TECHNICAL UNIVERSITY OF DENMARK

PROTOTYPING – Case Study Documentation

APPENDIX 4

Thesis:	Unlocking the value of Linked Building Data (LBD) - A Lean and integrated management process of temporary construction items (TCIs)
Chapter:	Chapter 6.2 – Case Study
Method:	Case Study (Yin, 2014)
Purpose:	This Case Study documentation will serve as a protocol for the consecutive steps for applying the developed prototype solution in a real case construction project. The documentation is structured to continue the prototyping process from the demo project. First, it introduces the case project and reveals the current state of the prototype solution. Subsequently, the different steps of applying the prototype in the case project are presented. Concluding this documentation, findings of the case study are summarized in the last chapter.
Author:	Alexander Schlachter
Student number:	182781

Table of Content

List of Figures

List of Tables

Chapter 1. Introduction

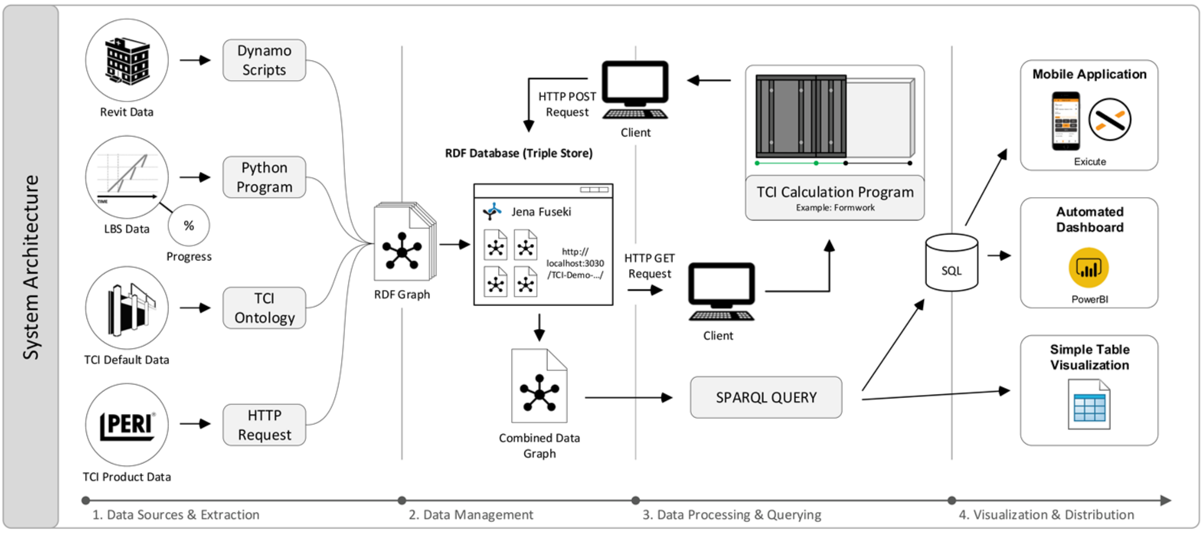


Figure 1: System Architecture of the proposed solution

The case study is part of the further development of the prototype and follows directly up on the first prototype phase, where the solution was developed with a small demo project. The case study is the second phase of prototyping and explores the applicability of the developed solution in a real case construction project and determine limitations and derive further developments of the prototype. The system architecture in figure 1 presents the solution workflow how it was developed for the demo project. In theory, the generic workflow should be applicable to all building projects with the same data sources. In the following paragraphs, the case study documentation tries to prove this assumption and documents the further development of the solution. First, the case study project is introduced in chapter 2. The focus lies on explaining the scope of the case study project as well as the different data sources, that are used. Furthermore, a further section verifies whether the simplifications of the prototype are still valid for the case study application. This will also outline the solution status that was applied in the case study. Following up on the case study introduction, chapter 3 of the case study documentation explores the different steps of applying the solution to the project. In chapter 4, the findings of the case study are derived from the application, which will explain the ongoing improvements that are made during the application as well as identify open issues and limitations of the solution. The last chapter then concludes the results of the case study.

Chapter 2. Case Study Project

By applying the prototype solution to a real construction project, the solution will be further developed, and its functionality is tested in a real construction project. In this case, the permission to use the project data was given by the public client Vejdirektoratet (The Danish Road Directorate) and the data, containing the 3D-Model and a location-based schedule, was provided by Exigo A/S. The case study is based on a public Danish construction project for a new healthcare science university faculty that is physically joint with the existing university and a new university hospital. The following table gives an overview of the project's main facts and characteristics:

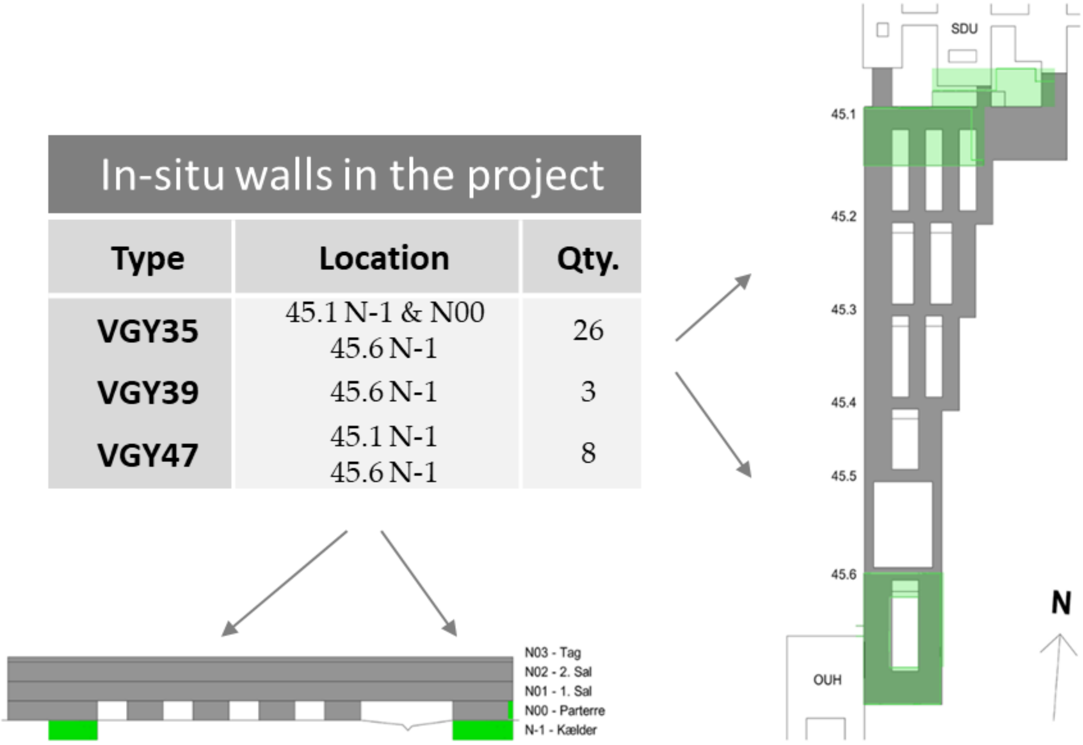
Project Name	SDU SUND
Location	Odense
Project Type	Public, New Construction, Rural
Building Type	Healthcare Science Faculty
Building Size	50.740 m²
Levels	Basement, Level 1-4
Building Sections	45.1 – 45.6
Value for Case Study	In-situ concrete walls are installed in the basement and serve as an application field for the developed prototype solution, creating a utilization plan for the required formwork
Used Data	• 3D-model (rvt-file) • Location-based schedule (vico-file)

Table 1: Case Study Project Information

As seen above, the project is divided into six building sections, reaching from 45.1 to 45.6. The following orthographic picture of the construction site from the 24.05.2020 shows the layout of these sections from the north (left) to the south (right).

Figure 2: Orthographic picture with project sections

The focus of the case study will be on the in-situ concrete walls in the basement and ground floor level of section 45.1 and the basement level of section 45.6. In total, the project contains 19 different types of in-situ walls which are all located in the specified locations. Before the solution can be applied to simply all in-situ walls, the model has to be analyzed in order to find out, if those items are actually in-situ concrete walls. This analysis revealed that most of the walls, that are specified as in-situ concrete walls, are actually parts of foundation blocks, columns, or beams. Thus, the wall selection was reduced to only contain the in-situ concrete walls which form the structural system of the basement in section 45.1 and 45.6, resulting in three different wall types that are shown in the following figure.



