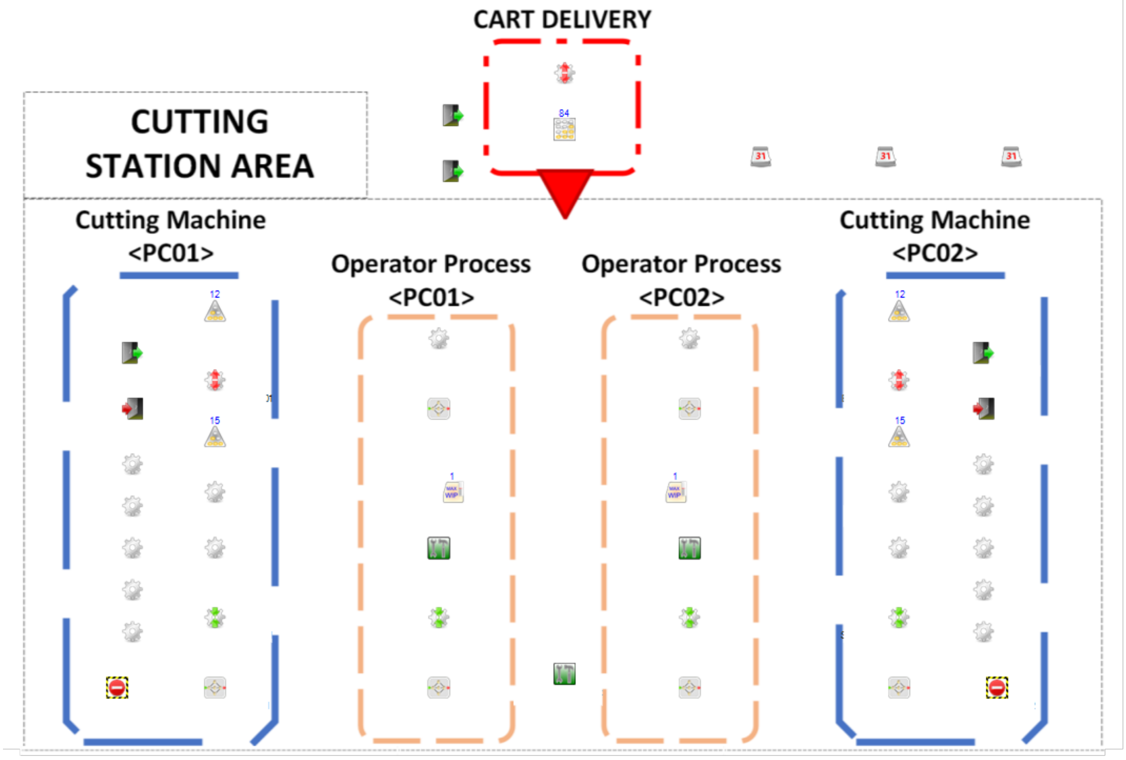

Simulation of a pneumatic tube cutting station

A case study in developing a framework for discrete event simulation model maintainability

Master's thesis in Production Engineering

Birta Hákonardóttir

Robert Hedlund

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Simulation of a pneumatic tube cutting station

A case study in developing a framework for discrete event simulation model maintainability

BIRTA HÁKONARDÓTTIR
ROBERT HEDLUND

Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
Gothenburg, Sweden 2023

Simulation of a pneumatic tube cutting station
A case study in developing a framework for discrete event simulation model maintainability
Birta Hákonardóttir & Robert Hedlund

Supervisor: Abbas Zreim, Volvo GTO
Supervisor: Viktor Bengtsson, Volvo GTO
Supervisor: Siyuan Chen, Department of Industrial and Materials Science
Examiner: Anders Skoogh, Department of Industrial and Materials Science

Master's Thesis 2023
Department of Industrial and Materials Science
Division of Production Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Image depicting the modelled cutting station in FACTS Analyzer.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Simulation of a pneumatic tube cutting station
A case study in developing a framework for discrete event simulation model maintainability
BIRTA HÁKONARDÓTTIR & ROBERT HEDLUND
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Volvo GTO is a truck manufacturing company that offers high customisation and tailor-suited vehicles towards their customers. One of their processes entails the production of pneumatic tubes that supply an array of functions within a truck. The tube production is complex and has high variation. Therefore, the capacity of the tube cutting station cannot be determined statically. In this thesis, a discrete event simulation (DES) model is built for the tube cutting station to determine its capacity. The model will also be used to explore other model parameters. However, building a discrete event simulation model is a time consuming process and therefore, often costly for companies. In some cases, a model will be used more often than once and needs to be maintained over time to be of use. In this thesis, the subject of maintainability of DES is discussed and this project is used as a case study.

Following this, two frameworks for maintainability are established. One for organisational maintainability and another for technical maintainability. The technical maintainability framework highlights internal support that can be applied to the model, while the organisational framework refers to external support that can be applied to the model to support model maintainability. The framework is established based on the results of a literature study and the authors' observations made during the process of modelling the DES model of the tube cutting station. Both frameworks consists of several criteria that should support the upkeep of the DES model long term.

Keywords: Discrete Event Simulation, Maintenance, Maintainability Framework

Acknowledgements

We would like to thank everyone at Volvo GTO for the help we got there. Particularly we want to thank our supervisors Abbas Zreim and Viktor Bengtsson, as well as Mikael Granbom for their guidance and support throughout the project. From Chalmers we would like to thank our tutor Siyuan Chen for his advise and guidance, and Anders Skoogh our examiner who provided ideas and great constructive feedback throughout the project.

Birta Hákonardóttir & Robert Hedlund, Gothenburg, June 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

DES	Discrete Event Simulation
HPL	Historical Production Log
JIS	Just-In-Sequence
JIT	Just-In-Time
ML	Machine Logs
PM	Preventive Maintenance
RM	Reactive Maintenance
SDLC	Software Development Life Cycle
SPT	Shortest Process Time
UFR	Unplanned Failure Replacement
UML	Unified Modelling Language

Contents

1

Introduction

1.1 Background

Volvo GTO is a Swedish automotive company specialising in the manufacturing and development of trucks. They offer their customers a high degree of customisation and the ability to tailor-make trucks. Recently new equipment for measuring, marking and cutting tubes for the pneumatic system of the truck was installed. In the future some changes could be made regarding the work performed at the station.

The current overall capacity of the equipment is unknown. An analysis to determine the daily capacity and the factors that affect it is highly beneficial for the company as the decisions made regarding changes to the station will be more informed. To determine the station's capacity and investigate the effect of different parameters, the process will be modelled using the FACTS Analyzer software.

Currently, the status of capacity is known to be affected by personnel, the downtime state of machines, and the variable number of tubes to be cut. But due to the dynamic nature of these parameters, it is impossible to define the capacity in a static way and a more detailed analysis of the process is required. This involves analysing relevant parameters to create a sufficiently accurate model usable by engineers within the company.

