Master Thesis

Construction Management & Engineering

Eindhoven University of Technology

COMPONENT BASED BUSINESS PROCESS AUTOMATION FOR AUTOMATING TASK-ORIENTED WORK INSTRUCTIONS FOR CONSTRUCTION WORKERS.

Component Based Business Process Automation for Automating Task-Oriented Work Instructions for Construction Workers.

L.H.M. (Leon) Vergeldt

Presentation date: 28-08-2017

GRADUATION COMMITTEE

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

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

ir. R.R.W.I. (Renato) Kindt

(This page is intentionally left blank)

COLOPHON

Project:	Master Thesis
Title:	Component Based Business Process Automation for Automation in the Construction Industry
Subtitle:	Component Based Business Process Automation for Automating Task-Oriented Work Instructions for Construction Workers.
Date:	24/08/2017
Defense date:	28/08/2017
Version:	2.0
Status:	Final

Author:	L.H.M. (Leon) Vergeldt
Student ID:	0853404
E-mail:	l.h.m.vergeldt@student.tue.nl
University:	Eindhoven University of Technology
Program:	Construction Management & Engineering

Graduation committee:
Prof. dr. ir. B. (Bauke) de Vries (chairman)
Eindhoven University of Technology
b.d.vries@tue.nl

Dr. dipl.-ing J. (Jakob) Beetz
Eindhoven University of Technology
j.beetz@tue.nl

ir. R.R.W.I. (Renato) Kindt
Eindhoven University of Technology
r.r.w.i.kindt@tue.nl

In collaboration with:
BBBDC – B&B Bouw Data Combinatie
Lage Kanaaldijk 115, 6212NA, Maastricht
Supervisor: Gerard van Booma
gerardvanbooma@bbbdc.nl

“A smooth sea never made a skilled mariner.” — Franklin D. Roosevelt.

PREFACE

With great pleasure, I present to you my Master Thesis for the completion of the study Construction Management & Engineering at Eindhoven University of Technology. This study has trained me to become a professional in the field of information management in construction. This program has been a truly valuable addition to my earlier studies in construction management and general construction, and my personal skills and knowledge.

Within the master track, I specialized myself in information management in construction, since I always enjoyed using computers to make the maximum use of its capacities to reach higher goals and higher efficiency. Besides this, I have been active in the field of construction for a long time, and always have been excited about large construction projects. I have trained myself to combine the best of both worlds, and will pursue this in my later career.

This research project has been a combination of both these interests, where I could use my knowledge from construction sites and workers, which I gained from my earlier studies and experiences. I also gained more knowledge and skills to make better use of information management for construction practices. Besides this, I learned to implement the practical experiences in a scientific research project on an abstract level.

To conclude, I would like to thank the people involved in my research project. Firstly, I would like to thank my first graduation supervisor Jakob Beetz for the help designing the research project in the initial stages, and his critical view to make sure the research was detailed. Secondly, I would like to thank my second graduation supervisor Renato Kindt for always being available and his vision and feedback on the overall view on the research project, and his observations on my personal growth. Thirdly, I would like to thank Gerard van Booma from BBBDC for his practical views on the overall goals of the research project and his help with gaining resources and contacts. Finally, I would like to thank Thomas Krijnen for his help in teaching the functioning of Python and its packages and his help with creating the final application.

I wish you a lot of pleasure reading this report.

Eindhoven, 30-05-17

Leon Vergeldt

TABLE OF CONTENTS

SUMMARY

In recent years, the architecture, engineering and construction (AEC) industry has been adopting digital solutions rapidly. These digital solutions, such as 3D modeling, document management systems, simulation and analysis lead to new possibilities, and eventually increase efficiency. The industry is struggling to implement these new possibilities and connect these systems as an integral solution. Building Information Modeling (BIM) is one of these techniques. BIM is a method to integrate geographical and non-geographical information from various stakeholders in order to enhance collaboration. By using BIM tools, 3D objects becomes more intelligent and additional information can be connected. BIM has multiple advantages over the traditional construction process, such as reducing errors, decrease construction time, improve collaboration, reduced rework, and increased profits. Currently, BIM is mainly used in the design and planning stage, and the maintenance phase of construction projects, but are often ignored in the construction phase.

The research goals were to apply the possibilities of a data-oriented approach and contribute to a change from a document-oriented construction process to a data-oriented process to make automation possible. This is done by developing an application which automatically creates Task-Oriented Work Instructions (TOWI) for construction workers based on open BIM data and aims to answer the following research question:

MAIN RESEARCH QUESTION

What is needed to make a (semi) automatic filter of Task-Oriented Work Instructions from open Building Information Models for the construction worker, and how can this be achieved?

This research project gives insight in and contribution to the transition from a document-oriented industry to a data-oriented industry and the possibilities that come with it. This research project discusses an approach to use BIM for an new information carrier for informing construction workers about their tasks. This research project continues on former research in applying TOWI as a new information carrier from Van Berlo & Natrop (2014) and Span (2015), and discusses the possibilities of automation of TOWI, by implementing and testing the business process automation (BPA) approach of Shi & Lee (2006) and Shi, Lee & Kuruku (2008).

Construction workers use paper-based drawings and specifics, such as plan views, elevation views, sections and detail books. These documents are designed for multiple purposes. The documents are designed for and used by multiple stakeholders in the chain. The documents consists of contradicting information, errors, irrelevant information, incorrect and aged information. Besides that, the information which is needed to perform a task, is scattered over multiple documents, which leads to a lot of searching time between different information sources.

TOWI are an overview of information designed for construction workers which includes all the relevant information which is needed to perform a task, and does not contain irrelevant information for that specific task. Both researchers as construction workers think TOWI will be very useful for their daily information source. TOWI will contribute to a more efficient and clear construction process. Construction workers will be informed better, which will lead to a

decrease of errors, less information searching time, less distraction, less interpretation and less information overload.

The TOWI are created based on the open BIM standard Industry Foundation Classes (IFC). IFC is an open standard for exchanging BIM-product data, including geometrical data as metadata.

To create the TOWI, the Business Process Automation (BPA) approach is used. This method identifies, automates, executes, and measures workflows. BPA is a method to identify business processes and to convert these processes into process components which can be automated. By connecting these automated components and executing them in a workflow engine, processes can be (partially) automated. To do so, the output (information demand) has to be determined precisely; next, the input should be checked for the availability of the information. When the information is available and the output is known, the process of creating the TOWI can be automated.

This process is implemented in the case study to automate the process of creating TOWI for the task 'placing and assembling window(frames)'. The BPA process was successfully used within this research project to automate the creation of TOWI by creating an application that automatically creates the TOWI for the aforementioned task.

The added value of this research project is not just automating the process of the creation of TOWI, but the concept of automation for construction in whole. The step to a data-oriented industry is ongoing, and comes with many new possibilities of automation as can be seen in this research project. Instead of creating documents with large amounts of information, data can be filtered to show only relevant information for its purpose. This allows IT-tools to be configured in new ways to filter information and give the user immediate insight the of the information they need.

Continuation of this research project can be done by applying knowledge management to save process models and its components to implement automation in other workflows. Also, it would be interesting be to further investigate the use of digital techniques on the construction site. Within this research project, the construction workers had a preference for paper-based methods, but the author thinks digital applications would be more efficient and accepted on the construction site. Furthermore, the author thinks, the process of creating TOWI could be easily adapted to more tasks, but this should be implemented and tested. The author would advise to rebuild the used method to a method which uses the workflow engine. Because of the user-friendliness and similar functioning.

SAMENVATTING

De afgelopen jaren heeft de bouwindustrie snel digitale oplossingen aangenomen. Deze digitale oplossingen, zoals 3D-modellering, document management systemen, simulatie en analyse leiden tot nieuwe mogelijkheden, en verhogen uiteindelijk efficiëntie. De industrie worstelt om deze nieuwe mogelijkheden te implementeren en deze systemen te verbinden als een integrale oplossing. Building Information Modeling (BIM) is een van deze technieken. BIM is een methode om geografische en niet-geografische informatie van verschillende belanghebbenden te integreren om de samenwerking te verbeteren. Door gebruik te maken van BIM-tools worden 3D-objecten intelligenter en kan extra informatie worden aangesloten. BIM heeft meerdere voordelen ten opzichte van het traditionele bouwproces, zoals het verminderen van fouten, het verminderen van de bouwtijd, het verbeteren van de samenwerking, verminderde herwerking en een hogere winst. Momenteel wordt BIM voornamelijk gebruikt in het ontwerp- en voorbereidingsfase en de onderhoudsfase van bouwprojecten gebruikt, maar wordt vaak in de uitvoeringsfase genegeerd.

De onderzoeksdoelstellingen waren de mogelijkheden van een data-georiënteerde aanpak toe te passen en te bijdragen aan een verandering van een documentgericht bouwproces naar een data-georiënteerd bouwproces om automatisering mogelijk te maken. Dit wordt gedaan door een applicatie te ontwikkelen die op basis van open BIM-gegevens automatisch taakgerichte werkinstructies (TGWI) genereert voor bouwarbeiders en beoogt de volgende onderzoeksvraag te beantwoorden:

ONDERZOEKSVRAAG:

Wat is nodig om een (semi) automatisch filter van taakgerichte werkinstructies van open gebouwinformatiemodellen voor de bouwvakker te maken en hoe kan dit worden bereikt?

Dit onderzoeksproject geeft inzicht in en levert bijdrage aan de overgang van een document-georiënteerde industrie naar een data-georiënteerde industrie en de mogelijkheden daarbij. Dit onderzoeksproject bespreekt een aanpak om BIM te gebruiken voor een nieuwe informatiedrager om bouwarbeiders te informeren over hun taken. Dit onderzoeksproject gaat voort op het eerdere onderzoek bij het toepassen van TGWI als een nieuwe informatiedrager van Van Berlo & Natrop (2014) en Span (2015). Daarnaast bespreekt het de mogelijkheden van automatisering van TGWI, door het implementeren en testen van de bedrijfsprocesautomatisering (BPA)-benadering Van Shi & Lee (2006) en Shi, Lee & Kuruku (2008).

Bouwarbeiders maken gebruik van op papier gebaseerde tekeningen en specificaties, zoals plattegronden, gevels, doorsneden en detailboeken. Deze documenten zijn ontworpen voor meerdere doeleinden. De documenten zijn ontworpen voor- en worden gebruikt door meerdere belanghebbenden in de keten. De documenten bestaan uit onder andere tegenstrijdige informatie, fouten, irrelevante informatie, onjuiste en oude informatie. Daarnaast is de informatie die nodig is om een taak uit te voeren, verspreid over meerdere documenten, wat leidt tot veel zoektijd tussen verschillende informatiebronnen.

TGWI zijn een overzicht van informatie dat is ontworpen voor bouwarbeiders die alle relevante informatie bevat die nodig is om een taak uit te voeren en bevat geen irrelevante

informatie voor die specifieke taak. Zowel onderzoekers als bouwwerkers denken dat TGWI zeer nuttig zal zijn voor hun dagelijkse informatiebron. TGWI zal bijdragen tot een efficiënter en duidelijker bouwproces. Bouwarbeiders worden beter geïnformeerd, wat leidt tot een daling van fouten, minder informatie zoektijd, minder afleiding, minder interpretatie en minder informatie overbelasting.

De TGWI is gemaakt op basis van de open BIM standaard Industry Foundation Classes (IFC). IFC is een open standaard voor het uitwisselen van BIM-productgegevens, inclusief geometrische data en metadata.

Om de TGWI te creëren wordt de aanpak van Business Process Automation (BPA) gebruikt. Deze methode identificeert, configureert, voert uit en meet werkprocessen. BPA is een methode om bedrijfsprocessen te identificeren en om deze processen om te zetten in procescomponenten die geautomatiseerd kunnen worden. Door deze geautomatiseerde componenten te verbinden en ze in een workflow-engine uit te voeren, kunnen processen (gedeeltelijk) geautomatiseerd worden. Om dit te doen moet de output (informatiebehoefte) exact bepaald worden; Vervolgens moet de invoer gecontroleerd worden op de beschikbaarheid van de informatie. Wanneer de informatie beschikbaar is en de uitvoer bekend is, kan het proces van het maken van de TGWI geautomatiseerd worden.

Dit proces is geïmplementeerd in een casestudy om het proces van het creëren van TGWI te automatiseren voor de taak 'het plaatsen en monteren van ramen / stelkozijnen'. Het BPA-proces werd succesvol gebruikt in dit onderzoeksproject om de creatie van TGWI te automatiseren door een applicatie te creëren die automatisch de TGWI voor de bovengenoemde taak genereert.

De toegevoegde waarde van dit onderzoeksproject is niet alleen het automatiseren van het proces van het genereren van de TGWI, maar ook het concept automatisering voor de bouw in het geheel. De stap naar een data-georiënteerde industrie is onderweg en komt met veel nieuwe mogelijkheden van automatisering zoals blijkt uit dit onderzoeksproject. Hiermee kunnen, in plaats van documenten te creëren met grote hoeveelheden informatie, gegevens worden gefilterd om alleen relevante informatie te tonen voor het bijbehorende doel. Hiermee kunnen IT-tools op nieuwe manieren worden geconfigureerd om informatie te filteren en de gebruiker onmiddellijk inzicht te geven in de informatie die ze nodig hebben.

Het voortzetten van dit onderzoeksproject kan worden gedaan door kennisbeheer toe te passen om de procesmodellen en onderdelen ervan op te slaan om automatisering in andere werkprocessen te implementeren. Ook zou het interessant zijn om het gebruik van digitale technieken op de bouwplaats verder te onderzoeken. Binnen dit onderzoeksproject hadden de bouwarbeiders een voorkeur voor papier-gebaseerde methoden, maar de auteur vindt dat digitale applicaties efficiënter zouden zijn en dat deze zullen worden geaccepteerd op de bouwplaats. Bovendien denkt de auteur dat het proces van het creëren van TGWI gemakkelijk kan worden aangepast voor meer (andere) taken, maar dit moet geïmplementeerd en getest worden. De auteur adviseert om de gebruikte methode te herbouwen op een methode die de workflow-engine gebruikt, vanwege de gebruiksvriendelijkheid en aangezien het op eenzelfde manier functioneert.

ABSTRACT

In recent years, the architecture, engineering and construction (AEC) industry has been adopting digital solutions rapidly. These digital solutions, such as 3D modeling, document management systems, simulation and analysis lead to new possibilities, and eventually increase efficiency. The industry is struggling to implement these new possibilities and connect these systems as an integral solution. Building Information Modeling (BIM) is one of these techniques. BIM is a method to integrate geographical and non-geographical information from various stakeholders in order to enhance collaboration. By using BIM tools, 3D objects becomes more intelligent and additional information can be connected. Currently, BIM is mainly used in the design and planning stage, and the maintenance phase of construction projects, but are often ignored in the construction phase.

The research goals were to apply the possibilities of a data-oriented approach and contribute to a change from a document-oriented construction process to a data-oriented process to make automation possible. This is done by developing an application which automatically creates Task-Oriented Work Instructions (TOWI) for construction workers based on open BIM data and aims to answer the following research question:

MAIN RESEARCH QUESTION

What is needed to make a (semi) automatic filter of Task-Oriented Work Instructions from open Building Information Models for the construction worker, and how can this be achieved?

TOWI are an overview of information designed for construction workers which includes all the relevant information which is needed to perform a task, and does not contain irrelevant information for that specific task. Both researchers as construction workers think TOWI will be very useful for their daily information source. TOWI will contribute to a more efficient and clear construction process.To create the TOWI, the Business Process Automation (BPA) approach is successfully used. This method identifies, automates, executes, and measures workflows. This process is implemented in the case study to automate the process of creating TOWI for the task ‘placing and assembling window(frames)’.

LIST OF ABBREVIATIONS

AEC-industry	Architecture, Engineering and Construction industry
BIM	Building Information Modeling
BPA	Business Process Automation
BPM	Business Process Modeling
BPMN	Business Process Modeling and Notation
BPR	Business Process Re-engineering
CAD	Computer Aided Design
IDM	Information Delivery Manual
IFC	Industry Foundation Classes
KM	Knowledge Management
KPI	Key Performance Indicator
LOD	Level Of Development
MVD	Model View Definitions
TOWI	Task-Oriented Work Instructions
TBM	Task Based Modeling
UML	Unified Modeling Language
PSD	Program Structure Diagrams
XML	eXtensible Markup Language

PART A: RESEARCH PROJECT

This section introduces the problem analysis, research background and the topics which are involved in this research project. Furthermore, the research design, approach and questions are explained.

1. INTRODUCTION

In recent years, the architecture, engineering and construction (AEC) industry has been adopting digital solutions rapidly. These digital solutions, such as 3D modeling, document management systems, simulation and analysis lead to new possibilities, and eventually increase efficiency. The industry is struggling to implement these new possibilities and connect these systems as an integral solution.

When looking at the history, the AEC-industry has a very conservative image towards innovation. Compared to other industries, the AEC-industry is known as an industry with low (0.22%) investments in Research and Development to improve innovation and automation (TNO, 2005). Where other industries such as the manufacturing industry has increased its productivity significantly, the AEC-industry has actually declined its productivity over the last 50 years (Teichholz, 2013). Teicholz (2013) accounts the increase in productivity in the manufacturing industry mostly to more accessible computerized processes for planning and coordination activities. About one-third to the adoption of automation, and about two-third to the adoption of a leaner (LEAN) approach.

The AEC-industry has been adapting digital solutions at a much lower pace. Figure 1 shows a roadmap of the use of (digital) information in construction projects. The figure defines 4 information levels (BIM Levels 0, 1, 2 and 3), and has added a timeline with the introduction of the techniques.

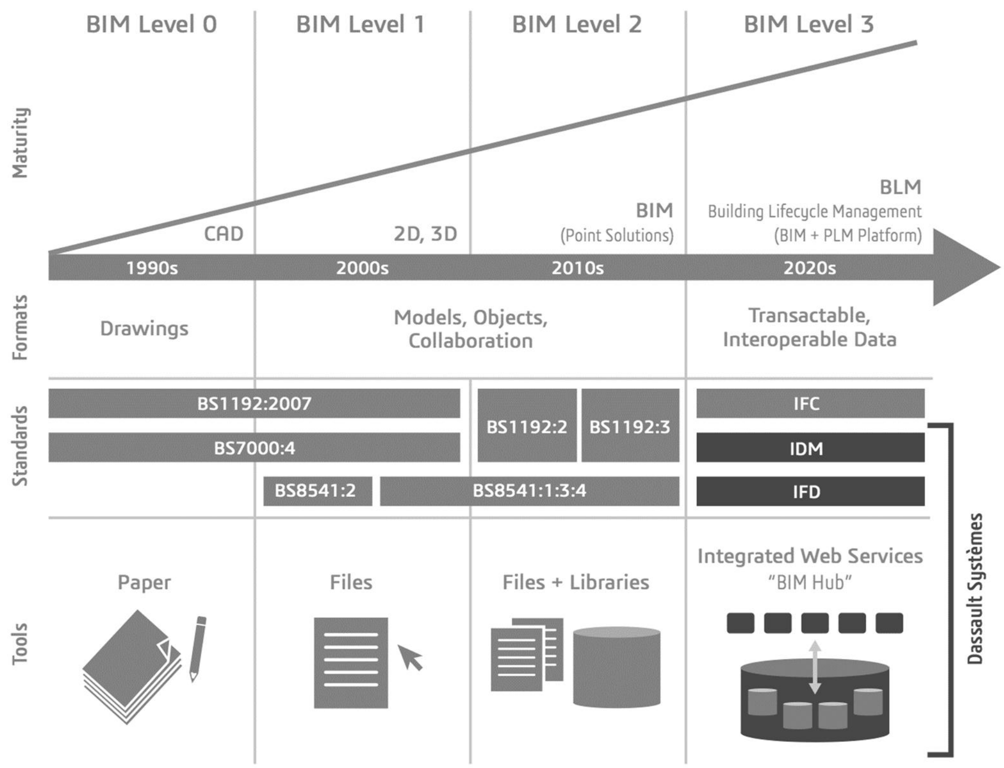


Figure 1: BIM levels (Marty, 2014)

In the 90s, computers became standardized in the industry. These came with new possibilities for calculations, logs and Computer Aided Design (CAD). CAD, eventually, became the industry standard for drawing construction projects. The drawings were then printed and send to the stakeholders. Nowadays, CAD is still the industry standard for most parties in the AEC-industry. Later, new communication methods, such as e-mail and file sharing came. This improved the design process speed. Also, 3D CAD was introduced which added an extra dimension to drawing. Complex structures could be visualized more easily.

2010 was the year that Building Information Modeling (BIM) became popular under early adopters in the AEC-industry. BIM is a method to integrate geographical and non-geographical information from various stakeholders in order to enhance collaboration. By using BIM tools, 3D objects becomes more intelligent and additional information can be connected. By using the BIM-approach for construction projects, the project information will increase over time, ideally without losing information when moving through different phases (figure 2). Nowadays, BIM is becoming standardized in the western world and mandated by some governments (e.g. United Kingdom, Norway and Singapore). Currently, BIM is mainly used in the design and planning stage, and the maintenance phase of construction projects, but are often ignored in the construction phase (intense black line figure 2).

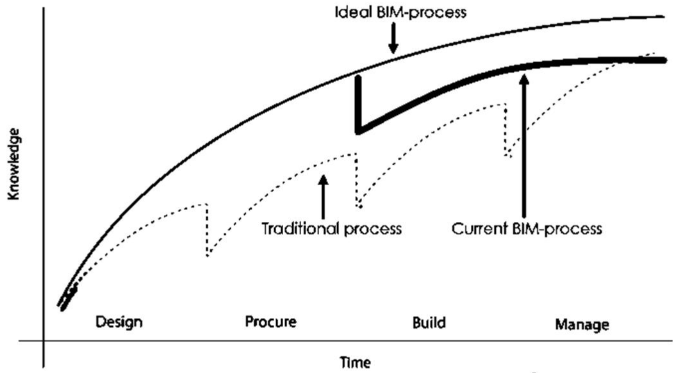


Figure 2: Sawtooth-curve of building information during lifecycle (Beetz, 2015) (adapted)

At the same period as the introduction of BIM, connected servers and cloud services became popular to share construction project information, such as documents and BIM-models (Figure 1). The next phase will change the AEC-industry from a document-oriented to an object, and data-oriented industry. Figure 3 shows that current formats are readable by humans, but not understood by computers because natural language is used. Using computer readable data will lead to new possibilities of automation (Figure 3).

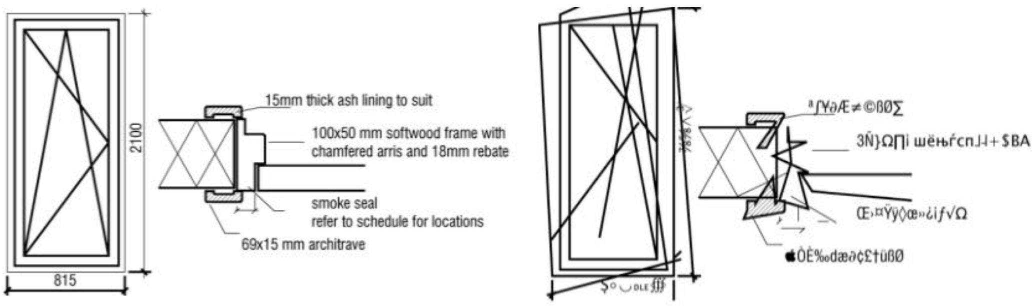


Figure 3: Natural language unreadable by computers (Beetz, 2015)

1.1 RESEARCH PROJECT

The current use of BIM makes the use of automation tools, such as energy analysis and queries for extracting quantities possible because BIM data is readable by computers. Using these tools makes the design process of construction projects more efficient. As stated before, BIM is not continued in the transition of the preparation phase to the construction phase. In many projects the BIM-process is abandoned. The (digital) information will be exported to traditional two-dimensional paper drawings, which are not computer readable and contain less information than the digital BIM-product. As a result, many benefits of using the digital techniques (up-to-date, intelligent, connected, etc.) disappear.

Currently, multiple applications for construction sites are available, such as Autodesk 360 (https://a360.autodesk.com) and Trimble Connect (http://connect.trimble.com), however most of the applications are mainly used for quality control and not helping the actual construction process. There is a market demand for a well-functioning application which can be used for the actual construction process, which makes use of the digital possibilities and replaces the traditional document-oriented methods.

RESEARCH GOALS

The research goals are to apply the possibilities of a data-oriented approach and contribute to the change from a document-oriented process to a data-oriented process to make automation possible for construction workers. This will be done by developing an application which automatically creates Task-Oriented Work Instructions (TOWI) for construction workers based on open BIM data and aims to answer the following research questions:

MAIN RESEARCH QUESTION

What is needed to make a (semi) automatic filter of Task-Oriented Work Instructions from open Building Information Models for the construction worker, and how can this be achieved?

SUB-QUESTIONS

1. How do Task-Oriented Work Instructions differ from the traditional approach?
    a. How are construction workers informed on their tasks in the traditional approach?
    b. What are Task-Oriented Work Instructions, and what are the expectations and requirements of Task-Oriented Work Instructions?
2. What information is needed for Task-Oriented Work Instructions according to literature and field professionals, and how is/can this information be stored in data?
3. How can the Task-Oriented Work Instructions filter be defined, captured in a generic way (that is easy to be reused and adapted in other scenarios) and how will it be processed?
4. How can the Task-Oriented Work Instructions filter contribute to the construction workers, and how are the results reviewed?

SOCIETAL RELEVANCE

This research project is relevant for the AEC-industry, because this will give insight and contribute in the transition from a document-oriented industry to a data-oriented industry and the possibilities that come with it. Besides this, there are issues in implementing the use of BIM on the construction site. This research projects discusses an approach to use BIM for an new information carrier for information construction workers about their tasks. In general, using a data-oriented approach will increase efficiency, since static documents such as pdf, and paper drawings will be eliminated due to data as main information carrier. Also will this research contribute to a more productive process, since by automation manual steps will be eliminated. Besides this, this research project will increase the knowledge on business process automation for construction purposes.

The developed application is especially interesting in contracting companies in the Netherlands which make use of the BIM-method as their leading information flow.

SCIENTIFIC RELEVANCE

This research project continues on former research in applying TOWI as a new information carrier from Van Berlo & Natrop (2014) and Span (2015), and discusses the possibilities of automation of TOWI, by implementing and testing the business process automation (BPA) approach of Shi & Lee (2006) and Shi, Lee & Kuruku (2008). The implications and limitations of applying this method in the AEC-industry will be discussed. Industry and researchers can utilize, adapt and research the designed application and approach further.

Van Berlo & Natrop (2014) and Span (2015) have researched a 'new' information carrier for informing construction workers. They researched so called task-oriented work instructions (Dutch: taakgerichte werkinstructies) to improve the communication and information supply for the construction workers. Instead of informing by using large information overviews such as floor plans, elevation views and section views, only task-related information is handed over. These task-oriented work instructions exist of all the work-related information, and are compiled on an A3 paper format. This communication method led to multiple advantages, such as less clashes during construction, less errors, less mistakes, easier interpretation, convenient information overviews, lower searching and understanding time, less distraction/ information overload, and task-oriented instructions (instead of plain information). Despite the advantages, these task-oriented work instructions have to be assembled/created manually, which is a time-consuming task, and also requires a lot of in depth and implicit knowledge about the construction process. As a result, only a selected group of employees is capable of creating these instructions. Besides this, for every different task a different instruction has to be created. The concept of TOWI will be discussed extensive in the literature review (chapter 3).

Shi & Lee (2006) and Shi and Lee & Kuruku (2008), have researched Business Process Automation to automate standardized processes. They have developed a method to implement BPA within processes. This process consists of four steps, 1) Define task, 2) Configure components, 3) Execute process, 4) Measure outcome. The complete process will be discussed extensively in chapter 5.

SCOPE

The project focus is on automating the information filtering process for product-related information (objects, dimensions, product info, etc.) for the construction phase, aimed at construction workers who are assigned to the task 'placing and assembling window (frames)'. The input which will be used for the 'filter' is based on the IFC (Industry Foundation Classes) format, which is a platform neutral open source format to describe construction industry data.

PROTOTYPE APPLICATION

This research project aims to make use of the digital BIM process to automate the process of creating TOWI by using the BPA process. To elaborate, an application will be developed that generates TOWI for construction workers by using (open) BIM input. This will decrease human interaction and therefore an increase of productivity, and will improve the information carrier for the construction workers by decreasing searching costs, increasing overview of the information and a decrease of information overload.

Figure 4 illustrates the process of information filtering. The process is as follows: a client has an request for information (RFI) with a certain information demand to be able to fulfil his task. Next, the exact information demand of the client is researched. Then, the information database (for example an integrated BIM-model) is checked for the availability of the information. Following, a process model and automation components are created to automate and filter the information demand process (see chapter 6 Business Process Automation). Therefore, a standardized filter profile will be used, created or adapted to the actual needs. From then on, the business process is executed by the application with the database as input information on the moment the information is needed. This process results in the relevant information for the client. Next, the client can redefine its information demand to improve the overall process in a circular process.

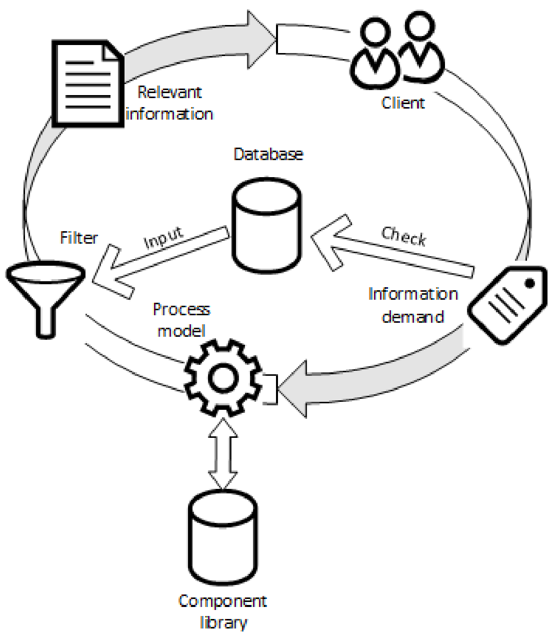


Figure 4: Information filtering process (own figure)

1.2 RESEARCH DESIGN

The research project is divided into two main parts. 1) research involving a literature review on the involved topics. 2) The implementation and prototype for an actual case. This leads to a combination of both a qualitative research method as more practical implementation/ prototyping. This approach requires a broad package of skills and activities, namely literature review, interviews, observation, data analysis, scripting, process design, application design and writing.

On Figure 5, the research approach, its results and required activities are depicted. This approach will be discussed in-depth.

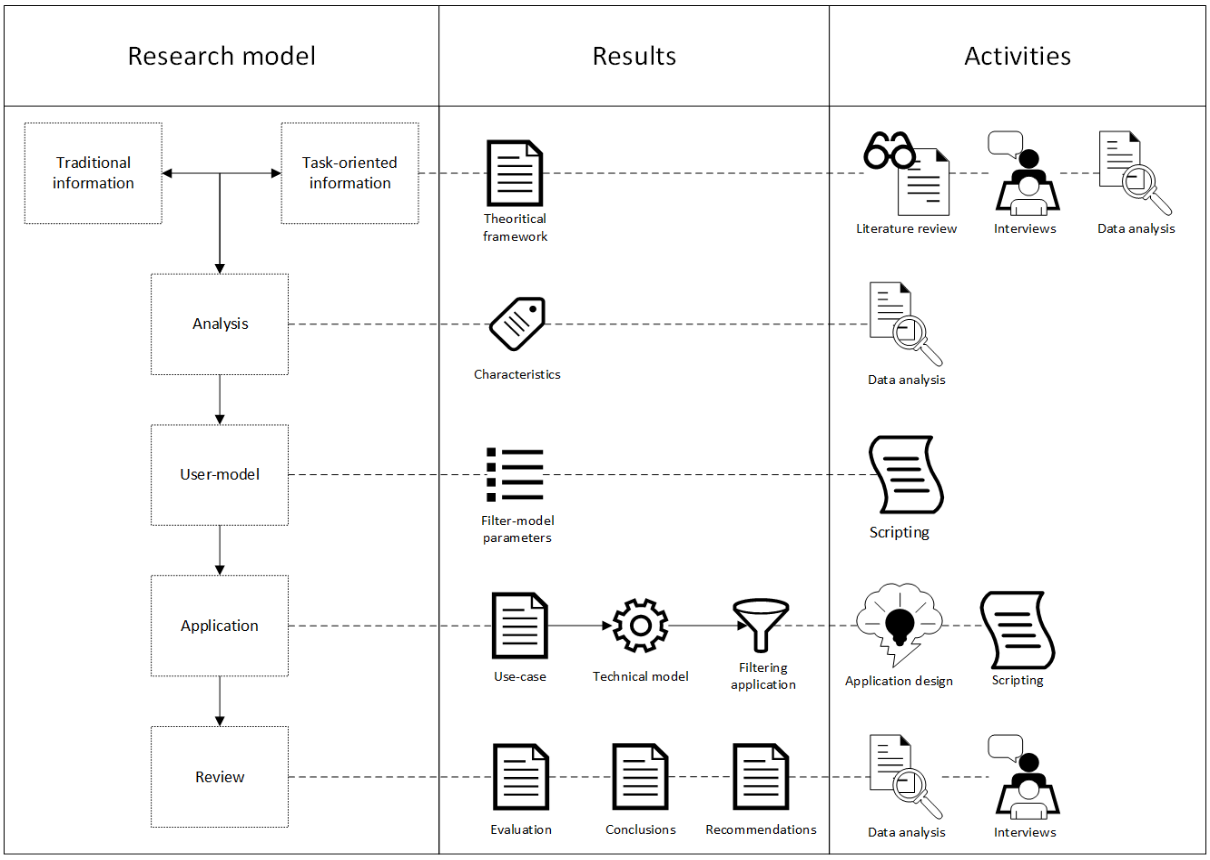


Figure 5: Research model

THEORETICAL FRAMEWORK

The theoretical framework aims to answer the first two questions. In the theoretical framework, the background of the keywords (Information in construction, task-oriented work instructions (TOWI), Building Information Modeling), and business process automation (BPA) are explained and discussed in-depth. Also, the relations between the different keywords are discussed. Next, the information demand for TOWI for the task placing and assembling window (frames) will be stated. Data for this part is gained by performing interviews with actors in the construction phase, reviewing literature and analyzing documents.

DEVELOP USER-MODEL

Based on the information demand, the software components (see chapter 5) will be created. The software components are created to generate the correct (digital) output as requested. The characteristics will be researched on their possibilities of filtering and availability of (meta)data. The results are (technical) functions which make the characteristics possible for filtering data.

DEVELOP APPLICATION

The fourth step aims to answer the third question. In this part, the application will be designed (front-end and back-end), and the design will be executed. The design principles and conditions will be designed in the use-case. Then, the technical background (i.e. the functional operation of the to be designed application) is designed in the technical model. The final application is the technical elaboration of the use-cases and the back-end scheme. The software components that are developed in the user-model are used to run the application. The application will be tested during the developing phase for direct improvement.

CASESTUDY

A prototype of the earlier described tool will be developed and its impact on the process will be discussed. The prototype will be tested on an actual case, namely the creation of task-oriented work instructions for construction workers performing the task: 'placing and assembling window (frames)'. This task will be discussed in-depth in chapter 2 information in construction.

REVIEW

The last part reviews the studied theories and developed application, which lead to the conclusions and recommendations. In this part, the application will be tested in the field to test the functioning, and the reviews of the construction workers are judged on the output of the tool, and possible improvements and possibly missing information. The initial findings in the theoretical framework are evaluated and discussed on the final output.

1.3 READING GUIDE

In the first chapter, the background, research goals, research questions and research design were discussed. In chapters 2, 3, and 4, literature will be reviewed and discussed on respectively information in construction, task-oriented work instructions and Building Information Modeling. Chapter 5 discusses the used framework and its approach. In chapter 6, the framework will be implemented on the case study, and discusses the use of the implementation. Chapter 7, discusses the implications of the used framework and effects on business processes. Chapters 8 and 9 discusses the research with conclusions and recommendations. The appendix can be found in chapter 11.

PART B: LITERATURE REVIEW

This section consists of the theoretical framework on three main topics: 1) Information in construction. 2) Task-oriented work instructions 3) Building Information Modeling and aims to answer the following research questions:

How does task-oriented information differ from the traditional approach?