In-situ walls in the project

Type	Location	Qty.
VGY35	45.1 N-1 & N00 45.6 N-1	26
VGY39	45.6 N-1	3
VGY47	45.1 N-1 45.6 N-1	8

Figure 3: In-situ walls in the case study project

In the next chapter, the prototype solution is applied to these three wall types to first, calculate the formwork demand for each wall element and then generate a TCI utilization plan by linking it together with the location-based schedule. However, before going through the different steps of the case application, a verification is needed whether the case study project is aligned with simplifications of the premises that are defined for the prototype solution (see Appendix 3 – Demo-Project Documentation).

a) Formwork is chosen to represent all TCIs that can be linked to a model object

This simplification is possible with the project of the case study, as in-situ concrete walls are used in the basement of building 45.1 and 45.6 which are subject to this study.

b) One dimensional calculation of vertical formwork for the construction of concrete walls

As the wall height of the regarded in-situ walls in the case study range between 4.22 m and 5.22 m, two formwork elements with each 2.70 m are stacked on top of each other to cover all walls. Thus, the formwork quantities are slightly over-dimensioned in their height, due to this simplification.

However, as this approach also reflects the utilization of formwork elements on the actual construction site, where two stacked formwork elements of the height 2.70 m are used to cast all in-situ walls (see Appendix 2 - Photo Documentation of Site Observation), the simplification is regarded valid.

c) Height of forms is always greater than the wall height, hence no need for consideration of heights

As justified before, two stacked formwork elements with a total height of 5.40 m are calculated to cover the wall height of all regarded walls. Hence, the simplification is still valid, although the result might be optimized.

d) Utilization of only a default set of forms with specific parameters

For this thesis, the use of the default set of formwork is sufficient to justify a proof of concept as the integration of a confirmed formwork set from a selected provider is considered and can easily update the default utilization plan. The panel height of the default formwork set in this case study, in comparison to the demo project, is reduced to the standard height of 2.70 m, as a stacked approach is applied here.

e) Assumption that walls in the building model are modelled as they are constructed, meaning that the wall separation in the model represent the sequencing of the construction activity which must be covered by formwork

A total of 5 walls from the regarded set of in-situ walls are split by the location management system in VICO office. Therefore, the calculated formwork elements cannot be linked to a single element instance from VICO which contains the task time information. This means that two schedule tasks are applied two one wall of the building model and therefore, the formwork elements occur twice in the TCI utilization plan. As this issue only concerns 5 walls, the overall result is still regarded as valid. Considering the fact that the split walls result in false information in the TCI utilization plan, this issue is reviewed in chapter 4 of this documentation.

f) Corner panels are not considered in the demo project

This simplification was removed in the case study, as corner elements are now included in the proposed solution. More information is provided in the following chapter.

g) Formwork is calculated for the entire length of each modelled wall

As corner elements are included in the case study application, the formwork calculation is considered more realistically. If the wall contains a corner ending, the wall-length has to be reduced respectively with the length of the corner elements. If no corner is detected, walers and plywood should be applied at the end of a wall sequence to close the formwork system and prevent the concrete from pouring out. Thus, the formwork covering has to reach slightly over the actual wall length to apply the plywood and walers. The consideration of the wall sequence however requires to already integrate the schedule information in the calculation process of the formwork elements in order to know where to close the formwork system, and this has not yet been done in the prototype solution. Hence, the status of the prototype solution, which is applied to the case study, calculates a full range of formwork elements for the given building elements but does not consider the formwork sequence.

After verifying the prototype simplifications and presenting the current status of the solution, that is applied to the case project, the documentation provided sufficient introduction to go through the steps of the case application. During the application, the prototype is further developed and adjusted to fit the needs of a real construction project. These adjustments are documented as well in the following chapter.

Chapter 3. Case Application

This chapter explores the four different steps of the system architecture which were applied to the case project. As the solution framework was already explained in detail in the demo-project documentation, this chapter will focus on the specific application on a big scale project and the adjustments and further development of the solution during the case application. Captures of the results are presented in this documentation and the whole content of the case study and all developed programs and tools can be found in the GitHub repository LBD-for-TCI (https://github.com/Alex-Schlachter27/LBD-for-TCI/tree/master).

The next chapter will then reflect on the case study findings and draw a conclusion.

3.1 Data Sources & Extraction

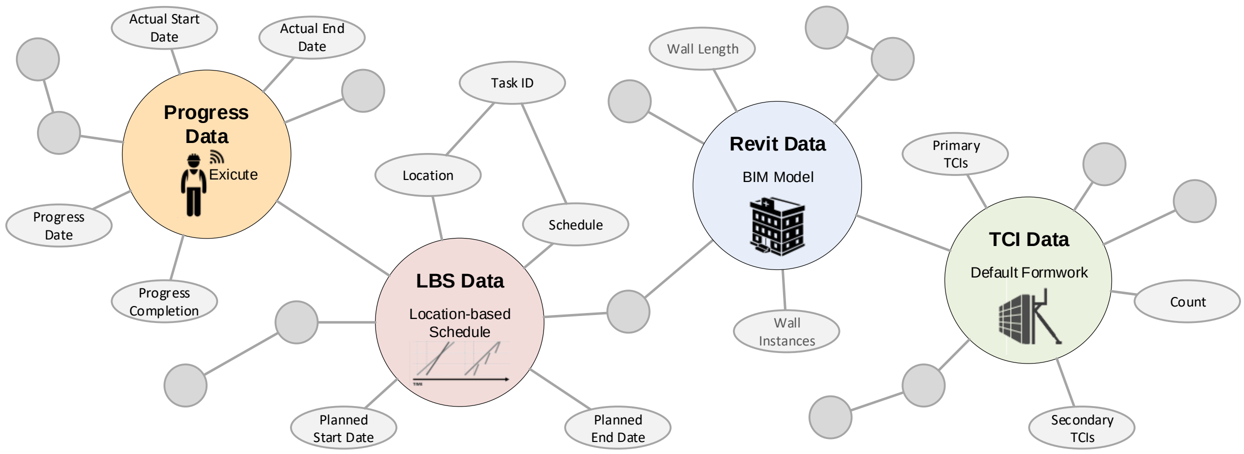


Figure 4: Data Sources with desired parameters

Figure 4 shows the different generic data sources, which are integrated into the prototype solution. In the case study, the dataset of TCIs is almost the same as the dataset in the demo project, providing the required information of the primary and secondary formwork elements in the default formwork set. The only difference is that, apart from the formwork panels, the primary formwork elements also contain standard corner elements. This allows to calculate the formwork demand more holistically and realistically, representing the real demand on the construction site. Furthermore, as mentioned earlier, the height of the formwork panels is now set to 2,70 m.

The building model of the project is a structural model, modelled in Revit, and containing all structural elements of the project. In order to only extract the walls, that are regarded in this case study, the Dynamo scripts are modified to extract only the information of the wall types VGY35, VGY38, and VGY47 (See figure 5).

Figure 5: Capture of 3D-view of the Revit model, only showing the regarded in-situ walls

This modification is applied to all Dynamo scripts that are extracting the model data, convert it into RDF-triples and write it directly to the triple store Jena Fuseki to create the Revit data graph for the case study. The following picture shows a capture of the Dynamo script that assigns the wall types to the RDF class.

