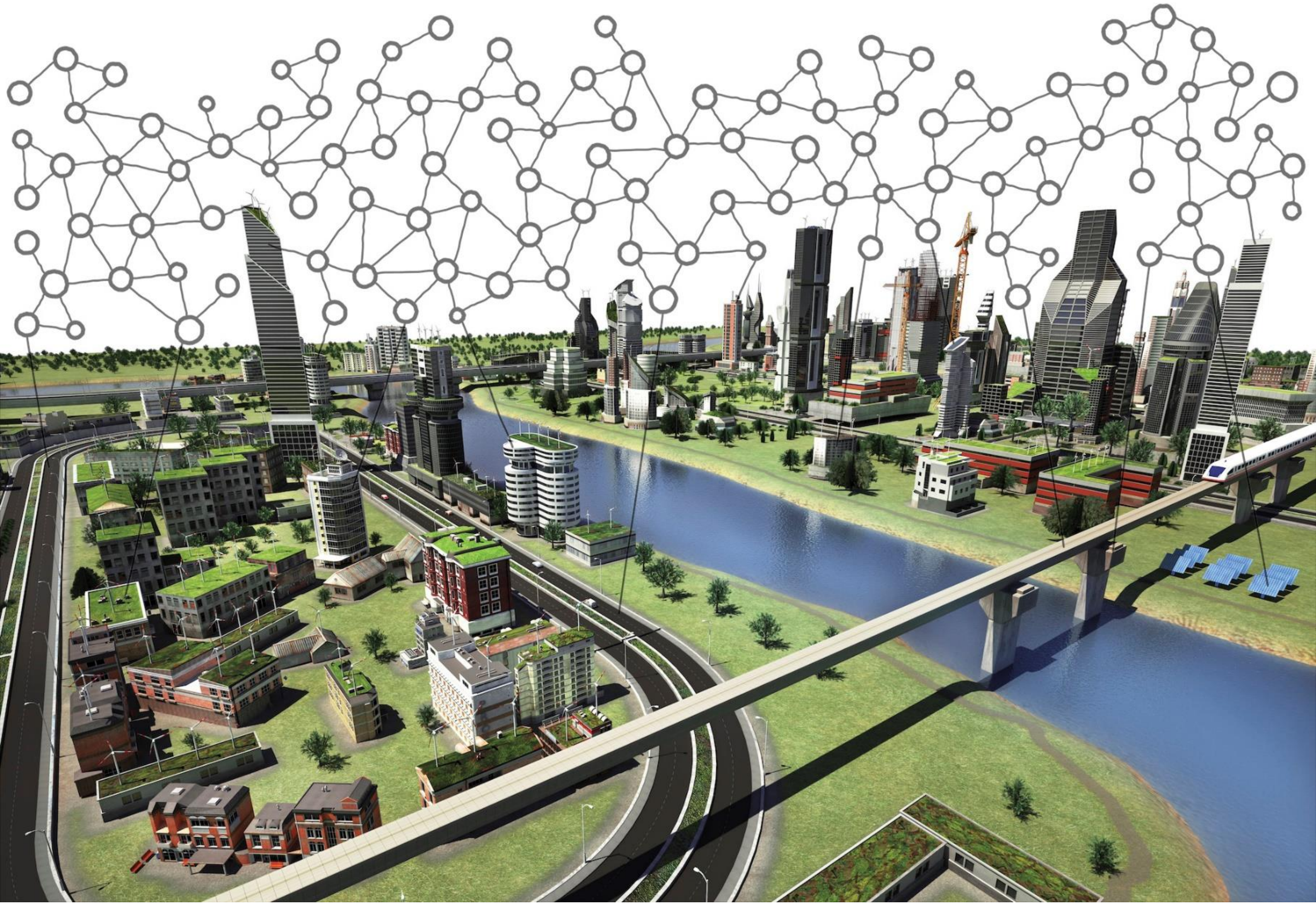


# SEMANTIC MODEL ENRICHMENT FOR BIM-ENABLED RISK-BASED OPERATION AND MAINTENANCE

*A case study approach with Industry Foundation Classes*

**M.H. (Matthijs) van de Riet**

February 18, 2016

*Construction Management & Engineering*

**2016**

# Semantic Model Enrichment for BIM-Enabled Risk-Based Operation and Maintenance

**A CASE STUDY APPROACH WITH INDUSTRY FOUNDATION CLASSES**


| | |
|---|---|
| Author | : M.H. (Matthijs) van de Riet |
| Student number | : 0874054 |
| Date | : February 11, 2016 |
| E-mail | : mh.vanderiet@gmail.com |
| Date of final presentation | : February 18, 2016 |

UNIVERSITY

| | |
|---|---|
| University | : Eindhoven University of Technology (TU/e) |
| Faculty | : Faculty of the Built Environment |
| Master track | : Construction Management & Engineering (CME) |

IN COLLABORATION WITH

| | |
|---|---|
| Company | : Arcadis Nederland |
| Division | : Mobility |
| Advisory group | : Information Management advisory group |

GRADUATION COMMITTEE

| | |
|---|---|
| Chairman (TU/e) | : prof.dr.ir. B. (Bauke) de Vries |
| Graduation supervisor (TU/e) | : dr.Dipl.-Ing. J. (Jakob) Beetz |
| External supervisor (Arcadis) | : Ir. B.J.A. (Bob) van Thiel |

Technische Universiteit Eindhoven University of Technology

ARCADIS Design & Consultancy for natural and built assets

# PREFACE

I am happy to present this thesis which is the result of my graduation project carried out in collaboration with the Eindhoven University of Technology and Arcadis. After six months of hard work I finished my thesis representing the end of my master study Construction Management & Engineering and my time as a student.

By conducting research in the field of Building Information Modeling (BIM), operation & maintenance, linked data and semantic web I had chosen a challenging topic. Especially because I was passionate about programming a real and usable application without any real programming experience in advance. Eventually, after a lot of YouTube tutorials, literature and online programming forms this turned out as I hoped. I consider the complete research as a great opportunity to learn and develop myself in the field of information management from which I will definitely profit in my career.

However, I could have never done this without the help and support of some people. During moments where I needed guidance and support they helped me to get back on track and push on. First of all, I would like to thank my supervisor Jakob Beetz (TU/e) for his support, guidance and technical discussions helping me to keep improving my work. In addition, I want to thank Bauke de Vries (TU/e) for the discussion we had and his overall supervision on the progress of my graduation project. And of course Bob van Thiel (Arcadis) for the valuable discussions we had and pointing me to the right connections within the company. Furthermore, when I got stuck with programming and did not know how to proceed Thomas Krijnen helped me to resolve any issues, for which I am grateful. Then, many thanks to Kris McGlinn (Trinity College Dublin) for sharing work on the RDF parser where I could build upon.

Last but not least, I want to thank all colleagues at Arcadis for helping me and the great time I had. I'm of course also grateful to all interviewees for making time and providing valuable input and insights in the current data struggles of the industry. Finally, I would like to thank my girlfriend, family and friends for all their help and support!

I hope you will enjoy reading and learn from this thesis as much as I did.

Matthijs van de Riet

Tilburg, February 2016

# SUMMARY

Integration of Building Information Modeling (BIM) working methods in post construction phases are not very common in the Dutch Architecture Engineering and Construction (AEC) industry yet. However, in terms of money, subsequent Operation and Maintenance (O&M) costs of a building over its lifecycle could amount to many times more than the construction costs. In addition, this exploitation phase of a building is the longest phase.

During a building's exploitation, diverse models or representations of a building are available with a vast amount of external data resulting in the need for distributed data management. This information and the BIM are usually stored locally and shared via email or project management systems and rarely connected across domains. While during exploitation, a lot of (re)usable data is generated, the industry is failing to enrich available models with this information. This inefficient information management causes significant costs. Furthermore, the lack of interoperability between software applications hampers the integration of BIM during the whole building's lifecycle.

The Industry Foundation Classes (IFC) standard is increasingly accepted by the AEC industry as interoperable file format due to the ongoing effort of BuildingSMART. IFC models are semantically rich because they capture not only the 3-dimensional geometry of objects, but metadata related to many other aspects of this object and the building as a whole. Semantic enrichment of these building models has the potential to facilitate a more optimal O&M processes by providing means to structure, preserve and visualize relevant data. In order to investigate this further, in this research, information management for risk-based O&M is optimized by semantic model enrichment. Risk-based O&M is a relatively novel approach in the industry for finding the optimal balance between structural reliability and lifecycle cost of deteriorating buildings. Frequently the Failure Mode and Criticality Analysis (FMECA) methodology is used to determine failure modes and associated risks for building objects. The discussed situation results in the following main research question: "How can IFC models be enriched semantically to improve risk-based operations & maintenance of an AEC/ FM project?". To be able to answer the question, first expert interviews were held to get a better understating of the current situation and future needs of the industry. The experts pointed out the inefficiency of current BIM project handover to O&M. In addition, experts recognize the potential of using 3D models for O&M to manage and structure information. Furthermore, making a good data selection prior to handover is essential.

Then, two prototype tools are developed to enable a facility manager to enrich an IFC model with risk-based O&M data. For this tool development and testing, one building of the 'Uithoflijn' project is selected of which an IFC model is available. This model concerns an overpass facilitating the tramway passing a canal.

Prototype tool one consists of three stages. The first stage enables the asset manager to extract all objects from an IFC model and write these to a FMECA sheet according to the NEN2767-4 object breakdown structure (OBS). Second, after completion of the FMECA by the asset manager, in stage two it is possible to write this FMECA information to IFC. Thereby, FMECA data can be viewed by any IFC viewer on the market. Finally, in stage three risk data can be visualized and IFC element properties (FMECA) can be viewed.

While a working prototype is the result some drawbacks of this first approach are discovered. The three main disadvantages are that this data in IFC is hard to query/reuse, versioning risks exist and the IFC file gets polluted. Due to these drawbacks, a second tool is developed using semantic web technologies. First, this second prototype tool transforms tabular FMECA data into RDF resulting in data with semantic meaning. Second, the tool

provides the possibility to view an IFC model with element properties and provide associated FMECA RDF data by selection of an element in 3D view. In addition, the ability exists to visualize RPN values, obtained from RDF, in terms of colors. No ontology which describes FMECA parameters seems to be available. Therefore, for this research a concise ontology is created. This ontology defines the FMECA parameters with associated data and value restrictions. This second tool eliminates the disadvantages of first approach, the data is queriable and reusable using SPARQL, no IFC pollution occurs and reduced versioning risks due to the separation of data repositories. In addition, this approach is more future proof due to the expected shift towards semantic web technologies within the industry. However, current IFC viewers are not compatible with semantic web technologies.

As an answer to the main research questions both approaches are viable, although the second approach is highly preferred. Recommendations are that any object, whether in IFC or other data format, contains an object classification and Globally Unique Identifier (GUID). In addition, optimally no object properties are stored in the IFC model itself due to the discussed disadvantages respecting the importance of good data selection prior to handover O&M. The shift towards semantic web technologies could enhance data sharing between the AEC and FM industry facilitating e.g. design for maintenance. While the FM industry is considered rather traditional the initiative and incentives for such a shift should come from building owners by stimulation of innovations through more extensive implementation of performance based contracts.

# SAMENVATTING

Implementatie van een Bouw Informatie Model (BIM) tijdens beheer en onderhoud (B&O) van een bouwwerk is nog niet erg gebruikelijk in de Nederlandse bouwsector. Echter, financieel gezien is deze fase veel omvangrijker in vergelijking met de ontwerp en bouwfase. Ook is deze exploitatiefase veruit de langste fase van een bouwwerk.

Gedurende de levensduur van een bouwwerk en daarmee ook de exploitatiefase, zijn er veel verschillende modellen of representaties met grote hoeveelheid externe gegevens beschikbaar waardoor gegevensbeheer van deze gedistribueerde informatie essentieel is. Deze informatie en het BIM zijn meestal lokaal opgeslagen en gedeeld via e-mail of project management systemen en zelden bestaat er een link tussen de verschillende bouwdisciplines. Terwijl tijdens de exploitatie veel (her)bruikbare gegevens worden gegenereerd slaagt de sector er niet in om een link tussen deze gegevens te leggen en zodanig op te slaan dat deze later makkelijk (her)gebruikt kunnen worden. Bovendien is het gebrek aan compatibiliteit tussen softwaretoepassingen belemmerend voor de integratie van BIM gedurende de gehele levenscyclus van een gebouw. Dergelijk inefficiënt informatiebeheer veroorzaakt daardoor aanzienlijke faalkosten.

De ‘Industry Foundation Classes’ (IFC) worden in toenemende mate door de bouwsector als interoperabel bestandsformaat geaccepteerd dankzij de voortdurende inspanningen van BuildingSMART. IFC modellen zijn semantisch rijk omdat ze niet alleen de 3-dimensionale geometrie van objecten bevatten maar ook de daaraan gerelateerde metadata op het gebied van vele aspecten. Semantische verrijking van deze modellen met B&O gegevens biedt potentieel voor een optimaler beheer- en onderhoudsproces door structureren en visualiseren van relevante gegevens. Om dit verder te kunnen onderzoeken is in dit onderzoek gekeken naar semantische modelverrijking met gegevens voor risicogestuurd B&O. Risicogestuurd B&O is een relatief nieuwe benadering voor de bouwsector waarbij er wordt gezocht naar een optimum tussen structurele betrouwbaarheid en onderhoudskosten van bouwwerken. Vaak wordt hiervoor de Failure Mode and Criticality Analysis (FMECA) toegepast waarbij de faalmodus en bijbehorend risico voor bouwelementen worden geïdentificeerd.

De bovenstaande situatie resulteert in de volgende hoofdvraag voor dit onderzoek: “Hoe kunnen IFC-modellen semantisch worden verrijkt om risicogestuurd beheer en onderhoud in de bouwsector te optimaliseren?”. Om deze vraag te kunnen beantwoorden zijn er eerst expert interviews gehouden om een beter beeld te krijgen van de huidige situatie en de toekomstige behoeften van de sector. De interviews bevestigde het vermoeden dat informatie na de bouw inefficiënt en vaak incompleet wordt overgedragen aan B&O. Hierbij blijkt het zeer belangrijk te zijn om een goede gegevensselectie te maken vóór de overdracht naar B&O. Bovendien erkennen de experts het potentieel van het gebruik van 3D BIM modellen voor B&O om informatie te structureren en beheren.

Vervolgens zijn er twee prototype tools ontwikkeld om de beheerder in staat te stellen een IFC model te verrijken met risico gestuurd B&O informatie. Voor deze toolontwikkelingen is een bouwwerk van het project ‘Uithoflijn’ als case gebruikt, van dit bouwwerk is een IFC model beschikbaar. Dit model betreft een tram viaduct over een kanaal.

De eerste prototype tool bevat drie stappen. De eerste stap maakt het mogelijk voor de beheerder om alle objecten uit het IFC model te extraheren en naar een FMECA werkblad te schrijven volgens de NEN2767-4 ‘Object Breakdown Structure’ (OBS). In de tweede stap, na afronding van de complete FMECA analyse door de beheerder, is het mogelijk om alle FMECA informatie uit de werkblad naar IFC te schrijven. Daarna is het mogelijk om deze

informatie met een van de vele IFC viewers op de markt te benaderen. Ten slotte, in de derde stap is het mogelijk om, naast toegang te bieden tot FMECA informatie in het model, risico's te visualiseren in het model. Ondanks dat dit een werkend en praktisch toepasbaar prototype betreft kunnen een aantal nadelen van deze methode worden onderscheden. De drie belangrijkste nadelen zijn dat informatie in IFC is moeilijk te doorzoeken en hergebruiken is (1), versierisico's ontstaan bij beheer van dezelfde informatie in IFC alsmede in een werksheet (2) en vervuiling van IFC bestand (3). Vanwege deze nadelen is er een tweede prototype tool ontwikkeld welke gebruik maakt van semantische web technologie. Eerst transformeert deze tool de tabulaire FMECA data naar RDF wat resulteert in data met semantische betekenis. Vervolgens is het mogelijk om het IFC model te bekijken en een element te selecteren waarna de eigenschappen uit IFC alsmede de bijbehorende FMECA informatie uit RDF wordt getoond. Daarnaast is het mogelijk om de risicowaardes, verkregen uit RDF, te visualiseren met kleuren. Omdat er geen FMECA ontologie beschikbaar is die de relevante parameters beschrijft is er een beknopte ontologie opgesteld in dit onderzoek. Deze ontologie definieert FMECA parameters met de bijbehorende datasoort en restricties. De tweede tool elimineert de nadelen van de eerste benadering, de informatie is doorzoekbaar en herbruikbaar met behulp van SPARQL, er treedt geen IFC vervuiling op en een vermindering van versierisico's vanwege de scheiding van informatie is het resultaat. Ook lijkt deze aanpak toekomstbestendiger vanwege de verwachte verschuiving naar semantische web technologieën binnen de sector. Echter, momenteel beschikbare IFC software is niet compatibel met semantische web technologie.

Als antwoord op de onderzoeksvraag zijn beide toolbenaderingen haalbaar en toepasbaar, hoewel de tweede benadering in hoge mate de voorkeur heeft. Aanbevelingen zijn dat ieder object, zowel in IFC als in een ander data formaat, een Globally Unique Identifier (GUID) en classificatie bevat. Optimaal gezien zouden objecteigenschappen ook niet worden opgeslagen in het IFC model zelf vanwege de besproken nadelen en daarbij rekening houdend met goede dataselectie vóór overdracht naar B&O. De verschuiving naar semantische web technologieën kunnen het delen van gegevens tussen ontwerp/bouw sector en B&O vergemakkelijken en daarmee bijvoorbeeld 'design for maintenance' faciliteren. Omdat de B&O sector als vrij traditioneel kan worden beschouwd moet het initiatief en stimulans voor een dergelijke verschuiving komen van de eigenaren van gebouwen door meer frequente toepassing van prestatiecontracten.

# INDEX

# 1 INTRODUCTION

During mid-1990s, great amount innovative and new ICT applications in the architecture engineering and construction (AEC) sector have been developed. Developments resulted in sophisticated CAD systems, where it was possible to enrich 3D models of buildings with, in addition to vectorial data, complementary data such as physical characteristics, unit costs, quantity take-offs, etc. This methodology became known as building information modelling (BIM) (Grilo & Jardim-Goncalves, 2010a).

In the AEC industry though, the focus of BIM integration is predominantly on (pre-) construction phases. Integration of BIM working methods in post construction phases is not very common in the Dutch AEC industry yet. So, while BIM processes are established for new buildings, the majority of existing buildings are not maintained, refurbished or deconstructed with BIM nowadays (Volk, Stengel, & Schultmann, 2014).

However, when speaking in terms of money, these subsequent operation and maintenance (O&M) costs of a building over its lifecycle could amount to many times more than the construction costs (Becerik-Gerber, Jazizadeh, Li, & Calis, 2012). These cost proportions are schematic represented in figure 1-1. In addition, with a time span of 30, 50 or sometimes even more years it is also the longest phase. Therefore, there is a growing interest in the use of BIM in the exploitation phase for coordinated, consistent, and computable building information/ knowledge management during the whole lifecycle (Becerik-Gerber et al., 2012).

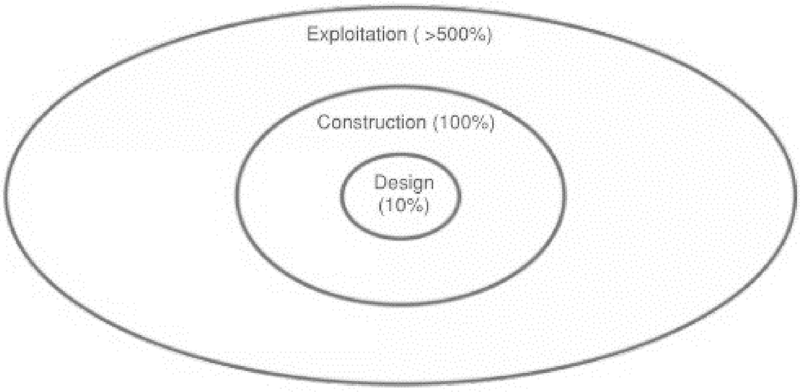



Figure 1-1: Schematic representation of building lifecycle costs (Verbaan, Visser, Koe, Boer, & Voet, 2014)

However, the lack of interoperability between software applications hampers the integration of BIM during the whole building’s lifecycle. The Industry Foundation Classes (IFC) are increasingly accepted by the AEC industry as interoperable file format due to the ongoing effort of BuildingSMART (Beetz, Leeuwen, & Vries, 2005). The use of IFC in every phase of a building’s lifecycle has the potential to prevent information loss and ease the handover process from one department to another.

Semantic enrichment of these building models has the potential to facilitate a more optimal O&M processes by providing means to structure, preserve and visualize relevant data. Pruvost et al. (2012) recognize the value of model enrichment (e.g. with linked models) providing the nested structure which can be of substantial help for performing risk-based O&M. Risk-based O&M is a relatively novel approach in the industry for finding the optimal balance between structural reliability and lifecycle cost of deteriorating buildings.

# 2 RESEARCH APPROACH

In this chapter the research approach will be elaborated to set a research outline. First, the problem definition and research gap are defined in section 2.1. Thereafter in section 2.2, sub-research questions are presented with associated main research question and objectives/ limitations of this research. In section 2.3 the research design is discussed by means of a methodological justification and research model. Finally, in section 2.4 expected results are presented to conclude this chapter.

## 2.1 Problem definition and research gap

The exploitation phase of a building is the longest and most expensive phase (Becerik-Gerber et al., 2012). In this context, a growing amount of scientific literature is available which indicate that the potential benefits of using BIM for O&M seem to be significant (e.g. Becerik-Gerber et al., 2012; Arayici, 2008; Akcamete, Akinci, & Garrett, 2010).

Within the exploitation phase of a building, vast amounts of information are being produced such as maintenance data, risk-based O&M data, energy use and occupancy patterns. In the current working methods, this information and the BIM are usually stored locally and shared via email or project management systems and are rarely connected across domains. In addition, information generated outside the model based on exports of for instance quantity takeoffs is seldom fed back into the model. This inefficient information management causes significant costs (Dankers, Geel, & Segers, 2014).

The BIM is generally the central point of information used in the construction process, more and more effort is put into using BIM during the exploitation phase of the building. However, information outside the BIM is usually not connected to the relevant elements inside the BIM (Dankers et al., 2014).

Actual BIM application in the Dutch AEC and Facility Management (FM) industry however remains behind. This is partly caused by interoperability issues between BIM standards and O&M software. This incapable interoperability is still a major obstacle in BIM data exchanges both in new and existing buildings (Volk et al., 2014). The effects are vast amounts of data losses after construction. This is visualized in figure 2-1, where the traditional paper based Design-Build process is compared with the collaborative BIM-based process in terms of information assets. As illustrated in figure 2-1, information is lost after each phase in the traditional Design-Build working method. The BIM-based working method has overcome this problem significantly with exception of the handover from construction to future phases. Though, interoperability issues are reduced by the implementation of an open model standard, called Industry Foundation Classes (IFC).

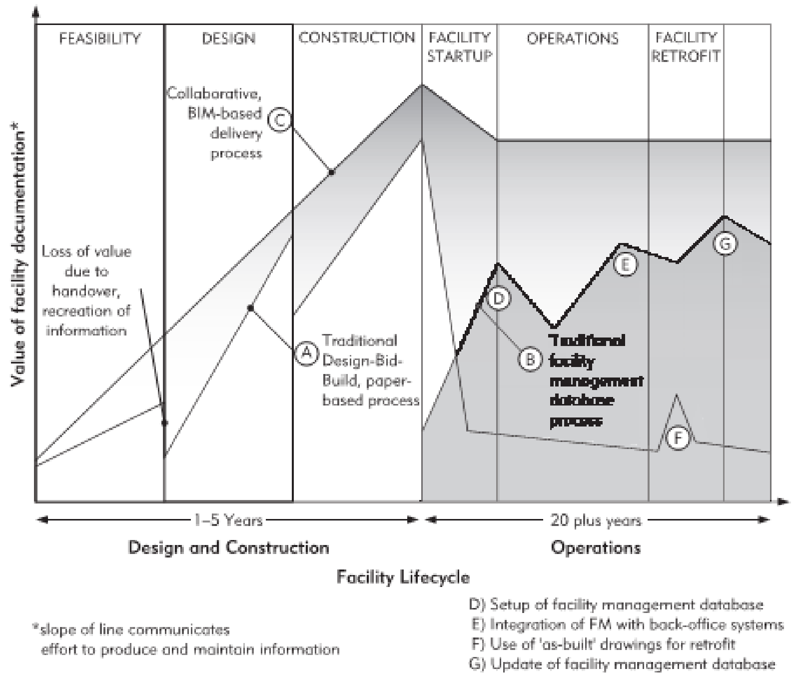



Figure 2-1: Comparison of traditional and BIM-based processes in terms of information assets (Eastman, Teicholz, Sacks, Liston, 2008)

Figure 2-1 also illustrates a lack of model enrichment during post-construction phases, while a lot of (re)usable data is generated. Currently O&M information is stored in proprietary file formats or non-queryable data structures. This is resulting in inefficient information use and limits O&M information preservation and reuse in future projects.
In order to overcome this, semantic enrichment of IFC models with O&M data is proposed in lots of scientific literature (i.a. Visser, Boer, & Voet, 2013; Belsky, Sacks, & Brilakis, 2015; Dankers et al., 2014; Vanlande, Cruz, & Nicolle, 2008; Beetz, Coebergh, Botter, Zlatanova, & de Laat, 2015).

## 2.2 Research question

For this research the following hypothesis is formulated: “By Semantic enrichment of IFC models O&M data can be preserved and structured efficiently”. In order to test this hypothesis the following central question will be answered:

**How can IFC models be enriched semantically to improve risk-based operations & maintenance of an AEC/ FM project?**

The following sub questions will be answered in this research:

1. What is the added value of Industry Foundation Classes (IFC) in Building Information Modeling use for O&M phases?
2. Which approaches can be employed to semantically enrich IFC model populations and what are their possibilities and limitations?

3. What are object classifications and how can they be of added value in risk-based operation & maintenance?
4. What are important IFC model conditions for handover towards the O&M phase?
5. What is risk-based operation & maintenance and how can this be applied practically?
6. How could risk-based O&M data be reused in future projects?

### 2.2.1 Research objectives and limitations

The aim of this research is to present an advice on how risk-based O&M can be integrated in BIM based working methods. The focus of this research is to enhance information management and preservation of risk-based O&M data. Therefore, a tool to integrate risk-based O&M and IFC models is developed. This tool should enable the facility manager to view and visualize O&M data in a 3D model environment. The tool will be tested on a real-world IFC model.
This tool is developed to facilitate a connection between IFC and FMECA risk analysis, other risk-based O&M methods are not covered. Further research and testing is needed to improve and test the prototype tool for practical integration.

## 2.3 Research design

In this section the research design is elaborated by a methodological justification first. Here, the use of desk research, interviews and tool development with case study is justified. Then, the research model is presented with accompanying explanation.

### 2.3.1 Methodological justification

To get a better understanding of the problem and the current available scientific literature, desk research is done in the first stage. Desk research is the process of gathering and analyzing information, already available in print or published on the internet (Businessdictionary, 2014). According to Hilbe (2014) for a researcher to successfully undertake new research they must contribute 'new knowledge' to this total store of knowledge through publication in the same way. In order to create 'new knowledge', available literature must be studied first to set a baseline.
Using the developed knowledge obtained by desk research, expert interviews will provide better understanding of the current situation in the AEC/ FM industry regarding the main topic of this research. Experts in the field of BIM and facility management of the engineering company Arcadis and other companies will be interviewed to get more knowledge about the industries latest developments and views on future needs. Based on expert views, a tool will be developed to support BIM and FM integration.
In general this tool must enable the facility manager to link, view and visualize O&M data within 3D building models. Python programming language and various modules such as IfcOpenShell, PyQt, RDFLib and OpenPyXL are used for development. IfcOpenShell is an open source software library that helps users and software developers to work with the IFC file format. IfcOpenShell uses Open CASCADE internally to convert the implicit geometry in IFC files into explicit geometry that any software CAD or modelling package can understand (ifcOpenShell, 2015).
The case study is a real world project called 'De Uithoflijn' commissioned by 'Regiotram Utrecht' (RTU). The Uithoflijn will be a tram connection between Utrecht Central Station and Utrecht Science Park De Uithof by 2018. For tool testing and validation, case models and

data are used. Case study research is inquiry focusing on describing, understanding, predicting and/or controlling the individual (Woodside & Wilson, 2003). Manageable parts of this case are used to test and validate if a working and usable tool is created.

### 2.3.2 Research model

The research model can be divided into four parts which represent the main phases of the research (figure 2-2). First, the objectives are set which are translated into research questions.

In the second phase, desk research is done to set the base of the research by investigating 'Building Information Modeling (BIM)', 'Risk-based operations & maintenance' and 'Semantic model enrichment'. With this literature study, a number of sub-research questions can be (partially) answered.

Third, a practical application with interviews and a case study will create insight into the practical possibilities and pros and cons of semantic IFC model enrichment. As stated in the previous section, a tool will be create which makes it possible to enrich a model with risk-based O&M information and the possibility to visualize this data in a 3D environment. This tool will be tested using the real-world case.

Fourth, tool reflection, interview conclusions and literature study will lead to the final conclusion. Here, the central research question is answered and recommendations for further research are proposed.

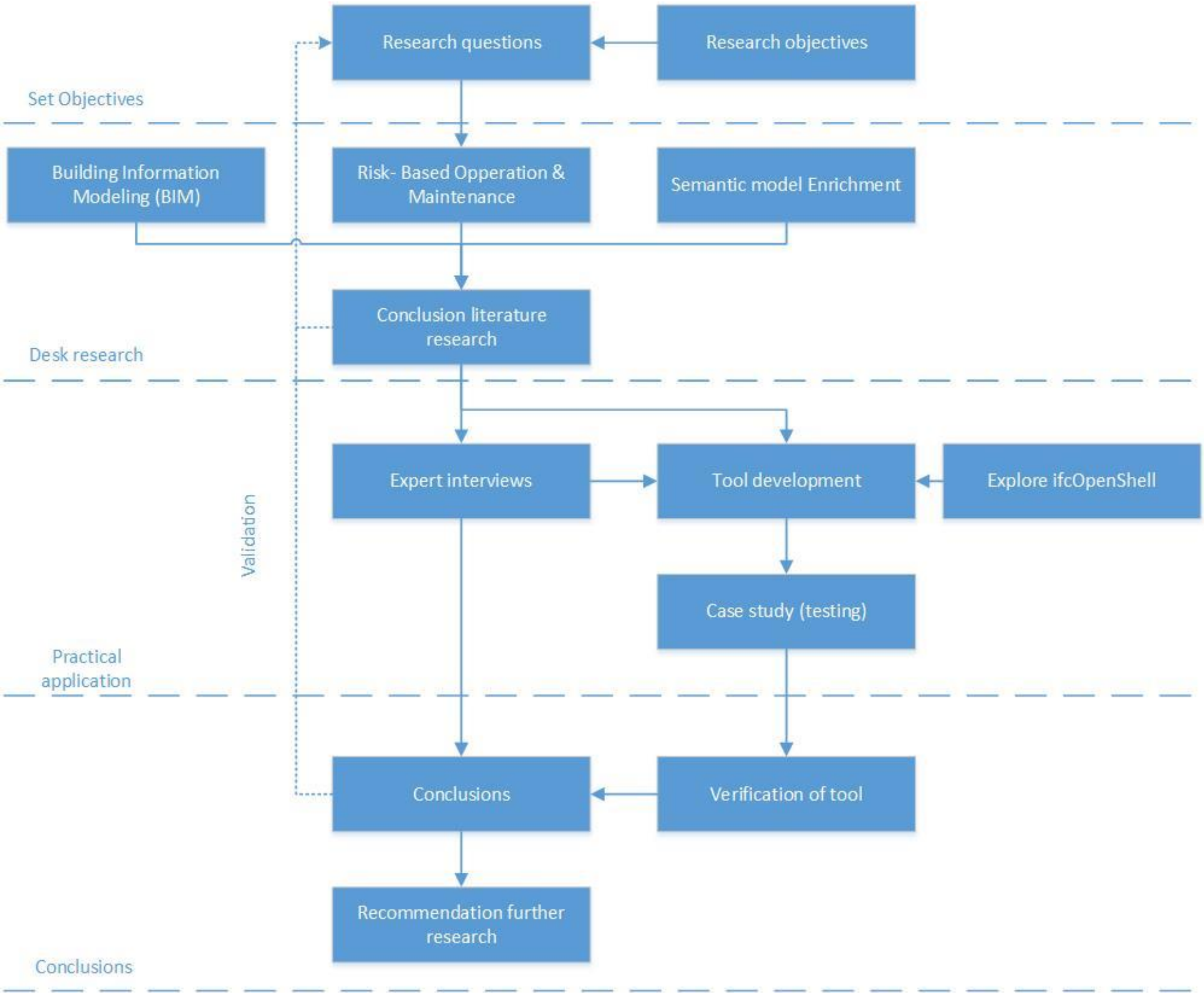



Figure 2-2: Research model

## 2.4 Expected results

By finding an answer to the central research question the expectation is that an advice can be presented on the preservation, reuse and optimization of risk-based O&M data using BIM methodologies. Results may be generalized to be able to present an advice on how data should be handed over to facility managers and reused by the AEC industry (figure 2-3). This seems challenging because currently both industries are rather disassociated.

In order to answer the research questions and present an advice, a tool will be developed which must be free to use and provide a simple interface to work with IFC models and risk-based O&M data. To be able to obtain practically relevant results, a real-world case will be used to test information management of risk-based O&M data and IFC.

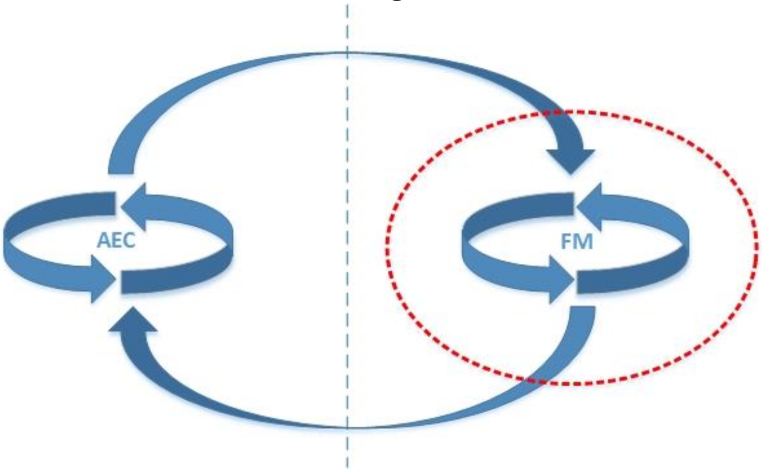



Figure 2-3: Expected generalizable results

# 3 BUILDING INFORMATION MODELING (BIM)

Nowadays Building Information Modeling (BIM) seems to be a buzzword and often used when talking about 3D models. However, BIM is much more than just a 3D model. In section 3.1, BIM is defined by discussing and comparing multiple definitions from the literature. In section 3.2, BIM maturity levels are explained to get insight in the extent in which BIM could be implemented. In section 3.3, BIM applications across the project lifecycle are elaborated in terms of current implementation in the industry and classification of BIM applications. Thereafter in section 3.4, interoperability is defined and current interoperability issues are discussed. In section 3.5, the open standard Industry Foundation Classes (IFC) is elaborated to get insight is the possibilities and limitations. Section 3.6, provides explanation of object classifications and concept libraries. Finally in section 3.7, the principles 'Level of Detail' and 'Level of Information' are discussed.

## 3.1 What is BIM?

Building Information Modeling (BIM) is much more than just a 3D model, nowadays it can be considered as a proven AEC technology applied in a steadily growing number of projects in the industry. Due to the fact that the extent in which BIM is applied differs significantly among companies and/or projects, many definitions exist. Currently, it also seems scientific literature has failed to reached a consensus about a single, widely- accepted BIM definition (Eastman, Teicholz, Sacks, & Liston, 2008). Therefore, multiple definitions are discussed to get insight in differences and to select the most complete one for this thesis.

Azhar, Khalfan, & Maqsood (2009) state that BIM is a revolutionary technology and process that has quickly transformed the way buildings are conceived, designed, constructed and operated. In contradiction to the conventional (3D) CAD systems which describe an AEC project by e.g. lines, arcs and circles and therewith independent views such as plans, sections and elevations, BIM models are defined in terms of building elements and systems such as spaces, walls, beams and columns.

In the research of Love, Simpson, Hill, & Standing (2013) BIM is defined as an emerging technology that can be used to improve the performance and productivity of an asset's design, construction, operation and maintenance process. Both definitions show that BIM is not only applicable to the design and construction phase. This is also supported by Howard (2006): " BIM is a methodology to manage the essential building design and project data in digital format throughout the building's lifecycle." This definition extends the first definitions by using the term lifecycle because now inception up to decommissioning phases are covered as well.

The NBIMS (2007) state that BIM should be a collective knowledge resource by defining BIM as: "a BIM is a digital representation of physical and functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility forming a reliable basis for decisions during its lifecycle."

