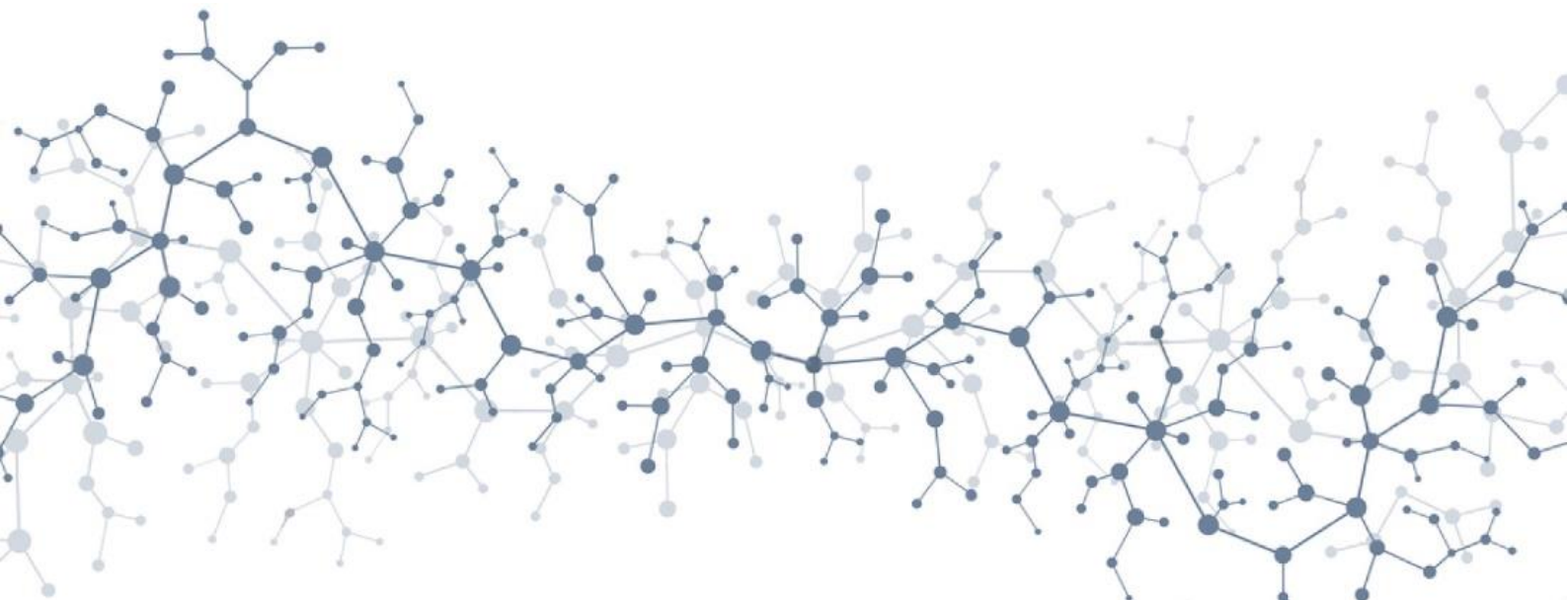
The aforementioned two aspects make the research relevant from a scientific perspective and strengthen the existing literature on requirements management and semantic enrichment of building information models by contributing to the concept of managing project requirements through the use of Linked Data principles.

2.6 Readers' guide

The report is organized into five main sections conceptually dividing the different parts of the thesis. Each of the sections contains several chapters, which, on their side, are composed by sub-chapters. **PART A:** PROBLEM DEFINITION AND RESEARCH METHODOLOGY of this report outlays the general introduction to the thesis, while **PART B:** LITERATURE REVIEW presents the literature research conducted on topics relevant for carrying out the thesis project. **PART C:** QUALITATIVE RESEARCH introduces the qualitative research, or expert interviews, held with the aim of validating the research problem of the thesis. Subsequently, **PART D:** METHODOLOGY elaborates on the method of the research which consists of two chapters and namely, the theoretical research on general engineering requirements and the development of a prototype. In the end, **PART E:** CONCLUSION AND DISCUSSION presents the conclusions and provides recommendations for future research.

PART B: LITERATURE REVIEW

This section provides the background and context of the research by elaborating upon relevant topics, which have induced the motivation for carrying out this thesis project. Chapter 3 gives an insight into the collaboration practices in the AEC industry, BIM, and the IFC standard, while in chapter 4 the topic of requirements management and traceability is elaborated on. Chapter 5 introduces the idea of Linked Data and the Semantic Web with their underlying technologies. In section 5.3 some of the limitations of BIM and the IFC schema are presented in order to give a better insight into the benefits of semantically enriched building information models. At last, in section 5.7 the conclusions from the literature research are drawn.

3   Building Information Modeling and the collaboration process

The AEC industry is a domain, characterized by the high diversity of stakeholders, participating and thus cooperating in the design, construction, and maintenance of a single construction project. Throughout the years the approaches to project collaboration have been drastically changing due to the realized need of a shift from the traditional way of working, implying a segregated, sequential process, where each professional domain works in isolation, to a more integrated process encouraging interoperability. Building Information Modeling (BIM), being one of the most promising technologies enabling this transition, has helped the industry to move forward by replacing the 2D drawing-based information exchange between domains with a 3D model-based approach, serving as a prerequisite for creating a collaborative working environment.

3.1   Collaboration in the design process

The design collaboration in its essence rests on the idea of each design party being able to integrate into their own work the design solutions of the other professionals involved in the process. Besides that, good collaboration suggests that each actor is able to properly evaluate the impact which the rest of the domains have on his/her solution and define accordingly the conditions for achieving quality of integration by discovering errors or proposing improvements. Therefore, the most important elements for achieving an effective collaborative environment are knowledge sharing management (Simeone and Cursi, 2016).
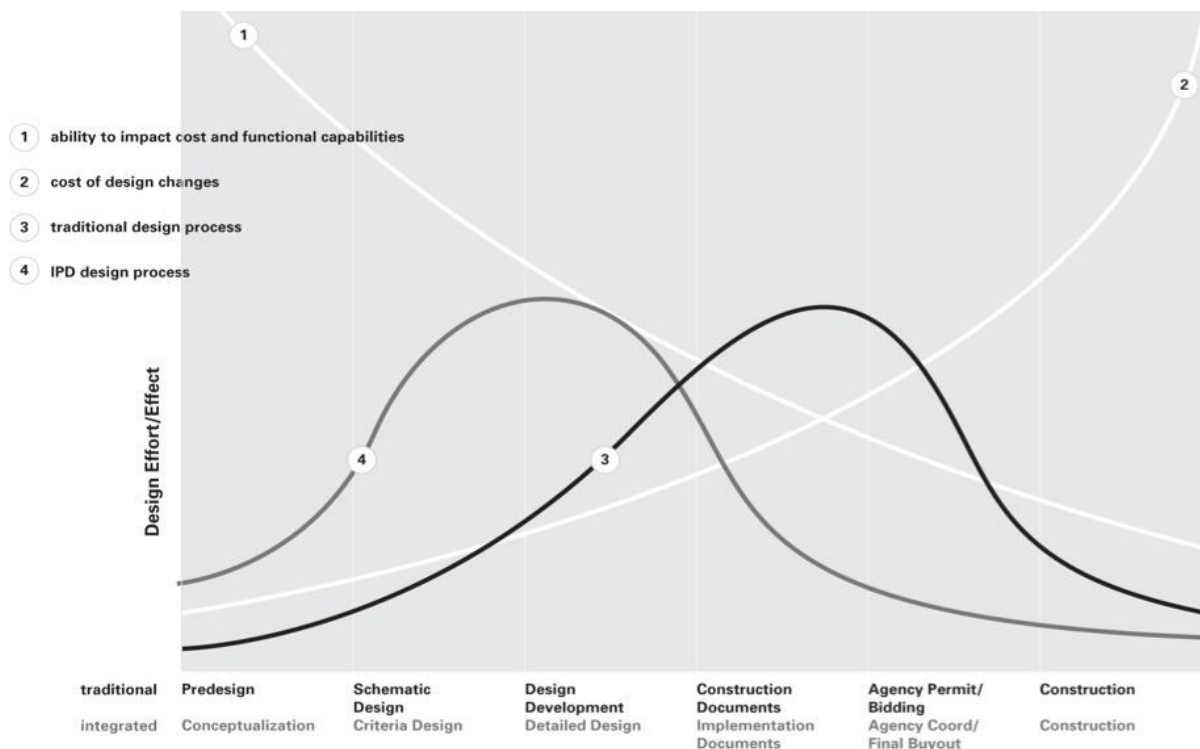
Figure 3: The MacLeamy curve depicting the traditional and integrated workflow (Lu et al. 2015)

Figure 3 above represents the MacLeamy Curve comparing the traditional project workflow with the BIM-centered workflow, as well as the implications resulting from them. It clearly displays the superiority of the latter in terms of overall project performance and therefore, highlights the importance of using BIM in the design phase.

The building industry has been often criticized for being wasteful, inefficient and fragmented due to the traditional, draft-centric way of working, where each design team is focused their own expertise within a project, without considering the project as a whole. As a way to address these inefficiencies, the AEC industry has started adopting a more integrated way of working, based on utilizing Building Information Modeling in order to encourage collaborative working and interoperability of project information (Serginson, Mokhtar, and Kelly 2013).

The benefits of Building Information Modelling for the AEC industry are related to facilitating practices such as stakeholder coordination (coordination of interfaces), better knowledge transfer and predictability of outcomes. These aspects subsequently lead to a higher factor of error detection (especially in the early design stages), mitigation of risks, and minimization of costs which otherwise might lead to budget overruns. BIM improves the efficiency of work, reduces waste, increases the value and the quality of the building projects and its integration on-site facilitates faster problem-solving (Azhar, S. and Hein, M. 2014).

3.2   Introduction to Building Information Modeling

The official definition of Building Information Modelling proposed by buildingSMART and the Royal Institute of British Architects (RIBA) describes BIM as a "digital representation of physical and functional characteristics of a facility, creating a shared knowledge resource for information about it, and forming a reliable basis for decisions during its life cycle, from earliest conception to demolition" (Abanda, F. H. 2013).

Another source defines BIM as "the process of generating, storing, managing, exchanging and sharing building information in an interoperable and reusable way", where all project stakeholders are able to provide their knowledge as an input which can be reused throughout the entire life cycle of the building. This suggests that the building information model represents "a cooperative system of unified views of the same building", where each individual view corresponds to the information of the specific professional domain it represents (e.g. MEP, maintenance), necessary for the correct realization of the corresponding tasks and processes (Farias, M. T., Roxin, A., and Nicolle, C. 2014).

As 2D CAD represents a building by its autonomous views, such as floor plans and sections, composed solely of graphical entities such as arcs, lines, and circles, in a BIM model the same objects are defined as building elements and systems. When a design change occurs, in the case of 2D CAD drawings all views, or drawings, need to be altered accordingly, whereas, in BIM, the alteration happens only once - in the 3D model. Therefore, a BIM model incorporates all information related to a building in series of "smart objects" (Azhar, S. and Hein, M. 2014).

3.3   BIM Maturity Levels

The BIM Maturity Model, displayed in *Figure 4* and further clarified in *Table 1* below, portrays the different levels of growth, or levels of computerization of processes, throughout a building project's life cycle. The model has been initially created by the British government and has been globally adopted as means for determining the degree of IT integration in the building process (Bouw Informatie Raad 2010).

Table 1: BIM maturity levels

BIM Level 0	is entirely document oriented, which suggests the use of 2D CAD drawings and other 'non-intelligent' ways to exchange information such as calculations in Excel.
BIM Level 1	is object oriented and also the first step towards properly implementing BIM as it involves working with 3D objects. In this way, the potential of linking external information, such as financial calculations or planning schedules exists but is not yet implemented.
BIM Level 2	is the collaboration level or also called '*little BIM*', at which one view model combines separate databases, each associated with their own model. Therefore, file based collaboration, as well as the possibility to link planning (4D) and cost calculations (5D) is enabled.
BIM Level 3	is the integrated level at which exchanges are only object based and can occur between different organizational entities by the means of an integrated web environment. The level is also called '*big BIM*' and information in the integrated environment continues to be shared in the operations and maintenance phase of the structure, or in other words - throughout the entire project life cycle.
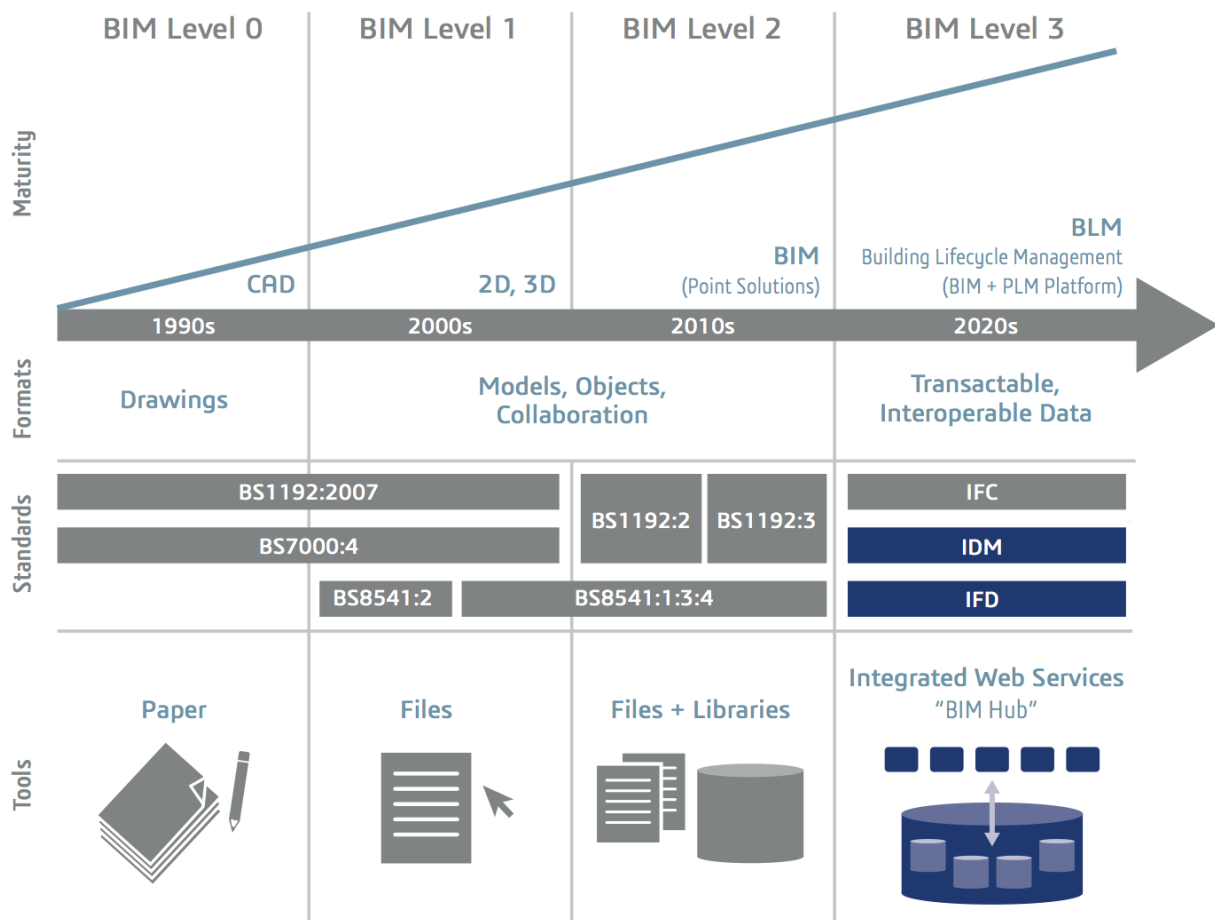
Figure 4: BIM maturity model by Mark Bew and Melvin Richards (Akio 2017)

3.4   BIM collaboration formats

The BIM collaboration formats allowing the exchange of various design, construction, and operations-related data can be divided into two categories – proprietary formats and open formats. Some examples for proprietary formats are Autodesk's RVT (Revit), NWD (Navisworks) and DWG (AutoCAD). The Industry Foundation Classes (IFC) and the Construction Operations Building Exchange (COBie) are examples of non-proprietary collaboration formats.

While the proprietary formats are developed by big software vendors and are usually readable and executable by their own software applications (and other permitted ones), the open BIM formats are vendor-neutral and can be read and edited by any software application (designingbuildings.co.uk 2014). Open formats are therefore essential for the successful collaboration within the building industry as they facilitate the interoperability between the different professional domains in a more efficient way.

The drawbacks of interoperability based on proprietary formats reside in the fact that each individual software application must develop and implement direct back and forth translators for all other target formats that it is meant to exchange data with. With the open interoperability standard, however, regardless of the type of exchange (file-based or server-based), the back and forth mapping is done from and to one single format, compatible with all other software applications (Kiviniemi, A. and Laakso, M. 2012). *Figure 5* below depicts the two conceptual scenarios.
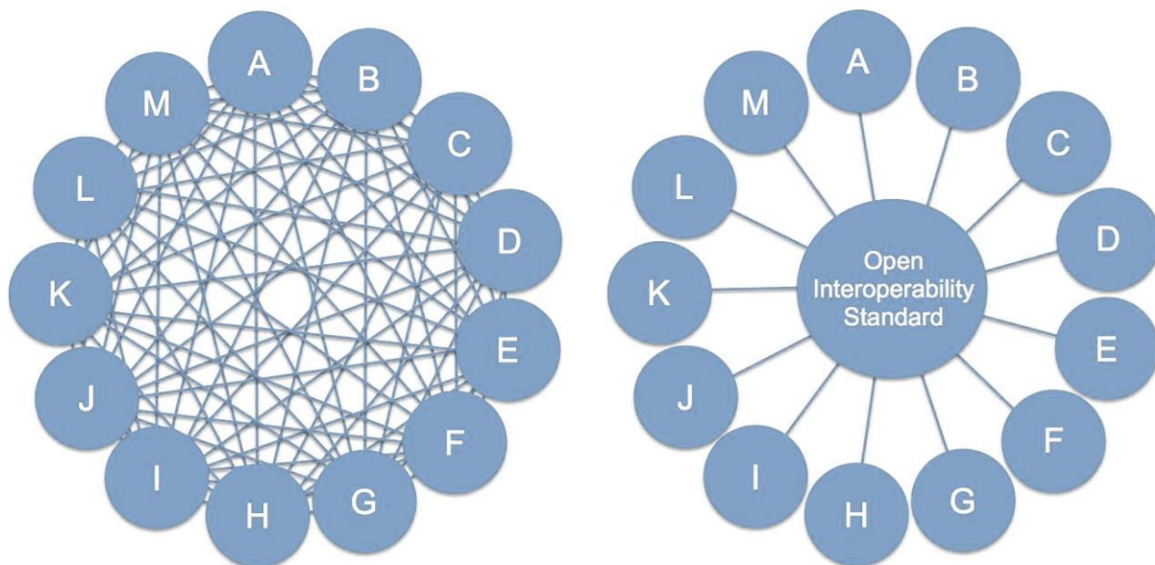
Figure 5: Direct translators vs. an open interoperability standard (Kiviniemi et al.,2012)

3.5   BuildingSMART's open standards

As a new approach to describing and displaying information in the building sector, BIM is able to combine different threads of information used in the design, construction, and operation processes into a single operating environment and thus, reducing significant amounts of documentation in paper form. In order to get the benefits of using BIM, however, the quality of both communication and information exchange needs to be assured (IfcWiki 2017).

Therefore, the buildingSMART consortium (formerly, the International Alliance for Interoperability), which aims at improving the exchange of information between software applications in the AEC industry, developed several basic methodology standards for openBIM

and namely, IDM, IFC, IFD, MVD and BCF (buildingSMART 2015). *Table 2* gives information on their uses.

Table 2: Basic standards for openBIM (buildingSMART 2015)

Name	What it does	Standard
IDM (Information Delivery Manual)	Describes processes	IDO 29481-1; ISO 29481-2
IFC (Industry Foundation Classes)	Transports information/data	ISO 16739
BCF (BIM Collaboration Format)	Change Coordination	buildingSMART BCF
IFD (International Framework for Dictionaries)	Mapping of Terms	ISO 12006-3 buildingSMART Data Dictionary
MVD (Model View Definitions)	Translates processes into technical requirements	buildingSMART MVD

3.6   Industry Foundation Classes (IFC)

IFC is an EXPRESS schema developed by buildingSMART to support the interoperability within the building industry and solve the issues of incompatibility between software applications coming from different Computer Aided Design (CAD) vendors. As a data definition language for modeling products, the EXPRESS schema consists of formalized concepts and the relations between them. By the use of entities, types and relationship attributes products can be described and conceptualized. The actual product data is contained in a STEP physical file (Schevers, H. and Drogemuller, R. 2005).

The IFC file format facilitates the interoperability between different software vendors by allowing them to share information in the same generic data format, which can be read by any software application, regardless of its internal data structure. Its schema contains concepts, such as classes, attributes, relationships, property sets, quantity definitions. IFC can be generally described as an object-oriented data model composed of class definitions describing elements and processes (Svetel and Pejanović 2010).

The IFC specification is written according to the EXPRESS data definition language and alternatively as an XML Schema specification in ifcXML format. It also incorporates reference data, which is represented as XML descriptions of property and quantity definitions. The IFC model consists of four information layers and namely, a Resource layer, a Core layer, an Interoperability layer and a Domains/Application layer. The Resource layer is important for low-level concepts, or objects that serve a general purpose and their definitions don't include a Globally Unique Identifier (GUID). The Core Layer contains the most abstract concepts. All concepts above this layer have a GUID and optionally - an owner history and a history information. The Interoperability layer describes objects or concepts related to two or more domains, whereas the Domains layer contains specific entity definitions associated with a specific professional discipline (buildingSMART 2013).

3.6.1 IFC Architecture

As already mentioned before, IFC represents an EXPRESS-based entity-relationship model arranged into an object-based inheritance hierarchy. In general, in an IFC file one can distinguish between rooted and non-rooted entities. The rooted entities have a GUID,

ownership history, name, description and are derived from the most abstract root class from all IFC definitions and namely, the *IfcRoot*. The class itself consists of the three abstract sub-concepts *IfcObjectDefinition*, *IfcRelationship* and *IfcPropertyDefinition* which are used to define object types, relationships among objects, as well as object properties (Wikipedia 2017b). The difference between rooted and non-rooted instances lies in the fact that rooted entities can be used independently, whereas non-rooted entities are not supposed to be used as separate instances because they don't possess a concept of identity - a GUID (buildingSMART 2017). *Figure 6* below gives an overview of the *IfcRoot* subtype tree.
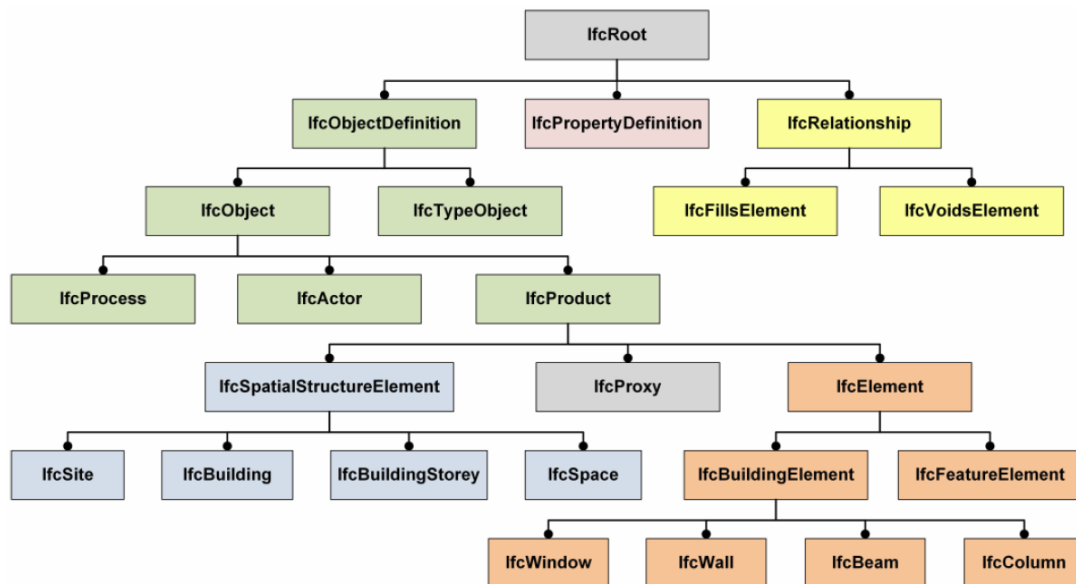
Figure 6: The upper layers of inheritance of the IFC data model (Borrmann et al. 2015)

4   Management of requirements in the AEC industry

In the building industry, client requirements are the basis of a project and they address general aspects such as overall goals, activities, and needs of the client, as well as very detailed characteristics of the product such as materials or special conditions. Due to the iterative nature of the design process, however, the emerging of new, derived requirements and the redefinition of old ones become unavoidable. The documentation supporting the process of requirement evolution is nonetheless usually not kept updated in accordance with the evolution of the design, which as a consequence results in requirements not being satisfied, or in a design which doesn't comply with the initial expectations. Therefore, according to (Kiviniemi 2005), even a simple link between requirement information and design tools can increase the traceability of documentation and the relations between product and requirements.

4.1   What are project requirements?

According to (Mogk, N. W. 2014) project requirements are the means to unambiguously state what a system should do, and they also serve as a basis of the contract between customer and provider. Due to the fact that the design of a system should be compatible with the wishes of the customer, which in the process of communication can evolve, the designer has to reach to these changes by adapting the design, which makes the modification of the project requirements unavoidable.

In his book "The requirements engineering handbook" (2004) R. Young defines a requirement as a necessary attribute in a system, which has the purpose of identifying capability, characteristic or quality factor of that system with the sole intent of bringing value and utility to the customer. The author further points out the importance of differentiation between "stated" and "real" requirements. While the "stated" requirements represent the wishes provided by the customer at the beginning of the design process, the "real" requirements reflect the verified needs of the user in relation to the particular system. Therefore, they can be also described as "derived" requirements.

Yu et al. (2010) further clarify the pivotal role of project requirements during the complete project life cycle by stating that they are not only defining the needs of the stakeholders and the conditions that the end product must meet to satisfy that need. In fact, project requirements are the central part of all processes related to the design and execution of a construction project and namely, project planning, risk management, trade off and change control etc. Therefore, according to the authors, they need to be "complete, unambiguous, consistent, feasible, solution neutral, traceable and necessary".

4.2   Types of project requirements

Kamara et al. (2002) have distinguished between four additional types of requirements apart from the client requirements and namely:

- Site requirements
- Environmental requirements
- Regulatory requirements
- Design and construction requirements
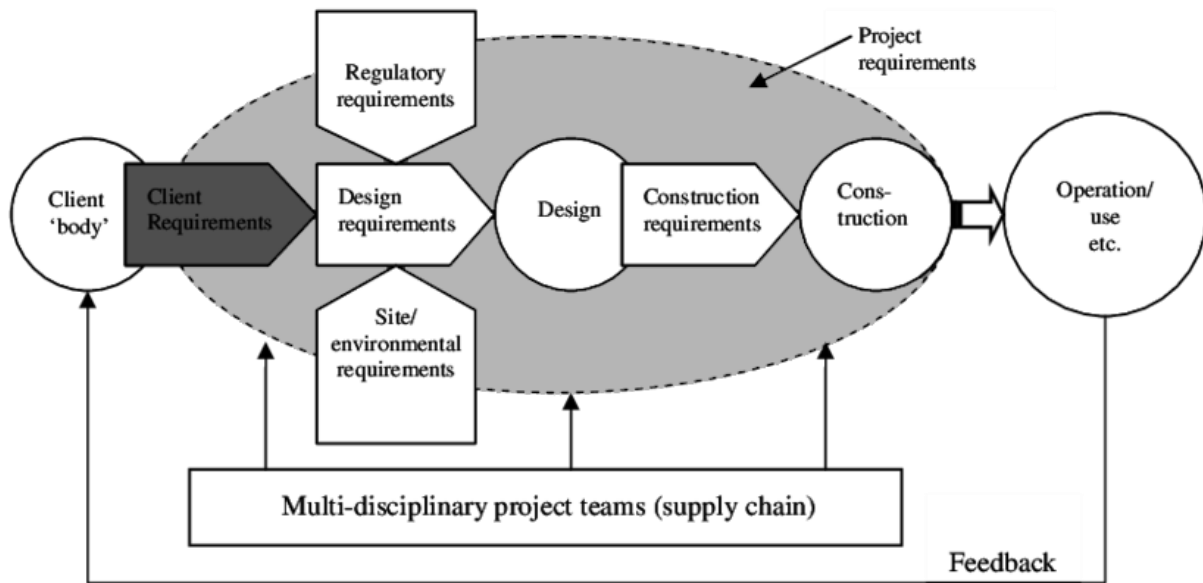
Figure 7: Interrelationship between project requirements (Kamara et al., 2002)

As displayed in *Figure 7* below, the client requirements in combination with regulatory, site and environmental requirements serve as an input for the design requirements, which produce the construction requirements. This serves as a proof that the design phase is the most requirement-intensive phase of a construction project.

4.3   Challenges of requirements management in the building industry

The AEC industry is often described as a highly fragmented and data intensive industry which also lacks automation partially due to its project-based nature. The construction industry not only depends on a large number of stakeholders with various domain-specific knowledge, but all of these professionals participate in different phases of the project and rarely during its entire lifecycle. In the presented context, data management within the industry has become highly fragmented and therefore, the traceability of information and documentation is also challenged (Beach et al. 2013).

As the industry nowadays makes use of both traditional and integrated contracts, the aforementioned fragmentation and its impact on data and requirements management present two different issues in regard to the contract type. Kamara et al. (2002) *(Figure 8)* pointed out the problem with the procedure of traditional building contracts by naming it " a sequential 'over the wall' syndrome for building projects". The challenges in traceability and continuity come from the fact that the stakeholders involved in 'downstream activities' such as contractors are usually not involved in 'upstream' tasks and decisions taken during the design processes. Each actor in the chain passes on their domain-specific interpretation of requirements to the next stakeholder.

However, the collaborative way of working imposed by integrated contracts presents a different issue and namely, the issue of processing client requirements in a way in which the collaboration between disciplines can be facilitated, which implies a neutral definition of requirements. All parties in an integrated team should be able to understand and work with the information without having to decode the interpreted version from the perspective of another professional (Kamara, Anumba, and Evbuomwan 2002).
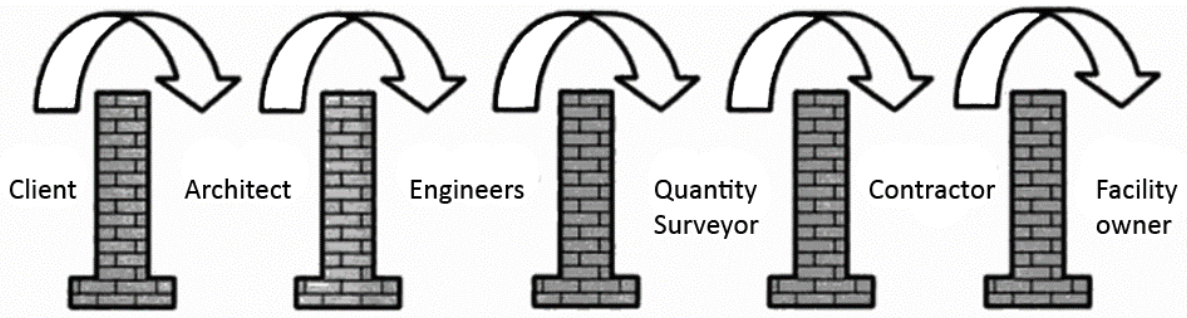
Figure 8: Sequential 'over the wall' syndrome for building projects (Kamara et al., 2002)

Jansson et al. (2013) express the opinion that requirements management throughout the project's life cycle if negatively impacted by the lack of transparency, results in design solutions which do not comply with the original needs of the client. As a result, the designs require iterations and rework. In support of Yu et al. (2010), Kamara et al. (2002) and Beach et al. (2013) the authors also point out the fragmentation within the industry as one of the major factors contributing to the lack of proper requirements management framework, which can facilitate the unambiguous use of information. As shown in *Figure 9* below, the so called "operational islands" within the building industry are formed due to the intersection between the stakeholder domains involved in a project and the building delivery phases of the process. Therefore, the problems of coordination for the traceability of information occurs both vertically (from each phase to the next) and horizontally (from discipline to discipline within a phase).
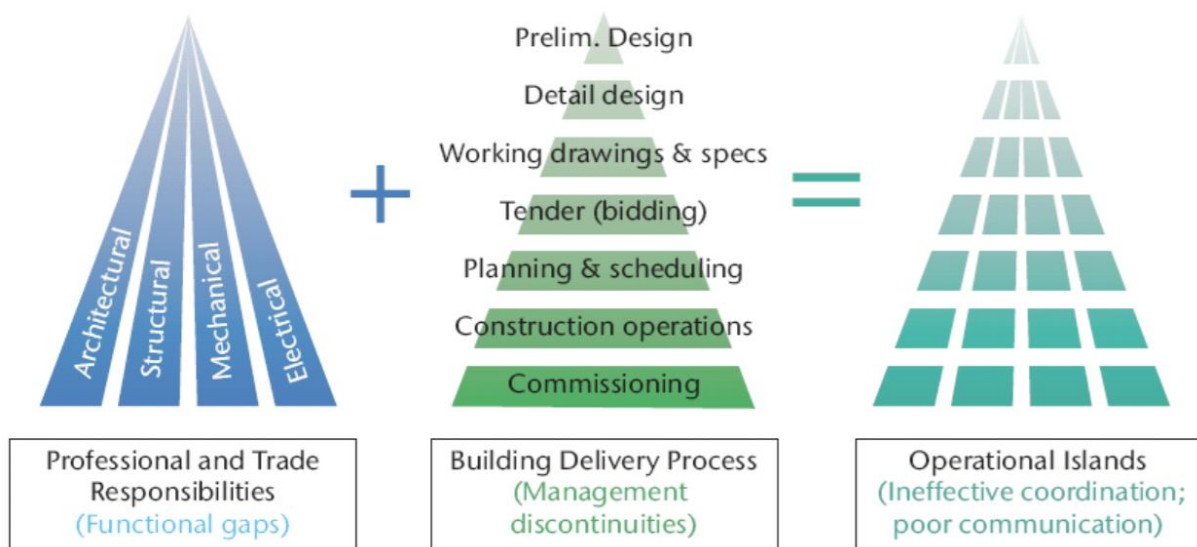
Figure 9: Fragmentation of the AEC industry - 'operational islands' (Jansson et al., 2013)

Yu et al. (2010) systemize the main problems in the management of requirements for construction projects. Similar to Kamara et al. (2002) and Beach et al. (2013) they recognize the issues of fragmentation and the high number of participants in a project. In addition, however, they also mention the need of continuous requirements effort throughout the entire project's lifecycle and the need of proper amount of time spent for working out a good requirements management policy. Yu et al. (2010) also point out the importance of well-documented updates as a way of tracking changes and feedback and the involvement of users for meeting the end-user's expectations.

To elaborate further on the work of Jansson et al. (2013) , Parsanezhad and Tarandi (2016) use the following schema *(Figure 10)* to depict Jansson's procedural view on the iterative process of design changes related to domain-specific feedback from one discipline to another which also results in the alteration of project requirements. As seen in the schema below, the needs of the customer are referred to as 'attributes', the word 'requirements' stands for functional characteristics, the 'parameters' present the design solution, while the 'variables' relate to the final design solution. The 'constraints' are the main conditions for formalizing the variables back to customer attributes through the feedback loop.
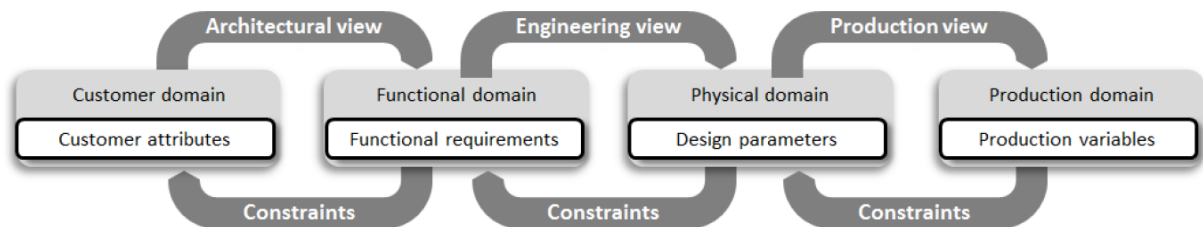
Figure 10: The procedural model of building requirements according to Jansson et al. (2013)

As Jallow et al. (2010) state, usually at the initial stages of a construction project, or the so called elicitation, the emphasis on requirements management is the highest as it serves as a basis for the design processes. Once the project reaches later delivery phases, the links between corresponding requirements do not exist, which again brings the issue of traceability which the aforementioned authors have also indicated. Furthermore, the inadequacy in documenting the changes of the requirements can also lead to misunderstandings between client and provider.

Jallow et al. (2010) go on to also criticize the briefing procedure which is regarded as an ongoing, continuous process throughout the building delivery phases where initial client requirements are gradually reinterpreted and transformed into different project phase-related levels of detail based on their content. The briefing documents, however, are rarely stored in a central repository where all stakeholders can have access to the information, instead, all parties hold their own copies of the brief and as the project goes on and changes occur, these changes are never kept or documented in the same manner. This creates chaos in the documentation and implies a more collaborative way of working with project requirements, just as (Kamara, Anumba, and Evbuomwan 2002) have already recommended.

In the book "Requirements engineering" by Hull et al. (2005) the traceability of requirements has been explained as "the understanding of how high-level requirements – objectives, goals, aims, aspirations, expectations, needs – are transformed into low-level requirements". In other words, requirements traceability in the relationship between the different layers of information or as mentioned earlier, the connections between the "operational islands" of a project.

As Kiviniemi (2005) claims, the major limitation in requirements engineering for the building industry is the lack of a consistent theory which links requirements to the design. Parsanezhad and Tarandi (2016) confirm that statement by explaining that in order to successfully manage project requirements, they need to be formalized. This formalization, however, relies on a broad range of terms, such as 'objective', 'goal', 'constraint', 'criteria', 'variable', 'parameter' and 'attribute', that need to be standardized because currently they are being used in different ways by the different academics.

4.4 Benefits of formalizing requirements management in the AEC industry

As Pegoraro et al. (2017) state, problems in buildings occur due to the complex and iterative nature of the design processes and also due to the many stakeholders involved in such projects. Quite often such problems can be traced back to the processing of requirements in the design phase of a structure which indicates the need for a well-designed framework for their management (Pegoraro et al. 2017).

According to the authors of the book "Requirements engineering" (2005) E. Hull, Jackson, and Dick, the benefits of a requirement management framework which facilitates requirement traceability throughout the project's life cycle are the increased confidence in meeting the initially set project objectives, the ability to assess the impacts of change, as well as the ability to measure progress. In addition to that, traceability allows for balancing cost against benefit for different product components and it puts into perspective the liabilities of subordinate organizations such as suppliers.

One of the most important aspects in making a requirements management framework effective according to (Ozkaya and Akin 2007) is also the traceability aspect, which concerns the ability to connect requirement with system components in an explicit way. Therefore, the authors suggest 3D software-based solutions for the creation of an integrated environment where the designers can manage and reuse requirement information in relation to form exploration. Such integrated environment would contribute to the following:

- Requirement data can be visualized with the help of other tools such as simulation software in order to better measure the design performance based on requirements;
- Design errors can be tracked during the design process and change management during design can be better supported;
- Obstacles occurring due to the lack of tool support which displays both the problem specifications and the design exploration for suitable solutions can be avoided.

4.5 BIM for the management project requirements

Amongst the research done in the field of requirements management and formalization for the building industry, several researchers have suggested the integration of project requirements and BIM as a possible solution to the traceability issue.

In his proposed model hierarchy consisting of requirements, design, production and maintenance model, Kiviniemi (2005) is one of the first researchers recommending the direct linking between project requirements and design representations by the use of design tools. Ozkaya and Akin (2007) also explored the possibility of computer-aided requirement traceability in the design phases of a project. Jansson et al. (2013) suggested that for the proper implementation of the proposed by them requirements management framework based on the axiomatic design theory, the use of BIM should be further investigated and their framework should be connected to construction classification and ontology.

Beach et al. (2013) have indicated that the BIM should serve as "a complete 4D virtual repository of all the data about the building from its conception through its demolition". Data such as management information, product information, and building performance information should be linked to the 3D geometric representation of the structure allowing the users to progressively add information as the project matures. In what concerns information which evolves in time such as project requirements, the authors point out three important

aspects to consider for the proper organization of information/documentation liked to the model in regard to the multi-user nature of the process. These aspects are versioning, composition, and derivation. For example, if the structural engineer adds information to an artifact based on the architect's work, he can use a relationship that indicates derivation.

In their research on requirements engineering and management for building design Pegoraro et al. (2017) have systematically investigated the problems and critical factors occurring during the different requirements processing stages. They have as well assessed the tools, techniques, and methods for the management of requirements that have been previously used or suggested by researchers. In their classification the BIM technologies show a direct link to six out of the eight requirements processing stages, indicating that requirements management through BIM is worth to be explored further.

4.6 Systems Engineering (SE) for the management of project requirements

According to the Directives of the US Department of Energy (2008), Systems Engineering is "A proven, disciplined approach that supports management in clearly defining the mission or problem; managing system functions and requirements; identifying and managing risk; establishing bases for informed decision-making; and verifying products and services meet customer needs." In other words, SE's main goal is making diverse engineering systems compatible with one another through the coordination of their design process and by involving all participating engineering parties.

Although the application of Systems Engineering (SE) approaches in the AEC industry can be of a great advantage both for the successful delivering of building projects and for the efficient management of the project's stakeholders and requirements, the building industry is the only engineering industry which hasn't actively involved SE techniques in its project and process management. With the emerging of Building Information Modelling (BIM), however, which is instrumental to SE and to the practices of coordination, knowledge transfer and error detection, Systems Engineering starts gaining popularity in the domain of the built environment (Kossiakoff, A. and Sweet, W. N. 2011).

4.6.1 SE Methodology

The SE methodology for dealing with complexity is mainly connected with the design and development processes of a system. The user's needs, translated into requirements, are used as a primary input, defining the first level of the system breakdown into its functional elements. The specified functions further influence the system's subdivision into physical elements, which on their part have an impact on the design model. This approach to projects is described by many sources as 'holistic' – main concentration on the whole, rather than the analysis of its separation parts; or the idea that the whole is more than merely the sum of its parts. Every separate component should be considered as influential to the final outcomes of a project and also should be taken into account from the very start (Kossiakoff, A. and Sweet, W. N. 2011).

The situation in the engineering and construction industry nowadays, however, presents a different picture of the development process and its functional breakdown. As Erik Aslaksen (2005) describes it in his report on Systems Engineering in the AEC industry – the development process is focused on the 'how' and not on the functional elements of the system or the 'what'. This is where both the focus and complexity of the engineering work are situated. The engineers' main goal is to figure out 'how' to execute the construction in the most cost- and

time-efficient manner but not on 'what' is to be built. The author also emphasizes on the fact that the first breakdown subdivisions in such a system should be connected to the different engineering disciplines such as structural, mechanical, electrical, etc. where the system elements will be the various kinds of works to be performed.

In other words, SE practices can be described as an approach, which focuses firstly on the needs and requirements of a particular project in order to achieve specific objectives, instead of focusing on a solution or the 'how'. The requirements are further translated into functions and then a framework which fits the project's objectives the best has been designed *(Figure 11)*.
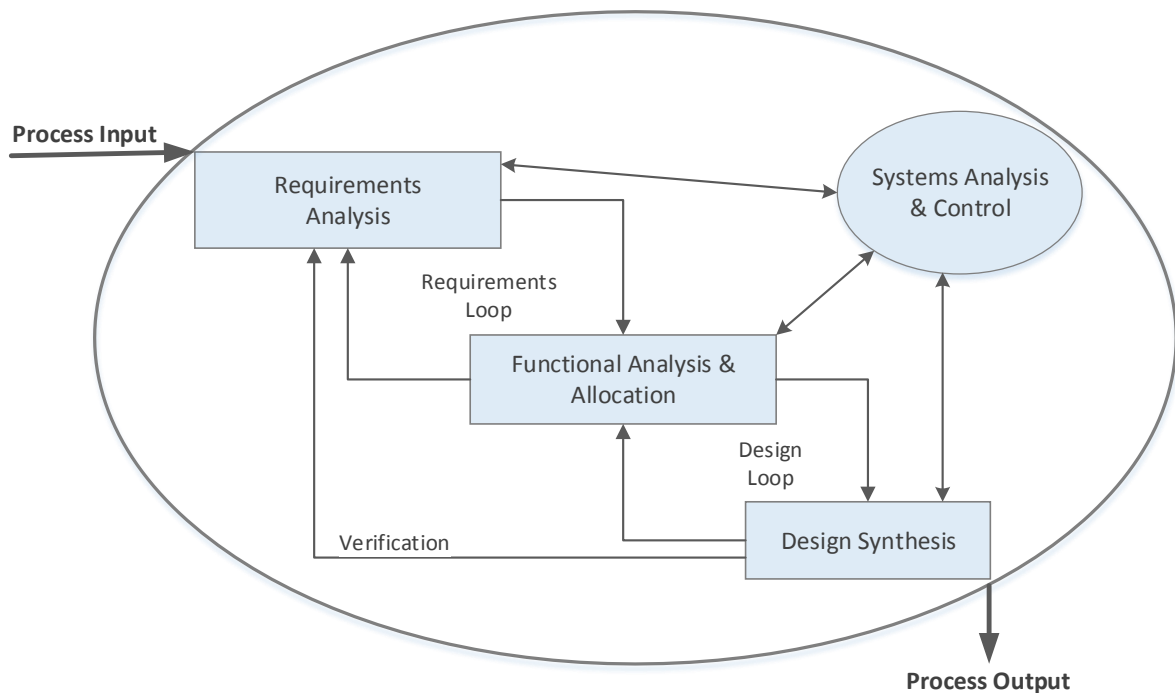
Figure 11: Fundamental activities of the Systems Engineering process (Coinsweb 2016)

On one hand, Building Information Modeling is a process in which the development and control of physical and functional characteristics of a structure can be showcased. By doing this in an early stage of the project, mistakes can be eliminated and thus, prevented from happening in a later project stage. On the other hand, Systems Engineering and its interdisciplinary approach, allow an overview of the complete lifecycle, so that specific requirements can be defined and verification methods can be established. (v. d. Liende, J. 2017) Therefore, by combining the two, the objective of complying with the project's requirements and to the client's demands becomes a more straightforward and better-integrated process, which reduces inefficiencies, mitigates project-related risks, prevents miscommunication, etc.