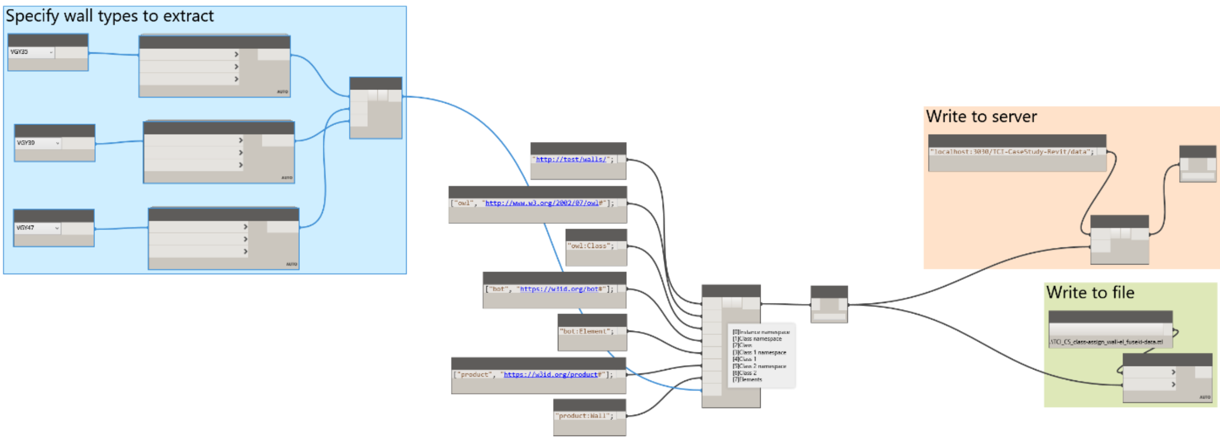


Figure 6: Dynamo script, extracting the regarded wall types

Figure 6 shows the resulting Revit graph in RDF.

```
wallinst:ef24efa4-4fef-4d22-b807-03a3af66a712-00bb15f6
        a                     product:Wall ;
        tci:pciType           product:Wall ;
        bot:adjacentElement   wallinst:ef24efa4-4fef-4d22-b807-03a3af66a712-00bb144c ,
                              wallinst:ef24efa4-4fef-4d22-b807-03a3af66a712-00bb16e9 ;
        props:Element_ID      "12260854" ;
        props:Revit_GUID      "ef24efa4-4fef-4d22-b807-03a3af66a712-00bb15f6" ;
        props:angle           90.0 ;
        props:area            38.18 ;
        props:building        "45.1" ;
        props:height          4.69 ;
        props:length          8.31 ;
        props:level_simple    "N00" ;
        props:wallType        rvt:Vgy35 .
```

Figure 7: Excerpt of the resulting Revit data graph showing one wall instance

The location-based schedule (LBS) of the project was established by Exigo A/S in the software VICO Office and contains schedule and location information of the whole project. This schedule is based on the same Revit model, and thus the same wall instances can be found in the dataset of the Revit model and the schedule. For the case study application, the schedule is not modified or adjusted to meet the requirements of the proposed solution. On the contrary, the intention is to use a common schedule as it is used in current construction projects. With this approach, the functionality of the solution can be proven, and potential limitations and further developments can be addressed based on the findings.

Figures 8 and 9 show captures of the LBS information from VICO Office. Figure 8 contains the flow-line view and figure 9 presents the location management system, which is dividing the building elements into different locations, that are used to create the schedule.

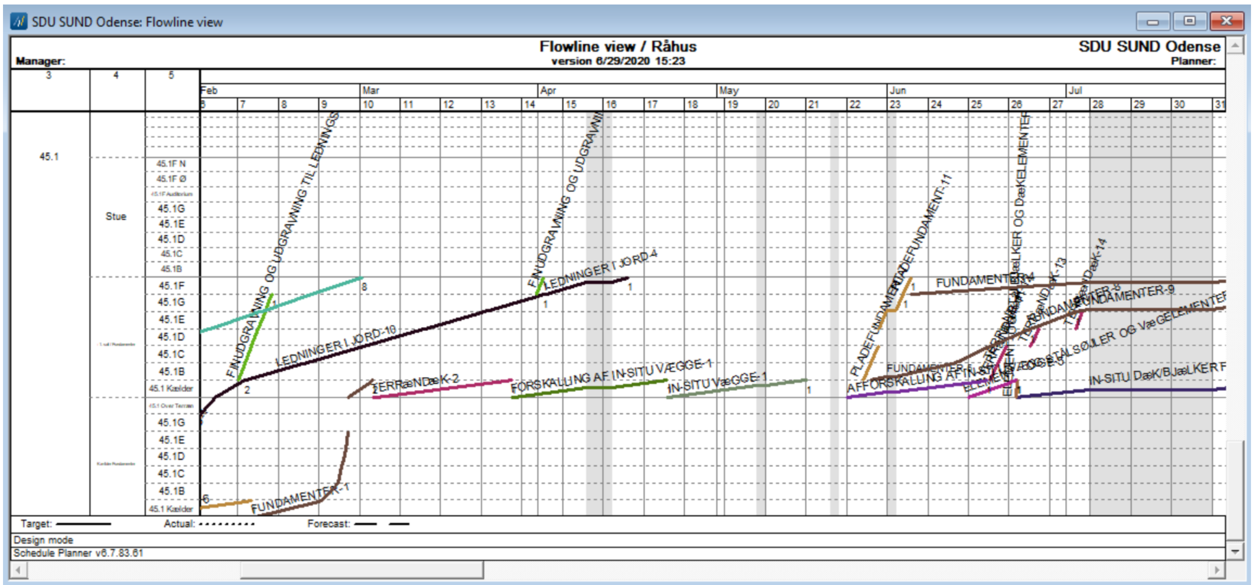


Figure 8: Location-based schedule of the case study project

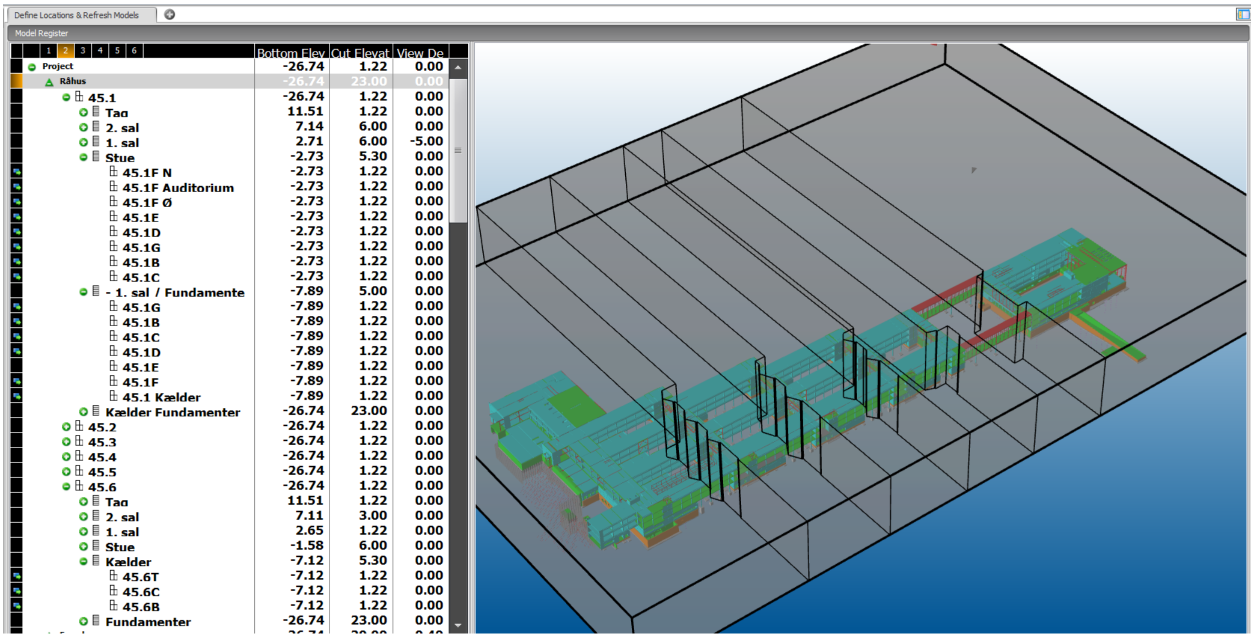


Figure 9: Location Management System of the case study project

As mentioned in the Demo-Project Documentation, a manual extraction of the LBS data was chosen for simplification as the data was quite small and handy. For a real construction project, this would require a lot of effort and would not fit in the scope of the thesis. Thus, a solution had to be developed to write the LBS data from VICO into the triple store. The solution to this problem is two-folded.

The first intention for bringing the information from VICO to the triple store was to use the software ExiLink from Exigo A/S, which was already introduced in the Demo-Project Documentation. This tool is based on a python code, that is able to go through the different datasets (Figure 10) of the VICO project, extracts the required schedule information, and writes the data in an SQL database. The structured data from the SQL database can then easily be converted into RDF-triples with an appropriate python script.