BuildingSMART is an international acknowledged platform for knowledge exchange regarding BIM. The aim is to improve the exchange of information between software applications used in the AEC/ FM industry. The definition of BuildingSMART (2012) seems to integrate the discussed definitions in the following:

*"BIM is a business process for generating and leveraging building data to design, construct and operate the building during its lifecycle. BIM allows all stakeholders to have access to the same information at the same time through interoperability between technology platforms." (BuildingSMART, 2012).*

This definition is supported by Arcadis (2015) by defining BIM as a process of creating and using one or more (3D) object orientated databases of a construction in its environment, relevant for the design, realization, maintenance and repurposing of that construction during its lifecycle.
The wide range of definition can be caused by the different levels on which BIM is implemented in the industry. These implementations levels are also known as maturity levels, discussed in the next section.

## 3.2 BIM maturity levels

The BIM maturity stages provided a systematic framework for the classification of BIM implementation. The Government Construction Client Group (GCCG) developed the UK Maturity model visualized in figure 3-1. Although many versions exist, the model created by the GCCG is widely used. By defining the levels from 0 to 3, different types of technical and collaborative working are categorized to create a better understanding of the processes, tools and techniques to be used (GCCG, 2011).

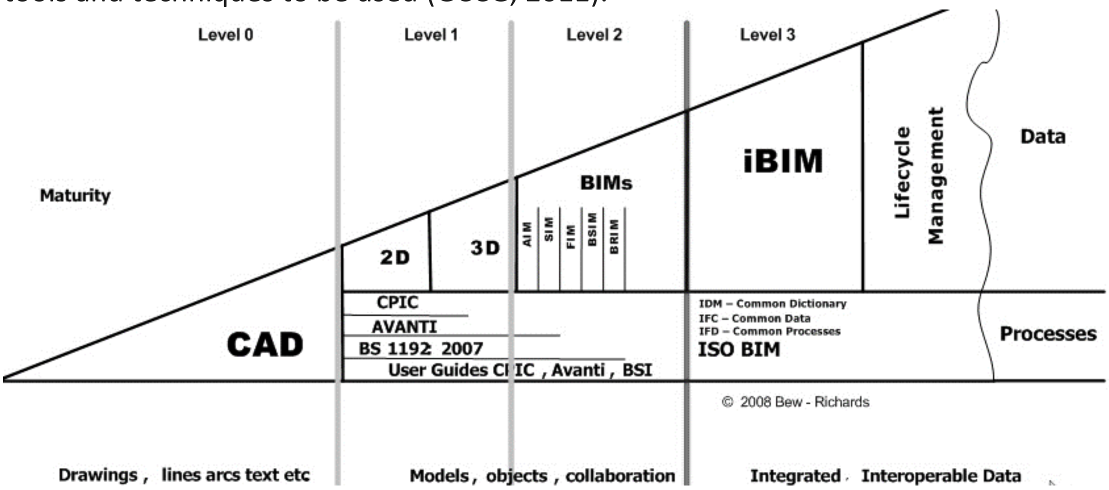



Figure 3-1: UK Maturity Model (GCCG, 2011)

As depicted in figure 3-1 four levels (ranging from 0 up to 3) in BIM implementation can be distinguished. These four levels are defined by Khosrowshahi & Arayici (2012) as pre-BIM (0), object-based (1), model-based (2) and network-based (3). The line on this graph represents the degree of automation and the integration of processes into building project lifecycles.

Level 0, or pre-BIM, is strictly document oriented meaning that everyone works with texts, lines, curves etc. on document level (BIR, 2008). This can be described as a 'non-intelligent' due to the lack of digital objects and therefore labeled as a pre-BIM level.
Level 1, object based, refers to the migration from 2D to 3D and object-based modelling and documentation (Succar, 2009). Thereby, this is the first step of implementing BIM by working with objects. This does not mean only working with 3D , 2D objects can be used and so can objects without any geometric description (BIR, 2008). Clearly-defined objects are used to which information or intelligence can be linked.
Level 2, model based, covers the progresses from modelling to collaboration and interoperability (Succar, 2009). This enables file based co-operation between departments in the same project. Hereby, the possibilities exist to link planning and cost calculations.
Level 3, network-based, is the level at which information between both known and unknown parties can be exchanged (BIR, 2008). This results in less strict project lifecycle boundaries and players interact in real time to generate real benefits from increasingly virtual workflows (Succar, 2009). At the end of level 3, information is shared over the lifecycle in an integrated environment.

## 3.3 BIM applications across the project lifecycle

In this section more insight is gained in BIM applications across a building's lifecycle. A building's lifecycle can be decomposed in project inception, feasibility, design, construction, handover, operation, maintenance and eventual demolition (Eadie, Browne, Odeyinka, McKeown, & McNiff, 2013).
Eadie et al (2013) conducted a survey on BIM use during construction stages in the UK. Results in table 3-1 show the actual application in the AEC/ FM industry per construction stage. The survey was conducted amongst almost 100 professionals within the construction industry. Expectations are that these numbers do not differ significantly in the Netherlands.

Table 3-1 BIM use during construction stages in the UK (Adapted from Eadie et al., 2013 )

| **Use during the construction project stages** | **Often %** | **Occasionally %** | **Never %** |
|---|---|---|---|
| Feasibility | 26.92 | 52.56 | 20.51 |
| Design | 54.88 | 42.68 | 2.44 |
| Preconstruction (Detail design & Tender) | 51.9 | 39.24 | 8.86 |
| Construction | 34.67 | 52 | 13.33 |
| Operation & Maintenance (O&M) | 8.82 | 45.59 | 45.59 |

Table 3-1 illustrates that BIM is most often used in the design (54.88%) and pre-construction stages (51.9%). Application during the construction stage and feasibility studies is even less common, 34.67% and 26.92% respectively. Often use of BIM during O&M stages are rare with 8.82%. BIM maturity levels are not taken into account in this research which could mean that however a large part applies BIM often during design phases low maturity levels are reached in this phase.

### 3.3.1 Application classification

Numerous scientific literature exists concerning the description and classification of BIM applications (e.g. Azhar et al., 2007; Becerik-Gerber et al., 2012 Eastman et al., 2008).
The comprehensive classification of Kreider & Messner (2013) is discussed in this section

because it is generic and the distinguished uses can apply to each phase of a facility's lifecycle. BIM applications are divided in five primary categories: gather, generate, analyze, communicate and realize (figure 3-2). Of these primary categories there are various subcategories that specify the BIM use.


Figure 3-2: The BIM use purposes (Kreider & Messner, 2013).

**Gathering** information can be facilitated by a BIM at various phases during a facility's life (Kreider & Messner, 2013). Secondary BIM uses are qualifying, monitoring, capturing and quantifying. These BIM uses are solely focused on the collection and organization of information.

In a facility's lifecycle almost every stakeholder will **generate** new information. Subcategories according to Kreider & Messner (2013) are prescribing, arranging and sizing facility elements to various levels of development. In the design phase, the engineers and architects will be the primary generators of information. During construction, the (sub-) contractors are the primary information generators. In addition, during O&M those maintaining the facility will generate information when updating or changing a facility.

The **analyzing** purpose of BIM includes those actions in which an examination of the facility elements is needed. Secondary uses are coordinating, forecasting and validating. Prior to this use, previous generated and gathered information is needed that will be analyzed to come to decisions.

**Communication** of facility information as primary use is intended to present information in such a way this information can be shared or exchanged. Visualize, draw, transform and document are the secondary uses. Communication optimization by BIM is often seen as one of the most valuable BIM uses.

**Realization** includes the use in which BIM data is used to make or control a construction element, more and more often without human interaction. Fabricate, assemble, control and regulate are secondary uses of this primary use. Eventually this can lead to improved productivity of both construction and O&M of buildings (Kreider & Messner, 2013).

## 3.4 Interoperability

The goal of full interoperability is far from being realized, in the AEC/ FM sector (Grilo & Jardim-Goncalves, 2010b). Full interoperability must result in a situation in which all different systems are able to communicate with each other using open standards, protocols and procedures. Eastman et al. (2008) defined BIM interoperability as the need to pass data between applications, allowing multiple types of experts and applications to contribute to the work at hand. This is extremely important due to the fact that more than 150 software applications are supporting BIM nowadays (Visser et al., 2013). These software packages have various BIM applications and are used in different phases of a building's lifecycle. A standard file format should ensure optimal cooperation between these software packages and thereby enhanced cooperation between AEC/ FM departments.

There is a need to develop an interoperable file format which is compatible with other software tools (Laakso & Kiviniemi, 2012). In addition to compatibility it is necessary this software can translate a model into an interoperable file format, in such a way that all of the object's information can be transferred correctly. In most cases it is a challenge for such a translation to retain all the information that the model contained in its original native file format (Grilo & Jardim-Goncalves, 2010b).

Due to the many different interactions between the various participants across a building's lifecycle, interoperability dimension is critical for the success of BIM (Grilo & Jardim-Goncalves, 2010b). Because interoperability relies on open standards, this concept is further elaborated and compared with proprietary formats in the next section.

### 3.4.1 Open standard versus proprietary formats

An open standard is a standard with an open standardization process which results in easy accessible documentation, no intellectual property right constraints, open participation in addition to independence and the sustainability of the standardization organization (BIR, 2015). Proprietary file formats are the exact opposite, these are not standardized in an open process and created by specific software developers thereby often only supported by their own software.

Interoperability using an open standard has many theoretical benefits in the application of BIM in the AEC/ FM industry. Without an open standard, each individual software application must develop direct translators back and forth for all other software application to communicate (Laakso & Kiviniemi, 2012). If an open standard is used instead, the application only needs be compatible with this open standard in order to be compatible with all other applications supporting that same standard. In figure 3-3, a graphical representation of this principle is visualized.

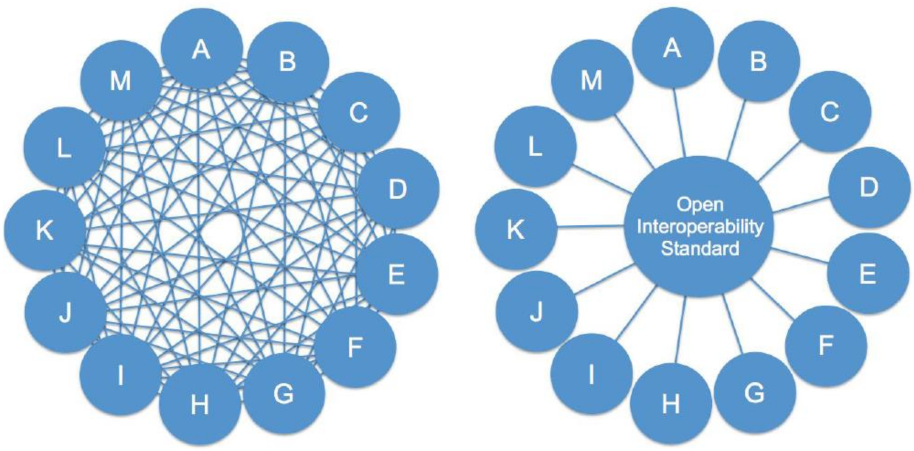



Figure 3-3: Direct translators vs. an open standard (Laakso & Kiviniemi, 2012).

The value of an open standard is supported by the NBIMS (2007), due to the elimination of integrating every application (and version) with every other application (and version). However, exchange of BIM data is dominated by proprietary solutions nowadays. This means in practice that construction projects are based on a solution in which all collaborators have software from the same or compatible vendors (Laakso & Kiviniemi, 2012).

Open BIM is the concept of having all the relevant model information in open formats, making them accessible and readable for anyone, and not locked into proprietary software formats (Hallberg & Tarandi, 2011). Open BIM is an initiative of BuildingSMART and several leading software vendors using the open IFC data model (BuildingSMART, 2012). Amongst the various data model formats, IFC is the only public, non-proprietary data model existing today formally adopted worldwide by different governments and agencies (Gupta, Cemesova, Hopfe, Rezgui, & Sweet, 2014). IFC is further elaborated in section 3.5.

## 3.5 Industry Foundation Classes (IFC)

Industry Foundation Classes (IFC) is an open and standardized data model intended to enable interoperability between building information modeling software applications in the AEC/FM industry (Laakso & Kiviniemi, 2012). The actively ongoing effort of BuildingSMART (formally known as the International Alliance for Interoperability, IAI) to bring together various software vendors and research institutions resulted in a standard that is increasingly accepted by the AEC/FM industry (Beetz et al., 2005).

Since the first definition of IFC in 1996 by IAI, the standard has seen a number of minor and major revisions (Steel, Drogemuller, & Toth, 2012). The latest version is IFC4, however currently the versions 2x2 and 2x3 are still popular.

IFC models contain not only the 3-dimensional geometry of objects, metadata related too many other aspects of the building are included as well. This makes IFC models semantically rich (Steel et al., 2012). While semantics are further elaborated in chapter 4, a simple example is provided: if we consider an instance of a window object, this window will be located in a wall, on a defined building level, within the building. It will have attributes associated with it that describe its thermal performance, price, manufacturer, window type, etc.

A great amount of the significant BIM tools currently used by the industry supports import and export of IFC files (Steel et al., 2012). As stated in section 3.4, it is essential that various analysis tools used in the AEC/ FM domain are interoperable with the non-proprietary open IFC schema (Gupta et al., 2014). However, success depends on the quality of IFC exports, i.e. the mapping between different software packages to IFC. Studies reveal that data losses occur because IFC format-based information exchange fails to provide complete interoperability (Oh et al., 2015; Pazlar & Turk, 2008). Oh et al. (2015) discovered that up to 78.8% of all objects can be lost in the process of exchanging information between IFC and Revit formats. In addition to objects, object properties (e.g., color, grid, layer, location, and view) were lost as well. This unsatisfying model handling is also proven in exchange scenarios using software of other large vendors (Pazlar & Turk, 2008). Continuous improvement of IFC import and export methods are required to minimize data loss.

Although IFC data losses are an issue, the potentials of open interoperability are substantial. It would enable the seamless flow of design, cost, project, production and maintenance information, thereby reducing process inefficiency throughout a building's lifecycle. Laakso and Kiviniemi (2012) consider the IFC effort as one of the most ambitious IT standardization efforts.

### 3.5.1 IFC-EXPRESS

The IFC specification is written according to the EXPRESS data definition language. IFC is accepted as an open international standard in the ISO 16739 (BuildingSMART, 2012).

As stated by Laakso and Kiviniemi (2012) BIM data is intended to be readable, editable, and shared between various systems. Therefore the file structure needs to be standardized. IFC relies on the STEP physical file format. A STEP file consists of two sections, a header section with information about the file itself and a data section with the description of entity instances. All objects in EXPRESS are called entities, with twenty-six defined base entities e.g. geometry, materials, properties (Eastman et al., 2008). By composing these base entities, commonly used objects are defined such as generic walls, floors, structural elements, building service elements, process elements, management elements, and generic features. Because IFC is specified as an extensible data model, the possibility emerges to elaborate base entities if required.

IFC relies on a hierarchical inheritance object structure, the objects are incorporated within an entity tree as illustrated by figure 3-4. Each level of the tree introduces different attributes and relations to an entity. For example, *IfcRoot* assigns a Global ID and other identifier information. *IfcProduct* defines the location of the wall and its shape. *IfcElement* has attributes which define the relationship of this element with others. Many of these attributes and relations are optional, this brings the option to exclude attributes from the export to IFC (Eastman et al., 2008). All independent entities generally contain the attributes 'Global Unique Identifier (GUID)', owner history, name and description.

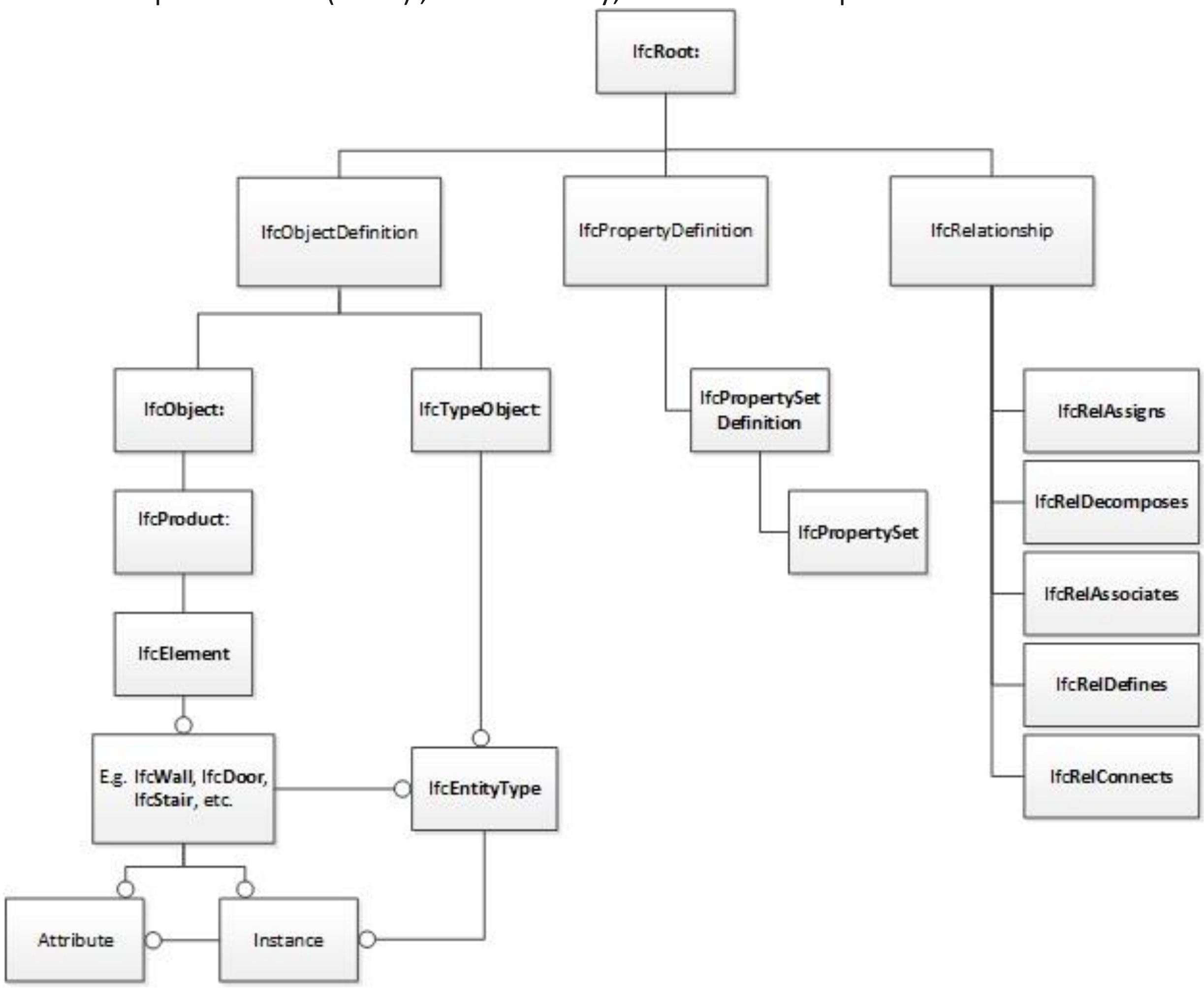



Figure 3-4: Simplified IFC2x3 schema (Heinen, 2015)

Complementary, BuildingSMART extended the scope of IFC-based exchanges beyond the IFC data model. Figure 3-5 shows other BuildingSMART concepts explained in section 3.5.2 and 3.5.3 and 3.5.4.

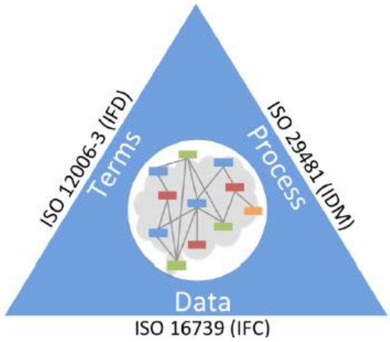


Figure 3-5: The BuildingSMART standards (BuildingSMART, 2011).

### 3.5.2 Information Delivery Manual (IDM)

Information Delivery Manual (IDM) describes a set of process maps, exchange requirements and functional parts, and has been recognized as the key feature that makes IFCs work (Grilo & Jardim-Goncalves, 2010b). An IDM provides a standardization framework to structure the information need for specific tasks of a building's lifecycle. By integrating IDM's, all participants can set when specific information is needed and in what quality.
The main purpose of an IDM is to make sure that the relevant data are communicated in such a way that they can be interpreted by the software at the receiving side (BuildingSMART, 2011). BuildingSMART has an IDM framework available for O&M, however this IDM is still in draft status and therefore not ready to implement in the industry.

### 3.5.3 Model View Definition (MVD)

A Model View Definition (MVD) defines a subset of the IFC schema that is needed to satisfy one or many exchange requirements (BuildingSMART, 2011). Often the complete IFC schema is not required, a MVD provides only the relevant information for a specific AEC/ FM department or task. The MVD acts like a filter of the IFC data schema to obtain only the required information for a specific purpose. For O&M purposes, Construction Operation Building information exchange (COBie) is the international standard to exchange contact an general building information as well as information about spaces, floors, zones, components, technical systems and equipment (Volk et al., 2014).

### 3.5.4 BuildingSMART Data Dictionary (bsDD)

The BuildingSMART Data Dictionary (bSDD) is a terminology standard for BIM libraries and ontologies (Volk et al., 2014). The bsDD, with its former name 'International Framework for Dictionaries (IFD)', is an ISO 12006-3 based ontology for the AEC industry to connect information from existing databases to IFC data models (BuildingSMART, 2013). The bSDD currently contains over 80k concepts along with approx. 200k natural language names and descriptions (DURAARK, 2013). Using bsDD, modeled objects are separated from its name and language and described using an ontology for the definition and storage of building model objects that can be reused on different projects (Steel et al., 2012). This helps to prevent ambiguities such as the example of the Norwegian "dør". The word "dør" in Norwegian is in a normal dictionary translated to "door" in English. However, the word "dør" refers to the door with its frame, while "door" in English only referrers to the door itself. When translating "dør" it should be "door set" in English. This simple example illustrates a critical problem which has to be overcome for BIM- based object oriented working methods.
The bsDD concepts relate to other concepts with objectified relationships and are assigned

with Global Unique Identifiers (GUID), names and descriptions. While not all stakeholders in every lifecycle phase of a building will use all described information, a unique and identifiable object can be created (figure 3-6).

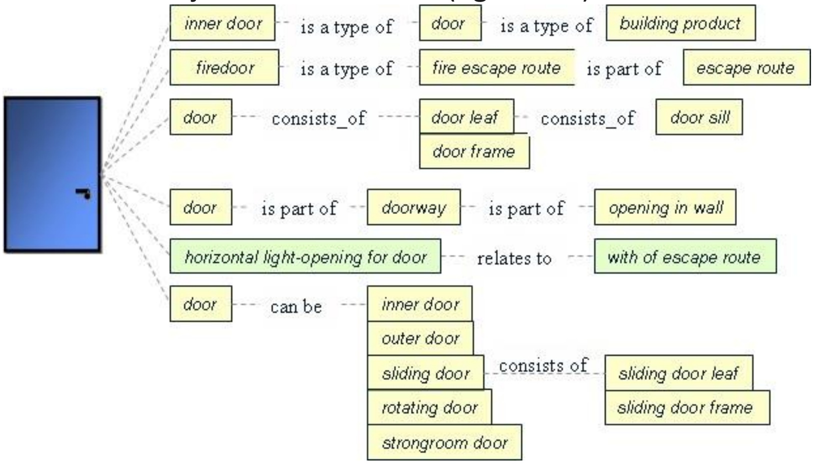



Figure 3-6: Concept with relationships (BuildingSMART, 2009)

Depending on the use case, an object view can be instantiated showing only the desired bsDD context. Figure 3-7 illustrates how a window concept can be described by a set of characteristics. These characteristics can originate from e.g. a window suppliers object specification. The different contexts of a window are illustrated in figure 3-7.

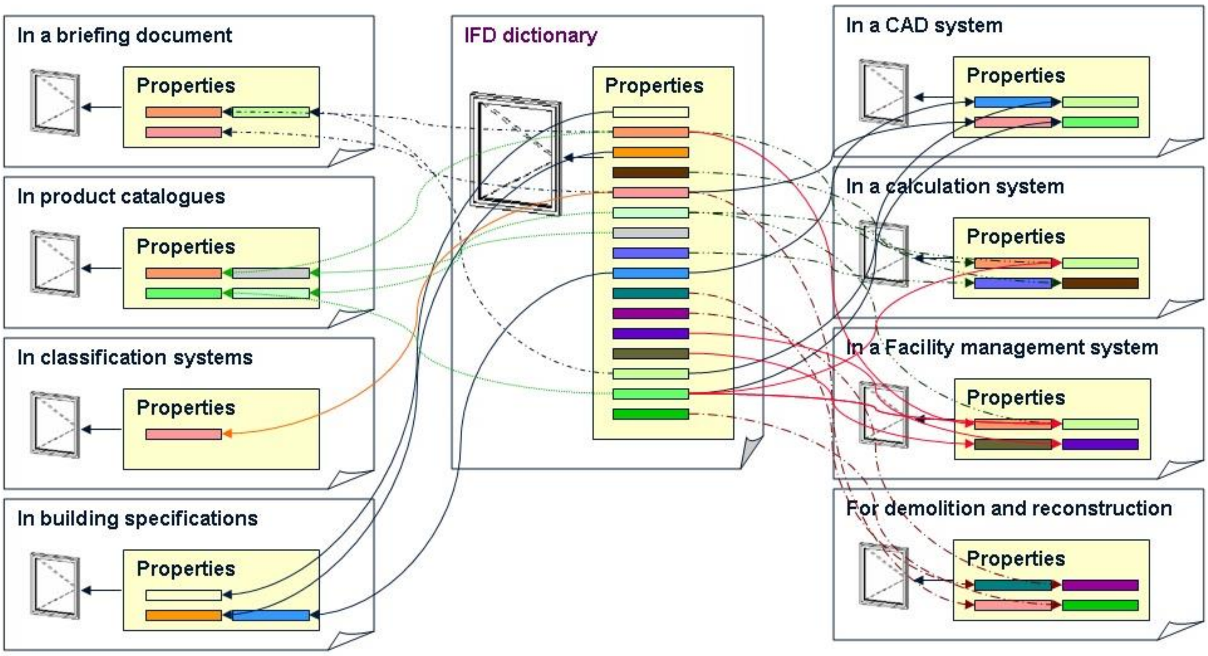



Figure 3-7: BsDD (IFD) as a mapping mechanism(BuildingSMART, 2009)

The bsDD does not describe instances of e.g. a window of a specific project or from a product database. This obligation is fulfilled by a BIM model in, for example, IFC format. The BuildingSMART Data Dictionary bsDD is being developed with an emphasis on buildings (Nederveen, Luiten, & Böhms, 2015). Infrastructural assets are not covered yet.

## 3.6 Classifications & Concept Libraries

Classifications are used to classify collections according to their similarity and providing a hierarchal structure to these collections. For interoperability within an object oriented working method, classifications are key. As stated by Pauwels et al. (2011) a proper classification framework is the key to manage and provide access to information. By annotating 3D models semantically with non-geometric information an improved search over the information is possible. For example, good classification enable informational model queries without being hindered by naming ambiguities such as 'tilting window' and 'window' because they belong to a similar class. Thereby, a standard classification implementation within a BIM model will allow same information sorting and retrieval across multiple platforms and at any stage in the building's lifecycle. In section 3.6.1 some common Dutch classifications are discussed. Thereafter in section 3.6.2, concept libraries are discussed which can basically be seen as a more extensive form of object classification.

### 3.6.1 Object Classification

According to ETIM (2015) object classification is simply a logical, unambiguous classification (taxonomy) of products in different product classes, designed so that anyone within the sector can communicate about those products without misunderstandings. Currently, many classification systems are developed both on national as international scale. The ISO 12006(-2/3):2015 defines a framework for the development of built environment classification systems. It identifies a set of recommended classification table titles for a range of information object classes according to particular views. While not common in the Netherlands, internationally used classifications based on this ISO standardization are e.g. Omniclass, DBK and Uniclass. A number of frequently used classification systems in the Netherlands are:

- **NL/SfB:** Classification method to group and code semantic building information during its lifecycle. It originates from the Swedish SfB- system published in 1947 (STABU, 2015). In total the NL/SfB consists of five parts e.g. encodings for a building's environment, functional parts and construction materials.
- **ETIM international:** The European Technical Information Model (ETIM) gives a listing of the most important technical characteristics of product classes to describe and find the products (ETIM, 2015). ETIM finds its application in the installation, construction and maritime sectors and is intends to share product classes clearly on an international scale.
- **IMGeo:** Information Model Geography (IMGeo) forms the standard for exchange of 3D geo-information and contains agreements about the exchange. This includes agreements relating to the legally compulsory Basis Registration Large-Scale Topography or BGT (Geonovum, 2015).
- **STABU BWBRD: '**STABU Bouwbreed' is a pragmatic classification for the technical description of entire building systems and installations (STABU, 2015). This classification system contains six modules and thereby applicable in each lifecycle phase of a building.
- **NEN2767:** Classification and standardization of inspection of the technical state of construction components introduced by the Netherlands Standardization Institute (NEN). Contains different parts of which the first defines a Methodology for measuring the condition of building and installation parts (NEN 2767-1). Second, the

NEN 2767-2 includes defect lists for measuring the condition of building and installation parts. Then, NEN 2767-3 is an aggregation of condition scores for measuring the condition of building and installation parts. Finally, NEN2767-4 is the standardization of the condition assessment for infrastructure specifically. In addition, it provides a breakdown of elements and components of infrastructural works and associated defect lists (NEN, 2010). The aggregation of condition assessment scores of management objects and areas is still under development.

### 3.6.2 Concept library

An concept library or also referred to as Object Type Library (OTL) is a library containing generic reusable objects covering the complete building lifecycle (LDAC, 2014). A BIM describing specific construction objects in various stages of its lifecycle is composed of data specific for this project. However, some information has a more generic nature and can be of use for the category the construction object belongs to. Therefore, this kind of category specific object information could be stored in a concept library that can be referenced by other BIM's that hold construction objects of the same category (Hoeber, Alsem, & Willems, 2015). By doing so, an enhancement of semantics is offered by concept libraries due to the reusability of data.

The AEC/ FM industry faces the challenge of improving effectiveness and efficiency of their processes in the areas of project management, FM and network management where information management seems to be crucial (Nederveen et al., 2015). Ideally, information that has been created at a particular location would be used and reused by other actors at other locations, inside or outside an organization.

In the Netherlands a nation-wide concept library with the name CB-NL is developed. In addition, the Dutch ministry of Infrastructure (Rijkswaterstaat) develops a concept library for its asset types (e.g. highways and waterways), while the Dutch railway authority (ProRail) is developing a concept library specific for railways (Hoeber et al., 2015). Eventually these infrastructural concept library initiatives will be harmonized by mappings to the nation-wide CB-NL. While the CB-NL is still in a conceptual stage it creates great object standardization and associated data reusability opportunities (Dankers et al., 2014). The CB-NL has adopted a linked data approach (Nederveen et al., 2015). Thereby not only combining before mentioned concept libraries but a great number of distinct standards, databases, and ontologies, including STABU, NET, NLCS, BID CROW, ImGeo, AQUO, INSPIRE, CORA RioNED, ETIM, and so forth (LDAC, 2014). This combination effort by CB-NL is illustrated in figure 3-8, where some standards have distinct mappings directly between them, they can all be connect by the nation-wide concept library CB-NL. It relies on links between the diverse ontologies, and will become a dedicated context in the BuildingSMART Data Dictionary (bsDD) (LDAC, 2014).

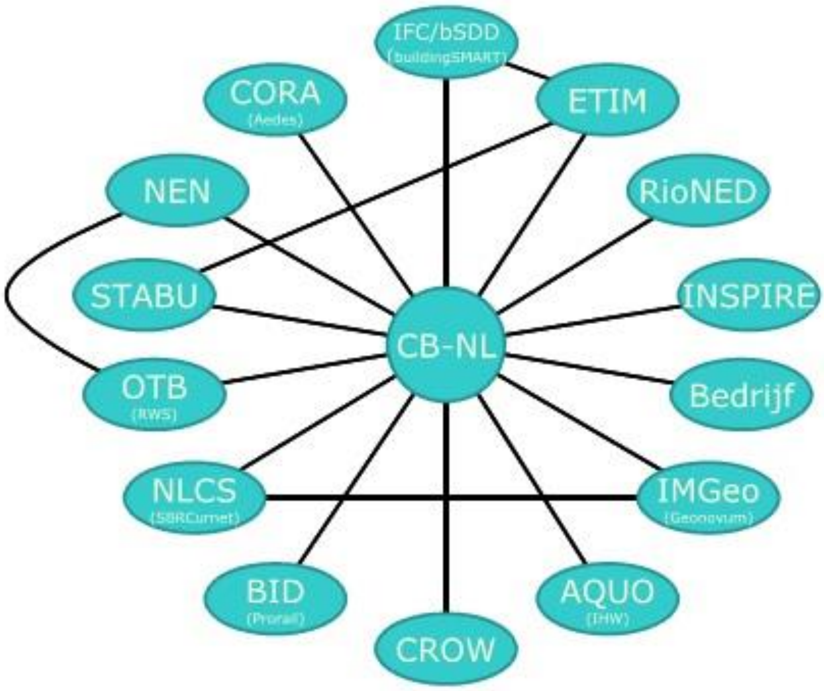



Figure 3-8: CB-NL (LDAC, 2014).

## 3.7 Level of detail (LOD) and level of information (LOI)

Level of Detail (LOD) is defined by Volk et al. (2014) as the geometric and non-geometric attribute information provided by a model, commonly referring to a point in time, lifecycle stage or to a contractual responsibility. Different levels of detail are defined to enable specific informational requirements to be set for e.g. a project phase. Love et al. (2013) defines LOD as an indicator of how much information is known about a model at a given time which increases in richness as an asset progresses throughout each project lifecycle phase. Generally, Levels of Detail values range from LOD100 to LOD500 with increments of 100. On this scale LOD100 contains only spatial objects or masses ranging up to LOD500 representing the as-built condition and is configured to a central data storage for integration into maintenance and operations systems (Love et al., 2013). For maintenance functionalities, the Construction Operations Building information exchange (COBie) standard defines a LOD for technical equipment, regarding type and location, make, model and serial numbers, tag, installation date, warranty and scheduled maintenance requirements (Volk et al., 2014).

Level of Information (LOI) refers to all non-graphical content of a model at a point of time or lifecycle stage. While scientific literature commonly only refers to LOD, LOI is a part of this broader definition. Hence, LOI is a proper subset of LOD thus:

$$LOI \subset LOD$$

Due to building complexity high levels of information for its facility management are required. This information is essential and thereby required to be available as highly accurate as possible (Lavy & Jawadekar, 2013).

During design phases architects and other engineers preliminary produce graphical data whereby the amount attribute data increases in time. Most valuable O&M information is added during construction to obtain an as-built situation (Verbaan, Visser, Koe, Boer, & Voet, 2014). For an asset manager BIM attribute data is more important than the graphics and visual data of the as-built model (Visser et al., 2013). This principle of proportional shift and transfer of information over time is illustrated in figure 3-9.

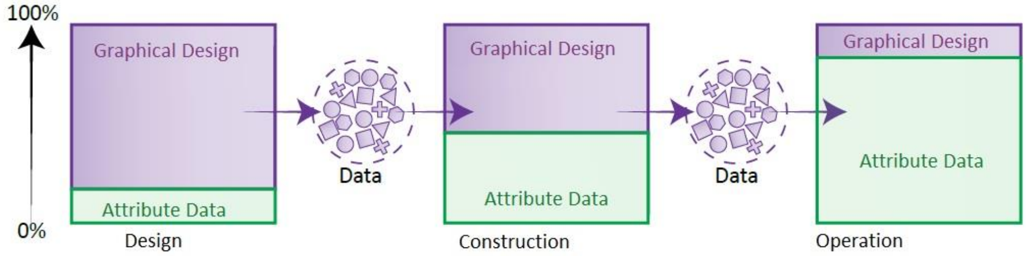



Figure 3-9: The significance, ratio and transfer of data per phase (Verbaan et al., 2014)

# 4 SEMANTIC MODEL ENRICHMENT

While BIM models are elaborated to a high level during design and construction of a building, they do not have to represent the 'as-built' situation. The actual as-built conditions can differ significantly from the design, and the as-used conditions can change extensively throughout a building's lifecycle (Huber, Adan, & Okorn, 2011). Therefore, 'as-designed' BIM models needs to be adjusted to create 'as-built' models. In addition, a lot of external data is available that is to some extend related to a specific building. Missing, obsolete or unstructured building information might result in ineffective project management, uncertain process results and time loss or cost increases in maintenance, retrofit or remediation processes (Volk et al., 2014). Koukias, Nadoveza, and Kiritsis (2013) consider management of a building's data as key to achieve optimization of FM.

A semantic model is a model in which semantic information is included describing the meaning of its instances. As discussed in chapter 3, IFC models can be considered as semantic models. Generally, diverse models or representations of a building are available with a vast amount of external data resulting in the need for distributed data management (figure 4-1). Additionally, datasets are distributed over various places and in possession of different stakeholders making the challenge only more complex. Hence, development towards a Web of Data becomes more relevant for the AEC/ FM industry (LDAC, 2014).

First, in section 4.1 the term semantic enrichment is elaborated. Thereafter in section 4.2, the semantic web is discussed in more detail and some relevant terms are defined. Finally, in section 4.3, approaches of semantic IFC enrichment are introduced and discussed in more detail.

