MASTER

The development of a Microsoft Excel based model checker to verify the completeness of a building model
research to the exchange of information between customer and supplier of a building process

van Dun, A.

Award date:
2017

Eindhoven University of Technology

The development of a Microsoft Excel based model checker to verify the completeness of a building model

Research to the exchange of information between customer and supplier of a building process

A. van Dun
Construction Management and Engineering
March 15, 2017

In collaboration with: Hurks bouw Eindhoven

Graduation committee

Prof.dr.ir. B. (Bauke) de Vries	University supervisor (chairman graduation committee)
Dr.dipl.-ing. J. (Jakob) Beetz	University supervisor
Ir. T.F. (Thomas) Krijnen	University supervisor
Ing. M.J.W.C. (Machiel) van den Brink	Company supervisor
Ir. R.H.P. (Rob) Verweij	Company supervisor

Colophon

General

Report:	The development of a Microsoft Excel based model checker to verify the completeness of a building model
Subtitle	Research to the exchange of information between customer and supplier of a building process
Date:	March 15, 2017
Place:	Eindhoven

Student:

Author:	Astrid van Dun
Student number:	0743848
University:	Eindhoven University of Technology
Master track:	Construction Management and Engineering
Chair:	Information Systems in the Built Environment

Graduation company

Company:	Hurks Bouw Eindhoven
Division:	Da Vinci Huis

Supervisors

Chairman
Prof.dr.ir. B. (Bauke) de Vries

First supervisor
Dr.dipl.-ing. J. (Jakob) Beetz

Second supervisor
Ir. T.F. (Thomas) Krijnen

Company supervisors
Ing. M.J.W.C. (Machiel) van den Brink
Ir. R.H.P. (Rob) Verweij

Preface

This report is the result of my graduation research carried out in collaboration with the Eindhoven University of Technology and Hurks Bouw Eindhoven. This thesis forms my final work for the master Construction Management and Engineering. It was a real learning experience to set up and conduct the complete research. I am happy to have challenged myself and that I learnt a lot about a promising field of the construction industry.

I want to thank the company Hurks Bouw Eindhoven for giving me the opportunity and support to conduct my graduation research. Special thanks to Machiel van den Brink and Rob Verweij for their input and guidance.

I also want to thank my university supervisors, especially Jakob Beetz always provided me new insights concerning my graduation subject.

Last but not least, I would like to thank my family and friends for pushing me towards my goal.

Contents

List of figures and tables

List of figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

List of tables

Summary

A good collaboration between the involved parties is key for the succeeding of a construction project. However, there is still room for improvement at the current traditional process. Due to a lack of unambiguously communication, mistakes are made which could have been prevented in the first place.

Applying BIM (Building Information Modeling) in the construction process is a way to, amongst others, prevent construction flaws. BIM is a methodolgy to accomplish integral design, construction and maintenance, resulting in a complete digital description of a building project. The essence of BIM is to inform and involve all partners of the building project and let them speak the same comprehensible language, which has a positive effect on the collaboration between all stakeholders.

However, the involved parties all have various preferences regarding the use of software. In order to be able to exchange information amongst the parties, the IFC schema is conceived. IFC (Industry Foundation Classes) is an open file format so the differences in file formats does not necessarily forms an obstacle for the exchange of data.

It is difficult to control the export from the native file to an IFC file. While it is of great importance for the user to trust that the essential information kept present in the IFC model. For this reason, model checkers are developed, a model checker validates the presence of certain information in the IFC file.

This problem definition resulted into the following main research question: 'how can the completeness of a building model be verified using a model checker?'. In order to be able to answer this question, a research has been conducted to the exchange of information between customer and supplier in a building process.

At first, it is determined at which particular building process the research is conducted. The thesis focuses on the exchange of data between customer and supplier of prefab concrete wall elements of the Da Vinci Huis process. Aim of the Da Vinci Huis concept is to combine the advantages of a standardized house, with the desire for freedom of choice in the design.

In order to verify the completeness of a building model, it has to be established when a building model can be considered complete. Based on interviews with the contractor and supplier of prefab wall elements, a list of parameters which the wall elements should comply to is created, so the building model can be considered complete.

Based on these parameters, and the preset requirements that the model checker should be user-friendly and open-sourced, initially is decided to develop a model checker based on the mvdXML schema. After applying the mvdXML checker it is concluded that the output of the checker, rarely corresponded with the outcome it should show. For this reason, the checker based on mvdXML is regarded as unreliable.

Since the mvdXML checker is considered not reliable enough, another type of model checker has been developed. The choice for the new checker has been based on the same principles as stated before. A model checker based on Microsoft Excel came out as the best option, mainly because applying it requires a low threshold for the user.

To use the model checker in Excel, the first step that needs to be taken is the export from the original file to IFC, important is to taken into account certain export settings. After the preparation of the IFC it is converted to an Excel file with the use of the IFC file analyzer of (Lipman, 2010).

Next step is to perform the check. At first the specifications about what needs to be checked should be indicated in the main sheet of the Excel workbook. The name of the Excel, the element which needs to be checked and the parameters the presence is checked on are addressed in this main sheet. After the performance of the check, the outcome is displayed on three different levels of detail.

Main objective of the thesis is to verify the completeness of a building model. After conducting the research it can be concluded that by developing a Microsoft Excel based checker the completeness of a building model can be verified. A low-threshold, easy to use model checker is developed. The model checker is able to verify the completeness of building projects, under the circumstances that the files are not too big and the information is stored in the right way.

Samenvatting

Een goede samenwerking tussen de verschillende bouwpartijen is van groot belang voor het slagen van een project. Echter is er in het huidige bouwproces nog ruimte voor verbetering. Door gebrek aan eenduidige communicatie worden er fouten gemaakt die in eerste instantie voorkomen hadden kunnen worden.

Een middel om bijvoorbeeld deze bouwfouten te voorkomen, is het gebruiken van BIM (Bouw Informatie Model) in het proces. BIM is onder andere een manier van werken waarbij in één bouwmodel integraal wordt samengewerkt met alle belanghebbenden, met als gevolg dat alle informatie op één plek te vinden is. Dit heeft een positief effect op de samenwerking tussen alle belanghebbenden.

Echter kunnen de verschillende partijen allemaal andere voorkeuren hebben met betrekking tot het gebruik van software. Om toch informatie onderling uit te kunnen wisselen is het IFC schema ontwikkeld. IFC (Industry Foundation Classes) is een open bestandsformaat ontwikkeld zodat de onderlinge verschillen geen belemmering hoeft te vormen voor het uitwisselen van informatie.

Het is alleen lastig te controleren of de export van het originele bestand naar IFC goed verloopt. Het is van groot belang dat de gebruiker erop kan vertrouwen dat de essentiële informatie aanwezig is in het IFC model omdat dit anders grote gevolgen kan hebben voor de kwaliteit van het bouwproject. Om die reden zijn er model checkers ontwikkeld. Een model checker controleert de aanwezigheid van bepaalde informatie in het IFC bestand zodat de gebruikter hier meer inzicht over heeft.

Deze probleemstelling heeft geleid tot de volgende hoofdonderzoeksvraag: ‘hoe kan de volledigheid van een bouwmodel geverifieerd worden door middel van een model checker?’. Om deze vraag te kunnen beantwoorden is er onderzoek gedaan naar het uitwisselen van informatie tussen klant en leverancier in een bouwproces.

Eerst is het specifieke bouwproces waar het onderzoek plaats vindt bepaalt. Het onderzoek heeft zich gericht op de bestandsuitwisseling tussen klant en leverancier van prefab betonnen wandelementen van het Da Vinci Huis proces. Het Da Vinci Huis is een woonconcept van Hurks Bouw Eindhoven die door middel van standaardisatie efficiëntie bereikt.

Om de volledigheid van een bouwmodel te kunnen verifiëren is het van belang vast te stellen wanneer een bouwmodel compleet is. Aan de hand van interviews met zowel de aannemer als de leverancier van de wandelementen is er een lijst met eigenschappen opgesteld die het bouwmodel moet bevatten, voordat het als compleet kan worden beschouwd.

Gebaseerd op deze eigenschappen, en de vooropgestelde eisen dat de model checker gebruiksvriendelijk en open sourced moet zijn, is er in eerste instantie voor gekozen om een model checker gebaseerd op het mvdXML schema te ontwikkelen. Na het in gebruik nemen van de mvdXML checker is echter geconcludeerd dat de output van de checker zelden overeen kwam met de uitkomst die het zou moeten laten zien. Om deze reden is de checker gebaseerd op mvdXML als te onbetrouwbaar geacht en wordt er in dit verslag niet verder op ingegaan.

Omdat de bovengenoemde checker niet betrouwbaar genoeg bleek te zijn, is er een andere model checker ontwikkeld. De keuze voor de nieuwe checker is gebaseerd op dezelfde uitgangspunten, als hierboven beschreven. Hieruit is een model checker in Microsoft Excel als beste optie uitgekomen, met name omdat deze zeer laagdrempelig in het gebruik is.

De eerste step voor het gebruik van de model checker in Excel het exporteren van het originele bouwmodel naar IFC. Na het gereedmaken van de IFC wordt deze middels de IFC file analyzer van (Lipman, 2010) geconverteerd naar een Excel bestand, en is het bouwmodel klaar om gecheckt te worden.

Voor het uitvoeren van de check worden eerst de specificaties over wat er gechekt moet worden aangegeven in de allereerste sheet van de Excel workbook. Hieronder vallen: de naam van het IFC Excel, het te checken element en de te checken eigenschappen. Na het uitvoeren van de check kan de uitkomst van de check op drie verschillende detailniveaus bekeken worden.

Het hoofddoel van deze thesis is het verifiëren van de volledigheid van een bouwmodel. Na het onderzoek te hebben verricht kan worden geconcludeerd dat dit door middel van de ontworpen Microsoft Excel checker mogelijk is. Er is een laagdrempelige model checker ontwikkeld die overzichtelijk is in het gebruik. De checker is geschikt om de volledigheid van bouwprojecten te verifiëren, mits de bestanden niet al te groot zijn en de informatie op de juiste manier is opgeslagen.

1

Introduction

The building industry is an environment in which collaboration is key for the success of a project. The industry requires constant exchanges of data and communication among different formats and stakeholders. New criteria are developed regularly, ranging from building codes and safety rules to techniques of fabrication. As the complexity level of the design and construction processes is increasing, traditional information resources, such as paper-based documents, cannot satisfy the requirements nowadays (Zhang Chi, 2015). Since the traditional building process does not satisfy the current developments and in order to increase productivity and quality, it is essential to shift from the current paper-based building process to a digital model-based building process. The concept of Building Information Modeling (BIM) is conceived to achieve this. This concept forms the base element for the graduation thesis.

BIM is a methodolgy to accomplish integral design, construction and maintenance, resulting in a complete digital description of a building project. The essence of BIM is to involve all partners of the building project and let them speak the same comprehensible language. As a consequence communication issues caused by confusion will drop down.

This graduation project intends to address the topic of BIM-interoperability within an organization. This work is carried out with the support of the company Hurks Bouw.

1.1 Motivation

Building Information Modeling (BIM) is a promising development in the architecture, engineering and construction (AEC) industry. The benefits it can provide when implemented in the building process are extensive. BIM is becoming very important for the construction process in the future. Uniform information, which can be reused in future projects, leads to more quality and a decrease in the amount of errors made, since the building has been built digitally on forehand. However, it is not very easy to put BIM in practice due to the novelty of the topic and the large complexity which shifting from a traditional process to a BIM process implies (Xiao & Noble, 2016).

The implementation of BIM in the construction process forms an interesting topic for graduation. On the one hand the advantages of applying BIM are, in theory, huge, on the other hand resentment against the implementation of BIM in the process exists. This gap between the novelty and innovative character of BIM and the current traditional construction process forms an interesting subject to research how to overcome these problems and put the theory into practice.

1.2 Problem definition

Goal of this section is to define the research problem by further specifying the most general problem and their solutions in the field of BIM, after the definition of the hypothesis, the research question can be established in chapter 1.4.

The problem definition starts at a very broad level and is specified further by defining the matching solutions. Every problem has its solution, though, every solution of a problem comes along with its own problem. The principle of this concept is visualized in Figure 1.2-1, the conceptual definition of the research problem is visualized by an inversed pyramide, which represents the specification of the problem starting at a broad level and ending with a much more defined solution.

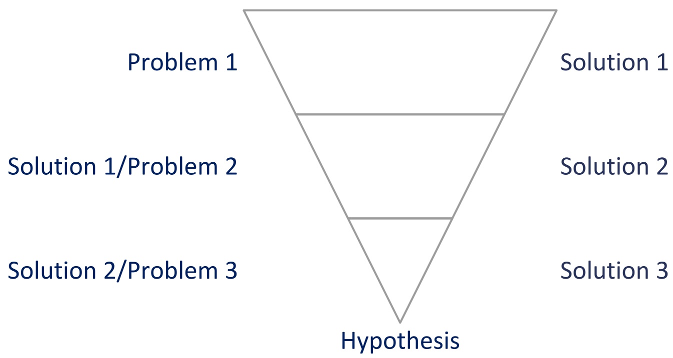


Figure 1.2-1: Principle of problem definition

Starting point is the current situation; the traditional building process. A research conducted by the company Tardif, Murray & Associates (a construction company from Canada) shows a good example of the complexity of the traditional building process in large projects. The construction company counted the number of participants and documents in one of their projects (Eastman, et al., 2011). As a result, they ended up with 420 participant companies, including all suppliers and sub-sub-contractors, 850 individual participants, 50 types of documents generated which total up

56.000 pages. This complexity level and the large amount of data makes such projects difficult to manage, the use of BIM would be ideal for this large projects, since all data is stored in one accessible place.

BIM does not only offer advantages when applied in big projects. The use of one accessible model leads to advantages for smaller projects as well, mainly concerning cost reduction, improved communication and a clearer structure. It is assumed that when BIM is applied in a correct way it offers many advantages compared to the traditional process.

With the use of BIM in the building process, the problem of interoperability comes along. Every participant in the process makes use of their own software. These are designed and developed to meet the wishes and requirements of the user. As a result, it is difficult to exchange files between participants of the building process since file formats differ from each other. Because of the differences in requirements of the users, combining all software into one would be inconvenient. For that reason, one common data exchange format for interoperability is conceived; the Industry Foundation Classes (IFC).

IFC is conceived to increase the interoperability between stakeholders. However, the issue regarding IFC is the high redundancy. It offers (too) many ways to define objects, relations, and attributes. As a result, users are not willing to rely on and accept IFC (Venugopal, Eastman, Sacks, & Teizer, 2012).

A solution for this problem is a model checker, this validation tool verifies the completeness by checking if certain information is stored in the IFC or not. An IFC checker offers potential as a guidance tool to control the use of IFC. Figure 1.2-2 shows this principle.

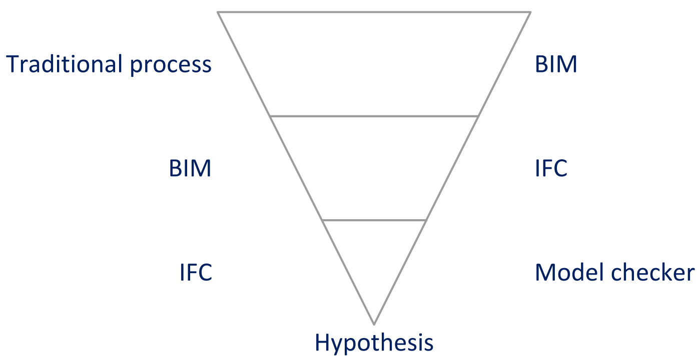


Figure 1.2-2: Problem definition

This leads to the following hypothesis:

A model checker can contribute to a more efficient construction process

1.3 Research scope

This research is applied at a process of the company Hurks Bouw Eindhoven. Since both the subject and the process of the whole company are way too broad, more specification is required. The main subject of the thesis is model checking, within this subject the thesis is focused on the exchange of data. Within the subject of exchange of data, the focus lies on the verification of parameters within an IFC file.

The research is conducted at the Da Vinci Huis department of the company Hurks Bouw Eindhoven. The focus within the construction process of the Da Vinci Huis, lies on the exchange of information between the contractor Hurks and the supplier of prefab concrete walls, Holcim.

1.4 Research Questions

From the problem definition, the research scope and the hypothesis the research questions has been drawn up. The main research question is:

> *How can the completeness of a building model be verified using a model checker?*

The subtitle is:

> *Research to the exchange of information between customer and supplier in a construction process.*

Several sub questions are developed to support the main research question. The sub-questions are categorized per chapter; 2. Literature review, 3. Data collection and 4. Model Design. Each sub question is briefly discussed below:

Sub question chapter 2, Literature review
It is essential to gain background knowledge before continuing with the rest of the thesis. First step in this research thesis is to provide understanding in the current situation on the research topic of automation and validation of exchange requirements of building elements. Goal of the following sub question is to get more insight about the fundamentals of model checking. The corresponding sub-research question is:

1. *Which key concepts regarding model checking can be distinguished?*

Sub question schapter 3, Data collection
At the current state it is not possible to provide an answer to the main research question, as stated in the beginning of this section since it requires a context. For this reason the first step is to define the construction process the research is conducted at, the corresponding sub question is:

2. *How can the construction process be defined within the context of this research?*

To gain insight concerning which part of the defined process requires improvement and to collect all required data, the following sub question is set up:

3. *What are the differences between the current and ideal situation of the defined construction process?*

After collecting all required information concerning the construction process, it needs to be established which parameters have to be present in a building model before it can be considered complete. This leads to following sub question:

> 4. *Which parameters need to be present in a building model before it can be considered complete?*

Sub question chapter 4, Model Design
After the collection of the required data and the establishement of a complete model, it is elaborated upon the design of the model. At first is determined which type of model checker forms the base for the final validation tool. Followed up by a detailled explanation of this developed model checker. This corresponding sub question is:

> 5. *Which type of validation tool is suited to verify the completeness of a building model and how can it be applied?*

These sub-research questions lead to three final products:

1. *A model checker which can be applied in the company*
2. *Instructions about how to use the model checker*
3. *Recommendations to extend Da Vinci Huis configurator*

1.5 Research model

Goal of this graduation project is to find an answer to the research question. In order to answer this research question, the thesis is divided into three main chapters, an introduction and a conclusion where the final answer of the research question is provided. The structure of the research is shown in Figure 1.5-1. Furthermore, a textual description for each chapter is given in this part.

Chapter 2: Literature review
The first part of the research is about collecting information on the topics BIM and model checking. In order to gain this information, a literature study is conducted, to get more insight in the key elements concerning BIM. The goal is to answer the first sub question related to this part. The outcome is used in the following phases and serves as starting point for the rest of the thesis.

Chapter 3: Data collection
Aim of this chapter is to collect the data required to answer the main research question. At first the definition of construction process is further defined, since the term used in the research question is too broad before it can be applied in the thesis. For this reason, the first step is to specify the construction process the research is applied on.

After that a definition is provided about what the differences are between the current and ideal situation of the construction process of the Da Vinci Huis, by analysing this building process.
This analysis is about the exchange of prefab concrete wall elements of the Da Vinci Huis. This concept is suitable as a case study for multiple reasons: The building process of the Da Vinci Huis differs from other construction processes because it is an iterative process. In a traditional way of building the process and the building itself are every time unique, in contrast to the Da Vinci Huis. Because this concept took place in the past and takes place in the future, data is available as well improving the process of the concept is relevant since the outcome can be used for the improvement in future projects.

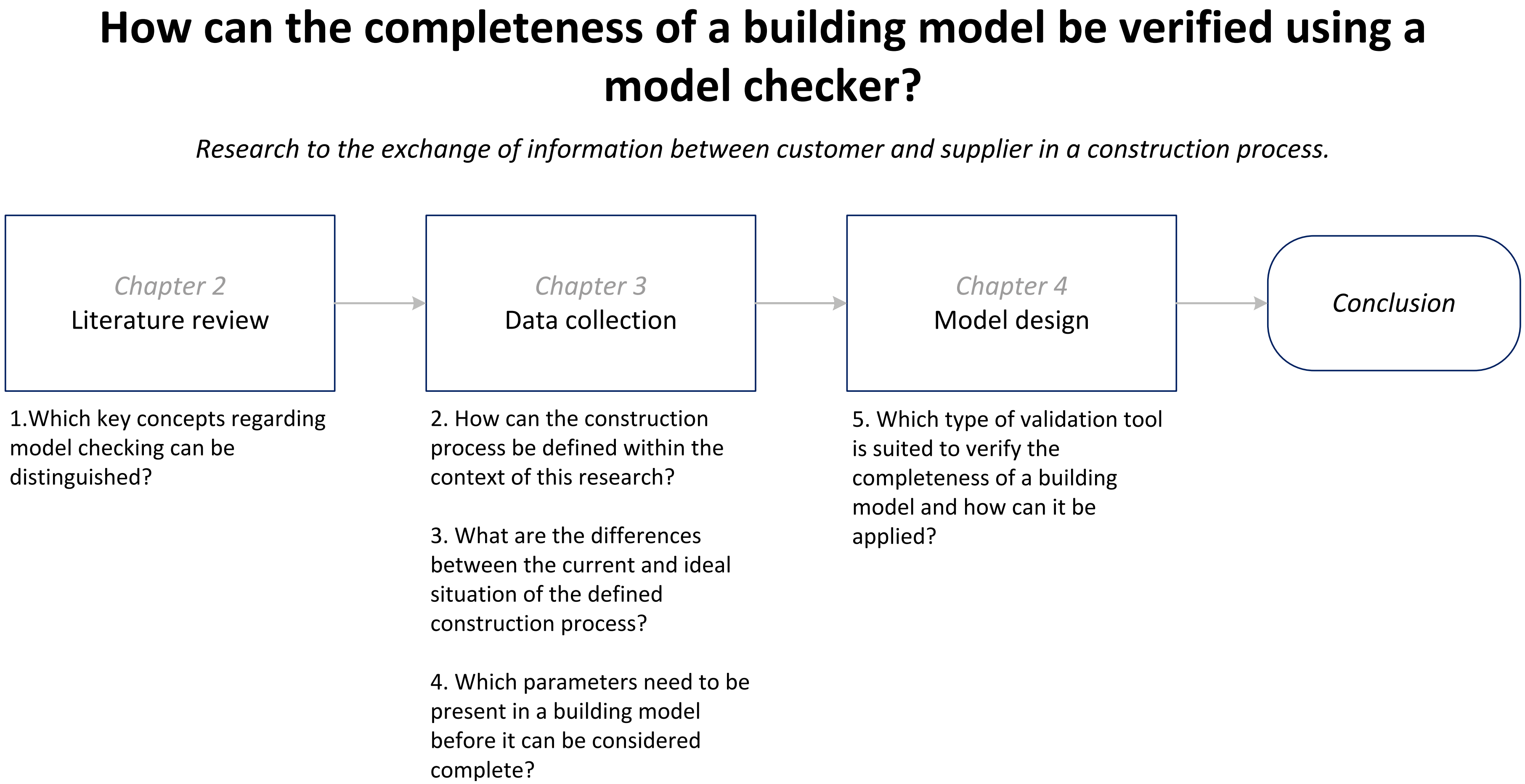


Figure 1.5-1: Structure of the research

When the construction process is established, more information about this process is required. This is done with the use of a Business Process Model and Notation (BPMN) diagram. Two situations are compared using this method; the current process and the ideal situation. The rest of the thesis is based on the differences between these two situations.

Next, it is essential to define a complete model. In order to be able to answer the main research question. This information is gained by interviewing all involved parties and by reviewing available data. These parameters form the base of the model checker.

Chapter 4: Model design
This chapter is about answering the last sub question: “Which type of validation tool is suited to verify the completeness of a building model and how can it be applied?” After the definition of a complete model, the next step is to choose a validation tool and apply it on the case. At first is elaborated upon the type of validation tool based on requirements. Next is explained how this model checker works.

1.6 Expected results

This graduation thesis aims to develop a model checkert to verify the completeness of a building model. It is expected that in the second chapter (literature review) more insight is given about the key concepts of model checking. In the third chapter the required data is collected for the development of the model checker. This is an interesting chapter since more insight regarding building processes is gained. The fourth chapter is expected to be the most challenging part. In this chapter a model checker is developed. Since I personally lack programming skills, the development of a tool will be challenging but also very informative.

2
Literature review

This literature review serves as starting point for the graduation thesis project of the master Construction Management and Engineering at the Eindhoven University of Technology. The literature review aims to provide understanding in the current situation on the research topic of automation and validation of exchange requirements of building elements.

Furthermore, the goal of the literature review is to give answer to the first sub question. The sub question elaborated in this chapter is:

1. Which key concepts regarding model checking can be distinguished?

Goal of this sub question is to get more insight about the fundamentals of model checking. The literature review starts with the general explanation of Building Information Modeling, the role of BIM in the building process, the advantages and disadvantages of the use of BIM and the applications. It is continued by a section about Data Interoperability where three initiatives, “Level of Development”, “BIM maturity level” and “Open standards”, are presented. Subsequently it is followed up by a sub-chapter about Industry Foundation Classes, where the application, difficulties and the structure are used to describe the Industry Foundation Classes (IFC). Furthermore, the standardization of IFC is described by using: “Information Delivery Manual”, “Model View Definition”, “International Framework for Dictionaries” and “BIM standards”. The last two sections deal with Model checking and Validation tools, where the types of model checking and validation tools are described and where is elaborated upon the model checking process. At the end of this chapter a conclusion is provided in which the answers of the first sub-research questions is presented.

2.2 BIM (Building Information Modeling)

The NBIMS (National BIM Standard) Initiative categorizes the Building Information Model (Eastman, Teicholz, Sacks, & Liston, 2011) in three ways:

1. *As a product*
2. *As an IT-enabled, open standards–based deliverable, and a collaborative process*
3. *As a facility lifecycle management requirement.*

This categorization is chosen because of its completeness. It claims that BIM is not just a product/a 3D model, but also a virtual representation of the actual building with elements containing information about its geometry, material type, costs, maintenance, fire resistance, location and so on.

In order to provide more insight about the main subject of this thesis, this sub-chapter provides information about the role of BIM in the building process, the advantages and disadvantages and the application of BIM.

The role of BIM in the building process

As mentioned in the introduction of this literature review, a shift from the traditional building process to a digital model-based process is required. Building Information Modeling is a concept which is developed to support this.

Traditionally in the architecture, engineering and construction industry, the exchange of information between participants of a project is done through paper or digital 2D drawings. This is a rather simple method but comes along with communication issues. In recent years as an answer to these communication issues, Building Information Modeling slowly started to become commonplace in the building profession.

The use of BIM in a project team affects all the stakeholders in their processes. It shifts from manually exchanging 2D drawings and non-digital communcation about the features of the building elements to finding all needed information in one model.

Models nowadays are observed as as a source of information in itself. Experienced users of BIM want to find information within a model itself and want to compare models in order to evaluate states of a model, differences in separate models or models from different points of time (Tamke, Jensen, Beetz, Krijnen, & Fjeld, n.d.).

The shift from a traditional process to a BIM excecution is schematically shown in Figure 2.2-1, which is based on (Sagarkar, n.d.). In the traditional approach all stakeholders are connected with each other, at the BIM approach everyone has access to the Information model where all information is saved.

Building Information Modeling is not only a product or model but also a way of working. With a much clearer structure it is strived to contribute to an improved process in multiple ways.

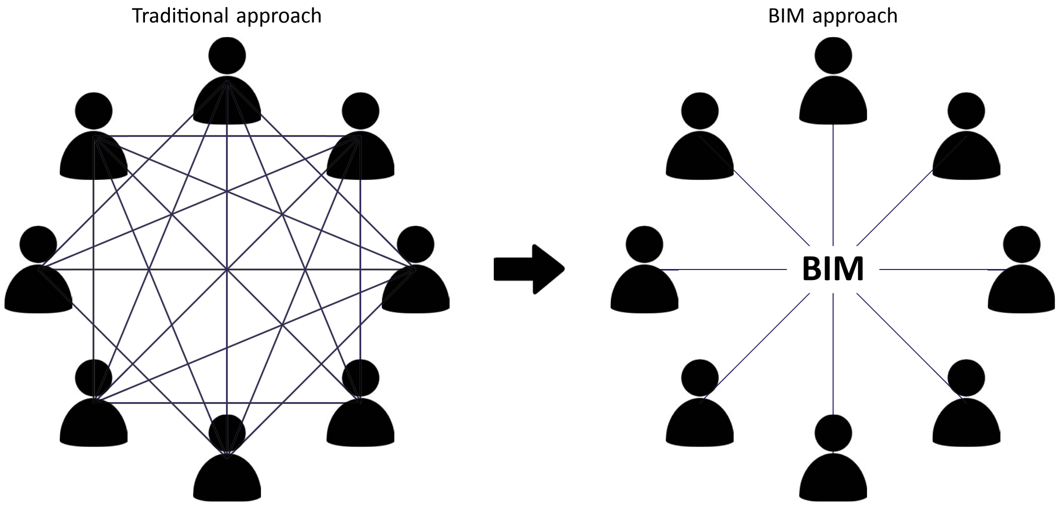


Figure 2.2-1: Shift from traditional process to a BIM approach

The exchange of information through Building Information Modeling provides improved sharing capabilities with large benefits. BIM can be of great value for all members of a project team, including architects, engineers, contractors and sub-contractos, facility owners and executors, as well as for building product suppliers who can model their products so it can be incorporated into the building model. Integration of design is dependent of the participation and collaboration of all design team members. In general, the earlier issues regarding building performance are considered in the design, the bigger the advantages and achievements in the end. When more project time has passed, the ability of influencing costs is decreasing. Figure 2.2-2 (Eastman et al., 2011, p. 164) illustrates this principle.