- *How are construction workers informed on their tasks in the traditional approach?*
- *What is task-oriented information, and what are the expectations and requirements of task-oriented information?*

What information is needed for task-oriented information according to literature and field professionals, and how is/can this information be stored in data?

2. INFORMATION IN CONSTRUCTION

2.1 INTRODUCTION

Before discussing information in construction, the meaning of information will be discussed shortly. According to dictionaries, information is:

1. *Information is that which informs. In other words, it is the answer to a question of some kind. It is thus related to data and knowledge, as data represents values attributed to parameters, and knowledge signifies understanding of real things or abstract concepts.* (Wikipedia, 2017)
2. *Data that is (1) accurate and timely, (2) specific and organized for a purpose, (3) presented within a context that gives it meaning and relevance, and (4) can lead to an increase in understanding and decrease in uncertainty.* (Business Dictionary, 2017)

Both definitions use sort of similar concepts to describe the meaning of information. Information is related to data and knowledge, and has the purpose to inform and decrease uncertainty. Within this report, information is:

Information informs people or resources, to decrease uncertainty and support in decision-making. Information is interpreted by people or resources which are influenced by knowledge. Information is transferred by communication.

INFORMATION IN CONSTRUCTION

The amount of information in construction projects is enormous. Within large or complex projects, thousands of documents are exchanged between the stakeholders (Ramaekers, 2013). To handle these amounts of information accurate, extensive and advanced information systems are required (Wallinga, 2011). Not managing the information flow efficiently, will lead to uncontrollable processes, a decrease of the final product, and legal conflicts. In other words, failure costs (Wallinga, 2011). According to research of USP Marketing Consultancy, inefficient information exchange and communication are the most significant (24%) causes of failure costs in construction (USP Marketing Consultancy, 2010). Thus, proper information management is a crucial skill to prevent failure costs in construction. The AEC-industry is a document-oriented industry where documents are exchanged and interpreted between the stakeholders.

2.2 CASE STUDY

This paragraph introduces the case study which is used within this research project. To identify what information is used on the construction site, more specifically by construction workers who place and assembles windows(frames) in residential buildings, interviews and observations were performed on 6 similar small to medium housing projects in the area of Eindhoven (The Netherlands). All construction workers worked in couples or triples, so 6 double semi-structured interviews were conducted. The results of the case study will be discussed throughout this chapter. The results of the interviews can be found in Appendix 1: Interviews construction workers.

In general, the procedure of the task placing and assembling window(frames) is as follows:

Preparations

1. Inner wall is complete for mounting the window(frame) and the window opening is present (fig 1);
2. Determine and indicate the meter level (Dutch: meterpeil) on the outside of the inner wall (fig 2);
3. Determine and indicate horizontal measurements on the outside of the inner wall, based on the grid (fig 2);
4. Calculate measurements including the mounting frame (Dutch: stelkozijn) or cavity frame (Dutch: spouwlat) and indicate measurements on the outside of the inner wall;
5. Place window(frame) fasteners (Dutch: kozijnankers) on the outside of the inner wall on the bottom + 1 side (fig 2);
6. Place other fasteners including tolerance (fig 3);
7. Write window mark on outside inner wall, so it can be identified easily (fig 4);

Placing the window(frame)

8. Locate window storage and move correct window (window mark == mark on wall) correct to window opening (fig 4 + 5)
9. Temporarily mount window (frame) to fasteners (fig 6);
10. Definitely mount window (frame) to fasteners on correct window depth (Dutch: negge) of outside exterior wall, by measuring from ropes (Dutch: metselkoorden) (fig 6).

1.

2.

3.

4.

5.

6.

2.3 INFORMATION TYPES

To be able to manage the information, it is essential to know the differences between the different types of information. Within the AEC-industry, there are two different information types, namely:

- Product-related information
- Process-related information

Product-related information is information which is relevant for the final product (e.g. bridge, road, wall or window). Examples of product-related information are: drawings, models, reports, calculations, documentation, tables, etc.

Process-related information is information which is relevant to the process (e.g. time, money, quality). Examples of process-related information are: schedules, prices, contracts, budgets, etc. Process-related information is based on product-related information. Some information resources can consist of both product, as process information (e.g. contract documents).

Besides the two types of information, information can be recorded either analog (paper), or digital (CAD-drawings, PDF-files, IFC-files, Excel-sheets, etc.).

INFORMATION TYPES FOR CONSTRUCTION WORKERS

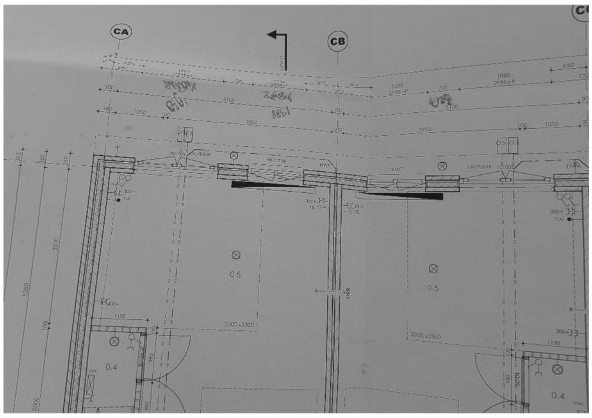
On the construction site, multiple information sources are required for completing tasks. Information is scattered over different analog information sources, such as plan views, section views, elevation views, details, quantities, work-plans, and sub-contractor specifications. These information sources are frequently inconsistent and do not match each other. Besides this, the information sources are often large paper documents (A0) which cover the whole project, or a major part of the project. This often results in information overload for the construction workers, especially when they are not familiar with the project. The information leads to uncertainties, searching time and costs and errors.

During practical research, the construction workers assigned to the task 'placing and assembling window(frames)' were asked about the current information use and its form. For this task, they used the following information sources: (number of times named in brackets)

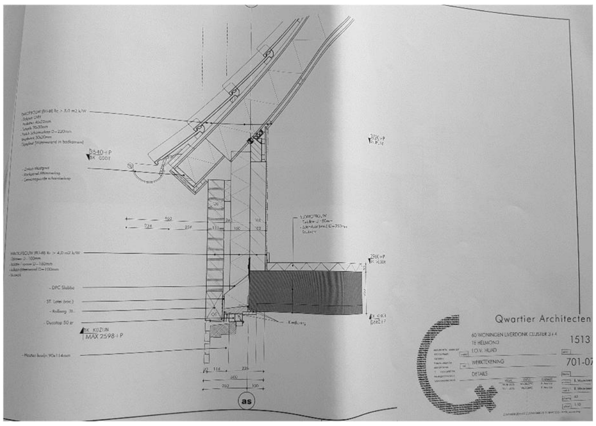
1. Floorplan (Dutch: Werktekening) (6)
2. Elevation view (5)
3. Detail book architect (5)
4. Detail book supplier (6)

Pictures of each document can be found on the next page.

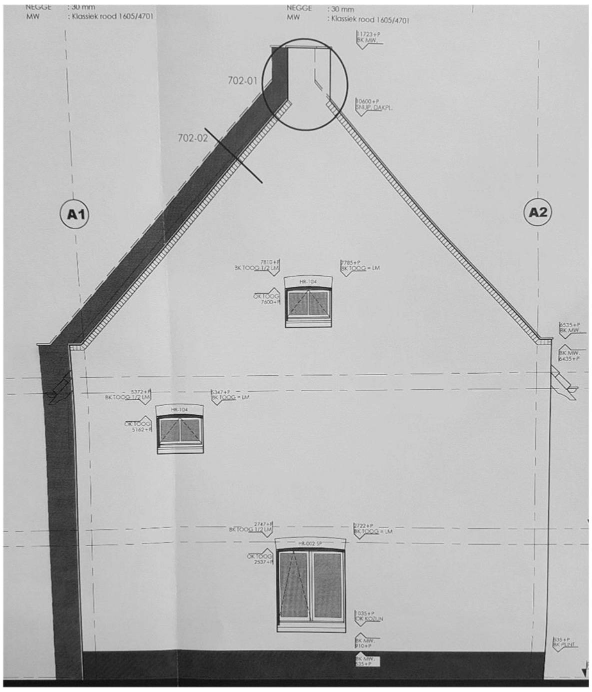
1.



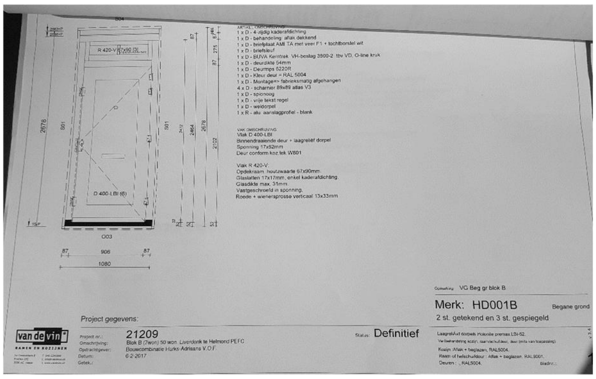
2.



3.



4.



So, all construction workers required the floorplan and detail book of the window supplier, and most construction workers required the elevation view and detail book of the architect to perform this task. Other documents, such as sections were mentioned, but not used for this task. Thus, on average 4 different information sources are required to perform the aforementioned task. All of the documents are product-related information sources and all documents are paper-based.

The construction workers were asked about their findings of the information sources. All used information sources were paper based. They mentioned the following issues: (number of times named in brackets)

- Contradicting information (6)
- A lot of searching time between different information sources (4)
- Many different documents required / scattered information (3)
- Errors in information (2)
- No consistency between different documents (2)
- Irrelevant information (2)
- Incorrect information / aged information (2)
- Not all the information available (1)

Most of these errors are related to the quality of the information, and especially the inconsistencies between the different information sources. These errors lead to uncertainties to the construction crew which lead to additional searching costs and to errors when the information is interpreted wrong or is incorrect.

2.4 INFORMATION TRANSFER

Information is transferred by an information flow and follows a standard protocol. According to Shannon (1948), the information flow exists of the sender, encoder, channel, decoder, and receiver of the information. The information flow starts either by a request for information (RFI) from a certain stakeholder (e.g. the site supervisor needs information about a product), or by informing a stakeholder by sending information (e.g. the contractor informs the client about the progress). Next, the message is send by a channel, for example a telephone, email, cloud, or voice. The channel is influenced by noise, which is a negative influencer. There are two sorts of noise:

- Internal noise, which is a disturbing factor which concerns the sender and the receiver. Examples are interpretation, emotions, obscure messages, information overload, unclear information.
- External noise, which is a disturbing factor which does not concern the sender or receiver, but external factors such as compatibility, unstable conversion, and network problems.

The information will then be decoded by the receiver's system and the message will be converted into a message which is understandable for the receiver.

INFORMATION FLOW CONSTRUCTION WORKERS

During practical research, construction workers assigned to the task 'placing and assembling window(frames)' were asked about the current flow of information. All information sources were paper based and were obtained by the site-superintendent. They mentioned the following issues: (number of times named in brackets)

- A lot of searching time between different information sources (4)
- Not all information available: a lot of movement between workplace and site-superintendent shack (4)
- Many different documents lead to searching/browsing (3)
- Incorrect information / aged information (2)
- Information overload leads to confusion and searching (2)
- Not all the information available (1)
- Specific information hard to find (1)
- Information is (too) late (1)
- No overview of information (1)
- Unclear information due to information overload (1)
- Unclear where information can be found (1)

The current format of information sources consists of a lot of internal noise (information overload, irrelevant information, unclear information, contradicting information) and information is scattered over multiple sources. The current format also consists of external noise (errors in information, aged information). As a result, the information sources have to be researched thoroughly and browse through multiple documents to identify the relevant information and its relations. The construction workers mentioned that the current flow of information, where information is scattered over different information sources, is not very efficient.

Besides noise, they mentioned that often information for specific situations/objects is hard to find, and they often need to consult with the site-superintendent.

The information sources are designed as general documents which can be used for a variety of tasks. These tasks can be construction site related (e.g. information for constructing and assembly of products) or preparation and organization related (e.g. contract documents, quantity take-off, sub-contractor bids, scheduling). As a result, the information on the documents consist of lots of information for different actors, so they can be used for different tasks. The information is not valuable for every task or stakeholder (e.g. construction workers are not interested in contract related information such as standards and laws). (Van Berlo & Natrop, 2014)

Besides the goal of making uniform information overviews which can be used for many disciplines, the information demand of the construction workers is unclear. As a result, the information is often incomplete (scattered information, impractical), irrelevant (information overload) and unclear (information overload). (Span, 2015; Tilley & Barton, 1997)

In addition to the aspects on the overview of information, the current information provision process is inefficient. Currently there is a paper-based information flow from the design team to the construction site. Often, the information sources are designed by design parties (architect, structural engineer), they send their drawings to the contractor by mail or email.

Next, the contractor sends the drawings to the construction site. Then, when the site-supervisor sorts the new drawings, and provides them to the construction workers, the information flow is completed. This process may take several weeks. Nowadays, digital techniques involve real-time updates. Compared with this, the current information flow process is very slow. This slow process leads to errors and information being late, which lead to delays in other processes (Kimoto, Endo, Iwashita, & Fujiwara, 2005).

To conclude, the current flow of information for construction workers is paper-based which comes with disadvantages such as, outdated, not available (at the worksite) and hard to find specific information. Also does the information carrier (paper drawings) consist of a lot of noise, both internal as external. Many of these issues are due to the fact that the documents are designed as general document to perform various tasks and are not designed for the target group (not task/role specific). As a result, multiple documents are required for the complete information, which results in lots of searching time (costs).

2.5 INFORMATION DEMAND CONSTRUCTION WORKERS

To be able to provide correct information to the construction workers, it is essential to identify the exact information demand. Depending on the users, organization, resources, manpower and the situation, multiple methods to identify the information demand can be used (Devadason, 2008). Since the focus of this research project is on construction workers, which are performing a similar task on similar projects, and it is known that the focus is on the information to perform this task, few methods are left over. Devadason (2008) says, that in such a situation, formal interviews with the users often are the best method to identify the information demand. For the interviews, he advises to prepare the interviews with a list with a rough estimate of anticipated information needs. This will help to confirm the actual needs and to eliminate pseudo needs. Besides this, it is important to distinguish the interests and the actual needs. To do so, asking questions as; what is the priority? How often do you need this? In what quantity do you need this? etc. can help to tell the difference. Finally, the information needs can be refined by asking for feedback, showing examples and discussing the output. Devadason (2008) mentions that asking 3-5 qualified people usually will identify general information needs.

To identify the information demand of the construction workers for the task 'placing and assembling window(frames)', 6 double semi-structured interviews including with document analysis and observation are performed. Draft information needs were used as input for the interviews and examples of TOWI were shown. The construction workers were asked to classify the information demand in four classes, essential (3), important (2), desirable (1), and unimportant (0). Since it was difficult to classify all the unimportant information, this class was not used. By identifying the exact needs, the remaining information can be classified as unimportant. To increase the level of overview, the unimportant information should be kept to a minimum.

USE OF CURRENT INFORMATION

The construction workers were asked about the information demand on the previously mentioned information sources for the task placing and assembling window(frames). They required the following information (number of times mentioned in cells):

Table 1: Information demand floor plan

Information type:	Essential	Important	Desirable
Location/orientation	4		
Horizontal dimensions based on grid	4		2
Mark/type	2		
Grid	1		
Wall structure		3	

Table 2: Information demand elevation view

Information type:	Essential	Important	Desirable
Location/orientation	3		
Bottom height relative to base level (Dutch: peil =0)	5		
Top height relative to base level (Dutch: peil =0)	3		
Mark/type	5		
Depth (Dutch: negge)	1	1	
Corresponding detail (number)			2

Table 3: Information demand detail book architect

Information type:	Essential	Important	Desirable
Depth (Dutch: negge)	5		

Table 4: Information demand detail book supplier

Information type:	Essential	Important	Desirable
Mark/type	2		
Bottom height relative to base level (Dutch: peil =0)	1		
Top height relative to base level (Dutch: peil =0)	1		
Mounting frame dimensions (Dutch: spouwlat / stelkozijn)	5		
Fastenings	1		
Depth (Dutch: negge)	3		
Form (Dutch: vorm)	1		

Table 5: Information demand in total

Information type:	Essential	Important	Desirable
Depth (Dutch: negge)	9	1	
Mark/type	9		
Location/orientation	7		

Bottom height relative to base level (Dutch: peil =0)	6		
Mounting frame dimensions (Dutch: spouwlat / stelkozijn)	5		
Horizontal dimensions based on grid	4		2
Top height relative to base level (Dutch: peil =0)	4		
Form (Dutch: vorm)	1		
Grid	1		
Wall structure		3	
Fastenings		1	
Corresponding detail (number)			2

In Table 5 it is summarized that in total, 12 different types of information were requested by the construction workers for the task placing and assembling window(frames). An interesting fact is the phenomenon that some information types (depth, mark, and orientation) are mentioned more times, than actual interviews were performed (6). This means that similar information is used and needed on different information sources. For example, the window mark was scattered over all four used documents, and the window depth was scattered over three documents. This confirms their opinion on the unclarity of where to find the required information. Other essential information types are the vertical and horizontal dimensions of the window based on base level or the grid. The least important information types have no direct connection with the windows itself, but are informing the construction workers about the location (grid, wall structure and corresponding detail). This emphasizes the importance of the orientation to navigate through the information.

It can be stated that the most important information types are the window depth (Dutch: Negge), window mark, window location, window height from base level, mounting frame dimensions, horizontal dimensions based on the grid, and orientation. For some interviewees, some information was not important because it was not relevant to the situation. For example, the horizontal location was determined based on the openings in the walls instead of the floor plan dimensions. Another example, was that the mounting frame dimensions were not important because they were always checked on-site.

The construction workers were also asked about information that was missing in the current format and would be useful for performing their tasks. Most of the construction workers did not miss any information. One construction worker would prefer water resistance of the information and another construction worker was interested in the type of brickwork and its measurements to make better decisions.

INFORMATION FILTERING

To conclude, currently construction workers have to search for relations between multiple different documents to find the information they need to perform their task. This leads to a lot of searching time and is prone to errors. For a relatively simple task, 4 different documents are required to find 7 attributes (Figure 6). Often these documents are not uniform and make construction workers decide what information is correct or have to ask their supervisor. This may lead to errors, time-loss and inefficiencies. The overview of information is unclear and different on each project. A main reason of this, is the fragmented character of the AEC-industry, where multiple stakeholders, such as subcontractors have their own procedures, strategies, methods and styles. There are no strict procedures about the form, detail and content. By applying Information Delivery Manuals (IDM), the information overview could become more clear.

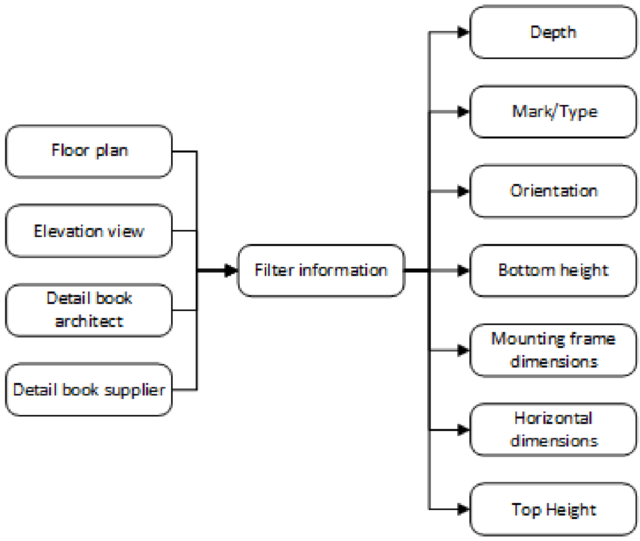


Figure 6: Current information filtering

DIGITAL INFORMATION FILTERING

To be able to filter the digital information, the data types should be known. There are many different data types, such as integers (0,1,2,3, etc.), strings (‘hello’, ‘world’, names, etc.), booleans (yes, no, true, false, etc.), reals (1.002, 5.96, 102.45, etc.), and Voids (null). Different data types require different filtering approaches. For example, working with geographical or numerical data as input may have different possibilities than textual data. For example, numerical data can be used for all kinds of calculations and can be used in simulation analysis. The possibilities with textual data are more limited since text is human readable and, however many research programs are going on, limited readable/understandable by computers.

Table 6: Data types information demand

Information demand	Data type:
Depth	Length = real
Mark/type	Text = string
Orientation	-
Bottom height	Length = real
Mounting frame dimensions	Length = real
Horizontal dimensions	Length = real
Top height	Length = real

Table 6 shows the translation from information demand to data types. By connecting the information demand with digital properties, the information can be filtered (Table 6).

Most of the data types of the information demand are related to geometrical positioning and have dimensions which are exact numbers connected to a coordinate-system. The window mark is textual data, which is a string. The data type 'orientation' is more complex to define, since this is related to the knowledge of the user. For example, angle to true North could be used, but is too limited. The data types will be discussed in-depth in chapter 6 where they are implemented in the application.

3. TASK-ORIENTED WORK INSTRUCTIONS

3.1 INTRODUCTION

As stated in the previous chapter information in construction, do construction workers use paper-based drawings and specifics, such as plan views, elevation views, sections and detail books. These documents are designed for multiple purposes. The documents are designed for and used by multiple stakeholders in the chain. For example, an elevation view is usually designed by an architect. The architect uses this view to express his design and communicate this with other stakeholders. The client uses this document to get his design approved by the municipality and uses this document in the contract. The estimators at the contractor use this document to calculate areas and materials. The planner at the contractor uses this to order materials and communicate with producers, such as window fabricators and prefabricated element manufacturers, and to hire employees or subcontractors. The site-supervisor of the contractor uses the document for organizing his logistics and schedule on the construction site and the construction worker uses the document for his orientation, window types and measurements.

Besides that the documents are designed for multiple purposes, and are used by many different stakeholders, information is scattered over multiple documents. As a result, multiple documents are needed to get an overview of the relevant information.

To emphasize, the standard contract documents are used for multiple purposes; however, since these documents are not detailed enough for many other purposes, new documents are designed based on the contract documents. For example, the window fabricator needs additional information to design and fabricate the actual window. Therefore, he creates a new window overview plan and a detail book.

With the introduction of BIM, there is questioned if the traditional formats of information overviews are still relevant. BIM can be used to filter and generate more specific information to make it more relevant to the stakeholder.

A relatively new approach (in the AEC-industry) to inform construction workers about their tasks is the concept of Task-Oriented Work Instructions (Van Berlo & Natrop, 2014; Span, 2015; Hopman, 2016). TOWI are an overview of information designed for construction workers which includes all the relevant information which is needed to perform a task, and does not contain irrelevant information for that specific task. The construction workers will get 1 overview to perform the task; as a result, they do not have to search for the relevant information in different information overviews, they have less possibilities to deviate from the designs (which is wrong), and will decrease the information overload. On Figure 7, an example of an TOI overview for placing floor reinforcement can be seen.

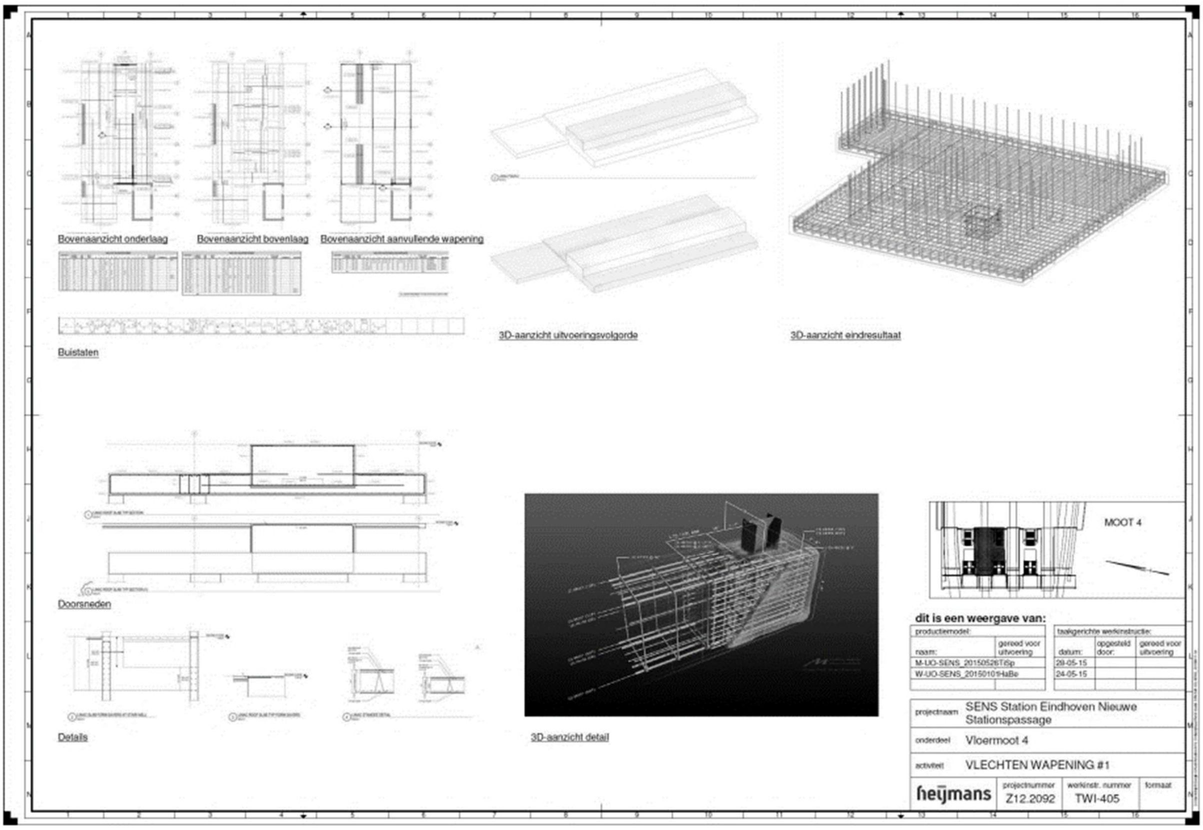


Figure 7: Example TOI placing floor reinforcement (Span, 2015)

Figure 7, shows different views of information merged on 1 overview. Plan views, sections, details and 3D views are combined to inform the construction worker. Normally, the construction worker should have used at least 3 different documents to get similar information. Another interesting fact are the possibilities of other types of information that can be added to the TOWI. Van Berlo & Natrop (2014) stated that weather conditions, phone numbers or contact persons, prescriptions and instruction manuals could be useful. Other researchers named the following information: activity interactions, sequence of work, approvals and responses, flow, coordination of work, material and equipment management, scheduling (acceleration, compressions), error tolerance, supervisor, material and equipment availability, safety, suitability of tools and equipment, travel distance to worksite, working conditions (lighting, noise, etc.) and work assignments (Matthews, et al., 2015).

Both product-related (geometry, tolerance) as process-related (weather conditions, phone numbers) have potential to inform the construction workers. These aspects can give extra insight to the construction workers about their own tasks, but also to related tasks to make them more informed. Many of these aspects will become important on larger construction sites, since it will be demanding to inform all construction workers personally.

In the interviews with construction workers with the task ‘placing and assembling window(frames)’, there was asked about some of these information types. The construction workers were not very interested in these information types, because they ‘already know the rules’ and ‘know how to perform their tasks’. No extra information types were requested by the interviewees. The author relates this to the size of the projects. The projects had between 14 and 100 residential buildings and between 2 and 6 similar roles (construction workers). These were relatively small construction sites, where personal instructions were manageable and a less hierarchic approach was applied. On larger sites, as researched by Matthews et al. (2015), travel distance was important, because a single walk could be up to 30 minutes. So the relevance of these additional aspects differ per situation. The correlation between the aspects and the project size has not been researched.

3.2 CREATING TASK-ORIENTED WORK INSTRUCTIONS

Span (2015) has developed a method to create task-oriented work instructions (TOWI) with the input of BIM-models. He has developed a process existing out of 4 steps. The process is as follows:

- Step 1: BIM design
- Step 2: Preparations task-oriented work instructions
- Step 3: Create task-oriented work instructions
- Step 4: Record feedback

STEP 1: BIM DESIGN

To be able to create the TOWI, the required information should be available and accessible in the BIM. Therefore, the information needs of the construction workers should be known and available in the BIM. Both standardized information needs, as project specific information needs can be used. The process of identifying and adding the information to the BIM is summarized in Figure 8. Firstly, the BIM designer of the correct discipline (e.g. architect, structural engineer, window producer) should be aware of the information needs. This knowledge is gained by either using standardized information needs from a database, or new/specific information needs from the construction worker. Then he has to work up the level of detail of the BIM, so it consists of the information which is required for the TOWI. Finally, the discipline models will be merged and checked and are ready to be used for creating TOWI. Since, this process can be used project-wide, and the information demand can be adapted, this is designed as an iterative process. Feedback is discussed in step 4.

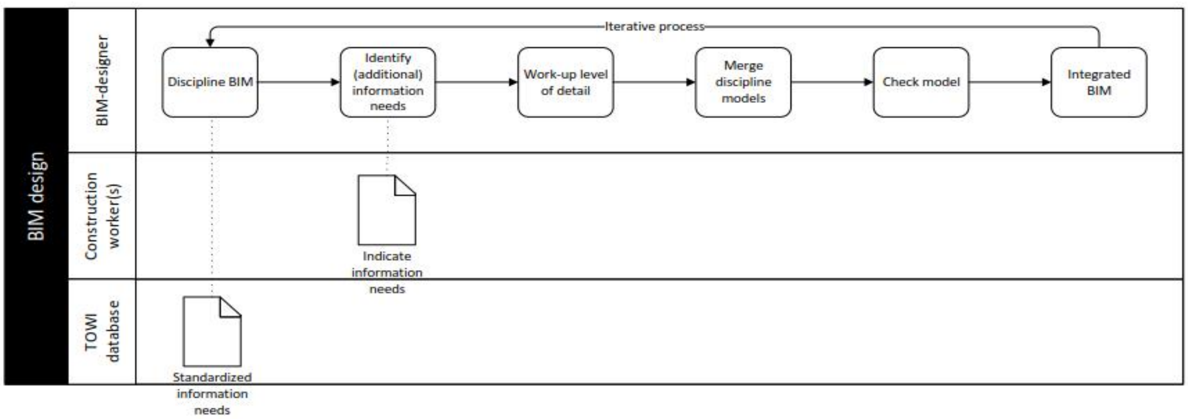


Figure 8: Identifying and adding TOWI information to the BIM

STEP 2: PREPARATIONS TASK-ORIENTED WORK INSTRUCTIONS

Next, preparations for the TOWI can be made. The integrated BIM is the input for this process. When construction on the site starts, the objects in the BIM have to contain all the necessary information and these objects have to change their status to production. Then a part of the BIM will be locked for production, so they cannot be modified anymore (Figure 9).

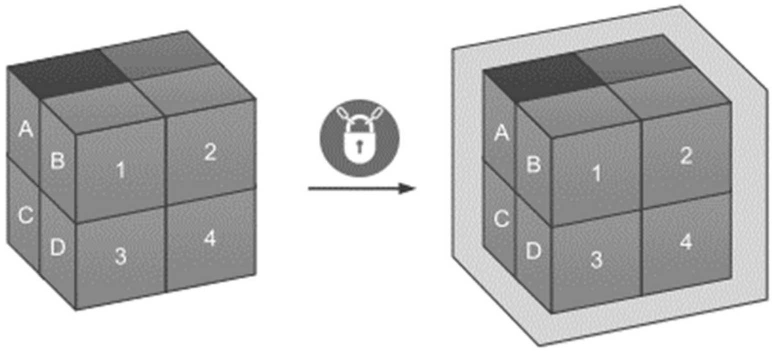


Figure 9: Locking part of BIM (Span, 2015)

Then, the form of the information overview, such as 3D-view, plan view, section view, etc. should be identified by communicating with the construction workers. The information overviews should contain the information as was requested in the information demand phase. Span (2015) has developed a standardized checklist to identify the information overviews.

STEP 3: CREATE TASK-ORIENTED WORK INSTRUCTIONS

When the information has reached its final state a part of the BIM is locked for production, the TOWI can be created. This is a manual process, which is done by the construction planner. The process starts by creating the required information overviews (making sections, plan views, 3D-views, etc.). Depending on the amount of information required by the construction workers, the size (A4, A3, A2, etc.) of the TOWI is decided. A maximum size of A3 is preferred, since many printers on construction site have a maximum paper size of A3. Construction workers do not care about the size, but about the readability of the information on the overviews. Next, geometrical and non-geometrical information is merged on the information overviews manually. TOWI related dimensions, grid, and colors have to be added manually. Then, non-BIM information, such as phone numbers, contact persons, and weather conditions, will be added to the views. Later, the layout for the TOWI will be designed, and information will be projected on a readable scale. Finally, project related information (Company information, logo's, dates, document number, etc.) will be added and then the TOWI is finished and can be generated to the requested format (Figure 10). Span (2015) has researched that currently, paper based output is the most realistic option, but this might change in the (near) future.

After creation of the TOWI, the TOWI will be checked for readability, completeness and practicability, by both the producer and the site-supervisor(s) (Figure 11). In case of no comments, the TOWI are complete. The site-supervisor can print the TOWI on the moment the information is needed.

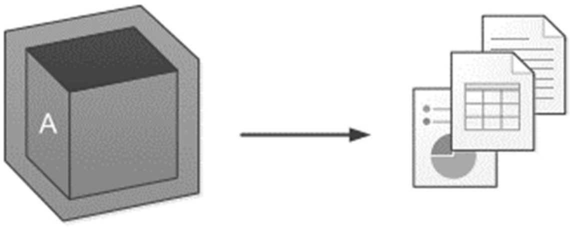


Figure 10: Creating TOWI from Locked-BIM (Span, 2015)

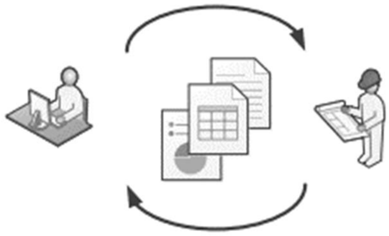


Figure 11: Check TOWI (Span, 2015)

STEP 4: RECORD FEEDBACK

The last step is to record feedback for both the TOWI, as the BIM-model. Firstly, the as-built situation will be analyzed and modification/deviations will be recorded by the site-supervisor. This feedback will be communicated to the designers, who will edit the BIM to create a so-called As-built BIM model. Experiences from the TOWI will be shared with the creator for possible improvements (Figure 12). In case a new task was emerged, standardized information needs will be recorded. This will result in a database with standardized processes to improve the quality and speed of creating TOWI (Figure 13).

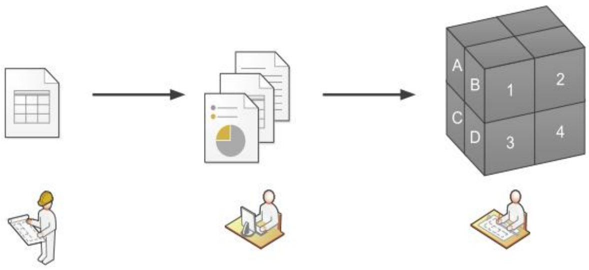


Figure 12: Site-feedback to BIM-model (Span, 2015)

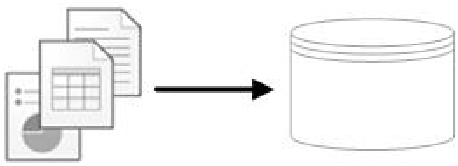


Figure 13: TOWI feedback to database (Span, 2015)

CONCLUSION

To conclude, the process of creating TOWI consists of many steps and a lot of consulting with the construction site is required to get the overviews right. The information demand and feedback should be managed in a companywide library of work packages. Complex is the part of locking parts for production, because some tasks have to be executed in an earlier phase than others, for example pouring concrete walls is done in an earlier phase than making the ceiling. Often, decisions still have to be made while construction starts. Therefore, locking parts should be connected to the tasks and its associated information demand. Overall, the process of creating TOWI is a complex and labor-intensive process, which is demanding for the BIM-product designer and requires early input from the site-supervisor(s).

3.3 TOWI AS INFORMATION OVERVIEW

Several research projects have looked at the advantages and disadvantages of TOWI as information carrier to the construction site. Both literature, as practice has been studied about the findings of TOWI. Table 7 summarizes the advantages and disadvantages which have been stated in the literature.

