



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

**VIRTUAL**  
manufacturing.



# Faster set-up for DES through integrated standardized template

Bachelor thesis in Mechanical Engineering, Production

Mikael Bergelind  
Lukas Thylin

DEPARTMENT OF INDUSTRIAL AND MATERIAL SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2023

DEPARTMENT OF INDUSTRIAL AND MATERIAL SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2023

[www.chalmers.se](http://www.chalmers.se)

---

**Examiner for the thesis work:**

Mélanie Despeisse

**Supervisors Chalmers:**

Ninan Theradapuzha Mathew

Qi Fang

**Supervisor Virtual Manufacturing AB:**

Oliver Karlsson



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

## **Abstract**

Discrete Event Simulation is one way to analyze and optimize production facilities. The company Virtual Manufacturing AB partly carries out simulation projects for companies with production facilities. In these types of projects, the company talks to the client and analyzes what data that needs to be collected to correctly build the model in a digital environment. The task of collecting the right data and then use that data to manually build the layout of the facility in simulation programs like Siemens Plant Simulation or Visual Components, are a time-consuming part of these projects. This thesis work therefore strived to reduce the amount of time it takes to get from the time when a project is started until real value adding work in the form of improvements to the layout is done. The time that could be saved leaves potential for a larger environmental focus within the projects through more time for the improving phase of the projects. By automating the build of the base model, the employee is relieved of the monotonous task of building the model manually.

In this thesis work there has been an information retrieval phase where a questionnaire survey, informal communication with the employees and a literature study has been carried out. The informal communication was carried out with employees that work with simulation projects in Siemens Plant Simulation and Visual Components. There was also a developing phase where the tasks of developing the template and the library files of these programs was done. An evaluating phase was also carried out where an experiment was done for Siemens Plant Simulation where the time difference of using the template for a real project was evaluated.

The project resulted in a standardized Excel template for gathering the general information needed to build a base model of a production facility to be used in simulation and improvement projects. This Excel template was connected to the libraries that Virtual Manufacturing AB has within the programs Siemens Plant Simulation and Visual Components. These libraries were adapted to connect with the Excel template and use the information within. The file in Siemens Plant Simulation is more capable and can import the data from the template, build the base model and simulate the run of the facility with the correct properties. The file in Visual Components is only able to read the data from the template correctly, but only the test file which is not connected to the template is able to build the model. The experiment that was carried out in Siemens Plant Simulation showed a manual set-up time for the project of 15 hours, and a set-up time of 45 minutes using the template and file. This resulted in a time difference of 14 hours and 15 minutes in the set-up phase of the project in the experiment.

# Contents

- ABSTRACT .....I
- LIST OF ABBREVIATIONS ..... IV
- LIST OF FIGURES ..... V
- LIST OF TABLES..... VI
  
- 1 INTRODUCTION ..... 1**
  - 1.1 General introduction of Discrete Event Simulation..... 1**
  - 1.2 Virtual Manufacturing AB ..... 1**
  - 1.3 Background ..... 1**
  - 1.4 Aim ..... 2**
  - 1.5 Why standardization? ..... 2**
    - 1.5.1 Environmental aspects ..... 2
    - 1.5.2 Social aspects ..... 4
    - 1.5.3 Economical aspects..... 4
    - 1.5.4 Significance of study ..... 5
  - 1.6 Limitations..... 5**
  - 1.7 Theoretical Framework ..... 6**
  
- 2 METHODOLOGY ..... 8**
  - 2.1 Data collection ..... 8**
    - 2.1.1 Literature study ..... 8
    - 2.1.2 Informal communication ..... 8
    - 2.1.3 Questionnaire survey ..... 9
  - 2.2 Library implementation ..... 9**
    - 2.2.1 Implementation into SPS ..... 9
    - 2.2.2 Implementation into VC ..... 10
  - 2.3 Testing and Verification ..... 10**
    - 2.3.1 The structure of the experiment ..... 11
  
- 3 RESULTS ..... 13**
  - 3.1 Data collection ..... 13**
    - 3.1.1 literature study..... 13
    - 3.1.2 Informal communication ..... 13
    - 3.1.3 Questionnaire survey ..... 14
  - 3.2 Library implementation ..... 15**
    - 3.2.1 The template ..... 15**
      - 3.2.1.1 Layout, machine data ..... 15
      - 3.2.1.2 Process time ..... 16
      - 3.2.1.3 Batch..... 16
    - 3.2.2 Siemens Plant Simulation ..... 17**
      - 3.2.2.1 Capabilities of the SPS-file ..... 17
      - 3.2.2.2 Template integration SPS ..... 18
      - 3.2.2.3 Detailed procedure of SPS program..... 22
    - 3.2.3 Visual Components ..... 23**
      - 3.2.3.1 General description ..... 23
      - 3.2.3.2 Excel Import block ..... 24

3.2.3.3 Template block .....	29
<b>3.3 Testing and verification</b> .....	<b>32</b>
3.3.1 Test SPS .....	32
3.3.2 Test VC.....	33
<b>4 DISCUSSION</b> .....	<b>34</b>
<b>4.1 The results</b> .....	<b>34</b>
4.1.1 Data collection.....	34
4.1.2 Library implementation .....	34
4.1.3 Testing & Verification .....	35
<b>4.2 Further development</b> .....	<b>35</b>
<b>4.3 Project execution, plan &amp; preparation</b> .....	<b>37</b>
<b>5 CONCLUSIONS</b> .....	<b>39</b>
<b>6 REFERENCES</b> .....	<b>40</b>
<b>APPENDICES</b> .....	<b>42</b>
<b>Appendix A</b> .....	<b>42</b>
<b>Appendix B</b> .....	<b>43</b>
Appendix B1 .....	43
Appendix B2 .....	45
Appendix B3 .....	45

## **List of abbreviations**

- AGV** – Automated Guided Vehicle
- AVI** – Availability
- CAD** – Computer Aided Design
- DES** – Discrete Event Simulation
- DPD** – Digital Product Development
- HMI** – Human Machine Interfaces
- MTTR** – Mean Time To Repair
- MTBF** – Mean Time Before Failure
- PLM** – Product Life Management
- PLC** – Programmable Logic Controller
- SPS** – Siemens Plant Simulation
- VC** – Visual Components
- VM** – Virtual Manufacturing AB
- WIP** – Work In Progress

**List of figures**

Figure 1: Illustration of project methodology..... 8

Figure 2: Illustration of library implementation..... 9

Figure 3: Methodology for measuring time difference between using the template and not using the template..... 11

Figure 4: Structure of the experiment procedures. .... 11

Figure 5: Example of data in sheet one of the template. .... 18

Figure 6: Example of printed objects in SPS. .... 18

Figure 7: Example of how data is automatically fed into Station1. .... 19

Figure 8: Example of how data is automatically fed into Buffer1. .... 19

Figure 9: Aspects of energy consumption in different stages within the template..... 19

Figure 10: Aspects of costs within the template..... 19

Figure 11: Example of automatically imported power input values into Station1..... 20

Figure 12: Example of automatically imported cost values into Station1. .... 20

Figure 13: Example of how Station1 can have different process times connected to different products..... 20

Figure 14: What the data for process times look like in the table for process times within SPS. .... 21

Figure 15: Illustration of what the data for batches looks like in the template to the left and then what it looks like in SPS to the right..... 21

Figure 16: Example of an automatically built layout in SPS..... 22

Figure 17: Picture of all the methods used in SPS to import data, build the layout, and run the simulation..... 22

Figure 18: Properties within Excel import block in VC..... 25

Figure 19: Picture of how to choose the right CSV file in VC. .... 25

Figure 20: Example of where the user needs to place the pathway within the code in VC. ... 26

Figure 21: Behaviors within the Excel import block in VC..... 26

Figure 22: Example of how the data within the notes in VC are split. .... 27

Figure 23: What the window with access to notes and buttons look like in the Excel import block in VC. .... 28

Figure 24: Example of what a built layout in VC can look like using only the buttons in the program. .... 29

Figure 25: Example of what the current four types of objects within VC looks like. .... 29

Figure 26: What the switch and the geometries for the different objects in VC looks like. .... 30

Figure 27: The row of code for declaring what value to assign to each object type in VC..... 30

Figure 28: Example of some of the properties within the template block..... 31

Figure 29: Behaviors of the template block in VC. .... 32

Figure 30: This is the first sheet of the Excel template. .... 44

Figure 31: This is the second sheet of the Excel template..... 45

Figure 32: This is the third sheet of the Excel template. .... 45

**List of tables**

Table 1: Theoretical framework. .... 7  
Table 2: Information within sheet one about general information in Excel template. .... 16  
Table 3: Information within sheet one about failures plus two aspects in Excel template. ... 16  
Table 4: Information within sheet one about power usage in Excel template. .... 16  
Table 5: Information within sheet one about costs in Excel template. .... 16  
Table 6: Information within sheet two of the Excel template..... 16  
Table 7: Information within sheet three of the Excel template. .... 17

# 1 Introduction

## 1.1 General introduction of Discrete Event Simulation

For optimizations made to production systems and factories, there are often simulations made beforehand. The benefits of simulations are that the production can stay running through the testing of different optimizations, and since no machines must be purchased there is also a great benefit when it comes to cost. These simulations are done digitally through different simulation programs. One type of such simulation is Discrete Event Simulation (DES). Every time an optimization is to be made for a new client the digital models also must be created from scratch. This is a process that takes a lot of time, both to model the facility, but also to go through all the raw data and filter out the necessities. The demand for DES is going up since there is a need for companies to always improve. The current ambition to reach industry 4.0 is also a driving factor for why optimizations through DES are sought after.

## 1.2 Virtual Manufacturing AB

This bachelor thesis was conducted at a company called Virtual Manufacturing (VM). Virtual Manufacturing was founded in 2006 and they mainly work within six different areas. These areas are as follows:

- Industrial management
- Assembly and production flow
- Robotics and automation
- Lean manufacturing products
- Virtual Manufacturing labs
- Manufacturing software and services

VM provides services to clients within all different types of industries and some of their values are safety and ergonomics, reduce WIP (work in progress) and increase quality. VM helps their clients to optimize their industries partly based on the values presented. The headquarters of VM are in Gothenburg. However, they have offices in other parts of Sweden and in other countries around the world as well. They also have employees in Scotland and Bangkok for example. In 2021 they had 49 employees according to (*Virtual Manufacturing Sweden AB | Info & Löner | Bolagsfakta*, n.d.).

## 1.3 Background

Virtual Manufacturing partly works with assembly & production flow. In this area they often help customers with their factories and help them optimize their manufacturing through simulations in programs like Visual Components (VC) and Siemens Plant Simulation (SPS) for example. To be able to help the customers they need quite a lot of information about the current numbers in production. They might need information about how many machines they have, how they are placed in the factory, how many products they produce in a certain timeframe, how much work in progress (WIP) they have, for example.

Today there are no standardized ways for the consultants working to obtain this information and organize it. Therefore, there is a need for a standardized template to use when gathering information. Another problematic area relates to the fact that it's quite time-consuming to go through all the data collected and then build a virtual twin (VT) in VC or SPS. Because of this there is also a need for an integration between the template and the tools to minimize the time duration from the moment that Virtual Manufacturing is given the task to carry out this service, until the time where the work done is value adding for the customer.

#### **1.4 Aim**

The aim of this project is to minimize the time it takes from when the company gets the order to optimize a factory, until the actual value adding work begins. This is crucial since most of the time is not value adding to the customer because a lot of time is spent just analyzing and building the facility in a program. On average 31% of the entire time spent on a project is used up by the input data management process (Skoogh & Johansson, 2007), which indicates that there is a need for optimizing the input data management. There are a lot of benefits to this, and it would benefit people both carrying out the projects and the customers. Employees working with the projects will go through less time consuming and monotonous tasks. The customers will receive their optimized facility faster, which is especially important as time goes on since the competitive environment grows larger by the day.

#### **The objectives of the project are as follows:**

- Minimize the time between from when VM receives an order from a company until real value adding work is being done. In this case value-adding work is the work done to improve upon the current state of the facility.
- Reduce the cost for DES projects through time reductions. The hope is that the reduced cost will encourage more companies to improve through optimizations.
- Make the work environment better for the employees at VM through a decrease in monotonous work tasks.
- Guide the employees to think more about environmental aspects when optimizing industries through a template that also focuses on such environmental aspects.

#### **1.5 Why standardization?**

The following chapter describes why a standardized template of a DES project would be beneficial regarding the following aspects.

##### ***1.5.1 Environmental aspects***

From an environmental standpoint it would be beneficial as less time is spent modeling the facility. Either more time is spent on improvements and more environmentally beneficial solutions and evaluations could be done. For example, there could be more time spent analyzing the environmental impact of different lengths of transport, different oils used or maybe heating/lighting of the facility. In short, other areas which could be redone to minimize emissions of carbon dioxide and usage of resources could be done in the same

timeframe of a project today. Another possibility for a project that has a shorter total time is that they cost less, and less money is spent by the customer for the project. This money could be spent on environmental solutions and evaluations instead. Also, shorter projects will lead to a couple of different possibilities for the company (in this case Virtual Manufacturing). They can carry out more projects in a year, and they can also hire more people which will also be a contributing ethical factor. This is because if they employ more people, they also contribute to more job opportunities for more people.

When optimizing a facility used in some kind of industry there is also a real possibility to affect the usage of different resources. 47,2% of the consumption of water was used up by industries in 2017 according to EEA-33 data. In 2017 it was the energy supply industries that used up 46,5% of the water usage within these total 47,2%. Of these 47,2% there were 13,1% of the water consumption that was used up by the manufacturing industry (*Industry — European Environment Agency, n.d.*). The way that shorter set-up times for DES projects can help here is through more time spent in the analyzing phases. If there is a possibility to shorten the total time of DES projects, there is also a possibility for a more thorough analysis of different machines and variants in the same timeframe as the old projects. Here it could be investigated if equipment using less resources could be used. This in turn has the potential to minimize the amount of water used for production.