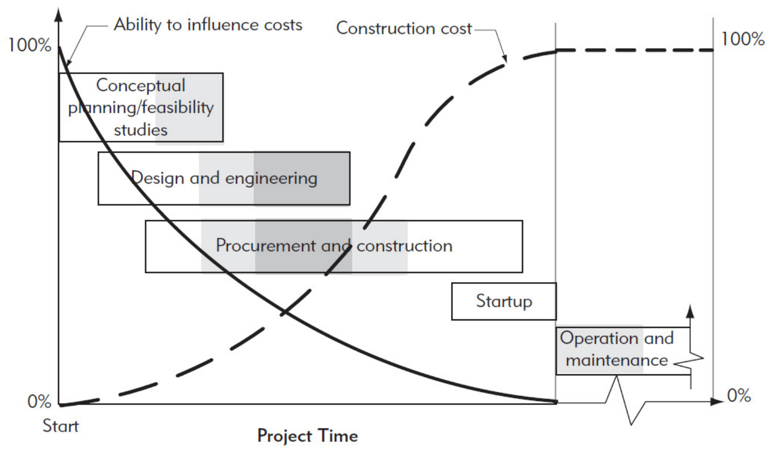


Figure 2.2-2: Illustration of cost influence on project time

The role of BIM in the process lies in the impact of early design decisions. A more optimized BIM process means early feedback about the design and less constructions costs because of early adapation in the project.

Besides, the building information model is always up-to-date because every building team member can access the model, and there is only one building information model. At the traditional building process, confusion about the latest versions of a model occurs, for this reason unncessary additional work is excecuted, which has a negative influence environmentally and financially.

Applying Building Information Modeling is more technically advanced than the traditional process, this makes it difficult for the non-professional client to understand. Particularly for elderly it is more challenging to accept this technology, despite the benefits (Heidari et al., 2014). A reason why it is more technically advanced is because of the exchanging of files forms a challenge, participants in the building process prefer using their own software. This leads to multiple file formats, which makes it difficult to exchange. Research, development and acceptance are crucial for the adaptation of BIM in the construction industry.

Applications of BIM
BIM technology helps to present the building design in three dimensional views and is also known for its use for communication and data exchange This virtual building plays a major role in the process of simulations, testing, refining and validation of building designs (Christiansson et al., 2011).

2.3 Data Interoperability

Since model and information exchanges are required between different project stakeholders and during different project phases, a key problem to industry users is data interoperability. (S. Zhang, Teizer, Lee, Eastman, & Venugopal, 2013). Interoperability is the ability to exchange data between applications, which improves workflows and sometimes facilitates their automation (Eastman et al., 2011). The benefit of Interoperability which is based on an open standard is that when using this, the file only needs to be translated twice; when sending and receiving. A visualization of this principle is shown in Figure 2.3-1, which is based on the figure of (Laakso & Kiviniemi, 2012 P. 137). When no common standard exists, each individual software application has to develop and implement their own translators in order to achieve exchanging of different file formats which costs a lot more effort. It can be concluded it is essential to strive for a higher level of standardization, only then involved parties can communicate in the same way and it will be much clearer how to cooperate using BIM.

Three initiatives which aim to develop data interoperability are presented. By either trying to enlarge standardization of a building model or challenging companies by labeling them. These three initiatives are: “Level of Development”, “BIM maturity level” and “Open standard”.

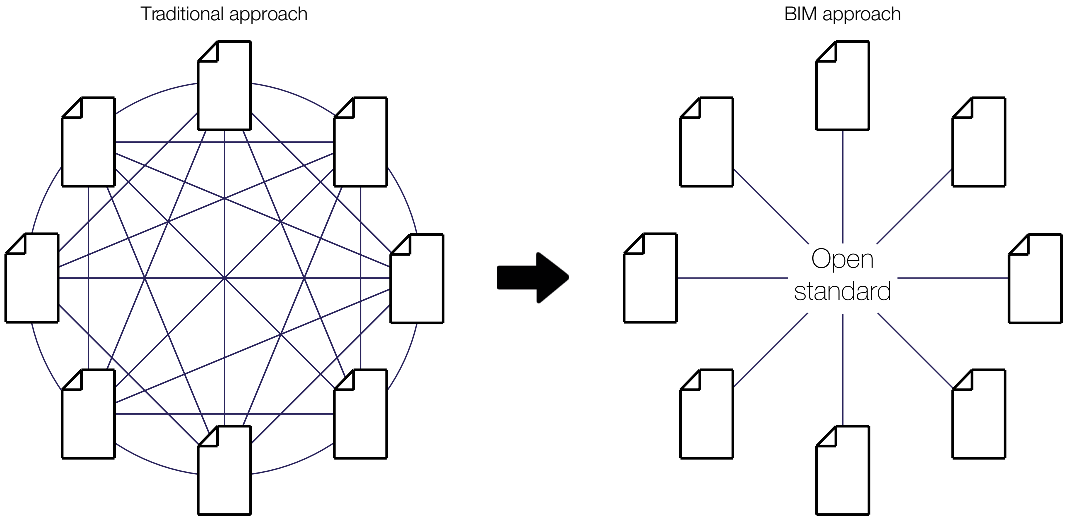


Figure 2.3-1: Visualization of principle of open interoperability

Level Of Development

The Level of Development concept is developed by the American Institute of Architects (AIA), aimed to define the individual development for each building element at various stages of time.

The Level of Development is a concept which is developed to define information models. The Level Of Development (LOD) specification is defined as a method which enables stakeholders in the Architecture, Engineering and Construction industry to specify the content and reliability of Building Information Models at different stages in the construction process (BIMForum, 2015).

The AIA developed the following five Levels (Bedrick, 2008):

LOD 100	Conceptual; the model contains a table
LOD 200	Approximate geometry; the table has dimensions
LOD 300	Precise geometry; the table has specific dimensions and finishing
LOD 400	Fabrication; the table is a model from a supplier
LOD 500	As-built; the table is verified as-built conducting information about supplier and accurate information about quantity, size, shape, location and orientation

A visualization of the differences between multiple levels of the Level of Development concept is shown in Figure 2.3-2, this figure is based on the information shown at (BIMstore, 2016).

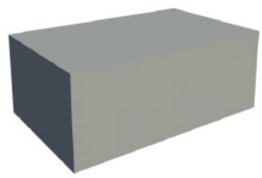
LODs are not defined by design phases, completion of the whole building or another deliverable. The reason for this approach is because there are no clearly defined design phases in the building industry. Besides, during the construction process various building elements are at a different Level of Development. For instance, at the schematic design phase the building information model can for instance consists of elements with a LOD 100 for eighty per cent, elements with a LOD 200 for ten per cent and elements with a LOD 300 for another ten per cent. Because of this it is not possible to define one model as a LOD 200 because of the difference in progress among the various elements.

It can be stated that a way to deal with BIM is the Level of Development. However, a disadvantage of this concept, and the main reason why it is not fully adopted yet in practice, is because the definition of what defines an object for instance LOD200 or LOD 300 is vague. Unless it is specificied, a well known example is the (NATSPEC, 2016). The concept of LOD offers a lot of potential as a way to set up clear agreements between stakeholders, though it is important that the LOD is correctly specified.

Level Of Development of a table

LOD 100

Description	**Dimensions**	**Manufacturer**
Table with four legs	L: W: H:	-
Model	**Purchase date**	**Finishing**
-	-	-



LOD 200

Description	**Dimensions**	**Manufacturer**
Table with four legs	L: 1200mm W: 800mm H: 500mm	IKEA
Model	**Purchase date**	**Finishing**
HEMNES	-	-

LOD 300

Description	**Dimensions**	**Manufacturer**
Table with four legs	L: 1200mm W: 800mm H: 500mm	IKEA
Model	**Purchase date**	**Finishing**
HEMNES	-	Dark brown

LOD 400

Description	**Dimensions**	**Manufacturer**
Table with four legs	L: 1200mm W: 800mm H: 500mm	IKEA
Model	**Purchase date**	**Finishing**
HEMNES	-	Dark brown, wood

LOD 500

Description	**Dimensions**	**Manufacturer**
Table with four legs	L: 1200mm W: 800mm H: 500mm	IKEA
Model	**Purchase date**	**Finishing**
HEMNES	03-03-2017	Dark brown, wood

Figure 2.3-2: A visualization of the LOD of a table

BIM Maturity level
The BIM Maturity level is a classification of the company's ability to operate and exchange information. The label can be very valuable when companies look for collaboration of a certain (BIM) level in the construction industry chain. The BIM maturity level concept provides a classification of BIM implementation. Figure 2.3.3 (Mark Bew and Mervyn Richards, 2008) shows a graph which visualizes an illustration of a roadmap for the whole sector. The BIM maturity level is part of the BIM Strategy paper of the Government of the United Kingdom (BIM Industry Working Group (BIWG), 2011) and was used because it provides an efficient visualization tool showing the different levels of BIM maturity in the Architecture, Engineering and Construction industry.

Figure 2.3-3 shows four levels of BIM maturity which are defined as follows (BIM Industry Working Group (BIWG), 2011 P. 40):

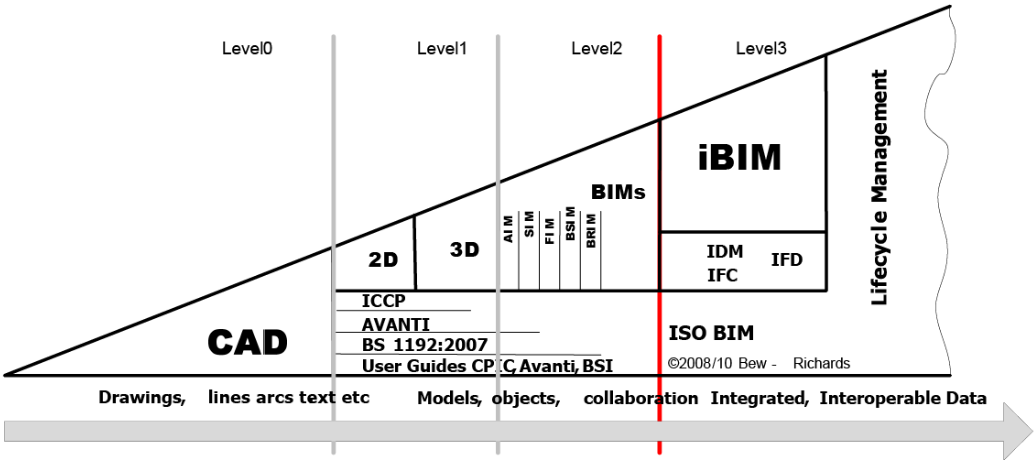


Figure 2.3-3: The BIM maturity levels

Level 0 represents unmanaged 2D CAD drawings. In this stage everyone works with texts, lines and curves on document level. This phase is labeled as a pre-BIM level because of the lack of digital objects.

Level 1 represents CAD in 2D or 3D format using standardized collaboration tools and providing common data environment with possible standard data structures and formats.

Level 2 refers to a 3D environment held in separate discipline BIM tools where data is attached to elements. This is the starting point of collaboration and interoperability because of open access between disciplines.

The red line stands for the required BIM maturity for tendering large public projects in the United Kingdom starting from 2016.

Level 3 represents fully open data processes and data integration, at this point information between parties can be exchanged.

The BIM maturity level has a similar problem as the Level Of Development. The BIM maturity level could be of great value, when agreed on the specific levels. In that case it would be a great classification system for companies which strive to use more BIM, and are willing to know which experience level the cooperating companies have. The BIM maturity level is not further elaborated upon in this thesis, since the application of the BIM maturity level is not aimed to improve the construction process directly.

Open standards
In order to be able to develop data interoperability, it is necessary to easily exchange data between the stakeholders. To achieve this, the open standards are adopted in the Architecture, Engineering and Construction industry.

Before continuing with the specification and explanation of the open standard, at first the general standard is defined. A standard can be defined as (de Vries, 2005):

> *''A standard is an approved specification of a limited set of solutions to actual or potential matching problems, prepared for the benefits of the party or parties involved, balancing their needs, and intended and expected to be used repeatedly or continuously, during a certain period, by a substantial number of the parties for whom they are meant.''*

This definition forms the base of the open standard. It addresses that a standard is made up to solve issues which occur more than one time, which is a requirement in the construction industry where collaboration is key for success.

Among the various data model standards developed, Industry Foundation Classes (IFC) is the only public, non-proprietary data model existing today which is formally adopted worldwide in the AEC sector by different governments and agencies (Gupta, Cemesova, Hopfe, Rezgui, & Sweet, 2014). As a general data model, IFC enables the exchange of data among various applications.

This standard is developed by (buildingSMART International Limited, 2013), to describe, exchange and share information in an open and neutral format. BuildingSMART is an international platform for knowledge exchange regarding BIM. The vision of BuildingSMART is: realization of the full societal, environmental and economic benefits of open sharable civil infrastructure and building asset information into commercial and institutional processes worldwide (BuildingSMART, 2016). The former International Alliance for Interoperability (AIA) aims to improve the exchange of information between software applications used in the construction industry.

The use of an open standard offers a lot of potential, mainly because it has a direct influence on the construction process. The open standard would be interesting for the progress of the construction process for further development.

2.4 Industry Foundation Classes

The Industry Foundation Classes is widely recognized as the common data exchange format for interoperability within the AEC industry (Eastman et al., 2011). IFC as an open standard offers a lot of potential for the improvement of the construction process using BIM. This sub-chapter provides an overview about the application of IFC, the difficulties when using IFC and zooms in on the structure of an IFC.

Application

The use of IFC as a standard format in exchange processes has been increasing as the industry begins to address the need of interoperability (Solihin, Eastman, & Lee, 2015), as was stated IFC represents geometry, relations, processes and material, performance, fabrication, and other properties, needed for design and production, using the EXPRESS language (Liebich & Wix, 1999). IFC thus aims at facilitating easy communication of construction-related information back and forth between BIM environments and other IFC-compatible software environments. Main goal is a more integrated design and construction process and thus an improved construction process with more quality and efficiency (Gallaher, O'Conor, Dettbarn, & Gilday, 2004).

Issues that come across when using IFC

Product model schemas such as IFC are rich but highly redundant, offering (too) many ways to define objects, relations, and attributes. Some experiments showed that various applications have different approaches how buildings should be modelled and mapped to the IFC schema and what information is required, which in real practices cause additional remodeling and extra effort (C Zhang, Beetz, & Weise, 2015). For many executions, it is required that the data needed for these processes is contained in models with proper representations by types, properties and names (C Eastman, Lee, Jeong, & Lee, 2009). (Oh, Lee, Hong, & Jeong, 2015) discovered that up to 78.8 percent of all objects can be lost in the process of exchanging information between IFC and Autodesk Revit formats. Not only objects, object properties (for instance; color, grid, layers, location and view) could get lost as well.

Even though information could get lost and there are uncertainties about how to model, the potential of adapting IFC is of huge importance. However, to improve the construction process with the use of IFC, it is important to be able to control the conversion of IFC so information does not get lost. The development of IFC as an open standard, improves interoperability when more standardization takes place.

Structure of IFC

BIM data is intended to be readable and editable between various systems. To accomplish this, the file structure needs to be standardized. To be able to work with an IFC file it is important to know how an IFC is built up. For that reason in this part is elaborated upon the multiple layers an IFC consists of. It is necessary to zoom in on the structure of IFC, since it is important to control the conversion of IFC so information does not get lost during the construction process.

An IFC model consists of multiple layers as shown in Figure 2.4-1. To understand how an IFC file is built up, the Industry Foundation Classes Release 4 by building SMART International Limited describes these layers (buildingSMART International Limited, 2013.). IFC consists of a set of schemas, each schema belongs to a layer. Each schema represents particular content, for instance; costs, control and structural elements. Any layer can reference to the same layer or to a layer below, but may not refer to a class from a higher layer. References within the same layer are only allowed in the Core and Resource layer.

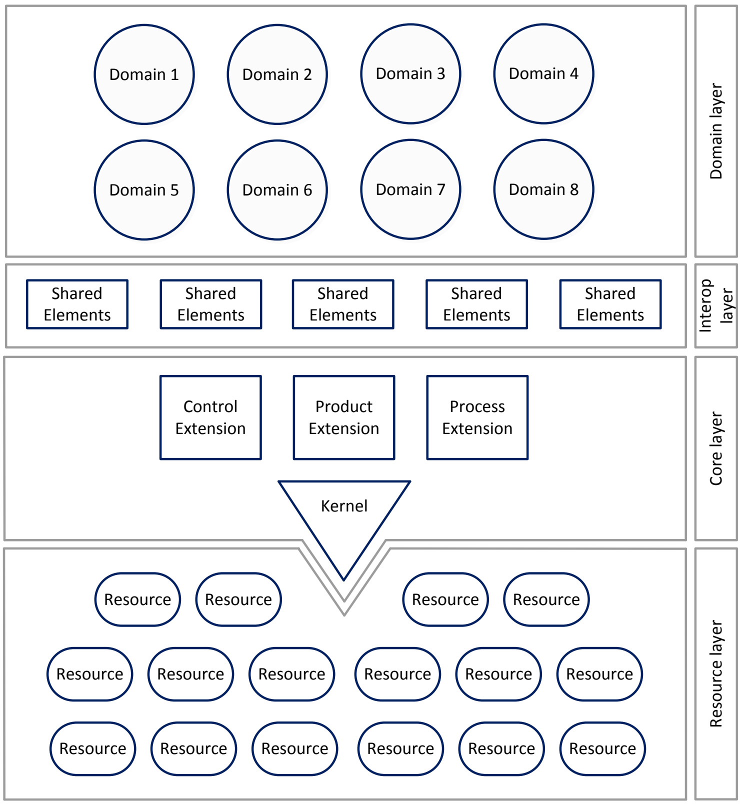


Figure 2.4-1: Layers of an IFC model

The data schema architecture of IFC defines four conceptual layers, each individual schema is assigned to exactly one conceptual layer.

1. Resource layer
Resource classes are used by classes in upper levels. An example of a resource class is the cost schema, all information concerning the concept cost is collected within this cost schema (IfcCostResource). Entities and types defined in this layer can be referenced by all entities in the layers below.

2. Core layer
The core layer consists of the kernel schema and the core extension schemas, containing the most general entity definitions. The Core layer provides the basic structure of the IFC object model and defines most abstract concepts which are specialized by higher layers of the IFC object model. The Kernel provides all the basic concepts required for IFC models it also determines the model structure and decomposition. Core Extensions provide extension or specialization of concepts defined in the Kernel. Each Core Extension is a specialization of classes defined in the Kernel. All entities defined in the core layer contain a unique identification, name, description, and change control information.

3. Interoperability layer
The shared element data schemas contain entity definitions which are specific to a general product, process or resource specialization used across several disciplines. Those definitions are typically utilized for inter-domain exchange and sharing of construction information.

4. Domain layer
The upper layer includes schemas containing entity definitions which are final specializations of products, processes or resources specific to a certain discipline. Entities defined in this layer are self-contained and cannot be referenced by any other layer.

Conclusion
In order to be able to exchange information amongst all the involved parties in the construction industry, it is important to easily share files between the parties. The Industry Foundation Classes (IFC) is widely recognized as the common data exchange format for interoperability within the AEC industry (Eastman et al., 2011). An IFC file is intended to be readable and editable between various systems. Though information gets lost in the process of exchanging information between IFC and Autodesk Revit formats. It is important to control the conversion of IFC to minimize the loss of information. The development of IFC as an open standard, improves interoperability when more standardization takes place. An IFC consists of multiple layers; resource layer, core layer, interoperability layer and domain layer.

2.5 Standardization of IFC

In the previous sub-chapter the problem that various applications map IFC in different ways is addressed. As a result inconsistencies exist, though it is essential to prevent this. In response to deal with the problems of inconsistencies in the assumptions and the big amount of data, some guidance needs to be provided to validate the models. The standard methodology of the Information Delivery Manual (IDM) and Model View Definitions (MVD) has been introduced to define required information for particular scenarios (See, Karlshoej, & Davis, 2012). These methodologies, the International Framework for Dictionaries and a short introduction to BIM standards, are explained further in this sub-chapter.

Information Delivery Manual
With IFC it is possible to exchange data amongst different players. However, a disadvantage concerning IFC is because it covers such a big amount of data, it can be unclear which data should be presented to which stakeholder. As a response on the big amount of data which comes along in the construction industry, the Information Delivery Manual (IDM) was introduced in 2007 as an official element of IFC standardization. The Information Delivery Manual methodology is used for existing documents or new processes and describes the associated information which have to be exchanged between parties. According to (BuildingSMART, 2010) it specifies:

- *Where a process fits and why it is relevant*
- *Who are the actors creating, consuming and benefitting from the information*
- *What is the information created and consumed*
- *How the information should be supported by software solutions*

These specifications are covered in an IDM, the major parts which cover these specifications are: process maps, exchange requirements, functional parts, and concepts.

> The process maps provide an understanding of who the involved actors are, how the activities are configured, and what information is required, consumed and produced at different stages of the process.
>
> An exchange requirement describes the information that must be exchanged in order to support a particular requirement at a particular stage of a project.
>
> Functional parts are individual units of information which software vendors use to support exchange requirements, describing the information by taking into account the requirements of the IFC data model.
>
> Concepts are connected directly to the IFC model and are implemented in functional parts. Concepts are capable of, but are not limited to, capturing the basic functionalities of a model and can be flexibly assigned to individual or whole entities (Wix 2007).

Model View Definition
Together with the IDM comes along the Model View Definition (MVD). A Model View Definition, or IFC View Definition, defines a subset of the IFC schema, which is required to satisfy one or many Exchange Requirements of the AEC industry (See et al., 2012). The stated goal of a MVD is: finding a useful balance between the wishes of users/customers and the possibilities of software developers, and documenting the outcome clearly (Lehtinen, 2006). A MVD can be applied to validate if the provided data conforms the Exchange Requirements. Development of model views that represent the appropriate subset of data required for an exchange is critical in defining the exact rule checking requirements that the populated models satisfy (W. Solihin & Eastman, 2015).

The MVD works as a filter for the IFC data schema. In a building information model multiple views are available. For instance, a view of the HVAC, construction technology or an architectural view. Since not every team member is interested in all the views, a MVD is applied to automatically validate if the provided data conforms to the exchange requirements per user, which are described in an Information Delivery Manual.

International Framework for Dictionaries
The International Framework for Dictionaries (IFD) is developing mappings of terms between different languages, for wide use in building models and interfaces. It describes what is exchanged by allowing the creation of dictionaries to connect information from databases to IFD models (Eastman et al., 2011).

Relation amongst IFC, IFD, IDM and MVD
Shortly, IFC is the format which represents the data, IFD about what is shared, IDM is about the content and timing in the process and MVD is a filter which determines which content is showed per user. The aim of these concepts is to provide guidance for the use of an IFC file.

BIM standards
Another way to guide IFC into standardization is the existence of a number of national, company- and project specific BIM standards and agreements which have been developed. Examples are the Dutch general service administration (Het Rijksvastgoedbedrijf, 2013) BIM Norm in The Netherlands (van Rillaer, Burger, Ploegmakers, & Mitossi, 2012), the Australian National Building Specification System (NATSPEC, 2016) and Statsbygg BIM Manual in Norway (Statsbygg, 2013). These developed exchange requirements or BIM standards have defined description-based rules that IFC building models in specific context should conform to (Chi Zhang et al., 2015). The BIM standards would be a good solution for improving the standardization in the BIM industry. However, multiple BIM standards are being used, which counteracts the original purpose of standardization. Research to BIM standards is required to conclude what would be the best composition of standards, so the ultimate standard could be set up and put into practice.

2.6 Model checking in general

In a collaboration environment like the building industry, being able to obtain information with sufficient quality is fundamental for the development of construction processes (C Zhang & Weise, 2012). If the quality level of the information is not sufficient enough, it leads to flaws in the building design and a disapproval of BIM in the construction process. However, one of the promising aspects that BIM can offer in the AEC industry is facilitating various rule checking and simulations for evaluating building designs in the earlier phases of a project (Eastman et al., 2011).

Rule checking applied in the construction industry is defined as a piece of software that does not modify a building design, but rather evaluates it on the basis of configured building objects (S. Zhang et al., 2013). Being able to check these requirements in an automated way is highly desirable for effective data exchange and high quality end results (Krijnen & van Berlo, 2016). It is of great importance that these requirements are checked automatically, otherwise the checking is too costly.

Forty percent of defects in the AEC industry can be related to flaws in the design process (Hjelseth & Nisbet, 2010). Model checking offers a lot of potential and is considered to be one of the biggest benefits of BIM. By checking a model before the construction of the building is executed, design related faults can be found and solved. Until recently, the only means to deal with the big amount of data which comes along with a construction process, were human cognition and organizational review processes (C Eastman et al., 2009). Automation of checking, where well-defined rules can be applied automatically with minimum user intervention, are increasingly needed (Nawari, 2012).

The biggest challenge with implementing rule checking is to integrate the knowledge transfer during the interpretation, implementation process and the actual usage of the rule. This should be done with minimum information loss (Wawan Solihin & Eastman, 2015). IFC model testing used to be done mainly through manual and visual inspection. A conflict that occurs is because the rules which check the presence of information are typically written in human-oriented languages which require significant knowledge in order to “interpret” them into a machine interpretable manner (W. Solihin & Eastman, 2015). A study by Fiatech confirmed that when the human interpretation is involved, inconsistencies are expected. Different individuals interpret the rules differently, often led by their experience and locality (Fiatech Regulatory Streamlining Committee, 2012). Since the reasoning and interpretation ability of the human brain is unlike anything implemented in computer systems, the computerization of this process poses a real challenge to the AEC industry (Nawari, 2012).

Experience in CORENET ePlanCheck, shows that interpretation is crucial in transforming the rules. Singapore's CORENET ePlanCheck project is so far the most serious attempt to automate code checking at the national level. Based on these experiences, it is made clear that the role of automated rule checking serves as a decision support system, where some user involvement may be necessary. However, the ultimate goal of the automated rule checking should be a fully automated system that enlightens experts to focus on what really matters for buildings, such as safety, sustainability and high environmental performance (W. Solihin & Eastman, 2015).

Types of model checking
An often used example of model checking is clash detection to validate if, for example, different types of pipes intersect each other. However, there is a common perception that model checking is about validation and yes/no answers (Hjelseth & Nisbet, 2010).

According to (Hjelseth & Nisbet, 2010) model checking is a way to share and utilize knowledge. Important to keep in mind is the fact that model checking does not guarantee a good design of a building, but can be used to avoid bad design solutions. Model checking is divided into four classes. Validating, guiding, adaptive design and content based checking.

The purposes of validation of model checking is to determine if the content in the model is accordance to a code, standard, regulative etcetera. Validating the model checking results in a Yes/No answer. There are two basic types of validation model checks. First, there is the geometry based checking and second, the compliancy checking. An example of geometry based checking is clash detection which might be the most common concept of geometry checking. The purpose of compliancy checking is to check if the design is according to the building codes and requirements which apply to the project.

(Wawan Solihin et al., 2015) defined that the technique of rule checking can be divided into seven categories, the seven categories are:

1. Syntactic aspects: checking for well-formedness of a building model according to set of standards, prior set conditions or other MVDs.
2. Building regulatory code: checks if the model is complied with the building code.
3. Specific client requirements: applied when exceptional requirements takes place.
4. Constructability and other contractor requirements: checks the construction design.
5. Safety during construction and maintenance: check supports decisions and searches for potential dangers.
6. Warrantee approval: the post-construction model is checked for issues that may affect the warrantee or cost to maintain.
7. BIM data completeness: checks the data completeness at the end for handover to the facilities management.

The categorizing of rules in this thesis can be useful when processing the rules in this research. When the exchange requirements of the supplier are known, the rules can be divided into one of these categories.