1.2 Aim

The project aims to create a Discrete Event Simulation (DES) model using FACTS Analyzer, describing the state of the pneumatic tube cutting station. Therefore, relevant parameters must be identified and evaluated to be used as input in the model. The model should predict the capacity of the tube cuttings station's current layout and the station's capacity when changes are made to different parameters. As a result, the model should be able to act as a supporting tool in decision-making regarding production setup.

The model built in this project should be sufficiently accurate whilst also being maintainable. In the future, engineers at Volvo GTO should be able to use and update the model. DES methodology has been researched (Banks, Carson, Nelson, & Nicol, 2010; Law, 2007). However, fairly little literature is available focusing on discrete event simulation model maintainability (Williams, 2002). Therefore, the project also aims to identify how a maintainable discrete event simulation model can be built, investigating internal and external support that can be applied. This is so that some of this support can be applied to the model built for Volvo GTO.

1.3 Purpose

The purpose of this project is to develop a simulation model for the tube cutting station. The model will act as decision making support for changes being made at the tube cutting station. The model will be developed to be used in the future after this project is finished. To make this more feasible, the subject of DES model maintainability will be explored from both an organisational and technical perspective and the model will be built for maintainability.

1.4 Research Questions

The goal of this project is to develop a model to explore the system's current state, including capacity predictions. The model should provide accurate results such that it is relevant enough to be of use. In addition, the concept of a maintainable model will be explored. The following questions will be answered:

- How can a simulation flow model be designed for maintainability?
- What organisational strategies can support the maintainability of a simulation flow model?

At the end of the project the model built in this project should be able to answer the question:

- What is the capacity of the tube cutting station?
- How do changes to model parameters affect the station's ability to meet the assembly line demand?

Finally, the model shall be evaluated based on the two frameworks of maintainability: Organisational maintainability and technical maintainability.

1.5 Delimitation

The project is set to a 30 credits of university studies, entailing 20 weeks of full-time studies, spanning between the dates 16th January 2023 and 5th June 2023. Therefore, some limitations will be applied to the project. Limitations and assumptions surrounding the model will be set in cooperation with different stakeholders. This includes:

- Assume a predictable and on time delivery of restocking units which resupplies the station with material for production.

- A limited amount of time will be spent on work time studies, as it is not a main focus area in the project. The process times for most worker interaction will be based on interviews, simplifications and estimations.

- Parameters in the model will not be thoroughly examined towards the model output. All parameters that are of interest will be explored as input to the model and verified that they affect the output as intended.

Further limitations regarding the model may be made in cooperation with the stakeholders of the project later if deemed necessary. It should also be addressed that the goal of this project is to do a current state analysis of the pneumatic tube cutting station and that the focus is not to identify process improvements. The project will be limited to exploring the system parameters in the model that are deemed relevant and exploring the current state capacity.

1.6 Structure

The report is structured in the following way. The first chapter details why this work has been performed. Chapter 2 lists relevant theory that has been in used to support this work. Chapter 3 describes the method that was undertaken to answer all research questions and ensure the results and the validity of its outcome. In chapter 4 the results of the study are shown. In chapter 5, a discussion is raised regarding the work and results presented in the study as well as its academic relevance. Here the sustainability aspect of the project is discussed and future research is suggested. Finally, in chapter 6 a conclusion is reached based on the results of the project.

2

Theory

2.1 Discrete Event Simulation

As defined by Banks et al. (2010), "Simulation is the imitation of the operation of a real-world process or system over time". Once a simulation model has been built and validated, it can be used to model a process and monitor its behaviour. It is a versatile tool that can be used for many different purposes (Banks et al., 2010). To do that, a model usually includes assumptions made about the operations of a system. The assumptions made regarding the system will then be expressed, usually by using objects of interest within the system or entities, as well as the connections between parts in the system. A simulation model has to accurately represent the system as it is meant to replicate its properties. However, a model will be a simplified replication of a system and a different level of detail will be required depending on the use case. The model should be sufficiently detailed so that the result of the simulation can be used to draw a conclusion regarding the system. However, the purpose of the simulation has a large impact on how a model is built, the same system can be modelled in multiple ways to answer multiple questions (Banks et al., 2010; Law, 2007).

Simulation is a flexible tool and is therefore used in a variety of industries, including manufacturing, healthcare, business processing, as well as logistics transportation, and distribution (Banks et al., 2010). Within the manufacturing industry applications includes bottleneck detection, optimising maintenance activities to improve manufacturing performance, optimising transportation systems, and layout planning. DES is commonly used within the manufacturing industry and can be used to solve these problems. DES is defined as an event-based discrete simulation, dynamic and stochastic (Banks et al., 2010; Law, 2007).

Dynamic: A dynamic simulation model represents a system that changes over time, so the passing of time has an influence on the system. On the contrary, a static simulation model represents the system at a certain time point (Banks et al., 2010).

Discrete: A discrete simulation model is when changes to the state of the model are made at discrete points of time. According to Law (2007), a system state is defined as "The collection of state variables necessary to describe the system at a particular time". In discrete event simulation events represent the time the model when the model state is calibrated again. In contrast, continu-

ous simulation models the state of a system that is changed continuously over time (Banks et al., 2010).
Stochastic: A stochastic simulation model contains random variables which yield random outputs. In discrete event simulation, different distributions can be used to trigger events such that different simulation runs may lead to different results. In contrast, a deterministic simulation model has no randomness and each event can be predetermined (Banks et al., 2010).

Software used for DES can be classified into 3 categories. The first is general-purpose programming languages such as Java, C or C++. The second is simulation programming languages and the third is simulation environments such as Automod, Anylogic or Arena. General-purpose languages are more time-consuming when modelling, but are high-level programming languages that offer more flexibility in the modelling. A simulation programming language is less time-consuming but offers limited flexibility in the modelling. Finally, a simulation environment is a specialised software for simulation. These softwares try to combine the flexibility of general languages while decreasing the time required to build the model. However, a simulation environment has to suit the problem in question as they have different strengths (Banks et al., 2010; Nance, 1996; Oscarsson & Moris, 2002).

Simulation environments can have inbuilt statistical tools to analyse the model's result, leading to more easily interpretable results. An output analyzer of the data can include summary statistics and data visualisation. Some simulation environments even support the design of experiments, sensitivity analysis and tools to support warm-up time determination (Banks et al., 2010).

