Technische Universiteit **Eindhoven** University of Technology

Mutation management in BIM models during Operations & Maintenance

Managing Operations & Maintenance building models with the use of Industry Foundation Classes and Linked Data

MASTER'S THESIS

Author	: N.A.M. (Niels) van de Ven
Student number	: 0772961
Date	: 23-02-2017
E-mail	: nielsvdven93@gmail.com
Date of final presentation	: 23-03-2017

UNIVERSITY

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

IN COLLABORATION WITH

Company	: Vgib Onderhoudsmanagement B.V.

GRADUATION COMMITTEE

Chairman (TU/e)	: prof.dr.ir. B. (Bauke) de Vries
Graduation supervisor (TU/e)	: dr.Dipl.-Ing. J. (Jakob) Beetz
Graduation supervisor (TU/e)	: T.F. (Thomas) Krijnen
External supervisor (Vgib)	: ir. R.B.G. (Rolf) Hornes

PREFACE

It is with great pleasure that I present this thesis as the result of my graduation project. The thesis represents the end of my master Construction Management & Engineering at Eindhoven University of Technology. The research in this thesis was carried out in collaboration with Vgib Onderhoudsmanagement BV. This collaboration has resulted in an interesting research that utilizes knowledge of the academic world to improve a persistent problem in the current practice in the construction industry.

By conducting research in the field of BIM, operations & maintenance and the Semantic Web, I had chosen a challenging topic due to the fact that I did not yet know that much about the latter and because I was passionate about programming a real and usable application. However, I did not have any real programming experience. Eventually, after watching a lot of YouTube tutorials and reading up, my efforts turned out as I initially hoped. I consider the complete research a great opportunity to develop myself in the field of information management, from which I will definitely profit in my further career.

To execute this research and to gain the knowledge in this field of expertise, I have received help and guidance from a number of people without which this research would not have been possible. First of all, I would like to extend my gratitude towards Jakob Beetz for his guidance in the process. Throughout the entire graduation period you have challenged me to think about further possibilities and opportunities with available techniques. Thank you for the continuous support and help in bringing this research further. Secondly, I want to thank Thomas Krijnen for all the help that you have given me. As I came to you as a layman in the field of programming, I cannot thank you enough for your help in developing the prototype of the application and the discussions we had about the use of the application. And of course, Rolf Hornes and Mathijs van Daal, thank you for the opportunity to conduct this research at Vgib. You have both helped me to improve the research with valuable discussions we had and the right connections for conducting the interviews.

I also want to thank my colleagues at Vgib for all the help and the great time I had the last six months. I am grateful to each of the interviewees for making the time and providing valuable input and insight in the current state of information management and how this can be improved. Last but not least, I would like to thank my girlfriend Angelique, my family and my friends for all the help and support!

I hope that every reader will enjoy reading and learning from this thesis report.

Niels van de Ven
Vinkel, March 2017

SUMMARY

The architecture, engineering and construction (AEC) industry has seen major changes over the last decades. Most buildings in the 80s and 90s were designed in 2D, which causes the need for physical or digital exchange of these 2D drawings. When the building changes after construction is completed, these drawings have to be altered and exchanged again. This process of information management obviously is susceptible for mistakes.

A way to lead this process that is gaining significant momentum in the construction sector nowadays is building information modelling (BIM). BIM encompasses every form of building-related information, and visualises this information in a 3D environment. However, BIM implementation in the construction sector still focuses on design and construction, whereas the number of companies focusing on operations and maintenance (O&M) that have adopted BIM is very limited. Given that the exploitation phase, often lasting 50 years or more, is the longest in a building's life cycle and the costs in this phase may amount to many times the design/construction costs, the potential of utilizing BIM seems significant.

Open standards, such as the Industry Foundation Classes (IFC) have the potential to integrate the different domains of design, construction and O&M. IFC is an open and standardized data model with the intent of enabling interoperability between different BIM software applications. IFC files not only contain the geometry of objects, but attributes in property sets and metadata relating to other objects in the building as well. However, in the exploitation phase building data often changes when representing the as-used situation. The IFC standard still lacks adequate handling of such changes, as well as recording history of objects in a BIM (JalyZada, Tizani, & Oti, 2014). Therefore information is often old and thus obsolete so that the building has to be visited for the sole purpose of regaining this information. Missing, obsolete or unstructured information results in ineffective project management, uncertain process results and time loss or cost increases in maintenance, retrofit or remediation processes (Volk, Stengel, & Schultmann, 2014).

As a result, the Semantic Web approach is gaining increasing attention within the AEC domain as promising strategy to tackle these problems. It provides a mechanism through which information silos can exist in a homogeneous format, so users are able to combine data from other relevant data sources. This thesis uses the Semantic Web approach, so called Linked Data, to resolve one of the main challenges for BIM implementation during O&M; the lack of up-to-date information (Kiviniemi & Codinhoto, 2014).

In order for the model to remain up-to-date, model versioning is crucial. To be able to determine whether changes in data represent the actual as-used situation, information about the origins of specific data and the processes that this data underwent is needed. This information is referred to as provenance and eases the process of data versioning. Therefore, the main objective of this thesis is to develop an application that facilitates a BIM-based workflow for efficient handling of mutations in the O&M phase. The BIM acts as a

central information access point. The application should enable the asset manager to visualize and alter O&M data in a 3D model environment. The provenance information of these changes needs to be captured in a maintenance history log, which asset managers are required to keep for e.g. fire alarm systems. These logs are currently often missing, and if present, manually kept next to other maintenance processes.

In the application area of O&M there is an increasing need to identify and capture non-geometrical information to secure an successful implementation of BIM, because buildings produce more data than ever before. Examples of this data production can be found in energy usage, utility information, occupancy patterns, weather data, inner climate of rooms, etc.. In order to manage a building, it is essential to use knowledge from across these various data sources. However, many barriers exist to the interoperability of these sources and there is only little interaction between these 'islands of information' (Curry, et al., 2013).

In addition, tracking changes in IFC-based information is difficult. Ideally to capture changes, IFC files utilize the concept of object versioning. As proposed by Nour and Beucke (2010), object versioning enables having several versions of the content (attribute values) of an object. This means that a new object version is only created when the object's attribute values change, otherwise the same object version is shared among several model versions. However currently model difference is tracked mainly using GUIDs, i.e. objects of the same GUIDs in the different model versions are compared. Incorrect management of these GUIDs, or operations that delete an object and create a new one in its place will render this process ineffective. Therefore, given the former, the Semantic Web approach is deemed promising to tackle these issues.

The official state of the art concerning the concept of versioning in Linked Data is called reification. Through reification, data is described in order to capture provenance information regarding when certain statements are made, who made them or other similar information (W3C, 2007). Using reification, changes in data sets can be captured and subsequently queried in order to gain insight in the history of the building and it's objects.

The conducted interviews have pointed out the current inefficiency of information management in O&M, which is mainly caused by a mismatch in informational needs between design/construction and O&M. The latter is often neglected in the creation of a BIM. The lack of up-to-date information is a universal issue that interviewees recognized as well, as they currently have very little protocols in place that ensure the information is up-to-date at all times. Nonetheless the potential of 3D models for O&M in order to manage and structure information appears to be significant and recognized among asset managers. The most important information that needs to incorporated in such a model for O&M, which is recognized by the interviewees as well as reviewed literature is:

- the surface area of rooms in a building;
- quantities of building elements and system components;
- product supplier information;
- functions of rooms;
- service history data;
- condition assessment of objects.

These aspects were taken into account in the development of the application. Previous to this development, a data set was provided with maintenance data with regards to the components of a fire alarm system in a building. For these components, maintenance history logs are mandatory. In order to connect this data set to an IFC building model, an ontology was created that expresses the data set with the Semantic Web standard. The ontology is visualised in figure 1.

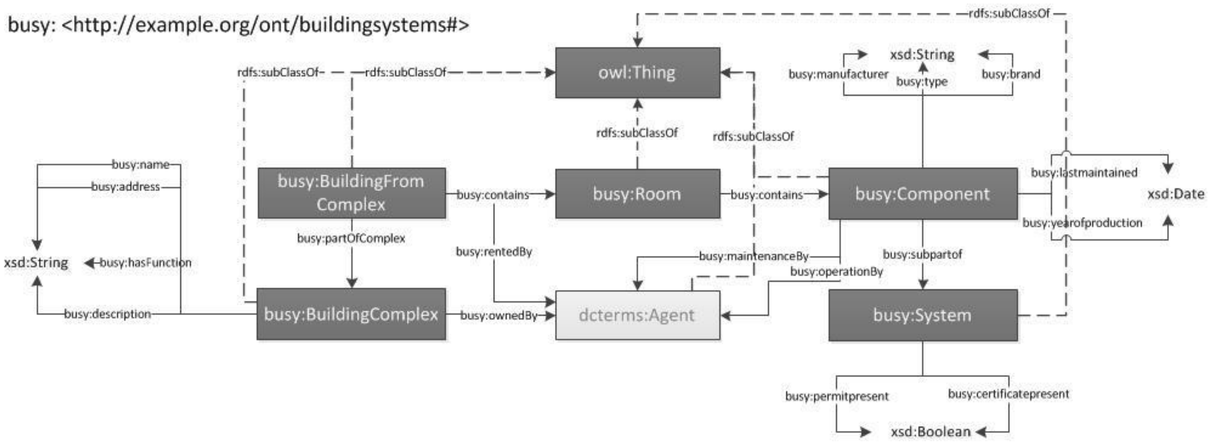


Figure 1: Ontology expressing the system data set

The application firstly connects the semantic data set to the IFC model through the use of common element indicators through which information can be accessed through the objects in the BIM. The application subsequently facilitates replacing components in the initial data set, as well as adding log entries which provide the date at which the component in question was last maintained. Figure 2 visualises this principle, where each replaced component has a relation to the one that replaced it. By querying the component that does not have this relation, the application makes sure that the information that is displayed is up-to-date. In addition to the new data instance, a separate IFC file is also created. This file contains an IFC element for each of the components that have been changed using the tool, with the current properties that are in the replaced component as well. This way, these IFC files act as changesets so that for each moment in time the current status at that time can be inspected.

Finally the application facilitates saving the log entries to a CSV file, which can be opened with Microsoft Excel. This makes sure that, along with the security that information is up-to-date, the mandatory maintenance history logs can always be extracted from the application and contain the information that is necessary for these files.

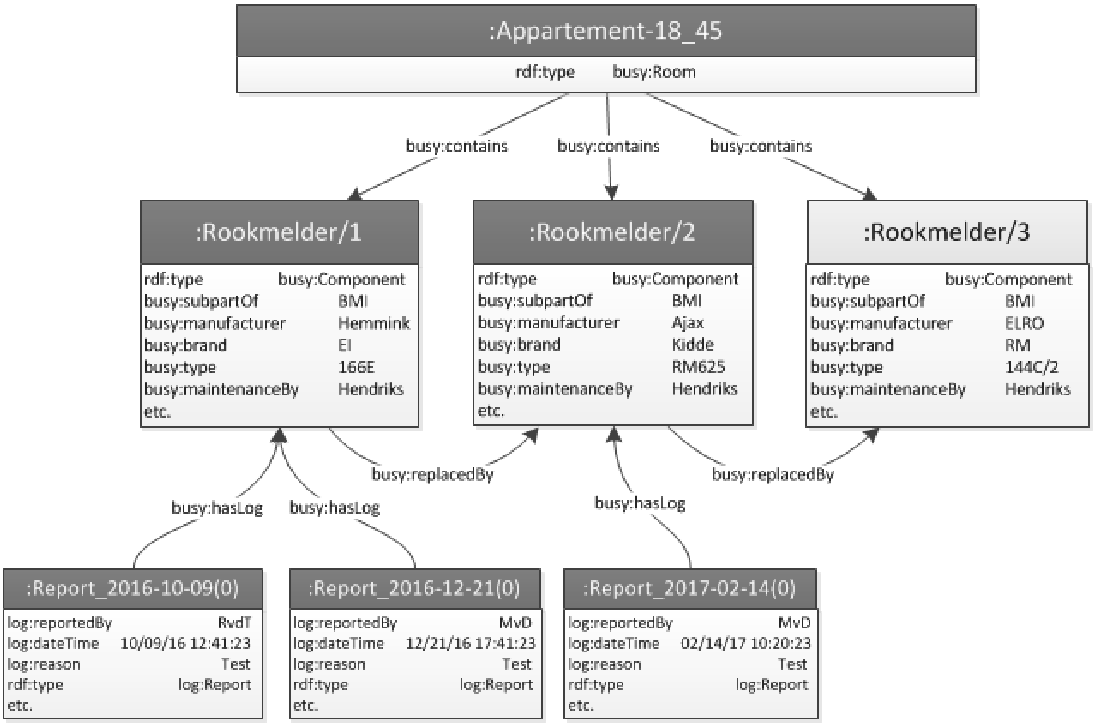


Figure 2: Schematic overview of developed versioning mechanism

The efforts to develop a working prototype have resulted in a usable and functioning tool. However, the bulk of current O&M working methods do not yet include working with BIM. Despite this, the proposed working method appears to be more future proof due to the expected shift towards Semantic Web technologies within the industry. This format offers improved data reusability and interoperability, for instance by using queries as demonstrated in the tool. This has the potential to enhance data sharing between AEC and FM industry, facilitating e.g. design for maintenance.

The main objective of this research was to develop a BIM-based workflow how changing information can be handled in the O&M phase. The core of this lies mostly in the use of the Semantic Web standard to connect the BIM to external data sets, and subsequently the ability to alter the properties of elements. This ensures that the information is always up-to-date. The addition of a single triple indicates the timeliness of this information. The usage of Semantic Web technologies enables this improvement, as data elements are made available as unique identifiers which can be addressed and accessed by the application. This way knowledge about elements in a building can be captured in an easy and retraceable way.

The improvement that has been demonstrated in this thesis lies in the use of BIM in combination with the Semantic Web standard in the visualisation and connection of data with 3D models and the security that data is up-to-date. In addition, this way the history of objects in a building can be inspected at all times. Here lies a big opportunity for the construction sector to improve information management in the exploitation phase.

SAMENVATTING

De architectuur, engineering en constructie (AEC) industrie heeft de afgelopen jaren grote veranderingen doorgemaakt. De meeste gebouwen uit de jaren ’80 en ’90 werden door middel van 2D-tekeningen ontworpen. Hierdoor vereist onderlinge overdracht van informatie tussen partijen digitale ofwel fysieke verzending van deze tekeningen. Bij wijzigingen is hernieuwde uitwisseling noodzakelijk, waardoor elke partij vaak op een informatiedrager werkt. Dit proces is duidelijk foutgevoelig.

Een manier om dit proces in te richten die de laatste jaren steeds meer voeten aan de grond krijgt in de bouwsector is het gebruik van Bouwwerk Informatie Modellen (BIM). BIM omvat elke vorm van gebouwinformatie en ontsluit deze informatie in een 3D-omgeving. Echter, BIM-implementatie in de bouwsector richt zich nog steeds voornamelijk op ontwerp en constructie, waar het aantal bedrijven in beheer & onderhoud dat BIM omarmd heeft zeer klein is. Gezien het feit dat de gebruiksfase, met een duur van 50 jaar of langer, de langste is in de levenscyclus en de kosten in deze fase vaak die van ontwerp/uitvoering tot vele malen overschrijdt, lijkt het inzetten van BIM in deze fase potentieel zeer winstgevend.

Open standaarden als de Industry Foundation Classes (IFC) hebben de potentie tot betere integratie tussen de domeinen van ontwerp, uitvoering en beheer & onderhoud. IFC is een open en gestandaardiseerd datamodel bedoeld voor interoperabiliteit te verbeteren tussen verschillende BIM softwarepakketten. IFC-bestanden bevatten niet alleen geometrie van objecten, maar ook eigenschappen in propertysets en metadata welke relateert aan andere objecten in het gebouw. Echter, in de gebruiksfase komt het vaak voor dat gebouwdata verandert om de as-used situatie te vertegenwoordigen. De IFC-standaard biedt nog steeds niet genoeg handvaten om deze veranderingen te verwerken, evenals het vastleggen van geschiedenis van de objecten in een BIM (JalyZada, Tizani, & Oti, 2014). Om deze reden is informatie vaak verouderd, zodat het gebouw nogmaals bezocht dient te worden met als enig doel om deze informatie (terug) boven tafel te krijgen. Ontbrekende, verouderde of ongestructureerde informatie zorgt voor ineffectief projectmanagement, onzekere procesuitkomsten en tijdverlies of kostenstijgingen in onderhoud en herontwikkeling (Volk, Stengel, & Schultmann, 2014).

Hierdoor krijgt de Semantic Web-aanpak meer en meer aandacht binnen het AEC-domein als een veelbelovende strategie om deze problemen aan te pakken. Het biedt een manier waarop losstaande informatiesilo’s kunnen bestaan in een homogeen format, zodat gebruikers data kunnen combineren vanuit andere relevante databronnen. Deze thesis gebruikt de Semantic Web-standaard, ofwel Linked Data, om een van de belangrijkste uitdagingen voor BIM-implementatie in beheer & onderhoud te verhelpen; het gebrek aan actuele informatie (Kiviniemi & Codinhoto, 2014).

Om ervoor te zorgen dat het model up-to-date blijft, is versiebeheer van het model noodzakelijk. Om te bepalen of veranderingen in data de actuele as-used situatie

vertegenwoordigen is informatie over de herkomst van data en de processen die deze data heeft ondergaan nodig. Deze informatie wordt *provenance* genoemd, en vergemakkelijkt het proces van versiebeheer van informatie. Daarom is de hoofddoelstelling van deze thesis om een applicatie te ontwikkelen die een workflow faciliteert gebaseerd op BIM-modellen welke veranderingen in informatie in beheer & onderhoud efficiënt verwerkt. Het BIM heeft hier de plek van centraal toegangspunt van informatie. De beheerder moet in de applicatie beheer- en onderhoudsdata in een 3D-omgeving kunnen inzien en aanpassen. De *provenance* informatie over deze veranderingen dient te worden vastgelegd in een onderhoudslogboek, welke verplicht zijn voor bijv. brandmeldinstallaties. Deze logboeken ontbreken vaak, en als zij aanwezig zijn, worden deze met de hand bijgehouden los van andere onderhoudsprocessen.

In het toepassingsgebied van beheer & onderhoud heerst een groeiende behoefte om niet-geometrische informatie vast te leggen om zo een succesvolle implementatie van BIM te verzekeren. Gebouwen produceren tegenwoordig namelijk meer data dan ooit tevoren, van energiegebruik en gebruiksinformatie tot beschikbaarheid en binnenklimaat van ruimtes. Om een gebouw effectief te beheren is het belangrijk om kennis van al deze databronnen te gebruiken. Er bestaan echter vele barrières voor de interoperabiliteit van deze bronnen, en er bestaat weinig interactie tussen deze 'informatie-eilanden' (Curry, et al., 2013).

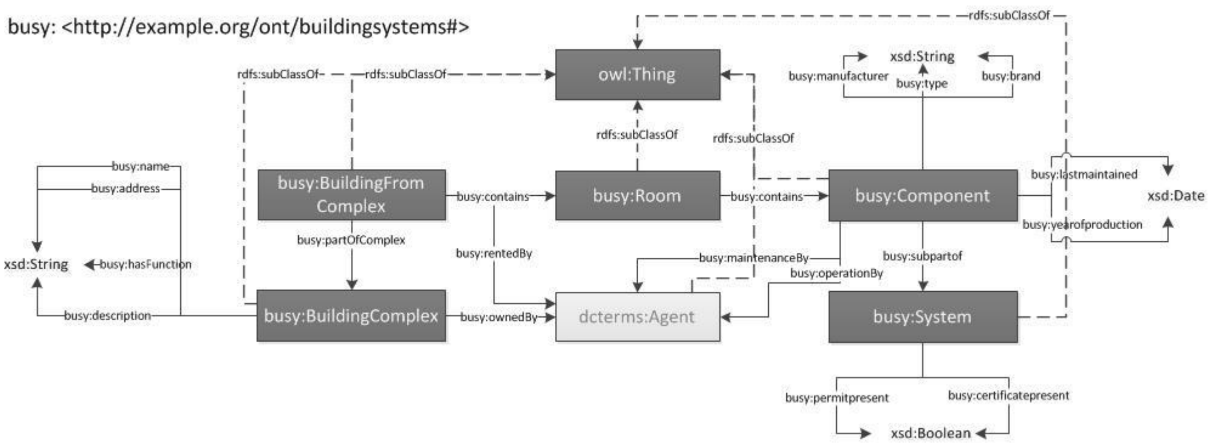
Daarbij komt dat het bijhouden van veranderingen in IFC-gebaseerde informatie problematisch is. Idealiter zouden IFC-bestanden het concept van *object versioning* toepassen om veranderingen vast te leggen. Zoals voorgesteld door Nour en Beucke (2010) houdt *object versioning* in dat er enkel een nieuwe versie van het object gemaakt wordt als de waardes van de eigenschappen van dat object veranderen. Zo niet, wordt dezelfde objectversie gedeeld door meerdere modelversies. Echter, veranderingen tussen modellen worden nu veelal bijgehouden door middel van GUIDs, d.w.z. objecten met dezelfde GUID in verschillende versies worden vergeleken. Onjuist beheren van deze GUIDs of bewerkingen die een object verwijderen en een nieuwe in zijn plaats creëren zorgen ervoor dat dit proces zijn werking verliest. Daarom, gezien het voorgaande, wordt de Semantic Web-standaard gezien als veelbelovend om deze problemen te aan te pakken.

De huidige state of the art met betrekking tot versiebeheer in Linked Data wordt reïficatie genoemd. Door middel van reïficatie wordt data beschreven zodat *provenance* informatie kan worden vastgelegd met betrekking tot wanneer bepaalde statements zijn gemaakt, wie ze heeft gemaakt en andere vergelijkbare informatie (W3C, 2007). Door middel van reïficatie kunnen veranderingen in datasets vastgelegd worden en bevraagd worden om zo inzicht te krijgen in de geschiedenis van het gebouw en haar objecten.

De afgenomen interviews hebben de huidige inefficiëntie van informatiebeheer gedurende beheer & onderhoud aangetoond, welke voornamelijk wordt veroorzaakt door een mismatch in informatiebehoefte tussen ontwerp/uitvoering en beheer & onderhoud. Die laatste wordt vaak niet meegenomen in het opzetten van een BIM. Het gebrek van actuele informatie is een universeel probleem dat de geïnterviewden allen herkenden, gezien het feit dat zij zeer weinig protocollen in werking hebben om informatie up-to-date te houden. Niettemin werd de potentie van 3D-modellen voor beheer & onderhoud om zo informatie te structureren en beheren herkend onder beheerders. De belangrijkste informatie die in een dergelijk model dient te worden opgenomen, welk wordt onderstreept door de geïnterviewden en literatuur, is het volgende:

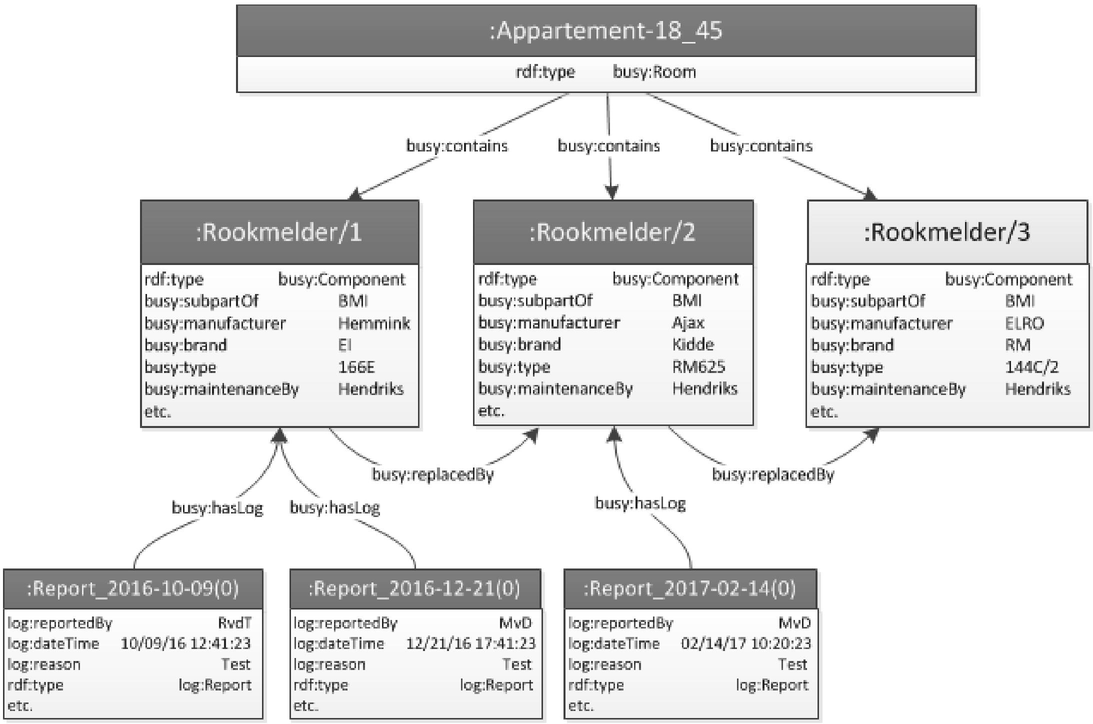
- oppervlakten van ruimtes in een gebouw;
- hoeveelheden van bouwelementen en installatiecomponenten;
- productinformatie van leveranciers;
- functies van ruimtes;
- onderhoudsgeschiedenis;
- conditiemetingen van objecten.

Deze aspecten zijn meegenomen in de ontwikkeling van de applicatie. Voorafgaand aan deze ontwikkeling is een dataset aangeleverd met onderhoudsinformatie over de componenten in een brandmeldinstallatie in een gebouw. Voor deze componenten is het wettelijk voorgeschreven om logboeken bij te houden. Om deze dataset te kunnen koppelen aan een IFC-gebouwmodel is er een ontologie opgesteld dat deze dataset uitdrukt door middel van de Semantic Web-standaard. De ontologie is weergegeven in figuur 1.



Figuur 1: Ontologie die aangeleverde installatie-dataset beschrijft

De applicatie koppelt eerst de semantische dataset met het IFC-model door middel van gemeenschappelijke elementindicatoren. Hierdoor kan informatie worden ontsloten door de objecten in het BIM. Vervolgens zorgt de applicatie ervoor dat componenten kunnen worden vervangen in de oorspronkelijke dataset en dat melding gedaan kan worden van onderhoud. Met zulke meldingen wordt de datum waarop de component laatst is onderhouden toegevoegd aan de dataset. Figuur 2 visualiseert dit principe, waarbij elke vervangen component een relatie krijgt naar de component die deze heeft vervangen. Door de component te zoeken die deze relatie niet heeft, zorgt de applicatie ervoor dat de informatie die wordt weergegeven altijd actueel is. In aanvulling op de nieuwe data-instantie wordt ook een afzonderlijk IFC-bestand gemaakt. Dit bestand bevat een IFC-element voor elke component die is vervangen in de tool, met de actuele informatie die aan de vervangen component is toegevoegd. Op deze manier vormen deze IFC-bestanden zogenaamde changesets, waarmee de huidige status voor elk moment in de tijd kan worden bekeken.



Figuur 2: Schematische weergave van ontwikkeld versiebeheer-mechanisme

Tot slot zorgt de applicatie ervoor dat de meldingen die met de tool gedaan worden, kunnen worden opgeslagen naar een CSV-bestand dat kan worden geopend met Microsoft Excel. Dit zorgt ervoor dat, naast de zekerheid dat informatie altijd up-to-date is, de verplichte logboeken altijd uit de applicatie gehaald kunnen worden en deze de informatie bevatten om deze logboeken consequent te vullen.

Het streven om een werkend prototype te ontwikkelen heeft geresulteerd in een bruikbare en werkende tool. Echter is het werken met BIM nog geen onderdeel van het merendeel van huidige beheer- & onderhouds-processen. Ondanks dit lijkt de voorgestelde werkmethode meer toekomstbestendig gezien de verwachte overgang naar Semantic Web-technologieën in de bouwsector. Dit format biedt beter hergebruik van informatie en interoperabiliteit, zoals met query's zoals in de tool gedemonstreerd. Dit biedt de potentie om het delen van informatie tussen de AEC en beheer & onderhoudssector te verbeteren.

De hoofddoelstelling van dit onderzoek was om een op BIM gebaseerde workflow te ontwikkelen hoe op een efficiënte manier omgegaan kan worden met veranderende informatie gedurende beheer & onderhoud. De kern van de oplossing hiervoor ligt voornamelijk in de inzet van de Semantic Web-standaard om het BIM te koppelen aan externe datasets en om vervolgens de eigenschappen van elementen te kunnen veranderen zoals gedemonstreerd in figuur 2. Dit verzekert dat de informatie altijd up-to-date is. Het toevoegen van een enkele relatie verzekert deze actualiteit. Het gebruik van Semantic Web-technologieën maakt deze verbetering mogelijk door middel van unieke indicatoren welke kunnen worden benoemd en benaderd door de applicatie. Op deze manier kan kennis over bouwelementen worden vastgelegd op een gemakkelijke en herleidbare manier.

De verbeteringen die gedemonstreerd zijn in deze thesis liggen in het gebruik van BIM in combinatie met de Semantic Web-standaard om zo data vanuit verschillende bronnen te kunnen koppelen en te visualiseren in 3D-modellen met de zekerheid dat data actueel is. Daarnaast kan op deze manier de geschiedenis van bouwelementen op elk moment kan worden bekeken. Hierin ligt een grote kans voor de bouwsector om informatiebeheer in de gebruiksfase van gebouwen te verbeteren.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF LISTINGS

LIST OF ABBREVIATIONS

3D	: three dimensional
AEC	: Architecture, Engineering and Construction
BIM	: Building Information Model(ling)
CAD	: Computer Aided Design
COBie	: Construction Operations Building information exchange
CSV	: Comma-Separated Values
GUID	: Globally Unique IDentifiers
FM	: Facility Management
IAI	: International Alliance for Interoperability
IFC	: Industry Foundation Classes
LOD	: Level Of Detail/Development
MEP	: Mechanical, Electric and Plumbing
MVD	: Model View Definition
O&M	: Operations & Maintenance
OWL	: Web Ontology Language
RDF	: Resource Description Framework
RDFS	: Resource Description Framework Schema
SPARQL	: SPARQL Protocol And RDF Query Language
TTL	: Turtle
URI	: Unique Resource Identifier

1 INTRODUCTION

In the last decade the architecture, engineering and construction (AEC) industry has seen major changes. In the 90s most buildings were designed in 2D, which still is common practice for a lot of companies nowadays. The handover of these drawings and other files to another party is done by sending either the digital file or the physical paperwork. When changes are made in the original drawing, later on the new version had to be sent again. This repeatable handover process is obviously susceptible for mistakes.

For several years, instant access to up-to-date information is becoming more and more standard practice in the AEC industry, due to more integrated project teams. These teams can extend to a wide variety of parties depending on the work required, e.g. the client, architect, engineers, contractors, project managers, MEP (mechanical, electrical and plumbing) engineers, etc.. A way to lead this process that is gaining significant momentum in the AEC sector nowadays is by using building information modelling (BIM). A building information model (BIM) is not just a 3D representation of the building design, but encompasses every form of building-related information – both geometrical and non-geometrical. The basic idea is that data regarding a building is created and is then centrally available for every party in the chain, from design to construction to operations and maintenance (Visser, de Boer, & van der Voet, 2013).

Integration of BIM in the AEC industry however still focuses predominantly on the design and construction phases and the number of companies that have adopted BIM in the post-construction phases is still very limited. Though BIM processes are established for new buildings, the majority of existing buildings are not maintained, refurbished or deconstructed using BIM nowadays (Volk, Stengel, & Schultmann, 2014). However, in terms of money, the subsequent operations and maintenance (O&M) costs of a building over its life cycle could amount to many times more than the design/construction costs (Becerik-Gerber, Jazizadeh, Li, & Calis, 2012). Figure 1 shows the proportional costs of the design phase, realization phase and exploitation phase. In addition, the exploitation phase is also the longest phase with a duration of 20 years, 50 years or sometimes even longer. Given that the exploitation phase is the most important phase in the life cycle of buildings, potential benefits of using BIM in this phase seem to be significant.

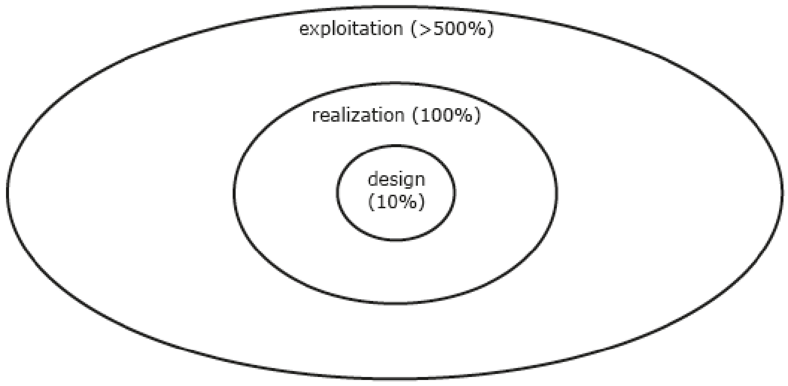


Figure 1: Financial proportions of the building and exploitation costs (Visser, de Boer, & van der Voet, 2013)

Open standards such as the Industry Foundation Classes (IFC) have the potential to integrate the different domains of design, construction and facility management. However in the exploitation phase, building information often changes when representing the as-used situation. These changes – albeit the replacement of a boiler or changing the layout of an entire floor – seldom need the complete model in order to process the changes made to the building. They generally only require a subset of the model, or even a small portion of the information behind the model. However, these changes are often not documented at all.

Even though BIM has shown numerous improvements, it has also been criticized in both the academic and corporate domain. This is due to the fact that the amount of non-geometrical information within the AEC domain, and especially data coming from the operational phase, cannot be adequately stored in a Building Information Model. The IFC standard alone does not suffice in enabling interoperability with systems outside of the AEC domain (Curry, et al., 2013). The sources of this data, including the BIM itself, are usually locally stored and often cannot connect with each other (Dankers, van Geel, & Segers, 2014). In addition, the IFC specification lacks adequate handling of the following (JalyZada, Tizani, & Oti, 2014):

- Changes made in the BIM and associated information;
- Information sharing and changes effected among different BIM models;
- Records of the history of objects in the BIM models;
- Records relating to earlier versions of the BIM models.

As a result, the Linked Data approach is increasingly gaining attention within the AEC domain as one of the most promising strategies to tackle these interoperability problems. It does so by separating the actual data from its authoring tools and relying on data representation in a Linked (Open) Data format (Pauwels, 2014). It thus provides a mechanism through which information silos can exist in a homogeneous format. By utilizing Linked Data to represent building data – such as the BIM – the user is able to combine data from other relevant domain silos. In doing so, organizations can generate and extract additional value from current stand-alone repositories across multiple domains (Curry, et al., 2013).

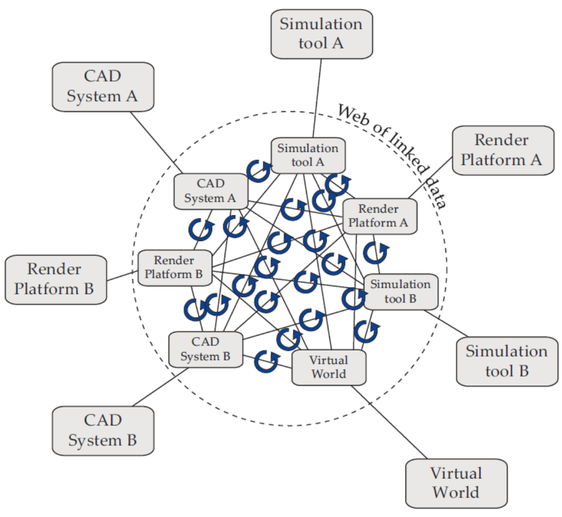


Figure 2: The Linked Data process of information exchange (Pauwels, 2014)

2 RESEARCH APPROACH

This chapter outlines the research approach. Therefore, the problem definition are defined in section 2.1, after which the research objective and research questions that follow from it are given in section 2.2. Section 2.3 discusses the research design through a methodological justification and a research model. Afterwards section 2.4 provides an outline of the thesis.

2.1 Problem definition

Nowadays, an increasing amount of companies in the AEC domain is using Building Information Modelling (BIM) in their business processes. The FM industry could also benefit from using BIM, but its use in practice is very limited. The amount of literature indicating potential benefits of BIM use for FM keeps expanding (Becerik-Gerber, Jazizadeh, Li, & Calis, 2012; Wetzel & Thabet, 2015; Volk, Stengel, & Schultmann, 2014; Roper & Payant, 2014).

In the FM phase, building owners have a wealth of external data available that is to some extent related to (one of) their building(s), e.g. cost estimation or product specifications (Dankers, van Geel, & Segers, 2014). However, such data is usually not linked to relevant elements inside of a BIM but stored in non-queryable data structures. This results in ineffective information use, limited re-use in future projects and a lot of double work when the information changes, which it often does during O&M to represent the as-used situation. In the case of small alterations, only partial drawings or schematics are used. This practically guarantees that the information will not be updated in the database. Drawings that are 10-20% out of date are seen as obsolete, which means a year's worth of failure to update ruins a good set of facility drawings (Roper & Payant, 2014). Therefore it seems practically relevant to ensure that drawings and, even more so, the information remains up-to-date. Kiviniemi and Codinhoto (2014) identify this lack of up-to-date information as one of the main challenges for BIM implementation during FM.

Lately, linking the BIM to external, heterogeneous data sets to improve information availability, accessibility and correlation is gaining scientific interest (Pauwels & Terkaj, 2016; Vanlande, Nicolle, & Cruz, 2008; Curry, et al., 2013; Schevers, et al., 2007). However, in order for the model to remain up-to-date, model versioning is crucial. An important enabler to efficiently manage changes between models is the capability to accurately detect changes from one version to another (Oraskari & Törmä, 2015). In order to determine whether changes in data represent the actual as-used situation, information about the origins of specific data and the processes that this data underwent is needed. This information is referred to as provenance (Petrinja, Stankovski, & Turk, 2007) and eases the exchange of versioning information (Zhao, Bizer, Gil, Missier, & Sahoo, 2010). It therefore seems beneficial to track provenance data.