5   Semantic Web and Linked Data

Over the recent years, the use of Semantic Web technologies has notably increased in the
domains of architecture, engineering, and construction (AEC). These technologies are typically
considered as complementary to the often used Building Information Modelling (BIM)
software (Pauwels, Zhang, and Lee 2017). The use of Semantic technologies in the AEC
industry is thereby stimulated by the need of overcoming interoperability issues among
software tools used by the diverse disciplines involved in construction projects.

5.1   Introduction to Linked Data and the Semantic Web

The term "Semantic Web" is a direct reference to the vision of the World Wide Web
Consortium (W3C) for the Web of Data, which is an upgrade of the Web of Document, or the
web as we currently know it. Tim Berners-Lee (1998), director of the World Wide Web
Consortium (W3C), defined the term as "the merging of human-readable documents with
machine understandable data", which implies that the data is not only machine-readable but
can also be interpreted or 'understood' correctly by computers. In his publication in Scientific
American (2001) Berners-Lee further clarifies the concept of the Semantic Web as "an
extension of the current web in which information is given well-defined meaning, better
enabling computers and people to work in cooperation".

The traditional Web (the Web of Documents) encodes information by the means of natural
language, pictures, videos etc., which suggests that the meaning of information is implicit, or
hidden in the context. Therefore, in order to be correctly interpreted by machines, this
information needs to be additionally described. Due to the fact that humans can understand
the way in which context contributes to meaning, they can interpret natural language
correctly, regardless of its ambiguity (Sack, H. 2014). Machines, however, cannot understand
certain content on their own unless it is made explicit. A simple example for natural language
ambiguity would be looking up the word *camel* on the web. The results generated by the
search engine would refer to the animal *camel*, the brand of cigarettes named *camel* and to
the brand of hiking shoes, also named *camel*. Therefore, this is an example of information
which is easily understood by humans but not by computers and thus, needs to be defined, or
made explicit, for each one of the three results generated. Enriching any data model with
semantic descriptions suggests that the data model describes the meaning of its instances.

In his web series on Knowledge Engineering (2014) Dr. Harald Sack explains how knowledge
can be made explicit by the means of "formal (structured) and standardized knowledge
representations", also named ontologies. In this way, machines are able to automatically
process the meaning of data, as well as to integrate and relate heterogeneous data in a logical
way. Furthermore, the automated deduction of implicit (hidden) information from existing
(evident) information is also facilitated. Describing information in an explicit way would result
in a global knowledge database of machine-interpretable data, where the content is
unambiguously annotated by ontology-based semantic metadata.

The term "Linked Data" can be described as being at the core of what the Semantic Web is all
about or as being the concrete means to achieving the Web of Data by enabling the extension
of the current Web with a global data space based on open standards (Abanda, 2013) Linked
Data is a method for publishing structured data in a way, in which it can be intertwined by the
use of semantics in a meaningful and structured manner, where the different data instances
have one to one and one to many relationships between each other. The main goal of the
concept is to maximize the value and usefulness of information by interconnecting different

datasets. The World Wide Web Consortium (W3C) outlined four rules that need to be considered in order to obtain Linked Open Data. These rules have evolved into the five stars of Linked Data, displayed in *Table 3* below. They represent the five stages towards complete data integration.

Table 3: The five stars of Linked Open Data according to Tim Berners-Lee (Wikipedia 2010)

★	Make data available on the web in any format (e.g. pdf, image, scans);
★★	Make data available as structured data (e.g. Excel);
★★★	Publish data in a non-proprietary open format (e.g. CSV);
★★★★	Use URIs to denote instances so that data becomes unique (e.g. RDF);
★★★★★	Link the data to other data in order to provide context as Linked Open Data (LOD).

5.2 Underlying technologies of the Semantic Web

The first level of semantic expressiveness is achieved by a flexible and generic language that allows to easily represent and combine information from diverse knowledge domains, namely RDF. The Semantic Web thus becomes a semantic network in which information is represented as directed labeled graphs (RDF graphs) (Pauwels, Zhang, and Lee 2017).

In addition, the next level of semantic expressiveness can be reached by using RDF vocabularies and ontologies. Being the key to the Semantic Web concept, ontologies can be described as knowledge structures used to formally describe domain knowledge through the creation of a framework of relevant concepts and the semantic connections between them. The RDFS (RDF Schema) vocabulary, containing class, subclass and datatype specifications, is one example of the aforementioned. In addition, the Web Ontology Language (OWL) reinforces the RDF Schema by allowing the creation of more complex statements such as the creation of classes, subclasses, properties and restrictions (Abanda, Tah, and Keivani 2013).

Figure 12: The Semantic Web technology stack (Guess, 2015)

Furthermore, information from a dataset can be retrieved by the means of semantic queries. The Simple Protocol and RDF Query Language (SPARQL) provides a mechanism to expose data sets and vocabularies in straightforward and understandable ways as so-called 'SPARQL endpoints' (Beetz et al. 2014). *Figure 12* above displays the Semantic Web technology stack, where some of the aforementioned technologies can be seen.

5.2.1 The Resource Description Framework (RDF)

The Resource Description Framework, developed under the guidance of the World Wide Web Consortium (W3C), is the primary infrastructure which enables the encoding, exchange, and reuse of structured metadata. This general-purpose data model enables metadata interoperability through the design of mechanisms that support the common logic of semantics, syntax, and structure (Miller, E. 1998). In its essence, the framework's structure consists of directed, labeled RDF graphs, composed by RDF triples. Each triple is, in fact, a subject-predicate-object statement, depicting facts and relations. This format enables various heterogeneous types of data to be linked together in a logical way and thus creating a web of information that is both machine- and human-readable (Corry et al. 2014). RDF is characterized also as self-describing due to the fact that the labels of the graph describe the data itself (Curry et al. 2013).

The individual parts of an RDF triple can be referred to via URIs which make a specific concept explicit by connecting it to a unique definition on the Web. Considering this precise definition, semantic data can be unambiguously described and queried. By referring multiple triples to the same resource, a network of information is developed, which spreads over the Web and thus, creates a Web of Data (Svetel and Pejanović 2010). Additional ways to denote a subject or an object of an RDF triple is through Blank nodes, which are nodes that don't carry any data on their own. Besides that, an object within a triple can be denoted by a Literal, usually used for values such as strings, dates, and numbers.

The statement '*The grass has the color green*', for example, consists of a subject '*the grass*', a predicate '*has the color*', and an object '*green*'. Using this basic structure, triples can be composed of more complex models, by using triples as objects and at the same time as subjects of other triples — for example, Suzy → said → (triples→can be →objects). The fact that an object and a subject are interchangeable entities in an RDF graph brings up the difference between RDF and object oriented design, where the typical approach is entity-attribute-value. (Wikipedia 2016).
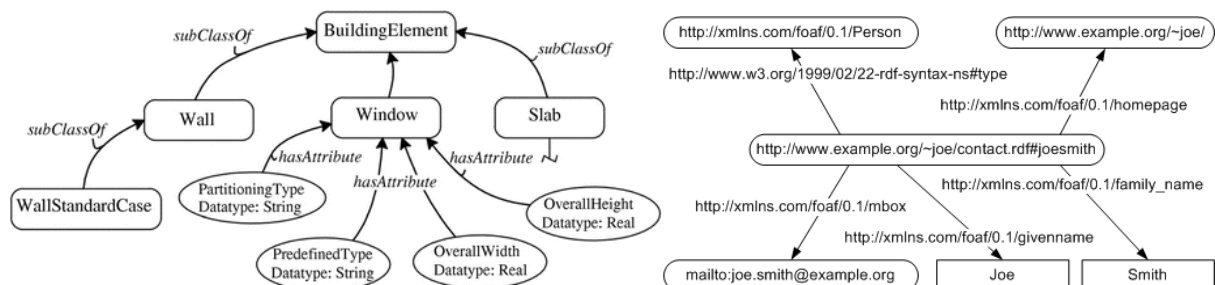
Figure 13: Examples of RDF graphs: left (Karan, Irizarry, and Haymaker 2015), right (Obitko 2007)

5.2.2 Ontology languages

Being a data model for representing information RDF describes concepts by giving them unique definitions, however, it cannot determine whether two or more diverse terms are related to the same concept. Therefore, the use of ontologies is necessary to provide the next level of semantic expressiveness (Svetel and Pejanović 2010).

"Ontology is the formal conceptualization of knowledge in a certain domain." (Ding et al. 2016) In its essence, an ontology is a data model which describes the main entities of a specific knowledge domain together with their relationships, properties, values and the rules that connect them. They allow for new knowledge (in the conceptual model) to be created, based on the already formalized one (Simeone and Cursi 2016).

By the means of linking concepts from different ontologies with each other, a network of ontologies can be created which can serve as a basis for interoperability and logical reasoning. For example, applications such as web services or agents can use the ontologies as a domain of discourse. The interrelation of these ontologies creates a basis for the exchange of information and therefore, creates a good premise for supporting interoperability (Schevers, H. and Drogemuller, R. 2005).

5.2.2.1 Resource Description Framework Schema (RDFS)

RDFS is the most basic schema language which specifies classes, subclasses, property and sub-property relationships as well as data types, or in other words - it extends the basic RDF vocabulary (Svetel and Pejanović 2010). As RDFS is the schema language for RDF, is serves the main purpose of clarifying the information that is expressed by RDF, or namely, the schema carries the information about the data stored as an RDF triple (Allemang and Hendler 2011b, 126).

5.2.2.2 Web Ontology Language (OWL)

The Web Ontology Language (OWL) further enhances the RDFS concepts by enabling the composition of more elaborate statements some of which are cardinality restrictions, type restrictions and complex class expressions (Pauwels, Zhang, and Lee 2017).
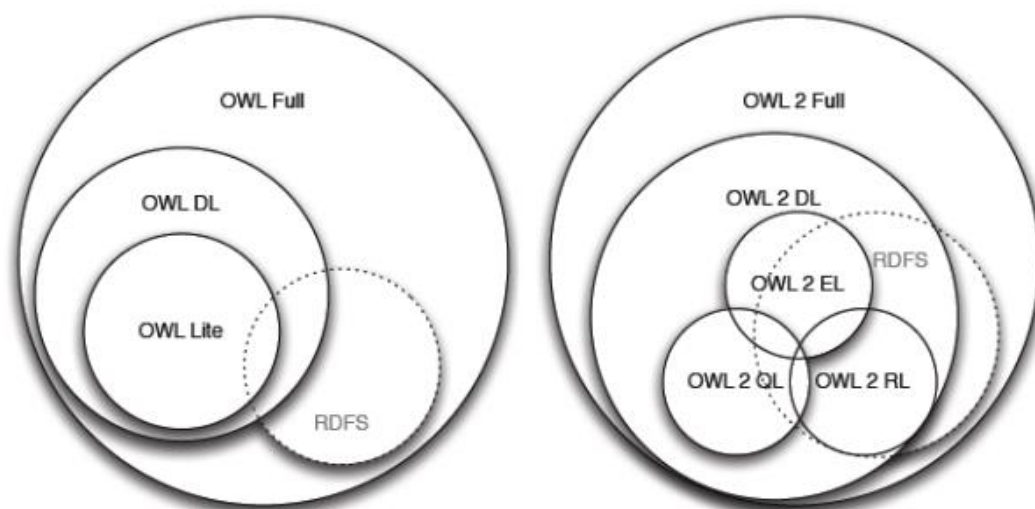
Figure 14: Comparison between OWL 1 and OWL 2 ontologies (Bergman 2010)

OWL, among the rest of the Semantic Web technologies, provides the most elaborate level of ontological functionality and it is based on Description Logics (DLs), which are a set of formal languages to represent certain domain knowledge. OWL has three versions, separated by their level of complexity and their ability to describe complex concepts and namely OWL lite, OWL DL and OWL Full. The syntaxes for writing OWL are XML (for web use) and the N3 notation, which is easily readable by humans (Beetz, J., Leeuwen, J. P., and de Vries, B. 2005).

The current version of OWL is OWL 2 which is subdivided into two parts and namely, syntax and semantics, where the semantics is the meaning of the ontologies, which can be expressed in different syntaxes. OWL 2 DL and Owl 2 Full are the two different ways to give meaning to the ontologies and according to the W3C documentation of OWL 2, there are five syntaxes and namely, RDF/XML, OWL/XML, Turtle, Manchester and functional style. The EL, QL and RL profiles (depicted in *Figure 14* above) contribute to the reasoning efficiency of the language (Zhang, L. and Issa, R. 2014).

5.2.3 IfcOWL

Due to the lack of ways to extend the semantic limitations of IFC, the ifcOWL ontology was suggested as an OWL representation of the IFC schema. It makes IFC data available in RDF format as directed labeled graphs. In this way, the Semantic Web technologies facilitate the data management practices by allowing the connection between building data and various other sources such as sensor measures, material classifications, manufacturer information, GIS data etc. (openBIMstandards 2007).

The conversion of the IFC EXPRESS schema into an OWL ontology was motivated by the ability of the Semantic Web technologies to provide a modeling environment that can deal with heterogeneous data, which in addition supports interoperability across various knowledge domains, the integration of distributed data and the efficiency in data reuse. The application of reasoning engines to infer new knowledge automatically was another appealing feature of the Semantic Web which triggered the interest of researchers to explore the possibilities for conversion to ifcOWL (Terkaj and Šojić 2015).

5.2.3.1 IfcOWL for describing of building information

Schrevers and Drogemuller (2005) are one of the first researchers to explore the mapping from IFC-EXPRESS to OWL DL and to outline some of the issues which the conversion can entail. Beetz et al. (2005) have also discussed an OWL notation of IFCs and its advantages over the XML schema, as well as the fields of its possible application. Other researchers have emphasized on the semantic limitations of the STEP technologies when used as definition languages in the building industry.

Beetz et al. (2008) have pointed out the lack of formal rigidness of the IFC schema and the need for logic-based and provable algorithms that rest on "a mathematically rigid theory such as used by OWL". Another disadvantage of the IFC mentioned by the same paper relates to the limited reuse and interoperability of the STEP technologies due to their unpopularity amongst developers and the inability to incorporate external ontology resources. In addition to that, within the STEP format the possibility for distribution of schemas and instances across networks, which is a central part of the Semantic Web vision, is also extremely limited.

Therefore, Beetz et al. (2008) suggested a method for the conversion of EXPRESS schemas to OWL for improving the reusability of the IFC standard.

Krima et al. (2012) have also recognized the disadvantages of the EXPRESS language related to the fact that it is not based on formal semantics and that there is a limited tool support for it. The authors emphasize on the lack of proper representation of concepts such as function and behavior within STEP and the fact that they are "beyond geometry information", which is what the STEP standard provides at the moment. Therefore, Krima et al. (2012) have specified the rules for defining the semantics of the STEP models in a formal logic by translating them into OWL. Thus, the semantic features added to the models are namely, consistency checking, inferencing capabilities, and decidability which allows the consistency and validity of EXPRESS schemas to be proven, as well as the opportunity of querying STEP files.

Zhang et al. (2014) present a case which highlights the benefits of mapping the IFC schema to OWL for the purpose of information retrieval from the IFC model. They argue that by combining the IFC technologies with ontologies the information stored in an IFC model can be accessed in an easier, more structured manner and that the process of knowledge management can be less prone to errors and inconsistencies.

In one of the most recent scientific publications by (Pauwels, P. et al. 2017) two improvements for the representation of geometric data in ifcOWL have been suggested and namely, the reorganization of the ontology into subsets for the reduction of data size and data complexity, and the creation of alternative frameworks for the serialization of collective data structures. As aggregate data is a commonality in engineering designs and the IFC schema list datatypes are not properly represented into ifcOWL, four representation approaches aiming at an improved mapping of geometric data from the IFC schema were explored.

5.2.4 SPARQL Protocol and RDF Query Language (SPARQL)

The SPARQL Protocol and RDF Query Language, or simply SPARQL, is the query language for accessing RDF data, made a standard by the Data Access Working Group (DAWG) of the World Wide Web Consortium and is, therefore, one of the essential technologies of the Semantic Web. The SPARQL query patterns are represented in Turtle format and can consist of triple patterns, conjunctions, disjunctions, and optional patterns (Wikipedia 2017a).

The way a SPARQL query works is by using a graph pattern including both resources and variables that are being matched against a data graph. The graph pattern is to specify what information needs to be taken from the graph and how the entities and variables relate to each other in it. Apart from retrieving information, SPARQL queries can be used for adding information to a named graph or to multiple named graphs or to also transform information (Allemang and Hendler 2011a).

5.2.4.1 SPARQL for the retrieval of building information

When it comes to the application of SPARQL for information retrieval in the building industry, this approach was made possible by the introduction of the ifcOWL ontology as an internationally recognized modeling standard (Krijnen, T. and Beetz, J. 2017). Some researchers have investigated the possibility of using SPARQL for retrieving building data from the IFC model, which is a practice that can also bring along some challenges.

As described by Zhang and Beetz (2016), the advantage of SPARQL rests on the fact that the language is applicable for querying data from various sources. This differentiates it from the other domain-specific query languages such as SQL. Therefore, SPARQL can be used for querying building data combined with data from other fields (e.g. regulatory data, sensor data). This concept, however, faces some challenges due to the inability of the query language (if used on its own), to retrieve useful relationships or properties from the building models. As a response to this issue, the authors propose the use of functional extensions on top of SPARQL, which define the missing explicitly defined or implied relations in the model. The authors' approach, however, also leaves open the issue related to the functional limitations of the query language and namely, its inability to obtain information connected to geometric data.

In the follow-up scientific publication on querying building data by the use of extended SPARQL functions, Zhang and Beetz (2017) emphasize on the fact that the IFC standard has originated as a way to create and exchange building data, while at the same time this data cannot be queried or subjected to any analysis tasks. In addition to that, the IFC schema does not include information such as product classifications, building regulations, requirements or data from relevant industries, which is commonly used in the AEC domain. Therefore, the paper presents two approaches for querying IFC-based building data in combination with regulatory data from another domain by using spatial and logical reasoning presented as a set of classified functions on top of SPARQL.

5.3   Semantic enrichment of building models

According to Belsky, Sacks, and Brilakis (2016), the semantic enrichment of building models implies "a process in which an expert system inference rule engine applies domain-specific rule sets to identify new facts about building objects and relationships in an input building model and adds them to the model". Lee, Sacks, and Eastman (2006) complement the definition by pointing out the parts which compose the semantics of a physical building object and namely, its form, its function, and its behavior. Therefore, the aforementioned inference rules simply contain the knowledge of professionals able to recognize and classify object instances, as well as their functions and behavior based on the context which the building model provides.

Object–oriented and parametric modeling are two concepts which have been adopted by the AEC industry from the domain of product modeling in the manufacturing industry. The concept of semantic enrichment for the purposes of supporting interoperability of product models, however, hasn't been yet implemented within the building industry domain (Belsky, Sacks, and Brilakis 2016). Although BIM has been developed specifically for the building industry, the Semantic Web and Linked Data technologies weren't. However, despite that fact, both approaches share the same goal of optimizing the knowledge and information management (Abanda, F. H. 2013).

5.4   Knowledge management issues and setbacks

Due to the emerging issues related to the lack of interoperability in the AEC industry, the building domain already looks into the idea of using semantic modeling and Triplestore (RDF) technology for facilitating processes and thus, improving the efficiency of communication, information exchange, storage, and retrieval. Especially in combination with open standards, the Semantic Web technologies present the opportunity of building a strong foundation for

the efficient management of information and knowledge in the different domains of the built environment (Abanda, Tah, and Keivani 2013).

5.4.1 Limitations of BIM

The deep fragmentation between domains in the AEC industry has caused knowledge and information to become dispersed and also often times untraceable. The unstructured nature of data management has caused knowledge gaps which currently are tackled through improved collaboration by the use of technologies such as BIM. Despite these efforts, however, the issue with information ambiguity and misinterpretation remains (Harrison, D., Donn, M., and Skates, H. 2003). One of the often mentioned interoperability challenges related to the semantic meaning of information have to do with the different conceptualizations of the same object within the different domains of the building industry, for e.g. the representation of the same column in the architectural and in the structural model (Belsky, Sacks, and Brilakis 2016).

According to Simeone and Cursi (2016), by looking at the current use of BIM during the design processes in the building industry, two main limitations of its methodology can be identified and namely, its restricted representation spectrum and its inefficiency in supporting the collaboration processes within a project. The first limitation addresses the fact that significant amounts of knowledge and information used in a project are not represented in the BIM model. Valuable information such as tradeoff matrixes, motivations about design choices, evolvement, and derivation of requirements etc. are lost during the design process as BIM represents only the updated or final result, without giving any insight into the process that leads to it. Therefore, the BIM approach turns out to be valuable for the sole purpose of documentation and final design representation in the end phase of a project.

The second limitation mentioned by Simone and Cursi (2016) related to the use of BIM in supporting the collaboration processes can be described as an inability to provide mutual comprehension between the project's stakeholders, where information sharing should be replaced by knowledge sharing. The authors further clarify that the low level of semantic representation in BIM has to do with the fact that data is linked to 'labels', rather than to concepts and their definition modalities, which exist only in the designers' 'head'.

Due to interoperability challenges, the optimal linking of BIM data to external datasets for the enrichment and complicity of knowledge has been prevented, which also contributes to the initially slow adoption rate of BIM in the AEC industry (Abanda, F. H. 2013). As Aubin et al. (2012) have described it, the most important part of BIM is, in fact, the 'I', or the Information aspect, which includes both graphical and non-graphical information, enclosed either in the model itself or available outside of it. The current BIM applications, however, neither fulfill nor facilitate the second functionality, as they barely provide interoperability amongst different tools or possibilities to successfully link other datasets to the model within the specific native software.

Resulting from both the fragmentation in the building industry and the incapability to connect information datasets from the different stakeholders in a project, one small requirement change in one of the BIM models (e.g. the architectural model), such as reducing the height of a door, can lead to modifications in all of the remaining models (e.g. structural model, MEP model) (Törmä, S. 2013). If not communicated, such small change cannot be noticed easily by other project participants, which would present a potential risk of not complying with a

project requirement, possible construction failures, additional costs etc. Therefore, the ability to automatically trace changes of demands or model modifications is also not well supported by the BIM tools currently in use.

5.4.2 Limitations of IFC

Although the development of the IFC open standard intended to eliminate the interoperability limitations arising from the use of different software packages by creating one neutral format for describing building information, it is far from solving the problem in its entirety. In the use-cases from Pazlar and Turk (2008) and Verstraeten et al. (2009), aimed at information data exchange through the IFC standard, the distortion and loss of information were confirmed. Due to the fact that almost always no exact mapping between the IFC description schema and the schema of the software application used is possible, complete data interoperability cannot be achieved. As a result of that, the IFC model is exchanged between the project's stakeholders "as just another information model, resulting again in multiple building models managed in parallel, containing different information about the same subject" (Pauwels, P. and De Meyer, R. 2011). The IFC models are oftentimes used in a read-only format due to the inability to make a complete conversion roundtrip – the "load-export-import-save cycle", without receiving as an output a model with changed structure, requiring extensive manual adjustments (Törmä, S. 2013).

The aforementioned findings are also supported in a report on 3D information exchange by Pieter Pauwels et al. (2011) who also emphasizes on the issues arising from the use of conversion tools for collaboration between the different stakeholders in a design project, the result of which leads to the undesired distortion and modification of data. He clarifies that many transition processes containing both geometric and product model information, when introduced in another application (product suite), become either oversimplified or lose their semantics.

Abanda et al. (2013) also point out the issue of information loss resulting from the "mistranslation between different syntaxes and/or semantics" due to the current approaches of conversion tools adopted in the AEC industry, as well as due to the inflexibility of open standards such as IFC. Some of the implications mentioned in the paper relate to errors and limitations in the design phase such as the inefficiency of processes due to repetitive work and the false interpretations of information by the different design domains (architecture, engineering, MEP).

In his report on Semantic Linking of Building Information Models, Seppo Törmä (2013) argues for the need of instance-level compatibility between building models and emphasizes the fact that IFC provides only type-level interoperability for the AEC domain, which doesn't completely tackle all interoperability issues. He supports the claim of Pauwels and De Meyer (2011) about the IFC model being "just another information model" by defining the core of the remaining interoperability problems. That is namely, the fact that the information exchange itself is facilitated by the IFC standard, however, the use of the exchanged information by the recipient still demands context interpretation and manual work.

Therefore, it can be argued that BIM and IFC solve the interoperability issue in the building industry domain only partially, as the process is still heavily dependent on proprietary standards for keeping the completeness of information in a 3D model. BIM tackles the issues on level communication exchange between stakeholders but at the same time performs

poorly on the level information management and traceability as it is currently primarily used for visualization, clash detection, building design and construction of an as-built model. When it comes to providing information to other applications such as building performance or calculation software, the use of BIM is merely there (Pauwels, P. and De Meyer, R. 2011).

5.5   Adopting Semantic Web technologies in the AEC industry

The ontology research in the construction industry can be divided into three main stages. During the first period, the term ontology is rarely used as the industry was more interested in exploring the opportunities of artificial intelligence (AI). During the second stage, after the year 2000, as the topic of knowledge managements becomes popular, the interest in ontology research within the industry begins to emerge with projects such as the e-COGNOS ontology which has the main objective of "consistent knowledge representation of construction knowledge items". The third stage of the ontology research in the AEC industry involves the ontology research on BIM, which continues to the present moment and involves feasibility studies on the IFC schema, the Web Ontology Language (OWL), the Geography Markup Language (GML) and others (Zhang, L. and Issa, R. 2014).

In their paper on BIM and Information Technology (IT) for project collaboration, Shan and Chua (2011) present three web technologies that could enable the improved efficiency in the building sector and thus, complement existing technologies such as BIM. One of the mentioned technologies is semantic search in relation to which the authors point out several benefits. The first one is the improvement of data and file management using ontology mapping, the second one is the ability to use complex search queries across a large number of data sources and the third one is the ability to create knowledge bases from 3D models by the means of semantic search engines and subsequently to share them.

In their literature review on Semantic Web technologies in the building industry, Pauwels et al. (2017) identify three main benefits of adopting the technologies in the design and construction industry and namely, achieving better interoperability, linking across domains, and using logical inference and proofs. The interoperability aspect relates to the aim of facilitating computers to understand the data they are working with and thus, being able to successfully load the same content in different software applications. In this way, the issue with different geometric representations of the same object would be tackled. While the interoperability aspect deals with the different depiction of the same content, linking across domains tackles the issue of combining different content such as geometry, cost and simulation data. The third benefit mentioned has to do with the possibility of using generic inference engines to infer additional information from the information in OWL or RDF based on basic description logic principles. In addition, the Semantic Web Rule Language (SWRL) can enable IF-THEN rules, which combined with building data and reasoning engine can facilitate the inference process.

The use of Web-based representation methods in building information models introduces not only the possibility of linking individual model instances to data on the Web but also builds the infrastructure for establishing a centralized architecture used by the different domains involved in a building project (Törmä, S. 2013). The created interoperability model deals with the issues of format conversion and information loss which occur in the processes of designing and constructing a building structure.

According to Pauwels et al. (2011), instead of the BIM plus IFC approach, a semantic web technology approach can improve the processing of 3D information, as well as minimize the loss of information due to conversion practices and thus, improve significantly the design workflow in the industry. The graph description associated with the semantic web approach would facilitate the relationship between schemas, coming from different applications. This would allow the coupling between 3D information and non-geometric data, as well as the aggregation of knowledge from different domains.

5.6   COINS and the SE Exchange Standard (SE-BIM)

COINS (Construction Objects and the INtegration of Processes and Systems) is a flexible standard for the exchange of BIM information and it also serves as a data exchange format by the means of a container for BIM related information. The standard has been adopted by the Dutch ministry of Transport, Public Works and Water Management as a way to manage complex information deliveries of construction projects, often consisting of combinations of various data structures (Rijkswaterstaat 2017).

The COINS standard has a core model which describes the essential elements necessary for the exchange of information and namely, objects, properties, relations and document references and a reference framework, which is an addition to the COINS core model. A reference frame is an extension of the core model, it contains knowledge needed for exchanging information about a specific domain (Coinsweb 2017).

Its core model is an extension of the OWL ontology, it is complementary to other standards from buildingSMART such as IFC, IFD Library, and IDM, and it facilitates the neutral data exchange between different software platforms. The information exchanged refers to documents, models, and object type libraries which are related to each other and it consists of BIM-data, SE data GIS-data, standards such as IFC, and file formats such as DWG, DXF and RVT (Coinsweb 2017).

The SE exchange standard, or also referred to as SE-BIM, will be implemented as the SE reference framework in COINS and will handle a broad array of building data such as project requirements, verification and validation processes, risks etc. The SE data will be exchanged within the COINS container, allowing information, in various formats, to be communicated in coherence (SE-BIM Werkgroep 2017).

Within SE-BIM all concepts of the SE domain and the relationships between them are defined semantically. The project data (instances) and its meaning (semantics) are structured according to the RDF syntax. For example, a requirement claim relates to a physical object. The semantic structure of the relationship would be as follows: subject (requirement) - relationship (refers to) - object (artifact).

The SE exchange standard does not address the way the SE practices of the different project parties should be arranged and how SE's activities should be performed. However, the standard limits the potential objects and potential relationships that both parties can use in their communication (SE-BIM Werkgroep 2017).

5.7   Conclusion

From the conducted literature research in relation to requirements management in the design phase, it can be concluded that the majority of papers propose a theoretical solution to the issues related to the management of project requirements, rather than an actual

implementation. In those papers, the focus is primarily set on requirements management in the elicitation phase, which to some extent covers the architectural project requirements but barely touches upon the engineering requirements derived from them and related to building codes and regulations. None of the scientific works suggested an approach for the management of project requirements through the use of Semantic Technologies apart from the COINS/SE-BIM standard.

From the conducted literature research in relation to the limitations of BIM and the IFC schema, as well as on the implementation of Semantic technologies, ifcOWL and SPARQL for the description and retrieval of building data, it can be concluded that although fundamental for the collaboration and interoperability within the industry, BIM and IFC are far from being able to provide complete and integrated management support for building data. Furthermore, the research into Semantic Technologies for the AEC domain provides solutions to the issue of data interoperability and integration, however, the implementation of these solutions depends on the standardization of domain-specific terms and the formalization of domain knowledge.

PART C: QUALITATIVE RESEARCH

This section introduces the interviews with exerts from the domains of Systems Engineering, Building Information Modeling, and structural engineering held for the validation of the research problem (*section 2.1*) and with the aim of answering some of the research questions (*section 2.2*). This section introduces the interview setup and the interview questions and in addition, elaborates on the interview findings and the conclusions drawn from them.
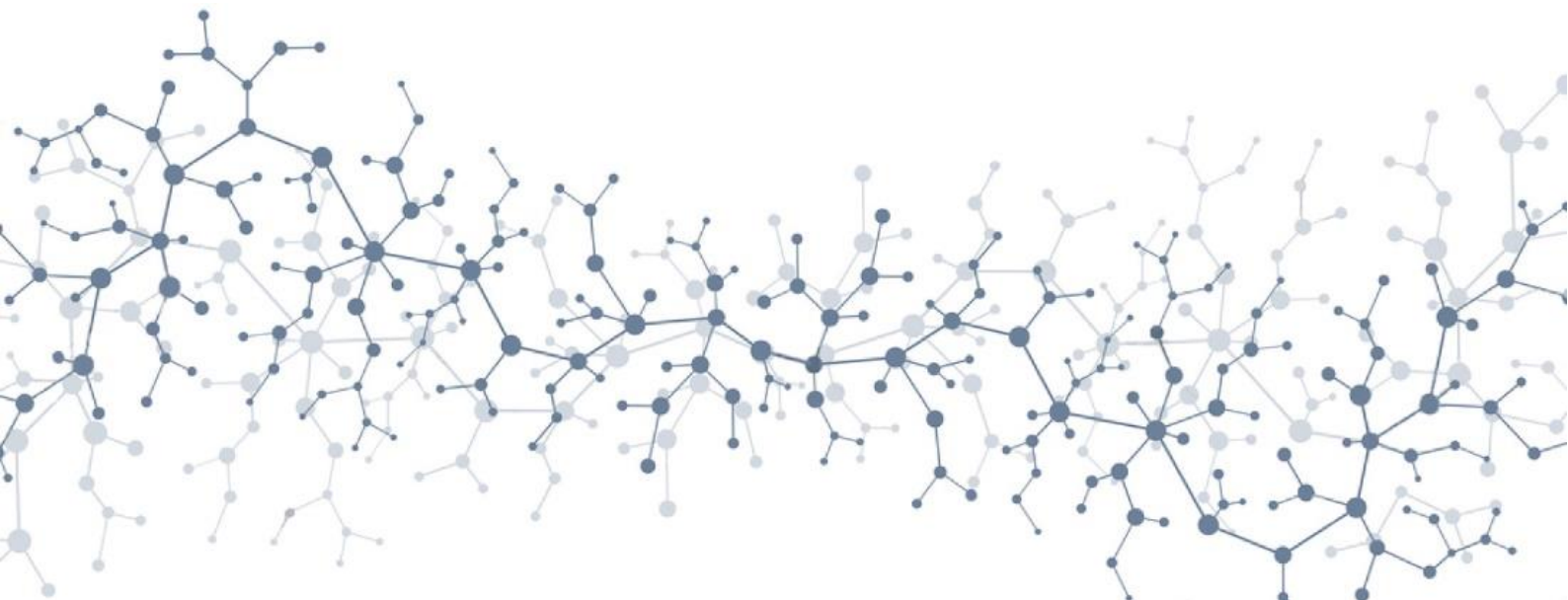
6   Expert interviews

The quantitative research was conducted in the form of semi-structured interviews. As this type of interviews has a fairly open framework, which allows focused, conversational, two-way communication, they can provide reliable, comparable qualitative data. They can confirm what is already known but also provide the opportunity for learning (Keller, 2006).

The interviews with experts were held in order to receive a more in-depth information concerning the data integration practices in the AEC industry and to confirm or deny the need of direct linking between project requirements and building information models. There were nine interviews held in total, with domain professionals from both V&L and from external companies. The opinions of two systems engineers, four BIM exerts, two structural engineers and one project manager in the structural engineering domain were gathered, processed and analyzed. The core businesses of the companies are diverse, as well as the organizations with which they cooperate. An overview of the interviewees and their functions can be seen in *Table 1* and the summarized transcripts from the interviews can be found in *Appendix XIII: Qualitative research: Summarized interview transcripts*.

Table 4: Record of interviews

Date	Company	Person	Function
07.03.2017	ProcessMinded	Chantal Laurijssens	SE process manager
07.03.2017	ProcessMinded	Adrian Dinca	SE & Relatics engineer
04.04.2017	Royal Haskoning DHV	Yves Scholtes	BIM manager/coordinator
04.04.2017	Royal Haskoning DHV	Jan-Henk Oldenburg	BIM advisor/coordinator
20.04.2017	Hendriks B&O	Joost van de Koppel	BIM manager
15.05.2017	Stam + De Koning	Stijn van Schaijk	BIM process manager
23.05.2017	Ingenieursbureau V&L	Lucas Verhelst	Structural engineer
25.05.2017	Ingenieursbureau V&L	Jurgen van der Aa	Structural engineer
25.05.2017	Ingenieursbureau V&L	Henk Verhoeven	Project manager/Engineer

6.1   Interview setup

The following interview questions were derived from the research problem described in *section 2.1* and with the help of the literature review conducted before their formulation. The questions can be divided into four main categories. The first category aims at introducing the interviewee and his/her company's field of work, while the second and third categories aim at addressing the current situation in the AEC industry with regard to BIM and information management, specifically in connection with the handling of requirements.

The last category of questions is intended to provide an expert opinion on the expected impact of connecting project requirements to building information models on the data management practices, the efficiency of design processes, and the mitigation of conformity-related risks. Thus, the interviews are essential to confirming or denying the practical relevance of the thesis research. The questions asked are the following:

General

1. Can you describe the core business of your organization?
2. What is your position in the organization and what are your main roles and responsibilities?

3. With what type of organizations do you most often cooperate?

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?

5. Which tangible benefits have you experienced through the implementation of BIM?

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?

Information management (requirements management)

7. How is documentation handled in your organization currently and how much project information is still stored in paper form?
8. How do you handle/systematize building requirements?
9. In which manner do you ensure that requirements information remains up-to-date?
10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?

6.2   Interview findings

The professionals interviewed for the qualitative research of this thesis provided an insight into the data integration practices of the AEC industry and confirmed the need for better interoperability between the different information sources within a project. The lack of a direct reference between the 3D model instances and the production information such as drawings, calculations, and reports are recognized as critical both by the BIM and the SE specialists. Regardless of whether the project's information is encoded into documents such as PDFs or stored in an information management system such as Relatics, the connection to the BIM model is always absent. Therefore, for effectively using all information sources, knowledge of the project is required.

All of the organizations participating in the interviews implement BIM in their processes and its main uses include keeping track of the project's progress, validating the project's design and detecting clashes between the different disciplines within a project. Besides that, BIM also plays an important role for cost and volume estimations, timeline phasing and communication.

The benefits pointed out by the respondents consist in the mitigation of risks such as time and cost overruns. The design can be easily validated and potentially risky and costly problems can

be prevented and solved before the execution phase and therefore, both the contractor and the client save money and time. The visual aspect of the model also contributes to diminishing communication errors and problems coming from misunderstandings between the design parties. Besides that, the information reuse aspect in using BIM contributes also to improving the quality of the design by having the opportunity to align different models and detect clashes on time. Also, the process of communication with the project's subcontractors is easier and faster with the common coordinated model. It allows all parties to discuss the best practices for construction and to find the best solution faster.

The many stakeholders involved in a project and the differences in knowledge about BIM that each stakeholder has are the biggest bottlenecks for the proper implementation of BIM currently. Getting everyone on the same page is an issue both internally within the companies and externally with other project parties because a lot of them are not BIM-ready. In addition, creating a model from which all parties can benefit is a challenge due to the fact that the different stakeholders have different interests and not always use the 3D model for the same purposes. Decisions on what information has to be integrated into the model need to be made from the beginning of the project and kept consistent throughout the whole process. Project stakeholders are usually overwhelmed with all the things they need to think about and when the pressure gets on due to deadlines, they tend to forget about keeping the BIM up to date. As a result, the model becomes invaluable. Therefore, the issues with the proper implementation of BIM could be also attributed to the lack of strict policies enforced by authorities such as the government which is the case in the United Kingdom.

When it comes to the information management practices within their organizations, the interviewees estimated that almost the entire project information is managed in the old-fashioned, traditional way - through PDFs and Excel sheets, while only a fraction of the important information is integrated into the BIM model. The reasons for that can be attributed to the fact that the client still wants to work with the traditional 2D drawings and regards the use of BIM as additional costs in the design phases. The contractors also play an important role because not many of them currently have incorporated BIM in their processes and are rather reluctant to use it. Sub-contractors have even lesser knowledge of BIM and therefore, they are yet another obstacle to moving towards a BIM-centered management of information. At this moment, a person working on a project has to link objects to documents and documents to objects in his/her head in order to keep the continuity of data and in order to be able to work effectively.

In relation to the handling and systematization of building requirements, the industry and the different stakeholders within it understand the process differently and this is primarily due to the heterogeneity of requirements with which the different domains have to work. While the architect focuses on the soft client requirements specified in the elicitation phase of the project, the structural engineer is more involved with the building law and the Eurocodes and identifies the information specified in them as the requirements directly related to his/her discipline. Another aspect related to the management of requirements is that for civil projects in the Netherlands the use of systems engineering is enforced by the government and therefore, a clear framework for the management of requirements for civil projects exists. For

building structures projects, however, due to the private clients and still widely used traditional contracts, the management of requirements is a vague topic. For such type of projects, some of the interviewees indicated that they try to systematize requirements mostly for internal use within their companies because the client doesn't demand them and therefore, other project parties usually don't feel the necessity to systematize them either. Even though some of the interviewees indicated that an effort has been made in the collection and definition of requirements, the process for their verification is executed only in the head of the professionals without the use of written proofs of conformity. Some of the interviewees even indicated that these verification practices often depend solely on the proactive behavior of the employees, rather than on policies or rules within the company stating that the project's compatibility with the requirements should be controlled and proved.

While some efforts in the direction of requirements management for building structures have been made, they are far from reaching the interoperability level which the industry needs for the mitigation of project risks associated with the lack of traceability in whether the project complies with the initially set goals and objectives. As one interviewee specified, it is quite rare that a project has met 100% of the initially set objectives from the elicitation phase after it's on-site completion. The most critical point which, according to the interviewees, is negatively impacted by the lack of requirements control is the transition from the final design phase to the construction phase because that is where the design meets the production information. That is also the point at which the information and respectively, the requirements, are most scattered into different documents. In addition, the final design is the phase where all problems need to be resolved as they cannot be shifted further in time because all project information sources get transferred to external parties who haven't been involved in the design phases before.

All interviewees expressed the opinion that establishing a link between the BIM model, the project's documentation and the requirements is a feature with obvious benefits. Such management approach would support the overview of all information datasets and it would be an effective way for checking whether the project is in-line with all requirements at any stage of the design or the construction. In fact, the interviewees suggested that the majority of project information should be linked directly to the BIM because all stand-alone PDFs that don't directly connect to anything are of a great importance for delivering the right quality of the project. It could be a very useful feature also for civil projects because so far Relatics doesn't properly relate to the geometrical design and the geometrical design, relate neither to the requirements specified in Relatics nor to the documentation from the DMS (document management system). The biggest benefit of such link, according to the professionals, rests in the fact that by relating documentation to object instances from the model, rather than relating just requirements to object instances, the project information becomes traceable and understandable also for external stakeholders. Two important features to consider according to them, however, are firstly the ability to filter the contents of the documentation and secondly, to make a distinction between the as-designed and as-built states of a project in relation to the requirements and the documentation.

When asked about the linking of which type of requirements would improve the information management process within a building project the most during the design phases, the interviewees' answers differed significantly. All requirements are important but they can be prioritized within each professional domain differently. Building specification requirements, the company's internal requirements, calculation requirements, contractual requirements, procurement requirements, and the requirements defined in the building law are some of the ones pointed out. What is the most important for the project in general, however, depends on the client and the starting points given by him during the elicitation phase.

In relation to the level of detail to which requirements should be associated with the design, there are two most popular opinions. The SE professionals and some of the BIM professionals insisted it should be an evolving process throughout the different project phases in order to get the most benefit from such an approach and in order to keep the consistency of data. Others, such as the structural engineers, expressed the opinion that the LOD 300 is the most appropriate phase as that is the phase where the most important and detailed information about a project is generated by all parties. The element-wise, rather than the systems-wise connection between requirements and model is preferred by the interviewed engineers due to the fact that they have no defined framework for the consideration of systems and therefore, such approach can cause confusion and probably also the repetition of data.

6.3   Conclusion

What can be concluded from the conducted qualitative research is that the AEC industry is shifting towards the integration of BIM for its management practices. However, for the proper implementation of BIM in a project, not only all stakeholders need to understand how to use it but their BIM-knowledge needs to be equal and corresponding to the complexity of the project. Although almost the entire project information for the building structures domain is managed in the old-fashioned, traditional way - through PDFs and Excel sheets, while only a fraction of the important data is integrated into the BIM model, the industry recognizes the need for better information interoperability practices. The handling of project requirements in the building structures domain if performed is done in an isolated manner and only for the in-house purposes of the specific company, rather than on a project level. The transition from the final design phase to the construction phase is identified as the most critical point at which the lack of requirements management practices and overview can have the biggest negative impact on the success of the project. An aspect contributing to that risk is the scattered and inconsistent nature of project documentation at that phase which in addition to that is usually encoded in stand-alone PDFs completely unrelated to the design. Therefore, the need for integration between design, requirements, and documentation, as well as the need for requirements integration between the different domains for the entire design and construction process, are recognized as necessary by the industry stakeholders. However, without enforcing strict policies for the management of information within the AEC industry and most specifically, in the building structures domain, the transformation towards more integrated and interoperable way of working can be very challenging.

PART D: METHODOLOGY

Firstly, this section gives an overview of the process and results from the initial research, related to the structuring and classification of general engineering requirements and secondly, the section discusses the prototype tool developed which facilitates the linking between project requirements and the structural model. Chapter 7 gives an insight into the step by step process of requirement data collection and analysis, while chapter 8 introduces the case study used for the development, the ontology created from the input of the initial research, and the actual prototype. At last, the tool is validated with the use of the case study introduced earlier and the limitations of the tool are discussed.

7 Determining general engineering requirements

As outlined by the project's objectives in *section 2.4*, the in-house research should result in a systematized schema presenting the general engineering requirements of the building structures project domain. This schema is the main input for the second research phase and namely, the tool development.

The process of data collection and analysis consisted of several sub-processes. Firstly, a study of the company practices was conducted. It included the revision of the company's quality manuals, the manuals on BIM, Systems Engineering, and the workflow specifications. The study was primarily used as a way of identifying the different categories of engineering requirements.