2.2 Banks Simulation Methodology

DES is an effective tool to solve problems in production if used correctly. To support the development of DES models and several methodologies have shown the importance of following a systematic methodology during the development process (Bokrantz et al., 2018). The Banks model is a widely accepted methodology to assure a successful simulation project and in total consists of 12 steps (Banks et al., 2010). Figure 2.1 shows steps in the Banks methodology as well as the flow of these steps. Here, the steps of the Banks simulation methodology are described with more detail (Banks et al., 2010):

1. **Problem formulation** is the first step. In the Banks methodology it is stated that the project should start with identifying the current problem and stating it clearly. In some cases, it may prove necessary to reconsider the problem later on in the project as more information is revealed.
2. **Objective setting** includes determining whether a simulation can solve the problem stated and the objective stated. An overall project plan should be created, including a timeline for the project, the people involved in the modelling and the expected end result of the project.

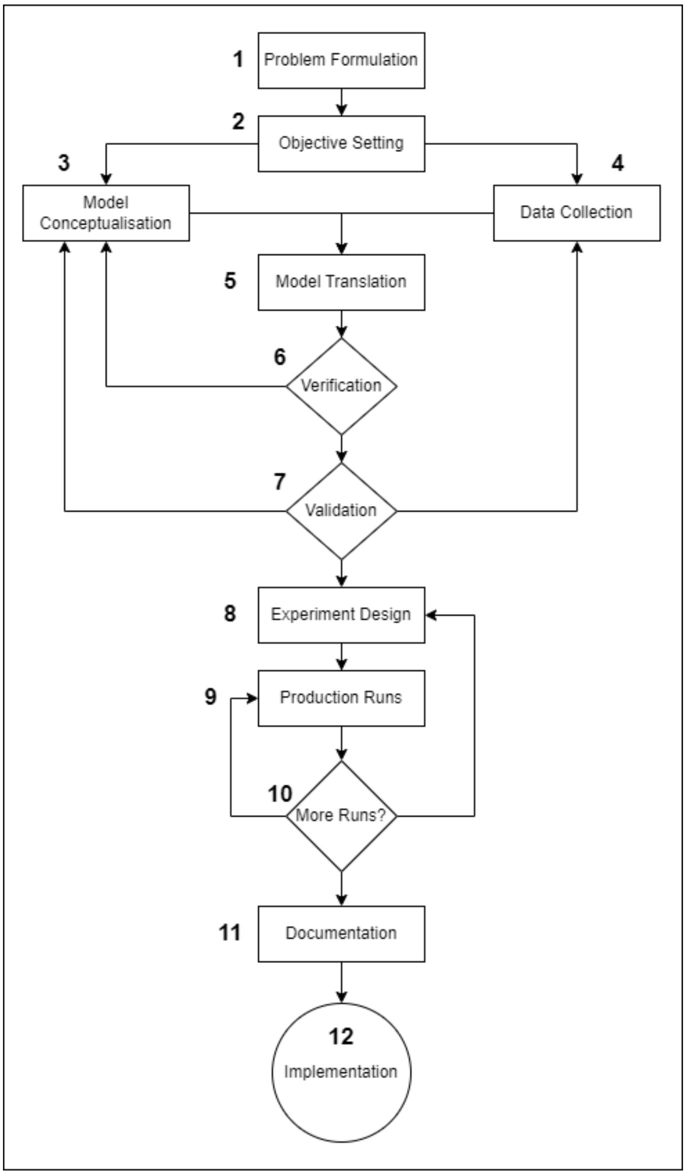

Figure 2.1: Steps in the Banks methodology (Banks et al., 2010)

3. **Model Conceptualisation** is the phase where the modeller decides how the physical system is translated to the model. This involves exploring the process and making appropriate simplifications of the system to be able to model it. A general rule when modelling a system is to avoid unnecessary details that do not increase the usefulness of the model´s results. If unnecessary features are added, it increases the size and complexity of the model. Therefore, it can lead to an increased time spent on model building and increased expenses. This means that the challenge for the modeller will be to translate the physical system into the relevant logic for the model, abstracting the essential features of the system, making assumptions regarding its behaviour, including defining the interaction between different components in the system and assumptions about the system's components. It is a good rule to start with a simple model and add detail later on to enrich the model. During the model conceptual-

isation phase, it may be beneficial to have the model user involved. Having the main user involved in the development process will lead to higher quality results in the model and the user's confidence in the model and the results of the project.

4. **Data Collection** phase is essential for accurate results from the model. It is a time consuming process and it is beneficial for the modeller to start the data collection process as early as possible (Skoogh & Johansson, 2007). The data required for the model is largely dependent upon the objective of the project, e.g., in this project the cutting times of the machines are of interest along with handling times, logistical routes and similar.
5. **Model translation** is when the conceptual model is created using some sort of computer software. The modeller must choose a software from the many different options available. In this project, the software chosen was FACTS Analyzer by EVOMA.
6. **Verification** step is when it is verified that the model function corresponds to the physical system. The structure of the model is explored to make sure the logical connections accurately represent the flow of the system. The input data also is verified to see if the data in the model provides a good representation of the process.
7. **Validation** is the process of confirming that the simulation model reflects the physical process it represents. This involves comparing the model to the physical system and making adjustments if required. The process is iterative and throughout the project, this will be repeated. This process will, if done correctly, ensure that there is no significant difference between the physical process and the model.
8. **Experimental design** In a simulation project alternative versions of processes can be created and compared. There can be multiple reasons for this, including eliminating system bottlenecks, optimising manufacturing activities or even to optimise logistical routes. In the experimental design, different alternatives are selected for comparison based on different performance measurements. The modeller also has to choose the simulation horizon, initialisation period and the number of replications. The initialisation can be used to simulate the system while in steady state i.e., in theory this is the state a system has reached when it has been run for an infinite amount of time (Law, 2007).
9. **Production runs and analysis** step involves measuring the performance of different alternatives based on the simulation runs.
10. **More runs?** The modeller has to determine if further experimentation is required and design appropriate ones if required.
11. **Documentation and reporting** of the project is important throughout the project and at the end of the project. Research has shown that documenting work performed and decisions made throughout the project can support the project. By including these regular reports the project can be kept on course and feedback regarding the project is more likely to be provided in a timely manner. Another aspect of documentation and reporting is the documentation of the final model. This is especially important if there are plans in place to use

the model further. The modeller should document decisions made, assumptions and how the model operates. With detailed documentation the model user can study the relationship between different input parameters and different output performance measurements. In addition to this, documentation of the model will increase the user's understanding of the model and therefore the credibility of the model and the model-building process.

12. **Implementation** of changes based on the model results is the final step in the Banks model. The extent of the changes depends on the result of the model, but by involving the model user throughout the development process the likelihood of vigorous implementation is increased.