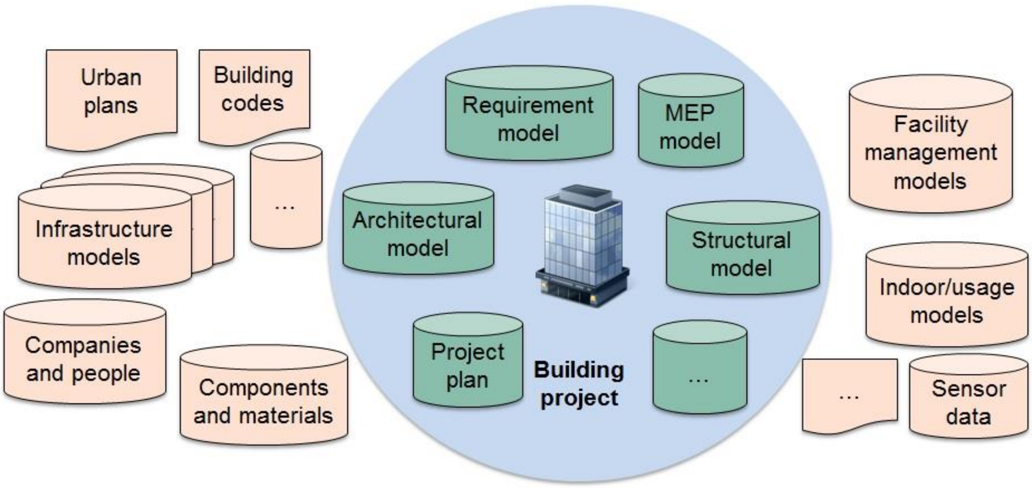



Figure 4-1: Distributed data management (LDAC, 2014)

## 4.1 Semantic enrichment

Semantic enrichment is defined by IBM (2015) as elaborating the content/context of data by tagging, categorizing, and/or classifying data in relationship to each other, to dictionaries, and/or other base reference sources. This means adding additional contextual information to some existing data set. Thus, semantic enrichment also includes the reference of external information resources, e.g. from building regulations, classification systems, or product data (Beetz, Dietze, Berndt, & Tamke, 2013). Due to the changing state of the built environment, this process of enrichment of existing datasets with knowledge and information should be an iterative process during the whole building's lifecycle.

Semantic enrichment of building models is defined by Belsky et al. (2015) as a process in which an expert system 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. The semantics of a building object are composed of its form, function, and behavior (Lee et al., 2006). These semantics are represented by its shape (3D geometry), material and mechanical properties, functional classification, topological and aggregation relationships with other objects, all of which have particular meaning. The relative locations of objects to one another are key determinants for their functional classification and topological/ aggregation relationships.

## 4.2 The semantic web

The semantic web is introduced in 2001 with the aim of turning the current web into a "web of data", thereby eliminating unstructured and semi-structured documents (Berners-Lee, Shadbolt, & Hall, 2006). Basically, the idea behind the semantic web is to add semantic metadata to the existing data in order to describe data content and their relations in a way so that the meaning of data can be processed by machines. Common linked data technologies are RDF, SPARQL, OWL, and SKOS (W3C, 2015b). The World Wide Web Consortium (W3C) has directed major efforts at specifying, developing, and deploying languages for sharing meaning. These languages provide a foundation for semantic interoperability (Berners-Lee et al., 2006). The need has increased for shared semantics and a web of data and information derived from it (Berners-Lee et al., 2006). Various industries have recognized the value semantic web applications, including the AEC/FM industry.

By using semantic web technologies, BIM models can be linked with external information from other ecosystems (product catalogues, libraries of design elements, public procurement requirements, etc.) on the internet (Costa & Madrazo, 2015).

According to Berners-Lee et al. (2006) four main rules must be taken into account for semantic web application. First, tings must be identified with Uniform Resource Identifiers (URIs). Without application of universal URIs data is not a part of the semantic web. The second rule is to integrate HTTP URIs to enable people to look up specific information connected with an URI. The third rule states that information connected to an URI should be useful in terms of well-formed ontologies, thereby structuring data following e.g. RDF, RDFS and OWL. The fourth and last rule defines the importance of linking to other URIs, thereby creating a web of information.

Linked Open Data (LOD) is Linked Data which is released under an open license, thereby not limiting free usage (W3C, 2015b). The five star Linked Open Data model is commonly mentioned in the literature, classifying the level of LOD integration. This model contains the following levels:

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

A key strength of the semantic web is the ability to create new links between data automatically, referred to as inference. Inference can be characterized by discovering new relationships due to the modelling of data as named relationships between resources. These new relationships could be added to the set of data or returned by a query. The semantic web principles can be the key to harmonization of information models, due to the ability to let software agents understand the meaning of data and create connections between data automatically thereby gaining new information. Based on this vision, ontologies can be used to capture the semantics of data, resolve semantic heterogeneities and optimize data quality and availability (Koukias, Nadoveza, & Kiritsis, 2013b). The semantic web stack illustrates the hierarchy of languages where each layer uses parts of the layer below (figure 4-2). To get a better understanding of the semantic web and current developments in the AEC industry, ontologies, Resource Description Framework (RDF) and IfcOWL are further elaborated.

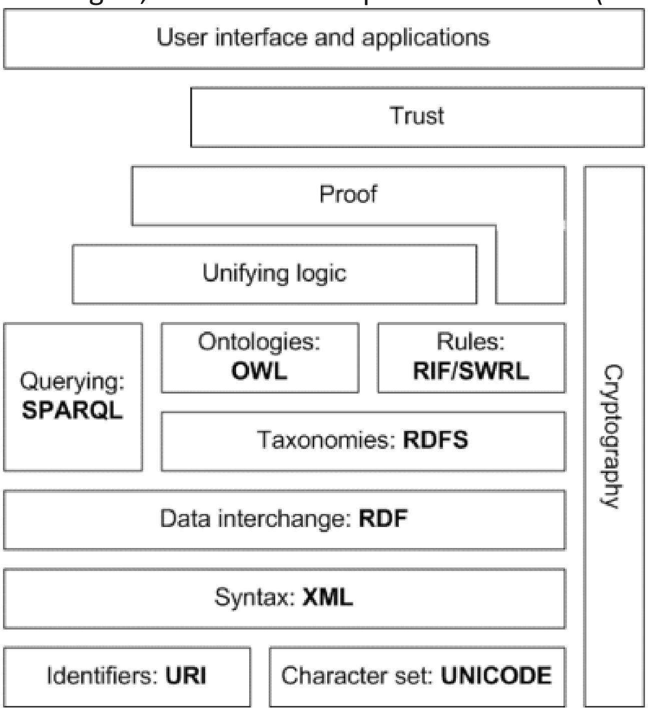


Figure 4-2: Semantic web stack (Berners-Lee et al., 2006).

### 4.2.1 Resource Description Framework (RDF)

Semantic web information is represented by triples, which are Resource Description Framework (RDF) expressions composed of subjects, predicates, and objects. By semantically linking all kinds of objects and subjects (resources) using predicates, large clouds of Linked Data can be created. This information is stored in RDF triple stores, which is a specific kind of graph database (Dimyadi, Pauwels, Spearpoint, Clifton, & Amor, 2015).

Figure 4-3 represents an RDF graph example containing four triples with information about a person. This RDF example includes various kinds of information. First, it contains an individual, in this case ‘Eric Miller’. Second, the RDF states that Eric Miller is a person indicating the type of ‘thing’. Third, properties of those ‘things’ are indicated such as the ‘mailbox’ property. And finally property values are identified which in this case is the email address for example. In terms of subject, predicate and object one instance from figure 4-3 is ‘me’, ‘fullName’ and ‘Eric Miller’ respectively.

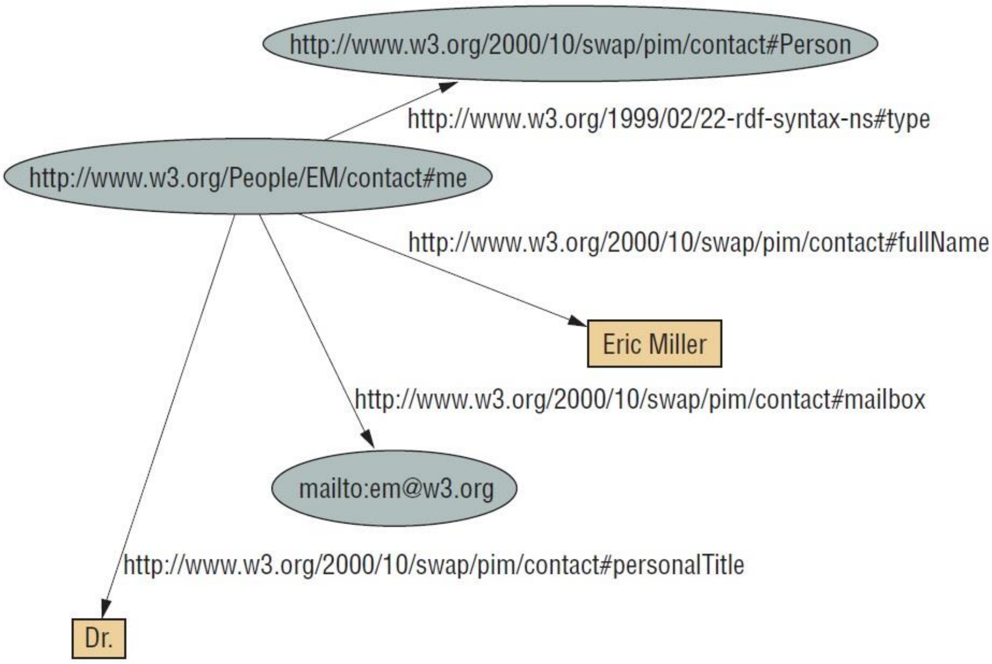



Figure 4-3: RDF graph representing Eric Miller (Berners-Lee et al., 2006).

By application of RDF specific Universal Resource Identifiers (URIs) are assigned to individual fields. URIs have a global scope, associating an URI with a resource means that anyone can link to it, refer to it, or retrieve a representation of it (Berners-Lee et al., 2006). Because RDF statements can be diagrammed as a directed graph representing facts, explicit links can provide an unambiguous reference that may refer to data specified in other graphs. Thereby, creating a network of linked data available for any application (Costa & Madrazo, 2015).
The primary query language for RDF graphs is SPARQL (SPARQL Protocol and RDF Query Language) able to retrieve and manipulate subgraph information (Dimyadi et al., 2015). SPARQL can be deployed across various data sources, whether the data is stored natively as RDF or viewed as RDF. In addition, it is possible to test values and query constrains by source RDF graph.

### 4.2.2 Ontologies

Ontologies define the concepts and relationships used to describe and represent an area of concern (W3C, 2015b). These common conceptualizations are also referred to as vocabularies (Berners-Lee et al., 2006). There is no obvious distinction between what is referred to as 'ontology' and 'vocabulary'. Generally, ontology is used for more complex and formal collections of terms, whereas vocabulary is used when such strictness is not necessarily important (W3C, 2015b). Ontologies are to improve data integration when ambiguities could exist on terms used in data sets or when extra knowledge can lead to the discovery of new relationships. Ontologies can capture the semantics of data, describing the knowledge for sharing in a specific domain and provide reasoning capabilities (Koukias et al., 2013b). Markup ontology languages are used to encode knowledge. The most used language created by the World Wide Web Consortium to describe ontologies in a formal way is the Web Language Ontology (OWL) (Costa & Madrazo, 2015). Providing a formal and explicit specification of a shared conceptualization defined by means of classes, attributes, values, relationships, roles and rules. The OWL recommendation is now at version 2.0 (OWL2) extending the capabilities of version 1.0. The OWL specification integrated several efforts.

The W3C recommendation presents various versions of OWL, depending on the degree of expressive power required (Berners-Lee et al., 2006). These different OWL profiles are illustrated in figure 4-4. OWL uses the Recourse Description Framework Schema (RDFS) concepts as subset to describe ontologies. Subsequently, RDFS took the basic RDF specification and extended it to support structured vocabularies. RDFS provides a minimal ontology representation language that the research community has adopted fairly widely (Berners-Lee et al., 2006). To conclude, RDF graphs can obtain an improved semantic structure using vocabularies or ontologies with the most basic elements to describe such ontologies available in RDFS vocabulary (Pauwels et al., 2011).

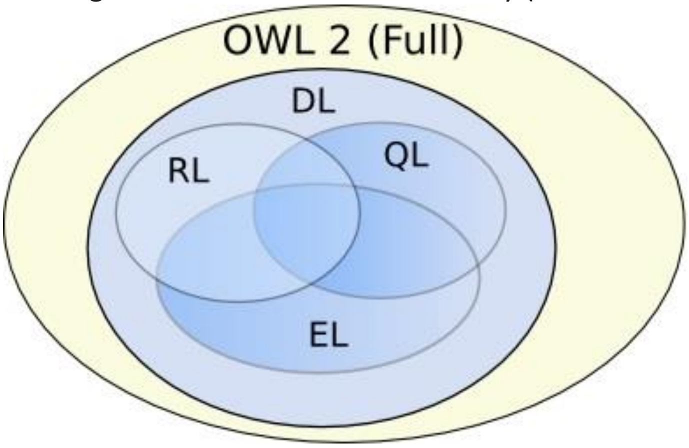



Figure 4-4: The various OWL2 profiles (LDAC, 2015)

### 4.2.3 IfcOWL

As discussed in section 3.5 the Industry Foundation Classes (IFC) model standard enables the exchange and representation of building data in a more interoperable way. However, the lack of mechanisms to extend the semantics of the model is identified as one of its major limitations (Costa & Madrazo, 2015). Therefore, Beetz, Van Leeuwen, and De Vries (2009) introduced an ontology for the building and construction sector based on IFC. By the introduction of semantic web technologies, IFC model information based on EXPRESS is proposed to be transformed into a semantically enhanced model encoded in OWL. This OWL version of the IFC schema named ifcOWL makes it possible to use semantic web technologies for BIM models. Amongst others, the benefits of transition to semantic web technologies and ifcOWL specifically are the ability to link different types of datasets of the same concept or elements, querying of data, publishing of data and reasoning with data (LDAC, 2014). As comparison, figure 4-5 illustrates the difference in approach in terms of “Old School” and “New School” of data management. The “Old School” approach is based on many technologies (STEP, XML, etc.) where one data structure is selected and applied. This central approach is relatively inflexible and static. In contrast, the “New School” approach has its fundaments in one logic-based technology which is the semantic web. Thereby, the ability emerges to interlink many data structures. This decentralized approach can be characterized as more flexible and dynamic.

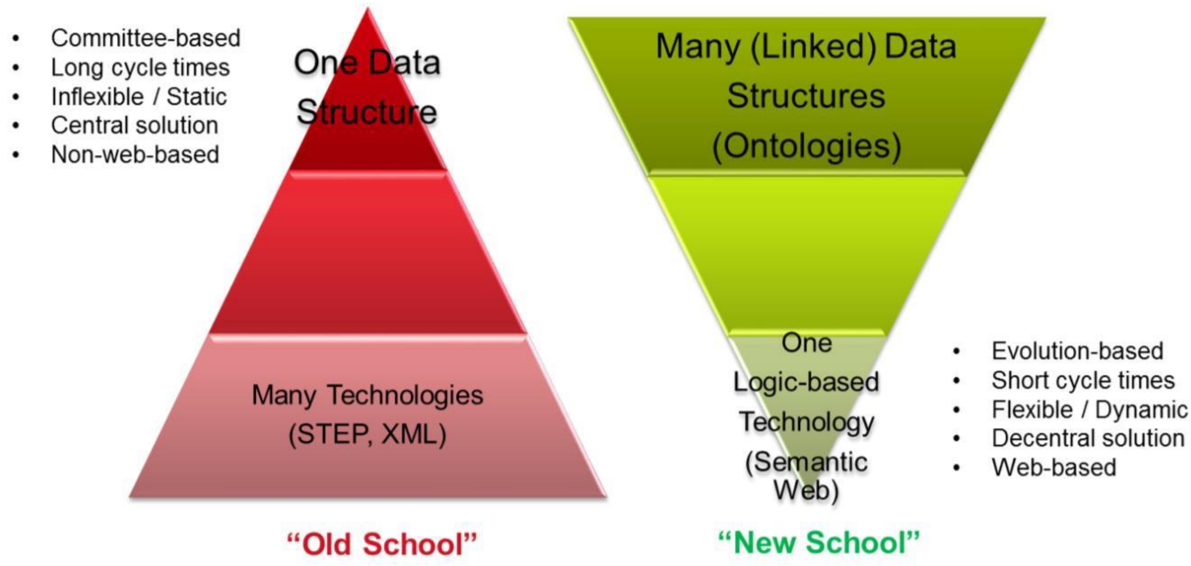



Figure 4-5: Opposition between traditional techniques and new techniques (LDAC, 2015)

Considering the features of the different OWL2 profiles as visualized in figure 4-4, multiple opportunities are available regarding the IFC conversion. On the one hand, one could aim at an ifcOWL ontology with as much type information as possible (high expressiveness). On the other hand, one could aim for an ifcOWL ontology that is in one of the less expressive profiles, but is more efficient at execution time (LDAC, 2015).
In order to standardize transformation rules, the development of ifcOWL is managed by a standardization body. The BuildingSMART Linked Data Working Group is responsible for building and maintaining the ifcOWL ontology. The group meets at regular intervals, both virtual and live, to keep track of and discuss possible ifcOWL ontology enhancements (Pauwels, Törmä, Beetz, Weise, & Liebich, 2015). The Linked Data Working Group is part of the Technical Room of BuildingSMART and closely interacts with the other working groups within the BuildingSMART organization. This work has resulted in a draft ifcOWL ontology.

## 4.3 Semantic enrichment of IFC

As stated, IFC models are semantically rich because they capture not only the 3-dimensional geometry of an object, but metadata related to many other aspects of this object and the building as a whole. Objects within the model will have attributes associated with it that describe its thermal performance, costing, fire safety performance, etc. (Steel et al., 2012). As discussed in section 3.5, IFC files are made of objects and connections between these objects. Object attributes describe the "business semantic" of the object. Connections between objects are represented by relation elements, for example *IfcRelDefinesByProperties*. Generally, a great amount of information can already be found in the semantically rich IFC model developed during a building's design and construction. Ranging from provenance data such as authorship, creation date and stakeholder roles for all instances descending from the *IfcRoot* class to information of individual components such as material and configuration properties (Beetz et al., 2013). Individual component metadata is generally stored in *IfcProperties* through *IfcPropertySets* included into the IFC model capabilities. This provides the possibility to extend and adjust object metadata within the model itself.

While some relevant O&M metadata is already present in the model by handover from design and construction phases, other data has to be added manually or through automated procedures by the asset manager. In such data networked structures, this process, referred to as semantic enrichment also includes the reference of external information resources e.g. from building regulations, classification systems, or product data (Beetz et al., 2013b). For O&M, this practically would mean that additional metadata is added or updated while geometry data would be adjusted less frequently. In the research of Dankers et al. (2014), researchers have chosen to enrich an IFC model by adding hyperlinks objects and thereby referencing to external information. An advantage of this IFC enrichment method would be that data in a web database is much easier to update to new or updated datasets then tags in an IFC property. However, this method still requires the manual interaction of adding the hyperlink itself to every single object in the model which could become a significantly large task for extensive models. Though, many of the significant BIM tools currently used by industry support import and export of IFC file (Steel et al., 2012). Thereby, by adding either metadata directly into the IFC model or indirectly by adding hyperlinks to the location of the associated data, a usable model for O&M can be the result.

Following the semantic web principle elaborated in section 4.2, with the well-known statement of “Anyone can say Anything about Anything” (AAA), relying on links between resources and is therefore a more decentralized situation. As stated earlier, additional arbitrary links between data resources can be created if the resources can be identified by an URI. In contrast to IFC, this does not require elaborate changes in the schema but is inherent to all RDF data sets (Beetz et al., 2013b). While the efforts of developing ifcOWL making good progress and certainly have potential of future adoption and implementation in the industry, currently the ontology is still in a conceptual stage. To overcome this issue, a transitional approach is suggested by the DURAARK project. DURAARK (Durable Architectural Knowledge) is funded by the European Commission with the collaborative aim of developing methods and tools for semantic enrichment and long-term preservation of architectural knowledge and data. The transitional approach implies simple implementation agreements whereby IFC models can be semantically enriched with arbitrary linked datasets without raising compatibility issues with existing commercial-of-the-shelf tools (DURAARK, 2013). In other words, the IFC model remains valid and can still be understood by available IFC viewers. Besides the current challenges of ifcOWL and its current conceptual stage, transformation of IFC models or partial sub-model chunks itself into RDF demand a considerable shift in technologies and would require an even higher implementation effort than schema extensions (Beetz, Coebergh, Botter, Zlatanova, & de Laat, 2015).

As graphically illustrated in figure 4-6, the transitional solution proposed by DURAARK (2013) allows the combination of STEP-based IFC models with RDF data. The suggested approach allows much more rigid semantics using a wide range of methods and technologies that can be applied and integrated into existing processes and tools.

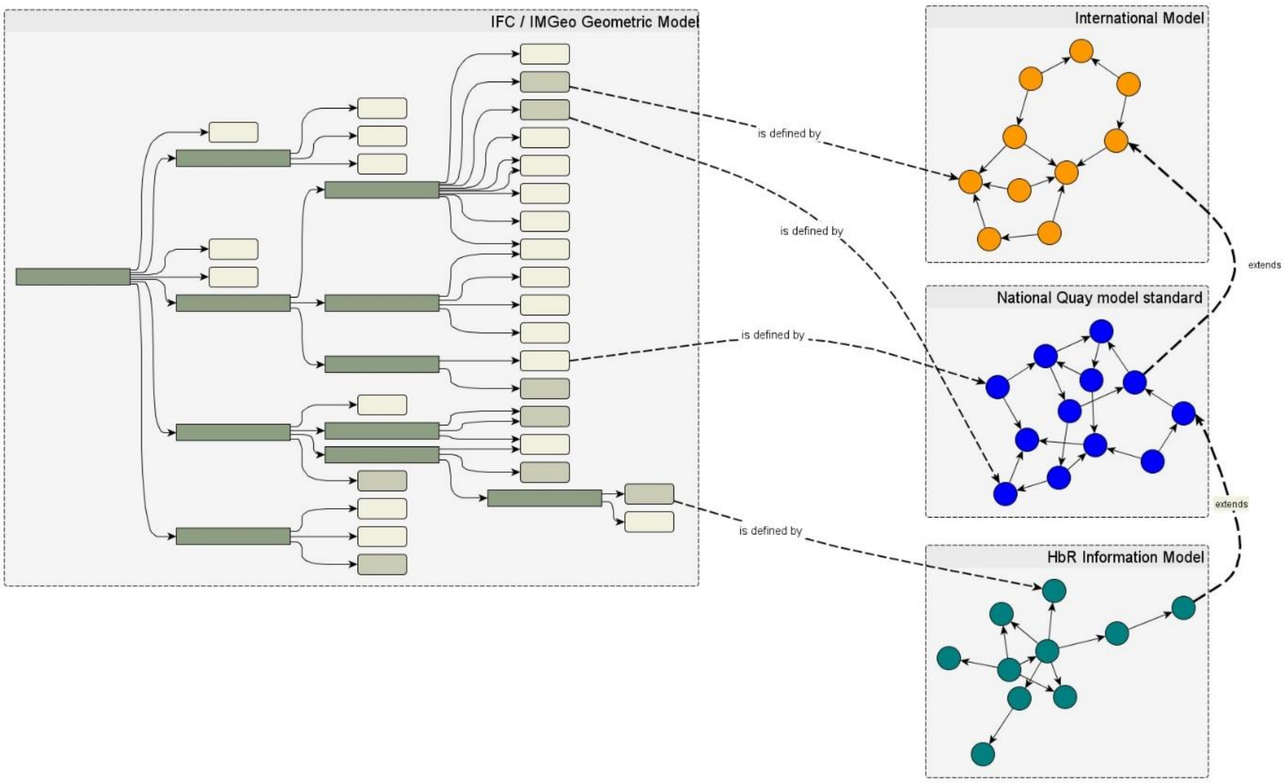



Figure 4-6: Web of Data technologies (LDAC, 2014)

# 5 RISK-BASED OPERATION AND MAINTENANCE

In this chapter, risk-based operation and maintenance (O&M) is further elaborated to get a better understanding of the principle and application possibilities. Dawood, Vukovic and Kassem (2015) point out that the visualization capabilities of BIM and their role in decision making for O&M tasks are receiving an increased attention, therefore risk-based O&M is further investigated. First, O&M strategies and performance based contracts are discussed in section 5.1 to introduce what incentives led to the application of risk-based O&M within the industry. Then in section 5.2, risk-based O&M is defined after which the Failure Mode Effect and Criticality Analysis (FMECA) methodology is further elaborated.

## 5.1 Operation and maintenance strategies

Building's operations and maintenance (O&M) includes all services required to assure the built environment will perform according to the functions for which a building was designed and constructed (WBDG, 2015). Typically O&M covers the day-to-day activities necessary for the building, its systems and equipment to perform following their intended function. For this research, operations and maintenance are combined into the common term O&M because a building cannot operate efficiency without being maintained.

Optimally, maintenance is carried out as little as possible while preserving the availability of a building according to the intended function. This means maintenance of its components is carried out infrequently. In other words, maintenance should be carried out only when necessary to ensure the continued, safe and profitable use of the building at acceptable levels of satisfaction or when there is the possibility of extending the useful life of the elements of the building (Horner, El-Haram, & Munns, 2001).

In maintaining a building, there are three basic strategic options available to the owner. There is the possibility of reducing demand for maintenance by addressing the actual cause of failure and identifying its consequences. In addition, it could be necessary to decide whether to repair or replace an item, and whether to carry out periodic maintenance at fixed intervals or simply to respond to the requests of the users. Horner et al. (2001) states that building maintenance can be divided into three basic strategies:

- Corrective maintenance;
- Preventive maintenance;
- Condition-based maintenance.

Corrective maintenance implies maintenance activities performed when an element actually has broken down. This includes the repair or replacement of an element that has failed or in other words, cannot perform its required function. Sometimes corrective maintenance is referred to as failure-based or unplanned maintenance.

Preventive maintenance includes all maintenance activities which reduce the probability of occurrence of failure and avoiding sudden failure. This type of maintenance is referred to as time-based maintenance, planned maintenance or cyclic maintenance (Horner et al., 2001).

Condition-based maintenance are the maintenance activities performed based on significant deterioration. Condition-based maintenance is performed when a change in condition and/or performance of an item is the principal reason for carrying out maintenance. Therefore, condition assessments must determine the actual condition of the building elements.

While the former strategies are basic but fundamental Frugugliettі, Pasqualato and Sagula (2012) are stating that due to the continuous technological evolution, the concept of maintenance has been renewed and there can be a shift identified from repair activities and an ad-hoc approach, to a more complex system of management, directed towards the prevention of faults and continuous improvement. Thereby, the task of maintenance is to cooperate throughout the whole lifecycle of a building, from design to deconstruction, with the aim of continuous improvement of operational availability and reduction of maintenance cost (Frugueglietti et al., 2012). However the current situation in the industry is that AEC and FM are still dissociated. The teams which are responsible for the FM processes are rarely those that have also participated in the design and construction of the building (Vanlande et al., 2008).

### 5.1.1 Performance based contract (PBC)

Performance based maintenance contracts (PBCs) give maintenance suppliers incentives to improve their way of working (Straub, 2009). This contracting method is adopted more and more by the industry. Due to the focus on performance of an asset and not on maintenance actions directly, contractors are stimulated to be more innovative and cost-effective to meet the performance criteria. The PBC defines the desired outcomes or results in its purest form and thereby does not detail how, when, or where to do the work (NCHRP, 2009). Fundamentally, the goal of a PBC is to pay a contractor based on the results achieved, not on the methods for performing the work. In a performance-based maintenance relationship, performance criteria are explicitly stated by the client.

Measures of performance are often expressed in terms of levels of service represented by specific rating scales. It enables maintenance contractors to assume responsibility for maintenance activities an provides room for own interpretation and a custom maintenance planning (Straub, 2009). However, in practice this is really challenging for those accustomed to standard procurement procedures such as low bid, best value, and qualifications-based selection, etc. (NCHRP, 2009). Though, the findings of Straub (2009) show that performance-based maintenance contracts reduce both direct and indirect costs compared to a competitive tendering approach. As a result to the more frequent application of PBCs, contractors are continuously improving their maintenance activities amongst others by integration of risk-based O&M which will be further elaborated in the next section.

## 5.2 Risk-based operation and maintenance methods

Risk-based O&M is about finding the optimal balance between structural reliability and lifecycle cost of deteriorating assets. Risk-based O&M is used to develop a detailed maintenance plan and/or strategy for safe and fault free operation in a cost effective manner (Khan & Haddara, 2004). In the construction industry, fault free operation can be defined as the time period a specific element meets its function. For example, in the case of highway lighting, at a point in time one light bulb of the entire lighting system may fail. However, the required function of the lighting is to provide a certain amount of lux at ground level. Even with one light bulb failing, the system could still meet this requirement. Therefore, the approximated associated risk of this failure mode is low. This results in cost effective maintenance due to the fact that a maintenance engineer is sent out only when multiple light bulbs are failing and not for every single failure separately.

When the ability of a system to withstand fault begins to decrease, the chances of functional failure increase. Functional failure identifies the time when an object is no longer able to satisfy the required function (Fruguglietti et al., 2012). This principle is illustrated in figure 5-1, where the threshold represents the level at which functional failure occurs.

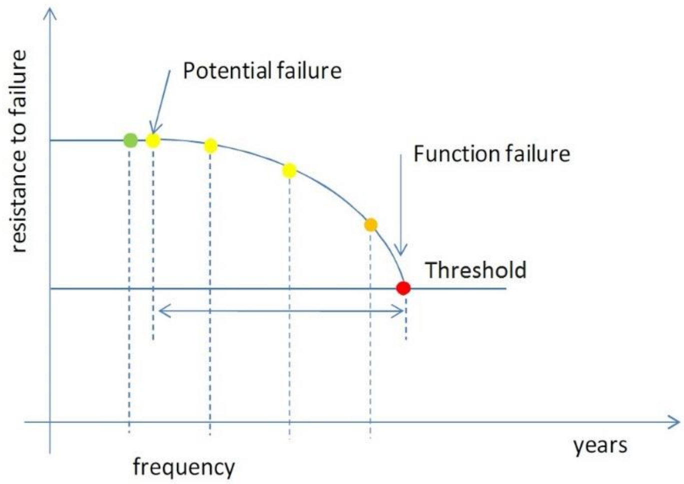



Figure 5-1: Objects resistance to failure (Fruguglietti et al., 2012).

Figure 5-2 illustrates the percentage of failures that have occurred over a period in time of individual elements within an imaginary building. This shows that generally element failure within such a system is not evenly distributed. Often the most unreliable elements account for the largest in-service failures (WMG, 2009). If these unreliable elements are identified during the concept and design phase, much more can be done to prevent future failures occurring. In the construction industry however, there is a tendency to leave probable risks and react to them when they have occurred, rather than dealing with them in advance (Pruvost et al., 2012).
Possible method for defining and calculating risks is the Failure Mode Effect and Criticality Analysis (FMECA) discussed in the next section.

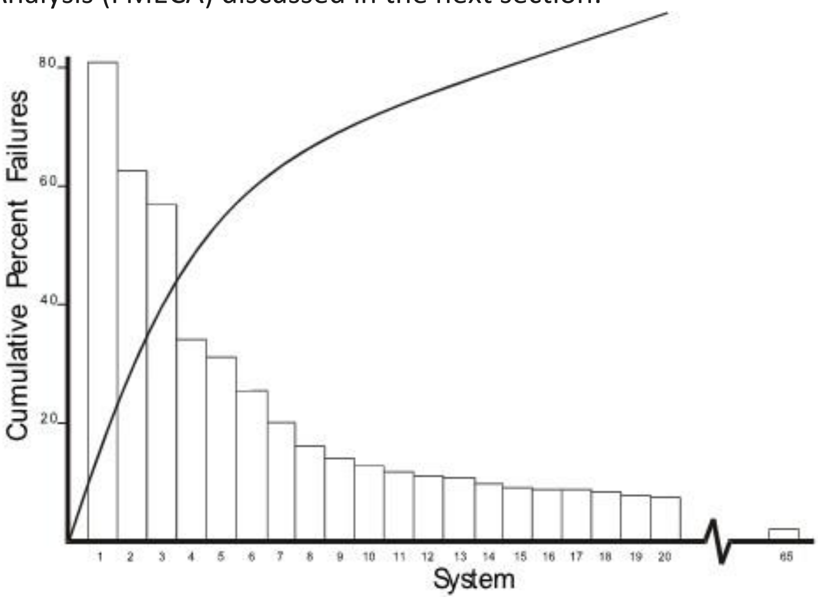



Figure 5-2: Cumulative failure plot (WMG, 2009)

### 5.2.1 Failure Mode Effect and Criticality Analysis (FMECA)

Failure Mode Effect and Criticality Analysis (FMECA) helps in determining the resulting consequences of a failure and evaluating the risks associated with the failure (Woldesenbet, 2014). FMECA has the potential to prioritize attributes, integrate with other tools and assist in making corrective actions. FMECA is an extension of the Failure Mode and Effect Analysis (FMEA) method with the addition of failure criticality calculation. FMECA is certainly no new methodology and has proven applications in various industries, including automotive, aeronautical, military, nuclear and electro-technical engineering (Zhou, Yu, & Zhang, 2015). In the construction industry, application of the methodology becomes more integrated especially for complex and high risk assets. For maintaining buildings with a high risk and complex nature in an acceptable condition and in a cost effective way FMECA is recommended (Fruguglietti et al., 2012).

The FMECA is a bottom-up method to highlight all possible failure modes, all causes of failures, effects produced by failures, possible solutions (Carmignani, 2009). This results in a clear pattern of maintenance needs and opportunities. Thereby, supporting in the identification of the appropriate maintenance policies, and in consequence in the definition of the activities related to the maintenance policies chosen (Fruguglietti et al., 2012). Generally, a FMECA analysis is done in a worksheet where the rows represent single elements of the reviewed system and the columns contain associated information regarding the element, failure mode, effect, cause, control processes, actions, criticality calculation, etc.

Furthermore, if a FMECA is not reviewed and updated regularly, it will lose its main capability of giving an actual overview of risks and taken countermeasures for improvement (Schmidt, Riedel, & Kangas, 2011). The FMECA update frequency depends on the risk levels involved. Besides the consideration of failure mode, effect and cause the criticality calculation provides a clear prioritization of failure modes with their associated risk. The literature discusses multiple methods for criticality calculation of which the calculation of a Risk Priority Number (RPN) is commonly applied in the AEC/FM industry. The RPN calculation takes three parameters into consideration, each correlated to a score range:

- Severity (S): Indicates the gravity of the effects of a failure which affect the system or consumer that uses the component. A scale of 1 to 10 is used to indicate the severity where 1 is not significant and 10 is very serious, dangerous or catastrophic (WMG, 2009).
- Occurrence (O): Indicates the probability of a particular failure occurring. Again, the scale is from 1 to 10 where 1 indicates that it is unlikely to occur and 10 indicates that it is almost certain to occur.
- Detection (D): Measure of the probability of detecting a failure mode by controls or inspections. The scale from 1 to 10 applies where 1 indicates that detection is highly likely and 10 almost impossible. Detectability should be related to the probability of detecting a failure before it takes place and not the detectability of the failure occurring.

The RPN is obtained from the product of these three parameters (Carmignani, 2009):

$$RPN = S \times O \times D$$

With a possible RPN range of:

$$RPN_{min} = 1;\ RPN_{max} = 1000$$

The higher the RPN of a failure mode, the greater the associated elements risk and reliability indicating the need to be re-examined. Subsequent modifications have the purpose of lowering the RPN in order to avoid a failure's occurrence or repetition. This often includes the RPN calculation before preventive measures are taken (Criticality a priori) and the revised RPN calculation (criticality a post) to evaluate the effect of defined preventive measures. The advantage of this method and the reason for its success and diffusion are due to the possibility of using linguistic expressions to which numerical values can be correlated through appropriate conversion charts (Carmignani, 2009).

However, many authors have highlighted the FMECA's numerous limits, questioning its reliability and theoretical robustness, as well as the advantages (Carmignani, 2009). For example, two FMECA's compiled by different authors often result in varying RPN's for the same failure mode. In addition, the FMECA technique completely neglects the different impacts between severity, occurrence and detection on the critical urgency failure calculation.

Figure 5-3 illustrates a simplistic FMECA for a pillar with one failure mode, effect, cause and RPN calculation. There is no threshold value for RPN and therefore it is not valid to say which RPN ranges can be considered low, medium and high. In other words, there is no value above which it is mandatory to take a recommended action or below which value no further action are required (Lipol & Haq, 2011).

Interestingly for this research, Pruvost et al. (2012) recognize the value of liked models providing the nested structure which can be of substantial help for performing risk chain analyses like FMECA. This is supported by Zhou et al. (2015) identifying the difficulties of sharing and reusing FMECA results in tabular form between enterprises.

| Item | Failure | Effect | Cause | Detectability | Probability of occurrence | Severity | RPN |
|---|---|---|---|---|---|---|---|
| pillar | swelling | Start of run-off phenomena, Traces of diffusion (dark spots), Washout, Permeation (or seepage water inflow), swelling, detachment | Presence of aggressive water (water is not collected) | 3 | 5 | 2 | 30 |

| | |
|---|---|
| Items with RPN value low | negligible |
| Items with RPN value medium | medium criticality |
| Items with RPN value high | High criticality |

Figure 5-3: RPN calculation (Fruguglietti et al., 2012).