Secondly, two projects from the domain of building structures and one project from the civil domain were selected and the processes and documentation of each were compared and analyzed. The documentation (E.g. calculations, advisory reports) of the two building structures projects served as a basis for the selection of general project requirements, which in order to confirm the validity of the results, were then compared to the Dutch building law (Bouwbesluit) and the European regulations (Eurocode) for structural/civil engineers. The Eurocodes were specifically used to determine which engineering requirements are material-neutral, or general for the different types of structural components.

In addition to that, throughout the entire research process several interviews, as well as one group session with several of the company's engineers were conducted in order to additionally confirm the soundness of the conclusions drawn at the previous stages. The interviews helped to clarify the connections between requirements and building component types such as columns, beams, floor slabs etc. Lastly, a general requirements schema was created depicting the necessary information, which needs to be proven and delivered for a building structures project at the end of the design phase. The schema is presented in detail in *section 7.5 Results*.

Each individual step of the process, displayed in *Figure 15*, is addressed as an independent sub-chapter in this section of the report.
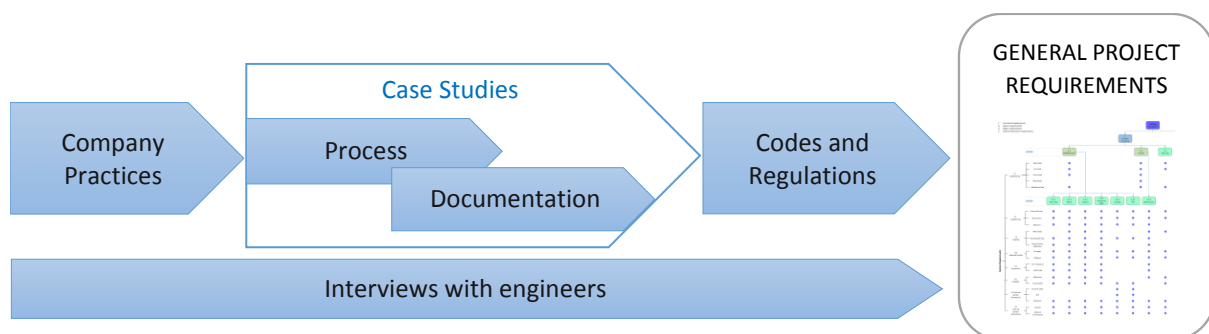
Figure 15: The process schema of determining general engineering requirements

7.1 A review of the company's practices

Ingenieursbureau Verhoeven en Leenders B.V. (V&L) is a structural engineering company which specializes in the design, calculation, and modeling of construction projects in the fields of building structures and infrastructure. Therefore, the company is divided into two working divisions and namely, a building structures division and a civil division. Considering the scope

of this thesis, the initial understanding of these divisions is that they differ significantly in the information management practices that they make use of.

7.1.1 The engineering design process

According to experts within the company, the strict policies on project management and organization of requirements in civil projects are imposed by the client (usually a governmental institution). *Rijkswaterstaat* (Dutch Ministry of Transport, Public Works, and Water Management), for example, requires from all stakeholders in a civil project to handle their information through Relatics.

Relatics is essentially a web-based platform used by projects to customize Systems Engineering (SE) and management of requirements applications completely to their needs. Tools such as Relatics are being implemented in civil works for the management of project information, providing project managers a defined framework for the control of data (Relatics B.V. 2017). In this way, the client can rely on a management tool for the maintenance of the project after its completion.

In order to gain a better insight into how requirements are formalized by the use of Systems Engineering and Relatics throughout the design and construction of a project, one of the civil projects of the company was investigated and namely, the bicycle park Vijfhoek. In the schema below *(Figure 16)* a flowchart displaying the sequencing of tasks during the design process of a project implementing SE is presented. The input and output documentation during each one of the activities is also displayed.

When we compare the well-structured workflow in the design of a civil project with the workflow within the building structures division, where no framework for the sequencing of tasks and the information exchange between stakeholders has been established, it becomes clear that project requirements are being poorly managed. On one hand, the lack of a management framework for explicitly documenting the decision-making process, the design iterations, and the tasks performed by the civil engineer results in the loss of valuable project information. On the other hand, the already present information such as calculations, reports, and estimations is kept in files (usually PDFs), unsynchronized with the 3D model which makes the traceability of requirement proofs inefficient and prone to errors.

Another aspect which additionally complicates the matter is the fact that when it comes to building structures projects, it is the architect who is more in contact with the client and carries out the first drafts of the project. In other words, the architect manages the demands of the client by incorporating them in the first architectural model that will be later sent to the structural engineer and modeler, who will cooperate in the creation of the structural design. Therefore, the requirements associated with the structure are a derivative of the architectural design.

When comparing both situations, it can be concluded that the roles of the engineer in civil and in building structures projects are slightly different. In civil projects, the engineer plays a primary role in the design processes and is usually one of the parties involved in the project from its early beginning. Therefore, he/she is in line with the demands of the client. In a building structures project, where the architect has the lead, requirements are already managed and defined by him/her in the elicitation process with the client at the beginning of the architectural design and therefore, the engineer plays a secondary role in the

requirement's management process. This scenario is typical for the traditional type of building contracts which is still widely used for building structures projects across the industry.
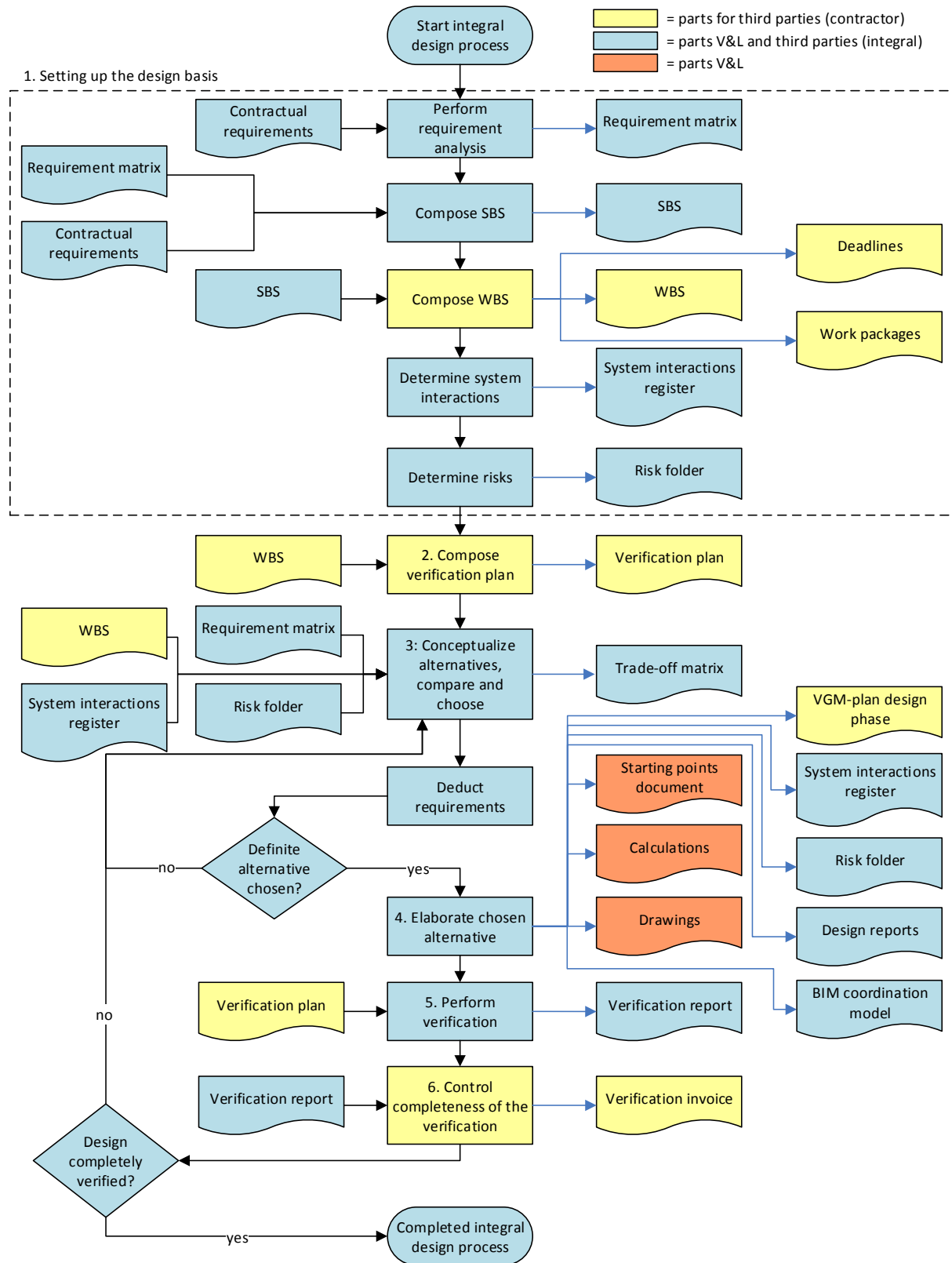

Figure 16: Design process flow chart with the use of SE principles (V&L)

7.1.2 Design phases

The project's design phases, the Levels of Detail/Development (LOD) associated with them, as well as all actors involved in the process and their responsibilities are defined in the Quality Manual (*Dutch: Kwaliteitshandboek*) of the company. This manual clarifies how each design phase should be carried out and which information/documents should be transferred between the collaborating parties. The manual also distinguishes between building structures projects and civil projects.

The different project design phases for building structures projects in which the structural engineers are usually involved are the following:

- **VO = Preliminary Design** *(Dutch: Voorlopig Ontwerp)* **= LOD200**
- **DO = Final Design** *(Dutch: Definitief Ontwerp*) **= LOD300**
- **UO = Execution design** *(Dutch: Uitvoering Ontwerp)*
  - **UO-PV = Design Execution** *(Dutch: Uitvoering Ontwerp – Productie Voorbereiding)* **= LOD350**
  - **UO-WT = Implementation Design** *(Dutch: Uitvoering Ontwerp – Werktekening)* **= LOD400**

The corresponding LOD levels for the different design phases of a building structures project and the tasks performed by both engineers and modelers are also outlined in the table below.

Table 5: Responsibilities of the engineer and modeler based on the LOD

L200 (VO)	
In this phase, the starting points for the structural design are defined.	
Modeler	The modeler makes 2D drawings of the primary supporting structure based on the 2D drawings (floor plans, sections) and 3D model provided by the architect.
Engineer	The structural engineer makes basic calculations of the loads for each floor based on the 3D model and 2D drawings (floor plans, sections) of the architect and also in consideration of the room categories in the building.
L300 (DO)	
In this phase, the structural design and the corresponding calculations are prepared.	
Modeler	The modeler creates the 3D structural model of the primary supporting structure, as well as the drawings of the construction details with the help of Revit or Tekla Structures.
Engineer	The engineer prepares the official engineering design calculations, consisting of calculations of individual elements and/or systems with their corresponding forces and reactions; he also determines the profiles and material quality of the elements.
L350 (UO-PV)	
In this phase, the structural design is prepared for further development by third parties. Data for this purpose shall be provided by other parties (contractor, architect, MEP consultant/engineer etc.).	
Modeler	The modeler applies the necessary changes/additions to the model including the deviations related to the structural openings.

Engineer	The engineer must calculate the openings and recesses in the structure which are classified into three categories and namely, structural, architectural and installation openings.
L400 (UO-WT)	
In this phase, the works for the in situ concrete are elaborated on for the execution of the structural design.	
Modeler	If possible, the modeler adds reinforcement data in the 3D model in Revit/Tekla. Otherwise, the same information is displayed in the 2D structural drawings.
Engineer	The drawings of the formwork for the concrete components poured in situ including measurements are prepared and eventually, a plan of the piles is also created.

7.1.3 Systems Engineering practices

Systems Engineering is an interdisciplinary approach for the management and organization of information within a complex system. The definition of the term, provided by INCOSE (International Council on Systems Engineering) (INCOSE 2015) defines it as 'An interdisciplinary approach and means to enable the realization of successful systems. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.' The main benefit of Systems Engineering for the AEC domain is the possibility to capture, structure and organize vast quantities of information while making the information easily traceable and accessible through the implementation of a tree structure. Although there can be various types of tree structures related to a construction project (e.g. process, requirements, objects), the ones that are useful for this thesis and therefore, will be used in this investigation, are the System Breakdown Structure (SBS) and the Requirement Breakdown Structure (RBS).

The SE handbook from V&L is based on the third edition of the official guidelines for Systems Engineering (ProRail, Rijkswaterstaat, and Bouwend Nederland 2013) written by six government and market parties in the Netherlands. According to the guidelines, the SBS is a hierarchical description of the physical parts of a certain structure, while the RBS is a summary of all the identified requirements in the project which should be directly related to the components from the SBS object tree. The SBS divides an entire system into different parts until an individual system element is obtained and the system cannot be further divided into components. Each component should obtain a unique number, which serves as an identifier and in addition, it should be assigned to a discipline responsible for its completion (e.g. structural engineering, MEP, architecture).

When talking about requirements, however, a distinction needs to be made between the different types of requirements that the SE guidelines identify. As displayed in *Table 6* below, there are 5 general categories for the classification of project requirements.

Table 6: Systems Engineering requirement categories

Functional requirements *(Dutch: Functie-eisen)*	Requirements relating to the functions which need to be realized; they indicate 'what the system should do'.

Aspect requirements *(Dutch: Aspecteisen)*	These are requirements relating to supporting functions or aspects of the system. For example, requirements regarding management and maintenance, design, and stability of the system.
Object requirements *(Dutch: Objecteisen)*	Requirements related to objects that have an impact on, for example, the shape, color, strength, and dimensions. These requirements arise as a result of the design choices of client and contractor.
System Interaction requirements *(Dutch: Raakvlakeisen)*	Requirements which come as a result of relations between the system and the system's environment (external requirements), as well as from interactions between the different components of the system (internal interactions/clashes). *An example of external interactions could be the nuisance caused to the neighbors of a construction site; an internal interaction could be the clash between physical objects in the model.*
Process requirements *(Dutch: Proceseisen)*	Requirements for activities which are necessary to be performed in order to successfully and timely achieve an objective (e.g. piling may take place from ... to ... hours).

To each requirement, a unique identification number is assigned, as well as a requirement description, parent requirement, responsible person and connection to an object (or objects) to which the requirement should be applied.

Each of the aforementioned requirement categories, with the exception of the Process requirements category, is later subdivided according to the hierarchy in *Figure 17*, depending on the level of detail of the project.
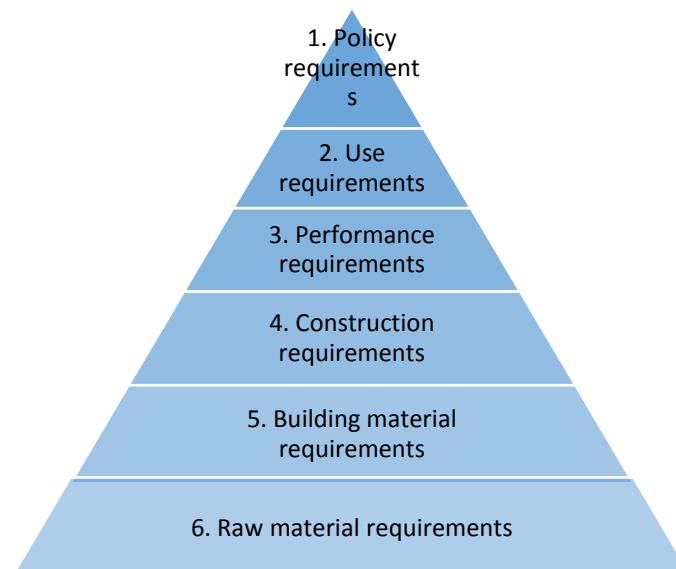

Figure 17: Requirements pyramid

1. **Policy requirements**

These are the requirements regarding amongst others, capacity and social security. They are intended for planners, urban designers, policy makers etc.

2. Use requirements

They relate to the functioning of a building structure. Examples for this are variations in movement, comfort level or safety. These requirements are inputs for architects and traffic engineering designers.

3. Performance requirements

They provide information on the expected performance of a structure. For example, they concern the embankment of pavements and are the basis knowledge which the structural engineer needs.

4. Construction requirements

They relate to the behavior of the structure, its sustainability, strength and stiffness, distortion, and are also part of the input for the designer.

5. Building material requirements

They determine the choice of materials and apply to the planning engineer and the contractor.

6. Raw material requirements

They relate to the raw materials comprising the various building materials. They are described in terms of tensile strength, maximum elongation, or particle-size distribution and are the necessary information for the manufacturers or for the sub-contractors.

7.2   Case studies

Three of the company's projects were studied during the research stage of the thesis in order to find similarities and differences not only in terms of requirements management but also in regards to the contents of the projects' documentation (e.g. calculations, advisory reports). The first two projects belong to the building structures division, while the third one is a part of the civil division. The civil project was used as an example of the successful implementation of the SE principles in a construction project. The three projects used are the following:

1. **Academy Vanderlande te Veghel** (Academy) – Building structures
2. **Nieuwbouw MAVO Schravenlant XL te Schiedam** (School) – Building Structures
3. **Fietsparkeergarage Vijfhoek** (Bike parking) - Civil

7.2.1 Process

Regarding the civil project *(Fietsparkeegarage Vijfhoek)*, Systems Engineering for the management of requirements was implemented through Relatics since its very beginning. As a result, the engineers working on the project used an SBS and an RBS for the organization of the work packages, for clash detection, and for keeping an updated record of the different tasks performed during the different design stages. The tree structures of the SBS and the RBS were used to gain an insight into how requirements are being formalized and related to physical objects from the design.

Upon taking a look at the building structures projects, only the school project *(Nieuwbouw MAVO Schravenlant XL te Schiedam)* counted on the principles of Systems Engineering for the management of project requirements. The way requirements were documented, however, consisted of an Excel sheet, initially provided by the company in charge of the preliminary design of the project. Therefore, the way of using SE was quite inefficient. Regarding the Academy project, no SE practices or any other type of tool for the capturing and managing of requirements information was used.

Here it should be also mentioned that the design phase at which V&L started working on each of the two building structures projects differs. To clarify further, the company was involved in

the Academy project since the preliminary design phase, while the preliminary design of the MAVO project was created by another company and V&L were involved only in the final design. This fact has additionally impacted the way requirements are organized in both projects because in the case of the MAVO project, where project teams changed, the continuity between the preliminary design phase and final design phase had to be assured and therefore, an attempt at structuring requirement information was made.
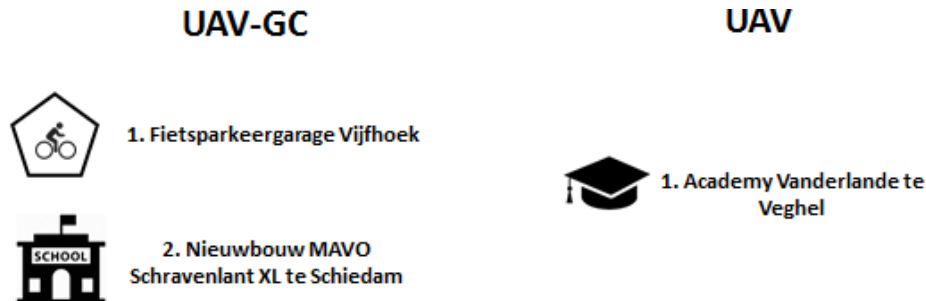

Figure 18: Case study projects in relation to the two types of contracts

Besides the already mentioned, the legal structure of both building structures projects is also different due to the difference in contractual agreements. The contract type of the Academy project is a UAV contract, which presumes a traditional, or in other words a sequential way of working where the architect leads and all other domains follow up on the work of the architect without having an impact on the architectural design. The MAVO project, however, has a UAV-GC contract, which indicates an integrated way of working between the project's stakeholders and therefore, the all of them participate in the decision-making processes in regards to design. In *Figure 19* below, the contractual relations between the stakeholders in both of the projects are presented.
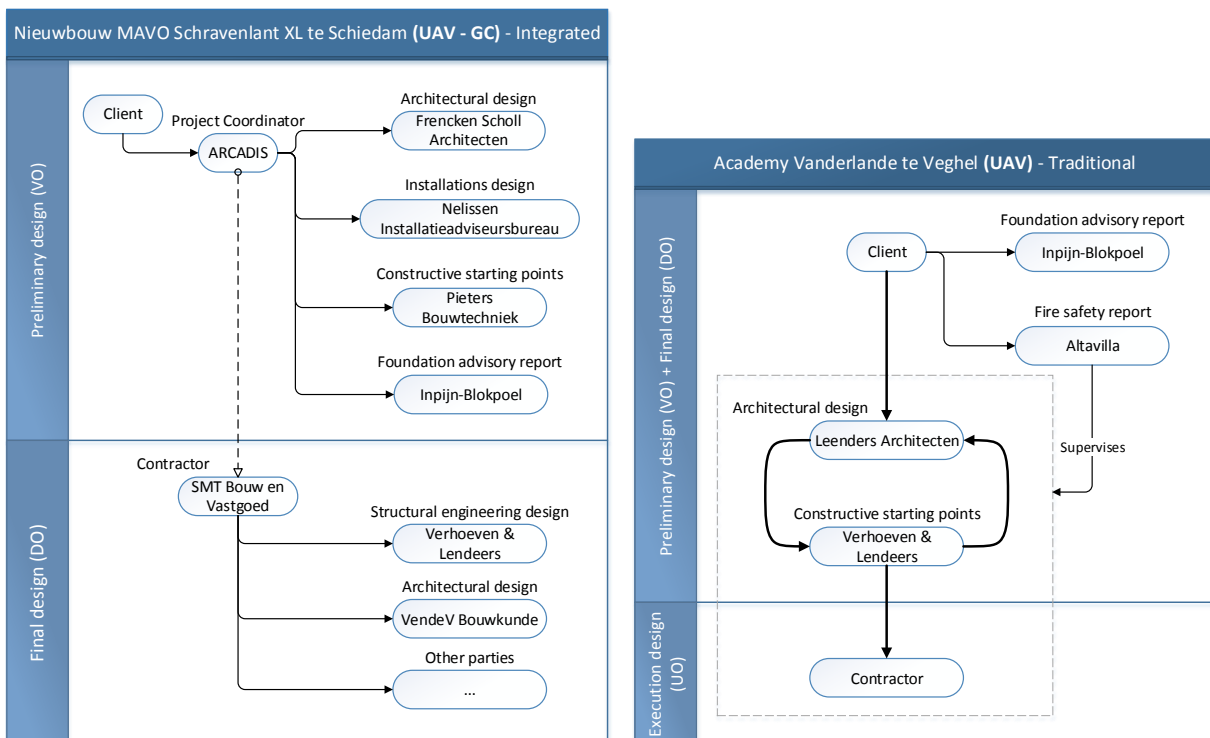

Figure 19: Contract collaboration MAVO (left); Traditional contract collaboration Academy (right)

7.2.2 Documentation

When revising the documentation of the two case studies, the calculation documents, the documents describing the starting points of the structural design, and the foundation, fire safety, and soil advisory reports contained the most important information for the accurate capturing of engineering requirements data. Due to the significant differences in the organization of information between the documents of the two case studies, their analysis was supported by the two engineers who calculated the structures.

By comparing the documentation of the two projects, the first draft of the general engineering requirements was created and categorized by the Systems Engineering requirement categories from *Table 6*. The draft was brought to a more precise level by comparing it with the official regulations for engineering structures in the Netherlands – the Dutch building law and more specifically, the Eurocode.

In order to receive a better overview of the information which was exchanged during the preliminary and final design phases, the sequence in which it was exchanged, the parties who generated it, and the parties who used it as an input for their activities, process and input/output schemas were created. In *Appendix II: MAVO communication workflow (preliminary design), Appendix III: MAVO communication workflow (final design), Appendix IV: MAVO input/output workflow (preliminary design) and Appendix V: MAVO input/output workflow (final design)* the schemas of the MAVO project are presented.

7.3 Codes and regulations

As already mentioned, the initial draft of general engineering requirements was created on the basis of the documentation from the two building structures projects. Due to the subjectivity of using a limited amount of case studies, however, the results had to be additionally supported. Therefore, a look at the Dutch building law and the Eurocodes was taken. The aforementioned were also used in the research of defining general project requirements, which apply to the type of structural element but are unrelated to the material from which the element is made of.

The Dutch building code (Bouwbesluit) is a collection of building regulations that all buildings in the Netherlands, such as homes, offices, shops, hospitals, etc. must meet. Exotic materials, exotic buildings, existing buildings, and big clients (such as governmental institutions) may have additional requirements. As the building code is the actual Dutch law which all parties in a project are obligated to follow, the Eurocodes, according to the law, are just one of the ways to prove the strength of a structure and they specifically refer to the structural engineering requirements that a building must comply with. This would suggest that the Eurocodes do not contain requirements regarding, for example, the amount of air or light that a certain room should have.

The Eurocodes are ten European standards which specify how structural design should be conducted within the European Union (EU). Their purpose is to provide a way of proving that a structure complies with the requirements for mechanical strength, stability, and safety in a situation of fire, as well as to serve as a basis for engineering contract specifications (Wikipedia, 2017).

For determining the validity of requirement information gathered from reviewing the calculation documents of the two case studies mentioned in the previous section, a look into

all 10 Eurocodes was taken. From *Table 7* presented below, it becomes evident that each Eurocode specifies a set of requirements connected to the materials used in a specific structure. Therefore, these specifications were useful for determining which engineering requirements are material-neutral, or in other words, which requirements are common for the calculation of specific building component regardless of its material type. The Eurocodes and their corresponding sub-sections can be found in *Appendix I: Eurocodes*.

Table 7: Eurocodes on structural design for the European Union (EU)

Name	Description	Standard
Eurocode	Basis of **structural design**	(EN 1990)
Eurocode 1	Actions on **structures**	(EN 1991)
Eurocode 2	Design of **concrete** structures	(EN 1992)
Eurocode 3	Design of **steel** structures	(EN 1993)
Eurocode 4	Design of **composite steel and concrete** structures	(EN 1994)
Eurocode 5	Design of **timber** structures	(EN 1995)
Eurocode 6	Design of **masonry** structures	(EN 1996)
Eurocode 7	**Geotechnical** design	(EN 1997)
Eurocode 8	Design of structures for **earthquake** resistance	(EN 1998)
Eurocode 9	Design of **aluminium** structures	(EN 1999)

7.4   Interviews

Throughout the entire process of requirement analysis and formalization, several interviews with engineers were conducted in order to additionally validate the conclusions drawn throughout the research. The interviews helped to clarify the connections between requirements and building element groups such as columns, beams, floor slabs etc. Besides that, a group session with several of the previously interviewed engineers was held at the end of the research, in order confirm the soundness of the results and in order to assure that the formalized schema is generic enough in order to be applied to a wide variety of building structures projects.

The three most important aspects for consideration during the group session were related to validating the objectives of the conducted research and namely:

- Is all of the information necessary for the handover to external parties (such as sub-contractors, principal, audit bodies etc.) captured in the results?
- Are all of the specified requirements generic?
- Are the classification of requirements data and its relation to building element types accurate and explicit enough?

The initial selection of requirements was therefore reduced and reorganized with the help of the MoSCoW method which is a prioritization technique used for management, business analysis, and software development. The method distinguishes between four categories of rules and namely, the 'must haves', the 'should haves', the 'could haves' and the 'won't haves' (DSDM Consortium 2008).

As the first category includes requirements which in any case have to be incorporated, the second category represents the requirements which would be strongly desirable to be incorporated. The requirements which fell under these two categories were included in the final requirements schema. Examples of 'must haves' are the forces which each element should bear and examples of 'should haves' are usability requirements such as crack width, or nuisance impact on the surroundings of a structure during construction.

7.5   Results

To sum up, the process of determining the general project requirements for building structures projects, as presented in *Figure 20*, involved a number of activities, the combined results of which were incorporated into a single schema, mapping project requirements to building element types.
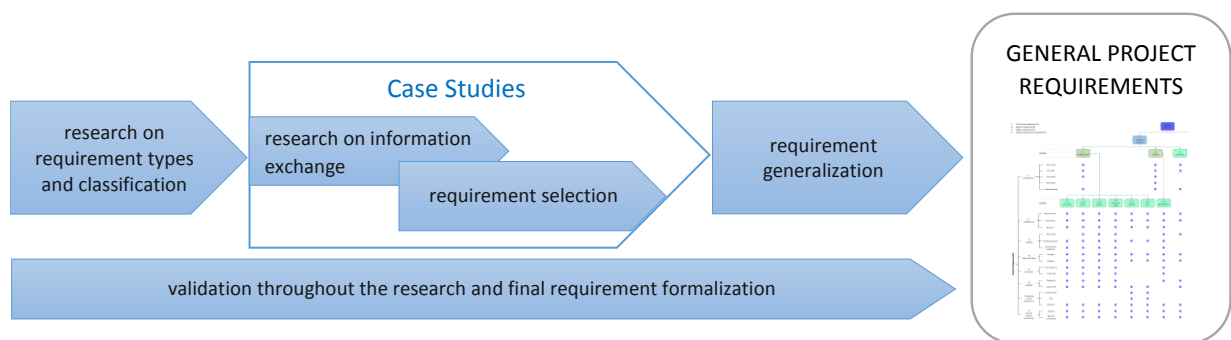
Figure 20: The activity sequence for determining the general engineering requirements

The schema, displayed in *Figure 21*, captures the general engineering requirements on structural elements which need to be proven at the end of the final design phase of a building structures project. This information also has to be delivered to external parties involved in the project phases following the design.

The schema represents a combination of an SBS (System Breakdown Structure) and an RBS (Requirements Breakdown Structure). In the SBS only the components of the primary structure of a building are considered due to the fact that they are the essential components which carry the load and on which the stability of the structure depends. By following the previously revised calculation reports, the primary structure is calculated both for the preliminary and for the final design phases with the difference that the estimates have a different level of detail.

In the preliminary design (LOD 200) the loads on the structure are calculated in relation to their impact on the entire building level, or respectively, on the entire surface of the facade. In this level of detail, the only components calculated individually are the staircases due to the fact that they don't belong to one particular storey.

In the calculations of the final design (LOD 300) the structure has been calculated in regards to the individual elements or systems of elements. The structural elements on a building level can, therefore, consist of columns, beams, floor slabs, structural walls, footings or piles. The only structural element on the facade, part of the primary structure and responsible for its stability - the wind braces, are also included in the schema.

From the in-house investigation, it became clear that not only the project requirements, but also the engineering systems of a structure are only in the minds of the engineers as tacit and implicit knowledge. Depending on the personal preferences of each engineer, every requirement applicable to a structural component can be proven in several different ways.
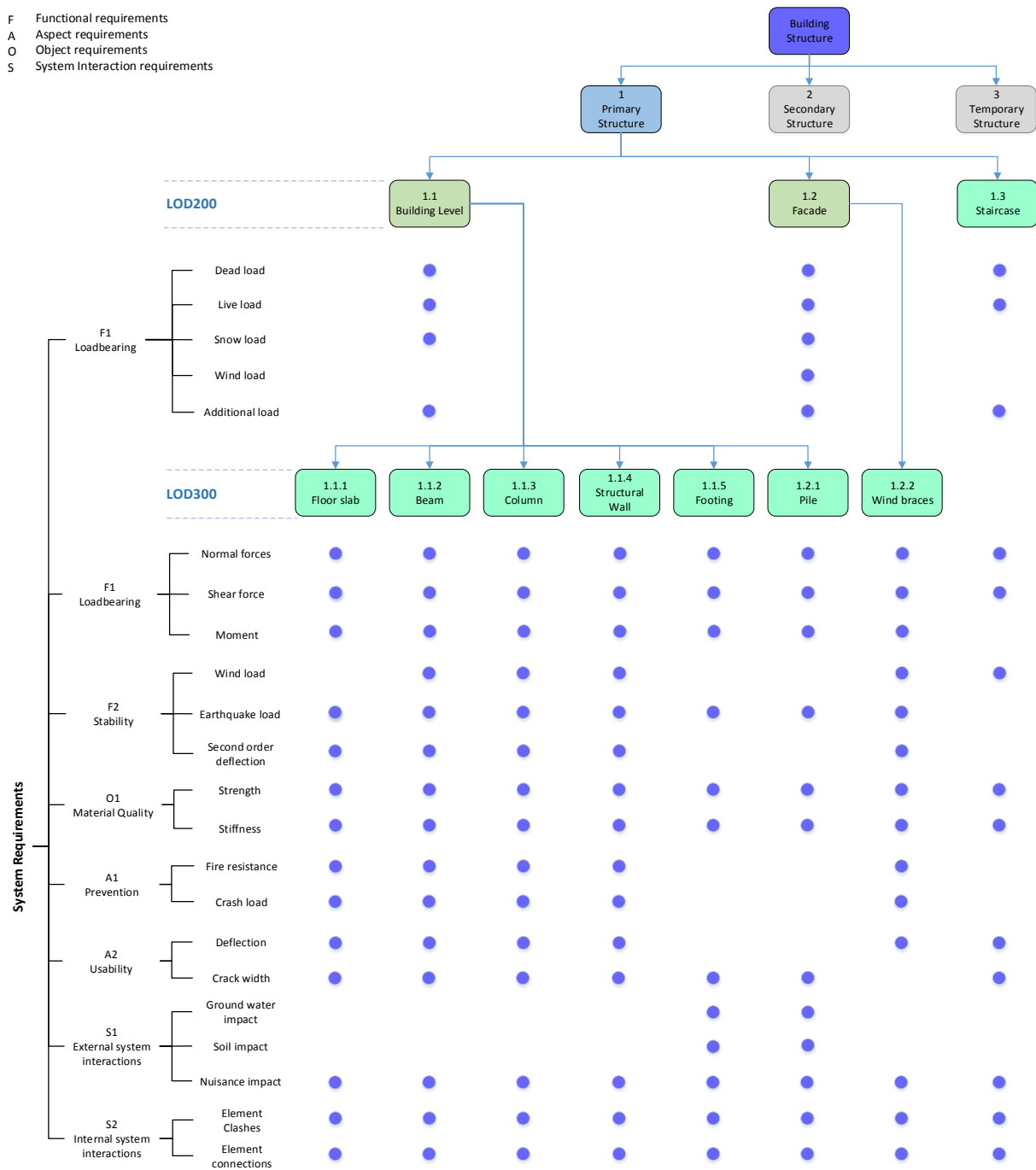

Figure 21: General engineering requirements on structural elements which need to be proven at the end of the final design phase of a building structures project

The building's stability can be proven by calculating single components on their own, by considering them as a part of a system, or even by calculating the entire structure as a whole.

In addition to that, a single engineering component can be proven by different systems based on the preference of the engineer who makes the calculation. Therefore, the notion of systems can be discovered only in the documentation of the project and it remains an aspect, highly dependable on the choice of the engineer. The geometrical model does not indicate systems and systems-relevant requirements are hard to be distinguished from the component-relevant requirements. For that reason, the systems approach to linking requirements requires a research on its own before being a plausible solution for this thesis and therefore, it was considered unreliable for keeping the completeness of the information.

The RBS differentiates between four requirement categories, consistent with the SE categories types discussed in *section 7.1.3* and namely, functional, aspect, object and system interactions requirements. The functional requirements are divided into two groups – '*loadbearing*' and '*stability*', the aspect requirements consist of the categories '*prevention*' and '*usability*', the object requirements refer to '*material quality*', and the system interactions requirements are divided into '*internal*' and '*external*'. Each of the aforementioned types incorporates several requirements which, as specified by the dots in the schema, certain component types need to comply to.

It is important to note that the dots represent the 'must haves' and 'should haves' in terms of which requirements need to be proven before the information handover to other parties takes place. The fact that some objects are not connected to a specific requirement by a dot does not indicate that the connection will never exist. It indicates that the information is not necessary and therefore, not calculated or demanded by the Dutch building law or the Eurocodes. Therefore, proving the validity of these relations is also unnecessary for the handover to external parties.

The main use of the formalized mapping between engineering requirements data and structural components is to determine the absolute minimum of information which should be provided and at the same time proven by the structural engineering team at the end of the final design. The creation of additional information is not excluded as a possibility, however, its presence or absence wouldn't impact in any way the validating of the stability of the structure or the work of external parties involved in the following stages of the project.

The requirements schema is the main input for the next step of the research and namely, the creation of an ontology and the mapping between project requirements, document proofs and the BIM model.

7.6   Conclusion

As mentioned previously, the process of determining the general project requirements for building structures projects resulted in a general requirements schema mapping engineering requirements to building element types. The main use of the schema created is to represent the 'must haves' and 'should haves' in terms of which requirements need to be proven before the engineering design handover to external parties takes place. The formalized matrix and the defined relations within it, however, shouldn't be accepted as a static or constant and the information it represents shouldn't be considered as exhaustive as in the different projects there will always be project-specific requirements that would require additional consideration. In the limitations and recommendations chapter of this thesis, some of the points for improvement will be introduced.

8   Tool development

This chapter discusses the development of a tool which enables the mapping between engineering requirements and model components based on the previously formalized requirements schema from *Figure 21*. The main use of the tool is to create a connection between the project's documentation and the BIM model for the better traceability of the relations between requirements and structure and for monitoring the process of requirement validation. The tool contributes to improving the efficiency within the company but also helps to find a practical solution for the design handover to external parties, where the integration between design and engineering data can reduce ambiguity and prevent risks.

Kiviniemi (Kiviniemi 2005) is one of the first researchers who recognize the advantages of creating a link between requirements and the product model. He states that even a simple connection between them can "increase the usage of requirements documentation throughout the design and construction process". The results of the qualitative research presented in *chapter 6*, also additionally support the need of creating a link between requirements, project documentation and object instances, which can lead to a more interoperable BIM.

For the proper implementation of the tool, first of all, the requirements matrix from *Figure 21* is translated into an information model, which facilitates the mapping between requirements and model instances. Secondly, the prototype tool is used to visualize the geometry model and display the requirements to which every component should comply to. In order to confirm that a certain object instance from the model has been verified against a specific requirement, the options to attach document proofs to the component and relate them to the requirement(s) that they prove for that component are facilitated by the tool. Once all elements in the model have been proved in relation to all requirements they must fulfill, the design handover to external parties can be carried out.

For the development of the information model and the application, the following programming software and modules were used:

- TopBraid Composer;
- Python 2.7, with the following libraries:
  - Python OCC 0.16;
  - PyQt 4;
  - IfcOpenShell 2.7-0.5.0;
  - RDFLib 4.2.1.

Parts of the Python source code from (van de Ven, N. 2017) was used as a basis for the development of the tool. In addition, for adapting the tool to the company's needs and practices the following platforms were also related to the prototype solution:

- RDF4J Workbench (Triplestore Server);
- FileZilla Client (FTP Server).

8.1   Case description

As a case study model is used for the development part of the thesis and namely, the new construction project of a secondary school for approximately 450 students - Mavo

Schravenlant XL te Schiedam (also used in the requirements research phase presented in *chapter 7*). The building consists of three storeys and incorporates an indoor gym located on the ground floor. The design of the school is also characterized by the protruding floors situated around a large central gallery with a skylight in the roof above it.

Mavo Schravenlant XL will be realized in the summer of 2017 and it intends to accommodate students, teachers, and staff. For this project, the existing buildings at the Burgemeester van Haarenlaan 952 and the Van Hogendorpstraat 103 and 105 in Schiedam will be demolished. The new building of approx. 4.000 m² gross floor area and the gym will be developed on the site emerged after the demolition.

Figure 22: MAVO Schravenlant XL (Frencken Scholl Architecten)

8.2   Process guidelines for the tool implementation

Based on the research conducted for the formalization of project requirements, a good understanding of the engineering practices in the final design phase of a building structures project was also obtained. With regard to this knowledge, a process schema serving as a guideline for the tool implementation was created. The schema integrates the functionalities which the tool must fulfill with the commonly performed engineering tasks.

Important to mention is that this thesis is complementary to another graduation project (Bernal, 2017), which focuses on the internal design validation through BCF and therefore, in the process schema, presented in *Appendix VI: Process schema – guidelines for the tool implementation,* the complete integration of tool functionalities from the two projects is presented.

The tasks relevant to the topic of this thesis, colored in red, consist of the visualization of the IFC geometry and the building element properties, the visualization of geometry in relation to proven and open requirements, and the linking of documents to both requirements and to model instances as requirement proofs. In addition, the ability to visualize the model instances based on their compliance with a specific requirement, as well as the ability to link document updates to already attached document proofs, based on the feedback of external parties after the design handover, must be also facilitated.

As the schema's main purpose is to present an overview of the sequencing of tasks after the integration of the tool within the design process of a building structures project, the use case diagram presented in the following section, elaborates on the process map by focusing on the exact functionalities of the tool, solely related to the topic of this thesis.

8.3   Use case diagram

In order to demonstrate how the user, in our case - the structural engineer, interacts with the tool, a use case diagram capturing the activities which he/she can perform is presented in *Figure 23* below. The diagram illustrates the core functionalities of the tool for handling requirements data and also the use of semantic web technologies such as RDF, ifcOWL and SPARQL for connecting BIM with external data repositories like the triplestore, where the requirements data is stored in the form of RDF triples. The triplestore contains not only requirements data but also the converted into ifcOWL geometry data of the specific project. Upon selection of a repository, the IFC file loaded in the tool by the structural engineer is linked to the requirements data and the converted geometry in the triplestore, allowing the IFC model instances to be queried for the requirements they need to comply with.

As already mentioned, the main user of the tool is the structural engineer, who is responsible for the specific project. He/she is, therefore, able to visualize and connect the IFC geometry with the RDF data stored online. Apart from that, through the tool, the engineer is able to browse a document of proof from the company's file server and link it to the element or to the multiple elements it relates to, while also being able to specify which requirements are being proven by it. The data generated by the engineer is stored as additional triples in the triplestore and the document is uploaded to an FTP server. In addition to that, the engineer can visualize the proven and open requirements for a particular model instance and query the model for instances which are linked to a particular requirement and/or proven on it.

It is important to specify that the process of attaching a document is intended to happen when the design component in question is fully calculated and a final version of the calculation report has been created. Only under these circumstances can the calculation document be a legitimate requirement proof which will be delivered in the design handover to external parties.

Furthermore, following the information handover process after the completion of the engineering design, in the case of a need based on the feedback from external parties, the engineer can update the version of a specific document previously linked. The previous version of the document remains in the triplestore and in the FTP server for the purpose of data traceability. The update is then also stored in the FTP server and the new information is also added to the triplestore.

In this case, the update of a calculation would suggest that a specific element or system, according to external parties, didn't meet certain criteria even after the finalized calculations of the structural engineer. As the practice shows, such scenario usually concerns a very limited amount of elements, which can be labeled 'problematic' and therefore, for the purpose of tracking these design components, the model can be queried for the elements with document updates. And the requirements on which these elements had to be recalculated can be also displayed, as well as the link to the FTP server location of the document update, which can be copied to a browser and opened from there.
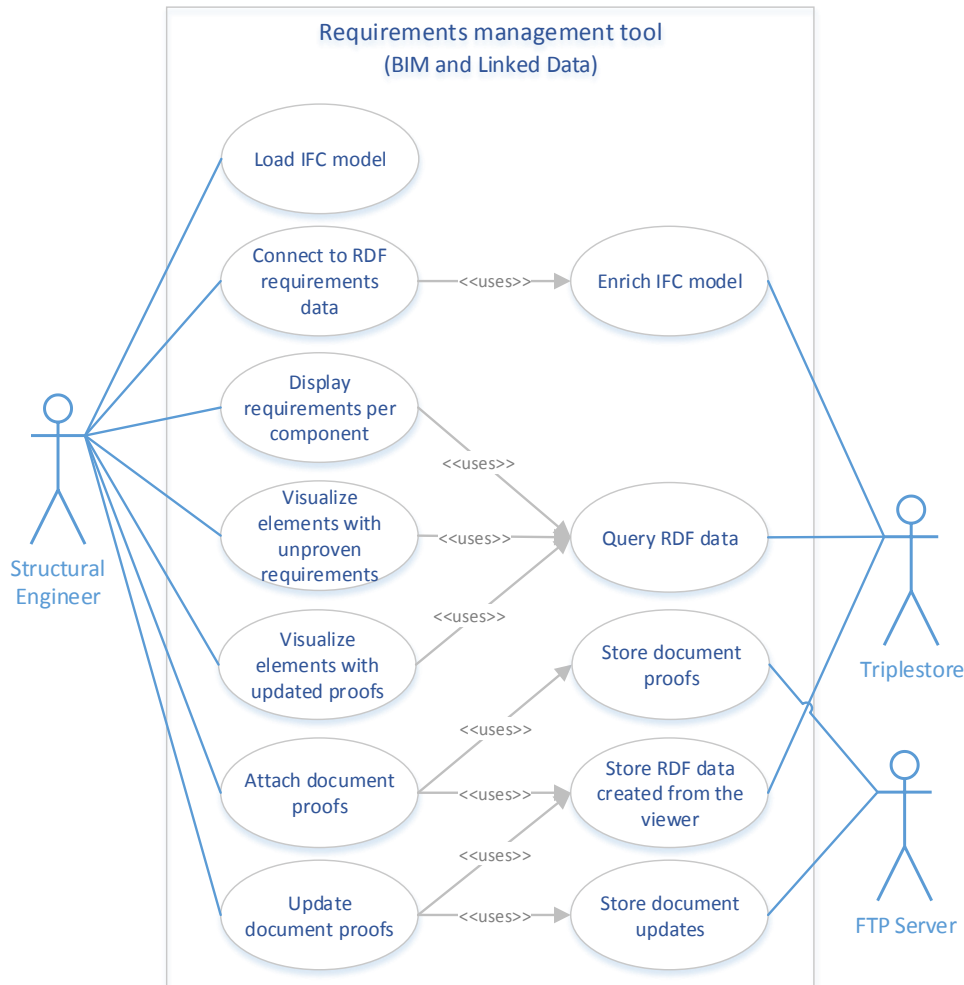
Figure 23: Use case diagram of the tool in connection to the triplestore and the file server

8.4 Ontology engineering

According to (Corcho and Fernandez-Lopez 2003), ontological engineering refers to 'the set of activities that concern the ontology development process, the ontology life cycle, the principles, methods, and methodologies for building ontologies, and the tool suites and languages that support them'.