One of the aims of the project is to develop a way of thinking for the employees at Virtual Manufacturing when it comes to the environmental aspects. This can be done by including a lot of environmental factors in the template, which will hopefully make the employees notice this. The template could focus on parameters and data saying something about the air pollution, water pollution, soil pollution as well as the health of the employees working within the facilities which are some of the ways that factories can affect the environment (*How Can Factories Affect The Environment? | Field, n.d.*). This is information that wouldn't have been gathered because it might have been forgotten. By doing this, the environmental aspects can be a greater part of the project. By adding focus on environmental aspects, the mindset of the employees hopefully becomes more focused on environmental questions overall which might influence the customers of VM to make changes for a better environment.

The production of energy did contribute to about 73,2% of the total greenhouse gas emissions on earth in 2020 (Ritchie et al., 2017). Therefore, there is a need to choose more environmentally friendly energy sources. If the template can influence the employees' thought process and focus more on environmental factors like energy and where it is sourced, then maybe some cost calculations and possible energy sources could be included in the delivery to the customer. This could influence more companies and factories to change to better energy sources, and by decreased demand the energy sources which are worse from an environmental standpoint could be eliminated or at least used less. This would also force a lot of companies supplying energy to change to more environmentally better solutions to generate energy. Also, if more companies source their energy from clean energy sources, these companies will have more customers which would mean an increased income and an expansion of those companies. This is an incentive for more people to focus on environmentally clean sources of energy.

### **1.5.2 Social aspects**

Time saved when not having to create the base model from scratch could be spent analyzing variations of the new and improved layout. For the same time duration as the projects today, the employee could make alternative versions of the layout with different ways to solve the different problems. This could be that the employee creates one version of the facility that uses only automated guided vehicles (AGV) for transport, and then one version that uses only human operators or conveyors for material handling for example. This would not only provide more options for the customer to choose from, but also make it easier to consider the different possibilities to solve a problem like material handling.

If the creation of a base model is autonomous, then the amount of repetitive work for the employees is reduced dramatically. For example, if a model has 30 machines, the employees must go into every machine and change the specific properties for each one. When there are many projects going on at once, this monotonous and time-consuming work can create a lot of stress for the employees. This can be very bad long term since stress and especially mental stress over a longer period really can affect how the brain functions. Phenomenon's like having problems with cognition, attention and the memory can be a result of stress. The capabilities to take in and process information can also be affected. Another change in behavior could be anxiety or other mood changes that come because of this stress. The long-term mental stress can also give a higher risk of inflammations in the brain which in turn can affect the heart as well (*Protect Your Brain from Stress - Harvard Health, 2021*). One goal is then to minimize this strain by using the template and special SPS and VC file to easily create the base model and save the employee from some parts of this stress. It is easy to get distracted from the real urgent tasks by having too much to do in a short amount of time. It is also a distraction to be interrupted with questions when the worktime should be spent on more urgent tasks and more time pressing tasks. The time stress is one of the profound sources of real pressure at work (*Eisenhower's Urgent/Important Principle - Using Time Effectively, Not Just Efficiently, n.d.*). Therefore, the template can help to speed up the less important processes so that more time is freed up to be spent on more urgent matters. The standardized template can also reduce the time spent looking through and analyzing the data. If the Excel template is sent out to the customer to be filled in, the only thing the employee must do is to make sure it looks correct. The data written by the customer in the excel template will be directly compatible with SPS and VC and the employee just must make sure that the placement of the file is correct.

### **1.5.3 Economical aspects**

The aim is also to reduce the costs that come with the planning of new layouts and make it cheaper for customers/companies to optimize their facilities thus encouraging more companies to do so. If the program can autonomously create a base version of a facility and time can be saved in the project, there will also be some financial benefits. The automation of tasks that result in higher productivity can result in faster economic growth, increased demand for more labor in the future, more consumption from customers and the creation of more jobs in the future. More jobs can be created from these optimizations since the money saved can be spent on new technology which will enable the possibilities for more advanced and new products (Skidelsky, n.d.). The time saved can be financially beneficial to the customer in two ways. Virtual Manufacturing in this case can take less money for the same

work because less time is spent on the project. There is also a possibility of getting more detailed layouts or maybe different versions of the same layout in the same timeframe that the projects would take before. The second one can be financially beneficial because of new ideas and solutions to consider which would benefit the customer in the long term with longer timespans between upgrades to the facilities. Or perhaps even an adventive edge.

#### **1.5.4 Significance of study**

Longer projects are more expensive. If the project takes a shorter time, less money is spent by the customer for the project. This money could be spent on environmental solutions or other solutions and evaluations instead. Also, shorter projects will lead to a couple of different possibilities for the company (in this case Virtual Manufacturing). They can carry out more projects in a year, and this might lead to the possibility to hire more people and do many more projects in a year. This will also be a contributing social factor, to get more people into a job by providing more jobs. The time saved could also be a direct cost saving factor for VM at which they can either save money, lower the cost for helping manufacturing companies or focus more on other areas as mentioned above. All aims of the project will therefore be taken into consideration by looking at environmental, social and economic aspects.

#### **1.6 Limitations**

- One limitation for this project is that the project will only focus on optimizations to production facilities like factories with machines, workers and AGV's for example. This limits the project to only focus on facilities which produce products.
- This project will only result in one file for SPS and one for VC that will be compatible with and able to import data from the Excel template.
- The placement of the objects when building the base model autonomously using the compatible SPS or VC file will be based on the coordinates placed in the Excel template.
- Through the initial research in the beginning of this project it was found that there are some programs/systems in place to help with the information gathering and filtering. These will not be covered/used in this project. These are tools like the GDM-tool (Generic Data Management tool) presented in (Skoogh et al., 2012).
- This project will not cover all the nine phases of the data input process in a DES project (Skoogh, Johansson. 2007). For instance, this project will not cover the activity of choosing where to gather data from.

## 1.7 Theoretical Framework

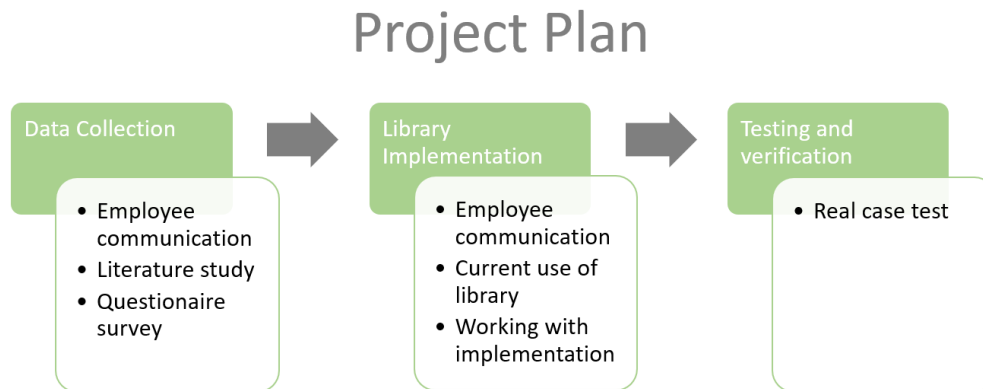
Word to describe	Description
PLC	A Programmable Logic Controller is a programmable system that is used to control other objects. PLC's are often used in industrial automation and mostly to control machines, engines, or functions. A PLC could be used to give directions to robots or other automation equipment. The computer languages used to program PLC's are standardized according to IEC-61131-3. To easily control a PLC and easily use the program within the PLC, it is often connected to an HMI ( <i>Vad Är PLC?   Elektroautomatik, n.d.</i> ).
Definition of Template	The definition of a template that is used in this report is the following. In this case a template is seen as a digital file with a constructed pattern to be used as a guide for making something (Hanna, n.d.). In this project the template is a file with a premade pattern to be filled in according to the pattern that guides the user as to what information is sought after.
VC	VC stands for Visual Components. This is a work tool for digital visualizations of facilities within production. The program is used for example when working with layout planning or production simulation. The company was founded in Helsinki, Finland in the year 1999. VC has now been owned by the robot company KUKA since December of 2017 ( <i>Visual Components - Wikipedia, n.d.</i> ).
SPS	SPS stands for Siemens Plant Simulation and is a work tool made by Siemens Digital Industries Software. This tool can be used for planning and visualizing a facility aimed at production. The software allows for modeling, analyzing, simulating, optimizing, and visualizing ( <i>Plant Simulation Software   Siemens Software, n.d.</i> ).
CAD	CAD is computer aided design ( <i>CAD Software   2D And 3D Computer-Aided Design   Autodesk, n.d.</i> ) . This is when computers are used to modify, optimize, or analyze the design of an object through a computer model of that same object. There are many different CAD programs available, and they are often used by both designers and engineers. CAD is one big part within both Digital Product Development (DPD) and Product Life Management (PLM).
DWG layout	DWG is a standardized file format for CAD models and is supported in most CAD programs. This file format is called DWG which stands for drawing ( <i>DWG – Wikipedia, n.d.</i> ). This DWG file is often in 2D but can also be done in 3D. This file is often used as guides for engineers or designers who work in CAD in DPD projects. The file either shows the operator details about the model or it can also be created after making the model on the computer to show the customer how the object will look from different angles. These DWG files can also act as drawing.
AGV	An AGV is an autonomous material handling system that does not need a human to operate. There are two main types of AGV's. The

	Line follow AGV systems which moves according to a path that is predefined by the operator and the natural navigation AGV system which do not follow any path. They are more advanced than the line follows AGV's and they use more advanced sensors and programs to be able to scan and map their surroundings ( <i>What Is an AGV System? Why Would You Need One? / MasterMover, n.d.</i> ).
Library	In this report the library is a digital library which in turn is a collection of digital objects. These can be objects like magazines or video and audio recordings for example, that are accessible electronically. These digital libraries allow for the possibility for the user to reach a large quantity of resources (Terrell, n.d.). In this project the word library will now refer to the digital libraries of Virtual Manufacturing within the different programs.
URI	URI stands for uniform resource identifier (What Is a URI? Definition and Explanation of the Identifier - IONOS, 2020). A uniform resource identifier enables computers to identify different things depending on application. In some cases, URI's can be used to identify different email addresses or websites. In the case of this report the URI's are used to identify the pathways for the CSV files within Visual Components.
Note	A note is a text window within the program Visual Components where the data is located. In this report the word note will be used in the context of importing or using data in Visual Components.

*Table 1: Theoretical framework.*

## 2 Methodology

The methodology chapter describes the systematic approach and methods used in the project. The three phases of the project are presented in *figure 1*.



*Figure 1: Illustration of project methodology.*

### 2.1 Data collection

This part of the methodology will handle the gathering of information. The data collection stage contains a literature study, informal communication, and a questionnaire survey. The questionnaire survey will be conducted initially to get an understanding of the issues and the thoughts of employees working with in the area. This will be followed by a literature study to gather more information. All of this will also be backed up by informal communications for understanding how the work is usually done and what the different programs and libraries looks like and are used.

#### 2.1.1 Literature study

A literature study will be conducted to gain a better understanding of what to include in the standardized Excel template. Information about how to categorize data for the discrete event simulation projects will also be researched. The main sources of information for the literature study will be scientific papers and other articles from mainly Chalmers library and google. Keywords for the literature study will be partly DES, Siemens Plant Simulation, Visual Components, and standardized template.

#### 2.1.2 Informal communication

A lot of information can be retrieved through conversations with the employees at the company. Especially when talking to the supervisor from VM, but also when meeting other staff in the lunchroom and asking about their work. This will be able to give information such as what different types of tasks that is being done at the company, who does what, how the work is divided amongst the employees, which programs that are mainly used, and to be able to see how the work is done today which will emphasize the need of the improvements that the projects investigate.

### 2.1.3 Questionnaire survey

A survey will be conducted in the initial stage of the project. The purpose of the survey is to gather information from the employee's perspective and what they think is important information to include in the template. Benefits of complementing the literature study with a survey is that more specific questions can be answered, and the employees have time to think about the questions. The survey can even be sent out to different companies to gather a broader knowledge of the topic if necessary. The content of the survey is what information the employees at VM would like to see included in the template (see more information in appendix A).

## 2.2 Library implementation

To standardize the process even more the template would also need to be connected to the libraries of VM. If the template is connected to the libraries used by VM it would also mean that the usage of the template would spread quicker amongst the employees. The procedure for the implementation looks similar for both VC and SPS.

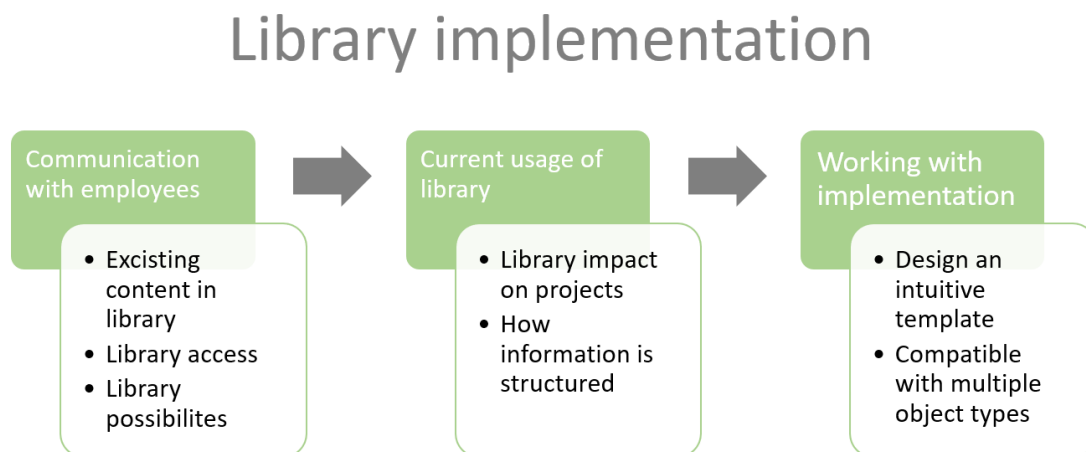


Figure 2: Illustration of library implementation.