Several ways to track differences have been introduced already. Oraskari and Törmä (2015) suggest algorithms in order to compute differences between successive versions of IFC files using Linked Data technologies, whereas JalyZada et al. (2014) propose an extension of the

IFC schema to include IfcVersion entities. To track and query provenance, Halpin and Cheney (2011) introduce named graphs, a key concept of the Semantic Web, whose purpose is to provide a record of how raw data in a data set changes over time. These named graphs group sets of Resource Description Framework statements, allowing descriptions to be made of that group of statement. This is also called reification (W3C, 2007). In addition, the concepts used in Linked Data might be subject to change as well (Kondylakis & Plexousakis, 2014; Plessers & De Troyer, 2005). Typically changes in the model as well as concepts used in the model are not recorded, as the data source is modified in place – if at all – and only the current state is kept (Meinhardt, 2015). Yet, such versioning is of particular value for O&M.

2.2 Research objective

The objective of the research is to gain insight in the O&M processes, the data required for those processes and problems and room for improvements in those processes. Given 1) the need for information models with interoperability between data sources in the O&M phase, 2) insufficient management of changing information and 3) the possibilities of Linked Data, it appears beneficial to investigate the added value of BIM as well as Semantic Web technologies in the O&M phase.

The main objective of this research is to develop an application, facilitating a BIM-based workflow for efficient handling of mutations in the FM phase. The BIM as such acts as a central information access point. This application should enable the facility manager to view, visualize and, if necessary, change O&M data in a 3D model environment. Additionally, changed data should be easily detected. Finally, for certain objects facility managers are required to keep a maintenance history log. The changes with respect to these objects should be captured in such a log, which are currently kept by hand apart from other maintenance processes. Further research and testing is needed to improve and test the prototype tool for practical integration.

2.2.1 Research questions

The main research question resulting from this problem definition is the following:

> *How can changing information in the Operations & Maintenance phase be handled efficiently, using 3D Building Information Models and Semantic Web technologies?*

In order to support this main research question, several sub-questions have been defined:

1. What are the most important processes in O&M, and which information is necessary for these processes?
2. What is Building Information Modelling and how can it facilitate information storage and exchange during the O&M phase?
3. What are important IFC model conditions for efficient use in the O&M phase?
4. What approach can be employed to track changes in information in BIM models?
5. How can different model versions be mapped using Linked Data?

2.2.2 Limitations

The scope of the project is restricted to the information aspect of fire safety. In addition, the application will be tested in a use case provided by an external partner.

2.3 Methodological justification

The research is conducted in three different stages. The combination of the three methods provides a solid foundation for the outcome of the research. Firstly, a literature review is conducted to gain knowledge about the subject, current research on BIM, Linked Data and fields of improvement. Secondly, in order to underline the literature review findings and to gain qualitative information, interviews are conducted. This way more knowledge is obtained with regards to the industry's latest developments and views on future needs. Finally, based on the findings of the literature review as well as the expert views, an application is developed to keep information up-to-date in a BIM in the FM phase, as well as form maintenance history logs. The workflow is then validated based on a pilot project.

These three methods not only generate a proper workflow to handle changing information, but also ensure that it will be supported by the end users. This way the facility management process gains significant efficiency. All of the information regarding real estate is present, clearly structured and always up-to-date, making the information reliable and eventually less personnel is required (Theeuwen & Smit, 2016).

2.3.1 Literature review

The literature review provides a deeper understanding regarding the subject and process. Different sources have been used for the literature review; books, white papers, scientific publications, conference contributions, master and doctoral theses and websites. The following key words are used:

- building information modelling;
- facility management;
- operations and maintenance;
- Linked Data;
- Semantic Web;
- Versioning OR model revision;

2.3.2 Interviews

Using the developed knowledge gained through the literature review and to underline the findings of this literature review, expert interviews will be conducted to provide better understanding of the current situation in the AEC and FM domain regarding the main topic of this research. Experts in the field of BIM, O&M as well as installation engineers that will perform the alterations will be interviewed. This way more knowledge can be obtained regarding the industry's latest developments and views on future needs. Based on these interviews, an application will be developed to support BIM integration in the FM domain.

For this research, semi-structured interviews were chosen above structured interviews. Using a semi-structured format, the interviewee has more space to give extra explanation or additional details about questions if need be. It is more open and slightly less formal than a fully structured interview. Because it is also more flexible, using semi-structured interviews results in a better understanding in the interviewee.

2.3.3 Pilot case

The developed application will create a 3D visualisation of the building for the facility manager in which information can be retrieved and queried through its objects. External data sets on the topic of fire safety from a real world use case will be connected to the BIM by using Linked Data. In addition, the information as well as placement of the objects can be changed and these changes can be viewed in a log.

To validate if the application is valid, a pilot case will be conducted. This pilot case is based on a BIM which is created on the basis of 2D drawings. When the model is ready and contains the exchange requirements that stem from the interviews, the application will be tested. The outcome will be analysed and flaws will be listed for future development. The pilot case is independent because the actual BIM will be created during the pilot case. Python programming language and various modules such as IfcOpenshell and RDFLib are used for development.

The case study is a real world project called ‘Ketsheuvel’ as commissioned by the housing association Zayaz in ‘s-Hertogenbosch. The building was the main office for Zayaz and was recently transformed to apartments. In tool testing and validation, case data is used and a model will be created based on the available drawings. Manageable parts of this case are used to test and validate if a working and usable application is created.

2.3.4 Research model

This section will provide an outline of the research model which will be used as a framework for this thesis, depicted in Figure 3. Firstly, the set research objectives are translated into research questions. Secondly, literature review is done to set the base of the research by investigating BIM, facility management and Linked Data in particular. The conclusions of this literature research, along with interviews with experts in the field of facility management as well as others in the AEC industry, will provide guidelines and recommendations for the development of the tool. The literature study aims to answer a number of sub-questions.

The development of a tool to facilitate efficient handling of mutations in the FM phase, as mentioned, will make up the third part of the research. In order to shape and further test the tool a case study will be used, as well as to gain insight into the practical possibilities and pros and cons of the use of BIM in FM. Finally, the tool must be verified and conclusions will be drawn regarding the research question as well as recommendations for future research.

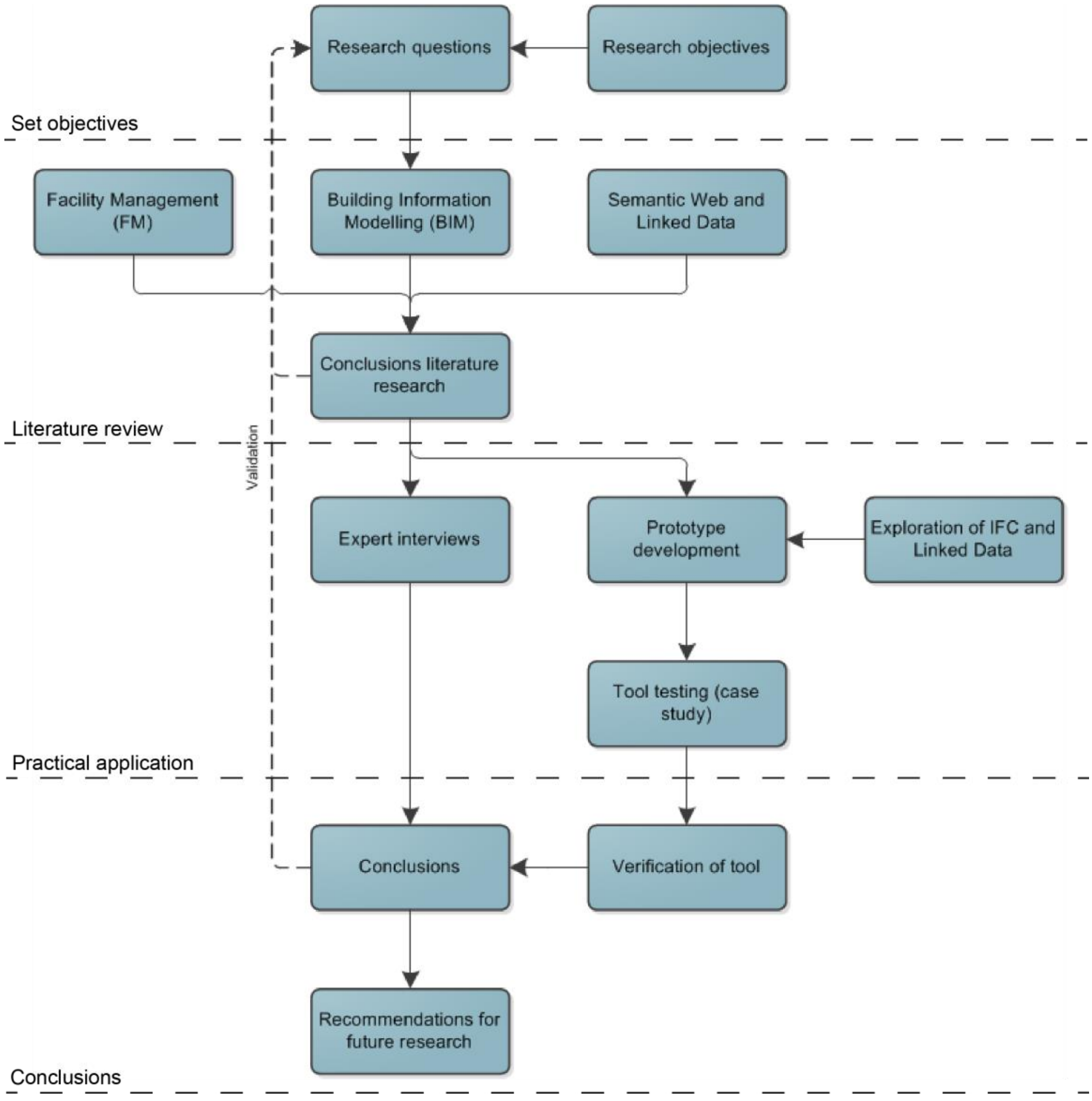


Figure 3: Research model

2.4 Thesis outline

This research starts with a literature review based on three aspects. Chapter 3 centres around two various practices in the exploitation phase of buildings, namely Facility Management (FM) and Operations & Maintenance (O&M). Chapter 4 provides an overview of the phenomenon of Building Information Modelling, which contains a definition, benefits and obstacles and an elaboration upon open standards. Given some of the drawbacks of these open standards, chapter 5 aims to elaborate upon the Semantic Web standard and how this can be used to alleviate these drawbacks. Chapter 6 summarizes the interviews that were held in order to form conclusions based on the qualitative part of the research. Chapter 7 sees the development and validation of a prototype tool that keeps information up-to-date in the O&M phase. Finally, chapter 8 provides the conclusions based on the previously defined research questions and chapter 9 forms recommendations for further research as well as for O&M companies to be able to implement BIM in their processes.

3 Facility management and Operations & Maintenance

The purpose of this part of the thesis is to explore the domain of facility management (FM) and operations and maintenance (O&M), in order to get a better understanding of the principles and application possibilities. Firstly a definition is given in section 3.1 for both these terms to be used throughout this research, as well as maintenance strategies. Subsequently the core activities are mentioned in section 3.2. Afterwards section 3.3 discusses differences in information management software, after which section 3.4 concludes with information requirements for O&M processes.

3.1 Facility management and Operations & maintenance

Facility management (FM) is defined as the integration of processes within an organisation to maintain and develop agreed services which support and improve the effectiveness of its primary activities (Comité Européen de Normalisation, 2006). FM constitutes an extensive field encompassing multidisciplinary and independent disciplines whose overall purpose is to maximise building functions while ensuring the wellbeing of occupants.

The operation and maintenance (O&M) of these facilities encompasses all of the services that are required to assure the facility will perform the functions for which it was designed and constructed (Sapp, 2015). It includes the day-to-day activities necessary for the building and its systems and equipment to perform their intended function, such as HVAC, electrical, plumbing, fire detection, but also the functioning of the doors, windows and outside infrastructure. For this research, operations and maintenance are combined into the common term O&M, because a building cannot operate efficiently without being maintained. Maintenance activities are divided into the following categories (Sullivan, Pugh, Melendez, & Hunt, 2010):

- Reactive maintenance, which is a strategy in which equipment is run until it breaks and no actions or efforts are taken to maintain the equipment as the designer originally intended to ensure design life is reached;
- Preventive maintenance, which contains actions taken on a time- or machine-run-based schedule that detect, preclude, or mitigate degradation of a component with the aim of sustaining or extending its useful life through controlling degradation to an acceptable level;
- Predictive maintenance, in which the onset of system degradation is measured and thereby allowing the reasons of this degradation to be eliminated or controlled prior to any significant deterioration in the component physical state. This type of maintenance differs from preventive maintenance by basing maintenance need on the actual machine condition rather than some pre-set schedule.

While these strategies are basic but fundamental, due to the continuous technological evolution, the concept of maintenance has been renewed. A shift is detected from repair activities and an ad hoc approach to a complex system of management, directed towards the prevention of the fault and continuous improvement. The task of maintenance in this context is to cooperate throughout the entire life cycle of a building, from design to demolition, with the intent of continuous improvement of operational availability and reduction of maintenance cost (Fruguglietti, Pasqualato, & Sagula, 2012). However, the current situation is that the AEC and FM industries are still dissociated, as teams concerned with the process of facility management are rarely those that have participated in the conception and construction of the building (Vanlande, Nicolle, & Cruz, 2008).

It is the task of O&M managers to keep the facility running to maximise efficiency and minimise financial costs involved. Therefore, optimally maintenance is carried out as little as possible while preserving the availability of a building according to their intended function. This means maintenance of its components is carried out infrequently. In other words, maintenance should solely be done if necessary to ensure the continued, safe and profitable use of the building at acceptable levels of satisfaction or when the possibility arises of extending the useful life of the elements of the building (Horner, El-Haram, & Munns, 2001). In order to fulfil this task, it is crucial for the O&M managers to be aware of the facility's state. Only when aware of the current state, maintenance can be planned preventively and carried out if necessary. This reduces the corrective maintenance actions and the downtime of the components, which keeps the facility in optimal condition.

O&M costs are often overlooked at the design phase by owners and project stakeholders although they can amount up to over half of the total building life cycle costs (Becerik-Gerber, Jazizadeh, Li, & Calis, 2012) and over five times the costs made in the design phase (Visser, de Boer, & van der Voet, 2013). A medium-sized, major headquarters can have 50,000 requests for service annually, and four times that number of preventive maintenance items corrected. Not only is this a high volume, but each of those service requests has a customer depending on the facility manager to respond effectively (Roper & Payant, 2014).

In order to adequately maintain buildings, rooms and its parts, information is crucial. Most of this information is paper-based. Facility managers have drawings at their disposal, saved in digital format as well as in a physical archive. Many of them have a map library (Dutch: cartotheek) in which PDF scans are saved. It may be central, but also completely static. The question whether or not a drawing is still accurate is often justified. Many drawings are either not up-to-date or absent altogether due to a missing original file or multiple versions in use (Verbaan, van der Voet, de Boer, Visser, & de Koe, 2014). This causes a lot of valuable time to be lost looking for and verifying the right 'as-built' building information as well as information about previous activities.

3.2 The O&M process

Though usually facility management plays a key role in the operation of a building, it has the most effect if incorporated during earlier phases of a building's life cycle. In general, the ideal approach implements the role of the facility manager in activities during the initial phases of a building's life cycle. Such an approach has multiple advantages, such as reducing the investment as well as the operational costs (Potkany, Vetrakova, & Babiakova, 2015). The tasks of the facility manager across the life cycle of a facility are listed in Figure 4.

Building life cycle phases

Investment objective, project preparation	Operational project design	Building Acquisition	Final building approval	Building use	Maintenance, servicing, reconstructions	Deconstruction
-counselling and procedure consultations, - defining of facility management requirements,	- consultancy and notes to individual project types, - design optimisation,	- continual control of task within individual projects,	- acceptance of building, operational documentation, - testing operation,	-coordination of support processes operation, - setting an effective and optimal building operation,	- calculation of effective use of building, -definition of maintenance, rebuilding and reconstruction requirements, - design of time frame schedules,	- deconstruction design, or calculation of further building effective use.

Tasks of a facility manager

Figure 4: An overview of facility managers' activities (Potkany, Vetrakova, & Babiakova, 2015)

The tasks of the facility manager in the exploitation phase of the building can be divided in two categories: 1) facility operations and 2) maintenance and repair (Roper & Payant, 2014). The former consists of areas such as inventory management, furniture installation, energy management and fire and life safety. In order to facilitate these areas, alterations management comes into play. Too often, partial drawings or schematics are used for small alterations or renovations. This practically guarantees that the information will not be updated in the database. Given that 30 per cent of all space is renovated annually in the average corporation, costly as-built drawings will be completely out of date within four years (Roper & Payant, 2014).

3.3 Software architecture in O&M

FM functions often require extensive information input from various fields and disciplines in order to adequately fulfil their purpose. Traditionally, FM information is organised and maintained in information systems such as Computerised Maintenance Management Systems (CMMS), Electronic Document Management Systems (EDMS), Building Automation Systems (BAS), etc.. The information that is required for such systems usually comes from different sources, is created and manipulated several times during the asset life cycle and is not synchronised between systems, which results in error-prone processes (Becerik-Gerber, Jazizadeh, Li, & Calis, 2012).

In his thesis, Bosch (2014) states that Dutch semi-public clients – government agencies acting as real estate clients, such as Rijkswaterstaat or educational organisations – use a variety of seven different information systems. This has led to the emergence of 'knowledge silos' that operate independently with their own systems. This causes the information to be extremely fragmented, stored across systems or across different servers and in different formats (spreadsheets or entered in databases).

3.4 Necessary information for O&M

During the design phase architects and other engineers preliminary produce graphical data, and the amount of information, captured in attribute data, increases over time. Most of the valuable O&M information is added during the construction in order to obtain an as-built situation (Verbaan, van der Voet, de Boer, Visser, & de Koe, 2014). The facility manager values attribute data more than the graphical data of the as-built model (Visser, de Boer, & van der Voet, 2013) This proportional shift in significance and addition of information over time is illustrated in Figure 5.

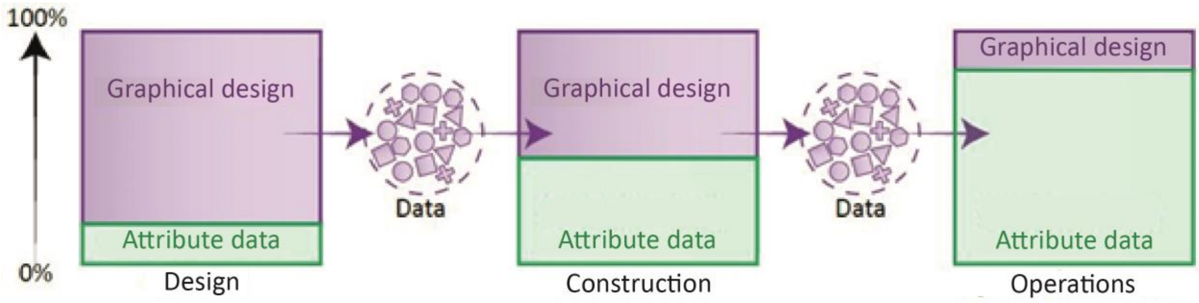


Figure 5: The significance, ratio and transfer of data per phase (Verbaan, van der Voet, de Boer, Visser, & de Koe, 2014)

The BIM protocol as set up by the national organisation promoting the interests of social housing associations in the Netherlands, Aedes, have listed a number of informational requirements a BIM should contain in order to be effective for housing associations. These include (Aedes Werkgroep BIM, 2016):

- Functions;
- Rooms with surface area;
- Geometrical objects;
- Inner walls;
- Installations;
- Ventilation elements;
- Safety systems;
- Materialisation and finishes;
- GIS-coordinates;
- Maintenance cycles;
- Tenant data;
- Prices.

4 BUILDING INFORMATION MODELLING (BIM)

The purpose of this section is to explore Building Information Modelling (BIM). Nowadays BIM seems to be a buzz word, and often used while talking about 3D models. However BIM is much more than simply just 3D models. Firstly a definition for BIM is provided, well as its benefits and potentials when implementing it in sections 4.1 and 4.2 respectively. Obstacles that can hinder BIM implementation are elaborated upon in section 4.3. Subsequently, section 4.4 explains the concept of 'Level of Development'. The application of BIM during the life cycle is investigated in section 4.5, after which section 4.6 elaborates on the concept of interoperability. Finally section 4.7 investigates Industry Foundation Classes, as well as other buildingSMART standards as Model View Definitions, Information Delivery Manuals and the buildingSMART Data Dictionary. The section concludes with classifications and the limitations of non-geometrical data in sections 4.8 and 4.9 respectively.

4.1 What is BIM?

In the last decade, BIM is becoming more and more adopted and is widely known in the Architecture, Engineering and Construction (AEC) industry. It is much more than only 3D models, and is considered a proven technology applied in a steadily growing number of projects in the industry. BIM has many definitions, due to the fact that many companies and/or projects apply BIM in different ways. In addition, scientific literature has yet to reach consensus regarding a single, widely-accepted definition for BIM (Eastman, Teicholtz, Sacks, & Liston, 2008) which has caused BIM to become a bit of an umbrella term recently. Therefore multiple definitions are discussed in order to get insight in the differences and to select the most complete one for use in this thesis.

Firstly it is wise to address the difference between Building Information Modelling, which describes the process of making the actual product which is the Building Information Model. The abbreviation BIM for both the process and the product are used interchangeably despite representing two essentially different parts in the new development in the AEC industry (Heinen, 2015). Grilo and Jardim-Concalves (2010) concur in that building information modelling should be seen as a dynamic process than a model per se.

The research of Love, Matthews, Simpson, Hill and Olatunji (2013) defines BIM as an emerging technology-focused methodology that can be used to improve the performance and productivity of an asset's design, construction, operation and maintenance process. This shows that BIM is not only applicable to the design and construction of buildings, but has its place during operation and maintenance as well. This is supported by the McGraw-Hill report called "The business value of BIM", a commonly referenced document by contractors, which defines BIM as the process of creating and using digital models for design, construction and/or the operations of projects (Young Jr., Jones, Bernstein, & Gudgel, 2009).

The National BIM Standard (NBIMS) defines BIM as 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 life cycle from inception onward (National Institute of Building Sciences, 2007).

BuildingSMART, an international acknowledged organization for knowledge exchange regarding BIM and the improvement of information exchange between software applications used in the AEC/FM industry. Their definition (BuildingSMART, 2012) seems to integrate the discussed definitions as follows:

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

This definition appears to be the most complete of the ones mentioned, and thus will be used throughout this thesis.

4.2 Benefits and potentials of BIM

Implementing BIM yields certain benefits. One of the biggest benefits is having early and accurate 3D visualisation of the building rather than it having to be generated from multiple 2D views. This increases insight in the building, given the fact that humans in general can derive overview of information from data better and faster if presented in a suitable visual format other than textual/numerical scripts or tables (Duncan & Humphreys, 1989). This phenomenon is also explained as the ability to large amounts of visual/diagrammatic information to be processed by the human visual perception system in parallel as opposed to the serial processing required for processing textual or numeric information (Ware, 2004; Larkin & Simon, 1987).

Through the use of a building model, the building is usable in early stages for producing cost estimates and checks against functional and sustainable requirements. Furthermore the usage of BIM enables earlier collaboration between disciplines, which shortens the design time and ensures insight in potential design problems in an earlier stage. This can be visualised through clash detection, which reduces errors during construction. By adding the dimension of construction planning to the model, making the BIM 4D, it is possible to simulate the construction process and show what the site would look like at any given point in the process, giving insight into potential opportunities to make the construction more efficient.

The BIM is able to provide accurate quantities from all objects in the model (e.g. area of walls, amount of doors, windows, etc.), as well as improved handover of facility information due to the information present in the model. Furthermore the model has the possibility to link maintenance and warranty information to the equipment and objects. In summary, the following benefits have been found (Eastman, Teicholtz, Sacks, & Liston, 2008; Volk, Stengel, & Schultmann, 2014):

- Visualisation;
- Clash detection;
- Analyses or simulations (daylight, fire, energy, structural, scheduling, etc.)
- Planning;
- Cost estimations;
- Quantity take-off;
- Improved collaboration;
- Interoperability between stakeholders.

4.3 Implementation obstacles

Despite the fact that benefits of using BIM are manifold, as described in the previous section, there remain a lot of barriers to achieve these benefits. Eadie, Browne, Odeyinka, McKeown and McNiff (2013) state that the top reason for not using BIM on projects is lack of expertise within the project team and/or organisation. Investment costs in training and software are therefore among the greater reasons that hinder the adoption of BIM.

Despite BIM being widely used in the AEC industry and growing attention towards facility owners and FM and O&M managers, a full correct implementation of BIM in the whole life cycle of the facility – or even the AEC industry alone – is relatively scarce. The table below list obstacles that hinder BIM implementation under four major categories found in reviewed literature (Eadie, Browne, Odeyinka, McKeown, & McNiff, 2013; Volk, Stengel, & Schultmann, 2014; Liu, van Nederveen, & Hertogh, 2016; Muz, 2014).

Table 1: Obstacles for implementing BIM (based on (Muz, 2014) and (Liu, van Nederveen, & Hertogh, 2016))

People	Process
• Fear of change and increased liability; • Poor understanding of the added value of BIM; • Inadequate BIM skills; • Reluctance to share information; • Conflicting obligations;	• Investment cost in new software and extra modelling effort too high; • No standardized processes and uniform categorisation; • Lack of immediate benefits; • Trust is still lacking; • Different requirements; • Costs of translating 2D drawings to 3D models often exceeds potential gains;
Technology	**Policy**
• Software functionality (liability); • Data safety; • Poor data interoperability (import/export of data); • Technology immaturity;	• Data and model ownership not yet clearly defined; • Limited use of relevant policies and contractual frameworks allowing for BIM-based project delivery;

The obstacles listed in Table 1 can be overcome, although some will take longer than others. Although the people-factor is potentially time-consuming to overcome, the other three factors can be overcome with research and pilot projects.

4.4 Level of Development

During the life cycle of buildings, various design, engineering, construction, maintenance and demolition functionalities, for example quantity take-off, clash detection or 4D planning, and possible applications such as Revit, Allplan or Solibri all require a different capability of BIM. Such functionalities are usually related to the concepts of 3D, 4D, 5D or even 6D BIM. The concept of 4D requires the connection of functional planning information to the building or space elements of the 3D model, while 5D is accomplished by adding the cost dimension. This allows systems to use this data in order to calculate cost analyses.

Volk et al. (2014) defines this as Level of Development (LOD), or the geometrical and non-geometrical attribute information provided by a model component, often referenced to a point of time, life cycle stage or a contractual responsibility. Generally the LOD values range from LOD100 to LOD500, where LOD100 contains only spatial objects or masses, and LOD500 represents the as-built situation. The national organisation promoting the interests of social housing associations in the Netherlands, Aedes, forms a good example by already taking into account Level of Development in their BIM protocol. They do not (yet) use values, but do take into account guidelines such as only modelling what is visually traceable (which e.g. means no foundations) and using a single type of material for each individual BIM component (Aedes Werkgroep BIM, 2016).

4.5 BIM applications during life cycle

BIM is applicable during all phases in a building's life cycle, which can be deconstructed in project inception, feasibility, design, construction, handover, operations, maintenance and eventual demolition (Eadie, Browne, Odeyinka, McKeown, & McNiff, 2013).

Eadie et al. (2013) have conducted a survey on the use of BIM across the project life cycle, of which the results have been adapted to Table 2. This survey was conducted among 90 professionals in the UK construction industry. These numbers are not expected to differ significantly in the Dutch industry.

Table 2: BIM use during the construction stages (Eadie, Browne, Odeyinka, McKeown, & McNiff, 2013)

Use during construction project stage	Often (%)	Occasionally (%)	Never (%)
Feasibility	26.9	52.6	20.5
Design	54.9	42.7	2.4
Preconstruction (detail design & tender)	51.9	39.2	8.9
Construction	34.7	52.0	13.3
Operations & Maintenance	8.8	45.6	45.6

Table 2 shows that BIM usage is most common in the design- and preconstruction phase both with over half of the respondents indicating that BIM is often used during this phase. Application during construction and feasibility studies are less common with 35% and 25% respectively, while often BIM use during O&M stages are rare with only 8.8%.

4.6 Interoperability

Interoperability is defined as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” (Institute of Electrical and Electronics Engineers, 1990). The goal of full interoperability in the AEC and FM sector is far from being realized (Grilo & Jardim-Goncalves, 2010).

Eastman et al. (2008) define interoperability for BIM as the need to pass data between applications and for multiple applications to jointly contribute the work at hand. This need is crucial, given that there are over 150 software packages supporting BIM nowadays (Visser, de Boer, & van der Voet, 2013). All of these types of software have various BIM applications and are used in different stages in a building’s life cycle. A standard file format should ensure optimal cooperation between these software packages and thereby enhance cooperation between AEC and FM departments.

This need can be defined in a bi-directional sense. On the one hand, there is a need for a standard, interoperable file format that is compatible with different types of software tools (Laakso & Kiviniemi, 2012). On the other hand, it is necessary that the software can translate a model into this interoperable file format in such a way that all of the object’s information can be transferred correctly. In most cases it proves to be a challenge for such translations to retain all of the information that the model contained in its original native file format (Grilo & Jardim-Goncalves, 2010).

4.6.1 Open standards vs. proprietary standards

An open standard is a standard with easily accessible documentation, no intellectual property rights, open participation procedures, independence and sustainability of the standardisation organisation (BIM Loket, 2015). Proprietary standards are the opposite of this, given that they are not standardised in an open process. These are mainly developed by the larger modelling software companies and quite interoperable among themselves, but not in relation to other vendors’ applications (Grilo & Jardim-Goncalves, 2010).

There are many theoretical benefits related to interoperability using an open standard in the application of BIM in the AEC and FM industry. Without such an open standard, each individual software application must develop direct translators for all other software applications to be able to communicate back and forth (Laakso & Kiviniemi, 2012). If, instead, an open standard is used, applications only need to be compatible with that standard in order to be able to communicate with all other applications that support this standard as well. Figure 6 visualises this principle.

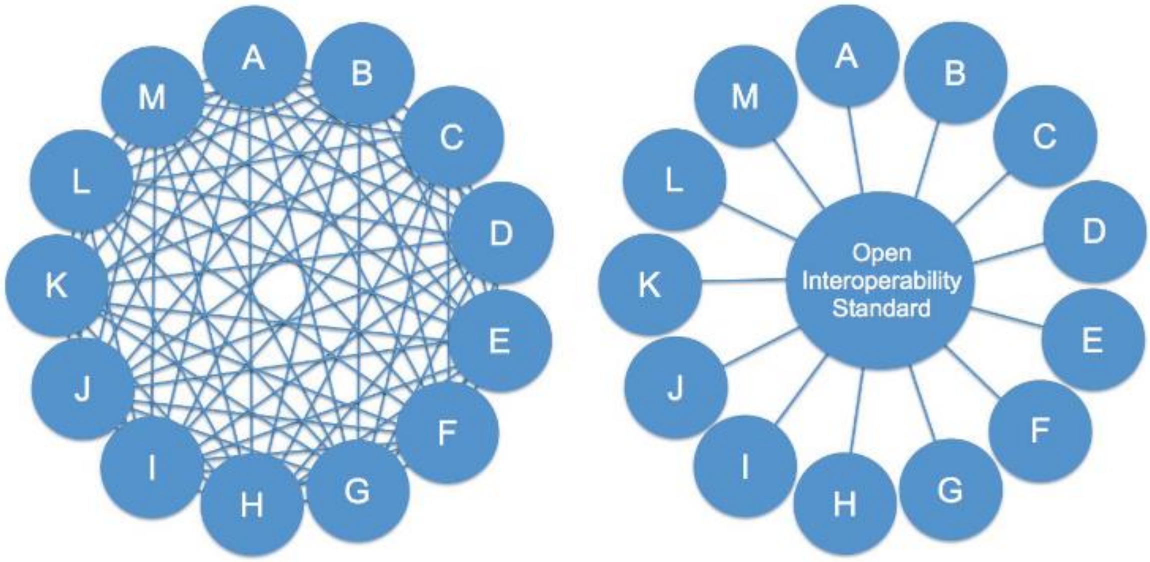


Figure 6: Direct communication between software vs. using an open standard (Laakso & Kiviniemi, 2012)

Many publications share and support the value of an open standard (National Institute of Building Sciences, 2007; Vanlande, Nicolle, & Cruz, 2008; Visser, de Boer, & van der Voet, 2013; Eastman, Teicholtz, Sacks, & Liston, 2008). However, the exchange of BIM data in practice nowadays is still dominated by proprietary solutions, meaning that many construction projects are based on a solution using software from the same vendor (Laakso & Kiviniemi, 2012).

Working with these open standards is the focal point of the concept called 'open BIM'. Open BIM is the concept of having all relevant building information in open formats, making them accessible and readable for everyone and not locked in proprietary software formats (Hallberg & Tarandi, 2010). It is an initiative of buildingSMART and several leading software vendors using the open IFC data model (BuildingSMART, 2012). Among the various data model formats, IFC is the only public, non-proprietary and well-developed data model for buildings and architecture existing today (Eastman, Teicholtz, Sacks, & Liston, 2008). The next section elaborates upon IFC.

4.7 Industry Foundation Classes

Industry Foundation Classes (IFC) is an open standard which enables the exchange of building data throughout the complete facility life cycle. It is an open and standardized data model with the intent of enabling interoperability between BIM software applications (Autodesk, Tekla, Bentley, etc.) in the AEC and FM industry (Laakso & Kiviniemi, 2012). IFC is developed and maintained by the non-profit organisation buildingSMART – formerly known as the International Alliance for Interoperability (IAI) – and their ongoing efforts have resulted in a standard that is increasingly accepted in the industry. Development on the first version of the IFC formally initiated in 1994, and nowadays the latest version is IFC4 which is defined in the ISO 16739 standard.

IFC models not only contain geometry of objects, but include metadata that relates to other objects of the building as well. For example, an IFC door is not just a simple collection of lines and geometrical primitives recognized as a door, but it is an “intelligent” object door which has a door’s attributes linked to a geometrical definition (Vanlande, Nicolle, & Cruz, 2008). Such attributes range from price, manufacturer to thermal and acoustic performances, etc.. This makes IFC models semantically rich.

The IFC data model is written according to the EXPRESS data definition language and relies on the physical file format STEP. A STEP file consists of two sections; a header section with information regarding the file itself, and a data section describing entity instances. All EXPRESS objects are called entities, and it defines twenty-six base entities such as materials, geometry and properties. These base entities are then composed to define commonly used building elements (e.g. generic walls, floors, building service elements, etc.). Because IFC is defined as an extensible, and object-oriented, data model, the base entities can be elaborated and specialized if need be (Eastman, Teicholtz, Sacks, & Liston, 2008).

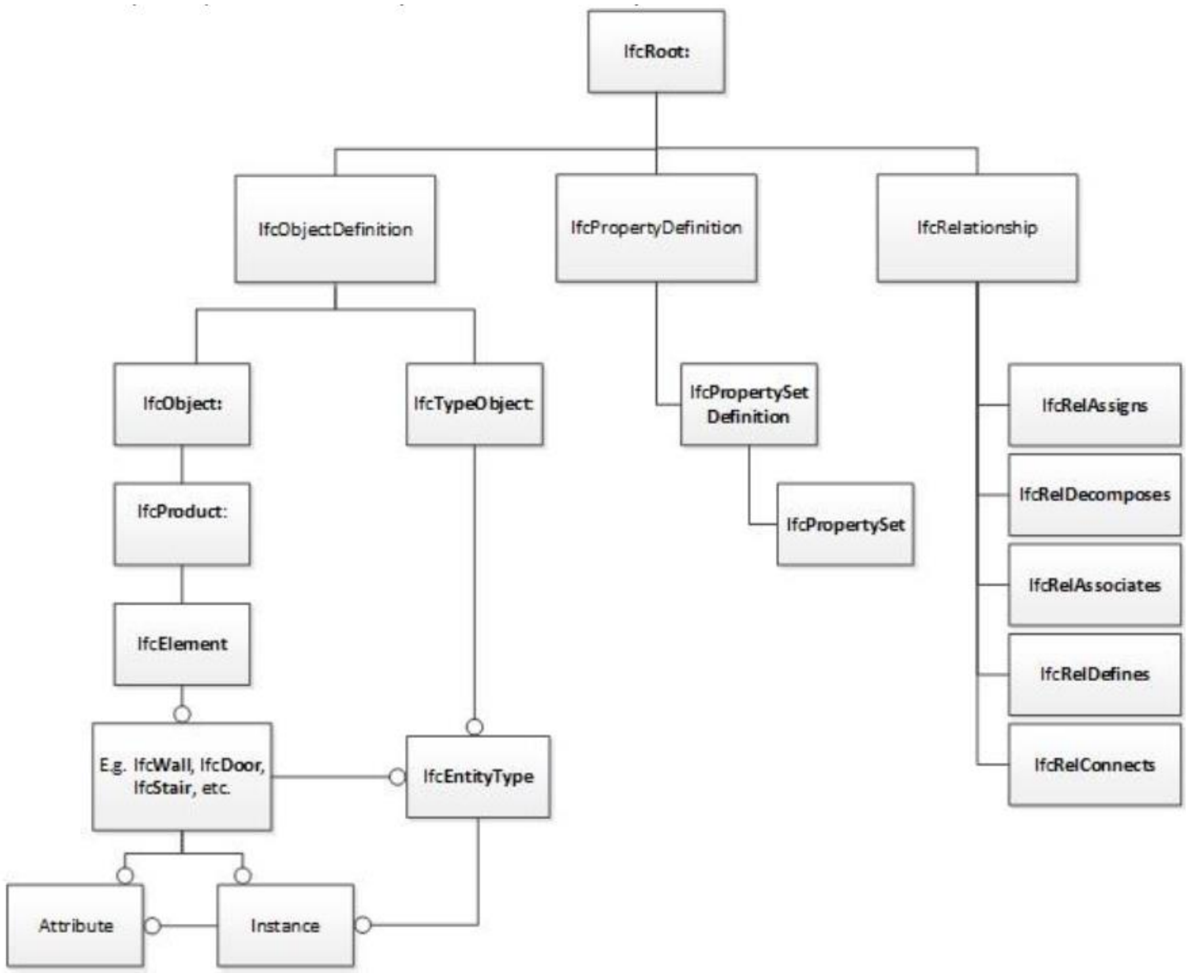


Figure 7: Simplified IFC2x3 Schema (Heinen, 2015)

The entities in an IFC file are related through an hierarchical inheritance structure. Figure 7 below visualises the hierarchy between objects in an entity tree. Each level of this tree introduces different attributes and relations to a certain entity. *IfcRoot* assigns a Global Unique Identifier (GUID) and other identifier information. *IfcProduct* defines the location of the object, for example a wall, and its shape, and *IfcElement* carries the relationship of this wall with other elements, such as wall bounding relationships and spaces that the wall separates (Eastman, Teicholtz, Sacks, & Liston, 2008). Many of these attributes and relations are optional, which allows the user to exclude attributes from the export to IFC.