As an ontologies' intended use is to capture domain knowledge in a generic way with the aim of providing a mutual understanding of that domain, ontologies also present the opportunity to share and reuse that knowledge across applications and groups (Pinto and Martins 2001). Reusability of existing knowledge resources as an input for building new domain ontologies has been acknowledged as an essential practice of ontology engineering and therefore, the first step of creating a new ontology should begin with research on knowledge resources whose domains overlap with its own target domain. Based on the content of the sources and the extent to which they overlap, there are two different reuse processes – merge and integration (Bontas, Mochol, and Tolksdorf 2005).

The main principles for ontology design according to (Corcho and Fernandez-Lopez 2003) are clarity in the intended meaning of defined terms, minimal encoding bias by relying upon actual terminology from the domain in question, and extendibility of the existing vocabulary. In addition, the authors also mention the importance of coherence in inferencing and the

minimal ontological commitments, achieved by defining only the essential domain knowledge, critical for the communication of information.

The main components of an ontology are classes, attributes, relations, and individuals, while ontologies might also include restrictions, axioms, rules, and events. The components are encoded using ontology languages such as OWL, RDFS, ifcOWL, covered in *chapter 5* of this report.

Although COINS and the SE reference framework SE-BIM handle project requirements and documentation, their data structure incorporates a really wide array of generalized concepts which also incorporate validation and verification practices, risks, stakeholders etc. The two standards aim at identifying all concepts within the construction field and the relations between them which creates an enormous data structure to which the user (e.g. engineers) need to adapt their data. Therefore, COINS and SE-BIM can be described as a top-down approach for the formalization of building information.

An agreement within the COINS group, however, suggests that this approach is not sufficient for considering the standard universally applicable. Both standards together represent a really large ontology, the structure of which needs to be tested with use cases from the industry. This approach can validate the correctness and completeness of definitions and can be described as case-study driven or as a bottom-up approach. The bottom-up approach consists in using company-related data as an input and structuring the data in the most convenient and logical way so that the structure can be compared to both standards where the issues and discrepancies can be discovered and the ontological definitions can be adjusted accordingly.

Therefore, it should be specified that this research project focuses on the bottom-up approach by developing an ontology which corresponds to the case-study based information taken from the previously formalized requirement schema (*Figure 21*, *section 7.5*). This chapter presents how the schema can be formally expressed as an information model and subsequently, associated to the IFC geometry of the structural model which has been converted into ifcOWL.

8.4.1 General requirements ontology

As mentioned earlier, the main input for the development phase of the project is the schema, displayed in *Figure 21*. The schema captures the general engineering requirements on structural elements which need to be proven at the end of the final design phase of a building structures project. This information also has to be delivered to external parties, involved in the project phases following the final design.

The schema in *Figure 24* represents a mapping between a general SBS of an engineering structure and an RBS from the requirements data formalized during the process of determining general project requirements (*chapter 7*). In the developed ontology, based on the requirements schema, only the SBS model components from LOD300 were considered as LOD200 covers only building storeys and general loads and the data is, therefore, far from sufficient for an adequate requirements mapping. The building components from LOD300 represent, in fact, the essential parts of a structure, on which the stability of a building depends.

The name of the ontology created is "General Requirements" and it incorporates the hierarchical definition of the RBS from the requirements schema. The ontology uses the prefix

"greq", which stands for "general requirements" and it corresponds to the following Unique Resource Identifier (URI):

```
http://example.org/ontology/generalrequirements#
```

As mentioned earlier, an ontology consists of, amongst others, classes, attributes, relations, and individuals. Classes represent concepts, which are taken in a broad sense and are usually organized in taxonomies through which inheritance mechanisms can be applied. Instances are used to represent elements or individuals in an ontology and belong to a specific class. Metaclasses, on the other hand, are classes whose instances are classes and allow for gradations of meaning (Corcho and Fernandez-Lopez 2003). Therefore, in order to represent the taxonomy of general engineering requirements as organized in the requirements schema, two levels of metaclasses were defined and namely, the class `greq:Requirement`, incorporating four other classes which classify the requirements into the requirement categories from the Systems Engineering guidelines presented in *Table 6* and namely:

```
greq:Functional_requirement,
greq:Aspect_requirement,
greq:Object_requirement,
greq:Systems_interaction_requirement.
```

Each of the aforementioned metaclasses is further divided into subclasses and the most detailed concept classifications from the schema are defined as the requirement instances of those classes. In *Figure 24* below the hierarchical representation of the requirement classes (blue) and the requirement individuals (purple) of the ontology.
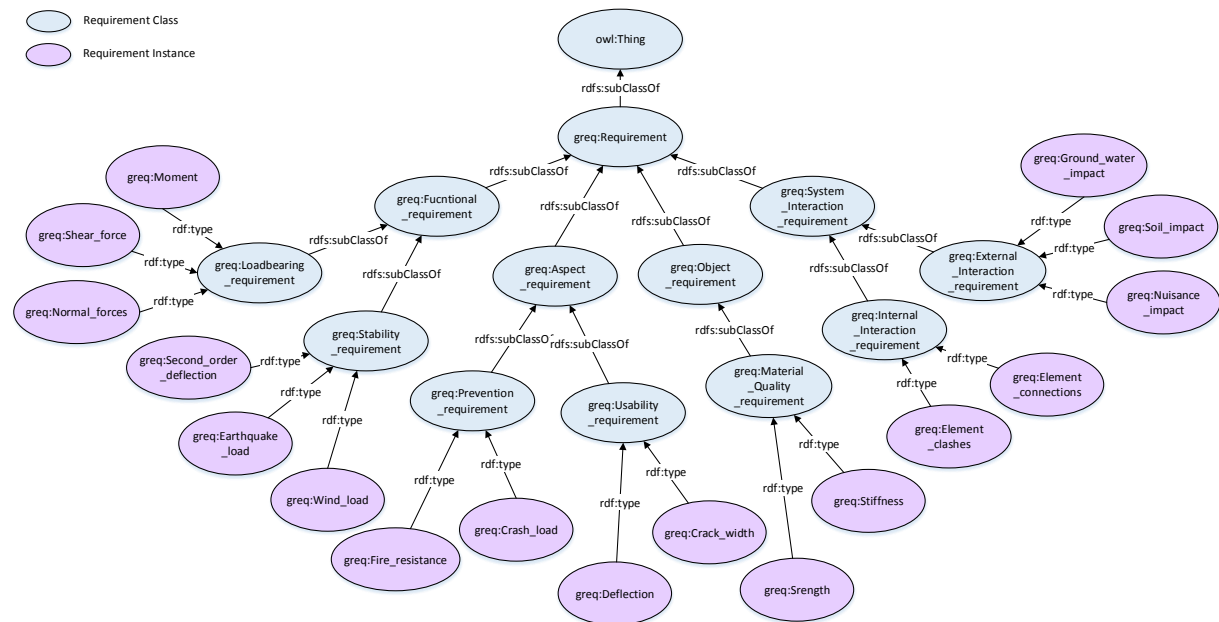
Figure 24: Hierarchical representation of the requirement classes and individuals of the ontology

Relations represent a type of association between concepts of the domain. The individuals in the ontology, namely the project requirements, are related to the building element types from the ifcOWL ontology through the following relation, displayed in red in *Figure 25*:

<div align="center">

`greq:isApplicableToObject`

</div>

The structural components of a building identified from the research - column, beam, pile, footing, floor slab, staircase, wind bracing and structural wall, are respectively represented in the ifcOWL ontology by the following classes: `ifc:IfcColumn`, `ifc:IfcBeam`, `ifc:IfcPile`, `ifc:IfcFooting`, `ifc:IfcSlab`, `ifc:IfcStair`, `ifc:IfcMember` and `ifc:IfcWallStandardCase`.

An example mapping of two of the general requirements from the requirements ontology `greq:Nuisance_impact` and `greq:Soil_impact` to building element classes from the ifcOWL ontology has been shown in *Figure 25*. The `greq:Nuisance_impact` relates to all eight building element types, indicating that all instances of these classes from the geometry model must comply with this requirement. The `greq:Soil_impact` refers to only two of the building element types and thus, only the model instances of these classes must comply with this requirement. The remaining relations between general requirements and building element types in the ontology have been defined in the same manner. Both requirements are also associated to their labels, or identifiers, by the property `greq:hasLabel`. The general requirements ontology can be seen in *Appendix VII: General requirements ontology* of this report.
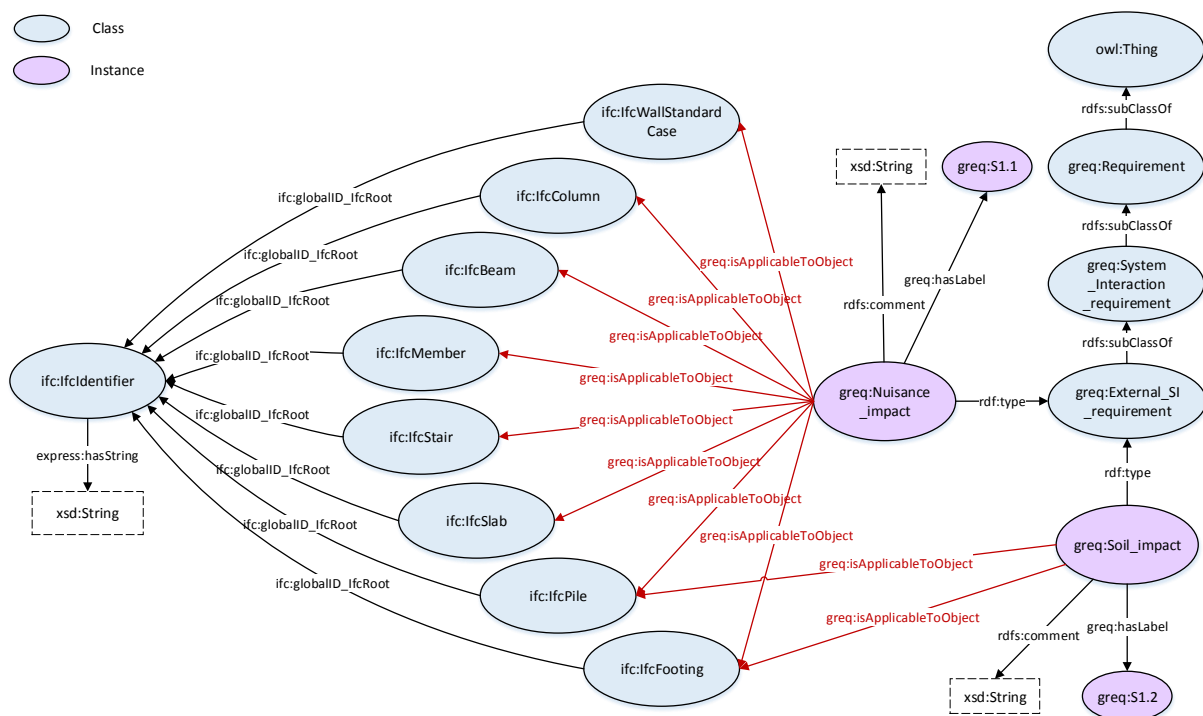
Figure 25: An example mapping between two general requirements from the requirements ontology and building element types from the ifcOWL ontology

8.5   Development of prototype - IFC and RDF 3D viewer

Prior to describing the prototype's interface and its functionalities, the way the desktop viewer interacts with the triplestore and the FTP server, both mentioned earlier, has to be showcased. *Figure 26* below illustrates the connection between the three platforms. An

important prerequisite for working with the 3D viewer is converting the IFC file of the 3D model into ifcOWL and storing the newly generated Named Graph in a repository on the triplestore next to the Named Graph of the previously introduced general requirements ontology. Thereafter, the IFC file of the structural model can be loaded in the prototype from the **Load IFC** button and visualized. Afterwards, in order to establish a connection with the triplestore and the data stored in it, the user needs to select the repository number in which the ifcOWL version of the visualized model has been previously stored together with the general requirements ontology. For doing that, the user needs to press the **Select a repository** button from the interface, displayed in *Figure 27*.
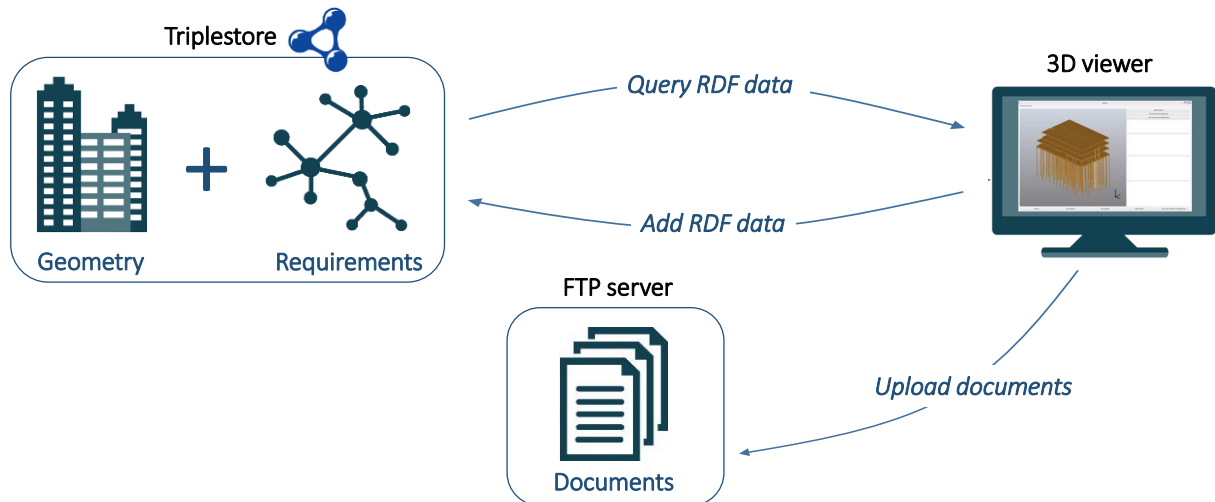
Figure 26: Communication between the 3D desktop viewer, the triplestore, and the FTP server
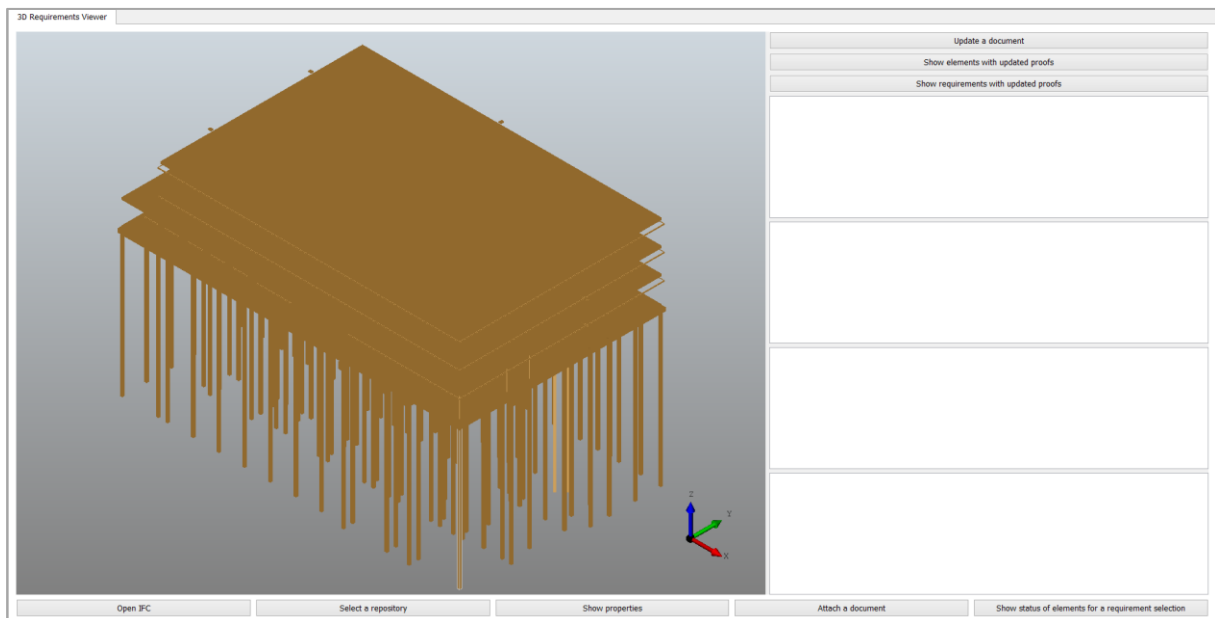
Figure 27: Interface of the 3D requirements viewer with an IFC model loaded into it

By selecting an object instance in the 3D view and pressing the **Show properties** button, the developed prototype enables the structural engineer to view the IFC properties of the elements in the model and in addition, the associated RDF data. The RDF data, stored in the

triplestore, consists of the requirements ontology presented in the previous section and the geometry of the IFC file converted into ifcOWL. When the user selects an element or multiple elements from the 3D view, by the use of SPARQL queries, the viewer is able to display both the open and the proven requirements associated with the selected object(s). The proven requirements also appear with the FTP location of the document that justifies the compliance of the element(s) with the requirement and the link can be copied to a browser for the document to be visualized and inspected.

Thereafter, the structural engineer can attach a document, proving the compliance of an element or group of elements with the certain requirement(s) by pressing the **Attach a document** button. This button opens a dialogue *(Figure 28)* where the user can browse for a document and specify the document type, document fragment, the date, the creator, as well as choose the requirements which the document proves for the specific element(s) selected in the 3D view. It is important to specify that the choice of requirements appearing in the dialogue is always based on the currently open requirements for that element so that no duplication of data can occur. When attaching a document proof to multiple elements at once, the dialogue box gives the user the option to attach a document to these elements only on the requirements that are open for all of them at the same time.
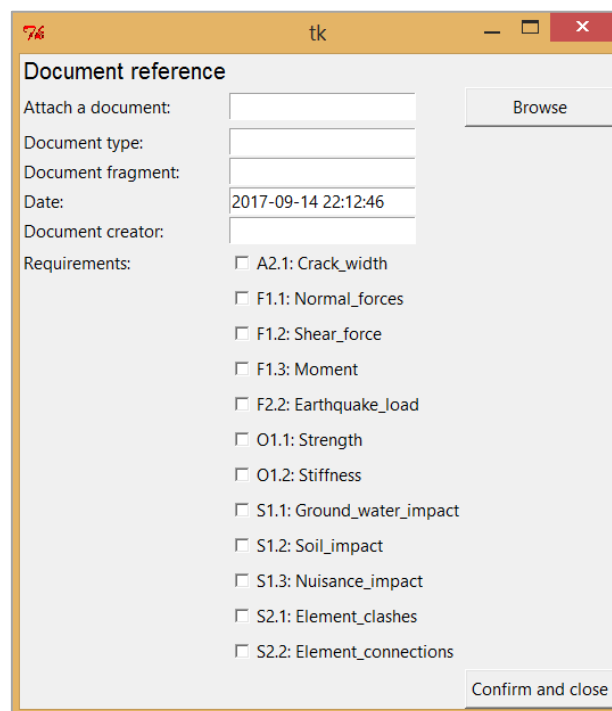
Figure 28: Document reference dialogue box

By pressing the **Close and confirm** button at the bottom right side of the dialogue box, a new context (Named Graph) is being automatically created in the triplestore location selected previously. The new context contains all of the information from the dialogue described in RDF format.

Afterwards, by pressing the **Show status of elements for a requirement selection** button, a drop-down menu with all previously identified general engineering requirements appears.

Upon a requirement selection, the model is queried based on that specific requirement in order to receive a visual validation on which model instances have been proven to comply with that requirement and which model instances still need to be proven on that requirement. The proven elements are colored in green and the to-be-proven elements are colored in red, while all remaining components that are unrelated to the selected requirement remain in their original color.

As indicated earlier, the interface doesn't allow the user to attach more than one document proving the compliance of an element to a specific requirement for the purpose of preventing duplication of data and documentation. The user is, however, able to update the version of a document proof attached previously by selecting an element related to that document and pressing the button **Update a document** which also opens a dialogue box *(Figure 29)*. In this dialogue box, both the document update and the change request which invoked the update can be attached, while also specifying the date, the creator and the change request initiator, and selecting the document to be updated. As for updating a document only the selection of a single model instance related to it is necessary. The dialogue box will display all documents attached to that instance. By pressing **Confirm and Close**, an additional Named Graph encoding the specified data into RDF format is created in the triplestore repository.
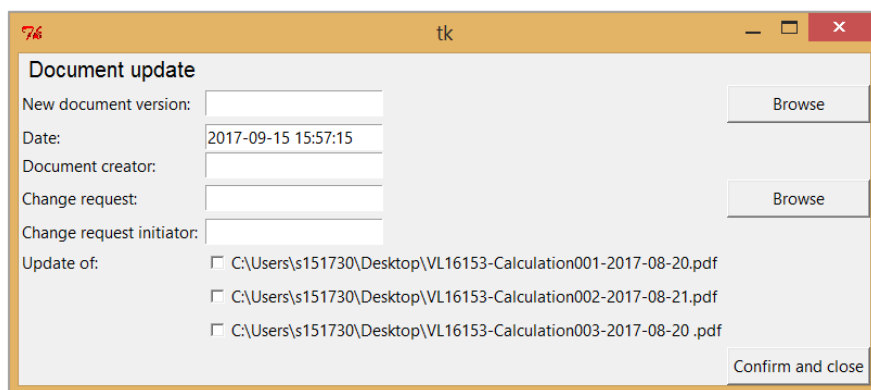
Figure 29: Document update dialogue box

Furthermore, by pressing the **Show elements with updated proofs** button, all elements which have document updates linked to them or in other words, all elements which were recalculated after the final design handover, are displayed in blue. Upon the selection of an element visualized in blue, the button **Show requirements with updated proofs** shows the requirement(s) on which an element has been updated and the link to the document update on the FTP server which can be copied in a browser and inspected.

As specified in *Figure 26*, the desktop viewer communicates with the triplestore by querying the database for information based on which it also creates additional semantic definitions stored in the triplestore. For achieving that, the two main functionalities of the viewer's back end are the ability to create additional semantic definitions and the ability to SPARQL query the server's repository. From the interface buttons discussed so far, the **Attach a document** and **Update a document** functionalities create additional RDF triples in the triplestore, while the rest of the actions performed in the viewer are executed with the help of queries for the retrieval of search-relevant information from the triplestore.

The added triples from the document reference, the update, and the change request are defined by additional definitions, which extend the "General Requirements" ontology presented in *section 8.4.1.* In addition, all of the documentation is being uploaded to an FTP server where the documents are stored in different directories with regard to their purpose and namely, 'Document reference', 'Document update' and 'Change request'.

The following section presents the extension of the "General Requirements" ontology, as well as an example of the linking between an element, its associated requirements and the files related to them (proofs, updates, and change requests). Furthermore, a look is taken into some of the SPARQL queries used in the back-end of the tool. In addition, two flowchart diagrams depicting the two purposes for which the tool can be used are presented in *Appendices XI* and *XII*. At last, the tool validation showcases all of the aforementioned steps in greater detail.

8.5.1 Extending the ontology

Apart from the defined ontological concepts related to engineering requirements, the tool allows the attachment of document references proving the compliance of model instances with the specific requirement(s). Furthermore, based on feedback in the form of a change request, a certain document reference can be updated. All three of the aforementioned files are stored on an FTP server. Therefore, additional ontological concepts, presented in *Figure 30*, were defined and namely: `greq:Document_Reference,` `greq:Document_Update, greq:Change_Request, greq:FTP_Location.`
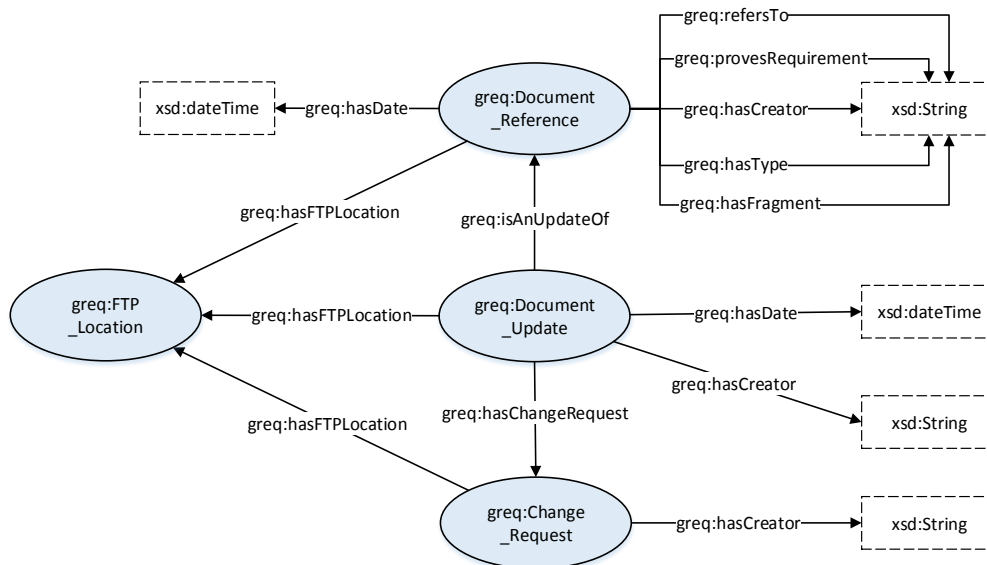
Figure 30: Ontology classes for the attachment of document proofs (part of the tool)

Figure 31 gives an example of the way an object instance (`greq:IfcPile_66981`) is proven by a `greq:Document_Reference` on two of the requirements `greq:Soil_impact` and `greq:Ground_water_impact` it needs to comply with. The document reference uses the predicate `greq:refersTo` to refer to the GUID of the selected element (in the tool) and the predicate `greq:provesRequirement` to relate to the labels of the requirements it proves for that specific element. The

`greq:Document_Reference` itself has an update related to a change request and all three documents have a location on the FTP server (`greq:hasFTPLocation`).
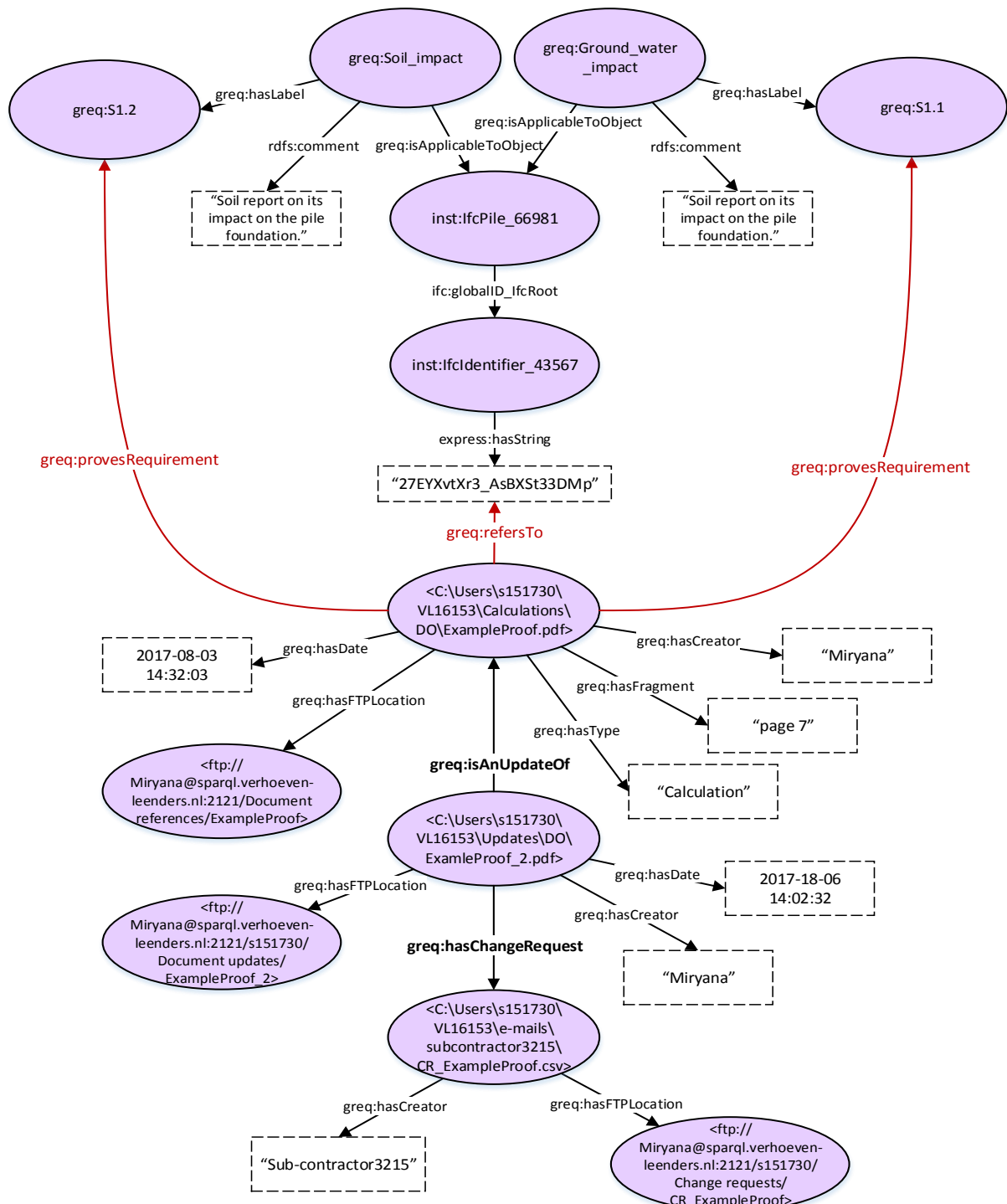
Figure 31: An example, demonstrating the way an object instance is proven on two of the requirements it needs to comply with by a document reference, which has an update based on a change request

Additional properties such as `greq:hasCreator` and `greq:Date` are also defined for all three types of documents, while the initial document proof also has the properties `greq:hasType` and `greq:hasFragment`, indicating the type of document (e.g. calculation, advisory report) and the section of the file (if multiple-page document) which

refers to the exact requirement. The additional ontological definitions added from the tool are presented in *Appendix VIII: Additional ontological definitions* from the tool.

8.5.2 SPARQL queries

As demonstrated in the flowchart diagrams from *Appendix XI: Flowchart: Check requirements and attach a document proof* and *Appendix XII: Flowchart: Attach a document update and a change request*, depicting the different purposes for which the tool can be used, the tool's functionalities rely primarily on the use of SPARQL queries. By storing the data related to a project in an RDF format and thus, having the possibility to retrieve the data necessary by the use of SPARQL queries, the use and the exchange of the data are performed in a more efficient way, supporting project interoperability and information reusability.

For the purpose of showcasing the results which the queries implemented in the tool yield, the SPARQL engine in the triplestore has been used to run some of them and to demonstrate the outcomes. It should be noted that the queries implemented the tool are ran for the particular element selection(s), or for the particular element GUID(s) and therefore, in the queries from *Listing 2* and *Listing 3* below, a GUID string has been inserted.

Firstly, a general query *(Listing 1)* has been attempted with the purpose of illustrating the way requirements data and geometry data, both stored in the triplestore, are being retrieved in a related way. Due to the length of the results, in *Table 8* only an excerpt has been displayed, showcasing the mapping between several beam instances and the 'Crack width' requirement.

Listing 1: SPARQL query retrieving requirement and geometry data

```
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Guid ?instance ?Label ?Requirement ?Type
WHERE {
    ?Requirement greq:isApplicableToObject ?ifctype .
    ?Requirement a ?Type .
    ?instance a ?ifctype .
    ?Requirement greq:hasLabel ?Label .
    ?instance ifc:globalId_IfcRoot ?guid_id .
    ?guid_id express:hasString ?Guid .
}
```

Table 8: Excerpt from the query results from Listing 1

Guid	Instance	Label	Requirement	Type
"okggG2MrnDqfHgrinIKo4k"	inst:IfcBeam_13967	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"okggG2MrnDqfHgrinIKo4m"	inst:IfcBeam_14035	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"okggG2MrnDqfHgrinIKo4o"	inst:IfcBeam_14188	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"okggG2MrnDqfHgrinIKo4q"	inst:IfcBeam_14371	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"okggG2MrnDqfHgrinIKo4s"	inst:IfcBeam_14439	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"okggG2MrnDqfHgrinIKo4u"	inst:IfcBeam_14507	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"okggG2MrnDqfHgrinIKo4w"	inst:IfcBeam_14575	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"1yKGrwOUz7cfs7M84w4sDL"	inst:IfcBeam_18349	greq:A2.1	greq:Crack_width	greq:Usability_requirement
"1yKGrwOUz7cfs7M84w4sDN"	inst:IfcBeam_18417	greq:A2.1	greq:Crack_width	greq:Usability_requirement

Secondly, the following query *(Listing 2)*, the results of which are appended in the "proven requirements" dialogue box of the viewer, displays all requirements on which the selected object instance has been previously proven. A similar query is used for appending the requirements of the object instance which haven't been proven yet in the "open requirements" dialogue box of the viewer. The difference between the two queries consists in using the `NOT  EXISTS'` filter, instead of the `EXISTS'` filter, shown below.

Listing 2: SPARQL query retrieving information on the requirements which have been proven for a selected element

```
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Label ?Requirement ?Type
WHERE {
    ?Requirement greq:isApplicableToObject ?ifctype .
    ?Requirement a ?Type .
    ?instance a ?ifctype .
    ?Requirement greq:hasLabel ?Label .
    ?instance ifc:globalId_IfcRoot ?guid_id .
    ?guid_id express:hasString "2mh5nJJP1Ed8tFMBbBfY4s" .
FILTER(
    EXISTS {
    ?Resource greq:refersTo "2mh5nJJP1Ed8tFMBbBfY4s" .
    ?Resource greq:provesRequirement ?Label .
        }
    )

}
```

Table 9: Query results from Listing 2

Label	Requirement	Type
greq:A2.1	greq:Crack_width	greq:Usability_requirement
greq:S1.1	greq:Ground_water_impact	greq:External_Interaction_requirement
greq:F1.1	greq:Normal_forces	greq:Loadbearing_requirement
greq:O1.2	greq:Stiffness	greq:Material_Quality_requirement

Thirdly, the following query *(Listing 3)* has been used in order to retrieve the document proofs related to an element for the purpose of attaching an updated version of that document and the change request, which has initiated the update. The updated version of the document is linked only to the document proof itself and not to the specific element selected in the viewer. Therefore, regardless of whether the same document has been used to prove the validity of multiple elements, only one of them needs to be selected in order for an update of the original file to be attached.

Listing 3: SPARQL query retrieving the document updates related to a selected model instance

```
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>


SELECT ?Resource
WHERE {
    ?instance ifc:globalId_IfcRoot ?guid_id .
    ?guid_id express:hasString "0AzhLiTWz0d8jdp0Al8hqR" .
    ?Resource greq:refersTo "0AzhLiTWz0d8jdp0Al8hqR" .

}
```

Table 10: Query results from Listing 3

Resource
<C:\Users\s151730\Desktop\VL16153-Calculation_System001-2017-08-20.pdf>
<C:\Users\s151730\Desktop\VL16153-Calculation002-2017-08-21.pdf>

Apart from validating the tool, the following section presents the ways in which the prototype can be used by demonstrating its functionalities the backbone of which consists of SPARQL queries as the ones discussed above.

8.6 Tool validation

This chapter intends to validate the soundness of the requirements ontology and the developed tool in regards to the correctness of information generated and also the visualization of that information. The tool is validated with the help of the case study model of the Mavo Schravenlant XL te Schiedam project introduced in *section 8.1*. The use of a second model for the tool validation is not necessary, as the developed ontology is generic enough to be compatible with different IFC models due to the fact that it is not highly dependent on the IFC schema. The ontology makes use only of building element classes, while the document reference functionality of the tool uses the objects' GUIDs.

Firstly, the requirements ontology *(Appendix VII: General requirements ontology)* is validated through the RDF Validation Service of the W3C Consortium which can be found at *https://www.w3.org/RDF/Validator/*. The validator confirmed the soundness of the RDF file by generating all triples defined in the data model. Secondly, the information created in the tool is verified in several steps. First of all, the IFC model of the MAVO project is loaded in the application and the triplestore repository, containing all additional data, is selected. Previously the model has been converted into ifcOWL and stored, together with the general requirements ontology, in the selected repository in the triplestore.

Afterwards, a building element from the model (a foundation pile) is selected in the viewer. As *Figure 32* shows, there are no proven requirements for this specific pile yet. Therefore, as an example, a document proof confirming the pile's compliance with three of the requirements indicated as "open" in the application *(Figure 33)* is selected and associated with both the pile's GUID and the labels of the three checked requirements. The association has been done by the use of the Python module `rdflib` with the help of which the data from

the pop-up dialogue box is saved in RDF format in the previously specified repository. The selected requirements are '*Nuisance impact*', '*Soil impact*' and '*Groundwater impact*', all proven by a single calculation report named '*VL16153-Calculation-2017-08-20*'.
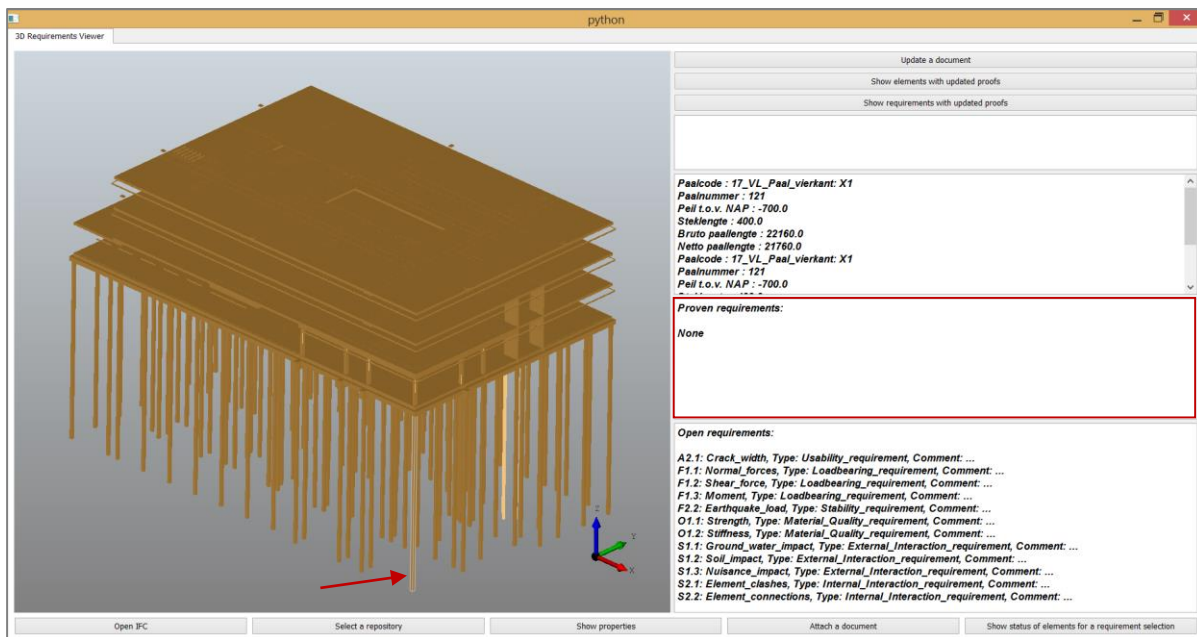

*Figure 32: "Open" and "proven" requirements for an element selection*
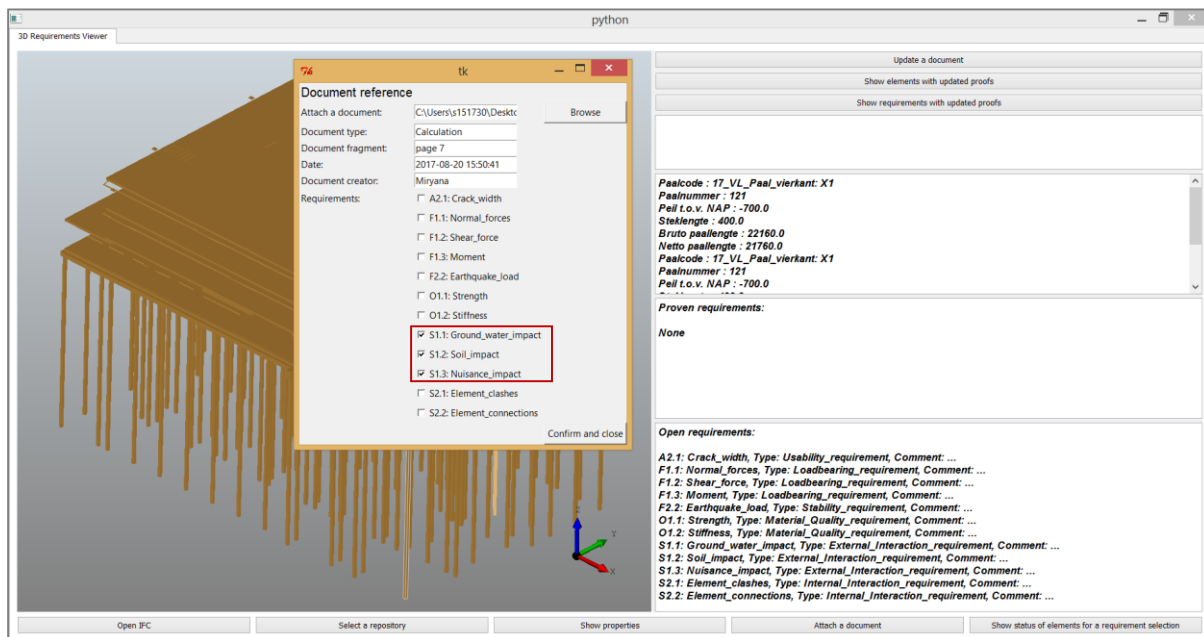
Figure 33: Attaching a document proof to an object instance in relation to three requirements

To confirm that the data has been generated in an RDF format, *Figure 34* shows the new triplestore context which has been created. Furthermore, *Figure 35* validates that the document proof has been uploaded to the FTP server under the 'Document references' folder. Additionally, *Figure 36* confirms that the next time the element is inspected, the viewer displays the three requirements in the "proven" requirement box with the FTP location of the documents that prove the conformity of the pile with the three requirements.