The first step is communication with employees where more knowledge about the libraries is collected. Information about what content that exists in the libraries already, which employees that have access to the library and what possibilities there are. The second stage of the implementation process is also about gathering information, but more through observation in combination with questions rather than just verbal communication. How the libraries are really used today and how the information within them is structured and used is to be seen in this stage. The last stage of the implementation is working with the implementation where the design of the template and getting the template to connect to the programs is worked on. The implementation of the templates into the programs needs to be considered when designing the template.

### 2.2.1 Implementation into SPS

There are some things that need to be done to successfully connect the template to the library used in VM for SPS. The first task is to gather more information about the library

through communication with the employees. The information that is then sought after is what the libraries look like, how to connect to or get access to them and what possibilities there are within them. As a second part of this there would be a need to see the possible use of the libraries in a real project. This is to see how they work with the library and to see how it functions. This will lead to a deeper understanding of how the implementation should be done to fit as well as possible into the way it is used. For example, giving the different objects certain names or placements that makes it easier for the employees to find them and quickly use them. The way that the information is structured within these libraries is also important so that it follows a logical pattern which allows for easy access. After the information about what the library looks like and how it is used the group would need access to the library so that the implementation of the template can be made.

Finally, the group needs to work with the implementation and making sure that it is organized as well as getting the Excel template and the SPS-file to be compatible. This work will also take some time since the information from the template needs to work with the different objects within SPS and the SPS file with the library needs to be able to give different values to the different objects depending on the type of object.

### **2.2.2 Implementation into VC**

The procedure for implementing the template into the library for VC is very similar to the one used to implement the template into the SPS library. Firstly, the employees at VM need to show what the library looks like, how it works and how it is used today. Then the group needs access to the library, then there needs to be some work done to implement the template and make sure everything works.

### **2.3 Testing and Verification**

After the implementations have been done there has to be a verification stage. This is to check if there are any differences in the time it takes to model a layout when using both the current way of working and the new one with the template. The test will be based on a real case. However, the experiment will not be a large-scale project because of demands to not occupy employees from VM that much. If the experiment were as large as a normal project that VM usually covers, then it would take too much time from their normal work, and they still have other tasks to complete as well. Therefore, the size of the project will be one of the smaller ones that VM usually handles.

To measure the time difference between the regular method compared to the template, the procedure presented in *figure 3* will be used.

## Testing and verification



Figure 3: Methodology for measuring time difference between using the template and not using the template.

The time it takes to complete the project will be clocked with a regular timer and there will be a time difference between the two methods. The first method is the method used today at VM, creating a layout manually in the software itself. The second method is using the standardized template to build the base model, and then making some small manual adjustments to the layout.

### 2.3.1 The structure of the experiment

The experiment procedure is described in *figure 4*.

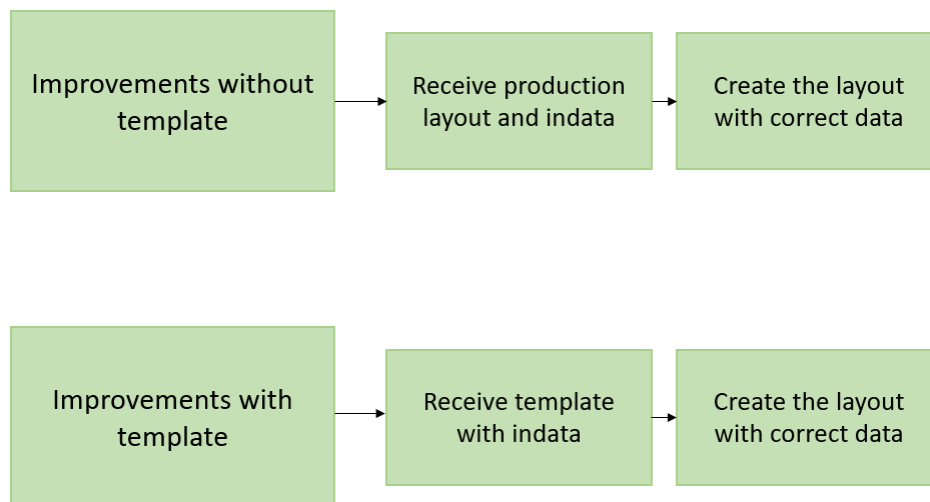


Figure 4: Structure of the experiment procedures.

1. The consultant at VM will have an ongoing project to work with, in this case a large manufacturing company. When all the in data and layout has been received, the consultant will start to manually build the manufacturing layout with the correct data and finally simulate the model. This is the improvements without the template.
2. The exact same model will be created, but this time with the standardized template. The consultant will now receive the template, plot the layout, and then make some small final changes. Here the employee will need to fill the template manually with

the information and then place the correct pathway for this template file within the program. After that the program will be able to automatically generate a base layout. This is the improvements with the template.

The time will be documented for each way of carrying out the project and these documented times will then be compared to get the result for the experiment. The time that will be documented is from when the employee starts with the project until a base model is built, and a simulation is possible.

## **3 Results**

In this chapter the results for the different parts of the project will be presented. The results chapter is divided into three different parts. The first part of this chapter is called data collection and in this part the results from the phase of the project where the information was gathered is presented. The second part is called library implementation and handles the results of the integration of the template into SPS and VC. This part also handles the capabilities of the template compatible files within the programs. The results are further broken down and divided into different parts for the results of each program integration. The third part is called testing and verification and describes the results of the test.

### **3.1 Data collection**

In this chapter the results of the data collection will be discussed.

#### **3.1.1 literature study**

Regarding the research done, the initial source of information about standardizations in DES projects was provided to the group by a supervisor from Chalmers. The group received a folder containing reports and documents that touch upon this subject. Furthermore, research in databases through Chalmers library was done. This research led to the discovery of the following articles:

- Skoogh, A., & Johansson, B. (2007).
- Skoogh, A., Johansson, B., & Stahre, J. (2012).
- Skoogh, A., Michaloski, J., & Bengtsson, N. (2010).

According to the articles, there are a few different ways to analyze and place meaning upon raw data. The raw data must be contextualized to have a meaning. Contextualization is to reason why the data was gathered and try to understand these reasons. By understanding this a better and clearer view of what information to filter out and gather keeping the purpose in mind is achieved. According to the same source the other parts are categorization, correction, calculation, and condensation. This would be that the data must be gone through to be filtered. There is data that is not needed, or the data might be disorganized and hard to look through. The contextualization is to understand why the data was collected, categorization is the key areas within the data, correction is to remove the invalid and wrong parts of the data which includes unnecessary data, calculation is the mathematical and statistical properties of the data and lastly condensation is to summarize the data in one place.

#### **3.1.2 Informal communication**

The group did look through these documents but got more knowledge about the information needed from various conversations with employees in Virtual Manufacturing. Through observations of how the employees at VM carried out their tasks, and through conversations in the break room for example a lot of information was compiled. Such information could be how they normally operate, how the information is normally organized when given to the employees and how much time is spent on a project.

Normally the employee gets introduced to the project through a meeting with the client either in person or through digital means. This meeting is arranged for the employee to get an understanding of what the client wants to do and what the current situation is like. In this meeting the client also gets understanding of possible durations for such projects and cost estimations. Also, if the objective is reasonable at all. Often the client is already prepared with data about process times, waiting times, downtimes, and number of machines for example. However, this information is not consistently organized when given to the employees at VM. This means that for each project the employee must either discuss the data with the client or try to analyze the thought process behind the categorization. This part of the process often takes a bit of time, and when the data is unorganized it takes a longer time before the employee notices if some critical data is missing. After the data is understood and complete, the employee can then start to build the digital model of the layout in the right program. The decision of using either Visual Components or Siemens Plant Simulation is based on personal preference of the employee and the preferences of the customer. In some cases, there might be a certain quality or capability within one of the programs that determines which program that is used. Then the employee will work on the project and provide progress reports to the customer along the duration of the project. In the end of the project there will be a larger presentation of the results. The employees might have multiple projects going on at once and they also help each other to solve tough problems.

### **3.1.3 Questionnaire survey**

The results from the questionnaire survey show that a lot of input data is wanted for the template. However, there were only two people who filled out the form which reduced how representative the survey was. Notice that both employees have experience in SPS respectively VC. The questions and answers of the form can be found in Appendix A.

The form expressed that both attendances believe that a standardized template would be useful to have when entering a new project and should include data of times, failures, capacities, number of machines etc. The input data comes from the customer and must sometimes be adjusted to fit in a specific distribution. Alongside the data provided by the customer, there is also often information given about the layout of the facility. This information comes in the form of a DWG layout. A DWG is a drawing of the layout and is described more in the list of abbreviations. This information clearly shows that the template not only has to handle the process time, but also what kind of process time it is. VM manufacturing usually works with constant process time or Weibull and Erlang distribution which would therefore be crucial to include in the template. Even more distributions would be useful to have as an option according to the survey. Weibull and Erlang are two different mathematical distributions. In SPS there is a possibility to change the distribution used for the process times.

The form also expresses some useful input data for AGV's. Some of the data is the speed of the AGV, number of AGV's, routing logic, how missions are assigned and what happens if they get close to each other. However, including AGV's in the template will make it more complicated and only some basic AGV integration will be possible. This is because AGV's are

seen as a suggestion and a wish to include. Therefore, it will not be prioritized and will only be handled if there is time for it. Also, more research needs to be done in order to understand how to use AGV's and what information that is needed.

### **3.2 Library implementation**

This section will cover how successful the integration between the Excel template and both Visual Components as well as Siemens Plant Simulation were.

#### **3.2.1 The template**

The project resulted in a standardized template. This template makes it possible to standardize the data collection process for DES projects. This Excel template now consists of three Excel sheets. These sheets contain different type of information and are split to have an easier categorization between the data. Having the data divided into different sheets also makes the template easier to read through having more smaller sheets than one large sheet of data.

The sheets are labeled as follows:

- Layout, machine data
- Process time
- Batch

##### **3.2.1.1 Layout, machine data**

This is the first sheet of the Excel template. This sheet contains data about the type of object, name of it and some properties. Some aspects of the power consumption and costs for the objects are also involved here. There is much information within the first template sheet. This sheet contains all the general information about the layout. Object types, object names, power consumption and process times are some of the information included for example. The information in this sheet is split into smaller sections within this report. This was done because the sheet is too large to have in one table. The information will be split into smaller sections categorized according to general information, failures plus other aspects, power usage and cost. When used all this data is in one single Excel sheet. The tables visualizing the information in the Excel sheet are filled with example data. *Table 2* contains the general information, *table 3* contains the information about failures plus two other aspects, *table 4* contains information about power usage and *table 5* contains information about cost. In *table 2* the columns for X, Y and Z are the coordinates of the placement of each object. In *table 3* the column called "precursor" is data about the following object. In *table 3* the example data describes that the following object after station 1 is station 2 for example. The column handling capacity in *table 3* is regarding the capacities of mainly buffers. See appendix B1 for a picture of the real design of the first sheet in the template.

Object Type	Identity	Frame	X	Y	Z
Object 1	Station1	1	0	0	100
Object 2	Station2	1	50	0	200
Object 3	Station3	1	100	0	300
Object 4	Station4	1	150	0	400
Object 5	Station5	1	200	0	500

Table 2: Information within sheet one about general information in Excel template.

MTBF	MTTR	AVI	Capacity	Precursor
5	2	9	4	Station 2
63	5	8	5	Station 3
2	7	5	6	Station 4
5	9	6	3	Station 5
8	5	3	2	

Table 3: Information within sheet one about failures plus two aspects in Excel template.

Power consumption working	Power consumption setting up	Power consumption operational	Power consumption failed	Power consumption standby
35	67	42	26	9
65	74	55	25	99
1	854	78	14	8
85	52	62	25	52
69	59	89	3	36

Table 4: Information within sheet one about power usage in Excel template.

Depreciation period	Investment cost	Operating cost
5	30000	642
2	500000	375
4	986333	444
8	632540	786
9	5578452	323

Table 5: Information within sheet one about costs in Excel template.

### 3.2.1.2 Process time

This is the second sheet of the template and contains information about the different process times for different products. *Table 6* shows the structure of this sheet. The values in the table are example values. Seven different products are presented in the table and there is a possibility to add more product variants if needed. See appendix B2 for a picture of the real design of the second sheet in the template.

Station	Product_A	Product_B	Product_C	Product_D	Product_E	Product_F	Product_G
object 1	5	2	3	4	58	9	6
object 2	6	3	6	8	5	2	12
object 3	2	1	4	5	8	6	3
object 4	1	2	5	6	3	2	4
object 5	5	2	56	3	1	4	8

Table 6: Information within sheet two of the Excel template.

### 3.2.1.3 Batch

The third and last sheet of the template contains information about the batch sizes of different products. *Table 7* illustrates what the sheet looks like and contains example values. See appendix B3 for a picture of the real design of the third sheet in the template.

Product	Batch
Product_A	8
Product_B	5
Product_C	6
Product_E	3
Product_F	1
Product_G	4

Table 7: Information within sheet three of the Excel template.

### 3.2.2 Siemens Plant Simulation

There was an already existing library in SPS which the project continued to build upon. This meant that the structure was already in place and the focus could be placed on further developing the functionality and capabilities of this library. Getting the library to be compatible with the Excel template was also important. The project resulted in a file in SPS that contained the code to both import data from an Excel file and to place the objects presented in the Excel file and assign the objects the right values. The program was able to read the information imported from Excel and determine what type of objects and give them different values depending on what type of object they were. If there were a station/machine presented in the Excel file, then the program knew this and assigned the machine the corresponding values. For instance, if there was a station that was written down in the Excel file, the program would notice this and then proceed to look at the right columns to find the data matching the machine. This could be data about processing times, which would not be assigned to a source or a buffer for instance.

The SPS project file is compatible with any Excel file that has the same layout as the template. It is also crucial that the right pathway for the exact Excel file used in the project is copied and placed in a specific module in the SPS file. This is the only installation needed to be done for the SPS file to be able to read the data from the Excel file. This measure is needed for the SPS file to know which Excel file to read and where it is located.