2.3 Data in DES

DES is a powerful and versatile tool, however, it is a time intensive project (Johansson et al., 2003). One of the large factors contributing to this is the data gathering required to build a model (Skoogh & Johansson, 2007). In a DES project, data is a large concern. For the model to accurately represent the properties of the system, data has to be gathered and evaluated (Banks et al., 2010). This can be a time consuming process. Research shows that, on average, gathering input data consumes 31% of time in DES projects is spent on data gathering (Skoogh & Johansson, 2007). Not only is the data gathering and evaluation process time consuming. High quality data is required to build an accurate DES model. In addition to data being required to build the model, Robinson and Bhatia (1995) argue that data is essential for validating simulation models.

2.3.1 Data Gathering

In the data gathering stage of a DES project the most common pitfall is poor data availability (Perera & Liyanage, 2000). Research has shown that only 6-7% of simulation projects have had all the required input data at the start of a project (Skoogh & Johansson, 2007; Johansson et al., 2003). The availability of data and data collection can have a big effect on a DES project, as data availability has to be considered when building the model and may affect how the model is built.

According to Robinson and Bhatia (1995), data can be divided into three categories based on if the data can be collected and the data availability. Category A is available data, meaning data that is readily accessible in different formats. Category B is not available data but collectable. Finally, C is non-available nor collectable data. Data belonging to category C is the hardest to obtain and must be based on assumptions and estimations. However, this also raises issues regarding the validity of the data and therefore the validity of the model. A useful technique to manage this issue is to perform a sensitivity analysis on the data in question and evaluate the sensitivity of the model output (Robinson & Bhatia, 1995).

In some cases, data that is critical in the model may be unavailable. Then, there are a number of ways to estimate that data. One way is through expert opinion, that is to talk to experts in the area. They can help describe the nature of the data, such that, if process times are being examined they may be able to describe the data variation and provide an estimate of the process times. Another way to estimate unavailable data is to consider the nature of the process, e.g., Weibull distribution is often used to model component failure. In some cases, physical or conventional limitations can be used to determine the limits of data. For example, an automated guided vehicle cannot travel faster than its programmed maximum speed. Finally, engineering data can be used. Frequently product or machine manufacturers provide information about the project that can be used to estimate data (Banks et al., 2010).

2.3.2 Data Quality

In a DES project, the quality of input data is of high importance. Low quality data can lead to misinformation and in a worst case scenarios it may lead to the wrong decision being made (Cai & Zhu, 2015; Mahanti, 2019; Han, Pei, & Tong, 2011). A framework proposed by Cai and Zhu (2015) defines five different data quality dimensions: Availability, usability, reliability, relevance, and presentation quality. The dimensions in the framework address commonly accepted and frequently encountered data quality issues.

> **Availability** refers to three different elements: authorisation, accessibility and timeliness. Availability addresses whether the data is accessible to the individual or organisation in a timely manner and that the individual or organisation has the right to use the data (Cai & Zhu, 2015).
> **Usability** can be divided into three different elements, definition/documentation, credibility and MetaData. This refers to whether the data is defined or documented, with a variable name for example. MetaData element refers to the risk of the data being misinterpreted due to a lack of knowledge regarding the dataset. The credibility element evaluates whether the data falls within an acceptable range, it has been checked for correctness by experts, and regularly audited by specialists. The credibility element also considers the source of the data (Cai & Zhu, 2015).
> **Reliability** is about making sure the data is trustworthy for the cause (Mahanti, 2019). The reliability dimension contains five different elements: accuracy, integrity, consistency, completeness and auditability. The accuracy element refers to whether the data is accurate and if it is a correct representation of the process. Consistency refers to the data being equal across different platforms, so even if it is located in different storage areas it should be equal in value and meaning. The integrity element refers to the data structure being complete. Data values should be standardised and characters in the data correct. The completeness element refers to the data set including all the relevant information. A complete data set will include all values of all components. The final element in the reliability dimension is auditability, which refers to whether the dataset can be validated in regards to integrity and accuracy within a reasonable timeframe (Cai & Zhu, 2015).

The Relevance dimension includes one element, fitness. This refers to how well the user demand and the data content correspond. The data needs to be relevant in relation to the purpose it is used for and contain enough width to be appropriate (Cai & Zhu, 2015; Mahanti, 2019).
Presentation quality dimension includes two elements: readability and structure. Readability refers to the structure of the data if the content and format are understandable and clear to the user. The data can then be explained in a clear manner using known information, attributes units, codes, or well defined terms. The structure elements refer to the state of the dataset, whether it is unstructured, semi-structured or structured (Cai & Zhu, 2015).

In the literature study several different data quality dimensions have been identified (Mahanti, 2019; Han et al., 2011). However for the purpose of this thesis, the above mentioned framework proposed by Cai and Zhu (2015) is used. By using this predefined framework for data quality issues with data can be identified early on. Based on a data quality analysis decisions regarding the data gathering process may be made early and steer the project towards better data sources. This will also increase time efficiency during the data gathering process.

2.3.3 Data Pre-processing

When large quantities of data are used there are multiple issues that can arise with the data, e.g., the data can have missing values, noise, or be inconsistent. Sometimes it may be necessary to combine multiple data sources for a project and then rules may be required to filter through the data so that no issues arise. Before data can be used in the project the data quality has to be assessed and data pre-processing performed if necessary (Han et al., 2011).

There are several methods that can be utilised in this case, including data cleaning, data integration, data reduction and data transformation. Data cleaning is the process of correcting inconsistencies and removing noise from data. Data integration is the process of combining data from multiple sources into one coherent data store. Data transformation or normalisation is when data is scaled to match a certain range. This may be done with a data set consisting of distance measurements to improve the efficiency and accuracy of the mining algorithm. Data reduction is when the size of the data collection is reduced through methods such as clustering, eliminating redundant features or aggregation. These methods can be used on the same data collection (Han et al., 2011).

2.3.4 Exploratory Data Analysis

When data has been collected about a process relevant information can often be extracted from that data. Exploratory data analysis involves exploring data from various perspectives to discover patterns that can provide further information. This will uncover previously unknown insights into the data. Several methods can be used in this process, including but not limited to, graphic visualisation, and looking into summary statistics and different data analysis tools (Morgenthaler, 2009).

2.3.5 Distributions

In reality, processes can rarely be represented as a constant and therefore a simulation analyst may need to analyse the input data to determine an appropriate distribution for the process. This is especially true in the case of complex processes. In a simulation project this is an important fact to keep in mind. When determining the distribution for a process there are several things to keep in mind, including the nature of the process being analysed and the shape and size of the probability density function. Considering the nature of the process is important because some distributions are a better fit for certain processes or activities. Then to get further information a density plot can be graphed and compared to the shape of different distributions (Banks et al., 2010).