In addition to the IFC standard alone, buildingSMART extends the scope of IFC-based exchanges beyond the IFC data model alone. Figure 8 shows the three buildingSMART standards for:

1. BIM models, namely the *Industry Foundation Classes*;
2. BIM processes, namely the *Information Delivery Manual*;
3. BIM object libraries, namely the *International Framework for Dictionaries* (renamed to buildingSMART Data Dictionary).

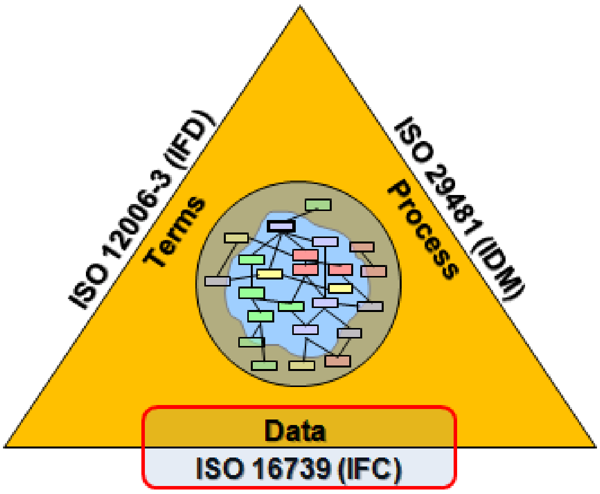


Figure 8: buildingSMART standards (buildingSMART, n.d.)

These standards are elaborated upon in following sections.

4.7.1 Information Delivery Manual (IDM)

The Information Delivery Manual (IDM) framework defines the exchange of process information in BIM through process maps, interaction maps and associated exchange requirements (Volk, Stengel, & Schultmann, 2014). IDM has been recognized as the key feature that makes IFCs work (Grilo & Jardim-Goncalves, 2010). IDM provides a framework to structure the information need for specific tasks in a building's life cycle, which main purpose is to ensure that the relevant data are communicated in such a way that they can be interpreted by software at the receiving end (buildingSMART, 2011).

4.7.2 Model View Definition (MVD)

Model View Definitions (MVD) denote subsets of the IFC schema. More often than not, the entire IFC schema is not necessary. Therefore MVDs only provide the information that is relevant for a specific AEC or FM department or task. It acts like a filter of the IFC data schema in order to only obtain the required information for a specific purpose. For the purpose of O&M, Construction Operation Building information exchange (COBie) is the predominant, vendor-neutral, international, standard MVD to exchange contact and general facility information as well as information about spaces, floors, components, technical systems and equipment (Volk, Stengel, & Schultmann, 2014).

COBie data is available in two main formats depending on the user and building stage. The data can be used in the IFC-format, which is used for information exchanges between machines during the design process. On the other hand, for human reading the engineering data is usually translated into a spreadsheet (e.g. Excel).

4.7.3 buildingSMART Data Dictionary (bSDD)

The buildingSMART Data Dictionary (bSDD), which has been renamed from ‘International Framework for Dictionaries’ (IFD), is a terminology standard for BIM libraries and ontologies, and certified in ISO 12006-3 (Volk, Stengel, & Schultmann, 2014). The idea of the bSDD is to create a central dictionary that overcomes implementation problems by defining objects frequently used in the AEC and FM industry, thereby linking existing databases to a BIM. This vocabulary currently contains around 80,000 concepts along with approximately 200,000 natural language names and descriptions (Beetz, et al., 2014).

The bSDD separates names and languages from the actual concept. Such a concept is unique in the bSDD, however a concept may have multiple names. In a world with many different languages, exchanging information between each other is naturally susceptive to misinterpretation and ambiguity. An example of this ambiguity is illustrated through the Norwegian word “dør”, which translates to “door” in English. However, “dør” refers to the door within its frame, while “door” in English only refers to the door itself. Therefore when translating “dør” it should be “door set”.

This example shows a critical problem which must be overcome for BIM-based object-oriented working methods. To solve this problem, the bSDD assigns Globally Unique IDs (GUID) to concepts. When exchanging concepts, names and words may still be added, making them easier to read for users. The added value is that a party can label their desired name to a certain concept, and when exchanging the other party can do the same, while still ensuring that they are both talking about the same concept. Although not every stakeholder in every life cycle phase of the building will use every piece of information described, they still create a unique and identifiable object. This is visualised in Figure 9.

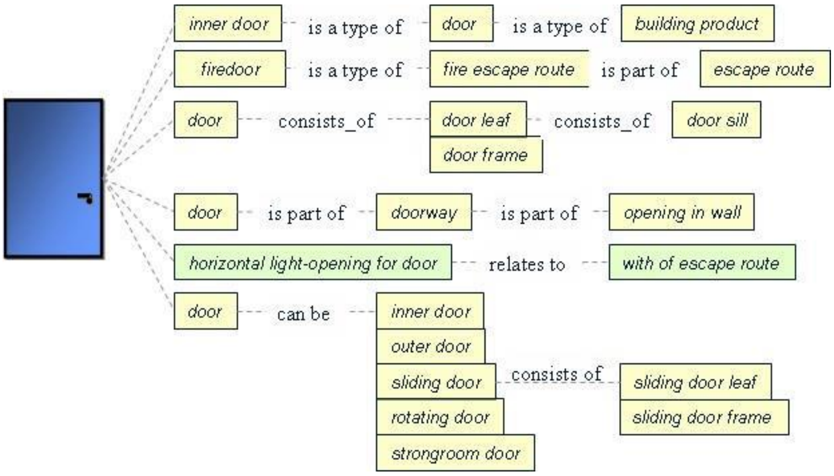


Figure 9: Relationships of a single concept (buildingSMART, 2008)

Depending on the organisation, project or use case, an object view can be instantiated showing only the desired bSDD context. Figure 10 shows the different contexts that make up all characteristics of a concept, in this example a window. It must be noted that the bSDD does not describe instances, which is done in file format (IFC). The bSDD holds templates of the concepts, while standards like IFC fill these templates.

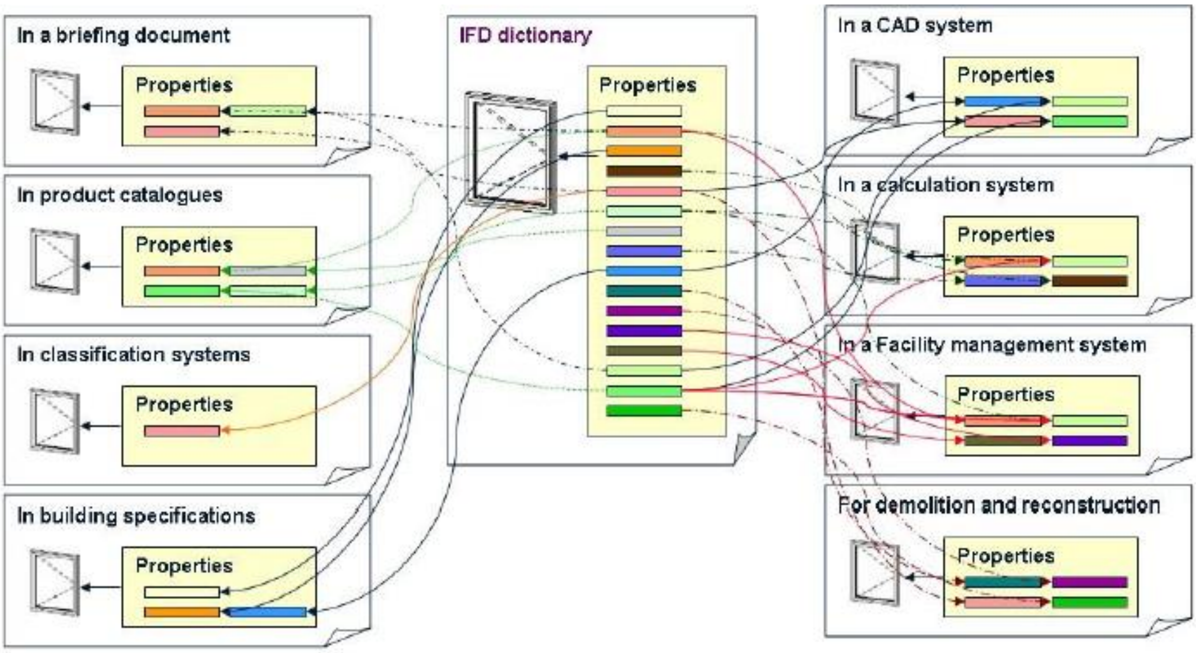


Figure 10: Concepts and its different contexts (buildingSMART, 2008)

4.8 Classifications

As the word itself already indicates, classifications are used to classify a collection of objects. They group data according to their similarity while also providing a hierarchical structure to that data. Classifications increase communication by using the same structure in more projects, thereby increasing the speed of interpreting the information. In addition to naming the classes and objects, a classification also applies a code to create an organized structure.

The purpose of the classification determines its structure, which is why many different classifications are currently in use.

Semantically annotating 3D models with non-geometrical information improves searches over this information. For instance, using a good classification enables model queries without being hindered by ambiguities in the information such as “door”, “door set” or “door leaf” because these belong to a similar class. In addition, implementing standard classifications within a BIM model allows sorting of similar information and retrieval of that information across multiple platforms and at any stage in the building’s life cycle. A number of frequently used classification systems in the Netherlands are:

- **NL/SfB**: this classification originates from the Swedish SfB-system, published in 1947, and consists of five parts (STABU, n.d.). The most known and most used is Table 1, which groups construction elements based on their functions, e.g. “inner walls” and “ceiling finishes”;
- **ETIM**: the European Technical Information Model (ETIM) lists the most important technical characteristics of product classes in order to find and describe them. ETIM finds its application mainly in the installation, construction and maritime sectors (ETIM, n.d.);
- **NEN 2767**: this classification originates from the Netherlands Standardization Institute (NEN) and is used for the standardisation of inspection of the technical state of construction components. This classification is applied to building objects as well as infrastructure;
- **STABU Bouwbreed**: this classification structures construction elements based on different work orders and is used for the technical description of entire building systems and installations. This system contains six modules and is thereby applicable for each life cycle phase of a building (STABU, n.d.).

4.9 Limitations of information exchange and versioning

In the application area of O&M, there is an increasing need to identify and capture non-geometrical information in order to secure a successful implementation of BIM. Nowadays buildings produce more data than ever before, from energy usage, utility information, occupancy patterns, weather data, etc.. In order to manage a building, it is important to use knowledge from across these various information sources.

Semantic models are models in which semantic information is included, describing the meaning of its instances. As such, IFC models are considered to be semantic models. However, it appears that approximately 90% of IFC models consist of geometrical entities, while 90% of the attributes are optional, almost rendering it just another exchange standard for geometrical data (such as CAD) (Beetz, 2009). In addition, many barriers exist to the interoperability of information sources and there is only little interaction between these ‘islands of information’ (Curry, et al., 2013).

In addition, tracking changes in IFC-based information is difficult. Ideally to capture changes, IFC files use the concept of object versioning. As proposed by Nour and Beucke (2010), object versioning enables having several versions of the content (attribute values) of an object. This means that a new object version is only created when the object's attribute values change, otherwise the same object version is shared among several model versions. Figure 11 shows this principle, where *IfcSite, IfcBuilding* and *IfcBuildingStorey* are shared between versions whereas each model version has a different version of *IfcWallStandardCase.*

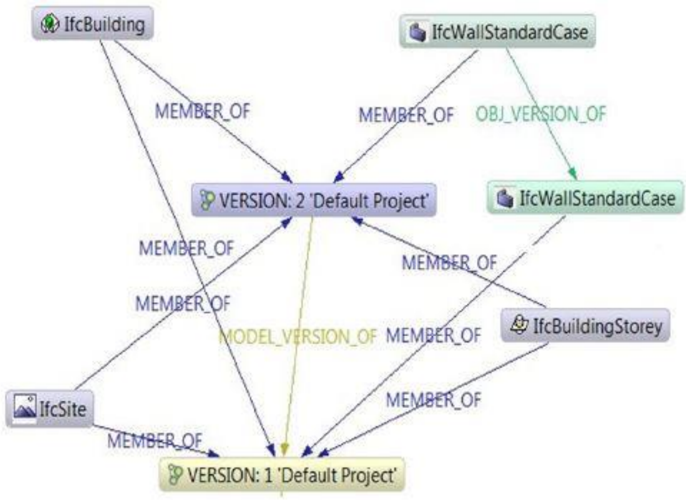


Figure 11: A simple IFC object versioning graph (Nour & Beucke, 2010)

Currently, model difference is tracked mainly using GUIDs, i.e. objects of the same GUIDs in the different models are compared. Incorrect management of these GUIDs, or operations that delete an object and create a new one in its place will render this process ineffective (Solihin, Eastman, & Lee, 2016; Nour & Beucke, 2010). Furthermore only a small minority of the instances captured in IFC is identified with GUIDs which is an attribute that is inherited from the *IfcRoot* entity as mentioned in section 4.7. However, most of the instances are not identified in this manner. For example, the geometric representations of the actual objects are more difficult to identify because these instances do not descend from the *IfcRoot* entity but from the *IfcGeometryResource* schema. These instances are only identified with line numbers of instances. Unfortunately though, these numbers are almost always different between versions (Oraskari & Törmä, 2015).

A promising approach in the AEC and FM industry to tackle the these problems in the exchange of non-geometrical information is called Linked Data, initiated by the W3C. The fundamental concept of Linked Data is that data is expressed in a more formal sense with the mind-set that it will be shared and reused by others. It is based on the so-called Semantic Web technologies for the representation, sharing and querying of structural data on the Web (Curry, et al., 2013). In general it uses languages such as OWL and RDF, which contrast with EXPRESS. Therefore these languages, as well as an elaborated view on the Semantic Web and Linked Data as a whole, are provided in the following section.

5 LINKED DATA

While BIM models are defined to a very high degree during design and construction of a building, they do not necessarily represent the 'as-built' situation. The actual as-built conditions can differ from the design significantly. In addition, the as-used building can change substantially throughout a facility's life cycle. Given this, the as-designed BIM models have to be altered and elaborated in order to create actual as-built models. Additionally, in the O&M phase building owners have a wealth of external data available which is to some extent related to (one of) their building(s). The participants in the building process use and produce much more information than containable inside a BIM, e.g. cost estimation, planning or product specifications (Dankers, van Geel, & Segers, 2014). Missing, obsolete or unstructured information results in ineffective project management, uncertain process results and time loss or cost increases in maintenance, retrofit or remediation processes (Volk, Stengel, & Schultmann, 2014).

Given the former, there appears to be a need for data to be integrated in a semantic sense, which is produced and used in a fundamentally different manner. Therefore, the purpose of this section is to explore the field of Linked Data as a means of solution. Due to the fact that this technology is still in its infancy within the AEC and certainly the FM industry, the origins of Linked Data are touched upon briefly in section 5.1. Afterwards, the Semantic Web approach is defined as well as the standards that create and link data across the web will be elaborated upon in sections 5.2 and 5.3. Section 5.4 looks into the standard query language for Linked Data; SPARQL. Finally, the chapter concludes with provenance and reification in section 5.5.

5.1 Semantic Web

The idea behind the Semantic Web dates back to 2001, which the World Wide Web Consortium (W3C) envisioned as the next step of networked information adding semantic metadata to textual information (Berners-Lee, Hendler, & Lassila, 2001). This way, data content and its relations can be described in a way that its meaning can be processed by machines as well as people. Thus the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise and community boundaries. Figure 12 visualises this linkage of data sets on the Semantic Web. The need has increased for shared semantics and a web of data and information derived from that data (Berners-Lee, Shadbolt, & Hall, 2006). A number of industries have recognized the value in semantic web applications, among which the AEC and FM industry.

Using these semantic web technologies, it becomes possible to connect BIM models to external information from other ecosystems, such as product catalogues, libraries of design elements, public procurement requirements, etc. over the internet (Costa & Madrazo, 2015). In doing so, building data such as BIMs are reusable in- as well as outside the AEC and FM domain and provides greater value across organisations (Curry, et al., 2013).

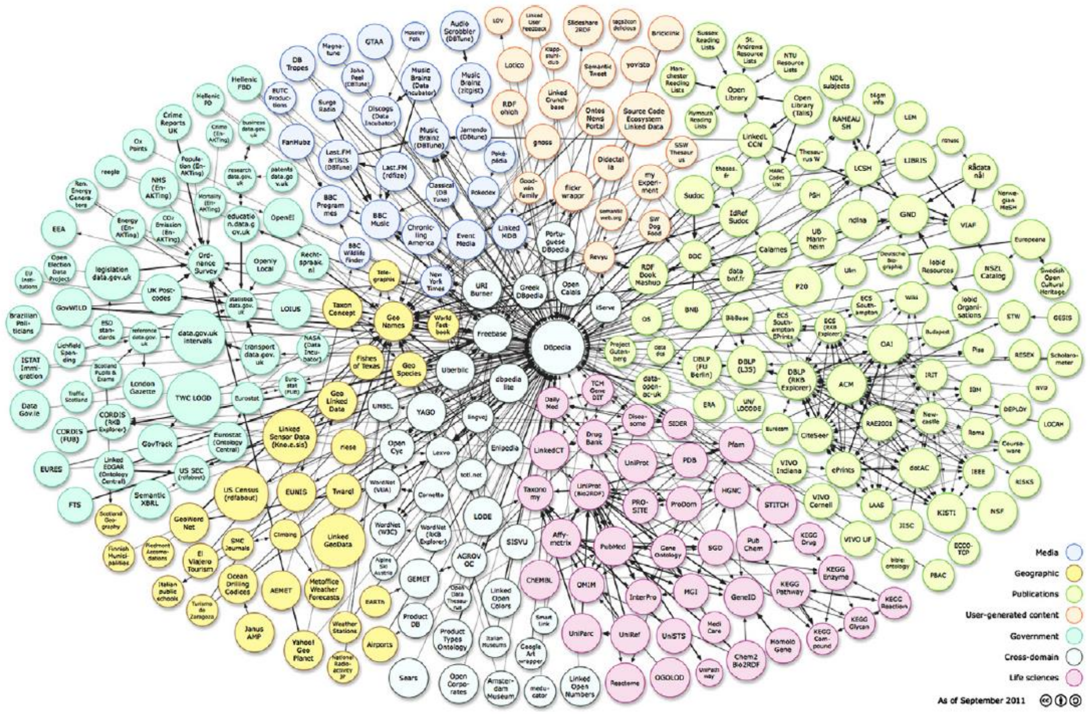


Figure 12: The linked open data cloud on the Semantic Web (Curry, et al., 2013)

Linked Open Data is Linked Data which has been released under an open license, thus not limiting free usage of this data. Many publications cover the five-star Linked Open Data model, which denotes the extent to which certain data is integrated to the Web (Berners-Lee, 2006). This model contains the following levels:

★ Make the data available on the web in any format (e.g. pdf, scans etc.);

★★ Make the data available as structured, machine-readable data (e.g. Excel);

★★★ Make the data available in a non-proprietary structured format (e.g. CSV);

★★★★ In addition to (3), use open standards to identify things (e.g. RDF);

★★★★★ Link the data to other data to provide context as Linked Open Data (LOD).

The semantic web stack illustrates the hierarchy of languages where each layer uses parts of the layer below, as seen in Figure 13. These standardized languages enable linking of concepts in different data sources, therefore connecting these sources. This is the essence of Linked Data. In essence, the term Linked Data refers to a set of best practices for the publishing and interlinking of structured data on the Web, the so called Linked Data principles. These are the following (Curry, et al., 2013):

1. Use Uniform Resource Identifiers (URIs) to identify things;
   A Uniform Resource Identifier is a string of characters used to identify a certain concept. Examples are buildings or walls, but also more abstract concepts like relations or groups of rooms.

2. Use URIs according to the HTTP standard;
   HTTP URIs enables the URI to be globally unique, along with a simple, well-understood retrieval mechanism. This way, people from across the globe are able to use these URIs to identify things which can then be referenced over the standard HTTP web protocol into a description of this identified concept.

3. Provide useful RDF metadata;
   This principle advocates the use of the standardized languages found in the Semantic Web stack – see Figure 13 – to publish structured data, for example RDF.

4. Include links to other data sets.
   The final, and most crucial principle of Linked Data promotes the use of hyperlinks in order to connect to concepts from other data sets.

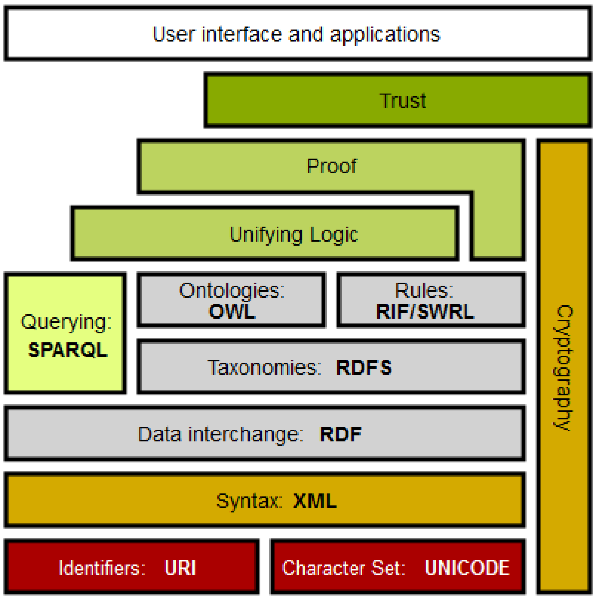


Figure 13: The Semantic Web stack (based on (Obitko, 2007))

Linked Data is published on the Web using the RDF data model to represent the data. However, RDF does not provide any domain-specific terms for describing the formal hierarchy of thins and relations among each other. This is done by ontologies. Therefore in order to obtain a better understanding of the Semantic Web, Linked Data and current developments in the AEC industry, the topics of ontologies, RDF, IfcOWL and SPARQL are further explored.

5.2 Resource Description Framework (RDF)

The basic machine-readable representational format used on the Semantic Web in order to represent information is the Resource Description Framework (RDF). RDF is a general method for encoding graph-based data that does not follow a predictable structure (Curry, et al., 2013). Describing data with the use of resources in RDF enables the connection between various data sets. In RDF data is described through the use of triple statements using various unique resources. These resources are defined with URIs. Triples are RDF expressions composed of subjects, predicates and objects. The subject denotes the resource, and the predicate denotes attributes of the resource as well as the relation between the subject and the object. An example of this logic is shown in Figure 14.

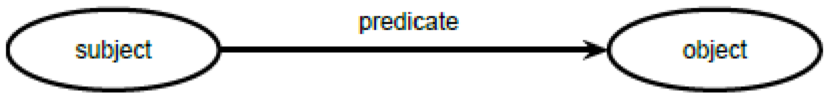


Figure 14: Visual representation of an RDF triple with subject, predicate and object

Figure 15 shows an example of an RDF graph that contains four triples with information about a person. Firstly, it denotes that the graph contains information about a person, thereby identifying the type of concept. The person is then given a name, in this case 'Eric Miller'. Thirdly, the 'mailbox' property is indicated. Finally, the 'personal title' is indicated, which in this case is 'Dr.'. The resources in this graph – 'me' and 'person – are represented with actual URIs, as well all of the predicates.

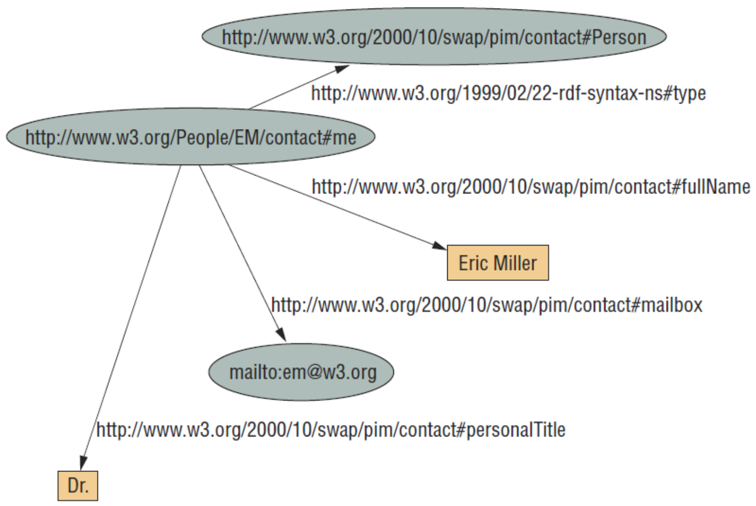


Figure 15: An RDF graph representing Eric Miller (Berners-Lee, Shadbolt, & Hall, 2006)

As mentioned, URIs have a global scope. This means when a resource is denoted with the use of an URI, anyone can link to it, refer to it, or retrieve a representation of it (Berners-Lee, Shadbolt, & Hall, 2006). RDF statements are diagrammed as a directed graph, representing facts. This way, by defining explicit links in these statements as unambiguous references that may refer to data that is specified in other graphs, it becomes possible to create a network of linked data available for any application (Costa & Madrazo, 2015).

The RDF graph of Figure 15 can be represented using various syntaxes. The most commonly used ones are RDF/XML, Turtle and N-Triples (Allemang & Hendler, 2011). The most often used syntax for machine-interpretable graphs is RDF/XML, though the turtle-notation is considered to be the most human-readable.

5.3 Ontologies

Ontologies define the concepts and relationships used to describe and represent an area of concern. Ontologies – or vocabularies – are used to classify the concepts that can be used in a particular application, characterise possible relationships and define possible constraints on using those concepts. There is no clear distinction between what is referred to as "ontologies" and "vocabularies". In general, ontology is used for more complex or formal collections of terms, while vocabulary is used when such strict formalism is not necessarily used (W3C, 2015). In practice, ontologies range from very complex (e.g. several thousands of concepts) to very simple (e.g. only one or two concepts) (W3C, 2015). The most used language to describe ontologies in a formal way is the Web Language Ontology (OWL), which provides a formal and explicit specification of a shared conceptualization that is defined by means of classes, attributes, values, relationships, roles and rules.

An example is the construct "`owl:sameAs`" construct, which is used to merge data from multiple sources. The use of this construct states that different resources actually represent the same real world entity. If such resources are determined to be the same, information with regards to those resources can be merged. Such a construct in turtle-syntax is the following:

```
ex:WallA owl:sameAs ex:WallE
```

OWL uses the Resource Description Framework Schema (RDFS) concepts as subset to define ontologies. In addition, RDFS took the basic RDF specification and extended it with a vocabulary to define classes, hierarchy and properties in order to provide a minimal ontology representation language that the research community has adopted fairly widely (Berners-Lee, Shadbolt, & Hall, 2006).

5.3.1 IfcOWL

As mentioned before, IFC models are semantically rich because they do not only contain information with regards to three-dimensional geometry but also contain metadata about the objects in relation to other aspects of this object and to the building as a whole. Although the IFC standard permits the representation of many domains of the AEC industry, the lack of mechanisms to extend the semantics of the model remains one of its major limitations (Costa & Madrazo, 2015).

In order to overcome these limitations, Beetz, van Leeuwen and de Vries (2009) have translated the EXPRESS language definition of IFC into an OWL-based notation: IfcOWL (buildingSMART, n.d.). This OWL version of the IFC schema uses Semantic Web technologies for BIM models, and opposed to the current IFC definitions, enables the extension of BIM models with new concepts and properties that facilitate its interpretation by applications (Beetz, 2009). In addition, this type of data formalization has the advantage of extracting partial models by using graph query languages, such as SPARQL (Costa & Madrazo, 2015).

5.4 SPARQL

The primary query language for RDF graphs is SPARQL, or the SPARQL Protocol And RDF Query Language, which is able to retrieve and manipulate subgraph information. SPARQL can be deployed across various data sources. The SPARQL query language relates closely with the RDF structure itself, due to the fact that the key element of a SPARQL query is the graph pattern. Such a pattern is essentially a smaller graph which includes both resources and variables, either known or unknown, that specifies what information needs to be retrieved from the RDF graph. SPARQL query patterns are represented in a variant of Turtle, one of the syntaxes commonly used for representing RDF as well (Allemang & Hendler, 2011). SPARQL has four distinct query forms; SELECT, CONSTRUCT, ASK and DESCRIBE. Each query form attempts to use solutions from pattern matching to form result sets or result RDF graphs (W3C, 2008). An example query is shown graphically in Figure 16:

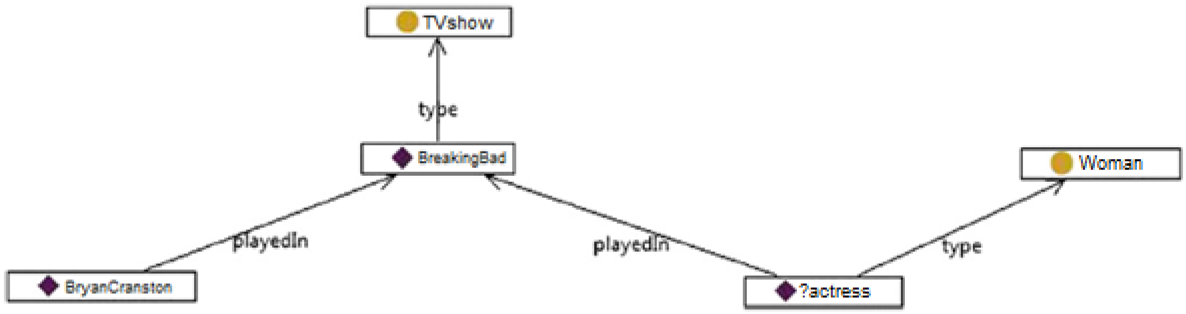


Figure 16: Example of a query pattern (based on (Allemang & Hendler, 2011))

This query pattern translates to the following SPARQL query:

```
SELECT  ?actress
WHERE { :BryanCranston :playedIn :BreakingBad .
        :BreakingBad rdf:Type :TVshow .
        ?actress :playedIn :BreakingBad .
        ?actress rdf:Type :Woman . }
```

This query asks which actress has played in the TV show *Breaking Bad* alongside its lead actor, Bryan Cranston. The line "*SELECT ?actress*" represents which piece of data is to be retrieved from the graph. The "*WHERE*" clause specifies the graph pattern that defines *?Actress* as subject of a type "woman" and defines that she has also played in a "TVshow" called *Breaking Bad.* This query, according to the International Movie Database, has two hits: Anna Gunn and Betsy Brandt (IMDb, n.d.).

5.5 Provenance and reification

As mentioned in section 2.1, provenance information captures the origins of specific data and processes that this data underwent (Petrinja, Stankovski, & Turk, 2007). In many cases new information which has come to light about concepts in the data are not recorded. Lack of provenance metadata and temporal annotations makes it difficult to understand how data sets develop with regard to the real world entities they describe (Mynarz, 2013). Zhao et al. (2010) have identified a number of requirements for provenance in order to be valuable in the versioning process as well as verification. These requirements are:

1. **Identity**: to be able to refer to the artefact for which provenance is described;
2. **Evolution**: to be able to describe provenance of evolving resources that change over time and to describe how new incarnations of a resource relate to each other;
3. **Entailment**: to distinguish what is directly stated by the entities and processes that produce the resource from other information that might be inferred from these statements;
4. **Publication**: to use some provenance representation language and link provenance statements to actual resource information;
5. **Querying**: to formulate and execute queries to find provenance for a given resource.

The IFC specification contains provenance information about the author of each entity included in the file, the application used for creating the file and the time that each entity was created or possibly modified. These data are captured in the following IFC entities; IfcOwnerHistory, IfcPersonAndOrganization, IfcPerson, IfcOrganization, IfcApplication, IfcAuditTrail and IfcTransaction (Petrinja, Stankovski, & Turk, 2007). Because among others, the IfcOwnerHistory is attached to the root entity, which most entities have as the topmost ancestor, provenance data can thus be tracked about the creation, modification and deletion of the entities in an IFC file.

The official state of the art concerning the representation of RDF together with metadata is reification (Zhao, Bizer, Gil, Missier, & Sahoo, 2010). Through reification, RDF statements are described using RDF in order to capture information about when statements are made, who made them or other similar information (W3C, 2007). The RDF language provides a built-in vocabulary that is intended for the description of RDF statements, consisting of the type `rdf:Statement`, and the properties `rdf:subject`, `rdf:predicate` and `rdf:object`. Using these for the reification of a statement, a URI is assigned to a statement so other statements are able to describe it.

Various alternative approaches to RDF reification have been published over the years. Mynarz (2013) provides an overview discussing the efficiency and successfulness of a wide range of provenance tracking approaches. Most of this literature focuses on two specific types: named graphs and quadruples. Quadruples extend RDF triples with a fourth element that is either a URI, a blank node or an identifier (Flouris, Fundulaki, Pediaditis, Theoharis, & Christophides, 2009). Flouris et al. (2009) use a form of quadruples where they use colours as a fourth element of triples to represent the source from which the triple was obtained. Named graphs, on the other hand, are sets of triples to which a URI is assigned, thus can be referenced by other graphs as a normal resource. This way, explicit provenance can be assigned to this collection of triples.

Using named graphs and quadruples, changes in data sets can be captured. An extension on the reification vocabulary already built into the RDF language, the ChangeSet vocabulary introduces the notion of a changeset, which encapsulates the delta between two versions of a resource description. The delta is represented by two sets of triples: additions and removals (Tunnicliffe & Davis, 2005). In addition, the vocabulary includes terms for describing metadata on the change, such as the creation date and preceding changeset.

The DURAARK (Durable Architectural Knowledge) project is funded by the European Commission with the aim of developing methods and tools for the long-term preservation of architectural knowledge and data. Their approach allows the combination of STEP-based IFC models with RDF data without raising issues regarding compatibility with existing commercial tools (Beetz, et al., 2014). In other words, the IFC model remains valid and can still be read by available IFC viewers. For the versioning of Linked Data sets the project uses named graphs as well, which cluster RDF triples into contexts.

The basic principle is to take an initial snapshot of an RDF data set which is stored into a first named graph context G_{t0}. When modification of the data set is measured at a later visit, the difference is measured as a changeset $S_{\Delta} = G_{t0} - G_{t1}$. This changeset contains the sets of added and removed triples grouped together in individual named graphs (Beetz, et al., 2014). An example is shown in Figure 17.

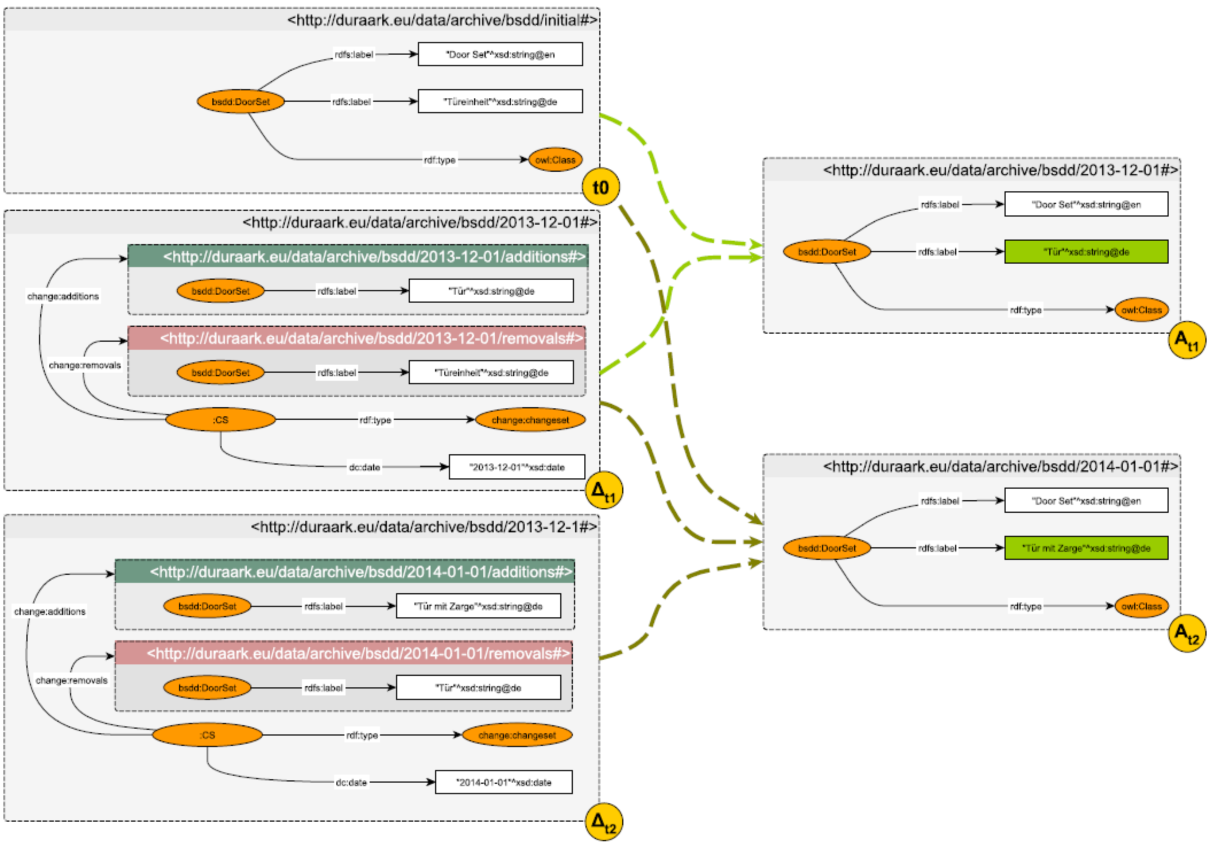


Figure 17: Schematic overview of versioning arbitrary RDF data using Named Graphs (Beetz, et al., 2014)

6 INTERVIEWS

Ten experts in the field of operations and maintenance and/or BIM are interviewed in order to determine how information management in O&M departments is currently done and what can be done to improve it so that information remains up-to-date. In addition, BIM implementation in the industry is discussed. Furthermore the experts are asked if information management can be improved by linking information to components in IFC models. These enriched models act as a network of information which leads to benefits when used during the O&M phase.