#### 3.2.2.1 Capabilities of the SPS-file

The Plant Simulation model together with the standardized Excel template can handle various types of objects, for instance, machines, buffers, and sources. The objects can be placed anywhere in the SPS program as each object has its own coordinates, which makes it easy to get a high validity layout from start. The objects can also be declared various types of names directly from the template, for instance a machine can be named a "turning machine".

Each object can be declared different kind of information. There are four groups of information that the program can handle:

- Times and failures
- Energy consumption
- Costs data
- Products and batch sizes

The program can handle process times for different machines, but also for different products. For instance, machine A does have a process time of 10 seconds for product A but 20 seconds for Product B. Also, the products can be produced in desired batch sizes. Mean time to repair (MTTR) and availability (AVI) is possible to declare for each machine as well as capacities for buffers. Every object in the program does have cost information as well as energy and power consumption in the different states. All this information mentioned above is written as input data in the excel template and then automatically plotted in SPS.

**3.2.2.2 Template integration SPS**

Following is an example on how the input data in the template will generate a layout in SPS.

Object Type	Identity	Frame	X	Y	Z	MTTR	AVI	Capacity
Source	Source1	1	600	50	150	2	3	
Station	Station1	1	600	100	150	4	6	
Buffer	Buffer1	1	600	150	150	6	9	3

Figure 5: Example of data in sheet one of the template.

Figure 5 shows that we want to plot a source, a machine and a buffer with the information written in the template. The SPS program plots the desired layout as shown in figure 6.

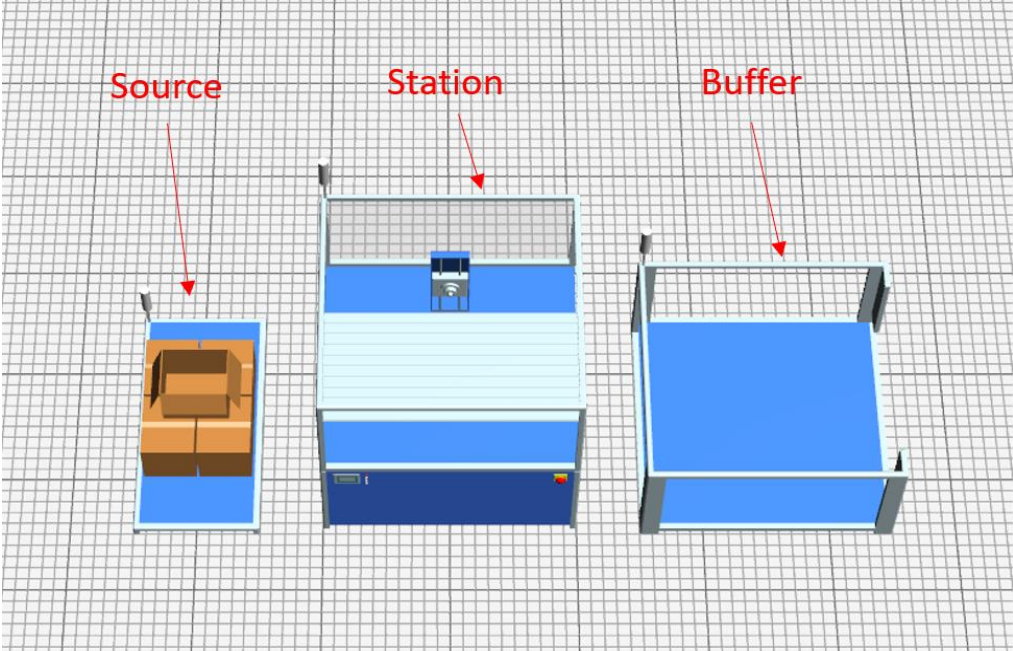


Figure 6: Example of printed objects in SPS.

Buffer1 and Station1 get the input data directly from the template and does not have to be manipulated manually in SPS. Figure 7 shows how the data for availability and MTTR have been automatically added into the properties for Station1, without the need for manual

input. *Figure 8* shows how the capacity of Buffer1 also was added automatically. The program reads the name of the object and correctly assigns the corresponding values.

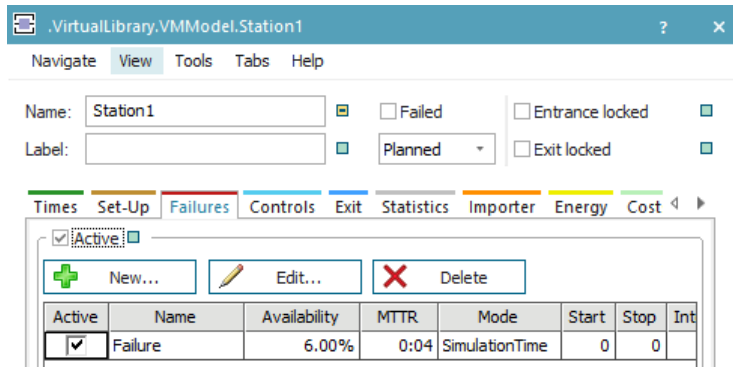


Figure 7: Example of how data is automatically fed into Station1.

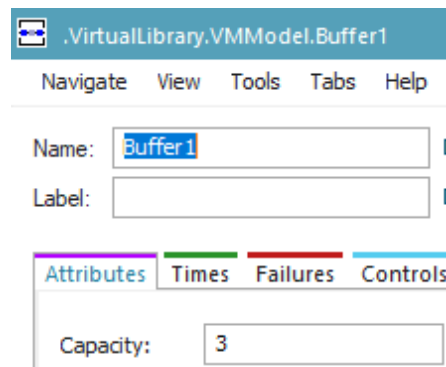


Figure 8: Example of how data is automatically fed into Buffer1.

The energy consumption and costs for the objects can also be filled in the template and then plotted in SPS. The energy consumption in different stages of the process is shown in *figure 9*, and the different cost aspects are presented in *figure 10*.

	Power Consumption Working	Power Consumption Setting Up	Power Consumption Operational	Power Consumption Failed	Power Consumption Standby
Source1	1.0				
Station1	2.0	0.6	0.4	0.5	0.8
Buffer1	1.1				

Figure 9: Aspects of energy consumption in different stages within the template.

	Depreciation Period	Investment Cost	Operating Cost
Source1			
Station1	10000	15000	2000
Buffer1	20000	30000	4000

Figure 10: Aspects of costs within the template.

Station1 automatically gets the power consumption and cost data from the template. These values are automatically imported into the machine. *Figure 11* shows an example of what it looks like when the program has successfully imported the power input values into Station1. *Figure 12* shows what it looks like when the values for costs have been imported into Station1.

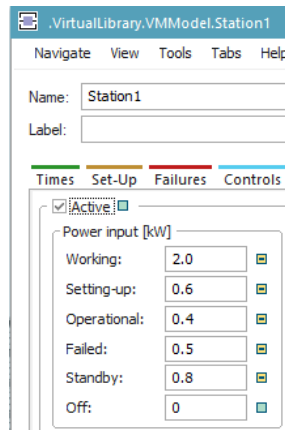


Figure 11: Example of automatically imported power input values into Station1.

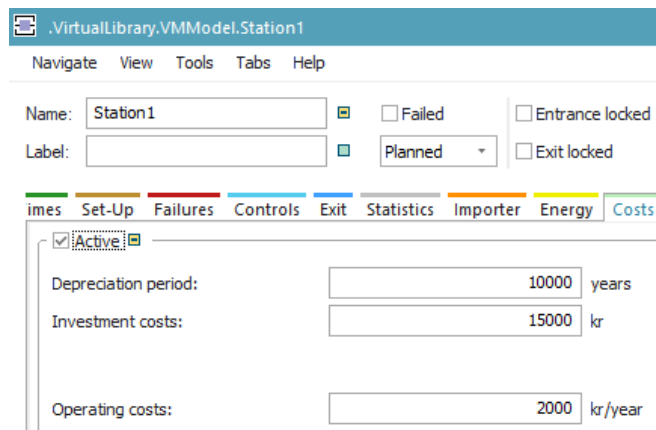


Figure 12: Example of automatically imported cost values into Station1.

As mentioned earlier, each machine can have different process times depending on the product. *Figure 13* shows what process time Station1 has depending on the product. The values presented are example values and they are presented in minutes.

Station	Product A	Product B	Product C	Product D	Product E	Product F	Product G
Station1	1	2	3	4	5	6	7

Figure 13: Example of how Station1 can have different process times connected to different products.


The information is put in a data table in SPS and then read by a code so that each machine gets the right process time. *Figure 14* shows what the process times from the template shown in *figure 13* now looks like in the right table within SPS.

	string 0	time 1
string	Product	Const
1	Product A	1.0000
2	Product B	2.0000
3	Product C	3.0000
4	Product D	4.0000
5	Product E	5.0000
6	Product F	6.0000
7	Product G	7.0000

Figure 14: What the data for process times look like in the table for process times within SPS.

Even batch sizes can be determined in the template and then used in the SPS file. This data is also put in a data table and then read by a code. Figure 15 shows what the data for batch sizes looks like in the template to the left and then what this same data looks like in the table within SPS to the right in the picture.

Product	Batch
ProductA	5
ProductB	10
ProductC	15
ProductD	20
ProductE	25
ProductF	30
ProductG	35



	object 1	integer 2	string 3
string	MU	Number	Name
1	.VirtualLibrary.MovableUnit....	5	Product A
2	.VirtualLibrary.MovableUnit....	10	Product B
3	.VirtualLibrary.MovableUnit....	15	Product C
4	.VirtualLibrary.MovableUnit....	20	Product D
5	.VirtualLibrary.MovableUnit....	25	Product E
6	.VirtualLibrary.MovableUnit....	30	Product F
7	.VirtualLibrary.MovableUnit....	35	Product G

Figure 15: Illustration of what the data for batches looks like in the template to the left and then what it looks like in SPS to the right.

Notice that the template can handle unlimited amounts of objects and products. Figure 16 shows a small simulation model totally made using the template and the connected SPS file. The SPS file is also capable of connecting the objects together. This means that the model is ready for a simulation without manual adjustments.

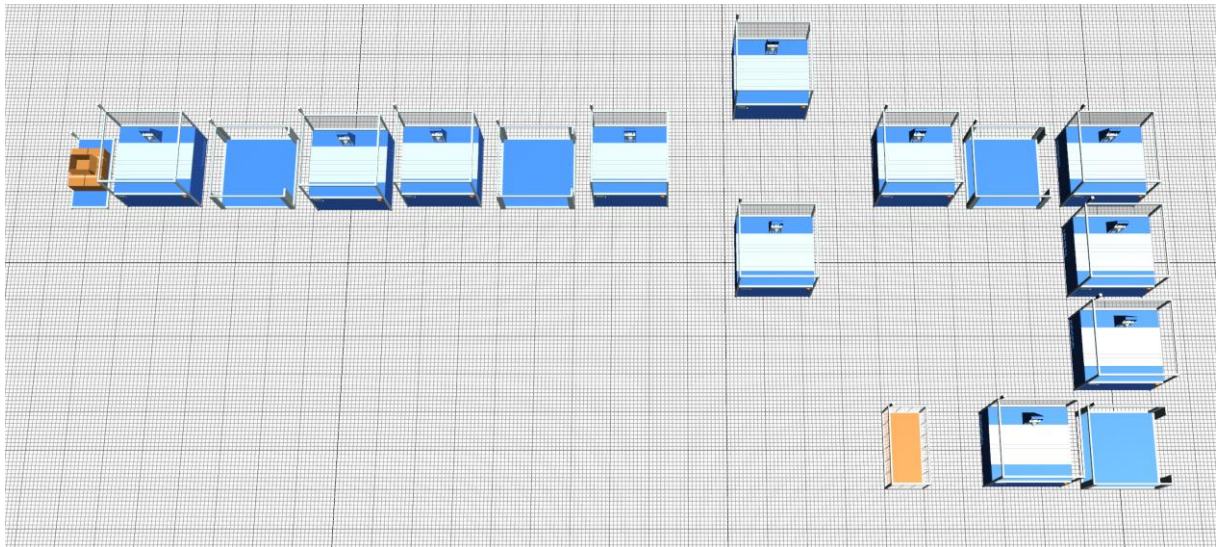


Figure 16: Example of an automatically built layout in SPS.

### 3.2.2.3 Detailed procedure of SPS program

The procedure by which the program works is as follows:

1. First, the SPS file establishes connection to the right Excel file through the correct file path. This path must be manually placed in the method LayoutReader.
2. Then the module CreateLayout is run. This method will place all the machines on the worksheet based on the coordinates that were given in the Excel file.
3. When all the machines are placed the module ReadValues is being called from the method CreateLayout. This means that CreateLayout will be run automatically. What CreateLayout does is to look for information regarding different parameters for the different objects in the Excel file. Then it assigns the already placed objects the right values for these parameters. This module is also capable of reading the object type and distinguishing between the different types of objects to not assign a value to an object which does not use that parameter.
4. ProcTimeReader and BatchReader must be read separately as they are in separate Excel sheets. Those methods read the processing time of the products in the different machines and in what batch sizes the products are being produced.
5. There is also a module called Delete which is used to clear the module of any data in the tables and all the placed objects. This one will mostly be used during this project and maybe not as much later when employees are using the template on real projects.

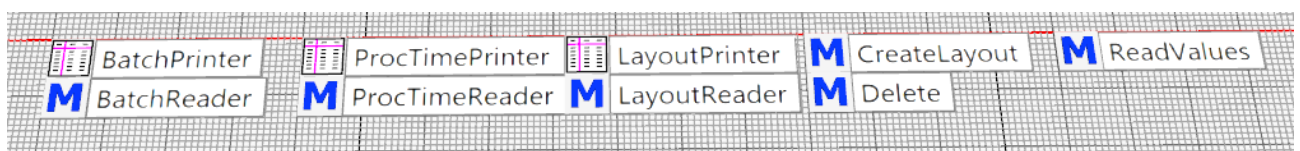


Figure 17: Picture of all the methods used in SPS to import data, build the layout, and run the simulation.