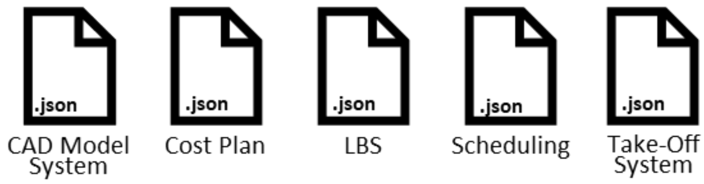


Figure 10: Datasets of the VICO project as JSON-files

As the software was more explored and workshops were held with employees from Exigo A/S, it was identified that ExiLink is only taking a few datasets from VICO in order to establish the schedule information. These datasets are "Cost Plan", "LBS" and "Scheduling" and contain all information for displaying the LBS-data in the solution of the company. The specific application in the thesis, however, requires to include the Element IDs of the wall instances in the building model in order to link the instance code of the LBS data to the same code in the Revit data.
Therefore, an additional program (**VICO-Extraction-Program.py**) has to be developed which is extracting the missing information from the VICO datasets and write it to the SQL-database. From there, the additional data, with the already existing schedule information, can be converted into RDF triples, transferred to the triple store, and integrated into the LBS data graph of the case study. This is being done with the program **VICO_SQL2RDF.js**. The key relationship between the instance code of the wall elements and the schedule information is the taskLoid, which is a generated instance code for each task. This task can be described with an individual combination of the location (locLoid) and the cost item (compLoid) in the VICO project as both items are used to create the schedule tasks and one specific location cost-item relation only describe one single task item. This means, that each task in the LBS-System can be described with the information of the location and cost-item. Furthermore, the compLoid of the cost item can be directly referenced to the wall instances that are summarized into one cost item.

Hence, the program has to extract information that contains the wall instances, as well as its relation to the compLoids and locLoids as the combination of both codes, is the key to link the wall instances to the schedule information.

The Element ID, that are originally created in Revit, are stored in the dataset "CAD Model System" and are summarized into take-off items of the same element type in the "Take-Off System". In this dataset, each wall instance in a take-off item also contains information about its location. Hence, the locLoid can be extracted in this step. The take-off items can also be found in the "Cost Plan" as they are directly linked to a cost item which is described by the compLoid. Going through the datasets "CAD Model System", "Take-Off System" and finally "Cost Plan", the developed Python-program (**VICO-Extraction-Program.py**) is able to create a SQL-table that includes the Element ID of the wall instances, the locLoid as well as the compLoid in order to link the wall instances to the tasks of the location-based schedule. This relationship between the VICO datasets is also visualized in the following figure.

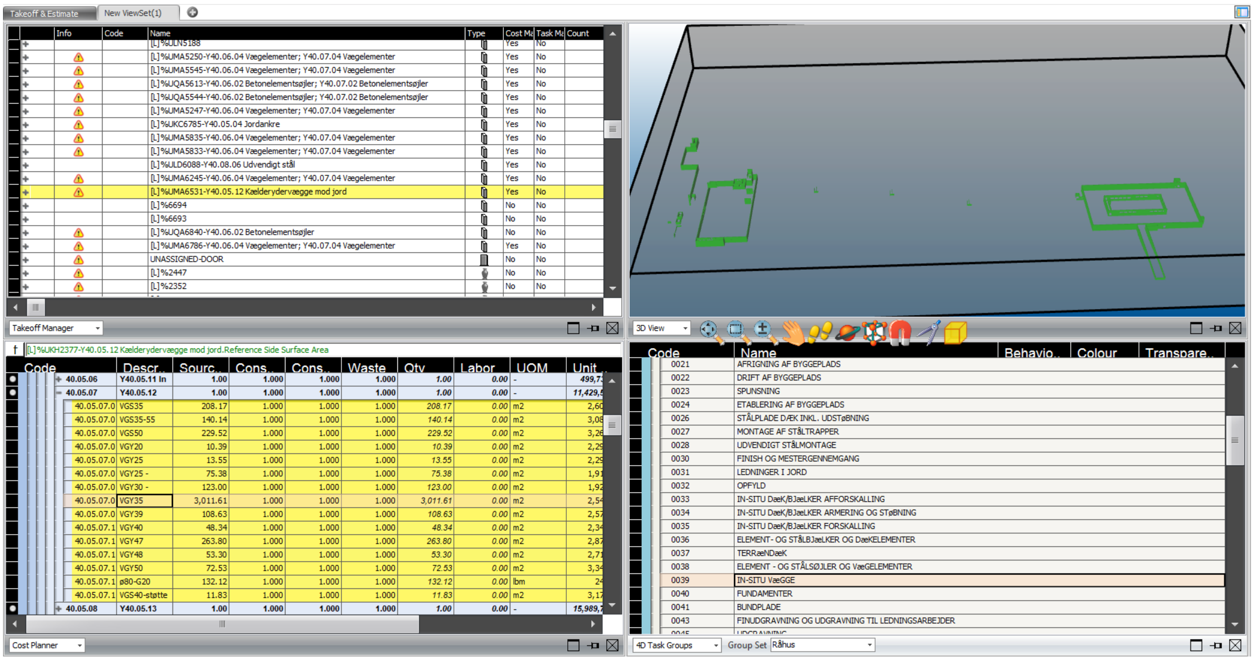


Figure 11: VICO Office interface in the case study project

Figure 11 shows the VICO Office interface with the four windows Take-Off Manager (containing the take-off items), Cost Planner (containing the cost items), and 4D Task Groups (containing the tasks). On the screen, the wall type VGY35 was selected in the Cost Planner and as it is linked to the other datasets, all items that are related to the selected wall type, are highlighted in yellow.

This proves the previously explained relation of the different datasets in the LBS software VICO. A capture of the resulting SQL-table is presented in figure 12.

ElementID	rvtGUID	locloid	toi	comploid
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15299.0.4232	15300.0.2227937	15300.0.98617492
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15300.0.45032304	15300.0.2227937	15300.0.98617492
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15300.0.53166144	15300.0.2227937	15300.0.98617492
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15300.0.53166171	15300.0.2227937	15300.0.98617492
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15300.0.53177359	15300.0.2227937	15300.0.98617492
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15300.0.53179181	15300.0.2227937	15300.0.98617492
2927850	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002cacea}	15300.0.38238460	15300.0.2227937	15300.0.98617492
2927851	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002caceb}	15299.0.4232	15300.0.2227937	15300.0.98617492
2927851	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002caceb}	15300.0.45032304	15300.0.2227937	15300.0.98617492
2927851	{13eb2559-6889-4e8e-9f20-6bb21ea0b923-002caceb}	15300.0.53168889	15300.0.2227937	15300.0.98617492

Figure 12: Resulting SQL-Table, containing the Element ID, locLoid and compLoid

As revealed in figure 12, one wall instance contains several locations. This is because VICO is assigning the wall instances to all the hierarchy levels of each location systems (shell construction and façade), the wall is located in. Therefore, each locLoid represents a different hierarchy level. In this application, however, only the lowest hierarchy level of the shell construction (Råhus) matters. This aspect is considered subsequently.

In the next step of the case study, the SQL-data, containing all required schedule information, is converted into RDF-triples and written into the LBS data graph of the triple store. Therefore, a Javascript-Program (**VICO_SQL2RDF.js**) was developed, which is able to communicate with the triple store, similar to the TCI calculation program that was developed in the demo project. Now instead of receiving the data from the triple store, calculating the TCI demands, and writing the results back to the triple store, the new program is receiving the SQL-data, converts it into RDF-triples and writes it to the empty LBS data graph. In this program, it is also specified that only the lowest hierarchy-level of the shell construction location system is considered. Furthermore, the program combines the datasets which were extracted through ExiLink with the dataset with wall instances by matching the located cost-items. By that, the wall instances are linked to the schedule tasks, which means that the LBS data graph contains all the required information for creating the TCI utilization plan by linking it to the other datasets in the Linked Data environment. Figure 15 is summarizing the data extraction process from the VICO datasets to the RDF data graph.