Table 7: Advantages and disadvantages TOWI

Advantages	Disadvantages
Less distraction (Van Berlo & Natrop, 2014; Span, 2015)	Labor-intensive to create TOWI (Van Berlo & Natrop, 2014)
Construction workers are better informed, which leads to higher quality and less failure costs (Van Berlo & Natrop, 2014; Span, 2015)	Inefficient in case of incomplete information (Van Berlo & Natrop, 2014)
Only relevant information / decrease of information overload (Van Berlo & Natrop, 2014; Natrop, 2017)	BIM-model should be used, updated and the leading information source (Span, 2015)
On demand and actual information (Van Berlo & Natrop, 2014; Matthews, et al., 2015)	Competent actors have to be available (Van Berlo & Natrop, 2014)
“No walking back to (and getting ‘stuck’ in) site-office” (Davies & Harty, 2013)	Can only be used for 1 task (Van Berlo & Natrop, 2014)
Possibilities of additional information (Van Berlo & Natrop, 2014; Span, 2015; Matthews, et al., 2015)	Strict information management and planning (Van Berlo & Natrop, 2014)
Not prone to changes in design (Natrop, 2017)	
No conflicting information (Matthews, et al., 2015)	
Complete information (Matthews, et al., 2015)	
Only 1 overview (Koedoot, 2016; Natrop, 2017)	
Similar to current information (paper-based), but better (Natrop, 2017)	
Less searching time (Van Berlo & Natrop, 2014; Span, 2015)	
Information is task-oriented (Van Berlo & Natrop, 2014)	
Relations between information (Natrop, 2017)	
Increased usability of information (Meža, Turk, & Dolenc, 2014)	

In short, the researchers are mainly positive about this ‘new’ way of informing construction workers, but there are still some hurdles to overcome. These hurdles are about the organization and the process of creating the TOWI, not the information carrier itself. To make use of TOWI, the process should be clear and aimed at the use of TOWI. Therefore, the process of Span (2015) could be used. This process however, includes the labor-intensive task of creating the TOWI.

On the other hand, using TOWI will lead to a more efficient and more clear construction process. Construction workers will be informed better, which will lead to a decrease of errors, less searching time, less distraction and less information overload. Besides this, the construction workers can be informed better, by using additional information. Matthews et al. (2015) found that additional information about the process enables a work force to see the status of shared equipment, what other work forces were doing, what spaces are available, etc. This made the work force make intelligent choices about their own work process.

The researchers also state that the importance of paper-based drawings is expected to decline when BIM becomes present on construction sites. Sharing live information using digital techniques will make the most out of the construction site, schedule, equipment, etc. When the new (digital) techniques make entrance to the construction site for construction workers, there will be no clear reason to maintain the current formatting. (Matthews, et al., 2015)

FINDINGS OF CONSTRUCTION WORKERS

In 6 double interviews, construction workers were asked about their findings about TOWI, and what is important when using TOWI. The construction workers were not familiar with TOWI; therefore, the researcher explained the content and showed an example.

Overall, all the interviewed construction workers were enthusiastic about the idea and saw a big potential in this ‘new’ information carrier. Two of the 12 interviewed construction workers were sceptic about TOWI, and found the current information carriers useful. The author relates this to the fact that they were working on a very small project (14 residential buildings) and had many responsibilities since there was no site-supervisor. Therefore, they were very familiar with the project and are very skilled in reading the information and its connections. However, they mentioned, that this technique could be interesting on larger projects where construction workers have less responsibilities and less time to get known with the project. For example, when they are only assigned to one task, such as sub-contractors or hired construction workers.

Table 8 summarizes the findings and importance of TOWI according to the construction workers (number of times named in brackets).

Table 8: Findings of construction workers on TOWI

Findings of TOWI	Importance of TOWI
Very useful (5)	The TOWI should be complete (6)
Less searching time (5)	It should be clear where the information belongs to (3)
All the required information on one overview (3)	The TOWI should be correct / trustworthy (3)
A lot of work to create the overviews (1)	The information should be water resistant (1)
Less need for interpretation (1)	The information should be on paper (1)
Another 'book' with information, many books for different tasks (2)	Information must be consistent, and changes must be processed (1)
Similar to method for making limestone walls, works fine for that task (1)	
Clear information (1)	
Loss of project overview (1)	

In short, the construction workers think TOWI will be very useful for their daily information source. They see the biggest advantage in the overview and searching time. Currently, they have to search the relations between multiple different documents, and base their opinion on that. They expect to need less interpretation. For the task making limestone walls, TOWI are already standardized as information carrier. Construction workers are very positive about this method. Some negative reactions include the loss of overall project overview and that the TOWI can only be used for one task.

According to the construction workers, TOWI should be complete, correct, trustworthy, up-to-date, paper-based, water resistant and the orientation should be clear.

To conclude, both researchers as construction worker are positive about TOWI and see potential in this form as information carrier. Important is that the information on the TOWI is complete and correct. The lack of introduction of TOWI in the (Dutch) AEC-industry can be concluded to two main reasons; namely, the labor-intensive process of creating TOWI and the unfamiliarity of this technique in the industry.

4. BUILDING INFORMATION MODELING

4.1 INTRODUCTION

A short introduction to BIM was given in the introduction of this report. This topic is important for this research project because BIM is getting increasingly popular in the AEC-industry, and has the possibilities to make better use of digital techniques in the industry. This research project uses BIM as input for the creation of TOWI.

As stated in the introduction, BIM is a method to integrate geographical and non-geographical information from various stakeholders in a single database, during the lifetime of a project in order to enhance collaboration. BIM has multiple advantages over the traditional construction process, such as reducing errors, decrease construction time, improve collaboration, reduced rework, increased profits, and more (Eastman, Teichholz, Sacks, & Liston, 2011).To elaborate, BIM can be separated into two parts, namely the BIM-product and the BIM-process.

BIM-PRODUCT

The BIM-product represents the data of the project. For example, the BIM-product of a residential building might exist of a 3D-model with intelligent objects which have both geometrical (lengths, volumes, surfaces, etc.) as non-geometrical data (metadata, weights, names, thermal-resistance, etc.). These intelligent objects make advanced analysis' possible, make relations with other objects and can be linked to other related information, such as costs, timelines, contracts, guarantees, etc. This approach of an integrated information system connects data to keep in control of the information.

The need for intelligent information models comes from efficiency. To be able to increase the efficiency, computers have to understand the models. The design of such integrated information system is complicated; therefore, information levels are created. These Levels Of Development (LOD) describe the phase of such intelligent object and how detailed the information of that object should be. There are 5 levels of development, namely LOD 100, LOD 200, LOD 300, LOD 350, and LOD 400 (BIM Forum, 2015). Where LOD 100 is the lowest and LOD 400 the highest. This research project uses the LOD of the BIM Forum. The exact specifics can be downloaded here: http://bimforum.org/lod/

To give an example how the levels develop on an object, the following window example is used:

- LOD 100: costs per window attached to walls
- LOD 200: window, generic/approximate size/shape/location
- LOD 300: exact window type/size/shape/location/material
- LOD 350: actual window type/size/shape/name / paint
- LOD 400: Actual mounting details/ guarantee / verified measurement

BIM-PROCESS

The BIM-process is the means of cooperation, management and information flow of the BIM-product. Cooperation of the BIM-product can be done by four different methods, namely (Figure 14):

1. little closed BIM
2. BIG closed BIM
3. little OPEN BIM
4. BIG OPEN BIM

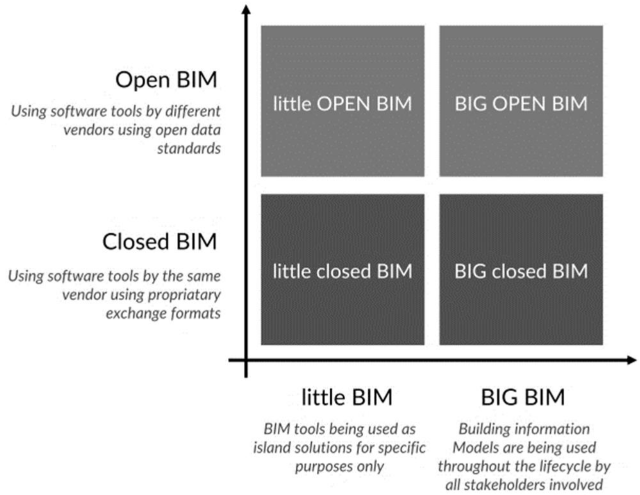


Figure 14: BIM-cooperation (Beetz, 2015)

Little BIM is the use of the BIM-product within a company internally for only one stakeholder/discipline. BIG BIM is where the BIM-product will be used by multiple stakeholders/disciplines, either within a company or with other companies. OPEN and closed BIM will be discussed in paragraph 4.2

BIG BIM strives to match the theoretical approach of BIM (Jernigan, 2010). In that case, BIM-products are created by multiple disciplines (e.g. architect, structural engineer, HVAC-designer) and merged (periodically) in one integrated product, also called ‘federated model’ (Figure 15).

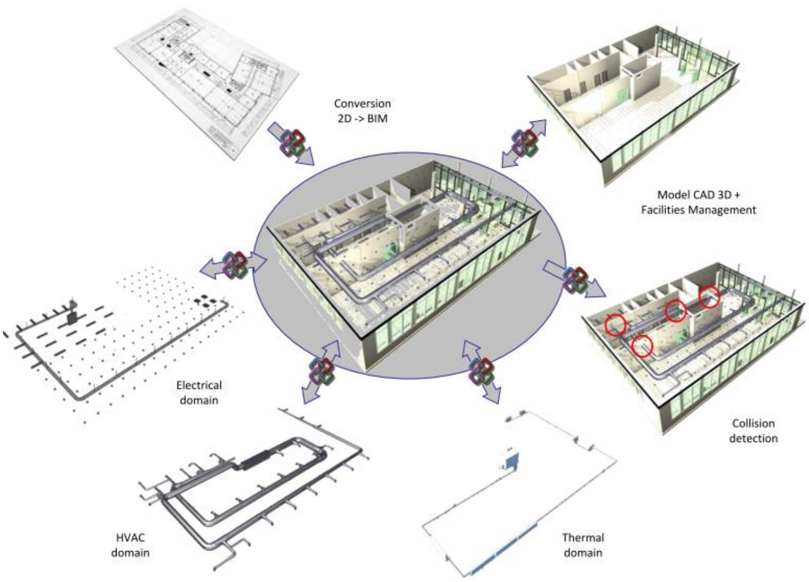


Figure 15: Integrated BIM-product (Beetz, 2016)

BIM leads to several advantages compared to the traditional design method. Firstly, 3D-models will be used which give more insight in the project. 2D drawings can be generated of this 3D model, which leads to consistent geometrical information. Furthermore, the stakeholders from the multiple disciplines make use of the same information, which leads to design efficiencies and a decrease of errors. The use of 3D-products leads to possibilities for the use of intelligent analysis software, such as clash control and model-check software which applies rules and calculations on the BIM-product to check the product on regulations and design errors. Also, might BIM lead to less (ideally none) rework and more reuse of information. (Eastman, Teichholz, Sacks, & Liston, 2011)

BIM ON THE CONSTRUCTION SITE

As stated in the introduction, BIM is mainly used in the process of designing and planning construction projects. During the construction phase, the BIM-products are flatted out and the traditional paper drawings, which are manually generated out of the BIM, will be used. The benefits that come with the intelligence and the visualization of the BIM-product are lost (Sacks, Radosavljevic, & Barak, 2010).

Recently, many researchers have tried to implement (a part of) BIM in the construction phase, by using new technologies, such as tablets (Meža, Turk, & Dolenc, 2014; Davies & Harty, 2013; Rebolj, Čuš Babič, Magdič, Podbreznik, & Pšunder, 2008; Park, Lee, Kwon, & Wang, 2013; Kimoto, Endo, Iwashita, & Fujiwara, 2005; Limpan, 2004), Augmented Reality (Uematsu & Saito, 2009; Van Berlo, Helmholt, & Hoekstra, 2009; Park, Lee, Kwon, & Wang, 2013; Madden, 2011; Meža, Turk, & Dolenc, 2014), and computers on site (Bråthen & Moum, 2015; Hewage & Ruwanpura, 2009). However these researchers speak of many advantages, many construction sites are not (yet) adapting these techniques. One of the main reasons are the high implementation costs. Also are these expensive tools sensitive for burglars and damage on the rough terrain.

Currently, tablet computers are making their way to the construction site. Many software vendors are selling advanced applications which can be used to track, monitor the construction process. These applications are mainly used for doing checks on quality and errors, not for improving the actual construction work.

4.2 OPEN BIM

OPEN and closed BIM are about the possibilities of sharing and using the BIM-product. The closed BIM method required all stakeholders to work with similar proprietary formats, such as Autodesk Revit's '.RVT' format. To join the BIM process, all stakeholders are required to use similar software to share and use the BIM-product. This is a major disadvantage, since a lot of specialty disciplines (steel engineers, mechanical engineers, window fabricators, etc.) use specialty software, designed for their daily processes. On the other hand, the OPEN BIM method makes use of open standards. As a result, all specialty disciplines can use their 'own' software and share and collaborate by exporting and importing the open standards.

BuildingSMART is the driving organization behind open, international standards in the AEC-industry. So far, buildingSMART has introduced 5 open standards, which will be introduced shortly.

1. Industry Foundations Classes (IFC) ISO 16739
2. International Framework for Dictionaries (IFD) ISO 12006-3
3. Information Delivery Manual (IDM) ISO 29481 ISO 29481-2

These three open standards are directly related to the BIM-product and BIM-process and designed in a triangle (Figure 16). The official buildingSMART definitions, which will be used in this research project are:

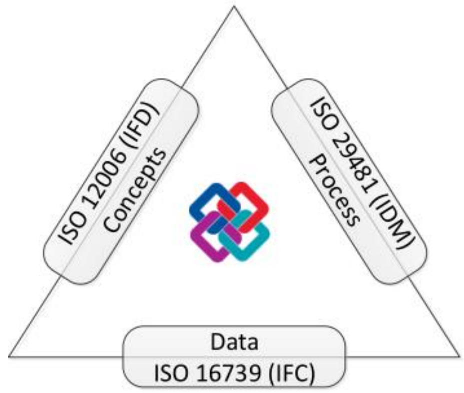


Figure 16: BuildingSMART open standards (Beetz, 2016)

IFC Industry Foundation Classes

"Industry Foundation Classes (IFC) is all about the sharing of information between project team members and across the software applications that they commonly use for design, construction, procurement, maintenance and operations. Data interoperability is a key enabler to achieving the goal of a buildingSMART process. BuildingSMART has developed a common data schema (IFC) that makes it possible to hold and exchange relevant data between different software applications." (buildingSMART, 2017)

IFD International Framework for Dictionaries

"The Data Dictionary is one of the core components of the buildingSMART technology. The bSDD is a reference library based on the IFD standard and intended to support improved interoperability in the building and construction industry.

The bSDD provides a flexible and robust method of linking existing databases with construction information to a buildingSMART based Building Information Model (BIM)." (buildingSMART, 2017)

IDM Information Delivery Manual

"BuildingSMART processes (IDMs) capture (and progressively integrate) business process whilst at the same time providing detailed specifications of the information that a user fulfilling a particular role would need to provide at a particular point within a project.

To further support the user information exchange requirements specification, IDMs also propose a set of modular model functions that can be reused in the development of support for further user requirements." (buildingSMART, 2017)

In short, IFC is an open standard for exchanging BIM-product data. IFD is a dictionary to improve communication by giving meaning to BIM-product objects. IDM captures the process to make sure the BIM-product has relevant information which can be used by other parties.

The other two standards are used to improve communication between different stakeholders in the BIM-process:

4. Model View Definition (MVD)
5. BIM Collaboration Format (BCF)

MVD Model View Definition

"Model View Definitions (MVDs) define the subset of the IFC data model that is necessary to support the specific data exchange requirements of the AEC industry during the life-cycle of a construction project.

A Model View Definition provides implementation guidance (or implementation agreements) for all IFC concepts (classes, attributes, relationships, property sets, quantity definitions, etc.) used within a particular subset. It thereby represents the software requirement specification for the implementation of an IFC interface to satisfy the exchange requirements." (buildingSMART, 2017)

Since the BCF standard is not relevant within this research project, it will only be discussed shortly. BCF is the open standard for communicating messages about the BIM-product objects, for example clashes or object-related notes.

4.3 BIM-PRODUCT

Since the BIM-product, based on the IFC file format, will be the input for generating the TOWI, this topic requires additional information. To be able to filter the information demand for the TOWI, the structure of IFC should be known.

Currently, IFC is the open data model for exchanging BIM-products. The IFC file format has had several iterations during its development. Presently, the IFC4 is the most recent iteration; however, IFC 2x3 is the most widely used version (Figure 17) (Borrmann, Beetz, Koch, & Liebich, 2015). The IFC file format focuses mainly on the exchange of building-related BIM-products. In future versions, infrastructure related classes will be added. IFC5 is currently being developed.

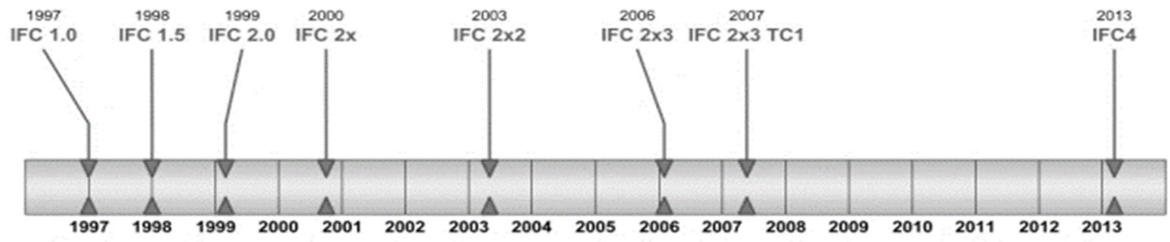


Figure 17: IFC-development timeline (Borrmann, Beetz, Koch, & Liebich, 2015)

IFC is designed with EXPRESS, which is a data modeling language to define object-oriented data models (such as IFC). The IFC file format follows the Standard for the Exchange of Product model data (STEP) procedure, which declares the procedure for product data modeling. Internally, objects are connected to each other by using relationships, and inverse relationships.

To elaborate, the IFC data model is a hierarchical tree model with both downward as upward relationships. On Figure 18, a basic hierarchical model of IFC is shown. This model makes the connection, between for example a IfcWindow and the IfcBuildingStorey possible. The Ifc-tree has IfcRoot as its starting point.

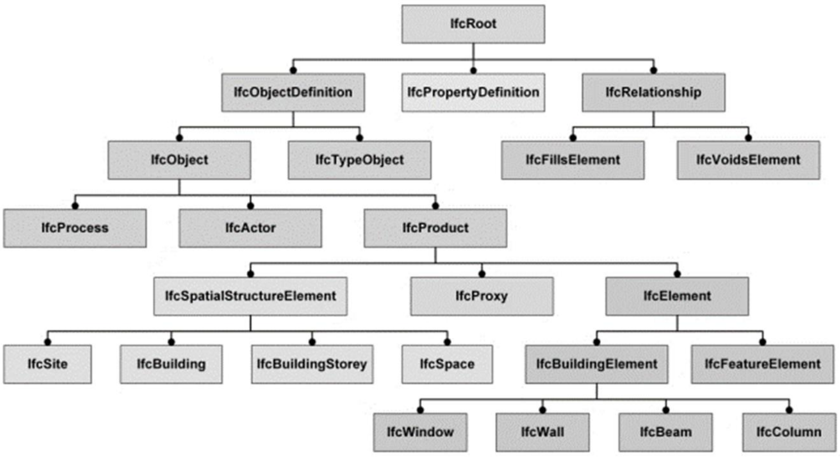


Figure 18: Hierarchical model IFC (Borrmann, Beetz, Koch, & Liebich, 2015)

The connections between the different classes (e.g. IfcDoor and IfcBuildingStorey) are made by relationships. On figure 19, a graphical scheme of an IfcDoor and (some of) its relationships with connected classes is shown. The connections are made by intermediate objects, which describe the relationships itself (e.g. HasProperties, OverallHeight, OwnerHistory, RelatedObject, RelatingPropertyDefinition). These intermediate objects are 'keywords' to navigate through the data model.

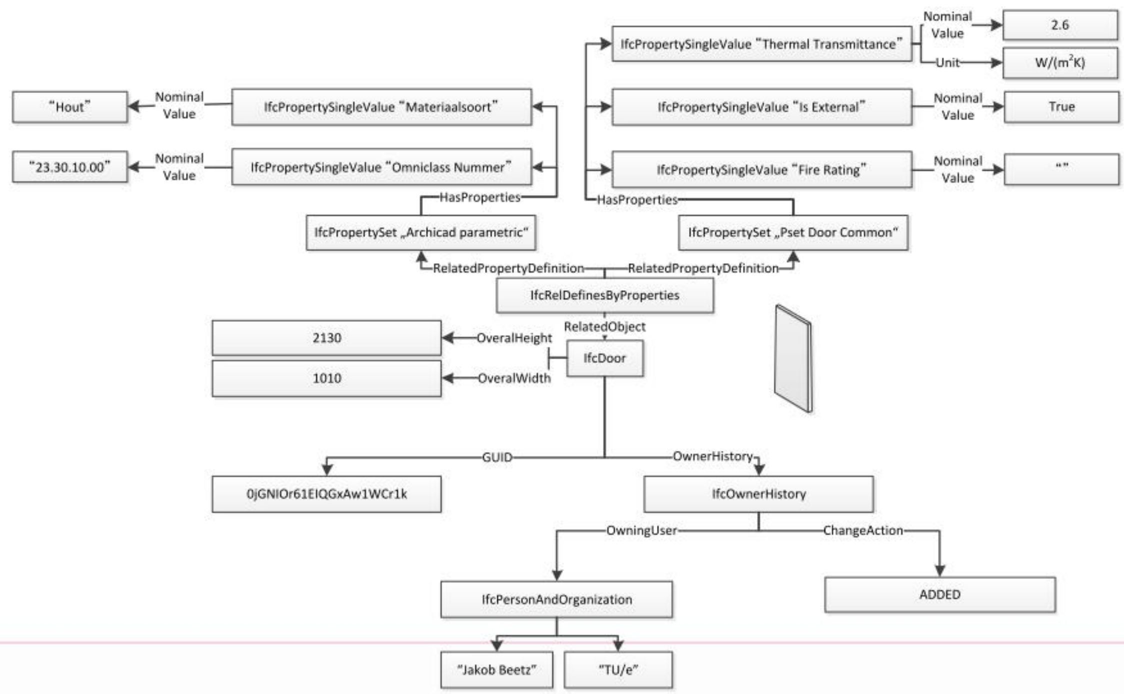


Figure 19: Graphical relationship model of an IfcDoor (Beetz, 2015)

Geometrical positioning in IFC is based on a local coordinate-system. Objects, such as walls, windows, doors, etc. have their coordinates placed in a local coordinate system of the related IfcBuildingStorey. The IfcBuildingStorey is located in the local coordinate system of the IfcBuilding, and so on.

The objects have either two-dimensional (IfcAxis2Placement2D) or three-dimensional coordinates (IfcAxis2Placement3D). Both two-dimensional coordinates, and three-dimensional coordinates are connected to both its local coordinate systems, as its parent coordinate system.

Three-dimensional coordinates also have a Z-axis, which is connected to a referenced direction (RefDirection). The referenced direction is the basic XYZ-axis (0,0,1), which targets upwards. The actual direction of the object is related to this reference direction and defines its actual direction. Figure 20, illustrates this concept.

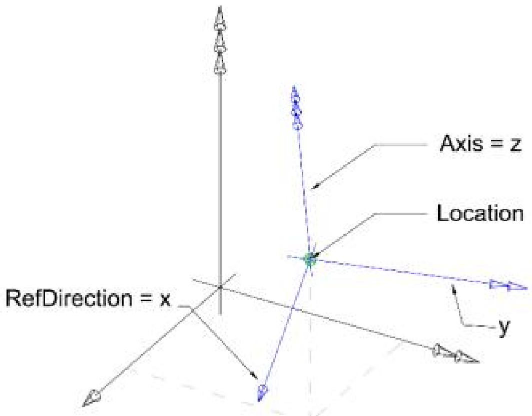


Figure 20: 3D-orientation IFC-object (Borrmann, Beetz, Koch, & Liebich, 2015)

NON-GEOMETRICAL INFORMATION

Besides geometrical information, is IFC able to handle intelligent (semantic) non-geometrical information. IFC-objects can have labels, names, properties, owners, processes, etc. This information is connected on a similar manner by making relationships. This non-geometrical information can be all kinds of data-types (Paragraph 3.4), such as strings for the object name and reals for the thermal transmittance of an object. Advanced BIM-modeling applications such as AUTODESK Revit, and Graphisoft's ARCHCAD can add unlimited extra properties to IFC-objects to increase the amount of information and support decision making.

More recent developments are applying linked data. Linked data is the term for linking data between different databases, to connect information e.g. PDF documents to IFC-objects. By linking data, the BIM-product might be able to encapsulate the entire integrated information model. Additional meaning to the objects can be created by linking the objects to an IFD-based dictionary.

These additional possibilities of IFC, makes the format appropriate to generate the information for TOWI.

BIM-PRODUCT INFORMATION WINDOWS

Since within this research project the building object 'window' is researched. The data for this specific object is researched. Several BIM-products were researched. Currently there is a wide range of modeling methods, and there are a lot of different approaches and methods used to reach the same results. For example, windows can be modeled as a window (which sounds logic), as a curtain wall or as 'generic model'. The result of all three methods might look the same, but the system (software) has a different understanding and treatment of the object. This results in wrong interpretations and information loss when exporting to the models/objects to the IFC file format. Firstly, they might be classified in the wrong class (e.g. IfcCurtainWall instead of IfcWindow). Secondly, the object might be exploded into small objects which leads to incorrect information handling, automation possibilities and use of semantic intelligence.

Besides, errors in the method of modeling, many different properties are available, which are related to the window objects. However, this is not an issue, but makes the models unnecessary large, and decreases the information overview. Three different window objects of window suppliers were analyzed and contained the following properties:

Object related properties:

Property:	
Length	Weight
Height	Sill height
Width	Frameoffset - external
Type mark	Fire Rating
Frame type	Acoustic Rating
Depth	Wind Load resistance
Material	Water tightness
Color(s)	Sound insulation
Direction	Burglar resistance
Handle height	Coating
Producer	Thermal resistance
Object GUID	Heat transfer coëfficiënt
Visual light Transmittance	

Location related properties:

Property:	
Floor	Distance
Bottom height (vertical position)	Depth
Upper height (vertical position)	Inner/outer

Process related properties:

Property:	
Schedule	Manufacturer
Supplier	Cost
Contact person	Youtube clip
Installation instructions	Classification number
Certificate	Mechanic
Technical description	(sub-)Contractor
URL	Assembly time

All of these properties can be relevant for some stakeholders. To prevent that the data models become too large, the information needs of objects should be decided before modeling the BIM-product. A well-functioning method to capture and communicate the information needs is to make use of an IDM (Paragraph 4.1).

Windows should be modeled as a Window object (not curtain wall, or other convenient object) to decrease the loss of information and keep the objects intelligent and useful, because IFC has object specific attributes. For example, a window has the attributes OverallHeight and OverallWidth. A IfcCurtainWall does not have these object specific entities. So, a window modeled as a curtain wall, will lose the extra intelligence.

PART C: AUTOMATION FRAMEWORK

This section gives an in-depth description of the BPA method that is used within this research project to approach the automation of TOWI. A short introduction was given in the introduction.

5. BUSINESS PROCESS AUTOMATON

As described in the introduction, this research project aims to automate the process of creating TOWI. To approach such a project, several software development processes are analyzed, to create an automation framework. In chapter 6, the framework will be implemented and tested on a real-life scenario.

5.1 INTRODUCTION

Business processes are all kinds of workflow processes that deal with the stakeholders of a company or project. Executing these processes consume time and resources (e.g. employee hours). Automating these processes is an effective approach for improving business efficiency and productivity by reducing the time and resources (Thomson, 1996).

The workflow of a business process consists of tasks (e.g. activities) and transactions (e.g. flows) between tasks. Executing the business process is characterized by executing the tasks in the process by following the sequence given by dependencies and information flows between tasks (http://www.wfmc.org).

Automating these business processes requires a standardized approach, since customizing the automation for every task, and/or company has proven to be a time consuming and expensive operation (Shi, Lee, & Kuruku, 2008). The AEC-industry is characterized by many non-standard projects and processes. Therefore, structure and standardization is important for this research project. Designing and defining a standardized approach is therefore necessary for the success of the automation of TOWI.

Shi,Lee, and Kuruku (2008), use a Task Based Modeling (TBM) approach to identify, model and standardize business processes. The TBM approach focuses on lower management tasks where similar tasks are required. For example, a purchasing process for construction products requires the following tasks: search for qualified suppliers, request for product offers, compare the offers, and place a purchase order. Performing the actions of management tasks, can either be simple or complex process.

In general, the process of modeling the business process, exists of the following steps (Shi, Lee, & Kuruku, 2008):

1. How to define a task as a generic component so that a business manager can understand it and use it for constructing business process models?
2. What is the data structure for the task component to carry specific management information and to communicate with other task components in the same process model?
3. How to create a business process model by using the provided task components?
4. How to measure the outcome of a task component?

These four questions can be summarized in an iterative process model (Figure 21) with four process steps: define task, configure components, execute process, and measure outcome. After measuring the outcome, the task can be redefined to improve the automation process and outcome.

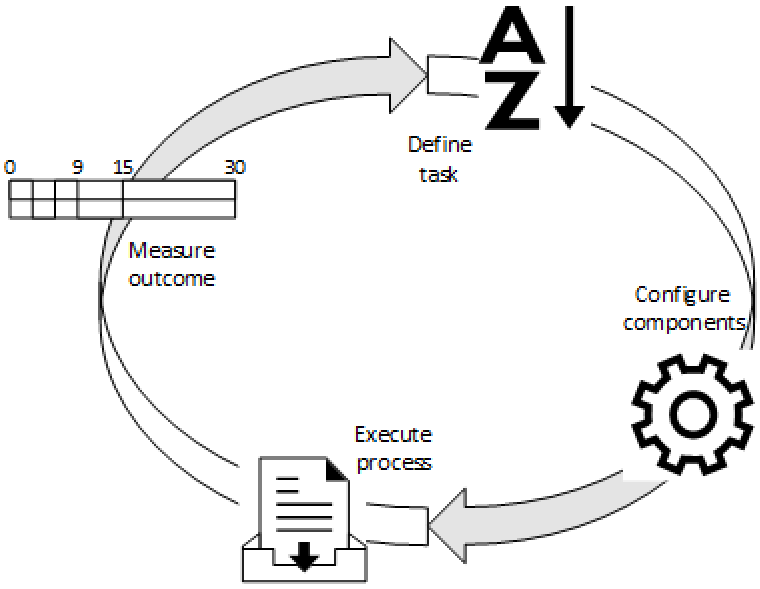


Figure 21: Process model Task Based Modeling (own figure)

The researchers (Shi, Lee, & Kuruku, 2008) used TBM to create reusable software components which can be used to automate the business processes. The process of automating TOWI will use a similar approach; therefore, their method and procedure will be used in this research project. Next, every step of this model will be discussed in-depth.

5.2 DEFINE TASK

The first step is to translate the business processes into understandable tasks. The process of TBM starts with the definition of a task. A task consists of the following features, which are summarized on figure 22:

- Action
- Method
- Object

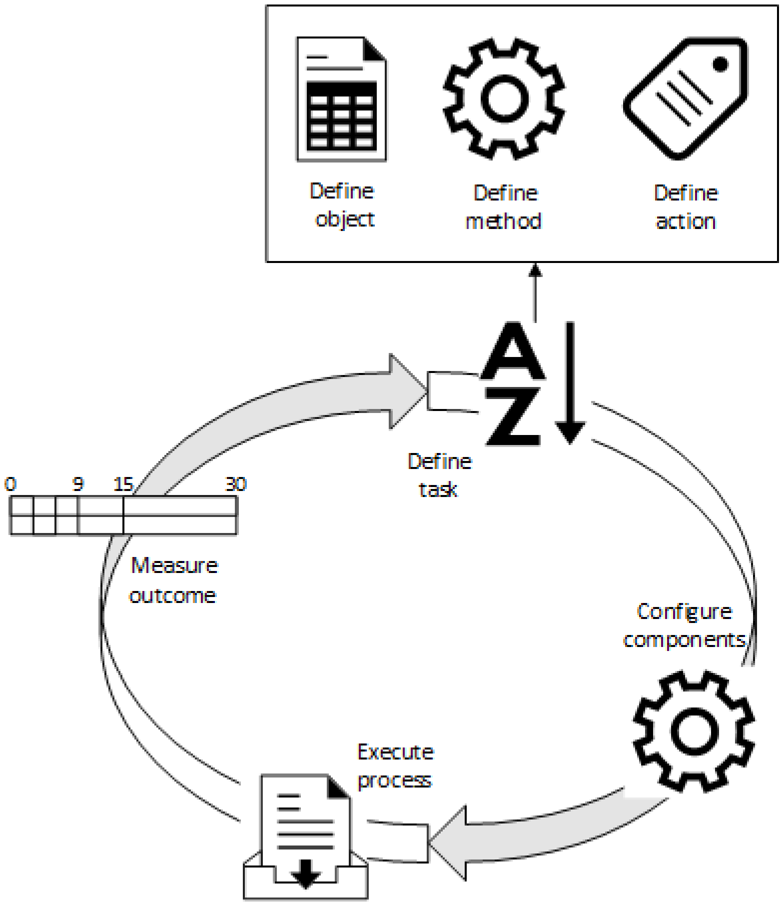


Figure 22: Define task

5.2.1 DEFINE ACTION

The definition of a task starts with the expression of an action. An action is characterized by a ‘Verb’. For example, in the task ‘send file X to stakeholder A, by mail’, the action which has to be taken is ‘send’. The Verbs are the keywords which describe the actions for the business processes. Other common actions in business are: analyze, schedule, calculate, plan, compare, determine, decide, make, design, create, etc.

For the TBM approach, it is necessary to identify the actions which are essential for performing the task. Next, these tasks have to be transformed in reusable software components.

INFORMATION NEEDS

To decide if the actions are essential for performing the tasks, the information demand has to be identified very specifically. Devadason (2008) has studied the process of identifying the information needs. He declares that identifying the information demand is a complex and diverse process which has to be tailored to a specific situation. The identifying process has to be modified to the type of organization, situation, manpower, resources and the type of persons. In short, he designed the following top-down process:

1. Study of the subject(s) / field(s) of interest to the organization/ user(s)
   e.g. dictionaries, encyclopedia, historical development, organizational sources.
2. Study of the organization and its environment
   e.g. objectives, functions, policies, technological environment.
3. Study of the immediate environment of user(s)
   e.g. unit background, organizational structure, information flow, information sources, recent problems, ongoing project.
4. Study of the user(s)
   e.g. type of user, user role, role activities, information flow, personal information, job description, user's publications, diary records.
5. Formal interview
   e.g. draft information needs, type of information services, existing information sources, unused information sources.
6. Identification and recording of information needs
   e.g. kind of information, information form, coverage of information, quantity of information, priorities.
7. Analysis and refinement of information needs
   e.g. feedback, examples, discussions.

Devadason comments, that it is not necessary to follow every step of the process, but should be tailored to the situation. Besides this, he mentions that it is not necessary to study the information needs of every user involved, but only a few qualified people (3-5). (Devadason, 2008)

5.2.2 DEFINE METHOD

The method of an action, defines how to perform that action for the specific needs. For example, in the task 'send file X to stakeholder A', the sender can use multiple methods to accomplish the task. The sender can use mail, e-mail, a memo, direct messaging, etc. to send the file to the receiver. To be able to automate the task, the method must be defined. The action as described in 5.2.1, can be supplemented with a method: Verb_method. The method is usually defined by using keywords such as, using, by, with, through, etc.