All the methods presented in *figure 17* consist of code that read information from the template and then plot the information in the tables (BatchPrinter, ProcTimeReader and LayoutPrinter).

The BatchReader method reads the batch size from the template and then plot the information in BatchPrinter. ProcTimeReader reads the process times for the machines and products from the template and then plot the information in ProcTimeReader. Layout Reader reads the layout from the template and then plots the layout in the LayoutPrinter. When CreatLayout is run, the layout is being created according to the layout in LayoutPrinter. The ReadValues method reads all the data from LayoutReader (which includes machine data) and gives each object the right information, for instance MTTR.

During the data collection stage of the project, it was concluded that the largest portion of the time a project takes up is to place all the different machines and manually put in all the different values. It does also take a lot of time to go through the information provided by the customer to find all the necessary data. By using the SPS base file and giving the template to the customer and letting them fill in the values, this time could potentially be reduced a lot. The aim of the field test was to see if this base file and standardized template in fact resulted in a considerable time difference compared to manually going through all the data and creating the layout in SPS.

### **3.2.3 Visual Components**

In this part of the report the results from the work in Visual Components is presented. This chapter handles the result of VC. The general description handles the overview of the capabilities within VC. The Excel import block and the template block is then presented in detail where behaviors are presented. In the case of Visual Components, the library did not look the same as the one VM has for SPS. In VC there was not a current library in the sense that there was a file with the needed programming to continue to improve upon. Instead, the work in VC started almost from scratch where the structure had to be made from the ground up. Therefore, the file that the work in VC resulted in is the new main part of the library for projects in VC. The integration of the template into the library of VC did not manage to reach the goals that were set up for the integration. The goal was to create a file that would be able to import data, build a base model using this data and then simulate the process. In the end the file only managed to import the data successfully, but the rest of the goals were not achieved. The test file which was not connected to the template managed to also build the layout, however, not simulate the process. This test file not connected to the template is more capable but is not a part of the library and was only used during development.

#### **3.2.3.1 General description**

The integration of the template to VC was successful. The only difference between SPS and VC is that the Excel file must be converted to a CSV format for VC to be able to read it. In SPS an Excel file can be read and the data can be placed within a table. Similarly, in VC the data from the CSV file can be read and placed in a note within VC. Then there is also the possibility to use the information from the note to create a layout. The file needs two objects to work. The first one is a template block. This block is programmed with varying geometry through a switch. The data for placement, MTTR, MTBF, availability, process times and precursor can also be placed within the built objects. The template works in such a way that it can take on all the properties mentioned above and change geometry depending on what type of object that is going to be placed. The other block that is needed is the block called "excelimport". This block has the code that reads the data from the CSV file and splits

it accordingly and places it in the note. It also has the code to build the layout by using the information in the note and looking at the template block. The pathways for the CSV files are also placed within this block.

The file works in such a way where the “excel import” block runs everything. This object initiates everything, handles the data in the note and takes commands from when the different buttons are pressed. Then with the information given to the “excelimport” object, it uses that information to make clones of the template block and change the geometry of it in various ways to visually illustrate the different types of objects. A buffer looks different than a station for the employee to differentiate between them more easily for example. However, there is a text besides all the objects in the layout also that displays the names of the objects. The template block then does not really do anything else than to be a starting point for the rest of the objects. The template block is only copied in a loop within the “excelimport” object where each iteration of the loop that builds the model looks at the data and creates a different looking copy of the template block.

To give specific properties based on the information from the note, the program looks at each column. For example, column one is the station name. Therefore, the code looks at column one to decide what the next object in the layout will be called. To find the right names and properties for each object it looks at the letters “VM” in the name. The program is currently programmed to search for those two letters to find the object names and to know what type of object it is. This also means that the objects must be named this way in the template as well. So, the objects must be called VMBuffer, VMStation and VMDrain for example. These letters are used to give all the objects a common identifier. This can be changed to whichever letters or signs that are desired. This must be changed both in the Excel template and in the code. The same goes for when a column is to be added or removed from the template. This is because currently each column header needs to be programmed into VC for the program to know what to expect.

This project resulted in two program files for VC. The first one is a test file which was used for development and understanding of VC. This file uses a test CSV file, and this project file can read the data from this CSV and then print all the objects within this CSV file correctly. The second file is the program file connected to the Excel template which the project resulted in. This program file can import the data from the four CSV files that are needed to read the data from the four Excel sheets. However, the program file that is connected to the template cannot use this information correctly to build a layout. Neither of the files can correctly simulate the process.

### **3.2.3.2 Excel Import block**

The Excel import block is the block that has all the code for importing and splitting the data and placing it in the right notes. It also has some of the code to build the model.

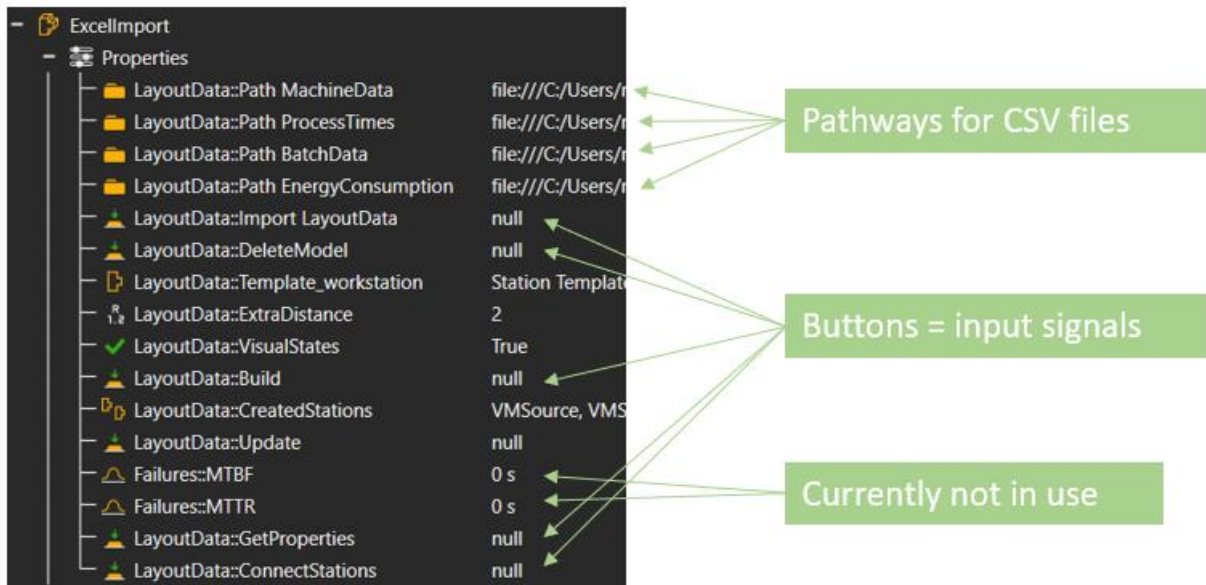


Figure 18: Properties within Excel import block in VC.

Presented in the *figure 18* are the properties that are placed in the Excel import block. The arrows show the unique resource identifiers (URI) which have the pathways for the CSV files within them. The reason for why there are four of those is that the CSV format is only capable of handling one Excel sheet of information. Therefore, the template that consists of 4 sheets of data needs to be saved down into 4 individual CSV files and placed in the right URI. This is done either by copying the pathway and placing it in the right input box, or it can be done through pressing the button to the right of the input box and choose the right file. This is shown in *figure 19*.

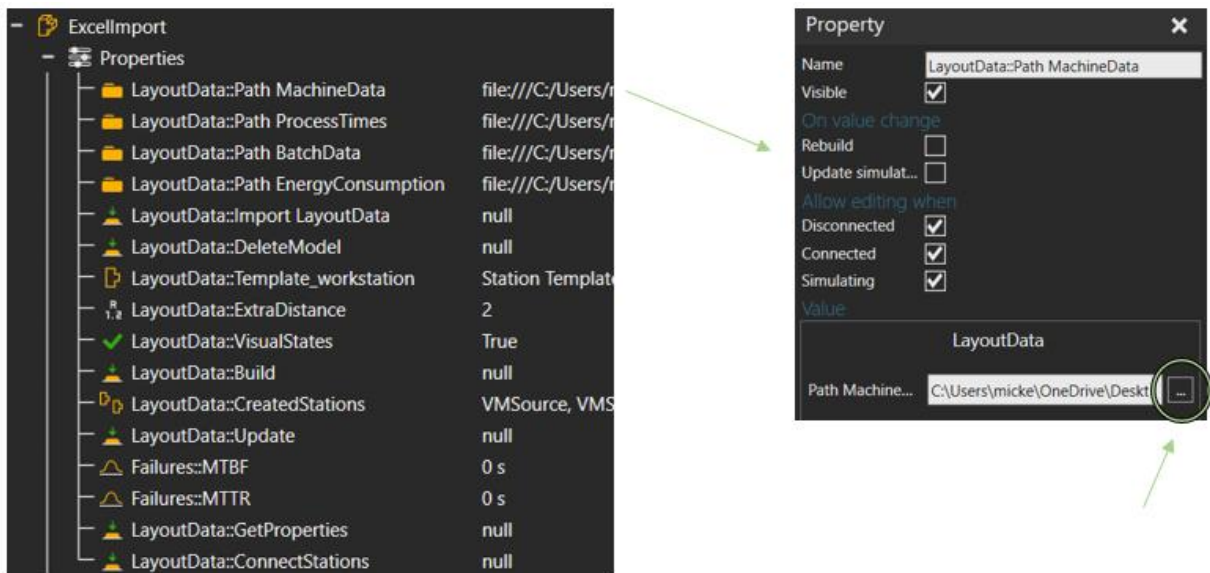


Figure 19: Picture of how to choose the right CSV file in VC.

There is also one python script for each pathway to import the data and split it correctly and place it in the right note. The pathway needs to be placed within this python script as well.

The pathway can easily be copied from the input field in *figure 19* and then be placed within the script like in *figure 20*.

```
5 uri = r' C:\Pathway_for_example.csv '
```

Figure 20: Example of where the user needs to place the pathway within the code in VC.

In the Excel import block, there are also some defined behaviors like the python scripts for example. The behaviors are presented in *figure 21*.



Figure 21: Behaviors within the Excel import block in VC.

Here there are many different python scripts. These are the ones with the yellow and blue symbol in front. The four first ones that start with “VM\_Read” are the ones that handle the information from the CSV files. The differences that exist between these four scripts has to do with how VC works. To read data from a CSV file the expected headings need to be presented. Since there are different headings in the different sheets they also differ between the scripts. The script called “BuildModel” is the one containing parts of the code needed to build the layout, the rest of that code and the information about the template block is located within the template block. The four notes following the scripts is the places where all the data is stored. For each sheet in the template there is also a corresponding note to keep that data in. The data is split in some ways. Firstly, the code tells the program to split the data into a new row every time it finds the first column again. Secondly the columns are separated with this sign “;”. Every time the program finds that sign it knows to handle that as a new column. This way it knows what information the user is trying to collect or use by knowing that all the machine names are located within the second column for example. An example of what the data within a note could look like is shown in the *figure 22* with example data.

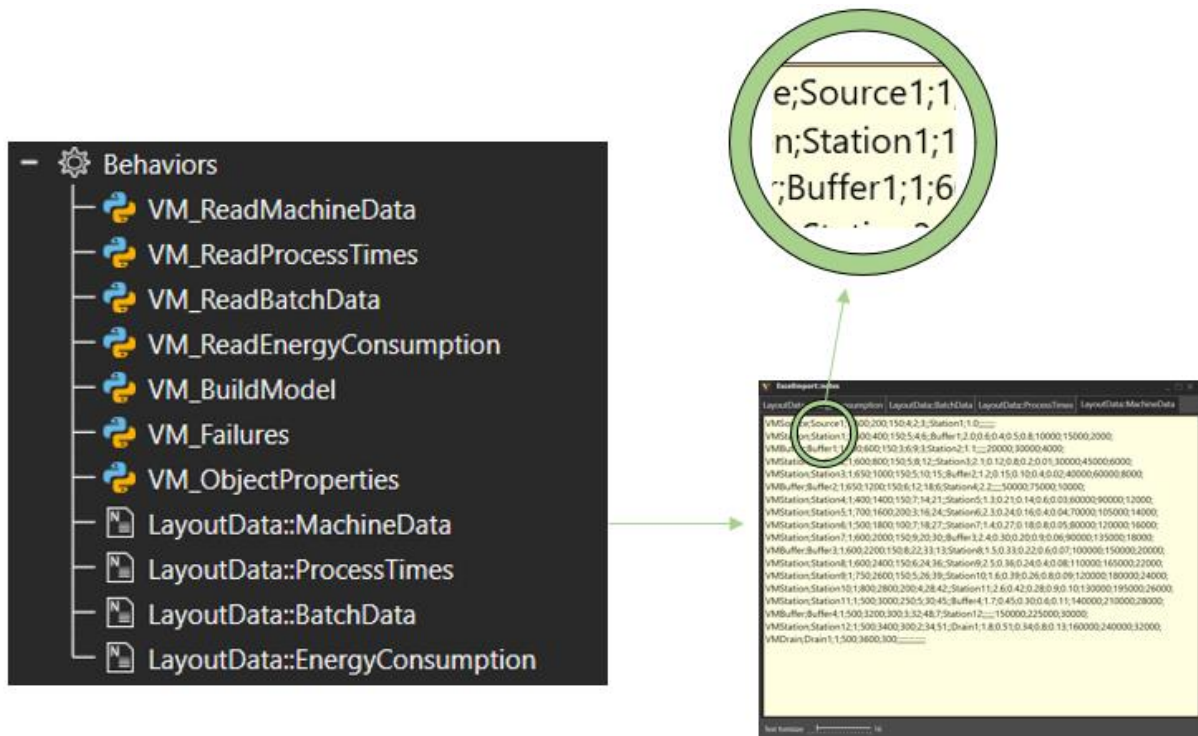


Figure 22: Example of how the data within the notes in VC are split.