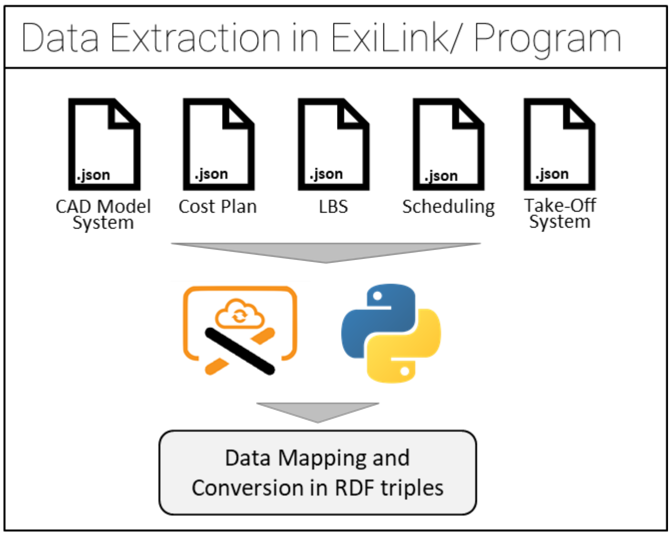


Figure 13: Visualization of the data extraction from VICO in the case study

Figure 14 is completing the data extraction from VICO, showing the resulting LBS data graph.

```
vicoinst:15300.0.104740129
        a                           lbs:VICOelement ;
        lbs:description             "Vgy35" ;
        lbs:elementType             "DERIVEDELEMENT" ;
        lbs:hasCompLoid             "15300.0.98617492" ;
        lbs:hasElementLoid          "15300.0.104740129" ;
        lbs:hasLocation             "45.1f n" ;
        lbs:haslocLoid              "15300.0.38277190" ;
        lbs:hasprojectGUID          "2e060258-64b3-41e5-b132-29d5a14ceb9f" ;
        lbs:hasschedLoid            "15300.0.99034466" ;
        lbs:hastakeOffID            "15300.0.2227937" ;
        lbs:hastaskLoid             "15300.0.99143062" ;
        lbs:hastaskName             "In-situ vÃ¦gge" ;
        lbs:taskActualEndDate       ""^^xsd:dateTime ;
        lbs:taskActualStartDate     ""^^xsd:dateTime ;
        lbs:taskPlannedEndDate      "2020-11-11T11:07:12Z"^^xsd:dateTime ;
        lbs:taskPlannedStartDate    "2020-11-06T07:57:57Z"^^xsd:dateTime ;
        lbs:taskProgressCompletion  "0"^^xsd:nonNegativeInteger ;
        lbs:taskProgressDate        ""^^xsd:dateTime ;
        props:Element_ID            "12260854" ;
        props:Revit_GUID            "{ef24efa4-4fef-4d22-b807-03a3af66a712-00bb15f6}" .
```

Figure 14: Excerpt of the resulting LBS data graph

3.2 Data Management

This chapter is exploring the data management of the proposed solution for the case study application. As data management is the same as in the first phase of the prototyping, it can be reviewed in detail in Appendix 3 – Demo-Project Documentation. The approach to combine the data in the triple store is still done manually, as the remote controlling engine called "comunica", which should allow executing SPARQL queries remotely in a Javascript-Program could not be implemented successfully.

3.3 Data Processing & Querying


```
{ wallinst: 'http://test/walls/38c866f8-c24e-4a2f-9c4c-fb0dadc921e6-0084ca43',
  ID: '8702531',
  length: 6,
  angle: 90,
  adjacentElements: [
    'http://test/walls/38c866f8-c24e-4a2f-9c4c-fb0dadc921e6-0084cab7',
    'http://test/walls/687443c2-11cd-4bb3-8210-54c6bfdfa10f-008b1e5c'],
  TCIsIn: [
    { TCIinst: 'http://test/tci/DefaultPanel270x240_0', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x240_1', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x240_2', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x240_3', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x120_0', length: 1.2 },
    { TCIinst: 'http://test/tci/DefaultPanel270x120_1', length: 1.2 }],
  TCIsOut: [
    { TCIinst: 'http://test/tci/DefaultPanel270x240_0', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x240_1', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x240_2', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x240_3', length: 2.4 },
    { TCIinst: 'http://test/tci/DefaultPanel270x120_0', length: 1.2 },
    { TCIinst: 'http://test/tci/DefaultPanel270x120_1', length: 1.2 }],
  secTCIs: [
    { TCIinst: 'http://test/tci/DefaultCoupler', weight: 4.58 },
    { TCIinst: 'http://test/tci/DefaultTieRod', weight: 4.43 },
    { TCIinst: 'http://test/tci/DefaultWingnut', weight: 2.58 },
    { TCIinst: 'http://test/tci/DefaultPushPullProp', weight: 22.8 }],
  TCIsCount: [
    { TCIinst: 'http://test/tci/DefaultPanel270x240', Count: 8 },
    { TCIinst: 'http://test/tci/DefaultPanel270x120', Count: 4 },
    { TCIinst: 'http://test/tci/TimberFilling', Count: 0 },
    { TCIinst: 'http://test/tci/DefaultCoupler', Count: 44 },
    { TCIinst: 'http://test/tci/DefaultTieRod', Count: 20 },
    { TCIinst: 'http://test/tci/DefaultWingnut', Count: 20 },
    { TCIinst: 'http://test/tci/DefaultPushPullProp', Count: 6 },
    { TCIinst: 'http://test/tci/DefaultPanel270x90', Count: 0 },
    { TCIinst: 'http://test/tci/DefaultPanel270x60', Count: 0 },
    { TCIinst: 'http://test/tci/DefaultPanel270x45', Count: 0 },
    { TCIinst: 'http://test/tci/DefaultInsideCorner270x20', Count: 0 },
    { TCIinst: 'http://test/tci/DefaultOutsideCorner270x55', Count: 0 },
    { TCIinst: 'http://test/tci/DefaultWaler', Count: 0 }],
  TimberFilling: [
    { TCIinst: 'http://test/tci/TimberFilling', Length: 0 },
    { TCIinst: 'http://test/tci/TimberFilling', Length: 0 }],
  Connections: [
    { wallinst: 'http://test/walls/38c866f8-c24e-4a2f-9c4c-fb0dadc921e6-0084cab7',
      ConnectionType: 'horizontal',
      angle: 0 },
    { wallinst: 'http://test/walls/687443c2-11cd-4bb3-8210-54c6bfdfa10f-008b1e5c',
      ConnectionType: 'horizontal',
      angle: 0 }],
  MonthlyRent: '5040.00',
  installationTimeTCI_h: '3.00',
  strippingTimeTCI_h: '12.00',
  dismantlingTimeTCI_h: '3.00' }
```


Figure 15: Output JSON Object of the TCI calculation program

Figure 15 shows the output JSON object of the TCI calculation program. The program itself is only slightly modified, compared to the version in the demo project, as it can now include corner elements. Two default corner elements are included in the dataset - DefaultInsideCorner270x20 & DefaultOutsideCorner270x55. In order to visualize the result of the TCI calculation program, the following figure shows an example wall structure, where the formwork elements are applied.

Each comment field contains information about the wall instance and type as well as its demand for primary formwork elements.