5.2.3 DEFINE OBJECT

The third part of defining a task is to define the object. The object is the item on which the action and method have effect. For example, in the task 'send file X to stakeholder A, by mail', 'file X' is the object. The action and method can be supplemented with the object: Verb_object_method

DETAILING OBJECTS

In many cases, the three steps verb, object, and method are not detailed enough to model the business process. Therefore, two additional detailing steps can be taken, namely: ‘What’ and ‘WhatElse’. Supplementing these steps to the syntax results in: Verb_object_method[What, WhatElse]

‘What’ is used to detail the direct object of the action. For example, in the task ‘send tenders to stakeholders’, the object ‘tenders’ may contain of multiple items, such as, a bid, a contract, a letter, additional information, etc. ‘What’ describes the detailed information that should be sent.

‘WhatElse’ is used to provide further information for detailing the indirect object of the action if applicable. In the example ‘send tenders to stakeholders’, ‘WhatElse’ describes details on the additional information, which are in this example the stakeholders.

Using the ‘What’ and ‘WhatElse’ makes it possible to reuse the components and perform the same task in different processes and actions.

5.3 CONFIGURE COMPONENTS

When the business processes are defined in such manner that they can be transformed into software components to make the automation possible, additional choices and checks are required. Firstly, the level of automation has to be determined. Secondly, the availability of input and possibility of output has to be checked. Thirdly, the components have to be set up and should be saved in a library. This part is summarized on figure 23.

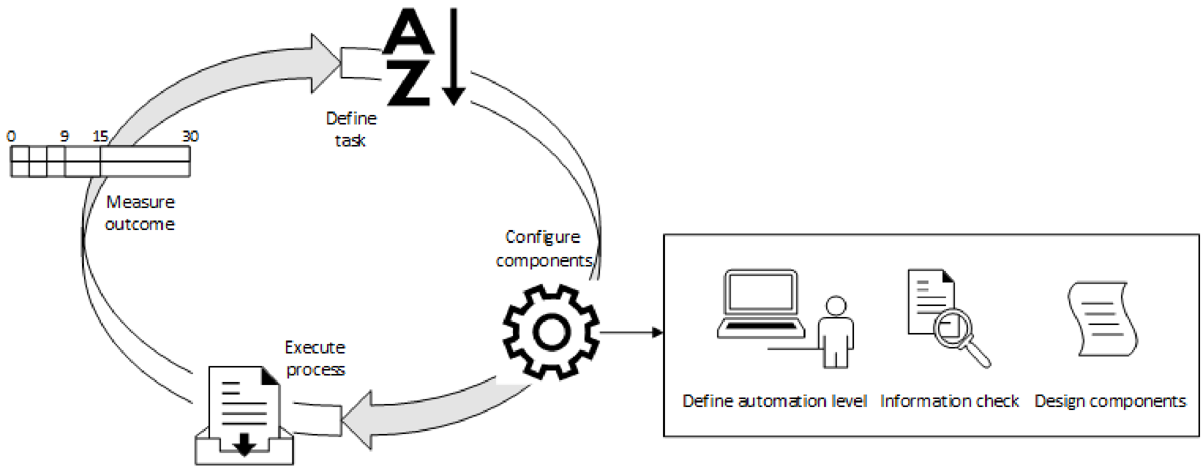


Figure 23: Configure components

5.3.1 DEFINE AUTOMATION LEVEL

Business processes are designed and executed by humans; however, with today's technology, many routine tasks can be automated. Still, many tasks require human involvement or decision making. To create the automating software components, the first step is to define the level of automation which is desired. There are three levels of automation, namely:

- Manual – A manual process is involved;
- Semi-automation – User's decision or confirmation is needed;
- Full automation – User's involvement is not needed.

MANUAL

The first, and lowest level of automation involves human interaction for a task to be automated. Some tasks are difficult to automate (e.g. creative processes, discussions, negotiation, actual situation), and therefore require human interaction. The results, however, may be important input for other tasks which can be automated. The results can be entered into the system manually (e.g. initial design, expected schedule, negotiated numbers, actual measurements), by using a dialog box for example.

SEMI-AUTOMATION

The second level of automation involves some human interaction for a task to be automated. This is the case when tasks need to be confirmed or need a decision before the task can be performed. In the case of semi-automation, the computer executes a task and shows the output to the user. The user, can either approve, adjust or dismiss the output, based on his own expectations or calculations.

FULL AUTOMATION

The third level of automation requires no human interaction. A task will be fully conducted by a computer. The computer program must be instructed on what input the action must be performed, and the task will be executed. For example, for every row in database x do something. In this case, 'database x' is the input that should be instructed to the computer program. If 'database x' would always be the same, it could be programmed in the procedure, but makes the procedure dependent.

The level of automation has to be decided early in the process of developing software components, to make sure the user(s) is able to control the information flow.

5.3.2 INFORMATION CHECK

Besides the automation level, the in- and output is very important, since this might be different within every company. In paragraph 5.2, the definition of a task was discussed. One part was to identify the information needs. The information needs should be equal to the output of the component(s). For example, if the project manager of a hospital requests information about the progress on the construction project, the output should contain scheduling information and information about the associated building objects.

To be able to automate the output, the input should be correct and consist of that information. Firstly, the input should be digitally comprehensive (e.g. Scheduling information in Excel sheets instead of written on paper, correct format). Secondly the input should be structured and able for a computer to read (e.g. digits instead of handwritten numbers). Thirdly, the information should be complete, to be able to generate complete output.

When the input information requirements and output information requirements are known, software components which make the automation possible, can be designed. To improve the output, the input information requirements can be adjusted. So, this process is different from the standard input-process-output (IPO) model (Figure 24):

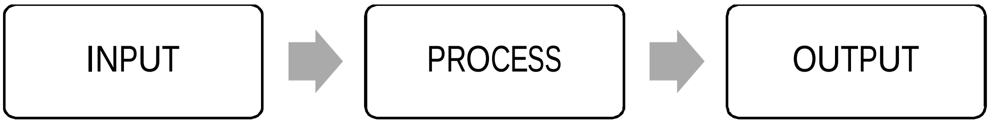


Figure 24: Standard IPO model (Grady, 1995)

For the business process automation approach, the focus is on the output (Figure 25). Firstly, the requested output should be identified. Next, the input should be checked to make sure the output can be automated by a computer. When valid, the software component(s) can be designed, utilized and improved.

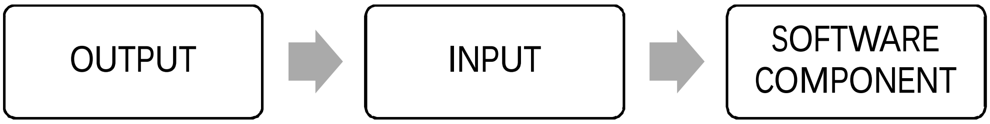


Figure 25: Automation component model (Own figure)

5.3.3 DESIGN COMPONENTS

When the task is defined, the automation level, output and input are known, the software (automation) components can be designed. Components are small parts of a piece of software that perform a single task. In programming language these components are often named functions. For example, a component to calculate the average of sum (X1+X2+Xn…), could be:

```
def calculate_average(sum,n):
    return sum/n
```

This component requires the input ‘sum’ and ‘n’, which may be determined by manual input, or other components. The words ‘sum’ and ‘n’ are called variables which can have any value, such as numbers, strings or even formula’s. However, the component ‘calculate_average’ requires numerical input since the components uses a division. A string would lead to a type error, since in this case a string cannot be divided by a number.

Components can be created for many different kinds of functions, such as mathematical equations, opening files, editing files, display information, etc. An efficient way of designing components is to make them reusable. This means that components are created for performing specific tasks where the input variables change, so the components can be used for multiple instances instead of 1. This method is also called Component-Based Development (Herzum & Sims, 2000).

To elaborate, the components are designed based on the definition of the task (Verb_object_method[What, WhatElse]). Where the 'What' and 'WhatElse' are assigned to the input variables.

The method described above is used to create the automation components for the application.

5.4 EXECUTE PROCESS

Once it is known how to define task and create corresponding components the process can be automated. To do so, the first step is to identify the process that has to be automated. A process is driven by a request, problem or need. To fulfill the needs, a process has to be performed. In general, such process involves the following steps: 1) raise a request; 2) instantiate an instance of a process model; and 3) execute the model task by task, and trace the dynamic status of the model (Shi, Lee, & Kuruku, 2008). This process is summarized on Figure 26, and will be discussed in-depth in this paragraph.

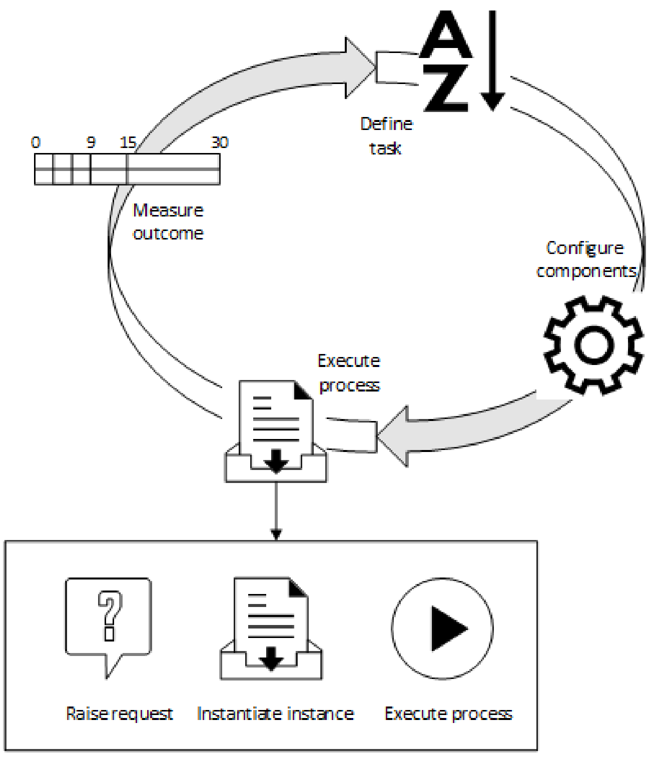


Figure 26: Execute process

5.4.1 RAISE REQUEST

Business processes are designed to fulfil the needs of a user. These needs can be a variety of information or solutions. In the definition of a task (paragraph 5.2) the information needs of the user are identified.

A user might be assigned to multiple (automated) tasks. Therefore, the user must be able to request the correct information corresponding to the to be performed task. The correct process model should be selected. To do so, a selection menu can be used.

5.4.2 INSTANTIATE INSTANCE OF PROCESS MODEL

When the request of a process model is completed, an instance of the process model can be created. The process model is a depicted version of the procedure which has to be executed to get the requested output. Multiple users with different responsibilities might be involved in developing the process model.

Process models are well known schema's for depicting business processes and designing software. These can be made from either actual processes or desired processes. To make sure industry and researchers are able to understand the processes, standardized techniques should be used. Well known techniques are Business Process Modeling and Notation (BPMN), Unified Modeling Language (UML), workflow diagrams and Program Structure Diagrams (PSD).

BPMN is an extensive method specifying business processes in a graphical manner. The objective of BPMN is to support both business as technical users to represent complex processes in an intuitive manner. UML is a ISO-certified modeling language with the purpose for software engineering. UML has a set of graphical notations to create object-oriented software systems. Workflow diagrams represent processes in a workflow and answer a given problem. Flowcharts are mainly used for simple processes and is a less extensive method for projecting processes. PSD uses a breakdown structure to depict processes in connected tree's. These structures are mainly used to show relationships between modules.

Since the goal is to identify and create workflows of information and tasks with its actions, methods, and object, workflow diagrams are used. An example of a business process depicted in a flowchart diagram can be seen in Figure 27.

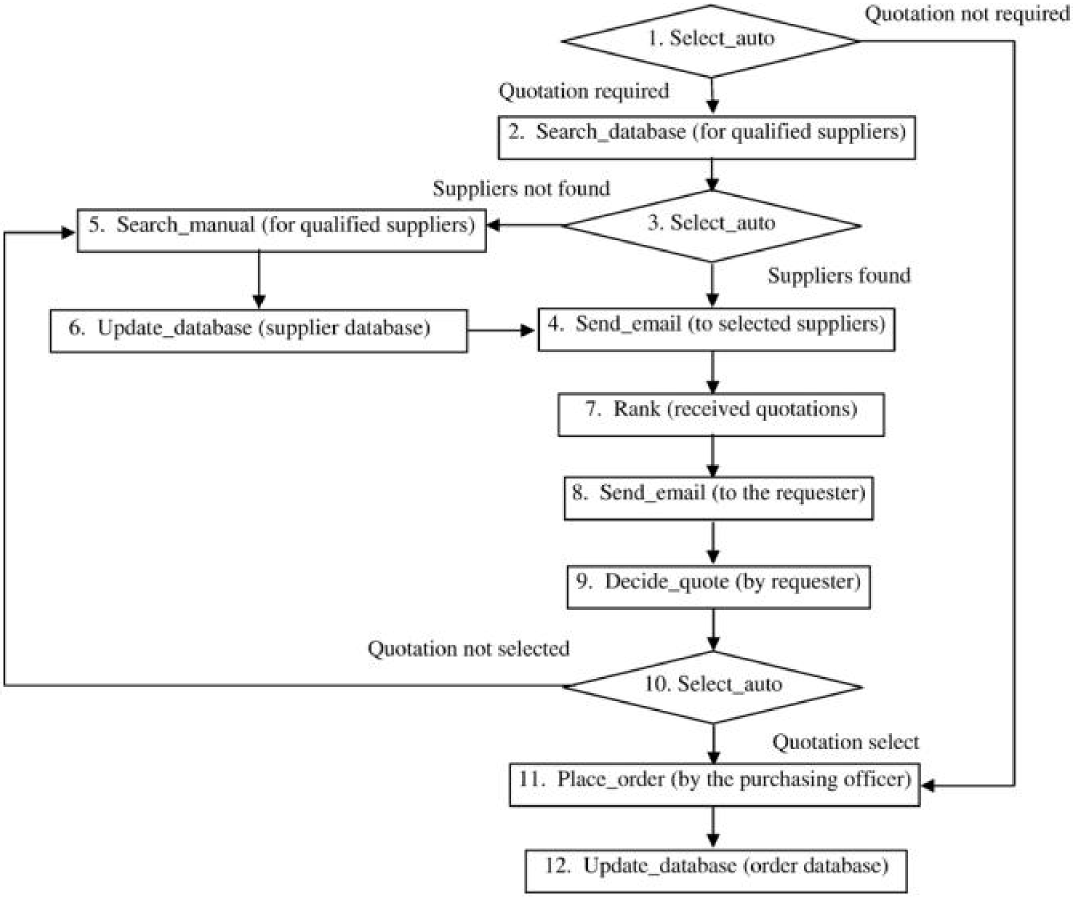


Figure 27: Flowchart diagram of business process 'purchasing order' (Shi, Lee, & Kuruku, 2008)

Figure 27 shows an example of a flowchart which is used as an example for the study of business process automation by Shi, Lee & Kuruku (2008). For every step in the process 'purchasing order', they defined the task (Action, Method, Object, What and WhatElse) and its automation level. Three examples are:

- Search_manual — a manual task. It involves a manual process to search for a list of qualified items;
- Send_email — an automated task. It sends an email with given content to the specified recipients;
- Rank — a semi-automated task. It requires a responsible person to manually rank a given list of items in an order; (Shi, Lee, & Kuruku, 2008)

This information can be shown into a table, as shown in Table 9, to create more overview for creating the automating software components.

Table 9: overview tasks purchase order (Shi, Lee, & Kuruku, 2008)

Task information and actions

Task code	Action	What	WhatElse	Description of action
1	Select_auto	2: QuotationRequired = yes 11: QuotationRequired = no	Nil	Select a route based on whether or not quotations are required
2	Search_database	Product: cement |delivery<RequiredDate	Suppliers — database name	Search the supplier database for qualified suppliers
3	Select_auto	4: QuaSuppliers>=1 5: QuaSuppliers<1	Nil	Select a route based on whether or not qualified suppliers are found
4	Send_email	"Request for quotations" & Product & Quantity & RequiredDate & ResponseDate & ContactPerson	QuaSuppliers	Send a message "Request for quotations" to the qualified suppliers
5	Search_manual	NewSuppliers	Nil	The purchase officer finds new suppliers and enters them to the system
6	Update_database	NewSuppliers	Suppliers	Update the database with new suppliers
7	Rank	ReturnedQuotations	Nil	The purchase officer ranks the returned quotations
8	Send_email	RankedQuotations	Requester	The ranked quotations are sent to the requester
9	Decide	RankedQuotations	Nil	The requester makes a decision on the acceptance of a quotation
10	Select_auto	11: SelectedQuotation = true 5: SelectedQuotation = false	Nil	Select a route based on whether or not the requester accepts a quotation
11	Place_order	SelectedQuotation|Product|Quantity	SelSupplier	The purchase officer places an order
12	Update_database	PO#|Product|Quantity	Orders — a database name	Update the order database

As you can see, the process of a purchase order consists of manual, semi-automated and automated components. This is due to the complexity of the process and the involvement of human decisions. Only some components have to be automated, where other components require human decisions. With the level of automation, processes can be automated to the limit as they are wished to be automated without losing control of the flow.

5.4.3 EXECUTE PROCESS

When the process model, its tasks and software components are designed and developed, the last step is to execute the process. The process is usually executed in the sequence of the process model. Depending on the requirements of the input and output software to execute the process has to be used. The data type (e.g. integer, string, Boolean) is important to know how it should be processed by the software.

Components are executed by a compiler. First the front end will be analyzed to make sure the components do not consist of errors and are executable. This process starts with a lexical analysis, which breaks the source code into smaller pieces to identify keywords, operators and variables. The next step is syntax analysis, which checks the grammar and structure of the components. The final step of the front end is the semantic analysis, which checks if the input of the variables is of the correct type, according to the lexical analysis. Secondly, the back end will be executed by synthesizing. The back end starts with allocation of the values of in the register. Then the code will be optimized, by transforming the values into faster/smaller, but equal forms. Finally, the to be executed code will be transformed into machine language and will be executed. (Wikipedia, 2017)

5.5 MEASURE OUTCOME

The last step in the BPA process is the measurement of the outcomes and its continuous improvement. Therefore, three additional steps have to be followed, namely user feedback, improve outcome and update library (Figure 28).

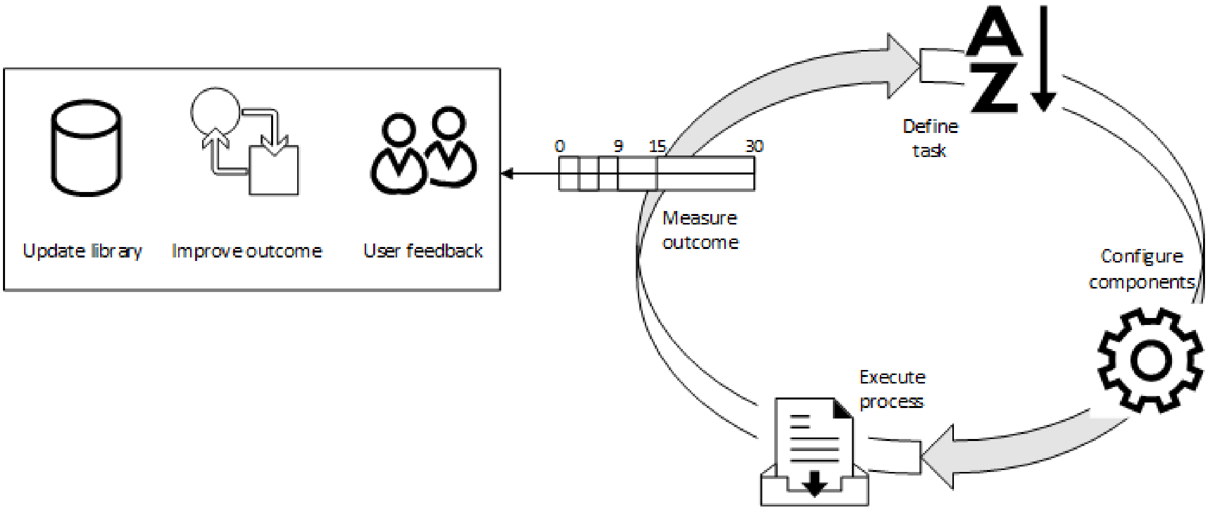


Figure 28: Measure outcome

5.5.1 USER FEEDBACK

User feedback is important to improve the process and the results of the automated process. Both software components can be improved, as the whole process by adding, changing or removing process steps.

To do so, the results have to be verified by competent actors, but preferably the actual users of the output, since they have to use the information. Before using the information, the users should give feedback on the completeness, clarity, usability and relevance. The actual (initial) output should be used as input for the feedback session. The feedback session should be adapted to the situation and users. Some output prefers written feedback, while other output requires a discussion or verbal feedback. For example, open- or semi-open interviews are useful, but time-consuming sessions to get feedback. The focus, however should be to increase the quality of the output.

While the output has been used, again, feedback should be collected. This is because unforeseeable information could have been missing while actually utilizing the output.

5.5.2 IMPROVE OUTCOME

After gathering feedback, the feedback should be processed. The feedback should be communicated with, or collected by the task designer who designs the process models and is involved with the creation of the software components. The feedback should be processed on either the process model, or the software components itself.

Besides this, standardized processes can be adapted to the information needs of the specific situation.

5.5.3 UPDATE LIBRARY

This paragraph discusses the component library which will aid the design of process models. First the library itself will be introduced, next a possible functioning of the library will be discussed, finally updating the library will be discussed. The library will only be discussed and is not implemented in this research project.

COMPONENT LIBRARY

Creating and designing software components for each project and task is like inventing the wheel. Many process steps require similar in- or output. Designing atomic automation components, makes them suitable and reusable. Collecting these software components in a database, makes them reusable for many projects and actors. These components can be used as basic building blocks for creating automated processes rapidly. Figure 29 shows this process, where the component library consists of many software components, which can be easily selected in the right sequence by the task designer to automate the process.

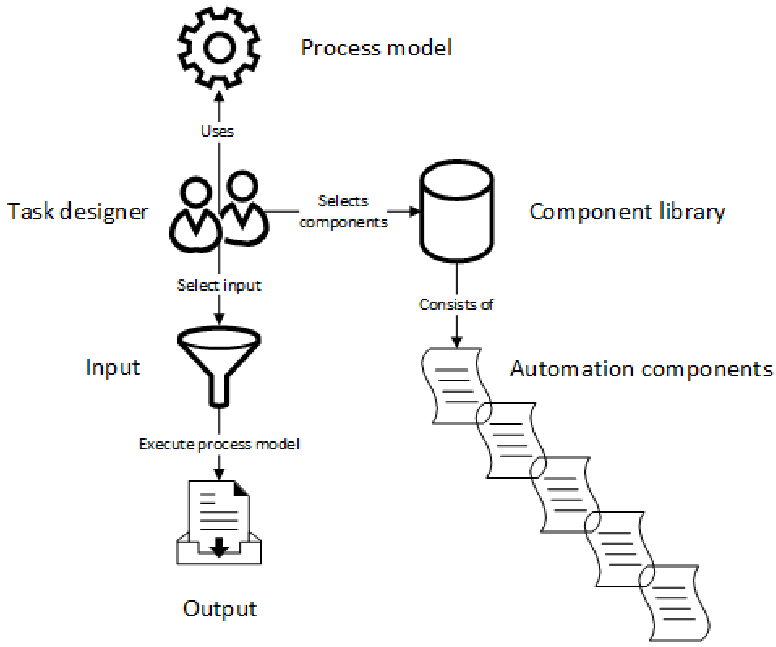


Figure 29: Component library

LIBRARY WORKFLOW ENGINE

A convenient way of using the software components and being able to rapidly create automated process models, is to make use of a workflow engine. A workflow engine is a piece of software within which the software components can be 'dragged and dropped' and can be linked and executed. Figure 30 shows an example of such software. Blocks of software components can be dragged to an empty process model. Manually, information can be selected if necessary. Next, these blocks can be connected to each other. By executing the workflow, the output will be generated.

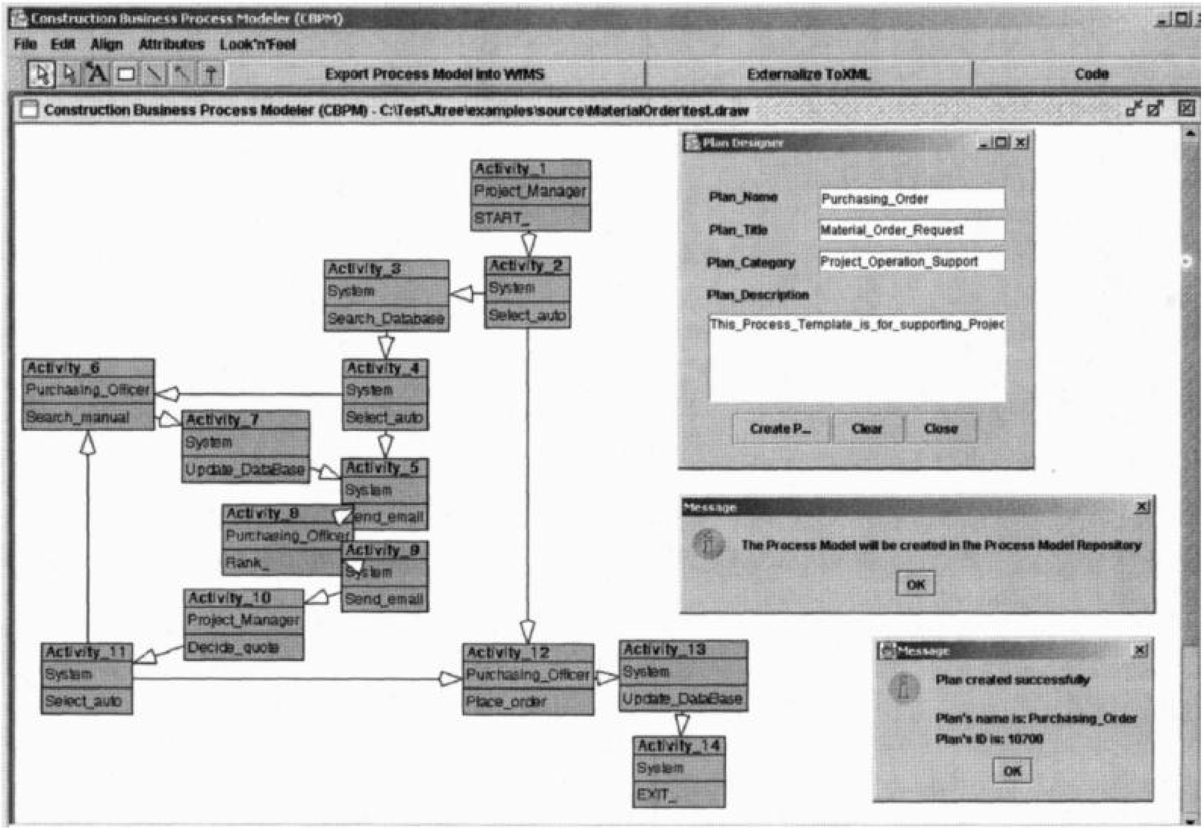


Figure 30: Workflow engine software (Lee & Shi, 2006)

Depending on the goals, an existing system could be used, or a new workflow engine could be developed, which uses the open software components. In the field of construction, Autodesk's Dynamo, is a well-known workflow engine which has plugins for BIM-modeling software Autodesk Revit, and this concept is also implemented in several Geographic Information System (GIS) software applications.

UPDATE LIBRARY

Collecting the software components and process models in the library, makes them able to reuse. By using the library as a growing database, the creation of automated process models will speed up. After several projects, the database will contain a lot more components, and automation will become easier. Besides this, the components have to be updated after feedback, and in case of changes in for example the database or software.

CONCLUSION

BPA is a process which exists of 4 main- and 12 sub steps. The goal of this approach is to create small components that (semi)automatically execute parts of a process. As a result recurring tasks can be (partially) automated, which increases efficiency and reduces human interaction. Shi,Lee, and Kuruku (2008) created an extensive process which can be adapted on all kinds of business processes. This research project has implemented this process and has applied it on the case of automatically creating TOWI. This implementation can be found in the next chapter (6). Chapter 7 discusses if the BPA approach was successful for this case and if it helped complete the goals.

PART D: IMPLEMENTATION

This section describes the creation, limitations and utilization (Chapter 6) of a working prototype based on the BPA-process (Chapter 5) for a case related to the AEC-industry. The case was selected early in the process and was the basis of this research.

This section aims to describe the process of creating automated solutions for the AEC-industry, more specifically automated solutions to create TOWI. This section aims to answer the following research questions:

- *How can the task-oriented information filter be defined, captured in a generic way (that is easy to be reused and adapted in other scenarios) and how will it be processed?*

- *How can the task-oriented information filter contribute to the construction workers, and how are the results reviewed?*

6. AUTOMATING TASK-ORIENTED WORK INSTRUCTIONS

As mentioned in Chapter 3 Task-oriented work instructions, one of the biggest issues of utilizing TOWI in construction projects is the labor-intensive task of composing the TOWI. The goal is to automate this process by applying the BPA process. A prototype application was developed to automate the process of composing TOWI for the task ‘placing and assembling window(frames)’ performed by construction workers on residential buildings by using the (open) BIM-product, as described in paragraph 2.2 case study.

To get a more in-depth understanding of the functioning of the prototype, a UML-use case diagram is developed. A use-case diagram is the description/visualization for the users of the application. The use-case describes the functioning of the application and its users in a graphical scheme and a descriptive table, as stated in ‘UML Distilled 3rd Edition’ by Martin Fowler (Fowler, 2003). The graphical UML use-case diagram of the proposed solution is shown in Appendix 2. The descriptive UML use-case description of the proposed solution is shown in Appendix 3.

6.1 DEFINE TASK

To automate the creation of TOWI, firstly the client has to be identified. In the case, the client is the construction worker (carpenter) who places and assembles windows(frames) in residential buildings. As described in chapter 2 (Information in construction) 6 double semi-structured interviews were conducted with construction performing the discussed task.

INFORMATION DEMAND

The information demand of the task ‘placing and assembling window(frames)’ are identified by performing interviews with construction workers related to this task. In paragraph 2.5, the information needs are summarized. According to Davadason (2008), 3-5 interviews with qualified actors are required to generalize the information demand. 6 interviews with 12 construction workers within 6 different companies were performed to generalize the information demand. Besides this, the information needs were practically equal at as well smaller as larger projects. Table 10 summarizes the information demand, its according BIM properties and the data types.

Table 10: Information demand, its properties and data type

Information demand	Properties	Data type:
Depth	FrameOffset - External	Length = real
Mark	Type Mark	Text = string
Orientation	-	-
Bottom height (base level)	-	Length = real
Mounting frame dimensions	-	Length = real
Horizontal dimensions based on grid	-	Length = real
Top height (base level)	-	Length = real

INFORMATION CHECK

Paragraph 4.3, discussed the availability of properties in the data of BIM-product window objects. When comparing the properties with the information demand, there can be noticed that only 2/7 of the properties are available (Table 10). This is the situation when the IfcWindow classification is used, and when these properties are exported to IFC (which should be set in the MVD, see Paragraph 4.2). By using other classifications, both available properties will be lost. Using an unexploded IfcWindow is therefore the first requirement which should be checked. Otherwise, the created components will not generate the correct results.

Orientation is a complex concept to encapsulate. Orientation is a function of the mind and depends on the capabilities of the person. Orientation can be provided by showing recognizable objects and a realistic scale of these objects. In the case of Window objects, orientation can be provided by showing the related objects (cavity, floors, structural walls, internal walls, stairs, etc.). By recognizing the positioning of objects, the needs of orientation will be fulfilled. Drawing these objects in the correct scale, also enhances the orientation. The amount of detail is not important for global orientation.

Mounting frame dimensions are important because these define the actual measurements for placing the window (frame). Since BIM-product modeling is relatively new in the AEC-industry, few companies are trained in modeling high detail models (LOD 350, or higher). As a result, the mounting frame is often not modeled, or exported in the correct class (IfcWindowLiningProperties). As a result, the information is not available in the BIM-product, and cannot be filtered. This information should be processed by the traditional way at this moment.

The construction workers use the grid for the horizontal dimensions. Therefore, the grid should be available in the BIM-product. With standard settings, the grid is not exported to IFC. In the MVD, the grid should be exported to IfcGrid. Without the grid, the horizontal measurements will not be automated.

The other two properties, bottom height and top height are related to the base level (Dutch: peilmaat). Data about these heights is related to their local coordinate system, not the base level, so this data cannot be used in a uniform setting. By using the global coordinate system, the correct data will be used.

To conclude, for generating TOWI for the aforementioned task, the status of the information is more important than the LOD. To elaborate, when certain information is final and will not change, a higher detail level is not relevant. Figure 31 shows a door in 5 different information levels (LOD 100, 200, 300, 350 and 400). At LOD 300, or even LOD 200, the dimensions, mark, and location are final. This is all the information which is needed by the construction workers. Extra information might be relevant to other stakeholders, but not the construction workers. The information demand is relatively low, and on a low level. So, the final status of the information is more important than the level of information. But higher information levels give more possibilities for automation.

Figure 31: LOD 100 - LOD 400 of a door (Hermans, 2015)

PROCESS OF AUTOMATING TOWI

Now the task, the user, the information demand, and the availability of (digital) information is known, the process to automate the TOWI can be designed. This process is based on the component design as discussed in Chapter 5. The sub steps of this process are merged due to its coherency.

Since, the construction workers are used to paper drawings, in the form of sections (including details), plan views and elevation views, both traditional forms of information, as new possibilities due to the use of 3D-models will be tested. The goal is to make automated sections (vertically and horizontally) of the three-dimensional geometry and make sure the information is complete with as little human interaction as possible. This resulted in the procedure (process model) as shown in Figure 32 and Table 11.

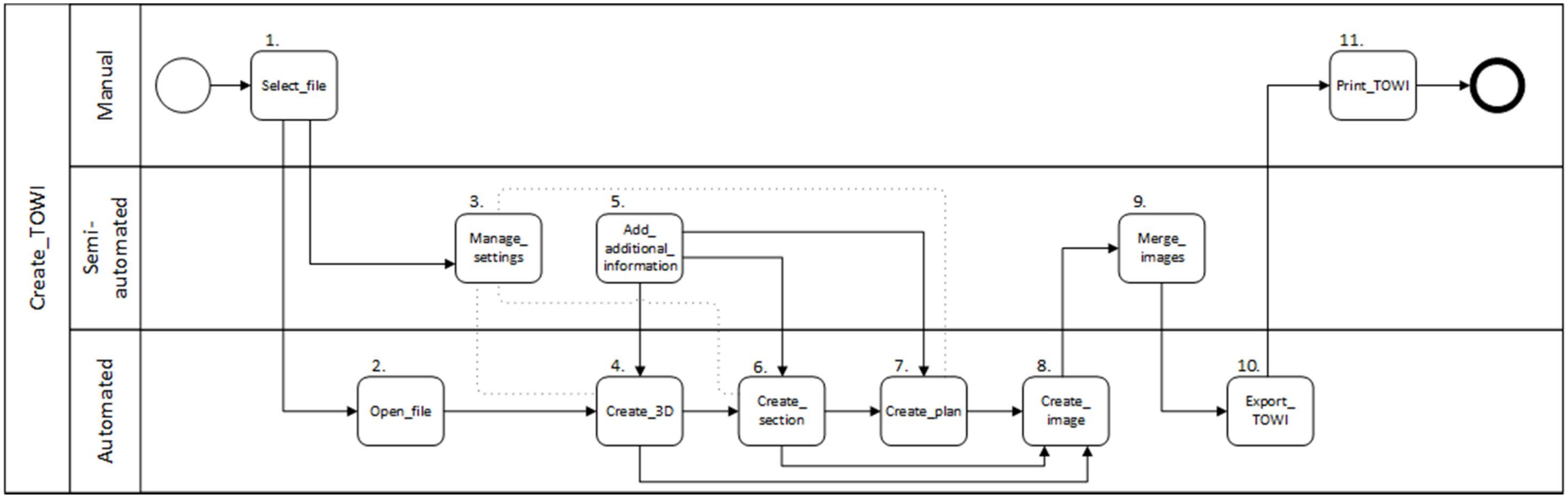


Figure 32: Process model automating TOWI

The process model has three ‘swim lanes’ which describe the level of automation which is related to the process step. The process starts with a manual process of selecting the IFC-model, and ends with the manual process of printing the TOWI. There are three semi-automated processes involved, which are ‘manage settings’, ‘add-additional information’ and ‘merge images’. These processes involve manual actions for selecting the additional information that should be presented, possibly changing settings, such as text sizes and colors, and manual actions for changing the layout of the output. The other processes can be fully-automated. Table 11, gives additional information about the process steps which are important for the component designer.