All the necessary buttons for input signals to the program are in the Excel import block as well. These are the button for importing data from all the CSV files. This one button is connected to all the four scripts and when this button is pressed all these scripts are run simultaneously. Then these scripts import, split, and place the data in the correct note. There is also a button for deleting the entire model. This button does not work if the build was unsuccessful. This button only works when everything managed to print correctly based on the data provided. Being able to delete all the machines by only triggering one input saves time. This is especially useful if the user quickly wants to delete the current layout, change the values in the data a bit and then build it all again. The data can be manipulated right in the note within VC and there is then no need to provide the pathway for a new CSV file. The changes made in the notes does not affect the data in the CSV file. There is also a button called “build”, and this button is the one which triggers the program to create the base layout by activating different scripts. One press of this button and the entire layout with the correct names and geometries are provided. The location of the objects in the program also corresponds to the data about coordinates provided through the CSV. This is shown in *figure 23*.

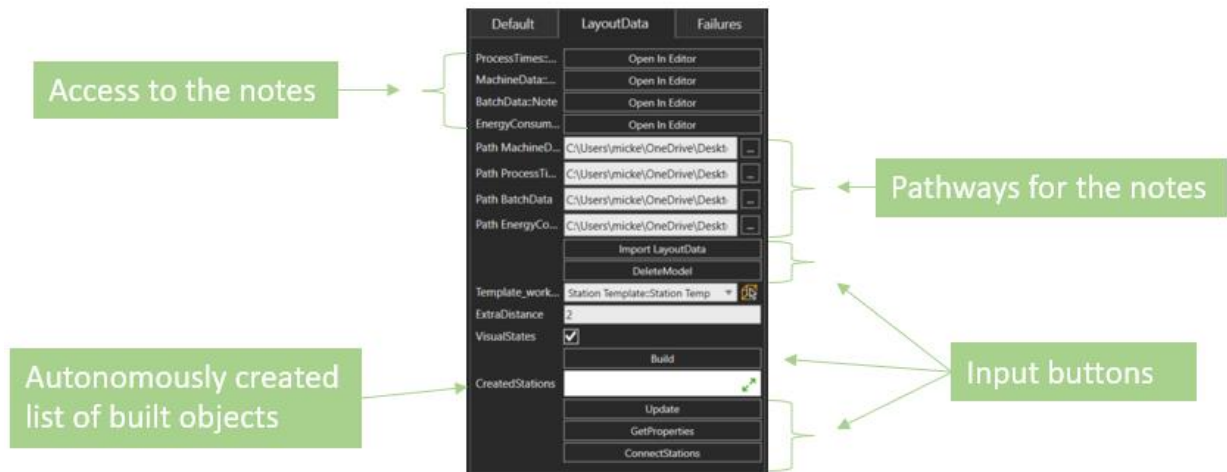


Figure 23: What the window with access to notes and buttons look like in the Excel import block in VC.

The buttons that are currently not in use are the buttons “Update”, “GetProperties” and “ConnectStations”, these are shown in *figure 23*. The one called “Update” is currently connected to do the exact same thing as the build button. The other ones are there partly to show the thought of how the template should work and because there has been some work on getting these buttons to work, but complications in other areas shifted focus. The white boxes right under the four buttons to access the notes are where the pathways are supposed to be declared so that the program knows where to find the information. Access to the four different notes with information is enabled through the four buttons in the top of the picture. Here the user can access them and see if the data is being presented correctly and there is a possibility to edit the data as well. The autonomously created list is a list that the program fills in as the layout is built. This box is empty in the picture since there was no layout built when the picture was taken.

With the use of the buttons “Import LayoutData” and “Build” the user can import data and build a base model accordingly. In *figure 24* a basic layout is presented. This layout is the one generated by the test file within VC and is built entirely using these buttons. In this case the picture is from when a test CSV file was used with random values within. This layout is therefore only a simple example of how it possibly could look when the template and the connected programmed VC file is used to build a base model.

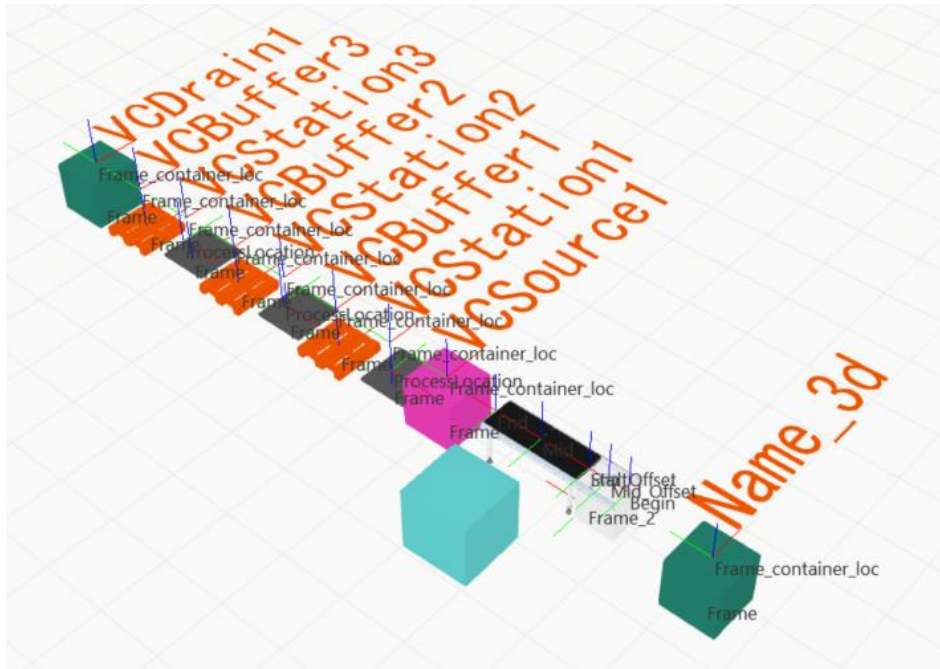


Figure 24: Example of what a built layout in VC can look like using only the buttons in the program.

### 3.2.3.3 Template block

The template block is the block that is being copied and printed repeatedly to build the base layout. This block also has the capability change geometry depending on what type of object that is to be printed. The current four accepted types of objects are buffers, sources, drains and stations. For this to work the template is programmed to accept one of four different geometries or colors. When printing the new objects, the program also knows what number to assign to each type of object. This number then corresponds to the right number for that specific geometry. The four different possible geometries are shown in *figure 25*. These geometries have been randomly chosen to differentiate between different object types when printed. They can be changed to other geometries at any time. The important thing is to change the appearances within the template block itself and not on the printed objects in the layout.

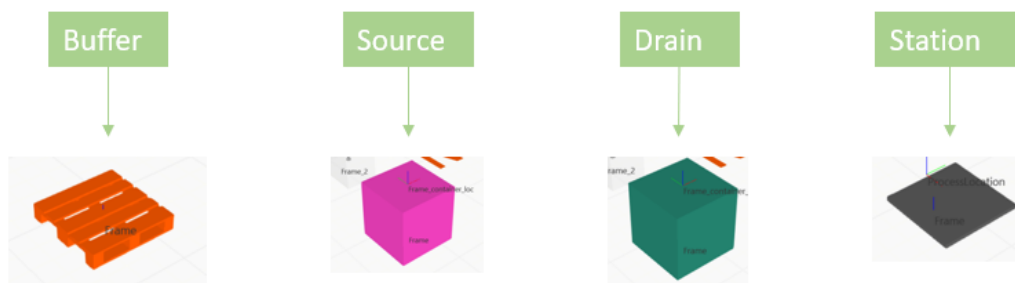


Figure 25: Example of what the current four types of objects within VC looks like.

The different geometries are presented as in *figure 26*. The source and drain which are only simple blocks have no “transform” within them. Instead, these are built up by simple geometries, in this case a block. The buffer and station have more complex geometries with more things to them. These are all placed under the “switch” which enables the possibility to

change between these four geometries depending on what type of object that is to be printed. This can all be seen in *figure 26*.

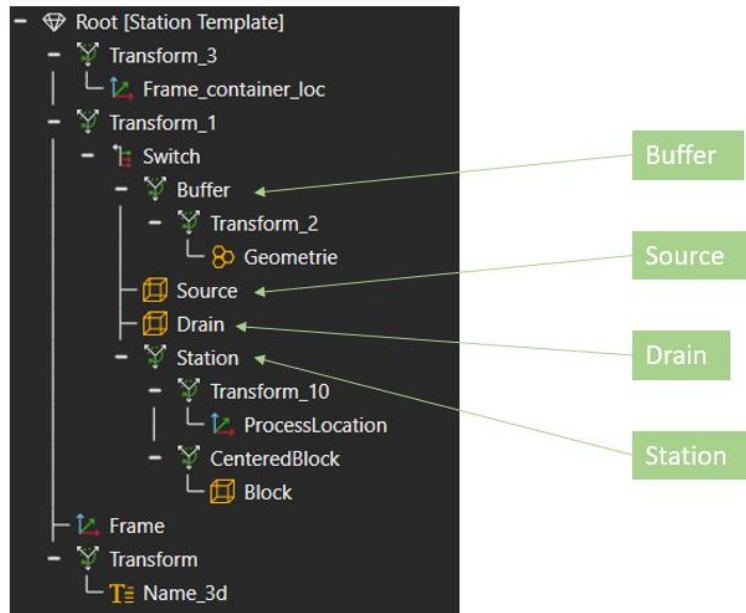


Figure 26: What the switch and the geometries for the different objects in VC looks like.

Figure 27 describes how the program is told what number each object type has. As stated in *figure 27* the object type “station” has been given the number four. When the script then runs the loop which is creating all the objects in the layout, the variable will have the value of four each time a row of data in the note contains a station. It will then also choose the fourth geometry when creating a copy of the template block.

```
dict_symbol_geo = {"VMStation":4, "VMBuffer":1, "VMSource":2, "VMDrain":3}
```

Figure 27: The row of code for declaring what value to assign to each object type in VC.

The properties that have been put into the template block are the ones presented in *figure 28*. The one called “Name3D” is responsible for name tags besides all the printed objects. This is so that it is stated clearly what every object is so that the one working with the layout does not have to count objects and try to figure out which station is station four for example. The one called “resource\_type” has to do with the geometry of the block. In this example it is of the value four, which means that it currently has the geometry of a station. Another interesting one is the one called “Next\_Obj” which is a place where the information about which station that is next in line. If this is station number four, then the next object for this station could be station number five for example. This information is used in the process of linking the objects together for simulation. The information that is placed in “Next\_Obj” is the data from the column precursor in the first sheet of the template.

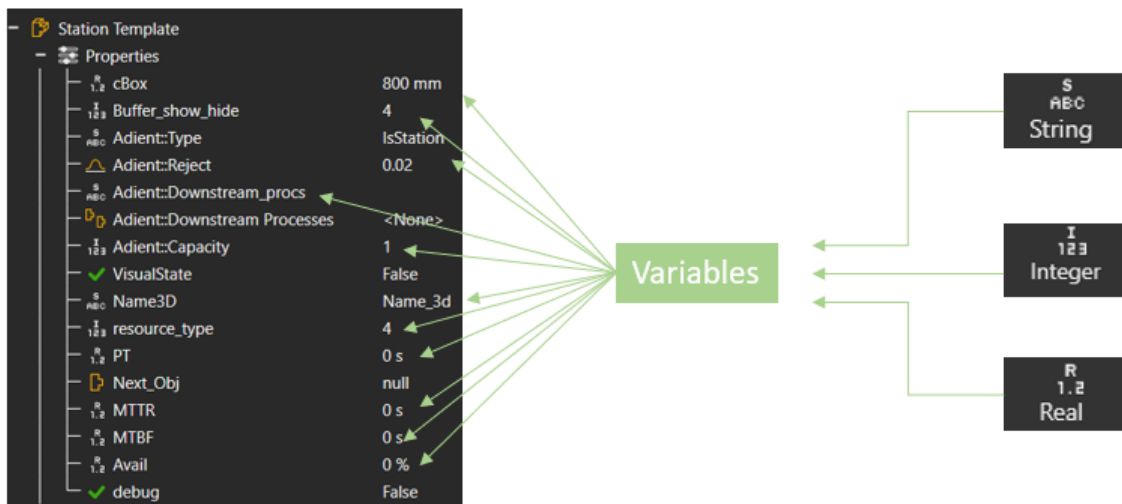


Figure 28: Example of some of the properties within the template block.

All the rows in *figure 28* that have arrows pointing towards them are the different variables that the template block has. Each newly printed object also has these same variables since the template block gets copied when the layout is built. The data from the note is what these variables are connected to, and the values therefore differs for each object. There are three different types of variables. There are strings, integers, and real variables. “PT” has the process time for the station, “MTTR” and “MTBF” have the mean time to repair and mean time before failure respectively and the capacity one holds the capacity of how many products the object can hold at one time. The rest of the properties have some function within getting the program to print properly or show the right design. The “cBox” one for example is a variable that is constant. This one is used for distancing and length, width, and depth of the objects. The program would work without this variable. However, the way that it is now set up is that there are many things depending on the value of this variable. For example, the length along all three axes for the cubes that make up the source and drain are decided by this variable. This way by changing the value of cBox the sizes of these blocks are directly and easily changed without having to make changes in the code. This variable can also be a part of the distancing between the objects in the layout. Currently the layout follows the exact coordinates of the data, but if the objects were to lie very close to each other or even collide, then this cBox could be used to create an extra offset.

The behaviors connected to the template block is the ones presented in *figure 29*. The signals are not currently used since the program is not capable of simulating the process properly. These signals are connected to the parts that flow through the layout. They give of and receive signals and the objects then also know what to do with certain signals. The reason why the notes does not have any data within them now is that they are not connected to the data currently. These notes were in use when the test CSV file was in use, but now when the template was used instead the functionality of these notes was not made possible. The use of these notes was to import just the specific data needed for that variable.