2.4 Verification

Verification of a simulation model is according to Law (2007) making sure that the model is run on reasonable data and assumptions. A model needs to be accurately debugged such that the logic of each connection and processes in a model is properly represented (Law, 2007; Banks et al., 2010). The model should run in a way that entities are flowing through the system in the intended way (Banks et al., 2010). Law (2007) lists eight techniques for verification, which are summarised here below:

1. Debug continuously for each implemented logic.
2. Allow multiple developers to review the logic.
3. Experiment with different settings and see if they change in a reasonable manner.
4. Use trace.
5. In instances where it is possible, use high aggregation of details to cover accurate processes.
6. Utilise animation to follow the flow of entities.
7. Compare the input distributed data with historical data distribution. Different programs handle variations differently.
8. Utilise available features presented by the modelling software to reduce instances of logic error.

2.5 Validation

According to Law (2007), validation is to confirm that the model represents the process which it is trying to simulate with sufficient accuracy. Law also claims that validation of a complex system can never fully be validated and can only be considered as an approximation at best. This is because the uncertainty caused by a lack of measurements or available data makes validation impossible. The output data of the model should to the highest extent be compared to what is measurable in reality to determine its accuracy (Law, 2007). The process of validating a model is iterative, where the model is compared to the real system, discrepancies identified and corrections made in accordance to those discrepancies (Banks et al., 2010).

Many different validation frameworks have been developed (Sargent, 2010). There can be multiple stages of validation in a DES project. This relates to the model itself, logical assumptions made during the project and the data used (Sargent, 2010, 1981).

A simplified approach for DES model validation was suggested by Sargent (1981) where three different categories of validation were identified: conceptual model validation, data validation, and operational validation. Throughout a simulation project necessary data is collected, and the process of evaluating whether the data in question is adequate and correct is called data validity. The process of determining whether assumptions made about the system and theories are correct is called technical validation. When a conceptual model of the system has been developed the technical validity is determined and adjustments are made if necessary. When the model has been built and verified the results of the model have to be validated, this is called operational validation. Operational validation is the process of ensuring that the simulation model is built with sufficient accuracy so that it fulfils the purpose of the simulation (Sargent, 1981).

2.6 Maintainability

As discussed in section 2.3, the process of building a DES model is time consuming and therefore a DES model can be expensive for a company to build (Johansson et al., 2003). In some cases, DES models are built to be used frequently over a long period. During this period many different modellers may be involved in updating the model and therefore some considerations need to be made when the model is developed (Williams, 2002). Maintainability is also something to consider when a DES model is made for a system that will undergo changes. If the exact changes are unknown when the model is being built considerations need to be made, so that the model can be updated with the system.

Definitions of maintainability vary. From a software perspective, maintainability is defined as "The ease of which a software system or component can be modified or corrected faults, improve performance or other attributes, or adapt to changed environment" (“IEEE Standard Glossary of Software Engineering Terminology”, 1990). From a hardware perspective it is defined as how easy it is to restore a physical product to its previous or original state and performance (“IEEE Standard Glossary of Software Engineering Terminology”, 1990). Based on this maintainability of DES model will be defined as the ease of which the model can be corrected, new features added or performance improved and how the model can adapt to changes made to the physical system in this project.

Maintenance regarding the upkeep of machines is usually divided into two different alternatives: Reactive maintenance (RM) and Proactive maintenance. Unplanned failure replacement (UFR) is one common RM strategy where no action is taken until the functions of the machine fail. The machine is then brought back to an operational state through repair, but there exists no official strategy to repair. Proactive maintenance, however, aim to lie a step a head of eventual breakdowns in by different strategies. A common Proactive maintenance strategy is Preventive maintenance (PM). PM can consist of a scheduled regular repair of a machine, and replacement of parts and components (Bidanda, 2023).

There are several advantages to building a DES model with maintainability in mind. A model built for maintainability has a lower cost of maintaining the model and the model is less likely to be misused leading to wrong results from the model. Not to mention that a model with high maintainability can be more user friendly, leading to a higher utilisation of the model (Williams, 2002).

To support maintainability in a DES model internal and external support can be applied. Internal support refers to support within the model or technical support, while external refers to support applied to the model from the outside or organisational support (Williams, 2002).

2.6.1 Organisational Support

External support for a maintainable model can be in the form of documentation and a clear structure within the company for maintaining the model. Documenting the source of data in the model is an important step. This includes the origin of the data, whether it is calculated, based on data sheets, information from the supplier, machine logs, time studies or any other source of information. In some cases statistical distribution software may be used to determine the data distribution and the details of that process should be documented as well. This includes the software used and details regarding the methods applied and the software's settings. All assumptions or simplifications of the process should also be documented and reasoned (Williams, 2002; Oscarsson & Moris, 2002).

For simulation environments, attempts have been made to establish a documentation standard. Oscarsson and Moris (2002) highlighted that documentation can be both internal and external. Documentation can be within the model, e.g., in the form of comments. External documentation is different and can be in the form of a 3D animation, a guide of the model or as a conceptual model. It is also highlighted that it can be beneficial to have different types of documentation depending on the type of audience. People involved at different levels of the model will require an understanding at different levels and when documenting models they should be documented at different levels. Three different levels of documentation are mentioned; low-level documentation for programmers, documentation for conceptual documentation, and finally when describing the system for people with no interest in details. For low-level documentation comments in code and flow charts of the process are preferred. If further detail is required, an Unified Modelling Language (UML) diagram can be used. Documentation of the conceptual model using Integration Definition 0 (IDEF0) was recommended. When more detail is required IDEF0 in combination with a flowchart can be used. Finally documenting the model for people with no interest in the model's detail a 3D simulation is ideal. Then the process can be visualised and understood easily due to the dynamic visualisation.

Furthermore, Sargent (2010) proposes an eight step process for validating a model. In the final step he proposes that a model that is intended to be used over a time horizon, there should be an established process to routinely review the validity of the model. Sargent goes on to claim that there is no specific standard way of scheduling what to do or when to do it, it is all case by case.

2.6.2 Technical Support

The internal support for a maintainable model can be applied early on in the development process. Decisions made regarding the software used in the project and the structure of a model can have an impact on the maintainability of the model. For each modelling project, the software used has to be chosen carefully. This is partially so that during the model-building phase the modeller can use constructs built within the software instead of creating unique ones themselves. Creating unique constructs within the model increases complexity and makes the model harder to understand. A good practice for modellers is to reuse constructs and templates that have been previously verified. This decreases the time spent modelling and verifying the model. A model reusing constructs or templates is also easier to maintain (Williams, 2002).

Complex models that contain more details are considerably more difficult to maintain (Piplani & Pauh, 2004). Brooks and Tobias (2000) defined simplification in DES projects as reducing the number of connections or components within the model. However, when a model is simplified, the nature of the simplification has to be considered, and to what extent it will affect the accuracy of the model (Brooks & Tobias, 2000).