The interviewed asset managers are mainly active in housing associations and healthcare institutions, whereas the BIM experts are active in the residential and utility construction sector. All of the interviews with summarized responses from the experts are listed in appendix I.

6.1 Interview findings

Interviewees recognize the inefficiency of information management during O&M in the Dutch industry. Currently, if at all, the contractor makes a selection of information just before the handover deadline which often results in handover of incorrect and incomplete project information. Nearly all of the interviewees emphasize the importance of identifying the informational needs of the client before handover.

Expectations are that BIM models can play an important role in improving the management of information for clients. One of the BIM experts quotes buildingSMART's chairman Patrick MacLeamy: "BIM, BAM, BOOM!" referring to the profits to be gained with the use of BIM in design and construction but most importantly in O&M (MacLeamy, 2014). However, even when there is a BIM model available, these are rarely handed over to the client. The BIM models nowadays are made for design and – increasingly – construction intentions, but the informational needs for O&M are often neglected. None of the interviewed asset managers currently use BIM, but they do recognize the value. By being able to visualise the building in 3D, efficiency and insight can be increased. For example, it is not necessary to go to the site to see what type of boiler needs to be replaced, but instead can be seen in the BIM model.

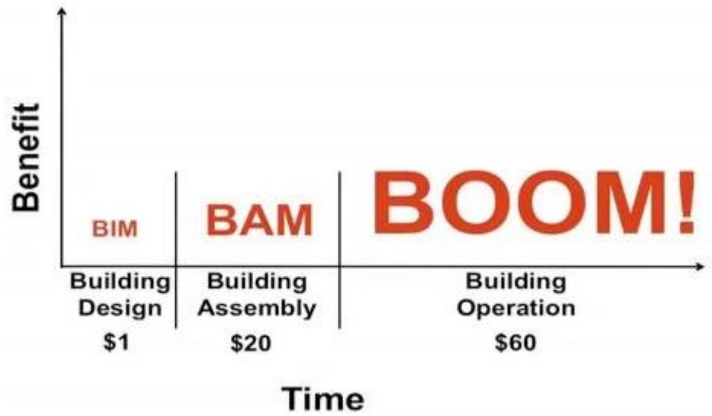


Figure 18: Benefits of using BIM in building's life cycle (MacLeamy, 2014)

Currently, most asset managers still work with files stored on a Windows Explorer-based server. In addition, a range of facility management information systems is used, but interviewees lack interoperability between those systems. Furthermore, most of the information they use is still paper-based. The potential of using open standards such as IFC to manage and structure is recognized, although points of concern are data losses after round trips from proprietary formats. Also, one of the interviewees stated that open standards are crucial, but it should not be necessary to have knowledge of the underlying EXPRESS technology in order to be able to work with IFC.

The interviewed asset managers state that they do not have all information available that they need for executing their tasks. Mandatory log files are usually missing. One of the asset managers indicated that such files, which are mandatory for e.g. fire alarms, can be extracted from the BIM software, whereas now they are manually filled (usually even written instead of digitally kept). A key issue that is identified among all asset managers is the lack of up-to-date information, if this information is digital at all. In addition, one of the interviewees identifies that often only standard contracts are found but that these are not specified to a project. If the information is available, it often takes a lot of time to search for it. One of the interviewees indicated that it would be beneficial if one's usual suppliers would have access to the BIM in case of renovations, so they can keep the information up-to-date in these cases.

When asked in which manner asset managers ensure that information remains up-to-date, they often responded that this does not yet happen adequately. In most cases, revisions are being done but there are no processes in place yet to ensure this is done in a uniform manner. Therefore drawings and documents are not always up-to-date. One interviewed asset manager even states that this is done in only 50% of the cases in his organization, whereas another stated that there currently is no way that this is ensured.

One of the asset managers pointed out that not every single piece of information needs to be included in a BIM. This could lead to an overkill of information. Without a good selection, all of the included information needs to be kept up-to-date during a building's life cycle in order for the model to remain reliable. Likewise, the BIM experts have pointed out that high Levels of Detail (LOD) are not needed for O&M but rather the focus must be on the information that is captured and accessible via the geometrical elements in the model.

Informational requirements for O&M should be defined in the early stages of a project. Ideally, the asset manager is already involved in the design and construction phase in order to oversee the model development and to ensure suitability for O&M. This way, the required information for O&M is already included in the model and does not have to be added after the model is completed. These requirements could be defined in an information delivery specification by the client, which the Dutch Department of Public Works (Rijkswaterstaat) already asks previous to a project.

For structuring and easy access to information, interviewees recognize the value of linking properties and documents to objects in a BIM. Thereby, the following data is mentioned in the context of O&M process optimizations:

- Functional requirements;
- Maintenance planning and costs;
- Quantity take-offs;
- Definition of rooms;
- Condition assessment information of objects;
- Product supplier information and user manuals;
- Warranties and life cycle information of products;
- Service history data;
- Legislation.

In contrast to the possibilities of BIM for O&M, asset managers do place some doubts regarding the transition to using BIM models. They state that the information should be well-defined, structured and manageable. The intent of the model should be clear to everyone that is going to use it. Keeping the model up-to-date is another issue. The one that was named the most is the human factor. Models should be accessible and understandable for all stakeholders, in order for BIM to yield the most benefits. In order to do this, several asset managers have mentioned the NL-SfB classification.

6.2 Conclusion

The interviews pointed out the inefficiency of information management in O&M. This is mainly caused by a mismatch in informational needs between design/construction and O&M. The latter is often neglected in the creation of a BIM. Nonetheless, the potential of 3D models for O&M in order to manage and structure information appears to be significant. However, the specific intent of the model should be clear beforehand so that only the information that is absolutely necessary is included in the model. If too much is included, the obligation to keep all this data up-to-date results in extra work. Setting up a BIM-protocol can be of great value to overcome this issue.

3D models are of value for structuring and easy access of O&M information. Thus the model has the potential to serve as central access point for this data, as opposed to the combination of Windows Explorer combined with a range of FM systems. In addition, there is a need for information to be kept up-to-date, as there are hardly any protocols in place to ensure this. One of the interviewees stated that this is improved by giving suppliers access to a BIM, so that they can directly ensure that the information is kept up-to-date. In order to further research linking information to 3D models for O&M and keeping this information up-to-date, a tool is developed. The working of this tool is discussed in the next chapter.

7 TOOL DEVELOPMENT

This chapter discusses the development of a tool that will enable the facility manager to efficiently handle mutations in the FM phase with the use of a BIM model. The goal of creating an information model in the first place is that it ultimately improves management of building-related information. The reviewed literature and expert interviews already show the potential of models by providing easy access to object information for asset managers in order to make informed decisions about its maintenance and optimal operations.

The tool should assist the facility manager in performing O&M by providing structured object information using IFC models and structure/preserve new O&M information for future use. However, the lack of up-to-date information has been identified as one of the main challenges for the implementation of BIM during O&M (Kiviniemi & Codinhoto, 2014). Following this, the developed tool facilitates altering the O&M data in a 3D model environment, thus ensuring that the information remains up-to-date with the actual situation. In addition, for the objects for which facility managers are required to keep a maintenance history log, changes with respect to these objects are captured in a log. In development of the application and model used in the application, the following programming software and modules are used:

- Revit 2015, using the IFC for Revit export extension[1];
- Qt Designer 4.8.6;
- Python 2.7, with the following libraries;
    - Python OCC 0.16;
    - IfcOpenShell 2.7-0.5.0;
    - PyQt 4;
    - RDFLib 4.2.1.

This chapter starts with a case description and the introduction of a real world IFC model used for development and testing in section 7.1. Afterwards in section 7.2 the user requirements that the application should fulfil are explained. Section 7.3 captures the interaction of the tool with its users, whereas section 7.4 provides a brief analysis of the data to be linked. Subsequently, section 7.5 covers the development of the prototype tool, and sections 7.6 and 7.7 elaborate upon validation and limitations of the tool, respectively.

7.1 Case description

The “Ketsheuvel” project encompasses transforming the former office building of housing association Zayaz to temporary housing for refugees who received residence permits as well as regular home seekers. The building is located at the Ketsheuvel in ‘s-Hertogenbosch. Zayaz is also the client, and Hendriks Bouw en Ontwikkeling is the contractor in combination with Schrijvers Technische Installaties for mechanical/electrical installations and plumbing.

[1] https://sourceforge.net/projects/ifcexporter/

Over the course of ten months the building was transformed and now holds 57 housing units (Schrijvers Technische Installaties, 2016). The housing units range from 20 to 46 m^2, and project completion is scheduled in early 2017 (Zayaz, 2016). Figure 19 and Figure 20 show the building as well as its location. The project design contains 2D information, but no 3D model was available. As a solution, a schematic BIM-model was created and exported as an IFC-file.

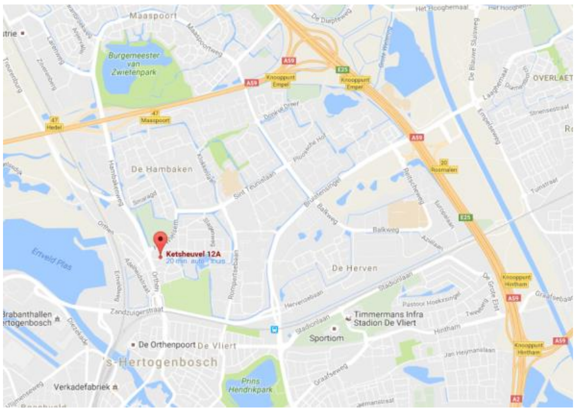


Figure 19: Location of Ketsheuvel

Figure 20: Ketsheuvel

7.2 Application requirements

Based on the objectives that are identified in section 2.2, the main functional requirements for the application are determined. The requirements have been prioritized using the MoSCoW-method. Through this technique the focus is kept on the identified business benefits. The "must haves" (M) are the requirements that must be incorporated in the end result, whereas the "should haves" (S) are strongly desired but still provide a viable solution if not implemented. The "could haves" (C) are desirable but will not be incorporated if the process does not allow for it. The "won't haves" (W) will explicitly not be included in the application (DSDM Consortium, 2008). The prioritized requirements are listed in Table 3.

Table 3: Prioritized list of requirements regarding the application functionality

	MUST
1	Link fire safety data and object information to an IFC model (including 2D drawings)
2	Visualize IFC models in 3D as well as the information about the object or building (e.g. fire compartments, ownership, type of space, etc.)
3	Create change log for certain objects (e.g. elevators, boilers, fire alarms, etc.)
4	Change object-related properties, including placement
	SHOULD
5	Warn if elements have changed since last time they were opened
	COULD
6	Be web-based
7	Visualise changes in 3D;
8	Display models in 3D as well as 2D floor plans;
	WON'T
9	Make animations (rendering)

7.3 Use case diagram

To capture the interaction of the tool with its users – the facility manager and/or the contractor – a use case diagram is developed and provided in Figure 21. This diagram depicts the core functions of the desired software architecture, based on the requirements listed in section 7.2. The tool uses the techniques of the semantic web in order to connect the BIM with external data repositories. Therefore the IFC files that contain the 3D model of the building are linked with RDF data, which is the file format in which Linked Data is constructed and which is queryable with the query language SPARQL.

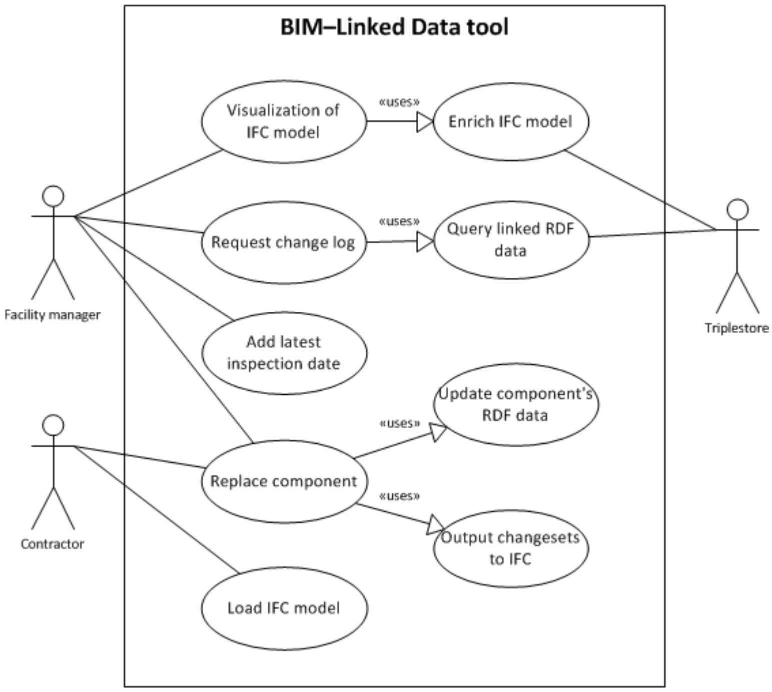


Figure 21: Use case diagram of the Linked Data tool

The use case diagram depicted in Figure 21 consists of two authorized actors, the facility manager and the contractor. The facility manager is the main user. The first task the tool needs to facilitate is to visualize the 3D model of the building. To this end, the triplestore enriches the IFC so that the facility manager can also view information connected to elements inside the model. The contractor loads the model into the tool. Due to the fact that contractors are the ones that will actually execute changes to a building in real life, in case of said changes, contractors will alter the building's geometry.

The tool enables its users to change components that are situated in one of the building's rooms. Changes can be made to parameters connected to the components. These changed components will be reflected in the RDF output of the tool, as well as partial IFC files that capture the changes made to the building. When properties of a selected component are altered, these changes are subsequently saved to an IFC output file. This will function as changeset, so that for each moment in time, the building's history can be accessed. In addition, based on inspection dates that can be added in the tool, the tool creates a change log. This change log is then written to a CSV file.

7.4 Data analysis

In order to script the tool's functionality, a case is obtained along with data sets to be linked to the building model. This use case is elaborated upon in section 7.1. This section provides a data analysis. Table 4 shows the main characteristics of the acquired data sets. Each of the data sets seem to be produced by a different actor in a different stage of the building life cycle. In addition, they are also produced by different applications and therefore exist of different information structures.

Table 4: Data to be linked in the application

	Building information	Fire safety data	Inspection reports
Actor	Cier Architecten	Zayaz	Vgib
Phase	Design	Construction	O&M
Tool	AutoCAD	MS Excel	MS Word
Structure	2D geometry in CAD	Data in tabular format	Plain text descriptions
Format	.dwg	.xlsx	.pdf

It is crucial for the building model to be set up in the IFC file format in order to enhance interoperability. However, the provided model is set up in the DWG-format consisting of 2D drawings. As a solution, a BIM-model was created in which the building is modelled on a schematic level. Using the IFC for Revit export extension mentioned in the beginning of this chapter, the building model is exported as an IFC file and provided in Figure 22.

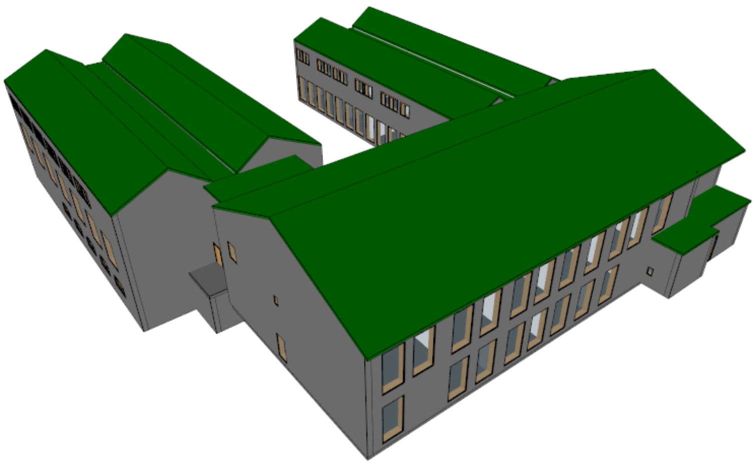


Figure 22: Ketsheuvel IFC model

Regarding the fire safety data, these are provided in a tabular format. This data set firstly specifies for a building in total if certain installations are present, for example how many fire extinguishers or smoke detectors. Subsequently, the data set specifies for the installation objects certain attributes, such as the supplier, type, date of the last maintenance, etc.. A detailed listing of the data set is provided in appendix VI. However, these parameters in the input data set are only specific to the building as a whole, which means that it is only specified how many of a certain component is situated in a building. Therefore, this data is further specified as to the rooms in which certain components are located.

7.4.1 Ontology engineering

The following chapter provides a thorough elaboration of how the previously described data sets can be formally expressed using OWL domain ontologies and subsequently interlinked. The scope of such an interlinked ontology is to define the O&M data with respect to installation components in an explicit way. This allows to connect and visualise this information in a 3D environment. Since reusability is one of the main principles to follow when developing ontologies, it is important to search for existing ontologies that best fit the data set. The following criteria need to be taken into account when selecting ontologies (Radulovic, et al., 2015):

i. the semantics of the class or property in the ontology is related to the term;
ii. if the term relates to a class, the class in the ontology has as many properties that correlate to the term as possible;
iii. the ontology that describes the class or property related to the search term is widely accepted and used.

In case no widely used ontology conforms to the criteria listed above, an own ontology is developed through the following steps: 1) definition of classes and class hierarchy, 2) definition of properties of classes (slots) and 3) definition of the facets of those slots (Noy & McGuinness, 2001). With the above in mind, each subsequent paragraph covers the development of an ontology of one data set.

7.4.1.1 Fire safety data

In controlling fire safety for buildings, certain information needs to be kept up-to-date. Legislation such as annex I of the Dutch building code and NEN regulations (2654-1) states that fire alarm systems and emergency lighting need to be managed and maintained (Ministry of Internal Affairs, 2012; NEN, 2015). In the same sense, the use of a log in order to track the maintenance of such systems is also mandatory. However, every company does this in a different way and uses concepts differently. As a result of this, there is no agreement for a formal ontology as of yet, which is able to describe the components of such systems and how and when they are maintained in an explicit and unambiguous way. This observation has been confirmed by several discussions with asset managers from Vgib, as well as other companies. Therefore it is chosen to develop an OWL ontology supporting the aforementioned scope, using TopBraid ontology design software. This ontology is included in appendix V.

A data set was provided by the housing corporation as to which information is necessary to maintain for their systems relating to fire safety. The columns used in this Excel data set were translated to comprise the actual OWL ontology, which is named “Building Systems”. This developed ontology uses the prefix “busy” for the Unique Resource Identifier (URI) `http://example.org/ont/buildingsystems#`. The design of this URI is specified further in section 7.5.1.

Figure 23 provides an overview of this established ontology. The ontology focuses on two main elements; a system made up by its components, and the building complex in which the system is placed. The building complex is further specified through a subclass representing the buildings in the complex and some basic properties containing information about the complex. Furthermore, the original data set groups the components per building complex. However, requirements have been formulated to specify the location of components by room. This way, the components can be queried per building as well as its location in that building. This also enables interlinking to the IFC data through:

```
busy:Room busy:contains busy:Component .
```

The `busy:Component` contains several properties with usual O&M information such as the manufacturer, brand and type of a component, as well as the party that provides maintenance. In addition, two `xsd:Date` properties have been added that describe when components are installed and when they were maintained last. This enables the user to query not only for all components in a room, but also all components maintained before a certain date.

```
busy:Component busy:lastmaintained xsd:Date .
```

Between the building complex and the components, the class `dcterms:Agent` represents the parties that have a role in the O&M phase; the owner and tenant of the building complex and the party that operates and/or maintains the system. In line with the principles of ontology development provided at the beginning of this section, this has been re-used from the widely used Dublin Core ontology. Finally, in order to define a hierarchical structure, all concepts of the ontology are a subclass of `owl:Thing`. In conclusion, the resulting ontology consists of seven classes with 18 properties. The ontology was named “Building Systems”, using the prefix “busy”.

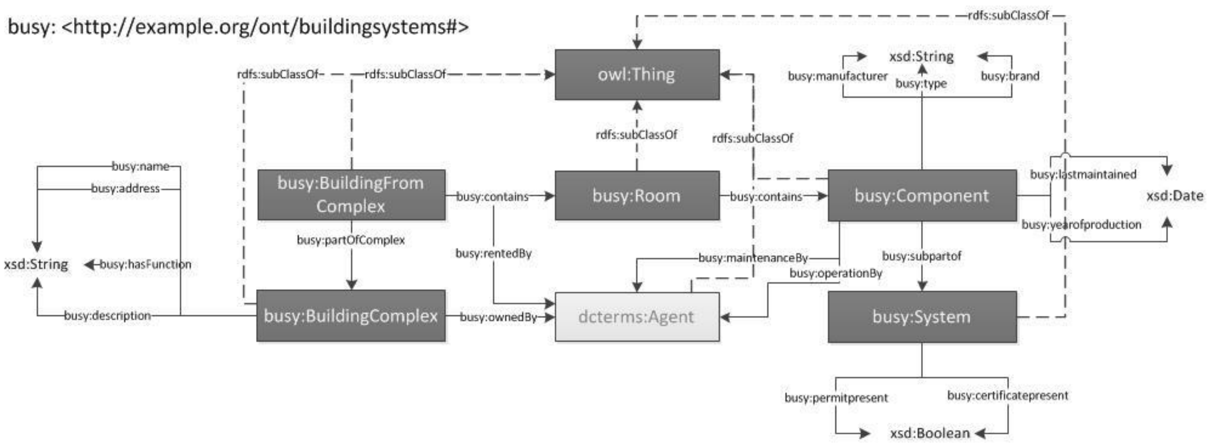


Figure 23: Overview of the established ontology formally expressing the system data set

The ontology offers some possibilities to be linked with the IFC data. The concept `busy:Room` is to be linked to the IFC concept `IfcSpace`. In addition, the concepts `busy:BuildingComplex` and `busy:BuildingFromComplex`, provided in the left part of Figure 23, are potentially linked to the IFC concepts `IfcProject` and `IfcBuilding`, respectively. This enables the data set to be connected to multiple projects, so that data can be analysed across the whole portfolio. Finally, the concept `busy:Component` can be connected to the objects in IFC files if an MEP reference model is included. Listing 1 indicates how this linkage might be achieved.

Listing 1: Example linkage of data set with IFC

```
busy:Room rdf:subClassOf ifc:IfcSpace .
busy:BuildingfromComplex rdf:subClassOf ifc:IfcBuilding .
busy:BuildingComplex rdf:subClassOf ifcowl:IfcProject .
```

7.4.1.2 Log entry data

In order to prevent the log data from polluting the actual data regarding the building systems, a separate ontology has been created which covers the necessary information to comprise the log entries. Figure 24 visualises the ontology that contains all of these aspects with regards to the log entries. The ontology is developed using TopBraid ontology software, and provided in appendix V in Turtle notation. Each log entry needs to contain:

- who made the entry;
- when the entry was saved;
- in which room the inspected component is situated;
- which component has been inspected, and;
- the reason the inspection was done.

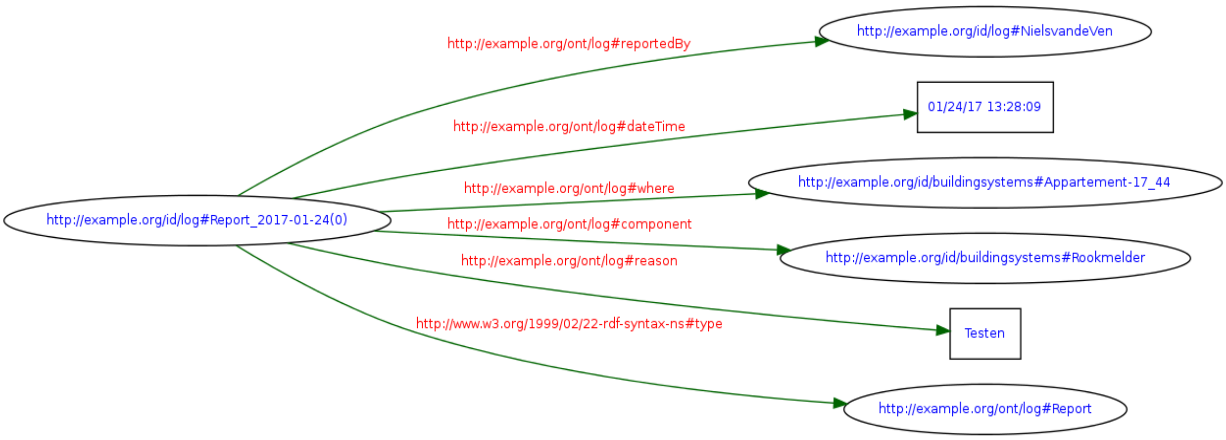


Figure 24: Overview of example log entry predicates and objects

7.4.1.3 IFC data

In order to lift IFC data to an ontological level, various EXPRESS to OWL conversion procedures have been proposed in the past. This has resulted in ifcOWL, previously mentioned in section 5.3.1, and provides an OWL representation of the IFC schema. The most recent mapping procedure is proposed by Pauwels and Terkaj (2016). This conversion allows ifcOWL to be structured according to the original IFC schema.

However, given the importance of the visualization aspect of the prototype tool to be developed in this research and the fact that there are hardly any adequate tools that support the visualisation of IFC models after conversion to ifcOWL, it was decided to stick with IFC files, which are easily visualised in 3D. Figure 25 schematically visualises the data linkage in which IFC elements are linked to the relevant parts of the RDF data set.

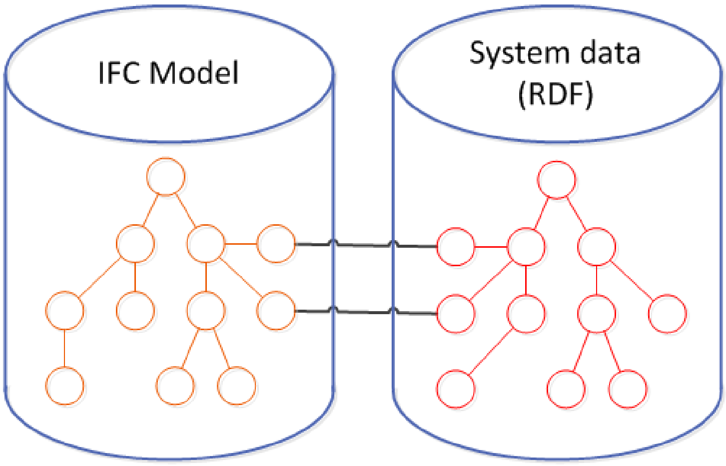


Figure 25: Schematic representation linkage between IFC and RDF data (based on (van der Riet, 2016))

7.5 Development of prototype

The developed prototype consists of two stages. The first stage, Tab A, enables the asset manager to transform tabular data to RDF in order to link this data to the relevant parts of the IFC model. Subsequently, the working of the second part of the tool, Tab B, is explained. This part enables visualisation of the IFC model along with the connected O&M data, as well as a versioning mechanism of this O&M data. In addition, Tab B contains several operations in order to ensure that the information remains up-to-date. The complete script of the prototype tool can be found in appendix II.

7.5.1 Prototype Tab A: RDF parser

This section provides an elaboration of the way in which the Linked Data sets are actually created according to the domain ontologies described in previous sections. Linked Data is created by following the four Linked Data principles, which are already mentioned in section 5.1; 1) using URIs for identification of resources, 2) using URIs according to the HTTP standard, 3) using open standards such as RDF and 4) including links to resources in other data sets.

Storing O&M data in semi-structured formats such as CSV, XML or JSON is generally only usable for other asset managers because there is no description of the actual data available. Using Semantic Web technologies and the capability of RDF, this issue can be mitigated. The conversion of such data to RDF results in data that has semantic meaning, is easy to interlink with other relevant data sources and can be queried with SPARQL (McGlinn, 2015). Tabular data can be converted into RDF in various ways, for example via tools such as OpenRefine. However, for flexibility reasons the Python programming language was chosen to conduct the conversion. Therefore, the first part of the tool facilitates the conversion of tabular O&M data into RDF, using the csv and RDFlib libraries. The Python code of this conversion is included in the complete Python script of appendix II.

The URIs for this research are carefully constructed according to the URI-strategy for Linked (Open) Data in the Dutch public sector which is founded upon the principles of 'Cool URIs for the Semantic Web (W3C, 2008)':

`http://{domain}/{type}/{concept}/{reference}`

The domain `http://example.org/` is chosen as base domain of the URI, due to this domain being free to use without needing prior coordination or permission. Afterwards the key terms `id` and `ont` are used as the types in order to indicate whether the URI refers to a data instance or an ontological concept. This term is followed by the concept belonging to the property in question. Finally, for the components a GUID is used as reference in order to ensure uniqueness of the URI. An example of a URI belonging to a data instance is (Overbeek & van den Brink, 2013):

`http://example.org/id/buildingsystems#Rookmelder/7e9eead0-c909-4dfa-b21a-f4fde112d6b0`

Figure 26 visualizes a part of the tool's output by means of an RDF graph. This graph shows triples with a system component as subject, and predicates and objects from the tabular O&M data itself. For an asset manager to be able to easily perform the conversion to RDF, a graphical user interface (GUI) was programmed with QT Designer. The designed GUI for both parts of the tool is included in appendix IV. For the steps and sequence of this conversion formulated in the Python script, an overview of this tool is visualized as a flowchart which can be found in appendix III. A snippet of the input data can be found in appendix VI, whereas the output (RDF) is partially included in appendix VII.

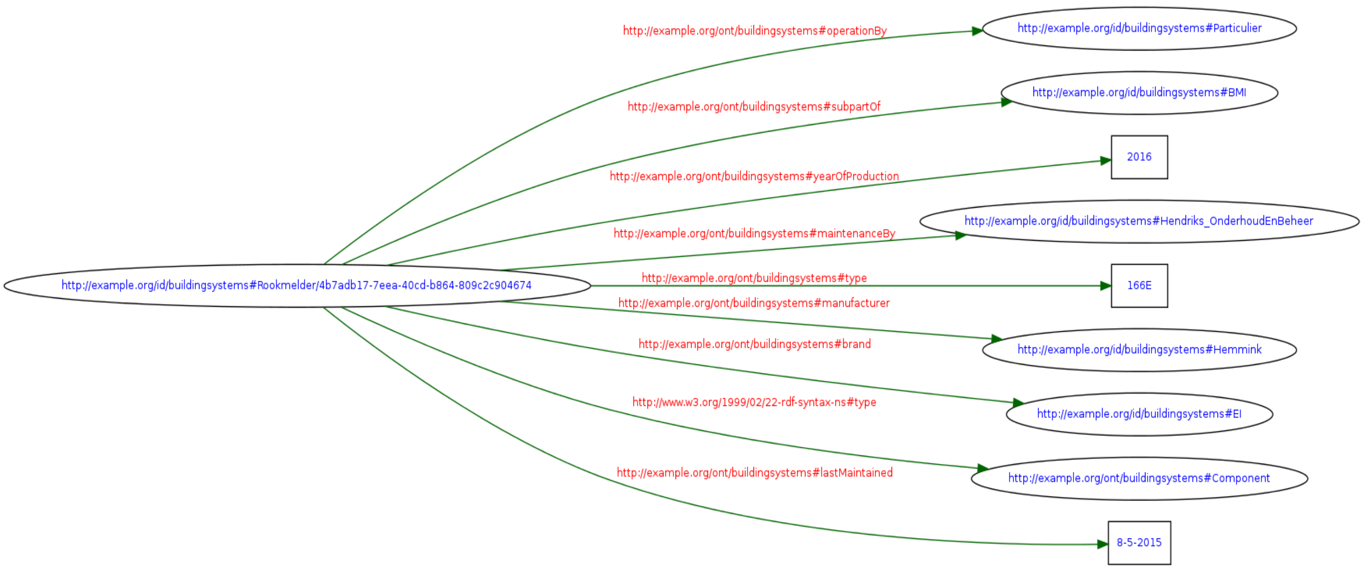


Figure 26: Partial RDF graph example after conversion

7.5.2 Prototype Tab B: IFC and RDF viewer

This part of the prototype tool enables the asset manager to view IFC properties with the addition of RDF data. The approach of enriching IFC files with arbitrary RDF triples is suggested by the DURAARK initiative (Beetz, et al., 2014). While nowadays scientific literature is vastly available on the transformation of the core model or partial model chunks into RDF, this demands an ample shift in technologies. Therefore this part of the tool provides the ability to visualise an IFC model with element properties and to provide associated system RDF data by selecting an element in the 3D view.

Given the reduced importance given to the Level of Detail in 3D, as mentioned by Verbaan, et al. (2014), the IFC model is visualised in terms of spaces. Per space in the IFC model, its respective system components are given with the RDF properties that relate to that specific component. This visualisation is shown in Figure 29. The ability to connect both types of data arises due to the use of common element indicators, in this case the name of the room. In the IFC model that has been created with Revit, the property set 'Pset_SpaceCommon' contains an instance of *IfcPropertySingleValue* where the name of the room is appended with the number of the space instance. This instance is unique and in absence of other classification systems is used to link the IFC file with the RDF data. This principle is seen in Figure 27, where the property instance is connected to an RDF triple that shows a component that the room in question contains.

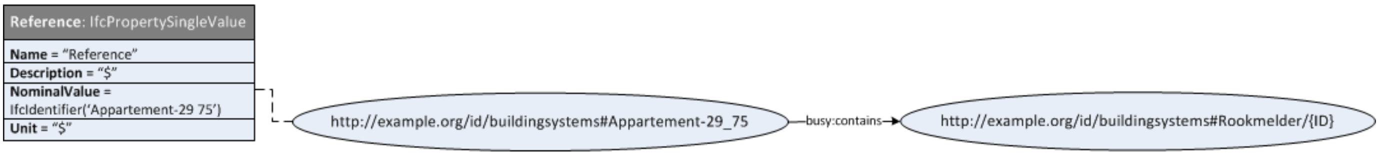


Figure 27: Linkage of IfcPropertySingleValue to RDF triple

Tab B first parses the IFC file using IfcOpenShell and PythonOCC, which results in a model visualisation where each room is given a default colour. After loading an associated RDF file, the graph is then parsed and a Python dictionary is created from the file with the names of spaces. In addition, a Python dictionary is created with GUIDs and corresponding IFC properties, such as the area of the selected room. When the user selects a room, the corresponding GUID is set as global variable and through these pre-established dictionaries, the corresponding IFC properties are displayed by activating the 'Show properties' button.

Given the linkage between the spaces and their counterparts in the RDF data, the bulk of Tab B's functionality is based upon the use of SPARQL queries. For example, Listing 2 shows the query that searches for components situated in a certain room. Such a query is used when a room is selected in order to fill the second window on the right side of Tab B with components that a room contains. The line that uses the prefix *FILTER NOT EXISTS* ensures that the most actual situation is depicted. By selecting the component in this window, the latest inspection date is queried as well as the other predicates and objects relating to the selected component. These are then appended to the bottom textbox, as can be seen in the GUI depicted in Figure 29.

Listing 2: Example query for components in a room

```
PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?c
WHERE
{
	?c rdf:type busy:Component .
	<http://example.org/id/buildingsystems#Appartement-02-18> busy:contains ?c .
	FILTER NOT EXISTS { ?c busy:replacedBy ?cnew }
}
```

In order to ensure that information remains up-to-date, two functions are added: ‘Add to log’ and ‘Change component’. The former adds the latest inspection date to the selected component, whereas the latter can be used when a component is replaced. When a component is replaced, a new data instance is created in the form of RDF triples which represent a component, with the ability to change manufacturer, brand or type. In addition, the triple that has been included in the query above is added to the old component. This triple, using the predicate `busy:replacedBy`, indicates that the old component has been replaced by the one that is inserted when activating this function. Figure 28 visualises this principle using a snippet of the data produced by Tab A. The instances in this figure use the namespace `http://example.org/id/buildingsystems#`, which keeps the instances separated from the concepts in the ontology.

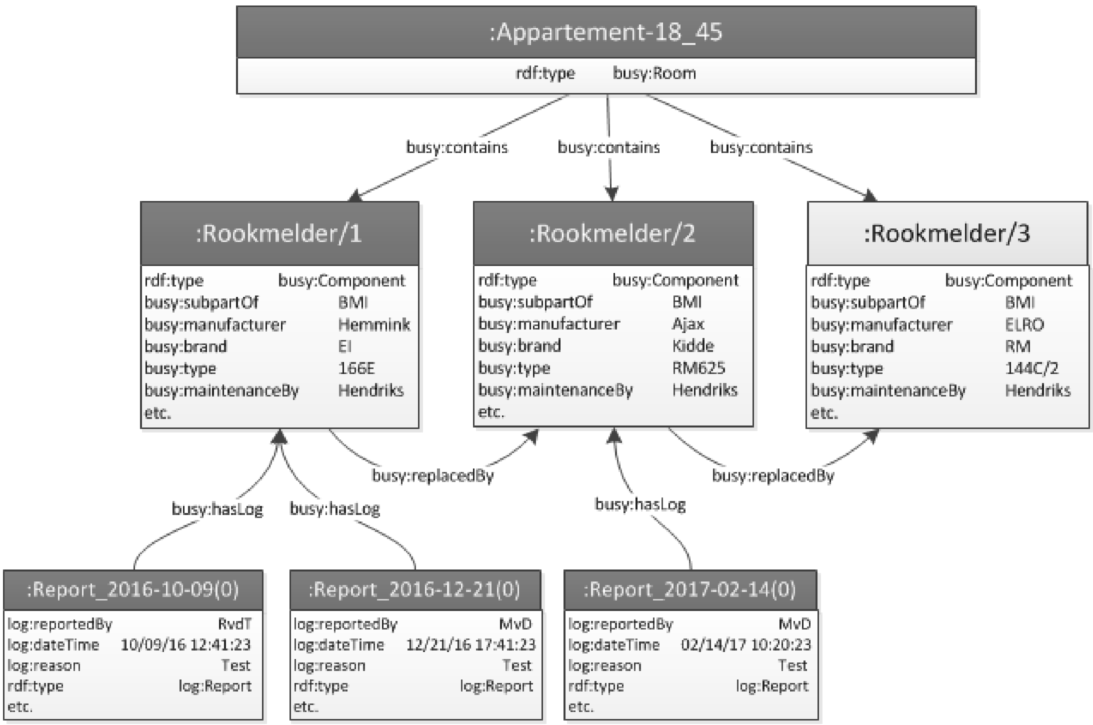


Figure 28: Schematic overview of developed versioning mechanism

The room “Appartement-18_45” contains one smoke detector. In Figure 28 this component has been replaced two times, where each component has different information relating to this component. In addition, the bottom instances represent the log entries which indicate that the components have been maintained. These log entry instances have been added using the “Add to log” function. The first component has been maintained twice, whereas the currently installed component has not been maintained yet. The component with the lighter colour is the current one, which can also be seen because it has no `busy:replacedBy` relation pointing to another instance.