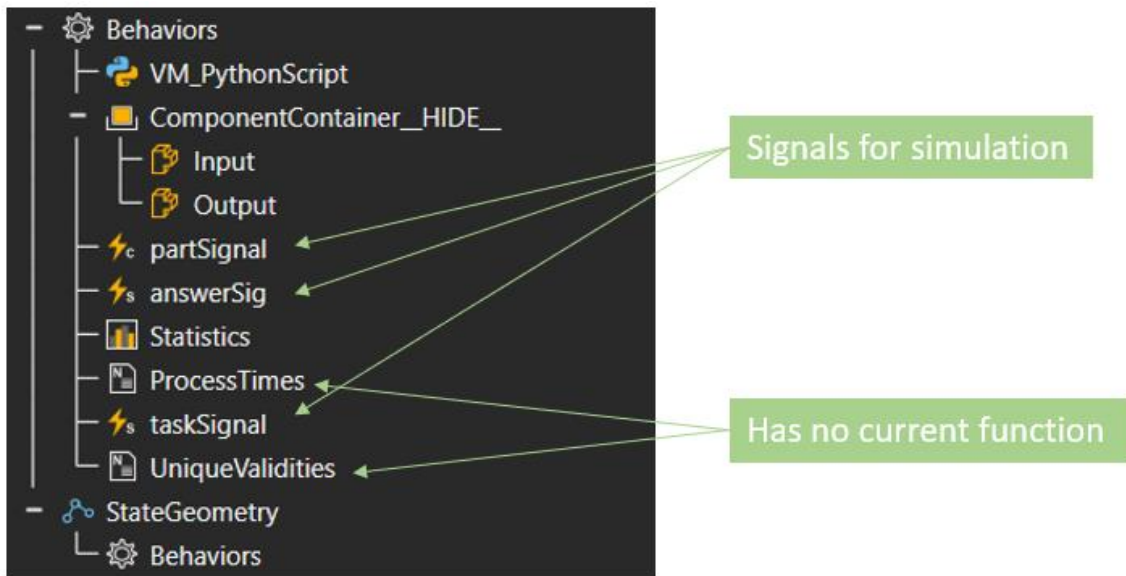


Figure 29: Behaviors of the template block in VC.

### 3.3 Testing and verification

This section will cover the results of the real-case test.

#### 3.3.1 Test SPS

The field test of the SPS base file and the template resulted in time saving when a real-case test was conducted at VM. The test was a smaller part of a larger project which an employee at VM was responsible for, and it took 15 hours to manually build the base model of this facility and make it ready for simulation. Overall, the most time during this project was spent on sorting data from the customer, setting up tables in SPS, placement of machines and get an accurate flow simulation, according to the employee at virtual manufacturing.

The real-case test included the following:

- 13 machines
- 4 buffers
- 1 source and 1 drain
- 112 different products
- A real DWG layout from the manufacturing company
- Several parallel stations

With the standardized template, the total time for generating the layout took 40 minutes and for completing the final layout 45 minutes. This means that the final changes to make the layout look like the DWG and connect parallel stations only took 5 more minutes after the plotting had been made. A total saving of 14 hours and 15 minutes was made when using the template. It is important to consider that once the employee used the template to build the base model, it was the second time that the same model was to be built.

Therefore, the employee had previous knowledge of what the facility looked like. However, this only effects the 5 minutes of manual adjustments since it is there the employee could have possibly been faster because it was the second time building the model. The first 40 minutes mostly consisted of manually filling in the Excel template with the right data.

### **3.3.2 Test VC**

There was a decision made to not carry out an experiment for Visual Components like the one that was done for Siemens Plant Simulation. This decision was based on the fact that the progress made in VC was not as great as the progress that was made in SPS. Furthermore, the program in SPS made it possible to simulate the process after the layout was built, but this was not the case for VC.

## **4 Discussion**

This chapter consists of the discussion of the different parts of this project. The discussion is divided into three different parts. The first part is the results part, where the results are discussed. The second part handles the potential future developments that are possible, and the third part handles the project execution, plan, and preparation.

### **4.1 The results**

This part of the discussion will handle the results. It is divided into the three phases of the project. These are data collection, library implementation and testing & verification.

#### **4.1.1 Data collection**

The literature study as well as informal communication with the employees at VM highlighted the need of organizing and categorizing the data to make it easier to understand and use. The main concepts that were important to have in mind when the template was built were the concepts contextualization, categorization, and condensation from the literature study. Contextualization to understand what data to focus on and what headings that needed to be included in the template. The categorization in the way that the information was divided into different parts and sheets within the Excel template. Lastly the condensation concept was important when making the template transparent and easy to read. The most important and mostly used data needed to be placed early in the template. The informal communication with employees working with discrete event simulation projects combined with the questionnaire survey contributed to the knowledge of what data to include in the template.

#### **4.1.2 Library implementation**

The project did result in a template and two template files in the programs SPS and VC. In this case, the project focused on making both programs compatible with the template. The project would have resulted in a more compatible and advanced template file if only one of these programs would have been chosen for the project. However, the decision was made to instead focus on showing the capabilities of the template itself and show that it is compatible with multiple programs. The reason why the progress in SPS is far greater than the progress in VC has to do with the limited access to the one employee who had sufficient skills in VC at VM. VC was therefore harder to work with. The access to support from VM was different between the programs, and VC requires more programming. With SPS there is a possibility to connect any Excel file to the SPS file and only the headings that the user wants needs to be programmed into the program. In the way that the VC program file is currently built-up using CSV files, the headings for all the columns in the CSV file needs to be manually programmed into the code. This was also one of the reasons why the VC library file is unable to build the model. It is because the modification and development of the Excel template was done for a long time in the project when the capabilities of SPS were continuously improved and built upon. The group decided to wait until the template was finished before connecting it to VC, since each new heading that was added after that would create issues in VC if they were not manually written in the code within VC. The group could have made two versions of the template. One that is connected to VC and one connected to SPS where improvements could be made. However, it was important that the exact same Excel template needed to be compatible with both programs. The program file in VC currently has

three scripts all for importing one CSV file each. The user therefore cannot connect a CSV file without also changing the expected headings for each CSV file.

One of the reasons why the VC file that is compatible with the template is useful for the employees, even though it currently is unable to correctly build the layout, is that the user can import the desired data into VC by only changing the pathway for each CSV file. No other work needs to be done if the Excel template is used. The template currently has four sheets. The fourth sheet does not contain new or necessary information currently. This is because the group started the process of further developing the template and to take the data about power consumption that is in the first sheet and move that data to the fourth sheet. This so that there could be an entire sheet dedicated to only environmental aspects. This however was not finished since this task was started quite late in the project and there was no time left to finish it. But this is the reason why there are four CSV files presented in the results chapter for Visual Components.

#### ***4.1.3 Testing & Verification***

The experiment carried out for SPS showed that there are possibilities and capabilities to shorten the time it takes to build the base model in DES projects. A time reduction of 14 hours and 15 minutes for a task that initially took 15 hours is a large reduction in time. To emphasize what type of difference this could do in real life it can be seen in the way that this single employee theoretically could initiate 20 projects of similar size in the same timeframe. It takes the same amount of time to build 20 base layouts using the template as it took to build one base layout manually. The standard practice is not to initiate too many projects, but instead finish the ones that are started, but this thought illustrates the difference the template has shown to make. In this case the experiment was based on a real project and was therefore representative of those types of projects that the template could be applied in. The experiment was only carried out once and by one employee. However, this employee almost entirely works with simulation projects in SPS and has been involved in many projects since employed. Even considering that the experiment was only carried out once, it still proved the capabilities and benefits of using the template with the connected library file. The experiment showed that if this template in combination with the library file is used for DES projects, then a bit of the monotonous work could be reduced for the employee and more time could be spent working with the improvements. This time could either be money saved for the customer or a deeper and more thorough analysis of the facility where there is also more time to analyze improvements.

#### ***4.2 Further development***

There are opportunities for further improvement of these template files and the template itself. The template can always be modified to fit better with any specific project but is rather complete as a base template that covers the most information needed. A big thing is that through the template files the information from the template is also being used in the right way and a lot of time can be saved. If there would have been more time available for further development of the template files, the group would have focused on the following things. Currently the different times that the template can handle, especially the process times, are only able to handle certain distributions. A way to further develop the SPS file would be to have it handle even more different distributions.

In VC the possibilities for further development are larger since the progress made in VC was less than the progress made in SPS. Without using any other data than the information currently available in the template there would be many improvements possible. The VC file could be made to handle the different distributions for different times. Be able to create batches and send different batch sizes depending on data from the template. Environmental factors such as energy consumption could also be programmed to use the information from the template. A lot of graphical improvements are also possible within VC, where the design of the different types of objects could be improved to be clearer and make it easier to understand what type of object that it is representing. Differences in size, shape, and color for example could be applied to the components as well so that it would be easier to differentiate between product a, b, and c. The program file could also be made to print all the objects successfully according to the imported data. This function was made possible in the test file. It is therefore not far away from being true for the program file using the template as well.

There are even more possible improvements to be made to the VC file after the progress is up to the same level as the current one in SPS. There can be more work put in to make the file be able to place the objects as they are placed in real life instead of just on a line. The program could also be able to place machines in parallel and send the components accordingly. The SPS integration can also be developed further to make even more complex simulations from start, using the template. AGV's would be useful to implement to the template, which requires a lot more work and knowledge, but would save even more time when building a production layout.

More variables including set-up time, recovery time and cycle time can be used in the template as well. This will improve the validity of the simulation model and save time for the engineer when building the layout. For further verifying the benefits of the use of the template in combination with the program files is to carry out more experiments. One or more experiments could be made in SPS where at least one of these experiments could use a larger layout. This experiment could investigate if the savings of time get larger for even larger layouts, thereby looking into if the benefits of using the template could be larger for big projects. Experiments could also be carried out in VC to investigate potential savings of time. There could also be experiments with different sizes of layouts constructed for VC.

In the limitations chapter it was described that the project would not handle the use of programs that handle data. The data is currently placed in the template manually by an employee either in a customer company or a VM employee. Therefore, a recommendation for further development would be to investigate possible connections between a data handling software and the template. It could be investigated if the template could be connected to these programs and be automatically filled with data to even further reduce the time that the start-up stages of DES projects take. The employee could then just download an already full excel template, place the pathway to that file in the correct places within the SPS file and then just run the model and a base model would be built. The template could possibly be connected to a data management tool directly which could fill the Excel template with the necessary information for the project and the employee would not have to fill it manually.

### **4.3 Project execution, plan & preparation**

The initial project plan was not entirely followed, and more time could have been spent analyzing the project plan. This plan could have been more detailed before execution of the project. More detail about how to carry out the survey and try to reach other companies as well to get information that is more representative of the industry as a whole and not only how one company is working. A plan should also have been made for a failed survey. This was the case in this project where the survey was not representative enough since there were only two participants. A detailed action plan could have been made and presented earlier. Due to late answers and new priorities throughout the project, the plan was not followed exactly.

There were some modifications to the plan during the project. However, the contents of the plan were followed, just not in the same order as the plan. The initial plan was to work in SPS first and then when we reached the right level, we would have the experiment to see whether it made any difference. After that, the group should have continued with the work in VC and then carried out the experiments for VC as well. The last step would have been to implement the program files into the existing libraries at VM, but the project turned out to go the other way around. There should have been a thorough interview with the supervisor early in the project to already at that stage understand what resources were available. This would have been especially important considering that the existing libraries looked very different and that there was a need to also construct a library within VC to build upon. The group started working with SPS as planned, however the work in VC began later than planned and the experiment was saved until the end of the project. Also, more capabilities than initially planned were added to SPS partly based on the suggestions from the questionnaire survey. This delayed the finished Excel template a bit since it was always improved upon with more capabilities which lead to it being connected to VC very late in the project. Therefore, a more detailed list of the exact expected contents and capabilities of the Excel template should have been included in the project plan. This would have made it possible to import the finished Excel template to VC in an earlier stage, and possible improvements in SPS could have been made in a separate version of both the SPS file and the Excel template. The finished one should have been saved when done and the improvements could have been done in a separate file.

The group started off working in SPS as planned but started to work within the library from the very beginning. A lot of progress was made fast, but the group thought that a test would be unnecessary at the time since there were months of time to improve the model. Therefore, the work continued with VC and progress was made in both programs before the group reached the end point where the improvements stopped and instead focused on working with the experiments.

The reason why the actions in the plan were flipped was because the group gained more information about both the programs and the challenges with them as the project went on. Initially the group did not know the status of the libraries, for instance when the group noticed that the library in SPS already existed and could be extended and built upon, it was decided to start from there. This decision was made because there was no point in building a project file from the ground up in parallel to the already existing template file. Especially since what this project needed to result in was partly the template itself, but then also two

template files within the programs for the template to be compatible and be able to use the information from the template in a correct way. One reason for why the group started working in SPS and not VC was because of the knowledge available at the location that the group was working from. The group worked from the office at VM where the people to which the group had closest contact worked in SPS. Therefore, the knowledge in SPS was easy to get access to and there were many employees who could assist when problems occurred. In VC there were only one employee with sufficient knowledge and skills in VC to help with the complex programming of the template file. This was why the work in VC started a bit later than when the work in SPS began because the knowledge was harder to come by.

The plan should then have been modified earlier on in the project when additional knowledge about the way VM works and what access the group would have to the different knowledge. The work in VC could have started before the work in SPS. This would probably have resulted in more progress made in VC, but how much more is uncertain. The main thing is that the progress in SPS would probably suffered a bit and the level of progress would probably not have been the same as the current results. However, since SPS was more widely used, it would maybe have been better to focus more on VC and instead focus on investigating and giving suggestions for how the employees could have continued the improvements of the library in SPS. This is because it is more likely that the employees would have focused on building the library in SPS than in VC which less people use. There are also benefits to the way that the project was carried out since the now current possibilities in SPS and the connection between the template and SPS can be implemented into the working procedure of the employees as of right now. The employees at VM can start using the template now and continue the improvements of the template based on what the specific use is going to be.