The structure of the rules, which are going to be checked on exchange requirements, have different levels of complexity. (W. Solihin & Eastman, 2015) have defined a classification for rules, according to the complexity of their structure. The structure of each rule is defined by the relations amongst the objects, attributes and entities. The following four classes have been defined, each class includes an example of a rule:

Class 1; rules that require a single or small number of explicit data
The rules in this class check the explicit attribute and information data that exist inside the building model. The data which needs to be checked is directly available in the model. An example of a rule within this class is:

Fire walls must have correct wall, door and window types.

Class 2; rules that require simple derived attribute values
Checks in this class are based on a single value or a small set of derived values, though it does not generate new data structures. Implicit relationships are often required to fulfill the checking requirements. Certain calculations may be involved in rules of this class. An example of a rule within this class is:

Space between two doors in a series shall be 48 inches (1219 mm) minimum plus the width of a door swinging into the space. Doors in a series shall swing either in the same direction or away from the space between the doors.

Class 3; rules that require extended data structure
The rules in this class require extensions to the data structure. Often the required extensions relies on geometrical, topological and other algorithms, data must be generated to execute the check. An example of a rule within this class is:

The Distance between detectors for flat ceiling shall not exceed:

a) 7m for areas other than corridors and
b) 10m for corridors

Class 4; rules that require a ''proof of solution''
Normal rules evaluate if there is compliancy in the model according to the requirements. This class of rules is a collection of rules that do not strictly ask for compliance or non-compliance, but requires a proof of solution. This requires a description how a model passes the rules, instead of just complying with the prescribed rules. An example of a rule within this class is:

Find a possible path to move large equipment into a building under construction.

Rule checking process
As stated before, the validation for IFC building models is becoming increasingly important in BIM-based collaboration processes. In order to get to an automatic rule checking process, (C Eastman et al., 2009 P. 1016) divided the rule checking process into four steps, as seen in Figure 2.6-1 ; rule interpretation and logical structuring of rules for their application, building model preparation where the necessary information required for checking is prepared, rule execution which carries out the checking and reporting checking results.

1. Rule interpretation
The first step in the rule checking process is about the interpretation of rules and is known as the most crucial step in the total rule checking process. The biggest challenge which comes along with the rule checking process is due to the fact that rules for building designs are written in a

natural language. The conflict occurs how these rules in human language can be interpreted by machine process without loss of data. This translation relies mainly on the type of rule and the intention of the rule check. Two ways of implementation can be defined, computer language encoded rules and parametric tables. The first one relies on the interpretation of the programmer translating the rule into a computer readable code. The other way of implementation is translating a rule in natural language in a logic way into computer coded language. The output of this phase is a certain amount of computer processable code, of which the level of flexibility, modularity and functionality largely depends on the expressiveness of the rule language chosen in the rule interpretation phase. This rule interpretation phase is thus of crucial importance for the remainder of the rule checking process.

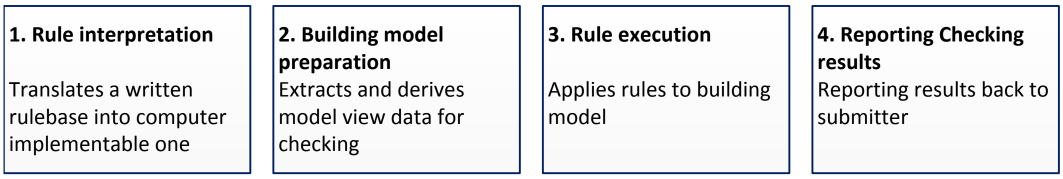


Figure 2.6-1: Rule checking process

2. Building model preparation

The second phase of the rule checking process is about the preparation of the building model itself. At the traditional process the objective is to obtain 2D drawings representations. The biggest requirement concerning drawings is that it must look visually correct. Since Building Information Modeling gained increased importance, a shift to a model oriented process have been taken place. Objects of a model nowadays are only considered to be that specific object when this is stated in the objects type and properties. The correctness of the model is crucial since building models are becoming large and complex data sets with the use of Building Information Modeling. The preferred solution is to automatically derive required rule checking data wherever possible, either within the model or the rule checking program.

Building models involve large datasets, even for the smaller scaled buildings. It would be useless to evaluate a whole model, as a rule only applies to certain aspects of a building. The use of the whole dataset of a building model makes the checking process ineffective. (C Eastman et al., 2009) proposed that separate model views should be used to derive needed data required for a specific type of rule checking and to extract subsets of an overall building model of rule checking functions.

3. Rule execution

In this phase the rules are applied to the model to check. At first the whole model needs to be checked for instance on the properties, names and required objects. In order to validate if the data needed for checking is available from the model and to make sure the data is complete and correct, the syntactic built up of checkers must be evaluated as well to ensure the checking works as required. Also the model views needs to be validated, when views are separately submitted inconsistencies can occur. The management of model versions consistency needs to be checked, when the design has change it is necessary the latest version of the model is checked.

4. Rule check reporting

In the last phase the results are reported. A rule has a fail, pass or error as result. Important for the evaluation of the rule checking is to visual publish all outcomes. Addressing the fails and errors as a result, but also to document the rules which passed. The outcome should be linked to the original rule, so that it is made easy to trace it to the original requirement.

Conclusion
One of the biggest advantages of BIM is the possibility to apply rule checking. With model checking the information exchange between several parties can be controlled. Different types of rules can be applied on the model in order to prevent failures on forehand. The biggest challenge with implementing rule checking is integrating knowledge transfer during the interpretation, implementation process and the actual usage of the rule. This should be done with minimum loss of information.

Multiple types of model checking and rules exist. The rule checking process can be divided into four steps; rule interpretation, building model preparation, rule execution and reporting checking results.

2.7 Validation tools

Projects in the building industry are dependent on the right information at the right time in order to succeed. As was concluded in chapter 2.6, a validation tool is required to conduct model checking. After the categorizing, classification and rule checking process of model checking, it is important to get to know more about the validation of IFC. Three types of validation tools are described in this sub-chapter.

Commercial checkers
An example of a validation tool is the commercial model checker. Three examples of the most used commercial model checkers are the Solibri Model Checker, Autodesk Navisworks and Tekla BIMsight. In this part only the Solibri Model Checker is discussed in detail, it represents the other commercial model checkers.

The Solibri Model Checker is a Java based model checker. It can read IFC files and check the models with preset rule libraries. The user can decide which rule needs to be checked from the rule library on the model. Since the Solibri Model Checker is a commercial tool, it is often too expensive to use for smaller companies. Besides, users cannot adjust the rule library by themselves, so users become dependent on the software vendor. It can be stated that a commercial model checker is not always the best choice and that there is a need for an alternative open sourced model checker.

MvdXML checker
Non-proprietary model checkers, such as the mvdXML Checker, solves these thresholds of proprietary model checkers. MvdXML is currently the only open standard dedicated for model view definition and IFC validation (Chi Zhang et al., 2015). The MVD Checker is a tool for IFC validation and is developed based on the open mvdXML standard so it is easily accessed, used and extended by end-users. A mvdXML file is an XML instance which is compatible in the mvdXML schema, so it can be developed by the official IfcDoc tool as well as common XML editors (Chipman, Liebich, & Weise, 2016).

IFCDoc stands for IFC Documentation Generator, which is a tool for creating mvdXML rules. It can check multiple IFC-files on multiple rules which are defined in XML. The tool produces one BIM Collaboration Format (BCF) report for each error. BCF introduces a workflow communication capability connected to IFC models. The idea is to separate the "communication" from the actual model (Stangeland, 2011). The mvdXML checker is based on the open source bimserver.org framework (Beetz & Berlo, 2010). Basically, the IFC instances and mvdXML files are the input of the checker while sets of BCF files are the output, as shown in Figure 2.7-1 (Chi Zhang et al., 2015 P. 29).

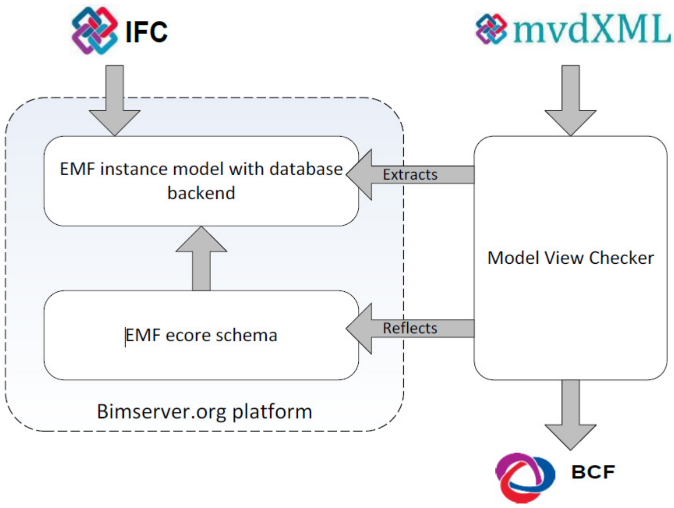
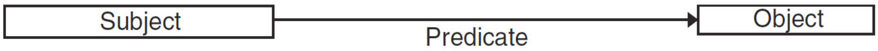


Figure 2.7-1: Concept of mvdXML

The mvdXML checker consists of three parts; the first part is about the interpretation of requirements and structuring validation rule-sets, the second part is about the execution of the checks and the last part is about the report generation.

The aim of the first part is to transform exchange requirements into rulesets. The implementation of model view rule-sets is based on the 1.1 version of the mvdXML standard (Chipman et al., 2016). When the mvdXML ruleset is completed, the mvdXML checker is able to check the rulesets on the IFC model. After the check has been executed, the mvdXML checker identified issues which are reported in a BIM Collaboration Format.

Figure 2.7-2: The ‘triple’ form of a RDF statement



Semantic Web

Another method for rule checking is based on using semantic web. Semantic web is a method where the extensive amount of data of the model is stored in various data sets which is made available as a web of data on the world wide web (W3C, 2011). The semantic web standard enables the possibility to create a web of linked data (Curry et al., 2013). The semantic web uses the Resource Description Framework (RDF) as a language to represent its graph structure. A RDF graph arises by applying a logical and operator to a range of logical statements. These statements are often referred to as ‘RDF triples’, consisting of a subject, a predicate and an object as shown in Figure 2.7-2. In addition, each concept and relation has a Unique Resource Identifier (URI) assigned to it, thereby making the RDF graph explicitly labelled (Pauwels et al., 2011 P. 509). Use of semantic web enables using more than just the IFC data, since the use of semantic web makes it possible to create linked data sets and the possibility to query IFC data.

Conclusion

Three types of validation tools are described, commercial checkers, validation tools based on mvdXML and validation tools based on semantic web. These three types of checkers distinguish from each other in multiple ways. Preset requirements determines, later on in this thesis, which type of checker is most suited for the case and which type is going to be developed further. However, in order to adjust and apply a model checker, at first the requirements needs to be established. In the next chapter a choice is made between the checkers presented in this sub-chapter.

2.8 Conclusion

The sub-research question related to this chapter is:

Which key concepts regarding model checking can be distinguished?

In this chapter five main key concepts which are relevant for continuation of this graduation thesis are distinguished; BIM, Data interoperability, Industry Foundation Classes, Model checking in general and Validation tools. A short summary of these concepts is provided:

BIM

The building industry requires repeated data exchanges and communication among different formats and stakeholders. New criteria are regularly developed, ranging from building codes, safety rules and techniques of fabrication. As the complexity level of the design and construction processes increases, traditional information resources cannot satisfy the required needs nowadays. In order to increase productivity and shifting from the current paper-based building process to a digital model-based building process, the concept Building Information Modeling (BIM) is conceived.

The Building Information Modeling can be defined in three ways:

1. As a product
2. As an IT-enabled, open standards–based deliverable, and a collaborative process
3. As a facility lifecycle management requirement.

BIM technology helps to present the building design in three dimensional views and is also known for its use for communication, data exchange and as a virtual building.

Data interoperability
Since model and information exchanges are inevitable between different project stakeholders and during different project phases, a major issue is data interoperability. Interoperability is the ability to exchange data between applications. It is based on an open standard, which means the file only needs to be translated twice; when sending and receiving. In order to improve interoperability between the team members it is essential to strive to more standardization, in this way involved parties communicate in the same way and it is much clearer how to cooperate using BIM.

There are several concepts developed to aim for standardized models during the construction process. The concepts of Level of Development and BIM maturity level give an impression of the process of the development of BIM. The Level of Development concept is developed to define and illustrate characteristics of model elements of different building systems at different Levels of Development. The BIM maturity level concept provides a framework for the classification of BIM implementation in three levels.

Industry Foundation Classes
An open standard offers a solution in exchanging files. Among the various data model standards, Industry Foundation Classes (IFC) is the only public, non-proprietary data model existing today formally adopted worldwide by different governments and agencies. BuildingSMART developed this data model standard to describe, exchange and share information in an open and neutral format.

BIM data is intended to be readable and editable between various systems. BuildingSMART International Limited released Industry Foundation Classes Releases 4 (IFC4) in 2013. In this release is established that an IFC file, amongst others, consists of objects, properties, entities and attributes which are all related to each other, and is built up out of four conceptual layers; the resource layer, core layer, interoperability layer and domain layer.

To deal with the problems of inconsistencies in the assumptions the standard methodology of the Information Delivery Manual (IDM) and Model View Definitions (MVD) has been introduced. The IDM is about which data when to share and MVD about what data is presented to which team member. The aim of these concepts is to provide guidance for the use of an IFC file.

Model checking
One of the promising aspects that applying BIM offers in the AEC industry is facilitating various rule checking and simulations for evaluating building designs in the earlier phases of a project. Rule checking applied in the construction industry is defined as a piece of software that does not modify a building design, but rather evaluates it on the basis of configured building objects.
The biggest challenge considering implementing rule checking is integrating the knowledge transfer during the interpretation, implementation process and the actual usage of the rule. This

should be done with minimum loss of information. Since the rules are typically written in human-oriented languages that require significant domain knowledge in order to “interpret” them into a machine interpretable manner. Model checking is divided into four classes. Validating, guiding, adaptive design and content based checking. There are two basic types of validation model checks. First there is the geometry based checking and second the compliancy checking. This categorizing can help to process the rules in this thesis. When the exchange requirements of suppliers are known, the rules can be divided into one of these categories.

Validation tools

A validation tool based on an open and standardized method is desired. Commercial model checkers are not open sourced, which means using one can be costly and users become dependent on the software vendor, since a commercial tool cannot be adjusted by the users. Currently mvdXML is the only open standard dedicated for model view definition and IFC validation. The MVD Checker is a tool for IFC validation and is developed based on the open mvdXML standard. Another method for rule checking is based on using semantic web. Semantic web is a method where the big amount of data of the building is stored in various data sets which are made available as a web of data on the World Wide Web.

3
Data collection

Aim of this chapter is to collect the required data to answer the main research question. At first the context of this research is defined. The main research question is linked to a case study, defining this case study and its corresponding construction process is the first step. Next is zoomed in on the construction process of this case study. Finally the definition of 'completeness' is established which is a key-concept of the main research question.

The corresponding sub-research questions are:

2. How can the construction process be defined within the context of this research?

3. What are the differences between the current and ideal situation of the defined construction process?

4. Which parameters need to be present in a building model before it can be considered complete?

3.1 Definition of the construction process

3.1.1 Introduction

The research question aimed to answer in this part is:

2. How can the construction process be defined within the context of this research?

The term 'construction process' needs more specification since the term is too broad now to be applied in this thesis. For this reason, the first step is to specify the construction process and the corresponding case study of the research. Starting point is the construction process of the whole company where this research is conducted at. Next, the specific department is determined, after this it is going to be established which part of the process is going to be elaborated upon. A visualization of the specification of the construction process is shown in figure 3.1-1.

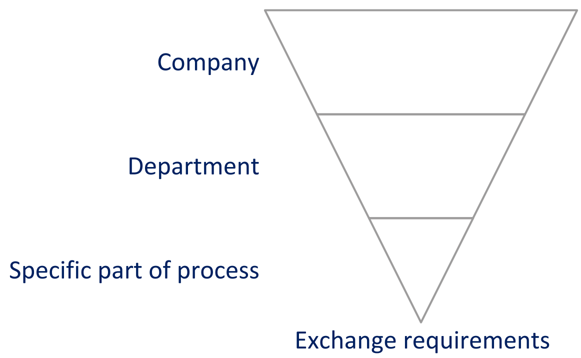


Figure 3.1-1: visualization of the specification of the construction process

3.1.2 Hurks

The starting point in the definition of the construction process and case study, is the company. The graduation thesis is conducted in collaboration with Hurks Bouw Eindhoven and executed at the department of plan development. Hurks is an umbrella enterprise consisting of multiple disciplines which cover the width of the entire construction sector. Hurks operates as a real estate developer, construction developer, supplier and business concern. Several activities are executed, consisting of building development and advice for construction and engineering. Hurks is specialized in urban (re)development, construction of hospitals and universities, high-rise buildings and transformation of buildings (Hurks, 2016).

3.1.3 Da Vinci Huis

The Da Vinci Huis concept is the subject of the case study. In 2013 Hurks launched this new housing concept, a new Design and Build concept which focusses on housing associations, investors, municipalities and developers to quickly and efficiently develop and execute dwellings. Aim of the Da Vinci Huis concept is to combine the advantages of a standardized house with the desire for freedom of choice in the design. The main goals of the concept are: Low costs, short development and construction time while still providing freedom of choice in the design. A configurator designed for the quick development of the concept, contributes in achieving this goal.

It is attempted to standardize the floor plans of the Da Vinci Huis projects as much as possible, while aiming to differentiate in the shell of the residential buildings in order to prevent uniform

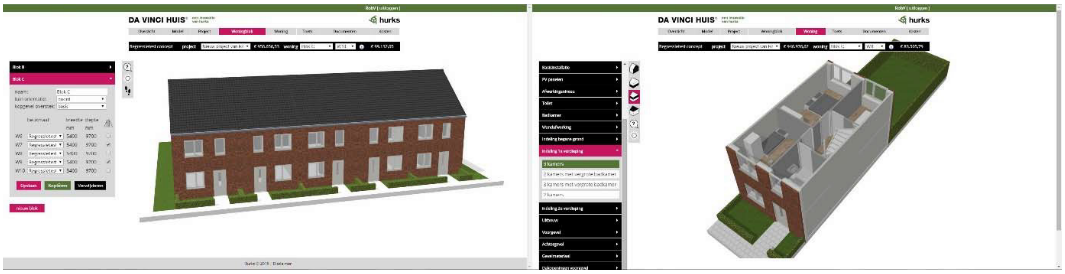


Figure 3.1.3-1: Two screenshots of the residential configurator, on the left side the creation of a housing block, on the right side the configuration of an individual dwelling

looking dwellings. Because standardization is at the expense of the freedom of choice of the customer, a configurator is conceived. Screenshots of the residential configurator are shown in Figure 3.1.3-1.

This configurator is an online tool which is able to compose the dwellings within half a day. In the configurator housing blocks can be created, it can be specified about how many lots a housing block should consist of and which dimensions every lot should have. On every lot one house can be placed. The configurator offers a choice between seven housing references of which a customer can choose from. Figure 3.1.3-2 shows these seven housing references. For each property multiple options are available. For instance it is possible to choose between different varieties of layout of the floors, the customer can add an extension to the property and the customer can choose between various window frames. In addition, a selection of choices is available regarding dormers, skylights, façade layout, roofing, gutters, HVAC, finishing of the interior etcetera (Hurks, 2016). In this way Hurks offers freedom of choice for the customer, while offering a low budget house due to the use of standardization.

Figure 3.1.3-2: The seven housing references

The concept of the Da Vinci Huis is suitable as a case study for multiple reasons. The building process of the Da Vinci Huis differs from other construction processes because it is a repetitive process. In the traditional way the process and the building itself are every time unique, unlike the Da Vinci Huis. Because the Da Vinci Huis concept is a repeated process it is more efficient than traditional projects. Since constructing a building in the traditional way is basically making a prototype every time, no chance is offered to filter out repetitive mistakes. Because the concept of Da Vinci Huis is an iterative process, possibilities occur to prevent making similar mistakes and improve this in the future. Besides, working with the same partners every time saves money due to more efficient contracting and better communication due to repeatedly collaboration.

Because this concept is applied in the past, floor maps and technical drawings are available. Since it is going to be applied in the future, improving the process of the concept is relevant.

3.1.4 Prefab concrete walls

This graduation thesis is limited to the prefab concrete walls in the Da Vinci House process. The concrete walls are considered suitable for research, since, in contrast to other elements, external walls are not modified during the process.

The supplier in this graduation thesis is Holcim Nederland. This is a supplier in the field of cement and raw materials. The company is the provider of prefab concrete wall elements for the Da Vinci Huis projects. The company is willing to exchange information in an IFC format.

3.1.5 Conclusion

The construction process of the graduation research is conducted at the company Hurks bouw Eindhoven, at the Da Vinci Huis concept and focuses on the exchange of information between customer and supplier of prefab concrete wall elements, as shown in Figure 3.1.5-1.

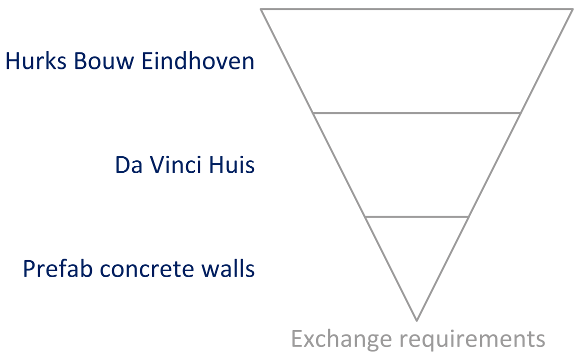


Figure 3.1.5-1: Definition of the construction process of the case

3.2 Visualization of Da Vinci Huis processes

3.2.1 Introduction

Since the main focus of the construction process is established in the previous sub-chapter, this part zooms in on the construction process of Da Vinci Huis. At first the main stakeholders are introduced. Thereafter, is zoomed in on the current and ideal situation of the exchange of data of prefab concrete wall elements using BPMN. For this reason, the following sub question concerning section 3.2 is drawn:

3. *What are the differences between the current and ideal situation of the defined construction process?*

In order to answer this sub-research question, both current and ideal situation are elaborated upon in this chapter. The differences between the current and ideal situation stands for the changes that need to be made to come to the best possible situation (the ideal situation). Later on in this thesis is attempted to change a part of the current situation into the ideal situation.

3.2.2 Stakeholders

In the Da Vinci Huis process the following main stakeholders can be distinguished:

Customer: multiple customer groups choose to apply the Da Vinci Huis concept. The most important customer groups are: social housing corporations, commercial investors, real estate developers and real estate investors.

Contractor: this party is responsible for the preparation and execution of the construction of the Da Vinci Huis dwellings.

Architect: could be the same as the draugtsman, the architect is responsible for the design of the shell of the houses.

Draughtsman: responsible and developer of the visual configurator, besides this party executes the architectural and technical drawings.

Municipality: is responsible for the granting of permits.

Supplier: in this thesis this group is represented by the supplier of prefab concrete walls.

3.2.3 Visualization of the current process

This analysis is not about the whole construction process of the Da Vinci Huis, but starts at the point where the customer approaches the contractor and ends with the start of the assembling of the construction. Not every part of the construction process is explained on a detailed level, otherwise too much irrelevant information is shown. The information in the BPMN schema is gained via Hurks, the contractor. In appendix 1 and 2 the complete BPMN schemas of the processes can be found.

The Business Process Model and Notation (BPMN) method is used as graphical representation for the process of the Da Vinci Huis. The Object Management Group developed this method and defined BPMN as a standard which provides businesses the capability of understanding their internal business procedures in a graphical notation and gives organizations the ability to communicate these procedures in a standard manner (Object Management Group, n.d.). This method is chosen because of the focus on the visualization of exchange requirements per stakeholder, which is an important aspect of this thesis.

Current Da Vinci Huis process

The first phase is the commercial part between customer and contractor and is shown in Figure 3.2.3-1, the customer wants to develop dwellings and approaches the contractor. The customer sends their wish list to the contractor, the contractor then decides if it is suited for the Da Vinci Huis concept or not. If the request is considered suitable the contractor meets the customer to present the concept of the Da Vinci Huis using the visual configurator and specify the customer's wishes. The outcome of the configurator are automatically generated drawings. These drawings are sent to an architect who designs the façade and roof, preferably based on a material/element library provided by Hurks. When the design is finished it is returned to the contractor, after the contractor approves it is up to the customer to approve or disapprove. If the contractor or customer does not agree with the design it is sent back to the architect to improve the design. This process continues this way until both parties are satisfied with the design.

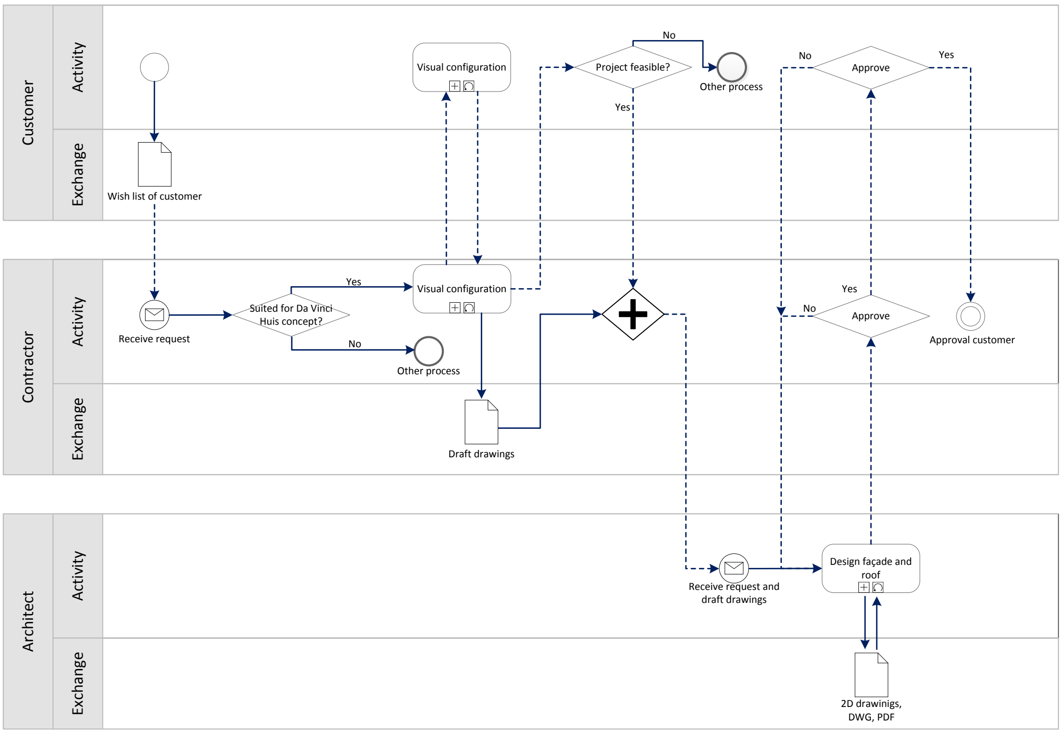


Figure 3.2.3-1: First part of current Da Vinci Huis process

Based on the visual configurator a global estimation of the prices is presented to the customer. Since the customer knows the design of the façade, roof and a conceptual visualization of the overall dwellings, the customer is aware of all necessary information about the design of the building and has to decide whether to continue with the project or not.

Figure 3.2.3-2 shows the process from the approval of the customer until the approval of production documents and architectural drawings. In the case the customer agreed to go on with the project, a request is sent to the draftsman company to improve the draft drawings of the visual configurator. The request for the draftsman is to convert the draft drawings into architectural drawings. The architectural drawings differ from the draft drawings because the specific wishes of the customers are not included in the draft drawings. Since the visual configurator has only a limited amount of options, the visual configurator cannot cover all the specific demands of the customer. Another task of the draftsman is to check the design against the building code, with a report as outcome. Both the architectural drawings and report of the building code are sent back and forth between draftsman and contractor until the contractor approves on the level of quality.

After this the contractor sends requests to the draftsman, municipality and supplier. The draftsman is requested to make technical drawings and improve them if the quality of the drawings is considered not good enough.

The municipality is requested to grant the construction permit. The granting of construction permits consists of two phases, the first phase takes about eight weeks, in this period of time the municipality decides whether to grant the construction permit or not. A permit for the Da Vinci Huis concept is always granted, even though it sometimes takes longer than eight weeks.