Table 11: Overview tasks create TOWI

Task code	Action	What	WhatElse	Description of action
1.	Select_file	IFC-file	Nil	Select the relevant IFC-file
2.	Open_file	1: Selected IFC-file	Nil	Open the IFC-file
3.	Manage _settings	4, 6, 7: Colors, line types, text sizes, background color	Nil	Manage and change settings as wished
4.	Create_3D	3D-view	Object highlight, Object mark	Generate the 3D-view with additional information
5.	Add _additional _information	4, 6,7: Text, property values, dimensions	Nil	Add additional information to views
6.	Create _section	Section-view	Bottom height, Top height, object mark, Depth, frame dimensions	Generate the section-view with additional information
7.	Create_plan	Plan-view	Horizontal dimensions, object highlight, frame dimensions	Generate the plan-view with additional information
8.	Create _image	Viewpoint-image	Nil	Dump image of viewpoint
9.	Merge _images	8: Viewpoint-images + template image	3D-view, section-view, plan-view, template	Merge template with viewpoint images
10.	Export _TOWI	9: Merged images	Nil	Save TOWI
11.	Print_TOWI	10: Exported TOWI	Nil	Print TOWI

6.2 CONFIGURE COMPONANTS

This paragraph discusses the software components which make the automation possible. The designed process model and task overview will be used as input for the creation of the components.

Since the TOWI have to be created out of the open IFC file format, the software components are created in programming language Python. More specifically, Python version 2.7.13 (32-bit) was used. Python was used since it is one of the best programming languages for beginners. Secondly, since there was a Python package called IfcOpenShell where python can be used to develop, manipulate IFC-files and is able to navigate through the data model and extract information. Since the IFC file format is the input for the TOWI, this package was a perfect fit.

IfcOpenShell is an open source software library based on Open CASCADE (OCC), which is a public geometrical package to create, manipulate and read geometrical models (e.g. three-dimensional models). The IfcOpenShell package makes use of PyQt4 Graphical User Interface (GUI) and viewer. The full Python script which is created for this research project is added in Appendix 4. (IfcOpenShell, 2017)

AUTOMATION COMPONENTS

With Python programming background and some additional training, it is possible to create the automation components. But before the components will be created, additional decisions have to be made about its functioning and the approach.

A. Decide information approach

Besides the procedure, the information approach has to be decided in an early phase. For example, an object (window) can be selected manually (by mouse or numerical selection), and information can be generated based on the selected object. This method is called object-oriented. This method suits well with IFC, since IFC is an object-oriented data model, and therefore applied in this research project. Manual selection has both advantages and disadvantages over automated selection. With manual selection, an object can be manually selected, which has the advantage of choice of which information has to be generated, especially when specific information is required. A disadvantage of this method is that this method is labor-intensive when many TOWI of objects have to be selected. For this reason, an automated selection approach is applied, which loops through the objects of the specific class (IfcWindow). Other objects might require a different approach.

B. Components

Next, the 11 components of the process model will be discussed piece by piece. To execute the components, Python packages OCC (OpenCascade, 2017), IfcOpenShell (IfcOpenShell, 2017), pythonOCC (PythonOCC, 2017), PyQt4 (Riverbank computing, 2017), and PIL (Pythonware, 2017) are required.

1. Select_file

The user, which for this case is either the site-supervisor or construction planner, opens the TOWI-generator application by executing the ‘start_generator’ file, and gets an empty viewer as shown in figure 33. Then he selects the most recent integrated IFC-file, by using a file-browser (figure 34). The IFC-file can either be a local file, or a cloud based IFC-file. The file-browser is a standard PyQt4-plugin. PyQt4 is a package for creating GUI.

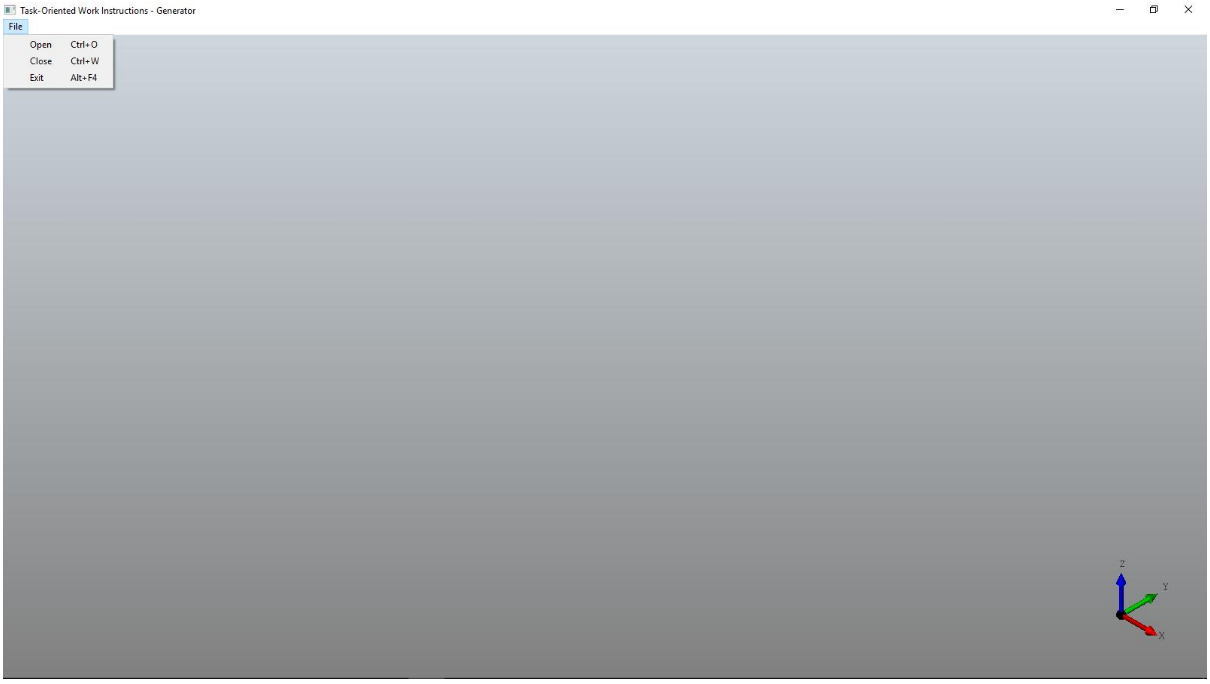


Figure 33: Start-screen TOWI-generator

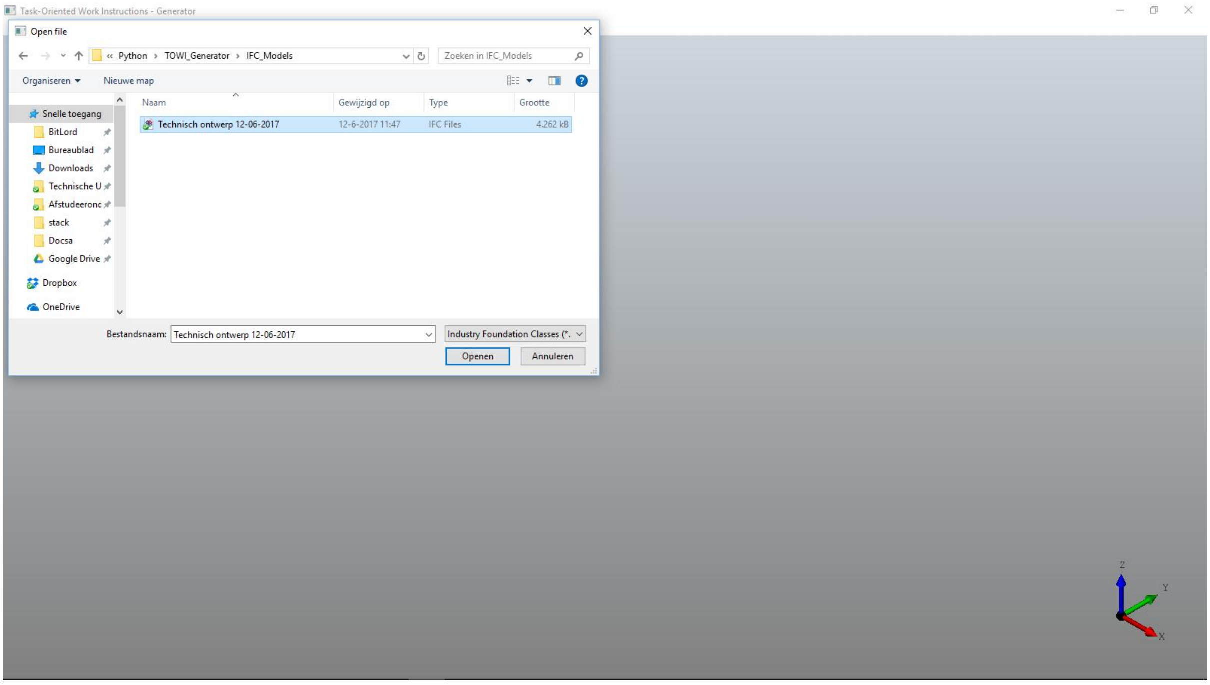


Figure 34: Windows file-browser

2. Open_file

By running the script or process model using a workflow engine, the variable to which the IFC-file was assigned will be opened. The IFC-file will be loaded in the script by the IfcOpenShell package, which is used to read and write IFC-files. If desirable, the IFC-model could be filtered to load less objects. For this project, the IfcOpeningElement, IfcSpace and IfcSite are not loaded because they hinder the results. Also IfcFurnishingElement is not loaded, because these objects are not relevant for this case and make the model unnecessary large. Another method is to not export these classes by setting the MVD to except these classes. All other IFC-classes are loaded in the model. Both actions are done by executing the following script:

```
def product_shapes_list(ifc_file): #Lines 40-52
    products = ifc_file.by_type("IfcProduct")
    product_shapes = []
    for product in products:
        if product.is_a("IfcOpeningElement") or product.is_a("IfcSite") or
product.is_a("IfcSpace") or product.is_a("IfcFurnishingElement"):
            continue
        else:
            try:
                shape = ifcopenshell.geom.create_shape(settings,
product).geometry
                product_shapes.append((product, shape))
            except:
                continue
    return product_shapes
```

3. Manage_settings

Managing settings involves telling the machine how to present information, such as text sizes, colors, unit handling, line aspects, etc. These settings have a direct effect on the presentation of the output, so these should be adapted to the wishes of the client (construction worker). Settings can both be set globally, as locally. For example, the color for all objects (global) can be set, but they can also be set for specific objects (local). Components are created for changing standard settings and applying new settings.

Standard line settings for drawing geometry are yellow. Since yellow lines are little visible on a white background, the line color is changed to black by using the following code:

```
# Line settings ( make the lines black) #Lines 151-154
def set_line_settings(display):
    black = Quantity_Color(Quantity_NOC_BLACK)

display.Context.DefaultDrawer().GetObject().LineAspect().GetObject().SetCol
or(black)
```

View settings:

The geometry can be presented in different modes, such as wireframe, filled objects or transparent objects. Standard filled objects are presented, but the mode can be changed by the following code.

```
def set_wireframe(canvas): # Set the view to Wireframe  #Lines 161-163
    canvas.SetModeWireFrame()
```

Figure 35 shows the modes transparent (left) and wireframe (right), where the geometry lines are presented in grey. As you can see, both 3D views are very crowded/cluttered, and not very clear. This is the result of complex integral 3D models, which consist of a lot of (geometrical) information. These modes have the advantage that you can see through them, and gives insight in the objects and make it possible to mark an object (black objects).

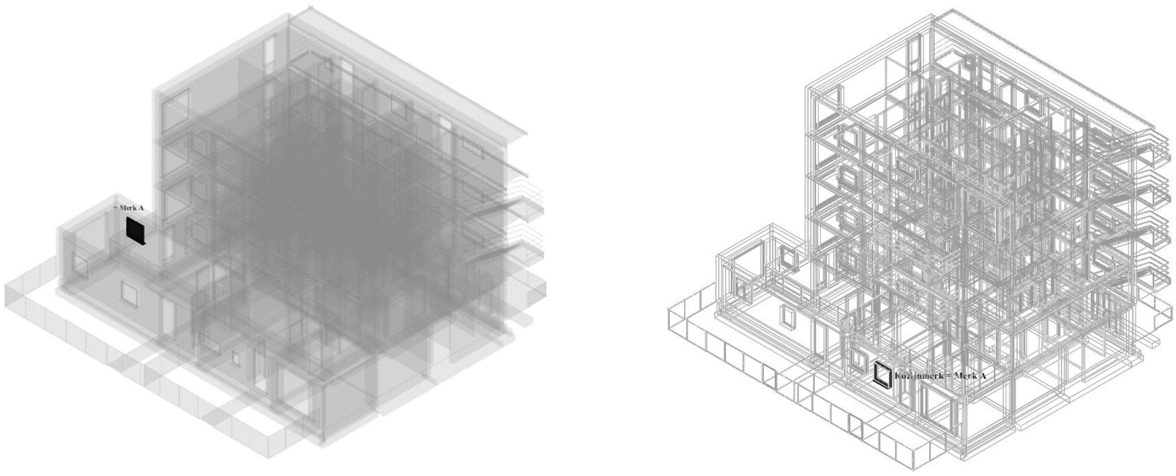


Figure 35: Grey shapes

Figure 36 shows the filled object mode and wireframe mode in colors. Again are both the 3D views very crowded. However the colored 3D model gives a clear view of the building, the marked object is not clear. The colored wireframe model has the advantage to see through the building, but is too cluttered to navigate.

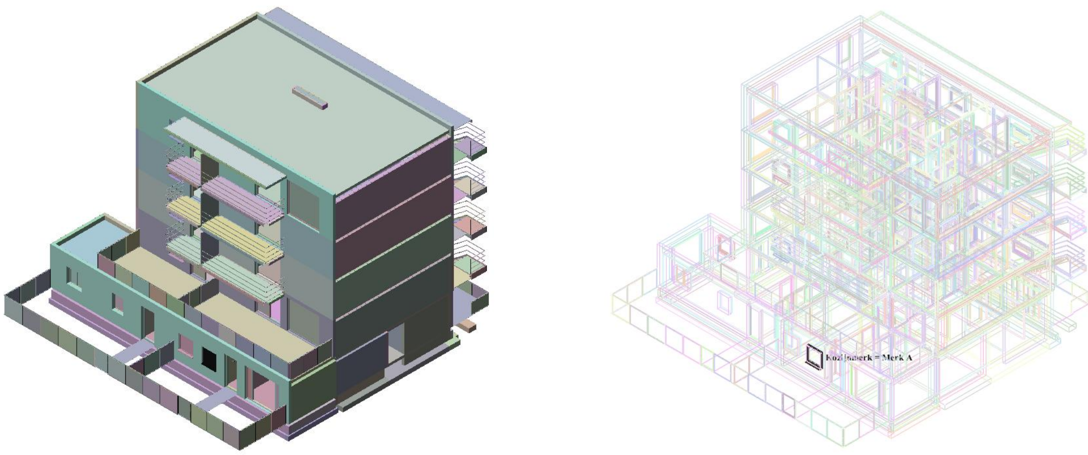


Figure 36: Colored solid model and colored wireframe model

All four 3D-models are parts of a much larger complex. Showing the total complex (8 times larger) would make the 3D model and marked object even less clear. Showing a smaller part of the model would lead to better results, but would make the user lose sight of the complete picture and would make orientation more complex. As a result, the 3D-model (grey wireframe) is still used on the TOWI for orientation purposes.

Figures 35 and 36 show 3D-models in isometric view. When changing views from 3D-models to 2D sections, the display has to change. The standard setting is isometric view, but for plan views a top view is required, for sections a section view is required, and for elevation views a side view is required. These settings are set by the following code:

Both isometric view and top view are standardized options in the OCC library.

Display settings Isometric, top-view, side-view, and fit-all:

```
def set_iso_view(display): # Set the view to Isometric #Lines 165-166
    display.View_Iso()

def set_top_view(display): # Set the view to Top #Lines 168-169
    display.View_Top()
```

The section view makes a perpendicular view connected to the selected object . The script (below) finds the original direction of the object and sets the view to the perpendicular direction.

```
def set_section_view(display,selection): # Set the view perpendicular to
the selected object #Lines 171-183
    if get_orig_x_direction_selection(selection) >0:
        xdir=get_orig_x_direction_selection(selection)*(-1)
    else:
        xdir=get_orig_x_direction_selection(selection)
    if get_orig_y_direction_selection(selection) >0:
        ydir=get_orig_y_direction_selection(selection)
    else:
        ydir=get_orig_y_direction_selection(selection)*(-1)
    display.View.SetUp(0,0,1)
    display.View.SetAt(xdir,ydir,get_orig_z_direction_selection(selection))
    display.View.SetEye(0,0,0)
```

The elevation view will be discussed in 8 B.

The last view setting is to erase the view to make space for the next to be generated view. If the object would not be erased, the view would be overwritten.

```
def erase_view(display): # Erase all objects on the view #Lines 202-203
    display.EraseAll()
```

4. Create_3D

Creating the 3D-view is done by the pythonOCC package, which is used by the IfcOpenShell package. This package is used to translate the IFC-object into a geometrical representation of the object. Executing the following code, will create and present the 3D-model with grey shapes. This function is built in the IfcOpenShell package. The following code will create and display a shape of every IFC-object within the model, except the unloaded objects. The result can be seen on figure 35 (left).

```
def show_grey_3D_shapes(display, product,product_shapes): #Lines 207-210
    for shape in product_shapes:
        grey=Quantity_Color(Quantity_NOC_MATRAGRAY)
        ifcopenshell.geom.utils.display_shape(shape[1],grey,
viewer_handle=display)
```

Additional settings can be set to manipulate the results, such as displaying in wireframe mode, transparent mode, and adding addition information (see below).

5. Add_additional_information

Besides managing the settings, additional, non-geometrical information has to be added to geometrical representations. Additional information can be texts, representation of properties, dimensions, highlighting, names, additional lines, etc.

Finding specific information in IFC can be a complex process because of its many hierarchal relationships and its object-oriented structure. By creating a script to navigate through the model, 'deep' information can be reached. By reusing and adapting the script, other information can be found. For example, a property of a window object is a NominalValue of a PropertySingleValue of a PropertySet of a RelatingType of a PropertyDefinition of a IfcWindow of a IFC-file. The script to reach this (and other) information can be found in **Lines 118-139** of the full source code.

Adding messages/ textual data:

All kinds of information can be added to the view as textual information (numbers, sentences, codes, etc.). This textual information is added by creating messages. This is a standard function of OCC. By connecting information from properties, as described above, to messages, additional information can be depicted on views. Figure 37 shows a created view, including (right) and without additional information (left). Both hand written text, as model data can be depicted. As a result, informative information can be added manual, but also information can be added fully automated by using the 'deep' information navigation script. Within the case, there was no data of the window frame available, so it was added manually. Also general information such as 'check actual measurements before installing' can be added to the views.

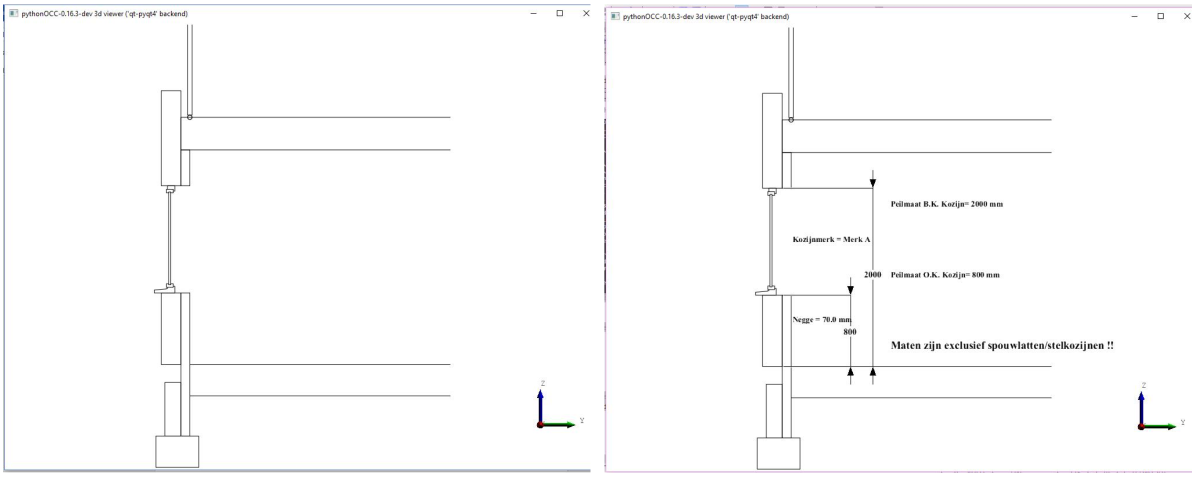


Figure 37: Empty section view (left), and section view with additional information (right)

Object highlighting:

To increase the orientation, the selected object is highlighted, to see of which object the information will be generated (Figure 38). Object highlighting is done by recoloring the selected object to black (for black and white use) by the following code. The code searches the current object by comparing the GUID with the selected shape. If this is correct, the object will be presented black.

```
def mark_selected_object_black(display,product,product_shapes): #Lines 212-216
    for shape in product_shapes:
        if shape[0].GlobalId == product.GlobalId:
            black=Quantity_Color(Quantity_NOC_BLACK)
            ifcopenshell.geom.utils.display_shape(shape[1], black
viewer_handle=display)
```

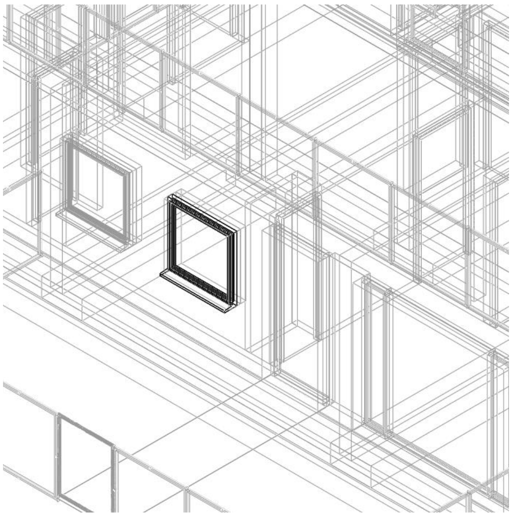


Figure 38: Object highlighting (black)

Create grid:
One of the parts of the information demand was to see the grid, and show the dimensions for horizontal positioning based on the grid. Therefore, the grid has been added to the view. Since the grid exists of two-dimensional lines (curves), the pythonOCC package skips drawing these lines, since they are not part of three-dimensional shapes. As a result, a script for adding the grid was written. As mentioned before, the grid has to be exported to an IfcGrid, otherwise there is no data to draw the grid. If there is no grid available in the model, this function will message that there is no grid available. The grid has the functions of increasing orientation and making horizontal positioning possible.

Because only a small part (plane) around the selected object (window), is shown, only the gridlines which cross that part are relevant. Otherwise all gridlines would be drawn, which would interfere with the viewing settings (Fit all). Therefore, the same plane as used for the plan view is reused. The plane's four coordinates are set by the following code. Therefore the function for finding the current object (window) coordinate is used and the plane range is determined by adding dimensions (VMIN, VMAX, UMIN, and UMAX). This concept can be seen on figure 40.

```
planey1=(get_global_y_coordinate_selection(selection)+VMIN)
planey2=(get_global_y_coordinate_selection(selection)+VMAX)
planex1=(get_global_x_coordinate_selection(selection)+UMIN)
planex2=(get_global_x_coordinate_selection(selection)+UMAX)
#Lines 261-264
```

As a result, gridlines which do not cross the plane are not drawn and gridlines which do cross are changed to the size of the plane. Therefore, the grid labels (e.g. A, B, C, 1, 2, 3) are always visible for the user. The grid labels are imported by finding the according AxisTag in the IFC-file. The grid labels are shown as messages and placed at the endlines of the grid in the plane.

The gridlines are drawn by comparing its own coordinates with the coordinate system of the plane of the current object. When the gridlines cross the plane, they are drawn. On figure 39, you can see a part of the building with the gridlines cut to the plane size and including labels.

When the gridlines are drawn, the dimension lines connecting to the grid and the current object has to be drawn. First, the gridline which is the nearest to the current object is calculated. This is done by the following code. Not calculating the nearest grid would result in many dimension lines, which makes the overview unclear and shows irrelevant information.

```
            #Calculate nearest vertical grid
            nearest_vertical=min(vertical_axis_list, key=lambda x:abs(x-
get_global_x_coordinate_selection(selection)))
```

The nearest gridline is also connected to the direction of the gridline, so the correct dimension can be calculated. The dimension is calculated by subtracting or adding the current object coordinates with the gridline coordinates. By using the following code, the dimensions lines are calculated and drawn:

```
                p3 =
OCC.gp.gp_Pnt(nearest_vertical,get_global_y_coordinate_selection(selection)
,set_section_height(selection))
                p4 =
OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection),get_global_y_coo
rdinate_selection(selection),set_section_height(selection))
                ld2=OCC.AIS.AIS_LengthDimension(p3,p4,section_plane)
canvas.Context.Display(ld2.GetHandle(),True)
#Lines 304-306
```

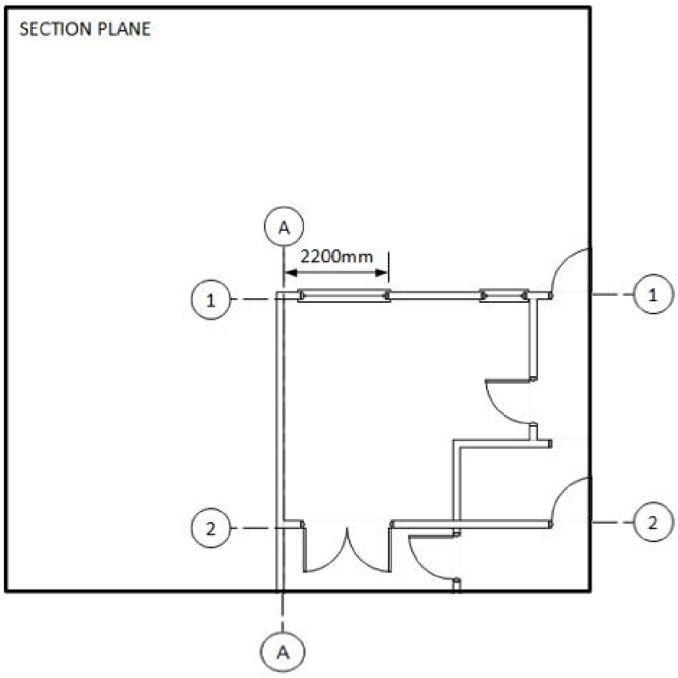


Figure 39: Drawing the IfcGrid, grid labels, section plane and object dimensions connected to grid

6. Create_section

Automatically generating vertical sections are done by making a section plane and drawing the shapes that intersect with the section plane on a selected section face. This method is based on the method of Van Strien (2017). He created a script which makes a section face by making wires of the objects which cross a section plane. By reconnecting the wires, the objects are recreated and the section view is complete. Within this project, his code is reused to create sections.

By connecting the section plane to the selected object (window) coordinates, and taking its perpendicular direction, the section plane will be automatically selected to the current object and make a section of it. The code of creating the sections and connecting them to the object can be found in **#Lines 231-258**

7. Create_plan

The floor plan will be generated with a similar method to create the (vertical) section. The plan, is a horizontal section of the 3D-model, and can be generated by the same method. Creating the section plan is also connected to the selected object (window). Because only the relevant part has to be shown, a section plane is set. Figure 40 shows the concept of a section plane, where the selected object is the center of the section plane. UMIN, UMAX, VMIN, and VMAX define the size of the section plane (12 x 12 meter in the example). The dimension of the plane can be changed by changing the UMIN, UMAX, VMIN, and VMAX values.

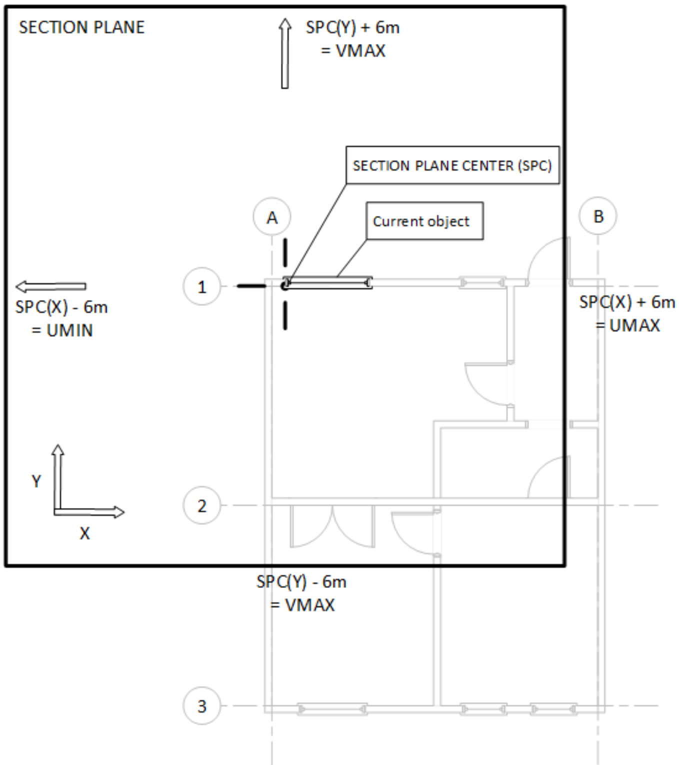


Figure 40: Section plane

Because the code is designed with small components (functions), they can be reused for many operations. For example, the function for creating the section can be used to create the (vertical) section, but also to create the horizontal section. Only the input values have to be changed (automatically). As a result, functions can be reused efficiently and the application is less 'hard-coded' and more flexible for reuse and expansion. The code to create the horizontal section can be found here:

```
    #First make a section plane to assign where the intersection should be
made #Lines 505-517
    section_plane = OCC.gp.gp_Pln(
        OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection),
get_global_y_coordinate_selection(selection),
set_section_height(selection)),
        OCC.gp.gp_Dir(get_x_direction_selection(selection),
get_y_direction_selection(selection), get_z_direction_selection(selection))
    ) #SEE NEXT PAGE
```

```
    # Make the section face and assign its dimensions
    UMIN = -6
    UMAX = 6
    VMIN = -6
    VMAX = 6
    section_face =
OCC.BRepBuilderAPI.BRepBuilderAPI_MakeFace(section_plane, UMIN, UMAX, VMIN,
VMAX).Face()
    # Make the section
make_section(canvas,selection,section_face,product_shapes_list(ifc_file))
```

By using the add additional information functions, additional information such as texts and parameters can be added to the plan view.

7B. Create_elevation view (added)

After feedback from construction workers about the first versions of the TOWI, an elevation view function was added to the application. The 3D-views did not give enough insight in the orientation and location of the selected object, due to its crowded and cluttered overview. Besides this, when the selected element was at the back of the 3D-view, it was not visible. Therefore the current method of orientation (elevation views) is reused and added to the application. Because many functions were created in the script, creating the elevation view was a relatively simple process. Only one new function had to be created, other functions could be reused.

The following function was created to set the elevation view, which is a side view, connected to the selected object (window). The function reuses the functions to find the direction of the object. Next, the original x, and y axis are analyzed to determine if the elevation is made on the correct side of the building. The script inverts the direction when the elevation is made on the opposite side. The result can be seen on figure 41.

```
def set_elevation_view(canvas,selection): #Lines 185-197
    if get_orig_x_direction_selection(selection) >0:
        coordx=get_orig_x_direction_selection(selection)*(-1)
    else:
        coordx=get_orig_x_direction_selection(selection)*(-1)
    if get_orig_y_direction_selection(selection) >0:
        coordy=get_orig_y_direction_selection(selection)
    else:
        coordy=get_orig_y_direction_selection(selection)

    canvas.View.SetUp(0,0,1)

canvas.View.SetAt(coordy,coordx,get_orig_z_direction_selection(selection))
    canvas.View.SetEye(0,0,0)
```

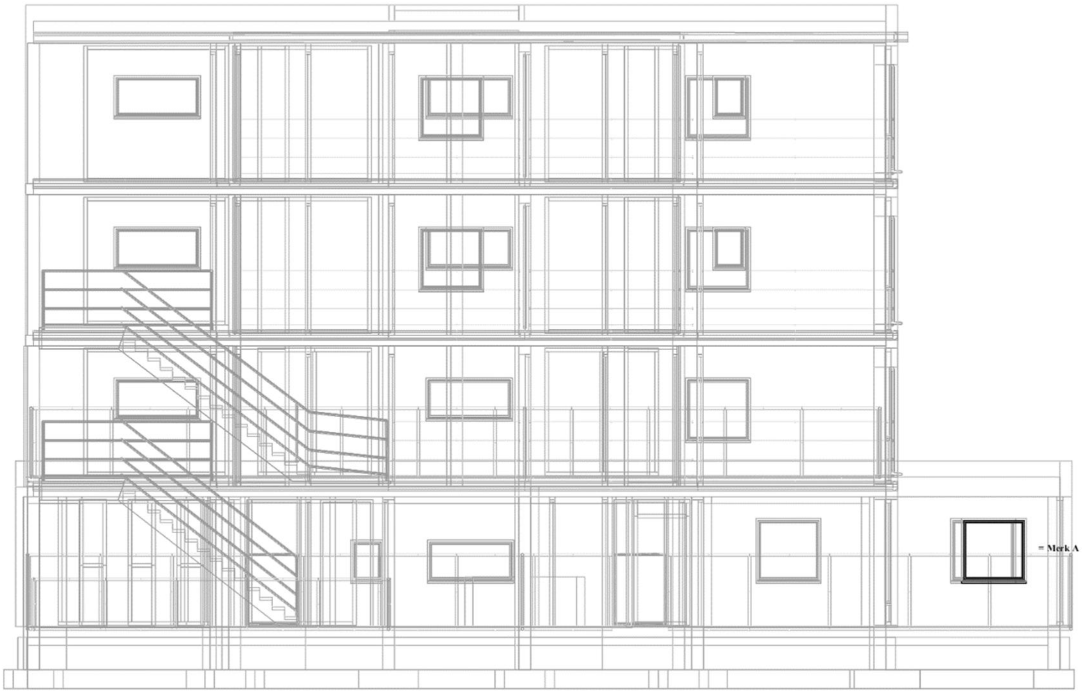


Figure 41: Automated Elevation view of selected object

8. Create_image

When the views consist of the required information, they can be exported. For the case study, the output should be paper-based, since the construction workers requested this. Therefore, the views can be exported to several formats, which can be plotted. Some examples are Scalable Vector Graphics (SVG), Portable Document Format (PDF), or image formats such as PNG or JPEG. PDF is a well-known format in the AEC-industry. The AEC-industry is less acquainted with the SVG-format, but this format has the advantage of keeping high resolution when zooming. Images are the third alternative, but have restricted resolution.