In addition to the new RDF data instance, a separate IFC file is also created. In this file, for each of the components that have been changed using the tool, an IfcBuildingElementProxy is created with the current properties that have been inserted in the RDF data set as well. This newly created IFC element is then placed in its respective room. This way, for each time that a change occurs, these created IFC files act as changesets so that for each moment in time the current status at that time can be inspected.

To be able to benefit from the visualization aspect of the 3D environment of the tool, a function is included. This function queries, after selection of a component, the rooms where this component is situated as well. Figure 29 shows an example of the result of such a query, where a room contains a smoke detector. The rooms in the screenshot that have been coloured red also contain smoke detectors.

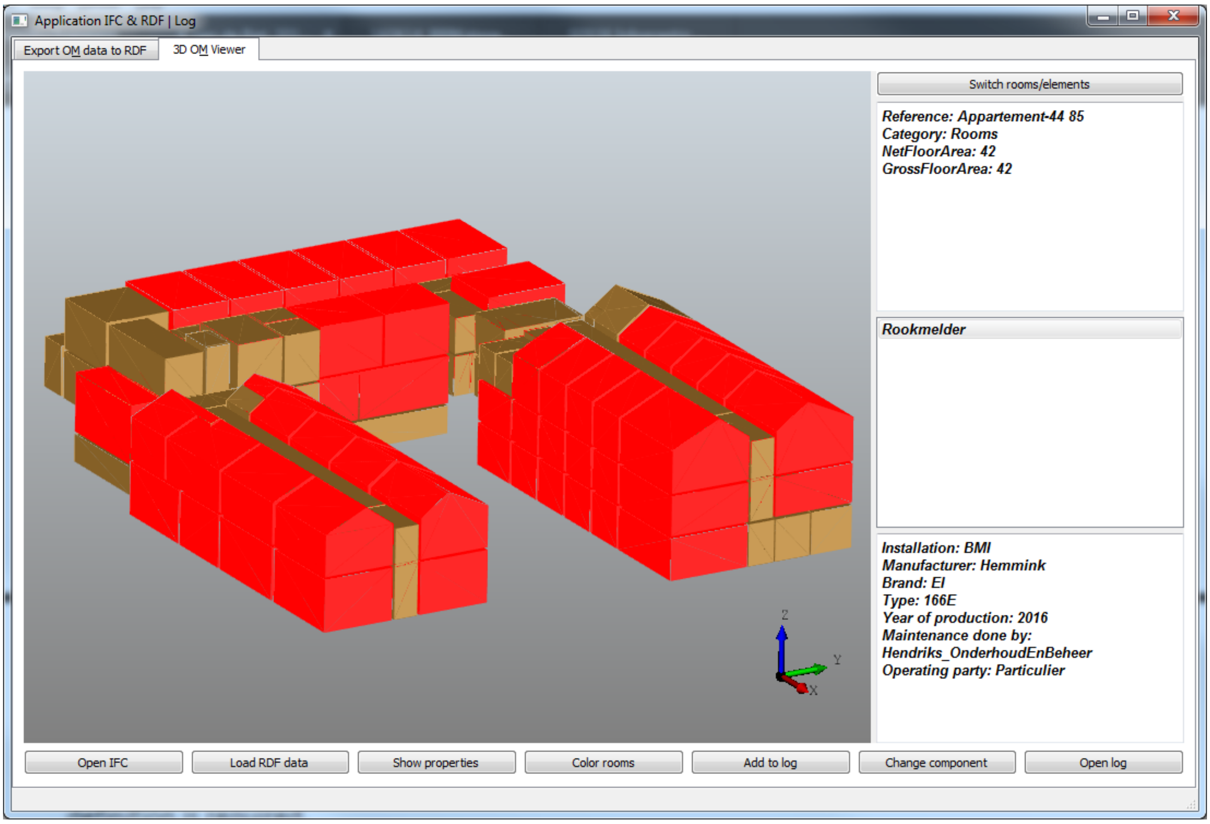


Figure 29: Graphical user interface of Tab B

Finally, the tool offers the functionality to save the log entries to a CSV file, which can be opened using Microsoft Excel. In line with the functions described before, this is also done with a SPARQL query. The query that is used, provided in Listing 3, and queries the properties that have been defined in the separate ontology, described in section 7.4.1.

Listing 3: SPARQL query used for log file creation

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX log: <http://example.org/ont/log#>

SELECT ?who ?when ?where ?what ?why
WHERE
{
      ?i log:reportedBy ?who .
      ?i log:dateTime ?when .
      ?i log:where ?where .
      ?i log:component ?what .
      ?i log:reason ?why .
      ?i rdf:type log:Report .
}
```

This SELECT query, given in Listing 3, aims to find the five aspects that make up the log file, and subsequently writes them to a Excel file. Table 5 provides an example of the output of this function. By adding this functionality, the mandatory maintenance history logs can be automatically extracted from the tool whereas currently these are filled manually apart from other maintenance processes.

Table 5: Output log file

Wie?	Wanneer?	Waar?	Wat?	Waarom?
NielsvandeVen	01/23/17 13:04:54	n.t.b._97	Rookmelder	Testen
NielsvandeVen	01/23/17 13:05:07	n.t.b._97	LED_armatuur	Testen
NielsvandeVen	01/23/17 13:06:23	Trappenhuis_108	Vluchtwegaanduiding	Testen
NielsvandeVen	01/24/17 11:23:11	Gang_92	Vluchtwegaanduiding	Testen
NielsvandeVen	01/24/17 11:23:26	Appartement-04_16	Rookmelder	Testen
NielsvandeVen	01/24/17 11:23:37	Gang_92	LED_armatuur	Testen
NielsvandeVen	01/24/17 13:28:01	Gang_111	LED_armatuur	Testen
NielsvandeVen	01/24/17 13:28:24	Gang_111	Vluchtwegaanduiding	Testen
NielsvandeVen	01/24/17 13:29:31	Appartement-17_44	Rookmelder	Testen

7.5.3 SPARQL queries

As mentioned in the previous section, the bulk of the tool's functionality is made up of SPARQL queries. Storing O&M data in RDF format with the ability to retrieve specific data using SPARQL queries allows the data to be shared and consumed in a more interoperable manner. Through fostering reusability of data, data sharing between the AEC and the FM industry can potentially be stimulated.

In order to demonstrate this type of querying data, the RDF data created with Tab A is locally published on a Fuseki server and simple SPARQL queries are run. The first example, given in Listing 4 shows how the rooms and components that have been replaced in a previous session can be retrieved from the RDF data using a SPARQL query. Such information is often lacking, which can result in replacement orders for components that already have been replaced.

Listing 4: Example SPARQL query to find replaced components

```
PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?r ?cnew
WHERE
{
	?r busy:contains ?c .
	?cnew rdf:type busy:Component .
	?c busy:replacedBy ?cnew .
}
```

In this case, the query results in four rooms which each contain a replaced component. The results of this query are illustrated in Figure 30.

Room	New component
<http://example.org/id/buildingsystems#Appartement-02_18>	<http://example.org/id/buildingsystems#Rookmelder/115a292b-bd4c-4005-8c4f-1f3b9621698f>
<http://example.org/id/buildingsystems#Appartement-18_45>	<http://example.org/id/buildingsystems#Rookmelder/c98a108f-3754-472c-ae62-83912db9a209>
<http://example.org/id/buildingsystems#Appartement-44_85>	<http://example.org/id/buildingsystems#Rookmelder/2d27c4a0-d3c3-40c0-8d4b-a3adf01b4336>
<http://example.org/id/buildingsystems#n.t.b._97>	<http://example.org/id/buildingsystems#LED_armatuur/07a47848-4489-416d-a77f-aa70e2755603>

Figure 30: Result SPARQL query for replaced components

The second example, provided in Listing 5, finds for each component which party manages the system. In reality, it is often unknown which party is responsible for managing, for example, the fire safety system or emergency lighting.

Listing 5: Example SPARQL query to find responsible party for managing components