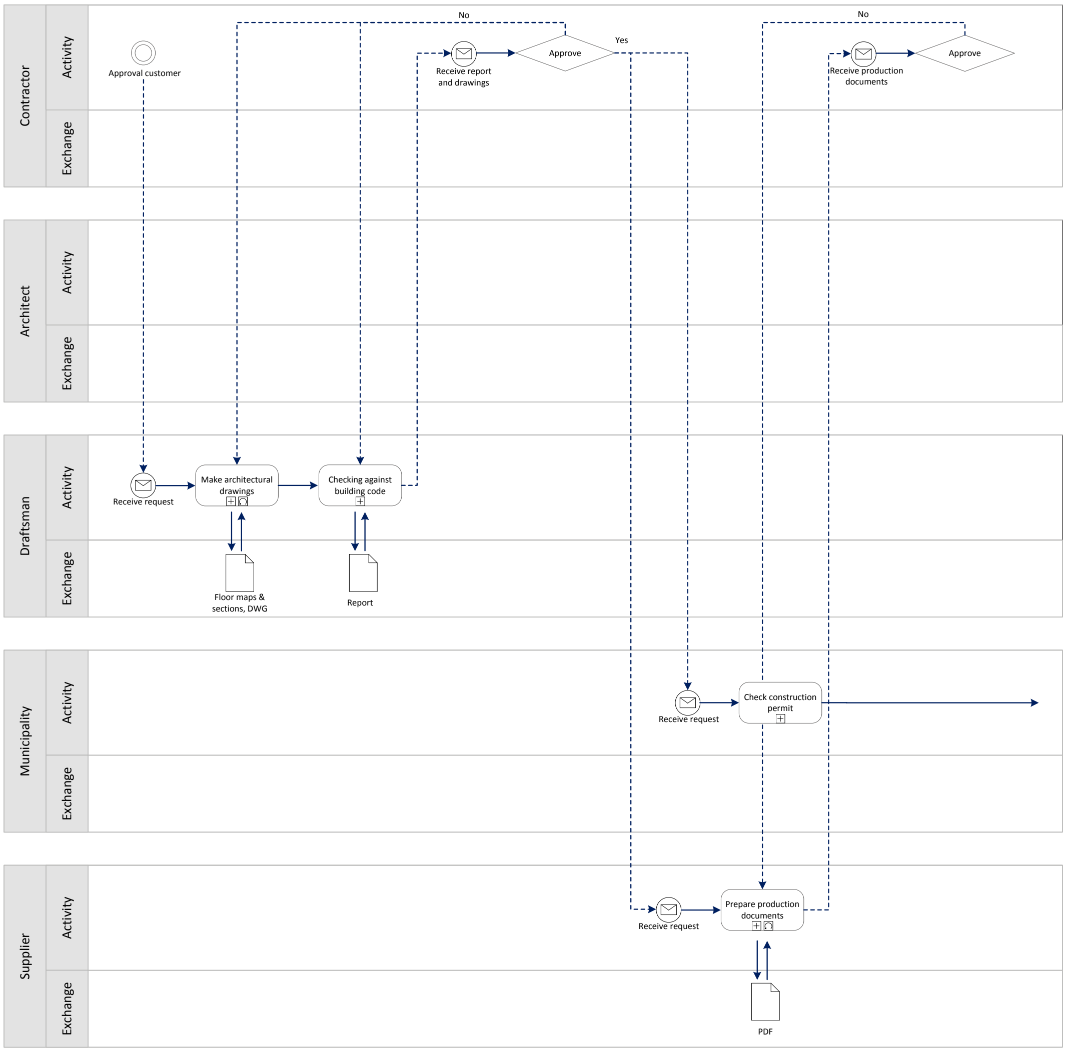


Figure 3.2.3-2: Second part of current Da Vinci Huis process

At the same time the supplier receives a request to prepare the production documents. In the production documents all the elements for production are individually described. For each individual element the necessary information is described which is required in order to be able to start production. After finishing the documents, the production documents are returned to the contractor who then decides if these are according to the design and technical drawings. If not, the files are sent back and forth until it is considered good enough. In the current situation this is a time-consuming job, since the controlling of the two dimensional documents is done manually. In the case a big construction project needs to be prepared, a huge amount of production documents comes along. Since it is seldom that production documents are approved the first time and because this is a task which requires to be done at each individual project, it is relevant to automate this part of the process.

The next phase of the process is shown in Figure 3.2.3-3. To move on to the next step, the approval of the production documents and the granting of the permit are needed. If this is the case, the contractor approves the production of elements and the supplier starts with the preparation of the building elements.

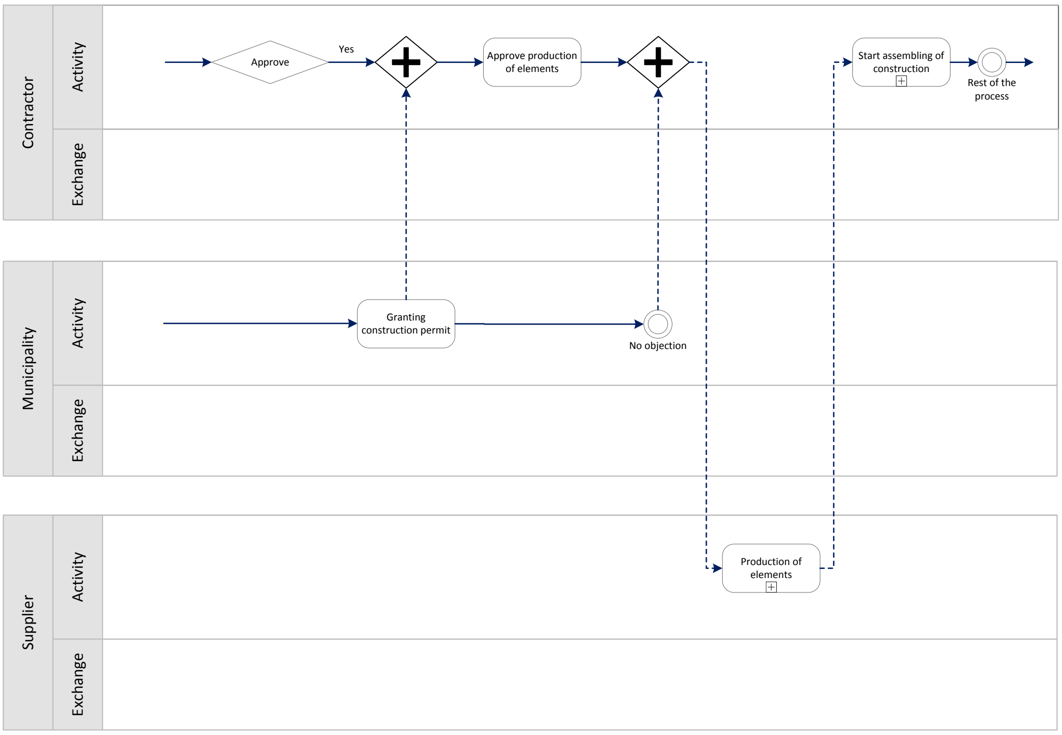


Figure 3.2.3-3: Third part of current Da Vinci Huis process

After this the second phase of the granting of the construction permit starts. In this phase neighbors, or other stakeholders, who suffer from an adverse effect caused by the development of the Da Vinci Huis, can lodge an objection during a period of six weeks. When no objection is lodged or the objection is not accepted, the construction permit does not form an obstruction anymore for the starting of the production of elements. After this the assembling of the construction starts, followed up by the rest of the process. This is not further described, since it is not important for this thesis.

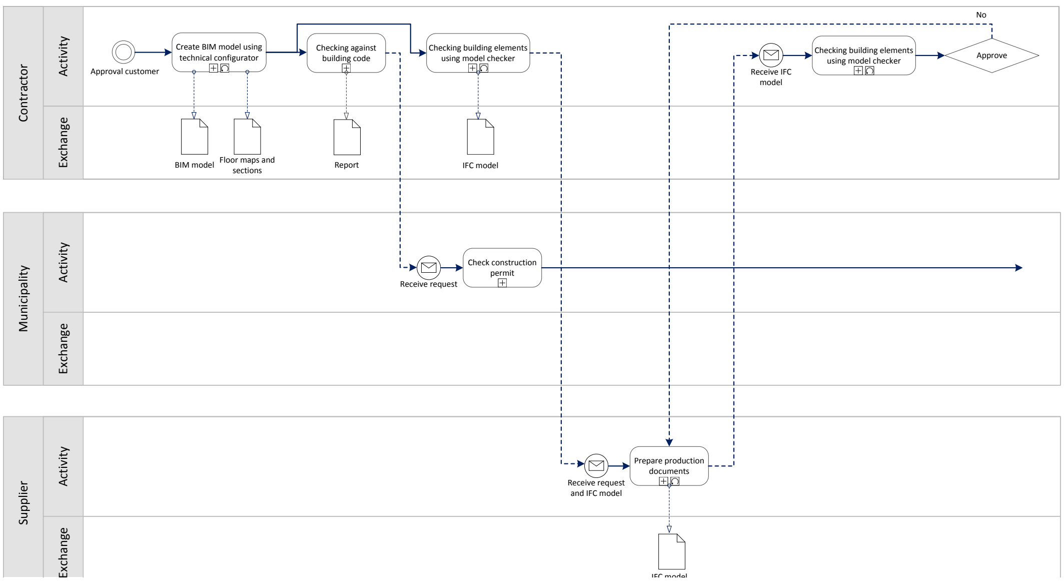


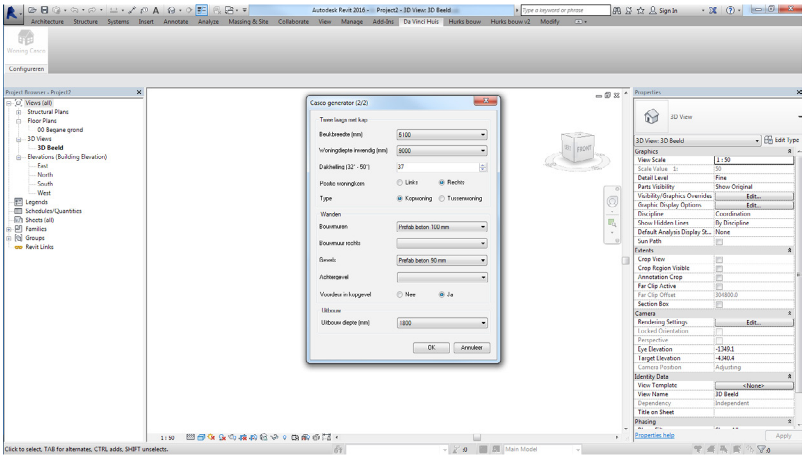
Figure 3.2.4-1: Part of the ideal Da Vinci Huis process

3.2.4 Visualization of the ideal Da Vinci Huis process

In this section the ideal situation is described, as well visualized by a BPMN schema. Since there is a lot of overlap between the current and ideal visualization, only a part of the BPMN schema is shown as seen in Figure 3.2.4-1, the complete visualization can be found in appendix 2.

The processes of the current and ideal situation are similar to the point of the approval of the customer, after that point the processes differ from each other.

After the approval of the customer the contractor continues the process by creating a BIM model using the technical configurator. This tool exists, though in development, and is designed by Hurks, the contractor. In this Autodesk Revit plug-in tool various options are available regarding the Da Vinci Huis, examples of the available options are; the length and width of the houses, the option for an extension, the angle of inclination for the roof and the use of materials, screenshots of this plug-in tool are shown in Figure 3.2.4-2.



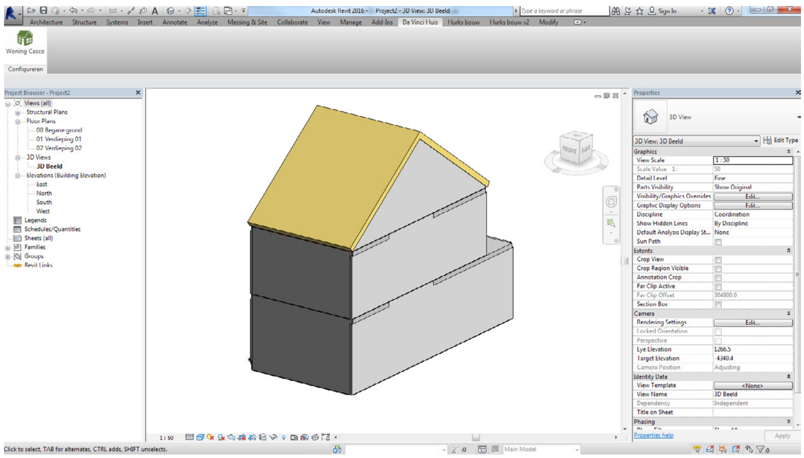


Figure 3.2.4-2: Screenshots of Revit plug-in tool

When the right options are selected, an Autodesk Revit model of the hull appears. In the ideal situation this tool is extended with more options, so the complete architectural and technical drawings can be created using this tool.

After the floor maps and sections are created they need to be checked against the building code. In the ideal situation this is done automatically. At the moment of writing, there does not exist a good working tool which aims to check the model against the building code. Since it is not the aim of this thesis to develop such a tool, this would be a recommendation for future development in order to achieve the ideal process.

Next, a request for the granting of the construction permit is sent, this follows the same procedure as described in the current situation. Another step taken after the creation of the BIM model is the automatically checking of the building elements using a model checker. This model checker checks if a model meets contains all necesarry information. It is of great importance that this check is executed, so the correct information is sent in the correct way to the supplier.
After the model has been checked and adapted, the 3D model is sent to the supplier. The supplier prepares the model for production and sends it back to the contractor, who checks the model again using the model checker. Since the checking is done automatically, a lot of time can be saved which is a big advantage. Another benefit is that manually checking is error-prone, when this can be avoided it saves a lot of unnecessary labor. When the model is approved, the process continues the same way as in the current process.

Exchange of information between contractor and supplier
As stated before, in the context of this thesis the focus lies on the exchange of information between the contractor and the supplier. Because this is an important part of the process for the continuation of the thesis, a detailed subscription of this part is described. This process is also visualized like the current and ideal situation processes in the previous chapters, with the use of BPMN. The information about the process and exchange of information which is shown in Figure 3.2.4-3 is conducted via an interview with the supplier of prefab concrete walls. This time only the current situation is visualized.

Current situation
The BPMN schema, Figure 3.2.4-3 starts with the approval of the contractor on the drawings and building code. The supplier receives a request and essential drawings in 2D, and focuses on the preparation of the wall elements. When the supplier receives the drawings of the walls, the drawings are manually checked by the supplier for flaws. Examples are too narrow lintels or walls longer than 11.40 meters. Next the supplier automatically adds elements to the drawings of the wall elements. These automatically generated elements are lifting hooks, strut sleeves and threaded sleeves for coupling plates. After this, the wall elements documentation is sent to the contractor to approve the production documents or not. The structural engineer is requested to calculate the amount of rebar needed in the wall elements. The supplier receives the calculations and adds the rebar into the drawings.

Last step is to add the information about the electricity pipelines and plumbing into the drawings of the wall elements. When it is sent back to the supplier, the final document is generated.

Ideal situation
In the current situation, everything is transferred in a two dimensional format. The supplier made clear to prefer the exchange of data two dimensionally. This is because two dimensional exchanges saves time for the supplier. When the supplier receives the information in three dimensional models, the supplier redraws everything manually into 2D DWG files because the structural engineer, electrician and plumber demand 2D files to add their information. The supplier

redraws everything manually, because the supplier assumes that parameters get lost when a 3D model is converted into 2D drawings.

The assumption of the supplier about parameters getting lost when 3D models are converted into 2D drawings is clear. However, a model checker is developed to check parameters and properties of elements and get control of this issue. In the ideal situation parameters do not vanish out of the model because of the conversion from 3D to 2D, so model exchange between contractor and supplier is done without any defects.

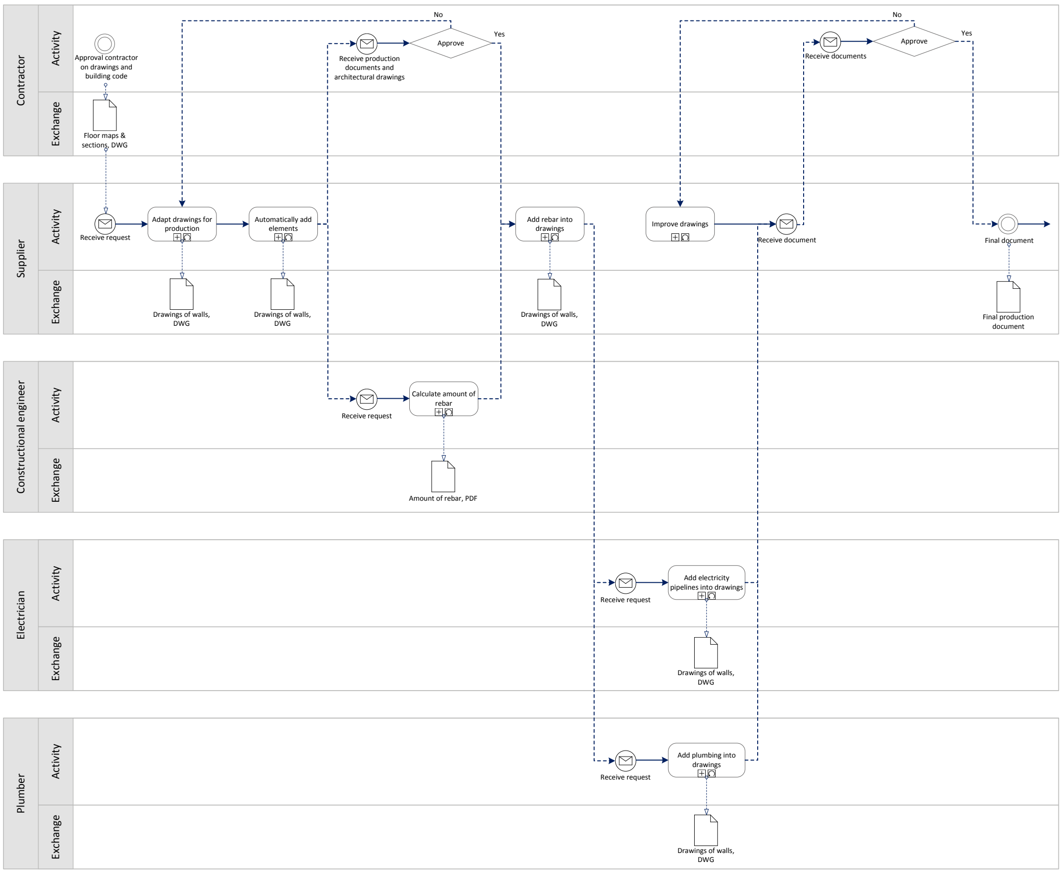


Figure 3.2.4-3: Detailed BPMN schema of information exchange between contractor and supplier

3.2.5 Conclusion

This chapter aims to answer the sub question 'What are the differences between the current and ideal situation of the defined construction process?' The answer of the question is approached in two ways. At first the current process is visualized with the use of BPMN, appendix 1. After the current process is established what the desired process should look like, appendix 2. The processes differ from the point after the approval of the customer about the façade and roof design. In the ideal situation the contractor, instead of the draftsman, generates the architectural and technical drawings with the use of the configurator. A model checker is used, before sending and after receiving the production documents of the supplier, to validate if all the necessary information is present in the model.

The main goal of this thesis is to verify the completeness of a building model using a model checker. in order to continue with the thesis it important to find out which specific parameters contribute to complete a building model. Chapter 3.3 elaborates upon this definition of a complete model.

3.3 Definition of the Exchange requirements

3.3.1 Introduction

Since the main focus of the construction process is established in the previous section, this part focuses on the determination of the parameters for the completion of a model, as shown in Figure 3.3.1-1. The corresponding sub-research question aimed to answer in this part is:

4. Which parameters need to be present in a building model before it can be considered complete?

It is necessary to establish which information is required for the exchanging of data between customer (Hurks) and supplier (Holcim), in order to achieve a single exchange of a complete model. To accomplish this, first it is important to know what is sent and what is checked in the current situation. Important to notice is that these questions are answered from the perspective of the contractor, because this thesis is conducted at the contractor's point of view. Next, a list is constructed including all parameters the BIM model should satisfy to, in order to be considered a complete model, suited to conduct only a single time exchange between contractor and supplier.

The information as presented here is generated by conducting interviews with Machiel van Den Brink (plan development), Rob Verweij (plan development) and Martijn Nieuwhoff (construction preparation) at the Da Vinci Huis department of the contractor Hurks, and with Dennis Nijs (draftsman) at Holcim, the supplier of prefab concrete walls. The approach of the interviews was semi-structured, the main issues which needed to come up during the interviews were noted on forehand, and functioned as rough guidelines and checklist.

3.3.2 What is sent?

At the current process Hurks sends a first version as a PDF or .dwg file to the supplier. These files contain architectural drawings, technical drawings, floor maps and elevations. However, the first versions are usually incomplete and need to be adjusted and checked. Usually three or four checking moments are required before the data of the building is finally approved.

3.3.3 What is checked?

Usually it takes three or four rounds before the information required for production is considered complete. In the first round the information is not yet complete, often due to the fact that not all information is available. For instance, the contractor has to wait for a response of a third party. The next rounds the contractor needs to check if the information is available and if it is processed in the production documents of the supplier.

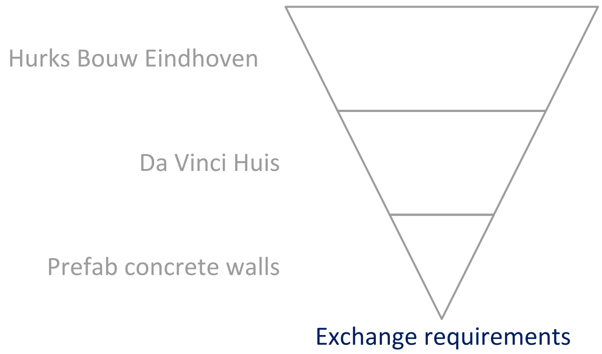


Figure 3.3.1-1: Concept of definition of Exchange Requirements

3.3.4 Parameters required to complete a model

To define the parameters which are required in order to complete the model, a distinction is made between the parameters required by the contractor and by the supplier. The research is conducted in the perspective of the contractor. However, the parameters required by the supplier are important to take into consideration as well. Otherwise the wall elements still do not satisfy the requirements of approval.

Not all the requirements of the supplier need to be taken into consideration. A distinction is made between the responsibilities of the contractor and those of the supplier. The contractor performs basic checks. In order to check more complex parameters of the wall elements, more specific knowledge is required, these checks should be conducted by the supplier.

An overview of the parameters can be found in appendix 3. In the first three columns the origin of the parameters are described. Those parameters are brought up via the interviews by either the contractor or the supplier, or it can be found in the data section. The data section is valuable information found in floor maps, sections or production documents of previous Da Vinci Huis projects, but did not come up during the interviews.

In the last column an overview is provided of the classification per parameter. This classification is based on the complexity of the rule according to (W. Solihin & Eastman, 2015), as explained in chapter 2.6 Model checking.

3.3.5 Exchange requirements based on the demands of the contractor

Following exchange requirements concerning prefab concrete wall elements, needs to be present in a building model before it is considered complete, according to the contractor.

NL/SfB

The NL/SfB, the ‘Elementenmethode’, is used for classification of building elements (BNA, 2005). This method is based on the Swedish SfB which is developed in 1947. It is an object orientated classification, used for ordering building objects in CAD systems, building estimations and documentation of related information. The NL/SfB consists of five tables (0 until 4), the most well-known table is the first one. This table is based on functions of the building parts. (BNA, 2005). Information concerning NLSfB method should be linked to the wall elements. It has a clear structure of classification and it is widely used and adapted.

Geometry
Other required information concerning prefab concrete wall elements is the geometry. When the supplier (Holcim) receives information, which does not have the .dwg extension, the wall elements are manually redrawn. Since it is not automatically generated, the process is error prone. Deviations appearing in the produced wall elements, have very negative influences. Because the contractor is responsible for correcting the geometry of the wall elements, there is a strong need to automate this process.

Weight
Usually the heaviest elements in a Da Vinci Huis project are the wall elements. The mass of a wall element is valuable information for the contractor. Since the selection of the crane is based on its capacity, which is determined by the heaviest element in the project. Quick access to this information is desired.

3.3.6 Exchange requirements based on the demands of the supplier

Following exchange requirements concerning prefab concrete wall elements, needs to be present in a building model before it is considered complete, according to the supplier of the wall elements.

Length
The maximum length of a wall element should, preferably, not exceed 11.40 meters. If the length of a wall element exceeds this limit, exceptional transport is required.

Height
Because of transport reasons, the height of a wall should not exceed 4.00 meters. Otherwise exceptional transport is required.

Weight
The weight of a wall element should not exceed 15.000 kilograms, because of transport limitations. However, this weight is rarely exceeded.

3.3.7 Exchange requirements based on information found in data

The exchange requirements concerning prefab concrete wall elements, found in floor maps, sections or production documents of previous Da Vinci Huis projects.

General project information
This is information which is necessary to be present in the model, in order to prevent confusion amongst stakeholders. The General project information consists of; project name, project description, organization name, author, date drawn, date final, project number, block number and address of the project.

Volume
Essential information in order to determine the weight of wall elements.

Type of concrete/density of concrete
Essential to determine the weight of wall elements.

Environmental classification
This is information found in data of the supplier. It is important this information is present in a model, in order to make a statement concerning the durability of the entire project. The environmental classification is based on the NEN-EN 206-1.

Wall ties (NL: spouwhaken)
This is data found in technical drawings of the supplier. This is information required when preparing a wall element in the executing phase.

Wall number
Every type of wall has its unique, company specific wall number. The supplier makes use of this numbering in order to know how often a type of wall occurs in a project, since a type of wall can appear just once or multiple times in a project.

3.3.8 Conclusion

The sub question aimed to answer in the part is, 'Which parameters need to be present in a building model before it can be considered complete?' An overview is created with information conducted in interviews with the contractor and the supplier of prefab concrete walls. This overview of the information which is required for a single, complete exchange between contractor and supplier can be found in appendix 3.

3.4 Options for extension of Revit plug-in tool

In this part is determined which parameters should be translated into options for the Autodesk Revit tool. The Autodesk Revit tool is the configurator which has been described in chapter 3.2.3 Da Vinci Huis. This configurator is a key element for the realization of the ideal situation in the Da Vinci Huis construction process. The exchange requirements, as defined in previous section, which are not already present in the outcome of the configurator should be added as options in this tool, so the configurator automatically generates a complete model in the future.

The options in this case equals the information which is required in order to consider the model complete, this corresponds with the goal as set up in the previous sub-chapter. The use of Autodesk Revit is assumed to be applied in this case, this means the parameters, as established in the previous sub-chapter, minus the standard information present in Revit equals the options for extension of the Revit plug in tool.

The following options for extension of the Revit plug in tool are:

- Project Description
- Date Final
- Type of concrete
- Density of concrete
- Wall ties
- Wall number
- Environmental classification
- NL/SfB

4
Model design

According to the hypothesis of this thesis, a model checker can contribute to a more efficient process In previous chapters is established on which part of the construction process is focused on and what the corresponding exchange requirements are in order to compose a complete building model. In this chapter the model checker, which verifies the completeness of a building model, is developed. The sub-research question belonging to this part is drawn as follows:

5. *Which type of validation tool is suited to verify the completeness of a building model and how can it be applied?*

First step in this chapter is to determine the requirements the model checker should comply to. Based on these requirements, a founded decision for a validation tool can be made. Thereafter, a detailed description of this model checker is provided.

In this chapter a distinction is made between how the checker works and how the checker should be used. How the checker works is described in the thesis on a detailed level. How the checker should be used is described in an appendix, though a short summary of the usage of the checker is provided.

At first, the model checker is designed only for the case of the Da Vinci Huis. However, after the development of the model checker specific for the case, it is tried to make the model checker more universal, so the checker could also be applied in other situations.

4.1 Selection between validation tools

In chapter 2.7 Validation tools, three types of validation tools are described, commercial checkers, validation tools based on mvdXML and validation tools based on semantic web technology. The choice of type of model checker is based on preset requirements, starting point for the requirements is that the checker should be able to be applied on the case of the Da Vinci Huis and should stimulate the development of model checking. These are the following requirements the model checker should comply to:

The validation tool should be able to apply rules with class 1 and 2

Rules with class 1 and 2 are rules that require a single or small number of explicit data and rules that require simple derived attribute values. The reason why it should be able to apply only rules with class 1 and 2 is that all individual Exchange Requirements of the Da Vinci Huis are either a class 1 or 2, this can be found in appendix 3.

The tool needs to be open sourced

If the tool is free and accessible for everybody, this enlarges the chance for future development and contributes to the adaptation of model checking in the building industry.

The model checker is user-friendly

The checker should be user-friendly, otherwise it will not be applied or further developed in the future. The validation tool should be easy to adjust and only a limited amount of extra specific software needs to be required. It should form a low threshold for the employee of the contractor to apply it.

The three validation tools, as elaborated upon in section 2.7, are the Commercial checker, mvdXML checker and a checker based on semantic web technology. Based on these three requirements one validation tool out of the three is eliminated on forehand. The commercial checker satisfies to the first and last requirement. However, a commercial checker is not open sourced, so two types of validation tools are left.