Another method of simplifying a model to support maintainability was discussed by Piplani and Pauh (2004), where the production of semiconductors was simulated. Two different methods were suggested, to aggregate or to condense the model. Aggregating refers to the use of product families, while condensing refers to reducing the number of elements in the model. The elements that can be reduced are identified by looking at the throughput, where high throughput components are deemed to have insignificant effect. This leads to fewer events in the simulation and faster runs.

Good modelling practices can provide internal support. For example, the use of embedded constants should be avoided. Especially in the case of variables that are of interest to the model user. These variables should be changeable by the user such that the effects of the change can be explored. The user of the model should be able to interact with the variables of interest in an easy way, for example by changing the variables on a spreadsheet connected to the model. The use of a spreadsheet also reduces the risk of input errors in the model, and reduces the maintenance time of the model and the time it takes to generate different scenarios within the model. Clear comments in the model can also help later when updates are required. Throughout the code information about constants, formulas and functions can be helpful for the user later on and will decrease the time spent on updating the model (Williams, 2002)

2.6.2.1 Software Maintenance and Maintainability Index

In software development, the issue of maintainability is an important consideration to make. Companies aim to maintain software in a cost efficient manner and a maintainable software will facilitate this aim. Schnappinger et al. (2020) researched expert judgement on maintainable software and what will make a code less maintainable. Although expert opinions vary and are to some extent based on personal preference results show that understandability of the code is a key feature.

Software development life cycle (SDLC) is a widely accepted framework in the development of software. SDLC consists of several steps whereof software maintenance is the last step (Gupta & Gayathri, 2022). According to both Christa et al. (2017) and Heričko and Šumak (2023), 70 % of resources and efforts are spent on maintenance activities considering a software project. However, software maintenance is broadly defined and consists of all actions undergone post product release to keep up the relevance of the software (Christa et al., 2017)

In their article, Heričko and Šumak (2023) compare different software maintainability evaluations in object-oriented systems. Maintainability index is a concept in software development described by a quantitative single number rating of the work-ability, improvement capability and changeability of the software. The value ranges between 0 and 100, where 100 indicates an exceptionally well maintainable foundation. The maintainability index is measured by four dimensions of software metrics relating to cognitive demand on maintaining the code. However, Heričko and Šumak go on to claim that there are different approaches for establishing the maintainability index, and depending on the application, different values are received. The following four metrics are usually considered for the maintainability index:

Cyclomatic Complexity: Measures the number of different and independent paths that is held by the code. For each instance that the code is divided in flows, it adds complexity to the system (Ebert et al., 2016). IBM (n.d.) provides some examples of this such as every time a branch appears in the code, e.g., IF-statements and other logical operands, additional complexity is added to the software.

Computational Complexity: Denoting the mental requirement on a continuation of upkeep of the software (Heričko & Šumak, 2023).

Software Size: Measures the grand size of the entire programming project in terms of content in the code, e.g., number of lines of code or statements

(Heričko & Šumak, 2023).
Code Documentation: Measures the relation between the amount of comment compared to the amount of code (Heričko & Šumak, 2023)

3

Method

This chapter describes the method that was used to reach the outcome of the project. In figure 3.1 the a structure of the tasks performed in the project are depicted.

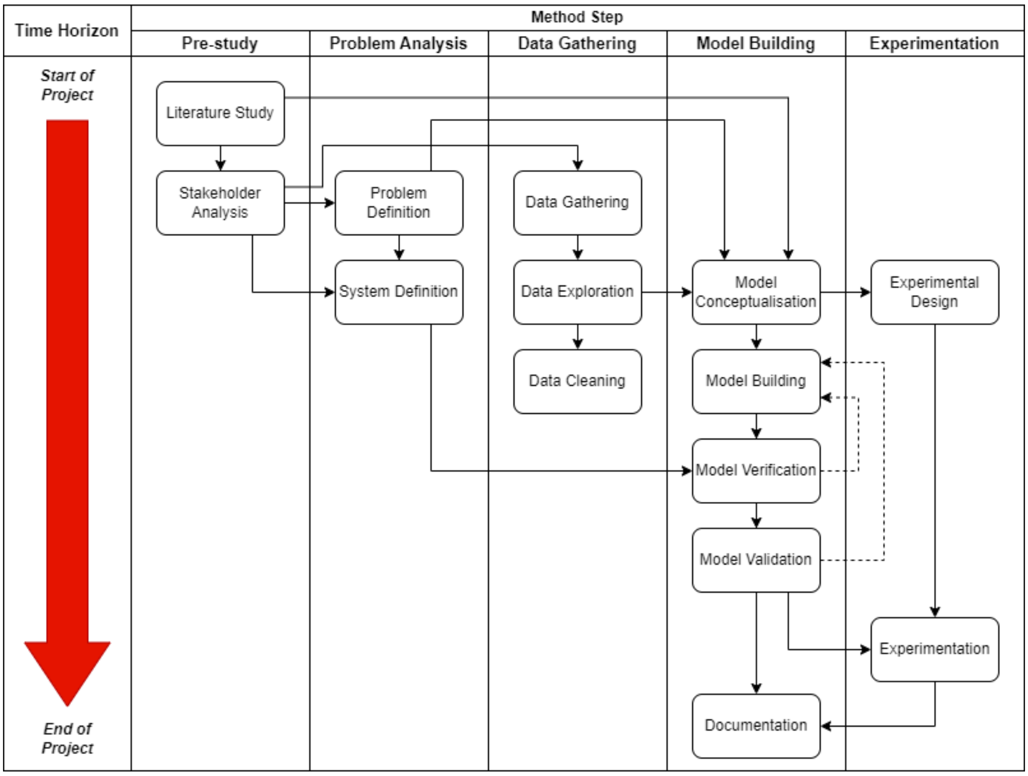

Figure 3.1: The methodology carried out in this project displayed in chronological order

During the project, step one through eleven of the Banks simulation project methodology was performed (Banks et al., 2010). As discussed in section 2.2, the Banks simulation methodology is repetitive and not a linear process (Banks et al., 2010). Therefore, that element was also included in this simulation project. In some ways, the methodology in this project did deviate from the Banks model. For example, to answer the research question a literature study was conducted, not only at the beginning of the project but rather the literature study was frequently revisited to support building a maintainable model.

3.1 Stakeholder Analysis

A stakeholder analysis can be used to identify who is interested in the outcome of the project, as well as who may have an effect on the project (Bryson, 2004).There are many different definitions of a stakeholder in literature, but they all have a common point: The stakeholder is someone who can affect an organisation's strategy. Some definitions are broader and will also consider stakeholders to be individuals or groups of individuals that will be affected by an organisation (Wang, Ge, & Lu, 2012; Bryson, 2004).