Subject	Predicate	Object
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:hasDocumentReferenceCreator	"Miryana"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:hasDocumentReferenceDate	2017-08-20 15:50:41
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:hasDocumentReferenceFragment	"page 7"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:hasDocumentReferenceType	"Calculation"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:hasFTPLocation	<ftp://Miryana@sparql.verhoeven-leenders.nl:2121/Requirement proofs/Document references/VL16153-Calculation001-2017-08-20.pdf>
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:provesRequirement	greq:S1.1
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:provesRequirement	greq:S1.2
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:provesRequirement	greq:S1.3
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	greq:refersTo	"2mh5nJJP1Ed8tFMBbBfY4s"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>	rdf:type	greq:DocumentReference

Figure 34: The new context in the triplestore containing the information previously generated
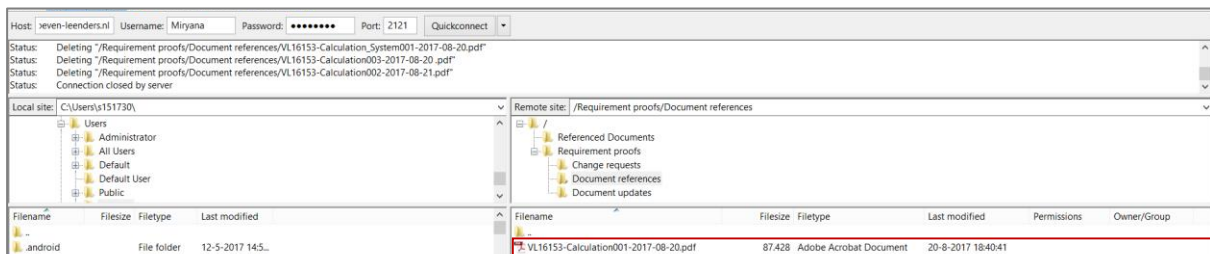

*Figure 35: The document uploaded on the file server under the 'Document references' directory*
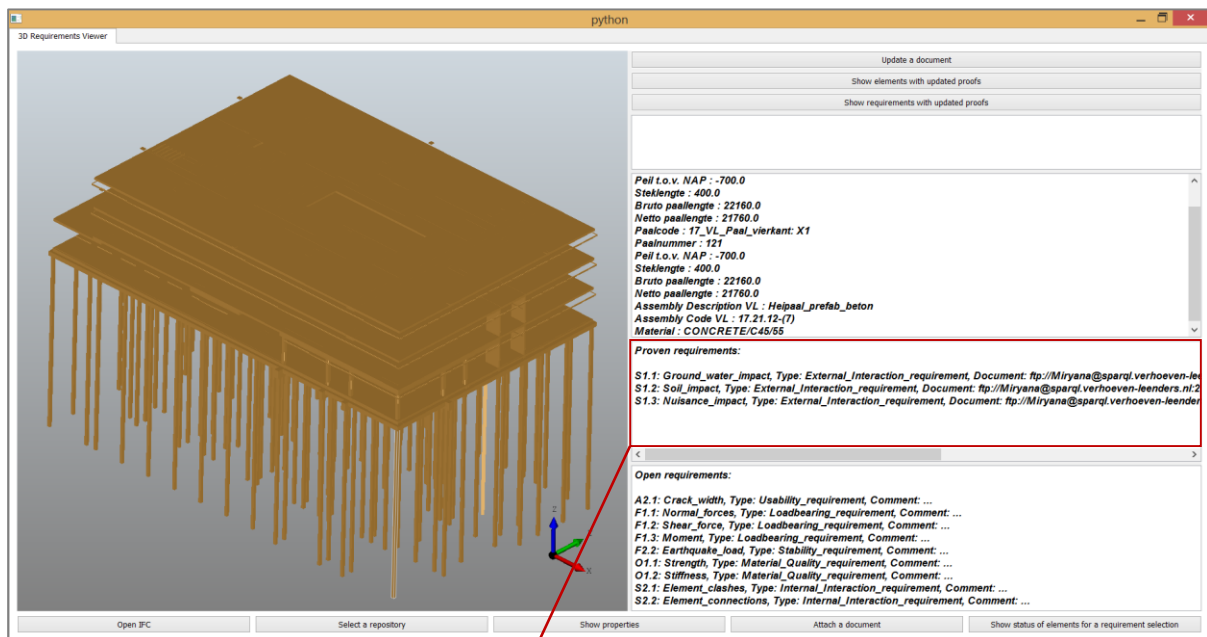
Figure 36: The requirements proven previously, displayed in the 'Proven requirements' section

The same steps are taken in order to also validate the scenario of a multiple-element selection. A system of four columns and one beam are selected simultaneously and a document, proving

the system's compliance with two of the requirements **"open" for all individual elements of the system**, is related to both all the GUIDs and the respective requirement labels *(Figure 37)*. In *Figure 38* it can be seen that the new RDF triples created in the triplestore indicate that the document has proven the conformity of the system to both the '*Fire resistance'* and the '*Nuisance impact'* requirements.
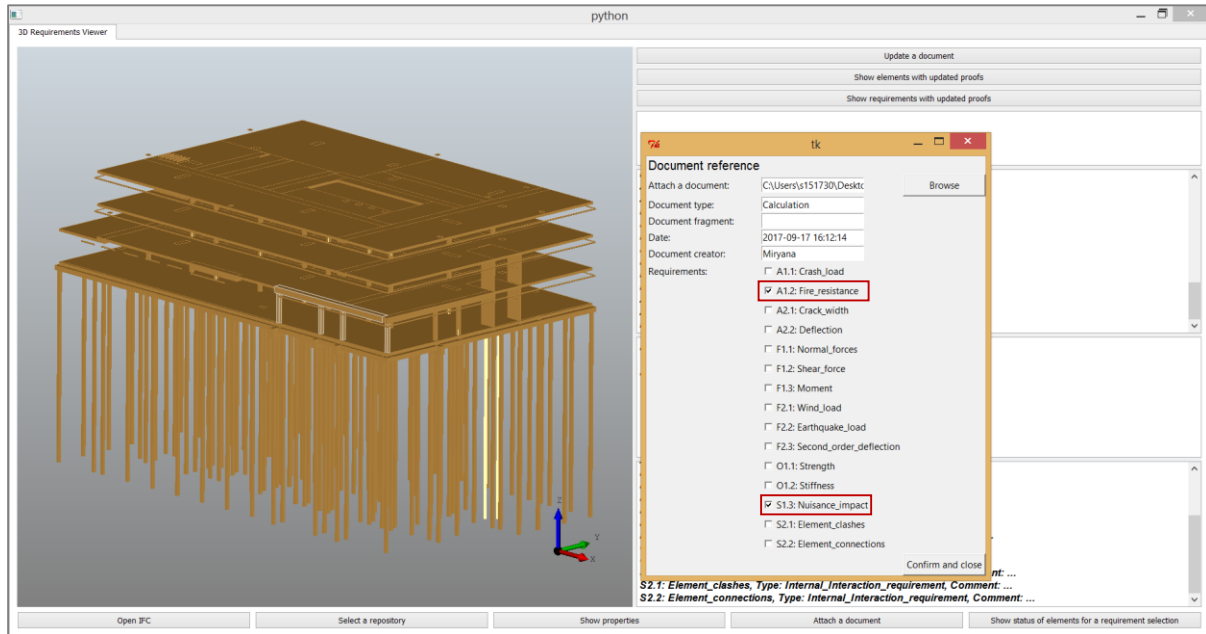

Figure 37: A system of one beam and four columns is proven on two general requirements

Subject	Predicate	Object
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:hasDocumentReferenceCreator	"Miryana"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:hasDocumentReferenceDate	2017-09-17 16:12:14
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:hasDocumentReferenceFragment	""
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:hasDocumentReferenceType	"Calculation"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:hasFTPLocation	<ftp://Miryana@sparql.verhoeven-leenders.nl:2121/Requirement proofs/Document references/VL16153-Calculation_System001-2017-08-20.pdf>
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:provesRequirement	greq:A1.2
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:provesRequirement	greq:S1.3
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:refersTo	"oAzhLiTWzod8jdpoAl8hqL"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:refersTo	"oAzhLiTWzod8jdpoAl8hqN"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:refersTo	"oAzhLiTWzod8jdpoAl8hqR"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:refersTo	"oAzhLiTWzod8jdpoAl8hrf"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	greq:refersTo	"2S$wVdH$X2C8W_VJo$Rb9o"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation_System001-2017-08-20.pdf>	rdf:type	greq:DocumentReference

Figure 38: The newly created triples in the triplestore

Next, the model is queried for the '*Nuisance impact'* requirement and all elements, previously proven on that requirement, (the pile, the four columns and the beam) are displayed in green, while all of the remaining model components, which still need to be proven on that requirement are displayed in red, as shown in ***Error! Reference source not found.***. In addition,

he model has been queried one more time for visualizing the model instances proven on their compliance with the '*Soil impact*' requirement *Figure 40*. The proven elements are displayed in green, the elements to be proven are displayed in red and all of the remaining element, not associated with that requirement remain unchanged.
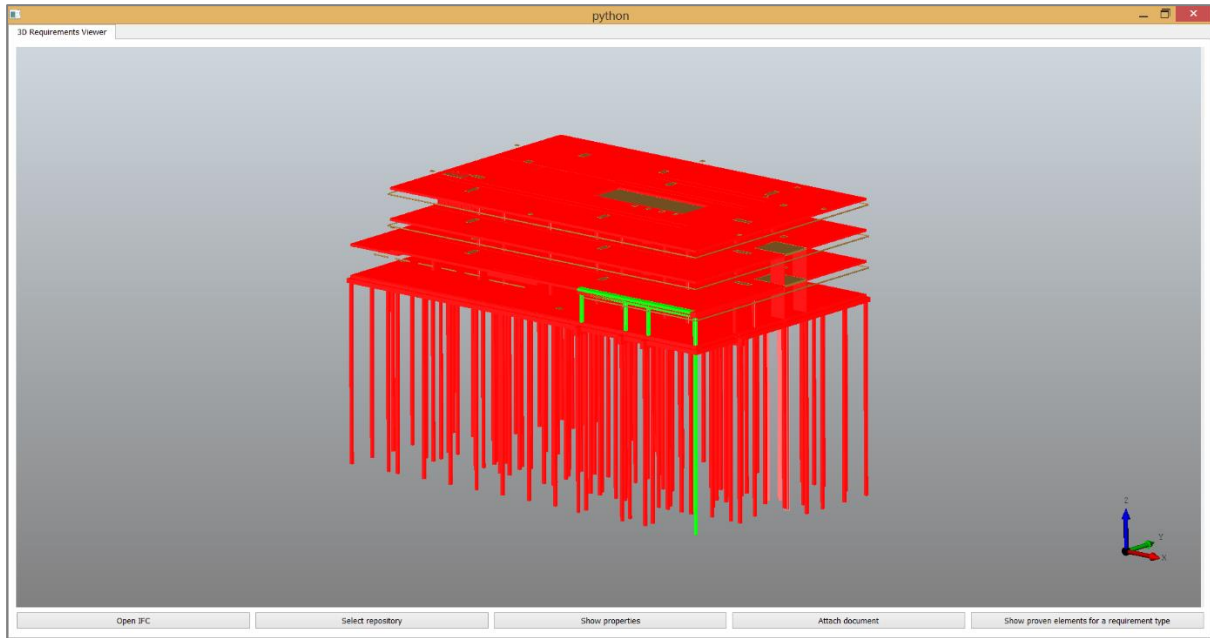
Figure 39: Elements proven (green) and elements to be proven (red) on 'Nuisance impact'
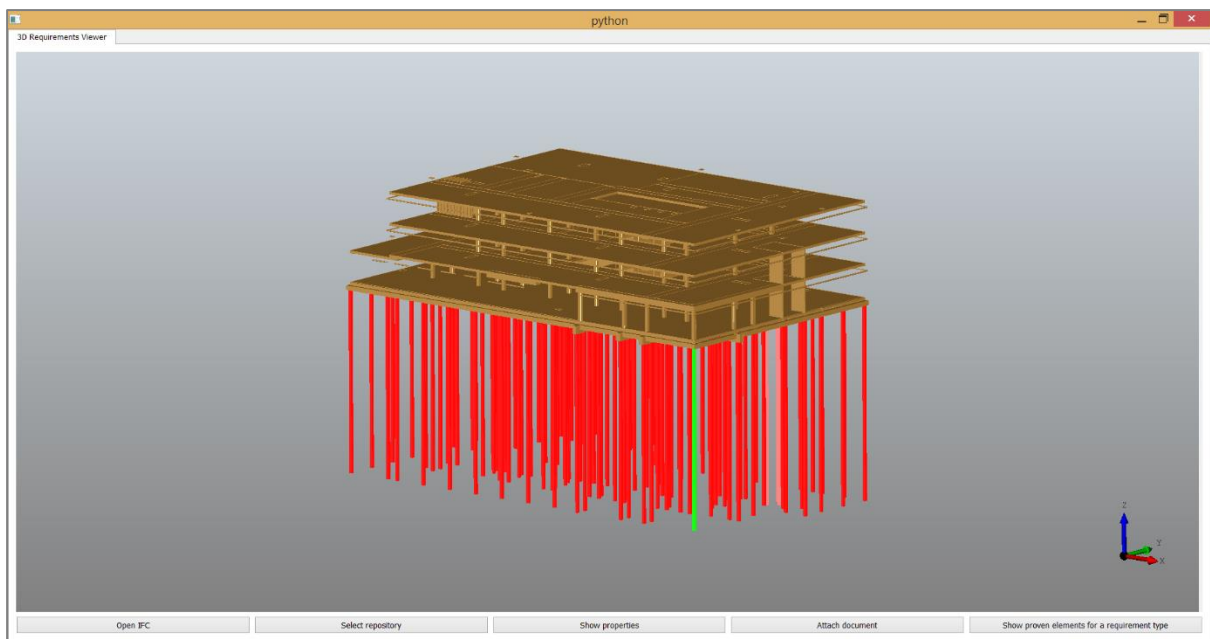
Figure 40: Elements proven (green) and elements to be proven (red) on 'Soil impact'

In the meantime, two more calculation documents have been attached to the pile. The third part of the validation has to do with updating one of the documents referenced previously based on a change request. *Figure 42* verifies that the new triples are stored in the triplestore, while *Figure 43* confirms that the document update and the change request document have been uploaded to the correct folders on the FTP server.
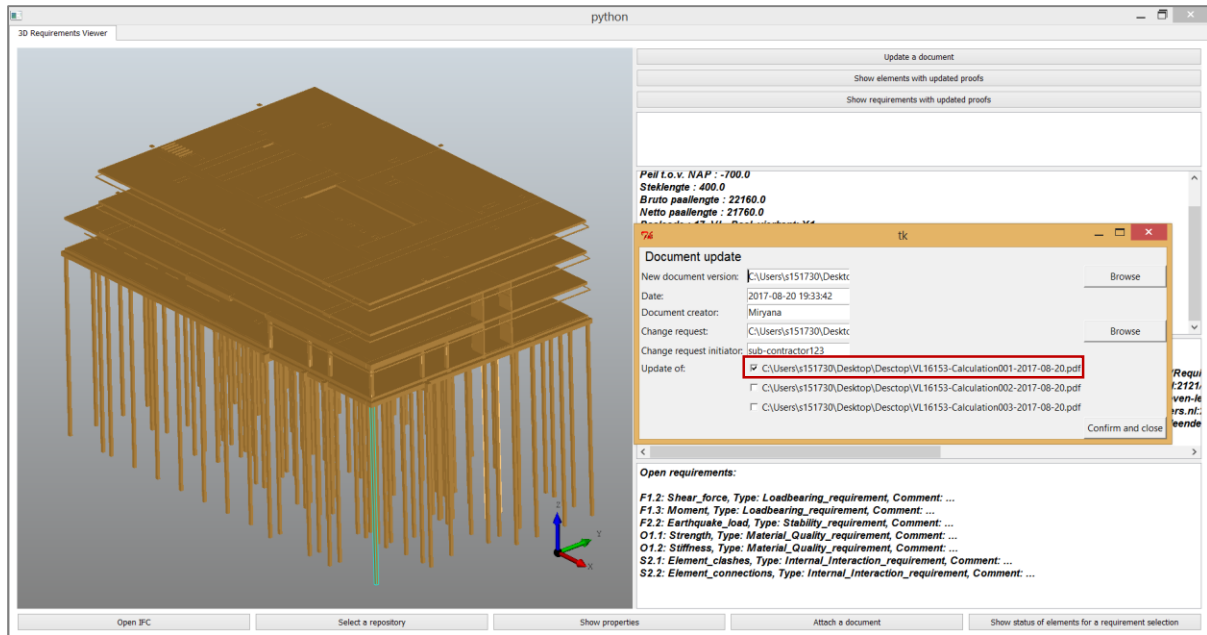
Figure 41: Updating a previously attached document reference based on a change request

Subject	Predicate	Object
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20_Update1.pdf>	greq:hasChangeRequest	<C:\Users\s151730\Desktop\Desctop\VL16153-ChangeRequest-2017-08-23.pdf>
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20_Update1.pdf>	greq:hasDocumentUpdateCreator	"Miryana"
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20_Update1.pdf>	greq:hasDocumentUpdateDate	2017-08-20 19:33:42
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20_Update1.pdf>	greq:hasFTPLocation	<ftp://Miryana@sparql.verhoeven-leenders.nl:2121/Requirement proofs/Document updates/VL16153-Calculation001-2017-08-20_Update1.pdf>
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20_Update1.pdf>	greq:isAnUpdateOf	<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20.pdf>
<C:\Users\s151730\Desktop\Desctop\VL16153-Calculation001-2017-08-20_Update1.pdf>	rdf:type	greq:DocumentUpdate
<C:\Users\s151730\Desktop\Desctop\VL16153-ChangeRequest-2017-08-23.pdf>	greq:hasFTPLocation	<ftp://Miryana@sparql.verhoeven-leenders.nl:2121/Requirement proofs/Change requests/VL16153-ChangeRequest-2017-08-23.pdf>
<C:\Users\s151730\Desktop\Desctop\VL16153-ChangeRequest-2017-08-23.pdf>	greq:hasInitiator	"sub-contractor123"

Figure 42: A new triplestore context, containing the document update and the change request
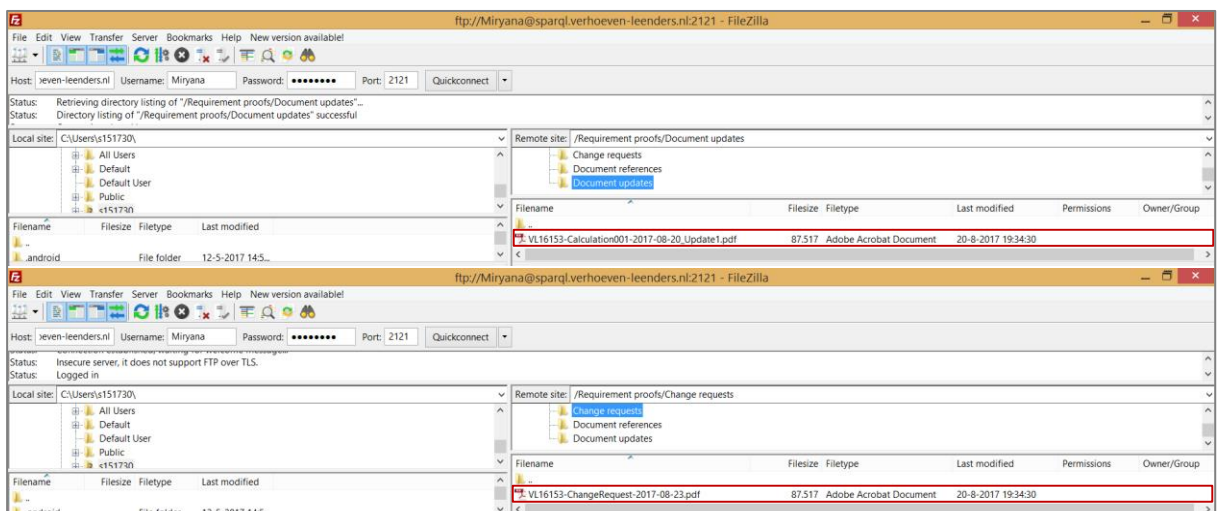
Figure 43: The document update and the change request uploaded in the FTP server

Thereafter, the model is queried for the elements which contain document updates and as seen in *Figure 44*, the pile related to the document update from *Figure 41 i*s displayed in blue. This indicates that the specific element connected to this document has been at issue, in the sense that it has been recalculated even after the final design handover.
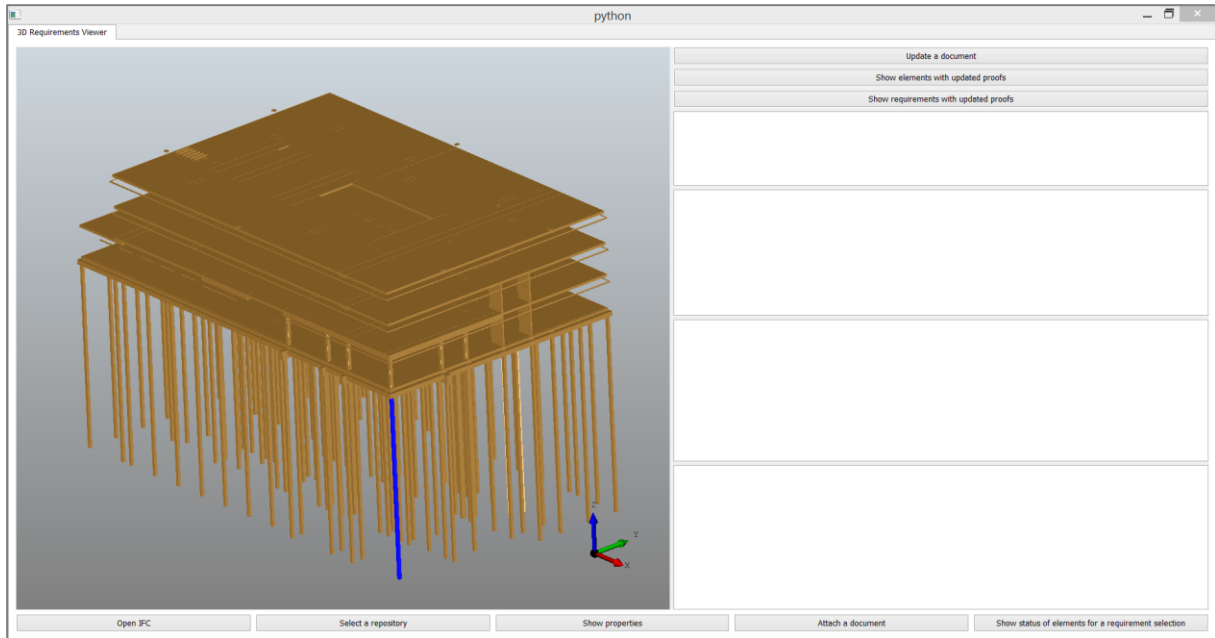
Figure 44: An element with a document update

Lastly, by selecting the element in blue and querying the repository for the requirements on which the pile has been updated, the three requirements to which the original document proof was attached in *Figure 33* are displayed together with the FTP location of the document update attached to the original proof in *Figure 41*.
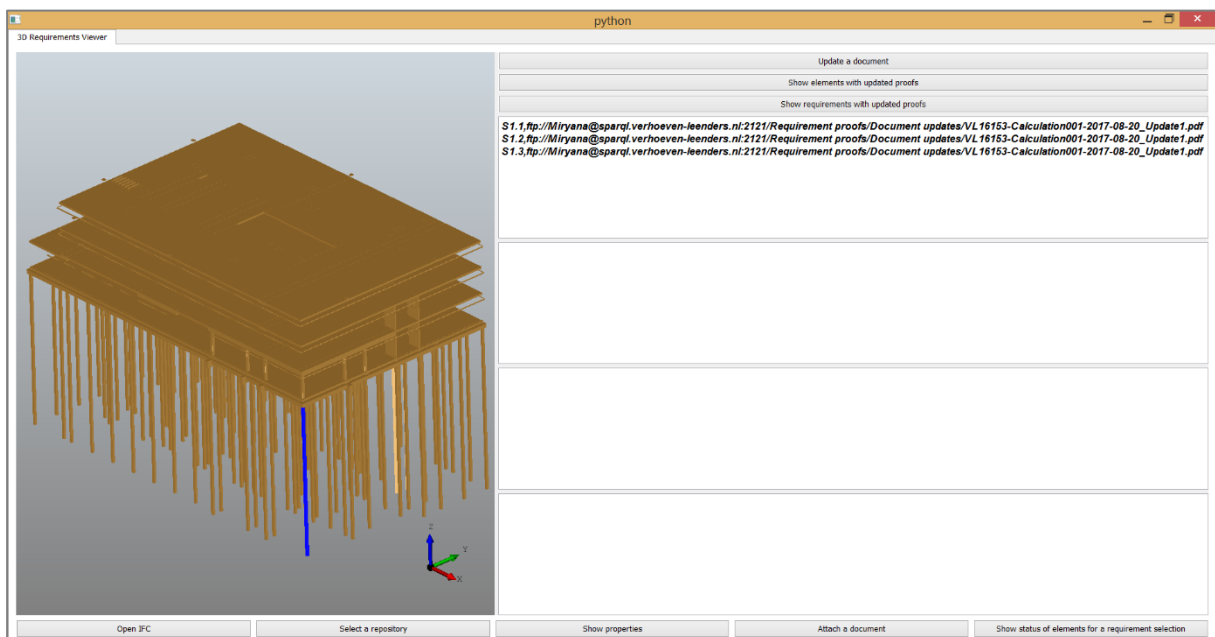
Figure 45: The requirements on which the pile has been updated

By copying the FTP address of the document in a browser:

ftp://sparql.verhoeven-leenders.nl:2121/Requirement%20proofs/Document%20updates/VL16153-Calculation001-2017-08-20_Update1.pdfof

an access to the file can be obtained *(Figure 46)* and the file can be revised *(Figure 47)*.
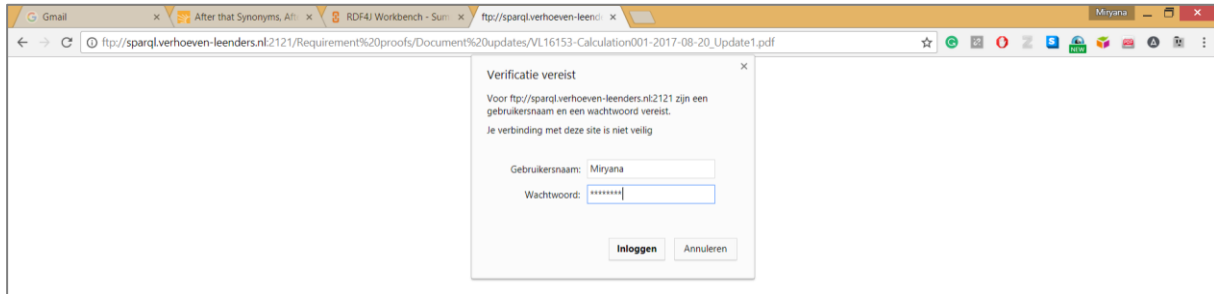
Figure 46: Accessing the FTP server location of the document update from a browser
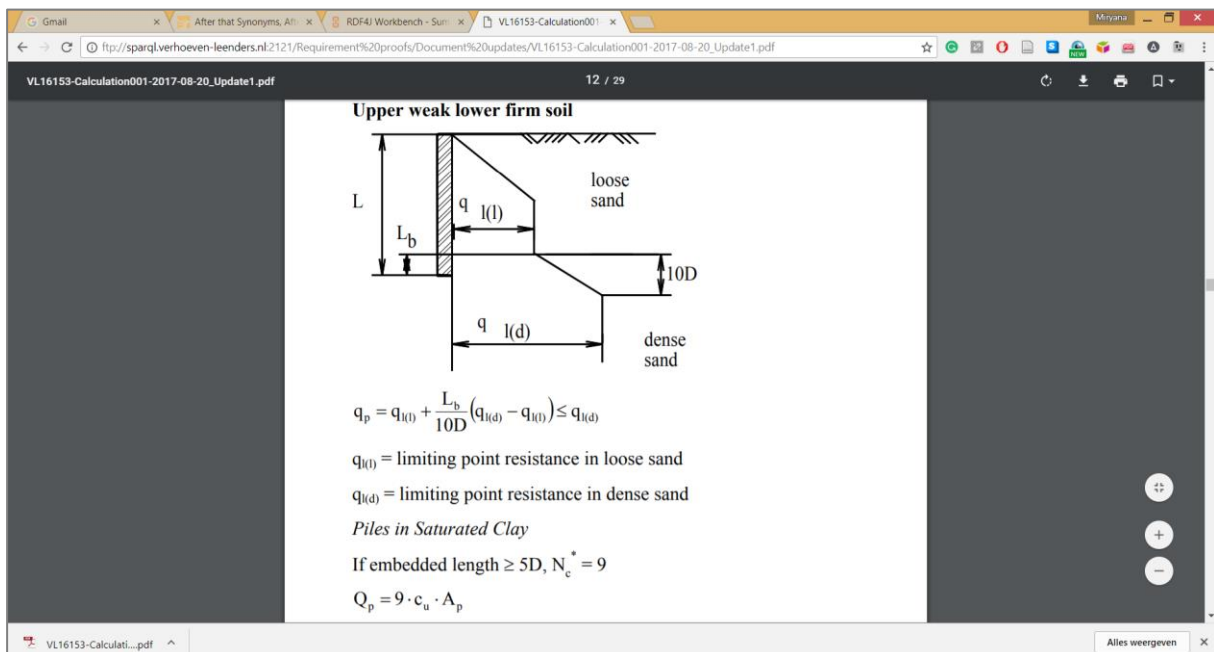
Figure 47: Document update accessed from the browser by its FTP location

8.7   Conclusion

As mentioned previously, the main use of the tool developed during the course of this thesis project is to create a connection between the project's documentation and the BIM model for the better traceability of the relations between requirements and structure and for monitoring the process of requirement validation. The tool's main contribution has to do with the fact that it showcases a way in which the company can improve the efficiency of information management within its own establishment but also in the process of information handover to external parties. It is also the main mechanism for the validation of the conducted research. The tool presented is, however, considerably immature in its development and functionalities and therefore, in the recommendations chapter of this thesis, some of the points for improvement will be introduced.

PART E: CONCLUSION AND DISCUSSION

In chapter 9 of this section, the main research question and the subquestions of the thesis are answered. Chapter 10 provides recommendations based on the knowledge gained from conducting the research by also firstly discussing the project's limitations. Thereafter, the chapter gives an insight into the possible directions for future research on the topic of requirements management for the design processes in the building structures engineering domain.
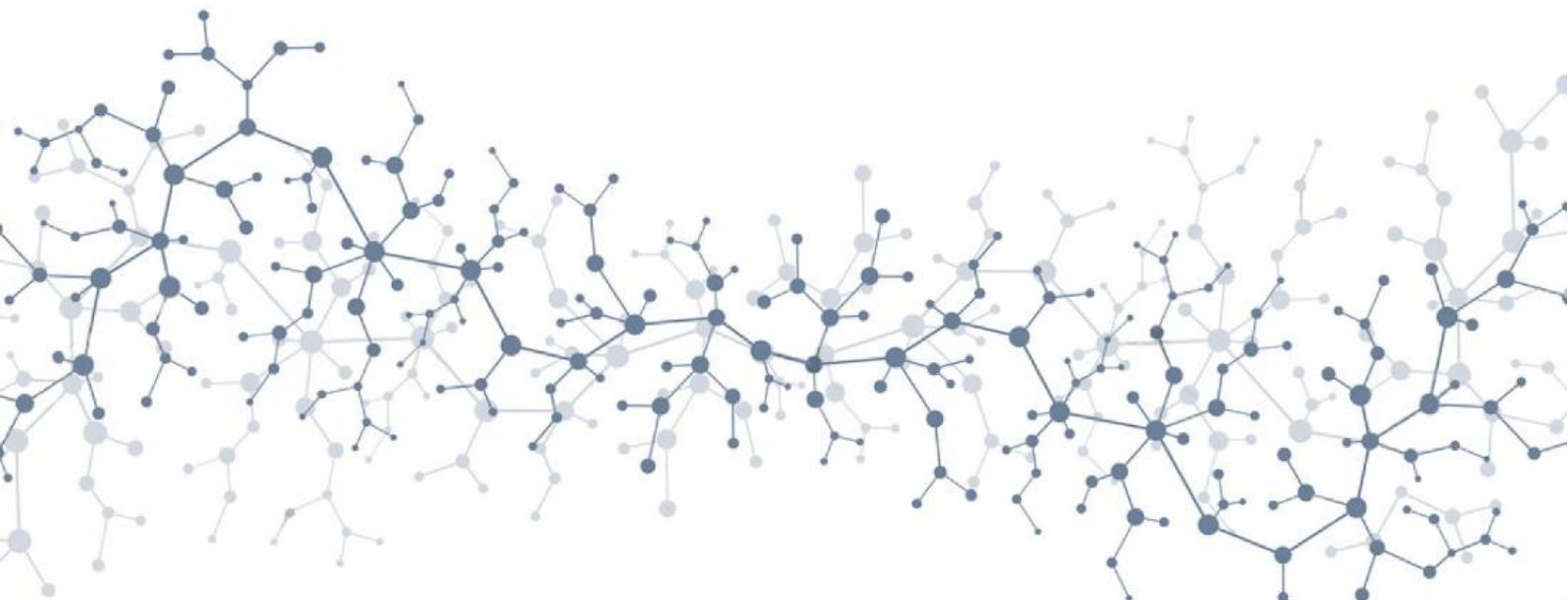
9 Conclusion

This chapter discusses the main objectives of the research by providing an answer to the research questions. The questions aim at investigating how the scattered and inconsistent nature of the requirements management practices in the engineering of building structures can be improved. The focal point of the expected results consists in achieving a certain level of integration between design, requirements, and documentation. Therefore, the main research question is formulated as follows: 'How can a mapping between project requirements, design, and documentation be created for the validation of the design's conformity with the requirements and for improving the traceability of information?' Prior to discussing the main research objective, however, the answers to the sub-questions will be given.

In order to improve the traceability of information and validate the design's conformity with its corresponding requirements, several aspects need to be considered and decided upon as a preparatory step. These aspects include the definition of general engineering requirements fundamental for the engineering design handover to external parties and the level of detail at which these requirements should be linked to the 3D geometry. Furthermore, these aspects also involve the means by which requirement conformity can be proven, the technological map for achieving the main research objective and the benefits the selected technological solution can bring to the data handling practices of the AEC industry.

The answers to the supporting research questions are presented next:

Which engineering requirements are fundamental in a building structures project (general requirements vs. project-specific requirements) and constitute the core of the final engineering design handover to external parties?

Based on the research conducted, in the building structures domain, the engineering requirements can be divided into several main categories. Depending on whether they relate to the function that an element or a system of elements need to realize, to supporting functions or aspects of these elements, to their properties, or whether they are requirements which come as a result of relations between the structure and the structure's environment.

General engineering requirements, in contrast to project-specific requirements, are the requirements on elements or systems which can be applied to any building structures project regardless of its size, complexity, social function or surroundings. These are also the requirements supported by the Dutch building law and the Eurocodes and therefore, they are essential for the final engineering design handover to external parties such as sub-contractors, manufacturers and audit commissions.

Based on the general engineering requirements a building structure must be validated on its overall stability and the ability to bear vertical and horizontal loads. In addition to that, the quality of the materials, contributing to the building element's strength and stiffness must be validated. Aspects such as prevention and usability, incorporating fire resistance of the primary structure, the crash load from an eventual impact with an object such as a vehicle, the deflection angle of displacement and the crack width in load bearing elements, also constitute an important part of the information necessary to be justified by the engineers.

Furthermore, the surrounding aspects of a building structure are also to be considered. They include groundwater impact on foundation elements, which results in additional loads and erosion, as well as loads coming from the soil. External system interactions such as the degree of the nuisance caused to the structure's environment are also to be taken into account by the engineering design team and justified by calculations and reports.

Internal system interactions consisting of possible clashes between physical objects in the design are likewise to be considered for the purpose of removing them. Information regarding the required types of connections between the different structural components must be also provided by the structural engineering team.

The formalized requirements matrix from *section 7.5* represents the 'must haves' and 'should haves' in terms of the information which needs to be completed and proven before the engineering design handover to other project stakeholders. Respectively, this is the information which the audit authorities also need for validating the stability of the building and for approving the design's compliance with the building law.

What are the levels of detail/development and on which LOD is it most beneficial to connect engineering requirements data with the BIM model (e.g. building storeys, spaces, elements, systems of elements)?

This research question will be answered both from the perspective of the common engineering design practices related to the levels of development (LOD) and from the perspective of the interviewees, who took part in the qualitative research of the thesis.

The creation of the Dutch Levels of Development is an attempt to formalize the level of information necessary in order for the design and engineering processes to operate effectively. The LOD initiative is an attempt to create a framework indicating what kind of information should be available in a model on a particular design elaboration level so that all project stakeholders are able to do their job adequately.

The project delivery phases for building structures projects during which the structural engineers are usually actively involved in the design process are the preliminary design (LOD200), the final design (LOD300) and the design execution (LOD350) phase. While the LOD200 deals with setting the starting points for the structural design based on the architectural drawings and geometrical model, the LOD300 dives deep into the detailed calculation of the engineering structure and its components. The LOD350 presents a continuation of LOD300 and the structural design is prepared for further development based on the input of third parties.

While the architectural domain, for example, works primarily with spaces, the structural engineering domain works both with individual building elements and with systems or groups of elements. These differences can be attributed to the project requirements which the two professional domains need to consider in their work. While the architect focuses on aspects such as function, usability, aesthetics and user comfort, the structural engineer concentrates on the primary, secondary and temporary structures and their stability. As structural components often belong to several spaces at the same time, using spaces seems to be an ineffective way to relate structural requirements to the design.

Due to the fact that during LOD200 the engineers regard the structure of a building as a system of storeys, the only requirements calculated at that stage are the abilities of these building storeys to carry load. From LOD300 onwards, however, all general engineering requirements are taken into consideration and the engineer performs the calculations on both element and systems level. Therefore, the LOD300 can be considered as the most suitable project phase, at which engineering requirements and structural design can be linked to each other as the same exact principle is used in all following project phases. The decision on whether to relate the requirements to individual element types or on a systems level, however, presents another aspect for consideration.

The decision to relate project requirements to component types, rather than to systems comes as a result of the in-house investigation of the thesis and the realization that systems in the engineering design context are not explicitly specified as building entities. Furthermore, their exact composition comes as a result of the personal preferences of each engineer. The notion of systems can be discovered only in the calculation reports of the specific project and it doesn't exist as a concept in the geometry model or in the consideration of requirements. A structural element can be also quite often assigned to multiple systems at once which brings along the same exact issue occurring when working with spaces as main entities for the structural engineering design. Due to the fact that there is no framework for identifying how the engineer decides to consider systems the possible variations for the calculation of a building structure are countless. For that reason, the systems approach to linking requirements with geometry requires a research on its own before being a reliable way of managing engineering requirements.

Upon asking the question "On which level of detail would it be most beneficial to connect (engineering) requirements and design?", the quantitative research yielded a similar conclusion as the aforementioned. The structural engineering and the SE professionals confirmed the usefulness of having traceability of requirement information on a building element level, rather than on a more general scale such as spaces or building storeys. The BIM experts indicated that relating engineering requirements to components or systems is a logical starting point, however, for the purpose of complete interoperability between domains, the architectural and the engineering view on the design should be related in the future. This nonetheless, also depends on changing the working styles of both parties and integrating their design processes.

How can structural engineering requirements be mapped to all object instances from the geometry model to which they relate to?

The formalized requirements matrix from *section 7.5* summarizes the outcomes of the research phase of this thesis by combining the results related to the previous two research questions. These results are the classification of general engineering requirements and the entities to which they are related to. Therefore, the matrix itself represents a knowledge structure which formally describes a specific domain knowledge and namely, the general structural engineering requirements related to the final design phase (LOD300) of building structures projects. The requirements, as well as the building elements, represent concepts which, as indicated by the dots in the matrix *(Figure 21)*, have relations between each other.

Being the key to the Semantic Web, ontologies can be described as knowledge structures used to formally describe domain knowledge through the creation of a framework of relevant concepts and the semantic connections between them (Abanda et al., 2013). Therefore, the

created requirements matrix creates the prerequisite for expressing the information it contains and namely, interrelated concepts, by the means of an ontology.

The IFC schema, which describes the object instances of a design, has its own OWL representation (ifcOWL), making IFC data available in RDF format. Therefore, due to the concept of reusability of existing knowledge resources as an input for building new domain ontologies, the structural components from the requirements matrix can be described by the means of the ifcOWL building element classes. Furthermore, an ontology, following the classification structure of the requirements tree from the matrix can be created. These newly defined concepts are then linked through object properties with all element classes from the ifcOWL ontology to which they relate to by the means of one-to-many relationships. Therefore, by converting the IFC geometry model into ifcOWL and by SPARQL querying both the ifcOWL geometry and the requirements ontology, all model instances, related to the respective IFC element classes are being associated with the requirements with which their object class needs to comply. A prerequisite for that is storing the two Named Graphs, namely the geometry model converted into ifcOWL and the requirements ontology, in a triplestore repository from where they can be also SPARQL queried.

How can the conformity of a particular model instance with its associated requirements be proven and the information reused also in later project delivery phases?

Generally, the conformity of engineering components with specific requirements on LOD300 is proven depending on the requirement in question, either by the calculation documents of the engineer or by the expert advisory reports. These documents contain very detailed object-oriented information which is often difficult to be systematized in a uniform way for all projects. Therefore, the most logical way of making the first step in proving the conformity of an element instance with a particular requirement is to connect that instance with the document (e.g. calculation report) which describes it in relation to the requirement.

For achieving that, additional ontological concepts need to be defined and added to the already developed requirements ontology. These concepts describe the document reference (e.g. calculation or advisory report), its eventual update and the change request that issued the update. The document describing a specific object instance from the model can be related to it by also additionally specifying on which requirement(s) this document proves the instance. This way of systematizing information in an RDF format enables the engineering design documentation to be related to the design in an unambiguous way. The project parties, such as the sub-contractors, who need to work with both the calculations and the model can, therefore, trace documentation in an efficient and infallible manner while also keeping track of the required performance of the objects at the same time.

What is the added value of semantically linking requirements and design and can this adjustment lead to a higher efficiency in the design process due to the mitigation of risks associated with design non-conformity with requirements?

This research question will be answered both from the perspective of the benefits associated with describing domain knowledge by the use of semantics and from the perspective of the interviewees who took part in the qualitative research of this thesis.

As ontologies' intended use is to capture domain knowledge in a generic way with the aim of providing a mutual understanding of that domain, ontologies also present the opportunity to

share and reuse that knowledge across applications and groups (Pinto and Martins, 2001). Several factors contribute to the added value of semantically describing a particular set of domain knowledge – the extensibility and reusability of data, the ability to retrieve search-specific sets of information and the software-independent nature of data handling. The aforementioned factors have the potential to improve the efficiency of communication between the stakeholders within a project by facilitating the vendor-neutral exchange, storage, and retrieval of information, which BIM and IFC haven't been able to tackle completely *(section 5.4).*

By the use of ontologies, the interoperability aspect, which aims at creating machine-readable data, can be achieved. Currently, the architectural model, the structural model, and the product models of the manufacturers describe the same content in different ways. A column in the architectural model is not recognized by the computer as being the same exact column in the structural model, and it also doesn't relate to the product specifications of the column type which the concrete manufacturer produces. Therefore, by the use of ontology libraries, also called object type libraries, the issues with the different geometric representation of the same object could be tackled.

The Semantic Web's AAA slogan: "Anybody can say Anything about Any topic." conveys the essential advantage of describing data through semantic technologies. This advantage consists in the opportunity to link any concept of any knowledge domain by the means of one-to-one or one-to-many relationships to another concept, or a number of concepts from a different knowledge domain. This is where the reusability aspect of ontologies comes into place and while the interoperability aspect deals with the different depiction of the same content, the linking across domains tackles the issue of combining different contents such as geometry, sensor data, cost data, manufacturer data etc.

In addition to the aforementioned, a semantic dataset can be easily extended by the addition of more concepts and relations while at the same time preventing repetition and redundancy of definitions. As the building industry is often described as data-intensive, the management and the retrieval of information become demanding and error-prone tasks. By storing data in an RDF format, however, the partial retrieval of only search-relevant information is enabled through the use of SPARQL queries. Besides that, semantically defined data carries the possibility of using reasoning engines for inferring information.

During the expert interviews from the quantitative research, the following question was asked: "Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?". As this question doesn't address the technology by the means of which the data is linked, it aims at providing an answer in regards to the usefulness of connecting the data itself.

Seven out of the nine interviewees expressed the opinion that an active link between project documentation (requirement- or non-requirement related) would bring significant benefits to their processes. Especially for the purpose of having all project information at one place and thus, being able to oversee project progress and alterations, and also in the case of old project revisions. Some of the professionals insisted that there should also be a defined limit to the amount of documentation linked to the geometry model in order to prevent information overload.

After answering all sub-questions, the main research question of the project can be discussed:

How can a mapping between project requirements, design and documentation be created for the validation of the design's conformity with the requirements and for improving the traceability of information?

Once both engineering requirements and object instances from the IFC model are described semantically through the conversion of the geometry to ifcOWL and through the creation of a requirements ontology, both Named Graphs can be stored in a triplestore repository. As storing data in an RDF format enables the selective retrieval of information through SPARQL queries, these queries can be implemented in the back end of a simple 3D viewer, where the IFC model can be visualized and different geometrical components selected.

Upon the selection of an element or multiple elements in the viewer, a query matches the GUID strings of both the IFC model loaded in the viewer application and the ifcOWL version of the model, stored in the triplestore. Due to the link between IFC object classes and requirements in the requirements ontology, the GUID(s) in question can be related to the requirements which they need to comply with.

In order to display the proven and open requirements for an element selection, the previously described query is, therefore, extended in two different ways. For the open requirements, the query selects those requirements for the chosen element(s), which haven't been proven by a document reference (e.g. calculation report). For the proven requirements, the query selects the ones which contain a document reference.

In order to prove a selected element on a specific requirement or multiple requirements, a document reference is related to the GUID string of that element, as well as to the requirement(s)' label. Additional information such as creator, date, time, and document type is also included. All data entries are subsequently saved as a new Named Graph in the project's repository on the triplestore.

In order to prove the compliance of multiple objects with one or more requirements, for preventing confusion or duplication of document proofs, these requirements need to be open for all individual elements in the selection. The document attachment would relate to both all the GUIDs of the elements in question and to the respective requirement labels.

One single document proof can be attached to multiple elements at multiple times while preserving the same URI due to the fact that for each attachment, a new repository context is created, which contains creation date and time in its title and therefore, prevents the overriding of data.

The possibility to update previously attached documents and at the same time to also relate the change requests which have initiated these updates enables the traceability of changes which have occurred after the engineering design handover. This provides a historical overview to all project parties of what has been recalculated and the reasons for it.

The direct link between documentation and design allows the project stakeholders who use the engineering calculations from the final engineering design as their main input, namely the sub-contractors and the manufacturers, to unambiguously find the necessary data. In addition, the engineers also have a practical solution for the design handover to audit authorities and to the client.

10 Limitations and recommendations

This chapter presents both the limitations of the research and the recommendations derived from them. The recommendations are presented in twofold – the ones related to the company and the ones related to further research. The company recommendations are closely related to the company's practices and workflow witnessed during the in-house research on processes and documentation, reported in *chapter 7*. The industry-related recommendations are, nonetheless, more general and from a broader opinion on the topic of requirements management during the engineering design for building structures.

10.1 Limitations of the research

The limitations of this thesis project are divided into two categories. The first category discusses the limitations of the initial research focused on the formalization of general engineering requirements for the final design phase of building structures projects. The second category explains the limitations of the ontology and the developed tool.

The first limitation of the theoretical research relates to the fact that the word "requirement" represents the information necessary to be generated and proven at the end of the final design phase, rather than an actual statement with numerical value and capacity. Furthermore, although related to the Dutch building law and the Eurocodes, the requirement information formalized in the requirements matrix has been gathered primarily from project documentation and interviews conducted within the company. Therefore, the requirements ontology created is compatible with the company's processes but probably would require some alterations if implemented by another organization. In addition to that, as the research focuses on the general, material-neutral engineering information, for this project material-related requirements, such as the environmental class of concrete *(Dutch: Milieuklasse)*, are not considered. Moreover, some requirements such as "nuisance impact" can incorporate a broad spectrum of aspects some of which could be simply linked on a project level, rather than on a component level.

It is also important to mention that a complete formalization of requirements data is unattainable due to the fact that there always will be projects with exotic materials, shapes or even functions, and for that reason, the handling of such requirements must be arranged for the specific project.