# 6 INTERVIEWS

Five experts in the field of BIM and/ or FM are interviewed to determine how information delivery to O&M departments currently works and what can be done to improve current working methods. In addition, implementations of BIM working methods within the industry are discussed. Furthermore, the experts are asked if transition of information to O&M could be optimized by linking information to elements in IFC models. These enriched models acts as a network of information which could be handed over and worked with during the O&M phase.

The interviewed asset managers are mainly active in infrastructural sector while the BIM experts are predominantly active in the residential and utility construction sector. All interviews with summarized expert responses can be found in appendix 1.

## 6.1 Interview findings

Interviewees recognize the inefficiency of current project handover to O&M in the Dutch AEC industry. Currently, contractors are making a selection of information to be handed over just prior to the handover deadline which often results in the handover of incorrect and incomplete project information. In addition, in some cases complete handover takes place years after the deadline or even skipped because handover costs are higher compared to the last client's payment.

Expectations are that BIM models created for design and construction can overcome this problem significantly. However, even when BIM models are available, these are not always handed over to the client/ asset manager. Currently, BIM models are made with design and construction objectives and O&M informational needs are not taken into account.

In addition, interoperability issues hamper the handover of BIM information to O&M. For example, the Dutch public works and water management agency called Rijkswaterstaat, uses over 150 different approaches and software packages for their asset O&M. Contracted parties are obliged to use/ deliver information for one of these software packages. Transition to open file standards, such as IFC, should prevent interoperability issues in the future.

All interviewees see the potential of the open standard IFC for O&M to manage and structure information. However, points of concern are data losses when models are exported to an IFC structure from proprietary formats.

In addition, one of the asset managers points out that models should not contain all possible related data. Because models created for construction are often highly rich of information, it is critical to make a good selection prior to handover. Without a good selection, all this information should be kept up to date during a building's lifecycle. Hence, if not updated, the complete model becomes unreliable. In line with this information selection, BIM experts point out that high Levels of Detail (LOD) are not important for O&M. However, focus should be more on information and therefore higher Levels of Information (LOI) are of much more importance. For O&M, LOD300 should be enough as visual representation and provide asset managers easy access to information.

Informational requirements for O&M should be defined in early project stages. Information exchange requirements could be defined in an information delivery specification by the client. An example of an information delivery specification is the Dutch RVB BIM standard. This standard defines the BIM informational requirements and can be part of a contract from the first stage of a construction project. Asset managers should ideally be involved in the design and construction phase to oversee the model development and ensure suitability for

O&M. Resulting in an optimal situation where handover as a project phase would not exist anymore.

In addition to optimization of handover as-build information, asset manager recognize possibilities of 3D models for O&M processes itself. For structuring and easy access, information could be linked to objects in a 3D model. Interviewees mention the following data to be linked or obtained from a model and see O&M process optimizations by doing so:

- Functional requirements (system engineering);
- Data for risk-based O&M (failure modes, failure probability, etc.);
- Maintenance planning;
- Object condition assessment information;
- Quantity take-offs;
- Product supplier information;
- Service history data;
- Maintenance costs.

In contrast to the possibilities, asset managers question if making a 3D model is worth the high investment for O&M. In particular for existing buildings where no model is available from pre- M&O phases. Additionally, added value of 3D models for line infrastructural assets (e.g. roads, train tracks) is questioned. For infrastructure, especially for complex structures (e.g. locks or tunnels) and the connection between various structures, 3D model value is acknowledged.

In all cases, object standardization and classifications are of great importance to make models accessible and understandable for all stakeholders. Preferably by application of 'Object Type Libraries' (OTLs) or classifications with mappings to other classification systems.

## 6.2 Conclusion

The interviews pointed out the inefficiency of current BIM project handover to O&M. This is mainly caused by the fact that currently models are made with design and construction objectives while post construction phases are not taken into account. Nevertheless, the potential of using 3D models for O&M to manage and structure information seems significant. However, making a good data selection prior to handover remains important. When not done correct, a lack of information results in rework during O&M while transferring to much information results in extra work due to the obligation to keep all this data up to date during a building's lifecycle.

In addition to handover optimization, models are of value for structuring and easy access of O&M information. However, asset managers are in doubt if making a 3D model for existing buildings is cost effective. Object standardization and classifications seem of great importance to make models accessible and understandable for all stakeholders. To further research object classification relevance and linking information to 3D models for O&M a tool is develop of which the working is discussed in the next chapter.

# 7 TOOL DEVELOPMENT

In this chapter a developed prototype tool to semantically enrich IFC models for O&M will be discussed. The goal of creating an information model in the first place is that it ultimately helps to improve the performance of the building. The reviewed literature and interviews show the potential of models by providing easy access to object information for an asset manager to make informed decisions about its maintenance and optimal operation.

With the development of BIM, knowledge sharing between facility management and design professionals has become possible. However, different stakeholders in the AEC industry are currently still working according to their own working habits and are afraid to cooperate with each other since inadequate interoperability is still a problem (Liu & Issa, 2013). Application of the open model standard IFC may overcome this problem partially.

The tool should assist an asset manager in risk-based O&M by providing object information using IFC models and structure/ preserve new O&M information for future use. Pruvost et al. (2012) proposes the principle of multi-model-based risk management. Following this proposal, the developed tool facilitates a mapping between FMECA and IFC models.

In addition to the lack of a mapping, a part of the problem resides in the absence of reusable risk documentation that classifies risks and remedial measures using common terms and descriptions. In each project, the relevant risks have to be newly identified and documented in varying forms, hindering the efficient build-up of risk models and their reuse in risk analysis applications (Pruvost et al., 2012).

For the tool development, the following programming software and modules are used:

- Python 2.7 64-bit;
- Python OCC 0.16;
- IfcOpenShell 2.7-0.5.0;
- OpenPyXL 2.2.6;
- lxml 3.4;
- PyQt 4;
- RDFLib 4.2.

This chapter starts with a case description in section 7.1 and the introduction of a real world IFC model for development and testing. Thereafter in section 7.2, prototype tool 1 is elaborated and conclusions in terms of possibilities and limitations are leading to the development of prototype tool 2, further elaborated in section 7.3. Finally, tool limitations and validation are discussed in section 7.4 and 7.5 respectively.

## 7.1 Case description

The 'Uithoflijn' project includes the construction of a tramway connection between Utrecht central station and the Utrecht Science Park (Uithof) in the Netherlands. The client is 'Regiotram Utrecht' (RTU) with a total project budget of €323 million (RTU, 2015). With a total length of 8 kilometers and 8 stops the tram connection will replace the current bus service which is dealing with capacity issues. Expectations are that by the year of 2020 up to 45.000 passengers will commute between Utrecht central station and the Uithof on a daily basis. Project completion is scheduled in the first semester of 2018. Project design contains

2D information of the line infrastructure and 3D models of the more complex structures such as bridges and overpasses.
For this research one building of this project is selected of which a 3D IFC model is available. This model concerns an overpass facilitating the tramway passing a canal as visualized in figure 7-1. This IFC model is selected for tool development and testing due to the relative richness of element properties and manageable size as illustrated in figure 7-2. Early testing showed however that standardized object classifications are not applied.


Figure 7-1: Location IFC model 'Uithoflijn' (RTU, 2015)

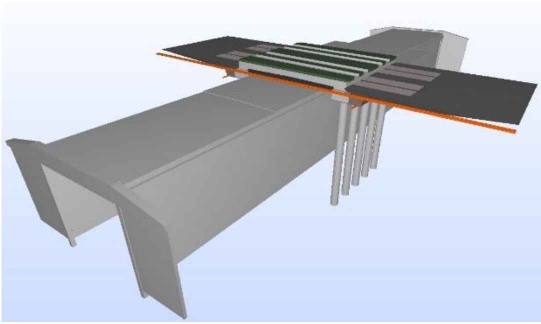


Figure 7-2: IFC model overpass (RTU, 2015)

## 7.2 Prototype tool 1

Prototype tool one consists of three stages. The first stage enables the asset manager to extract all objects from an IFC model and write these to a FMECA sheet according to the NEN2767-4 object breakdown structure (OBS). Second, after completion of the FMECA by the asset manager, in stage two it is possible to write this FMECA information to IFC. Finally, in stage three risk data can be visualized and IFC element properties can be viewed. The complete script of prototype tool 1 can be found in appendix 2.

### 7.2.1 Tool 1.1: Export Object Breakdown Structure

From the interviews with asset managers the traditional process of making a FMECA is clarified. By making this analysis the first step is to fill the FMECA template with as-built information in terms of all building element in an object breakdown structure (OBS) according to the NEN2767-4. Hereby, sheets can get as long as hundreds of rows for extensive projects where each row represents an element class. When a BIM is available this process could be automated on condition that the BIM is modeled using element classifications with a mapping to the NEN2767-4. For this tool the BIM case model is enriched with object classifications first to be able to continue with this model. Therefore, element classifications are considered as a pre model condition.

In order to automate the process of extracting an OBS according to the NEN2767-4 from the BIM, tool 1.1 starts by extracting all classifications. For this model the Rijkswaterstaat (RWS) OTL classification is applied which is stored in the IFC model as an *IfcPropertySingleValue*. The relation within an IFC model between the *IfcPropertySingleValue* and an *IfcBuildingElement* is illustrated in figure 7-3. Classifications in the case model are stored as *NominalValue* with *IfcPropertySingleValue* name 'Type Mark'. The tool starts by extracting all available RWS OTL classifications from the BIM and writing them to a Python list.

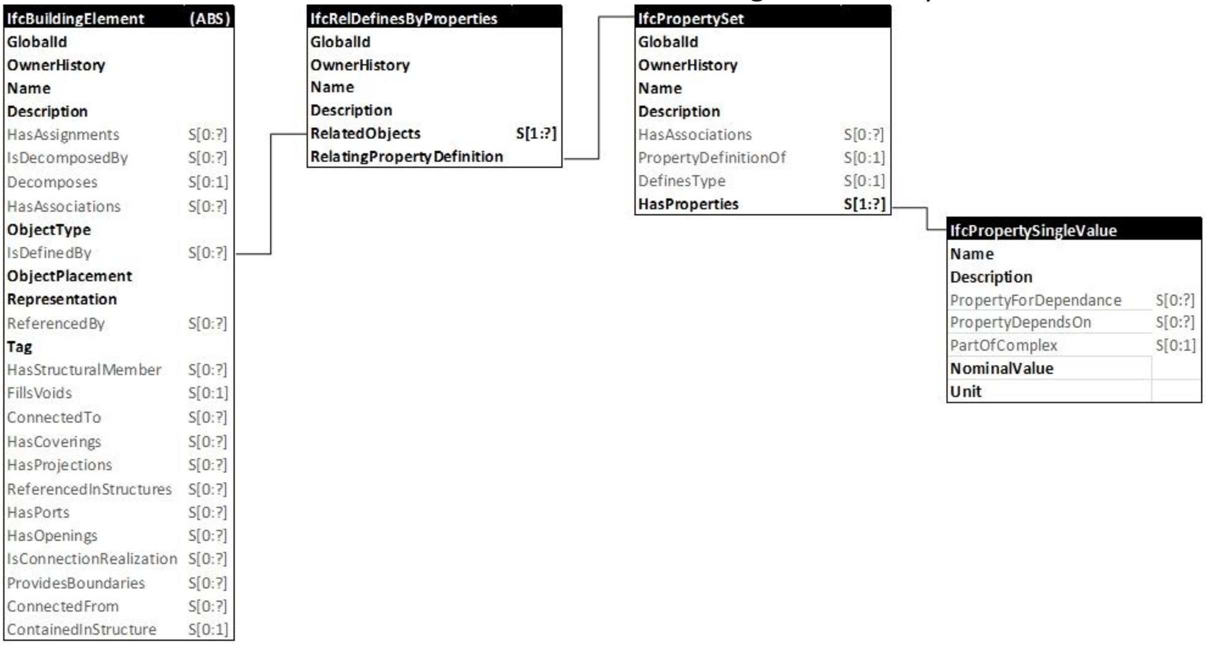


Figure 7-3: IfcBuildingElement to IfcPropertySingleValue (From: BuildingSMART, 2013)

Thereafter, the previously created list with classifications is mapped to the NEN2767-4 O&M standard classification. For each classification in the list a new python list is created containing the management objects, elements and structural parts with NEN2767-4 codes. Mapping rules are stored in an xml file decomposition visualized in figure 7-4.

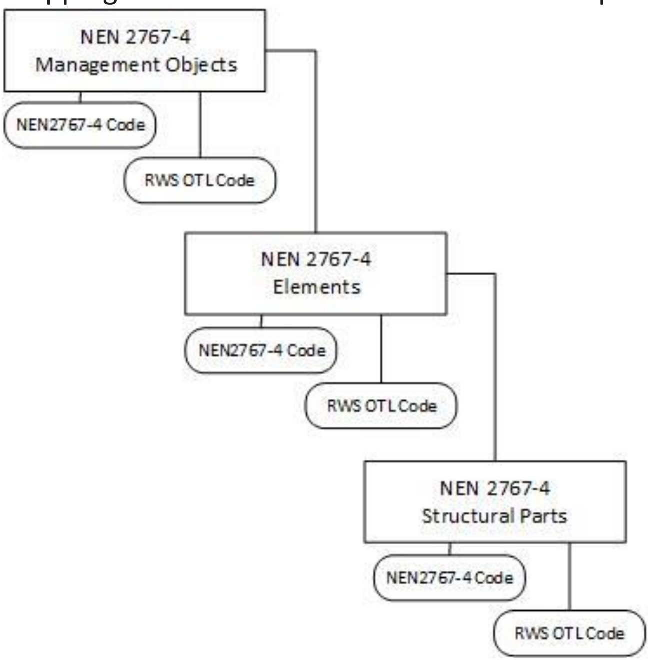


Figure 7-4: Decomposition .xml mapping NEN2767-4 and Rijkswaterstaat OTL

Finally, tool 1.1 writes the python list containing the OBS to a FMECA template which is saved to the user's hard drive. Figure 7-5 illustrates the result translated in English from left to right as Management Objects, Elements, Structural Parts and NEN Codes. In addition, column headings of the other relevant values for FMECA are written to the sheet. This provides the asset manager with a significant time reduction in completing a FMECA and a potential fault free result. Appendix 3.1 contains a flow diagram of the complete process of tool 1.1.

| **Beheerobjecten** | **Element** | **Bouwdeel** | **NEN Code** |
|---|---|---|---|
| Overkluizingen | Kerende constructie | Combiwand | 1121 |
| Overkluizingen | Hoofddraagconstructie | Rijdek | 1461 |
| Overkluizingen | Fundering | Grondverbetering | 1234 |
| Overkluizingen | Hemelwaterafvoer (HWA) | Aansluitleiding | 1013 |
| Overkluizingen | Hoofddraagconstructie | Sleufonderbouw | 2038 |

Figure 7-5: Result tool 1.1

For the asset manager to perform all actions combined in tool 1.1 a simple graphical user interface (GUI) is provided as visualized in appendix 4.

### 7.2.2 Tool 1.2: IFC model enrichment

To obtain easy access and visualization possibilities of FMECA information, tool 1.2 writes this data to the BIM. In this case semi-structured tabular data from a FMECA Excel sheet is written to an IFC model. This IFC model is hereby semantically enriched because additional meaning of an element is added. This principle is schematically illustrated in figure 7-6.

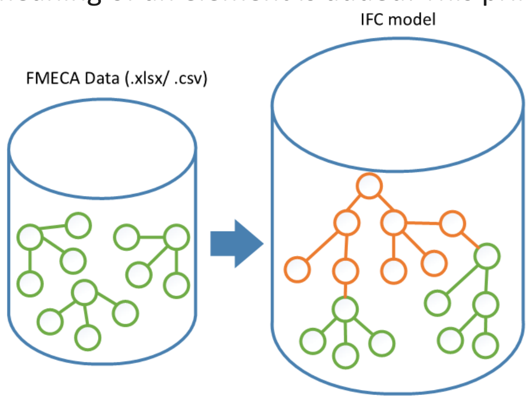



Figure 7-6: Schematic representation IFC model enrichment

For semantic enrichment of IFC with FMECA information, tool 1.2 first creates a python dictionary from the FMECA Excel sheet. Evidently, the FMECA has to be completed first by an asset manager. Subsequently, a Python dictionary is created from IFC containing OTL codes with corresponding elements. Testing revealed not all classifications where stored within IFC using *IfcRelDefinesByProperties* as visualized in figure 7-3. Some elements are linked with their classification code using *IfcRelDefinesByType* and one extra class named *IfcTypeObject* (e.g. *IfcWallType*). These relations are visualized in figure 7-7 where the *IfcPropertySet* and *IfcPropertySingleValue* are linked to *IfcTypeObject* respectively. Hence, the tool checks in what way element classifications are stored in order to create the dictionary.

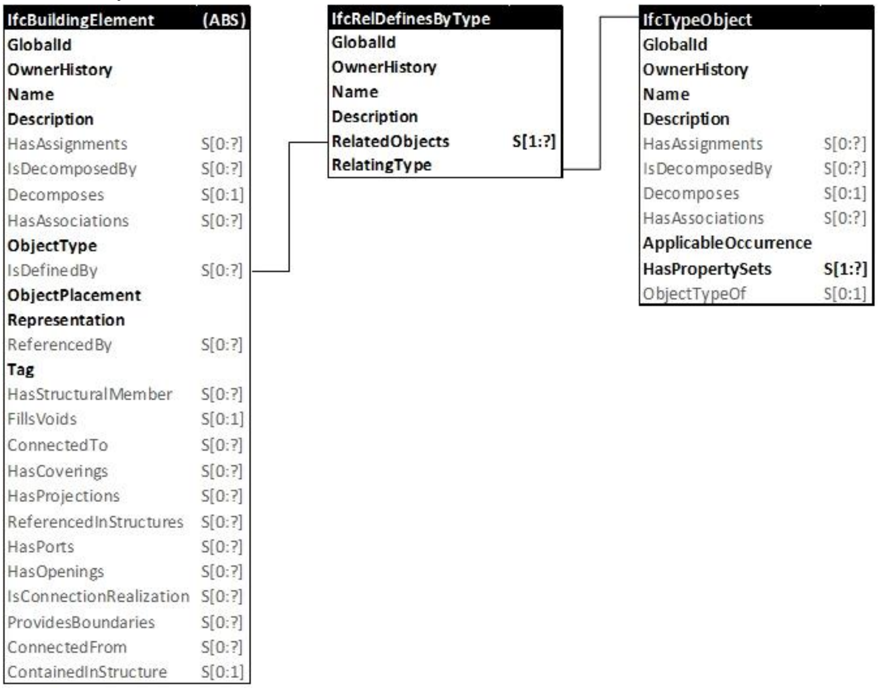


Figure 7-7: IfcRelDefinesByType and IfcTypeObject relation (From: BuildingSMART, 2013)

Finally, using both dictionaries new *IfcPropertySets*, *IfcRelDefinesByProperties* and *IfcPropertySingleValues* are created representing the extracted FMECA information. These processes are graphically displayed in a flowchart in appendix 3.2. A GUI is provided to enable an asset manager to preform above described operations easily (appendix 4).

### 7.2.3 Tool 1.3: IFC property viewer

For risk visualization the last tabs provides an IFC viewer with the ability to represent elements by corresponding risk colors. Colors are derived from the Risk Priority Number (RPN) values stored as *IfcPropertySingleValue* by tool 1.2. Elements get the color green, yellow or red for the values RPN < 30, 30 ≤ RPN < 100 and RPN ≥ 100 respectively.

By risk visualization the asset manager can locate high risk objects clearly and by object selection raise all associated properties (e.g. material, manufacturer, warranty information, geometric data). Figure 7-8 illustrates the risk visualization and selection of a foundation pile with corresponding IFC properties.

The tool first creates a geometric representation of the loaded IFC file using PythonOCC and IfcOpenShell modules. Thereafter, from the RPN values a python dictionary is created with colors and element IDs to enable model parsing using the defined colors. Subsequently, each element is added to a new dictionary with corresponding IFC properties. For user selection within the viewer, the elements GUID is stored as global variable and corresponding properties from the dictionary are shown. The flowchart of this python code can be found in appendix 3.3.

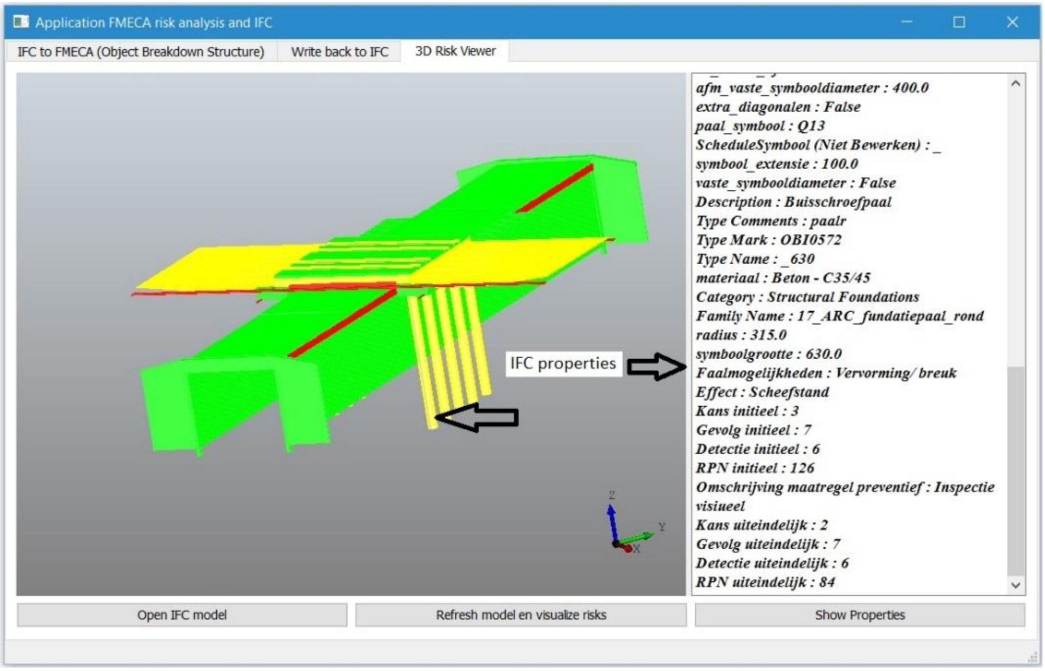



Figure 7-8: Graphical user interface tool 1.3

### 7.2.4 Conclusion tool 1

Development of tool 1 has resulted in a working prototype and by showing its abilities to asset managers the value of data visualization and data linking for O&M is recognized. Extracting all elements from a model by using a mapping to classification systems for O&M, in this case the NEN2767-4, time saving and failure reduction potentials can be significant compared to traditional methods. In addition, storing generated O&M data within an IFC model results in visualization and easy information exchange possibilities. By storing O&M data in the IFC itself, data can be accessed using any IFC viewer on the market.

However, discussion with asset managers and testing revealed some drawbacks of storing O&M data within the IFC model. Three main disadvantages have been uncovered. First, element properties stored within IFC files are harder to query/ reuse. Questions such as: 'What is the average RPN value defined in all past projects for objects with NEN2767-4 classification 1023?' are hard to answer. Second, by writing properties to IFC from an external application (in this case the FMECA sheet) versioning risks will exist. After adjustments within the external application a tool should facilitate removal of old IFC properties and replaced by updated values. Third, IFC file pollution with properties for various purposes (facility condition assessment properties, risk-based O&M properties, maintenance planning data, functional requirement, service history data, etc.) could result in error prone models, particularly due to the relatively long time span of the exploitation phase. Hence, a second prototype is proposed in the next section aiming to overcome the discussed disadvantages of writing O&M data directly to IFC.

## 7.3 Prototype tool 2

Due to the disadvantages of storing O&M data within the IFC file itself as concluded in the previous section, a second tool is developed. In this section the working is discussed to be able to draw conclusions in terms of pros and cons in comparison to the previous tool. Contrarily to tool 1, in this section the linkage of datasets is explored with the expectation to overcome data reusability, versioning and IFC pollution issues. This data linking is schematically visualized in figure 7-9 where IFC elements are linked to the relevant parts of RDF data. The complete Python script of prototype tool 2 can be found in appendix 5.

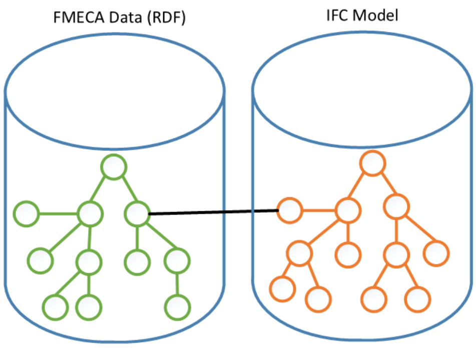



Figure 7-9: Schematic representation IFC-RDF linking

In section 7.3.1, the working of the first part of this tool is discussed which transforms tabular FMECA data to RDF by adding semantic meaning using CSVW. Thereafter in section 7.3.2, an IFC property & RDF viewer is proposed and further elaborated. Subsequently in section 7.3.3, SPARQL queries are introduced to query FMECA RDF data to show data reusability which eventually could enhance collaboration between the AEC and FM industry. Finally in section 7.3.4, tool prototype reflection and conclusions are provided.

### 7.3.1 Tool 2.1: RDF parser

Storing O&M data in (semi-) structured formats (e.g. csv, xml, JSON) are generally only usable for colleague asset managers due to the lack of description of the actual data. This also applies to FMECA data which is stored in tabular form and therefore difficult to share and reuse between enterprises (Zhou et al., 2015). Hence, the FMECA data needs to be transformed to achieve knowledge sharing and allows reusability. By making use of semantic web technologies and the capability of RDF the above stated issues can be overcome. Conversion to RDF results in a data format with semantic meaning, easy to interlink with other relevant data and could be queried using SPARQL (McGlinn, 2015). Tool 2.1 facilitates the transformation of tabular FMECA data into RDF based on the python RDF parser from Brian Walshe, Diarmuid Ryan (Trinity College Dublin, KDEG) and Markus Ackermann (University Leipzig, AKSW). Several optimization modifications are done in terms of RDF subject generation from the tabular data.

CSVW is a simple way of providing sematic meaning for CSV values in a JSON document supporting conversion of data into RDF (McGlinn, 2015). First some general properties are defined in the CSVW such as delimiter, URL and modification date. Then, in an array of column descriptors e.g. names, datatypes and property URL's are set. In addition, for this research the primary key contains a column reference to the property that holds element classifications which become the RDF subject after conversion. The CSVW document to describe FMECA tabular data semantics can be found in appendix 6.

No ontology which describes FMECA parameters seems to be available. Therefore, for this research a concise ontology is created using TopBraid ontology design software and is included in appendix 7. This ontology defines the FMECA parameters with associated data and value restrictions.

A GUI enables an asset manager to perform the conversion to RDF easily as shown in appendix 4. Figure 7-10 visualizes a part of the tool's output by means of an RDF graph. This graph is showing triples with an element classification as subject, predicates as defined by the FMECA method and objects from the tabular data itself.


Figure 7-10: RDF graph example after conversion

For the steps and sequence of these steps of RDF conversion formulated in the Python script, an overview of this tool is visualized as a flowchart which can be found in appendix 8.1. In addition, the output (RDF) is partially included in appendix 9.

### 7.3.2 Tool 2.2: IFC property & RDF viewer

In line with tool 1.3, which provides the possibility of IFC property viewing and visualization, this tool (2.2) has the ability to show IFC properties with the addition of RDF data. This approach to enrich IFC Step Physical Files with arbitrary RDF triplets is suggested by Beetz et al. (2015). Enrichment of IFC models with RDF triplets offers the use of exiting vocabularies and datasets available as linked open data while using the widely accepted and implemented IFC model standard. While nowadays a vast amount of scientific literature is available on the transformation of the core model or partial sub-model chunks itself into RDF, these approaches demand a considerable shift in technologies and a high implementation effort (Beetz et al., 2015).

Hence, this tool provides the possibility to view an IFC model with element properties and provide associated FMECA RDF data by selection of and element in a 3D view. In addition, the prototype has the ability to visualize RPN values, obtained from RDF, in terms of colors.

The ability to link this data arises due to the use of either common element indicators or element -indicators with mappings between them in both data repositories. This principle is visualized in figure 7-11 showing on the left an *IfcPropertySingleValue* instance containing the element classification and on the right one associated RDF triple. As discussed, for this research two different classification systems are used hence an external classification mapping definition is required.


Figure 7-11: IfcPropertySingleValue linked to RDF triple

The tool first parses the IFC model using PythonOCC and IfcOpenShell resulting in non-colored model visualization. Then, after loading an associated RDF file the graph is parsed and a python dictionary is created from the file containing element classifications and corresponding RPN values. In addition, a Python dictionary is created from the IFC file containing the same element classification, using the classification mapping, and associated element IDs. Thereafter, both dictionaries are combined in a new Python dictionary eventually containing element IDs and risk colors derived from RPN values. Using this dictionary the model can be refreshed in the GUI, visualizing the model with appropriate risk colors. Furthermore, for IFC property browsing a Python dictionary with GUID and corresponding IFC properties is created. In addition, a dictionary from the RDF file is created with the conversion in line with RPN value extraction from RDF. Eventually, this dictionary contains element GUIDs with corresponding RDF predicates and objects. For an element selection by a user, the corresponding GUID is set as global variable and using the dictionaries corresponding IFC properties and RDF predicates and objects are shown as illustrated in figure 7-12. The flowchart describing the complete working of this tool can be found in appendix 8.2.

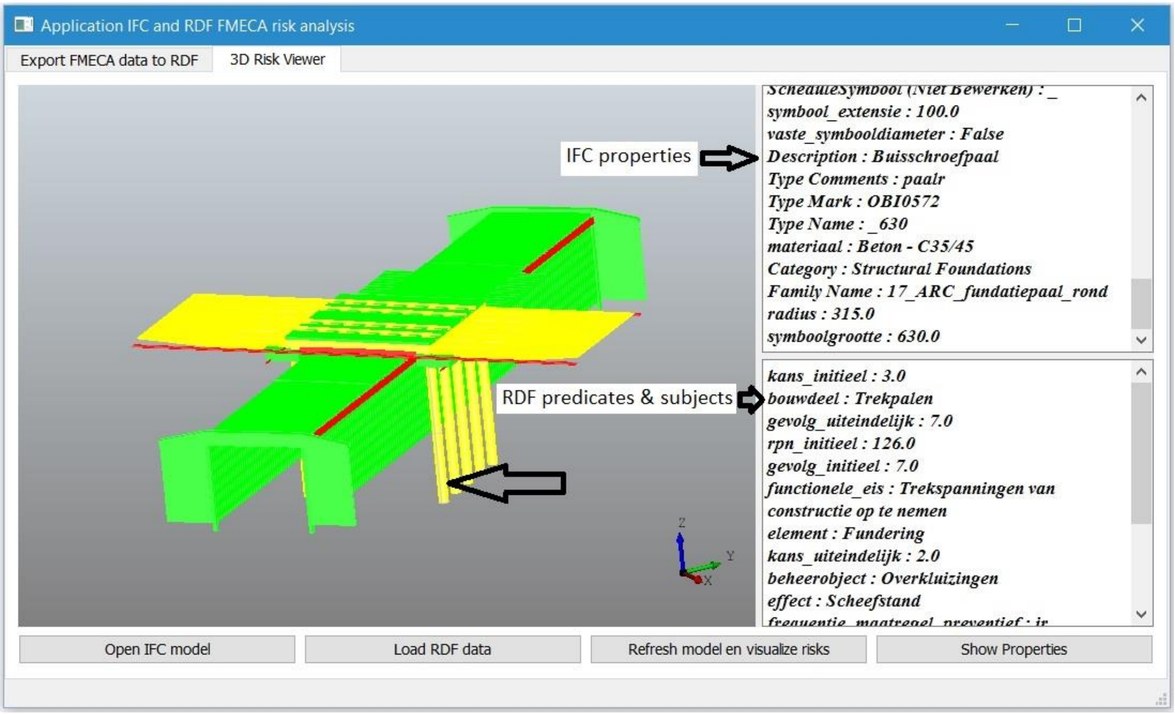



Figure 7-12: Graphical user interface tool 2.2

### 7.3.3 SPARQL Queries

SPARQL is a query language for RDF with capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. In addition, SPARQL supports extensible value testing and constraining queries by source RDF graphs (W3C, 2008). Query results are so-called 'SPARQL endpoints'.

Storing O&M data using RDF with the ability to retrieve specific data by SPARQL queries allows sharing and consuming data in a more interoperable way. Data reusability has the potential of stimulating data sharing between AEC and FM industry. To demonstrate, the RDF FMECA data created using tool 2.1 are locally published using Fuseki server and simple SPARQL queries are run. The first example shows how failure modes defined for specific element classes can be obtained from the RDF by a SPARQL query. Nowadays, failure modes are often redefined for every single project. Results could give the asset manager some guidelines that might be complemented with project specific failure modes. For retrieving failure modes, or 'faalmogelijkheden' in Dutch, the following query is used:

```
PREFIX ab: <https://www.isbe.tue.nl/ontology/fmeca/>

SELECT ?Faalmogelijkheden ?NEN_Code
WHERE
{
  ?NEN_Code ab: faalmogelijkheden ?Faalmogelijkheden .
}
```

For the sake of this research, the amount of failure modes defined for each element classification instance is limited to one. Hence, the results are as illustrated in figure 7-13.

```
| Faalmogelijkheden                                                              | NEN_Code                                                                     |
=================================================================================================================================================================
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.2038> |
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1328> |
| "Scheurvorming"^^<http://www.w3.org/2001/XMLSchema#String>                      | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1013> |
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1686> |
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1153> |
| "Verzakking"^^<http://www.w3.org/2001/XMLSchema#String>                         | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1234> |
| "Wapeningscorrosie door carbonatatie"^^<http://www.w3.org/2001/XMLSchema#String> | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1121> |
| "Scheurvorming"^^<http://www.w3.org/2001/XMLSchema#String>                      | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1604> |
| "Wapeningscorrosie door chloriden"^^<http://www.w3.org/2001/XMLSchema#String>   | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1126> |
| "Verzakking"^^<http://www.w3.org/2001/XMLSchema#String>                         | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1638> |
| "Wapeningscorrosie door chloriden"^^<http://www.w3.org/2001/XMLSchema#String>   | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1461> |
| "Afbrokkeling"^^<http://www.w3.org/2001/XMLSchema#String>                       | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.2140> |
| "Wapeningscorrosie door chloriden"^^<http://www.w3.org/2001/XMLSchema#String>   | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1680> |
| "Scheurvorming"^^<http://www.w3.org/2001/XMLSchema#String>                      | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1137> |
| "Scheurvorming"^^<http://www.w3.org/2001/XMLSchema#String>                      | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1024> |
| "Wapeningscorrosie door chloriden"^^<http://www.w3.org/2001/XMLSchema#String>   | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1687> |
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1179> |
| "Breuk van tussenlaag"^^<http://www.w3.org/2001/XMLSchema#String>               | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.2179> |
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1557> |
| "Vervorming/ breuk"^^<http://www.w3.org/2001/XMLSchema#String>                  | <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1051> |
```

Figure 7-13: Result SPARQL query failure modes

The second SPARQL query example aims to retrieve an average RPN value from past projects for a specific element classification. Resulting average has the potential to identify high risk elements in e.g. conceptual designs. By doing so, O&M costs may be reduced. In the following example only one FMECA RDF file is queried, however higher precision could be archived by larger samples. The example shows how an average RPN can be retrieved from elements with NEN2767-4 classification '1638':

```
SELECT (AVG(?double) AS ?average_RPN_NEN1638)
WHERE{
 <https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1638>
 <https://www.isbe.tue.nl/ontology/fmeca/rpn_uiteindelijk> ?double;
}
```

Figure 7-14 shows the result of the second query example revealing an average RPN of '24' for object with NEN2767-4 classification '1638'. This outcome may influence e.g. design decisions in future projects.

```
-----------------------
| average_RPN_NEN1638 |
=======================
| 24.0e0              |
-----------------------
```

Figure 7-14: Result SPARQL query average RPN

### 7.3.4 Conclusion tool 2

Efforts to develop a working second prototype resulted in a usable and functioning tool. Compared to the first prototype IFC pollution is avoided by linking to an external data repository. The ability to link this data arises due to the use of either common element indicators or element indicators with mappings between them in both data repositories. In addition, potential versioning risks may be reduced due to the separation of datasets. However, current risk-based O&M working methods do not include working directly in a native RDF file. Therefore, data transformation is still required involving versioning issues. Though, this working method seems to be more future proof due to the complete expected shift towards semantic web technologies within the industry. RDF offers improved data reusability e.g. by SPARQL queries as demonstrated. This could help enhance data sharing between the AEC and FM industry facilitating for example design for maintenance.

## 7.4 Tool validation

In this section the tools are validated in terms of correctness and completeness of generated data and data visualizations. In section 7.4.1, the three components of tool 1 are validated to reveal possible errors and reflect on them. Thereafter in section 7.4.2, the second tool is validated and data/ visualization deviations are discussed.