All three export format options were built in the package OCC. The goal was to use SVG because its high resolution when zooming. Another option was exporting to PDF. Unfortunately, both exports led to errors while exporting. The geometry was not exported as depicted in the viewer, due to errors in the export function. The image export function, however, led to the correct output. Since, images could also be used for printing, and the output was requested on paper, this solution was still usable. As a result the views are 'dumped' to Portable Network Graphics (PNG) files, because they have a lossless compression, compared to for example (JPEG). The dumps are created by the standard 'Dump' function, which makes an image of the viewer:

```
canvas.View.Dump(sectionview)
```

9. Merge_images

The next step is to merge the created images on one sheet. Python package Python Image Library (PIL) is used to merge and manipulate the images. First, two templates are designed, a A3 landscape template, and a A3 portrait template. This format is chosen because this was the favorite format as researched by Span (2015). A standard A3-image has 4963 x 3509 pixels. The template can be split in smaller boxes to paste the images (exported views). Also additional information, such as company logo's, dates, and stamps can be added to the template. For the tests, the portrait format was split in 3 'boxes', the landscape format was split in 4 'boxes'. Figure 42 shows the 4-box landscape format. Adding boxes is unlimited, as is the size of these boxes (with exceptions to the template size). The boxes are based on coordinates, which are the pixels of the image. The images have a four-coordinate system (left, top, right, bottom). The templates and boxes are selected by the following code:

```
def create_image(selection): #Lines 548-562
    template_landscape=os.getcwd()+'\\TOWI\Template-A3-Liggend.png'
    template_portrait=os.getcwd()+'\\TOWI\Template-A3-Staand.png'

    # Een A3= 2970mm x 4200mm en 3508 pixels bij 4961 pixels
    sheet_landscape = (4963, 3509)
    sheet_portrait = (3509, 4963)

    # Coordinates (left,top,right,bottom)
    # 4-boxes landscape format:
    full_box= (0, 0, 4963, 3509)
    top_left_box = (163, 150, 2488, 1750)
    top_right_box = (2488, 150, 4813, 1750)
    bottom_left_box = (163, 1750, 2488, 3350)
    bottom_right_box = (2488, 1750, 4813, 3350)
```

Figure 42: A3-landscape template including 4 boxes

To the boxes, images can be set, which are created variables in the script. The following code sets the images to the boxes, as shown in figure 42:

```
boxes = [
    [full_box,template_landscape],
    [top_left_box, threeDview],
    [top_right_box, planview],
    [bottom_left_box, sectionview],
    [bottom_right_box,elevationview]
] #Lines 564-570
```

The images have to fit to the boxes. Therefore a small script is written that resizes the images, which calculates the ratio of the image and compares it with the ratio of the box. Then it resized the image, so the image stays in the same ratio, but maximizes itself within the box. This is done by the following code:

```
# Resize images and paste them on an empty sheet or template
#Lines 599-619
sheet = PIL.Image.new('RGB', sheet_landscape)
for box, img in boxes:
    img = PIL.Image.open(img)
    box_width = float(box[2] - box[0])
    box_height = float(box[3] - box[1])
    img_width = float(img.size[0])
    img_height = float(img.size[1])

    box_ratio = box_width / box_height
    img_ratio = img_width / img_height

    if box_ratio > img_ratio:
        img_width = box_height / img_height * img_width
        img_height = box_height
    else:
        img_height = box_width / img_width * img_height
        img_width = box_width

    img = img.resize((int(img_width), int(img_height)))
    sheet.paste(img, (int(box[0]), int(box[1])))
```

10. Export_TOWI

Finally, the merged image has to be saved. Since every TOWI, has to be a unique image, so it will not be overwritten, a number will be added. As a result, the images will have a unique identifier and will be saved in a predefined folder, which can be changed. When the image is saved, it will be shown directly. An example can be seen on figure 43, the saved documents can be seen on figure 44. This is done by the following code:

```
# save and show sheet #Lines 621-627
# For every TOWI, make a new image
if os.path.exists('TOWI\werkinstructie.png'):
    sheet.save('TOWI\werkinstructie_{}.png'.format(int(time.time())))
else:
    sheet.save('TOWI\werkinstructie.png')
sheet.show()
```

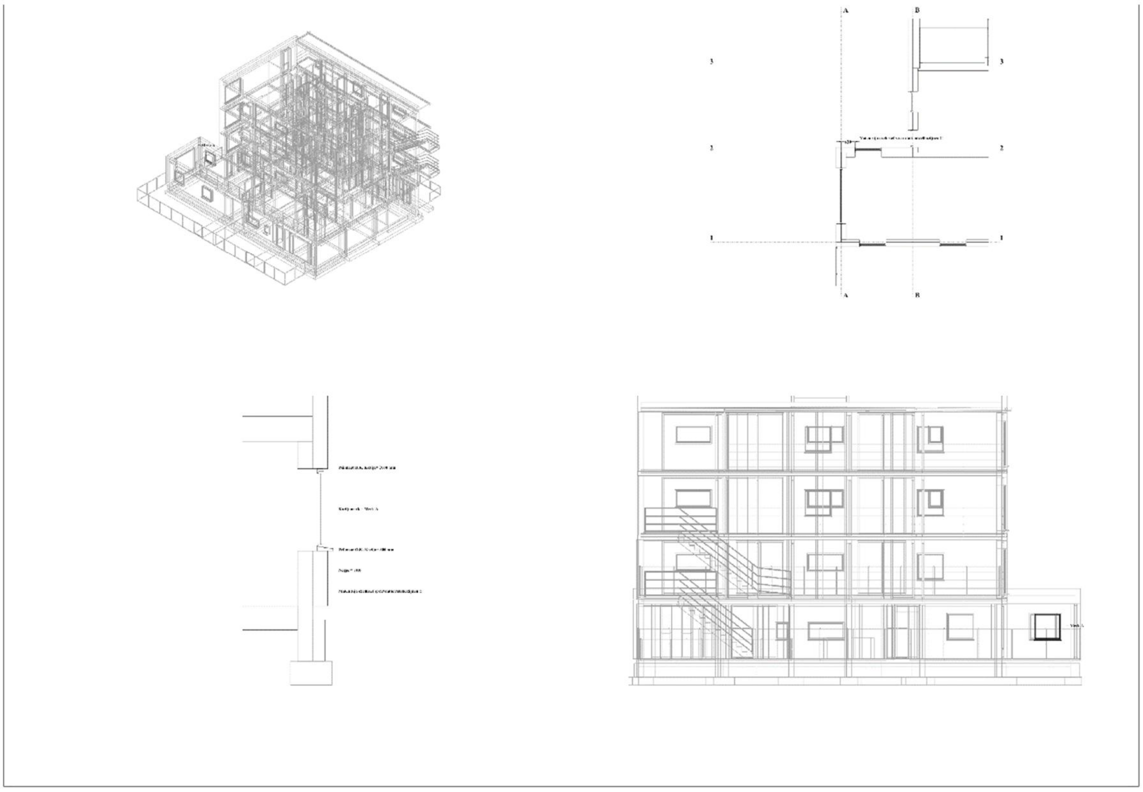


Figure 43: Exported TOWI

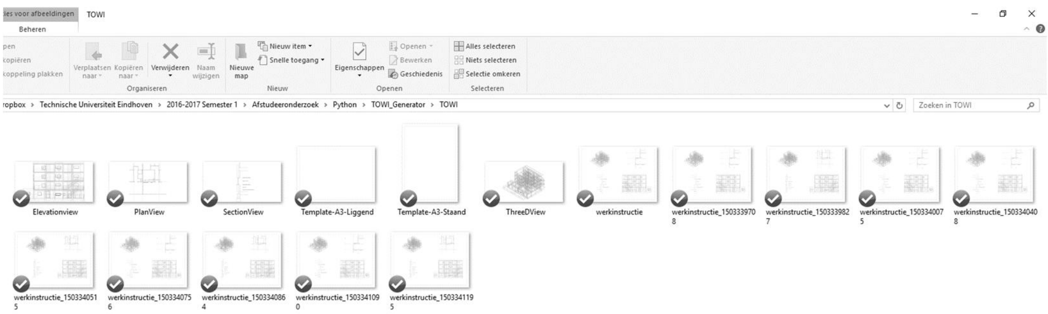


Figure 44: Windows folder with exported TOWI

11. Print_TOWI

Finally, the TOWI can be printed on either the construction office or construction site.

6.3 EXECUTE PROCESS

The BPA-approach discussed three steps, namely raise request, instantiate process model, and execute process. Because the BPA-approach does not have a sequential process, there is not an exact match with the applied process. For example, the process starts with an information demand (raise request). Also, the process model is created before creating the software components. However, the process still has to be executed.

The components are designed to read and use IFC-files by using the IfcOpenShell package in the programming language Python. Eindhoven University of Technology has developed an application which uses this package, including the OpenCasCade package and the PythonOCC package. The application is called 'tue_viewer' created by Thomas Krijnen and Jakob Beetz. The application uses PyQt4, which is a package for creating a Global User Interface (GUI) for among others Python (Riverbank Computing, 2017). This application was used to implement and test the components. The application was altered on some small aspects to fit to the TOWI concept. This resulted in the application 'TOWI-Generator'. The altered code of the 'tue_viewer' can be found in appendix 4.

The used method has problems with data management. A 32-bit version of Python was used, which has a limit of about 2GB of RAM. When loading large models (50MB+), the generation of TOWI, will lead to a huge increase of RAM data, and will run out of data after a few generated TOWI. Therefore, a 64-bit version of Python should have been used, and the data model should be optimized, so the increase in RAM would be lower.

Another disadvantage of the used method is that the application does not give insight in the process (what is happening behind the scenes?). Only users, who know how to read code (python) are able to understand the underlying process. A standard employee of an construction company does not contain such skill.

Currently, there is not library connected to the application, so it is not possible to save process models and components (except for copy and pasting code). Also, there is no feedback process designed to evaluate comments and change these in the designed components.

For this project, the use of the 'tue_viewer' application did fit the scope, but would not be recommended for future continuation of this research project. Therefore, either the application should be rebuild to fit goals for flexible building and execution of process models. The author advises to use an workflow engine as discussed in paragraph 5.5. Then the process models could be created as blocks of code (components), and could be dragged and dropped in a user friendly interface. Within the AEC-industry, AutoDesk Dynamo is such an tool, which has capability to read, edit and write 3D-models including metadata.

Such a tool would be a perfect match with the component based BPA-method. Within Dynamo, process models can be saved and reused. Also components can be designed in a user-friendly interface, and is able to apply the Python programming language. After several projects, the library would be filled with reusable components which can be dragged into a process model to set up automation processes. Besides this, the components can be adapted easily to change its in- or output.

6.4 MEASURE OUTCOME

The final part was to measure the outcome. Therefore, some of the interviewed construction workers were asked for feedback. The goals of the feedback were:

1. To verify the information demand as was captured in the interviews;
2. To get their opinion on their thoughts about TOWI;
3. To identify missing information/ improvements;
4. To get their opinion on how to expand this concept.

Their feedback was asked by discussing the exported and printed versions of the TOWI, in A3-landscape (both 3 + 4 boxes) and A3-portrait modes. Of the 12 previously interviewed construction workers, 5 were asked for feedback (3 different projects).

1. Verify information

The first goal was to verify the information demand to make sure it was captured correctly, and the TOWI should still fulfil their needs. All interviewees verified the information demand, and no additional information was requested.

2. Additional feedback

After showing the results, the second goal was to get additional feedback about the concept of TOWI and their thoughts about the presented results. The interviewees like the concept of TOWI and getting clear overviews of information. Important is that the information should be trustworthy and complete. Otherwise, they will have to fall back on other documents (plan views, details ,etc.) . However, they are sceptic about losing total project overview.

3. Missing information and improvements

The third goal was to identify missing information and improvements on the presented results. In the first interview, the 3-boxes TOWI with a 3D-view, section view, and plan view was presented. The construction workers commented on the use of the 3D-view for orientation purposes. Since, the image was not clear, the orientation/ location of the selected object was not clear. As a result, an extra component for creation elevation views (8B in paragraph 6.2), was created. When discussing the (new) results with the other construction workers, the orientation and location of the object was clear. Other comments were that text sizes were too small, the images were missing names (e.g. plan view, section view, elevation view), the sheets were missing a clear title, the 3D-view is not clear, other texts are used on the construction site, gridlines are too light (low contrast on white paper), and that the images were too small and should be fitted better on the A3 paper, since it contains a lot of whitespace. The TOWI, however, contained all the information that was required, so they would be able to use this document to actually perform their task. Besides this, this would lead to more clarity, especially by using unique identifiers for each object. On the other hand, this would make TOWI for every window in the building, when many similar windows have to be installed on a similar way, this process is not very efficient. Besides this, it is a more complex process to find the allocated window on the construction site, because of all the unique identifiers, while other windows are exact the same. They like the idea of TOWI on paper over TOWI on digital applications, such as tablets or smartphones.

4. Concept expansion

The last goal was to get their opinion on how to expand this concept to other tasks and functionalities. The interviewees think this concept can be reuses for many tasks, and is especially interesting for sub-contractors who are not (and do not have to be) familiar with the project, and are assigned to complete a certain task. Also on large-sized projects where construction workers have more specific roles (less responsibility), this concept will be interesting. The interviewees like the idea of having task specific information. This concept can be implemented directly for the tasks: making walls (every material), wall and floor openings, dormers (dakkapellen), columns and doors.

CONCLUSION TOWI

To conclude, TOWI are most effective in situations where information from many different sources have to be combined, such as placing windows. This increases the overview of information. However, by not showing the complete picture, the construction workers lose total project overview. Still, they think TOWI can give clear information for performing their tasks, but the information should be complete, trustworthy, readable, and clear. The 3D-view does give extra insight for the orientation, but should be clear to be valuable. The construction workers had comments about using a TOWI for every object, even when similar objects have to be installed. The author understands this issue, but thinks this will lead to a higher quality, since mistakes will not be made, when slight changes are not recognized in the original method. To solve the logistic issue to find the allocated object 1) an intelligent systems which recognizes the object similarities can be used, or 2) QR-codes can be added to the objects, which show the according TOWI (for which digital applications are required).

The results are not quantified and are based on the opinions of the construction workers and could be classified as qualitive results.

PART E: EVALUATION, IMPLICATIONS AND CONCLUSIONS

This section, discusses the impacts of the aforementioned approach on processes in the AEC-industry (Chapter 7), and if the approach and result was successful. Besides this, it discusses the results of the research by answering the main research question and the sub-questions (Chapter 8). Furthermore, recommendations for future research are provided and explained (Chapter 9).

7. EVALUATION

7.1 INTRODUCTION

This chapter reflects and discusses on the previously described theory, used method, implementation, results and its impact on the AEC-industry.

Within the AEC-industry, information is becoming more intelligent in a fast pace. Where a few decades ago, paper drawings and documents were the main information carriers; nowadays, a data-oriented approach (intelligent objects including metadata) is possible. This data-oriented approach leads to new possibilities, such as data analysis, information filtering (searches and query's), and calculations. This transition from a document-oriented industry to a data-oriented industry requires a culture change and new insights on information/data management.

This research project has successfully researched a process to change from an original document-oriented approach to a data-oriented approach to make automation possible. Therefore, the information demand is analyzed in-depth, and was changed to data. As a result, automation could be applied to automatically generate documents based on the data. This process leads to more efficiency and quality. Human interaction, where labor-intensive tasks have to be performed, can be automated. Data can be reused for many automation purposes, where duplications of information will decrease. Also, information can be generated on the moment that the information is needed. Important is that the data-model which is used for automation, is managed and up-to-date.

The added value of this research project is not just automating the process of the creation of TOWI, but the concept of automation for construction in whole. The step to a data-oriented industry is ongoing, and comes with many new possibilities of automation as can be seen in this research project. Instead of creating documents with large amounts of information, data can be filtered to show only relevant information for its purpose. This allows IT-tools to be configured in new ways to filter information and give the user immediate insight the of the information they need.

Key Performance Indicators (KPI) of this process are:

- (open) BIM must be the leading information source within the project;
- A higher level of information (metadata and geometrical data) within the BIM-product leads to more possibilities for automation. By focusing on the output (what is needed), the requested level of information should be identified;
- Automation falls or stands with identifying the information demand precisely;
- Activities within the industry (or company) have to be analyzed to standardize workflows and create process map, which are input for automation;
- These analyses should be performed by specialists who both understand business and automation. Examples are information managers and business analysts.

7.2 IMPLICATIONS

For this process to work, data-model should be enriched by both geometrical data and metadata. Adding geometrical data can be automated when using intelligent 3D-objects, for example in Autodesk Revit, all kinds of calculations of an object can be added in parameters. This leads to possibilities of parametric designs, but also more intelligent data which can be used within other automation tools and applications, such as the TOWI-Generator, but also structural analysis, and more. Data can be reused. By identifying how the data flows through the processes, the crucial (key) information can be identified. When the information junction points are identified, they can be managed and controlled. Also less important and irrelevant information can be identified, which leads to a more lean data process.

Steps to consider when transitioning to a data-oriented strategy are:

- Think about the strategy. What are the goals of the project or organization, how can information / data management support these and to what scope;
- Identify what information is required to reach these goals;
- Focus on the output, and define what information is required to get this output;
- Save the information demand in IDM's to communicate the information demand with stakeholders;
- Request IDM's of network parties to increase collaboration and identify similar information/data. As a result, information/data can be reused, instead of recreated which leads to duplicates of information;
- Make use of tools (lists, diagrams, libraries, etc.) to get an overview of the information flow and needs;
- Make use of tools to check the quality and availability of the information, such as query's in databases, and 3D-model checkers.

Eventually, by using IDM's through the whole sector will lead to a more clear information / data process. For example, when a window fabricator clearly states its information demand to the contractor, the contractor can deliver the information as wished. By adding its own information demand (IDM), the window fabricator knows what information to add. This process increases communication and efficiency. When all project stakeholders, state their information demand and this is delivered, the BIM-product should match perfectly with the goals of the project, and each organization.

Besides this, a data-oriented approach also makes machinery automation possible, because data can be understood by computers, which gives new possibilities of robotics in construction.

8. CONCLUSIONS

This chapter aims to answer the main research question and the sub-questions.

MAIN RESEARCH QUESTION:

What is needed to make a (semi) automatic filter of task-oriented work instructions from open Building Information Models for the construction worker, and how can this be achieved?

Important is to focus on the output. What information does the user require, is the most important question. By using IDM's the information demand of each user/stakeholder can be identified. By designing activity diagrams, the workflow of the users can be identified and its in-, and output can be defined. When the activities, in- and output are known, the process can be automated. The Business Process Automation method encapsulates this process, and was successfully used within this research project to automate the creation of TOWI.

The added value of this research project is not just automating the process of the creation of TOWI, but the concept of automation for construction in whole. The step to a data-oriented industry is ongoing, and comes with many new possibilities of automation as can be seen in this research project. Instead of creating documents with large amounts of information, data can be filtered to show only relevant information for its purpose. This allows IT-tools to be configured in new ways to filter information and give the user immediate insight the of the information they need.

Key Performance Indicators (KPI) of this process are:

- (open) BIM must be the leading information source within the project;
- A higher level of information (metadata and geometrical data) within the BIM-product leads to more possibilities for automation. By focusing on the output (what is needed), the requested level of information should be identified;
- Automation falls or stands with identifying the information demand precisely;
- Activities within the industry (or company) have to be analyzed to standardize workflows and create process map, which are input for automation;
- These analyses should be performed by specialists who both understand business and automation. Examples are information managers and business analysts.

For this process to work, data-model should be enriched by both geometrical data and metadata. Adding geometrical data can be automated when using intelligent 3D-objects. Quality data can be reused and by identifying how the data flows through the processes, the crucial (key) information can be identified. When the information junction points are identified, they can be managed and controlled. Also less important and irrelevant information can be identified, which leads to a more lean data process.

Steps to consider when transitioning to a data-oriented strategy are:

- Think about the strategy. What are the goals of the project or organization, how can information / data management support these and to what scope;
- Identify what information is required to reach these goals;
- Focus on the output, and define what information is required to get this output;

- Save the information demand in IDM's to communicate the information demand with stakeholders;
- Request IDM's of network parties to increase collaboration and identify similar information/data. As a result, information/data can be reused, instead of recreated which leads to duplicates of information;
- Make use of tools (lists, diagrams, libraries, etc.) to get an overview of the information flow and needs;
- Make use of tools to check the quality and availability of the information, such as query's in databases, and 3D-model checkers.

SUB-QUESTIONS:

- *How do task-oriented work instructions differ from the traditional approach?*
    - *How are construction workers informed on their tasks in the traditional approach?*
    - *What are task-oriented work instructions, and what are the expectations and requirements of task-oriented work instructions?*

Task-oriented work instructions or TOWI differ from traditional construction document by its content. Where traditional construction documents are created to inform many stakeholders with many different tasks, such as planning, quantity take-off or mounting windows, are TOWI created for only 1 task/stakeholder. TOWI have many advantages over traditional construction documents, such as a decrease of information overload, less distraction, they can be generated on demand and include actual (live) information, there are possibilities to include additional information (e.g. weather forecasts, product information, timelines), the construction workers need only 1 information overview, the information is more useful and construction workers have less searching time to find the information.

On the other hand, it is a labor-intensive task to create TOWI. To make use of TOWI, strict information management and planning are essential, and the BIM-product should be up-to-date and the leading information source of the project. If the information is incorrect, TOWI are ineffective. Besides this, TOWI can only be used for 1 task, where traditional information serves multiple purposes.

Currently, for the task 'placing and assembling window(frames)', the construction workers use only product-related information (Elevation views, plan views, detail books) in analog (paper) format. The elevation and plan views are large A0 sized paper drawings, which cover a large part of the project. The detail books are object or location related information sheets on A3 or A4 paper. Multiple documents are required to inform the construction workers completely. According to the construction workers, these documents often consist of contradicting information, require a lot of searching time, have scattered information and have a lot of irrelevant information.

Furthermore, the construction workers mentioned that information often was hard to find, not always available (at their location) and leads to confusion due to information overload and inconsistencies. Currently, the information flow is organized paper-based to instruct construction workers. The information flow from design to the construction site may take several weeks. Nowadays, digital techniques involve real-time updates. Compared with this, the current information flow process is very slow. This slow process leads to errors and information being late, which may lead to delays in other processes.

Important is that the TOWI should be complete, correct, trustworthy, readable, up-to-date, and the orientation should be clear. In addition, the use of TOWI is due to responsibilities of construction workers more relevant on larger construction projects. The lack of introduction of TOWI in the (Dutch) AEC-industry can be concluded to two main reasons; namely, the labor-intensive process of creating TOWI and the unfamiliarity of this technique in the industry.

- *What information is needed for task-oriented information according to literature and field professionals, and how is/can this information be stored in data?*

Within this research project, one task and its information demand and data is researched. Depending on the task, the information demand will change. For the researched task, the information demand and its data is summarized in table 12.

Table 12: Information overview TOWI

Information demand	Properties	Data type:
Depth	FrameOffset - External	Length = real
Mark	Type Mark	Text = string
Orientation	-	-
Bottom height (base level)	-	Length = real
Mounting frame dimensions	-	Length = real
Horizontal dimensions based on grid	-	Length = real
Top height (base level)	-	Length = real

The construction workers require a relatively small amount of information to perform their task. Where currently multiple documents are required, this information is merged on one TOWI-sheet. Most data types are related to dimensions and are not standardized in the BIM-product. Therefore small components were designed to calculate the ‘new’ properties and add them to the TOWI. A more complex information demand is the concept of orientation, since orientation is a function of the mind and depends on the capabilities of the user. Orientation was not captured with a data type, but with overviews of information, such as 3D-views, plan-views, and elevation views. By applying a realistic scale and showing recognizable object, such as walls, stairs, windows and doors, the concept of orientation was fulfilled.

- *How can the task-oriented work instructions filter be defined, captured in a generic way (that is easy to be reused and adapted in other scenarios) and how will it be processed?*

To create the TOWI, the Business Process Automation approach was used. This method identifies, automates, executes, and measures workflows. This concept can be seen on figure 45. By firstly identifying the task which has to be automated in the format of Verb_object_method[What, WhatElse], the process designer can design a workflow. The next step is to identify the exact information needs (output) for the user. To do so, the method of

Devadason (2008) can be used, which created a method to identify the information demand for different levels within an organization. Then, the automation level of each task has to be determined, to make the process fully, or partially automated, also manual tasks can include the workflow.

Next an process model can be created (workflow-model). The process model encapsulates all possible steps of the to be performed task, and what input and output is required.

When the tasks are identified, the process model is completed, the output is known, and the level of automation is known, software component which perform the task can be created. By developing small components which perform a single task, they can be adapted and reused in other cases. Important is that the components generate the correct output, therefore the input should be correct. As a result the quality and availability of the information should be checked. Checking applications can be used to automate this process.

When the components are complete, they can be connected as stated in the process model. A workflow engine can be used to 'drag and drop' components and connect them with each other. This approach was not used within this research project, but the advice would be to use such tool.

The final step is to check if the results are as requested by validating the results with the user. This is a continuous process. By using a workflow engine, the process can be checked by executing the process modeling, while developing. Feedback should be evaluated before actual use of the information, and after use of the information.

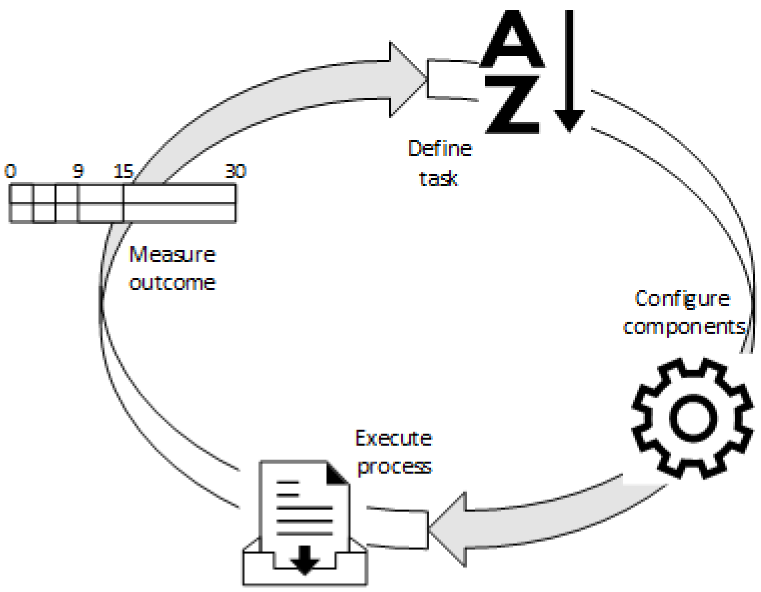


Figure 45: Business Process Automation method

Business Process Automation did fit the goals of this research project and the approach did fit the case study. The BPA-process, however, does not have a sequential approach which leads users to differ from this approach. The BPA-process should therefore be analyzed to create a sequential process for better implementation in the automation design. Also, in the original sources, some aspects were treated (too) slightly. Therefore, additional sources had to be used. The process of identifying information needs did not mention how to apply this. Also the step of checking the information was not treated, and was added to this process.

The created application ‘TOWI-Generator’, has problems with data management, due to low file limit limitations of Python 32-bits. A 64-bits version should have been used for increased possibilities. Also does the current method give not insight in the background process (what is happening behind the scenes?). This method is only useful for people who know how to read code (Python). A workflow engine works with visual process blocks, and is therefore more user-friendly. Also, no library for saving processes and components is implemented. This is an important step to implement this approach in business, but was not in the scope of this research project.

- *How can the task-oriented work instructions filter contribute to the construction workers, and how are the results reviewed?*

Both researchers as construction workers think TOWI will be very useful for their daily information source. TOWI will contribute to a more efficient and clear construction process. Construction workers will be informed better, which will lead to a decrease of errors, less information searching time, less distraction, less interpretation and less information overload. Also additional information can be added to inform them even better and let them make more intelligent choices.

The TOWI should be able to give total project overview. TOWI are most effective in situations where information from many different sources have to be combined, such as placing windows. The TOWI should be complete, trustworthy, readable, and clear. A three dimensional view gives additional insight for orientation, but should be clear. Since this was not the case, an extra elevation view was added, which solved the problem.

9. RECOMMENDATIONS

On many aspects, this research project can use continuation. The concept of TOWI is accepted as very useful for construction workers, but is rarely implemented on the construction site. A standardized application to identify the information needs would give insight in the current information flow process and if the current method matches the information needs of the user. For example knowledge management systems could help save the information and knowledge for continuous improvement.

Another focal point, would be to further investigate the use of digital techniques on the construction site. Within this research project, the construction workers had a preference for paper-based methods, but the author thinks digital applications would be more efficient and accepted on the construction site. Currently the results are save to ‘dumb’ formats, but intelligent formats could be kept by using digital techniques. This could even increase the overview and would change the process to a data-oriented process instead of an document oriented process.

This research project was focused on only one task. The author thinks, this method could be easily adapted to more tasks, but this should be implemented and tested. The author would advise to rebuild the used method to a method which uses the workflow engine. Because of the user-friendliness and similar functioning. Workflow engines often consist of libraries to save process models and components, to increase the speed of automating processes.

10. REFERENCES

Beetz, J. (2015). Lecture Building Information Modeling. Eindhoven: Eindhoven University of Technology.

Beetz, J. (2015, December 12). Lecture Industry Foundation Classes. Eindhoven: TU/e.

Beetz, J. (2016, 12 21). 7m900_week5_IFCIntro. Eindhoven: Eindhoven University of Technology.

BIM Forum. (2015). *LOD Specification 2015.* - : BIM Forum.

Borrmann, A., Beetz, J., Koch, C., & Liebich, T. (2015). Industry Foundation Classes. In -, - (pp. 1 - 39). -: -.

Bråthen, K., & Moum, A. (2015). Bridging the gap: taking BIM to the construction site . *CIB W78 conference 2015.* Eindhoven.

buildingSMART. (2017, 27 05). *buildingSMART* . Retrieved from open standards the basics: http://buildingsmart.org/standards/technical-vision/open-standards-101/

Business Dictionary. (2017, 04 03). *Information*. Retrieved from Business Dictionary: http://www.businessdictionary.com/definition/information.html

Davies, R., & Harty, C. (2013). Implementing 'Site BIM': A case study of ICT innovation on a large hospital project. *Automation in Construction*, 15-24.

De Vries, B. (2016, February). Reshaping the future. *!IntervisiE*, p. 22.

Devadason, F. J. (2008). A Methodology for the Identification of Information Needs of Users, Potential users and Non-users. *Kansas Library Conference* (pp. 1- 48). Wichita, KS: The Foundation for any Library and Information Activity.

Eastman, C., Teichholz, P., Sacks, R., & Liston, K. (2011). *BIM Handbook* (2nd ed.). New Jersey: John Wiley & Sons.

Fowler, M. (2003). *UML Distilled Third Edition.* New Jersey: Pearson Education.

Grady, J. O. (1995). *System Engineering Planning and Enterprise Identity.* Taylor & Francis.

Hermans, R. (2015). *Informatievoorziening in het voorbereidingsproces van houten gevelkozijnen binnen een BIM-proces.* Eindhoven: TU/e.

Herzum, P., & Sims, O. (2000). *Business Components Factory: A Comprehensive Overview of Component-Based Development for the Enterprise.* New York: John Wiley & Sons, Inc.

Hewage, K. N., & Ruwanpura, J. Y. (2009). A novel solution for construction on-site communication - the information booth. *Canadian Journal of Civil Engineering*, 659 - 671.

Hopman, J. (2016). Afstudeeronderzoek BIM in de realisatiefase. Rotterdam: Hogeschool Rotterdam.

IfcOpenShell. (2017, 08 10). *IfcOpenShell*. Retrieved from IfcOpenShell-python: http://ifcopenshell.org/python.html

Jernigan, F. (2010). *BIG BIM, little bim.* FAIA.

Kimoto, K., Endo, K., Iwashita, S., & Fujiwara, M. (2005). The application of PDA as a mobile computing system on construction management. *Automation in Construction*, 500-511.

Koedoot, P. (2016, 12 09). LinkedIn Discussion BIM on the construction site.

Lee, D.-E., & Shi, J. J. (2006). Construction Business Automation System. *Journal of Construction Engineering and Management*, 88 - 96.

Limpan, R. R. (2004). Mobile 3D visualization for steel structures. *Automation in Construction*, 119-125.

Madden, L. (2011). *Proffessional Augmented Reality Browsers for Smartphones; Programming for Junaio. Layar and Wikitude.* West Sussex: Wiley.

Marty, R. (2014, October 30). End-To-End Collaboration Enabled by BIM Level 3.

Matthews, J., Love, P. E., Heinemann, S., Chandler, R., Rumsey, C., & Olatunj, O. (2015). Real time progress management: Re-engineering processes for cloud-based BIM in construction. *Automation in Construction*, 38-47.

McKinsey & Company. (2017, March 30). *Reinventing construction: a route to higher productivity.* Retrieved from McKinsey & Company: http://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/reinventing-construction-through-a-productivity-revolution

Meža, S., Turk, Ž., & Dolenc, M. (2014). Component based engineering of a mobile BIM-based augmented reality system. *Automation in Construction*, 1-12 .

National Institute of Building Sciences buildingSMART alliance. (2012). *National BIM Standard - United States Version 2.* Washington: National Institute of Building Sciences.

Natrop, M. (2017, 05 25). *Solidu*. Retrieved from Bim op de bouwplaats informatie aanbod: http://solidu.nl/bim-op-de-bouwplaats-informatie-aanbod/

OpenCascade. (2017, 10 08). *Opencascade*. Retrieved from https://www.opencascade.com/

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A Model for Types and Levels of Human Interaction with Automation. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS*, 286 - 297.

Park, C.-S., Lee, D.-Y., Kwon, O.-S., & Wang, X. (2013). A framework for proactive construction defect management using BIM, augmented reality and ontology-based data collection template. *Automation in Construction*, 61 - 71 .

PythonOCC. (2017, 10 08). *PythonOCC*. Retrieved from http://www.pythonocc.org/

Pythonware. (2017, 08 10). *Python Image Library*. Retrieved from http://www.pythonware.com/products/pil/

Ramaekers, B. (2013). BIM. *TVVL Magazine*, 30.

Rebolj, D., Čuš Babič, N., Magdič, A., Podbreznik, P., & Pšunder, M. (2008). Automated construction activity monitoring system. *Advanced Engineering Informatics*, 493-503.

Riverbank Computing. (2017, 08 10). *PyQt*. Retrieved from Riverbank Computing: https://www.riverbankcomputing.com/software/pyqt/intro

Riverbank computing. (2017, 08 10). *PyQt4*. Retrieved from https://www.riverbankcomputing.com/software/pyqt/download

Sacks, R., Radosavljevic, M., & Barak, R. (2010). Requirements for building information modelling based lean production management systems for construction. *Automation in Construction*, 641-655.

Shannon, C. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 623 - 656.

Shi, J., Lee, D.-E., & Kuruku, E. (2008). Task-based modeling method for construction business process modeling and automation. *Automation in Construction*, 633 - 640.

Span, T. (2015). *Het Opstellen van Taakgerichte Werkinstructies.* Eindhoven: TU/e.

Teichholz, P. (2013). Labor-Productivity Declines in the Construction Industry: Causes and Remedes (Another Look). AEQbytes.

Thomson, G. I. (1996). The need for an enterprise resource management measurement/forecasting infrastructure. *Proceedings of Resource Management and Performance Evaluation of Enterprise Computing Systems* (pp. 467 – 478). San Diego: Computer Measurement Group.

Tilley, P., & Barton, R. (1997). Design and Documentation Deficiency - Causes and Effects. *Proceedings of the First International Conference on Construction Process Re-engineering*, (pp. 703 - 712). Gold Coast, Australia.

TNO. (2005, June). *Onderzoeksrapport innovatie in de bouw.* Retrieved from TNO: https://www.tno.nl/media/2512/onderzoeksrapport-innovatie-in-de-bouw.pdf

Uematsu, Y., & Saito, H. (2009). Multiple planes based registration using 3D projective space for Augmented Reality. *Image an Vision Computing*, 1484-1496.

USP Marketing Consultancy. (2010). *Slechte informatie-uitwisseling en communicatie grootste oorzaak faalkosten.*

Van Berlo, L. A., & Natrop, M. G. (2014). BIM op de Bouwplaats: Het genereren van taakgerichte tekeningen op verzoek, vanuit de bouwkeet. Wenen.

Van Berlo, L., Helmholt, K. A., & Hoekstra, W. (2009). Augmented Reality on the Construction Site. *Augmented Reality on the Construction Site* (pp. -). Sydney: -.

Van Strien, E. (2017, May 30). *Using IfcOpenshell and pythonOCC to generate cross sections directly from an IFC file*. Retrieved from IfcOpenShell Academy: http://academy.ifcopenshell.org/using-ifcopenshell-and-pythonocc-to-generate-cross-sections-directly-from-an-ifc-file/

Wallinga, A. (2011). *Virtueel Bouwen.* Bouwend Nederland.