In addition to the aforementioned, the theory of Systems Engineering has been implemented on a rather superficial level. Through the SBS and RBS, used for the creation of the requirements matrix, and through the primary categorization of general requirements, the project touches upon some of the SE concepts. However, when taking into account the requirements ontology, it can be argued that due to the linking of requirements solely on a component level, the SE's 'holistic' approach to structures, where the main focus is on the whole rather than on its components, has been contradicted. While the requirements ontology incorporates an RBS tree, the same cannot be said for the SBS as requirements are related only to physical components without a regard to the building as an entity. Furthermore, SE-based management systems usually relate requirements data not only to physical components, as in this research, but also to functional components, to validation and verification activities and to risks, to name a few.

In what concerns relating document proofs to the GUID of an element, this approach can be only applied to a final design model, which is unlikely to undergo component replacements. Using the prototype during the actual design exploration phase (LOD200) for the purpose of tracking changes can be problematic due to the fact that if an element is replaced by another, the IFC model exported will logically generate a different GUID for that element. Therefore, the information of the replacement will not be recorded and at the same time, the data generated for the previous component should be also somehow tracked and removed. Such scenario would be not so detrimental in a final design phase such as LOD300 where element replacements are more unlikely to occur and if so, would involve a limited amount of components.

When it comes to the limitations of the developed prototype, they come in threefold. Firstly, the functionalities of the tool are at this moment limited only to the proof of requirement information based on document attachment. If actual numerical values of requirements are to be proven in the future, significant changes to the back end of the prototype would be necessary. Secondly, the tool interface does not provide the user with the ability to make corrections or erase data which has been falsely referenced. Thirdly, currently, there is no functionality in the tool allowing the user to pack and export data for the purpose of transferring it to other parties.

An obvious challenge when using large BIM models in relation to an extensive database of RDF information is that the performance of the tool could significantly slow down due to its dependence on queries. Another inconvenience which the technical implementation creates is the fact that the IFC model needs to be converted into ifcOWL and stored in the triplestore repository prior to the use of the prototype for mapping geometry, requirements, and documentation. Instead, having the option to do the IFC conversion into RDF as well as to store the created data in the triplestore from the prototype interface itself would be a more convenient solution for the user.

10.2 Recommendations

From the requirements research phase of the thesis project, it was witnessed that *Verhoeven en Leenders* uses several manuals for the formalization of various practices. The manuals on Systems Engineering and Quality (of products and processes), as well as the workflow specifications on the implementation of BIM, are serving as a good guideline for the common understanding of how information and projects, in general, should be handled.

Therefore, for the implementation of semantic technologies with the purpose of requirements management within any organization, it is advisable to first look into several aspects such as responsibilities, actors, rights and legal implications. While the first two aspects relate to the organization's internal processes, the latter ones affect the organization's information exchange with external parties and the ownership of the exchanged information.

Additionally, an IFC modeling protocol for the uniform and standardized way of creating structural engineering designs should be established due to the differences in modeling preferences amongst the different stakeholders. While at this basic stage of development, the requirements ontology created relates only to the object classes from the IFC schema, in later stages, when the information becomes more refined and the ontology structure – more complex, the need for creating a uniform and consistent modeling guidelines will become necessary. An issue encountered at this moment within the company relates to the fact that

IfcMember, which is the object class corresponding to wind bracing elements, is not used properly. Instead, wind bracing is currently being modeled with the object class *IfcBeam*.

In relation to the formalized requirements ontology, the data incorporated in it should undergo a stepwise improvement and refinement over time. The most important objective to be achieved is to replace the use of documents for proving the compliance of design with requirements by comparing actual numerical values based on sets of minimum and maximum constraints. In this way, proper requirement validation and rule checking can be enabled also throughout the design process itself. As expected, not all requirements can be defined in numerical terms and therefore, the related documentation which proves their validity could still be implemented. It, however, should not be the main approach for proving design conformity with requirements because it still entails a high probability of making mistakes.

The suggested objective is, nonetheless, obtainable only in a step-by-step manner related to the processing of significant amounts of data. The most suitable follow-up for extending the ontology would be to look into materials and material-related object requirements. This implies the classification of concrete-, steel-, timber- and brick-specific information and the derivation of particular conditions which each material has to meet also depending on the type of building element it is used for.

Subsequently, a possible continuation would be the selection of requirement data which needs to be specified in the form of written statements. Afterwards, these statements have to be related to capacities (minimum and maximum constraints) in a way, in which object-based reasoning can be also enabled.

In relation to the prototype, it would be advisable to investigate the possibility of extending the functionality of the tool by adding the option of exporting semantic data as an ICDD (Information Container for Data Drop). As mentioned earlier, this thesis is complementary to another graduation project (Bernal, 2017), which focuses on requirements and design validation through BCF. Therefore, after the integration of the two prototypes, the tool would incorporate functionalities related to object-oriented data, project documentation and verification BCFs, all three of which are integral parts of the ICDD standard.

Apart from describing building data in a semantic way, managing that data by encoding information into the titles of the Named Graphs can be developed as a useful feature which provides an additional layer of definition allowing for a more advanced management and retrieval of data. As the name of the graph can act as a gatekeeper to the underlying data, some of the limitations of the research can be addressed. Finding and erasing falsely mapped data sources, for example, can be facilitated by this approach as every entry in the viewer interface creates a new Named Graph. By having the option to query just for the graph itself, rather than for the actual information it carries, this approach assures that all wrongly associated triples can be erased at once.

The management of requirements in the context of this research was considered from the limited perspective of the design processes in the building structures domain for the design handover to external project stakeholders. Therefore, project requirements were regarded in isolation from their predeceasing specifications in the form of soft client requirements from the elicitation phase and in the form of architectural/functional requirements from the design exploration phase. As the structural engineering requirements are a derivative from the aforementioned two, for the purposes of interoperability between all project stakeholders the

bigger picture of requirements management practices needs to be considered when implementing it in an actual project so that all parties can have the overview of how their domain-specific requirements influence the requirements of the other project parties.

For example, the client's demand for a sports facility in the building translates into functional, spatial and aesthetical requirements for the architect according to which the space gets assigned to a specific room category. The room category further indicates the loads which the underlying structure needs to carry and these loads translate into forces within the different structural engineering components (e.g. columns and beams) that bear the sports facility. The loadbearing capabilities of the supporting elements then translate into structural connections between the engineering components, as well as indicate the quality of concrete and reinforcement which the manufacturers need to consider.

Therefore, the relations between all these requirements need to be traced throughout the design process of a building structure because they are not created in isolation within each domain but are the direct consequences of the design choices which the previous party has taken. Once engineering requirements are represented in the context of the conditions they were derived from and the decisions which they influenced, their evolution throughout the design processes within a building structures project can be traced which, therefore, assists interoperability.

10.3 Future research

Due to the limitations of time, research scope and the initial research stage for the formalization of general project requirements on one hand, and the broadness of the chosen topic on the other, many aspects of its essence couldn't be considered in this research. Therefore, based on the acquired knowledge, this section discusses some of the research which can be carried out in the future.

As the conducted research serves the purpose of facilitating the final design handover to external parties such as sub-contractors and manufacturers, and especially in the case of material- related extension to the ontology, a topic for further investigation could be the possibility to associate the engineering components with object type libraries from the manufacturers. By enabling the data compatibility with external vocabularies and datasets, suggesting the reusability of ontological concepts, a more interoperable and integrated way of data and requirement handling can be achieved.

As mentioned in the limitations of the research, Systems Engineering concepts weren't implemented in the development of the thesis but rather used as a marginal topic for understanding how requirements can be systematically handled and organized in civil projects. The knowledge gained afterwards was used for the sole purpose of identifying the layers of inheritance within the object trees, rather than with the intent to implement SE in the project.

As the thesis focuses on building structures and also based on the previous discussions, related to the differences between the two types of engineering domains, it becomes clear that the use of SE-based management systems such as Relatics is already implemented in civil projects but not in building structures projects due to the lack of client policies enforcing it. Some of the complementary reasons for that relate to the leading role of the architect and the fact that architectural design deals primarily with spaces rather than building components and the fact that building structures projects are still primarily bounded by traditional contracts.

Therefore, a research into the possible implementation of SE for that domain must focus first and foremost on the relations and information exchange between these two parties in order to ensure the continuation of information and the explicit relations between concepts.

SE is usually associated with integrated projects because the processes of the stakeholders need to be integrated, not only their data – the model or the outcomes. Therefore, looking into the integrated process of the stakeholders within a building structures project and identifying which changes have to be made there is important because once a process where each professional is consciously and consistently applying the SE concepts and storing the data at the same time is developed, then a proper implementation of SE practices can be achieved.

As systems are an essential part of the working practices in the structural engineering domain, the possible incorporation of dynamic assemblies of components and their relation to project data such as requirements can be investigated. The essential challenges of dealing with systems consist in the dynamicity and ambiguity of the different possible compositions and the fact that a single component can be a part of multiple systems at the same time. Therefore, the notion of systems can be discovered only in the documentation of the project and it remains an aspect, highly dependable on the choice of the engineer to prove a certain component within the context of a specific system.

While some of the ambiguity can be resolved by enforcing policies, the technical aspects facilitating such way of working should be researched upon. The notion of systems needs to be firstly included in all project-relevant data sources such as, for example, in the geometrical model of the project. The systems approach to requirements management would entail investigating the core differences between systems-relevant and component-relevant requirements and whether such concept would bring more clarity and accuracy to the formulation of requirements or on the contrary, cause unnecessary complexity and ambiguity.

Bibliography

Abanda, F. H. 2013. "Exploring the Relationships between Linked Open Data and Building Information Modelling." http://www.irbnet.de/daten/iconda/CIB_DC26507.pdf.

Abanda, F. H., J. H. M. Tah, and R. Keivani. 2013. "Trends in Built Environment Semantic Web Applications: Where Are We Today?" *Expert Systems with Applications* 40 (14): 5563–77. doi:10.1016/j.eswa.2013.04.027.

Akio. 2017. "Leveraging the Full Value of BIM's Interoperability Potential." *3D PERSPECTIVES*. May 2. http://blogs.3ds.com/perspectives/leveraging-full-value-bims-interoperability-potential-2/.

Allemang, Dean, and Jim Hendler. 2011a. "Chapter 5 - Querying the Semantic Web—SPARQL." In *Semantic Web for the Working Ontologist (Second Edition)*, 61–112. Boston: Morgan Kaufmann. doi:10.1016/B978-0-12-385965-5.10005-6.

Allemang, Dean, and Jim Hendler. 2011b. "Chapter 7 - RDF Schema." In *Semantic Web for the Working Ontologist (Second Edition)*, 125–52. Boston: Morgan Kaufmann. doi:10.1016/B978-0-12-385965-5.10007-X.

Aslaksen, E.W. 2005. "Systems Engineering and the Construction Industry," January. http://www.gumbooya.com/Systems_and_Engineering/Systems%20Engineering%20and%20the%20Construction%20Industry.pdf.

Aubin, Mr Paul F., Mr Darryl McClelland, Mr Martin Schmid PE, Mr Gregg Stanley, Mr Michael Brumm, and Mr Ron Bailey. 2012. *The Aubin Academy: Revit MEP 2013*. CreateSpace Independent Publishing Platform.

Azhar, S., and Hein, M. 2014. "Building Information Modeling (BIM): Benefits, Risks and Challenges," August. https://www.researchgate.net/publication/237569739_Building_Information_Modeling_BIM_Benefits_Risks_and_Challenges.

Beach, Thomas H., Omer F. Rana, Yacine Rezgui, and Manish Parashar. 2013. "Cloud Computing for the Architecture, Engineering & Construction Sector: Requirements, Prototype & Experience." *Journal of Cloud Computing: Advances, Systems and Applications* 2 (April): 8. doi:10.1186/2192-113X-2-8.

Beetz, J, W van den Braak, R Botter, S Zlatanova, and R de Laat. 2014. "Interoperable Data Models for Infrastructural Artefacts – a Novel IFC Extension Method Using RDF Vocabularies Exemplified with Quay Wall Structures for Harbors." In *eWork and eBusiness in Architecture, Engineering and Construction*, edited by Ardeshir Mahdavi, Bob Martens, and Raimar Scherer, 135–40. CRC Press. doi:10.1201/b17396-26.

Beetz, J., Leeuwen, J. P., and de Vries, B. 2005. "An Ontology Web Language Notation of the Industry Foundation Classes." https://pure.tue.nl/ws/files/2329360/Metis209539.pdf.

Beetz, J., Leeuwen, J. P., and de Vries, B. 2008. "IfcOWL A Case of Transforming EXPRESS Schemas Intot Onotlogies," July. https://pdfs.semanticscholar.org/1e8c/91b84f00dfd6d5aa5bcabbca0476015dd48f.pdf?_ga=1.90375186.904748179.1492202631.

Belsky, Michael, Rafael Sacks, and Ioannis Brilakis. 2016. "Semantic Enrichment for Building Information Modeling." *Computer-Aided Civil and Infrastructure Engineering* 31 (4): 261–74. doi:10.1111/mice.12128.

Berbers-Lee, T. 2001. "The Semantic Web." *Scientific American*. May. https://www.scientificamerican.com/article/the-semantic-web/.

Bergman. 2010. "Metamodeling in Domain Ontologies." http://www.mkbergman.com/wp-content/uploads/kalins-pdf/singles/913.pdf.

Bernal, F. 2017. "Internal Design Validation Attestation in BIM Models: A Collaboration of Semantic Web Technologies and BIM Collaboration Format."

Berners-Lee, T. 1998. "The World Wide Web: A Very Short Personal History." May. https://www.w3.org/People/Berners-Lee/ShortHistory.html.

Bontas, Mochol, and Tolksdorf. 2005. "Case Studies on Ontology Reuse.pdf." http://www.inf.fu-berlin.de/users/mochol/papers/i-KNOW05.pdf.

Borrmann, Beetz, Koch, and Lieblich. 2015. "Industry Foundation Classes-A Vendor-Independent Data Model for the Entire Life Cycle of a Building.pdf." In .

Bouw Informatie Raad. 2010. "Dutch BIM Levels." http://www.bouwinformatieraad.nl/.

buildingSMART. 2013. "IFC4 Introduction." March. http://www.buildingsmart-tech.org/ifc/IFC4/final/html/introduction.htm.

buildingSMART. 2015. "Open Standards - the Basics." *buildingSMART*. http://buildingsmart.org/standards/technical-vision/open-standards-101/.

buildingSMART. 2017. "IfcRoot." http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifckernel/lexical/ifcroot.htm.

Coinsweb. 2016. "Introductie Systems Engineering - CoinsWiki." March. http://www.coinsweb.nl/wiki/index.php/Introductie_Systems_Engineering.

Coinsweb. 2017. "Essentials - COINS2." http://www.coinsweb.nl/wiki2/index.php/Essentials.

Corcho, and Fernandez-Lopez. 2003. "Ontological Engineering: Principles, Methods, Toold and Languages.pdf." In *Ontologies for Software Engineering and Software Technology*. https://pdfs.semanticscholar.org/a139/caadd497e1fb33bb8741d48101efdce4f7c7.pdf.

Corry, Edward, James O'Donnell, Edward Curry, Daniel Coakley, Pieter Pauwels, and Marcus Keane. 2014. "Using Semantic Web Technologies to Access Soft AEC Data." *Advanced Engineering Informatics* 28 (4): 370–80. doi:10.1016/j.aei.2014.05.002.

Curry, Edward, James O'Donnell, Edward Corry, Souleiman Hasan, Marcus Keane, and Seán O'Riain. 2013. "Linking Building Data in the Cloud: Integrating Cross-Domain Building Data Using Linked Data." *Advanced Engineering Informatics* 27 (2): 206–19. doi:10.1016/j.aei.2012.10.003.

designingbuildings.co.uk. 2014. "File Formats for BIM - Designing Buildings Wiki." https://www.designingbuildings.co.uk/wiki/File_formats_for_BIM.

Ding, L. Y., B. T. Zhong, S. Wu, and H. B. Luo. 2016. "Construction Risk Knowledge Management in BIM Using Ontology and Semantic Web Technology." *Safety Science* 87 (August): 202–13. doi:10.1016/j.ssci.2016.04.008.

DSDM Consortium. 2008. "DSDM Atern Handbook (2008)." *Agile Business Consortium*. https://www.agilebusiness.org/content/moscow-prioritisation-0.

Farias, M. T., Roxin, A., and Nicolle, C. 2014. "A Rule Based System for Emantical Enrichment of Building Information Exchange," August. http://ceur-ws.org/Vol-1211/paper2.pdf.

Guess, A.R. 2015. "Top Trends in Semantic Technology for 2016." *DATAVERSITY*. December 21. http://www.dataversity.net/top-trends-in-semantic-technology-for-2016/.

Harrison, D., Donn, M., and Skates, H. 2003. "Applying Web Services within the AEC Industry: Enabling Semantic Searching and Information Exchange through the Digital Linking of the Knowledge Base," January. https://www.researchgate.net/publication/228805728_Applying_web_services_within_the_AEC_industry_enabling_semantic_searching_and_information_exchange_through_the_digital_linking_of_the_knowledge_base.

Hull, Elizabeth, Ken Jackson, and Jeremy Dick. 2005. *Requirements Engineering*. Springer Science & Business Media.

IfcWiki. 2017. "IDM - IfcWiki." http://www.ifcwiki.org/index.php/IDM.

INCOSE. 2015. "What Is Systems Engineering." http://www.incose.org/AboutSE/WhatIsSE.

Jallow, A.K., Demian, P., Baldwin, A. N., and Anumba, C. J. 2010. "An Integrated Requirements Management System for Construction Projects." http://www.engineering.nottingham.ac.uk/icccbe/proceedings/pdf/pf149.pdf.

Jansson, G., Schade, J., and Olofsson, T. 2013. "Requirements Management for the Design of Energy Efficient Buildings," September. http://www.itcon.org/papers/2013_16.content.09557.pdf.

Kamara, John M., Chimay J. Anumba, and Nosa F. O. Evbuomwan. 2002. *Capturing Client Requirements in Construction Projects*. Thomas Telford.

Karan, Irizarry, and Haymaker. 2015. "BIM and GIS Integration and Interoperability Based on
        Semantic Web Technology." *ResearchGate*, August.
        https://www.researchgate.net/publication/280612643_BIM_and_GIS_Integration_and_Inter
        operability_Based_on_Semantic_Web_Technology.

Kiviniemi, A., and Laakso, M. 2012. "The IFC Standard - A Review Of History Development And
        Standardization.pdf," May.
        https://helda.helsinki.fi/dhanken/bitstream/handle/10138/36870/Laakso_Kiviniemi_2012_T
        he_IFC_Standard_A_Review_Of_History_Development_And_Standardization.pdf?sequence=
        2&isAllowed=y.

Kiviniemi, Arto. 2005. "Requirements Management Interface to Building Product Models."

Kossiakoff, A., and Sweet, W. N. 2011. *Systems Engineering Principles and Practice*. 2nd Edition.
        http://eva.sepyc.gob.mx:8383/greenstone3/sites/localsite/collect/ciencia1/index/assoc/HAS
        H3e18.dir/22050022.pdf.

Krijnen, T., and Beetz, J. 2017. "An Efficient SPARQL Engine for Unindexed Binary-Formatted IFC
        Building Models." http://www.semantic-web-journal.net/system/files/swj1625.pdf.

Krima, S., Barbau, R., and Foufou, S. 2012. "OntoSTEP: OWL-DL Ontology for STEP."
        https://www.nist.gov/publications/ontostep-owl-dl-ontology-step-0.

Lee, Ghang, Rafael Sacks, and Charles M. Eastman. 2006. "Specifying Parametric Building Object
        Behavior (BOB) for a Building Information Modeling System." *Automation in Construction*,
        Knowledge Enabled Information System Applications in Construction, 15 (6): 758–76.
        doi:10.1016/j.autcon.2005.09.009.

Lu, Peng, Fung, and Rowlinson. 2015. "MacLeamy Curve - With Figures." *ResearchGate*, November.
        https://www.researchgate.net/publication/283427563_MacLeamy_Curve_-_With_figures.

Miller, E. 1998. "An Introduction to the Resource Description Framework." May.
        http://www.dlib.org/dlib/may98/miller/05miller.html.

Mogk, N. W. 2014. "A Requirements Management System Based on an Optimization Model of the
        Design Process." *Procedia Computer Science*, 2014 Conference on Systems Engineering
        Research, 28 (January): 221–27. doi:10.1016/j.procs.2014.03.028.

Obitko. 2007. "RDF Graph and Syntax - Introduction to Ontologies and Semantic Web."
        http://www.obitko.com/tutorials/ontologies-semantic-web/rdf-graph-and-syntax.html.

openBIMstandards. 2007. "ifcOWL – openBIMstandards.org."
        http://openbimstandards.org/standards/ifcowl/.

Ozkaya, Ipek, and Ömer Akin. 2007. "Tool Support for Computer-Aided Requirement Traceability in
        Architectural Design: The Case of DesignTrack." *Automation in Construction* 16 (5): 674–84.
        doi:10.1016/j.autcon.2006.11.006.

Parsanezhad, and Tarandi. 2016. "Formalized Requirements Management in the Briefing and Design
        phase,A Pivotal Review of Literature." *ResearchGate*, September.
        https://www.researchgate.net/publication/309040291_Formalized_requirements_managem
        ent_in_the_briefing_and_design_phase_A_pivotal_review_of_literature.

Pauwels, P., and De Meyer, R. 2011. "Interoperability for the Design and Construction Industry
        through Semantic Web Technology."
        https://biblio.ugent.be/publication/1077599/file/1077602.pdf.

Pauwels, P., Krijnen, T., Beetz, J., and Terkaj, W. 2017. "Enhancing the ifcOWL Ontology with an
        Alternative Representation for Geometric Data," April.
        http://www.sciencedirect.com/science/article/pii/S0926580517301826.

Pauwels, Pieter, Davy Van Deursen, Jos De Roo, Tim Van Ackere, Ronald De Meyer, Rik Van de Walle,
        and Jan Van Campenhout. 2011. "Three-Dimensional Information Exchange over the
        Semantic Web for the Domain of Architecture, Engineering, and Construction." *Artificial
        Intelligence for Engineering Design, Analysis and Manufacturing* 25 (04): 317–332.

Pauwels, Pieter, Sijie Zhang, and Yong-Cheol Lee. 2017. "Semantic Web Technologies in AEC Industry:
        A Literature Overview." *Automation in Construction* 73 (January): 145–65.
        doi:10.1016/j.autcon.2016.10.003.

Pazlar, Tomaž, and Žiga Turk. 2008. "Interoperability in Practice: Geometric Data Exchange Using the IFC Standard." *Journal of Information Technology in Construction (ITcon)* 13 (24): 362–80.

Pegoraro, Camila, Istefani Carísio de Paula, Camila Pegoraro, and Istefani Carísio de Paula. 2017. "Requirements Processing for Building Design: A Systematic Review." *Production* 27. doi:10.1590/0103-6513.212116.

Pinto, and Martins. 2001. "A Methodology for Ontology Integration," October. https://dl.acm.org/citation.cfm?id=500759.

ProRail, Rijkswaterstaat, and Bouwend nederland. 2013. "Guideline for Systems Engineering." November. https://www.leidraadse.nl/welcome?source=flag.

Relatics B.V. 2017. "Relatics." *Relatics*. Accessed August 11. https://www.relatics.com/product/.

Rijkswaterstaat. 2017. "Producten en standaarden." Webpagina. https://www.rijkswaterstaat.nl/zakelijk/werken-aan-infrastructuur/efficienter-werken/bouwwerk-informatie-model/producten-en-standaarden.aspx.

Sack, H. 2014. "Knowledge Engineering with Semantic Web Technologies." *openHPI*. Mai. https://open.hpi.de/courses/semanticweb2014.

Schevers, H., and Drogemuller, R. 2005. "Converting the Industry Foundation Classes to the Web Ontology Language." http://eprints.qut.edu.au/9347/1/9347.pdf.

SE-BIM Werkgroep. 2017. "Toelichting Op Systems Engineering Uitwisselingsstandaard Voor de Nederlandse Bouwketen."

Serginson, Michael, George Mokhtar, and Graham Kelly. 2013. "A Theoretical Comparison of Traditional and Integrated Project Delivery Design Processes on International BIM Competitions." *International Journal of 3-D Information Modeling* 2: 52–64.

Shen, L., and Chua, D. 2011. "Application of BIM and IT for Project Collaboration," September. http://www.ppml.url.tw/EPPM/conferences/2011/download/SESSION3/67_76.pdf.

Simeone, Davide, and Stefano Cursi. 2016. "The Role of Semantic Enrichment in Building Information Modelling." *Tema:Tempo, Materia, Architettura* 2 (2): 22–30. doi:10.17410/tema.v2i2.105.

Steel, James, Robin Drogemuller, and Bianca Toth. 2010. "Model Interoperability in Building Information Modelling." *Software and Systems Modeling* 11 (1): 99–109.

Svetel, I., and M. Pejanović. 2010. "The Role of the Semantic Web for Knowledge Management in the Construction Industry." *Informatica* 34 (3). http://www.informatica.si/index.php/informatica/article/view/307.

Terkaj, Walter, and Aleksandra Šojić. 2015. "Ontology-Based Representation of IFC EXPRESS Rules: An Enhancement of the ifcOWL Ontology." *Automation in Construction* 57 (September): 188–201. doi:10.1016/j.autcon.2015.04.010.

Törmä, S. 2013. "Semantic Linking of Building Information Models." http://www.cs.hut.fi/~sto/Torma-2013-BIM-LD.pdf.

US Department of Energy. 2008. "Managing Design and Construction Using Systems Engineering for Use with DOE O 413.3A," September. https://science.energy.gov/~/media/opa/pdf/processes-and-procedures/doe/g4133-1_Managing_Design_Construction.pdf.

v. d. Liende, J. 2017. "The Power of Metaphors." Accessed April 3. http://jeroenvanderliende.com/the-power-of-metaphors/?lang=en.

van de Ven, N. 2017. "Mutation Management in BIM Models during Operations & Maintenance: Managing Operations & Maintenance Building Models with the Use of Industry Foundation Classes and Linked Data."

Verstraeten, R., Rauwels, P., De Meyer, R., Meeus, W., Van Campenhout, J., and Lateur, G. 2009. "IFC-Based Calculation of the Flemish Energy Performance Standard." https://biblio.ugent.be/publication/482379/file/6814885.pdf.

Wikipedia. 2010. "5 Star Linked Data - Government Linked Data (GLD) Working Group." https://www.w3.org/2011/gld/wiki/5_Star_Linked_Data.

Wikipedia. 2016. "Semantic Triple." *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Semantic_triple&oldid=738439464.

Wikipedia. 2017a. "SPARQL." *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=SPARQL&oldid=770096562.

Wikipedia. 2017b. "Industry Foundation Classes." *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=Industry_Foundation_Classes&oldid=774795046
.

Wikipedia,. 2017. "Eurocodes." *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=Eurocodes&oldid=779062285.

Young, R. 2004. *The Requirements Engineering Handbook*.
http://www.acqnotes.com/Attachments/The%20Requirements%20Engineering%20Handboo
k%20by%20Ralph%20R.%20Young.pdf.

Yu, A. T. W., Chan, E. H. W., and Shen, Q. G. 2010. "Managing Employers Requirements in
Construction Industry.pdf," May.
https://www.researchgate.net/profile/Ann_Yu2/publication/47870199_Managing_employer
s%27_requirements_in_construction_industry/links/0fcfd50bc632bb7019000000/Managing-
employers-requirements-in-construction-industry.pdf.

Zhang, C., and Beetz, J. 2016. "Querying Linked Building Data Using SPARQL with Functional
Extensions." In *ResearchGate*.
https://www.researchgate.net/publication/308057797_Querying_Linked_Building_Data_Usi
ng_SPARQL_with_Functional_Extensions.

Zhang, C., and Beetz, J. 2017. "BimSPARQL - A Case of Extending SPARQL Functions for Querying
Linked Building Data." http://www.semantic-web-journal.net/system/files/swj1628.pdf.

Zhang, L., and Issa, R. 2014. "Development of IFC-Based Construction Industry Ontology for
Information Retrieval from IFC Models," January.
https://www.researchgate.net/publication/267857855_Development_of_IFC-
based_Construction_Industry_Ontology_for_Information_Retrieval_from_IFC_Models.

Appendix I: Eurocodes

- ***Eurocode**: Basis of structural design* (EN 1990)

- ***Eurocode 1**: Actions on structures* (EN 1991)
    Part 1-1: Densities, self-weight, imposed loads for buildings (EN 1991-1-1)
    Part 1-2: Actions on structures exposed to fire (EN 1991-1-2)
    Part 1-3: General actions - Snow loads (EN 1991-1-3)
    Part 1-4: General actions - Wind actions (EN 1991-1-4)
    Part 1-5: General actions - Thermal actions (EN 1991-1-5)
    Part 1-6: General actions - Actions during execution (EN 1991-1-6)
    Part 1-7: General actions - Accidental Actions (EN 1991-1-7)
    Part 2: Traffic loads on bridges (EN 1991-2)
    Part 3: Actions induced by cranes and machinery (EN 1991-3)
    Part 4: Silos and tanks (EN 1991-4)

- ***Eurocode 2**: Design of concrete structures* (EN 1992)
    Part 1-1: General rules, and rules for buildings (EN 1992-1-1)
    Part 1-2: Structural fire design (EN 1992-1-2)
    Part 1-3: Precast Concrete Elements and Structures (EN 1992-1-3)
    Part 1-4: Lightweight aggregate concrete with closed structure (EN 1992-1-4)
    Part 1-5: Structures with unbonded and external prestressing tendons (EN 1992-1-5)
    Part 1-6: Plain concrete structures (EN 1992-1-6)
    Part 2: Reinforced and prestressed concrete bridges (EN 1992-2)
    Part 3: Liquid retaining and containing structures (EN 1992-3)

- ***Eurocode 3**: Design of steel structures* (EN 1993)
    Part 1-1: General rules and rules for buildings (EN 1993-1-1)
    Part 1-2: General rules - Structural fire design (EN 1993-1-2)
    Part 1-3: General rules - Supplementary rules for cold-formed members (EN 1993-1-3)
    Part 1-4: General rules - Supplementary rules for stainless steels (EN 1993-1-4)
    Part 1-5: Plated structural elements (EN 1993-1-5)
    Part 1-6: Strength and Stability of Shell Structures (EN 1993-1-6)
    Part 1-7: General Rules - Supplementary rules for planar plated structural elements with out of plane loading (EN 1993-1-7)
    Part 1-8: Design of joints (EN 1993-1-8)
    Part 1-9: Fatigue (EN 1993-1-9)
    Part 1-10: Material Toughness and through-thickness properties (EN 1993-1-10)
    Part 1-11: Design of Structures with tension components (EN 1993-1-11)
    Part 1-12: High Strength steels (EN 1993-1-12)
    Part 2: Steel Bridges (EN 1993-2)
    Part 3-1: Towers, masts, and chimneys (EN 1993-3-1)
    Part 3-2: Towers, masts, and chimneys - Chimneys (EN 1993-3-2)

Part 4-1: Silos   (EN 1993-4-1)
Part 4-2: Tanks   (EN 1993-4-2)
Part 4-3: Pipelines   (EN 1993-4-3)
Part 5: Piling   (EN 1993-5)
Part 6: Crane supporting structures   (EN 1993-6)

- *Eurocode 4: Design of composite steel and concrete structures*   **(EN 1994)**
    Part 1-1: General rules and rules for buildings   (EN 1994-1-1)
    Part 1-2: Structural fire design   (EN 1994-1-2)
    Part 2: General rules and rules for bridges   (EN 1994-2)

- *Eurocode 5: Design of timber structures*   **(EN 1995)**
    Part 1-1: General – Common rules and rules for buildings   (EN 1995-1-1)
    Part 1-2: General – Structural fire design   (EN 1995-1-2)
    Part 2: Bridges   (EN 1995-2)

- *Eurocode 6: Design of masonry structures*   **(EN 1996)**
    Part 1-1: General – Rules for reinforced and unreinforced masonry structures   (EN 1996-1-1)
    Part 1-2: General rules – Structural fire design   (EN 1996-1-2)
    Part 2: Design, selection of materials and execution of masonry   (EN 1996-2)
    Part 3: Simplified calculation methods for unreinforced masonry structures   (EN 1996-3)

- *Eurocode 7: Geotechnical design*   **(EN 1997)**
    Part 1: General rules   (EN 1997-1)
    Part 2: Ground investigation and testing   (EN 1997-2)
    Part 3: Design assisted by field testing   (EN 1997-3)

- *Eurocode 8: Design of structures for earthquake resistance*   **(EN 1998)**
    Part 1: General rules, seismic actions and rules for buildings   (EN 1998-1)
    Part 2: Bridges   (EN 1998-2)
    Part 3: Assessment and retrofitting of buildings   (EN 1998-3)
    Part 4: Silos, tanks and pipelines   (EN 1998-4)
    Part 5: Foundations, retaining structures and geotechnical aspects   (EN 1998-5)
    Part 6: Towers, masts and chimneys   (EN 1998-6)

- *Eurocode 9: Design of aluminum structures*   **(EN 1999)**
    Part 1-1: General structural rules   (EN 1999-1-1)
   Part 1-2: Structural fire design   (EN 1999-1-2)
   Part 1-3: Structures susceptible to fatigue   (EN 1999-1-3)
    Part 1-4: Cold-formed structural sheeting   (EN 1999-1-4)
   Part 1-5: Shell structures   (EN 1999-1-5)