### 7.4.1 Validation prototype tool 1

The separate components of tool 1 are validated in a reverse order because the last tool (1.3) should expose inconsistencies caused by prior tool components. For example, if tool 1.1 failed to export all element classifications to the FMECA template, non-colored elements should be visible in the model view of tool 1.3. In addition, when tool 1.2 failed to write FMECA data back to all elements in the model some element should be non-colored and lack a FMECA property set. After close visual inspection the conclusion can be drawn that all elements in the model do either have a green, yellow or red color. Hence, tool 1.1 has exported all element classifications and tool 1.2 created an FMECA property set for each element in the model. IFC model checking software, in this case Solibri Model Checker (SMC), confirmed that for each element a FMECA property set is available (figure 7-15).

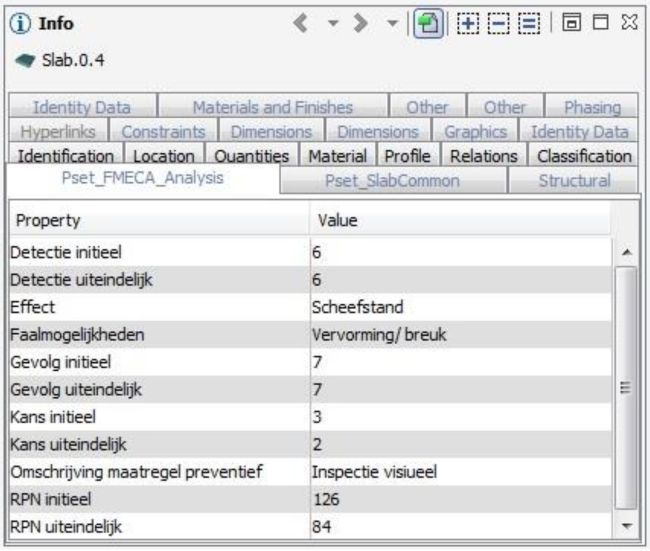


Figure 7-15: Example FMECA property set by Solibiri model viewer

However, by checking the Object Breakdown Structure exported by tool 1.1 some inconsistencies are discovered. This tool extracts all objects with RWS OTL classification from the IFC model and queries the NEN2767-4 hierarchy to obtain the parent element. Checking of the NEN2767-4 hierarchy revealed some structural parts can have different parent elements as illustrated in figure 7-16. In order to omit this error, adaption of the NEN2767-4 classification system is required.

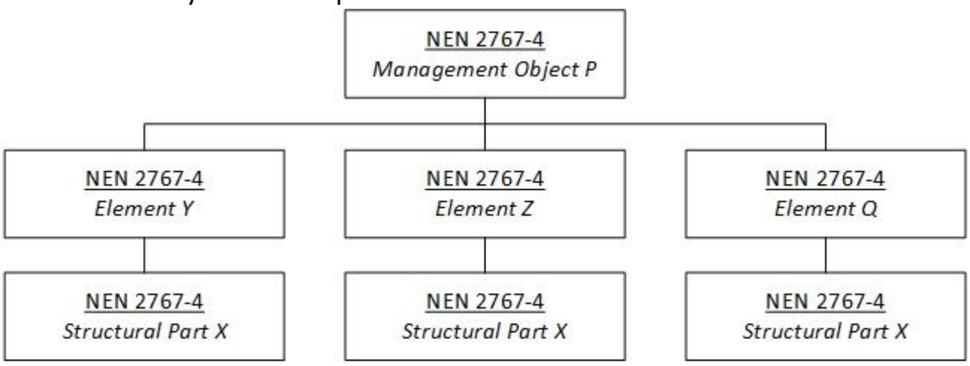


Figure 7-16: NEN2767-4 parent element inconsistency

### 7.4.2 Validation prototype tool 2

Output validation of prototype tool 2.1 is done using the RDF validation service of W3C. The RDF file exported by this tool is parsed at this server resulting in RDF triples, optionally the RDF graph and the announcement that the document validated successfully (figure7-17).

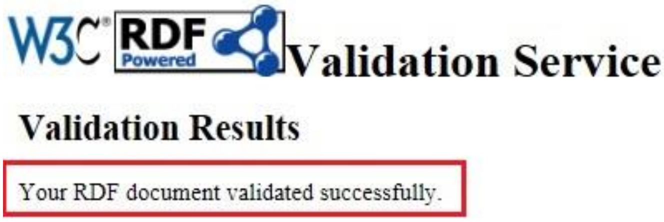



Figure 7-17: RDF validation (W3C, 2015).

To test the completeness of the generated RDF file the second and last tool is used. The second tool, tool 2.2 relies on the RDF file for risk visualization and data representation. In the case that not all data is included by transformation of tabular FMECA data into RDF some elements are not visualized according to the risk colors. In addition, when data would be lost in transformation to RDF, incomplete property representation would be noticeable easily. Detailed inspection showed no situations of incomplete or incorrect data representation. However, one remarkable inconsistency in element color representation was discovered (figure 7-18). In depth research showed this was not attributable to missing or incorrect RDF data, appropriate information is available in the file.

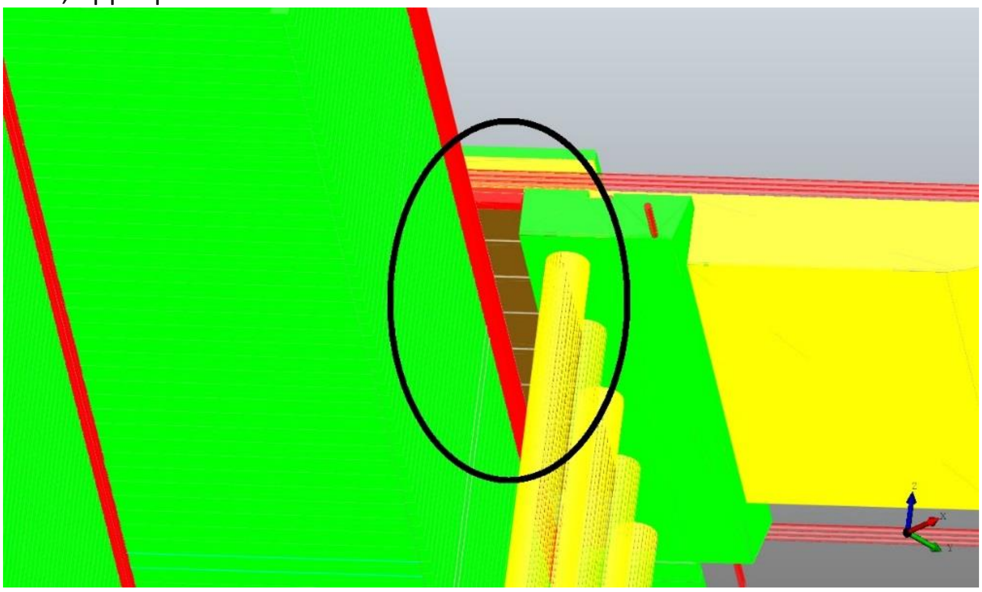


Figure 7-18: Risk visualization error tool 2.

Hence, another failure cause should be appointed. Incorrect visualization and no RDF property representation of the element visualized in figure 7-18 are caused by faulty classification mappings. The RWS OTL to NEN2767-4 classification mapping shows an inconsistency for this specific element. The NEN2767-4 classification is mapped to multiple RWS OTL classifications. Figure 7-19 illustrates this specific classification mapping inconsistency.

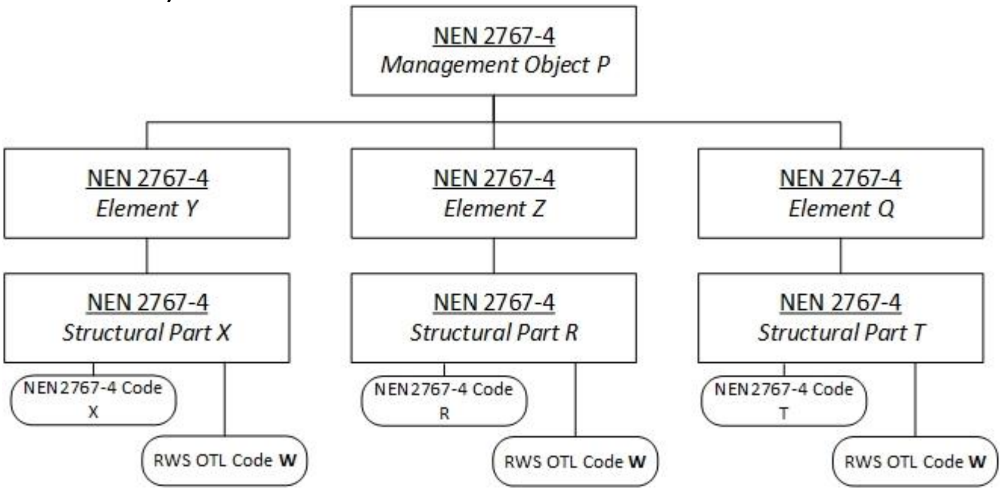



Figure 7-19: Classification mapping inconsistency

## 7.5 Tool limitations

Due to the tools' prototype status some limitations can be distinguished. First, large BIM models are expected to reduce the prototype's performance significantly. Tests using the case model of around 5 megabytes proved to cause no issues. Furthermore, development is based on IFC version 2x3, slight modifications are needed for IFC2x4 compatibility.

The application is designed to work exclusively with models containing Rijkswaterstaat OTL element classifications and O&M data based on the NEN2767-4 classification. Suitability with comparable element classifications requires slight script adaptations.

Due to some FMECA simplifications, the application only handles risk-based O&M data on element level. Information of parent classes are not taken into account. In addition, only one failure mode for each element is defined while in practice an element may have multiple failure modes. Expectations are that, due to the purpose of this research, simplifications are of no influence on the outcomes due to emphasis on information management and not on the data itself.

# 8 CONCLUSION

The main research question is "How can IFC models be enriched semantically to improve risk-based operations & maintenance of an AEC/ FM project?**".** This question can be answered by discussing the six sub-research questions. Therefore, the sub-questions are discussed first and thereafter the main research question is answered.

- What is the added value of Industry Foundation Classes (IFC) in Building Information Modeling use for O&M phases?

The added value of Industry foundation Classes (IFC) for BIM-enabled O&M is in particular the ability to structure O&M information efficiently by linking to the geometric representation of building elements. Thereby providing easy access to O&M information by enabling any user to select an element and access all associated information. In addition, due to the link between elements and related information, data visualization possibilities emerge as demonstrated in chapter 7. Furthermore, extraction of as-built information (e.g. quantities) from the model, results in more efficiency of FM and less mistakes. IFC provides an interoperable model standard which is widely accepted by the industry and therefore a lot of compatible software applications exist. Interviewed asset managers acknowledge the added values as discussed while questioning the cost effectiveness of such approaches if no as-built IFC model is handed over from preliminary design/ construction phases.

- Which approaches can be employed to semantically enrich IFC model populations and what are their possibilities and limitations?

Approaches to semantically enrich IFC model populations are discussed and applied in chapter seven. The first approach, demonstrated by tool 1 can be to directly enrich the model with O&M data by adding new or updated *IfcPropertySets*. This capability is standard available in the IFC standard, however by application and testing of this approach some limitations are discovered. First, information concealed in the semantic structure of IFC is harder to query and reuse. Second, versioning risks exist if by every adjustment of the native data repository and export has to be made to IFC and distributed amongst all stakeholders. Finally, by enrichment of IFC with all possible associated O&M information, file pollution can result in error prone models. An advantage however is the ability to view and access this data using the wide variety of IFC viewers available.

Another approach, as demonstrated by tool 2, can be to utilize semantic web technologies by conversion of associated model information to RDF. Hereby, a common element indicator or element indicators with mappings between them are essential. Testing this approach revealed that disadvantages of the previous approach are avoided. However, except the tool created for this research, there are currently no software applications available with IFC and RDF compatibility.

- What are object classifications and how can they be of added value in risk-based operation & maintenance?

Object classifications are used to classify object collections according to their similarity and providing a hierarchal structure to these collections. For interoperability within an object oriented working method, classifications are essential. Classifications facilitate that anyone within the sector can communicate about those products without misunderstandings. The value of object classifications for risk-based O&M is to standardize information (e.g. failure modes, risk values) according to the classification for future reuse.

- What are important IFC model conditions for handover towards the O&M phase?

In an optimal situation IFC models are handed over representing the as-built situation. In addition, standardized object classification and Globally Unique Identifier (GUID) for every object are indispensable. Furthermore, optimally no object properties should be in the IFC model itself due to the disadvantages as discussed at the second sub-question. Preferably, object properties are available using semantic web technologies with the application of common element indicators (classification). In contrast to the often assumed preferred situation of handing over all possible and highly detailed information to O&M, interviews revealed that this is not recommendable. Making a good data selection prior to handover is essential. When not done correct, a lack of data results in rework during O&M while transferring to much information results in extra work due to the obligation to keep all this data up to date during the entire exploitation and demolition phase.

- What is risk-based operation & maintenance and how can this be applied practically?

Risk-based O&M is about finding the optimal balance between structural reliability and lifecycle cost of deteriorating buildings. Risk-based O&M is used to develop a detailed maintenance plan and/or strategy for safe and fault free operation in a cost effective manner. Practically, risk-based O&M can be applied according to the FMECA methodology which is generally performed in tabular form.

- How could risk-based O&M data be reused in future projects?

By application of an object classification and semantic web technologies, risk-based O&M data can be reused in future projects. This implies description of data using Resource Description Framework (RDF) enabling SPARQL queries to be employed. SPARQL queries on the basis of object classifications can, for example, result in an average risk obtained from all in the past defined risk values of various projects belonging to a specific object class.

Finally, based on the conclusions regarding all sub-research questions the main research question “How can IFC models be enriched semantically to improve risk-based operations & maintenance of an AEC/ FM project?**”** can be answered.
IFC models can be enriched semantically to improve risk-based O&M of an AEC/ FM project directly within the IFC model structure itself by adding FMECA *IfcPropertySets as*

*demonstrated by tool 1*. However, as discussed this brings the disadvantages of IFC pollution, versioning risks and lack of data reusability. Therefore, a second methodology is tested and applied which avoids these disadvantages by application of semantic web technologies (tool 2). The FMECA tabular data is transformed to RDF by adding semantic meaning, triples from the resulting RDF can then be linked to IFC objects due to the application of classifications in both data repositories. While avoiding the disadvantages of the first approach, visualization and property viewing results are the same. Research outcomes are considered scientific relevant due to the vast amount of (distributed) data and models available in current AEC/FM projects while a link between them is often missing. Scientific research shows a trend of transforming distributed into a 'web of data'. This includes the implementation of semantic web technologies where this research contributes to further industry implementation. The societal relevance concerns the importance of maintaining and obtaining high quality buildings in a low-cost, efficient and effective way where this research has brought proven enhancements. While the proposed approach seems very promising a shift in technologies is required. While the FM industry is considered rather traditional the initiative and incentives for such a shift should come from building owners by stimulation of innovations through more extensive implementation of performance based contracts.

## 8.1 Further research

Due to the scope of this research and associated delimitations, further research can be appointed. First, more extensive model testing of the tools is required to guarantee optimal stability. This includes testing with an IFC model where classifications are stored using *IfcRelAssociatesClassification* and *IfcClassification.*

Second, unless applicable and usable tools are created, further development of these tools would result in significant benefits. Further development can be done in terms of compatibility with more vocabularies and datasets available as linked open data (e.g. sensor data, facility condition data, historical data, occupant data, etc.). Prospects are that visualization of e.g. sensor data and condition related information will provide great advantages.

Third, research concerning BIM application in the deconstruction/ demolition phase of a building would contribute to complete lifecycle implementation. And therewith, closing material loops and transition towards e.g. a circular economy.

Fourth, while the FMECA methodology provides guidance for creating an efficient and low-cost maintenance planning, human interpretation remains important. Further research could explore possibilities of automatic generation of a maintenance planning from FMECA data.

Finally, versioning of data repositories by dereferenceable URIs in IFC and RDF require further attention. These so called dereferenceable URI's may point towards the latest FMECA data or a specific version of the concept library. This approach of creating a semantic digital archive is proposed by DURAARK (2013) and key for long term data preservation.

# 9 REFERENCES

Akcamete, a, Akinci, B., & Garrett, J. H. (2010). Potential utilization of building information models for planning maintenance activities. *Proceedings of the International Conference on Computing in Civil and Building Engineering*, 151–157.

Arayici, Y. (2007). Towards building information modelling for existing structures. *Structural Survey*, *26*(3), 210–222. http://doi.org/10.1108/02630800810887108

Arcadis. (2015). BIM according to Arcadis, (March).

Azhar, S., Hein, M., & Sketo, B. (2007). Building Information Modeling ( BIM ): Benefits , Risks and Challenges.

Azhar, S., Khalfan, M., & Maqsood, T. (2009). Building Information Modeling (BIM): Now and Beyond.

Becerik-Gerber, B., Jazizadeh, F., Li, N., & Calis, G. (2012). Application areas and data requirements for BIM-enabled facilities management. *Journal of Construction …*, (March), 431–442. http://doi.org/10.1061/(ASCE)CO.1943-7862.0000433.

Beetz, J., Coebergh, W., Botter, R., Zlatanova, S., & de Laat, R. (2015). Interoperable data models for infrastructural artefacts - a novel IFC extension method using RDF vocabularies exemplified with quay wall structures for harbors. *eWork and eBusiness in Architecture, Engineering and Construction*, 135–140. http://doi.org/http://dx.doi.org/10.1061/9780784413616.071

Beetz, J., Dietze, S., Berndt, R., & Tamke, M. (2013). Towards The Long-Term Preservation of Building Information Models, 209–217.

Beetz, J., Dietze, S., Berndt, R., & Tamke, M. (2013). Towards The Long-Term Preservation of Building Information Models.

Beetz, J., Leeuwen, J. P. Van, & Vries, B. De. (2005). An Ontology Web Language Notation of the Industry Foundation Classes. *Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction*, 193–198.

Beetz, J., van Leeuwen, J., & de Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *23*(01), 89. http://doi.org/10.1017/S0890060409000122

Belsky, M., Sacks, R., & Brilakis, I. (2015). Semantic Enrichment for Building Information Modeling. *Computer-Aided Civil and Infrastructure Engineering*, *00*, n/a–n/a. http://doi.org/10.1111/mice.12128

Berners-Lee, T., Shadbolt, N., & Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, *21*(3), 96–101. http://doi.org/10.1109/MIS.2006.62

BIR. (2008). Dutch BIM Levels, 0–1.

BIR. (2015). Open BIM Standards, (2). Retrieved from www.bouwinformatieraad.nl

BuildingSMART. (2009). IFD Library. *Framework*, (April), 1–9. Retrieved from https://www.uneto-vni.nl/cms/streambin.aspx?requestid=EA0F4F6F-560D-4D15-9BD6-0B651404B812

BuildingSMART. (2011). Information Delivery Manuals. Retrieved October 9, 2015, from http://iug.buildingsmart.org/idms/

BuildingSMART. (2012). Technical vision: openBIM. Retrieved September 10, 2015, from http://www.buildingsmart.org/standards/technical-vision/

BuildingSMART. (2013). BuildingSMART. Retrieved December 11, 2015, from http://www.buildingsmart-tech.org

Businessdictionary. (2014). Online business dictionary. Retrieved October 9, 2015, from http://www.businessdictionary.com/definition/desk-research.html

Carmignani, G. (2009). An integrated structural framework to cost-based FMECA: The priority-cost FMECA. *Reliability Engineering and System Safety*, *94*(4), 861–871. http://doi.org/10.1016/j.ress.2008.09.009

Costa, G., & Madrazo, L. (2015). Connecting building component catalogues with BIM models using semantic technologies: an application for precast concrete components. *Automation in Construction*, *57*, 239–248. http://doi.org/10.1016/j.autcon.2015.05.007

Dankers, M., Geel, F. Van, & Segers, N. M. (2014). A web-platform for linking IFC to external information during the entire lifecycle of a building. *Procedia Environmental Sciences*, *22*, 138–147. http://doi.org/10.1016/j.proenv.2014.11.014

Dawood, N., Vukovic, V., & Kassem, M. (2015). BIM FOR FACILITIES MANAGEMENT : EVALUATING BIM STANDARDS IN ASSET REGISTER CREATION AND SERVICE LIFE PLANNING, *20*(January), 313–331.

Dimyadi, J., Pauwels, P., Spearpoint, M., Clifton, C., & Amor, R. (2015). Querying a Regulatory Model for Compliant Building Design Audit. *Proceedings of the 32nd CIB W78 Conference 2015, 27th - 29th October 2015, Eindhoven, The Netherlands*.

DURAARK. (2013). *Ontological Framework for a Semantic Digital Archive*.

Eadie, R., Browne, M., Odeyinka, H., McKeown, C., & McNiff, S. (2013). BIM implementation throughout the UK construction project lifecycle: An analysis. *Automation in Construction*, *36*, 145–151. http://doi.org/10.1016/j.autcon.2013.09.001

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2008). *BIM Handbook*.

ETIM. (2015). ETIM Classification. Retrieved January 8, 2016, from http://www.etim-international.com

Fruguglietti, E., Pasqualato, G., & Sagula, R. (2012). The "Maintenance Manual" for important infrastructure project, from the design to the implementation after construction. *Bridge Maintenance, Safety, Management, Resilience and Sustainability*, 3058–3065. Retrieved from http://www.gruppo-sina.it/pdf/public/GS_1289_FPAP_PDF.PDF

GCCG. (2011). A Report for the Government Construction Client Group – BIM working strategy Client Group, (March), 1–7.

Geonovum. (2015). Standaarden BGT|IMGeo. Retrieved January 8, 2016, from http://www.geonovum.nl

Grilo, A., & Jardim-Goncalves, R. (2010a). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*, *19*(5), 522–530. http://doi.org/10.1016/j.autcon.2009.11.003

Grilo, A., & Jardim-Goncalves, R. (2010b). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*, *19*(5), 522–530. http://doi.org/10.1016/j.autcon.2009.11.003

Gupta, A., Cemesova, A., Hopfe, C. J., Rezgui, Y., & Sweet, T. (2014). A conceptual framework to support solar PV simulation using an open-BIM data exchange standard. *Automation in Construction*, *37*, 166–181. http://doi.org/10.1016/j.autcon.2013.10.005

Hallberg, D., & Tarandi, V. (2011). On the use of open BIM and 4D visualisation in predictiv life cycle management system for construction, *16*, 445–466. Retrieved from 2011_26.content.05100_On the use of open bim and 4d visualisation in a predictive life cycle management system for construction works.pdf

Heinen, J. (2015). *The added value of Building Information Models in the Operation & Maintenance processes . Mapping between Building Information Models and a FCA*. Eindhoven University of Technology.

Hilbe, J. M. (2014). *Research methods for Engineers*. http://doi.org/http://dx.doi.org/10.1017/CBO9781139542326.004

Hoeber, J., Alsem, D., & Willems, P. (2015). “The management of information over the life-cycle of a construction project using open-standard BIM.” *Proc. of the 32nd CIB W78 Conference 2015, 27th-29th 2015, Eindhoven, The Netherlands*.

Horner, R. M. W., El-Haram, M. a., & Munns, A. K. (2001). Building maintenance strategy: a new management approach. *Journal of Quality in Maintenance Engineering*, *3*(4), 273–280. http://doi.org/10.1108/13552519710176881

Howard, R. (2006). Describing the Changes in Architectural Information Free-Form Architectural Expression. *ITcon*, *11*(January), 395–408. Retrieved from http://itcon.org/2006/29/

Huber, D., Adan, A., & Okorn, B. (2011). Automating the Creation of As-built Building Information Models Methods for Automatically Modeling and Representing As-built Building Information Models. *Proceedings of 2011 NSF Engineering Research and Innovation Conference, Atlanta, Georgia*.

Khan, F. I., & Haddara, M. (2004). Risk-Based Maintenance (RBM): A new approach for process plant inspection and maintenance. *Process Safety Progress*, *23*(4), 252–265. http://doi.org/10.1002/prs.10010

Khosrowshahi, F., & Arayici, Y. (2012). Roadmap for implementation of BIM in the UK construction industry. http://doi.org/10.1108/09699981211277531

Koukias, A., Nadoveza, D., & Kiritsis, D. (2013a). Semantic Data Model for Operation and Maintenance of the Engineering Asset. In *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services* (pp. 49–55). Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-40361-3_7

Koukias, A., Nadoveza, D., & Kiritsis, D. (2013b). Semantic Data Model for Operation and Maintenance of the Engineering Asset. In *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services* (pp. 49–55).

Kreider, R., & Messner, J. (2013). The Uses of BIM: Classifying and Selecting BIM Uses. *State College-Pennsylvania*, (September), 0–22. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:The+Uses+of+BIM+Classifying+and+Selecting+BIM+Uses#5

Laakso, M., & Kiviniemi, A. (2012). The IFC standard-a review of history, development, and standardization. *Journal of Information Technology in …*, *17*(May), 134–161. Retrieved from https://helda.helsinki.fi/handle/10138/35215

Lavy, S., & Jawadekar, S. (2013). A Case Study of Using BIM and COBie for Facility Management.

LDAC. (2014). *Joint Workshop on Linked Data in Architecture and Construction 2014*. Retrieved from http://linkedbuildingdata.net/events/ldac2014/

LDAC. (2015). 3rd International Workshop on Linked Data in Architecture and Construction.

Lipol, L., & Haq, J. (2011). Risk analysis method: FMEA/FMECA in the organizations. *International Journal of Basic & Applied …*, (October). Retrieved from http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=20771223&AN=69907107&h=23GdvURWF9D80qPpy16nm80MYw40fn3QgUgdBhG5sPo3QpBvPB39Pk9FLGnrio6ZJG2FTcgTas2OkY2+dWSWtg==&crl=c

Liu, R., & Issa, R. R. a. (2013). BIM for facility management: Design for maintainability with BIM tools. *ISARC 2013 - 30th International Symposium on Automation and Robotics in Construction and Mining, Held in Conjunction with the 23rd World Mining Congress*, (210), 572–580. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84893529225&partnerID=tZOtx3y1

Love, P. E. D., Simpson, I., Hill, A., & Standing, C. (2013). From justification to evaluation: Building information modeling for asset owners. *Automation in Construction*, *35*, 208–216. http://doi.org/10.1016/j.autcon.2013.05.008

McGlinn, G. (2015). CSVW Tutorial. Retrieved October 2, 2015, from http://phaedrus.scss.tcd.ie/buildviz/csvw/CSVW Tutorial.pdf

NBIMS. (2007). *National Building Information Modeling Standard*. *Building*. Retrieved from http://www.buildingsmartalliance.org/index.php/nbims/

NCHRP. (2009). *Performance-Based Contracting for Maintenance*. *Transportation*.

Nederveen, S., Luiten, B., & Böhms, M. (2015). Linked Data for Road Authorities. *Proc. of the 32nd CIB W78 Conference 2015, 27th-29th 2015, Eindhoven, The Netherlands*.

Oh, M., Lee, J., Hong, S. W., & Jeong, Y. (2015). Integrated system for BIM-based collaborative design. *Automation in Construction*, *58*, 196–206. http://doi.org/10.1016/j.autcon.2015.07.015

Pauwels, P., Törmä, S., Beetz, J., Weise, M., & Liebich, T. (2015). Linked Data in Architecture and Construction. *Automation in Construction*, *57*, 175–177. http://doi.org/10.1016/j.autcon.2015.06.007

Pauwels, P., Van Deursen, D., De Roo, J., Van Ackere, T., De Meyer, R., Van de Walle, R., & Van Campenhout, J. (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. http://doi.org/10.1017/S0890060411000199

Pazlar, T., & Turk, Ž. (2008). Interoperability in practice: Geometric data exchange using the IFC standard. *Electronic Journal of Information Technology in Construction*, *13*(July 2007), 362–380.

Pruvost, H., Benevolenskiy, a, Schapke, S., & Scherer, R. (2012). Considering Project Risks in Reference Modeling of Construction Processes. *Proceedings of the 19th EG-ICE International Workshop*, 1–10. Retrieved from https://www.cms.bv.tum.de/eg-ice2012/proceedings/papers/53.pdf

RTU. (2015). Uithoflijn. Retrieved December 21, 2015, from http://www.uithoflijn.nl

Schmidt, R., Riedel, G. J., & Kangas, K. (2011). Risk assessment using design review based on failure mode. *Reliability and Maintainability Symposium (RAMS), 2011 Proceedings-Annual*, 1–6.

STABU. (2015). Elementenmethode NL/SfB. Retrieved January 8, 2016, from http://www.stabu.org/diensten/nlsfb/#.Vo9qdhXhCUk

Steel, J., Drogemuller, R., & Toth, B. (2012). Model interoperability in building information modelling. *Software and Systems Modeling*, *11*(1), 99–109. http://doi.org/10.1007/s10270-010-0178-4

Straub, A. (2009). Cost savings from performance-based maintenance contracting. *International Journal of Strategic Property Management*, *13*(3), 205–217. http://doi.org/10.3846/1648-715X.2009.13.205-217

Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*, *18*(3), 357–375. http://doi.org/10.1016/j.autcon.2008.10.003

Vanlande, R., Cruz, C., & Nicolle, C. (2008). IFC and buildings lifecycle management. *Journal of Automation in Construction*, *18*, 70–78. http://doi.org/10.1016/j.autcon.2008.05.001>

Verbaan, W., Visser, E., Koe, D. De, Boer, J. De, & Voet, L. Van Der. (2014). Beter beheer met bim.

Visser, E., Boer, J. De, & Voet, L. Van Der. (2013). *BIM IN BEHEER BIEDT KANSEN*.

Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction*, *38*, 109–127. http://doi.org/10.1016/j.autcon.2013.10.023

W3C. (2008). SPARQL Query Language for RDF. Retrieved January 3, 2016, from http://www.w3.org/TR/rdf-sparql-query/#introduction

W3C. (2015a). RDF Validation Service. Retrieved January 4, 2016, from http://www.w3.org/RDF/Validator/

W3C. (2015b). Semantic Web. Retrieved January 11, 2016, from http://www.w3.org/standards/semanticweb/

WBDG. (2015). Facilities Operation & Maintenance. Retrieved October 20, 2015, from https://www.wbdg.org/om/om.php

WMG. (2009). Failure Modes, Effects & Criticality Analysis: Product Excellence Using Six Sigma. *University of Warwick*.

Woldesenbet, A. K. (2014). Highway Infrastructure Data and Information Integration & Assessment Framework : A Data- Driven Decision-Making Approach.

Woodside, A. G., & Wilson, E. J. (2003). Case study research methods for theory building. *Journal of Business & Industrial Marketing*, *18*(6/7), 493–508. http://doi.org/10.1108/08858620310492374

Zhou, A., Yu, D., & Zhang, W. (2015). A research on intelligent fault diagnosis of wind turbines based on ontology and FMECA. *Advanced Engineering Informatics*, *29*(1), 115–125. http://doi.org/10.1016/j.aei.2014.10.001

# 10 APPENDICES

**APPENDIX 1 – EXPERT INTERVIEWS**

**APPENDIX 2 – PYTHON SCRIPT TOOL 1**

**APPENDIX 3 – FLOWCHARTS TOOL 1**

**APPENDIX 4 – GRAPHICAL USER INTERFACES (GUI'S)**

**APPENDIX 5 – PYTHON SCRIPT TOOL 2**

**APPENDIX 6 – CSVW JSON**

**APPENDIX 7 – FMECA ONTOLOGY**

**APPENDIX 8 – FLOWCHART TOOL 2**

**APPENDIX 9 – PARTIAL RDF**

# Appendix 1: Expert interviews

In this appendix all interviews are enclosed. Appendix 1.1 contains the interview questions in English. Appendix 1.2 contains a Dutch summary of the respondent's answers.

## Appendix 1.1: Interview Questions (English)

| Name: | Company/ position: |
| --- | --- |

1. How would you rate efficiency of current (traditional) handover processes between construction phase and O&M? Which approach/ software is currently used And what are the pros and cons?

2. Have you used Building Information Modeling (BIM) on any previous/current projects? If yes, also in relation to operation and maintenance (O&M) phases?

3. Do you think BIM could improve handover process as referred in the previous question?
   i. If yes, how?
   ii. If no, why?

4. Should the client's role be passive or active during the handover process, and why?

5. Do you think linking files to a BIM model will have potential in the O&M field?
   a. If no, why not?
   b. If yes, which files would you definitely link to this model?
      i. user manuals;
      ii. warranty documents;
      iii. Product information on suppliers website;
      iv. Product data sheet;
      v. Condition reports;
      vi. Condition assessment photo's;
      vii. ……………
      viii. ……………

6. Do you think adding new attributes/ linking files in a 3D environment will be useful when structuring information? Both in building as in infrastructural projects?

7. Do you think IFC model files are appropriate to act as a BIM model to which (all) other information is linked?

8. Which features of an application used to link data to an object within a BIM model should certainly be implemented (e.g. automatic file linking, querying linked files, etc.)? Can you describe a related use case?

## Appendix 1.2: Summarized interviews (Dutch)

| | |
|---|---|
| **Naam:** *Marcel van Bavel* | **Bedrijf/ functie:** *BASED Tilburg, Eigenaar* |

1. Hoe zou u de efficiëntie van de huidige (traditionele) overdracht tussen realisatie en beheer en onderhoud (B&O) waarderen? Op welke manier vindt de overdracht plaats en welke software wordt hiervoor gebruikt? Wat zijn de voor- en nadelen?

*Niet efficiënt, de overdracht is zeer belangrijk maar tot op heden een groot probleem. Daarom zijn er momenteel veel verschillende ontwikkelingen gaande en applicaties op de markt die focussen op BIM voor beheer en onderhoud (B&O). Door de Engelse overheid wordt BIM vanaf 2016 voorgeschreven op alle projecten wat aantoont dat er steeds meer vraag komt naar oplossingen.*

2. Heeft u Building Information Mondeling (BIM) in eerdere/ lopende projecten gebruikt? Zo ja, ook met betrekking tot de B&O fase?

*Ja, ik ben onder andere bezig met een project voor Schiphol om te onderzoeken hoe BIM gegevens overgedragen moeten gaan worden naar O&B.*

3. Denkt u dat BIM het overdrachtsproces van realisatie naar B&O kan verbeteren/ optimaliseren?
   a. Zo nee, waarom niet?
   b. Zo ja, hoe?

*Absoluut, wel is het erg belangrijk om na te denken over het Level of Detail (LOD) en Level of Information (LOI) van het over te dragen model. Voor O&B is een hoog LOD helemaal niet interessant, de geometrische informatie dient uitsluitend als visuele weergave voor de beheerder om informatie behorende bij een object terug te vinden. Een over te dragen model zou prima overgedragen kunnen worden in LOD300 in plaats van LOD500. De LOI is veel belangrijker. Meer informatie is te vinden in de PAS1192-2/-3.*

4. Moet de rol van de opdrachtgever actief of passief zijn tijdens de overdracht, en waarom?

*Actief, een klant moet bewust zijn dat ze een goede uitvraag doen. De vraag stellen: moeten we niet toch een ander systeem aanschaffen?*

5. Heeft het koppelen van documenten/ informatie aan een BIM-model potentie betreffende de B&O fase?
   a. Indien nee, waarom niet?
   b. Zo ja, welke documenten/ informatie zouden zeker gelinkt moeten worden aan het model?
      i. Gebruikershandleidingen
      ii. Garantietermijnen
      iii. Productinformatie op de website van de leverancier
      iv. Productinformatie document
      v. Conditierapporten
      vi. Conditiefoto's
      vii. …..

*Handig zou het zijn als niet alleen de (onoverzichtelijke) conditierapporten gekoppeld zijn, maar vooral de huidige conditie van een object. Zo heb je in een oogopslag inzicht in de situatie.*
*Ook een directe link naar een stap voor stap handleiding voor bepaalde handelingen zou interessant zijn. Bijvoorbeeld een handleiding op 'Ikea' niveau: 'hoe vervang ik een filter van deze luchtbehandelingsinstallatie?'.*
*Daarnaast is het bruikbaar om in de toekomst het model te kunnen bevragen met vragen als: 'Bij welke groep in de groepenkast hoort deze lamp?' of 'Welke luchtbehandelingsinstallatie hoort bij dit afzuigpunt?'.*

6. Denkt u dat het toevoegen van nieuwe eigenschappen/ koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie? Zowel in woning- en utiliteitsbouw als in infrastructurele projecten?

*Absoluut, het idee achter COBie is hier een voorbeeld. Hoewel COBie sheets niet erg overzichtelijk zijn is vooral het gedachtegoed om informatie te filteren en weer te geven voor B&O goed. Echter ook in COBie sheets zit ook veel informatie die de stempel 'nice to have' zou kunnen krijgen en daarmee niet noodzakelijk zijn voor goed B&O.*

7. Denkt u dat IFC-modellen geschikt zijn om te dienen als middelpunt waaraan (alle) andere relevante informatie gekoppeld wordt?

*Prima bestandsformaat, het is GUID gebaseerd en daar gaat het om. Mocht er een ander formaat zijn die dit ook ondersteund en breed geaccepteerd zijn zou dit ook prima zijn. Voor de bouw is IFC momenteel leidend dus zeker geschikt. Momenteel zijn er veel ontwikkelingen op dit gebied en meerdere IFC viewers te gebruiken tijdens B&O zoals:*

*A. OGDB viewer van Fimble;*
*B. BIM+ viewer;*
*C. Autodesk view and data API;*
*D. BIM Sync;*
*E. Trimble connect, Trimble Field 3D*
*F. xBIM*

8. Welke functies zou een applicatie, bedoeld om documenten/ eigenschappen te koppelen aan een object in een IFC-model, zeker ook nog moeten hebben (bijvoorbeeld automatisch documenten/ eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u daarbij een bijbehorende use case beschrijven?