```
PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?Responsible_party ?Component
WHERE
{
	?r busy:contains ?Component .
	?Component rdf:type busy:Component .
	?Component busy:operationBy ?Responsible_party .
}
```

This yields the following results, shown in Figure 31. This shows that the smoke detectors placed in the apartments are the responsibility of the tenants themselves, whereas the smoke detectors as well as emergency lighting units are the responsibility of the housing association that owns the building. Although the data set only specifies the responsible parties for either the corporation or the tenants, the `busy:operationBy` relation provides the possibility to connect the O&M data set to the tenants system in order to make this relation more specific.

Responsible_party	Component
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#Rookmelder/662bac5f-d4b8-4e9d-b3ce-2cdcccc3bc53>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#Rookmelder/48f683b5-fc60-49e7-8f47-4e84e25e8912>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/1840ef10-33f7-4ded-b53a-79d4e9e08926>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#Vluchtwegaanduiding/ae405962-08f9-4a4d-b012-86514c74e251>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/a428053e-02c1-4b51-8837-c6a4c23216ef>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/e1e5d20b-8a25-43a3-88d3-789d2cfda72d>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/cba397ab-8473-43a1-b649-a09a2417d637>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/8ae5a940-f2f1-4db9-8680-2c4f682e4146>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/b42f5adb-266b-4f53-86b6-580e0092867c>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#Vluchtwegaanduiding/c9a57786-c3ad-40f0-b036-ffea3a5550a1>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/ec00eaf4-a0fc-4a12-8b95-718282f8061a>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#Vluchtwegaanduiding/4337c16b-bc22-4c80-8fd2-10b504ec2e88>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/50ab5670-1f8f-4d20-819b-e2c7750085bd>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/d0eca652-7289-42be-af60-fa8e4bccc8b3>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#Vluchtwegaanduiding/f8fec375-19d6-43c2-ad4f-8148ac4eea03>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/4d77b5ae-d790-4293-a455-20f913272281>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/3aa38080-fb2e-4ea4-ab3b-6d33837b1afd>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/6242af9c-292b-4247-8b86-5908b242cdfd>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/0c5cf4c8-8e56-45d3-9d8d-1828c48f4344>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/2e89c242-597e-4cc3-92f3-3f5587e72ecb>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/7ac202db-f490-4059-a465-bc7a4851418d>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/557d4906-9e73-4169-851d-4b9787abb740>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/e90513f0-7ce1-4a30-ac1c-514faeeb914a>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/a33adec2-8691-4c7b-9175-a4ff91c5ab50>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/8f9295af-fa22-444d-96c0-ec48a6941804>
<http://example.org/id/buildingsystems#Zayaz>	<http://example.org/id/buildingsystems#LED_armatuur/e183b39c-4c30-433b-a126-a0e0687ee2e7>
<http://example.org/id/buildingsystems#Particulier>	<http://example.org/id/buildingsystems#Rookmelder/199c87a5-bf74-4cf6-a3d0-7c5397ea30ef>

Figure 31: Result SPARQL query for managing system components

Depending on the question at hand, there are hardly any values that cannot be retrieved through the use of SPARQL queries.

7.5.4 Conclusion prototype

Efforts to develop a working prototype have resulted in a usable and functioning tool. IFC files are easily linked to external data repositories. The ability to connect these data sources arises through common element indicators or by mapping indicators in both data sources. By separating data sets and usage of the RDF format in the way that the prototype tool proposes, combined with the use of SPARQL queries, versioning can be done more easily. Thus, the risks with regards to versioning are avoided, while the mandatory maintenance history logs can be easily extracted as well.

However, the bulk of current O&M working methods do not yet include working with BIM, let alone working with RDF files. Therefore, data transformation is still required which involves versioning issues. Despite this, the proposed working method appears to be more future proof due to the expected shift towards Semantic Web technologies within the industry. The RDF format offers improved data reusability and interoperability, for instance by using SPARQL queries as demonstrated in the tool as well as the examples in the previous section. This has the potential to enhance data sharing between AEC and FM industry, facilitating e.g. design for maintenance.

7.6 Tool validation

This section provides an elaboration on the validation of the developed tool in terms of correctness and completeness of generated data and data visualizations. Firstly, validation of the output of Tab A of the prototype is carried out using the RDF validation service by W3C. The RDF output file created by the parser in Tab A is parsed at this server, which results in RDF triples, optionally the RDF graph, and the announcement that the document has validated successfully. The RDF file as it was exported from the tool has indeed validated, as can be derived from the announcement provided in Figure 32.

Figure 32: RDF validation (W3C, 2006)

Apart from the completeness of the generated RDF file, the conformance of the data set to the ontology defined in section 7.4.1 is also tested. Each of the concepts contained in the CSV data set is translated to its counterpart defined in the Building Systems ontology. Two properties from the developed ontology are left out of the conversion, namely `busy:permitPresent` and `busy:certificatePresent`, because no permits or certificates were linked. Apart from this, the generated RDF file conforms fully to the defined ontology.

In order to test the completeness of the generated RDF file, Tab B is used. This part relies on the RDF file for visualization of the RDF data connected to the rooms in the IFC files. If data is lost in the conversion of the tabular O&M data to RDF, some rooms that have components attached to them will yield no results or will not be visualized in colour if this query is run. Furthermore incomplete property representation would be easily noticeable.

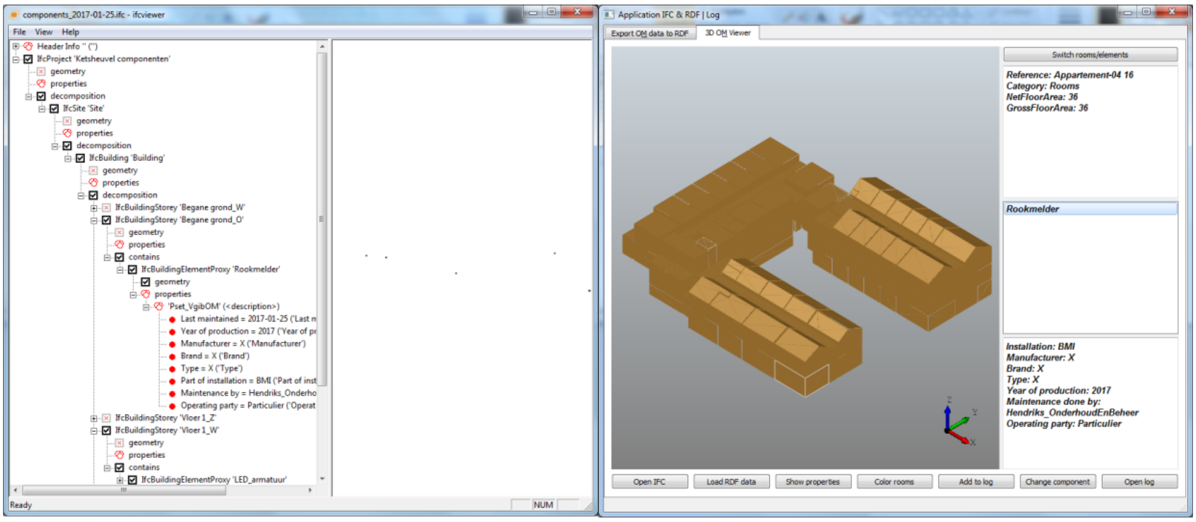


Figure 33: Validation of changed component parameters (IFC viewer and prototype tool)

Detailed inspection yielded no situations of incomplete or incorrect data representation. In addition, the versioning mechanism of the tool, which is schematically depicted in Figure 28, is validated. The input IFC model and the RDF data is loaded in the tool, and a number of components are replaced. Afterwards, the output RDF file should reflect these changes, as should the IFC output of the components. Figure 33 shows both the tool with changed parameters and the output IFC file generated from these changes. Inspection of these files showed no situations of incorrect data representation.

7.7 Tool limitations

In this thesis a prototype has been developed for a versioning mechanism combined with a 3D viewer for IFC & RDF data representation, based on the Semantic Web standard. This prototype has yielded an easy to use and open way of accessing and assessing data in a model towards O&M. By using the tool, the need to comprehend the complex data structures of the IFC schema is reduced. Thus, the use of IFC is made easier and less error prone. However, aside from benefits, developing of the prototype also has some limitations.

The first of the limitations is expected to be found in the size of the IFC files. Large BIM models will probably reduce the prototype's performance significantly. Tests using the case model of nearly 3 megabytes have proven to cause no issues. In addition, the development of the prototype is based on version 2x3 of the IFC schema, which means in order to be compatible with the latest version – IFC4 – some modifications may be necessary.

In addition, the application is designed to work exclusively with a tailored template regarding input data. Therefore, in order to adapt to changing informational need over time, small script modifications are needed. Through small script additions, the prototype can be made compatible with element classifications, such as NEN 2767 (NEN, 2012). However, this is currently still lacking due to the absence of element classifications in the case study. Similarly the application only handles O&M data on element level, specifically the rooms. Although the script can be altered to be able to handle IFC components as well, it is expected that simplification for the purpose of this research will be of no influence on the outcome due to the emphasis on information management and not the data itself.

8 CONCLUSION

The main research question of this thesis is "How can changing information in the Operation & Maintenance phase be handled efficiently using 3D Building Information Models and Semantic Web technologies?". This section provides an answer to this question, by firstly discussing the sub-research questions.

What are the most important processes in O&M, and which information is necessary for these processes?

The operation and maintenance (O&M) of buildings encompasses all services that are required to assure the facility will perform the functions for which it was designed. It includes the day-to-day activities necessary for the building and its systems and equipment to perform their intended function. It is the task of O&M managers to keep the facility running to maximise efficiency and minimise financial costs involved. Therefore planning needs to be done so that maintenance can be carried out as little as possible and order to do so, it is vital for the O&M manager to be aware of the facility's actual state. Tasks of an O&M manager can be divided in two categories; facility operations and maintenance and repair. This contains aspects such as inventory management, energy management and fire safety.

Concerning the information that is needed in order to perform adequate O&M, attribute data is valued over graphical detail. Therefore the tool developed in chapter 7 parses IFC models as rooms as opposed to detailed building components. However, the 3D visualisation was deemed very effective during the conducted interviews. An example of this can be found in the ability to see where elements are located in a building so that elements can be easily retrieved. The organisation of Dutch housing associations, Aedes, has named a number of aspects deemed necessary for effective O&M (Aedes Werkgroep BIM, 2016). The most vital of these, as confirmed by the interviews that were conducted as part of this thesis, are:

- the surface area of rooms in a building;
- quantities of building elements and system components;
- product supplier information;
- functions of rooms;
- service history data;
- condition assessment of objects.

What is Building Information Modelling and how can it facilitate information storage and exchange during the O&M phase?

Building Information Modelling (BIM) is considered a business process for generating and leveraging building data to design, construct and operate a building during its life cycle. BIM models have the ability to structure O&M information efficiently by linking to geometrical representations of building elements, thus providing easy access to O&M data by enabling users to select elements and access its associated data. Storing non-geometrical information in IFC files can be achieved through storage of properties within an *IfcPropertySet.* However, this has some limitations, such as difficulty in querying and reusing data in IFC models.

The predominant open standard for the exchange of non-geometrical information in BIM is called Construction Operations Building information exchange (COBie). However, COBie is structured according to a specific syntax – either tabular templates or the complex IFC schema – which lacks formal semantics. This hinders reuse of the data and impedes potential verification tasks. In addition, asset managers may view them as "yet another spreadsheet", which can add to the confusion in the current situation of information management. One of the key conclusions of the interviews was in order to efficiently facilitate information storage and exchange in O&M, it is vital to make a good data selection prior to handover. Otherwise if too much information is transferred, all of the information needs to be kept up-to-date which results in extra work.

What are important IFC model conditions for efficient use in the O&M phase?

Ideally, IFC models are handed over representing the as-built situation. The use of GUIDs for each object is imperative and, additionally, in an optimal situation no object properties should be in an IFC model itself, because information concealed in the data structure of IFC is difficult to query and re-use. Also, for every adjustment of the native data repository an extra IFC export needs to be made manually and distributed. Preferably object properties are present using Semantic Web technologies as demonstrated in this thesis, using common element indicators. Using standardized object classifications is supported in literature, but due to the absence of classifications in the case study this was not used in development.

Contrary to the often assumed preferable situation where the construction team hands over every highly detailed piece of information possible to the O&M phase, the expert interviews revealed that this situation is far from recommendable. This often leads to reworking the information in order to make it usable and manageable, as well as extra work due to the obligation to keep all of this information up-to-date during the whole exploitation phase of a building. Making an adequate data selection prior to handover is therefore crucial for efficient use of building models in the O&M phase. This selection might be done using COBie or another handover MVD, but needs to be agreed upon by all involved parties.

What approach can be employed to track changes in information in BIM models?

The current practice in tracking differences between models is via version deltas and to compare the GUIDs of objects. However, incorrect management of these GUIDs or operations which delete an object and create new ones in its place will render both processes completely ineffective. In addition, most entities in an IFC file are not identified in this manner, because these do not inherit from `IfcRoot`. These entities are only identified with line numbers that are only valid within a single file and not between versions. Therefore this thesis explores the use of the semantic web standard to track changes in BIM models and related non-geometrical data.

In order to implement Semantic Web standards in a BIM-centred working method, the tool first converts associated model information to RDF. Afterwards, by using a common element indicator the RDF data is linked to elements in an IFC model. The tool facilitates altering data connected to IFC elements and subsequently inserts these updated properties into the RDF dataset. In addition, a component is added to an IFC file which acts as a changeset. This way service history of these components can be viewed at any given moment in time, while the IFC remains valid and readable with available IFC viewers.

How can different model versions be mapped using Linked Data?

Currently there are few methods that adequately handle versioning in IFC models, let alone using Linked Data. Therefore for this research a tool was developed that employs the concept of object versioning. In this concept, a new object version is created when the object's attribute values change, whereas otherwise the same object version is shared among versions. It does so with the use of Linked Data. By updating the attached properties of components in the RDF dataset and subsequently creating an output IFC file of these components, the current and actual as-maintained version of the building and its building elements is secured. However, apart from the tool created in this thesis, there are no software applications available with sufficient IFC and RDF compatibility yet, though it is expected to be the direction the market is moving towards in the near future.

Finally, based on the answers to the sub-research questions, an answer can be formulated to the main research question: "How can changing information in the Operation & Maintenance phase be handled efficiently using 3D Building Information Models and Semantic Web technologies?".

Firstly, IFC models can be semantically enriched in order to improve the management of information in the O&M phase. By applying semantic web technologies, the tabular O&M data is transformed to RDF and linked to objects in an IFC file. This has resulted in a prototype application where the model and associated properties can be viewed. This application then demonstrates how the properties of elements in an IFC file can be altered so that the data is constantly up-to-date, thus ruling out risks of versioning. Usage of Semantic Web technologies makes this improvement possible as the data elements are made available as unique identifiers which can be addressed and accessed by the application. The addition of a single triple indicates the timeliness of information. This enables the capturing of knowledge about these elements in an easy and retraceable way.

The relevance of the outcomes of this research lies in the tremendous amount of data and models available in current AEC/FM projects, and the amount of annual service requests in a building which entails risks of versioning. Scientific research shows the trend of transforming building information into a 'web of data'. This trend includes implementing semantic web technologies, where this thesis contributes to further industry adoption by providing a solution for one of the major problems in BIM implementation in O&M.

The societal relevance of this research concerns importance of obtaining and maintaining high-quality buildings in an economic, efficient and effective way where this research has brought forward benefits in order to do so. While the proposed approach seems quite promising, a shift in technologies is necessary. However, the FM industry is considered rather traditional. Therefore the initiative for a shift as proposed in this thesis should come from building owners by stimulating innovations through performance-based contracts.

9 Recommendations

This chapter proposes a number of recommendations resulting from this research. These can be divided in company- and sector-related ones and recommendations providing an outlook for further research and development.

9.1 Company recommendations

The company recommendations described in this section can be viewed as recommendations for the AEC and FM industry in working with BIM and aligning informational requirements for that.

First and foremost, for FM companies that consider implementing BIM it is crucial to know what goals to strive towards through the use of BIM. Setting up a BIM-protocol can help to define this goal, as well as other aspects which the interviews have demonstrated as problematic in the current situation. Such aspects include roles, rights and responsibilities of actors involved in the BIM process, legal aspects and modelling agreements.

Secondly, it is recommendable to define a standard information requirement. Because asset managers will gain a more directing role with BIM, they will process information instead of producing it. To be able to simply and uniformly process information, the asset manager needs to indicate to internal as well as external partners which information is necessary, in which format this needs to be saved and how it should be handed to them. Defining and specifying these requirements is necessary. Ideally, the asset manager is already involved in the early stages of a project in order to oversee model development and to ensure that the informational requirements for O&M are incorporated in the model.

A last recommendation relates to the standardization of information in general. In line with the recommendation regarding information requirements, information in general should be standardized as well. The implementation of object type libraries (OTL) can improve this. Some AEC companies have already defined OTLs to standardize their way of working. However, the way these libraries are built up varies greatly. A library based on open standards might be useful to address this variety, because e.g. the Semantic Web standard enable the user to address many forms of data simultaneously. Storing all building elements in such a library along with their properties and characteristics, thus creating a connected database, has the potential to enhance efficiency for asset managers. With this library as main connection towards each discipline, data can be combined to be able to plan maintenance, make simulations and communicate efficiently towards suppliers. In case this library is reusable and extendable using open standards, developing this library can benefit from continuous improvement. This improvement favourably occurs stepwise.

9.2 Further research

Due to the scope of this research, as well as the limited timeframe, not every relevant development can be taken into account in this research. In order to encourage utilizing this potential, further research can be recommended based on the obtained knowledge.

Firstly, more extensive model testing of the tools is required to guarantee optimal stability. This includes testing with an IFC model where classifications are used such as the NL-SfB, NEN 2767 or STABU classification. These will then probably be stored in the IFC concepts *IfcRelAssociatesClassification* and/or *IfcClassification.* Furthermore, in this research only company-specific data was used. In line with the recommendation mentioned in the previous section, a standard information input can be designed. In addition, the process is only validated based on one BIM which was created using Revit.

Secondly, while a working and usable tool was created, further development of this tool will result in significant benefits. Further development can be achieved through compatibility with more vocabularies and datasets that have been made available as Linked Open Data. Such datasets range from sensor data and facility condition data to occupant data. It is assumed that visualisation of sensor data or condition-related information provides great advantage for O&M managers. In addition, in line with the use case diagram provided in section 7.3, different rights can be provided for different roles. For example, the asset manager as well as contractors need to be able to login to the system, whereas the contractor might not have as many rights with regard to the altering of information as the asset manager.

Thirdly, apart from keeping connected data up-to-date in a BIM, research can be conducted into the ways the IFC model can be kept up-to-date as well. However, given that the IFC model in the framework of this research is more static than the more dynamic data sets, updating the IFC files needs to be secured in contracts. For example, when a contractor is hired to perform renovations to a building, contracts may need to oblige the contractor to update the IFC as well in order for the model to represent the actual as-used situation.

Fourthly, the asset manager may benefit greatly from additional research into the use of BIM and how several types of analysis can be done from a BIM. Examples of such analysis include energy usage, maintenance planning and the condition of building elements.

Finally, while the concept of versioning is at the forefront of this thesis, versioning by dereferenceable URIs can be further explored. These so-called dereferenceable URIs would be designed to point towards the latest O&M data or a single specific version of the concept library. The DURAARK initiative has proposed this approach to create a digital semantic archive for long-term data preservation (Beetz, et al., 2014).

10 BIBLIOGRAPHY

Aedes Werkgroep BIM. (2016). *Regie over informatie: BIM-protocol voor woningcorporaties.* Den Haag: Aedes.

Allemang, D., & Hendler, J. (2011). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL* (2nd ed.). Waltham, MA, USA: Morgan Kaufmann Publishers.

Becerik-Gerber, B., Jazizadeh, F., Li, N., & Calis, G. (2012). Application Areas and Data Requirements for BIM-Enabled Facilities Management. *Journal of Construction Engineering and Management*, 431-442.

Beetz, J. (2009). *Facilitating distributed collaboration in the AEC / FM sector using Semantic Web technologies.* Eindhoven, The Netherlands: Eindhoven University Press.

Beetz, J., Lindlar, M., Dietze, S., Gadiraju, U., Edvardsen, D., & Bjorkhaug, L. (2014). *D.3.3.2 - Ontological Framework for a Semantic Digital Archive.* DURAARK.

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 Manufactering*, 89-101.

Berners-Lee, T. (2006). *Linked Data - Design Issues*. Retrieved September 23, 2016, from W3C.org: https://www.w3.org/DesignIssues/LinkedData.html

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 28-37.

Berners-Lee, T., Shadbolt, N., & Hall, W. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems*, 96-101.

BIM Loket. (2015). *Wat zijn open standaarden?* Retrieved September 15, 2016, from BIM Loket: http://www.bimloket.nl/116

Bosch, A. (2014). *Beheren 2.0: Beheren van bouwwerken met BIM.* Delft: TU Delft.

buildingSMART. (2008). *IFD Library.* Retrieved from IFD developers wiki: https:///www.unetovni.nl/cms/streambin.aspx?requestid=EA0F4F6F-560D-4D15-9BD6-0B651404B812

buildingSMART. (2011). *Information Delivery Manuals*. Retrieved September 16, 2016, from buildingSMART.org: http://iug.buildingsmart.org/idms

BuildingSMART. (2012). *The BIM Evolution Continues with OPEN BIM.* Retrieved July 5, 2016, from http://bimie.nl/images/bestanden/OPEN_BIM_ExCom_Agreed_Description_20120131.pdf

buildingSMART. (n.d.). *Linked Data*. Retrieved September 27, 2016, from buildingSMART-tech.org: http://www.buildingsmart-tech.org/future/linked-data

buildingSMART. (n.d.). *Specifications*. Retrieved September 16, 2016, from buildingsmart-tech.org: http://www.buildingsmart-tech.org/specifications

Comité Européen de Normalisation. (2006). Facility Management - Part 1: Terms and definitions. In *EN 15221-1* (pp. 3-14). Brussels. Retrieved from http://www.bifm.org.uk/bifm/about/facilities

Costa, G., & Madrazo, L. (2015). Connecting building component catalogues with BIM models using semantic technologies: an application for precast concrete components. *Automation In Construction*, 239-248.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., & O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics*, 206-219.

Dankers, M., van Geel, F., & Segers, N. (2014). A web-platform for linking IFC to external information during the entire lifecycle of a building. *12th International Conference on Design and Decision Support Systems in Architecture and Urban Planning, DDSS 2014* (pp. 138-147). Eindhoven: Elsevier Procedia.

DSDM Consortium. (2008). *DSDM Atern Handbook.* Retrieved October 14, 2016, from DSDM Consortium: https://www.dsdm.org/content/moscow-prioritisation-0

Duncan, J., & Humphreys, G. (1989). Visual search and stimulus similarity. *Psychology Review*, 433-458.

Eadie, R., Browne, M., Odeyinka, H., McKeown, C., & McNiff, S. (2013). BIM implementation throughout the UK construction project lifecycle: An analysis. *Automation In Construction*, 145-151.

Eastman, C., Teicholtz, P., Sacks, R., & Liston, K. (2008). *BIM Handbook* (2nd ed.). Hoboken, New Jersey: John Wiley & Sons, Inc.

ETIM. (n.d.). *Classification | Introduction*. Retrieved September 27, 2016, from ETIM: http://www.etim-international.com/classification/introduction

Flouris, G., Fundulaki, I., Pediaditis, P., Theoharis, Y., & Christophides, V. (2009). Coloring RDF Triples to Capture Provenance. *The Semantic Web - ISWC 2009* (pp. 196-212). Chantilly, VA, USA: Springer Berlin Heidelberg.

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* (pp. 3058-3065). Stresa, Lake Maggiore, Italy: CRC Press.

Grilo, A., & Jardim-Goncalves, R. (2010). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*, 522-530.

Hallberg, D., & Tarandi, V. (2010). On the use of Open BIM and 4D visualisation in a predictive life cycle management system for construction works. *Journal of Information Technology in Construction*, 445-466.

Halpin, H., & Cheney, J. (2011). Dynamic Provenance for SPARQL Updates Using Named Graphs. *TaPP'11* (pp. 4:1-4:6). USENIX Association.

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

Horner, R., El-Haram, M., & Munns, A. (2001). Building maintenance strategy: a new management approach. *Journal of Quality in Maintenance Engineering, 3*(4), 273-280.

IMDb. (n.d.). *Breaking Bad (TV Series 2008-2013)*. Retrieved September 26, 2016, from IMDb: http://www.imdb.com/title/tt0903747/?ref_=ttfc_fc_tt

Institute of Electrical and Electronics Engineers. (1990). *IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries.* New York, NY.

JalyZada, A., Tizani, W., & Oti, A. (2014). Building Information Modelling (BIM) - Versioning for collaborative design. *International Conference on Computing in Civil and Building Engineering 2014* (pp. 512-519). Orlando, FL, USA: American Society of Civil Engineers (ASCE).

Kiviniemi, A., & Codinhoto, R. (2014). Challenges in the Implementation of BIM for FM - Case Manchester Town Hall Complex. *International Conference on Computing in Civil and Building Engineering 2014* (pp. 665-673). Orlando, FL, USA: American Society of Civil Engineers (ASCE).

Kondylakis, H., & Plexousakis, D. (2014). Exploring RDF/S Evolution using Provenance queries. *Workshop Proceedings of the EDBT/ICDT 2014 Joint Conference* (pp. 176-181). Athens, Greece: CEUR.

Laakso, M., & Kiviniemi, A. (2012). The IFC standard - a review of history, development and standardization. *Journal of Information Technology in Construction*, 134-161.

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words? *Journal of Cognitive Science*, 65-100.

Liu, Y., van Nederveen, S., & Hertogh, M. (2016). Understanding effects of BIM on collaborative design and construction: An empirical study in China. *International Journal of Project Management*, n.d.

Love, P., Simpson, I., Hill, A., & Standing, C. (2013). From justification to evaluation: building information modelling for asset owners. *Automation in Construction*, 208-216.

MacLeamy, P. (2014). *The Promise of BIM.* Retrieved November 1, 2016, from BIMForum: http://bimforum.org/2014/04/28/boston-bimforum-presentations-2/

McGlinn, K. (2015). *CSVW Tutorial.* Retrieved from Trinity College Dublin: http://phaedrus.scss.tcd.ie/buildviz/csvw/CSVW%20Tutorial.pdf

Meinhardt, P. (2015). *Versioning Linked Datasets: Towards Preserving History on the Semantic Web.* Universität Potsdam. Potsdam: Hasso-Plattner Insitut.

Ministry of Internal Affairs. (2012). *Bijlage I: Brandmeldinstallaties*. Retrieved January 19, 2017, from Bouwbesluit Online 2012: https://rijksoverheid.bouwbesluit.com/Inhoud/docs/wet/bb2012_nvt/artikelsgewijs/hfd6/afd6-5/art6-20

Muz, R. (2014). *Extending Building Information Modeling beyond Design/Construction project phase*. Retrieved September 15, 2016, from Graduation Presentation by Rafal Muz on Prezi: https://prezi.com/q45y2ogmdnz2/graduation-presentation/

Mynarz, J. (2013). *Capturing temporal dimension of linked data*. Retrieved November 16, 2016, from blog.mynarz.net: http://blog.mynarz.net/2013/07/capturing-temporal-dimension-of-linked.html

National Institute of Building Sciences. (2007). *National Building Information Modeling Standard.* n.d.: National Institute of Building Sciences.

NEN. (2012). *Conditiemeting*. Retrieved February 1, 2017, from NEN: https://www.nen.nl/NEN-Shop/Conditiemeting.htm

NEN. (2015). *Norm voor beheer, controle en onderhoud van brandmeldinstallaties herzien*. Retrieved January 19, 2017, from NEN: https://www.nen.nl/NEN-Shop/Bouwnieuwsberichten/Norm-voor-beheer-controle-en-onderhoud-van-brandmeldinstallaties-herzien.htm

Nour, M., & Beucke, K. (2010). Object Versioning as a basis for design change management within a BIM context. In W. Tizani (Ed.), *Computing in Civil and Building Engineering,* *Proceedings of The International Conference* (pp. 147-152). Nottingham, UK: Nottingham University Press.

Noy, N., & McGuinness, D. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology.* Stanford, CA, USA: Stanford knowledge systems laboratory technical report KSL-01-05.

Obitko, M. (2007). *Introduction to Ontologies and Semantic Web*. Retrieved September 23, 2016, from Semantic Web Architecture: http://obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html

Oraskari, J., & Törmä, S. (2015). RDF-based signature algorithms for computing differences of IFC models. *Automation In Construction*, 213-221.

Overbeek, H., & van den Brink, L. (2013). *Boek/URI-strategie*. Retrieved December 2, 2016, from Platform implementatie Linked Open Data: http://www.pilod.nl/wiki/Boek/URI-strategie#URI-patroon

Pauwels, P. (2014). Supporting Decision-Making in the Building Life-Cycle Using Linked Building Data. *Buildings, 3*, 549-579.

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation In Construction*, 100-133.

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation In Construction*, 100-133.

Petrinja, E., Stankovski, V., & Turk, Z. (2007). A provenance data management system for improving the product modelling process. *Automation in Construction*, 485-497.

Plessers, P., & De Troyer, O. (2005). Ontology Change Detection using a Version Log. *4th International Semantic Web Conference, ISWC* (pp. 578-592). Galway, Ireland: Springer Berlin Heidelberg.

Potkany, M., Vetrakova, M., & Babiakova, M. (2015). Facility Management and Its Importance in the Analysis of Building Life Cycle. *4th World Conference on Business, Economics and Management, WCBEM* (pp. 202 – 208). Kuşadası: Elsevier.

Radulovic, F., Poveda-Villalón, M., Vila-Suero, D., Rodríguez-Doncel, V., García-Castro, R., & Gómez-Pérez, A. (2015). Guidelines for Linked Data generation and publication: An example in building energy consumption. *Automation in Construction*(57), 178-187.

Roper, K., & Payant, R. (2014). *The Facility Management Handbook* (4th ed.). New York: AMACOM.

Sapp, D. (2015). *Facilities Operations & Maintenance*. Retrieved August 26, 2016, from https://www.wbdg.org/om/om.php

Schevers, H., Mitchell, J., Akhurst, P., Marchant, D., Bull, S., McDonald, K., & Drogemuller, R. (2007). Towards digital facility modelling for Sydney opera house using IFC and semantic web technology. *Journal of Information Technology in Construction*, 347-362.

Schrijvers Technische Installaties. (2016). *Ketsheuvel Den Bosch*. Retrieved October 17, 2016, from Schrijvers Technische Installaties: http://sti-oss.nl/projects/herbestemming-kantoor-zayaz-te-s-hertogenbosch/

Solihin, W., Eastman, C., & Lee, Y. (2016). A framework for fully integrated building information models in a federated environment. *Advanced Engineering Informatics*, 168-189.

STABU. (n.d.). *Elementenmethode NL/SfB*. Retrieved September 27, 2016, from STABU: http://www.stabu.org/diensten/nlsfb/#.V-psajUXdsk

Sullivan, G., Pugh, R., Melendez, A., & Hunt, W. (2010). Chapter 5 Types of Maintenance Programs. In *Operations & Maintenance Best Practices* (pp. 5.1-5.9). U.S. Department of Energy.

Theeuwen, W., & Smit, P. (2016). *BIM biedt business voor beter beheer.* Hogeschool Utrecht, Instituut voor Gebouwde Omgeving. Utrecht: Roeleveld-Sikkes Architects.

Tunnicliffe, S., & Davis, I. (2005). *Changeset*. Retrieved October 12, 2016, from vocab.org: http://vocab.org/changeset/schema.html

van der Riet, M. (2016). *Semantic Model Enrichment for BIM-Enabled Risk-Based Operation and Maintenance.* Eindhoven: Eindhoven University of Technology.

Vanlande, R., Nicolle, C., & Cruz, C. (2008). IFC and building lifecycle management. *Automation In Construction*, 70-78.

Verbaan, W., van der Voet, L., de Boer, J., Visser, E., & de Koe, D. (2014). *Beter Beheer Met BIM: van informatiemodel naar informatiemanagement.* Amsterdam: Hogeschool van Amsterdam.

Visser, E., de Boer, J., & van der Voet, L. (2013). *BIM in het beheer biedt kansen.* Amsterdam: Hogeschool van Amsterdam.

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.

W3C. (2006). *RDF Validation Service*. Retrieved January 24, 2017, from W3C: https://www.w3.org/RDF/Validator/rdfval

W3C. (2007). *RDF Primer - Turtle Version*. Retrieved October 11, 2016, from W3C.org: https://www.w3.org/2007/02/turtle/primer/

W3C. (2008). *Cool URIs for the Semantic Web*. Retrieved December 14, 2016, from https://www.w3.org/TR/cooluris/#cooluris

W3C. (2008). *SPARQL Query Language for RDF*. Retrieved September 26, 2016, from W3C Recommendation: https://www.w3.org/TR/rdf-sparql-query/#QueryForms

W3C. (2015). *Ontologies*. Retrieved September 23, 2016, from W3C: https://www.w3.org/standards/semanticweb/ontology

Ware, C. (2004). *Information Visualisation.* USA: Morgan Kaufmann.

Wetzel, E., & Thabet, W. (2015). The use of a BIM-based framework to support safe facility management processes. *Automation In Constuction*, 12-24.

Young Jr., N., Jones, S., Bernstein, H., & Gudgel, J. (2009). *The Business Value of BIM: Getting Building Information Modelling to the bottom line.* Retrieved from SmartMarket Report: http://images.autodesk.com/adsk/files/final_2009_bim_smartmarket_report.pdf

Zayaz. (2016). *Aanpak Ketsheuvel 12a*. Retrieved October 17, 2016, from Zayaz.nl: http://www.zayaz.nl/Onze_projecten/Zayaz_ver_bouwt/Herontwikkeling/Ketsheuvel_12a

Zhao, J., Bizer, C., Gil, Y., Missier, P., & Sahoo, S. (2010). *Provenance Requirements for the Next Version of RDF.* Retrieved October 6, 2016, from W3C.org: https://www.w3.org/2009/12/rdf-ws/papers/ws08

11 APPENDICES

Appendix I – Expert interviews

Appendix II – Python script tool

Appendix III – Flowcharts of both parts of tool

Appendix IV – Graphical User Interfaces (GUIs)

Appendix V – Ontologies (.ttl)

Appendix VI – Partial input (.csv)

Appendix VII – Partial RDF input

Appendix I: Expert interviews

Name:	Company:	Date:

General

1. Can you indicate what the core business of your organization is?
2. What is your role in the organization, and with what organizations do you often cooperate?

Experience with BIM

3. Are you familiar with BIM?
    a. If yes, how would you describe BIM?
4. Does your organization work with BIM?
5. Which tangible benefits have you experienced through BIM implementation?
6. Which bottlenecks do you foresee regarding the transition to BIM in O&M?

Information management

7. How are building files handled in your organization nowadays?
    a. How much of the information is hardcopy (%)?
    b. If digital, how? (e.g. folders on server/in FMIS)
8. Do you have all of the information you need in executing your tasks available at all times?
    a. If not, which information not/why not?
    b. How many of the up-to-date information is with suppliers?
9. Which management information do you now posess, and which would you like to have?
10. In which manner do you ensure that information remains up-to-date?
    a. In case of changing information (architectural, MEP, etc.), how are these changes processed?

Information and BIM

11. Do you think linking information/files to a BIM will have potential in the O&M field?
    a. If not, why not?
    b. If yes, which files would you definitely link to a BIM?
        i. E.g. user manuals, warranty documents, product information on supplier's website, product data sheet, condition assessment report, photos, etc. …
12. Do you think adding new attributes/linking files in a 3D environment will be useful when structuring information?
13. Do you think IFC model files are adequate acting as a BIM to which other information is linked?
14. What features should an application used to disclose building information in O&M and process changes in models and/or information, certainly have? (e.g. automatic file linking, querying linked files)? Could you describe a related use case?

Changes to a building

15. Do you have experience with maintenance in a building?
    a. Could you describe a related work process?
16. When replacing components or otherwise changing installations, in what way do you notify the client (e.g. phone call/document/extensive logs/etc.)?
    a. If not at all, who does?

Questions specifically for asset managers
Questions specifically for suppliers

Appendix I.I: Summarized interviews (Dutch)

Naam: Gerrie Mühren	**Bedrijf:** Autodesk	**Datum:** 19-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Wat wij proberen te bereiken: als je iets wil zoeken, kom je bij Google uit. Als je iets wil kopen, kom je bij Amazon. Als je iets wil maken, kom je bij Autodesk. Hoe wij als Autodesk in de wereld willen staan is niet als leverancier van software maar een toegevoegde waarde bieden: "the future of making things".

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Projectmanager bij Autodesk Consulting. Niet direct 'into' de software, maar begeleid aannemers door te luisteren naar wensen en plaats daar de juiste mensen bij. Begeleiden van dit proces.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Buitengewoon Irritant Modewoord. Grootste bouw-ontwikkeling sinds jaren, maar helaas een modewoord geworden laatst waardoor de betekenis helaas verwatert.

4. Werkt uw organisatie met BIM?

Ja, levert software om BIM mogelijk te maken (Revit, Navisworks, Inventor, etc.)

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Visualisatie (hololens). Bijvoorbeeld in een schoolgebouw kunnen de schoonmakers inlezen welke lokalen intensief gebruikt worden, en de lokalen die niet gebruikt zijn hoeven dus ook niet worden schoongemaakt. Dit maakt het dubbel zo effectief. Een ander voorbeeld is als er lekkages optreden in een ziekenhuis, wordt vaak een afdeling afgesloten en complete plafonds opengebroken moeten worden. Tegenwoordig zet je een 3D-bril op, en kijk je in je model waar je leidingen en knooppunten zitten. Zo kun je gericht onderhoud uitvoeren.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

De mens: "Wij doen niet aan BIM, dus daaag." Je moet het voor de mensen bereikbaar maken. Voor iedereen kan het voordelen hebben, maar je moet die voordelen wel duidelijk communiceren naar de mensen.

Informatie en BIM

7. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. ~~Zo nee, waarom niet?~~
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. **Gebruikershandleidingen**, **garantietermijnen**, **productinformatie** op website van de leverancier, productinformatie document, conditierapporten, foto's…

Wetgeving is essentieel, alsmede informatie m.b.t. veiligheid. Door data te koppelen wordt het proces een stuk efficiënter. Daarnaast een voorbeeld op het gebied van operations: in een bank wordt een kluis geïnstalleerd op de tweede verdieping, maar hoe komt die daar?

8. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Ja, daar geloof ik zeker in. Het is gewoon al heel fijn als je installatie-informatie kunt koppelen aan je model. Stel je komt iets tegen waarvan je denkt "Ho eens, dat is levensgevaarlijk zoals dat daar staat." Dan kun je dit direct terugkoppelen en veranderen.

9. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Ja, maar hier moet ik wel een kanttekening bij plaatsen. Moeten dit per se IFC-modellen zijn. Je moet niet heel veel verstand van IFC hoeven hebben, maar je moet ermee kunnen werken. Het moeten open standaarden zijn, maar je moet er niet in hoeven duiken omdat je dan mensen kwijtraakt in dit denkproces. Niet iedereen hoeft programmeerkennis te hebben om software te maken, en om IFC te gebruiken hoeft niet iedereen kennis te hebben van IFC.

10. Acht u Semantic Web-technieken geschikt veranderende informatie bij te houden in een BIM-model voor beheer?

Ik heb hier weinig verstand van, maar het is een gegeven dat IFC is gebaseerd op de STEP-technologie van de jaren 80 van de vorige eeuw. En dat is eigenlijk te gek voor woorden. OWL zou hier een rol kunnen spelen, maar verder dan dat kan dus ik niet beoordelen of het hiervoor geschikt is maar ik ben het ermee eens dat je hierin mee moet gaan.

11. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Het zou goed zijn om hele specifieke productinformatie te kunnen inzien in een dergelijk model. In de wereld van brandveiligheid gaat er hierin een hele wereld open, omdat brandmelders tegenwoordig ook een stuk geavanceerder zijn dan vroeger. Hiervoor is informatie op het gebied van compartimentering ook interessant.

Naam: Gerard Alferink	**Bedrijf:** Triade	**Datum:** 20-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Wij als Triade leveren zorg en huisvesting aan cliënten met een verstandelijke beperking. Dit verschilt van cliënten die altijd zorg nodig hebben tot cliënten die werk hebben en waar af en toe dagbesteding voor wordt geregeld.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Projectleider huisvesting. In feite doen wij nieuwbouwprojecten en grote renovaties aan panden. Onze organisatie werkt vaak met aannemers, installateurs, onderhoudspartijen, adviseurs, andere contractpartners; de 'usual suspects' op het gebied van bouwen eigenlijk.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ik heb ervan gehoord, maar nog niet gezien hoe het in de praktijk werkt. Ik ben op de hoogte van de toepassing in het ontwerpproces, maar daar houdt het op.

4. Werkt uw organisatie met BIM?

Nee.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Het grootste voordeel waar ik van gehoord heb (maar niet aan den lijve ondervonden) is 3D coor-dinatie. Maar geen tastbare voordelen zelf ondervonden.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

De organisatie moet eraan toe zijn. Je moet je informatiebeheer op orde hebben, voordat je je aan BIM wil gaan wagen. Dat is bij ons nog lang niet zo.

Informatiebeheer

7. Hoe zijn gebouwdossiers in uw organisatie nu geregeld?
    a. Hoeveel van de documenten/informatie is hardcopy (%)?
    b. Indien digitaal, hoe? (bijv. mappen op een server/in systemen)

Wij hebben hiervoor een digitaal systeem in werking, WISH. Dat is een programma dat diverse modules heeft waarin je vastgoedregistratie kan worden vastgelegd wat je basis wordt voor beheer & onderhoud. Hier kun je weer modules aan hangen voor contractbeheer, tekeningen, etc.. Daarnaast zitten er nog documenten op de netwerkschijf, dus niet al het digitale zit in dit WISH-systeem. Contracten zitten er redelijk in, tekeningen nog nauwelijks. Ik schat dat de informatie die nu hardcopy is ongeveer 60% beslaat tegenover 40% digitaal.

8. Is alle informatie die u nodig hebt in uitvoering van uw taken altijd beschikbaar?
    a. Zo niet, welke informatie niet/waarom niet?
    b. Hoeveel van de actuele informatie is bij leveranciers?

Nee, niet alle informatie is beschikbaar. Tekeningen zijn vaak niet actueel. Ook qua contracten zie je vaak dat er alleen een standaard te vinden is, maar geen bijgewerkte voorwaarden of bijbehorende contracttekeningen oid. En als de informatie er al is, schort het ook vaak aan de structuur wat het moeilijk maakt ze te zoeken.

9. Over welke managementinformatie beschikt u nu en waarover zou u willen beschikken?

Ik ben geen manager, maar de managers beschikken over ‘business intelligence’. Voor projecten gebruiken wij dit ook, met daarnaast projectbegrotingen en aanvullende projectinformatie. Hiermee kunnen we hier ook op sturen. Ik kan niet goed zeggen wat er nog mist omdat ik geen manager ben.

10. Op welke wijze borgt u dat informatie actueel blijft? (onderhouden van informatie)
    a. In geval van veranderende informatie (bouwkundig, installaties, etc), hoe worden deze wijzigingen verwerkt?

Revisies moeten worden geborgd. Ik schat dat dit bij ons nu in 50% van de gevallen gebeurd. Bij updates van contracten wordt dit aangepast en gaat het oude contract eruit in het WISH-systeem. Als ik een project oplever, dan zet ik alle revisiestukken vanuit het project naar een beheersmap.

Informatie en BIM

11. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. Zo nee, waarom niet?
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto’s…

Ik kan niet goed inschatten hoe dit zou werken. Maar het zou zeker helpen om het inzichtelijker te maken. Voorwaarde is wel dat ze actueel blijven.

12. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Zou zeker helpen bij het aanbrengen van structuur inderdaad, brengt zoektijd terug.

13. Acht u IFC-modellen geschikt als middelpunt waar relevante informatie gekoppeld wordt?

Geen ervaring met IFC, dus kan hier geen nuttig antwoord op geven.

14. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Ik kan me voorstellen dat als je zoiets hebt, dat je op een pand klikt en alle bewonersinformatie kunt zien. Dat je contracten, je tekeningen etc. meteen kunt zien per pand. In WISH gaat dit per module, deze haken niet echt op elkaar in. Daarnaast moet je rechten kunnen instellen per gebruiker (lezen, schrijven, etc). Ook zou het handig zijn om verzameldocumenten op thema te kunnen opknippen oid.

Naam: Paul Noom	**Bedrijf:** Kropman Installatietechniek	**Datum:** 20-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

De core business ligt in de bouw, het realiseren van de installaties in gebouwen. In de kern zijn we installateur. Ontwerp & Techniek is hier een onderdeel van maar uiteindelijk draait het om bouwen en onderhouden.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Landelijk manager van de afdeling Ontwerp & Techniek (O&T) binnen Kropman, en wij zitten veel in nieuwbouwprojecten. Daarnaast hebben wij ook een rol in het service-traject, wat meer richting onderhoud gaat. We werken vaak met aannemers, productontwikkelaars, engineering; we zijn de spin in het web eigenlijk. We werken ook veel direct met klanten in het geval van aanbestedingen.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ik ben er zeker bekend mee, heb binnen Kropman een aantal jaar geleden het eerste 'BIM-project' gedaan met Revit. We tekenen wel al jaren in 3D, wat een onderdeel van BIM is. Modelleren doen we op het moment ook in Techline. BIM zie ik als een mooie ontwikkeling. Je merkt wel dat je eerder geconfronteerd wordt om keuzes te maken, en daar moet je als bedrijf op ingericht zijn.

4. Werkt uw organisatie met BIM?

Ja, we zijn in tegenstelling tot sommige andere bedrijven nog niet zover dat we BIM-processen in werking hebben. De I in BIM, daar lopen we nog wel in achter. Het is tegenwoordig lastig om goede modelleurs aan te trekken. De service-tak moet eraan toe komen, maar die zijn zeker nog niet zover.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Clashdetectie is een erg groot voordeel. Het geeft duidelijker en overzichtelijker knooppunten aan in het ontwerp vergeleken met 2D. Inzicht is ook zeker gebaat bij BIM. Dit kan in de toekomst helemaal groeien, waarbij de uitvoerder met een iPad of met een Hololens de bouwplaats opgaat.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Het kennisniveau van mensen is een groot knelpunt in de omschakeling naar BIM. Voor het maken van 2D tekeningen vroeg men meestal MBO'ers, maar je ziet dat een modelleur moet opereren op een hoger kennisniveau, is eigenlijk engineer. Dit is een van de hoofdknelpunten. Het up-to-date houden van je model is een tweede. Daarnaast zie je dat opdrachtgevers, als ze al BIM gebruiken, vaak niet weten wat ze ermee willen. Je kunt wel een model uitvragen voor beheer & onderhoud op bijvoorbeeld LOD 500, maar als je niet weet wat je ermee wil doen heb je er nog niets aan.

Wijzigingen in een gebouw

7. Heeft u ervaring met het onderhoud van gebouwen?
    a. Kunt u een dergelijk werkproces beschrijven?

De afdeling O&T bestrijkt eigenlijk alle facetten, dus zeker ervaring mee. Het komt voor dat we een ontwerp maken, dit is uitgevoerd en dat er tijdens beheer klachten zijn. Dan analyseren we het probleem en lossen we dit op. Ook komt Servicewel eens met de vraag 'hoe heb je dit bedacht?' Daarbij krijgen we terugkoppeling over wat fout ging. We leveren hierbij ook input aan MJOP's.

8. Bij het vervangen van componenten (of anderszijds aanpassen van installaties), hoe stelt u de opdrachtgever hiervan op de hoogte? (vb. telefoontje/document/uitgebreide log/etc.)
    a. Zo nee, wie doet dit wel?

Sowieso krijgt de opdrachtgever een factuur. Daarnaast hangt het af van het type contract. Je hebt contracten waarbij als er wat stukgaat we langsgaan. Daarnaast hebben we prestatie-contracten. Hier wordt onderhoud op preventieve basis uitgevoerd. Sowieso wordt altijd de opdrachtgever op de hoogte gesteld. Dit kan op basis van een telefoontje, maar op eigen initiatief stellen we vaak een lijst op met wat veranderd is.

Informatie en BIM

9. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. ~~Zo nee, waarom niet?~~
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Jazeker, al is het heel erg afhankelijk van wat de klant ermee wil. Modellen worden snel heel groot/zwaar. Geometrie is belangrijk, maar vooral de informatie die je eraan hangt. De voorbeelden zijn allemaal goed. Los daarvan zou een automatische koppeling een voordeel zijn.

10. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Ja, maar de gebruiker moet hier wel mee om kunnen gaan. De meesten zijn nog niet zo ver zodat ze wat uit een BIM-model of Revit kunnen halen. Hier komt het eerder genoemde kennisniveau weer terug. Dit is eventueel te verbeteren met een gebruiksvriendelijke interface. Al zou dit zeker helpen bij het structureren van informatie omdat alles in een model is opgeslagen.

11. Acht u IFC-modellen geschikt als middelpunt waar relevante informatie gekoppeld wordt?

Ja heel erg. Hiermee dwing je andere partijen niet om dezelfde software-pakketten ook te kopen als die jij gebruikt (of andersom). Al moet er wel een protocol duidelijk zijn hoe hiermee om te gaan, want anders gaat veel data verloren bij de omzet naar IFC en weer terug.

12. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Het zou makkelijk zijn om een koppeling te hebben tussen je model en een database die allebei de kanten op werkt. Zo gauw een pomp vervangen wordt en verwerkt in de database, zou er een melding in je model moeten verschijnen die zegt dat het type pomp is veranderd.

Naam: Bert van Hulsen	**Bedrijf:** PostNL	**Datum:** 21-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Het verwerken van post, zowel brieven als pakketten als ongeadresseerde post.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Ik ben verantwoordelijk voor het beheer en onderhoud van gebouwgebonden installaties. Wij zijn een regieorganisatie, wat betekent dat één iemand verantwoordelijk voor de installaties is; dat ben ik. Ik werk vaak met de installateurs, adviesbureaus, aannemers en andere leveranciers.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ja, en ik zou het omschrijven als een doorontwikkeling van tekenmethodieken en informatie-technologie. Omdat er altijd plat getekend werd, zorgde dit voor knelpunten tussen installateurs en bouwkundigen. Je krijgt met BIM beter inzicht in je ontwerp, en op een andere manier werken.

4. Werkt uw organisatie met BIM?

Nee, ik heb me daar wel in verdiept. Maar omdat onze organisatie met name krimpende is, en onze gebouwen vaak bestaande zijn en weinig “spannend” zijn qua structuur, twijfel ik of het voor ons een waardevolle ontwikkeling is.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Ikzelf niet, maar zit ook bij de ISSO in contactgroepen. Daar wordt voornamelijk gesproken over de waarde van coordinatie en zogenaamde “clashdetectie”.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Ik moet gokken, maar ik denk dat een van de uitdagingen is om ervoor te zorgen dat de informatie goed gedefinieerd is. Anders krijg je een overkill aan informatie. Daarnaast is uniformiteit een issue, in b&o wordt vaak de NL-SfB codering gebruikt, of STABU, en het Rijksvastgoedbedrijf gebruikt dan weer vaak Rgd-BOEI. Het moet beheersbaar en gestructureerd zijn.

Informatiebeheer

7. Hoe zijn gebouwdossiers in uw organisatie nu geregeld?
    a. Hoeveel van de documenten/informatie is hardcopy (%)?
    b. Indien digitaal, hoe? (bijv. mappen op een server/in systemen)

Bijna alles is digitaal op het moment. Wij werken met een sharepoint-achtige omgeving, geen vaste werkplekken meer. Daar staat alle informatie in, van huurovereenkomsten, koopcontracten, bouwgegevens, naar tekeningenbeheer, verbouwingen, noem het maar op. Alle informatie die betrekking heeft op een pand wordt gesorteerd per pand. 10-20% is hardcopy, maar alleen als het voor het bevoegd gezag nodig is. Tekeningen brengen we bij Vgib onder.

8. Is alle informatie die u nodig hebt in uitvoering van uw taken altijd beschikbaar?
    a. ~~Zo niet, welke informatie niet/waarom niet?~~
    b. Hoeveel van de actuele informatie is bij leveranciers?

Ja, omdat dit altijd vanuit de sharepoint digitaal is op te vragen. AutoCAD is hierop een uitzondering, daarvoor hebben we een paar laptops staan. Leveranciers hebben ook indien nodig toegang tot een webportal van deze sharepoint.

9. Over welke managementinformatie beschikt u nu en waarover zou u willen beschikken?

Voor projecten is er regelmatig overleg, waar verslagen uitrollen met informatie. Met onze leveranciers die regelmatig onderhoud doen op locatie is meestal 1x per maand overleg. Daarnaast kijken we naar planningen, hoe verloopt communicatie, financiële afwikkeling. Dit is voldoende om onze eigen organisatie te informeren. Mis hierin weinig, anders huur ik adviesbureau in.

10. Op welke wijze borgt u dat informatie actueel blijft? (onderhouden van informatie)
    a. In geval van veranderende informatie (bouwkundig, installaties, etc), hoe worden deze wijzigingen verwerkt?

Afhankelijk van welke informatie. Als het gaat om keuringen bijvoorbeeld, daarvoor staan prestatie-eisen in het contract. Voor de belangrijkste eisen hierin hebben we een soort dashboard met teller die oranje of rood wordt naar mate de tijdsduur verstrijkt. Op het gebied van tekeningen haalt naar behoefte van de klant de adviseur huisvesting de tekeningen bij Vgib op. Er wordt een ontwerp gemaakt, besproken met de klant en gekozen, en na uitvoering worden de revisietekeningen terug bij Vgib geüpload.

Informatie en BIM

11. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. Zo nee, waarom niet?
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Ja, hier zit zeker toekomst in. De kanttekening hierbij is dat het wel beheersbaar blijft en de informatie moet wel zinvol blijven. Voorbeelden zijn allemaal goed, productspecificaties en acties voor de beheerder zou hierbij kunnen worden geteld. Informatie op websites zie ik overigens alleen maar als bedreiging, omdat websites te onderhevig zijn aan veranderingen.

12. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

In het ontwerpproces absoluut. De 3D-omgeving is als het gebouw eenmaal staat nu minder van belang omdat het van tekeningen net zo goed kan worden afgelezen. Het zou wel handig zijn als je bijvoorbeeld je armatuurtypes wil veranderen, om dan te kunnen kijken of er genoeg ruimte is met je kanalen. Het heeft zeker voordelen, maar je moet het niet overschatten.

13. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Zou kunnen, maar niet bekend mee.

14. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Het automatiseren van koppeling van eigenschappen is sowieso verstandig, omdat anders bij bijvoorbeeld het vervangen van een pomp dit moet worden aangepast in je tekening, in je database, en in je elementenlijst. Dubbel werk is nooit gewenst, en dit garandeert dat dit op een van deze plekken wordt vergeten en gaat het dus fout.

Naam: Joost van de Koppel	**Bedrijf:** Hendriks Bouw & Ontw.	**Datum:** 25-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Ontwikkelen en bouwen.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

BIM-manager. Werkt vaak met alle partijen binnen Hendriks, ontwikkelende tak, E+W, Hendriks onderhoud en beheer, architecten, modellerende partijen, leveranciers, e.d..

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ja, BIM bestaat uit twee dingen, Building Information Model (alleen het model), Modelling (alleen het proces wat leidt tot een model) maar ik beschrijf het vaak als Building

Information Management. Dat integreert deze, en naast het model en het bouwen van het model ook het beheren ervan.

4. Werkt uw organisatie met BIM?

Ja, bouw- en installatietak, maar beheer & onderhoud nog niet (gepland in 2017).

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Kostenreductie. Minder tijd nodig op de bouw, minder doorlooptijd nodig. Werkt daarnaast goed voor je imago als innovatief bedrijf.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Cultuuromslag per definitie. Daarnaast is er een mismatch tussen informatiebehoefte tussen de bouw en het beheer van een gebouw. Deze afstemming ontbreekt op het moment. De bouwer weet niet wat de beheerder nodig heeft, maar de beheerder vraagt het ook niet uit.

Informatie en BIM

7. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    c. Zo nee, waarom niet?
    d. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Ja. Voorbeelden zijn goed. Aanvulling hierop is de meerjarenonderhoudbegroting (MJOB) op zichzelf. Voorbeelden zijn hierop van invloed.

8. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Ja. Een BIM ís in feite al een datastructuur. Het gaat erom dat de data er is, maar ook dat deze gestructureerd is en je snapt wat de samenhang is tussen deze bronnen.

9. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Ja, bij uitstek een goede manier om allerlei andere informatie context te geven. In IFC zit data, dus is al een databron an sich, maar kan vooral gebruikt worden om andere data te koppelen. Ook is het toekomstbestendig.

10. Acht u Semantic Web-technieken geschikt veranderende informatie bij te houden in een BIM-model voor beheer?

Ken ik niet genoeg om een zinnig antwoord te geven.

11. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Het hangt af van de gebruiker. Web-based zou een meerwaarde zijn, om overal op je browser erbij te kunnen. Tot slot lijkt het me goed om zoekopdrachten kunt doen ook in de geschiedenis van panden en/of objecten.

Naam: Ronald van Aggelen	**Bedrijf:** Root b.v.	**Datum:** 26-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Ik ben mede-eigenaar van twee bedrijven: Root bv, en SlimLabs. De ene doet bouwkundige engineering en de andere richt zich puur op BIM-software.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Directeur en mede-eigenaar. We werken vaak met aannemers en (semi-)publieke eigenaren.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ja, en ik denk dat definities zinloos zijn omdat iedereen zo een eigen beeld vormt terwijl we de blik op BIM verliezen.

4. Werkt uw organisatie met BIM?

Ja, we hebben Flatt.io ontwikkeld om informatie te koppelen aan IFC; denk aan huurdersmanagement, ruimtemanagement. Daarnaast zijn we ook bezig met sensoren om real-time gegevens te koppelen aan ruimtes.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Het bespaart geld, werkt efficiënter. Output is veel sneller te leveren.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Niemand weet met BIM waar het precies over gaat. Iedereen denkt dat het een soort vanzelfsprekendheid is dat als je een BIM hebt, dat je het kunt gebruiken voor beheer &

onderhoud. Dit is niet waar, want deze modellen zijn bedoeld om gebouwen te máken en niet om te beheren. De inventarisatie van informatiebehoefte is hierbij cruciaal.

Informatie en BIM

7. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    e. Zo nee, waarom niet?
    f. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Ja, in principe kan alle informatie nuttig zijn. De afweging moet gemaakt worden welke informatie het meest oplevert, en welk proces het dient.

8. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Zie antwoord op de vorige vraag. Heeft wel potentie om bij te dragen aan het interpreteren van informatie. Soms wil je het zien adhv de decompositie van gebouwen, soms wil je een andere structuur kunnen gebruiken.

9. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Het moet zeker een plaats hebben in het koppelen van de relevante informatie. Echter bij het koppelen van informatie is er mijns inziens geen middelpunt. Het is zeker van waarde om data een plek te geven en te kunnen visualiseren, maar het is geen middelpunt.

10. Acht u Semantic Web-technieken geschikt veranderende informatie bij te houden in een BIM-model voor beheer?

Ja, interessant is om na te denken over de houdbaarheid van data, en hoe data muteert. Hierin moet een scheiding gemaakt worden tussen geometrie en de daaraan gerelateerde eigenschappen.

11. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Automatisch koppelen heeft zeker meerwaarde, al zet ik mijn vraagtekens bij het gedeelte "in een IFC-model". Het bevragen van informatie heeft meerwaarde. Kwaliteitscontrole is een aandachtspunt, alsmede informatie ontsluiten over de gebouwen heen. Je kunt hierbij je vraagtekens stellen bij het geometrisch detailniveau (denk aan componenten in een database i.p.v. gemodelleerd in 3D)

Naam: Rob Huijsmans	**Bedrijf:** Zorgbalans	**Datum:** 28-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Het leveren van zorg en ondersteuning aan de kwetsbare mensen in de samenleving, zodat zij zo lang mogelijk de regie over hun eigen leven kunnen behouden. Hierbij is onze strategie m.b.t. vastgoed “huren tenzij…”.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Manager vastgoed. We hebben 3 panden in eigen beheer, daarnaast werken we dus vaak met corporaties waar we panden van huren. Daarnaast bij nieuwbouw bijvoorbeeld aannemers en installateurs en dergelijke.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ja, bekend mee.

4. Werkt uw organisatie met BIM?

Nee. In nieuwbouwprojecten zetten sommige van onze verhuurders wel BIM in, maar bestaande bouw nog niet.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Conflicten worden veel beter opgelost (lees: clashes).

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Medewerkers zijn er nog niet op toegerust, te weinig capaciteit. Tekeningbeheer is nu al slecht op orde, laat staan in 3D.

Informatiebeheer

7. Hoe zijn gebouwdossiers in uw organisatie nu geregeld?
    a. Hoeveel van de documenten/informatie is hardcopy (%)?
    b. Indien digitaal, hoe? (bijv. mappen op een server/in systemen)

Voor tekeningen gebruiken we een webviewer (Vgib), daar zouden we graag meer informatie in laden (certificaat brandmelder, onderhoudsplan, etc.). Dit zit nu in Windows Verkenner, huidige basisstructuur van de informatie. Dat integrale mis ik op dit moment. Voor de panden in eigen beheer hebben we een gebouwbeheersysteem (Priva), maar dit is niet goed digitaal ontsloten. Ik schat dat 25% van de informatie hardcopy is, maar alléén hardcopy nauwelijks meer.

8. Is alle informatie die u nodig hebt in uitvoering van uw taken altijd beschikbaar?

   a. Zo niet, welke informatie niet/waarom niet?
   b. Hoeveel van de actuele informatie is bij leveranciers?

Nee, niet alle informatie is altijd beschikbaar. We beschikken uberhaupt niet altijd over actuele tekeningen, laat staan digitaal. Informatie is daarnaast absoluut niet toegankelijk voor leveranciers.

9. Over welke managementinformatie beschikt u nu en waarover zou u willen beschikken?

Exploitatiekosten en klanttevredenheidsscores zou ik als manager vastgoed heel interessant vinden.

10. Op welke wijze borgt u dat informatie actueel blijft? (onderhouden van informatie)
    a. In geval van veranderende informatie (bouwkundig, installaties, etc), hoe worden deze wijzigingen verwerkt?

Nu nog onvoldoende. Tekeningen worden gereviseerd, maar het proces hoe daarmee wordt omgegaan is nog onvoldoende duidelijk, laat staan of het gevolgd wordt.

Informatie en BIM

11. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. Zo nee, waarom niet?
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
       i. Bijv. gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Absoluut. Productinformatie, certificaten en foto's oplevering zouden hierbij waardevol zijn.

12. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Ja, maar informatiebehoefte van de verschillende partijen is daarvoor wel een aandachtspunt. Wat voeg je toe aan het model, en heeft dat waarde.

13. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Ja, maar onvoldoende verstand van IFC an sich.

14. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Sorteringen in informatie/documenten kunnen maken op basis van de NL-SfB codering. Wat heb ik allemaal, en evt. ook wat heb ik allemaal (bewust) niet.

Naam: Emile Bourquin	**Bedrijf:** Ymere	**Datum:** 28-10-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Optimaal faciliteren van de medewerkers

2. Kunt u aangeven hoe uw functie/afdeling in de organisatie opereert en met welke afdelingen/organisaties deze vaak samenwerkt?

Manager Facilitair Bedrijf. Als Facilitair Bedrijf vallen we onder Directeur Concernzaken (HR, I&A, Continue Verbeteren, Directiestaf, Communicatie, Klantcontact en FB). Onze afdeling bestaat uit 4 aandachtsgebieden: Huisvesting, Services, Veiligheid, Gezondheid en Welzijn en Inkoop. We werken veel samen met de andere afdelingen van Concern, afdeling Vastgoedbeheer (verantwoordelijk voor onderhoud aan de woningen), Afdeling BOG (bedrijfs onroerend goed), Juristen en Secretariaat

Ervaring met BIM

3. Bent u bekend met BIM?
   a. Zo ja, hoe zou u BIM omschrijven?

Niet bekend mee.

4. Werkt uw organisatie met BIM?

Nee.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

-

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

-

Informatiebeheer

7. Hoe zijn gebouwdossiers in uw organisatie nu geregeld?
   a. Hoeveel van de documenten/informatie is hardcopy (%)?
   b. Indien digitaal, hoe? (bijv. mappen op een server/in systemen)

Ik schat dat 80% van de informatie hardcopy is. Wat we digitaal hebben staat verspreid op ofwel een interne server (Verkenner) ofwel een Facility Management Informatie Systeem (FMIS).

8. Is alle informatie die u nodig hebt in uitvoering van uw taken altijd beschikbaar?
   a. Zo niet, welke informatie niet/waarom niet?

b. Hoeveel van de actuele informatie is bij leveranciers?

Nee, dossiers zijn wel eens incompleet of zoek. Ik schat dat 50% van de actuele informatie op een gegeven moment bij leveranciers is.

9. Over welke managementinformatie beschikt u nu, en over welke managementinformatie zou u willen beschikken?

Geen managementinfo. Zou hier graag over beschikken.

10. Op welke wijze borgt u dat informatie actueel blijft? (onderhouden van informatie)
    a. In geval van veranderende informatie (bouwkundig, installaties, etc), hoe worden deze wijzigingen verwerkt?

Wijzigingen komen voornamelijk door huisvestingsprojecten. De leveranciers is dan verantwoordelijk voor de bijstelling van de plattegronden.

Informatie en BIM

11. Heeft het koppelen van documenten of informatie aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. Zo niet, waarom niet?
    b. Zo ja, welke informatie zou <u>zeker</u> aan een BIM gelinkt moeten worden?
        i. Bijv. gebruikershandleidingen, garantietermijnen, productinformatie op de website van de leverancier, productinformatie document, conditierapporten, conditiefoto's, ……

Jazeker. Voorbeelden zijn goed. Aanvulling in de vorm van functies, prestaties, definities van ruimtes, installaties, elementen, veiligheid, kosten en onderhoudsplannen, fysieke tekeningen.

12. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Als iedereen zijn rol en de mogelijkheden goed invult zeker.

13. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Ja

14. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Veranderingen koppelen aan onderhoud in de tijd (MOP/ MIP)

Naam: Stef de Vries	**Bedrijf:** Wardenburg beveiliging & telecom	**Datum:** 03-11-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Installaties op het gebied van beveiliging. Dit verschilt van brandbeveiliging tot inbraak tot cctv en telecom. Op dit gebied doen we ook service en onderhoud.

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Ik ben hoofd tekenaars. Hierbij werken we voornamelijk samen met andere installateurs, alsmede met aannemers en opdrachtgevers.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ja bekend met BIM, we tekenen in Revit i.c.m. Stabicad.

4. Werkt uw organisatie met BIM?

Ja, zie 3. Inmiddels hebben we 7 projecten waarbij we in 3D tekenen met behulp van Revit.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

In 3D is visualisatie een belangrijk voorbeeld. We kunnen bijvoorbeeld direct de hoogte van ruimtes aflezen. Daarnaast is de informatie in je model erg handig; types, nummering, hoe rookmelders geprogrammeerd zijn, etc..

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Ik mis de mogelijkheid nog om extra eigenschappen toe te voegen. Bijv. wanneer in bedrijf gesteld, maar ook storingen. Daarnaast kost het meer werk, je wordt gedwongen eerder na te denken en beslissingen te nemen. Hierop moet je wel bedacht zijn.

Wijzigingen in een gebouw

7. Heeft u ervaring met het onderhoud van gebouwen?
    b. Kunt u een dergelijk werkproces beschrijven?

Niet echt ervaring met onderhoud gezien mijn achtergrond als tekenaar. Kan me wel een proces voorstellen waarbij een melder vervangen moet worden. Dan kijk je in je Revitmodel voor het type etc. en ga je naar locatie en vervang je de melder. Vervolgens ga je terug naar je model en werk je de informatie bij.

8. Bij het vervangen van componenten (of anderszijds aanpassen van installaties), hoe stelt u de opdrachtgever hiervan op de hoogte? (vb. telefoontje/document/uitgebreide log/etc.)
    a. Zo nee, wie doet dit wel?

Dit gaat vaak op basis van een telefoontje of document. Daarnaast wordt er altijd een factuur of werkbon opgesteld. Er is niet echt sprake van een uitgebreide log.

Informatie en BIM

9. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. Zo nee, waarom niet?
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Bijv. gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Hier zit zeker potentie in. Een directe koppeling zou het makkelijker maken om informatie up-to-date te houden. Hierdoor zou je in theorie zelfs direct de koppeling naar de monteur kunnen leggen die on-site het model aanpast.

10. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Ja.

11. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Ja dit is geschikt, maar de gebruiker moet hier wel mee kunnen werken. Ik verwacht dat de gebruiker op dit moment zeker nog wel een soort cursusje nodig heeft om met IFC-bestanden om te kunnen gaan. Aan de andere kant zou je het ook bij lijstjes kunnen houden, waarbij bij het wijzigen van componenten in het lijstje ook direct het model wordt bijgewerkt.

12. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Dit is tweeledig; aan de ene kant moet er een tweezijdige koppeling zijn waarbij wijzigingen in de lijstjes moeten resulteren in wijzigingen in het model en vice versa. Aan de andere kant moet het gebruiksvriendelijk zijn, omdat monteurs en gebruikers het gaan gebruiken. Dit zijn over het algemeen niet de groepen met de meeste kennis van IFC/BIM/tekeningen in het algemeen.

Naam: Niels Vader	**Bedrijf:** Reinier van Arkel	**Datum:** 09-11-16

Algemeen

1. Kunt u aangeven wat de core business van uw organisatie is?

Verlenen van zorg, specifieker geestelijke gezondheidszorg (GGZ).

2. Wat is uw rol in de organisatie en met welke organisaties werkt deze vaak?

Coördinator huisvesting, ik geef leiding aan de afdeling vastgoed. Ik ben binnengekomen als projectleider van nieuwbouw en renovaties. Hierbij hebben we ongeveer 150 gebouwen in gebruik, waarvan 50 à 60 in eigendom. Hierbij ben ik veel in gesprek met zorgdirecteuren, op het gebied van huisvesting veel met aannemers en installateurs.

Ervaring met BIM

3. Bent u bekend met BIM?
    a. Zo ja, hoe zou u BIM omschrijven?

Ja. Terug in 1993 was BIM niet meer dan een 3D-model met een beperkt aantal gegevens erin. Nu zie ik BIM meer voor wat het is, een verandering in de manier waarop partijen samenwerken. De aannemer, ontwerper, installateur, beheerder zitten allemaal vanaf dag 1 om tafel met een model.

4. Werkt uw organisatie met BIM?

Nee.

5. Welke tastbare voordelen door implementatie van BIM heeft u meegemaakt?

Ik heb gemerkt dat visualisatie voor gebruikers (leken) een ontzettend groot voordeel is. Zij hebben geen inzicht op het gebied van plattegronden, dus op deze manier is dit veel beter te visualiseren.

6. Welke knelpunten voorziet u t.a.v. de omschakeling naar BIM in het beheer & onderhoud?

Geld en tijd. Het is nogal een investering, zeker nadat een tijdje geleden onze panden in 2D allemaal al zijn ingemeten. Dit geld is er, zeker in deze tijd in de zorg, niet. Van de hoogte (3D) is niet veel bekend, maar de vraag die ik hierbij heb is wat levert het op. Ik reken namelijk vooral op m^2.

Informatiebeheer

7. Hoe zijn gebouwdossiers in uw organisatie nu geregeld?
    a. Hoeveel van de documenten/informatie is hardcopy (%)?
    b. Indien digitaal, hoe? (bijv. mappen op een server/in systemen)

We hebben heel veel informatie digitaal. Ik durf het niet te schatten in procenten, maar voor mijzelf heb ik 98% van de informatie die ík nodig heb digitaal beschikbaar. Er is ook een archief, daar haal ik de andere 2% vandaan. De digitale informatie is qua tekeningen in

AutoCAD-bestanden waar ook informatie uit te halen is (bijv. polylijnen waar oppervlaktes uit te halen zijn). We werken met een Windows Verkenner-structuur, met een structuur op basis van adres. Daarnaast hebben we informatie in Planon, met een database-achtige structuur en tekeningen daarin. De brontekeningen hierin staan beveiligd voor aanpassingen.

8. Is alle informatie die u nodig hebt in uitvoering van uw taken altijd beschikbaar?
    a. Zo niet, welke informatie niet/waarom niet?
    b. Hoeveel van de actuele informatie is bij leveranciers?

Nee, niet alle informatie is beschikbaar. Gebruikshandleidingen missen bijvoorbeeld. Heeft ook te maken met het feit dat binnen de organisatie van ca. 1300 man er 2 kennis hebben van Planon waarvan ik er niet één ben. Kengetallen heb ik vaak ook niet m.b.t. gas-water-licht of beheerskosten. Van *actuele* informatie is maar heel weinig bij leveranciers. Het zou ontzettend interessant zijn als je vaste leveranciers mee mogen kijken met je in jouw BIM als ze werkzaamheden uitvoeren.

9. Over welke managementinformatie beschikt u nu en waarover zou u willen beschikken?

Verbruik van gas-water-licht als nulsituatie zijn handig als je ingrepen wil doorrekenen. Terugverdientijd, maar ook kengetallen incl. redenatie daarachter. De managementinformatie die ik nu gebruik zijn vierkante meters. Dat is echt de basis; VVO en BVO. Verkeerde vierkante meters kan al resulteren in 10% extra kosten.

10. Op welke wijze borgt u dat informatie actueel blijft? (onderhouden van informatie)
    a. In geval van veranderende informatie (bouwkundig, installaties, etc), hoe worden deze wijzigingen verwerkt?

Dit wordt op het moment op geen enkele manier geborgd.

Informatie en BIM

11. Heeft het koppelen van informatie of documenten aan een BIM-model potentie m.b.t. de beheer- & onderhoudsfase?
    a. ~~Zo nee, waarom niet?~~
    b. Zo ja, welke informatie zou zeker aan een BIM gelinkt moeten worden?
        i. Bijv. gebruikershandleidingen, garantietermijnen, productinformatie op website van de leverancier, productinformatie document, conditierapporten, foto's…

Productspecificaties, zoals materiaal, maar ook onderhoudsinterval. Eigenlijk zou je zoveel mogelijk informatie digitaal al willen voor je het gebouw hebt gezien. Omgevingsinfo, denk aan Google Maps. Datum invoer gegevens, zo heb je bijvoorbeeld voor WOZ-waardes wat context. Financiële gegevens van gebouwen.

12. Denkt u dat het toevoegen van nieuwe eigenschappen/koppelen van documenten in een 3D-omgeving nuttig zal zijn bij het structureren van informatie?

Ja. Een 3D-omgeving draagt sowieso al bij in het visuele voordeel wat ik al beschreef. Een structuur op basis van het gebouw zie ik ook als zeer nuttig.

13. Acht u IFC-modellen geschikt als middelpunt waaraan relevante informatie gekoppeld wordt?

Ja. Idee achter IFC is de basis van BIM.

14. Welke functies zou een applicatie, bedoeld om gebouwinformatie in beheer te ontsluiten en veranderingen in modellen/informatie te verwerken in een IFC-model, zeker ook nog moeten hebben (bijv. automatisch documenten/eigenschappen koppelen, zoekopdrachten in alle gekoppelde informatie, enz.)? Kunt u een bijbehorend voorbeeld beschrijven?

Het moet gebruiksvriendelijk zijn, zodat wanneer Jantje weggaat, Pietje er direct mee kan werken. Zoekopdrachten zijn ook erg handig.

Appendix II: Python Script

Appendix II.I: Main tool

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

from collections import defaultdict
from operator import itemgetter

from PyQt4 import QtCore, QtGui
from PyQt4.QtCore import QObject, pyqtSignal
from OCC.Display.backend import get_backend
get_backend("qt-pyqt4")
import OCC.Display.qtDisplay
from OCC.Display.qtDisplay import qtViewer3d

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

import rdflib
from rdflib import Graph, Literal, URIRef, Namespace, XSD, RDF, RDFS, BNode
from rdflib.namespace import NamespaceManager
import csv
import urllib
import re

import CreateIFCElement

busy = Namespace ("http://example.org/ont/buildingsystems#")
busyI = Namespace ("http://example.org/id/buildingsystems#")
log = Namespace ("http://example.org/ont/log#")
logI = Namespace ("http://example.org/id/log#")

guid_selection = None

class ProductViewer(qtViewer3d):
    def __init__(self, *args):
        qtViewer3d.__init__(self)
        self.objects = {}

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

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

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

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

    def mouseReleaseEvent(self, *args):
        qtViewer3d.mouseReleaseEvent(self, *args)
        if self._display.selected_shape:
            global guid_selection
            global selected_shape
            selected_shape = self._display.selected_shape
```

```
            guid_selection = (self.objects[ProductViewer.Hash(self._display.selected_shape)])

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

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

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

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

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

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

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

        self.display.FitAll()

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

        #Instantiating the tabs
        global filename
        self.filename = None

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

        self.rdf_tab = QtGui.QWidget()
        self.tabs.addTab(self.rdf_tab, "Export O&M data to RDF")

        self.viewer_tab = QtGui.QWidget()
        self.tabs.addTab(self.viewer_tab, "3D O&M Viewer")

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

        self.setGeometry(100, 100, 850, 550)
        self.setWindowTitle("Application IFC & RDF | Log")

        #Calling both the tabs
        self.tab_rdf()
        self.tab_3dview()

#----------tab 1----------
    def tab_rdf(self):

        vbox = QtGui.QVBoxLayout()
        hbox = QtGui.QHBoxLayout()
        hbox2 = QtGui.QHBoxLayout()
        hbox3 = QtGui.QHBoxLayout()
        vbox2 = QtGui.QVBoxLayout()
        vbox3 = QtGui.QVBoxLayout()

        vbox.addLayout(hbox)
```

```
        vbox.addLayout(hbox2)
        hbox2.addLayout(vbox2)
        hbox2.addLayout(vbox3)
        vbox3.addLayout(hbox3)
        self.rdf_tab.setLayout(vbox)

        vbox2.setSpacing(50)
        vbox3.setSpacing(50)
        hbox3.setSpacing(10)

        csv_to_rdf_introduction = QtGui.QLabel(
"""
This tab provides operations to open CSV data and transform the data into triples in an RDF
file. This operation enables the data to be linked to the rooms in the IFC file of the
building.

This operation is tailored to the system information provided by Zayaz.

The next tab enables the user to visualise IFC files in a 3D environment and connect the RDF
data to the elements in the IFC model.
"""
              )
        csv_to_rdf_introduction.setMaximumHeight(150)
        csv_to_rdf_introduction.setWordWrap(True)

        openCSV_btn_title = QtGui.QLabel("1. Open a CSV file that contains O&M data:")
        openCSV_btn = QtGui.QPushButton("Open CSV file", self)
        self.openCSV_btn_path = QtGui.QLineEdit(self)
        self.openCSV_btn_path.setPlaceholderText("Open the required O&M data here")
        openCSV_btn.clicked.connect(self.open_csv)

        writetoRDF_btn_title = QtGui.QLabel("2. Write the O&M data to an RDF file:")
        writetoRDF_btn = QtGui.QPushButton("Write to RDF", self)
        writetoRDF_btn.clicked.connect(self.writecsv_to_rdf)

        hbox.addWidget(csv_to_rdf_introduction)
        vbox2.addWidget(openCSV_btn_title)
        vbox2.addWidget(writetoRDF_btn_title)
        hbox3.addWidget(self.openCSV_btn_path)
        hbox3.addWidget(openCSV_btn)
        vbox3.addWidget(writetoRDF_btn)

    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))
        return self.csvFilePath
        self.openCSV_btn_path.setText(self.csvFilePath)

    def writecsv_to_rdf(self):
        g = Graph()
        namespace_manager = NamespaceManager(g)
        namespace_manager.bind('busy', busy, override=False)

        with open(self.csvFileName, "r") as csvfile:
            reader = csv.DictReader(csvfile, delimiter=';')
            for row in reader:
                comp = row['Component']+"/"+str(uuid.uuid4())

                #Kolommen in csv toewijzen aan type volgens ontologie
                g.add( (URIRef(busyI+row['Complexnummer']), RDF.type,
URIRef(busy.BuildingComplex)) )
                g.add( (URIRef(busyI+row['Gebouwen behorende tot complexnummer']), RDF.type,
URIRef(busy.BuildingFromComplex)) )
                g.add( (URIRef(busyI+row['Gebouwen behorende tot complexnummer']),
busy.partOfComplex, URIRef(busyI+row['Complexnummer'])) )

                g.add( (URIRef(busyI+row['Corporatie']), RDF.type,
URIRef(URIRef(busy.Organization))) )
                g.add( (URIRef(busyI+row['Complexnummer']), busy.ownedBy,
URIRef(busyI+row['Corporatie'])) )

                g.add( (URIRef(busyI+row['Huurder pand']), RDF.type,
URIRef(busy.Organization)) )
                g.add( (URIRef(busyI+row['Gebouwen behorende tot complexnummer']),
busy.rentedBy, URIRef(busyI+row['Huurder pand'])) )
```

```
                g.add( (URIRef(busyI+(row['Ruimte']+'_'+row['Ruimte_id'])), RDF.type,
URIRef(busy.Room)) )
                g.add( (URIRef(busyI+row['Gebouwen behorende tot complexnummer']),
busy.consistsOf, URIRef(busyI+(row['Ruimte']+'_'+row['Ruimte_id']))) )
                g.add( (URIRef(busyI+row['Installatie']), RDF.type, URIRef(busy.System)) )
                g.add( (URIRef(busyI+(row['Ruimte']+'_'+row['Ruimte_id'])), busy.contains,
URIRef(busyI+comp)) )

                if row['Component'] == "":continue
                g.add( (URIRef(busyI+row['Leverancier onderhoud']), RDF.type,
URIRef(busy.Organization)) )
                g.add( (URIRef(busyI+comp), RDF.type, URIRef(busy.Component)) )
                g.add( (URIRef(busyI+comp), busy.maintenanceBy, URIRef(busyI+row['Leverancier
onderhoud'])) )
                g.add( (URIRef(busyI+comp), busy.subpartOf, URIRef(busyI+row['Installatie']))
)
                g.add( (URIRef(busyI+comp), busy.operationBy, URIRef(busyI+row['Beheerder
installatie'])) )

                g.add( (URIRef(busyI+row['Beheerder installatie']), RDF.type,
URIRef(busy.Organization)) )

                #Toewijzen properties aan complex en component
                g.add( (URIRef(busyI+row['Complexnummer']), busy.name,
Literal(row['Complexnaam'], datatype=XSD.string)) )
                g.add( (URIRef(busyI+row['Complexnummer']), busy.address,
Literal(row['Adresgegevens'], datatype=XSD.string)) )
                g.add( (URIRef(busyI+row['Complexnummer']), busy.hasFunction,
Literal(row['Functie gebouw'], datatype=XSD.string)) )
                g.add( (URIRef(busyI+row['Complexnummer']), busy.description,
Literal(row['Toelichting'], datatype=XSD.string)) )

                g.add( (URIRef(busyI+comp), busy.manufacturer, URIRef(busyI+row['Fabrikant']))
)
                g.add( (URIRef(busyI+comp), busy.brand, URIRef(busyI+row['Merk'])) )
                g.add( (URIRef(busyI+comp), busy.type,
Literal(row['Type'],datatype=XSD.string)) )
                g.add( (URIRef(busyI+comp), busy.lastMaintained, Literal(row['Datum laatst
onderhouden'],datatype=XSD.date)) )
                g.add( (URIRef(busyI+comp), busy.yearOfProduction, Literal(row['Bouwjaar
installatie'],datatype=XSD.gYear)) )

##              g.add( (busyI+row['Installatie'], busy.permitpresent,
Literal(row['Omgevingsvergunning aanwezig?'],datatype=XSD.boolean)) )
##              g.add( (busyI+row['Installatie'], busy.certificatepresent, Literal(row['Geldig
inspectiecertificaat aanwezig?'],datatype=XSD.boolean)) )

        g.serialize('%s.rdf'%self.csvFileName, format='xml')
        print "RDF file successfully generated! Data written to %s.rdf."%self.csvFileName

#--------tab 2---------
    def tab_3dview(self):
        #Initializing a split-view layout
        self.propertybox = QtGui.QTextBrowser()
        font = QtGui.QFont("Arial",10,QtGui.QFont.Bold,True)
        sizePolicy = QtGui.QSizePolicy(QtGui.QSizePolicy.Fixed,
QtGui.QSizePolicy.MinimumExpanding)
        self.propertybox.setFont(font)
        self.propertybox.setSizePolicy(sizePolicy)
        self.componentbox = QtGui.QListWidget()
        self.componentbox.setFont(font)
        self.componentbox.setSizePolicy(sizePolicy)
        self.propertybox2 = QtGui.QTextBrowser()
        self.propertybox2.setFont(font)
        self.propertybox2.setSizePolicy(sizePolicy)

        item = QtGui.QListWidgetItem()
        self.componentbox.addItem(item)
        self.componentbox.itemClicked.connect(self.comp_clicked)

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

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