A model checker based on the semantic web based as well a model checker based on mvdXML could satisfy to all requirements. However, the principles of the semantic web are more suitable for making connections between data sets, while mvdXML is better suited for checking the presence of parameters in IFC files. In this case the use of a model checker based on mvdXML is considered the most suitable option.

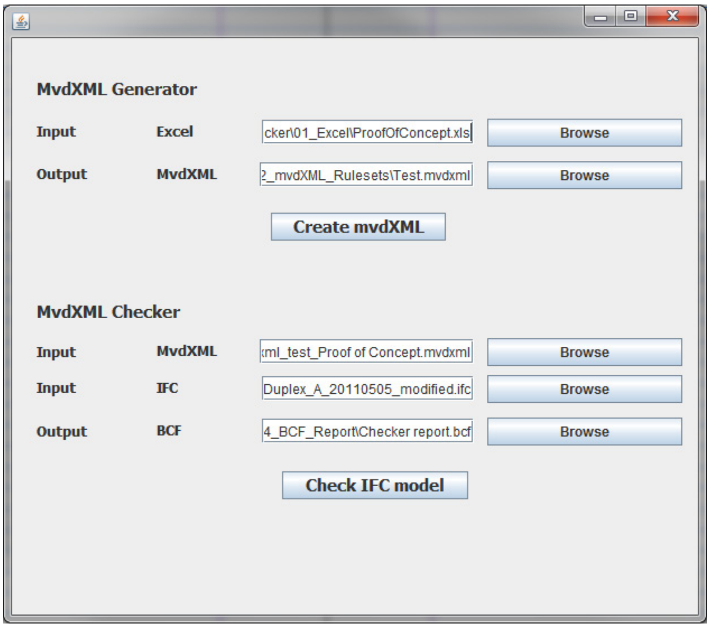


Figure 4.2.1-1: Interface of mvdXML Generator and Checker

4.2 MvdXML checker

4.2.1 Introduction

MvdXML is a data schema to capture Model View Definitions. An application which checks IFC models based on mvdXML has been developed (Zhang et al., 2015). This application is called mvdXML Generator and Checker and is a non-proprietary model view checker based on open standards to validate IFC building models. In Figure 4.2.1-1 the user interface is shown.

4.2.2 Process of mvdXML checker

This chapter describes the process of rule checking of the mvdXML Generator and Checker. The mvdXML checker consists of three parts; the first part is about the interpretation of requirements and structuring validation rule-sets, the second part is about the execution of the checks and the last part is about the report generation.

The structure of this chapter is based on the description of the rule checking process of (Eastman et al., 2009 P. 1016), as has been described in chapter 2.6 Model checking. The categories are; Rule interpretation, Building model preparation, Rule execution and Rule check reporting. In Figure 4.2.2-1 a simplified overview of the overall process of the mvdXML checker is provided. This figure is used in the explanation below to clarify the process.

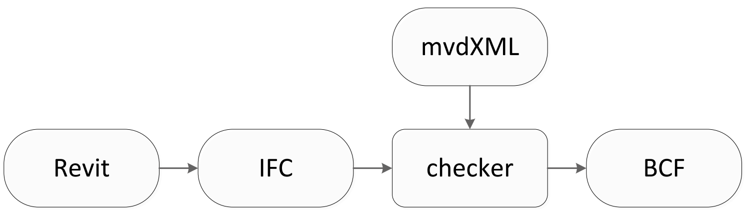


Figure 4.2.2-1: Simplified overview of the process of mvdXML checker

1. Rule interpretation

The first step in the rule checking process is about the interpretation of rules and the most crucial step in the total rule checking process (Zhang et al., 2015). In Figure 4.2.2-1: this step is the creation of mvdXML which is loaded into the checker. This first step in the rule checking process is similar to the first step of the mvdXML checker. The first part of the mvdXML checker is about the interpretation of requirements and structuring validation rule-sets.

The occurring conflict in this first phase is how the rules written in human language can be interpreted by a machine process without loss of data. This is achieved by generating rulesets in the mvdXML format. The mvdXML refers to an electronic format for representing such model view definitions. A MVD stands for a Model View Definition which describes contents of data to be exchanged in specific scenarios (Chipman et al., 2016).

Two ways of creating a mvdXML format are described in this section. The first way is by the use of IfcDoc the other way is by using a template.

Generating rules with the use of IfcDoc

The IFC Documentation Generator is released by BuildingSMART (BuildingSMART, n.d.). The purpose of the application is to read and write mvdXML rules by providing an interface which can be used for defining all content within mvdXML. A screenshot of the IFC Documentation Generator is shown in Figure 4.2.2-2. A mvdXML can be built up with the use of certain concepts like; model view, exchange requirement, concept, concept template, IfcObject, entities and attributes, which are all related to each other. A detailed explanation concerning these relationships can be found in appendix 4.

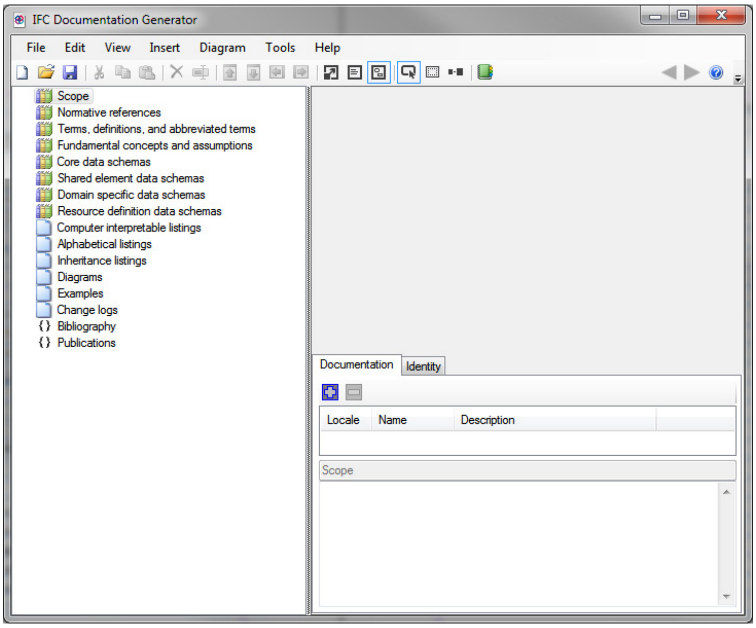


Figure 4.2.2-2: Screenshot of the IFC Documentation Generator

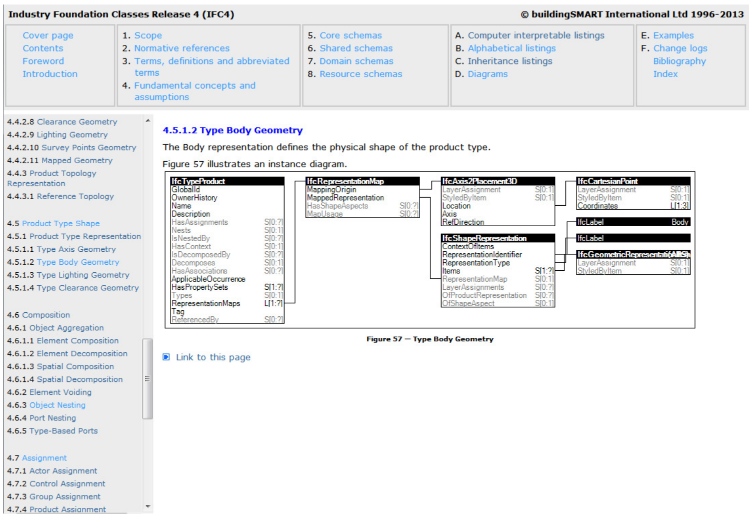


Figure 4.2.2-3: Screenshot of the IFC4 release

The structure of these rules are based on the IFC4 release (buildingSMART International Limited, n.d.). A screenshot of the IFC4 release is shown in Figure 4.2.2-3. A baseline could be added to the IFC Documentation Generator, in this case one does not have to start from scratch when converting mvdXML rules. The IFC Documentation Generator is difficult to use, profound knowledge of IFC, mvdXML and the IfcDoc tool is required. In this sub-chapter no further explanation about the concepts and relations of mvdXML are given. A detailed explanation is given concerning the concepts and relations of mvdXML in appendix 4, because it does not further support the process of th The creation of mvdXML rules is a complex process. Since IfcDoc is a valuable, however a difficult tool to use, a template is created for automatic generation of mvdXML rules by (J. Weerink, 2016). The spreadsheet enables domain end-users to specify requirements. IFC Support syntax is developed to convert the requirements into mvdXML rulesets. The generation of mvdXML files from the spreadsheet can be operated from a user-interface as shown in Figure 4.2.2-4.

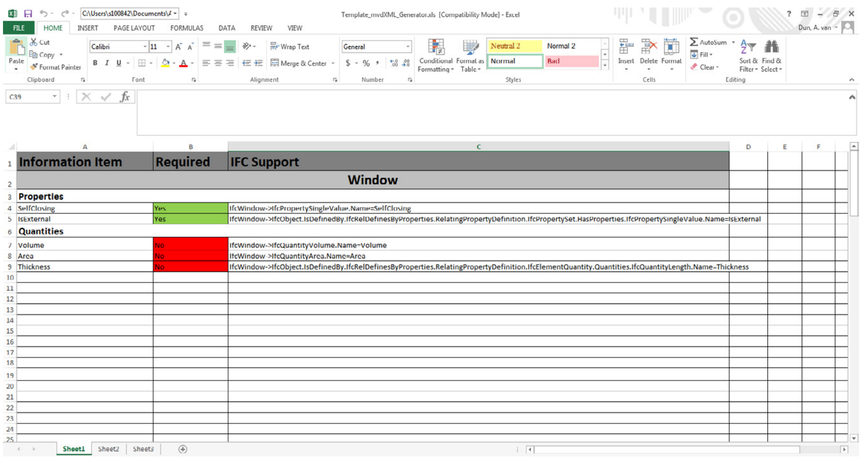


Figure 4.2.2-4: Screenshot of mvdXML Excel template

The template consists of three columns. The first column "Information Item" classifies the rule type and can be used to add a name to each rule. The second column "Required", specifies whether each rule should be converted by the mvdXML Generator or not. The third column "IFC Support" is a string which is transformed into mvdXML format by the mvdXML Generator. This string is based on the NATSPEC Object matrix, (NATSPEC BIM Object Element Matrix, 2008). The BIM Object/ Element Matrix is a Spreadsheet/Worksheet used for identifying and tracking BIM information during the project. One of the reasons why this matrix is used in practice is the completeness of it and the availability of the IFC support per information item, which is necessary for the third column.

2. Building model preparation
The building model preparation is in this case the conversion from the Autodesk Revit model to the IFC model, the first two blocks on the left in Figure 4.2.2-1. It is assumed that Autodesk Revit is used as software tool for the creation of the original file, since the configurator of the Da Vinci Huis is an Autodesk Revit plug-in. The conversion of an Autodesk Revit file is a process which can be controlled, partly because of the large variation in options Autodesk Revit supports and because of the auditability IFC offers. IFC is text based, which means a user can easily lookup using a text editor whether information is present or not. When the building model of Revit is converted into IFC, it needs to be saved in a PropertySet, so it can be checked in the Excel checker. A more detailed explanation about which options are required is provided in the user manual in appendix 5.

3. Rule execution
After the mvdXML rules and the building model are completed, the mvdXML checker is able to check the rulesets on the IFC. The mvdXML checker converts the IFC file into an EXPRESS schema of the IFC file, EXPRESS is a data specification language which consists of language elements that allow a data definition and specification of constraints on the data defined (ISO 10303-11:2004, n.d.). Next it is converted into an Eclipse Modelling Framework (EMF), which is a modeling framework and code generation facility for building tools and other applications based on a structured data model. (Richard Gronback, n.d.) Which is used to generate Java classes for entities of the IFC. The IFC objects and attributes from the instance file can be extracted by the developed mvdXML file. Depending on rule types in mvdXML, these values are checked to evaluate whether their existence, quantities, contents, uniqueness and conditional dependencies fulfil requirements or not (Chipman et al., 2016).

4. Rule check reporting
When the mvdXML checker executed the check, the last phase, the rule check reporting starts. The mvdXML checker captures the generated results in a BIM collaboration Format (BCF) report. BIM analysis software, such as Solibri Model Viewer or Tekla BIMSight, can be applied to display the BCF file. The BCF report consists of a markup and viewpoint file, the issues are stored in the markup file, the viewpoint file gives insight in the location of the issue.

4.2.3 Structure of mvdXML

As has been stated in previous sub-chapter, using mvdXML offers multiple benefits. However, mvdXML is difficult to work with when lacking profound knowledge. For future use and development it is desired to have a clear description of mvdXML, IFC and IfcDoc and their interrelationships. Since it becomes easier for an unexperienced user to apply the mvdXML checker, the development of the mvdXML checker becomes more accessible. The purpose of this thesis is to improve the process of the Da Vinci Huis, with the use of a mvdXML checker, this means the next step is to provide a detailed review about mvdXML, starting with a short description of XML. The goal is to get a grip on the structure of mvdXML model, which is related to the IFC structure and how it can be generated in the Ifc Documentation Generator. This detailed explanation has been put into one document, which can be found at appendix 4.

4.2.4 Conclusion model checker based on mvdXML

It is attempted to put the mvdXML Generator and Checker into practice. However, without success. In Figure 4.2.2-1 a simplified overview of the process of the mvdXML checker is shown. Based on these steps in combination with an example it is tried to find out why the use of a mvdXML based checker did not work.

The following example is used: a building model is created in Revit, containing three different external walls. One external wall is loadbearing, the other two walls are not. After the conversion of the Revit model into an IFC file, it can be checked using a text editor whether the parameter 'LoadBearing' is linked to the corresponding wall. If positive, multiple mvdXML files, all varying in the description of the parameter 'LoadBearing', are checked on the IFC model with the use of the mvdXML Generator and Checker.

In total 72 unique BCFzip files, a composition of BCF files generated each check, are generated. The example described above is one of the many variations which are generated. Below all the outcomes are collected and explained based on the example case.

1. None of the issues are reported, though some issues should appear – 15 times

Based on the example two issues are expected, because two walls do not have the parameter loadbearing, however it happened 15 times that none of the issues are reported though some issues sould have appeared.

2. All issues are reported, but none of the issues should appear – 33 times

In this case three issues appear in the BCF file, though only two issues should appear.

3. Some issues are reported, but not all of those issues should appear – 22 times

This outcome is applicable when more elements are checked. An example: if an IFC model contains four walls and two of them are loadbearing. Three issues, two loadbearing and one not loadbearing, appear as outcome.

4. Desired outcome – 2 times

Two issues linked to the right walls are addressed.

With only twice the desired result out of 72 attempts, the conclusion is drawn that the mvdXML Generator and Checker is too unreliable to apply. It is unknown why the mvdXML checker did not generate the required outcome. It could be a flaw in the checker, an implementation bug or a version problem. Another possibility could be that the mvdXML files and IFC files which are applied were not correctly formulated. Though, even if this is the case and it could be fixed easily, the mvdXML checker is still considered to be not user-friendly. Apparently it forms a challenge to generate a well performing mvdXML and corresponding IFC file. Because it is too error prone it is unlikely it will be adopted soon in a business environment.

For these reasons a new model checker is going to be developed, based on the same requirements as stated in section 4.1 Selection between validation tools.

4.3 Microsoft Excel checker

4.3.1 Explanation of development checker in Microsoft Excel

As concluded before, the mvdXML checker is considered not user-friendly, therefore the decision was made to develop a new type of checker. After conducting some research to alternatives to build a validation tool, a Microsoft Excel workbook is developed for the design of a model checker. The main reason to use Microsoft Excel is the low threshold for the employees of the contractor to make use of it.

A tool designed in Microsoft Excel meets all the preset requirements. It is able to apply rules with class 1 and class 2, the workbook itself is open sourced, user-friendly and it can be adjusted by employees of the company, since no specific knowledge and skills are required, furthermore only a limited amount of extra software is required. Since Microsoft Excel is a common software program it is assumed the employees of the company can easily access it. The creation of a user friendly model checker is the key feature of the new tool.

Goal of the new model checker is to create a validation tool which can perform a quick check and which can be easily accessed and adjusted by the employees of the contractor.

4.3.2 Introduction

Goal of this sub-chapter is to elaborate upon the model checker created in Microsoft Excel which is especially created for the purpose of the thesis. In this section all aspects concerning the model checker are elaborated. A distinction is made between the background of the checker; how the checker works, and the user-interface of the model checker; how the checker should be used.

First, the background of the checker is elaborated upon. A user manual concerning the model checker is provided in this thesis, it can be found in appendix 5. This manual focuses on the use of the Excel checker rather than the background processes.

The sub-chapter starts with the structure of the Excel file. Continued by a summary about the process of the Microsoft Excel checker in order to get an overall insight of how the checker works. Next, a more detailed description about the process of the background, is provided. Like the mvdXML Generator and Checker, the structure is based on the description of the rule checking process of (Eastman et al., 2009). The categories are; Rule interpretation, Building model preparation, Rule execution and Rule check reporting.

Since the development of the model checker is, at first, only based on the case of the Da Vinci Huis, the last part is dedicated to explaining the structure of checker, so it can be used in different situations other than the Da Vinci Huis.

4.3.3 Process of the Microsoft Excel checker

Structure of the Excel checker file

The Excel Checker is built up out of three types of sheets in order to keep the structure clear. The first type is the main sheet where the user has to fill in the requirements for conducting the check. The second type of sheets are the sheets displaying the outcome of the check. The last type are the sheets where the rules are executed. A sheet can have multiple functions.

Check IFC	Function
1. Overview	Filling in requirements and provides most general overview of the outcome
1_1 Project Information	Define general project information parameters
1_2 Present parameters	Overview of present parameters of element
1_2_1 PropSets rel to element	Find PropertySets related to element
1_2_2 Rel IfcPropSingleValue	Find IfcPropertySingleValues related to element
2. Overview per Paramater	Shows outcome of parameters per Element type
3. All element information	Collection of all available information
3_1 Object Name	Defines object name
3_2_X Value #X	Matches parameter with individual element
3_3 Combined value	Calculates the outcome of the combined value
3_4 Wall Assembly Code	Defines Wall Assembly code of element

Compare geometries	Function
1. Overview comparison	Compares volume of two models based on GUID
1_1 Find values	Finds volume per element

Table 4.3.3-1: Overview of types of sheets

In table 4.4.3-1 an overview of the types of sheets is provided. In order to conduct the check, two different workbooks are created; Check IFC and Compare geometries.

Summary about the process of the Microsoft Excel checker

In Figure 4.4.3-1 a simplified overview of the process of the Microsoft Excel checker is drawn. This figure is used in the explanation below to clarify the process.

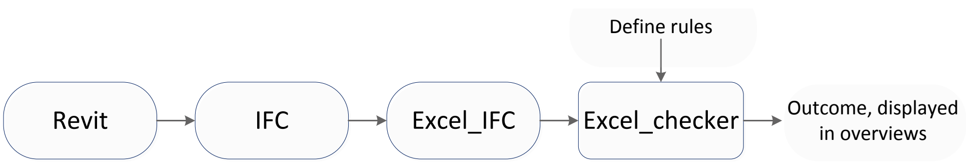


Figure 4.3.3-1: simplified overview of the process of the Microsoft Excel checker

It is of importance for conversion of the Autodesk Revit file to IFC to take into account certain settings, in order to obtain the information. Concerning the process of the Microsoft Excel checker, the first step is the creation of a model in Autodesk Revit. When this model is finished, it is converted into an IFC file.

Next step is to convert the IFC file to an Excel file. This process ensures the IFC to be readable for the Microsoft Excel checker. Again, certain settings must be applied so necessary information is maintained. To conduct this conversion from IFC to an Excel file, the IFC file analyzer of (R. R. Lipman, n.d.) is used. The IFC File Analyzer generates an Excel file from an IFC file, with the Excel_IFC as outcome. It contains one work sheet for each type of entity in the IFC file and a summary sheet (R. Lipman, 2010). The IFC File Analyzer reads single or multiple IFC files and reports the results of the coverage analysis in a spreadsheet application. The analyzer uses the IFCsvr ActiveX component (IFCsvr ActiveX component, 2013) to read and parse information from an IFC file. Thereafter, the rules applied on the model are defined. The main sheet of the Excel_ Checker is about addressing the issues which are checked.

When the parameters are set, the rules are executed on the building model. Information about, for instance the presence of certain parameters of an element, can be easily looked up in the file; in three overview sheets in Excel_checker the outcome is displayed on different levels of detail.

1. Rule interpretation

The first step in the rule checking process is about the interpretation of rules and is considered as the most crucial step in the total rule checking process. In this case the rule interpretation phase is about defining the rules. The interpretation of rules in the Excel_Checker file is defined by three levels. The building model, elements and parameters. A conceptual explanation of the relationships amongst the key elements of the rule interpretation is visualized in Figure 4.3.3-2.

Building model

The building model is the highest level concerning the definition of rules. It defines which building model the rules should be applied on. The building model is loaded into the Microsoft Excel checker by manually filling in the file name in the destined cell.

Elements

The element determines which part of the building model the rules should be applied on. The possible elements in the Microsoft Excel checker are; walls, windows, columns, doors, beams, slab and spaces. These options can be found in a drop down list in the destined cell on the main page of the Microsoft Excel checker.

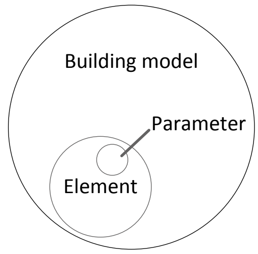


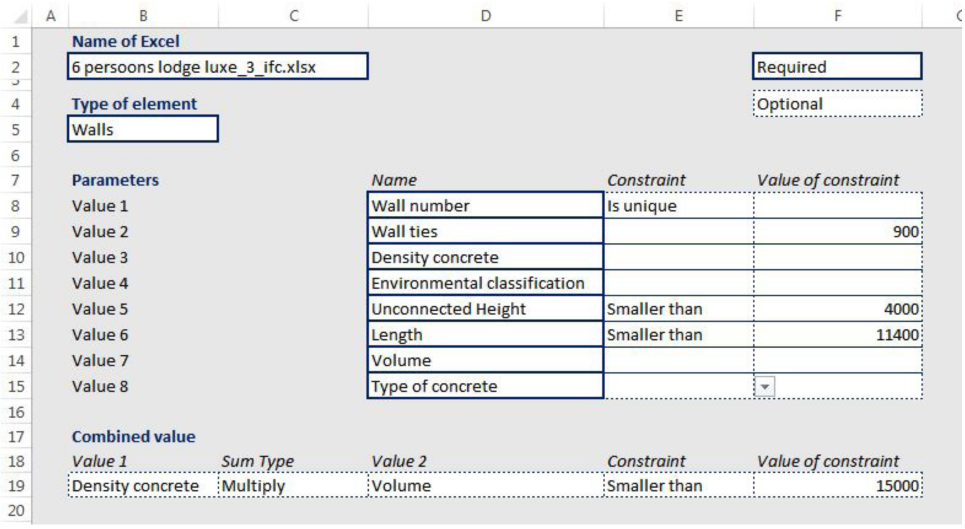
Figure 4.3.3-2: conceptual explanation of the relationships amongst the concepts of the rule interpretation

Parameters

The parameters define which part of information of an element the presence should be checked on. The possible parameters per element of the Microsoft Excel checker are presented in sheet 1_2 Present parameters and calculated in sheet 1_2_1 PropSets rel to element and 1_2_2 Rel IfcPropSingleValue. The possible parameters per element are shown as options in a drop down list per value. For each check one can choose eight parameters which are checked on presence.

Apart from the standard static value, it is also possible to create and check a combined value. For example, the value 'weight' is a dynamic parameter which is dependent on the volume and density of the material. However, this parameter cannot be easily added in Autodesk Revit. By combining these values in the model checker, the parameter 'weight' can be created and checked.
The constraints are linked to the parameters, the constraints can apply one rule to a parameter, apart from checking just the presence. It can check if a value of an element is unique, if it is smaller, bigger or equal to a certain value. If a user wants to apply more than one constraint on a parameter, it is possible by filling in twice the same parameter and different constraint as a value.

In Figure 4.3.3-3 a screenshot of the main sheet, 1. Overview, of the IFC check workbook is shown. In this sheet one has to fill in the name of Excel, the type of element, the parameters and constraints. The sheet is not extindible, however it is built up in a generic way, so it can be applied in other cases as well. In chapter 4.3.4 a more detailed explanation is provided.



	A	B	C	D	E	F
1		**Name of Excel**				
2		6 persoons lodge luxe_3_ifc.xlsx				Required
4		**Type of element**				Optional
5		Walls				
6						
7		**Parameters**		*Name*	*Constraint*	*Value of constraint*
8		Value 1		Wall number	Is unique	
9		Value 2		Wall ties		900
10		Value 3		Density concrete		
11		Value 4		Environmental classification		
12		Value 5		Unconnected Height	Smaller than	4000
13		Value 6		Length	Smaller than	11400
14		Value 7		Volume		
15		Value 8		Type of concrete		
16						
17		**Combined value**				
18		*Value 1*	*Sum Type*	*Value 2*	*Constraint*	*Value of constraint*
19		Density concrete	Multiply	Volume	Smaller than	15000
20						

Figure 4.3.3-3 screenshot of the main sheet

2. Building model preparation

The second phase of the rule checking process is about the preparation of the building model itself and relates to the first three blocks in Figure 4.4.3-1. The correctness of the model is crucial since building models are becoming extensive and complex sets of data with the use of Building Information Modeling. In this case the preparation of the Building model is divided into three steps:

1. Completion of the model using native software

In the case of Da Vinci Huis the original software is Autodesk Revit. The use of other construction related software is possible as well, the only requirement is that the software supports a decent IFC converter. Preferably a converter with many options concerning the conversion of the original file into IFC.

2. Conversion to IFC

When the building model is complete and contains all necessary information, the model is converted into an IFC file. It is necessary to take the export options into account for the conversion of the original file into IFC, in order to obtain important parameters.

3. IFC to Excel

Next step is to convert the IFC file to an Excel file. This process ensures the IFC to be readable for the Microsoft Excel checker. Again, certain settings must be applied so necessary information is maintained. To conduct this conversion from IFC to an Excel file, the IFC file analyzer of (Lipman, n.d.) is used. The IFC File Analyzer generates an Excel file from an IFC file. The name of this file needs to be filled in below the Name of Excel cell in the main sheet.

3. Rule execution

In the previous two phases the data and rules are established. In this phase the rules are checked on the model. At first the necessary information concerning the structure of the IFC when exported to Microsoft Excel is provided to give some basic background information. Thereafter is explained how the rule check is conducted.

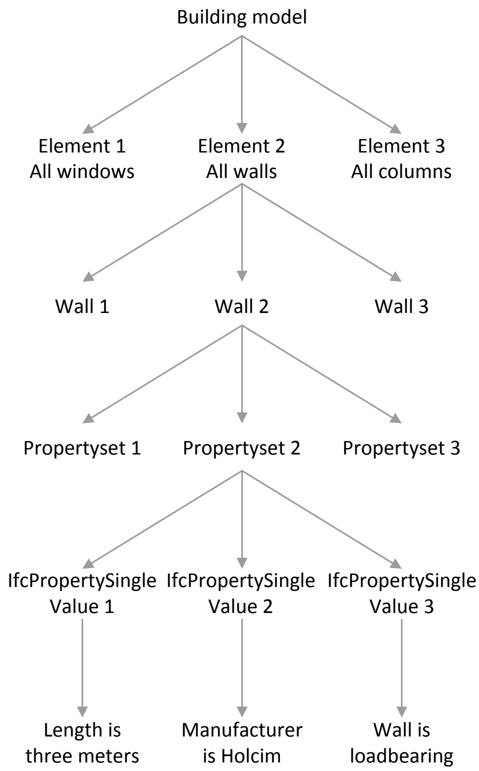


Figure 4.3.4-4: A conceptual image of the description of the structure of the IFC

In order to clarify the description of the structure of the IFC, a conceptual image is added, figure 4.4.3-4. The upper layer is the IFC, this building model consists of group of elements. Examples of these groups of elements are; walls, windows, columns, doors, slabs beams and spaces. All these groups are described in its own sheet if information concerning these elements is present. This sheet handles all cases of these groups, where each individual element of the group is extruded vertically. Each individual element has one or more propertysets. The IfcPropertySet is a combination of multiple properties. Each individual propertyset consists of one or more IfcPropertySingleValues. A propertysinglevalue defines a property object which has a single (numeric or descriptive) value assigned.

Example of a rule check
A simplified example is used to clarify this description, the case is drawn as follows:

> *A user wants to know if the information concerning the material of the wall elements are present in the model, and if the values are unique.*

The execution of the rule check consists of two parts, at first all individual parts of the element group (the walls) are accessed for all lookups. This is done based on the Globally Unique Identifier (GUID) of the elements. A GUID provides a way of uniquely identifying an object. The lookup is based on the GUID of the elements to make sure all elements are displayed. As shown in Table 4.4.3-2.