*Uitgangspunten voor informatie-uitwisseling moeten worden vastgelegd in een Informatie leveringsspecificatie (ILS). Het is cruciaal om eerst een model te checken door de informatie te valideren met het ILS. Valideren of bijvoorbeeld alle velden gevuld zijn in het model en of de basisopbouw van het model goed is.*
*Functies om het model te bevragen en waarbij resultaat wordt weergegeven in applicatie zouden handig zijn. De resultaten hoeven niet altijd in het IFC model opgeslagen te worden. Dit kan resulteren in een grafisch overzicht van de condities van objecten in een model waardoor je in één oogopslag inzicht hebt in alle objecten met bijv. een 'rode' conditie.*

| **Naam:** *Floris van Ruth* | **Bedrijf/ functie:** *Arcadis, Project manager WV&I – Asset Management* |
|---|---|

1. Hoe zou u de efficiëntie van de huidige (traditionele) overdracht tussen realisatie en beheer en onderhoud (B&O) waarderen? Op welke manier vindt de overdracht plaats en welke software wordt hiervoor gebruikt? Wat zijn de voor- en nadelen?

*Groot verschil tussen nieuwbouw of bestaande bouw. Bij bestaande bouw worden er prestatiecontracten voor verschillende objecten opgesteld. Hiervoor worden vaak verschillende beheermanagement systemen gebruikt, onder andere Kerngis, Dtb en Ultimo. In Kerngis worden verschillende attributen aan een object gekoppeld. Voor kunstwerken wordt de applicatie Disc vooral gebruikt. Door grote variatie aan toegepaste beheermanagement systemen ontstaan er interoperabiliteitsproblemen. Vaak verplichting om te werken met software van opdrachtgever, bijvoorbeeld van Rijkswaterstaat (RWS).*
*Het probleem blijft over het algemeen dat gegevens niet volledig of onjuist zijn.*
*Continu dient daarom de controle plaats te vinden betreffende: actualiteit, betrouwbaarheid en compleetheid. Het is een illusie dat beheermanagementgegevens exact hetzelfde zijn als as-built situatie. Meestal zijn verschillende stakeholders die wijzigingen aanbrengen en daardoor is het slecht te beheersen.*

2. Heeft u Building Information Mondeling (BIM) in eerdere/ lopende projecten gebruikt? Zo ja, ook met betrekking tot de B&O fase?

*Wel object georiënteerd werken op objectniveau waardoor informatie gekoppeld kan worden, echter geen toepassing van een gezamenlijk bouwmodel.*
*Bij beheer en onderhoud van nieuwbouw is BIM makkelijker toe te passen dan bij bestaande projecten. In praktijk wordt lijninfra vaak in 2D gemodelleerd, puntobjecten (bijv. sluis of tunnel) kunnen wel in 3D gemodelleerd zijn. Echter is het belangrijk om af te vragen of 3D modellen de investering waard zijn, vaak brengt dit hoge kosten met zich mee. Daarnaast moeten opdrachtgevers ook mee gaan in deze innovatie en niet vasthouden aan traditionele methodes.*

3. Denkt u dat BIM het overdrachtsproces van realisatie naar B&O kan verbeteren/ optimaliseren?
    a. Zo nee, waarom niet?
    b. Zo ja, hoe?

*Ja, DBFM of DBM contracten zou dit zeker voordelen opleveren. Bestaande situatie blijft lastig vanwege de investering die gemaakt moet worden om de huidige situatie te modelleren. Hierbij moet je uiteraard de vraag stellen of deze investering zoveel toegevoegde waarde heeft dat het rendabel is.*

4. Moet de rol van de opdrachtgever actief of passief zijn tijdens de overdracht, en waarom?

*Opdrachtgever heeft een actieve, toetsende rol. De opdrachtgever moet de mogelijkheid bieden om bij hun gegevens te kunnen zodat de opdrachtnemer informatie in systeem opdrachtgever kan verwerken.*

5. Heeft het koppelen van documenten/ informatie aan een BIM-model potentie betreffende de B&O fase?
    a. Indien nee, waarom niet?
    b. Zo ja, welke documenten/ informatie zouden zeker gelinkt moeten worden aan het model?
        i. Gebruikershandleidingen
        ii. Garantietermijnen
        iii. Productinformatie op de website van de leverancier
        iv. Productinformatie document
        v. Conditierapporten
        vi. Conditiefoto’s
        vii. …..

*Ja, het zou interessant zijn om faalmechanisme te koppelen aan objecten volgens de "Failure Mode Effect & Criticality Analysis" (FMECA).*
*In de huidige werkmethode worden objecten gerangschikt volgens een uniforme decompositie met standaard gegevens. Relatics en een SEMM model worden gebruik om informatie te koppelen aan objecten.*

*Informatiebehoefte is zeer verschillend tussen opdrachtgeverskant en de aannemer. Het is belangrijk om een goede selectie te maken in alle beschikbare informatie. Zodat overbodige informatie er uit gefilterd wordt. Het heeft niet de voorkeur om zomaar alles in een model te stoppen, dit moet namelijk ook up-to-date gehouden worden. Als informatie veroudert is kan het ook niet meer worden gebruikt.*

6. Denkt u dat het toevoegen van nieuwe eigenschappen/ koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie? Zowel in woning- en utiliteitsbouw als in infrastructurele projecten?

*Ja en nee, 2D is vaak voldoende bij infrastructurele projecten, ook in 2D is het koppelen van informatie zeer bruikbaar en dit gebeurt ook al. Alleen bij zeer lastige infrastructurele projecten is 3D van meerwaarde (bijv. tunnels, bruggen en sluizen).*

7. Denkt u dat IFC-modellen geschikt zijn om te dienen als middelpunt waaraan (alle) andere relevante informatie gekoppeld wordt?

*Niet bekend met IFC.*

8. Welke functies zou een applicatie, bedoeld om documenten/ eigenschappen te koppelen aan een object in een IFC-model, zeker ook nog moeten hebben (bijvoorbeeld automatisch documenten/ eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u daarbij een bijbehorende use case beschrijven?

*Toegang tot historische informatie van een object zoals informatie over de leverancier, onderhoudshistorie, storingsinformatie en onderhoudskosten zou zeker bruikbaar zijn. Soms wordt deze informatie zelfs gebruik makende van traditionele methodes nog niet opgeslagen in beheermanagementsystemen. Als dit wel gebeurt is dit link naar een eenvoudige database met storingsinformatie, onderhoudshistorie, onderhoudskosten.*

| | |
|---|---|
| **Naam:** *Jos Bakker* | **Bedrijf/ functie:** *Arcadis, Senior Advisor WV&I, Asset Management* |

1. Hoe zou u de efficiëntie van de huidige (traditionele) overdracht tussen realisatie en beheer en onderhoud (B&O) waarderen? Op welke manier vindt de overdracht plaats en welke software wordt hiervoor gebruikt? Wat zijn de voor- en nadelen?

*Momenteel inventariseren aannemers pas vlak voor de overdracht naar beheer en onderhoud hoe en wat er overgedragen moet worden. Dit resulteert in een inefficiënte overdracht. Vanuit de opdrachtgever worden er vaak goede informatiespecificaties geleverd waarin beschreven wordt wat er overgedragen moet worden.*
*Het komt zelfs voor dat aannemers de een gedeelde van de laatste termijn afwegen tegen de kosten die gemaakt moeten worden om informatie goed over te dragen. Het kan zo zijn dat de kosten voor informatieoverdracht hoger zijn dan het resterende termijn waar een aannemer recht op heeft. In het verleden hebben aannemers op basis hiervan wel eens besloten om overdacht niet plaats te laten vinden.*
*Vanuit BIM beter omdat er actualisaties tijdens project plaats vinden.*

2. Heeft u Building Information Mondeling (BIM) in eerdere/ lopende projecten gebruikt? Zo ja, ook met betrekking tot de B&O fase?

*Ja, in ongeveer 80 procent van alle projecten wordt BIM toegepast. Dit omvat onder andere object georiënteerd werken en aan locatie gekoppelde informatie. RWS zal vermoedelijk medio 2015 een prestatiecontract als BIM beproevingsproject aanwijzen. Dit proces kan nog beter worden georganiseerd gebruik makende van 'Object Type Libraries' (OTL).*

3. Denkt u dat BIM het overdrachtsproces van realisatie naar B&O kan verbeteren/ optimaliseren?
   i. Zo nee, waarom niet?
   ii. Zo ja, hoe?

*Hier heb ik weinig ervaring mee, vooral vanwege de klantvraag. Momenteel wordt gebruik gemaakt van een combinatie van System Enigeering (Relatics), GIS beheersoftware van ARCADIS (Atrium) en een beheer management systeem (BMS) van opdrachtgever. In de situatie zouden Relatics en Atrium nog beter moeten samenwerken, hier ontbreekt namelijk de directe koppeling. Hierdoor ontstaan er tijdens beheer onduidelijkheden over wat de eisen behorende bij een object zijn. Wat waren ook alweer de eisen aan dit object in Relatics?*

*Rijkswaterstaat (RWS) is momenteel een grote slag aan het slaan om beheersystemen op orde te krijgen, momenteel maken ze namelijk gebruik van meer dan 150 soorten systemen. RWS heeft een objecttypenbibliotheek om zo eenduidig concepten te beschrijven gedurende de hele levenscyclus van fysiek gebouwde objecten. Deze is echter meer opgesteld vanuit het oogpunt: 'Welke informatie hebben we?' in plaats van 'welke informatie hebben we nodig?'. Dit betekent dat er veel te veel informatie in deze specificatie opgeslagen wordt. Het is dus belangrijk om een selectie te maken betreffende wel informatie overdragen moet worden en op wat voor detailniveau.*

4. Moet de rol van de opdrachtgever actief of passief zijn tijdens de overdracht, en waarom?

*Actief, die doet een kwaliteitscheck. De opdrachtgever moet namelijk de vraag stellen of hetgeen overgedragen conform afspraken en planning is.*

5. Heeft het koppelen van documenten/ informatie aan een BIM-model potentie betreffende de B&O fase?
    a. Indien nee, waarom niet?
    b. Zo ja, welke documenten/ informatie zouden zeker gelinkt moeten worden aan het model?
        i. Gebruikershandleidingen
        ii. Garantietermijnen
        iii. Productinformatie op de website van de leverancier
        iv. Productinformatie document
        v. Conditierapporten
        vi. Conditiefoto's
        vii. …..

*Ja, waardevol zou een koppeling naar de functionele eisen zijn. Zo kan makkelijk de vraag gesteld worden tijdens beheer: 'Voldoet het object nog aan de gestelde eisen?'. Daarnaast zou een koppeling naar een Excel sheet met het faalgedrag zeer bruikbaar zijn. Ook makkelijk inzicht in hoeveelheden en oppervlaktes zou van toegevoegde waarde zijn.*
*Belangrijk blijft het om een onderscheid te maken tussen de informatie bedoeld voor de aannemer en de informatie bedoeld voor de opdrachtgever/ eigenaar.*

6. Denkt u dat het toevoegen van nieuwe eigenschappen/ koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie? Zowel in woning- en utiliteitsbouw als in infrastructurele projecten?

*Ja, zoals aangegeven bij vorige vraag. Uiteindelijk zou dit risico gestuurd beheer en onderhoud moeten faciliteren. Bijvoorbeeld door middel van een export naar een risicomatrix, per element of bouwdeel.*

7. Denkt u dat IFC-modellen geschikt zijn om te dienen als middelpunt waaraan (alle) andere relevante informatie gekoppeld wordt?

-

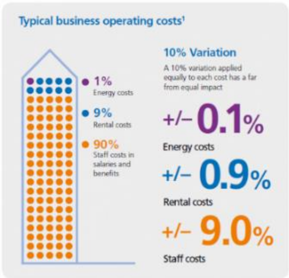
8. Welke functies zou een applicatie, bedoeld om documenten/ eigenschappen te koppelen aan een object in een IFC-model, zeker ook nog moeten hebben (bijvoorbeeld automatisch documenten/ eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u daarbij een bijbehorende use case beschrijven?

*Faciliteren van risico gestuurd beheer per tracé met onderliggende objecten. Hiervoor dient de informatiebehoefte in kaart te worden gebracht. Ook zou een koppeling tussen SE en beheer onderhoud systeem zou van toegevoegde waarde zijn.*

| **Naam:** *Ronald van Aggelen* | **Bedrijf/ functie:** *Root B.V., Directeur* |
|---|---|

1. Hoe zou u de efficiëntie van de huidige (traditionele) overdracht tussen realisatie en beheer en onderhoud (B&O) waarderen? Op welke manier vindt de overdracht plaats en welke software wordt hiervoor gebruikt? Wat zijn de voor- en nadelen?

*Het is de uitdaging om modellen te maken waar volgende fases/ andere disciplines mee verder kunnen. Belangrijk is om een scheiding te maken tussen strategisch en operationeel beheer. Globaal is de kostenverdeling bij kantoren bijvoorbeeld 10% energie/ gebouw tegenover 90% personeelskosten. Een efficiëntieverhoging van 10% levert daardoor veel meer op in de zin van personeelskosten dan energiekosten, dit is weergegeven in figuur 1. Met andere woorden, start met optimalisaties die het meeste rendement opleveren.*




Figuur 10-1: Typical business operating costs. Bron: Health, Wellbeing & Productivity in Offices, 2014

2. Heeft u Building Information Mondeling (BIM) in eerdere/ lopende projecten gebruikt? Zo ja, ook met betrekking tot de B&O fase?

*Root B.V. heeft de applicatie Flatt ontwikkeld om informatie te koppelen aan objecten in IFC. Hiermee is het bijvoorbeeld mogelijk om bewonersgegevens te koppelen aan een ruimte (huurdersmanagement).*
*We maken ook modellen van bestaande bouw (met een lager detailniveau), vooral gefocust op ruimtemanagement. Dit maakt GIS-achtige analyses mogelijk op gebouwniveau.*
*Daarnaast zijn we ook bezig met sensoren om realtime gegevens te koppelen aan ruimtes, bijvoorbeeld temperatuurgegevens en CO2 niveaus.*

3. Denkt u dat BIM het overdrachtsproces van realisatie naar B&O kan verbeteren/ optimaliseren?
    iii. Zo nee, waarom niet?
    iv. Zo ja, hoe?

*Ja, het model helpt bij het valideren van de informatiekwaliteit. Het kan echter nog steeds onduidelijk zijn wat de beheerder nodig heeft aan informatie. Model kan dienen als kapstok voor informatie. Belangrijk blijft dat in vroege fases vastgelegd moet worden welke informatie nodig is voor beheer en onderhoud, Root B.V. gebruikt de RVB BIM norm als leidraad.*

4. Moet de rol van de opdrachtgever actief of passief zijn tijdens de overdracht, en waarom?

*Absoluut actief, hier moet je als eigenaar/ opdrachtgever controle over hebben. Grootste kostenpost voor eigenaar bevindt zich na de realisatie van een object!*

5. Heeft het koppelen van documenten/ informatie aan een BIM-model potentie betreffende de B&O fase?
    a. Indien nee, waarom niet?
    b. Zo ja, welke documenten/ informatie zouden zeker gelinkt moeten worden aan het model?
        i. Gebruikershandleidingen
        ii. Garantietermijnen
        iii. Productinformatie op de website van de leverancier
        iv. Productinformatie document
        v. Conditierapporten
        vi. Conditiefoto's
        vii. …..

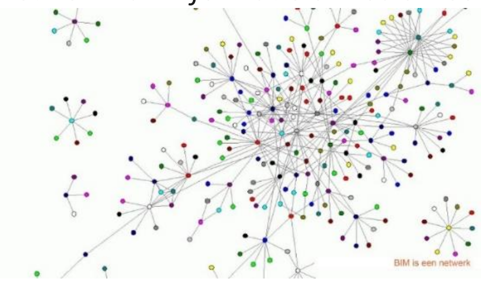
*Ja, in principe kan alle informatie nuttig zijn. Afweging dient gemaakt te worden welke informatie het meeste oplevert (kosten/ baten).*

6. Denkt u dat het toevoegen van nieuwe eigenschappen/ koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie? Zowel in woning- en utiliteitsbouw als in infrastructurele projecten?

*Zie antwoord vorige vraag. Geen ervaring met infrastructurele projecten.*

7. Denkt u dat IFC-modellen geschikt zijn om te dienen als middelpunt waaraan (alle) andere relevante informatie gekoppeld wordt?

*Ja, IFC is een internationale open standaard. Ik denk dat alles aan alles gekoppeld wordt (many to many), maar dat soms informatie indirect gekoppeld is en soms direct. Het is een netwerk van informatie en zoals weergegeven in figuur 2.*




Figuur 10-2; BIM is een netwerk. Bron: http://www.networkweaving.com/blog/2006/04/network-mapping_23.html

8. Welke functies zou een applicatie, bedoeld om documenten/ eigenschappen te koppelen aan een object in een IFC-model, zeker ook nog moeten hebben (bijvoorbeeld automatisch documenten/ eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u daarbij een bijbehorende use case beschrijven?

*Automatische verrijking heeft zeker meerwaarde. Validatie blijft een groot aandachtspunt, op basis van welke eigenschappen (NL/SfB, naam, type, geometrisch, etc.) kan de koppeling gemaakt en gevalideerd worden?*

| **Naam:** *Tom Borst* | **Bedrijf/ functie:** *Arcadis, Teamleader structure / BIM program manager* |
|---|---|

*Tom Borst is constructieadviseur gebouwen van oorsprong, momenteel werkzaam als BIM programmamanager binnen divisie gebouwen. Hierbij onder andere advies betreffende MJOP en risico gestuurd beheer en onderhoud.*

1. Hoe zou u de efficiëntie van de huidige (traditionele) overdracht tussen realisatie en beheer en onderhoud (B&O) waarderen? Op welke manier vindt de overdracht plaats en welke software wordt hiervoor gebruikt? Wat zijn de voor- en nadelen?

*De overdracht naar beheer en onderhoud (B&O) blijft nog een hardnekkig probleem. Bij de projecten die wij hebben worden modellen over het algemeen gemaakt met als doel de realisatie. Er zijn een aantal voorbeelden van opdrachten voor Arcadis om bestaand vastgoed te modelleren en dit model te gebruiken voor tijdens de B&O fase. Als er al vanaf ontwerp met een BIM model gewerkt wordt hangt het van de opdrachtgever af of het model uiteindelijk daadwerkelijk overgedragen wordt. Het is echter in alle gevallen belangrijk dat er voorkant vastgesteld is hoe gegevens aan het eind opgeleverd dienen te worden.*

2. Heeft u Building Information Mondeling (BIM) in eerdere/ lopende projecten gebruikt? Zo ja, ook met betrekking tot de B&O fase?

*Binnen divisie gebouwen worden alle nieuwbouwprojecten volgens BIM principes uitgevoerd. Uitsluitend als vooraf vastgesteld is dat BIM modellen worden gebruikt voor B&O, worden modellen wel eens overgedragen maar dit komt niet vaak voor. Het komt voor dat B&O partijen geen BIM modellen willen gebruiken omdat dit efficiëntere werkprocessen oplevert en daardoor minder uren kunnen boeken op een project.*
*Het probleem is dat nooit van tevoren vastgesteld is welke informatie nodig is voor goed B&O. Vaak stelt Arcadis deze eisen zelf vast d.m.v. aannames. Uiteindelijk worden modellen dan vaak achteraf helemaal aangepast naar wensen van de beheerder om het geschikt te maken voor B&O.*

3. Denkt u dat BIM het overdrachtsproces van realisatie naar B&O kan verbeteren/ optimaliseren?
    v. Zo nee, waarom niet?
    vi. Zo ja, hoe?

*Ja, in een optimale situatie bestaat er eigenlijk geen echte overdrachtsfase meer. Benodigde gegevens voor beheer en onderhoud moeten in de modellen zitten die volgens protocollen zijn gemodelleerd. Dit betekent dan standaardisatie en classificatie van objecten goed geregeld moeten zijn om model voor iedereen toegankelijk en begrijpelijk te maken. Idealiter worden de beheerders al betrokken bij het ontwerp en realisatie zodat het model geschikt is voor de B&O fase. Als voorbeeld zou de beheerder de door hem gebruikte classificatie al tijdens realisatie aan het model kunnen toevoegen.*

4. Moet de rol van de opdrachtgever actief of passief zijn tijdens de overdracht, en waarom?

*Actieve rol van beheerder/ opdrachtgever is zeker vereist. Al vanaf het begin zouden de beheerders betrokken moeten zijn bij het proces. Zo zouden beheerders al tijdens het ontwerpproces invloed kunnen uitoefenen op het uiteindelijke model.*

5. Heeft het koppelen van documenten/ informatie aan een BIM-model potentie betreffende de B&O fase?
    c. Indien nee, waarom niet?
    d. Zo ja, welke documenten/ informatie zouden zeker gelinkt moeten worden aan het model?
        i. Gebruikershandleidingen
        ii. Garantietermijnen
        iii. Productinformatie op de website van de leverancier
        iv. Productinformatie document
        v. Conditierapporten
        vi. Conditiefoto's
        vii. …..

*Het koppelingen van Relatics informatie waarin de eisen gedefinieerd zijn is mogelijk van waarde. Het koppelen van informatie kan momenteel ook geregeld worden in de beheersoftware genaamd Archibus. Risico gestuurd beheer en onderhoud is geïmplementeerd in het MJOP op basis van inzichten van de planner. Een koppeling tussen een BIM model en het daarbij behorende MJOP heeft de potentie om de planner te ondersteunen en de kwaliteit te verhogen.*

6. Denkt u dat het toevoegen van nieuwe eigenschappen/ koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie? Zowel in woning- en utiliteitsbouw als in infrastructurele projecten?

*Op zich wel, 3D modellen bieden de mogelijkheid om objecten te lokaliseren tijden B&O, zoals een ventilatiekanaal in een verlaagd plafond waar onderhoud aan gepleegd dient te worden. Dit levert kan een hogere efficiëntie opleveren voor de beheerorganisatie.*

7. Denkt u dat IFC-modellen geschikt zijn om te dienen als middelpunt waaraan (alle) andere relevante informatie gekoppeld wordt?

*Dit zou kunnen door middel van IFC vooral naar de opdrachtgever toe. Toch merken we dat er informatieverlies optreed bij de export naar IFC, vooral als een IFC weer ingeladen wordt in modelleersoftware zoals Revit.*

8. Welke functies zou een applicatie, bedoeld om documenten/ eigenschappen te koppelen aan een object in een IFC-model, zeker ook nog moeten hebben (bijvoorbeeld automatisch documenten/ eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u daarbij een bijbehorende use case beschrijven?

*Binnen Revit modellen zouden risicoprofielen inzichtelijk gemaakt kunnen worden door middel van kleuren. Uit ervaring blijkt dat beheerders vaak informatie gestuurd zijn en de waarde van dit soort innovaties niet altijd erkennen. Ook hebben wij als divisie behoefte aan een mapping tussen de NEN2767 classificatie en NL-SfB. Ook een koppeling tussen beheer en onderhoudsplanningen en het model zou interessant zijn zoals eerder aangegeven. Transitie naar implementatie van een OTL waarin beheer en onderhoudsparameters zitten is nog ver* *weg. Binnen de divisie gebouwen zijn OTL's minder geïmplementeerd vanwege het grote aanbod aan objecten in vergelijking met de divisie Mobiliteit.*

# Appendix 2: Python script tool 1

```
import sys
import os
import uuid
import ifcopenshell
import ifcopenshell.geom
settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)

from collections import defaultdict
import openpyxl
from openpyxl import Workbook
try:
    template_wb = openpyxl.load_workbook('FMECA_template.xlsx', data_only=True)
    template_sheet = template_wb.get_sheet_by_name('FMECA')
except: print "FMECA template not found. OBS export not possible!"

from lxml import etree
try: xml_file = open('Decompositie_NEN_OTL.xml')
except: print "RWS OTL to NEN2767-4 mapping rules not found. Add mapping rules to continue."
tree = etree.parse(xml_file)

from PyQt4 import QtGui,QtCore
from PyQt4.QtCore import QObject, pyqtSignal
try: from OCC.Display.qtDisplay import qtViewer3d
except: print "trying old version"
try: from OCC.Display.pyqt4Display import qtViewer3d
except: print "no occ path set"

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

guid_selection = None

# Create 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, *args)
        # Create an empty dictionary to keep track of products
        self.objects = {}
    @staticmethod
    def Hash(shape):
        # Return a (semi-)unique number for a shape
        return shape.HashCode(1<<30)
    def Show(self, key, shape, color=None):
        # Add the shape and number to the dictionary of objects
        self.objects[ProductViewer.Hash(shape)] = key
        if color is None:
            self._display.DisplayShape(shape)
        else:
            self._display.DisplayColoredShape(shape, color)
    def mouseReleaseEvent(self, *args):
        # Process selection by parent class
        qtViewer3d.mouseReleaseEvent(self, *args)
        if self._display.selected_shape:
            # Retrieve the key (GlobalId, but can be anything) of the selected
            # object, by computing the Hash again and comparing with the dictionary
            global guid_selection
            guid_selection = (self.objects[ProductViewer.Hash(self._display.selected_shape)])

#main Class of the Geometry Viewer
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()

        #Status bar
        self.main.statusBar()
```

```python
        self.display = self.main.canvas._display

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

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

    def geometry_box(self):
        #self.display = self.main.canvas._display
        #self.display = self.main.canvas._display.ifcopenshell.geom.utils.initialize_display()

        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)

        self.display.FitAll()

#Main Class of the Graphical User Interface
class Main(QtGui.QMainWindow):
    def __init__(self, parent=None):
        self.parent=parent
        print "my parent is",self.parent
        #Instantiating the tabs

        global filename
        self.filename = None

        self.tabs = QtGui.QTabWidget()

        self.tab_OBS = QtGui.QWidget()
        self.tab_to_ifc = QtGui.QWidget()
        self.tab_viewer = QtGui.QWidget()

        self.tabs.addTab(self.tab_OBS, 'IFC to FMECA (Object Breakdown Structure)')
        self.tabs.addTab(self.tab_to_ifc, 'Write back to IFC')
        self.tabs.addTab(self.tab_viewer, '3D Risk Viewer')

        #Implementing the OCC Viewer
        super(Main, self).__init__()
        self.canvas = ProductViewer(self)
        #self.canvas = ifcopenshell.geom.utils.initialize_display()

        #Calling the tab methods
        #first tab
        self.OBS_tab()

        #second tab
        self.to_ifc_tab()

        #third tab
        self.canvas_tab()

        self.setCentralWidget(self.tabs)

        self.setGeometry(200, 300, 800, 450)
        self.setWindowTitle('Application FMECA risk analysis and IFC')


#----------------------------Tab_1---------------------------------------------------------
    def OBS_tab(self):

        vbox = QtGui.QVBoxLayout()
        hbox = QtGui.QHBoxLayout()
        h2box = QtGui.QHBoxLayout()
        h3box = QtGui.QHBoxLayout()
        h4box = QtGui.QHBoxLayout()
        h5box = QtGui.QHBoxLayout()

        vbox.addLayout(hbox)
        vbox.addLayout(h2box)
        vbox.addLayout(h3box)
        vbox.addLayout(h4box)
        vbox.addLayout(h5box)

        #Set Layout for Tab Page
        self.tab_OBS.setLayout(vbox)
```

```
        # textual introduction of the tab and functionalities
        OBS_tab_introduction = QtGui.QLabel("""In this tab an infrastructural facility manager can export an Object
        Breakdown Structure (OBS) from the IFC model. This OBS is written to an Failure mode, effects, and
        criticality analysis (FMECA) Excel sheet. This enables the facility manager to asses objects according to
        FMECA specifications.""")
        OBS_tab_introduction.setWordWrap(True)
        OBS_tab_introduction.setMaximumHeight(50)

        #Button open IFC
        OBS_btnOpenIFC_title = QtGui.QLabel("1. Open an IFC model:")
        OBS_btnOpenIFC = QtGui.QPushButton("Open IFC model", self)
        OBS_btnOpenIFC.clicked.connect(self.open_ifc_file_OBS)
        OBS_btnOpenIFC.setStatusTip('Open a .ifc file')

        #ComboBox select object
        ComboBox_title = QtGui.QLabel("2. Select the model type (according to NEN 2767-4):")
        #Setting up a comboBox
        self.comboBox = QtGui.QComboBox(self)

        #Define possible items
        nen_beheerobjecten = ['Aanleginrichtingen', 'Aquaducten', 'Begraafplaatsen', 'Beluchtingstanks',
                              'Bezinkbassins','Bosgebieden','Bruggen (beweegbaar)','Bruggen (vast)','Buffers',
                              'Bunkers','Coupures','Dammen','Dijken','Drinkwaterinlaten','Duikers','Duingebieden',
                              'Ecoducten','Ecoducten','Faunatunnels','Gemalen','Gistingstanks','Grondkeringen',
                              'Havens','Hevels','Individuele Behandeling van Afvalwater (IBA)','Infragebonden
                              gebouwen',
                              'Kabels & Leidingen
                              derden','Kades','Kanalen','Landbouwgebieden','Meren','Natuurgebieden',
                              'Object Overstijgende Voorzieningen','Onderdoorgangen','Ontvangst- en verdeelwerken',
                              'Open
                              Tunnelbakken','Opslagtanks','Overkluizingen','Parken','Persleidingen','Procesbassins',
                              'Recreatieve gebieden','Rioleringen','Rivieren','Schutsluizen','Sifons','Spoorwegen',
                              'Sport- en spelvoorzieningen','Spuisluizen','Stormvloedkeringen','Stuwen','Terreinen',
                              'Tunnels','Uiterwaarden','Verkeerscentrales','Verzorgingsplaatsen','Viaducten',
                              'Wateren en watergangen (lijnvormig), overig','Wateren en watergangen (niet
                              lijnvormig), overig',

                              'Waterreguleringswerken','Wegen','Windmolens','Zandfilterbassins','Zee'

                              ]

        # Add items to comboBox
        for beheerobject in nen_beheerobjecten:
            self.comboBox.addItem(beheerobject)

        # Button export to sheet
        OBS_btnExport_title = QtGui.QLabel("3. Export OBS to sheet:")
        OBS_btnExport = QtGui.QPushButton("Export to FMECA sheet", self)
        OBS_btnExport.clicked.connect(self.write_to_Excel)

        #Button open FMECA sheet with OBS
        OBS_openSheet_title = QtGui.QLabel("4. Open FMECA sheet with OBS:")
        OBS_openSheet = QtGui.QPushButton("Open FMECA", self)
        OBS_openSheet.clicked.connect(self.open_fmeca_OBS)

        #Add to layout
        hbox.addWidget(OBS_tab_introduction)
        h2box.addWidget(OBS_btnOpenIFC_title)
        h2box.addWidget(OBS_btnOpenIFC )
        h3box.addWidget(ComboBox_title)
        h3box.addWidget(self.comboBox)
        h4box.addWidget(OBS_btnExport_title)
        h4box.addWidget(OBS_btnExport)
        h5box.addWidget(OBS_openSheet_title)
        h5box.addWidget(OBS_openSheet)

    def open_ifc_file_OBS(self):
        self.filenameOBS = QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","Industry Foundation Classes
        (*.ifc)")
        self.ifc_file_OBS = ifcopenshell.open(self.filenameOBS)

    def open_fmeca_OBS(self):
        try:
            os.startfile('FMECA_with_OBS.xlsx')
        except:
            print "No FMECA file with OBS found!"

    def get_comboBox_selection(self):
        #Returns the selected value of comboBox
        return str(self.comboBox.currentText())

    def get_otl_model_codes(self):
        #Extract all RWS OTL codes from the Ifc model and add them to a list
        singlevalues = self.ifc_file_OBS.by_type("IfcPropertySingleValue")
        OTL_in_model = []
        for singlevalue in singlevalues:
```

```
            if singlevalue.Name == 'Type Mark':
                OTL_in_model.append(singlevalue.NominalValue.wrappedValue)
        return list(set(OTL_in_model))

    def get_NEN_objects(self, otl_code):
        #Create subtree from individual 'beheerobject'
        for beheerobject in tree.xpath("//Beheerobject"):
            if beheerobject.get('Beheerobject') == self.get_comboBox_selection():
                new_tree = etree.ElementTree(beheerobject)
        #print elements from subtree with corresponding otl code
        elements = new_tree.xpath("//Bouwdeel_Code[@RWS_OTL=$code]", code = otl_code)#[0]
        for element in elements:
            parent = element.getparent()
            parent2 = parent.getparent()
            return self.get_comboBox_selection(), parent2.get('Element'), parent.get('Bouwdeel'),
            element.get('Bouwdeel_Code')

    def write_to_Excel(self):
        try:
            for code in self.get_otl_model_codes():
                template_sheet.append(self.get_NEN_objects(code))
                template_wb.save("FMECA_with_OBS.xlsx")
            print 'Exported to Excel!'
        except:
            print "Not exported, check model!'."


#----------------------------Tab_2----------------------------------------------------------
    def to_ifc_tab(self):
        vbox = QtGui.QVBoxLayout()
        hbox = QtGui.QHBoxLayout()
        h2box = QtGui.QHBoxLayout()
        h3box = QtGui.QHBoxLayout()
        h4box = QtGui.QHBoxLayout()
        h5box = QtGui.QHBoxLayout()

        vbox.addLayout(hbox)
        vbox.addLayout(h2box)
        vbox.addLayout(h3box)
        vbox.addLayout(h4box)
        vbox.addLayout(h5box)

        #Set Layout for Tab Page
        self.tab_to_ifc.setLayout(vbox)

        # textual introduction of the tab and functionalities
        write_tab_introduction = QtGui.QLabel("""In this tab FMECA data can be written to the IFC model. First, the
        facility manager has to complete the FMECA analysis using the file generated in the previous tab. Then
        following the steps below, a FMECA property set is created for each element and added to the model. The
        result is a new IFC model with added FMECA data.""")
        write_tab_introduction.setWordWrap(True)
        write_tab_introduction.setMaximumHeight(50)

        # button and description for opening an IFC file
        openIFC_write_title = QtGui.QLabel("1. Open an IFC model:")
        openIFC_write_btn = QtGui.QPushButton("Open IFC file", self)
        openIFC_write_btn.clicked.connect(self.open_ifc_file_write)

        # button and description for opening an FMECA .xlsx file
        openFMECA_title = QtGui.QLabel("2. Open an FMECA Excel analysis:")
        openFMECA_btn = QtGui.QPushButton("Open FMECA", self)
        openFMECA_btn.clicked.connect(self.open_fmeca_completed)

        # Button write to IFC
        toIFC_btnWrite_title = QtGui.QLabel("3. Write FMECA data to IFC:")
        toIFC_btnWrite = QtGui.QPushButton("Write to IFC", self)
        toIFC_btnWrite.clicked.connect(self.write_to_ifc_per_code)

        #Button open IFC model with FMECA property set
        IFC_openModel_title = QtGui.QLabel("4. Open IFC model with FMECA property set:")
        IFC_openModel_btn = QtGui.QPushButton("Open completed IFC", self)
        IFC_openModel_btn.clicked.connect(self.open_completed_ifc)

        #Add to layout
        hbox.addWidget(write_tab_introduction)
        h2box.addWidget(openIFC_write_title)
        h2box.addWidget(openIFC_write_btn)
        h3box.addWidget(openFMECA_title)
        h3box.addWidget(openFMECA_btn)
        h4box.addWidget(toIFC_btnWrite_title)
        h4box.addWidget(toIFC_btnWrite)
        h5box.addWidget(IFC_openModel_title)
        h5box.addWidget(IFC_openModel_btn)

    def open_fmeca_completed(self):
        filenameFMECA = QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","Excel-worksheet (*.xlsx)")
```

```python
        self.FMECA_completed_wb = openpyxl.load_workbook('%s' %(filenameFMECA), data_only=True)
        self.FMECA_completed_sheet = self.FMECA_completed_wb.get_sheet_by_name('FMECA')

    def open_ifc_file_write(self):
        self.filenameWrite = QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","Industry Foundation Classes
(*.ifc)")
        self.ifc_file_Write = ifcopenshell.open(self.filenameWrite)

    def fmeca_dict(self):
        # Creates a dictionary from the FMECA Excel sheet with NEN code as key and cells as values
        self.nen_to_fmeca_values = defaultdict(list)
        row_count = self.FMECA_completed_sheet.get_highest_row()+1
        column_range = range(5, 18)
        row_range = range (2, row_count)
        for r in row_range:
            for c in column_range:
                cell_value = self.FMECA_completed_sheet.cell(row=r,column=c).value
                code = self.FMECA_completed_sheet.cell(row=r,column=4).value
                self.nen_to_fmeca_values[code].append(cell_value)

    def elm_to_prop_dict(self):
        # Creates a dictionary from the Ifc file with otl code as key and element id's as values
        self.property_id_to_elem_id = defaultdict(list)
        for elem in self.ifc_file_Write.by_type("IfcProduct"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        if prop.Name == 'Type Mark':
                            #print prop.NominalValue.wrappedValue
                            self.property_id_to_elem_id[prop.NominalValue.wrappedValue].append(elem.id())
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByType"):
                    for propsets in rel.RelatingType.HasPropertySets:
                        for prop in propsets.HasProperties:
                            if prop.Name == 'Type Mark':
                                self.property_id_to_elem_id[prop.NominalValue.wrappedValue].append(elem.id())

    def nen_otl_mapping(self, nen_code):
        # Return RWS OTL code for nen2767-4 code input
        element = tree.xpath("//Bouwdeel_Code[@Bouwdeel_Code=$code]", code = nen_code)[0]
        return element.get('RWS_OTL')

    def write_to_ifc(self, nen_code):
        Faalmogelijkheden = self.nen_to_fmeca_values[nen_code][1]
        Effect = self.nen_to_fmeca_values[nen_code][2]
        Kans_initieel = self.nen_to_fmeca_values[nen_code][3]
        Gevolg_initieel = self.nen_to_fmeca_values[nen_code][4]
        Detectie_initieel = self.nen_to_fmeca_values[nen_code][5]
        RPN_initieel = self.nen_to_fmeca_values[nen_code][6]
        Omschrijving_maatregel_preventief = self.nen_to_fmeca_values[nen_code][7]
        Kans_uiteindelijk = self.nen_to_fmeca_values[nen_code][9]
        Gevolg_uiteindelijk = self.nen_to_fmeca_values[nen_code][10]
        Detectie_uiteindelijk = self.nen_to_fmeca_values[nen_code][11]
        RPN_uiteindelijk = self.nen_to_fmeca_values[nen_code][12]

        create_guid = lambda: ifcopenshell.guid.compress(uuid.uuid1().hex)
        owner_history = self.ifc_file_Write.by_type("IfcOwnerHistory")[0]

        # Create and assign property set
        property_values = [
            self.ifc_file_Write.createIfcPropertySingleValue("Faalmogelijkheden", None,
            self.ifc_file_Write.create_entity("IfcText", str(Faalmogelijkheden)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Effect", None,
            self.ifc_file_Write.create_entity("IfcText", str(Effect)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Kans initieel", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(Kans_initieel)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Gevolg initieel", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(Gevolg_initieel)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Detectie initieel", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(Detectie_initieel)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("RPN initieel", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(RPN_initieel)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Omschrijving maatregel preventief", None,
            self.ifc_file_Write.create_entity("IfcText", str(Omschrijving_maatregel_preventief)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Kans uiteindelijk", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(Kans_uiteindelijk)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Gevolg uiteindelijk", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(Gevolg_uiteindelijk)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("Detectie uiteindelijk", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(Detectie_uiteindelijk)), None),
            self.ifc_file_Write.createIfcPropertySingleValue("RPN uiteindelijk", None,
            self.ifc_file_Write.create_entity("IfcInteger", int(RPN_uiteindelijk)), None)
        ]

        # Set up a list of IFC elements with same OTL code property
        pset_elements = []
```