There could have been more preparative work in the beginning of the project to retrieve more information early on about the possibilities with the resources available during the project. More in depth plans could also have been made where a more detailed timeframe could have been developed and a clearer distribution of tasks among the members of the group. Unexpected days off could have resulted in problems if the plan would have been very detailed, since there might not have been space within the plan for catching up when work has not been done in time. However, this would perhaps have been easy to incorporate into the plan anyway. During the project there has been numerous unplanned sick days, mandatory lectures and even a combination of these together did create multiple days where no progress was made.

## 5 Conclusions

This study managed to deliver an Excel template, one file in Siemens Plant Simulation connected to this template, one library file in Visual Components connected to the template and a successful experiment which showed that using the template and the connected SPS file saves time. The group managed to analyze what general data that is usually needed for DES projects which include process simulation. After finding the necessary data to include for creating a DES project, a template was made. This template contains the main information needed to build a base layout for simulation projects. The template is not specific and detailed in a way that it limits the use to only certain projects. However, it is suited mostly towards DES projects working with production facilities. The template was successfully implemented into the libraries of both Visual Components and Siemens Plant Simulation. However, the program file for SPS is more advanced. Even though the VC program file is not as advanced, the group successfully made a test file in VC which can build a base layout based on simple data. The project resulted in a new library in VC which can be continuously built and improved upon. This new library file in VC is not able to build the layout using the information from the Excel. However, it can import the data from the different sheets in the Excel template and place the data in the correct note. The group carried out an experiment to verify potential reductions of time when using the template for DES projects. The experiment was only carried out in SPS since the VC file connected to the template is incapable of building the layout. The experiment was based on a real case project that VM carried out for a customer. When this project was done initially it took 15 hours to manually build the base layout and get it ready to simulate. Using the template, the time of 15 hours was reduced to 45 minutes where it took 40 minutes to fill the template correctly, import it into SPS and build the base model. The last 5 minutes was used for minor manual adjustments. Therefore, the group is satisfied with the results of the project since it made a difference in time for the initial stages of DES projects. Another reason why the group is contempt with the project is that there now exists a library to continue to improve upon in VC like the one that was already established in SPS.

There are some future developments that can be done. This is especially true for the VC library which is still in the early stages of development. There are some improvements that can be done to get the VC file to build the base model properly and to be able to simulate. It is true for both programs that there are development opportunities to further enlarge the capabilities of the files. These libraries could either be expanded to incorporate other types of simulation projects than the ones focusing on production facilities, or they could be narrowed down to be even more specialized to one type of projects.

## 6 References

- Air pollution*. (n.d.). World Health Organization . Retrieved February 28, 2023, from [https://www.who.int/health-topics/air-pollution#tab=tab\\_1](https://www.who.int/health-topics/air-pollution#tab=tab_1)
- CAD Software | 2D And 3D Computer-Aided Design | Autodesk*. (n.d.). Retrieved May 9, 2023, from <https://www.autodesk.com/solutions/cad-software>
- Cockerham, J. (2020, April 2). *What is Industry 4.0 | Definition, Industry Types & Future Benefits*. ManagerPlus. <https://managerplus.io/officecorp.com/blog/industry-4-0>
- DWG – Wikipedia*. (n.d.). Retrieved February 23, 2023, from <https://sv.wikipedia.org/wiki/DWG>
- Eisenhower's Urgent/Important Principle - Using Time Effectively, Not Just Efficiently*. (n.d.). MindTools. Retrieved February 28, 2023, from <https://www.mindtools.com/al1e0k5/eisenhowers-urgentimportant-principle>
- Hanna, K. T. (n.d.). *What is a template?* Retrieved February 24, 2023, from <https://www.techtarget.com/whatis/definition/template>
- How Can Factories Affect The Environment? | Field*. (n.d.). Retrieved February 28, 2023, from <http://www.field.org.uk/how-can-factories-affect-the-environment/>
- Industry — European Environment Agency*. (n.d.). Retrieved February 28, 2023, from <https://www.eea.europa.eu/themes/industry>
- Plant simulation software | Siemens Software*. (n.d.). Retrieved May 8, 2023, from <https://plm.sw.siemens.com/en-US/tecnomatix/products/plant-simulation-software/>
- Protect your brain from stress - Harvard Health*. (2021, February 15). Harvard Health Publishing. <https://www.health.harvard.edu/mind-and-mood/protect-your-brain-from-stress>
- Ritchie, H., Roser, M., & Rosado, P. (2017, May). *CO<sub>2</sub> and Greenhouse Gas Emissions - Our World in Data*. Our World in Data. <https://ourworldindata.org/co2-and-greenhouse-gas-emissions>
- Skidelsky, R. (n.d.). *What effect does automation have on the economy? | World Economic Forum*. Retrieved February 28, 2023, from <https://www.weforum.org/agenda/2019/09/the-economic-consequences-of-automation/>
- Skoogh, A., & Johansson, B. (2007). *TIME-CONSUMPTION ANALYSIS OF INPUT DATA ACTIVITIES IN DISCRETE EVENT SIMULATION PROJECTS 5GEM-5G-Enabled Manufacturing View project REWIND-Enabling REuse, REmanufacturing and REcycling Within INDUSTRIAL systems View project TIME-CONSUMPTION ANALYSIS OF INPUT DATA ACTIVITIES IN DISCRETE EVENT SIMULATION PROJECTS*. <https://www.researchgate.net/publication/235719405>
- Skoogh, A., Johansson, B., & Stahre, J. (2012). Automated input data management: Evaluation of a concept for reduced time consumption in discrete event simulation. *SIMULATION*, 88(11), 1279–1293. <https://doi.org/10.1177/0037549712443404>
- Skoogh, A., Michaloski, J., & Bengtsson, N. (2010). Towards continuously updated simulation models: Combining automated raw data collection and automated data processing. *Proceedings - Winter Simulation Conference*, 1678–1689. <https://doi.org/10.1109/WSC.2010.5678901>
- Terrell, K. H. (n.d.). *What is a digital library? – TechTarget Definition*. WhatIs.Com. Retrieved March 7, 2023, from <https://www.techtarget.com/whatis/definition/digital-library>

Trentesaux, D., & Caillaud, E. (2020). Ethical stakes of Industry 4.0. *IFAC-PapersOnLine*, 53(2), 17002–17007.  
<https://doi.org/10.1016/J.IFACOL.2020.12.1486>

*Vad är PLC? | Elektroautomatik.* (n.d.). Retrieved February 24, 2023, from <https://elektroautomatik.se/automationsguider/vad-ar-plc/>

*Virtual Manufacturing Sweden AB | Info & Löner | Bolagsfakta.* (n.d.). Bolagsfakta. Retrieved February 23, 2023, from [https://www.bolagsfakta.se/5567172084-Virtual\\_Manufacturing\\_Sweden\\_AB](https://www.bolagsfakta.se/5567172084-Virtual_Manufacturing_Sweden_AB)

*Visual Components - Wikipedia.* (n.d.). Retrieved February 24, 2023, from [https://en.wikipedia.org/wiki/Visual\\_Components](https://en.wikipedia.org/wiki/Visual_Components)

*What is a URI? Definition and Explanation of the Identifier - IONOS.* (2020, March 23). <https://www.ionos.com/digitalguide/websites/web-development/uniform-resource-identifier-uri/>

*What is an AGV System? Why would you need one? | MasterMover.* (n.d.). Retrieved February 24, 2023, from <https://www.mastermover.com/en-us/news/what-is-an-agv-system/>

## Appendices

In this part of the document all appendices can be found.

### Appendix A

Appendix is the form complete with the answers. This survey was sent out to 6 people, where only two people answered. The survey was only sent out to employees at VM that have previous knowledge of digital tools for DES. These employees were picked by the supervisor from VM. There were 6 questions in total, but the last questions was optional in the sense that it only requested potential additional information.

- 1. Using the following options, how would you categorize the need for a standardized template to make it easier to bring in data and generate a basic layout?**

Here 1 person answered, "very useful-would save me quite some time" and the other answered "ok useful-it would be nice to have but not necessary".

- 2. What do the projects of optimizing in VC and SPS generally include/ look like? For example, do you get data from the company, or do you gather it yourself, do you create the layout from scratch every time? (This question is mostly to give us a better understanding of the problem).**

**Person 1:** The basic data (such as cycle time and nr of machines, parallell etc. Often given with a Flowchart or DWG Layout to see how to products flow in a system.) is first given and a basic model can be modelled up, then setup and failures can be discussed depending on how much knowledge they have about here own production. Buffers can be something the want to look at, How big should our buffers be?

**Person 2:** The raw data is always gathered by the company. Sometimes I have to adjust the data to fit a model for example when I need distributions. Layout as a dwg file is always provided from the customer if there is one. The layout is the model is always created from scratch.

- 3. Using the following options, which of the following data do you think is important for the template?**
  - **Number of components in the system (machines, buffers etc.)**
  - **Times (Set-up, Recovery, Cycle time etc.)**
  - **Failures (MTTR, MTBF, Availability etc.)**
  - **Buffer (Capacity, Times, Failures etc.)**
  - **If some parts of the system are linear or parallel**
  - **Possibility to differentiate between different products and their production volumes**

Here both people selected all the above.

- 4. Are there any other data that you would like to see included in the template that was not mentioned in the second question?**

Person 1: What do the costumer want out of the simulation. Name of the stations. Production plan.

Person 2: If operators are needed, what the do and the amount. Shift is applicable.

- 5. We would like to perhaps include data for AGV's in the template as well, what data would be important to have for the AGV's?**

Person 1: Speed. how many? can be something they want to know as well Routes. when orders arrive. How many orders. DWG layout. Possible priority of orders.

Person 2: Routing Logic(What happens when AGV get to close to each other) How missions are assigned(First available) Speed Charging.

**6. Are there any other things that would be good for us to know?**

No answers.

**Appendix B**

This appendix contains pictures of the real Excel template.

***Appendix B1***

The first sheet of the Excel template handles general information about the layout such as names of objects and object types. The real design of the first sheet in the Excel template is presented here.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	Object	Identif	Frame	X	Y	Z	MTBF	MTTR	AMT	Capacity	Premontor	Power Consumption	Working	Setting Up	Operational	Failed	Standby	Depreciation Period	Investment Cost	Operating Cost
2	Source	Station1	1	600	50	150	10	20	0.8	0.4	0.4	0.5	0.2	0.2	0.1	2000	15000	2000		
3	Buffer	Buffer1	1	600	100	150	11	20	0.8	0.4	0.4	0.5	0.2	0.2	0.1	2000	15000	2000		
4	Station	Station2	1	600	200	150	12	20	0.8	0.4	0.4	0.5	0.2	0.2	0.1	2000	15000	2000		
5	Station	Station3	1	600	300	150	13	20	0.8	0.4	0.4	0.5	0.2	0.2	0.1	2000	15000	2000		
6	Station	Station4	1	600	400	150	14	20	0.8	0.4	0.4	0.5	0.2	0.2	0.1	2000	15000	2000		
7	Station	Station5	1	600	500	150	15	20	0.8	0.4	0.4	0.5	0.2	0.2	0.1	2000	15000	2000		
8	Station	Station6	1	700	425	200	16	21	0.21	0.14	0.14	0.4	0.4	0.3	0.3	5000	30000	12000		
9	Station	Station7	1	700	425	200	17	21	0.24	0.18	0.18	0.4	0.4	0.4	0.4	7000	105000	14000		
10	Station	Station8	1	600	500	100	18	21	0.27	0.18	0.18	0.4	0.4	0.5	0.5	8000	120000	18000		
11	Station	Station9	1	600	500	150	19	20	0.30	0.20	0.20	0.3	0.3	0.6	0.6	9000	135000	18000		
12	Buffer	Buffer3	1	600	500	150	20	20	0.33	0.22	0.22	0.3	0.3	0.7	0.7	10000	150000	20000		
13	Station	Station10	1	700	600	150	21	20	0.36	0.26	0.26	0.3	0.3	0.8	0.8	11000	165000	20000		
14	Station	Station11	1	700	600	200	22	20	0.39	0.28	0.28	0.3	0.3	0.9	0.9	12000	180000	24000		
15	Station	Station12	1	800	600	200	23	20	0.42	0.30	0.30	0.3	0.3	1.0	1.0	13000	195000	26000		
16	Station	Station13	1	800	600	250	24	20	0.45	0.34	0.34	0.3	0.3	1.1	1.1	14000	210000	28000		
17	Buffer	Buffer4	1	500	600	300	25	20	0.48	0.34	0.34	0.3	0.3	1.2	1.2	15000	225000	30000		
18	Station	Station14	1	500	600	300	26	20	0.51	0.34	0.34	0.3	0.3	1.3	1.3	16000	240000	32000		
19	Drain	Drain1	1	500	500	300	27	20	0.54	0.34	0.34	0.3	0.3	1.4	1.4	17000	255000	34000		
20																				
21																				

Figure 30: This is the first sheet of the Excel template.

### Appendix B2

This is what the second sheet of the Excel template looks like. This sheet handles the process times.

	A	B	C	D	E	F	G	H
1	Station	Product_A	Product_B	Product_C	Product_D	Product_E	Product_F	Product_G
2	Station1	1	2	3	4	5	6	7
3	Station2	2	4	6	8	10	12	14
4	Station3	3	6	9	12	15	18	21
5	Station4	4	8	12	16	20	24	28
6	Station5	5	10	15	20	25	30	35
7	Station6	6	12	18	24	30	36	42
8	Station7	7	14	21	28	35	42	49
9	Station8	8	16	24	32	40	48	56
10	Station9	9	18	27	36	45	54	63
11	Station10	10	20	30	40	50	60	70
12	Station11	11	22	33	44	55	66	77
13	Station12	12	24	36	48	60	72	84
14								

Figure 31: This is the second sheet of the Excel template.

### Appendix B3

The third sheet of the Excel template handles batch sizes for the seven different product variants in the template.

	A	B
1	Product	Batch
2	ProductA	5
3	ProductB	10
4	ProductC	15
5	ProductD	20
6	ProductE	25
7	ProductF	30
8	ProductG	35

Figure 32: This is the third sheet of the Excel template.