## Appendix II: MAVO communication workflow (preliminary design)
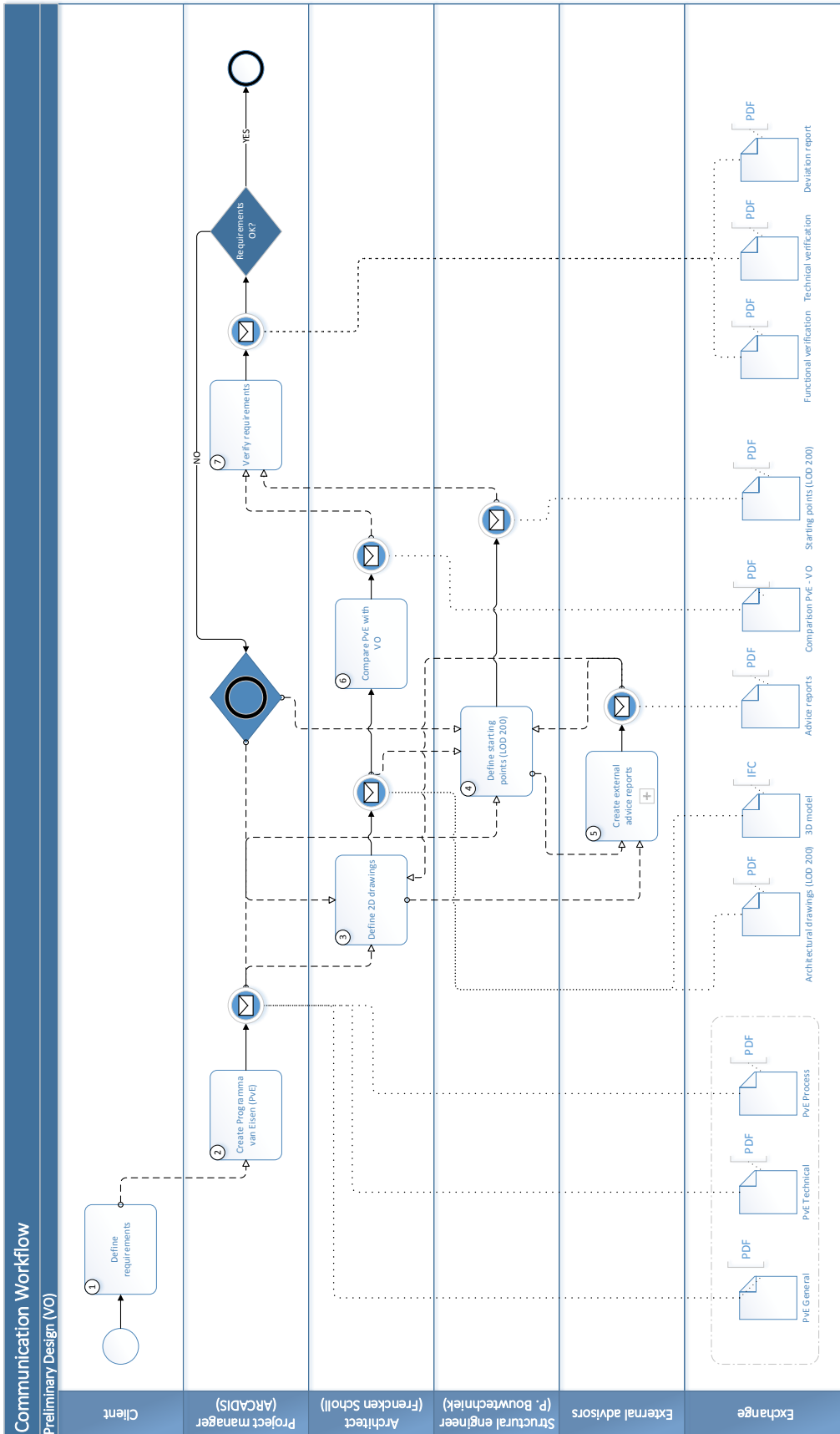
## Appendix III: MAVO communication workflow (final design)
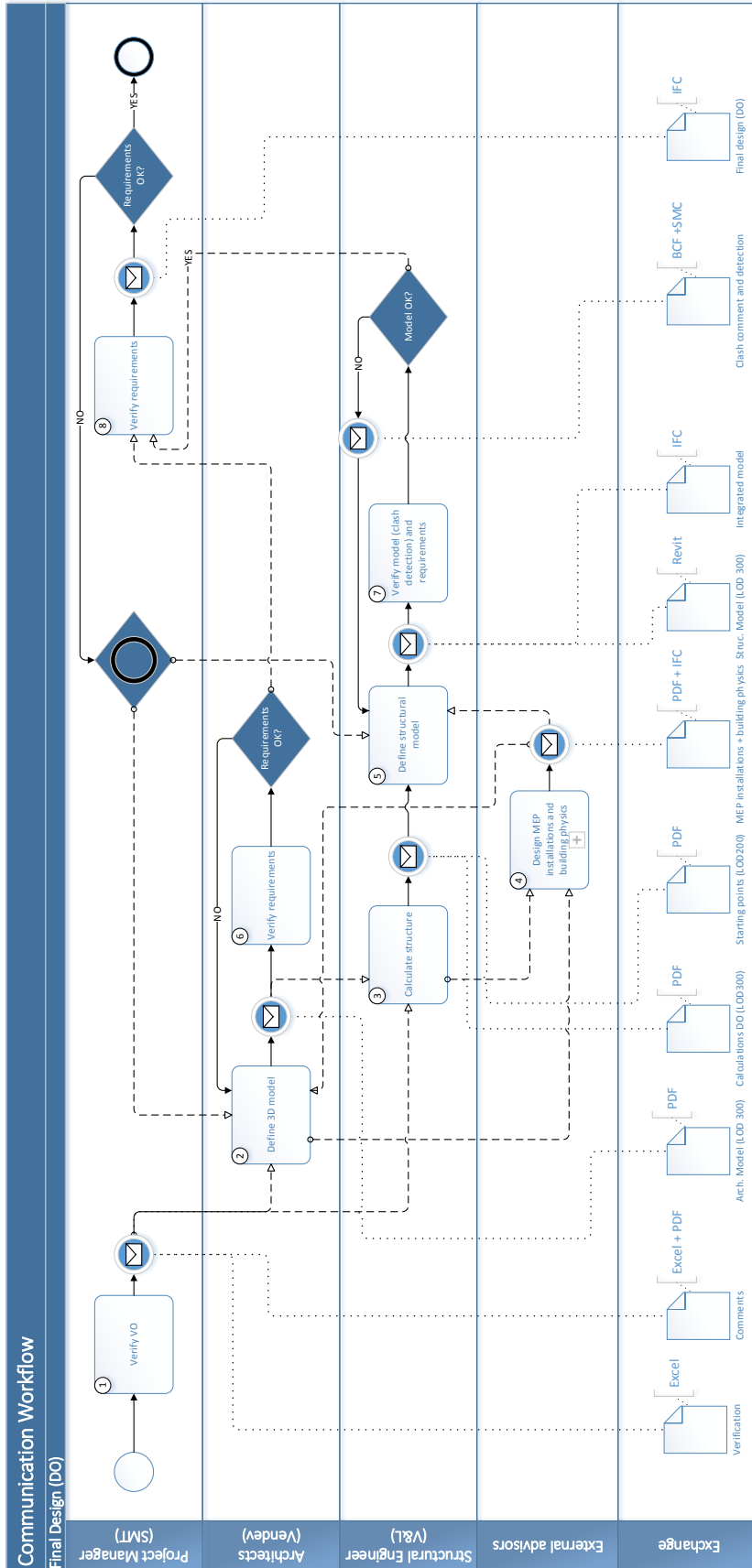
## Appendix IV: MAVO input/output workflow (preliminary design)
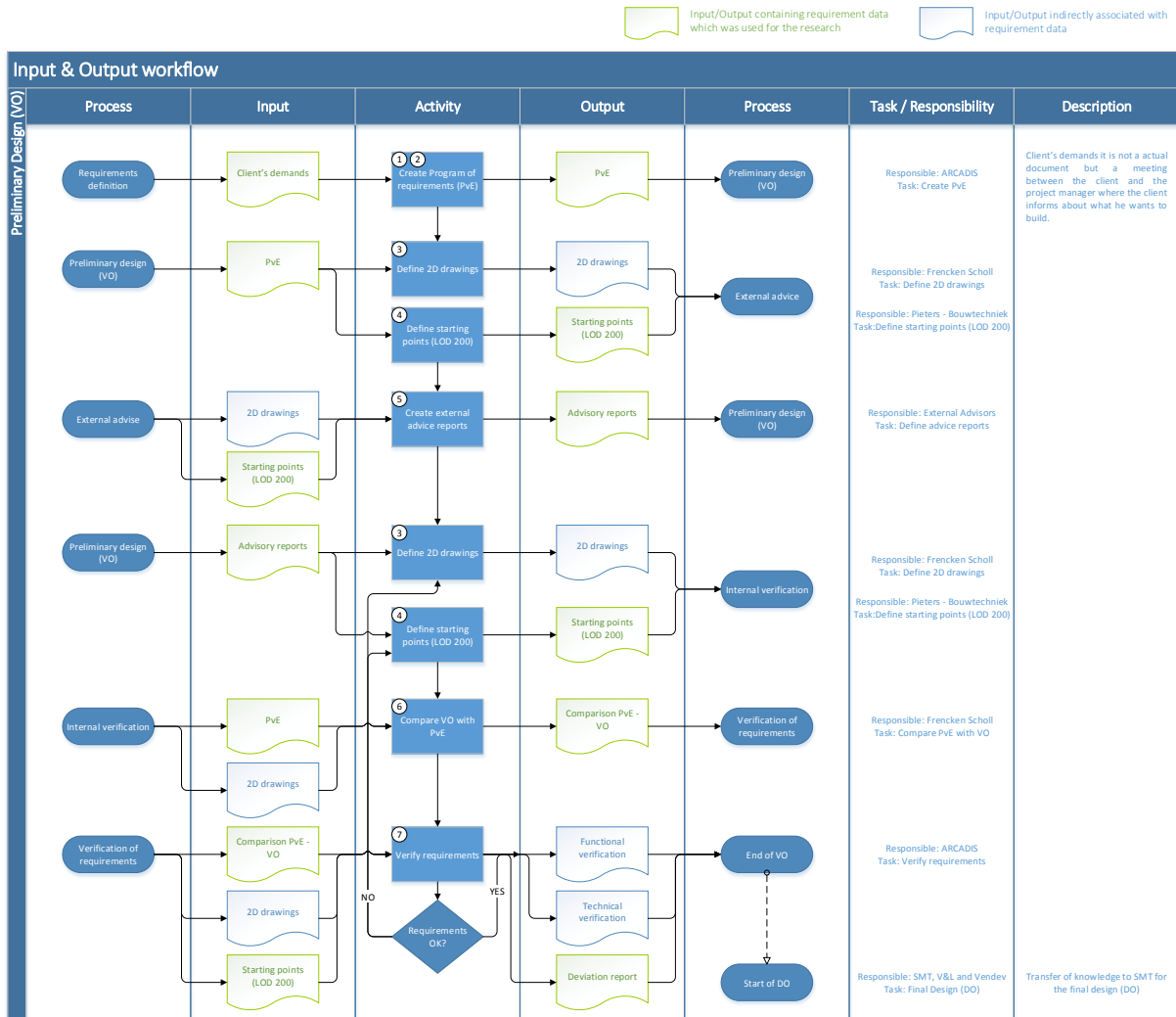
Appendix V: MAVO input/output workflow (final design)
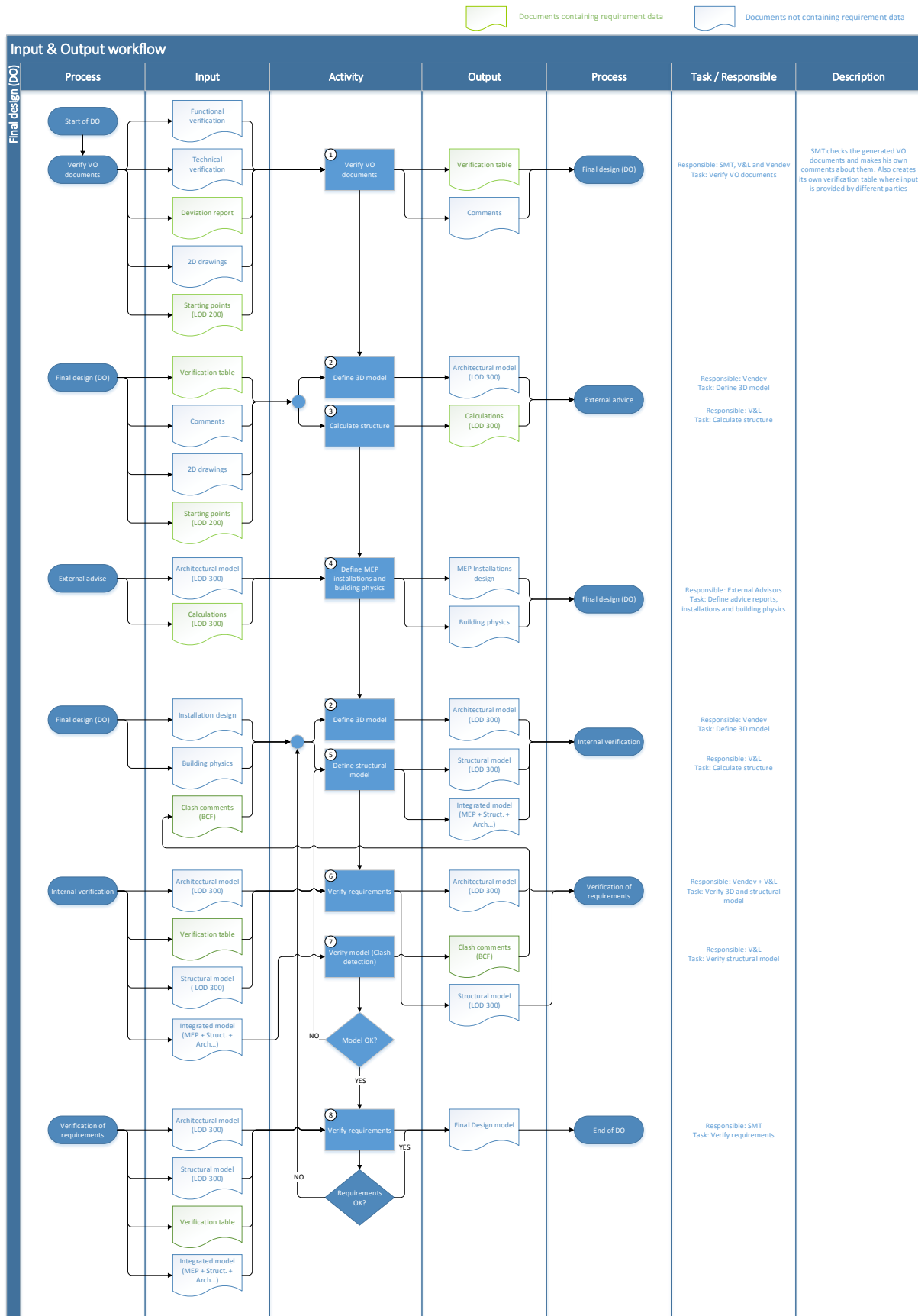
Appendix VI: Process schema – guidelines for the tool implementation
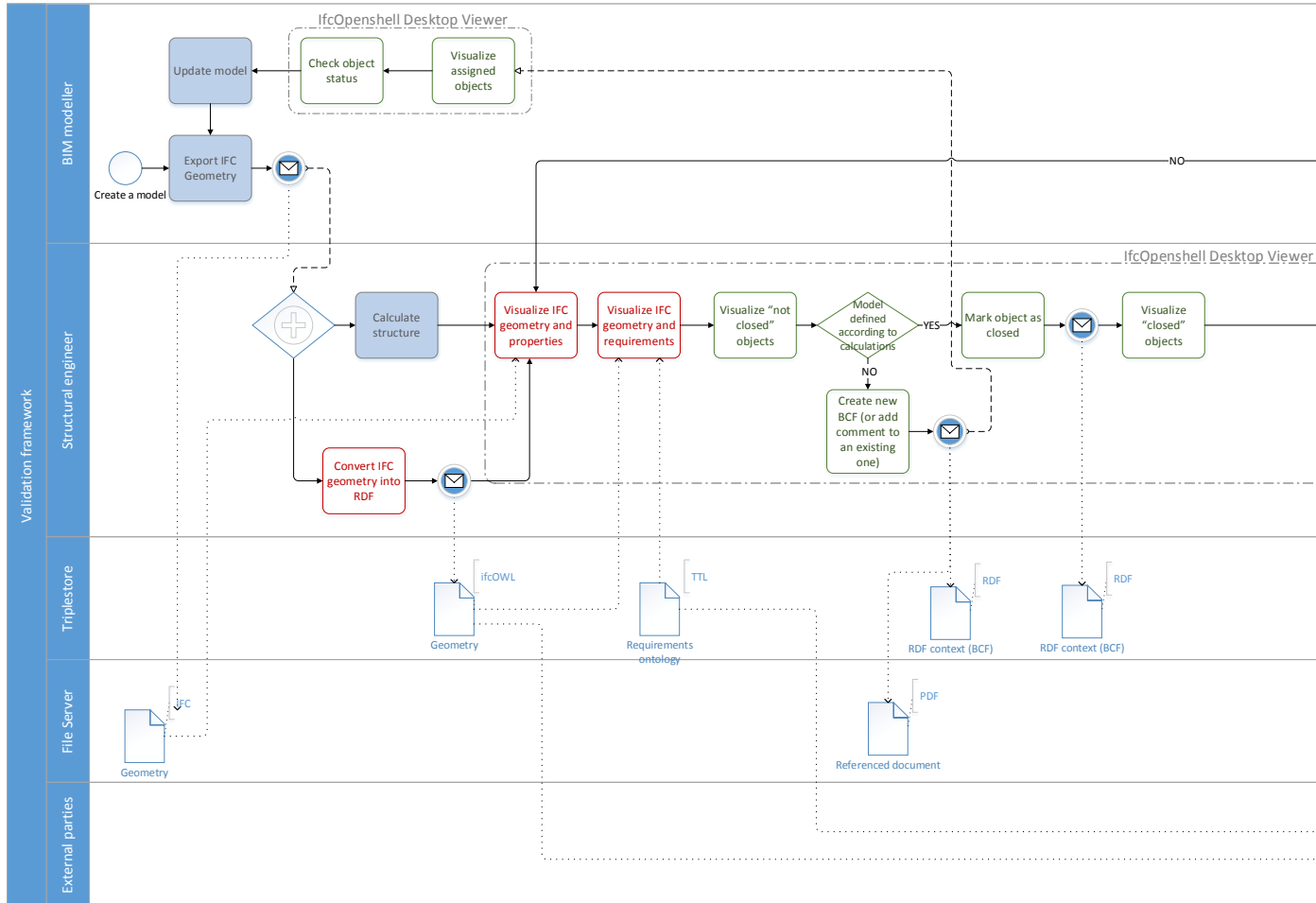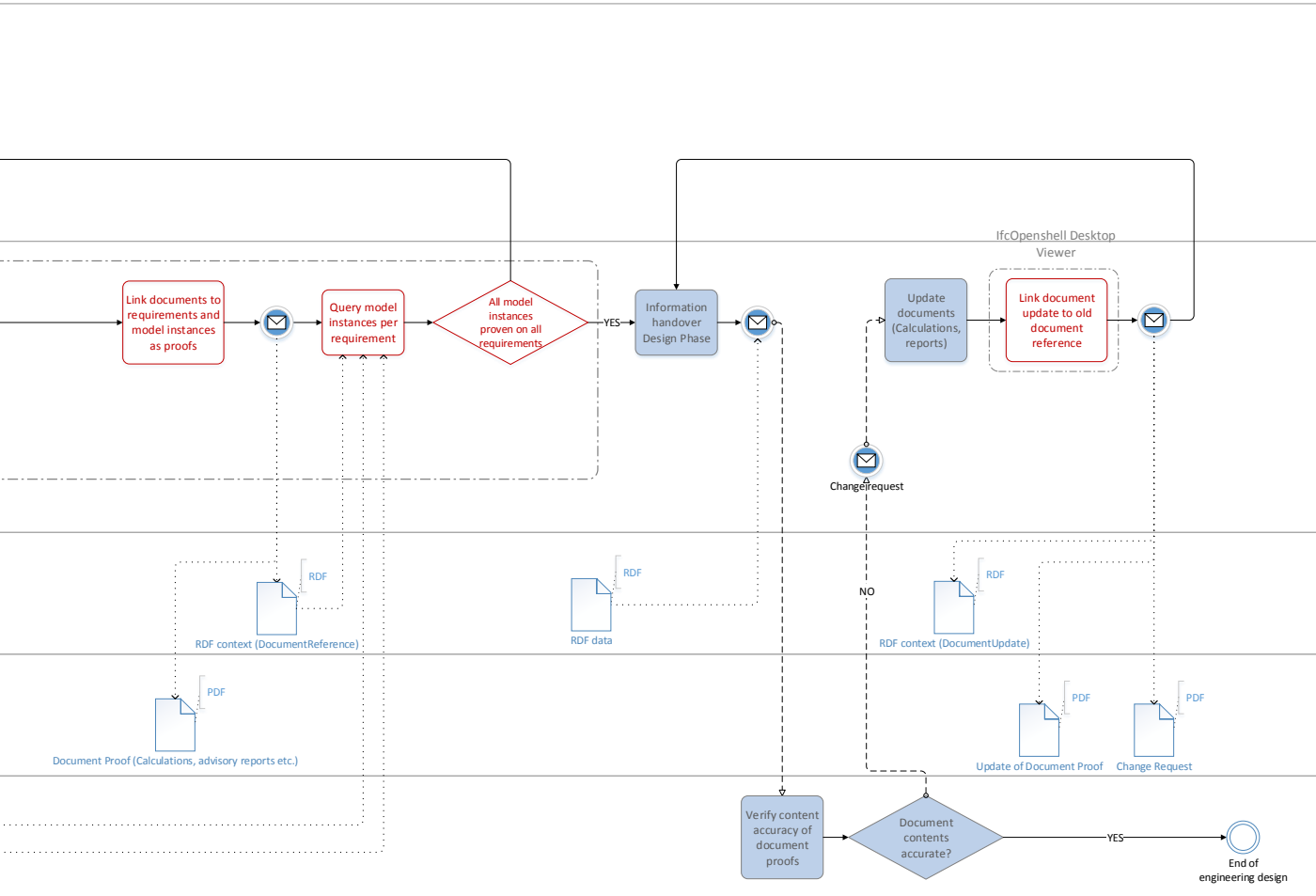
Appendix VII: General requirements ontology

```
# baseURI: http://example.org/ontology/generalrequirements
# prefix: greq

@prefix greq: <http://example.org/ontology/generalrequirements#> .
@prefix ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
.
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .


<http://example.org/ontology/generalrequirements>
  rdf:type owl:Ontology ;
  owl:versionInfo "Created with TopBraid Composer" ;
.


#######   Classes   #######

greq:Requirement
  rdf:type owl:Class ;
  rdfs:comment "General structural engineering requirements for
building structures." ;
  rdfs:subClassOf owl:Thing ;
.
greq:Label
  rdf:type owl:Class ;
  rdfs:comment "Indentifier of the requirement." ;
  rdfs:subClassOf owl:Thing ;
.
greq:Functional_requirement
  rdf:type owl:Class ;
  rdfs:comment "Requirements relating to the functions which need to
be realized; they indicate 'what the system should do'." ;
  rdfs:label "F" ;
  rdfs:subClassOf greq:Requirement ;
.
greq:Aspect_requirement
  rdf:type owl:Class ;
  rdfs:comment "Requirements relating to supporting functions or
aspects of the system." ;
  rdfs:label "A" ;
  rdfs:subClassOf greq:Requirement ;
.
greq:Object_requirement
  rdf:type owl:Class ;
  rdfs:comment "Requirements which arise as a result from the design
choices of client and contractor and have an impact on, for example,
the shape, color and strength of the object." ;
  rdfs:label "O" ;
  rdfs:subClassOf greq:Requirement ;
.
greq:System_Interaction_requirement
```

```
   rdf:type owl:Class ;
   rdfs:comment "Requirements which come as a result from relations
between the system and the system's environment (external
requirements), as well as from interactions between different
components within the system (internal interactions/clashes)." ;
   rdfs:label "S" ;
   rdfs:subClassOf greq:Requirement ;
.
greq:Prevention_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:Aspect_requirement ;
.
greq:Usability_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:Aspect_requirement ;
.
greq:Loadbearing_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:Functional_requirement ;
.
greq:Stability_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:Functional_requirement ;
.
greq:Material_Quality_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:Object_requirement ;
.
greq:External_Interaction_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:System_Interaction_requirement ;
.
greq:Internal_Interaction_requirement
   rdf:type owl:Class ;
   rdfs:subClassOf greq:System_Interaction_requirement ;
.

#######   Object Properties   #######

greq:isApplicableToObject
   rdf:type owl:FunctionalProperty ;
   rdf:type owl:ObjectProperty ;
   rdfs:domain greq:Requirement ;
   rdfs:label "Property for linking general requirements to building
element types." ;
.
greq:hasLabel
   rdf:type rdf:Property ;
   rdfs:domain greq:Requirement ;
   rdfs:range greq:Label ;
   owl:inverseOf greq:identifiesRequirement;
   rdfs:label "Property for linking requirements to lables." ;
.
greq:identifiesRequirement
   rdf:type rdf:Property ;
```

```
    rdfs:domain greq:Label ;
    rdfs:range greq:Requirement ;
    owl:inverseOf greq:hasLabel;
    rdfs:label "Property for linking labels to requirements." ;
.


#######   Data Properties   #######

greq:hasCreationDate
    rdf:type owl:DatatypeProperty ;
    rdfs:domain greq:Requirement ;
    rdfs:label "Date on which the general requirement instance is
created." ;
    rdfs:range xsd:date ;
.


#######   Indivisuals   #######

greq:Crack_width
    rdf:type greq:Usability_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:A2.1 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcFooting ;
    greq:isApplicableToObject ifc:IfcPile ;
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcStair ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
.
greq:Crash_load
    rdf:type greq:Prevention_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:A1.1 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcMember ;
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
.
greq:Deflection
    rdf:type greq:Usability_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:A2.2 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcMember ;
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcStair ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
.
```

```
greq:Earthquake_load
  rdf:type greq:Stability_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:F2.2 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcPile ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
greq:Element_clashes
  rdf:type greq:Internal_Interaction_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:S2.1 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcPile ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcStair ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
greq:Element_connections
  rdf:type greq:Internal_Interaction_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:S2.2 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcPile ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcStair ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
greq:Fire_resistance
  rdf:type greq:Prevention_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:A1.2 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
  rdfs:label "A1.2" ;
.
greq:Ground_water_impact
  rdf:type greq:External_Interaction_requirement ;
```

```
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:S1.1 ;
    greq:isApplicableToObject ifc:IfcFooting ;
    greq:isApplicableToObject ifc:IfcPile ;
    rdfs:comment "..." ;
  .
greq:Moment
    rdf:type greq:Loadbearing_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:F1.3 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcFooting ;
    greq:isApplicableToObject ifc:IfcMember ;
    greq:isApplicableToObject ifc:IfcPile ;
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcStair ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
  .
greq:Normal_forces
    rdf:type greq:Loadbearing_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:F1.1 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcFooting ;
    greq:isApplicableToObject ifc:IfcMember ;
    greq:isApplicableToObject ifc:IfcPile ;
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcStair ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
  .
greq:Nuisance_impact
    rdf:type greq:External_Interaction_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:S1.3 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcFooting ;
    greq:isApplicableToObject ifc:IfcMember ;
    greq:isApplicableToObject ifc:IfcPile ;
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcStair ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
  .
greq:Second_order_deflection
    rdf:type greq:Stability_requirement ;
    greq:hasCreationDate "2017-08-20"^^xsd:date ;
    greq:hasLabel greq:F2.3 ;
    greq:isApplicableToObject ifc:IfcBeam ;
    greq:isApplicableToObject ifc:IfcColumn ;
    greq:isApplicableToObject ifc:IfcMember ;
```

```
    greq:isApplicableToObject ifc:IfcSlab ;
    greq:isApplicableToObject ifc:IfcWallStandardCase ;
    rdfs:comment "..." ;
.
greq:Shear_force
  rdf:type greq:Loadbearing_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:F1.2 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcPile ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcStair ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
greq:Soil_impact
  rdf:type greq:External_Interaction_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:S1.2 ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcPile ;
  rdfs:comment "..." ;
.
greq:Stiffness
  rdf:type greq:Material_Quality_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:O1.2 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcPile ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcStair ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
greq:Strength
  rdf:type greq:Material_Quality_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:O1.1 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcFooting ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcPile ;
  greq:isApplicableToObject ifc:IfcSlab ;
  greq:isApplicableToObject ifc:IfcStair ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
```

```
greq:Wind_load
  rdf:type greq:Stability_requirement ;
  greq:hasCreationDate "2017-08-20"^^xsd:date ;
  greq:hasLabel greq:F2.1 ;
  greq:isApplicableToObject ifc:IfcBeam ;
  greq:isApplicableToObject ifc:IfcColumn ;
  greq:isApplicableToObject ifc:IfcMember ;
  greq:isApplicableToObject ifc:IfcStair ;
  greq:isApplicableToObject ifc:IfcWallStandardCase ;
  rdfs:comment "..." ;
.
greq:A1.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Crash_load ;
.
greq:A1.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Fire_resistance ;
.
greq:A2.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Crack_width "" ;
.
greq:A2.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Deflection ;
.
greq:F1.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Normal_force ;
.
greq:F1.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Shear_force ;
.
greq:F1.3
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Moment ;
.
greq:F2.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Wind_load ;
.
greq:F2.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Earthquake_load ;
.
greq:F2.3
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Second_order_deflection ;
.
greq:O1.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Strength ;
.
```

```
greq:O1.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Stiffness ;
.
greq:S1.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Ground_water_impact ;
.
greq:S1.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Soil_impact ;
.
greq:S1.3
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Nuisance_impact ;
.
greq:S2.1
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Element_clashes ;
.
greq:S2.2
  rdf:type greq:Label ;
  greq:identifiesRequirement greq:Element_connections ;
.
```

Appendix VIII: Additional ontological definitions from the tool

```
# baseURI: http://example.org/ontology/generalrequirements
# prefix: greq

@prefix greq: <http://example.org/ontology/generalrequirements#> .
@prefix ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>.
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .


<http://example.org/ontology/generalrequirements#>
  rdf:type owl:Ontology ;
  owl:versionInfo "Created with TopBraid Composer" ;
.

#######     Classes      #######

greq:FTP_Location
  rdf:type owl:Class ;
  rdfs:label "Document location on a file server." ;
  rdfs:subClassOf owl:Thing ;
  owl:disjointWith greq:Document_Reference ;
  owl:disjointWith greq:Document_Update ;
  owl:disjointWith greq:Change_Request ;

 .
greq:Document_Reference
  rdf:type owl:Class ;
  rdfs:subClassOf owl:Thing ;
  owl:disjointWith greq:Change_Request ;
  owl:disjointWith greq:Document_Update ;
  owl:disjointWith greq:FTP_Location ;
.
greq:Document_Update
  rdf:type owl:Class ;
  rdfs:subClassOf owl:Thing ;
  owl:disjointWith greq:Change_Request ;
  owl:disjointWith greq:Document_Reference ;
  owl:disjointWith greq:FTP_Location ;
.
greq:Change_Request
  rdf:type owl:Class ;
  rdfs:subClassOf owl:Thing ;
  owl:disjointWith greq:Document_Reference ;
  owl:disjointWith greq:Document_Update ;
  owl:disjointWith greq:FTP_Location ;
.

#######    Object Properties    #######

greq:isAnUpdateOf
  rdf:type owl:FunctionalProperty ;
```

```
  rdf:type owl:ObjectProperty ;
  rdfs:domain greq:Document_Update ;
  rdfs:label "Property for linking a document update to the original
document." ;
  rdfs:range greq:Document_Reference ;
.
greq:hasChangeRequest
  rdf:type owl:ObjectProperty ;
  rdfs:domain greq:Document_Update ;
  rdfs:label "Property for linking change request to the document
update." ;
  rdfs:range greq:Change_Request ;
.
greq:hasFTPLocation
  rdf:type owl:FunctionalProperty ;
  rdf:type owl:ObjectProperty ;
  rdfs:domain greq:Document_Reference ;
  rdfs:domain greq:Document_Update ;
  rdfs:domain greq:Change_Request ;
  rdfs:label "Property for determining the location of a document on
a file server." ;
  rdfs:range greq:FTP_Location ;
.

#######   Data Properties   #######

greq:hasFragment
  rdf:type owl:DatatypeProperty ;
  rdfs:domain greq:Document_Reference ;
  rdfs:label "Property refering to a specific page of a document." ;
  rdfs:range xsd:string ;
.
greq:hasDate
  rdf:type owl:DatatypeProperty ;
  rdfs:domain greq:Document_Reference ;
  rdfs:domain greq:Document_Update ;
  rdfs:domain greq:Change_Request ;
  rdfs:range xsd:dateTime ;
.
greq:hasType
  rdf:type owl:DatatypeProperty ;
  rdfs:domain greq:Document_Reference ;
  rdfs:label "Property refering to the type of document proof." ;
  rdfs:range xsd:string ;
.
greq:hasCreator
  rdf:type owl:DatatypeProperty ;
  rdfs:domain greq:Document_Reference ;
  rdfs:domain greq:Document_Update ;
  rdfs:domain greq:Change_Request ;
  rdfs:label "Property identifying the person linking a document." ;
  rdfs:range xsd:string ;
.
greq:provesRequirement
  rdf:type owl:DatatypeProperty ;
```

```
    rdfs:domain greq:Document_Reference ;
    rdfs:label "Property identifying the requirement which a document
reference proves" ;
    rdfs:range xsd:string ;
.
greq:refersTo
    rdf:type owl:DatatypeProperty ;
    rdfs:domain greq:Document_Reference ;
    rdfs:label "Property refering to the GUID of an element." ;
    rdfs:range xsd:string ;
.
```

Appendix IX: Python script

```python
import os
import datetime
import sys
import time
import uuid

import ifcopenshell
import ifcopenshell.geom

settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)

from collections import defaultdict

from PyQt4 import QtCore, QtGui
from OCC.Display.backend import get_backend

get_backend("qt-pyqt4")
import OCC.Display.qtDisplay
from OCC.Display.qtDisplay import qtViewer3d

from OCC.gp import *
import OCC.Bnd, OCC.BRepBndLib
from OCC.Aspect import Aspect_GT_Rectangular, Aspect_GDM_Lines
from OCC.BRepPrimAPI import BRepPrimAPI_MakeBox

import Query1
import Query2
import Query3
import Query4

import urllib
import httplib2

from Tkinter import *
from Tkinter import W, E
from tkFileDialog import askopenfilename

import Tkinter as tk

from ftplib import FTP

from rdflib import Graph, Literal, URIRef, Namespace, XSD, RDF
from rdflib.namespace import NamespaceManager

import SPARQLWrapper
from SPARQLWrapper import RDFXML

greq = Namespace("http://example.org/ontology/generalrequirements#")

guid_selection = None

#This class creates a custom viewer, that keeps record of shapes and an
associated shape, which is returned upon selection
class ProductViewer(qtViewer3d):
    def __init__(self, *args):
        qtViewer3d.__init__(self)
        self.objects = {}
```

```python
    @staticmethod
    def Hash(shape):
        return shape.HashCode(1 << 30)


    displayed_shapes = {}


    def Show(self, key, shape, color=None):
        self.objects[ProductViewer.Hash(shape)] = key
        qclr = OCC.Quantity.Quantity_Color(.35, .25, .1,
OCC.Quantity.Quantity_TOC_RGB)
        ais = self._display.DisplayColoredShape(shape, qclr)
        self.displayed_shapes[key] = ais
        self._display.FitAll()


    def Color_Repaint(self, key):
        ais = self.displayed_shapes[key]
        qclr = OCC.Quantity.Quantity_Color(.35, .25, .1,
OCC.Quantity.Quantity_TOC_RGB)
        ais.GetObject().SetColor(qclr)


    def ColorWhenRequirementIsProven(self, key):
        ais = self.displayed_shapes[key]
        qclr = OCC.Quantity.Quantity_Color(0, 0.7, 0,
OCC.Quantity.Quantity_TOC_RGB)
        ais.GetObject().SetColor(qclr)


    def ColorAsRequirementMustBeProven(self, key):
        ais = self.displayed_shapes[key]
        qclr = OCC.Quantity.Quantity_Color(1, 0, 0,
OCC.Quantity.Quantity_TOC_RGB)
        ais.GetObject().SetColor(qclr)


    def ColorAsElementHasAnUpdate(self, key):
        ais = self.displayed_shapes[key]
        qclr = OCC.Quantity.Quantity_Color(0, 0, 1,
OCC.Quantity.Quantity_TOC_RGB)
        ais.GetObject().SetColor(qclr)


    def mouseReleaseEvent(self, *args):
        qtViewer3d.mouseReleaseEvent(self, *args)
        if self._display.selected_shape:
            global guid_selection
            global selected_shape
            selected_shape = self._display.selected_shape
            guid_selection = [self.objects[ProductViewer.Hash(x)] for x in
self._display.selected_shapes]


#This class holds the locaton where the results from the dropdown menu must
be stored
class PlaceHolderStringClass:
    def __init__(self,aString):
        self.string = aString


#This class creates the dropdown requirement menu of the 'Show status of
elements for a requirement selection' button
class SelectRequirement:
    def __init__(self, master,stringClass):
        #stringClass is an instance of PlaceHolderStringClass
        self.StringClassInstance = stringClass
```

```python
        self.root = master
        self.label1 = Label(master, text="Select requirement:")
        self.label1.grid(row=0, column=0,sticky=E+W)

        # Requirement enumeration for the dropdown menu
        choices = {'A1.1', 'A1.2', 'A2.1', 'A2.2', 'F1.1', 'F1.2', 'F1.3',
'F2.1', 'F2.2', 'F2.3', 'O1.1', 'O1.2', 'S1.1', 'S1.2', 'S1.3', 'S2.1',
'S2.2'}
        self.selection = StringVar(master)
        self.dropDown = OptionMenu(master,self.selection,*choices)
        self.selection.set('A1.1')
        self.dropDown.grid(row=0, column=1,sticky=E+W)

        # Ok and Cancel button of the dropdown menu
        self.buttonOK = Button(master, text="OK", command=self.Select)
        self.buttonOK.grid(row=1, column=0,sticky=E+W)
        self.buttonCancel = Button(master, text="Cancel", command=
self.Close)
        self.buttonCancel.grid(row=1, column=1,sticky=E+W)

    # Function adopting the input from the dropdown menu
    def Select(self):
        self.StringClassInstance.string = self.selection.get()
        self.root.destroy()
    def Close(self):
        self.StringClassInstance.string = ""
        self.root.destroy()

# Class creating the dialogue box for attaching a document
class DocumentAttachment(tk.Tk):

    def open_file(self):
        global file_path
        filename = askopenfilename()
        file_path = os.path._getfullpathname(filename)
        return file_path

    def __init__(self, dateandtime, requirementList, resolvedRequirements):
        tk.Tk.__init__(self)

        # Document reference
        self.label1 = tk.Label(self, text="Document reference", font="Arial
12")
        self.label1.grid(row=0, column=0, sticky=E + W)

        # Attached document
        self.label2 = tk.Label(self, text="Attach a document: ")
        self.label2.grid(row=1, column=0, sticky=W)

        self.entry1 = tk.Entry(self)
        self.entry1.grid(row=1, column=1, sticky=W)

        # Browse button
        browse_button = tk.Button(self, text="Browse",
command=self.select_file)
        browse_button.grid(row=1, column=2, sticky=W+E)

        # Document type
        self.label3 = tk.Label(self, text="Document type: ")
        self.label3.grid(row=2, column=0, sticky=W)
```

```python
        self.entry2 = tk.Entry(self)
        self.entry2.grid(row=2, column=1, sticky=W)

        # Document fragment
        self.label4 = tk.Label(self, text="Document fragment: ")
        self.label4.grid(row=3, column=0, sticky=W)

        self.entry3 = tk.Entry(self)
        self.entry3.grid(row=3, column=1, sticky=W)

        # Date
        self.label5 = tk.Label(self, text="Date: ")
        self.label5.grid(row=4, column=0, sticky=W)

        self.filedate = tk.StringVar()
        self.filedate.set(dateandtime)

        self.entry4 = tk.Entry(self, textvariable=self.filedate)
        self.entry4.grid(row=4, column=1, sticky=W)

        # Document creator
        self.label6 = tk.Label(self, text="Document creator: ")
        self.label6.grid(row=5, column=0, sticky=W)

        self.entry5 = tk.Entry(self)
        self.entry5.grid(row=5, column=1, sticky=W)

        # Requirements proven by the document
        self.label7 = tk.Label(self, text="Requirements: ")
        self.label7.grid(row=6, column=0, sticky=W)

        #Remove elements that are already solved in at least one of the
selected elements/guids in the 3D viewer
        a = sorted(list(set(requirementList.split("\r\n"))-
set(resolvedRequirements.split("\r\n"))))
        rown=6
        self.checkbuttonList = []
        self.checkVariables = []
        self.requirementString = []

        # Adopting the open requirements for the element selection as check
boxes in the document attachment dialogue box
        for item in a:
            if (item != "Open requirements:") and (item != ""):
                self.checkVariables.append(IntVar(value=0))
                self.checkbuttonList.append(Checkbutton(self,
text=(item.split(",")[0]).strip(), variable=self.checkVariables[-1]))
                self.checkbuttonList[-1].grid(row=rown, column=1, sticky=W)
                rown += 1

        #Close button
        close_button = tk.Button(self, text="Confirm and close",
command=self.close)
        close_button.grid(row=rown, column=2, sticky=W+E)

    def select_file(self):
        fp = self.open_file()
        self.entry1.delete(0, "end")
        self.entry1.insert(0, fp)

    def close(self):
```

```python
        # All entry values from the dialogue box are added to a list
        self.result = result = []
        result.append(self.entry1.get())
        result.append(self.entry2.get())
        result.append(self.entry3.get())
        result.append(self.entry4.get())
        result.append(self.entry5.get())

        # The entry values from the checkboxes in the dialogue box are also
appended
        for idx,c in enumerate(self.checkbuttonList):
            if self.checkVariables[idx].get():
                self.result.append(c.cget("text"))

        self.destroy()

    def mainloop(self):
        tk.Tk.mainloop(self)
        return self.result

    def getCheckList(self):
        return self.checkbuttonList

# Class creating the dialogue box for attaching document updates and change
requests
class DocumentUpdate(tk.Tk):

    def open_file(self):
        global file_path
        filename = askopenfilename()
        file_path = os.path._getfullpathname(filename)
        return file_path

    def __init__(self, dateandtime, getReferencedDocuments):
        tk.Tk.__init__(self)

        # Document reference
        self.label1 = tk.Label(self, text="Document update", font="Arial
12")
        self.label1.grid(row=0, column=0, sticky=E + W)

        # Attached document
        self.label2 = tk.Label(self, text="New document version: ")
        self.label2.grid(row=1, column=0, sticky=W)

        self.entry1 = tk.Entry(self)
        self.entry1.grid(row=1, column=1, sticky=W)

        # Browse button
        browse_button1 = tk.Button(self, text="Browse",
command=self.select_file)
        browse_button1.grid(row=1, column=2, sticky=W + E)

        # Date
        self.label3 = tk.Label(self, text="Date: ")
        self.label3.grid(row=2, column=0, sticky=W)

        self.filedate = tk.StringVar()
        self.filedate.set(dateandtime)

        self.entry2 = tk.Entry(self, textvariable=self.filedate)
```

```python
        self.entry2.grid(row=2, column=1, sticky=W)

        # Document creator
        self.label4 = tk.Label(self, text="Document creator: ")
        self.label4.grid(row=3, column=0, sticky=W)

        self.entry3 = tk.Entry(self)
        self.entry3.grid(row=3, column=1, sticky=W)

        # Change request
        self.label5 = tk.Label(self, text="Change request: ")
        self.label5.grid(row=4, column=0, sticky=W)

        self.entry4 = tk.Entry(self)
        self.entry4.grid(row=4, column=1, sticky=W)

        # Browse button
        browse_button2 = tk.Button(self, text="Browse",
command=self.select_change_request_file)
        browse_button2.grid(row=4, column=2, sticky=W + E)

        # Change requiest party
        self.label6 = tk.Label(self, text="Change request initiator: ")
        self.label6.grid(row=5, column=0, sticky=W)

        self.entry5 = tk.Entry(self)
        self.entry5.grid(row=5, column=1, sticky=W)

        # Requirements proven by the document
        self.label7 = tk.Label(self, text="Update of: ")
        self.label7.grid(row=6, column=0, sticky=W)

        rown=6
        self.getReferencedDocuments = []
        self.checkbuttonList = []
        self.checkVariables = []

        # Adopting the resources (document references) already attached to
the element selection as check boxes in the document update dialogue box
        for line in getReferencedDocuments.split("\r\n"):
            if (line != "") and (line != "Resource"):
                self.checkVariables.append(IntVar(value=0))
                self.checkbuttonList.append(Checkbutton(self, text=line,
variable=self.checkVariables[-1]))
                self.checkbuttonList[-1].grid(row=rown, column=1, sticky=W)
                rown += 1

        #Close button
        close_button = tk.Button(self, text="Confirm and close",
command=self.close)
        close_button.grid(row=rown, column=2, sticky=W+E)

    def select_file(self):
        fp = self.open_file()
        self.entry1.delete(0, "end")
        self.entry1.insert(0, fp)

    def select_change_request_file(self):
        fp = self.open_file()
        self.entry4.delete(0, "end")
        self.entry4.insert(0, fp)
```

```python
    def close(self):
        # All entry values are added to a list
        self.result = result = []
        result.append(self.entry1.get())
        result.append(self.entry2.get())
        result.append(self.entry3.get())
        result.append(self.entry4.get())
        result.append(self.entry5.get())

        # All entry values from the checkboxes are also appended
        for idx,c in enumerate(self.checkbuttonList):
            if self.checkVariables[idx].get():
                self.result.append(c.cget("text"))

        # close UI
        self.destroy()

    def mainloop(self):
        tk.Tk.mainloop(self)
        return self.result

    def getCheckList(self):
        return self.checkbuttonList

# Main class of the application
class initUI(object):
    def __init__(self, *args):
        # Constructing an application
        app = QtGui.QApplication(sys.argv)

        # Viewer initialization
        self.main = Main(self)
        self.main.show()
        self.main.canvas.InitDriver()
        self.main.statusBar()
        self.display = self.main.canvas._display

        # Methods to feed the viewer with content
        self.geometry_box()
        self.geometry_grid()

        # Raise a system exit
        sys.exit(app.exec_())

    def geometry_box(self):
        box = BRepPrimAPI_MakeBox(10., 10., 10.).Shape()
        self.display.DisplayShape(box)
        self.display.FitAll()

    def geometry_grid(self):
        ax3 = gp_Ax3(gp_Pnt(0, 0, 0), gp_Dir(0, 0, 1))
        self.display.GetViewer().GetObject().SetPrivilegedPlane(ax3)
        self.display.GetViewer().GetObject().SetRectangularGridValues(0, 0,
10, 10, 0)

self.display.GetViewer().GetObject().SetRectangularGridGraphicValues(10,
10, 0)

self.display.GetViewer().GetObject().ActivateGrid(Aspect_GT_Rectangular,
Aspect_GDM_Lines)
```

```python
        self.display.FitAll()


# Main class of the Graphical User Interface
class Main(QtGui.QMainWindow):
    def __init__(self, parent=None):
        self.parent = parent
        QtGui.QMainWindow.__init__(self)

        # Instantiating the tab
        global filename
        self.filename = None
        self.repo = None

        self.tabs = QtGui.QTabWidget()
        self.setCentralWidget(self.tabs)

        self.viewer_tab = QtGui.QWidget()
        self.tabs.addTab(self.viewer_tab, "3D Requirements Viewer")

        self.setGeometry(100, 100, 850, 550)

        # Implementing the OCC viewer
        self.canvas = ProductViewer(self)

        # Calling the tab
        self.tab_3dview()


    # --------tab 1-------------------------------------------------
    ---------------
    def tab_3dview(self):
        # Initializing a split-view layout
        self.propertybox = QtGui.QTextBrowser()
        font = QtGui.QFont("Arial", 10, QtGui.QFont.Bold, True)
        sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed,
QtGui.QSizePolicy.MinimumExpanding)
        sizePolicy1 = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed,
QtGui.QSizePolicy.Ignored)

        # Property box for the IFc properties of the elemnt selection
        self.propertybox.setFont(font)
        self.propertybox.horizontalScrollBar().setValue(0)
        self.propertybox.setLineWrapMode(0)
        self.propertybox.setSizePolicy(sizePolicy)

        # Property box for the proven requirements of the element selection
        self.propertybox3 = QtGui.QTextBrowser()
        self.propertybox3.setFont(font)
        self.propertybox3.horizontalScrollBar().setValue(0)
        self.propertybox3.setLineWrapMode(0)
        self.propertybox3.setSizePolicy(sizePolicy)

        # Property box for the open requirements of the element selection
        self.propertybox2 = QtGui.QTextBrowser()
        self.propertybox2.setFont(font)
        self.propertybox2.horizontalScrollBar().setValue(0)
        self.propertybox2.setLineWrapMode(0)
        self.propertybox2.setSizePolicy(sizePolicy)
```

```python
        # Property box for requirements on which the element selection has
been updated
        self.propertybox4 = QtGui.QTextBrowser()
        self.propertybox4.setFont(font)
        self.propertybox4.horizontalScrollBar().setValue(0)
        self.propertybox4.setLineWrapMode(0)
        self.propertybox4.resize(self.propertybox4.width(), 20);

        # Define a widget for the 3D viewer
        center = QtGui.QWidget()

        # Define and set layout
        mainLayout = QtGui.QHBoxLayout(center)
        viewer_hbox = QtGui.QHBoxLayout()
        viewer_vbox = QtGui.QVBoxLayout()

        # Define all buttons in the layout
        viewer_open_ifc_btn = QtGui.QPushButton("Open IFC", self)
        viewer_open_ifc_btn.clicked.connect(self.open_ifc_file)
        viewer_open_rdf_btn = QtGui.QPushButton("Select a repository",
self)
        viewer_open_rdf_btn.clicked.connect(self.open_rdf_data)
        viewer_show_prop_btn = QtGui.QPushButton("Show properties", self)

viewer_show_prop_btn.clicked.connect(self.viewer_get_property_by_GUID)
        viewer_attach_calculation_btn = QtGui.QPushButton("Attach a
document", self)

viewer_attach_calculation_btn.clicked.connect(self.attach_calculation_file)
        viewer_show_objects_proven_by_documents_btn =
QtGui.QPushButton("Show status of elements for a requirement selection",
self)

viewer_show_objects_proven_by_documents_btn.clicked.connect(self.show_objec
ts_proven_by_documents)
        viewer_update_document_btn = QtGui.QPushButton("Update a document",
self)
        viewer_update_document_btn.clicked.connect(self.update_document)
        viewer_show_objects_with_updated_document_btn =
QtGui.QPushButton("Show elements with updated proofs", self)

viewer_show_objects_with_updated_document_btn.clicked.connect(self.show_obj
ects_with_document_updates)
        viewer_show_update_of_requirement_btn = QtGui.QPushButton("Show
requirements with updated proofs", self)

viewer_show_update_of_requirement_btn.clicked.connect(self.show_updated_req
uirements_for_object_with_document_updates)

        # Define the position of all buttons and property boxes in the
layout
        splitter = QtGui.QSplitter(QtCore.Qt.Horizontal)
        splitterH = QtGui.QSplitter(QtCore.Qt.Vertical)
        splitter.addWidget(self.canvas)
        splitter.addWidget(splitterH)
        splitterH.addWidget(viewer_update_document_btn)
        splitterH.addWidget(viewer_show_objects_with_updated_document_btn)
        splitterH.addWidget(viewer_show_update_of_requirement_btn)
        splitterH.addWidget(self.propertybox4)
        splitterH.addWidget(self.propertybox)
        splitterH.addWidget(self.propertybox3)
```

```python
        splitterH.addWidget(self.propertybox2)

        viewer_vbox.addWidget(splitter)
        viewer_vbox.addLayout(viewer_hbox)
        self.viewer_tab.setLayout(viewer_vbox)
        viewer_hbox.addWidget(viewer_open_ifc_btn)
        viewer_hbox.addWidget(viewer_open_rdf_btn)
        viewer_hbox.addWidget(viewer_show_prop_btn)
        viewer_hbox.addWidget(viewer_attach_calculation_btn)
        viewer_hbox.addWidget(viewer_show_objects_proven_by_documents_btn)

        self.count = 0

    # Function for adopting information into the property boxes
    def viewer_get_property_by_GUID(self):
        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                      "No IFC file loaded!",
                                      "Please, load a model first!")
            return
        self.propertybox.clear()
        self.guid_to_prop_dict()

        if guid_selection == None:
            QtGui.QMessageBox.warning(self, "Select element first!")

            return
        self.propertybox.clear()
        self.guid_to_prop_dict()

        for element in self.ifc_file.by_type("IfcProduct"):
            if element.is_a("IfcBuildingElement"):
                self.canvas.Color_Repaint(element.GlobalId)

        # Append IFC properties to property box for GUID selection
        for x in guid_selection:
            for category, property_or_material in self.guid_to_prop[x]:
                if category == 'prop':
                    if property_or_material.NominalValue.wrappedValue ==
'':
                        continue
                    else:
                        self.propertybox.append("%s : %s"
%(property_or_material.Name,
property_or_material.NominalValue.wrappedValue))
                elif category == 'material':
                    self.propertybox.append("%s : %s" % ("Material",
property_or_material))

        if not self.repo:
            QtGui.QMessageBox.warning(self,
                                      "No repository selected!",
                                      "Please, select a repository
first!")
            return
        self.propertybox2.clear()
        self.propertybox3.clear()

        # Append RDF properties (requirements) to property box for GUID
selection
        self.requirementList = ""
```

```python
        # Append RDF properties (open requirements) to property box for
GUID selection
        for x in guid_selection:
            aux = str(Query1.get_requirement(x, self.repo))
            #self.requirementList += str(Query1.get_requirement(x,
self.repo))
            if len(aux) <= 25:
                self.requirementList +="" + "\r\n" + "None"+ "\r\n"
            else:
                self.requirementList +="" + "\r\n" +aux

        self.resolvedRequirement = ""
        # Append RDF properties (proven requirements) to property box for
GUID selection
        for x in guid_selection:
            aux2= str(Query2.get_open_requirement(x, self.repo))
            if len(aux2) <= 25:
                self.resolvedRequirement +="" + "\r\n" + "None"+ "\r\n"
            else:
                self.resolvedRequirement +="" + "\r\n"+aux2

        # Append the intersection between proven and open requirements for
all selected GUIDS
        a = sorted(list(set(self.requirementList.split("\r\n")) -
set(self.resolvedRequirement.split("\r\n"))))
        self.propertybox2.append("Open requirements:\r\n")
        for item in a:
            self.propertybox2.append(item)
        b = sorted(list(set(self.resolvedRequirement.split("\r\n")) -
set(self.requirementList.split("\r\n"))))
        self.propertybox3.append("Proven requirements:\r\n")
        for item in b:
            self.propertybox3.append(item)

    def guid_to_prop_dict(self):
        # Append IFC properties of elements to dictionary
        self.guid_to_prop = defaultdict(list)
        for elem in self.ifc_file.by_type("IfcBuildingElement"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties") and
rel.RelatingPropertyDefinition.is_a("IfcPropertySet"):
                    for prop in
rel.RelatingPropertyDefinition.HasProperties:
                        self.guid_to_prop[elem.GlobalId].append(('prop',
prop))
            for rel in elem.HasAssociations:
                if rel.is_a("IfcRelAssociatesMaterial"):
                    if rel.RelatingMaterial.is_a("IfcMaterial"):
                        for material in rel.RelatingMaterial:

self.guid_to_prop[elem.GlobalId].append(('material', material))

    # Main function behind the 'Attach calculation' button
    def attach_calculation_file(self, filename=None):
        dateandtime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        if guid_selection == None:
            QtGui.QMessageBox.warning(self, "No element selected!",
                                      "Select an element first!")

            return
```

```python
        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                "No document attached!",
                                "Please, attach a document first!")
            return

        # Establishing connection with the FTP server for uploading the
attached documentation
        def ftp_filelocation():
            localfile = result[0]
            if localfile == "":
                print "No file has been attached"
            else:
                host = 'sparql.verhoeven-leenders.nl'
                #host = '192.168.5.2'
                username = 'Miryana'
                password = '05091991'
                remotefile_directory = '/Requirement Proofs/Document
references'

                ftp = FTP()
                ftp.set_debuglevel(2)
                ftp.connect(host, 2121)
                ftp.login(username, password)
                ftp.cwd(remotefile_directory)

                fp = open(localfile, 'rb')
                ftp.storbinary('STOR %s' % os.path.basename(localfile), fp,
2121)
                fp.close()
                print localfile + " has been stored in " +
remotefile_directory

                reffile = os.path.basename(str(file_path))
                ftpfile_directory = 'ftp://Miryana@sparql.verhoeven-
leenders.nl:2121/Requirement proofs/Document references/' + str(reffile)
                # ftpfile_directory =
'ftp://Miryana@192.168.5.2:2121/Requirement document references/' +
str(reffile)
                ftp.quit()
                return ftpfile_directory

        # Creating RDF data from the input of the attach a document
dialogue box
        def RDF_graph_ReferenceDocument():

            g = Graph()

            greq =
Namespace("http://example.org/ontology/generalrequirements#")
            namespace_manager = NamespaceManager(g)
            namespace_manager.bind('greq', greq)

            refers_to = URIRef(greq.refersTo)

            # RDF classes
            document_reference = URIRef(result[0])
            element = Literal(guid_selection)

            # RDF DocumentReference properties
            hasFTPlocation = URIRef(greq.hasFTPLocation)
```

```python
            hasdocumenttype = URIRef(greq.hasDocumentReferenceType)
            hasdocumentfragment = URIRef(greq.hasDocumentReferenceFragment)
            hasdocumentdate = URIRef(greq.hasDocumentReferenceDate)
            hasdocumentcreator = URIRef(greq.hasDocumentReferenceCreator)
            proves_requirement = URIRef(greq.provesRequirement)

            for element in guid_selection:
                g.add((document_reference, refers_to, Literal(element)))

            # RDF DocumentReference triples
            g.add((document_reference, RDF.type, greq.DocumentReference))
            g.add((document_reference, hasFTPlocation,
URIRef(ftp_filelocation())))
            g.add((document_reference, hasdocumenttype, Literal(result[1],
datatype=XSD.string)))
            g.add((document_reference, hasdocumentfragment,
Literal(result[2], datatype=XSD.string)))
            g.add((document_reference, hasdocumentdate, Literal(result[3],
datatype=XSD.dateTime)))
            g.add((document_reference, hasdocumentcreator,
Literal(result[4], datatype=XSD.string)))
            for c in range(5, len(result)):
                # g.add((document_reference, proves_requirement,
Literal(result[c].split(":")[0])))
                g.add((document_reference, proves_requirement,
greq[result[c].split(":")[0]]))

            rdfdata = g.serialize(format='pretty-xml')

            # Uploading data to the chosen repository
            repository = self.repo
            reffile = os.path.basename(str(file_path))

            # graph = 'file://C:/fakepath/RequirementDocumentReference.rdf'
            graph = 'file://C:/fakepath/DocumentReferences/'+ str(reffile)
+ "_" + dateandtime
            params = {'context': '<' + graph + '>'}
            endpoint = "http://sparql.verhoeven-
leenders.nl/repositories/%s/statements?%s" % (repository,
urllib.urlencode(params))
            #endpoint = "http://192.168.5.2/repositories/%s/statements?%s"
% (repository, urllib.urlencode(params))
            data = rdfdata
            (response, content) = httplib2.Http().request(endpoint, 'PUT',
body=data, headers={'content-type': 'application/rdf+xml'})
            print "Response %s" % response.status

        app =
DocumentAttachment(dateandtime,self.requirementList,self.resolvedRequiremen
t)
        result = app.mainloop()
        RDF_graph_ReferenceDocument()

    # Shows all objects proven or to be proven for a requirement selection
    def show_objects_proven_by_documents(self):
        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                "No IFC loaded!",
                                "Please load a model first!")
            return
```

```python
        if not self.repo:
            QtGui.QMessageBox.warning(self,
                                      "No repository selected!",
                                      "Please, select a repository
first!")
            return

        root = Tk()
        root.wm_title("Select Requirement")
        self.placeWhereTheResultsMustBeStored= PlaceHolderStringClass("")

        #creates a window and specifiy where the selection from the
dropdown menu must be stored (placeWhereTheResultsMustBeStored)
        window =
SelectRequirement(root,self.placeWhereTheResultsMustBeStored)
        #runs the window and waits for the user to click OK or Cancel
        root.mainloop()

        #makes a copy of the selection and puts it in self.label
        self.label = self.placeWhereTheResultsMustBeStored.string
        self.me = greq + "".join(self.label)
        sparql = SPARQLWrapper.SPARQLWrapper("http://sparql.verhoeven-
leenders.nl/repositories/" + self.repo)
        # sparql =
SPARQLWrapper.SPARQLWrapper("http://192.168.5.2/repositories/"+repo)
        sparql.setMethod(SPARQLWrapper.POST)
        sparql.setReturnFormat(RDFXML)

        # Qeury which retrieves all GUIDs with document attachments for a
requirements selection
        q1 = """
                PREFIX greq:
<http://example.org/ontology/generalrequirements#>
                PREFIX ifc: <http://www.buildingsmart-
tech.org/ifcOWL/IFC2X3_TC1#>
                PREFIX express: <http://purl.org/voc/express#>
                PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

                SELECT ?guid
                WHERE {
                ?Requirement greq:isApplicableToObject ?ifctype .
                ?Requirement greq:hasLabel <%s> .
                ?instance a ?ifctype .
                ?instance ifc:globalId_IfcRoot ?guid_id .
                ?guid id express:hasString ?guid .
                FILTER(
                    EXISTS {
                        ?Resource greq:refersTo ?guid .
                        ?Resource greq:provesRequirement <%s> .
                        }
                    )
                }""" % (self.me, self.me)

        sparql.setQuery(q1)
        results = sparql.query().convert()
        list_of_proven_elements = results.split("\r\n")

        # Qeury which retrieves all GUIDs without document attachments for
a requirements selection
        q2 = """
```

```
                    PREFIX greq:
<http://example.org/ontology/generalrequirements#>
                    PREFIX ifc: <http://www.buildingsmart-
tech.org/ifcOWL/IFC2X3_TC1#>
                    PREFIX express: <http://purl.org/voc/express#>
                    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

                    SELECT ?guid
                    WHERE {
                            ?Requirement greq:isApplicableToObject ?ifctype
.
                            ?Requirement greq:hasLabel <%s> .
                            ?instance a ?ifctype .
                            ?instance ifc:globalId_IfcRoot ?guid_id .
                            ?guid_id express:hasString ?guid .
                    FILTER(
                    !EXISTS {
                            ?Resource greq:refersTo ?guid .
                            ?Resource greq:provesRequirement <%s> .
                            }
                        )
                    }""" % (self.me, self.me)

        sparql.setQuery(q2)
        results = sparql.query().convert()
        list_of_open_elements = results.split("\r\n")

        # Elements of the retrieved GUIDs from both queries are colored in
respect to whether they are proven
        # on the requirement, to be proven on it or whether the
requirements doesn't apply to them
        for element in self.ifc_file.by_type("IfcBuildingElement"):

            if element.GlobalId in list_of_proven_elements:
                if element.Representation:

self.canvas.ColorWhenRequirementIsProven(element.GlobalId)
            elif element.GlobalId in list_of_open_elements:
                if element.Representation:

self.canvas.ColorAsRequirementMustBeProven(element.GlobalId)
            else:
                    self.canvas.Color_Repaint(element.GlobalId)

    # Main function of the 'Document update' button
    def update_document(self):
        dateandtime = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        if not self.repo:
            QtGui.QMessageBox.warning(self,
                                      "No repository selected!",
                                      "Please, select a repository
first!")
            return

        for element in self.ifc_file.by_type("IfcProduct"):
            if element.is_a("IfcBuildingElement"):
                self.canvas.Color_Repaint(element.GlobalId)

        if len(guid_selection) > 1:
            QtGui.QMessageBox.warning(self,
                                      "More than one elelement selected!",
```

```python
                                            "Please, select only one element!")
            return

        self.getReferencedDocuments =
str(Query3.get_document_proofs(guid_selection, self.repo))

        # Establishing connection with the FTP server for uploading the
change requests
        def ftp_filelocation_change_request():
            localfile = result[3]
            if localfile == "":
                print "No file has been attached"
            else:
                host = 'sparql.verhoeven-leenders.nl'
                #host = '192.168.5.2'
                username = 'Miryana'
                password = '05091991'
                remotefile_directory = '/Requirement proofs/Change
requests'

                ftp = FTP()
                ftp.set_debuglevel(2)
                ftp.connect(host, 2121)
                ftp.login(username, password)
                ftp.cwd(remotefile_directory)

                fp = open(localfile, 'rb')
                ftp.storbinary('STOR %s' % os.path.basename(localfile), fp,
2121)
                fp.close()

                reffile = os.path.basename(str(localfile))
                ftpfile_directory = 'ftp://Miryana@sparql.verhoeven-
leenders.nl:2121/Requirement proofs/Change requests/' + str(reffile)
                # ftpfile_directory =
'ftp://Miryana@192.168.5.2:2121/Requirement document references/' +
str(reffile)
                ftp.quit()
                return ftpfile_directory

        # Establishing connection with the FTP server for uploading the
document updates
        def ftp_filelocation():
            localfile = result[0]
            if localfile == "":
                print "No file has been attached"
            else:
                host = 'sparql.verhoeven-leenders.nl'
                #host = '192.168.5.2'
                username = 'Miryana'
                password = '05091991'
                remotefile_directory = '/Requirement proofs/Document
updates'

                ftp = FTP()
                ftp.set_debuglevel(2)
                ftp.connect(host, 2121)
                ftp.login(username, password)
                ftp.cwd(remotefile_directory)

                fp = open(localfile, 'rb')
```

```python
                ftp.storbinary('STOR %s' % os.path.basename(localfile), fp,
2121)
                fp.close()

                reffile = os.path.basename(str(localfile))
                ftpfile_directory = 'ftp://Miryana@sparql.verhoeven-
leenders.nl:2121/Requirement proofs/Document updates/' + str(reffile)
                # ftpfile_directory =
'ftp://Miryana@192.168.5.2:2121/Requirement document references/' +
str(reffile)
                ftp.quit()
                return ftpfile_directory

        # Creating RDF data from the input of the document update dialogue
box
        def RDF_graph_UpdateDocument():

            g = Graph()

            greq =
Namespace("http://example.org/ontology/generalrequirements#")
            namespace_manager = NamespaceManager(g)
            namespace_manager.bind('greq', greq)

            # RDF classes
            document_update = URIRef(result[0])
            change_request = URIRef(result[3])
            element= Literal(guid_selection)

            # RDF DocumentUpdate properties
            hasFTPLocation = URIRef(greq.hasFTPLocation)
            hasdocumentdate = URIRef(greq.hasDocumentUpdateDate)
            hasdocumentcreator = URIRef(greq.hasDocumentUpdateCreator)
            haschangerequest = URIRef(greq.hasChangeRequest)
            haschangerequestinitiator = URIRef(greq.hasInitiator)
            is_an_update_of = URIRef(greq.isAnUpdateOf)

            # RDF DocumentUpdate triples
            g.add((document_update, RDF.type, greq.DocumentUpdate))
            g.add((document_update, hasFTPLocation,
URIRef(ftp_filelocation())))
            g.add((document_update, hasdocumentdate, Literal(result[1],
datatype=XSD.dateTime)))
            g.add((document_update, hasdocumentcreator, Literal(result[2],
datatype=XSD.string)))
            if result[3] == "":
                g.add((document_update, haschangerequest, Literal("None")))
            else:
                g.add((document_update, haschangerequest, change_request))
                g.add((change_request, hasFTPLocation,
URIRef(ftp_filelocation_change_request())))
                g.add((change_request, haschangerequestinitiator,
Literal(result[4], datatype=XSD.string)))
            # g.add((document_reference, proves_requirement,
Literal(result[5], datatype=XSD.string)))
            for c in range(5, len(result)):
                g.add((document_update, is_an_update_of,
URIRef(result[c])))

            rdfdata = g.serialize(format='pretty-xml')
```

```python
            # Uploading data to the chosen repository
            repository = self.repo
            reffile = os.path.basename(str(result[0]))

            # graph = 'file://C:/fakepath/RequirementDocumentReference.rdf'
            graph = 'file://C:/fakepath/DocumentUpdates/'+ str(reffile) +
"_" + dateandtime

            params = {'context': '<' + graph + '>'}
            print params
            endpoint = "http://sparql.verhoeven-
leenders.nl/repositories/%s/statements?%s" % (repository,
urllib.urlencode(params))
            #endpoint = "http://192.168.5.2/repositories/%s/statements?%s"
% (repository, urllib.urlencode(params))
            data = rdfdata
            (response, content) = httplib2.Http().request(endpoint, 'PUT',
body=data, headers={'content-type': 'application/rdf+xml'})
            print "Response %s" % response.status

        app = DocumentUpdate(dateandtime, self.getReferencedDocuments)
        result = app.mainloop()
        RDF_graph_UpdateDocument()

    # Function for displaying all objects with document updates
    def show_objects_with_document_updates(self, guid):
        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                "No IFC loaded!",
                                "Please load a model first!")
            return

        if not self.repo:
            QtGui.QMessageBox.warning(self,
                                    "No repository selected!",
                                    "Please, select a repository
first!")
            return

        for element in self.ifc_file.by_type("IfcProduct"):
            if element.is_a("IfcBuildingElement"):
                self.canvas.Color_Repaint(element.GlobalId)

        sparql = SPARQLWrapper.SPARQLWrapper("http://sparql.verhoeven-
leenders.nl/repositories/" + self.repo)
        # sparql =
SPARQLWrapper.SPARQLWrapper("http://192.168.5.2/repositories/"+repo)
        sparql.setMethod(SPARQLWrapper.POST)
        sparql.setReturnFormat(RDFXML)

        # Query for retrieving all GUIDs with document attachments which
are related to document updates
        q3 = """
        PREFIX greq: <http://example.org/ontology/generalrequirements#>
        PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
        PREFIX express: <http://purl.org/voc/express#>
        PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

        SELECT ?guid
        WHERE {
            ?Update greq:isAnUpdateOf ?Resource .
```

```python
                ?Resource greq:refersTo ?guid .
        }"""

        sparql.setQuery(q3)
        results = sparql.query().convert()
        list_of_elements_with_updates = results.split("\r\n")

        # Coloring the elements with GUIDs retrieved from the q3 query
above
        for element in self.ifc_file.by_type("IfcBuildingElement"):

            if element.GlobalId in list_of_elements_with_updates:
                if element.Representation:
                    self.canvas.ColorAsElementHasAnUpdate(element.GlobalId)

    # Displaying the requirements on which the colored/selected elements
were updated and the FTP location on the document updates
    def show_updated_requirements_for_object_with_document_updates(self):

        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                      "No IFC file loaded!",
                                      "Please, load a model first!")
            return

        if not self.repo:
            QtGui.QMessageBox.warning(self,
                                          "No repository selected!",
                                          "Please, select a repository
first!")
            return

        if guid_selection == None:
            QtGui.QMessageBox.warning(self, "Select element first!")

            return
        self.propertybox4.clear()

        for x in guid_selection:
            aaa = str(Query4.get_requirements(x, self.repo))
            for line in aaa.split("\r\n"):
                if (line != "") and (line != "Label,FTP"):
                    self.propertybox4.append(line)

    # Loading an IFC file in the viewer application
    def open_ifc_file(self, filename=None):
        self.filename = QtGui.QFileDialog.getOpenFileName(self, 'Open
file', ".", "Industry Foundation Classes (*.ifc)")
        if self.filename:
            self.parent.display.EraseAll()
            self.propertybox.clear()
        self.parse_ifc(self.filename)

    # Selecting a triplestore repository
    def open_rdf_data(self):
        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                      "No IFC file loaded!",
                                      "Please, load a model first!")
            return
```

```python
        self.repo = str(QtGui.QInputDialog.getText(self, "Choose a
repository", "Repository ID:")[0])

        return


    # Parsing the IFC file loaded in the viewer application by component
type
    def parse_ifc(self, filename):
        self.created_shapes = {}
        self.ifc_file = ifcopenshell.open(filename)
        elements = self.ifc_file.by_type("IfcProduct")
        for element in elements:
            if element.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings,
element).geometry
                shp = self.canvas.Show(element.GlobalId, ifcgeom, None)
                self.created_shapes[element.GlobalId] = shp
        print "IFC file successfully loaded!"


    # Closing the viewer application
    def closeEvent(self, event):
        closevariable = QtGui.QMessageBox.question(self,
                                    "Confirm Exit",
                                    "Are you sure you want to exit
?",
                                    QtGui.QMessageBox.Yes |
QtGui.QMessageBox.No)
        event.ignore()

        if closevariable == QtGui.QMessageBox.Yes:
            event.accept()

init = initUI()
```