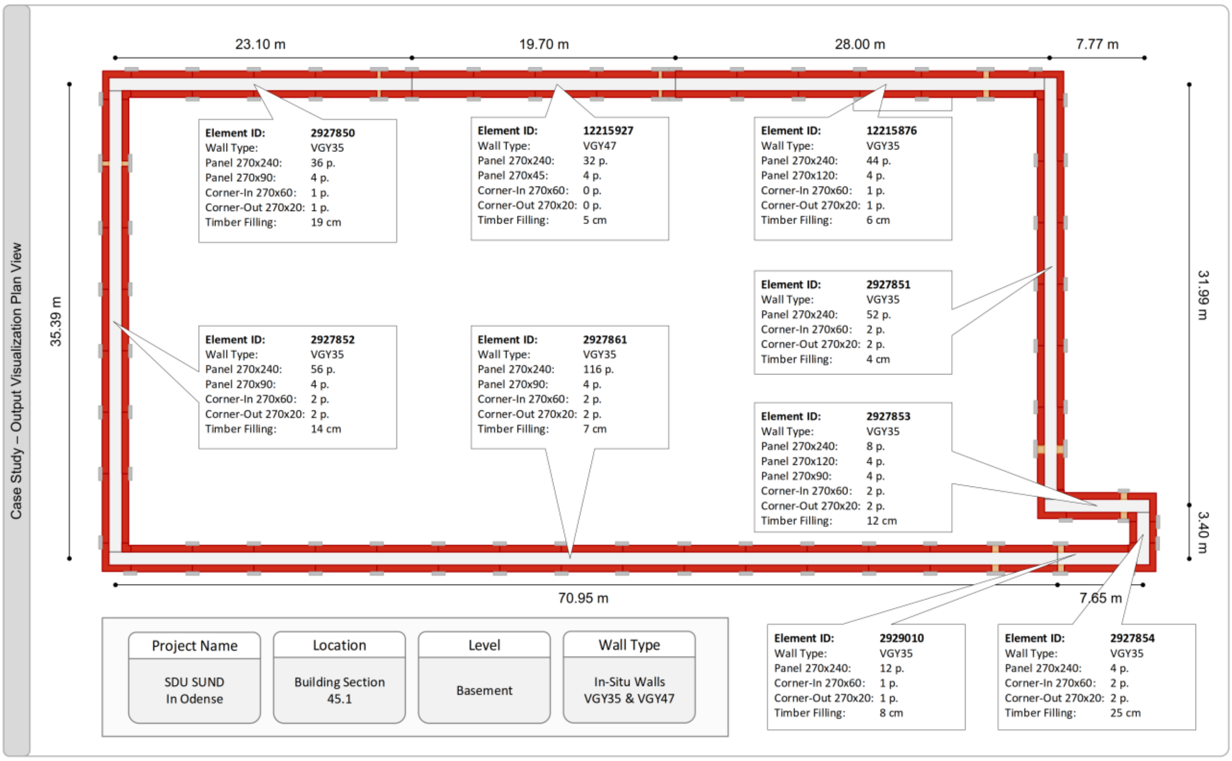


Figure 16: Output visualization plan view

In 3D, the formwork calculation can be imagined as figure 17 illustrates.

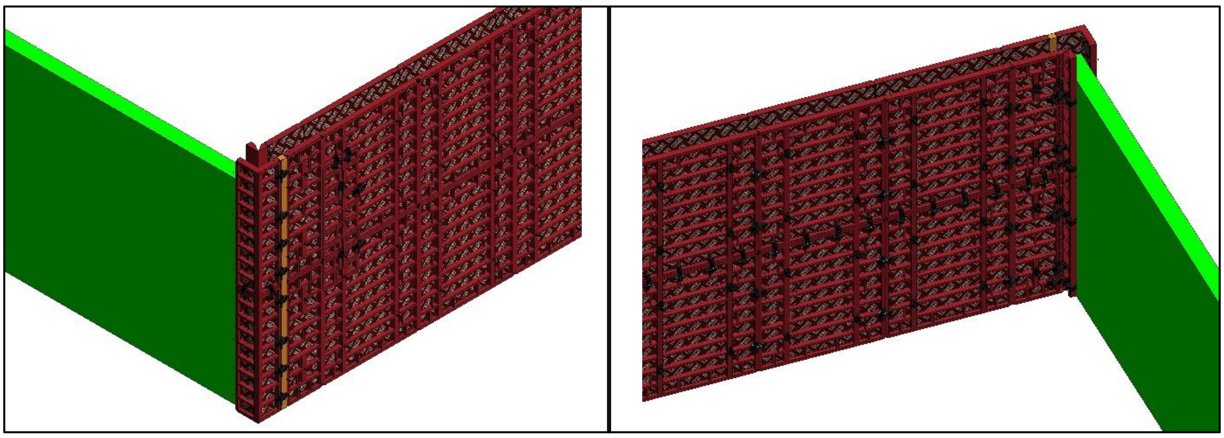


Figure 17: 3D-visualization of the formwork calculation

The program is then writing the data back to the triple store, where a combined data graph is generated out of the just calculated TCI quantities and the three datasets Revit data, LBS data, and TCI data. As in the demo project, this rich dataset can then be used to further visualize the data according to the needs of the relevant stakeholder who will benefit from the solution.

3.4 Data Visualization & Distribution

First, the data has to be transferred to the SQL database, which is used to link the data to the visualization tools. As visualization tools, the Exicute mobile application, as well as the Power BI dashboard, are selected again as in combination, the tools are able to reach all relevant stakeholders, who are in need of the TCI utilization plan, with a tailor-made visualization.
In contrast to the Power BI dashboard, which was further developed and improved, the Exicute App is still only a conceptualization based on a wireframe. The actual effort of developing the app extension to include TCI quantities for each regarded task is not an issue as the platform that is using data from the SQL-database is already existing. The actual issue is that the Exicute platform is still under development and therefore, a consistent app extension could not be developed within the scope of the thesis project. However, the concept of using the data also in a mobile app for the construction workers is still considered as a valuable extension to the developed solution.
In the case study application, more focus was put into the further development of the Power BI dashboard, as this visualization allows to include much more valuable insights into the TCI utilization plan, which will benefit the project or site manager on the construction site. Compared to the already existing dashboard from the demo project, two more pages were developed to give more insight into the developed project data and maximize the added value. The following list provides all the features, the new dashboard incorporates to visualize the TCI utilization plan.

- Project overview and dashboard content
- Exploded building view of building model with different locations allowing to select specific locations
- Time slicer to specify the regarded period or specific date
- Selection tool to specify the specific TCI type to be reviewed in the dashboard
- TCI Allocation graph showing the quantities of the TCI utilization on a time axis
- Comparison between static stock and dynamic stock (Static stock is calculated with the peak amount of TCI demand in the project and dynamic stock is representing the actual TCI demand with a buffer of 10%)

- Graph showing an accumulated cost comparison of TCIs on the construction site, based on the comparison between the static stock and dynamic stock.
- List of all TCIs used in the project
- List of all PCIs which are constructed in the project and supported by the TCIs during the construction activities
- Gantt-Diagram, showing each task and its linked TCI information as well as the progress of the task (if received from the construction site)

The list of important aspects to include in the Power BI visualization resulted in the development of a four-paged dashboard that provides all necessary information regarding TCIs from a management perspective. The following figures subsequentially show the four dashboard pages and explain the content.

Figure 18: Dashboard Page 1 - *Project Overview*

Figure 18 presents the first page of the dashboard, giving a project overview, and providing insights into the following three main pages. The first page is only a static information box, that provides the user with project information on the left side of the page and with the content of the dashboard on the right side. Here, three pages are listed, containing all the above-mentioned features to visualize the full content of the TCI utilization plan. The content of these pages is explained below.