Wall name	*GUID*
Wall A	09685hfd8
Wall B	31574pbs3
Wall C	95112vch6
Wall D	74396ihh4
Wall E	99632mas2

Table 4.3.3-2: Lookup based on GUID

The next step is to find all IfcPropertySingleValues of the required parameters as defined on the main page and match it with the GUID of the individual elements. To achieve this, the single values are linked to the propertysets, the propertysets are again matched with the GUID of the individual elements, Table 4.4.3-3 shows this principle.

IfcPropertySingleValue	*NominalValue*	*IfcPropertySet*	*GUID*
Material	Brick	256	09685hfd8
Material	Brick	285	95112vch6
Material	Concrete	317	74396ihh4

Table 4.3.3-3: Find all IfcPropertySingleValues and link to GUID

Since the single values are linked to the GUID of the individual elements, it can be concluded which wall elements lack the specific single values, as showed in Table 4.4.3-4.

Wall name	*GUID*	*NominalValue*
Wall A	09685hfd8	Brick
Wall B	31574pbs3	
Wall C	95112vch6	Brick
Wall D	74396ihh4	Concrete
Wall E	99632mas2	

Table 4.3.3-4: Link value to element name

The constraints, as drawn on the main page, is based on the value of the IfcPropertySingleValue. In the example below, Table 4.4.3-5, the constraint 'Is unique' is applied.

Wall name	*GUID*	*NominalValue*	*Constraint*
Wall A	09685hfd8	Brick	No
Wall B	31574pbs3		
Wall C	95112vch6	Brick	No
Wall D	74396ihh4	Concrete	Yes
Wall E	99632mas2		

Table 4.3.3-5: Apply constraints on the values

Of all exchange requirements, as established in Chapter 3, the presence of following parameters are calculated in this way; NL/SfB, Length, Height, General project information, Volume, Type of concrete, Density of concrete, Environmental classification, Wall ties and Wall number. Each value has its own sheet where the GUID and Nominal value are linked. The parameter 'Weight' is created by combining the Density of concrete and Volume parameter.

As stated in the introduction of this section, the model checker should be able to check all exchange requirements of chapter 3.5. Most of them are covered way as described. However, one of the main activities the contractor performs is manually checking whether the geometry of two wall elements are similar. To automate this process a separate Excel file is created, in order to improve the performance of the main IFC checker file. The geometry check is based on the principle that a GUID of an element stays the same when the dimensions are changed.

In figure 4.4.3-5 a screenshot of the Excel file is shown. To conduct the check one has to fill in the names of both IFC Excel files and the element that needs to be checked.

	B	C	D	E	F
1		**Name of Excel 1**		**Name of Excel 2**	
2		6 persoons lodge luxe_3_ifc.xlsx		6 persoons lodge luxe_3_2_ifc.xlsx	
3					
4		**Element**			
5		Walls			
6					
7		**GUID of wall elements not the same**			
8		20ByCA$dXEFeTAIGKcGWqP			
9					
10					
11					
12					
13					
14					
15					
16					
17					
18		**GUID**	**Volume**	**Volume**	**Is the same?**
19		20ByCA$dXEFeTAIGKcGWos	0.0377039999999898	0.0377039999999898	TRUE
20		20ByCA$dXEFeTAIGKcGWou	0.504239999999997	0.504239999999997	TRUE
21		20ByCA$dXEFeTAIGKcGWo$	0.381840000000001	0.381840000000001	TRUE
22		20ByCA$dXEFeTAIGKcGWo_	0.198318111133256	0.198318111133256	TRUE
23		20ByCA$dXEFeTAIGKcGWoz	2.35449999999998	2.35449999999998	TRUE
24		20ByCA$dXEFeTAIGKcGWr3	1.11496018970329	1.11496018970329	TRUE
25		20ByCA$dXEFeTAIGKcGWr1	0.826253810296682	0.826253810296682	TRUE
26		20ByCA$dXEFeTAIGKcGWrN	1.56404700000072	1.56404700000072	TRUE
27		20ByCA$dXEFeTAIGKcGWrL	1.29575550000002	1.29575550000002	TRUE
28		20ByCA$dXEFeTAIGKcGWrR	1.8804638098043	1.8804638098043	TRUE
29		20ByCA$dXEFeTAIGKcGWrO	0.522217499999999	0.522217499999999	TRUE
30		20ByCA$dXEFeTAIGKcGWrV	0.654272499999999	0.654272499999999	TRUE

4.3.3-5: screenshot of Compare geometries Excel file

When all necessary information is filled in, the volumes of each element are established. The volume described in IFC is very precise, so if the geometry of an element changes, the volume always changes as well. If the volumes of two individual elements differ from each other, the GUID of this element is shown.

	B	C	D	F	G	I	J	L	M	O	P	R	S	U	
1	**All wal information**														
7															
8			*Value 1*		*Value 2*		*Value 3*		*Value 4*		*Value 5*		*Value 6*		*Va*
9	*GUID*	*Element Name*	*Wall number*	*Const.*	*Wall ties*	*Const.*	*Density conc.*		*Environment*	*Const.*	*Unconnected*	*Const.*	*Length*	*Const.*	*Vo*
10	20ByCA$dXEFeTAIGKcGWos	Basic Wall:EPS 80 mm		TRUE							600	TRUE	745.49999999	TRUE	0
11	20ByCA$dXEFeTAIGKcGWou	Basic Wall:EPS 80 mm		TRUE							600	TRUE	10424.999999	TRUE	0
12	20ByCA$dXEFeTAIGKcGWo$	Basic Wall:EPS 80 mm		TRUE							600	TRUE	7955.0000000	TRUE	0
13	20ByCA$dXEFeTAIGKcGWo_	Basic Wall:EPS 80 mm		TRUE							600	TRUE	4171.6273152	TRUE	0
14	20ByCA$dXEFeTAIGKcGWoz	Basic Wall:Prefab betonwand 100 mm		TRUE							2620.0000000	TRUE	10104.999999	TRUE	2
15	20ByCA$dXEFeTAIGKcGWr3	Basic Wall:Prefab betonwand 100 mm		TRUE							2620.0000000	TRUE	4990.3726847	TRUE	1
16	20ByCA$dXEFeTAIGKcGWr1	Basic Wall:Prefab betonwand 100 mm		TRUE							2420.0000000	TRUE	4086.6273152	TRUE	0
17	20ByCA$dXEFeTAIGKcGWrN	Basic Wall:Prefab betonwand 90 mm		TRUE							2620.0000000	TRUE	7875.0000000	TRUE	1
18	20ByCA$dXEFeTAIGKcGWrL	Basic Wall:Prefab betonwand 150 mm		TRUE							2800.0000000	TRUE	7875.0000000	TRUE	1
19	20ByCA$dXEFeTAIGKcGWrR	Basic Wall:Prefab betonwand 90 mm		TRUE	No		60		D2/G6		5377	FALSE	7774.9999999	TRUE	1
20	20ByCA$dXEFeTAIGKcGWrO	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	3850	TRUE	0
21	20ByCA$dXEFeTAIGKcGWrV	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	3850	TRUE	0
22	20ByCA$dXEFeTAIGKcGWrU	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	975	TRUE	0
23	20ByCA$dXEFeTAIGKcGWrT	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	4214.9999999	TRUE	0
24	20ByCA$dXEFeTAIGKcGWrS	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	2250	TRUE	0
25	20ByCA$dXEFeTAIGKcGWrZ	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	1185	TRUE	0
26	20ByCA$dXEFeTAIGKcGWrY	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	2270	TRUE	0
27	20ByCA$dXEFeTAIGKcGWrX	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE	Yes				h7/0o		2450.0000000	TRUE	609.99999999	TRUE	0
28	20ByCA$dXEFeTAIGKcGWrW	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	1755.0000000	TRUE	0
29	20ByCA$dXEFeTAIGKcGWrd	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	2580	TRUE	0
30	20ByCA$dXEFeTAIGKcGWrc	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	694.99999999	TRUE	0
31	20ByCA$dXEFeTAIGKcGWrb	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	2580	TRUE	0
32	20ByCA$dXEFeTAIGKcGWra	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	2579.9999999	TRUE	0
33	20ByCA$dXEFeTAIGKcGWrh	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	1715	TRUE	0
34	20ByCA$dXEFeTAIGKcGWrg	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	670	TRUE	0
35	20ByCA$dXEFeTAIGKcGWrf	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	1175	TRUE	0
36	20ByCA$dXEFeTAIGKcGWru	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	975	TRUE	0
37	20ByCA$dXEFeTAIGKcGWqQ	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	10504.999999	TRUE	4
38	20ByCA$dXEFeTAIGKcGWqP	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	8035.0000000	TRUE	2
39	20ByCA$dXEFeTAIGKcGWqO	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	8035.0000000	TRUE	1
40	20ByCA$dXEFeTAIGKcGWqa	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	694.99999999	TRUE	0
41	20ByCA$dXEFeTAIGKcGWAa	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	10504.999999	TRUE	3
42	20ByCA$dXEFeTAIGKcGWAm	Basic Wall:EPS 80 mm		TRUE							530	TRUE	2048	TRUE	0
43	20ByCA$dXEFeTAIGKcGWAt	Basic Wall:EPS 80 mm		TRUE							600	TRUE	110.00000000	TRUE	0
44	20ByCA$dXEFeTAIGKcGWAs	Basic Wall:EPS 80 mm		TRUE							530	TRUE	2148	TRUE	0
45	20ByCA$dXEFeTAIGKcGWAr	Basic Wall:EPS 80 mm		TRUE							600	TRUE	110.00000000	TRUE	0
46	20ByCA$dXEFeTAIGKcGWAq	Basic Wall:EPS 80 mm		TRUE							530	TRUE	2048	TRUE	0

Figure 4.3.3-6: screenshot of overview of most detailed level

4. Reporting checking results

The last phase is focused on displaying the outcome of the rule execution. In the Excel_Checker the outcome is displayed on three different levels.

At the sheet '3 All element information', the information is displayed with the highest level of detail. All information regarding the individual elements and the matching outcome of the values and constraints are shown in this sheet. A screenshot of this level is shown in Figure 4.4.3-6.
Since one does not always want to know all information the next level shows a less detailed overview. In the sheet '2 Overview per Parameter', is per element name the outcome of the parameters categorized. It is stated whether all parameters are complete or uncomplete, combined with the number of the amount of missing values per parameter per element name. A screenshot of this level is shown in Figure 4.4.3-7.

Overview per parameter

ObjectType	Total	Overall	Wall number	Wall ties	Density concrete	Environmental class	Unconnected Height	Length
Basic Wall:EPS 80 mm	12	*Not complete*	Not complete (12)	Not complete (12)	Not complete (12)	Not complete (12)	Complete	Complete
Basic Wall:Prefab betonwand 100 mm	3	*Not complete*	Not complete (3)	Not complete (3)	Not complete (3)	Not complete (3)	Complete	Complete
Basic Wall:Prefab betonwand 90 mm	2	*Not complete*	Not complete (2)	Not complete (1)	Not complete (1)	Not complete (1)	Complete	Complete
Basic Wall:Prefab betonwand 150 mm	1	*Not complete*	Not complete (1)	Not complete (1)	Not complete (1)	Not complete (1)	Complete	Complete
Basic Wall:Ytong cellenbeton G4/600 70 mm	13	*Not complete*	Not complete (13)	Not complete (12)	Not complete (13)	Not complete (12)	Complete	Complete
Basic Wall:Ytong cellenbeton G5/800 100 mm	5	*Not complete*	Not complete (5)	Not complete (5)	Not complete (5)	Not complete (5)	Complete	Complete
Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm	4	*Not complete*	Not complete (4)	Not complete (4)	Complete	Complete	Complete	Complete

Figure 4.3.3-7: screenshot of overview per parameter

Sheet '1. Overview' is the most general overview. In this overview per element name whether any value is missing or if all information per element name is complete is displayed. Directly after conducting the check, the user can see at the same page if all information is complete or incomplete.

Concerning the display of values in the Checker_Excel, it is taken into account that certain software tools automatically add a value to a parameter. For instance, Autodesk Revit automatically fills in 'Enter your address here' at the parameter Address. It is tried to filter out these standard values, by mentioning 'Standard Value' when one of these standard values appear.

4.3.4 Using the Microsoft Excel checker at other cases

Since the development of the model checker is mainly based on the checking of parameters of the case of the Da Vinci Huis, this part is dedicated to the explanation of how the checker could be used in situations other than the Da Vinci Huis.

The Microsoft Excel checker is universally developed, so the checker can also be applied in other situations. The export of IFC to IFC_Excel is conducted using a uniform format, so the structure of the data is always the same. This makes it possible to access all required values in the Excel checker file.

The lookups conducted in the Excel Checker are all based on formulas containing the name of the IFC_Excel and the sheet name. Again, because the sheets are named the same, uniform formulas can be conducted using a reference to the filename at the main page.

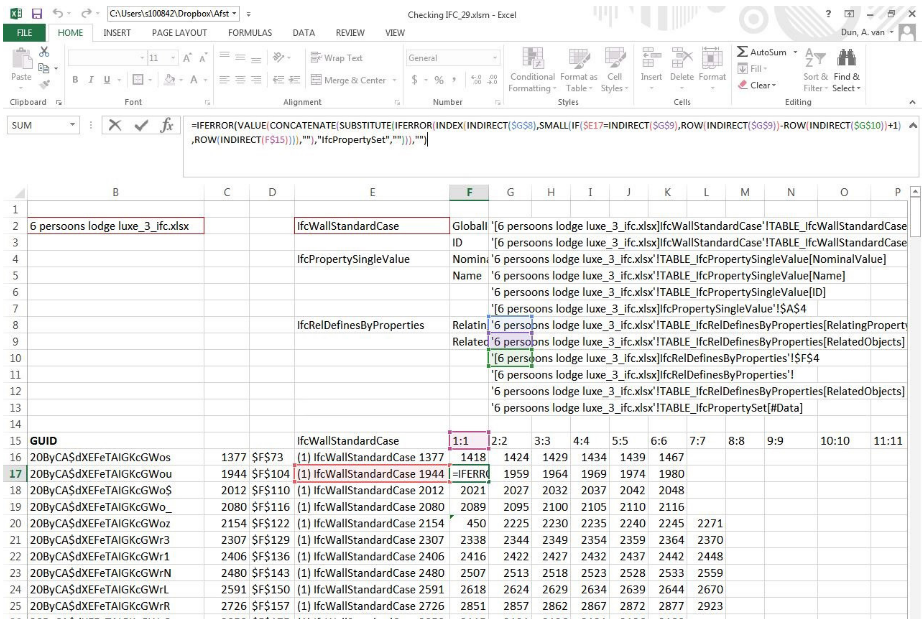


Figure 4.3.3-8: example of a calculation sheet

Figure 4.4.3-8 shows an example of a sheet where the required values are established. The name of the model, 6 persoons lodge luxe_3_ifc.xlsx, is established at the main sheet and referred to in cell B2. IfcWallStandardCase, located in cell E2, is the sheet containing all information of the wall elements. In column G, the file path references are concatenated out of different cells. If another model is loaded into the Excel checker or the element changes, the filepath references change automatically.

Not all IFC files appeared to be suitable for the Excel checker. A high quality of IFC is required, containing parameters which are stored based on the structure of IfcPropertySets and IfcPropertySingleValues. Random IFC files have been checked, most of them appeared to be not suitable for the check, since the IFC files were lacking relevant information or contained parameters which are not stored based on the required structure. However, when the export setting as described in the manual in appendix 5 are taken into account, the IFC files are suited to check.

4.3.5 Conclusion

The sub-research question aimed to answer in this part of the thesis is ‘Which type of validation tool is suited to verify the completeness of a building model and how can it be applied?’

The first step in this chapter was to determine the requirements the checker should satisfy to. At first a model checker based on mvdXML is developed. However, when put into practice the mvdXML based checker appeared to be not user-friendly, which is one of the essential requirements concerning the checker, so a new kind of validation tool is developed.

As a response the key element of the new checker focuses on creating a user-friendly model checker. A tool made in Microsoft Excel meets all the preset requirements. The developed workbook in Excel is open-sourced and it can apply rules with class 1 and 2. A checker developed in Excel is user-friendly because of the low threshold to open the software. Besides, the average employee knows how to work with Excel since no specific knowledge is required. Though the model checker can be slow, because it has to interact constantly between the IFC Excel, where all information is stored, and the checker Excel, which conducts the check and contains a lot of formulas, it is still faster than manually checking. Besides, the Excel checker can be applied at other cases. Because of the use of drop-down lists, the user is limited to only those options which can be checked.

An IFC checker, like the Microsoft Excel checker, can be applied to validate exchange requirements of prefab concrete wall elements between customer and supplier.

Conclusion

Goal of this research is to verify the completeness of a building model using a model checker. This chapter starts with the problem definition leading to the main research question as formulated in chapter one. Thereafter, the research question is explained based on the sub questions. Finally, the recommendations concerning future development are discussed.

Starting point for the graduation thesis is a review concerning the traditional process, due to a lack of unambiguously communication at the traditional process, mistakes are made which could have been prevented in the first place. BIM offers a solution where all data is stored in one accessible place, in order to improve communication and obtain a higher level of quality compared to the traditional construction process.

However, with the use of BIM in the building process, the problem of interoperability appears. The involved parties all have various preferences regarding the use of software. In order to be able to exchange information amongst the parties, the IFC schema is conceived.

Though, IFC is highly redundant which is considered as a downside of the extension. It offers many ways to define objects, relations, and attributes. As a result, users are not willing to rely on and accept IFC as exchange format. A solution for this limitation is a model checker; a validation tool which checks if certain information is stored in the IFC.

This problem definition resulted into the following main research question:

'How can the completeness of a building model be verified using a model checker?'

In order to be able to answer this question, a research has been conducted to the exchange of information between customer and supplier in a building process.

At first, it is determined at which particular building process the research is conducted. The thesis focuses on the exchange of data between customer and supplier of prefab concrete wall elements of the Da Vinci Huis process. Aim of the Da Vinci Huis concept is to combine the advantages of a standardized house, with the desire for freedom of choice in the design.

At this construction process is zoomed in on a detailed level in order to gain more knowledge concerning what needs to be improved. Main goal of this thesis is to verify the completeness of a building model, in order to reach this goal, it has to be established when a building model can be considered complete. Based on interviews with the contractor and supplier of prefab wall elements, a list of parameters which the wall elements should comply to is created.

The choice for a model checker is based on three requirements. It should be user-friendly, open sourced and able to apply the composed list of parameters. Based on these demands a model checker is developed based on Microsoft Excel. By using Excel a high level of usability for the company employee is obtained.

A Microsoft Excel based checker can verify the completeness of a building model. Exchange requirements for the completion of a model are created and translated into rules. As a result, when conducted a check in Microsoft Excel it can be directly seen what information is missing or does not satisfy to the rules. When using a model checker one can be sure whether all required information concerning the model is present in the IFC file or not.

The model checker is at first designed for the specific case of the Da Vinci Huis and the Hurks employee. To find out if a model checker can be applied at other construction processes as well and in order to provide a more general answer, the model checker is applied on other cases than the Da Vinci Huis process. The outcome is that not all building models are suited, depending on the quality of IFC, file size and structure. However, to answer the main research question: an IFC checker, like the Microsoft Excel checker, can be applied to verify the completeness of a building model.

Recommendations for future development

This graduation research aims to stimulate the implementation of BIM on a small scale. As stated in the literature review, the implementation of BIM can offer multiple benefits. However, certain challenges have to be addressed and developed in order to achieve full implementation in the construction industry.

The first improvement to address is that more standardization concerning the structure of IFC is required. During this graduation process it is made clear that the way an IFC model is built up strongly dependents on the original software and export options. If all information containing in IFC is stored in a uniform way it would improve the usability of IFC, since it becomes easier to develop basic software which can apply rule checking. It is recommended to conduct a research providing more insight concerning the conversion and its options from original file to an IFC file, for all types of exports of different kinds of software and decide on what is the best way how an IFC should be built up. In other words, strive to a uniform structure of IFC files.

With the current checker geometries of elements in IFC files cannot be checked on a detailed level. Even though possibilities exist to insert a viewer plug-in in Excel, other software tools might be more convenient to conduct these kinds of checks.

There is room for improvement concerning the overview of the outcome of the model checker. An option would be to create a link with a .BCF report, which is the common report extension for the exchange of issues. With this report it is possible to provide more insight concerning the physical location of the issue.

A limitation of the checker is the performance of the Excel checker when large IFC files are checked. It is difficult to establish the limitation size of an Excel or IFC file. The required calculation time is, for instance, dependent on the type of computer. A newer type of computer with more processors does not necessarily mean a smaller calculation time, though the opposite is expected. Besides, the limitation size is difficult to state since it is depending on how long someone is willing to wait.

An opportunity for development of the Excel checker lies in reducing the calculation time. An example is applying a MvD before calculating so a part of the data is filtered. As a result, not all data of the IFC needs to be checked, which has a positive effect on the calculation time.

The current version of the Excel checker makes use of formulas which refer to data in a different workbook, the IFC Excel. This is a time consuming job for the total calculation. An opportunity lies in improving this by, for instance, combining the formulas and data in the same workbook.

The drop down list at the parameters part in the main sheet is generated after calculation. However, using this drop down list first and calculating again afterwards requires two complete calculations, while only a part of the calculation is needed to generate the possible options. A button which is developed to only calculate the options for parameters, prevents the checker to conduct a complete calculation twice.

Bibliography

Bedrick, J. (2008). AIA, Organizing the Development of a Building Information Model, 7–10.

Beetz, J., & Berlo, L. Van. (2010). BIMSERVER . ORG – AN OPEN SOURCE IFC MODEL SERVER, (Weise 2006), 16–18.

BIM Industry Working Group (BIWG). (2011). Strategy Paper for the Government Construction Client Group From the BIM Industry Working Group. Department of Business, Innovation and Skills, URN 11, (March), 1–107.

BIMForum. (2015). Level of Development Specification. BIM Forum, 195.

BIMstore. (n.d.). Wat is LOD (Level of Development)? Retrieved October 11, 2016, from http://bimstore.nl/bim-info/over-bim/39-wat-is-lod-level-of-development

BNA. (2005). Elementenmethode NL/SfB, 215. Retrieved from http://www.stabu.org/wp-content/uploads/2015/07/NL-SfB_BNA_Boek_2005-ISBN-10-90-807626-3-6.pdf

BuildingSMART. (n.d.). Vision & Mission. Retrieved October 4, 2016, from http://buildingsmart.org/about/vision-mission/

BuildingSMART. (2010). Information Delivery Manual Guide to Components and Development Methods. buildingSMART, 1–84.

buildingSMART International Limited. (n.d.). Industry Foundation Classes Release 4 (IFC4). Retrieved October 10, 2016, from http://www.buildingsmart-tech.org/ifc/IFC4/final/html/

Chipman, T., Liebich, T., & Weise, M. (2016). mvdXML, 1.1, 49.

Christiansson, P., Svidt, K., Pedersen, K. B., Advisor, C., Dybro, U., & Manager, J. (2011). User participation in the building process. Journal of Information Technology in Construction, 16(July 2010), 309–335.

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, 27(2), 206–219. http://doi.org/10.1016/j.aei.2012.10.003

de Vries, H. J. (2005). IT Standards Typology. Advanced Topics in In- formation Technology Standards and Standardization Research (Vol. 1).

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

Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors. Building (Vol. 2). http://doi.org/10.1002/9780470261309

Fiatech Regulatory Streamlining Committee. (2012). AutoCodes project: phase 1, proof-of-concept final report, (March), 20.

Gallaher, M. P., O'Conor, A. C., Dettbarn, J. L., & Gilday, L. T. (2004). Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. Nist, 1–210. http://doi.org/10.6028/NIST.GCR.04-867

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

Heidari, M., Allameh, E., De Vries, B., Timmermans, H., Jessurun, J., & Mozaffar, F. (2014). Smart-BIM virtual prototype implementation. Automation in Construction, 39, 134–144. http://doi.org/10.1016/j.autcon.2013.07.004

Het Rijksvastgoedbedrijf. (2013). RVB BIM Norm, 31.

Hjelseth, E., & Nisbet, N. (2010). Overview of concepts for model checking. Proceedings of the CIB W78 2010: 27th International Conference –Cairo, Egypt, 16–18.

Hurks. (2016). Hurks. Retrieved from http://www.hurks.nl/en/about-us/organisation/

IFCsvr ActiveX component. (2013). Retrieved March 6, 2017, from (http://tech.groups.yahoo.com/group/ifcsvr-users/

ISO 10303-11:2004. (n.d.). Express language. Retrieved February 14, 2017, from http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38047

Krijnen, T., & van Berlo, L. (2016). Methodologies for requirement checking on building models. http://doi.org/ISBN: 978-90-386-4144-7

Laakso, M., & Kiviniemi, A. (2012). The IFC standard - A review of history, development, and standardization. Electronic Journal of Information Technology in Construction, 17(May), 134–161.

Lehtinen, S. (2006). The useful minimum.

Liebich, T., & Wix, J. (1999). Highlights of the development process of industry foundation classes. 8th International Conference on Durability of Building, 1997(January 1997). Retrieved from http://www.irbdirekt.de/daten/iconda/CIB2249.pdf

Lipman, R. (2010). Developing Coverage Analysis for Ifc Files. Proceedings of CIB W78 2010, (1), 1–10.

Lipman, R. R. (n.d.). IFC-File-Analyzer. Retrieved from https://www.nist.gov/services-resources/software/ifc-file-analyzer

Mark Bew and Mervyn Richards. (2008). UK maturity model. Retrieved January 13, 2017, from http://www.bimtaskgroup.org/wp-content/uploads/2012/07/q15.jpg

Master Thesis_Jesse Weerink. (2016).

NATSPEC. (2016). National BIM Guide v1.0 Sep 2011.doc, (September 2011).

NATSPEC BIM Object Element Matrix, (2008).

Nawari, N. (2012). The challenge of computerizing building codes in BIM environment. Proceedings of the 2012 Asce International Conference on Computing in Civil Engineering.

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

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. Automation in Construction, 20(5), 506–518. http://doi.org/10.1016/j.autcon.2010.11.017

Richard Gronback. (n.d.). EMF. Retrieved February 14, 2017, from http://www.eclipse.org/modeling/emf/

Sagarkar, V. (n.d.). Traditional Approach vs BIM Approach. Retrieved January 13, 2017, from https://www.linkedin.com/pulse/traditional-approach-vs-bim-varunkumar-sagarkar-igbc-ap

See, R., Karlshoej, J., & Davis, D. (2012). An Integrated Process for Delivering IFC Based Data Exchange, (1), 53. Retrieved from http://iug.buildingsmart.org/idms/

Solihin, W., & Eastman, C. (2015). A Knowledge Representation Approach to Capturing BIM Based Rule Checking Requirements Using Conceptual Graph. Proc. of the 32nd CIB W78 Conference 2015, 27th-29th October 2015, Eindhoven, The Netherlands.

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

Solihin, W., Eastman, C., & Lee, Y.-C. (2015). Toward robust and quantifiable automated IFC quality validation. Advanced Engineering Informatics, 29(3), 739–756. http://doi.org/10.1016/j.aei.2015.07.006

Stangeland, B. (2011). BIM Collaboration Format. buildingSMART, 1, 1–3.

Statsbygg. (2013). Statsbygg BIM Manual Version 1.2.1. Sbm, 1, 1–98. Retrieved from http://www.statsbygg.no/Files/publikasjoner/manualer/StatsbyggBIM-manual-ver1-2-1eng-2013-12-17.pdf

Tamke, M., Jensen, M. M., Beetz, J., Krijnen, T., & Fjeld, D. (n.d.). Building Information Deduced, 141, 1–11.

van Rillaer, D., Burger, J., Ploegmakers, R., & Mitossi, V. (2012). Kingdom Relations Rgd BIM Standard Colophon, (July), 1–29.

Venugopal, M., Eastman, C. M., Sacks, R., & Teizer, J. (2012). Advanced Engineering Informatics Semantics of model views for information exchanges using the industry foundation class schema. Advanced Engineering Informatics, 26(2), 411–428. http://doi.org/10.1016/j.aei.2012.01.005

W3C. (2011). What is the Semantic Web. Semantic Web for the Working Ontologist, 1. http://doi.org/10.1016/B978-0-12-385965-5.10001-9

Xiao, H., & Noble, T. (2016). BIM's impact on the project manager. Procs 30th Annual ARCOM Conference, (1–3 September 2014), 693–702.

Zhang, C., Beetz, J., & Weise, M. (2015). INTEROPERABLE VALIDATION FOR IFC BUILDING MODELS USING OPEN STANDARDS Interoperability issues of the Industry Foundation Classes, 20(November 2014), 24–39.