Wikipedia. (2017, May 29). *Compiler*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Compiler

Wikipedia. (2017, 04 03). *Information*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Information

Wikipedia. (2017, May 30). *Python*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Python_(programming_language)

11. APPENDICES

- Appendix 1: Interviews construction workers
- Appendix 2: Use-case diagram
- Appendix 3: Use-case description
- Appendix 4: Script TOWI generator components
- Appendix 5: Application script

Appendix 1: Interviews construction workers

Interview

Project BIM op de bouwplaats	Besproken documenten 3D afbeelding kozijn, voorbeeld TGWI
Datum gesprek 23-03-2017	Onderwerp Proces kozijnen stellen
Interviewer Leon Vergeldt	Functie Timmerman
Geïnterviewde(n) Timmerploeg Adriaans	Duur: +/- 0,75 uur

1 Opening

1.1 BIM op de bouwplaats

Ik ben bezig een afstudeeronderzoek over de verbetering van de informatie voor de bouwvakkers. Een relatief nieuwe ontwikkeling is het fenomeen taakgerichte werkinstructies. Dit houdt in dat traditionele informatie zoals werktekeningen, aanzichten, doorsneden, detailboekjes, etc. worden vervangen door één overzicht met daarop alle informatie die benodigd is om de taak uit te voeren. In dit interview wil ik te weten komen welke informatie benodigd, gewenst, of overbodig is om een taak uit te voeren. In dit geval het stellen van kozijnen.

1.2 Wat is uw rol en wat zijn uw verantwoordelijkheden?

Timmerman. Ik houd me vooral bezig met de maatvoering en het stellen van profielen voor het stellen van kozijnen en de metselaars.

2 Informatie op de bouwplaats

2.1 Gericht op de informatie, wat zijn de grootste problemen die u dagelijks in de praktijk tegenkomt?

Zoekwerk in de verschillende informatiebronnen. Niet alle informatie ter beschikking à naar uitvoerderskeet lopen. Informatie voor specifieke situatie lastig te vinden.

2.2 Wat zijn de grootste nadelen van de huidige informatievoorziening?

Veel zoekwerk naar de juiste, samenhangende informatie.

2.3 Wat vind je van taakgerichte informatie?

Zoals je het beschrijft klinkt dat wel handig. Je hebt dan alle benodigde informatie op één overzicht. Hierdoor hoef je minder te bladeren door de verschillende informatiebronnen. Ik denk dat het wel even wennen is. Ook is dat veel werk voor de architecten. Ze voegen nu al vaak maten niet toe, omdat het veel werk is om deze toe te voegen.

2.4 Wat is van belang voor een taakgerichte werkinstructie?

Dat alle informatie erop staat. Dus de juiste maatvoeringen, diepte, hoogte, en horizontale maatvoering vanuit de assen of het hoekpunt waar de profiel komt te staan. Het metselwerk is door de architect op koppenmaat gemaatvoerd. Daartussen komen de kozijnen.

2.5 Wat vind je van een stappenplan zoals Lego- of Ikea-handleiding voor uitvoering?

Niet gevraagd.

Werkwijze stellen kozijnen:

We gaan als volgt te werk bij het maatvoeren van kozijnen.

1. Stellen profielen op hoeken en e.v.t. dilatatievoegen
2. Overnemen meterpeil naar buitenkant binnenmuur
3. Overnemen horizontale maatvoering naar buitenkant binnenmuur
4. Bepalen hoogte kozijnankers door middel van optellen/aftrekken ten opzichte van meterpeil, inclusief spouwlat/stelkozijn.
5. Boren kozijnankers aan de hand van horizontale en verticale maatvoering aan de onderkant en één zijkant. Kozijnankers waterpas stellen.
6. Boren kozijnankers aan andere zijde, inclusief speling voor plaatsen kozijnen
7. Kozijnmerk op binnenwand schrijven
8. Kozijn met juiste kozijnmerk naar locatie hijsen
9. Kozijn tijdelijk monteren op juiste hoogte en horizontale positionering
10. Kozijn op juiste diepte (negge) en waterpas monteren, aan de hand van metselkoorden aan stelprofielen. Dus buitenkant kozijn ten opzichte van buitenkant buitenmetselwerk.

2.6 Welke van de volgende informatie gebruik je voor het stellen van kozijnen?

- Plattegrond	X
- Aanzicht	X
- Doorsnede	
- Detailboekje	X
- Kozijndetailboekje	X
- Kozijnstaat	
- Productinformatie	
- Werkinstructies	
- Werkplan	
- Planning	
- Anders, namelijk:	

2.7 Welke informatie heb je wel en welke niet van de verschillende informatiebronnen nodig? Dus welke informatie heb je écht nodig om het (stel) kozijn te kunnen plaatsen? Welke informatie is overbodig en welke is gewenst.

0= Onbelangrijk
1= Gewenst
2= Belangrijk
3= Essentieel

Plattegrond:

Wel: Belangrijkheid (0 t/m 3):

Maatvoering t.o.v. hoeken of assen (3)
Oriëntatie (3)
Gevelpakket (3)

Niet: Belangrijkheid (0 t/m 3):

Aanzicht:

Wel: Belangrijkheid (0 t/m 3):

Kozijnmerk (3)

Peilmaat onderkant kozijn (3)
E.v.t Negge (2)
Bijbehorend detail (3)

Niet: Belangrijkheid (0 t/m 3):

Detailboekje kozijnenfabrikant:
Wel: Belangrijkheid (0 t/m 3):
Negge (3)
Afmetingen spouwlat/stelkozijn (3)

Niet: Belangrijkheid (0 t/m 3):

Anders, namelijk: Belangrijkheid (0 t/m 3):

2.8 Is er nog informatie die je mist? Zo ja, hoe belangrijk zou je dit vinden?

Nee

2.9 Wat vind je van de overzichtelijkheid van de informatie? Wat is er duidelijk en wat is er minder duidelijk?

2.10 Wat zou er verbeterd kunnen worden ten behoeve van de overzichtelijkheid?

Samenhang
Soms krijg je kozijntekeningen met de bijbehorende details op hetzelfde blad. Dat geeft veel overzichtelijkheid en een stuk minder zoekwerk naar de samenhang tussen de verschillende informatiebronnen.

Interview

Project	Besproken documenten
BIM op de bouwplaats	-
Datum gesprek 29-03-2017	Onderwerp Proces stellen kozijnen
Interviewer Leon Vergeldt	Functie Timmerman
Geïnterviewde(n) Timmermannen Bots Bouwgroep	Duur: 30 minuten

1 Opening

1.1 BIM op de bouwplaats

Ik ben bezig een afstudeeronderzoek over de verbetering van de informatie voor de bouwvakkers. Een relatief nieuwe ontwikkeling is het fenomeen taakgerichte werkinstructies. Dit houdt in dat traditionele informatie zoals werktekeningen, aanzichten, doorsneden, detailboekjes, etc. worden vervangen door één overzicht met daarop alle informatie die benodigd is om de taak uit te voeren. In dit interview wil ik te weten komen welke informatie benodigd, gewenst, of overbodig is om een taak uit te voeren. In dit geval het stellen van kozijnen.

1.2 Wat is uw rol en wat zijn uw verantwoordelijkheden?

Timmerman

2 Informatie op de bouwplaats

2.1 Gericht op de informatie, wat zijn de grootste problemen die u dagelijks in de praktijk tegenkomt?

Informatie die niet overeenkomt. Veel verschillende tekeningen nodig.

2.2 Wat zijn de grootste nadelen van de huidige informatievoorziening?

Veel verschillende tekeningen nodig. Niet altijd alle informatie bij de hand waardoor we bij de uitvoerder in de keet moeten kijken.
Tegenwoordig gebruiken ze ook al veel tablets, dan heb je in ieder geval alle informatie en hoef je niet telkens terug naar de keet voor meer informatie.

2.3 Wat vind je van taakgerichte informatie?

Dat zou wel handig zijn, dan hoef je minder in de papieren te zoeken.

2.4 Wat is van belang voor een taakgerichte werkinstructie?

De tekeningen moeten dan wel kloppen. Daarnaast is het belangrijk dat je weet over welk kozijn je het hebt.

2.5 Wat vind je van een stappenplan zoals Lego- of Ikea-handleiding voor uitvoering?

Niet gevraagd.

2.6 Welke van de volgende informatie gebruik je voor het stellen van kozijnen?

- Plattegrond	x	
- Aanzicht		
- Doorsnede		
- Detail(boekje)	x	
- Kozijnstaat	x	+ kozijndetails
- Productinformatie		
- Werkinstructies		
- Werkplan		
- Planning		
- Anders, namelijk:		

2.7 Welke informatie heb je <u>wel en welke niet</u> van de verschillende informatiebronnen nodig? Dus welke informatie heb je <u>écht</u> nodig om het (stel) kozijn te kunnen plaatsen? Welke informatie is overbodig en welke is gewenst.

0= Onbelangrijk
1= Gewenst
2= Belangrijk
3= Essentieel

Plattegrond:
Wel: Belangrijkheid (0 t/m 3):
Horizontale maatvoering vanuit stramien (1)
Merk (3)
Oriëntatie (3)
Gevelpakket(3)
Bij dit project is de horizontale maatvoering minder belangrijk omdat de maatvoering op basis van de sparing in de betonnen wand wordt gedaan. Er wordt hier met wild verband gemetseld, dus de maatvoering van de architect is niet zo strikt, dan wanneer halfsteens verband toegepast zou worden. Hier is de sparing maatgevend.

Niet: Belangrijkheid (0 t/m 3):
Rest

Detailboekje:
Wel: Belangrijkheid (0 t/m 3):
Negge (3)
We komen net van de uitvoerderskeet af omdat we de neggemaat van de kozijnen moesten weten. Deze staat bij de uitvoerder in het detailboekje van de architect.

Niet: Belangrijkheid (0 t/m 3):

Kozijnstaat:
Wel: Belangrijkheid (0 t/m 3):
Merk (3)
Vorm (3)
Ankers (3)
Verticale maatvoering bovenkant (stel)kozijn (3)
Verticale maatvoering onderkant (stel)kozijn (3)

Niet: Belangrijkheid (0 t/m 3):

Anders, namelijk: Belangrijkheid (0 t/m 3):

2.8 Is er nog informatie die je mist? Zo ja, hoe belangrijk zou je dit vinden?

Het zou fijn zijn als de neggemaat in het kozijnenboekje staat, want nu moeten we dit elke keer bij de uitvoerder in de keet gaan opzoeken.

2.9 Wat vind je van de overzichtelijkheid van de informatie? Wat is er duidelijk en wat is er minder duidelijk?

Veel verschillende tekeningen nodig, dat leidt tot veel zoekwerk.

2.10 Wat zou er verbeterd kunnen worden ten behoeve van de overzichtelijkheid?

Wat jij voorstelt, dus alle benodigde informatie op één blad.

Interview

Project	Besproken documenten
BIM op de bouwplaats	-
Datum gesprek 05-04-2017	Onderwerp Proces stellen kozijnen
Interviewer Leon Vergeldt	Functie Timmerman
Geïnterviewde(n) Timmermannen BurgtBouw	Duur: 40 min

1 Opening

1.1 BIM op de bouwplaats

Ik ben bezig een afstudeeronderzoek over de verbetering van de informatie voor de bouwvakkers. Een relatief nieuwe ontwikkeling is het fenomeen taakgerichte werkinstructies. Dit houdt in dat traditionele informatie zoals werktekeningen, aanzichten, doorsneden, detailboekjes, etc. worden vervangen door één overzicht met daarop alle informatie die benodigd is om de taak uit te voeren. In dit interview wil ik te weten komen welke informatie benodigd, gewenst, of overbodig is om een taak uit te voeren. In dit geval het stellen van kozijnen.

1.2 Wat is uw rol en wat zijn uw verantwoordelijkheden?

Timmerman. Als de uitvoerder er niet is, zijn wij het aanspreekpunt op de bouwplaats. Verder doen we productiewerk.

2 Informatie op de bouwplaats

2.1 Gericht op de informatie, wat zijn de grootste problemen die u dagelijks in de praktijk tegenkomt?

Informatie die niet overeenkomt. Laatst hadden we hier bijvoorbeeld een detail van de architect dat niet was bijgewerkt. Hierdoor hebben we het anders gemaakt, dan dat het moest worden. Dit heeft tot veel herstelwerkzaamheden geleid. De informatie kwam dus niet overeen tussen de verschillende informatiebronnen.

2.2 Wat zijn de grootste nadelen van de huidige informatievoorziening?

Sommige informatie is maanden oud, terwijl andere recent gemaakt is. Wij moeten het doen met de informatie die we krijgen.

2.3 Wat vind je van taakgerichte informatie?

Twijfel, het MOET dan kloppen. Wij ontdekken nog vaak fouten doordat we de werktekeningen, details en aanzichten ‘controleren’. Bij taakgerichte informatie moet je volledig op de informatie vertrouwen. Er treden vaak nog wijzigingen op, die moeten dan wel verwerkt zijn.
Met name op grotere bouwplaatsen zal dat interessant zijn, omdat bouwvakkers vaak minder betrokken zijn bij het project, waardoor ze minder thuis zijn in de tekeningen van dat project. Als ze dan één overzicht krijgen, dan scheelt dat extra veel uitzoekwerk.

2.4 Wat is van belang voor een taakgerichte werkinstructie?

Dat de informatie correct en te vertrouwen is.
Op de bouwplaats werken we in ieder geval nog graag met papieren tekeningen.

2.5 Wat vind je van een stappenplan zoals Lego- of Ikea-handleiding voor uitvoering?

Niet gevraagd.

2.6 Welke van de volgende informatie gebruik je voor het stellen van kozijnen?

- Plattegrond	x
- Aanzicht	x
- Doorsnede	
- Detail(boekje)	x
- Kozijnstaat	x
- Productinformatie	
- Werkinstructies	
- Werkplan	
- Planning	
- Anders, namelijk:	

2.7 Welke informatie heb je wel en welke niet van de verschillende informatiebronnen nodig? Dus welke informatie heb je écht nodig om het (stel) kozijn te kunnen plaatsen? Welke informatie is overbodig en welke is gewenst.

0= Onbelangrijk
1= Gewenst
2= Belangrijk
3= Essentieel

Plattegrond:
Wel: Belangrijkheid (0 t/m 3):
Horizontale maatvoering (1).
Niet van toepassing bij dit project omdat sparingen in wanden maatgevend zijn.

Niet: Belangrijkheid (0 t/m 3):

Aanzicht:
Wel: Belangrijkheid (0 t/m 3):
Merk (3)
Hoogtemaat onderkant kozijn t.o.v. peil
Hoogtemaat bovenkant kozijn t.o.v. peil

Niet: Belangrijkheid (0 t/m 3):

Detailboekje:
Wel: Belangrijkheid (0 t/m 3):
Negge (3)

Niet: Belangrijkheid (0 t/m 3):

Kozijnstaat:

Wel: Belangrijkheid (0 t/m 3):

Afmetingen spouwlat (3)

Niet: Belangrijkheid (0 t/m 3):

Anders, namelijk: Belangrijkheid (0 t/m 3):

2.8 Is er nog informatie die je mist? Zo ja, hoe belangrijk zou je dit vinden?

Nee niet dat ik weet.

2.9 Wat vind je van de overzichtelijkheid van de informatie? Wat is er duidelijk en wat is er minder duidelijk?

We zijn het zo gewend.

2.10 Wat zou er verbeterd kunnen worden ten behoeve van de overzichtelijkheid?

Misschien zoals jij zegt alle informatie op één blad, maar dat ligt aan de grootte van het project. Voor ons is de huidige informatie duidelijk omdat we gewend zijn om zo te werken.

Interview

Project BIM op de bouwplaats	Besproken documenten 3D-afbeelding kozijn
Datum gesprek 23-03-2017	Onderwerp Proces stellen kozijnen
Interviewer Leon Vergeldt	Functie Timmerman
Geïnterviewde(n) Timmerploeg Hurks	Duur: +/-1 uur

1 Opening

1.1 BIM op de bouwplaats

Ik ben bezig een afstudeeronderzoek over de verbetering van de informatie voor de bouwvakkers. Een relatief nieuwe ontwikkeling is het fenomeen taakgerichte werkinstructies. Dit houdt in dat traditionele informatie zoals werktekeningen, aanzichten, doorsneden, detailboekjes, etc. worden vervangen door één overzicht met daarop alle informatie die benodigd is om de taak uit te voeren. In dit interview wil ik te weten komen welke informatie benodigd, gewenst, of overbodig is om een taak uit te voeren. In dit geval het stellen van kozijnen.

1.2 Wat is uw rol en wat zijn uw verantwoordelijkheden?

Timmerman. Verantwoordelijk voor het stellen van kozijnen op de juiste locatie en ervoor zorgen dat de metselploeg vooruit kan door middel van het stellen van profielen.

2 Informatie op de bouwplaats

2.1 Gericht op de informatie, wat zijn de grootste problemen die u dagelijks in de praktijk tegenkomt?

Verschillende informatiebronnen die niet overeenkomen. Informatie verspreid over meerdere verschillende informatiebronnen. Het komt ook weleens voor dat verkeerde kozijnen worden geplaatst, doordat de stickers op de kozijnen niet blijken te kloppen. Met name bij kozijnen in spiegelbeeld en andersom gaat dit geregeld fout. Ook vervelend is het wanneer de kozijnmerken van de architect verschillen met die van de kozijnenfabrikant. Dit leidt tot veel zoekwerk en verwarring. Bij dit project gaat dit toevallig een keer goed.

2.2 Wat zijn de grootste nadelen van de huidige informatievoorziening?

Dat de informatie verspreid is over verschillende bronnen. Verder moeten we best wel veel dingen zelf bereken omdat de maatvoering vaak vanuit de kozijnlocatie wordt gemeten en niet vanuit het stelkozijn of spouwlat. Deze moeten we dan weer op-/ of aftellen.

2.3 Wat vind je van taakgerichte informatie?

Goed concept. Je hebt dan alle benodigde informatie op een blad, waardoor je niet meer door de verschillende informatiebronnen hoeft te zoeken. Het nadeel ervan is dat je dan weer een reeks met blaadjes krijgt, maar die alleen voor die ene specifieke taak zijn. Voor het stellen van het metselwerk is weer andere informatie nodig, zo krijg je een reeks aan boeken die je allemaal maar voor één taak kan gebruiken.

2.4 Wat is van belang voor een taakgerichte werkinstructie?

Het is heel erg belangrijk dat je weet over welk kozijn het gaat. Dus bij welk gebouw en waar in dat gebouw. Op een grote tekening heb je het overzicht dat je weet waar je bent en weet waar je het over hebt.

De volgende informatie is van belang voor tgwi: hoogtemaatvoering, onderkant stelkozijn/spouwlat (niet kozijn) ten opzichte van peilmaat. Het kozijnmerk. Horizontale maatvoering ten opzichte van assen/stramienen. De neggemaat, dus afstand van buitenkant kozijn tot buitenkant metselwerk. Welke ankers gebruikt moeten worden.

2.5 Wat vind je van een stappenplan zoals Lego- of Ikea-handleiding voor uitvoering?

Opzicht wel handig, maar dat is een nieuwe manier van bouwen. Ik denk dat de bouw best wel traditioneel is. Zoiets zal wel wennen zijn.

2.6 Welke van de volgende informatie gebruik je voor het stellen van kozijnen?

- Plattegrond	x
- Aanzicht	x
- Doorsnede	
- Detailboekje	x
- Kozijndetailboekje	x
- Kozijnstaat	
- Productinformatie	
- Werkinstructies	
- Werkplan	
- Planning	
- Anders, namelijk:	

2.7 Welke informatie heb je <u>wel en welke niet</u> van de verschillende informatiebronnen nodig? Dus welke informatie heb je <u>écht</u> nodig om het (stel) kozijn te kunnen plaatsen? Welke informatie is overbodig en welke is gewenst.

0= Onbelangrijk
1= Gewenst
2= Belangrijk
3= Essentieel

Plattegrond:
Wel: Belangrijkheid (0 t/m 3):
Stramienen (3)
Maatvoering kozijnen ten opzichte van stramienen (3)
Wandenpakket (3)
Kozijnen (3)
Maatvoering stramienen (2)
Binnenmuren (2)
Trappen(1) (ter oriëntatie)
Verdieping (3) (ter oriëntatie)

Niet: Belangrijkheid (0 t/m 3):
De rest (0)
Installaties (0)

Aanzicht:
Wel: Belangrijkheid (0 t/m 3):
Kozijnmerk (3)
Peilmaat onderkant kozijn (3)
Detailnummer (3)
Negge (3)

Locatie/oriëntatie (3)

Niet: Belangrijkheid (0 t/m 3):
Renvooi(0)

Detailboekje kozijnfabrikant:
Wel: Belangrijkheid (0 t/m 3):
Neggemaat (3)
Merk (3)
Dikte spouwlat (3)

Niet: Belangrijkheid (0 t/m 3):
Specificaties kozijn (0)

Detailboekje:
Wel: Belangrijkheid (0 t/m 3):
Extra informatie indien onduidelijkheden.
Neggemaat (3)

Niet: Belangrijkheid (0 t/m 3):
Wandopbouw (0)

Anders, namelijk: Belangrijkheid (0 t/m 3):
n.v.t.

2.8 Is er nog informatie die je mist? Zo ja, hoe belangrijk zou je dit vinden?

Gewicht is niet belangrijk omdat tegenwoordig alle kozijnen met de hijskraan worden geplaatst. Planning is niet interessant. Ik hoef niet te weten wanneer ik deze taak uit moet voeren.

2.9 Wat vind je van de overzichtelijkheid van de informatie? Wat is er duidelijk en wat is er minder duidelijk?

Er staat te veel informatie op, waardoor het soms verwarrend en een zoekwerk is.

2.10 Wat zou er verbeterd kunnen worden ten behoeve van de overzichtelijkheid?

Irrelevante informatie weghalen
Relevante informatie samenvoegen

Interview

Project BIM op de bouwplaats	Besproken documenten Werktekening, aanzichten, kozijndetails
Datum gesprek 05-04-2017	Onderwerp Informatievoorziening stellen kozijnen
Interviewer Leon Vergeldt	Functie Uitvoerder
Geïnterviewde(n) Jan Staps, Uitvoerder Stam + De Koning	Duur: 50 min

1 Opening

1.1 BIM op de bouwplaats

Ik ben bezig een afstudeeronderzoek over de verbetering van de informatie voor de bouwvakkers. Een relatief nieuwe ontwikkeling is het fenomeen taakgerichte werkinstructies. Dit houdt in dat traditionele informatie zoals werktekeningen, aanzichten, doorsneden, detailboekjes, etc. worden vervangen door één overzicht met daarop alle informatie die benodigd is om de taak uit te voeren. In dit interview wil ik te weten komen welke informatie benodigd, gewenst, of overbodig is om een taak uit te voeren. In dit geval het stellen van kozijnen.

1.2 Wat is uw rol en wat zijn uw verantwoordelijkheden?

Uitvoerder, aansturen en organiseren van de bouwplaats.

2 Informatie op de bouwplaats

2.1 Gericht op de informatie, wat zijn de grootste problemen die u dagelijks in de praktijk tegenkomt?

Veel zoekwerk tussen veel verschillende tekeningen.
Maatvoeringen die niet overeenkomen tussen verschillende documenten (werktekening + wanduitslag)

2.2 Wat zijn de grootste nadelen van de huidige informatievoorziening?

Veel verschillende tekeningen nodig
Verwarrend en onduidelijk. De tekeningen staan zó vol met informatie, waarvan veel informatie irrelevant is. Er staan ook vaak veel maatvoeringen op, maar niet degene die jij nodig hebt.
Eigenlijk zou iedereen met 3D modellen om moeten kunnen gaan, dan zouden de bouwvakkers veel beter geïnformeerd kunnen worden.

2.3 Wat vind je van taakgerichte informatie?

Dat lijkt me erg goed. Nu hebben de timmermannen veel verschillende documenten nodig en moeten ze zelf de informatie interpreteren. Het zou zeker prettig zijn als alle informatie op één blad zou staan.
Wat jij laat zien zou zeker kunnen werken.

2.4 Wat is van belang voor een taakgerichte werkinstructie?

Dat alle informatie erop staat: onderkant kozijn, bovenkant kozijn, negge, merk, horizontale maatvoering, geveltekening voor oriëntatie. 3D-model voor oriëntatie zou ook kunnen.

2.5 Wat vind je van een stappenplan zoals Lego- of Ikea-handleiding voor uitvoering?

Ja dat zou wel kunnen werken ja.

2.6 Welke van de volgende informatie gebruik je voor het stellen van kozijnen?

- Plattegrond	x	
- Aanzicht	x	
- Doorsnede		
- Detail(boekje)	x	
- Kozijnstaat	x	
- Productinformatie		
- Werkinstructies		
- Werkplan		
- Planning		
- Anders, namelijk:	x	Detailboek kozjinfabrikant

2.7 Welke informatie heb je <u>wel en welke niet</u> van de verschillende informatiebronnen nodig? Dus welke informatie heb je <u>écht</u> nodig om het (stel) kozijn te kunnen plaatsen? Welke informatie is overbodig en welke is gewenst.

0= Onbelangrijk
1= Gewenst
2= Belangrijk
3= Essentieel

Plattegrond:
Wel: Belangrijkheid (0 t/m 3):
Horizontale maatvoering vanuit stramien (3)
Merk (3)
Oriëntatie (3)

Niet: Belangrijkheid (0 t/m 3):

Aanzicht:
Wel: Belangrijkheid (0 t/m 3):
Merk (3)
Peilmaat O.K. Kozijn (3)
Peilmaat B.K. Kozijn (3)
Oriëntatie (3)

Niet: Belangrijkheid (0 t/m 3):

Detailboekje:
Wel: Belangrijkheid (0 t/m 3):
Negge (3)

Niet: Belangrijkheid (0 t/m 3):

Anders, namelijk: Detailboek kozijnfabrikant Belangrijkheid (0 t/m 3):
Afmetingen spouwlatten (3)
Merk (3)

2.8 Is er nog informatie die je mist? Zo ja, hoe belangrijk zou je dit vinden?

Type metselwerk en lagenmaat, gewenst

2.9 Wat vind je van de overzichtelijkheid van de informatie? Wat is er duidelijk en wat is er minder duidelijk?

Er staat erg veel informatie op, waardoor je overweldigd wordt door alle informatie. Er staan zó veel maten op de tekeningen. Daarnaast zijn er ook erg veel tekeningen, waardoor je snel de verkeerde pakt.

2.10 Wat zou er verbeterd kunnen worden ten behoeve van de overzichtelijkheid?

Wat jij voorstelt. Dat alle informatie op één blad staat, dat zou erg handig zijn. Je moet er dan niet té veel informatie op zetten, alleen wat nodig is.

Interview

Project BIM op de bouwplaats	Besproken documenten Voorbeeld TGWI, Ikea-handleiding
Datum gesprek 03-04-2017	Onderwerp Productieproces stellen kozijnen
Interviewer Leon Vergeldt	Functie Voorman + timmerman
Geïnterviewde(n) Voorman + timmerman Teunissen BV	Duur: 60 min

1 Opening

1.1 BIM op de bouwplaats

Ik ben bezig een afstudeeronderzoek over de verbetering van de informatie voor de bouwvakkers. Een relatief nieuwe ontwikkeling is het fenomeen taakgerichte werkinstructies. Dit houdt in dat traditionele informatie zoals werktekeningen, aanzichten, doorsneden, detailboekjes, etc. worden vervangen door één overzicht met daarop alle informatie die benodigd is om de taak uit te voeren. In dit interview wil ik te weten komen welke informatie benodigd, gewenst, of overbodig is om een taak uit te voeren. In dit geval het stellen van kozijnen.

1.2 Wat is uw rol en wat zijn uw verantwoordelijkheden?

Maatvoering en productiewerk

2 Informatie op de bouwplaats

2.1 Gericht op de informatie, wat zijn de grootste problemen die u dagelijks in de praktijk tegenkomt?

De informatie is (te) laat aanwezig, informatie tussen verschillende bronnen komt niet overeen, geen tijd om te verdiepen in het geheel waardoor je geen overzicht over het project hebt, onjuiste informatie.

2.2 Wat zijn de grootste nadelen van de huidige informatievoorziening?

Niet alle informatie beschikbaar

2.3 Wat vind je van taakgerichte informatie?

Goed plan. Net als bij het lijmen van kalkzandstenen elementen, krijg je dan overzichten met alle informatie die je nodig hebt. Dit werkt erg handig, is duidelijk en overzichtelijk.

2.4 Wat is van belang voor een taakgerichte werkinstructie?

De informatie moet volledig en juist zijn. Het zou handig zijn als de informatie ingesealed zou zijn.

2.5 Wat vind je van een stappenplan zoals Lego- of Ikea-handleiding voor uitvoering?

Als het duidelijk is, dan is dat zeker handig.

2.6 Welke van de volgende informatie gebruik je voor het stellen van kozijnen?

- Plattegrond	x
- Aanzicht	x
- Doorsnede	
- Detail(boekje)	x
- Kozijnstaat	x
- Productinformatie	
- Werkinstructies	
- Werkplan	
- Planning	
- Anders, namelijk:	

2.7 Welke informatie heb je wel en welke niet van de verschillende informatiebronnen nodig? Dus welke informatie heb je écht nodig om het (stel) kozijn te kunnen plaatsen? Welke informatie is overbodig en welke is gewenst.

0= Onbelangrijk
1= Gewenst
2= Belangrijk
3= Essentieel

Plattegrond:
Wel: Belangrijkheid (0 t/m 3):
Horizontale maatvoering (3)

Niet: Belangrijkheid (0 t/m 3):

Aanzicht:
Wel: Belangrijkheid (0 t/m 3):
Merk (3)
Orientatie(3)
Peilmaat O.K. Kozijn (3)
Peilmaat B.K. Kozijn (3)

Niet: Belangrijkheid (0 t/m 3):

Detailboekje:
Wel: Belangrijkheid (0 t/m 3):
Negge (3)

Niet: Belangrijkheid (0 t/m 3):

Kozijnstaat:

Wel: Belangrijkheid (0 t/m 3):

Afmetingen spouwlat/stelkozijn (3)

Niet: Belangrijkheid (0 t/m 3):

Anders, namelijk: Belangrijkheid (0 t/m 3):

2.8 Is er nog informatie die je mist? Zo ja, hoe belangrijk zou je dit vinden?

Nee, maar als de papieren ingesealed waren, dan zou dat fijn zijn.

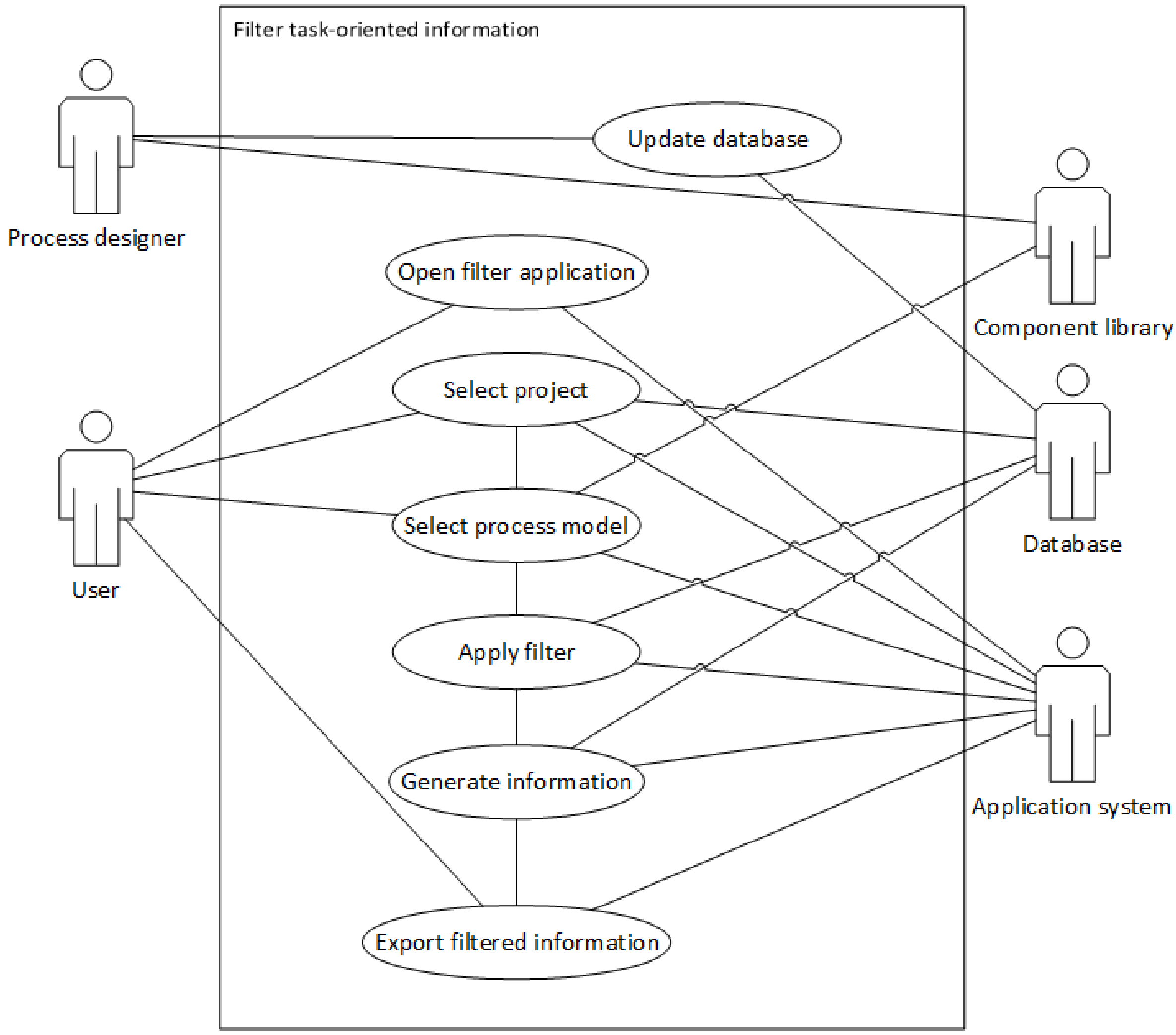
2.9 Wat vind je van de overzichtelijkheid van de informatie? Wat is er duidelijk en wat is er minder duidelijk?

Informatie staat verspreid over verschillende documenten. Op elk project is het weer anders. De ene keer staat de informatie op het ene document, de andere keer op een ander document. Er is vooraf geen duidelijkheid waar de informatie gevonden kan worden.

2.10 Wat zou er verbeterd kunnen worden ten behoeve van de overzichtelijkheid?

Zoals ik al aangaf, voor het lijmen van wanden hebben we een lijmboekje. Daarop staat per wand alle informatie die je nodig hebt. Dit werkt erg overzichtelijk. Je hebt daarvoor alleen het lijmboekje nodig. Andere informatiebronnen zijn niet nodig.