To facilitate the success of a project, identifying stakeholders can be the key. The literature claims that the support of a stakeholder can be crucial to ensure a project's success (Jepsen & Eskerod, 2009; Bryson, 2004). According to Bryson (2004), a stakeholder analysis is beneficial in a project's early phases as a tool for finding information from key stakeholders. Failure to attend to stakeholders' needs or utilising of resources may result in a project failing or yielding poor results (Bryson, 2004).

Many different guidelines regarding how a stakeholder analysis can be conducted have been researched. In most, the first step is to identify all stakeholders in the project. There are several challenges present in the stakeholder analysis, including: identifying stakeholders and determining their importance to the project and identifying their expectations (Jepsen & Eskerod, 2009). According to Wang et al. (2012) there is no one methodology for a stakeholder analysis, but commonly consists of the three following steps:

1. Identifying stakeholders
2. Categorising different stakeholders
3. Mapping the relationship of different stakeholders

There are no guidelines on how far the stakeholder analysis method should branch. This is an evaluation that has to be conducted for each case by the one who performs the analysis (Bryson, 2004).

At the start of the project it was unclear how precisely the model should be constructed to answer relevant questions. Different departments within the company had an interest in the outcome of the project and therefore it was deemed important to identify different expectations towards the project. It was also unclear who could aid in the project. To gain further insight into the problem and the system, a stakeholder analysis was conducted. The analysis was conducted in the early phases of the project to gain a better understanding of who or what may be affected by the project.

The first step in the stakeholder analysis carried out in this project was to identify stakeholders based on the scope of the project and aims. This includes both people that have an interest in the project or its outcome, as well as people who can in some way affect the outcome of the project. Since the group of the identified stakeholders was small and the students unfamiliar with the structure of the company, it was deemed beneficial to interview all of the stakeholders.

Each interview was semi-structured and the interview questions focused on identifying different expectations of the project and what or who might affect or be affected by the project. In the interview, the interviewees were also questioned about whether they knew of anyone else who might be interested in the project or anyone who may have an impact on the project. The goal with this questions was to identify if any stakeholders had been missed in the initial assessment and if so locate those stakeholders. Table 3.1 shows the stakeholders identified in this analysis, divided by whether they have an interest in the outcome of the project or if they can influence the outcome of the project. In some cases, the identified stakeholders consist of entire departments within the company. In such cases, it was deemed sufficiently accurate to interview a representative of these departments.

Table 3.1: Stakeholders in the project

Outcome/Result stakeholders	**Process/realisations stakeholders**
Company - Volvo GTO: Interest in gaining further knowledge about the production system	Chalmers examiner
User: Interest in having a model for decision support.	Chalmers tutor
Developer: Interest in having a model for decision support.	Supervisor
Logistical department at Volvo GTO: Interested in the investigating the effect of the logistics flow	Workers at Volvo trucks
Students	Students

The stakeholder interviews provided a deeper understanding of the problem presented in this project and a better understanding of the processes. Each stakeholder could communicate what he or she expected from the project. In addition, the stakeholder analysis helped identify ways to support the project, both in regards to a more in-depth knowledge of the process and access to data for the model.

3.2 Problem Definition

Based partly on the information obtained during the stakeholder analysis, a detailed problem definition was established. The main goal of the project was to provide a model of the pneumatic tube cutting station that reflects the process and considers relevant parameters. The model should be easily understandable by the user, such that the detail level of the model had to be chosen with care.

One of the main challenges in modelling this process is the variance of the tubes in production. The number of tubes vary greatly between product variants and vehicle specifications, where delivery locations for product variants, the number of tubes, as well as each tube length and diameter vary.

The model should act as a decision-making support when exploring changes to the station. Therefore, the output of the model should be communicated in a clear way that can be understood. A commonly used unit within the company is trucks manufactured per shift and it would be ideal if the model could report results using those units.

Based on the stakeholder interviews a few different scenarios of the model use were construed. Ideas about the number of carts in circulation, the number of chassis numbers per cart, batch sizes, changes to the logistical flow and product demand were of particular interest. Therefore the model should be built with sufficient detail so that these factors can be further explored in the future. Another aspect highlighted by the stakeholders was maintainability. Since the machines are new and changes to the station are ongoing the model should be built so that these future changes can be more easily implemented in the model.

3.3 System Definition

The pneumatic tube production is a mix of sequencing and batching. Sequenced deliveries are more frequent, and the tubes are delivered in carts to 17 different delivery locations within the production. However, not every truck will require tubes at each delivery location, this depends on the specification for each truck. The batched products, providing continuous supply, are also delivered to various locations on the production line, but the deliveries are less frequent and a small number of the total cut tubes. The production line uses a two-bin system for both the sequenced tube deliveries and the batched ones.

The tube cutting process can be initiated in two ways. Either through scanning a bar code on a cart or by manually selects delivery location on a computer connected to the machine. The machine will fetch information about the tubes required at each location and starts cutting the tubes. In the case of sequenced tube cutting, information about vehicle specifications is utilised to produce for each delivery location according to the requirements of the vehicles. For each tube produced, the machine measures, prints information and cuts. The machine cuts one tube at a time, but batches tubes in bundles and later deliver several tubes at the same time to the worker. This is depending on the use-point within the delivery location's assembly. The machine will switch between tube diameters based on the orders and can deliver several different tubes with different lengths and diameters at the same time. Additionally, the machine has the option of optimising cut sequence to minimise the diameter change time. When the tubes are delivered to the worker they are tied together using a cable tie with an identifying label attached. The tubes are placed in a cart in the correct sequence slot.

The production for carts is made similar to Just-in-Time (JIT) and Just-in-Sequence (JIS) pull production. JIT aims to deliver material right as it is needed to reduce inventory. JIS is further specified to the delivery of the material that is to be used in a specific order in the assembly, both at the right time, and in the correct and sorted order for assembly (Boysen et al., 2015). In this case, the material for each cart is placed in slots of each cart that determines the assembly order. Each cart is assigned to a specific delivery location and depending on the location different logistical routes are used. Here, five different delivery methods are used: delivered by tube cutting station workers, delivered by workers at the receiving station, towed to the delivery location, carried by forklift, and the most common delivery method, via tug-train.

Workers at the tube cutting station are also responsible for some maintenance activities, including cleaning the equipment and replacing tube coils. During cleaning, the machine is not operational. Additionally, the machine can have downtime if a coil of tubes runs out and must be replaced. If the coil is changed without consuming the buffer length of tube, downtime is avoided. The number of tubes, length and diameter vary depending upon where they are used and the specifications of each vehicle. The length of the tubes is between 110 mm and 6000 mm and have five different diameters. As such, the cycle time of the machine varies, even though the cord feeding is constant, the feeding speed depends on the diameter of tube and the length of each tube varies. Since other activities are also performed by the workers assigned to this station, the effect on those activities will be investigated and determined how this affects the machine capacity.