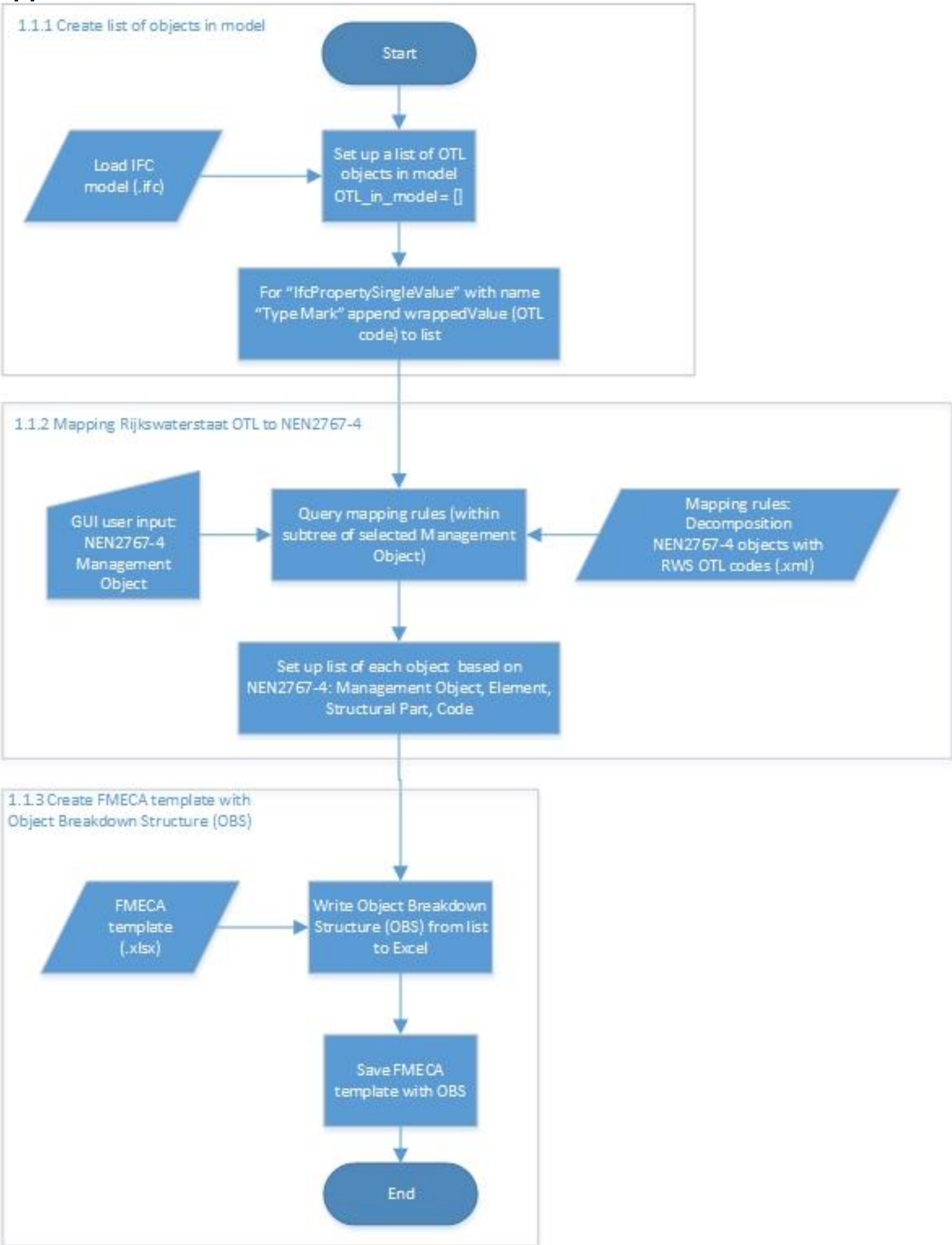
```
        for elm in self.property_id_to_elem_id[self.nen_otl_mapping(nen_code)]:
            pset_elements.append(self.ifc_file_Write.by_id(elm))
        # Create IfcPropertySet element and IfcRelDefinesByProperties element
        property_set = self.ifc_file_Write.createIfcPropertySet(create_guid(), owner_history, "Pset_FMECA_Analysis",
        None, property_values)
        self.ifc_file_Write.createIfcRelDefinesByProperties(create_guid(), owner_history, None, None, pset_elements,
        property_set)
        # Write these elements to a new ifc file with new name:
        self.new_IfcFileName = str(("%s_with_FMECA.ifc") %(os.path.splitext(os.path.basename("%s"
        %self.filenameWrite))[0]))
        self.ifc_file_Write.write(self.new_IfcFileName)

    def write_to_ifc_per_code(self):
        # Set up dictionaries by running defined functions:
        self.fmeca_dict()
        self.elm_to_prop_dict()
        # Make a list of all NEN codes from the FMECA Excel sheet
        nen_code_list = []
        row_count = self.FMECA_completed_sheet.get_highest_row()+1
        row_range = range (2, row_count)
        for r in row_range:
            nen_code_list.append(self.FMECA_completed_sheet.cell(row=r,column=4).value)
        # Run write_to_ifc function for NEN code in NEN code list
        for nen_code in nen_code_list:
            if nen_code == 'None':
                continue
            else:
                self.write_to_ifc(nen_code)

    def open_completed_ifc(self):
        try:
            os.startfile(self.new_IfcFileName)
        except:
            print "No IFC model with FMECA property set found!"


#----------------------------Tab_3----------------------------------------------------------
    def canvas_tab(self):
        #Initialization for a full screen layout
        """canvas_layout = QtGui.QVBoxLayout()
        canvas_layout.addWidget(self.canvas)
        self.tab_viewer.setLayout(canvas_layout)"""

        #Initialization for a split view layout
        self.console = QtGui.QTextBrowser()
        font = QtGui.QFont("Times",10,QtGui.QFont.Bold,True)
        self.console.setFont(font)

        #Define widget for viewer
        center = QtGui.QWidget()

        #Define and set layout
        mainLayout = QtGui.QHBoxLayout(center)
        mainLayout.setContentsMargins(0, 0, 0, 0)

        splitter = QtGui.QSplitter(QtCore.Qt.Horizontal)
        splitter.addWidget(self.canvas)
        splitter.addWidget(self.console)

        splitter.setStretchFactor (0, 10)
        splitter.setStretchFactor (1, 1)

        canvas_btnOpenIFC = QtGui.QPushButton("Open IFC model", self)
        canvas_btnOpenIFC.clicked.connect(self.open_ifc_file)
        canvas_getGuid = QtGui.QPushButton("Refresh model en visualize risks", self)
        canvas_getGuid.clicked.connect(self.parse_ifc_with_risk_color)
        canvas_btnShowProp = QtGui.QPushButton("Show Properties", self)
        canvas_btnShowProp.clicked.connect(self.canvas_getGuid)

        vbox_layout = QtGui.QVBoxLayout()
        hbox_layout = QtGui.QHBoxLayout()
        vbox_layout.addWidget(splitter)
        vbox_layout.addLayout(hbox_layout)
        self.tab_viewer.setLayout(vbox_layout)
        hbox_layout.addWidget(canvas_btnOpenIFC)
        hbox_layout.addWidget(canvas_getGuid)
        hbox_layout.addWidget(canvas_btnShowProp)

    def canvas_getGuid(self):
        #self.console.setText(guid_selection)
        self.console.clear()
        self.guid_to_prop_dict()
        for guid in self.guid_to_prop[guid_selection]:
            if guid.NominalValue.wrappedValue == '':
                continue
            else:
```

```
            self.console.append( "%s : %s" %(guid.Name, guid.NominalValue.wrappedValue))

    def guid_to_prop_dict(self):
        self.guid_to_prop = defaultdict(list)
        for elem in self.ifc_file.by_type("IfcProduct"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        self.guid_to_prop[elem.GlobalId].append(prop)
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByType"):
                    for propsets in rel.RelatingType.HasPropertySets:
                        for prop in propsets.HasProperties:
                            self.guid_to_prop[elem.GlobalId].append(prop)

    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.console.clear()
        self.parse_ifc(self.filename)

    def parse_ifc(self,filename):
        self.ifc_file = ifcopenshell.open(filename)
        products = self.ifc_file.by_type('IfcProduct')

        for product in products:
            if product.is_a('IfcOpeningElement'):continue
            if product.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings, product)

                # Display shape
                self.canvas.Show(product.GlobalId, ifcgeom.geometry)

                # Update display
                self.canvas._display.Repaint()

    def parse_ifc_with_risk_color(self):
        self.parent.display.EraseAll()
        self.console.clear()
        self.ifc_file = ifcopenshell.open(self.filename)
        products = self.ifc_file.by_type('IfcProduct')

        self.elem_rpn_color_dict()
        for product in products:
            if product.is_a('IfcOpeningElement'):continue
            if product.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings, product)

                # Find appropriate color, can be <None>

                color = None
                for element, clr in self.product_rpn.iteritems():
                    if product.id() == element:
                        color = clr
                        break

                # Display shape
                self.canvas.Show(product.GlobalId, ifcgeom.geometry, color)

                # Update display
                self.canvas._display.Repaint()

    def elem_rpn_color_dict(self):
        # Creates a dictionary from the Ifc file with otl code as key and element id's as values
        self.property_id_to_elem_id_rpn = defaultdict(list)
        for elem in self.ifc_file.by_type("IfcProduct"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        if prop.Name == 'RPN uiteindelijk':
                            #print prop.NominalValue.wrappedValue
                            self.property_id_to_elem_id_rpn[prop.NominalValue.wrappedValue].append(elem.id())
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByType"):
                    for propsets in rel.RelatingType.HasPropertySets:
                        for prop in propsets.HasProperties:
                            if prop.Name == 'RPN uiteindelijk':
                                self.property_id_to_elem_id_rpn[prop.NominalValue.wrappedValue].append(elem.id())
        self.product_rpn = dict()
        for rpn, elements in self.property_id_to_elem_id_rpn.iteritems():
            for elem in elements:
                if rpn < 30:
                    self.product_rpn[elem] = "GREEN"
                elif rpn <100:
                    self.product_rpn[elem] = "YELLOW"
```

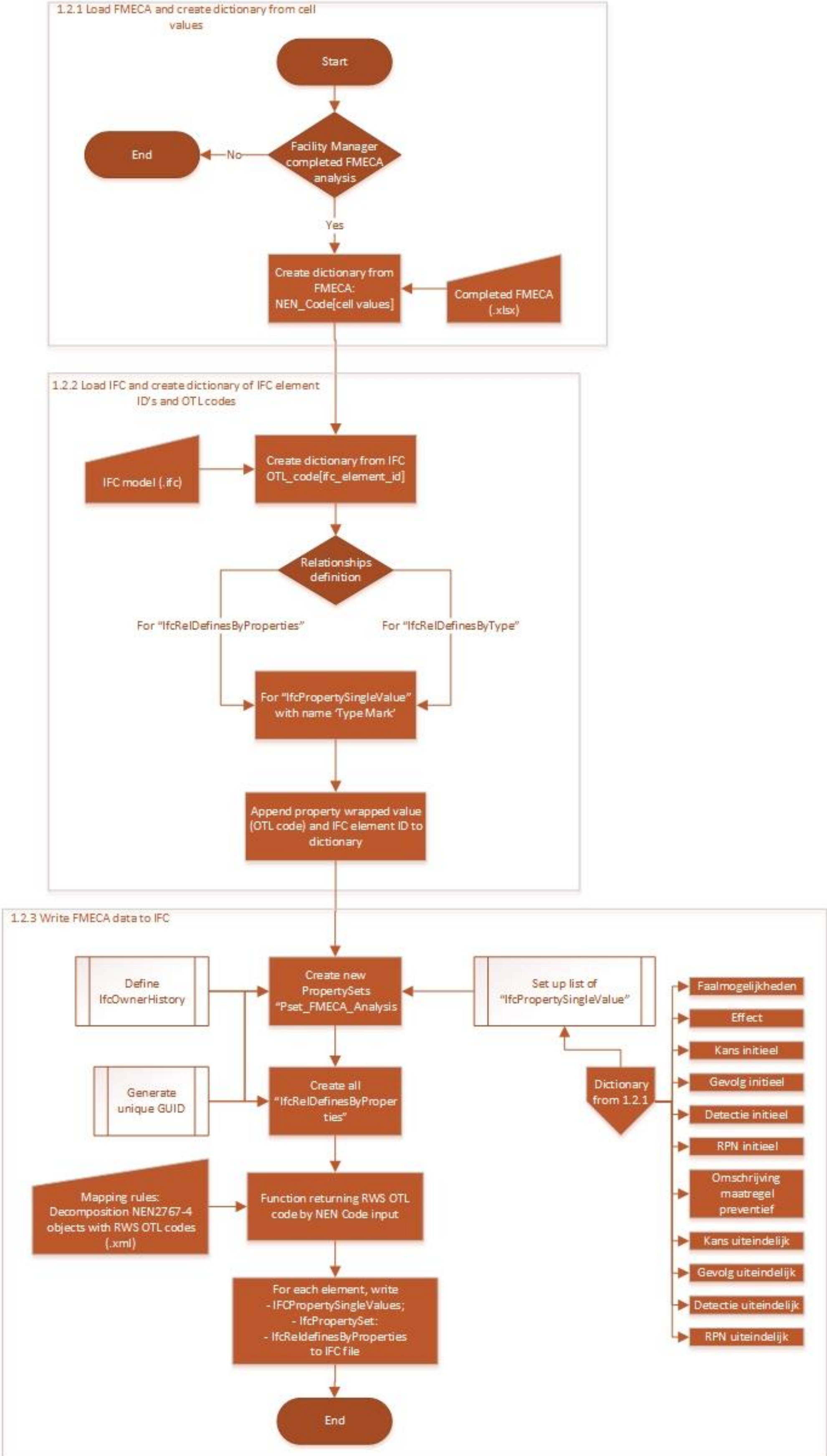
```
            else:
                self.product_rpn[elem] = "RED"

init = initUI()
```

# Appendix 3: Flowcharts tool 1

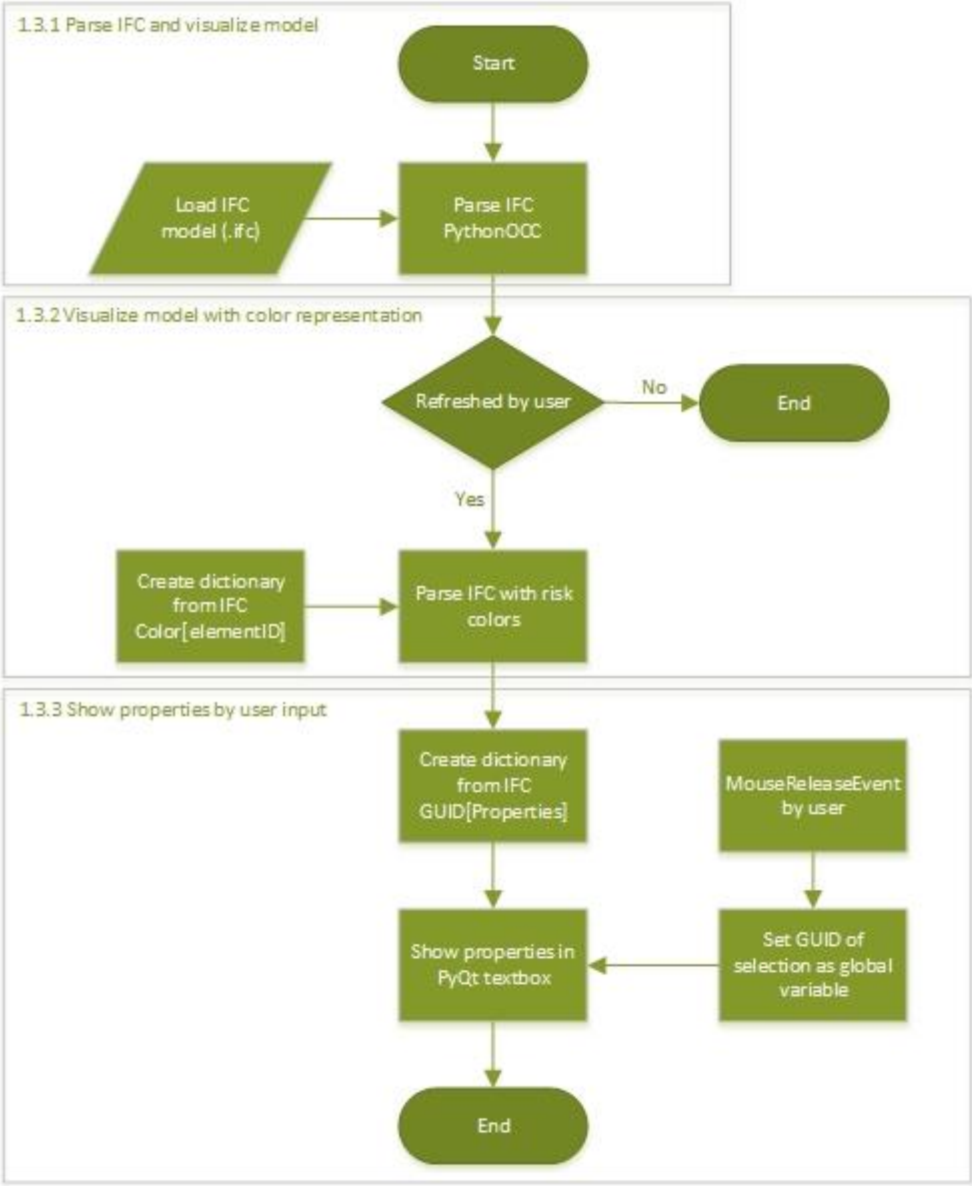
## Appendix 3.1: flowchart tool 1.1



## Appendix 3.2: flowchart tool 1.2

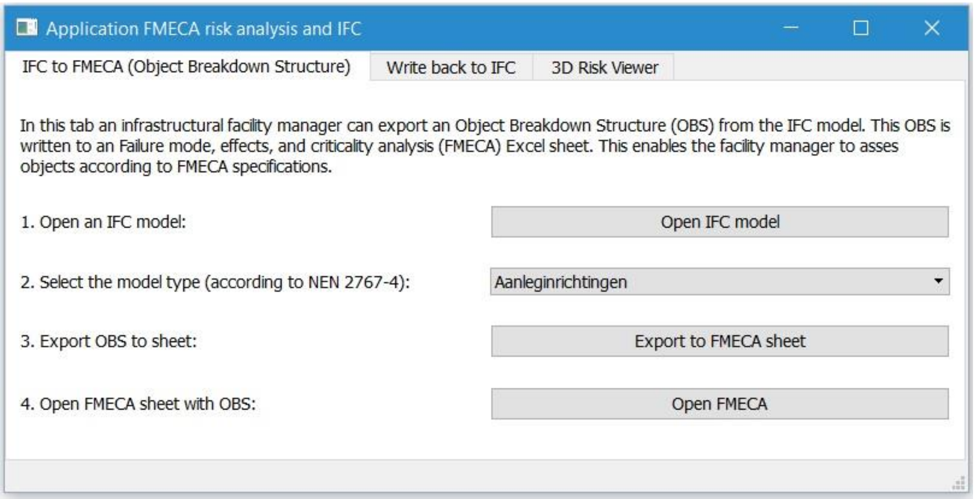


### Appendix 3.3: flowchart tool 1.3

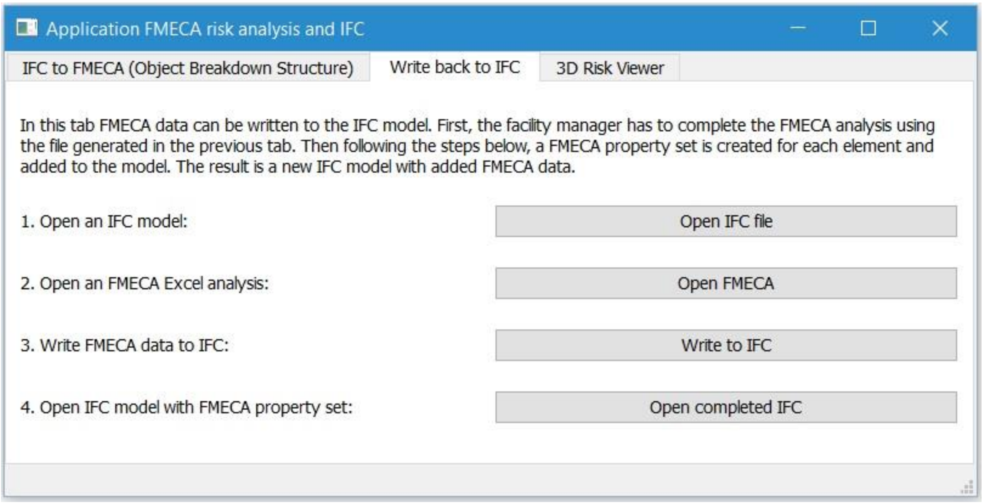


# Appendix 4: Graphical User Interfaces (GUI's)

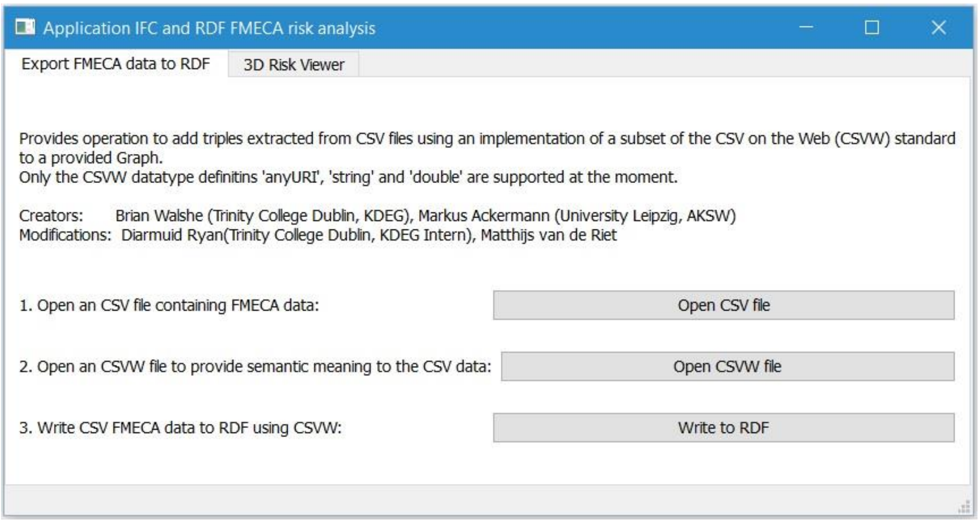
## GUI tool 1.1




## GUI tool 1.2




## GUI tool 2.1



# Appendix 5: Python script tool 2

```
import sys
import os
import uuid
import ifcopenshell
import ifcopenshell.geom
settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)

from collections import defaultdict

from lxml import etree
try: xml_file = open('Decompositie_NEN_OTL.xml')
except: print "RWS OTL to NEN2767-4 mapping rules not found. Add mapping rules to continue."
tree = etree.parse(xml_file)

from PyQt4 import QtGui,QtCore
from PyQt4.QtCore import QObject, pyqtSignal
try: from OCC.Display.qtDisplay import qtViewer3d
except: print "trying old version"
try: from OCC.Display.pyqt4Display import qtViewer3d
except: print "no occ path set"

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

import rdflib
from rdflib import Graph, Literal, URIRef, Namespace, XSD, RDF, RDFS, BNode
import csv
import urllib
import re
from operator import itemgetter

MAX_LINES_TO_PROCESS = -1
CSVW = Namespace("http://www.w3.org/ns/csvw#")
DCAT = Namespace("http://www.w3.org/ns/dcat#")
DC = Namespace("http://purl.org/dc/terms/")

guid_selection = None


# Create 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, *args)
        # Create an empty dictionary to keep track of products
        self.objects = {}
    @staticmethod
    def Hash(shape):
        # Return a (semi-)unique number for a shape
        return shape.HashCode(1<<30)
    def Show(self, key, shape, color=None):
        # Add the shape and number to the dictionary of objects
        self.objects[ProductViewer.Hash(shape)] = key
        if color is None:
            self._display.DisplayShape(shape)
        else:
            self._display.DisplayColoredShape(shape, color)
    def mouseReleaseEvent(self, *args):
        # Process selection by parent class
        qtViewer3d.mouseReleaseEvent(self, *args)
        if self._display.selected_shape:
            # Retrieve the key (GlobalId, but can be anything) of the selected
            # object, by computing the Hash again and comparing with the dictionary
            global guid_selection
            guid_selection = (self.objects[ProductViewer.Hash(self._display.selected_shape)])

#main Class of the Geometry Viewer
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()

        #Status bar
        self.main.statusBar()

        self.display = self.main.canvas._display

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

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

    def geometry_box(self):
        #self.display = self.main.canvas._display
        #self.display = self.main.canvas._display.ifcopenshell.geom.utils.initialize_display()

        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)

        self.display.FitAll()


#Main Class of the Graphical User Interface
class Main(QtGui.QMainWindow):
    def __init__(self, parent=None):
        self.parent=parent
        print "my parent is",self.parent
        #Instantiating the tabs

        global filename
        self.filename = None

        self.tabs = QtGui.QTabWidget()

        self.tab_rdf = QtGui.QWidget()
        self.tab_viewer = QtGui.QWidget()

        self.tabs.addTab(self.tab_rdf, 'Export FMECA data to RDF')
        self.tabs.addTab(self.tab_viewer, '3D Risk Viewer')

        #Implementing the OCC Viewer
        super(Main, self).__init__()
        self.canvas = ProductViewer(self)
        #self.canvas = ifcopenshell.geom.utils.initialize_display()

        #Calling the tab methods
        #first tab
        self.canvas_tab()

        #second tab
        self.rdf_tab()

        self.setCentralWidget(self.tabs)

        self.setGeometry(200, 300, 800, 450)
        self.setWindowTitle('Application IFC and RDF FMECA risk analysis')


#----------------------------Tab_1------------------------------------------------------------
    def rdf_tab(self):

        vbox = QtGui.QVBoxLayout()
        hbox = QtGui.QHBoxLayout()
        h2box = QtGui.QHBoxLayout()
        h3box = QtGui.QHBoxLayout()
        h4box = QtGui.QHBoxLayout()

        vbox.addLayout(hbox)
        vbox.addLayout(h2box)
        vbox.addLayout(h3box)
        vbox.addLayout(h4box)
        self.tab_rdf.setLayout(vbox)

        csv_to_rdf_introduction = QtGui.QLabel(
'''
```

```
Provides operation to add triples extracted from CSV files using an implementation of a subset of the CSV on the Web
(CSVW) standard to a provided Graph.
Only the CSVW datatype definitins 'anyURI', 'string' and 'double' are supported at the moment.

Creators:       Brian Walshe (Trinity College Dublin, KDEG), Markus Ackermann (University Leipzig, AKSW)
Modifications:  Diarmuid Ryan(Trinity College Dublin, KDEG Intern), Matthijs van de Riet
'''
        )

        csv_to_rdf_introduction.setWordWrap(True)
        csv_to_rdf_introduction.setMaximumHeight(130)

        openCSV_btn_title = QtGui.QLabel("1. Open an CSV file containing FMECA data:")
        openCSV_btn = QtGui.QPushButton("Open CSV file", self)
        openCSV_btn.clicked.connect(self.open_csv)

        openCSVW_btn_title = QtGui.QLabel("2. Open an CSVW file to provide semantic meaning to the CSV data:")
        openCSVW_btn = QtGui.QPushButton("Open CSVW file", self)
        openCSVW_btn.clicked.connect(self.open_csvw)

        toRDF_btnWrite_title = QtGui.QLabel("3. Write CSV FMECA data to RDF using CSVW:")
        toRDF_btnWrite = QtGui.QPushButton("Write to RDF", self)
        toRDF_btnWrite.clicked.connect(self.run_csv_to_rdf)

        hbox.addWidget(csv_to_rdf_introduction)
        h2box.addWidget(openCSV_btn_title)
        h2box.addWidget(openCSV_btn)
        h3box.addWidget(openCSVW_btn_title)
        h3box.addWidget(openCSVW_btn)
        h4box.addWidget(toRDF_btnWrite_title)
        h4box.addWidget(toRDF_btnWrite)

    def open_csv(self):
        self.csvFilePath = QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","Comma-separated values (*.csv)")
        self.csvFileName = str(os.path.basename('%s' %self.csvFilePath))

    def open_csvw(self):
        self.csvwFileName = str(QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","CSV on the Web(*.csvw)"))

    def run_csv_to_rdf(self):
        g = Graph()
        CSVWtoRDF = self.CSVWtoRDF
        converter = CSVWtoRDF(g)
        converter.loadCSVW(self.csvFilePath, self.csvFileName, self.csvwFileName)
        converter.writeToFile("%s.rdf"%self.csvFileName)

    class CSVWtoRDF:
        def __init__(self, rdfGraph):
            self.graph = rdfGraph

        def loadCSVW(self, csvFilePath, csvFileName, csvwFileName,  mappingResourceIRI=None):

            """if csvwFilename == None:
                csvwFilename = csvFilename + ".csvw"""

            csvwLD = Graph().parse(csvwFileName, format='json-ld')


            for prefix, expansion in csvwLD.namespaces():
                self.graph.bind(prefix, expansion, override=False)


            if mappingResourceIRI == None:
                mappingResource = self._findMappingResource(csvwLD, csvFileName)
            else:
                mappingResource = URIRef(str(mappingResourceIRI))

            if next(csvwLD.predicate_objects(mappingResource), 'NA') == 'NA':
                raise RuntimeError("mapping description %s does not exist"%mappingResource)

            def anyObject(subject=None, property=None):
                #print "(%s, %s)"%(subject, object)
                return next(csvwLD.objects(subject, property) )

            delim = anyObject(mappingResource, CSVW.delimiter)

            csvFile = open(csvFilePath, "r")
            csvData = csv.reader(csvFile, delimiter=str(delim))

            schemaRes = anyObject(mappingResource, CSVW.tableSchema)

            tableNode = URIRef("%s#table" % csvFileName)
            self.graph.add((tableNode, RDF.type, CSVW.Table))

            dcatNode = BNode()
            self.graph.add((dcatNode, RDF.type, DCAT.Distribution))
```

```
        self.graph.add((dcatNode, DCAT.downloadURL, URIRef(csvFileName)))

        self.graph.add((tableNode, DCAT.distribution, dcatNode))

        mappedColumnsNames = []
        datatypeForColumn = dict()
        propertyForColumn = dict()
        columnList =  csvwLD.value(schemaRes, CSVW.column)
        primaryKey = str(anyObject(schemaRes, CSVW.primaryKey))

        while columnList != None and csvwLD.value(columnList, RDF.first)!=None:
            column = csvwLD.value(columnList, RDF.first)
            columnName = csvwLD.value(column, CSVW.name)
            propertyRes = anyObject(column, CSVW.propertyUrl)
            self.graph.add((propertyRes, RDF.type, RDF.Property))
            self.graph.add((propertyRes, RDFS.label, anyObject(column, CSVW.title)))
            self.graph.add((propertyRes, DC.description, anyObject(column, DC.description)))
            mappedColumnsNames += columnName
            datatypeForColumn[str(columnName)] = str(anyObject(column, CSVW.datatype))
            columnList = csvwLD.value(columnList, RDF.rest)
            propertyForColumn[str(columnName)] = propertyRes

        urlTemplate = anyObject(schemaRes, CSVW.aboutUrl)
        groups = re.match("^(.*?)\{([A-Za-z0-9\-_]+)\}(.*)$", urlTemplate)
        pre = groups.group(1)
        post = groups.group(3)
        nameCol = groups.group(2)

        """csvHeader = []
        for h in csvData.next():
            if h == primaryKey:
                continue
            else:
                csvHeader.append(h)
        print csvHeader"""

        csvHeader = csvData.next()
        cellname2Index = self._cellToIndexMapping(csvHeader)
        idIndex=-1
        if nameCol!="_code":
            idIndex = csvHeader.index(nameCol)

        linesRead = 0
        for line in csvData:
            linesRead += 1
            id_fragment=str(linesRead)
            if idIndex!=-1:
                id_fragment = urllib.quote_plus(line[idIndex])
            #subject = URIRef("%s%s%s" % (pre, id_fragment, post))

            #self.graph.add((tableNode, CSVW.row, subject))
            for cellname, i in cellname2Index.items():
                subject = URIRef("%s%s%s" % (pre, line[cellname2Index[primaryKey]], post))
                if cellname == primaryKey:
                    continue
                else:
                    predicate = propertyForColumn[cellname]
                    obj = self.makeObject(datatypeForColumn[cellname], line[i])
                self.graph.add((subject, predicate, obj))
            if(MAX_LINES_TO_PROCESS > 0 and linesRead >= MAX_LINES_TO_PROCESS): break


    def _findMappingResource(self, csvwLD, csvFileName):
            def mappingResFilter(subject):
                return isinstance(subject, URIRef) and str(subject).endswith(csvFileName)

            mappingCandidates = set(filter(mappingResFilter, map(itemgetter(0), csvwLD)))

            if not mappingCandidates:
                raise RuntimeError("unable to find mapping description")
            else:
                if len(mappingCandidates) > 1:
                    raise RuntimeError("more than one mapping descrption candidate")

            return next(iter(mappingCandidates))

    def _cellToIndexMapping(self, csvHeader):
        return dict(zip(csvHeader, range(0, len(csvHeader))))

    def _sanitizeSID(self, sid):
        return sid.replace('_', '-')

    def makeObject(self, datatypeStr, val):
        if datatypeStr == "anyURI":
            return URIRef(val)
        else:
```

```
                return Literal(val, datatype=datatypeStr)

        def printN3(self):
            print(self.graph.serialize(format='n3'))

        def writeToFile(self, fileName, format="xml"):
            self.graph.serialize(open(fileName, "w"), format)


#----------------------------Tab_2---------------------------------------------------------
    def canvas_tab(self):
        #Initialization for a split view layout
        self.console = QtGui.QTextBrowser()
        font = QtGui.QFont("Times",10,QtGui.QFont.Bold,True)
        self.console.setFont(font)
        self.console2 = QtGui.QTextBrowser()
        self.console2.setFont(font)

        #Define widget for viewer
        center = QtGui.QWidget()

        #Define and set layout
        mainLayout = QtGui.QHBoxLayout(center)
        mainLayout.setContentsMargins(0, 0, 0, 0)

        splitter = QtGui.QSplitter(QtCore.Qt.Horizontal)
        splitterH = QtGui.QSplitter(QtCore.Qt.Vertical)
        splitter.addWidget(self.canvas)
        splitter.addWidget(splitterH)
        splitterH.addWidget(self.console)
        splitterH.addWidget(self.console2)

        splitter.setStretchFactor (0, 10)
        splitter.setStretchFactor (1, 1)

        canvas_btnOpenIFC = QtGui.QPushButton("Open IFC model", self)
        canvas_btnOpenIFC.clicked.connect(self.open_ifc_file)
        canvas_btnOpenRDF = QtGui.QPushButton("Load RDF data", self)
        canvas_btnOpenRDF.clicked.connect(self.load_rdf_data)
        canvas_getGuid = QtGui.QPushButton("Refresh model en visualize risks", self)
        canvas_getGuid.clicked.connect(self.parse_ifc_with_risk_color)
        canvas_btnShowProp = QtGui.QPushButton("Show Properties", self)
        canvas_btnShowProp.clicked.connect(self.canvas_getGuid)

        vbox_layout = QtGui.QVBoxLayout()
        hbox_layout = QtGui.QHBoxLayout()
        vbox_layout.addWidget(splitter)
        vbox_layout.addLayout(hbox_layout)
        self.tab_viewer.setLayout(vbox_layout)
        hbox_layout.addWidget(canvas_btnOpenIFC)
        hbox_layout.addWidget(canvas_btnOpenRDF)
        hbox_layout.addWidget(canvas_getGuid)
        hbox_layout.addWidget(canvas_btnShowProp)

    def canvas_getGuid(self):
        #self.console.setText(guid_selection)
        self.console.clear()
        self.guid_to_prop_dict()
        for guid in self.guid_to_prop[guid_selection]:
            if guid.NominalValue.wrappedValue == '':
                continue
            else:
                self.console.append( "%s : %s" %(guid.Name, guid.NominalValue.wrappedValue))
        self.console2.clear()
        self.rdf_dict_prop()
        self.dict_guid_otl()
        # Append properties to textbox for GUID input
        for elm in self.rdf_subject_to_object[self.otl_nen_mapping(self.otl_to_guid[guid_selection][0])]:
            for prop in elm:
                self.console2.append("%s : %s" %(prop, elm[prop]))

    def guid_to_prop_dict(self):
        self.guid_to_prop = defaultdict(list)
        for elem in self.ifc_file.by_type("IfcProduct"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        self.guid_to_prop[elem.GlobalId].append(prop)
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByType"):
                    for propsets in rel.RelatingType.HasPropertySets:
                        for prop in propsets.HasProperties:
                            self.guid_to_prop[elem.GlobalId].append(prop)

    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.console.clear()
        self.parse_ifc(self.filename)

    def load_rdf_data(self, filename=None):
        self.RDFfilename = QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","Resource Description Framework
        (*.rdf)")
        self.rdf_graph = rdflib.Graph()
        self.rdf_graph.parse(str(self.RDFfilename))

    def parse_ifc(self,filename):
        self.ifc_file = ifcopenshell.open(filename)
        products = self.ifc_file.by_type('IfcProduct')

        for product in products:
            if product.is_a('IfcOpeningElement'):continue
            if product.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings, product)

                # Display shape
                self.canvas.Show(product.GlobalId, ifcgeom.geometry)

                # Update display
                self.canvas._display.Repaint()

    def parse_ifc_with_risk_color(self):
        self.parent.display.EraseAll()
        self.console.clear()
        self.ifc_file = ifcopenshell.open(self.filename)
        products = self.ifc_file.by_type('IfcProduct')

        self.dict_elm_id_fmeca_color()
        for product in products:
            if product.is_a('IfcOpeningElement'):continue
            if product.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings, product)

                # Find appropriate color, can be <None>
                color = None
                for element, clr in self.product_rpn.iteritems():
                    if product.id() == element:
                        color = clr
                        break

                # Display shape
                self.canvas.Show(product.GlobalId, ifcgeom.geometry, color)

                # Update display
                self.canvas._display.Repaint()

    def dict_rdf_nen_rpn(self):
        # Create dictionary with from rdf file with NEN code and associated RPN value
        self.rdf_subject_to_rpn = {}
        for subj, pred, obj in self.rdf_graph:
            p = subj.split('/')[-1]
            if 'NEN_Code' in p:
                subject = p.split('.')[-1]
                predicate = pred.split('/')[-1]
                if predicate == 'rpn_uiteindelijk':
                    self.rdf_subject_to_rpn[subject] = int(float(obj))

    def dict_nen_elm_id(self):
        #Create dictionary with NEN code and associated IFC element id's
        self.nen_to_elem_id= defaultdict(list)
        for elem in self.ifc_file.by_type("IfcProduct"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        if prop.Name == 'Type Mark':
                            #print prop.NominalValue.wrappedValue
                            nen_code = self.otl_nen_mapping(prop.NominalValue.wrappedValue)
                            self.nen_to_elem_id[nen_code].append(elem.id())
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByType"):
                    for propsets in rel.RelatingType.HasPropertySets:
                        for prop in propsets.HasProperties:
                            if prop.Name == 'Type Mark':
                                nen_code = self.otl_nen_mapping(prop.NominalValue.wrappedValue)
                                self.nen_to_elem_id[nen_code].append(elem.id())

    def dict_elm_id_fmeca_color(self):
        self.dict_rdf_nen_rpn()
        self.dict_nen_elm_id()
        #Create dictionary with RPN value and associated IFC element id's
        self.property_id_to_elem_id_rpn = defaultdict(list)
```

```
        for nen, rpn in self.rdf_subject_to_rpn.iteritems():
            for id in self.nen_to_elem_id[nen]:
                self.property_id_to_elem_id_rpn[rpn].append(id)
        #Create dictionary with IFC element id and associated FMECA color
        self.product_rpn = dict()
        for rpn, elements in self.property_id_to_elem_id_rpn.iteritems():
            for elem in elements:
                if rpn < 30:
                    self.product_rpn[elem] = "GREEN"
                elif rpn <100:
                    self.product_rpn[elem] = "YELLOW"
                else:
                    self.product_rpn[elem] = "RED"

    # Function which returns NEN code for OTL code input using xml mapping rules
    def otl_nen_mapping(self, otl_code):
        # Return nen2767-4 code for RWS OTL code input
        element = tree.xpath("//Bouwdeel_Code[@RWS_OTL=$code]", code = otl_code)[0]
        return element.get('Bouwdeel_Code')

    def rdf_dict_prop(self):
        # Create dictionary from RDF with properties
        self.rdf_subject_to_object = defaultdict(list)
        for subj, pred, obj in self.rdf_graph:
            p = subj.split('/')[-1]
            if 'NEN_Code' in p:
                subject = p.split('.')[-1]
                predicate = pred.split('/')[-1]
                d =  {}
                self.rdf_subject_to_object[subject].append({str(predicate) : str(obj)})

    def dict_guid_otl(self):
        # Create dictionary from IFC with GUID and associated OTL code
        self.otl_to_guid = defaultdict(list)
        for elem in self.ifc_file.by_type("IfcProduct"):
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        if prop.Name == 'Type Mark':
                            self.otl_to_guid[elem.GlobalId].append(prop.NominalValue.wrappedValue)
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByType"):
                    for propsets in rel.RelatingType.HasPropertySets:
                        for prop in propsets.HasProperties:
                            if prop.Name == 'Type Mark':
                                self.otl_to_guid[elem.GlobalId].append(prop.NominalValue.wrappedValue)

    def prop_viewer(self, guid):
        self.rdf_dict_prop()
        self.dict_guid_otl()
        # Append properties to textbox for GUID input
        for elm in self.rdf_subject_to_object[self.otl_nen_mapping(self.otl_to_guid[guid][0])]:
            for prop in elm:
                print ("%s : %s" %(prop, elm[prop]))

init = initUI()
```