```
        viewer_open_ifc_btn = QtGui.QPushButton("Open IFC", self)
        viewer_open_ifc_btn.clicked.connect(self.open_ifc_file)
        viewer_open_rdf_btn = QtGui.QPushButton("Load RDF data", self)
        viewer_open_rdf_btn.clicked.connect(self.open_rdf_data)
        viewer_show_prop_btn = QtGui.QPushButton("Show properties", self)
        viewer_show_prop_btn.clicked.connect(self.viewer_getGUID)
        viewer_color_btn = QtGui.QPushButton("Color rooms", self)
        viewer_color_btn.clicked.connect(self.color_ifc_with_rdf_obj)
        viewer_add_report_btn = QtGui.QPushButton("Add to log", self)
        viewer_add_report_btn.clicked.connect(self.window_add_to_log)
        viewer_change_comp_btn = QtGui.QPushButton("Change component", self)
        viewer_change_comp_btn.clicked.connect(self.window_change_comp)
        viewer_create_log_btn = QtGui.QPushButton("Open log", self)
        viewer_create_log_btn.clicked.connect(self.create_log_file)
        viewer_rooms_objects_btn = QtGui.QPushButton("Switch rooms/elements", self)
        viewer_rooms_objects_btn.clicked.connect(self.switch_rooms_elements)

        splitter = QtGui.QSplitter(QtCore.Qt.Horizontal)
        splitterH = QtGui.QSplitter(QtCore.Qt.Vertical)
        splitter.addWidget(self.canvas)
        splitter.addWidget(splitterH)
        splitterH.addWidget(viewer_rooms_objects_btn)
        splitterH.addWidget(self.propertybox)
        splitterH.addWidget(self.componentbox)
        splitterH.addWidget(self.propertybox2)

        viewer_vbox.addWidget(splitter)
        viewer_vbox.addLayout(viewer_hbox)
        self.viewer_tab.setLayout(viewer_vbox)
        viewer_hbox.addWidget(viewer_open_ifc_btn)
        viewer_hbox.addWidget(viewer_open_rdf_btn)
        viewer_hbox.addWidget(viewer_show_prop_btn)
        viewer_hbox.addWidget(viewer_color_btn)
        viewer_hbox.addWidget(viewer_add_report_btn)
        viewer_hbox.addWidget(viewer_change_comp_btn)
        viewer_hbox.addWidget(viewer_create_log_btn)

        self.count = 0
        self.rdf_graph = None

        self.g1 = Graph()
        namespace_manager = NamespaceManager(self.g1)
        namespace_manager.bind('log', log, override=False)

    def viewer_getGUID(self):
        if not self.filename:
            QtGui.QMessageBox.warning(self,
                                      "No IFC loaded!",
                                      "Please load a model first!")
            return
        self.propertybox.clear()
        self.guid_to_prop_dict()
        #Append IFC properties to textbox for GUID input
        for guid in self.guid_to_prop[guid_selection]:
            if guid.is_a('IfcQuantityArea'):
                self.propertybox.append("%s: %s" %(guid.Name, int(guid.AreaValue)))
            elif not guid.is_a('IfcPhysicalSimpleQuantity'):
                self.propertybox.append("%s: %s" %(guid.Name, guid.NominalValue.wrappedValue))

        if self.rdf_graph == None:
            QtGui.QMessageBox.warning(self,
                                  "No RDF data loaded!",
                                  "Please load RDF data first.")
            return
        self.componentbox.clear()
        self.dict_guid_name()
        #Append component names to list widget
        self.room = busyI + "".join(self.name_to_guid[[guid_selection][0]])
        self.qres = """PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?c WHERE { ?c rdf:type busy:Component . <%s> busy:contains ?c .
FILTER NOT EXISTS { ?c busy:replacedBy ?cnew } }"""%self.room
        if len(self.rdf_graph.query(self.qres)) == 0:
            item = QtGui.QListWidgetItem("No components present in this room!")
            self.componentbox.addItem(item)
        else:
```

```
            for result in self.rdf_graph.query(self.qres):
                comp = "".join(result).split('#')[-1]
                item = QtGui.QListWidgetItem("".join(comp).split('/',1)[0])
                self.componentbox.addItem(item)
        self.propertybox2.clear()
        self.canvas._display.Repaint()

    def guid_to_prop_dict(self):
        #Append IFC properties of rooms to dictionary
        self.guid_to_prop = defaultdict(list)
        self._storey = defaultdict(list)
        for elem in self.ifc_file.by_type("IfcProduct"):
            if elem.is_a('IfcSpace'):
                shapes = self.canvas.ColorBack(elem.GlobalId)
                for st in elem.Decomposes:
                    self._storey[elem.GlobalId].append(st.RelatingObject.LongName)
            for rel in elem.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    if rel.RelatingPropertyDefinition.is_a("IfcElementQuantity"):
                        for q in rel.RelatingPropertyDefinition.Quantities:
                            if q.is_a("IfcPhysicalSimpleQuantity"):
                                self.guid_to_prop[elem.GlobalId].append(q)
                    else:
                        for prop in rel.RelatingPropertyDefinition.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.propertybox.clear()
        self.parse_ifc(self.filename)

    def open_rdf_data(self, filename=None):
        self.RDFfilename = QtGui.QFileDialog.getOpenFileName(self, 'Open file',".","Resource
Description Framework (*.rdf)")
        self.rdf_graph = Graph()
        self.rdf_graph.parse(str(self.RDFfilename))
        print "RDF data successfully loaded!"

    def parse_ifc(self,filename):
        self.created_shapes = {}
        self.ifc_file = ifcopenshell.open(filename)
        rooms = self.ifc_file.by_type("IfcSpace")
        for room in rooms:
            if room.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings, room).geometry
                shp = self.canvas.Show(room.GlobalId, ifcgeom, None)
        print "IFC file successfully loaded!"

    def color_ifc_with_rdf_obj(self,item):
        guidOfRooms = []
        current = self.componentbox.currentItem().text()
        if "No components present" in current:
            return
        for comp in self.rdf_graph.query(self.qres):
            if current in "".join(comp):
                qdate = """PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?r WHERE { ?r busy:contains ?c . FILTER (regex(str(?c), "%s") )
                FILTER NOT EXISTS { ?c busy:replacedBy ?cnew } }"""%current
                print "Selected component also in %s other
rooms."%len(self.rdf_graph.query(qdate))
                for row in self.rdf_graph.query(qdate):
                    result = "".join(row).split('#')[-1].replace('_',' ')
                    guidOfRooms.append("".join(self.guid_to_name[result]))
        for room in self.ifc_file.by_type('IfcSpace'):
            if room.Representation:
                if room.GlobalId in guidOfRooms:
                    shapes = self.canvas.Color(room.GlobalId)
                else:
                    shapes = self.canvas.ColorBack(room.GlobalId)
        self.canvas._display.Repaint()

    def dict_guid_name(self):
        #Create dictionary from IFC with GUID and associated room name
```

```
        self.name_to_guid = defaultdict(list)
        self.guid_to_name = defaultdict(list)
        for room in self.ifc_file.by_type("IfcSpace"):
            for rel in room.IsDefinedBy:
                if rel.is_a("IfcRelDefinesByProperties"):
                    if rel.RelatingPropertyDefinition.is_a("IfcElementQuantity"):continue
                    for prop in rel.RelatingPropertyDefinition.HasProperties:
                        if prop.is_a("IfcPropertySingleValue"):
                            if prop.NominalValue.is_a("IfcIdentifier"):
                                name = prop.NominalValue.wrappedValue.replace(' ', '_')
                                self.name_to_guid[room.GlobalId].append(str(name))
                                self.guid_to_name[prop.NominalValue.wrappedValue].
append(room.GlobalId)

    def comp_clicked(self, item):
        self.propertybox2.clear()
        if "No components present" in item.text():
            return
        #Fill bottom textbox with latest inspection date
        for comp in self.rdf_graph.query(self.qres):
            comp_ = "".join(comp)
            if item.text() in comp_:
                qdate = """PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX log: <http://example.org/ont/log#>
SELECT ?d WHERE { <%s> busy:hasLog ?l . { SELECT * WHERE {?l log:dateTime ?d .}
 } } ORDER BY DESC(?d) LIMIT 1"""%comp_
                for row in self.rdf_graph.query(qdate):
                    self.propertybox2.append("Last maintained at: %s"%row)
        #Fill bottom textbox with additional O&M data
                qprop = """PREFIX busy: <http://example.org/ont/buildingsystems#>
SELECT ?i ?m ?b ?t ?y ?mp ?o WHERE { <%s> busy:subpartOf ?i ;
                                    busy:manufacturer ?m ;
                                    busy:brand ?b ;
                                    busy:type ?t ;
                                    busy:yearOfProduction ?y ;
                                    busy:maintenanceBy ?mp ;
                                    busy:operationBy ?o}"""%comp_
                for row in self.rdf_graph.query(qprop):
                    self.propertybox2.append("Installation: %s\nManufacturer: %s\nBrand:
%s\nType: %s\nYear of production: %s\nMaintenance done by: %s\nOperating party:
%s"%(row[0].split('#')[-1],row[1].split('#')[-1],row[2].split('#')[-
1],row[3],row[4],row[5].split('#')[-1],row[6].split('#')[-1]))

    def window_add_to_log(self):
        if guid_selection == None:
            QtGui.QMessageBox.warning(self,
                                      "Select room first!",
                                      "Please select room before adding log entry.")
            return
        if len(self.rdf_graph.query(self.qres)) == 0:
            QtGui.QMessageBox.information(self,
                                          "No components present!",
                                          "This room contains no components to inspect!")
            return

        #Initialize dialog to add entries to log
        self.add_to_log = QtGui.QDialog()
        self.add_to_log.__init__(self)

        self.add_to_log.setWindowTitle("Dialog")
        self.add_to_log.setGeometry(500, 500, 300, 100)
        log_vbox_main = QtGui.QVBoxLayout()
        log_hbox = QtGui.QHBoxLayout()
        log_vbox = QtGui.QVBoxLayout()
        log_vbox2 = QtGui.QVBoxLayout()

        log_title = QtGui.QLabel("Add outage/maintenance for log:")
        buttonBox = QtGui.QDialogButtonBox()
        buttonBox.setOrientation(QtCore.Qt.Horizontal)
        buttonBox.setStandardButtons(QtGui.QDialogButtonBox.Cancel|QtGui.QDialogButtonBox.Ok)
        buttonBox.accepted.connect(self.add_log_entry)
        buttonBox.rejected.connect(self.add_to_log.reject)

        self.add_to_log.setLayout(log_vbox_main)
        log_vbox_main.addWidget(log_title)
        log_vbox_main.addLayout(log_hbox)
        log_vbox_main.addWidget(buttonBox)
```

```
        log_hbox.addLayout(log_vbox)
        log_hbox.addLayout(log_vbox2)

        insert_name_title = QtGui.QLabel("Name:")
        self.insert_name = QtGui.QLineEdit(self)
        insert_room_title = QtGui.QLabel("Room:")
        self.insert_room = QtGui.QLineEdit(self)
        self.insert_room.setReadOnly(True)
        self.insert_room.setText("".join(self.name_to_guid[[guid_selection][0]]))
        insert_comp_title = QtGui.QLabel("Component(s):")
        self.insert_comp = QtGui.QComboBox(self)
        for comp in self.rdf_graph.query(self.qres):
            self.insert_comp.addItem(("".join(comp).split('#')[-1]).split('/',1)[0])
        insert_reason_title = QtGui.QLabel("Reason:")
        self.insert_reason = QtGui.QLineEdit(self)

        log_vbox.addWidget(insert_name_title)
        log_vbox.addWidget(insert_room_title)
        log_vbox.addWidget(insert_comp_title)
        log_vbox.addWidget(insert_reason_title)
        log_vbox2.addWidget(self.insert_name)
        log_vbox2.addWidget(self.insert_room)
        log_vbox2.addWidget(self.insert_comp)
        log_vbox2.addWidget(self.insert_reason)

        self.add_to_log.exec_()

    def add_log_entry(self):
        if (self.insert_name.text() or self.insert_reason.text()) == "":
            QtGui.QMessageBox.warning(self,
                                      "Fields still empty!",
                                      "Please fill in all the fields!")
            return
        #When fields are filled, entry is added to dataset
        for comp in self.rdf_graph.query(self.qres):
            comp_ = "".join(comp)
            if self.insert_comp.currentText() in comp_:
                name = logI + self.insert_name.text()
                reason = self.insert_reason.text()
                current_date = time.strftime("%Y-%m-%d")
                log_id = "Report_%s(%s)" %(current_date, self.count)
                log_url = logI + log_id
                log_filename = "%s_test.rdf" %(log_id)

                self.g1.add( (URIRef(log_url), RDF.type, URIRef(log.Report)) )
                self.g1.add( (URIRef(log_url), log.reportedBy, URIRef(name)) )
                self.g1.add( (URIRef(log_url), log.dateTime, Literal(time.strftime("%c"),
datatype=XSD.dateTime)) )
                self.g1.add( (URIRef(log_url), log.where, URIRef(self.room)) )
                self.g1.add( (URIRef(log_url), log.component, URIRef(comp_)) )
                self.g1.add( (URIRef(log_url), log.reason, Literal(reason,
datatype=XSD.string)) )
                self.rdf_graph.update("""
PREFIX busy: <http://example.org/ont/buildingsystems#>
PREFIX log: <http://example.org/ont/log#>
INSERT DATA { <%s> busy:hasLog <%s> ;
                   busy:lastMaintained "%s" .
              <%s> log:dateTime "%s" .
}"""%(comp_,log_url,current_date,log_url,current_date))
                print "Log entry saved successfully!"
            self.add_to_log.close()
            self.count += 1

    def window_change_comp(self):
        if guid_selection == None:
            QtGui.QMessageBox.warning(self,
                                      "Select room first!",
                                      "Please select room before adding log entry.")
            return
        if len(self.rdf_graph.query(self.qres)) == 0:
            QtGui.QMessageBox.information(self,
                                          "No components present!",
                                          "This room contains no components to replace or
maintain!")
            return

        #Initialize dialog to add entries to log
```

```
        self.comp_change = QtGui.QDialog()
        self.comp_change.__init__(self)

        self.comp_change.setWindowTitle("Dialog")
        self.comp_change.setGeometry(500, 500, 300, 100)
        log_vbox_main = QtGui.QVBoxLayout()
        log_hbox = QtGui.QHBoxLayout()
        log_vbox = QtGui.QVBoxLayout()
        log_vbox2 = QtGui.QVBoxLayout()

        log_title = QtGui.QLabel("Replace components:")
        buttonBox = QtGui.QDialogButtonBox()
        buttonBox.setOrientation(QtCore.Qt.Horizontal)
        buttonBox.setStandardButtons(QtGui.QDialogButtonBox.Cancel|QtGui.QDialogButtonBox.Ok)
        buttonBox.accepted.connect(self.change_comp)
        buttonBox.rejected.connect(self.comp_change.reject)

        self.comp_change.setLayout(log_vbox_main)
        log_vbox_main.addWidget(log_title)
        log_vbox_main.addLayout(log_hbox)
        log_vbox_main.addWidget(buttonBox)
        log_hbox.addLayout(log_vbox)
        log_hbox.addLayout(log_vbox2)

        insert_room_title = QtGui.QLabel("Room:")
        self.insert_room = QtGui.QLineEdit(self)
        self.insert_room.setReadOnly(True)
        self.insert_room.setText("".join(self.name_to_guid[[guid_selection][0]]))
        insert_comp_title = QtGui.QLabel("Component:")
        self.insert_comp = QtGui.QComboBox(self)
        for comp in self.rdf_graph.query(self.qres):
            self.insert_comp.addItem(("".join(comp).split('#')[-1]).split('/',1)[0])
        insert_mnfctr_title = QtGui.QLabel("Manufacturer:")
        self.insert_mnfctr = QtGui.QLineEdit(self)
        insert_brand_title = QtGui.QLabel("Brand:")
        self.insert_brand = QtGui.QLineEdit(self)
        insert_type_title = QtGui.QLabel("Type:")
        self.insert_type = QtGui.QLineEdit(self)

        log_vbox.addWidget(insert_room_title)
        log_vbox.addWidget(insert_comp_title)
        log_vbox.addWidget(insert_mnfctr_title)
        log_vbox.addWidget(insert_brand_title)
        log_vbox.addWidget(insert_type_title)
        log_vbox2.addWidget(self.insert_room)
        log_vbox2.addWidget(self.insert_comp)
        log_vbox2.addWidget(self.insert_mnfctr)
        log_vbox2.addWidget(self.insert_brand)
        log_vbox2.addWidget(self.insert_type)

        self.comp_change.exec_()

    def change_comp(self):
        for comp in self.rdf_graph.query(self.qres):
            if self.insert_comp.currentText() in "".join(comp):
                for subj,pred,obj in self.rdf_graph:
                    if "".join(comp) in subj:
                        oldcomp = subj
                        if 'subpartOf' in pred:
                            self.inst = obj
                        if 'maintenanceBy' in pred:
                            self.MP = obj
                        if 'operationBy' in pred:
                            self.OP = obj

        self.mnfctr = busyI + self.insert_mnfctr.text()
        self.brand = busyI + self.insert_brand.text()
        self.ctype = Literal("%s"%(self.insert_type.text()), datatype=XSD.string)
        self.YOP = Literal(time.strftime("%Y"), datatype=XSD.gYear)
        self.current_date = Literal(time.strftime("%Y-%m-%d"), datatype=XSD.date)
        newcomp = busyI + self.insert_comp.currentText() + "/" + str(uuid.uuid4())
        if (self.insert_mnfctr.text() and self.insert_brand.text()) == "":
            QtGui.QMessageBox.warning(self,
                                      "Fields still empty!",
                                      "Please fill in all the fields!")
            return
        qres = """PREFIX busy: <http://example.org/ont/buildingsystems#>
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT { <%s> rdf:type busy:Component ;
              busy:lastMaintained "%s" ;
              busy:yearOfProduction "%s" ;
              busy:manufacturer <%s> ;
              busy:brand <%s> ;
              busy:type "%s" ;
              busy:subpartOf <%s> ;
              busy:maintenanceBy <%s> ;
              busy:operationBy <%s> .
        <%s> busy:replacedBy <%s> .
        <%s> busy:contains <%s> . }
WHERE { <%s> rdf:type busy:Component .
        <%s> busy:maintenanceBy <%s> ;
             busy:operationBy <%s> .
        <%s> busy:contains <%s> . }
"""%(newcomp,self.current_date,self.YOP,self.mnfctr,self.brand,self.ctype,self.inst,self.MP,se
lf.OP,oldcomp,newcomp,self.room,newcomp,oldcomp,oldcomp,self.MP,self.OP,self.room,oldcomp)
        print "Old component: %s\nNew component: %s"%(oldcomp,newcomp)
        self.rdf_graph.update(qres)

        try:
            self.ifcfile
        except AttributeError:
            self.ifcfile = None
        if self.ifcfile is None:
            self.create_ifc_file()
            self.write_comp_to_ifc()
        else:
            self.write_comp_to_ifc()
        print "Maintenance processed successfully!"
        self.comp_change.close()

    def create_ifc_file(self):
        self.create_guid = lambda: ifcopenshell.guid.compress(uuid.uuid1().hex)
        self.ifcfile = ifcopenshell.open(CreateIFCElement.temp_filename)
        self.owner_history = self.ifcfile.by_type("IfcOwnerHistory")[0]
        project = self.ifcfile.by_type("IfcProject")[0]
        self.context = self.ifcfile.by_type("IfcGeometricRepresentationContext")[0]

        site_placement = CreateIFCElement.create_ifclocalplacement(self.ifcfile)
        site = self.ifcfile.createIfcSite(self.create_guid(), self.owner_history, "Site",
None, None, site_placement, None, None, "ELEMENT", None, None, None, None, None)
        building_placement = CreateIFCElement.create_ifclocalplacement(self.ifcfile,
relative_to=site_placement)
        building = self.ifcfile.createIfcBuilding(self.create_guid(), self.owner_history,
"Building", None, None, building_placement, None, None, "ELEMENT", None, None, None)
        for storey in self.ifc_file.by_type("IfcBuildingStorey"):
            storey_placement = CreateIFCElement.create_ifclocalplacement(self.ifcfile,
relative_to=building_placement)
            name = storey.LongName
            elevation = storey.Elevation
            building_storey = self.ifcfile.createIfcBuildingStorey(self.create_guid(),
self.owner_history, name, None, None, storey_placement, None, name, "ELEMENT", elevation)
            container_storey = self.ifcfile.createIfcRelAggregates(self.create_guid(),
self.owner_history, "Building Container", None, building, [building_storey])
        container_site = self.ifcfile.createIfcRelAggregates(self.create_guid(),
self.owner_history, "Site Container", None, site, [building])
        container_project = self.ifcfile.createIfcRelAggregates(self.create_guid(),
self.owner_history, "Project Container", None, project, [site])

    def write_comp_to_ifc(self):
        for storey in self.ifcfile.by_type("IfcBuildingStorey"):
            if storey.LongName == self._storey[guid_selection][0]:
                plcmnt = storey.ObjectPlacement
        comp_placement = CreateIFCElement.create_ifclocalplacement(self.ifcfile,
relative_to=plcmnt)
        polyline = CreateIFCElement.create_ifcpolyline(self.ifcfile, [(0.0, 0.0, 0.0), (0.2,
0.0, 0.0)])
        axis_representation = self.ifcfile.createIfcShapeRepresentation(self.context, "Axis",
"Curve2D", [polyline])

        space_shape = []
        bbox = OCC.Bnd.Bnd_Box()
        for space in self.ifc_file.by_type("IfcSpace"):
            shape = selected_shape
            space_shape.append((space, shape))
```