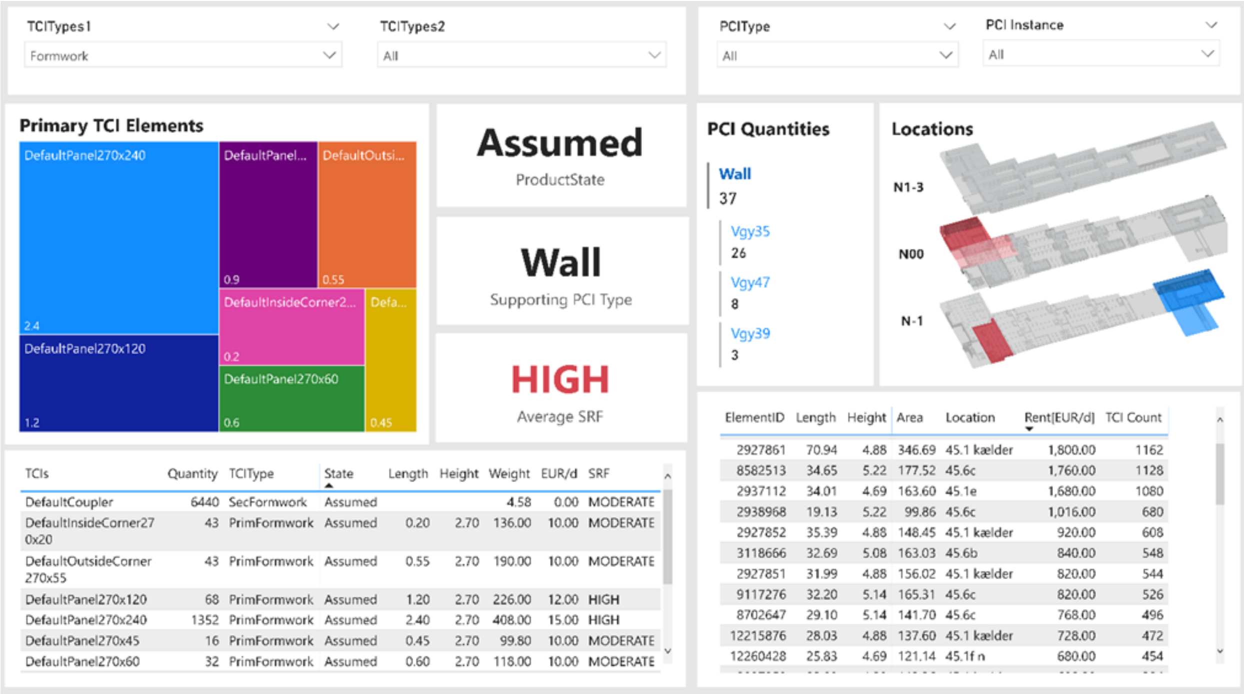


Figure 19: Dashboard Page 2 – *TCIs/PCIs*

Figure 19 shows the second page, the user finds when reviewing the Power BI report. This page is divided into two sections – left and right – and provides information about the temporary construction items (TCIs) as well as permanent construction items (PCIs). The left-hand side of this page reveals the type and some other useful information of all the formwork elements, which are used in the case study project. It also gives some clear notes about the product state of the used TCIs, which PCI they are supporting, and what average Safety Risk Factor the formwork elements have. On the right-hand side of the page, a list of all PCIs which are constructed in the project and supported by the TCIs during the construction activities is provided. In this case, all included PCIs are wall elements. Furthermore, the dashboard provides information about the wall types and their quantities in the project as well as the location, the walls are constructed.

This page shall give the user an initial overview of what TCIs as well as PCIs are used to create the TCI utilization plan, in order to better understand and analyze the next page, which is the main page of the developed dashboard.

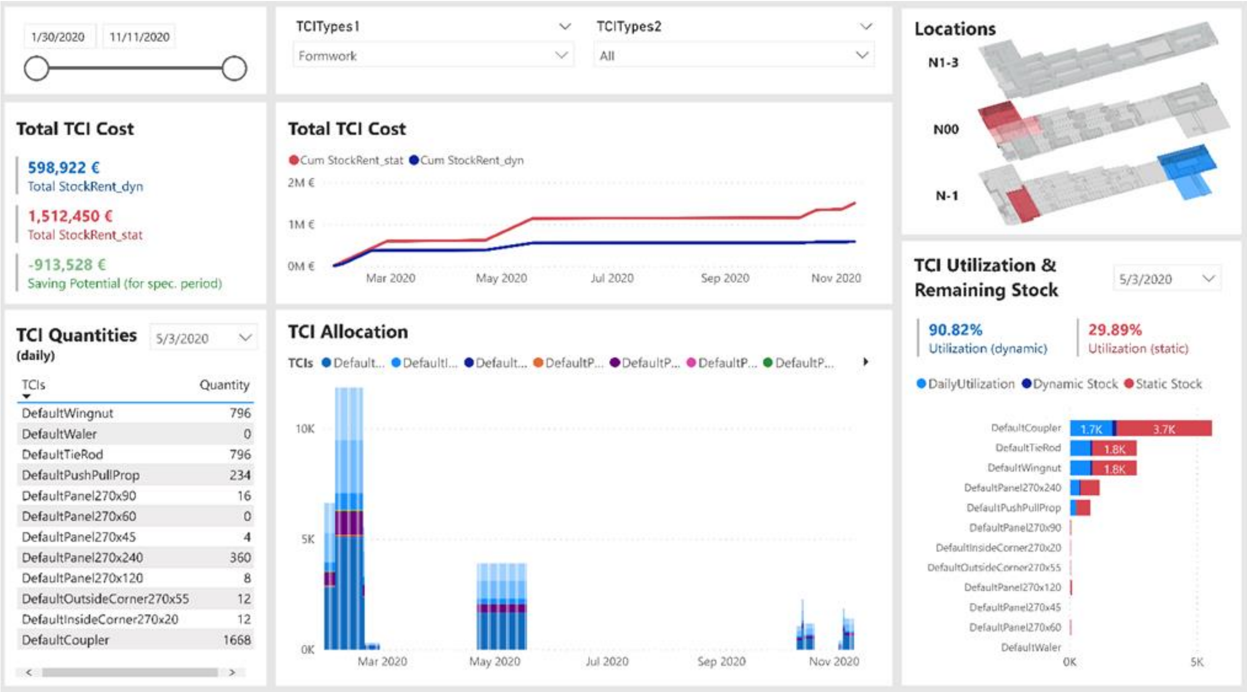


Figure 20: Dashboard Page 3 – *TCI Utilization*

Figure 20 contains the main page of the dashboard, which is exactly the same dashboard that was already developed in the demo project, now only containing the data of the case project. In the central position, the dashboard provides an overview of the TC utilization over time for the whole construction period. It takes into account each day, the tasks that are scheduled that day, the building elements that this task contains as well as the calculated TCI-demand of each building element. Thus, it gives a clear overview of what TCIs are needed on each project day on the construction site. As the solution is also built on a location-based schedule, a location filter on the upper right side of the dashboard allows two either select a location to receive the tasks and TCI quantities for this specific location or the user can select one specific day in order to specify the TCI quantities of this day as well as where the tasks of this day are located on site. On the lower left side, a simple table also reveals the TCI quantities of each day, if the user simply wants to check what items are needed on a specific day.

A further consideration that was already introduced in the demo project is the comparison between a static and dynamic stock of formwork elements. The static stock represents the current practice in the construction industry, where the site manager is ordering formwork elements based on his estimation and with the intention to cover the estimated peak amount of formwork demand plus a contingency buffer.

This approach leads to high quantities of formwork elements on site, costing avoidable rent and also use up valuable space on the construction site. In contrast, the proposed solution enables a dynamic stock with a just-in-time delivery approach, where the communication with the TCI provider is enhanced and formwork elements are ordered and delivered as they are needed on site. This results in a much lower number of formwork elements on site, saving money space and time. And all as a result of the created transparency of the TCI utilization.
This comparison is shown in three different parts of the dashboard. On the lower right side, the dashboard contains the TCI utilization in percent of a specific date, considering both the static (red) and dynamic (dark blue) stock approach. The second part in the upper center of the dashboard, a cost comparison over time shows the different cost development of both stock approaches, clearly identifying lower total costs of renting formwork elements with the dynamic approach. Just left to this, another visual presents the total costs of both approaches for the whole project as well as the potential savings in rent, a construction site could generate by implementing such a dynamic stock for formwork elements.