# Appendix 6: CSVW JSON


```
{
    "@id": "https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv",
    "@context": ["http://www.w3.org/ns/csvw",
    {
        "@language": "nl",
        "dcterms": "http://purl.org/dc/terms/",
        "xsd": "http://www.w3.org/2001/XMLSchema#",
        "fmeca": "https://www.isbe.tue.nl/ontology/fmeca/",
        "cbnl": "http://api.cbnl.org/sparql/CBNL/statements/",
        "ifc": "http://www.buildingsmart.org/ontology/IFC/",
        "uom": "www.w3.org/2007/ont/unit/",
        "ssn": "http://purl.oclc.org/NET/ssnx/ssn/"
    }],
    "delimiter": ";",
    "@type": ["Table",
    "dcat:DataSet"],
    "url": "https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv",
    "dcterms:title": "FMECA Data",
    "dcterms:description": "Table of FMECA analysis",
    "dcterms:keywords": ["risk",
    "operation and maintenance"],
    "dcterms:modified": "2015-11-17",
    "tableSchema": {
        "columns": [{
            "name": "Beheerobject",
            "title": "Beheerobject",
            "dcterms:description": "Naam van Beheerobject",
            "datatype": "xsd:String",
            "propertyUrl": "cbnl:beheerobject",
            "required": true
        },
        {
            "name": "Element",
            "title": "Element",
            "dcterms:description": "Naam van Element",
            "datatype": "xsd:String",
            "propertyUrl": "cbnl:element",
            "required": true
        },
        {
            "name": "Bouwdeel",
            "title": "Bouwdeel",
            "dcterms:description": "Naam van Bouwdeel",
            "datatype": "xsd:String",
            "propertyUrl": "cbnl:bouwdeel",
            "required": true
        },
        {
            "name": "NEN Code",
            "title": "NEN2767 Code Bouwdeel",
            "dcterms:description": "Code van Bouwdeel",
            "datatype": "xsd:String",
            "propertyUrl": "cbnl:code",
            "required": true
        },
        {
```

```
        "name": "Functionele eis",
        "title": "Functionele eis",
        "dcterms:description": "Functionele eis",
        "datatype": "xsd:String",
        "propertyUrl": "fmeca:functionele_eis",
        "required": true
    },
    {
        "name": "Faalmogelijkheden",
        "title": "Faalmogelijkheden",
        "dcterms:description": "Mogelijke Faaleigenschappen",
        "datatype": "xsd:String",
        "propertyUrl": "fmeca:faalmogelijkheden",
        "required": true
    },
    {
        "name": "Effect",
        "title": "Faal Effect",
        "dcterms:description": "Effect van Falen",
        "datatype": "xsd:String",
        "propertyUrl": "fmeca:effect",
        "required": true
    },
    {
        "name": "Kans (initieel)",
        "title": "Kans (initieel)",
        "dcterms:description": "Kans op Falen Zonder Maatregel",
        "datatype": "xsd:double",
        "propertyUrl": "fmeca:kans_initieel",
        "required": true
    },
    {
        "name": "Gevolg (initieel)",
        "title": "Gevolg (initieel)",
        "dcterms:description": "Gevolg van Falen Zonder Maatregel",
        "datatype": "xsd:double",
        "propertyUrl": "fmeca:gevolg_initieel",
        "required": true
    },
    {
        "name": "Detectie (initieel)",
        "title": "Detectie (initieel)",
        "dcterms:description": "Moeilijkheid van Detectie bij Falen Zonder Maatregel",
        "datatype": "xsd:double",
        "propertyUrl": "fmeca:detectie_initieel",
        "required": true
    },
    {
        "name": "RPN (initieel)",
        "title": "Risk Priority Number (RPN)",
        "dcterms:description": "RPN Zonder Maatregel",
        "datatype": "xsd:double",
        "propertyUrl": "fmeca:rpn_initieel",
        "required": true
    },
    {
        "name": "Omschrijving maatregel (preventief)",
        "title": "Omschrijving maatregel (preventief)",
        "dcterms:description": "Preventieve Maatregel",
        "datatype": "xsd:String",
        "propertyUrl": "fmeca:omschrijving_maatregel_preventief",
        "required": true
    },
    {
        "name": "Frequentie maatregel (preventief)",
        "title": "Frequentie maatregel (preventief)",
        "dcterms:description": "Frequetie van Uitvoeren Preventieve Maatregel",
        "datatype": "xsd:String",
        "propertyUrl": "fmeca:frequentie_maatregel_preventief",
        "required": true
    },
```

```
        {
            "name": "Kans (uiteindelijk)",
            "title": "Kans (uiteindelijk)",
            "dcterms:description": "Kans op Falen na Maatrel",
            "datatype": "xsd:double",
            "propertyUrl": "fmeca:kans_uiteindelijk",
            "required": true
        },
        {
            "name": "Gevolg (uiteindelijk)",
            "title": "Gevolg (uiteindelijk)",
            "dcterms:description": "Gevolg van Falen na Maatregel",
            "datatype": "xsd:double",
            "propertyUrl": "fmeca:gevolg_uiteindelijk",
            "required": true
        },
        {
            "name": "Detectie (uiteindelijk)",
            "title": "Detectie (uiteindelijk)",
            "dcterms:description": "Moeilijkheid van Detectie na Maatregel",
            "datatype": "xsd:double",
            "propertyUrl": "fmeca:detectie_uiteindelijk",
            "required": true
        },
        {
            "name": "RPN (uiteindelijk)",
            "title": "Risk Priority Number (RPN)",
            "dcterms:description": "RPN na Maatregel",
            "datatype": "xsd:double",
            "propertyUrl": "fmeca:rpn_uiteindelijk",
            "required": true
        }],
        "primaryKey": "NEN Code",
        "aboutUrl":
        "https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.{_code}"
    }
}
```

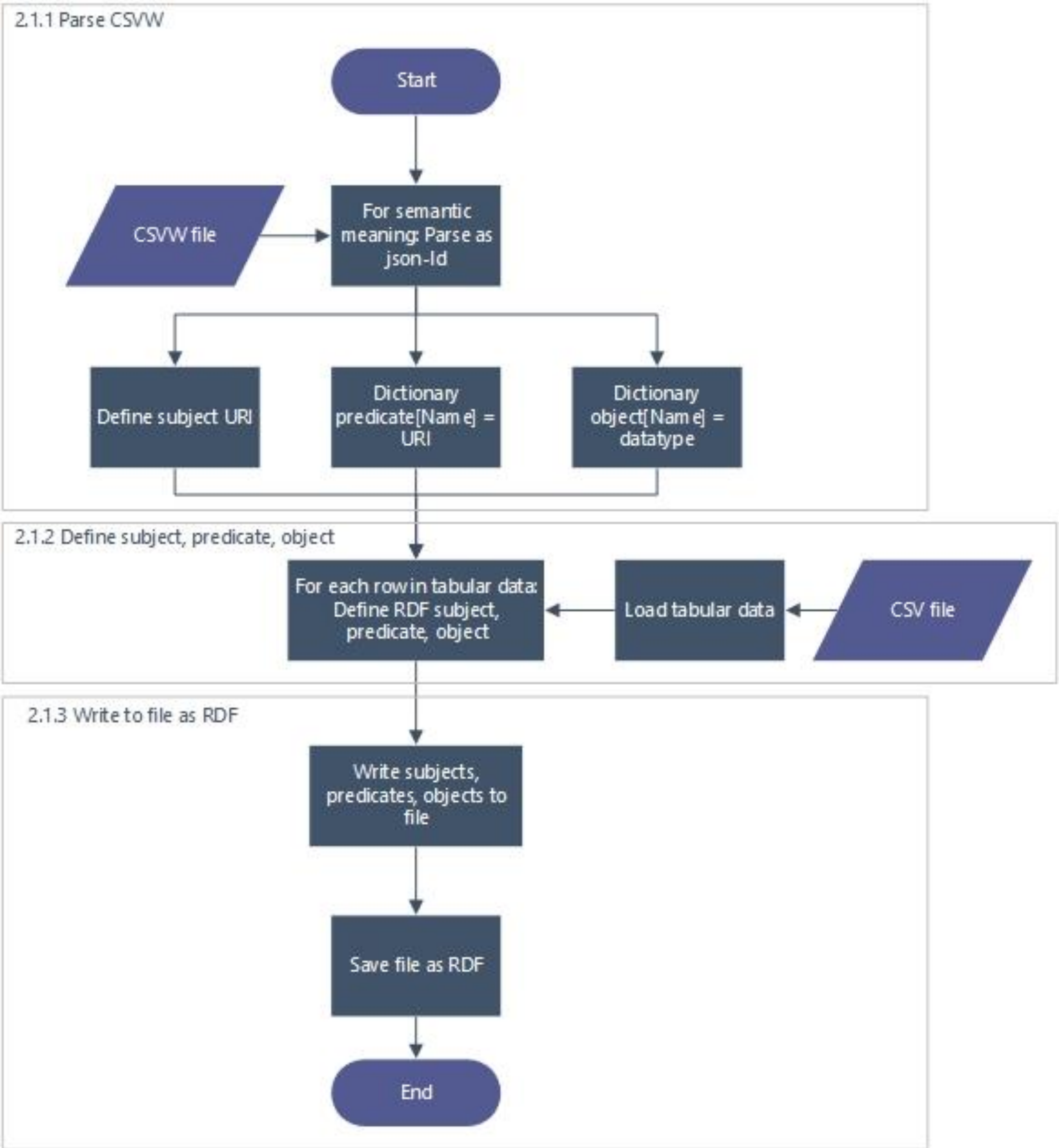
# Appendix 7: FMECA ontology (.ttl)

```
# baseURI: https://www.isbe.tue.nl/ontology/fmeca

@prefix fmeca: <https://www.isbe.tue.nl/ontology/fmeca#> .
@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#> .

owl:detectie_initieel
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Detectability initial"@en-gb ;
  rdfs:label "Detectie initieel"@nl-nl ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 10 ;
  xsd:minInclusive 1 ;
.
owl:detectie_uiteindelijk
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Detectability final"@en-gb ;
  rdfs:label "Detectie uiteindelijk"@nl-nl ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 10 ;
  xsd:minInclusive 1 ;
.
owl:effect
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Effect"@en-gb ;
  rdfs:label "Effect"@nl-nl ;
  rdfs:range xsd:string ;
.
owl:faalmogelijkheden
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Faalmogelijkheden"@nl-nl ;
  rdfs:label "Failure modes"@en-gb ;
  rdfs:range xsd:string ;
.
owl:frequentie_maatregel_preventief
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Frequency preventive measure"@en-gb ;
  rdfs:label "Frequentie preventieve maatregel"@nl-nl ;
  rdfs:range xsd:string ;
.
owl:functionele_eis
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Functional requirement"@en-gb ;
  rdfs:label "Functionele eis"@nl-nl ;
  rdfs:range xsd:string ;
.
owl:gevolg_initieel
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Gevolg initieel"@nl-nl ;
  rdfs:label "Severity initial"@en-gb ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 10 ;
```

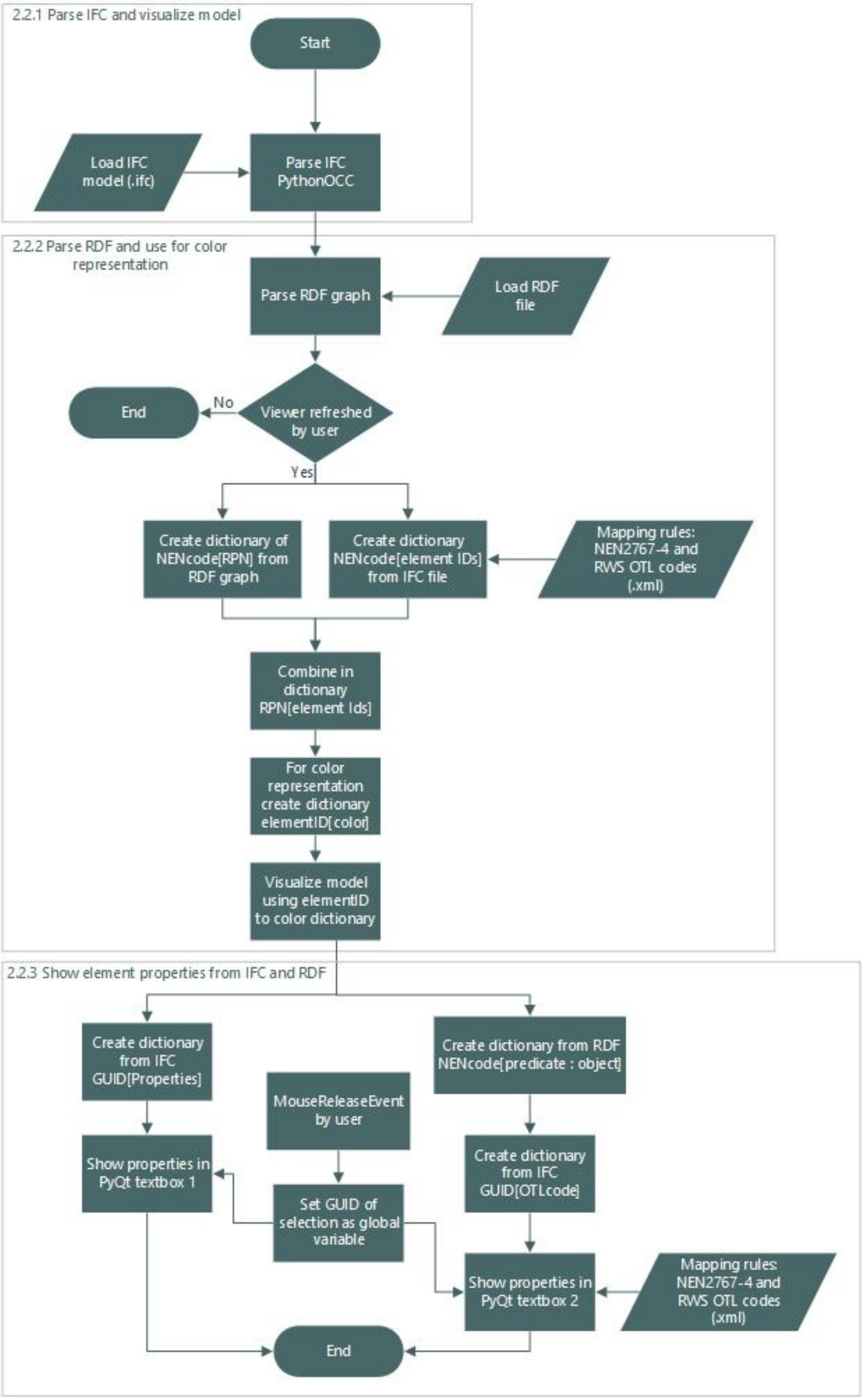
```
  xsd:minInclusive 1 ;
.
owl:gevolg_uiteindelijk
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Gevolg uiteindelijk"@nl-nl ;
  rdfs:label "Severity final"@en-gb ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 10 ;
  xsd:minInclusive 1 ;
.
owl:kans_initieel
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Kans initial"@en-gb ;
  rdfs:label "Occurrence initieel"@nl-nl ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 10 ;
  xsd:minInclusive 1 ;
.
owl:kans_uiteindelijk
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Kans uiteindelijk"@nl-nl ;
  rdfs:label "Probability final"@en-gb ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 10 ;
  xsd:minInclusive 1 ;
.
owl:omschrijving_maatregel_preventief
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "Description preventive measure"@en-gb ;
  rdfs:label "Omschrijving preventieve maatregel"@nl-nl ;
  rdfs:range xsd:string ;
.
owl:rpn_initieel
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "RPN initial"@en-gb ;
  rdfs:label "RPN initieel"@nl-nl ;
  rdfs:range xsd:integer ;
  xsd:minInclusive 1 ;
  xsd:minInclusive 1000 ;
.
owl:rpn_uiteindelijk
  rdf:type owl:DatatypeProperty ;
  rdfs:domain fmeca:nen_code ;
  rdfs:label "RPN final"@en-gb ;
  rdfs:label "RPN uiteindelijk"@nl-nl ;
  rdfs:range xsd:integer ;
  xsd:maxInclusive 1000 ;
  xsd:minInclusive 1 ;
.
<https://www.isbe.tue.nl/ontology/fmeca>
  rdf:type owl:Ontology ;
  owl:versionInfo "Created with TopBraid Composer"^^xsd:string ;
.
fmeca:bouwdeel
  rdf:type owl:Class ;
  rdfs:label "Bouwdeel"@nl-nl ;
  rdfs:label "Building component"@en-gb ;
  rdfs:subClassOf owl:Thing ;
.
fmeca:nen_code
  rdf:type owl:Class ;
  rdfs:label "NEN code"@en-gb ;
  rdfs:label "NEN code"@nl-nl ;
  rdfs:subClassOf fmeca:bouwdeel ;
.
```

# Appendix 8: Flowcharts tool 2

## Appendix 8.1: flowchart tool 2.1



## Appendix 8.2: flowchart tool 2.2



# Appendix 9: Partial RDF


```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:cbnl="http://api.cbnl.org/sparql/CBNL/statements/"
   xmlns:dcat="http://www.w3.org/ns/dcat#"
   xmlns:dcterms="http://purl.org/dc/terms/"
   xmlns:fmeca="https://www.isbe.tue.nl/ontology/fmeca/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/ontology/fmeca/detectie_uiteindelijk">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <dcterms:description xml:lang="nl">Moeilijkheid van Detectie na Maatregel</dcterms:description>
    <rdfs:label xml:lang="nl">Detectie (uiteindelijk)</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1557">
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:kans_uiteindelijk>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Scheefstand</fmeca:effect>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Trekpalen</cbnl:bouwdeel>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:kans_initieel>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:gevolg_initieel>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vervorming/ breuk</fmeca:faalmogelijkheden>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Fundering</cbnl:element>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">84.0</fmeca:rpn_uiteindelijk>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:detectie_uiteindelijk>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">126.0</fmeca:rpn_initieel>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:gevolg_uiteindelijk>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:detectie_initieel>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Trekspanningen van constructie op te nemen</fmeca:functionele_eis>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1461">
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:gevolg_initieel>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Instorting</fmeca:effect>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:detectie_uiteindelijk>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Rijdek</cbnl:bouwdeel>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:gevolg_uiteindelijk>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">mnd</fmeca:frequentie_maatregel_preventief>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Hoofddraagconstructie</cbnl:element>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:detectie_initieel>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Controleren
    afdichtingsconstructie</fmeca:omschrijving_maatregel_preventief>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Stabiele en conforabele ondergond voor passerend verkeer</fmeca:functionele_eis>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">45.0</fmeca:rpn_initieel>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Wapeningscorrosie door chloriden</fmeca:faalmogelijkheden>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">36.0</fmeca:rpn_uiteindelijk>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">5.0</fmeca:kans_initieel>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4.0</fmeca:kans_uiteindelijk>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
  </rdf:Description>
  <rdf:Description rdf:about="http://api.cbnl.org/sparql/CBNL/statements/code">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <dcterms:description xml:lang="nl">Code van Bouwdeel</dcterms:description>
    <rdfs:label xml:lang="nl">NEN2767 Code Bouwdeel</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/ontology/fmeca/rpn_uiteindelijk">
    <dcterms:description xml:lang="nl">RPN na Maatregel</dcterms:description>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label xml:lang="nl">Risk Priority Number (RPN)</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1039">
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:gevolg_initieel>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">8.0</fmeca:rpn_uiteindelijk>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Dien beschadigingen door menselijk toedoen te voorkomen</fmeca:functionele_eis>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4.0</fmeca:detectie_uiteindelijk>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Antivandalismevoorziening Algemeen</cbnl:bouwdeel>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vermindering estetische waarde</fmeca:effect>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:kans_initieel>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vervorming/ breuk</fmeca:faalmogelijkheden>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Antivandalismevoorziening</cbnl:element>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:kans_uiteindelijk>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">12.0</fmeca:rpn_initieel>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:gevolg_uiteindelijk>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4.0</fmeca:detectie_initieel>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1687">
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overbrengen belastingen van overspanning naar constructie</fmeca:functionele_eis>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Instorting</fmeca:effect>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:gevolg_uiteindelijk>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Oplegbalk</cbnl:bouwdeel>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Oplegging</cbnl:element>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:gevolg_initieel>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:kans_initieel>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:detectie_uiteindelijk>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">mnd</fmeca:frequentie_maatregel_preventief>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:detectie_initieel>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Controleren
    afdichtingsconstructie</fmeca:omschrijving_maatregel_preventief>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">63.0</fmeca:rpn_initieel>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Wapeningscorrosie door chloriden</fmeca:faalmogelijkheden>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">63.0</fmeca:rpn_uiteindelijk>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:kans_uiteindelijk>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1013">
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:detectie_uiteindelijk>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">162.0</fmeca:rpn_uiteindelijk>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:kans_initieel>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:detectie_initieel>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Verminderde afvoer</fmeca:effect>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Hemelwaterafvoer (HWA)</cbnl:element>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Controleren waterdoorvoer</fmeca:omschrijving_maatregel_preventief>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">162.0</fmeca:rpn_initieel>
```

```
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Aansluitleiding</cbnl:bouwdeel>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Scheurvorming</fmeca:faalmogelijkheden>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:kans_uiteindelijk>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">mnd</fmeca:frequentie_maatregel_preventief>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Afvoer van hemelwater op en rondom beheerobject</fmeca:functionele_eis>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:gevolg_initieel>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:gevolg_uiteindelijk>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1234">
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:gevolg_initieel>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Verzakking</fmeca:faalmogelijkheden>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:gevolg_uiteindelijk>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">5.0</fmeca:detectie_uiteindelijk>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">70.0</fmeca:rpn_initieel>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:kans_uiteindelijk>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:kans_initieel>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Fundering</cbnl:element>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Scheefstand</fmeca:effect>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">70.0</fmeca:rpn_uiteindelijk>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Voldoende draagkrachtig voor constructie</fmeca:functionele_eis>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">5.0</fmeca:detectie_initieel>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Grondverbetering</cbnl:bouwdeel>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1686">
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">168.0</fmeca:rpn_uiteindelijk>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vervorming/ breuk</fmeca:faalmogelijkheden>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">8.0</fmeca:gevolg_uiteindelijk>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Oplegging</cbnl:element>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:kans_uiteindelijk>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:kans_initieel>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Oplegblok</cbnl:bouwdeel>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">168.0</fmeca:rpn_initieel>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Instorting</fmeca:effect>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overbrengen belastingen van overspanning naar constructie</fmeca:functionele_eis>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">8.0</fmeca:gevolg_initieel>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:detectie_uiteindelijk>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">7.0</fmeca:detectie_initieel>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1051">
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Instorting</fmeca:effect>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:kans_uiteindelijk>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vervorming/ breuk</fmeca:faalmogelijkheden>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overbrengen belastingen van overspanning naar constructie</fmeca:functionele_eis>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:gevolg_initieel>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:detectie_initieel>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:kans_initieel>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Fundering</cbnl:element>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:rpn_initieel>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:detectie_uiteindelijk>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:rpn_uiteindelijk>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:gevolg_uiteindelijk>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Balk</cbnl:bouwdeel>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1179">
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:detectie_initieel>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Dwarsdrager</cbnl:bouwdeel>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:gevolg_initieel>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Hoofddraagconstructie</cbnl:element>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:gevolg_uiteindelijk>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:kans_initieel>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vervorming/ breuk</fmeca:faalmogelijkheden>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:detectie_uiteindelijk>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Instorting</fmeca:effect>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:rpn_uiteindelijk>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:kans_uiteindelijk>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">9.0</fmeca:rpn_initieel>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Dient belasting van bovenliggende constructie te kunnen dragen</fmeca:functionele_eis>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/ontology/fmeca/gevolg_uiteindelijk">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label xml:lang="nl">Gevolg (uiteindelijk)</rdfs:label>
    <dcterms:description xml:lang="nl">Gevolg van Falen na Maatregel</dcterms:description>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1638">
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Voldoende draagkrachtig voor constructie</fmeca:functionele_eis>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4.0</fmeca:gevolg_initieel>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">4.0</fmeca:gevolg_uiteindelijk>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:detectie_uiteindelijk>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Verzakking</fmeca:faalmogelijkheden>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:kans_initieel>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">1.0</fmeca:kans_uiteindelijk>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">24.0</fmeca:rpn_uiteindelijk>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">24.0</fmeca:rpn_initieel>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Scheefstand</fmeca:effect>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:detectie_initieel>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Onderbouw</cbnl:element>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Zandbed</cbnl:bouwdeel>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1729">
    <cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Geleideconstructie</cbnl:element>
    <cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Trottoirband</cbnl:bouwdeel>
    <fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:detectie_uiteindelijk>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Vervorming/ breuk</fmeca:faalmogelijkheden>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Scheefstand</fmeca:effect>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:detectie_initieel>
    <fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">24.0</fmeca:rpn_uiteindelijk>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:kans_uiteindelijk>
    <fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:gevolg_uiteindelijk>
    <fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">24.0</fmeca:rpn_initieel>
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">6.0</fmeca:gevolg_initieel>
    <cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Overkluizingen</cbnl:beheerobject>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">2.0</fmeca:kans_initieel>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Inspectie visiueel</fmeca:omschrijving_maatregel_preventief>
    <fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Afwerking vloerveld tbw HWA afvoer</fmeca:functionele_eis>
  </rdf:Description>
  <rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1024">
    <fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">3.0</fmeca:gevolg_initieel>
    <fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">jr</fmeca:frequentie_maatregel_preventief>
    <fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">10.0</fmeca:detectie_initieel>
    <fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double">5.0</fmeca:kans_uiteindelijk>
    <fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Waterdoorlating</fmeca:effect>
    <fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Controleren scheurvorming</fmeca:omschrijving_maatregel_preventief>
    <fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String">Scheurvorming</fmeca:faalmogelijkheden>
    <fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double">5.0</fmeca:kans_initieel>
```

```
    &lt;fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Dient te fungeren als waterbarriere&lt;/fmeca:functionele_eis&gt;
    &lt;fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;3.0&lt;/fmeca:gevolg_uiteindelijk&gt;
    &lt;cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Voegovergang&lt;/cbnl:element&gt;
    &lt;fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;150.0&lt;/fmeca:rpn_initieel&gt;
    &lt;cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Afdichtingsprofiel&lt;/cbnl:bouwdeel&gt;
    &lt;fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;10.0&lt;/fmeca:detectie_uiteindelijk&gt;
    &lt;cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Overkluizingen&lt;/cbnl:beheerobject&gt;
    &lt;fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;150.0&lt;/fmeca:rpn_uiteindelijk&gt;
  &lt;/rdf:Description&gt;
  &lt;rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.2038"&gt;
    &lt;fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Vervorming/ breuk&lt;/fmeca:faalmogelijkheden&gt;
    &lt;cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Overkluizingen&lt;/cbnl:beheerobject&gt;
    &lt;fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;3.0&lt;/fmeca:kans_uiteindelijk&gt;
    &lt;fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;90.0&lt;/fmeca:rpn_uiteindelijk&gt;
    &lt;fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Inspectie visiueel&lt;/fmeca:omschrijving_maatregel_preventief&gt;
    &lt;cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Hoofddraagconstructie&lt;/cbnl:element&gt;
    &lt;fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;5.0&lt;/fmeca:gevolg_uiteindelijk&gt;
    &lt;fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;5.0&lt;/fmeca:gevolg_initieel&gt;
    &lt;fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;150.0&lt;/fmeca:rpn_initieel&gt;
    &lt;fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Scheefstand&lt;/fmeca:effect&gt;
    &lt;cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Sleufonderbouw&lt;/cbnl:bouwdeel&gt;
    &lt;fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Overbrengen belasting naar bodem&lt;/fmeca:functionele_eis&gt;
    &lt;fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;5.0&lt;/fmeca:kans_initieel&gt;
    &lt;fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;6.0&lt;/fmeca:detectie_uiteindelijk&gt;
    &lt;fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;jr&lt;/fmeca:frequentie_maatregel_preventief&gt;
    &lt;fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;6.0&lt;/fmeca:detectie_initieel&gt;
  &lt;/rdf:Description&gt;
  &lt;rdf:Description rdf:about="https://www.isbe.tue.nl/csvw/FMECA_Overkluizing_Minstroom.csv/NEN_Code.1153"&gt;
    &lt;fmeca:effect rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Instorting&lt;/fmeca:effect&gt;
    &lt;cbnl:element rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Hoofddraagconstructie&lt;/cbnl:element&gt;
    &lt;fmeca:gevolg_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;9.0&lt;/fmeca:gevolg_uiteindelijk&gt;
    &lt;fmeca:detectie_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;3.0&lt;/fmeca:detectie_uiteindelijk&gt;
    &lt;fmeca:frequentie_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;jr&lt;/fmeca:frequentie_maatregel_preventief&gt;
    &lt;fmeca:kans_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;1.0&lt;/fmeca:kans_initieel&gt;
    &lt;fmeca:omschrijving_maatregel_preventief rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Inspectie visiueel&lt;/fmeca:omschrijving_maatregel_preventief&gt;
    &lt;fmeca:detectie_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;3.0&lt;/fmeca:detectie_initieel&gt;
    &lt;cbnl:beheerobject rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Overkluizingen&lt;/cbnl:beheerobject&gt;
    &lt;fmeca:faalmogelijkheden rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Vervorming/ breuk&lt;/fmeca:faalmogelijkheden&gt;
    &lt;fmeca:functionele_eis rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Moet afschuiving van spoorconstructie voorkomen&lt;/fmeca:functionele_eis&gt;
    &lt;fmeca:kans_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;1.0&lt;/fmeca:kans_uiteindelijk&gt;
    &lt;fmeca:gevolg_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;9.0&lt;/fmeca:gevolg_initieel&gt;
    &lt;fmeca:rpn_uiteindelijk rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;27.0&lt;/fmeca:rpn_uiteindelijk&gt;
    &lt;fmeca:rpn_initieel rdf:datatype="http://www.w3.org/2001/XMLSchema#double"&gt;27.0&lt;/fmeca:rpn_initieel&gt;
    &lt;cbnl:bouwdeel rdf:datatype="http://www.w3.org/2001/XMLSchema#String"&gt;Diagonaal (vakwerkligger)&lt;/cbnl:bouwdeel&gt;
  &lt;/rdf:Description&gt;
&lt;/rdf:RDF&gt;
```