```
            OCC.BRepBndLib.brepbndlib_Add(shape, bbox)
        bounding_box_center = ifcopenshell.geom.utils.get_bounding_box_center(bbox)

        extrusion_placement = CreateIFCElement.create_ifcaxis2placement(self.ifcfile,
(bounding_box_center.X(),bounding_box_center.Y(),bounding_box_center.Z()), (0.0, 0.0, 1.0),
(1.0, 0.0, 0.0))
        point_list_extrusion_area = [(0.0, -0.1, 0.0), (0.2, -0.1, 0.0), (0.2, 0.1, 0.0),
(0.0, 0.1, 0.0), (0.0, -0.1, 0.0)]
        solid = CreateIFCElement.create_ifcextrudedareasolid(self.ifcfile,
point_list_extrusion_area, extrusion_placement, (0.0, 0.0, 1.0), 0.2)
        body_representation = self.ifcfile.createIfcShapeRepresentation(self.context, "Body",
"SweptSolid", [solid])
        product_shape = self.ifcfile.createIfcProductDefinitionShape(None, None,
[axis_representation, body_representation])
        comp = self.ifcfile.createIfcBuildingElementProxy(self.create_guid(),
self.owner_history, str(self.insert_comp.currentText()),None, None, comp_placement,
product_shape, None)

#Create and assign updated propertyset
        property_values = [
            self.ifcfile.createIfcPropertySingleValue("Last maintained", "Last maintained",
            self.ifcfile.create_entity("IfcText", str(self.current_date)), None),
            self.ifcfile.createIfcPropertySingleValue("Year of production", "Year of
production",
            self.ifcfile.create_entity("IfcInteger", int(self.YOP)), None),
            self.ifcfile.createIfcPropertySingleValue("Manufacturer", "Manufacturer",
            self.ifcfile.create_entity("IfcText", str(self.mnfctr.split('#')[-1])), None),
            self.ifcfile.createIfcPropertySingleValue("Brand", "Brand",
            self.ifcfile.create_entity("IfcText", str(self.brand.split('#')[-1])), None),
            self.ifcfile.createIfcPropertySingleValue("Type", "Type",
            self.ifcfile.create_entity("IfcText", str(self.ctype)), None),
            self.ifcfile.createIfcPropertySingleValue("Part of installation", "Part of
installation",
            self.ifcfile.create_entity("IfcText", str(self.inst.split('#')[-1])), None),
            self.ifcfile.createIfcPropertySingleValue("Maintenance by", "Maintenance by",
            self.ifcfile.create_entity("IfcText", str(self.MP.split('#')[-1])), None),
            self.ifcfile.createIfcPropertySingleValue("Operating party", "Operating party",
            self.ifcfile.create_entity("IfcText", str(self.OP.split('#')[-1])), None)]
# Create IfcPropertySet element and IfcRelDefinesByProperties element
        property_set =
self.ifcfile.createIfcPropertySet(self.create_guid(),self.owner_history,"Pset_VgibOM",None,pro
perty_values)

self.ifcfile.createIfcRelDefinesByProperties(self.create_guid(),self.owner_history,None,None,[
comp],property_set)
        for storey in self.ifcfile.by_type("IfcBuildingStorey"):
            if storey.LongName == self._storey[guid_selection][0]:
                self.ifcfile.createIfcRelContainedInSpatialStructure(self.create_guid(),
self.owner_history, "Building Storey Container", None, [comp], storey)

    def create_log_file(self):
#Sum up all of the log entries and write them to CSV file
        self.g1.serialize("Log_test.rdf", format='xml')
        file_exists = os.path.isfile("output_log.csv")
        with open("output_log.csv", "ab") as output:
            headers = ['Wie?','Wanneer?','Waar?','Wat?','Waarom?']
            writer = csv.DictWriter(output, delimiter=';', fieldnames=headers)
            if not file_exists:
                writer.writeheader()
            for row in self.g1.query("""
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX log: <http://example.org/ont/log#>
SELECT ?who ?when ?where ?what ?why WHERE { ?i log:reportedBy ?who .
                                            ?i log:dateTime ?when .
                                            ?i log:where ?where .
                                            ?i log:component ?what .
                                            ?i log:reason ?why .
                                            ?i rdf:type log:Report . }"""):
                writer.writerow({'Wie?':row[0].split('#')[-1], 'Wanneer?':row[1],
'Waar?':row[2].split('#')[-1], 'Wat?':(row[3].split('#')[-1]).split('/',1)[0],
'Waarom?':row[4]})
            print "Log generated successfully!"

    def switch_rooms_elements(self):
        products = self.ifc_file.by_type("IfcProduct")
        self.parent.display.EraseAll()
        self.propertybox.clear()
```

```
        for product in products:
            if product.is_a("IfcOpeningElement"):continue
            if product.is_a("IfcSpace"):continue
            if product.Representation:
                ifcgeom = ifcopenshell.geom.create_shape(settings, product).geometry
                self.canvas.Show(product.GlobalId, ifcgeom)

    def closeEvent(self,event):
        result = QtGui.QMessageBox.question(self,
                      "Confirm Exit",
                      "Are you sure you want to exit ?",
                      QtGui.QMessageBox.Yes| QtGui.QMessageBox.No)
        event.ignore()

        if result == QtGui.QMessageBox.Yes:
            event.accept()
            self.rdf_graph.serialize("Ketsheuvel_output_%s.rdf"%self.current_date,
format='xml')
            self.ifcfile.write(CreateIFCElement.filename)

init = initUI()
```

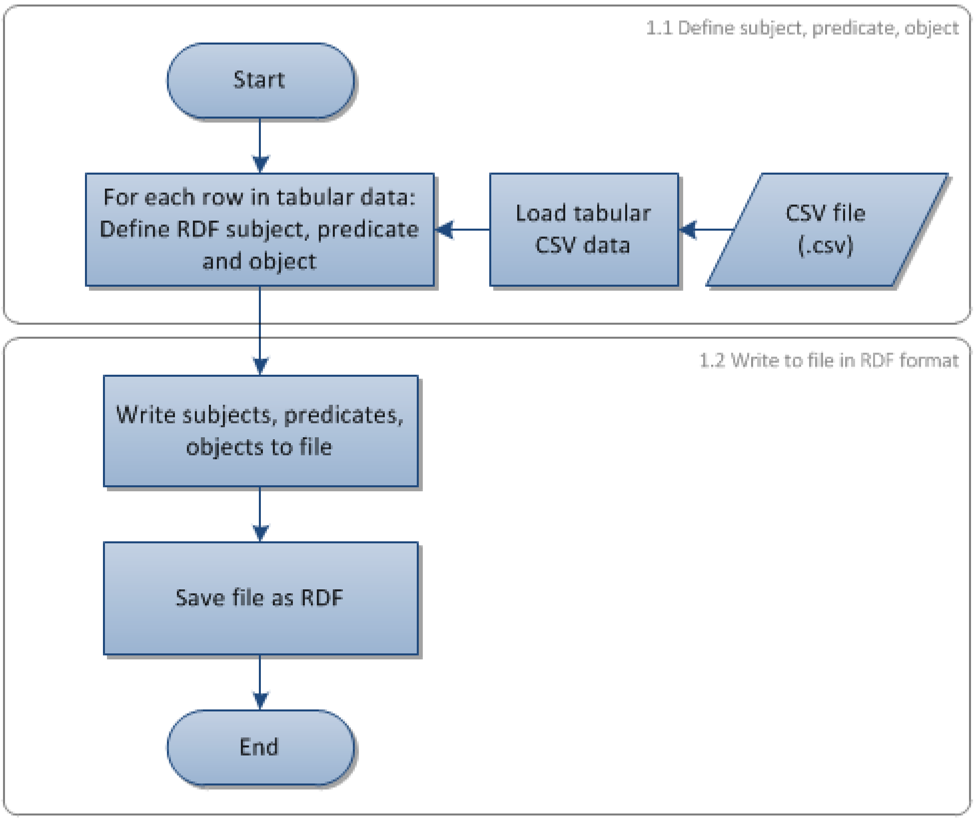
Appendix II.II: Helper functions for creating IFC files

```
import uuid
import time
import tempfile
import ifcopenshell

O = 0., 0., 0.
X = 1., 0., 0.
Y = 0., 1., 0.
Z = 0., 0., 1.

# Creates IfcAxis2Placement3D from Location, Axis and RefDirection specified as Python tuples
def create_ifcaxis2placement(ifcfile, point=O, dir1=Z, dir2=X):
    point = ifcfile.createIfcCartesianPoint(point)
    dir1 = ifcfile.createIfcDirection(dir1)
    dir2 = ifcfile.createIfcDirection(dir2)
    axis2placement = ifcfile.createIfcAxis2Placement3D(point, dir1, dir2)
    return axis2placement

# Creates IfcLocalPlacement from Location, Axis and RefDirection, specified as Python tuples,
and relative placement
def create_ifclocalplacement(ifcfile, point=O, dir1=Z, dir2=X, relative_to=None):
    axis2placement = create_ifcaxis2placement(ifcfile,point,dir1,dir2)
    ifclocalplacement2 = ifcfile.createIfcLocalPlacement(relative_to,axis2placement)
    return ifclocalplacement2

# Creates an IfcPolyLine from a list of points, specified as Python tuples
def create_ifcpolyline(ifcfile, point_list):
    ifcpts = []
    for point in point_list:
        point = ifcfile.createIfcCartesianPoint(point)
        ifcpts.append(point)
    polyline = ifcfile.createIfcPolyLine(ifcpts)
    return polyline

# Creates an IfcExtrudedAreaSolid from a list of points, specified as Python tuples
def create_ifcextrudedareasolid(ifcfile, point_list, ifcaxis2placement, extrude_dir,
extrusion):
    polyline = create_ifcpolyline(ifcfile, point_list)
    ifcclosedprofile = ifcfile.createIfcArbitraryClosedProfileDef("AREA", None, polyline)
    ifcdir = ifcfile.createIfcDirection(extrude_dir)
    ifcextrudedareasolid = ifcfile.createIfcExtrudedAreaSolid(ifcclosedprofile,
ifcaxis2placement, ifcdir, extrusion)
    return ifcextrudedareasolid

create_guid = lambda: ifcopenshell.guid.compress(uuid.uuid1().hex)


# IFC template creation

filename = time.strftime("components_%Y-%m-%d.ifc")
timestamp = time.time()
```

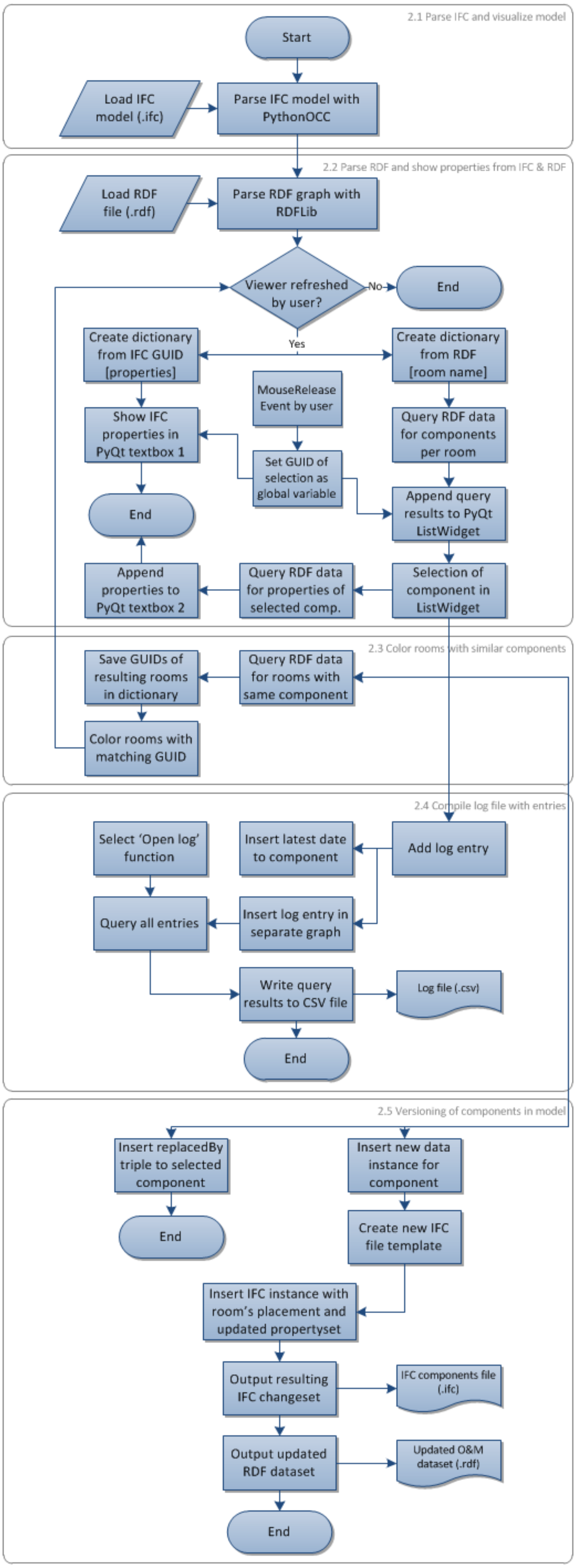
```
timestring = time.strftime("%Y-%m-%dT%H:%M:%S", time.gmtime(timestamp))
creator = "Niels van de Ven"
organization = "Vgib"
application, application_version = "IfcOpenShell", "0.5"
project_globalid, project_name = create_guid(), "Ketsheuvel componenten"

# A template IFC file to quickly populate entity instances for an IfcProject with its
dependencies
template = """ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('ViewDefinition [CoordinationView]'),'2;1');
FILE_NAME('%(filename)s','%(timestring)s',('%(creator)s'),('%(organization)s'),'%(application)
s','%(application)s','');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;
DATA;
#1=IFCPERSON($,$,'%(creator)s',$,$,$,$,$);
#2=IFCORGANIZATION($,'%(organization)s',$,$,$);
#3=IFCPERSONANDORGANIZATION(#1,#2,$);
#4=IFCAPPLICATION(#2,'%(application_version)s','%(application)s','');
#5=IFCOWNERHISTORY(#3,#4,$,.ADDED.,$,#3,#4,%(timestamp)s);
#6=IFCDIRECTION((1.,0.,0.));
#7=IFCDIRECTION((0.,0.,1.));
#8=IFCCARTESIANPOINT((0.,0.,0.));
#9=IFCAXIS2PLACEMENT3D(#8,#7,#6);
#10=IFCDIRECTION((0.,1.,0.));
#11=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.E-05,#9,#10);
#12=IFCDIMENSIONALEXPONENTS(0,0,0,0,0,0,0);
#13=IFCSIUNIT(*,.LENGTHUNIT.,$,.METRE.);
#14=IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#15=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#16=IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
#17=IFCMEASUREWITHUNIT(IFCPLANEANGLEMEASURE(0.017453292519943295),#16);
#18=IFCCONVERSIONBASEDUNIT(#12,.PLANEANGLEUNIT.,'DEGREE',#17);
#19=IFCUNITASSIGNMENT((#13,#14,#15,#18));
#20=IFCPROJECT('%(project_globalid)s',#5,'%(project_name)s',$,$,$,$,(#11),#19);
ENDSEC;
END-ISO-10303-21;
""" % locals()

# Write the template to a temporary file
temp_handle, temp_filename = tempfile.mkstemp(suffix=".ifc")
with open(temp_filename, "wb") as f:
    f.write(template)
```

Appendix III: Flowcharts of both parts of tool

Flowchart tool Tab A

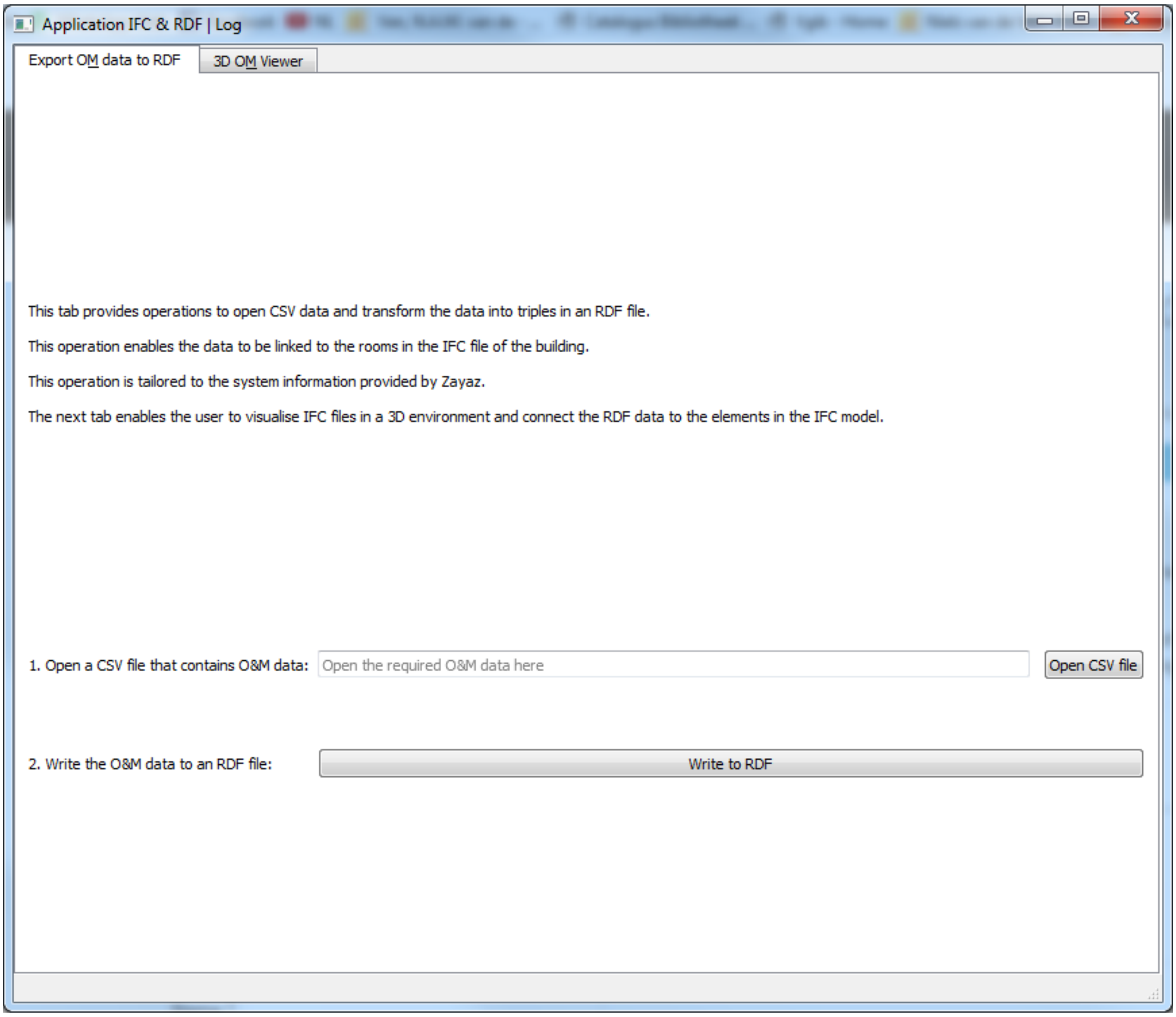


Flowchart tool Tab B

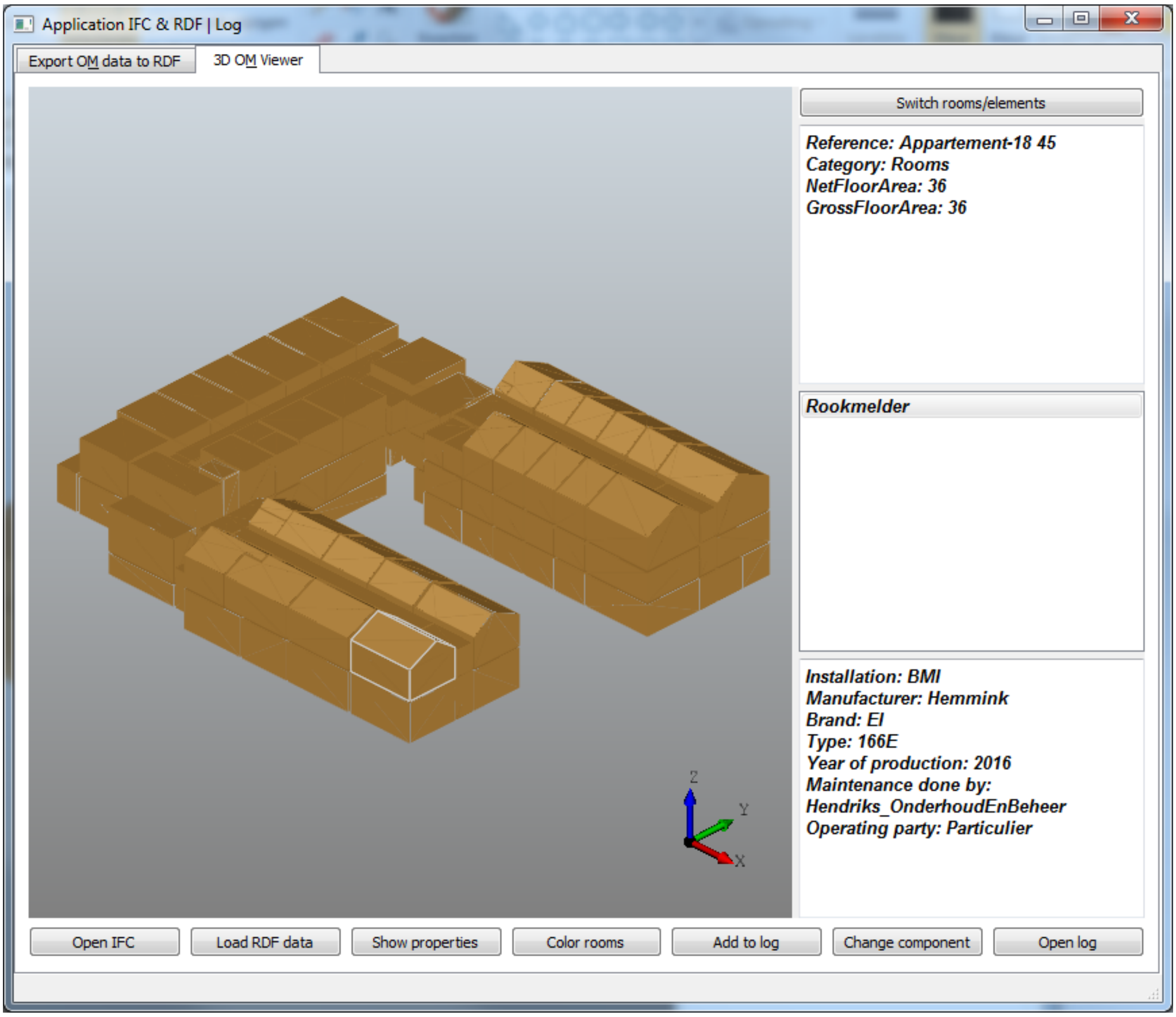


Appendix IV: Graphical User Interfaces (GUIs)

GUI Tab A



GUI Tab B


Application IFC & RDF | Log
Export OM data to RDF
3D OM Viewer
Switch rooms/elements
Reference: Appartement-18 45
Category: Rooms
NetFloorArea: 36
GrossFloorArea: 36
Rookmelder
Installation: BMI
Manufacturer: Hemmink
Brand: EI
Type: 166E
Year of production: 2016
Maintenance done by:
Hendriks_OnderhoudEnBeheer
Operating party: Particulier
Z
Y
X
Open IFC
Load RDF data
Show properties
Color rooms
Add to log
Change component
Open log

Appendix V: Ontologies (.ttl)

Ontology for building systems

```
# baseURI: http://example.org/ont/buildingsystems

@prefix busy: <http://example.org/ont/buildingsystems#> .
@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#> .
@prefix dcterms: <http://purl.org/dc/terms/>

<http://example.org/ont/buildingsystems>
  rdf:type owl:Ontology ;
  owl:versionInfo "Created with TopBraid Composer"^^xsd:string ;
.
busy:BuildingComplex
  rdf:type owl:Class ;
  rdfs:label "building complex"^^xsd:string ;
  rdfs:subClassOf owl:Thing ;
.
busy:BuildingFromComplex
  rdf:type owl:Class ;
  rdfs:label "building belonging to complex"^^xsd:string ;
  rdfs:subClassOf busy:BuildingComplex ;
.
busy:Component
  rdf:type owl:Class ;
  rdfs:label "component of system"^^xsd:string ;
  rdfs:subClassOf busy:System ;
.
dcterms:Agent
  dcterms:description "Examples of Agent include person, organization and
software agent."@en ;
  rdf:type rdfs:Class ;
  rdfs:label "A resource that acts of has the power to act."^^xsd:string ;
.
busy:Room
  rdf:type owl:Class ;
  rdfs:label "room"^^xsd:string ;
  rdfs:subClassOf owl:Thing ;
.
busy:System
  rdf:type owl:Class ;
  rdfs:label "building system"^^xsd:string ;
  rdfs:subClassOf owl:Thing ;
.
busy:address
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:BuildingComplex ;
  rdfs:label "address data"^^xsd:string ;
  rdfs:range xsd:string ;
.
busy:brand
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "brand of component"^^xsd:string ;
  rdfs:range xsd:string ;
.
```

```
busy:certificatepresent
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:System ;
  rdfs:label "inspection certificate present"^^xsd:string ;
  rdfs:range xsd:boolean ;
.
busy:contains
  rdf:type owl:ObjectProperty ;
  rdfs:domain busy:BuildingFromComplex ;
  rdfs:domain busy:Room ;
  rdfs:label "contains"^^xsd:string ;
  rdfs:range busy:Component ;
  rdfs:range busy:Room ;
.
busy:description
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:BuildingComplex ;
  rdfs:label "Description"^^xsd:string ;
.
busy:hasFunction
  rdf:type owl:FunctionalProperty ;
  rdfs:domain busy:BuildingComplex ;
  rdfs:label "functional use"^^xsd:string ;
  rdfs:range xsd:string ;
.
busy:lastmaintained
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "last maintained at"^^xsd:string ;
  rdfs:range xsd:date ;
.
busy:maintenanceby
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "maintenance party"^^xsd:string ;
  rdfs:range dcterms:Agent ;
.
busy:manufacturer
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "manufacturerer of component"^^xsd:string ;
  rdfs:range xsd:string ;
.
busy:name
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:BuildingComplex ;
  rdfs:label "name"^^xsd:string ;
  rdfs:range xsd:string ;
.
busy:operationby
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "operating party"^^xsd:string ;
  rdfs:range dcterms:Agent ;
.
busy:ownedby
  rdf:type owl:ObjectProperty ;
  rdfs:domain busy:BuildingComplex ;
  rdfs:label "is owned by"^^xsd:string ;
  rdfs:range dcterms:Agent ;
.
```

```
busy:permitpresent
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:System ;
  rdfs:label "permit present?"^^xsd:string ;
  rdfs:range xsd:boolean ;
.
busy:rentedby
  rdf:type owl:ObjectProperty ;
  rdfs:domain busy:BuildingFromComplex ;
  rdfs:label "is rented by"^^xsd:string ;
  rdfs:range dcterms:Agent ;
.
busy:subpartof
  rdf:type owl:ObjectProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "component of system"^^xsd:string ;
  rdfs:range busy:System ;
.
busy:type
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "has type name"^^xsd:string ;
  rdfs:range xsd:string ;
.
busy:yearofproduction
  rdf:type owl:DatatypeProperty ;
  rdfs:domain busy:Component ;
  rdfs:label "year when installed"^^xsd:string ;
  rdfs:range xsd:date ;
.
```

Ontology for log entries

```
# baseURI: http://example.org/ont/log#

@prefix log: <http://example.org/ont/log#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://example.org/ont/log#>
  rdf:type owl:Ontology ;
  owl:versionInfo "Created with TopBraid Composer"^^xsd:string ;
.
log:Component
  rdf:type owl:Class ;
  rdfs:label "Which component is concerned?"^^xsd:string ;
  rdfs:subClassOf owl:Thing ;
.
log:DateTime
  rdf:type owl:DatatypeProperty ;
  rdfs:domain log:Report ;
  rdfs:label "When reported?"^^xsd:string ;
  rdfs:range xsd:dateTime ;
.
log:Report
  rdf:type owl:Class ;
  rdfs:label "One notification of maintenance/solving an
outage"^^xsd:string ;
  rdfs:subClassOf owl:Thing ;
```

```
.
log:Zone
  rdf:type owl:Class ;
  rdfs:label "Which room holds the component?"^^xsd:string ;
  rdfs:subClassOf owl:Thing ;
.
log:component
  rdf:type owl:ObjectProperty ;
  rdfs:domain log:Report ;
  rdfs:label "Where reported?"^^xsd:string ;
  rdfs:range log:Component ;
.
log:reason
  rdf:type owl:DatatypeProperty ;
  rdfs:domain log:Report ;
  rdfs:label "Why reported?"^^xsd:string ;
  rdfs:range xsd:string ;
.
log:reportedBy
  rdf:type owl:ObjectProperty ;
  rdfs:domain log:Report ;
  rdfs:label "Who reported?"^^xsd:string ;
  rdfs:range owl:NamedIndividual ;
.
log:where
  rdf:type owl:ObjectProperty ;
  rdfs:domain log:Report ;
  rdfs:label "Where reported?"^^xsd:string ;
  rdfs:range log:Zone ;
.
```

Appendix VI: Partial input (CSV)

Corporatie	Complexn	Gebouwe	Complexnaam	Toelichtin	Huurder pand	Functie ge	Adresgegevens	Ruimte	Installatie	Component	Beheerder installatie	Leverancier o	Fabrikant	Merk	Type	Bouwjaar	Datum laa
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-01	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	20-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-02	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	21-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-03	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	3-4-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-04	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	5-5-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-05	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	6-5-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-06	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	7-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-07	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	8-5-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-08	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	9-5-2017
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Gang	NV	Vluchtwegaanduiding	Zayaz	Hendriks_Onc	ElboTechr	KM	KMU003	2016	10-5-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Gang	NV	LED_armatuur	Zayaz	Hendriks_Onc	ElboTechr	DL	DLE023	2016	10-5-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Entree									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Entree	NV	Vluchtwegaanduiding	Zayaz	Hendriks_Onc	ElboTechr	KM	KMU003	2016	20-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Entree	NV	LED_armatuur	Zayaz	Hendriks_Onc	ElboTechr	DL	DLE023	2016	21-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Werkkast	NV	LED_armatuur	Zayaz	Hendriks_Onc	ElboTechr	Luxa	37624	2016	3-4-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Toilet_H									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Toilet_D									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Archief									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Lift									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Trappenhuis									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Berging_techniek									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Berging									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Technische_ruimte									
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Spreekkamer	NV	LED_armatuur	Zayaz	Hendriks_Onc	ElboTechr	DL	DLE023	2016	20-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-09	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	21-5-2016
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-10	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	3-4-2015
Zayaz	TCL-0xxx	Ordune	Ordune	Vergunnir	Particulier	Wonen	Ketsheuvel 12a,	Appartement-11	BMI	Rookmelder	Particulier	Hendriks_Onc	Hemmink	EI	166E	2016	5-5-2015

Appendix VII: Partial RDF Input


```xml
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
   xmlns:busy="http://example.org/ont/buildingsystems#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#LED_armatuur/0ec2ee12-7f6c-4ff6-b605-ac987886db1d">
    <busy:yearOfProduction rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2016</busy:yearOfProduction>
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Component"/>
    <busy:maintenanceBy rdf:resource="http://example.org/id/buildingsystems#Hendriks_OnderhoudEnBeheer"/>
    <busy:lastMaintained rdf:datatype="http://www.w3.org/2001/XMLSchema#date"></busy:lastMaintained>
    <busy:operationBy rdf:resource="http://example.org/id/buildingsystems#Zayaz"/>
    <busy:manufacturer rdf:resource="http://example.org/id/buildingsystems#ElboTechnology"/>
    <busy:brand rdf:resource="http://example.org/id/buildingsystems#DL"/>
    <busy:type rdf:datatype="http://www.w3.org/2001/XMLSchema#string">DLE023</busy:type>
    <busy:subpartOf rdf:resource="http://example.org/id/buildingsystems#NV"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Appartement-25_68">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#Rookmelder/bc54459d-7a98-45dd-b40d-3d109d995452"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Entree_10">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#/bc89cf18-2838-4d5e-a6f6-b99e55df2888"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Rookmelder/744df71a-8a43-4e40-9296-3c33b50b5691">
    <busy:brand rdf:resource="http://example.org/id/buildingsystems#EI"/>
    <busy:manufacturer rdf:resource="http://example.org/id/buildingsystems#Hemmink"/>
    <busy:subpartOf rdf:resource="http://example.org/id/buildingsystems#BMI"/>
    <busy:type rdf:datatype="http://www.w3.org/2001/XMLSchema#string">166E</busy:type>
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Component"/>
    <busy:maintenanceBy rdf:resource="http://example.org/id/buildingsystems#Hendriks_OnderhoudEnBeheer"/>
    <busy:operationBy rdf:resource="http://example.org/id/buildingsystems#Particulier"/>
    <busy:yearOfProduction rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2016</busy:yearOfProduction>
    <busy:lastMaintained rdf:datatype="http://www.w3.org/2001/XMLSchema#date"></busy:lastMaintained>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Appartement-30_74">
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#Rookmelder/8693a5ac-f390-4769-8750-01ead402be27"/>
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#TCL-0xxx">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#BuildingComplex"/>
    <busy:hasFunction rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Wonen</busy:hasFunction>
    <busy:ownedBy rdf:resource="http://example.org/id/buildingsystems#Zayaz"/>
    <busy:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Ordune</busy:name>
    <busy:address rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Ketsheuvel 12a, 5231 PT 's-Hertogenbosch</busy:address>
    <busy:description rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Vergunning voor 10 jaar, getransformeerd van kantoren naar woningen voor statushouders en starters.</busy:description>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Toilet_109">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#/a74ee7c0-ac0e-41b6-981d-4e03aa3e0852"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Appartement-17_44">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#Rookmelder/c4735bea-ce3e-452d-b18b-a0c56bf128ee"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Gang_92">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
```

```
    <busy:contains
rdf:resource="http://example.org/id/buildingsystems#Vluchtwegaanduiding/5936b997-1d47-426d-
b667-2a5859ffe255"/>
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#LED_armatuur/02b385f9-
9535-49db-8654-6846c5ee94e5"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://example.org/id/buildingsystems#Hendriks_OnderhoudEnBeheer">
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Organization"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#Toilet_H_95">
    <busy:contains rdf:resource="http://example.org/id/buildingsystems#/0380460d-f477-4295-
b6b2-3cc1d1016ada"/>
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Room"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/id/buildingsystems#LED_armatuur/945573ad-
79b5-4059-b60c-fe754d8fac9d">
    <busy:yearOfProduction
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2016</busy:yearOfProduction>
    <busy:subpartOf rdf:resource="http://example.org/id/buildingsystems#NV"/>
    <busy:maintenanceBy
rdf:resource="http://example.org/id/buildingsystems#Hendriks_OnderhoudEnBeheer"/>
    <busy:type rdf:datatype="http://www.w3.org/2001/XMLSchema#string">DLE023</busy:type>
    <busy:manufacturer rdf:resource="http://example.org/id/buildingsystems#ElboTechnology"/>
    <busy:lastMaintained
rdf:datatype="http://www.w3.org/2001/XMLSchema#date"></busy:lastMaintained>
    <busy:brand rdf:resource="http://example.org/id/buildingsystems#DL"/>
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Component"/>
    <busy:operationBy rdf:resource="http://example.org/id/buildingsystems#Zayaz"/>
  </rdf:Description>
  <rdf:Description
rdf:about="http://example.org/id/buildingsystems#Vluchtwegaanduiding/95d9bda8-55c6-4c92-b5dd-
334067a016c6">
    <busy:maintenanceBy
rdf:resource="http://example.org/id/buildingsystems#Hendriks_OnderhoudEnBeheer"/>
    <busy:type rdf:datatype="http://www.w3.org/2001/XMLSchema#string">KMU003</busy:type>
    <rdf:type rdf:resource="http://example.org/ont/buildingsystems#Component"/>
    <busy:operationBy rdf:resource="http://example.org/id/buildingsystems#Zayaz"/>
    <busy:subpartOf rdf:resource="http://example.org/id/buildingsystems#NV"/>
    <busy:lastMaintained
rdf:datatype="http://www.w3.org/2001/XMLSchema#date"></busy:lastMaintained>
    <busy:brand rdf:resource="http://example.org/id/buildingsystems#KM"/>
    <busy:manufacturer rdf:resource="http://example.org/id/buildingsystems#ElboTechnology"/>
    <busy:yearOfProduction
rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2016</busy:yearOfProduction>
  </rdf:Description>
</rdf:RDF>
```