Appendix 2: Use case diagram


Filter task-oriented information
Process designer
Update database
Component library
Open filter application
Select project
User
Database
Select process model
Apply filter
Generate information
Application system
Export filtered information

Appendix 3: Use case description

Use-case	Task-oriented information filter
Actors	Process designer Component library User Database Application system
Description	Open the application, apply filters and filter the database(s). Then, export task-oriented information
Pre-conditions	Database online Database up-to-date
Flow of events	1. User opens application 2. User selects project 3. User selects process model 4. Application applies filters 5. Application generates information 6. User prints exported filtered information
Exception flow	-
Post-conditions	-
Result	User has task-oriented information

Appendix 4: Script TOWI generator

```
import OCC.Geom

import OCC.Graphic3d
import OCCUtils
import OCC.Bnd
import OCC.BRepBndLib
import OCC.BRep
import OCC.BRepAlgoAPI
import OCC.BRepBuilderAPI
import OCC.ShapeAnalysis
import OCC.TopTools
import OCC.TopoDS

from OCC.Quantity import *
from OCC.TCollection import *
from OCC.Aspect import *

from OCC.TopoDS import topods
from OCCUtils import wire
import ifcopenshell
import ifcopenshell.geom

#imports for LengthDimension
import OCC.AIS
import OCC.gp
from OCC.Display.SimpleGui import init_display
from OCC.BRepPrimAPI import BRepPrimAPI_MakeBox

#imports for images
import PIL.Image
import PIL.ImageOps

import os
import inspect
import time

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


def product_shapes_list(ifc_file):
    products = ifc_file.by_type("IfcProduct")
    product_shapes = []
    for product in products:
        if product.is_a("IfcOpeningElement") or product.is_a("IfcSite") or
product.is_a("IfcSpace") or product.is_a("IfcFurnishingElement"):
            continue
        else:
            try:
                shape = ifcopenshell.geom.create_shape(settings,
product).geometry
                product_shapes.append((product, shape))
            except:
                continue
    return product_shapes


# Geometrical functions
def get_global_x_coordinate_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
```

```
    Global_X_coor_selection=product.transformation.matrix.data[9]
    return Global_X_coor_selection

def get_global_y_coordinate_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Global_Y_coor_selection=product.transformation.matrix.data[10]
    return Global_Y_coor_selection

def get_global_z_coordinate_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Global_Z_coor_selection=product.transformation.matrix.data[11]
    return Global_Z_coor_selection

def get_orig_x_direction_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Orig_X_dir_selection=product.transformation.matrix.data[0]
    return Orig_X_dir_selection

def get_orig_y_direction_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Orig_Y_dir_selection=product.transformation.matrix.data[1]
    return Orig_Y_dir_selection

def get_orig_z_direction_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Orig_Z_dir_selection=product.transformation.matrix.data[2]
    return Orig_Z_dir_selection

def get_x_direction_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    X_dir_selection=product.transformation.matrix.data[6]
    return X_dir_selection

def get_y_direction_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Y_dir_selection=product.transformation.matrix.data[7]
    return Y_dir_selection

def get_z_direction_selection(selection):
    settings = ifcopenshell.geom.settings()
    product = ifcopenshell.geom.create_shape(settings, selection)
    Z_dir_selection=product.transformation.matrix.data[8]
    return Z_dir_selection

def get_selection_related_floor_z(selection,minmeasurement): # Find the
object-related floor level height
    if not None:

floor_base_Z=selection.ObjectPlacement.RelativePlacement.Location.Coordinat
es[2]

floor_base_Z=selection.ObjectPlacement.PlacementRelTo.RelativePlacement.Loc
ation.Coordinates[2]
```

```
floor_base_Z=selection.ObjectPlacement.PlacementRelTo.PlacementRelTo.RelativePlacement.Location.Coordinates[2]

floor_base_Z=(selection.ObjectPlacement.PlacementRelTo.PlacementRelTo.PlacementRelTo.RelativePlacement.Location.Coordinates[2])*minmeasurement
    return floor_base_Z

# Additional information functions
def show_property_value(display,selection,property,x,y,z,name,ifc_file): # Show property value
    try:
        for PropertyDefinition in selection.IsDefinedBy:
            for i in ifc_file.by_type("IfcRelDefinesByType"):
                if PropertyDefinition == i:
                    for i in PropertyDefinition.RelatingType.HasPropertySets:
                        for propertyset in ifc_file.by_type("IfcPropertySet"):
                            if propertyset == i:
                                for propertysinglevalue in propertyset.HasProperties:
                                    # For example 'Type Mark' or 'FrameOffset - External'
                                    if propertysinglevalue.Name == property:
                                        if True:
                                            message=name + '= '+str(propertysinglevalue.NominalValue[0])

display.DisplayMessage(OCC.gp.gp_Pnt(x,y,z),str(message),height=15,message_color=(0,0,0),update=False,top=True)
                                        elif True:
                                            message=name + '= '+str(propertysinglevalue.Unit[0])

display.DisplayMessage(OCC.gp.gp_Pnt(x,y,z),str(message),height=15,message_color=(0,0,0),update=False,top=True)
                                        else:
                                            print "property unknown"

    except:
        print "no property"

def show_additional_text(canvas,product,x,y,z,message,size):

canvas.DisplayMessage(OCC.gp.gp_Pnt(x,y,z),str(message),height=size,message_color=(0,0,0),update=True,top=True)


def set_lengthdimension_settings(display):# LengthDimension settings
    black = Quantity_Color(Quantity_NOC_BLACK)

display.Context.DefaultDrawer().GetObject().DimensionAspect().GetObject().ArrowAspect().GetObject().SetLength(0.1)

display.Context.DefaultDrawer().GetObject().DimensionAspect().GetObject().SetCommonColor(black)

display.Context.DefaultDrawer().GetObject().DimensionAspect().GetObject().SetArrowTailSize(0.1) # extension of the arrow tail line
```

```
def set_line_settings(display): # Line settings ( make the lines black)
    black = Quantity_Color(Quantity_NOC_BLACK)

display.Context.DefaultDrawer().GetObject().LineAspect().GetObject().SetCol
or(black)

def set_grid_setting(display):# Grid settings ( make the gridlines dashed)

display.Context.DefaultDrawer().GetObject().LineAspect().GetObject().SetTyp
eOfLine(Aspect_TOL_DOTDASH) # Linetype

display.Context.DefaultDrawer().GetObject().LineAspect().GetObject().SetWid
th(0.3) # Line width


# View settings
def set_wireframe(canvas): # Set the view to Wireframe
    canvas.SetModeWireFrame()

def set_iso_view(display): # Set the view to Isometric
    display.View_Iso()

def set_top_view(display): # Set the view to Top
    display.View_Top()

def set_section_view(display,selection): # Set the view perpendicular to
the selected object
    if get_orig_x_direction_selection(selection) >0:
        xdir=get_orig_x_direction_selection(selection)*(-1)
    else:
        xdir=get_orig_x_direction_selection(selection)

    if get_orig_y_direction_selection(selection) >0:
        ydir=get_orig_y_direction_selection(selection)
    else:
        ydir=get_orig_y_direction_selection(selection)*(-1)
    display.View.SetUp(0,0,1)
    display.View.SetAt(xdir,ydir,get_orig_z_direction_selection(selection))
    display.View.SetEye(0,0,0)

def set_elevation_view(canvas,selection): # Set the view perpendicular to
the selected object
    if get_orig_x_direction_selection(selection) >0:
        coordx=get_orig_x_direction_selection(selection)*(-1)
    else:
        coordx=get_orig_x_direction_selection(selection)*(-1)
    if get_orig_y_direction_selection(selection) >0:
        coordy=get_orig_y_direction_selection(selection)
    else:
        coordy=get_orig_y_direction_selection(selection)

    canvas.View.SetUp(0,0,1)

canvas.View.SetAt(coordy,coordx,get_orig_z_direction_selection(selection))
    canvas.View.SetEye(0,0,0)

def set_fit_all_view(display): # Set the view to fit all
    display.FitAll()

def erase_view(display): # Erase all objects on the view
    display.EraseAll()
```

```
# 3D specific functions
def show_grey_3D_shapes(display, product,product_shapes):
    for shape in product_shapes:
        grey=Quantity_Color(Quantity_NOC_MATRAGRAY)
        ifcopenshell.geom.utils.display_shape(shape[1],grey,
viewer_handle=display)

def mark_selected_object_black(display,product,product_shapes):
    for shape in product_shapes:
        if shape[0].GlobalId == product.GlobalId:
            ifcopenshell.geom.utils.display_shape(shape[1],
viewer_handle=display)


# Section specific functions
def set_section_cut_location_x(selection):
    distance_from_object=0.1
    return (distance_from_object*get_orig_x_direction_selection(selection))

def set_section_cut_location_y(selection):
    distance_from_object=0.1
    return (distance_from_object*get_orig_y_direction_selection(selection))

def set_section_height(selection):
    section_height = (get_global_z_coordinate_selection(selection)+0.1)
    return section_height

def make_section(display, selection,section_face,product_shapes):
    # Each product of the building is intersected with the horizontal face
    for product, shape in product_shapes:
        try:
            section = OCC.BRepAlgoAPI.BRepAlgoAPI_Section(section_face,
shape).Shape()
            # The edges of the intersection are stored in a list
            section_edges = list(OCCUtils.Topo(section).edges())
            if len(section_edges) > 0:
                edges = OCC.TopTools.TopTools_HSequenceOfShape()
                edges_handle =
OCC.TopTools.Handle_TopTools_HSequenceOfShape(edges)
                wires = OCC.TopTools.TopTools_HSequenceOfShape()
                wires_handle =
OCC.TopTools.Handle_TopTools_HSequenceOfShape(wires)
                # The edges are copied to the sequence
                for edge in section_edges:
                    edges.Append(edge)
                # A wire is formed by connecting the edges

OCC.ShapeAnalysis.ShapeAnalysis_FreeBounds.ConnectEdgesToWires(edges_handle
, 1e-5, True, wires_handle)
                wires = wires_handle.GetObject()
                # From each wire a face is created
                for i in range(wires.Length()):
                    wire_shape = wires.Value(i+1)
                    wire = topods.Wire(wire_shape)
                    face =
OCC.BRepBuilderAPI.BRepBuilderAPI_MakeFace(wire).Face()
                    # Display the wires and make them black
                    black=Quantity_Color(Quantity_NOC_BLACK)
                    display.DisplayColoredShape(wire,black)
```

```
        except:
            continue

def
make_grid(canvas,selection,grid,grid_base_x,grid_base_y,vertical_axis_list,
horizontal_axis_list,section_plane,maxmeasurement,minmeasurement,UMIN,UMAX,
VMIN,VMAX):
    planey1=(get_global_y_coordinate_selection(selection)+VMIN)
    planey2=(get_global_y_coordinate_selection(selection)+VMAX)
    planex1=(get_global_x_coordinate_selection(selection)+UMIN)
    planex2=(get_global_x_coordinate_selection(selection)+UMAX)
    try:
        for axis in grid.UAxes:
            # Set grid coordinates
            x1= axis.AxisCurve.Points[0].Coordinates[0]
            x2= axis.AxisCurve.Points[1].Coordinates[0]
            y1=((get_global_y_coordinate_selection(selection)+VMIN)-0.5)
            y2=((get_global_y_coordinate_selection(selection)+VMAX)+0.5)
            distance_axis_basepoint_x1= (grid_base_x + x1)*minmeasurement
            distance_axis_basepoint_x2= (grid_base_x + x2)*minmeasurement

            # Put axis in list
            vertical_axis_list.append(distance_axis_basepoint_x1)

            #Calculate nearest vertical grid
            nearest_vertical=min(vertical_axis_list, key=lambda x:abs(x-
get_global_x_coordinate_selection(selection)))

            #Makes the line for the grid
            if distance_axis_basepoint_x1 >= planex1 and
distance_axis_basepoint_x1 <= planex2:
                gridpoint1=OCC.gp.gp_Pnt(distance_axis_basepoint_x1,y1,0)
                gridpoint2=OCC.gp.gp_Pnt(distance_axis_basepoint_x1,y2,0)
                gridpoint1_handle =
OCC.Geom.Geom_CartesianPoint(gridpoint1).GetHandle()
                gridpoint2_handle =
OCC.Geom.Geom_CartesianPoint(gridpoint2).GetHandle()
                gridline=OCC.AIS.AIS_Line(gridpoint1_handle
,gridpoint2_handle)
                canvas.Context.Display(gridline.GetHandle(),True)

                # Show gridlabel pnt1
                x=(distance_axis_basepoint_x1+0.1)
                y=y1
                z=0
                message=axis.AxisTag
                size=25
                show_additional_text(canvas,selection,x,y,z,message,size)

                # Show gridlabel pnt2
                x=(distance_axis_basepoint_x1+0.1)
                y=(y2-0.2)
                show_additional_text(canvas,selection,x,y,z,message,size)

            if get_orig_x_direction_selection(selection) != 0:
                p3 =
OCC.gp.gp_Pnt(nearest_vertical,get_global_y_coordinate_selection(selection)
,set_section_height(selection))
                p4 =
OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection),get_global_y_coo
rdinate_selection(selection),set_section_height(selection))
```

```
                ld2=OCC.AIS.AIS_LengthDimension(p3,p4,section_plane)

                TP2 = ld2.GetTextPosition()
                diff = p3.XYZ() - TP2.XYZ()
                diff.SetZ(0)
                diff = diff / 30. * 29 #distance is 15m/30*29 = 0.5 meter
                TP2 = OCC.gp.gp_Pnt(TP2.XYZ() + diff)

TP2.SetX((nearest_vertical+(get_global_x_coordinate_selection(selection)))/
2)
                ld2.SetTextPosition(TP2)


current_value_metres_ld=OCC.AIS.AIS_LengthDimension.GetValue(ld2)

OCC.AIS.AIS_LengthDimension.SetCustomValue(ld2,(current_value_metres_ld*max
measurement))


            else:
                continue
        try:
            canvas.Context.Display(ld2.GetHandle(),True)
        except:
            print 'This is not an X-axis grid'

        for axis in grid.VAxes:
            # Set grid coordinates
            x1=((get_global_x_coordinate_selection(selection)+UMIN)-0.5)
            x2=((get_global_x_coordinate_selection(selection)+UMAX)+0.5)
            y1= axis.AxisCurve.Points[0].Coordinates[1]
            y2= axis.AxisCurve.Points[1].Coordinates[1]
            distance_axis_basepoint_y1=(grid_base_y + y1)*minmeasurement
            distance_axis_basepoint_y2=(grid_base_y + y2)*minmeasurement

            # Put axis in list
            horizontal_axis_list.append(distance_axis_basepoint_y1)

            # Calculate nearest horizontal axis
            nearest_horizontal=min(horizontal_axis_list, key=lambda
x:abs(x-get_global_y_coordinate_selection(selection)))

            #Makes the line for the grid
            if distance_axis_basepoint_y1 >= planey1 and
distance_axis_basepoint_y1 <= planey2:
                gridpoint1=OCC.gp.gp_Pnt(x1,distance_axis_basepoint_y1,0)
                gridpoint2=OCC.gp.gp_Pnt(x2,distance_axis_basepoint_y1,0)
                gridpoint1_handle =
OCC.Geom.Geom_CartesianPoint(gridpoint1).GetHandle()
                gridpoint2_handle =
OCC.Geom.Geom_CartesianPoint(gridpoint2).GetHandle()
                gridline=OCC.AIS.AIS_Line(gridpoint1_handle
,gridpoint2_handle)
                canvas.Context.Display(gridline.GetHandle(),True)
                # Show gridlabel pnt1
                x=(x1)
                y=(distance_axis_basepoint_y1+0.1)
                z=0
                message=axis.AxisTag
                size=25
                show_additional_text(canvas,selection,x,y,z,message,size)
```

```
                # Show gridlabel pnt2
                x=(x2)
                y=(distance_axis_basepoint_y1+0.1)
                show_additional_text(canvas,selection,x,y,z,message,size)

            # Make the lengthdimension
            if get_orig_y_direction_selection(selection) != 0:
                p3 =
OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection),nearest_horizont
al,set_section_height(selection))
                p4 =
OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection),get_global_y_coo
rdinate_selection(selection),set_section_height(selection))
                ld2=OCC.AIS.AIS_LengthDimension(p3,p4,section_plane)

                TP2 = ld2.GetTextPosition()
                diff = p3.XYZ() - TP2.XYZ()
                diff.SetZ(0)
                diff = diff / 20 * 19 #distance is 15m/30*29 = 0.5 meter
                TP2 = OCC.gp.gp_Pnt(TP2.XYZ() + diff)

TP2.SetY((nearest_horizontal+(get_global_y_coordinate_selection(selection))
)/2)
                ld2.SetTextPosition(TP2)


current_value_metres_ld=OCC.AIS.AIS_LengthDimension.GetValue(ld2)

OCC.AIS.AIS_LengthDimension.SetCustomValue(ld2,(current_value_metres_ld*max
measurement))

        try:
            canvas.Context.Display(ld2.GetHandle(),True)
        except:
            print 'This is not an Y-axis grid'

    except:
        print "no grid available"

# Execute the script
# Show the 3D view including settings
def
create_3D_view(canvas,selection,maxmeasurement,minmeasurement,normalmeasure
ment,ifc_file):
    show_grey_3D_shapes(canvas, selection,product_shapes_list(ifc_file))
    mark_selected_object_black(canvas,
selection,product_shapes_list(ifc_file))
    set_wireframe(canvas)
    # set_elevation_view(canvas,selection)
    set_iso_view(canvas)
    set_fit_all_view(canvas)

    # Set coordinates for property + show property value
    x=(get_global_x_coordinate_selection(selection)-0.5)
    y=(get_global_y_coordinate_selection(selection)-0.5)

z=(get_global_z_coordinate_selection(selection)+((selection.OverallHeight*m
inmeasurement)))+0.5
    name=' '
    show_property_value(canvas,selection,'Type Mark',x,y,z,name,ifc_file)
```

```
    # Make the image
    global threeDview
    threeDview=os.getcwd()+'\\TOWI\ThreeDView.png'
    canvas.View.Dump(threeDview)
    # Delete the view
    erase_view(canvas)

def create_elevation_view(canvas,selection,minmeasurement,ifc_file):
    show_grey_3D_shapes(canvas, selection,product_shapes_list(ifc_file))

mark_selected_object_black(canvas,selection,product_shapes_list(ifc_file))
    set_elevation_view(canvas,selection)
    set_fit_all_view(canvas)


    # Set coordinates for property + show property value
    x=(get_global_x_coordinate_selection(selection)-0.2)
    y=(get_global_y_coordinate_selection(selection)-0.2)

z=(get_global_z_coordinate_selection(selection)+((selection.OverallHeight*m
inmeasurement)/2))
    name=' '
    show_property_value(canvas,selection,'Type Mark',x,y,z,name,ifc_file)
    # Make the image
    global elevationview
    elevationview=os.getcwd()+'\\TOWI\Elevationview.png'
    canvas.View.Dump(elevationview)
    # Delete the view
    erase_view(canvas)

# Make and show the section view including settings
def
create_section(canvas,selection,maxmeasurement,minmeasurement,normalmeasure
ment,ifc_file):
    set_lengthdimension_settings(canvas)

    #First make a section plane to assign where the intersection should be
made
    section_plane = OCC.gp.gp_Pln(

OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection)+set_section_cut_
location_x(selection),
get_global_y_coordinate_selection(selection)+set_section_cut_location_y(sel
ection), get_global_z_coordinate_selection(selection)),
    OCC.gp.gp_Dir(get_orig_x_direction_selection(selection),
get_orig_y_direction_selection(selection),
get_orig_z_direction_selection(selection))
    )

    # Make the section face and assign its dimensions
    UMIN = -2.6
    UMAX = 2.6
    VMIN = -1
    VMAX = 1
    section_face =
OCC.BRepBuilderAPI.BRepBuilderAPI_MakeFace(section_plane, UMIN, UMAX, VMIN,
VMAX).Face()

    # Make the section

make_section(canvas,selection,section_face,product_shapes_list(ifc_file))
```

```
    # Show additional text
    x=(get_global_x_coordinate_selection(selection)+0.4)
    y=(get_global_y_coordinate_selection(selection)+0.4)
    z=(get_global_z_coordinate_selection(selection))
    message="Peilmaat O.K. Kozijn=
"+str(int((get_global_z_coordinate_selection(selection)*maxmeasurement)))+
" mm"
    size=15
    show_additional_text(canvas,selection,x,y,z,message,size)

    # Show additional text
    x=(get_global_x_coordinate_selection(selection)+0.4)
    y=(get_global_y_coordinate_selection(selection)+0.4)

z=(get_global_z_coordinate_selection(selection)+(selection.OverallHeight*mi
nmeasurement))
    message="Peilmaat B.K. Kozijn=
"+str(int(((get_global_z_coordinate_selection(selection)*maxmeasurement)+(s
election.OverallHeight*normalmeasurement))))+" mm"
    size=15
    show_additional_text(canvas,selection,x,y,z,message,size)

    # Show additional text
    x=(get_global_x_coordinate_selection(selection)+0.4)
    y=(get_global_y_coordinate_selection(selection)+0.4)
    z=(get_selection_related_floor_z(selection,minmeasurement)+0.2)
    message="Maten zijn exclusief spouwlatten/stelkozijnen !!"
    size=15
    show_additional_text(canvas,selection,x,y,z,message,size)

    # Set coordinates for property + show property value
    x=(get_global_x_coordinate_selection(selection)+0.4)
    y=(get_global_y_coordinate_selection(selection)+0.4)

z=(get_global_z_coordinate_selection(selection)+((selection.OverallHeight*m
inmeasurement)/2))
    name='Kozijnmerk '
    show_property_value(canvas,selection,'Type Mark',x,y,z,name,ifc_file)

    # Set coordinates for property + show property value
    x=(get_global_x_coordinate_selection(selection)+0.4)
    y=(get_global_y_coordinate_selection(selection)+0.4)
    z=(get_selection_related_floor_z(selection,minmeasurement)+0.5)
    name='Negge '
    show_property_value(canvas,selection,'FrameOffset -
External',x,y,z,name,ifc_file)

    # Set the view
    set_section_view(canvas,selection)
    set_fit_all_view(canvas)

    # Make the image
    global sectionview
    sectionview=os.getcwd()+'\\TOWI\SectionView.png'
    canvas.View.Dump(sectionview)
    erase_view(canvas)
```

```
def
create_plan(canvas,selection,maxmeasurement,minmeasurement,normalmeasuremen
t,ifc_file):
    set_lengthdimension_settings(canvas)
    set_line_settings(canvas)
    set_grid_setting(canvas)

    #First make a section plane to assign where the intersection should be
made
    section_plane = OCC.gp.gp_Pln(
        OCC.gp.gp_Pnt(get_global_x_coordinate_selection(selection),
get_global_y_coordinate_selection(selection),
set_section_height(selection)),
        OCC.gp.gp_Dir(get_x_direction_selection(selection),
get_y_direction_selection(selection), get_z_direction_selection(selection))
    )
    # Make the section face and assign its dimensions
    UMIN = -6
    UMAX = 6
    VMIN = -6
    VMAX = 6
    section_face =
OCC.BRepBuilderAPI.BRepBuilderAPI_MakeFace(section_plane, UMIN, UMAX, VMIN,
VMAX).Face()
    # Make the section

make_section(canvas,selection,section_face,product_shapes_list(ifc_file))

    # Show additional text
    x=(get_global_x_coordinate_selection(selection)+0.2)
    y=(get_global_y_coordinate_selection(selection)+0.6)
    z = 0
    message="Maten zijn exclusief spouwlatten/stelkozijnen !!"
    size = 15
    show_additional_text(canvas,selection,x,y,z,message,size)

    # Grid settings
    grid=ifc_file.by_type("IfcGrid")[0]

grid_base_x=grid.ObjectPlacement.PlacementRelTo.PlacementRelTo.RelativePlac
ement.Location.Coordinates[0]

grid_base_y=grid.ObjectPlacement.PlacementRelTo.PlacementRelTo.RelativePlac
ement.Location.Coordinates[1]

    # Empty lists
    vertical_axis_list=[]
    horizontal_axis_list=[]
    # Make the grid

make_grid(canvas,selection,grid,grid_base_x,grid_base_y,vertical_axis_list,
horizontal_axis_list,section_plane,maxmeasurement,minmeasurement,UMIN,UMAX,
VMIN,VMAX)

    # Set the view
    set_top_view(canvas)
    set_fit_all_view(canvas)

    # Make the image
    global planview
    planview=os.getcwd()+'\\TOWI\PlanView.png'
```

```python
    canvas.View.Dump(planview)
    erase_view(canvas)


def create_image(selection):
    template_landscape=os.getcwd()+'\\TOWI\Template-A3-Liggend.png'
    template_portrait=os.getcwd()+'\\TOWI\Template-A3-Staand.png'

    # Een A3= 2970mm x 4200mm en 3508 pixels bij 4961 pixels
    sheet_landscape = (4963, 3509)
    sheet_portrait = (3509, 4963)

    # Coordinates (left,top,right,bottom)
    # 4-boxes landscape format:
    full_box= (0, 0, 4963, 3509)
    top_left_box = (163, 150, 2488, 1750)
    top_right_box = (2488, 150, 4813, 1750)
    bottom_left_box = (163, 1750, 2488, 3350)
    bottom_right_box = (2488, 1750, 4813, 3350)

    boxes = [
        [full_box,template_landscape],
        [top_left_box, threeDview],
        [top_right_box, planview],
        [bottom_left_box, sectionview],
        [bottom_right_box,elevationview]
    ]

    # # 3-boxes landscape format:
    # full_box= (0, 0, 4963, 3509)
    # left_box = (163, 150, 1713, 3350)
    # middle_box = (1713, 150, 3263, 3350)
    # right_box = (3263, 150, 4813, 3350)

    # boxes = [
        # [full_box,template_landscape],
        # [left_box, threeDview],
        # [middle_box, planview],
        # [right_box, sectionview]
    # ]

    # # 3-boxes portrait format
    # full_box= (0, 0, 3509, 4963)
    # top_box = (150, 163, 3350, 1713)
    # middle_box = (150, 1713, 3350, 3263)
    # bottom_box = (150, 3263, 3350, 4813)

    # # Set images to the boxes
    # boxes = [
        # [full_box,template_portrait],
        # [top_box, threeDview],
        # [middle_box, planview],
        # [bottom_box, sectionview]
    # ]

    # Resize images and paste them on an empty sheet or template
    sheet = PIL.Image.new('RGB', sheet_landscape)
    for box, img in boxes:
        img = PIL.Image.open(img)
        box_width = float(box[2] - box[0])
        box_height = float(box[3] - box[1])
```

```
        img_width = float(img.size[0])
        img_height = float(img.size[1])

        box_ratio = box_width / box_height
        img_ratio = img_width / img_height

        if box_ratio > img_ratio:
            img_width = box_height / img_height * img_width
            img_height = box_height
        else:
            img_height = box_width / img_width * img_height
            img_width = box_width

        img = img.resize((int(img_width), int(img_height)))
        sheet.paste(img, (int(box[0]), int(box[1])))

    # save and show sheet
    # For every TOWI, make a new image
    if os.path.exists('TOWI\werkinstructie.png'):
        sheet.save('TOWI\werkinstructie_{}.png'.format(int(time.time())))
    else:
        sheet.save('TOWI\werkinstructie.png')
    sheet.show()


def run(ifc_file, canvas):
    canvas.set_bg_gradient_color(255,255,255,255,255,255)
    units=ifc_file.by_type("IfcUnitAssignment")[0]
    canvas.View.TriedronErase()
    if units.Units[0].Prefix == 'MILLI':
        maxmeasurement = 1000
        minmeasurement = 0.001
        normalmeasurement = 1
    else:
        maxmeasurement = 1000
        normalmeasurement = 1000
        minmeasurement = 1

    objects = ifc_file.by_type("IfcWindow")
    object_number=0

    while object_number < len(objects):
        selection=ifc_file.by_type("IfcWindow")[object_number]
        create_3D_view(canvas,
selection,maxmeasurement,minmeasurement,normalmeasurement,ifc_file)
        create_elevation_view(canvas,selection,minmeasurement,ifc_file)
        create_section(canvas,
selection,maxmeasurement,minmeasurement,normalmeasurement,ifc_file)
        create_plan(canvas,
selection,maxmeasurement,minmeasurement,normalmeasurement,ifc_file)
        create_image(selection)
        object_number=object_number+1
```

Appendix 5: Application script

```
from __future__ import print_function

import os
import sys
import time
import operator
import functools

import OCC.AIS

from collections import defaultdict, Iterable, OrderedDict

os.environ['QT_API'] = 'pyqt4'
try:
    from pyqode.qt import QtCore
except:
    pass

from PyQt4 import QtGui, QtCore

try:
    from OCC.Display.pyqt4Display import qtViewer3d
except:
    import OCC.Display

    try:
        import OCC.Display.backend
    except:
        pass

    try:
        OCC.Display.backend.get_backend("qt-pyqt4")
    except:
        OCC.Display.backend.load_backend("qt-pyqt4")

    from OCC.Display.qtDisplay import qtViewer3d


from ifcopenshell.geom.main import settings, iterator
from ifcopenshell.geom.occ_utils import display_shape

from ifcopenshell import open as open_ifc_file
from ifcopenshell import get_supertype

# Depending on Python version and what not there may or may not be a
QString
try:
    from PyQt4.QtCore import QString
except ImportError:
    QString = str

# Imports for TOWI
import TOWI_generator

class configuration(object):
    def __init__(self):
        try:
            import ConfigParser
            Cfg = ConfigParser.RawConfigParser
        except:
            import configparser
```

```
            Cfg = configparser.ConfigParser(interpolation=None)

        self.conf_file = os.path.expanduser(os.path.join("~",
".ifcopenshell", "app", "snippets.conf"))
        if self.conf_file.startswith("~"):
            conf_file = None
            return

        self.config_encode = lambda s: s.replace("\\",
"\\\\").replace("\n", "\n|")
        self.config_decode = lambda s: s.replace("\n|",
"\n").replace("\\\\", "\\")

        if not os.path.exists(os.path.dirname(self.conf_file)):
            os.makedirs(os.path.dirname(self.conf_file))

        if not os.path.exists(self.conf_file):
            self.config = Cfg()
            self.config.add_section("snippets")

        self.config = Cfg()
        self.config.read(self.conf_file)

    def options(self, s):
        return OrderedDict([(k, self.config_decode(self.config.get(s, k)))
for k in self.config.options(s)])


class application(QtGui.QApplication):

    # """A pythonOCC, PyQt based IfcOpenShell application
    # A graphical 3d view"""

    class viewer(qtViewer3d):

        instanceSelected = QtCore.pyqtSignal([object])

        @staticmethod
        def ais_to_key(ais_handle):
            def yield_shapes():
                ais = ais_handle.GetObject()
                if hasattr(ais, 'Shape'):
                    yield ais.Shape()
                    return
                shp = OCC.AIS.Handle_AIS_Shape.DownCast(ais_handle)
                if not shp.IsNull(): yield shp.Shape()
                return
                mult = ais_handle
                if mult.IsNull():
                    shp = OCC.AIS.Handle_AIS_Shape.DownCast(ais_handle)
                    if not shp.IsNull(): yield shp
                else:
                    li = mult.GetObject().ConnectedTo()
                    for i in range(li.Length()):
                        shp =
OCC.AIS.Handle_AIS_Shape.DownCast(li.Value(i+1))
                        if not shp.IsNull(): yield shp
            return tuple(shp.HashCode(1 << 24) for shp in yield_shapes())

        def __init__(self, widget):
            qtViewer3d.__init__(self, widget)
```

```
        self.ais_to_product = {}
        self.product_to_ais = {}
        self.counter = 0
        self.window = widget

    def initialize(self):
        self.InitDriver()
        self._display.Select = self.HandleSelection

    def load_file(self, f, setting=None):

        if setting is None:
            setting = settings()
            setting.set(setting.USE_PYTHON_OPENCASCADE, True)

        v = self._display

        t = {0: time.time()}
        def update(dt = None):
            t1 = time.time()
            if t1 - t[0] > (dt or -1):
                v.FitAll()
                v.Repaint()
                t[0] = t1

        terminate = [False]
        self.window.window_closed.connect(lambda *args:
operator.setitem(terminate, 0, True))

        t0 = time.time()

        it = iterator(setting, f)
        if not it.initialize():
            return

        old_progress = -1
        while True:
            if terminate[0]: break
            shape = it.get()
            product = f[shape.data.id]
            ais = display_shape(shape, viewer_handle=v)
            ais.GetObject().SetSelectionPriority(self.counter)
            self.ais_to_product[self.counter] = product
            self.product_to_ais[product] = ais
            self.counter += 1
            QtGui.QApplication.processEvents()
            if product.is_a() in {'IfcSpace', 'IfcOpeningElement'}:
                v.Context.Erase(ais, True)
            progress = it.progress() // 2
            if progress > old_progress:
                print("\r[" + "#" * progress + " " * (50 - progress) +
"]", end="")
                old_progress = progress
            if not it.next():
                break
            update(0.2)

        print("\rOpened file in %.2f seconds%s" % (time.time() - t0, "
"*25))

        update()
```

```
        def select(self, product):
            ais = self.product_to_ais.get(product)
            if ais is None:
                return
            v = self._display.Context
            v.ClearSelected(False)
            v.SetSelected(ais, True)

        def set_color(self, product, red, green, blue):
            qclr = OCC.Quantity.Quantity_Color(red,green,blue,
OCC.Quantity.Quantity_TOC_RGB)
            ais_shape = self.product_to_ais.get(product)
            ais = ais_shape.GetObject()
            ais.SetColor(qclr)
            self.update()

        def get_selection_set(self,model):
           selection_set =[]
           for p in model.by_type("IfcProduct"):
               ais = self.product_to_ais.get(p)
               if ais != None:
                   if self._display.Context.IsSelected(ais):
                       selection_set.append(p)
           return selection_set

        def set_transparency (self, product, transp):
            ais_shape = self.product_to_ais.get(product)
            ais = ais_shape.GetObject()
            ais.SetTransparency(transp)

        def toggle(self, product_or_products, fn):
            if not isinstance(product_or_products, Iterable):
                product_or_products = [product_or_products]
            aiss = list(filter(None, map(self.product_to_ais.get,
product_or_products)))
            last = len(aiss) - 1
            for i, ais in enumerate(aiss):
                fn(ais, i == last)

        def toggle_visibility(self, product_or_products, flag):
            v = self._display.Context
            if flag:
                def visibility(ais, last):
                    v.Erase(ais, last)
            else:
                def visibility(ais, last):
                    v.Display(ais, last)
            self.toggle(product_or_products, visibility)

        def toggle_wireframe(self, product_or_products, flag):
            v = self._display.Context
            if flag:
                def wireframe(ais, last):
                    if v.IsDisplayed(ais):
                        v.SetDisplayMode(ais, 0, last)
            else:
                def wireframe(ais, last):
                    if v.IsDisplayed(ais):
                        v.SetDisplayMode(ais, 1, last)
            self.toggle(product_or_products, wireframe)
```

```
        def HandleSelection(self, X, Y):
            v = self._display.Context
            v.Select()
            v.InitSelected()
            if v.MoreSelected():
                ais = v.SelectedInteractive()
                inst =
self.ais_to_product[ais.GetObject().SelectionPriority()]
                self.instanceSelected.emit(inst)


    class window(QtGui.QMainWindow):

        TITLE = "Task-Oriented Work Instructions - Generator"

        window_closed = QtCore.pyqtSignal([])

        def __init__(self):
            QtGui.QMainWindow.__init__(self)
            self.setWindowTitle(self.TITLE)
            self.menu = self.menuBar()
            self.menus = {}

        def closeEvent(self, *args):
            self.window_closed.emit()

        def add_menu_item(self, menu, label, callback, icon=None,
shortcut=None):
            m = self.menus.get(menu)
            if m is None:
                m = self.menu.addMenu(menu)
                self.menus[menu] = m

            if icon:
                a = QtGui.QAction(QtGui.QIcon(icon), label, self)
            else:
                a = QtGui.QAction(label, self)

            if shortcut:
                a.setShortcut(shortcut)

            a.triggered.connect(callback)
            m.addAction(a)


    def makeSelectionHandler(self, component):
        def handler(inst):
            for c in self.components:
                if c != component:
                    c.select(inst)
        return handler

    def __init__(self, settings=None):
        QtGui.QApplication.__init__(self, sys.argv)
        self.window = application.window()
        self.canvas = application.viewer(self.window)

        splitter = QtGui.QSplitter(QtCore.Qt.Horizontal)
        splitter2 = QtGui.QSplitter(QtCore.Qt.Horizontal)
```

```
        splitter2.addWidget(self.canvas)


        splitter.addWidget(splitter2)
        splitter.setSizes([100,700])
        splitter2.setSizes([400,200])
        self.window.setCentralWidget(splitter)
        self.canvas.initialize()

        self.components = [self.canvas]
        self.files = {}

        self.window.add_menu_item('File', '&Open', self.browse,
shortcut='CTRL+O')
        self.window.add_menu_item('File', '&Close', self.clear,
shortcut='CTRL+W')
        self.window.add_menu_item('File', '&Exit', self.window.close,
shortcut='ALT+F4')


self.canvas.instanceSelected.connect(self.makeSelectionHandler(self.canvas)
)
        self.settings = settings

    def change_visibility(self, tree, inst, flag):
        self.canvas.toggle_visibility(insts, flag)

    def change_displaymode(self, tree, inst, flag):
        self.canvas.toggle_wireframe(insts, flag)

    def start(self):
        self.window.showMaximized()
        sys.exit(self.exec_())


    def browse(self):
        filename = QtGui.QFileDialog.getOpenFileName(self.window, 'Open
file',".","Industry Foundation Classes (*.ifc)")
        self.load(filename)

    def clear(self):
        self.canvas._display.Context.RemoveAll()
        self.files.clear()

    def load(self, fn):
        if fn in self.files: return
        f = open_ifc_file(str(fn))
        TOWI_generator.run(f, self.canvas._display)

if __name__ == "__main__":
    application().start()
```