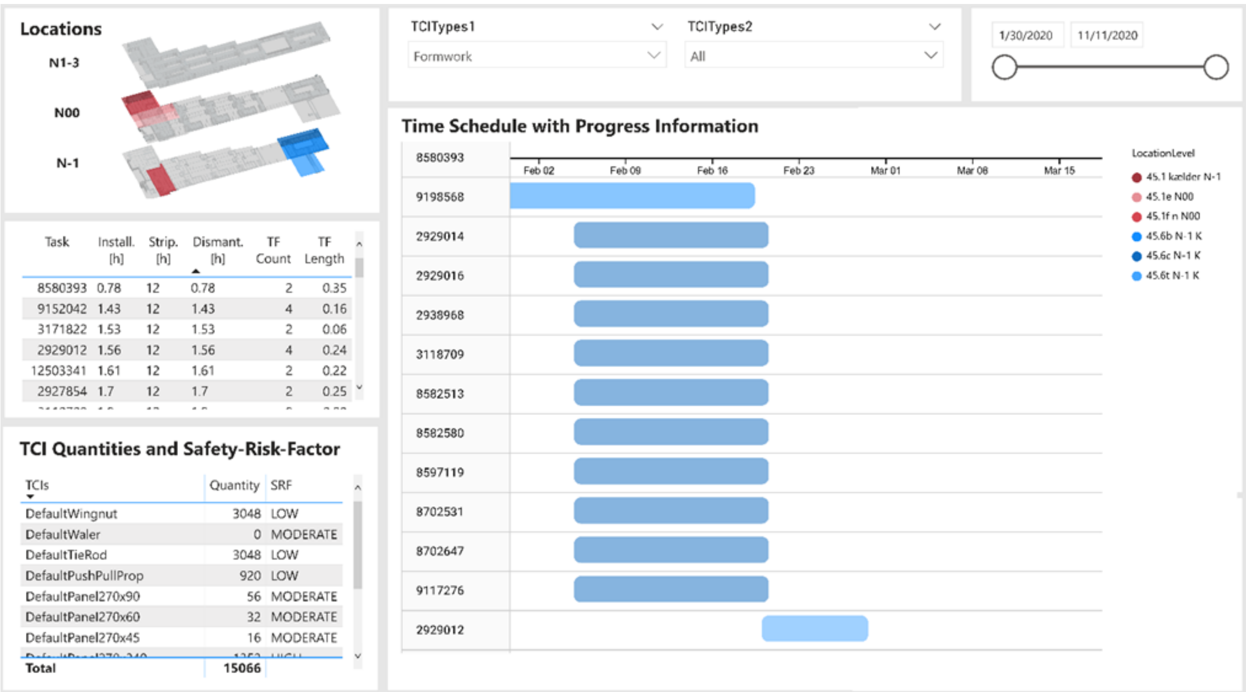


Figure 21: Dashboard Page 4 – *TCI Tasks*

The last page of the developed dashboard in the case study is presented in figure 21, containing similar information as the last page, but visualizes it in another way. This dashboard incorporates a Gantt-diagram to easily select a specific task and then receive the TCI-quantities as well as its location, shown in the dashboard. In addition to this main feature, the TCI quantity table also contains a Safety-Risk-Factor (SRF) which identifies the TCI items that are most risky to install, mostly because of their size and weight. In the center-part of the left side, a small table also shows information of each task, how much time it takes to install the formwork elements, how long the concrete has to rest until the formwork can be dismantled again, and finally how long it takes to dismantle the elements again. This calculation is based on the number of formwork elements and the volume of the wall. For a more precise formwork time consideration, other factors as the concrete type and weather conditions should be considered.

In conclusion, the developed three-folded dashboard with an additional overview page is a powerful tool for visualizing the TCI utilization plan. It clearly targets the management perspective of a contractor who is receiving an easily understandable and visually appealing presentation of location and time-based TCI-information in order to better plan and manage the construction site.

Chapter 4. Findings & Conclusion

In the previous chapter, the case application of the proposed solution was discussed in detail and all aspects and benefits of the further developed prototype were presented. However, the case study also identified some limitations and issues of the proposed prototype solution. By presenting these limitations and issues but also mentioning positive findings from the application on a real construction project, this chapter concludes the case study documentation with an objective reflection of the current prototype status.

Firstly, the case study application was successfully determined with the proof that the developed prototype is applicable to a real case construction project. All steps of the system architecture function well with the applied set of data, especially after automating the process of extracting the LBS data from VICO Office. Thus, the second phase of prototyping a solution that is able to automatically plan formwork elements and generate a location and time-based utilization plan is considered complete. Furthermore, it is assumed that the developed solution can be applied to any big scale construction project with similar specifications as the selected case study project.

Secondly, the current solution also contains some limitations and issues that are identified in the case study or earlier in the process of prototyping. The following list is summarizing these findings which shall be considered in the further development of the solution.

- The current version of the solution only considers formwork as a representative of TCIs. To offer a holistic tool for planning and managing TCIs on a construction project, it should include several different TCI-types.
- The formwork-specific solution should also be extended to include the whole range of items that are used to construct in-situ walls (e.g. pouring platforms, safety rales)
- In addition to the previous point, the solution is currently also not able to consider the sequence of the formwork cycle as already mentioned in chapter 2, in which the simplifications are explained. If a wall has no corner, the ending must be closed with walers and plywood at the end of a wall sequence to prevent the concrete from pouring out. The plywood must have the height of the formwork, the width of the concrete thickness, and an approximate thickness of 2 cm. Then, tree walers shall be placed around the open wall ending to support the plywood and hold the concrete in place.

- The wall thickness of the walls for the case study is manually set to a default of 350 mm. This number is set statically in the demo project, as all walls had a thickness of 400 mm. In the case study, this number was then manually adjusted to 350 mm as 70% of the regarded in-situ walls have a thickness of 350 mm. Currently, the solution is not able to automatically adjust the formwork calculation to the thickness of the regarded wall. This does not influence the formwork calculation for straight walls. The calculation of formwork corner elements, however, is directly affected by the wall thickness as it determines how much length the corner takes up from the total wall length of a wall. As the walls in the case study have variating thicknesses of 350, 390, and 470 mm the calculation of the corner elements is slightly incorrect for some walls which do not have a thickness of 350 mm and contain a corner. Further development should enable the calculation engine of the solution to flexibly adjust the calculation of the corner elements, based on the wall thickness of the modelled wall.

- As already mentioned in chapter 2, the solution assumes that the modelled wall geometries are aligned with the construction process, meaning that a location manager in the location-based schedule will not cut a wall into two, because the wall is modelled too big and does not fit in the location system of the project. This limitation must be addressed in the early stages of a project to ensure that the model meets the requirements to use the proposed solution. An alternative to the early quality assurance of the model is that the contractor adjusts the model as the location-based schedule is created.

- The solution, moreover, does not consider walls with T-junction correctly as they are identified as corners and the solution consequently applies a corner element. The wall 12502993 in section 45.6 of the case project (see figure 22), for example, has two opening walls. Currently, only up to two wall connections per wall (on each ending) are supported by the developed prototype. Thus, further development of the solution has to also consider more complex wall connections.

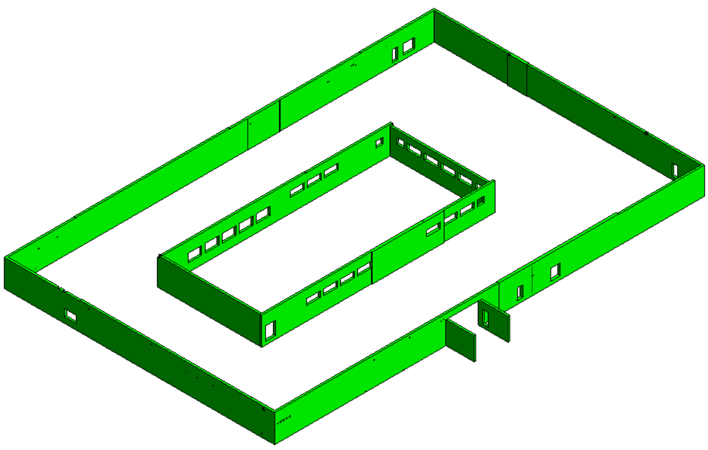


Figure 22: Orthogonal wall opening in the case project'

- The Linked Data remote controlling engine "comunica" would enable a more automated data management in the triple store, where the data of the solution is hosted. It shall be investigated why the engine is currently not working with the triple store of Jena Fuseki and further development shall automate the data management part of the solution.

- The second simplification of the prototype restricts the formwork calculation engine to only consider the formwork application in one dimension as it is assumed that the formwork elements are always higher than the wall elements. A two-dimensional consideration would not only improve the calculation for vertical formwork but also allows to apply the solution for calculating the formwork demand for horizontal concrete structures as concrete slabs. Furthermore, the algorithm to apply the formwork elements consecutively to the regarded wall can be improved. Currently, the algorithm is trying to place the least amount of formwork elements on the given wall geometry and is then applying a timber filling. In some cases, it might be more realistic to apply two smaller formwork panels if this results in a smaller or even no timber filling. Therefore, more advanced algorithms shall be utilized in the solution to improve the formwork calculation e.g. to include more wall connections, have a more efficient calculation, and a two-dimensional consideration.

- The developed solution aims to visualize the developed TCI utilization plan for both the construction workers and the management level and by that utilizes two visualization tools. Currently, only the dashboard is developed as a functioning solution, while the mobile application remains a concept. The next step of prototyping would be to develop the conceptual extension of the Execute mobile application to distribute the TCI utilization plan to the construction site and workers.

With this reflection on the findings of the case study, this documentation provides an objective view on the current status of the prototype solution and helps to classify and understand the level of development, the current prototype is in.

Chapter 5. Bibliography

Yin, R. K. (2014). *Case study research: Design and methods* (5. edition): Sage.