Appendix X: Python script - Additional queries

```python
import SPARQLWrapper
from SPARQLWrapper import RDFXML

# Query which retrieves all open requirements for an element selection
def get_requirement(guid,repo):


    q = """
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Label ?Requirement ?Type ?Description
WHERE {
    ?Requirement greq:isApplicableToObject ?ifctype .
    ?Requirement a ?Type .
    ?instance a ?ifctype .
    ?Requirement greq:hasLabel ?Label .
    ?Requirement rdfs:comment ?Description.
    ?instance ifc:globalId_IfcRoot ?guid_id .
    ?guid_id express:hasString "%s" .

FILTER(
    NOT EXISTS {
    ?Resource greq:refersTo "%s" .
    ?Resource greq:provesRequirement ?Label .
}
)
}""" % (guid,guid)

    sparql = SPARQLWrapper.SPARQLWrapper("http://sparql.verhoeven-
leenders.nl/repositories/"+repo)
    sparql.setQuery(q)
    sparql.setMethod(SPARQLWrapper.POST)
    sparql.setReturnFormat(RDFXML)
    ret = ""
    try:
        results = sparql.query().convert()
    except:
        return "There was an error with the selection of repository: "+repo


    lines = results.split("\r\n")

    for requirement in lines:
        #only consider actual requirements
            if requirement.find("#") != -1:
            #split using the # sign
                tokens = requirement.split("#")
                ret = ret +
tokens[0].replace("http://example.org/ontology/generalrequirements","") \
                + tokens[1].split(",")[0] + ": " +tokens[2].split(",")[0] +
", Type: " + tokens[3]+ "\r\n"

    return  ret
import SPARQLWrapper
```

```python
from SPARQLWrapper import RDFXML

# Query which retrieves all proven requirements for an element selection
and the FTP location of the document proofs
def get_open_requirement(guid,repo):


    q = """
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Label ?Requirement ?Type ?FTP
WHERE {
    ?Requirement greq:isApplicableToObject ?ifctype .
    ?Requirement a ?Type .
    ?instance a ?ifctype .
    ?Requirement greq:hasLabel ?Label .
    ?instance ifc:globalId_IfcRoot ?guid_id .
    ?guid_id express:hasString "%s" .
    ?Resource greq:hasFTPLocation ?FTP .
FILTER(
    EXISTS {
    ?Resource greq:refersTo "%s" .
    ?Resource greq:provesRequirement ?Label .
}
)

}""" % (guid,guid)

    sparql = SPARQLWrapper.SPARQLWrapper("http://sparql.verhoeven-
leenders.nl/repositories/"+repo)
    sparql.setQuery(q)
    sparql.setMethod(SPARQLWrapper.POST)
    sparql.setReturnFormat(RDFXML)
    ret = ""
    try:
        results = sparql.query().convert()

    except:
            return "There was an error with the selection of repository:
"+repo

    lines = results.split("\r\n")
    for requirement in lines:
        #only consider actual requirements
        if requirement.find("#") != -1:
            #split using the # sign
            tokens = requirement.split("#")
            ret = ret +
tokens[0].replace("http://example.org/ontology/generalrequirements", "") \
                + tokens[1].split(",")[0] + ": " +
tokens[2].split(",")[0] + ", Type: " \
                + tokens[3].replace(",","", Document: ") + "\r\n"

    return  ret
```

```python
import SPARQLWrapper
from SPARQLWrapper import RDFXML

# Query which retrieves document attachments for a selected element
def get_document_proofs(guid, repo):

    aux = str(guid).replace("[u'", "").replace("']", "")
    q = """
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Resource
WHERE {
    ?instance ifc:globalId_IfcRoot ?guid_id .
    ?guid_id express:hasString "%s" .
    ?Resource greq:refersTo "%s" .

}"""% (aux, aux)
    sparql = SPARQLWrapper.SPARQLWrapper("http://sparql.verhoeven-
leenders.nl/repositories/"+repo)
    #sparql =
SPARQLWrapper.SPARQLWrapper("http://192.168.5.2/repositories/"+repo)
    sparql.setQuery(q)
    sparql.setMethod(SPARQLWrapper.POST)
    sparql.setReturnFormat(RDFXML)

    results = sparql.query().convert()

    return results
```

```python
import SPARQLWrapper
from SPARQLWrapper import RDFXML

# Query which retrieves the requirement IDs or labels on which an element
selection has been updated together with the FTP location of the document
update
def get_requirements(guid, repo):
    aux = str(guid).replace("[u'", "").replace("']", "")
    q = """
PREFIX greq: <http://example.org/ontology/generalrequirements#>
PREFIX ifc: <http://www.buildingsmart-tech.org/ifcOWL/IFC2X3_TC1#>
PREFIX express: <http://purl.org/voc/express#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Label ?FTP
WHERE {

    ?Resource greq:refersTo "%s" .
    ?Resource greq:provesRequirement ?Label .
    ?Requirement greq:hasLabel ?Label.
    ?Update greq:hasFTPLocation ?FTP .

 FILTER(
    EXISTS {
    ?Update greq:isAnUpdateOf ?Resource .

}
)
}"""% (aux)
    sparql = SPARQLWrapper.SPARQLWrapper("http://sparql.verhoeven-
leenders.nl/repositories/"+repo)
    sparql.setQuery(q)
    sparql.setMethod(SPARQLWrapper.POST)
    sparql.setReturnFormat(RDFXML)

    ret = ""
    results = sparql.query().convert()

    lines = results.split("\r\n")

    for requirement in lines:
            #only consider actual requirements
        if requirement.find("#") != -1:
                #split using the # sign
            tokens = requirement.split("#")
            ret = ret + tokens[0].split(",")[0]+ ":
"+tokens[1].split(",")[0]+", Type: "\
                +tokens[2].replace(",",", Document: ")+ "\r\n"

    return  ret
```

Appendix XI: Flowchart: Check requirements and attach a document proof

Appendix XII: Flowchart: Attach a document update and a change request

```
                        ┌──────────────┐
                        │    Start     │
                        └──────┬───────┘
                               │
                               ▼
  ╱────────────╲       ┌──────────────┐
 ╱ Load IFC     ╲─────▶│ Parse IFC    │
╱   model        ╲     │ model with   │
╲                ╱     │ PythonOCC    │
 ╲──────────────╱      └──────┬───────┘
                               │
                               ▼
                        ┌──────────────┐
                        │ Enter        │
                        │ repository ID│
                        │ to connect to│
                        │ the          │
                        │ triplestore  │
                        └──────┬───────┘
                               │
                               ▼
                        ┌──────────────┐
                        │ Element(s)   │
                        │ selection    │
                        │ (GUID of     │
                        │ selection is │
                        │ set as global│
                        │ variable)    │
                        └──────────────┘
```

Start

Load IFC model

Parse IFC model with PythonOCC

Enter repository ID to connect to the triplestore

Element(s) selection (GUID of selection is set as global variable)

Query RDF data for document proofs related to the selected element

New context in the triplestore

Document update uploaded to the file server

Update a document proof based and attach a change request

Append query results in Tkinter frame widget as checkboxes

Change request uploaded to the file server

Color elements with updated document proofs

Query RDF data for elements with document updates

End

Appendix XIII: Qualitative research: Summarized interview transcripts

20.04.2017	Hendriks B&O	Joost van de Koppel	BIM Manager

General

1. Can you describe the core business of your organization?
Hendriks Bouw en Ontwikkeling is a company primarily involved in the construction and development of real estate such as residential projects, office buildings and healthcare facilities. We also sometimes design the entire building including the MEP facilities and operate and maintain the project afterwards.

2. What is your position in the organization and what are your main roles and responsibilities?
I am a BIM manager responsible for everything related to BIM such as assisting various project teams and improving the BIM processes of the company, keeping track of innovations in BIM, overseeing the contact arrangements with suppliers and sub-contractors to improve collaboration practices.

3. With what type of organizations do you most often cooperate?
Usually we cooperate with external clients such as residential cooperations, also with users/buyers when we develop our own real estate, sub-contractors and suppliers.

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?
Yes, we use BIM both in the office and on site. BIM is primarily implemented as a tool to optimize the design and construction processes and therefore, to reduce the costs of a project. Besides that, we implement it in order to also be innovative in general, to improve the image of the company, and to be an interesting employer. But again, the most important purpose for us is using it as a tool to optimize the building/core process in general.

5. Which tangible benefits have you experienced through the implementation of BIM?
The benefits for us consist in having a tool which can perform validation and clash detection but is also helpful in coordinating the different stakeholders for quantity takeoff and further, to reduce time for the site manager. There are also benefits which BIM brings in the calculation phase to determine quantities and price.

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?
What is always difficult are the many stakeholders involved in a project and the fact that everyone needs to be included in the process but it is really difficult to be on the same level of understanding about BIM. There are still many stakeholders are not BIM ready. The interoperability between software applications is not that big of an issue anymore as we have the IFC standard for the information exchange. With suppliers, however, sometimes working with IFC models isn't a very good approach as the software they use is not very compatible with IFC and information can get lost.

Information management (requirements management)

7. How is documentation handled in your organization currently and how much project information is still stored in paper form?

We handle about 70-80% of the documentation in the old-fashioned traditional way - through PDFs, we also use Excel and try to integrate only the important information in BIM. The engineering is what is done through BIM and the communication part is being currently done by PDFs but also by BCFs, however I wouldn't say that there is a consistent or uniform way for handling information as it all still depends on the person. Towards the end of a project for information that is not in the BIM model such as finishings, there is almost no use of the model in the communication between stakeholders. 2D drawings are not used as often in the office but are used extensively by the builders and this isn't a bad thing because the content counts in this case more.

8. How do you handle/systematize building requirements?
We don't use a platform for that and use only the initial architectural soft requirements defined by the client and stored as a PDF. We have, however, our own in-house general requirements on the BIM and for that we follow the Hendrix IDM (Information Delivery Manual) which is also a must for the subcontractors we are working with and everyone involved in a process should be aware of it.

9. In which manner do you ensure that requirement information remains up-to-date?
-

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
Definitely, in the construction phase because that is where the design meets the production information. That is where information is most scattered into different documents which also includes the requirements.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
Yes, because several graduation students conducted similar studies in our company. If you can relate information to objects, it becomes way clearer than relating just a requirement to an object. Also the relationship between the requirements and the objects is well defined for other, external stakeholders.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
The fire safety requirements and the calculations regarding ventilation and insulation, the advisory reports made by external parties such as the sound reports, structural loads and reactions. Basically, all stand-alone PDFs that don't directly connect to anything but are from a great importance for delivering the right quality of the project.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
For smaller residential projects the architect also models the structural part but doesn't want to take the responsibility for making mistakes, so the structural engineer goes into the architectural model to correct his part. If we can attach documentation to the design, it will be way clearer to both parties what the work of the other party represents. Therefore, probably LOD300 as there the most project data is usually generated which also constitutes the input for the construction phase is most suitable.

07.03.2017	ProcessMinded	Chantal Laurijssens	SE process manager
07.03.2017	ProcessMinded	Adrian Dinca	SE & Relatics engineer

General

1. Can you describe the core business of your organization?
We are a consultancy which specializes in the design, implementation and optimization of processes in building projects through Systems Engineering.

2. What is your position in the organization and what are your main roles and responsibilities?
We are both Systems Engineers.

3. With what type of organizations do you most often cooperate?
Generally, we collaborate with contractors on projects related to both the civil and building structures sector but Systems Engineering is usually implemented in civil projects more than in building structures projects.

Information management (requirements management)

7. How is documentation handled in your organization and in the organizations that you currently work with on projects and how much project information is still stored in paper form?
The companies that we work with are mostly digitalized they use VISI which is a document management system with authorization and workflow integrated into it for formal communication and the client doesn't want to have hard copies anymore, however, documentation is indeed still in PDF format or Excel format and is sometimes but not always connected to the information management systems such as Relatics.

8. How do you handle/systematize building requirements for clients?
Everything is done in Relatics and it is important to mention that civil and building structures projects are very different in the management of requirements. In civil projects the client – usually a governmental institution demands the use of Systems Engineering, in building structures projects, however, that is not the case. The processes and contracts within this domain are also different from the one is the civil domain where projects follow stricter policies.

9. In which manner do you ensure that requirement information remains up-to-date?
Firstly, we analyze the contract documents and extract the requirements from there. Afterwards, we send them to the client for validation and approval so that we know whether we can build our process on them. Then we go to Relatics, we create the SBS of a project and the work packages and then when we verify the requirements, we make the link to the document management system in order to verify them through documentation.

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
At the very beginning, right after the contract is signed is the most appropriate time to start with requirement specification. Every stage can be negatively impacted and when starting at the beginning, it can save time in later stages, From the final design or LOD 300 onwards, the systematization of requirements is really hard due to the fact that they need to be deducted from previous phases. In order to prevent difficulties, therefore, it is important to adapt to the process of the designers early on. Also the design needs to be validated from the contractor's side in order to assure the buildability of the design.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
It could be a very useful feature because so far Relatics doesn't properly relate to the geometrical design and the geometrical design, or BIM model, doesn't relate to the requirements specified in Relatics or to the documentation from VISI. If you have a contract with COINS, it is almost necessary because you need to link requirements information and model information all in one container.

For building structures projects we extract the information from the 3D model and import it in Relatics. The elements are, however, extracted as an Excel sheet with all references and put it into Relatics, then there is a lot of manual work in relating requirements to the object components. And when COINS comes into place, then yet another relationship component is being added which needs to be related to everything else. The COINS OTL is imported again as Excel in Relatics and needs to be associated to the information already contained on the platform.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
All of them.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
Both the general and the very detailed requirements need to be associated to the model, which implies that all of the aforementioned, with the requirements derived from them need to be connected on all level of detail. The systems engineering SBS and the 3D modeling SBS are not always the same but need to be linked throughout the entire design and construction process to keep the consistency of data.

04.04.2017	Royal Haskoning DHV	Jan-Henk Oldenburg	BIM advisor/coordinator

General

1. Can you describe the core business of your organization?
We are a civil engineering company which works in the domain of rail, waterworks, transport but also building structures.

2. What is your position in the organization and what are your main roles and responsibilities?
I am a BIM advisor and for specific projects I'm the BIM coordinator.

3. With what type of organizations do you most often cooperate?
We are the main contractor so we cooperate with both designers and sub-contractors. We are also often involved as a contractor in the early stages of a project such as the sketching/concept phase and preliminary design.

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?
Yes, we implement BIM. Sometimes it is asked by the client for keeping track of the project's progress and as a means to validate design and detect clashes between the different disciplines. It is more or less a validation tool for us but we also show it to the client in order to show the advantages of using it. The client usually gets only the final model for maintenance and operations purposes and the documentation generated throughout the process; there is, however, no history or traceability between these two aspects.

5. Which tangible benefits have you experienced through the implementation of BIM?
Reducing risks of really costly mistakes during the construction phase is the biggest benefit we get from BIM. The design can be easily validated and potentially risky and costly problems can be prevented and solved before the execution phase and thus, save us and the client both money and time. In the preliminary and final design phases using BIM is more expensive but then money are saved in the construction phase and this is the biggest benefit which we try to convey to our clients. We want to basically show them the value of using a properly constructed BIM.

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?
BIM is a complex thing to understand and there's a lot of people working on projects that don't know what BIM actually is and if you have a team that isn't fully committed to the way BIM actually works, it's simply not going to work. Project stakeholders are overwhelmed with all the things they need to think about and when the pressure gets on due to deadlines, people tend to forget about the BIM. After the pressure goes away, they realize that they are behind with the BIM model and it is usually already worthless to keep using.

Information management (requirements management)

*7. How is documentation handled in your organization currently and h*ow much project information is
Almost 100% of the information is handled in PDF form because the client still wants to work with the traditional 2D drawings. But in this case also the contractor plays a role because not a lot of them have

incorporated BIM in their processes and are reluctant to use it. Sub-contractors have even lesser knowledge of BIM and therefore, they are yet another obstacle to the complete adoption of BIM.

8. How do you handle/systematize building requirements?
For civil projects we use Relatics and Excel but primarily Excel because Relatics is complicated for the older team members and project managers. It, however, depends on the client how we are going to manage requirements. In civil projects it is mandatory to use Systems Engineering because the client is usually a governmental institution. For building structures projects we try to systematize requirements mostly for internal use because the client doesn't demand them and therefore, other project parties don't feel the necessity to integrate processes. Therefore, we verify our own requirements.

9. In which manner do you ensure that requirement information remains up-to-date?
Usually there is a person assigned who supervises the process and puts the information in the right order. There is also a database of general requirements in our system so, in the BIM model a reference to these numbers is usually made in order to know to which requirements an element is applicable to but we don't prove in written form that the requirements are verified. It is more of a knowledge that the project's stakeholders know for themselves.

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
The final design phase or namely, the phase during which the execution drawings and documents are being generated. During the concept design and the preliminary design you can still get away with not doing things properly but in the detailed design you need to be precise. It is the phase where all problems need to be resolved as they cannot be shifted further in time.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
There is an obvious benefit to that but the main issue there is how we are going to have everybody wanting that system.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
Linking the company's internal requirements and documentation to actual 3D models would save time especially in traditional contracts where validating requirements is really time consuming.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
It is the most beneficial to do it from the start of a project because in later phases it can become time consuming and not that effective. After all, if implemented, this type of approaching projects would be mostly beneficial for the entire design and construction process. A phase that definitely has to be looked into first is the final design phase because the amounts of information there are the most and the detail of specifications and requirements is high. For that reason, it can be argued that this phase usually contains the complete as-planned data. The as-built situation should be also regarded however.

04.04.2017	Royal Haskoning DHV	Yves Scholtes	BIM Coordinator/Manager

General

1. Can you describe the core business of your organization?
We are a civil engineering company which works in several different domains.

2. What is your position in the organization and what are your main roles and responsibilities?
I am a BIM coordinator for some projects and general BIM manager of the division.

3. With what type of organizations do you most often cooperate?
We cooperate with designers, suppliers and sub-contractors primarily.

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?

Most often it is just traditional 2D drawings but BIM is also used for cost and volume estimations and timeline phasing where we can talk about 4D and 5D. So we use it for many different purposes, from tracking of costs, time and validation, to using it simply as a virtual reality model for showcasing the design.

5. Which tangible benefits have you experienced through the implementation of BIM?

The benefits are consistency in the management and design processes, reduction of errors because they can be prevented before the construction and in addition, BIM also saves time of cost estimators and all supporting disciplines as things such as volumes and measures can be extracted from the model by just several clicks with the mouse.

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?
It is very expensive and it takes a lot of time and clients always compare it to the traditional 2D and the costs of such process

Information management (requirements management)

7. How is documentation handled in your organization currently and how much project information is still stored in paper form?
Pretty much 100% of it as the senior engineers prefer to use PDFs and 2D plans and even the DMS systems are a bit of a challenge for them and they understand paper the best.

8. How do you handle/systematize building requirements?
We use BOX - a cloud data system with version control in it but the documentation in it is still in PDF format or in Excel format and ultimately I can describe it as being a very messy system to manage a project with. Drawings and documentation get lost quite often. Because I am responsible for food and beverages projects such as breweries Relatics is rather rare and it depends on the project and the client as to whether it will be used as a management system. For civil projects it is widely used as the government demands it but for other projects it isn't being implemented.

9. In which manner do you ensure that requirement information remains up-to-date?
There is not really a practice for that as we keep the requirements within the documentation and they are rarely systematized in a document altogether.

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
During construction or after the definite design phase because of the amounts of documentation and the fact that this documentation is transferred usually to parties who haven't been involved in the previous design phases but still need to understand what the specifics of the project are.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
Yes, because the geometrical visualization needs to be there and for achieving complete interoperability it needs to relate to all project information and documentation, not only to the requirements.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
There are many requirements which are important, in fact, all requirements are important but they can be prioritized within each professional domain. For us here requirements such as building specification requirements, calculation requirements, contractual requirements, procurement requirements, and asset management requirements are really important. What is the most important for the project in general depends on the client and the starting points given by him from the initiation.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
From the concept design phase or LOD 200 it is the most effective way as it saves time throughout the next phases. But when thinking about the 20/80 rule (20% effort and 80% results), then the final design phase or LOD300 is the most appropriate and requirements should be linked elementwise because systems-wise would very difficult. Linking on a systems level is not necessarily better as it could cause confusion.

15.05.2017	**Stam + De Koning**	**Stijn van Schaijk**	**BIM process Manager**

General

1. Can you describe the core business of your organization?
We (VolkerWessels) are an enterprise company which incorporates telecom, rail, construction and real estate. Stam De Koning in particular is involved in construction and real estate both for VolkerWessels and for external clients/investors.

2. What is your position in the organization and what are your main roles and responsibilities?
I'm a BIM process manager and I oversee the BIM processes and the quality of the BIM models.

3. With what type of organizations do you most often cooperate?
During the preconstruction phase we cooperate with designers and engineers and during the construction phase we mostly cooperate with sub-contractors.

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?
Before starting a project we define the project goals and the project ambition. Then we estimate what information we need for that ambition and based on that we define the information that we put in the BIM model. Therefore, for each project it is different but in general we track costs, time and validate the completeness of the design, as well as detect clashes. We have also a basic in-house IDM for the way a BIM model has to be constructed and for every project we need to follow the rules set there.

5. Which tangible benefits have you experienced through the implementation of BIM?
The major benefit, in my opinion, is the information reuse at this moment because it saves a lot of time. Also the visual aspect of the model which diminishes communication errors and problems coming from misunderstandings between the design parties plays a very important role for us. The information reuse contributes also to quality because you are able to align different models and see clashes on time. Also the process of communication with the project's subcontractors is easier and faster with the common coordinated design. It allows them to discuss the best practices for construction for example and to find the best solution faster.

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?
The dealing with many partners who have different interest in regard to the model is the hardest part of working with BIM. Architect want nice visualizations for example, while we care about constructability aspects and cost/time ratios. So creating a model where all parties can benefit from it is quite difficult. Every project has also different partners with different ways of communicating and working. The most important alignment is using the right tools for the right processes but a deeper issue is that we also have to deal with a lot of people who are not used to using IT tools in general. Therefore, the issue is not only the process but also the tools and the proper use of the tools by all stakeholders.

Information management (requirements management)

7. How is documentation handled in your organization currently and how much project information is still stored in paper form?

Documentation is handled primarily in paper form so I would say 40% BIM and 60% unstructured and unlinked data such as PDFs and Excel sheets. You have to link objects to documents and documents to objects in your head if you want to have continuity of data and be able to work effectively on a project.

8. How do you handle/systematize building requirements?
The initial program of requirements put in an excel sheet. Based on this excel sheet we check if the requirement is fulfilled within the model and whether there is documentation related it but everything is done in a proactive manner and there aren't strict rules set which is a weak point on our part. Drawings are generated from the model so the requirement is proved based on the model or on the drawing - depends on the perception of the person checking. The best thing is rely on the model because you can implement rule checkers there but not on the 2D drawing.

9. In which manner do you ensure that requirement information remains up-to-date?
We translate the data in the model and compare it with the program of requirements. There is a place where we store all project data and it is called Trimble connect. It is a data management platform and all projects are stored there, also the PDF files and models from all the stakeholders. You can attach a PDF to a floor slab for example which is very useful for monitoring the processes but the biggest question is what do we connect in order to keep the consistency of the model and the information it contains. For every project delivery phase at the end we freeze the model so that in the next phase the comparison can be made as to whether the project is still on track with the initial objectives. Models checked against frozen state can also make the traceability of design conformity with the requirements better.

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
In the construction phase usually because that is when the actual product gets produced.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
Yes, because we are trying to do it now right now and it definitely provides a better overview of the entirety of information sources. The bidirectional relations between the requirements documentation and the model objects are important in this sense. Also what is important is that we distinguished between the as-designed and as-built states of a project in relation to their requirements and documentation.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
Everything related to maintenance data because the people within the project know more or less where to find the documents but the maintenance company hasn't been involved in the project but needs to know which documentation relates to what in element in the model. They need to see location, requirements and what the object is all about.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
At least during the LOD300 where all of the production information is elaborated on but in my opinion it should be an evolving process throughout the different project phases in order to get the most benefit from such an approach.

23.05.2017	Verhoeven en Leenders	Lucas Verhelst	Structural engineer

General

1. Can you describe the core business of your organization?
Our company specializes in structural engineering for civil works and for building construction projects mostly in concrete and steel.

2. What is your position in the organization and what are your main roles and responsibilities?
I'm a structural engineer and modeler.

3. With what type of organizations do you most often cooperate?
We cooperate with architecture firms, contractors, sub-contractors, manufacturers and consultants.

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?

For seeing construction errors early in the design and for taking proactive actions to correct the errors.

Therefore, we use BIM for validation and verification mainly and it helps us as we can think about problems before they happen, which is the biggest difference with using only 2D Cad drawings.

5. Which tangible benefits have you experienced through the implementation of BIM?

There's definitely the higher efficiency in the design processes and the aspect of risk mitigation such as the mitigation of budget and time overruns.

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?
The many stakeholders involved in a project and the differences in knowledge about BIM that each stakeholder has is the biggest bottleneck at this moment. Getting everyone on the same page is an issue both internally within the company and externally with other parties. There are always conservative voices also within our company and also the habits that people have developed play a big role. I believe that the modelers are far ahead of the engineers in terms of BIM knowledge.

Information management (requirements management)

7. How is documentation handled in your organization currently and how much project information is still stored in paper form?
Still a great deal of the documentation is in PDF and Excel format. So probably 90%.

8. How do you handle/systematize building requirements?
For civil projects we use Relatics because the client demands the use of Systems Engineering practices. For private projects the main way we handle requirements is through e-mails or phone calls, which, of course, is not the way it should be done.

9. In which manner do you ensure that requirement information remains up-to-date?
We usually consider the codes and regulations as our primary requirements and not so much the program of requirements which comes from the client as it is related to the architectural design primarily. We have a subscription to the NEN website database and a person from the company tracks the changes related to the codes and lets everyone else know what is new. Project-specific requirements are primarily in the head of the engineers and remain undocumented.

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
For us LOD300 is a very important phase because we generate the most of our information then. I think, however, that there should be an overview of the entire design process because some things such as the decision on a foundation system need to be decided far ahead.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of nonconformity)?
It helps support the overview of all information datasets and it is an easy way to check whether the requirements are considered or regarded correctly. Way easier to get an overview rather than going in Relatics and then checking the BIM model afterwards.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
I think the most important requirements are the one from the codes such as material requirements, loads, physical requirements on sizes of objects, etc. Basically everything necessary for the production of our calculations.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
The most crucial phase is the final design phase or the LOD300. The LOD350 is also important but it is not always executed for every project. In earlier stages there are a lot of requirements that can be left out because they are also way too vague and get elaborated on at LOD300.

25.05.2017	Verhoeven en Leenders	Jurgen van der Aa	Structural engineer
25.05.2017	Verhoeven en Leenders	Henk Verhoeven	Project manager /Engineer

General

1. Can you describe the core business of your organization?
We are an engineering company which specialized both in civil and building structures projects.

2. What is your position in the organization and what are your main roles and responsibilities?
I'm a structural engineer.

3. With what type of organizations do you most often cooperate?
We most often cooperate with architects, contractors, subcontractors and consulting companies.

Experience with BIM

4. Does your organization work with BIM and for which purposes do you implement it within a project?
Yes, we use BIM primarily for validation and verification purposes and for clash detection.

5. Which tangible benefits have you experienced through the implementation of BIM?
The real connections between the different elements are much more visible in comparison to the 2D drawings where such details are sometimes omitted. With 2D drawings the difficult parts are neglected and BIM allows us to specify details in an early stage so problems are more detectable throughout the design phases. Also the clashed with other domains are detected much earlier. Although there is more effort at the beginning of the project's design, later during construction time and money are saved.

6. Which bottlenecks do you foresee regarding the recent transition to BIM-centered project management?
In a very early stage you have to make a lot of final decisions in comparison with before when these decision could have been taken even in the final design project phase. In this case the contractor and the MEP engineers are sometimes reluctant to take decisions so early in the process because they are likely to pay more this way than if they buy the materials and technical systems later on because the subcontractors are more likely to have the time span for negotiating higher prices. Getting the contractor on board at an early design stage generally has many benefits related to the reduction of costs during construction.

Information management (requirements management)

7. How is documentation handled in your organization currently and how much project information is still stored in paper form?
It is handled primarily in paper (PDF) form and in Excel for building structures projects. Of course, there is a BIM model but it doesn't contain all of the information of the project. For civil projects there is Relatics and BCF but still the majority of information is contained in PDFs.

8. How do you handle/systematize building requirements?
In the building structures sector it is the toughest to manage requirements. We usually use the Eurocodes as a guideline for the engineering requirements but the requirements defined by the user remain in the program of requirements because the requirements defined there have to do more with the architect than with us. Yes, both engineering and architectural requirements are connected but it is more in the heads of the engineers than documented on paper or on a platform such as Relatics.

9. In which manner do you ensure that requirement information remains up-to-date?
-

10. At which stage of a construction project is it most likely to encounter difficulties in the management process due to lack of traceability between design and requirements?
All of the information the stakeholders coming after us need is in the calculation reports that the engineers produce. Sometimes, however, they don't know how to read the calculations. The structural engineers, therefore, quite often also have to coordinate the suppliers because questions always arise such as positioning of elements for example.

Information management and BIM implementation

11. Do you think that linking requirement documentation (information) to a BIM could have potential benefits for the construction field (such as mitigating risks of non-conformity)?
Yes, but things should be easy to filter because some elements might have 50 pages of information and maybe I need only 2 sentences which are relevant to my work. Therefore, if I can reach these 2 sentences instantaneously it will improve my way of working but if I have to look for them, then it won't do a lot. The architects have more requirements especially at the initial phases of a project so the functionality would be more useful for them. The engineers have repetitive requirements and making the link can be time consuming. But I think that such approach is very beneficial for the owner or operations company that maintains the building afterwards.

12. The linking of which type of requirements, in your opinion, would improve the information management process within a building project the most during the design phase?
Pretty much everything that is in the calculation and advisory reports.

13. On which level of detail would it be most beneficial to connect (engineering) requirements and design?
Probably on both elements and systems level but the problems is that systems aren't strictly defined so every engineer would consider them differently and this may cause confusion. But any of both would suggest LOD300 as the level at which it will be most beneficial as that is where the most important and detailed engineering information is generated by us.