Zhang, C., & Weise, M. (2012). Model view checking : automated validation for IFC building models.

Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M. (2013). Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. Automation in Construction, 29, 183–195. http://doi.org/10.1016/j.autcon.2012.05.006

Appendix

Content

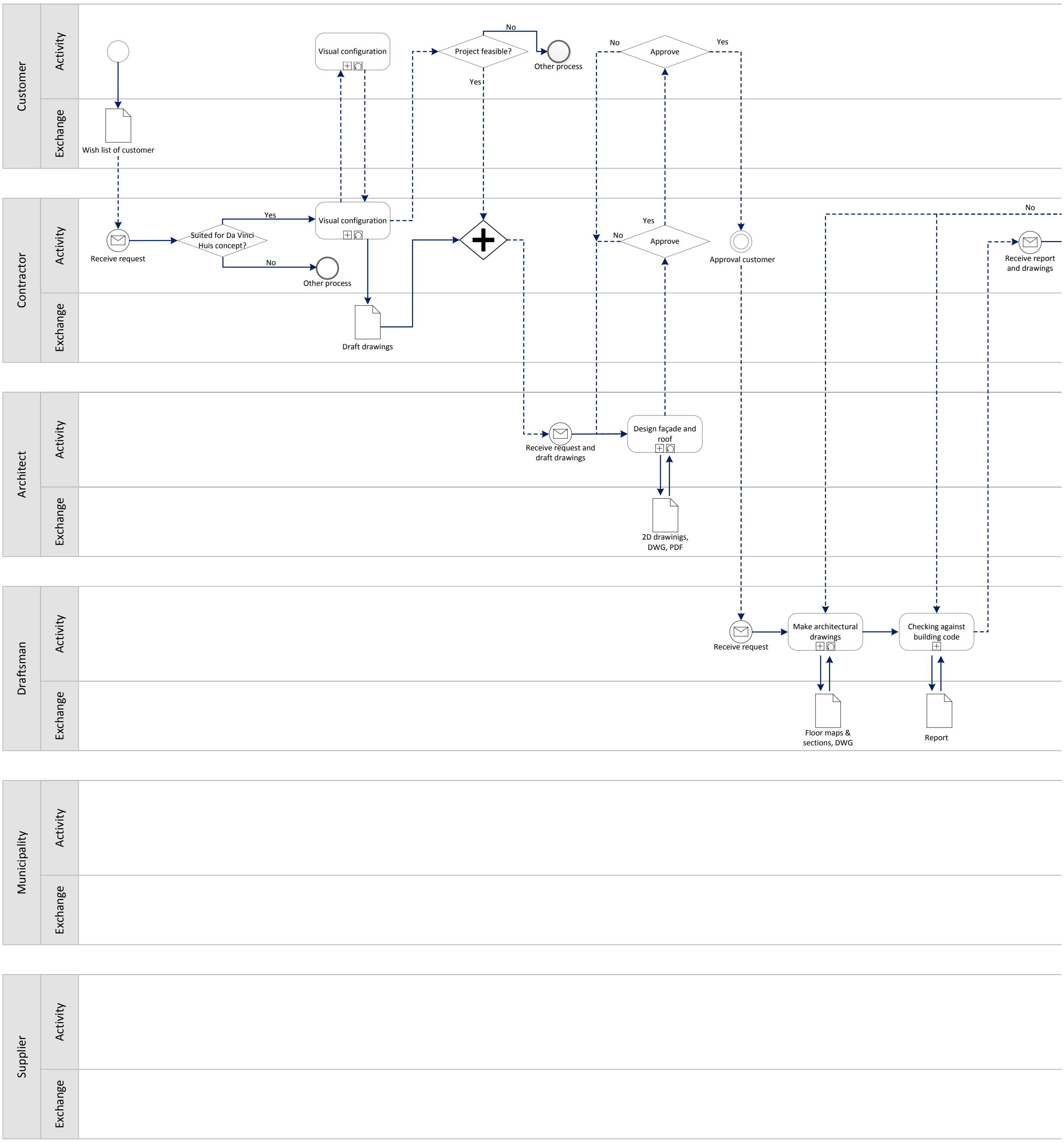
List of figures and tables

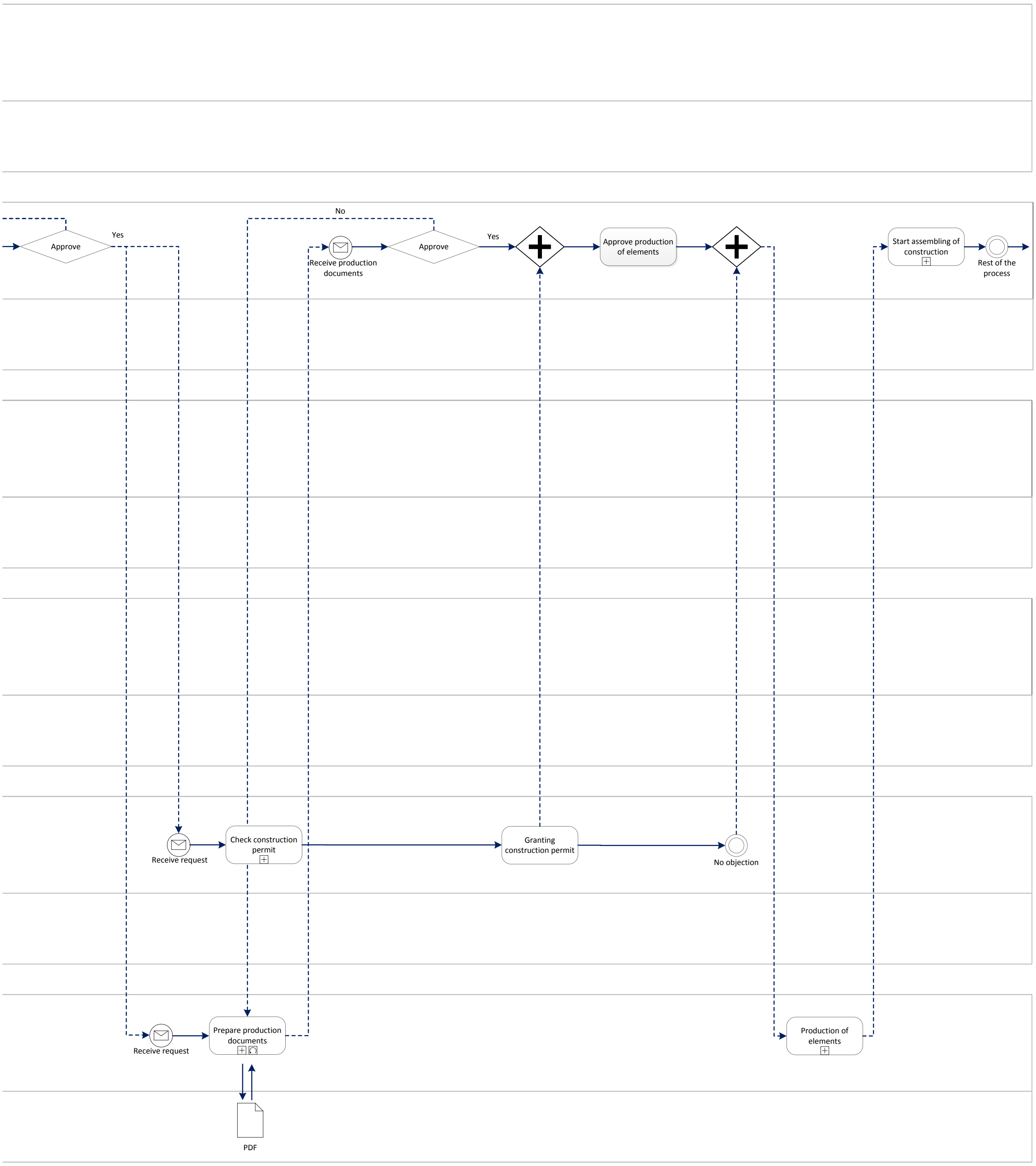
Appendix 4

Appendix 5

Appendix 1

Current process of the Da Vinci Huis in BPMN

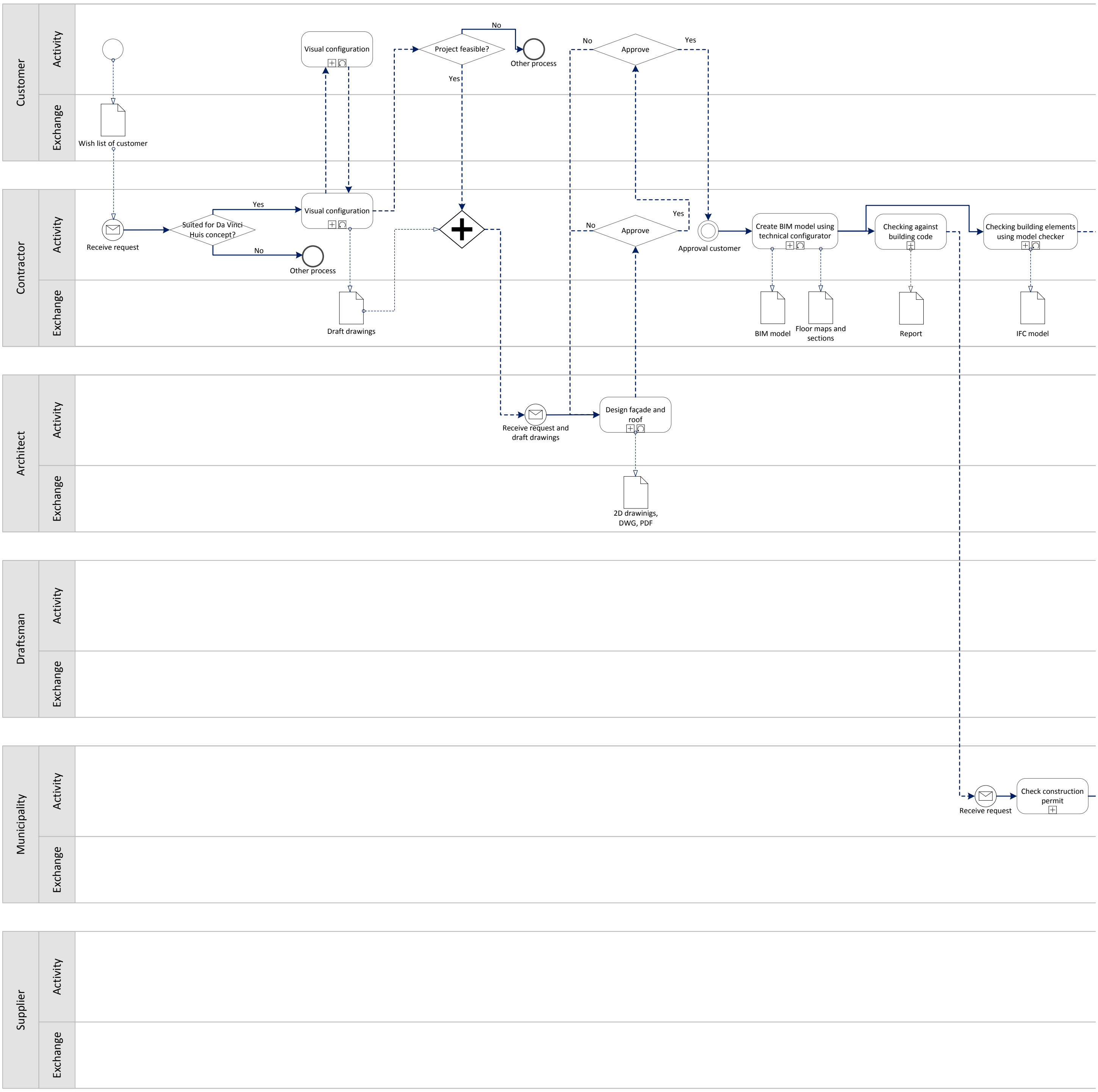


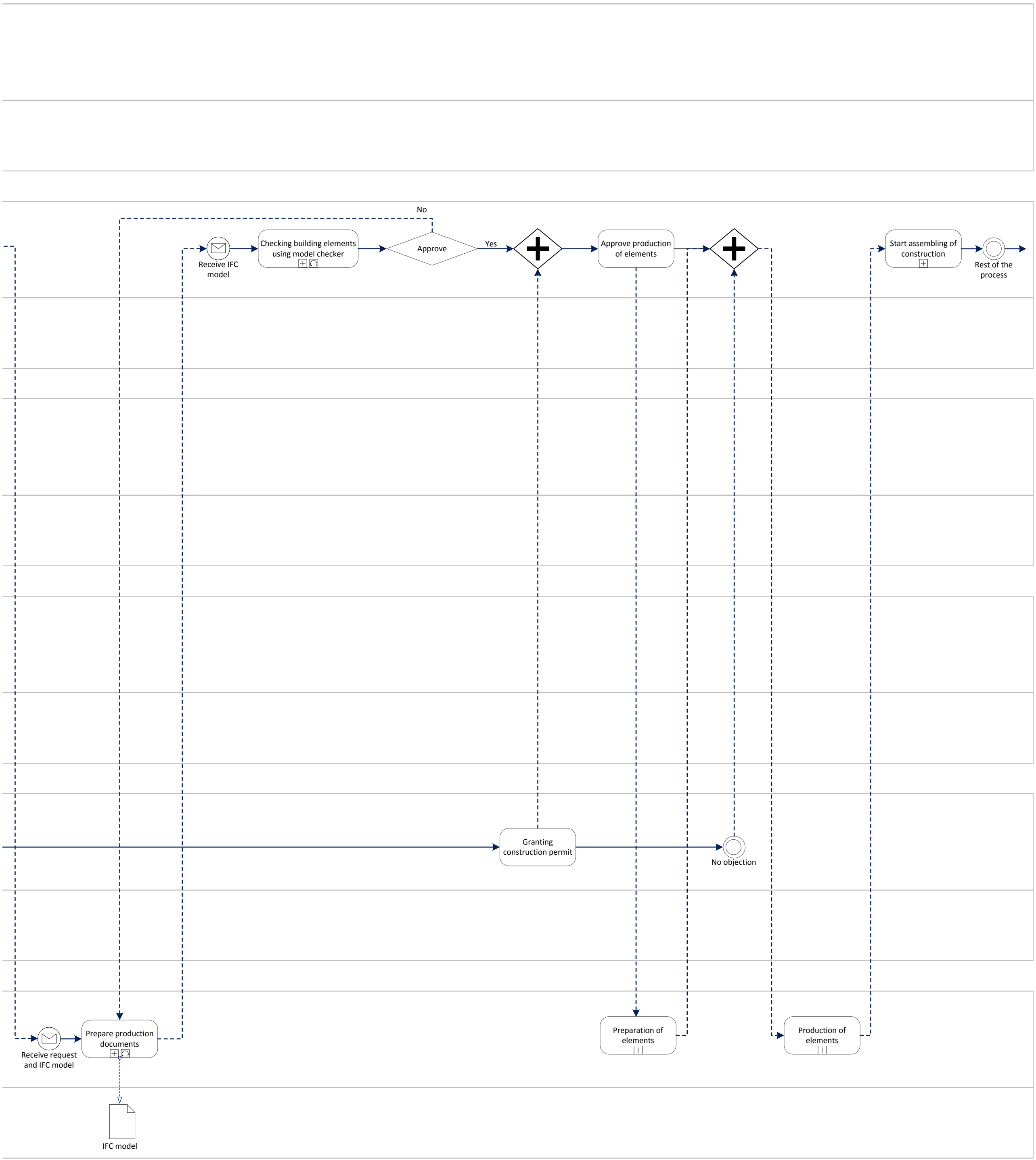


No
Yes
Approve
Receive production documents
Approve
Yes
Approve production of elements
Start assembling of construction
Rest of the process
Receive request
Check construction permit
Granting construction permit
No objection
Receive request
Prepare production documents
PDF
Production of elements

Appendix 2

Ideal process of the Da Vinci Huis in BPMN





No
Receive IFC model
Checking building elements using model checker
Approve
Yes
Approve production of elements
Start assembling of construction
Rest of the process
Granting construction permit
No objection
Receive request and IFC model
Prepare production documents
Preparation of elements
Production of elements
IFC model

Appendix 3

Overview of required parameters

	Contractor	*Supplier*	*Data*	*Rule Class*
General Project Information				
Project name			x	1
Description			x	1
Organization name			x	1
Author			x	1
date drawn			x	1
date final			x	1
Project Number			x	1
position			x	1
block			x	1
site			x	1
Wall				
Physical properties				
Length	x	x	x	1
Length < 11.40 meters		x		2
Width	x	x	x	1
Height	x	x	x	1
Height < 4.00 meters		x		2
Area			x	1
Volume			x	1
Weight	x	x	x	1
Weight < 15.000 kilograms	x	x		2
Type of concrete	x	x	x	1
Density of concrete			x	1
Environmental classification			x	1
Wall ties			x	1
Wall number			x	1
NL/SfB	x			1

Appendix 4

Detailed explanation concerning the concepts and relations of mvdXML

Introduction

This document is created as a starters guide for anyone who gets involved with the mvdXML Generator and Checker. The mvdXML Generator and Checker is an open model view checker based on open standards to validate IFC building models. This checker is difficult to use and requires some knowledge about mvdXML, IFC and the IFC Documentation Generator. In this document connections are shown between various important concepts of mvdXML and IFC, in order to get a better understanding of the mvdXML Generator and Checker and IfcDoc.

MvdXML is developed to check rules on IFC files, this means that in mvdXML the rules are described and it contains some of the structure of IFC in order to be able to check the rules on IFC. MvdXML is based on the markup language XML and is adjusted in order to be checkable on IFC. Since it is a language, one should first learn the basics of the language before being able to apply it.

An example of a detailed description of mvdXML is (Chipman et al., 2016). This is a specification of a standardized format to define and exchange Model View Definitions with Exchange Requirements and Validation Rules. Though it is a very complete description, it is difficult to understand since it is written in a very descriptive way. Connections between the different subjects are unclear and background knowledge is required. For future use and development it is important to have a clear description of mvdXML, IFC and IfcDoc. Since it will become easier for an unexperienced user to use the mvdXML checker and the development of the mvdXML checker will become more accessible.

1. XML (eXtensible Markup Language)

EXtensible Markup Language (XML) describes a class of data objects called XML documents and describes the behavior of computer programs which process them (W3C, 2014).
The language is developed by the World Wide Web Consortium. The eXtensible Markup Language has been designed to store and transport data. At the process of rule checking the conflict which occurs is about how the rules written in human language can be interpreted by machine processes without loss of data. Since XML is both readable for machines and humans, it is a suitable solution for the stated problem.

XML structure

XML documents should, begin with a XML declaration which specifies the used version of XML. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The content of the XML is described by rules which form a tree structure. It starts with an attribute (note, in the example below), is further described by entities (in this example; to, from, heading and body) followed up by the information in human language.

<note>
<to>Henk</to>
<from>Piet</from>
<heading>Reminder</heading>
<body>Don not forget our meeting!</body>
</note>

In the example one can see that XML is using a structure. Each line starts by using a ‘<’, followed by text and closing with the use of ‘>’. If the rule should be closed, a / is added after the <. This is the way all rules should be described, if deviated from this, an error will appear when it is launched in a web browser.

2. IFC structure & IfcDoc

This subchapter will shortly explain the structure of an IFC and how it is related to the IfcDoc. The data schema architecture of IFC defines four conceptual layers, each individual schema is assigned to exactly one conceptual layer. As can be seen in Figure 2.1 the structure of the IFC can be exactly found in IfcDoc. Even though the names of the layers in IFC differ from the data schemas in IfcDoc, the structure is equal. Since Figure 2.1 provides a clear understanding of the connection between IFC and IfcDoc more information concerning the content of the layers/data schemas is provided.

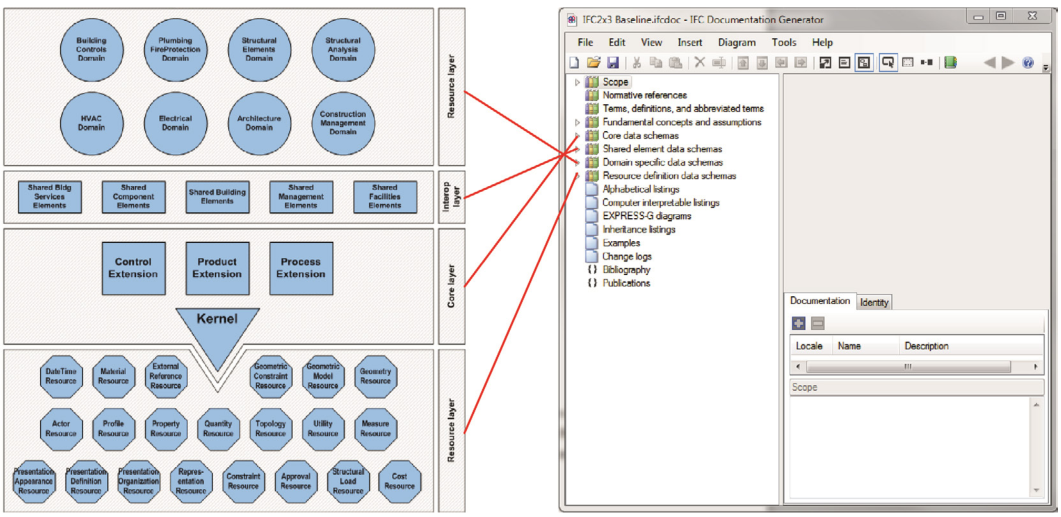


Figure 2.1 Link between IFC and IfcDoc

1. Resource layer/Resource definition data schemas

Resource classes are used by classes in the higher levels. An example of a resource class is the cost schema, all information concerning the costs is collected within this cost schema (IfcCostResource). Entities and types defined in this layer can be referenced by all entities in the layers below.

2. Core layer/Core data schemas

The core layer consists of the kernel schema and the core extension schemas, containing the most general entity definitions. The Core layer provides the basic structure of the IFC object model and defines the most abstract concepts. These concepts are further defined by higher layers of the IFC object model. The Kernel provides all the basic concepts required for IFC models, it also determines the model structure and decomposition. All entities defined in the core layer contain a unique identification, name, description, and change control information.

3. Interoperability layer/shared element data schemas
The shared element data schemas contain entity definitions which are specific to a general product, process or resource specialization which is used across several disciplines, those definitions are typically utilized for inter-domain exchange and sharing of construction information.

4. Domain layer/domain specific data schemas
The highest layer include schemas containing entity definitions that are final specializations of products, processes or resources specific to a certain discipline. Entities defined in this layer cannot be referenced by any other layer (buildingSMART International Limited, n.d.).

3. mvdXML

This chapter is about the explanation of mvdXML. Wherever it is possible, a connection between mvdXML, IfcDoc and IFC will be made. The explanation is conducted based on the following subjects:

- Model view & Exchange requirement
- Concept template
  - Entity and attribute
  - Constraint
- Concept
  - Concept root
- IfcObject

For each subject at first a definition is given. An additional explanation is given if this is considered necessary. Followed up by showing the subject in the context of a mvdXML and IfcDoc.

3.1 Model view and Exchange requirement

Definitions

Model view:	Subset of a schema, representing the data structure required to fulfil the data requirements within one or several exchange scenarios.
Exchange requirement:	An exchange requirement documents the information needed between two or more parties to be exchanged in support of a particular business requirement at a particular stage of a project.

A Model View Definition can be applied to validate if the provided data conforms to the Exchange Requirements.

Explanation

Situation:	You won the lottery and you want to buy a car.
Demands:	The car must be red and must be a Ferrari.
Model view:	Ferrari dealer
Exchange requirement:	Red car

In this case your model view is the Ferrari dealer, it is possible to choose between various dealers, but you limit yourself to the Ferrari dealer. When entering the Ferrari dealer, the exchange requirements is a red car, since that is needed in order to satisfy the demands.

The location of the Exchange Requirements and Model view in a mvdXML file is located in Figure 3.1-1.

Figure 3.1-1 Exchange Requirements and Model View located in mvdXML

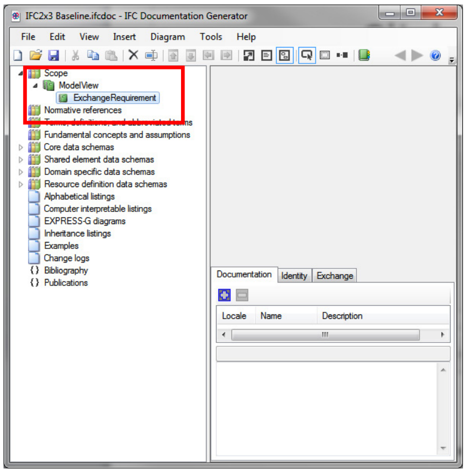


Figure 3.1-2 Exchange Requirements and Model View located in IFC Documentation Generator

As can be seen in Figure 3.1-2 at first a ModelView is inserted in Scope in the IfcDoc. At least one ModelView is required, though it is possible to add more ModelViews. Next, a ExchangeRequirement is inserted in ModelView. Like a ModelView, at least one ExchangRequirement is required but it is possible to add more ExchangeRequirements to the model.

3.2 Concept and concept template

In this part at first concept and concept template together and the connection between the two subjects are defined. After that it is split up and started with the explanation of ‘concept template’, followed up by ‘concept’.

Definitions

Concept:	Rules on using a subset of the schema structure identified as a concept template to enable a certain functionality within the context of a concept root contained in a model view.
Concept template:	The specification of a subset of the schema structure to enable a certain unit of functionality.

The Concept Template defines the structure related Concepts should comply to.

Explanation

Concept:	Lemonade Applicable for one time
Concept template:	Glass Reusable

Both need each other, lemonade has to fit into the structure of the glass

3.2.1 Concept template

In the layer ‘Fundamental concepts and assumptions’ of the IfcDoc a ConceptTemplate can be inserted. Later on, in any entity of IFC will be referred to the ConceptTemplate. Using the Template button in IfcDoc the structure of the template can be constructed. The structure of the template is based on the structure as defined in IFC4 (buildingSMART International Limited, n.d.), and is built up out of entities and attributes.In figure 3.2.1-1 the Concept template located in mvdXML is displayed and in Figure 3.2.1-2 the Concept template, located in IFC Documentation Generator is visualized.

```xml
<ConceptTemplate applicableEntity="IfcObject" applicableSchema="IFC4" name="" uuid="46fba748-b6c7-48a3-b81a-af259c83aaa4">
  <Rules>
    <AttributeRule AttributeName="IsDefinedBy" Cardinality="_asSchema">
      <EntityRules>
        <EntityRule Cardinality="_asSchema" EntityName="IfcRelDefinesByProperties">
          <AttributeRules>
            <AttributeRule AttributeName="RelatingPropertyDefinition" Cardinality="_asSchema">
              <EntityRules>
                <EntityRule Cardinality="_asSchema" EntityName="IfcElementQuantity">
                  <AttributeRules>
                    <AttributeRule AttributeName="Quantities" Cardinality="_asSchema">
                      <EntityRules>
                        <EntityRule Cardinality="_asSchema" EntityName="IfcQuantityArea">
                          <AttributeRules>
                            <AttributeRule AttributeName="Name" Cardinality="_asSchema" RuleID="IfcQuantityAreaName"/>
                          </AttributeRules>
                        </EntityRule>
                      </EntityRules>
                    </AttributeRule>
                  </AttributeRules>
                </EntityRule>
              </EntityRules>
            </AttributeRule>
          </AttributeRules>
        </EntityRule>
      </EntityRules>
    </AttributeRule>
  </Rules>
</ConceptTemplate>
```

Figure 3.2.1-1 Concept template, located in mvdXML

In the layer ‘Fundamental concepts and assumptions’ of the IfcDoc a ConceptTemplate can be inserted. Later on, in any entity of IFC will be referred to the ConceptTemplate. Using the Template button in IfcDoc the structure of the template can be constructed. The structure of the template is based on the structure as defined in IFC4 (buildingSMART International Limited, n.d.), and is built up out of entities and attributes.

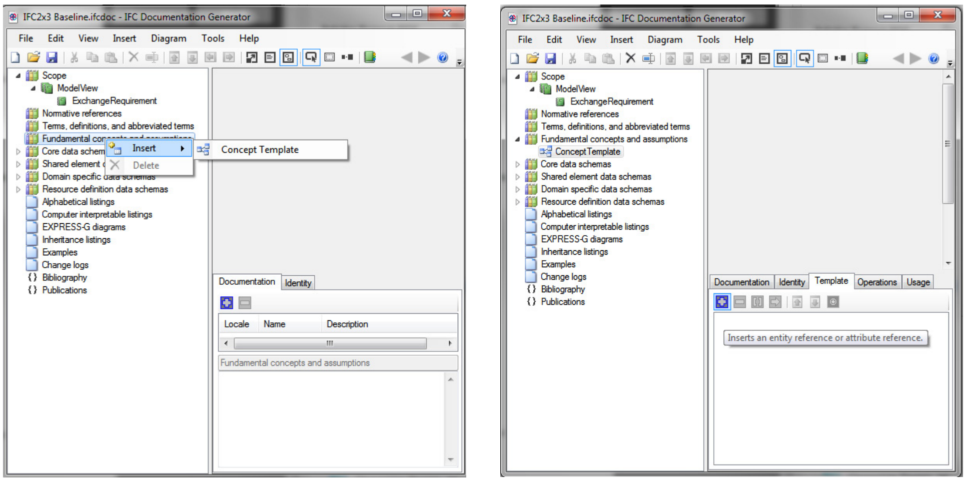


Figure 3.2.1-2 Concept template, located in IFC Documentation Generator

Definitions

Entity:	Class of information defined by common attributes and constraints.
Attribute:	Unit of information within an entity, defined by a particular type or reference to a particular entity.
EntityRule:	Represents the specification of an entity (or value type) referenced by an attribute, either as a scalar reference or a reference from within a collection.
AttributeRule:	Represents the specification of an attribute on an entity, with related constraints, and/or entity rules.
Constraint:	Applicable in EntityRule and AttributeRule, represents restriction on an attribute.

Explanation

If entity is a book, title, price and ISBN would be attributes of the book. The information that describes the main object.

3.2.2 Concept

All the information of IFC is divided over four layers, which can also be found in IfcDoc. These four layers are built up out of information units, which consist of, amongst other thing, types, entities, global rules, functions, property sets and more. In the entities a Model View can be assigned, next a concept can be inserted and described. Besides, requirements can be added in the Concept. In figure 3.2.2-1 the Concept located in mvdXML is displayed and in Figure 3.2.2-2 the Concept, located in IFC Documentation Generator is visualized.

```
<ConceptRoot applicableRootEntity="IfcWindow" name="" uuid="6c03fe12-98d2-4329-b4a7-9147d5892568">
  <Concepts>
    <Concept name="" override="false" uuid="6e6498d9-93c9-496d-a7a2-f9b6b6b58e92">
      <Definitions>
        <Definition>
          <Body/>
        </Definition>
      </Definitions>
      <Template ref="46fba748-b6c7-48a3-b81a-af259c83aaa4"/>
      <Requirements>
        <Requirement/>
      </Requirements>
      <Rules>
        <TemplateRule Parameters="IfcQuantityAreaName[Value]='Area'"/>
      </Rules>
    </Concept>
  </Concepts>
</ConceptRoot>
```

Figure 3.2.2-1 Concept, located in mvdXML

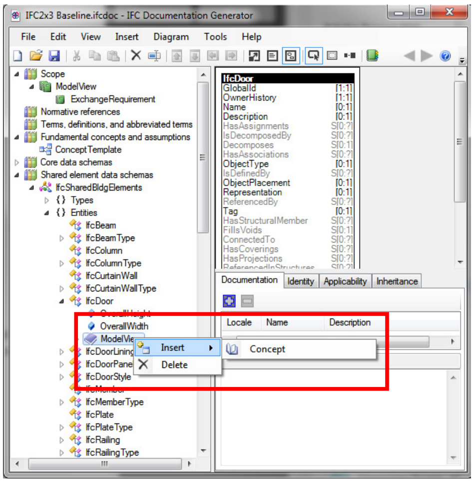


Figure 3.2.2-2 Concept, located in IFC Documentation Generator

All the information of IFC is divided over four layers, which can also be found in IfcDoc. These four layers are built up out of information units, which consist of, amongst other thing, types, entities, global rules, functions, property sets and more. In the entities a Model View can be assigned, next a concept can be inserted and described. Besides, requirements can be added in the Concept.

Definition

Concept root: An entity of a schema, used to assign concepts to describe the required functionality.

3.3 IfcObject

Definitions

IfcObject: Main components of the raw building (or carcass)

IfcObject consists of:

IfcWindow	IfcSlab
IfcDoor	IfcRoof
IfcWall	IfcStair
IfcBeam	IfcRamp
IfcColumn	IfcCovering

These can be found in the layers of the IFC model.

IfcObject, defined in concept template, as showed in Figure 3.3-1, referenced in concept as visualized in Figure 3.3-2.

Figure 3.3.1. IfcObject, defined in concept template

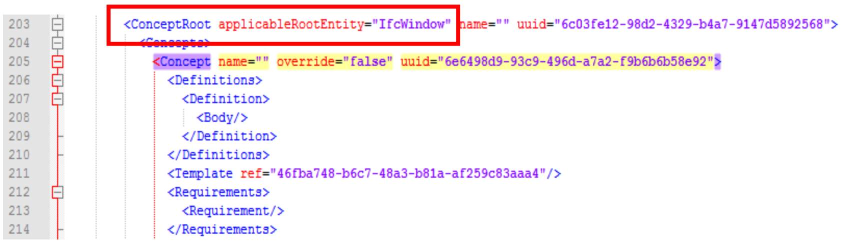


Figure 3.3.2. IfcObject, referenced in concept

4. IfcDoc & IFC & mvdXML

Table 4.1 shows an overview of links between all the information classes as were named in this sub-chapter.

Table 4.1 Overview IfcDoc, IFC and mvdXML

In IfcDoc	In IFC	In mvdXML
Scope	x	Model view and Exchange Requirement
Normative references	x	x
Terms, definitions, and abbreviated terms	x	x
Fundamental concepts and assumptions	x	Concept template
Core data schemas	Core layer	Concept & IfcObject
Shared element data schemas	Interop layer	
Domain Specific Data schemas	Resource layer (up)	
Resource Definition data schemas	Resource layer (low)	

Appendix 5

User manual of Excel checker

Introduction

Aim of this user manual is to provide insight concerning how to use the Microsoft Excel checker. This explanation is done by using the case study of the Da Vinci Huis.

The structure of this manual is according to the automatic rule checking process of (C Eastman et al., 2009), which is referred to several times in the graduation thesis. However, a different order is maintained. Each rule checking process block represents a chapter in this manual. In Figure 1 the structure of this guide is visualized.

1. Building model preparation	**2. Rule interpretation**	**3. Rule execution**	**4. Reporting Checking results**
Extracts and derives model view data for checking	Translates a written rulebase into computer implementable one	Applies rules to building model	Reporting results back to submitter

Figure 1: Structure of the chapters of the Excel checker manual

In this instruction guide several files are used for the checking of a building model. All files which are used in this manual can be found via following link: *https://www.dropbox.com/home/Files graduation project A van Dun*

1. Building model preparation

This phase is about the conversion of an Autodesk Revit to IFC file. This is a very important step, since all important information should maintained during the file conversion. At first the building model needs to be prepared. This means, necessary information needs to be added to the model before it can be considered complete. Next, the file needs to be exported.

1.1 Add manually parameters to Autodesk Revit file

Since Autodesk Revit does not offer all parameters required to consider a building model complete, sometimes it is required to add these manually conducting following steps:

1. Select Project Parameters under the tab Manage, Settings

2. Click Add...

3. Select Shared parameter (Figure 1.1-1)

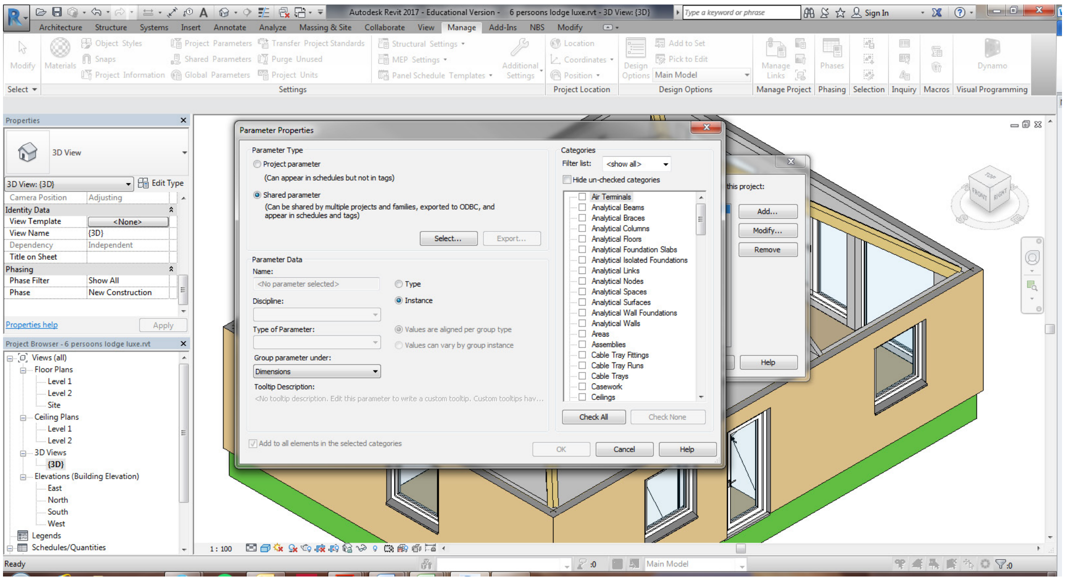


Figure 1.1-1 Screenshot of select shared parameter

4. Browse to the file Parameters Revit.txt and select

5. Add each parameter group to corresponding category

   For example: Block number, Category Project Information, Group Parameter under: Other (Figure 1.1-2)

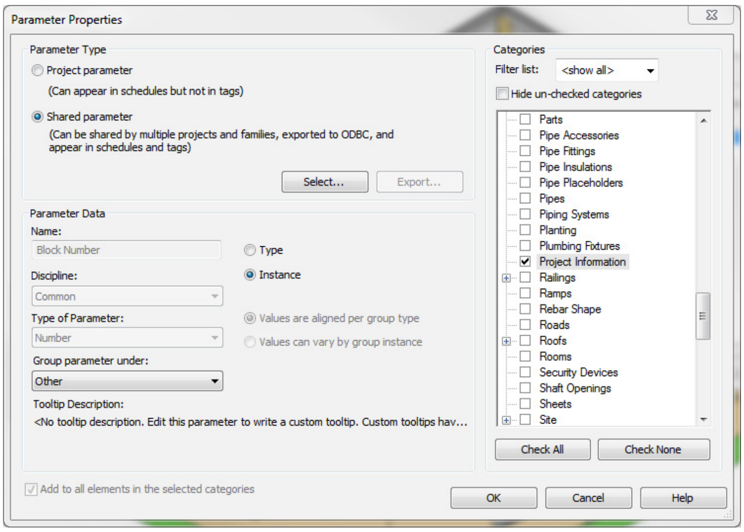


Figure 1.1-2 Screenshot of add parameter group to corresponding category

1.2 Add manually Assembly code to Autodesk Revit file

A requirement for a complete model is the presence of the NI/SfB code. This code belongs to the Assembly code in Autodesk Revit. The file containing all information of NI/SfB applicable for Revit, is named 160209_NLRSv2.5.2_NL-SfB classificatiecode variantelementen BNA versie 1991_v1.txt and can be found in the Dropbox folder.

1. Select Assembly Code at the tab Manage, Settings, Additional Settigs

2. Select Relative

3. Browse and load 60209_NLRSv2.5.2_NL-SfB classificatiecode variantelementen BNA versie 1991_v1.txt

4. Assign code to all related paramets

1.3 Export Autodesk Revit to IFC file

1. Select Export as IFC in home tab

2. Adjust destination file path

3. Modify Setup

4. Choose at Property Sets (Figure 1.3-1)
   Export Revit property sets
   Export IFC common property sets

5. Set Level of Detail, high

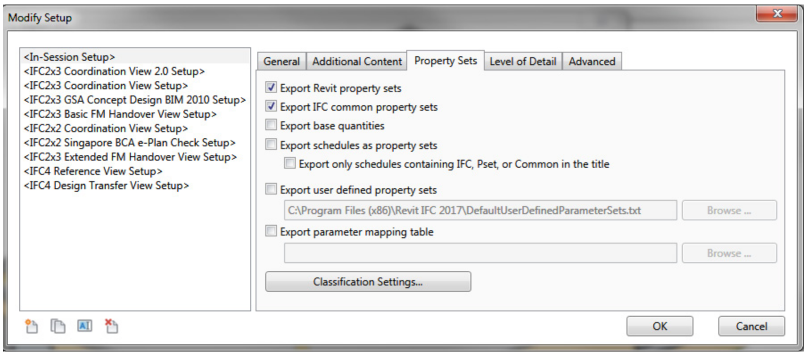


Figure 1.3-1 Options Property Sets

1.4 Export IFC to Excel file

Download IFC File Analyzer at https://www.nist.gov/services-resources/software/ifc-file-analyzer
Or open IFC-File-Analyzer.exe at the Dropbox.

When the IFC file analyzer is opened:

1. Select options in the menu

2. Select in the process section (Figure 1.4-1):
   Building elements
   Property
   Relationship
   Other
   Include GUID

3. Select in the Expand section:

IfcLocalPlacement

IfcAxis2Placement

4. In spreadsheet tab select

Tables: Generate Tables for Setting and FIltering

5. Load IFC model in software tool and generate spreadsheet

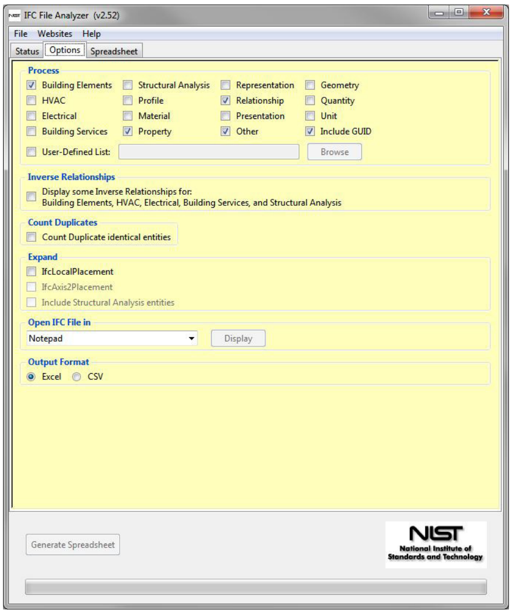


Figure 1.4-1 Screenshot of options of IFC file analyzer

2. Rule interpretation

The rule interpretation phase is about establishing the rules which are going to be applied on elements. In this part a distinction is made between the file Exel_Checker and Excel_IFC. Two types of checks are explained, the checking on parameters and a comparison of geometries of two models.

2.1 Check on parameters

1. Open the file Excel_Checker.xlsx in the Dropbox folder. Make sure the Excel_IFC file is open as well, if this file is not open, the file paths in the Excel_Checker required for making references does not work. In figure 2.1-1 the main sheet is shown.

2. First step is to fill in the name of Excel_IFC in the left upper corner, make sure to spell the name right, otherwise the checker does not work.

3. Next is to decide the type of element the rules are applied on, the possible options are collected in a drop-down list.

4. After that, one can calculate the workbook, this leads to a drop-down list of available parameters for the element

5. Next, fill in the parameters of which the presence needs to be checked on the element, using the drop-down list which has been calculated in previous step.

6. The next step is optional; fill in the constraints and combined value, again by using a drop down list, and the value of the constraint.

7. Calculate the workbook

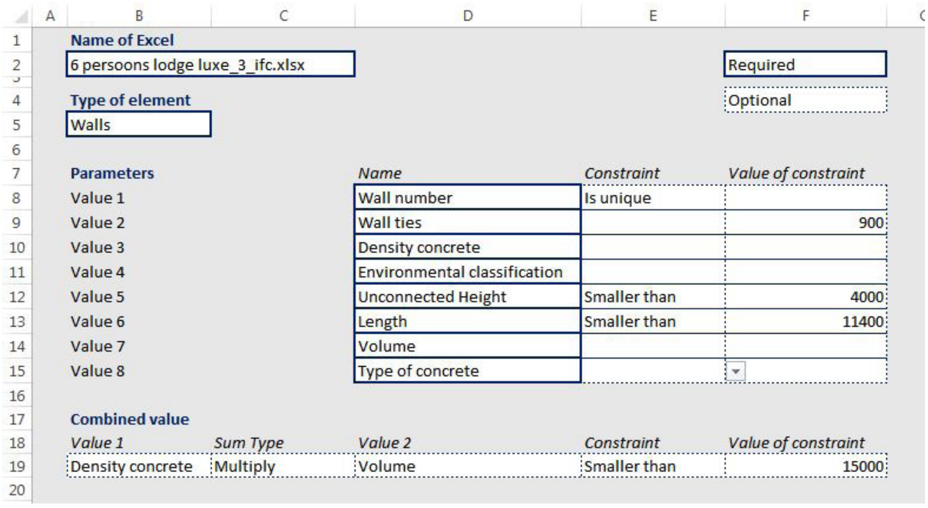


Figure 2.1-1 Screenshot of main sheet

2.2 Compare geometries

In order to conduct this check, two Excel_IFC files are required which are originally from the same model

1. Open the file Compare Geometries.xlsx (Figure 2.2-1)

2. Fill in the names of the Excel_IFC files

3. Again, decide on the element which needs to be checked

4. Calculate the Workbook

	B	C	D	E	F
1		**Name of Excel 1**		**Name of Excel 2**	
2		6 persoons lodge luxe_3_ifc.xlsx		6 persoons lodge luxe_3_2_ifc.xlsx	
3					
4		**Element**			
5		Walls			
6					
7		**GUID of wall elements not the same**			
8		20ByCA$dXEFeTAlGKcGWqP			
9					
10					
11					
12					
13					
14					
15					
16					
17					
18		**GUID**	**Volume**	**Volume**	**Is the same?**
19		20ByCA$dXEFeTAIGKcGWos	0.0377039999999898	0.0377039999999898	TRUE
20		20ByCA$dXEFeTAIGKcGWou	0.504239999999997	0.504239999999997	TRUE
21		20ByCA$dXEFeTAIGKcGWo$	0.381840000000001	0.381840000000001	TRUE
22		20ByCA$dXEFeTAIGKcGWo_	0.198318111133256	0.198318111133256	TRUE
23		20ByCA$dXEFeTAIGKcGWoz	2.35449999999998	2.35449999999998	TRUE
24		20ByCA$dXEFeTAIGKcGWr3	1.11496018970329	1.11496018970329	TRUE
25		20ByCA$dXEFeTAIGKcGWr1	0.826253810296682	0.826253810296682	TRUE
26		20ByCA$dXEFeTAIGKcGWrN	1.56404700000072	1.56404700000072	TRUE
27		20ByCA$dXEFeTAIGKcGWrL	1.29575550000002	1.29575550000002	TRUE
28		20ByCA$dXEFeTAIGKcGWrR	1.8804638098043	1.8804638098043	TRUE
29		20ByCA$dXEFeTAIGKcGWrO	0.522217499999999	0.522217499999999	TRUE
30		20ByCA$dXEFeTAIGKcGWrV	0.654272499999999	0.654272499999999	TRUE

Figure 2.2-1 Screenshot of compare geometries

	A	B	C	D	E	F	G
6							
7		**Parameters**		*Name*	*Constraint*	*Value of constraint*	
8		Value 1		Wall number	Is unique		
9		Value 2		Wall ties		900	
10		Value 3		Density concrete			
11		Value 4		Environmental classification			
12		Value 5		Unconnected Height	Smaller than	4000	
13		Value 6		Length	Smaller than	11400	
14		Value 7		Volume			
15		Value 8		Type of concrete			
16							
17		**Combined value**					
18		*Value 1*	*Sum Type*	*Value 2*	*Constraint*	*Value of constraint*	
19		Density concrete	Multiply	Volume	Smaller than	15000	
20							
21		**General Project Information**					
22		Project name	Beekse Bergen vakantiewoningen				
23		Project Description	6 persoons lodge luxe				
24		Organization name	Hurks Bouw Eindhoven				
25		Author	R. Verweij				
26		Date drawn	42747				
27		Date final	Not present				
28		Project Number	76670				
29		Block Number	1				
30		Site	Beekse Bergen 1, 5081 NJ Hilvarenbeek				
31							
32		**ObjectType**		**Overall**			
33		Basic Wall:EPS 80 mm		Not complete			
34		Basic Wall:Prefab betonwand 100 mm		Not complete			
35		Basic Wall:Prefab betonwand 90 mm		Not complete			
36		Basic Wall:Prefab betonwand 150 mm		Not complete			
37		Basic Wall:Ytong cellenbeton G4/600 70		Not complete			
38		Basic Wall:Ytong cellenbeton G5/800 10		Not complete			
39		Basic Wall:EPS Isolatie + stucwerk 150 +		Not complete			

Figure 4.1-1 Screenshot of the most general overview

3. Rule execution

This phase is not important for the user manual, more information about the background ot the rule execution can be found in the thesis in chapter 4.

4. Report checking results

4.1 Check on parameters

After the rule execution the outcome in the Excel_Checker is displayed on three different levels.

Sheet '1. Overview' offers the most general overview, Figure 4.1-1. In this overview is displayed per element name whether any value is missing or if all information per element name is complete. Directly after conducting the check, one can see at the same page if all information is complete or incomplete.

Since one does not always want to know all information the next level shows a more detailed overview. In the sheet '2 Overview per Parameter', are per element name the outcome of the parameters categorized. It is stated whether all parameters are complete or uncomplete, combined with the number of the amount of missing values per parameter per element name. A screenshot of this level is shown in Figure 4.1-2.

Overview per parameter

ObjectType	*Total*	*Overall*	*Wall number*	*Wall ties*	*Density concrete*	*Environmental class*	*Unconnected Height*	*Length*
Basic Wall:EPS 80 mm	12	*Not complete*	Not complete (12)	Not complete (12)	Not complete (12)	Not complete (12)	Complete	Complete
Basic Wall:Prefab betonwand 100 mm	3	*Not complete*	Not complete (3)	Not complete (3)	Not complete (3)	Not complete (3)	Complete	Complete
Basic Wall:Prefab betonwand 90 mm	2	*Not complete*	Not complete (2)	Not complete (1)	Not complete (1)	Not complete (1)	Complete	Complete
Basic Wall:Prefab betonwand 150 mm	1	*Not complete*	Not complete (1)	Not complete (1)	Not complete (1)	Not complete (1)	Complete	Complete
Basic Wall:Ytong cellenbeton G4/600 70 mm	13	*Not complete*	Not complete (13)	Not complete (12)	Not complete (13)	Not complete (12)	Complete	Complete
Basic Wall:Ytong cellenbeton G5/800 100 mm	5	*Not complete*	Not complete (5)	Not complete (5)	Not complete (5)	Not complete (5)	Complete	Complete
Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm	4	*Not complete*	Not complete (4)	Not complete (4)	Complete	Complete	Complete	Complete

Figure 4.1-2 screenshot of overview per parameter

At the sheet '3 All element information', the information is displayed with the highest level of detail. All information regarding the individual elements and the matching outcome of the values and constraints are shown in this sheet. A screenshot of this level is shown in Figure 4.1-3.

4.2 Compare geometries

Directly after comparing the geomertries of two files, the GUID of the elements of which the volumes differ appear in the main sheet.

	B	C	D	F	G	I	J	L	M	O	P	R	S	U	
1	**All wal information**														
7															
8			*Value 1*		*Value 2*		*Value 3*		*Value 4*		*Value 5*		*Value 6*		*Va*
9	*GUID*	*Element Name*	*Wall numbe*	*Const.*	*Wall ties*	*Const.*	*Density conc*		*Environment*	*Const.*	*Unconnectec*	*Const.*	*Length*	*Const.*	*Vo*
10	20ByCA$dXEFeTAIGKcGWos	Basic Wall:EPS 80 mm		TRUE							600	TRUE	745.49999999	TRUE	0
11	20ByCA$dXEFeTAIGKcGWou	Basic Wall:EPS 80 mm		TRUE							600	TRUE	10424.999999	TRUE	0
12	20ByCA$dXEFeTAIGKcGWo$	Basic Wall:EPS 80 mm		TRUE							600	TRUE	7955.0000000	TRUE	0
13	20ByCA$dXEFeTAIGKcGWo_	Basic Wall:EPS 80 mm		TRUE							600	TRUE	4171.6273152	TRUE	0
14	20ByCA$dXEFeTAIGKcGWoz	Basic Wall:Prefab betonwand 100 mm		TRUE							2620.0000000	TRUE	10104.999999	TRUE	2
15	20ByCA$dXEFeTAIGKcGWr3	Basic Wall:Prefab betonwand 100 mm		TRUE							2620.0000000	TRUE	4990.3726847	TRUE	1
16	20ByCA$dXEFeTAIGKcGWr1	Basic Wall:Prefab betonwand 100 mm		TRUE							2420.0000000	TRUE	4086.6273152	TRUE	0
17	20ByCA$dXEFeTAIGKcGWrN	Basic Wall:Prefab betonwand 90 mm		TRUE							2620.0000000	TRUE	7875.0000000	TRUE	1
18	20ByCA$dXEFeTAIGKcGWrL	Basic Wall:Prefab betonwand 150 mm		TRUE							2800.0000000	TRUE	7875.0000000	TRUE	1
19	20ByCA$dXEFeTAIGKcGWrR	Basic Wall:Prefab betonwand 90 mm		TRUE	No		60		D2/G6		5377	FALSE	7774.9999999	TRUE	1
20	20ByCA$dXEFeTAIGKcGWrO	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	3850	TRUE	0
21	20ByCA$dXEFeTAIGKcGWrV	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	3850	TRUE	0
22	20ByCA$dXEFeTAIGKcGWrU	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	975	TRUE	0
23	20ByCA$dXEFeTAIGKcGWrT	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	4214.9999999	TRUE	0
24	20ByCA$dXEFeTAIGKcGWrS	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	2250	TRUE	0
25	20ByCA$dXEFeTAIGKcGWrZ	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	1185	TRUE	0
26	20ByCA$dXEFeTAIGKcGWrY	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	2270	TRUE	0
27	20ByCA$dXEFeTAIGKcGWrX	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE	Yes				h7/0o		2450.0000000	TRUE	609.99999999	TRUE	0
28	20ByCA$dXEFeTAIGKcGWrW	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	1755.0000000	TRUE	0
29	20ByCA$dXEFeTAIGKcGWrd	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	2580	TRUE	0
30	20ByCA$dXEFeTAIGKcGWrc	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	694.99999999	TRUE	0
31	20ByCA$dXEFeTAIGKcGWrb	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	2580	TRUE	0
32	20ByCA$dXEFeTAIGKcGWra	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	2579.9999999	TRUE	0
33	20ByCA$dXEFeTAIGKcGWrh	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	1715	TRUE	0
34	20ByCA$dXEFeTAIGKcGWrg	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	670	TRUE	0
35	20ByCA$dXEFeTAIGKcGWrf	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	1175	TRUE	0
36	20ByCA$dXEFeTAIGKcGWru	Basic Wall:Ytong cellenbeton G5/800 100 mm		TRUE							2450.0000000	TRUE	975	TRUE	0
37	20ByCA$dXEFeTAIGKcGWqQ	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	10504.999999	TRUE	4
38	20ByCA$dXEFeTAIGKcGWqP	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	8035.0000000	TRUE	2
39	20ByCA$dXEFeTAIGKcGWqO	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	8035.0000000	TRUE	1
40	20ByCA$dXEFeTAIGKcGWqa	Basic Wall:Ytong cellenbeton G4/600 70 mm		TRUE							2450.0000000	TRUE	694.99999999	TRUE	0
41	20ByCA$dXEFeTAIGKcGWAa	Basic Wall:EPS Isolatie + stucwerk 150 + 10 mm		TRUE			50		XC1, XC3		2600.0000000	TRUE	10504.999999	TRUE	3
42	20ByCA$dXEFeTAIGKcGWAm	Basic Wall:EPS 80 mm		TRUE							530	TRUE	2048	TRUE	0
43	20ByCA$dXEFeTAIGKcGWAt	Basic Wall:EPS 80 mm		TRUE							600	TRUE	110.00000000	TRUE	0
44	20ByCA$dXEFeTAIGKcGWAs	Basic Wall:EPS 80 mm		TRUE							530	TRUE	2148	TRUE	0
45	20ByCA$dXEFeTAIGKcGWAr	Basic Wall:EPS 80 mm		TRUE							600	TRUE	110.00000000	TRUE	0
46	20ByCA$dXEFeTAIGKcGWAq	Basic Wall:EPS 80 mm		TRUE							530	TRUE	2048	TRUE	0

Figure 4.1-3 screenshot of overview of most detailed level

Bibliography

Appendix 4

buildingSMART International Limited. (n.d.). Industry Foundation Classes Release 4 (IFC4). Retrieved October 10, 2016, from http://www.buildingsmart-tech.org/ifc/IFC4/final/html/

Chipman, T., Liebich, T., & Weise, M. (2016). mvdXML, 1.1, 49.

W3C. (2014). No Title. Retrieved December 1, 2016, from https://www.w3.org/TR/1998/REC-xml-19980210#dt-xml-doc

Appendix 5

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