Furthermore, the tube cutting station that is the focus of this thesis, is just one supporting an array of processes relating to the truck production. The tubes are either delivered directly to the assembly line or to another sub-assembly station in which the consumption of pneumatic tubes is done ahead of time of the actual truck assembly.

3.4 Model Detail Level

In a simulation project the level of detail in the model is very important. The modeller should model the system with enough detail so as to get a sufficiently accurate result that answers questions posed. However, adding unnecessary details that do not increase the usefulness of the model or the models results, only increases the amount of work required to build the model (Banks et al., 2010).

In this project the level of detail was hard to determine due to the large variety of tubes in truck variants. An overview of the detail levels discussed can be seen in table 3.2. Another challenge was data availability and determining process time distributions with such large variety. A requirement by the stakeholders was that the model result should return an easily readable and understood metric. Results should matching common used metrics within the company, trucks per shift. There was also interest in exploring tube deliveries to specific deliver locations on the production line. Taking a few factors that were deemed important into consideration four different detail levels were explored.

Table 3.2: Comparison of detail levels

	Highest detail level Tube level	**Medium detail level Bundle level**	**Medium detail level Chassis level**	**Lowest detail level Cart level**
Communicate result	Complicated hard to read model	Hard to read results	Results can easily be communicated	Results can easily be communicated
Model Complexity	Very high	High	Medium	Low
Accuracy	Accounts for worker interactions. Accounts for variety of the delivery locations	Accounts for variety of the delivery and worker interaction with the machine	Accounts for variety of the delivery locations but loses some details regarding worker interactions	Accounts for variety of the delivery locations but loses some details regarding worker interactions
Maintainability	Very hard to maintain when any changes are made to the product	Easier to maintain when changes are made to the product	Hard to maintain when changes are made to the product	Very hard to maintain when any changes are made to the product
Data access	Good	Good	Good	Decent
Validation	Very hard to validate data and results	Data and conceptual model can be partially validated	Data and conceptual model can be partially validated	Data and conceptual model can be partially validated

Based on this analysis, a combination of bundle and chassis detail level was chosen, where tube bundles are processed at the tube cutting station for the production of chassis. By including bundles, the product variety can be considered with greater detail as well as the worker interaction at the cutting machines. Other interactions with the tubes in the model are on a chassis level. By building the assembly line at chassis level the model will result in a readable outcome, chassis per shift.

3.4.1 Conceptual Model

As mentioned in section 3.4 the lowest detail level in the model is at bundle level. When the machine cuts tubes, bundles of tubes are created and subsequently the operator gathers the bundled tubes. However, as in the physical process tubes are placed in carts and transported to delivery locations throughout the plant. Figure 3.2 shows a conceptual model for the simulation model. The conceptual model only shows the flow of a cart through one cutting machine, but there exist two.

The process starts when an empty cart arrives at the tube cutting station. The carts in the model are created to be delivered to a specific specific delivery location at the assembly line. If one of the two cutting machines is available, the cart is taken by an operator to be loaded on that station. This involves placing the cart by the machine and scanning a bar code on the cart or choosing the correct delivery location at the computer. At the cutting stations tubes are cut and delivered in bundles to the operator, the number of bundles and the tubes in the bundle are determined by the design of the truck it matches. When a bundle has been cut it is placed in a cart by the operator. Bundles are placed in a specific order in the cart, such that they match the truck they will be assembled onto later. When all bundles have been cut and placed in the cart by the operator the cart is delivered by the operator to a buffer at the cutting station.

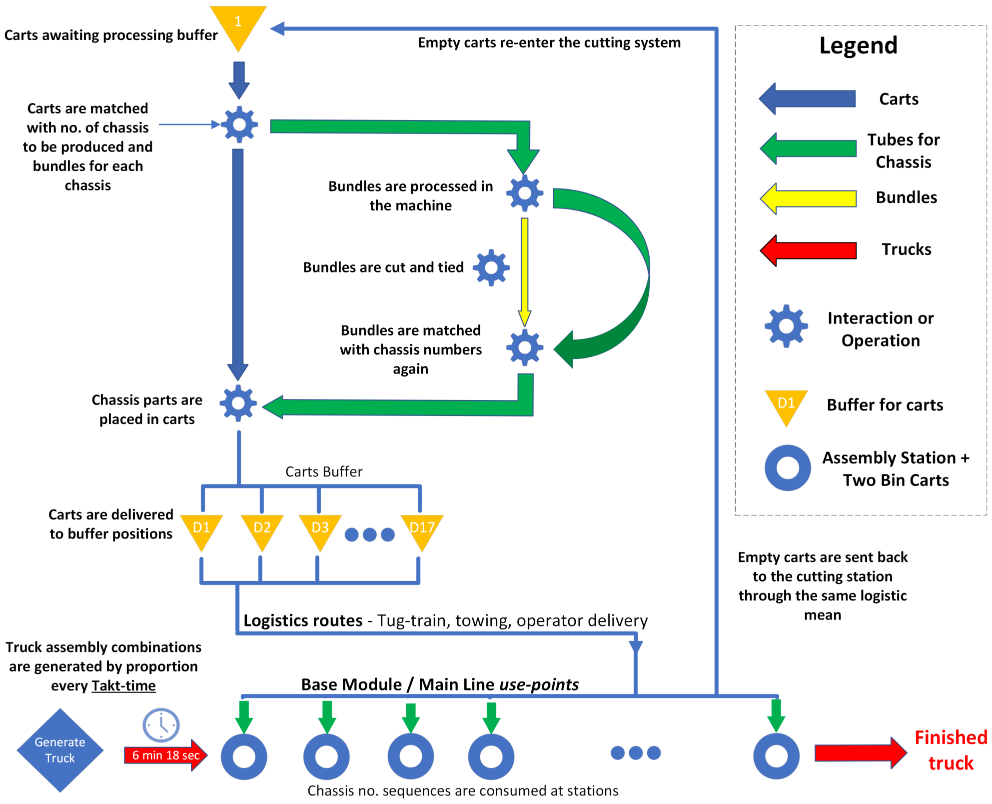

Figure 3.2: Conceptual model of the product flow

When a cart runs out at the assembly line, it is delivered to the correct delivery location at the assembly line via the corresponding logistical routes for that delivery location. Upon arrival at the correct delivery location, the cart is placed in a buffer. When the cart at the delivery location has been consumed a new cart is delivered from the delivery location buffer and the empty cart is circulated back to the cutting machine station through the same logistical route as before.

3.5 Model Design

In this section, the model building is described in its most essential parts. The model is based on the system definition mentioned in section 3.3 and the detail level chosen in section 3.4.

3.5.1 Model Variants

For a simulation model, the right level of detail is essential. Too low detail and it gets insufficiently accurate. On the opposite, too high detail makes the model partly too complex and slower to run (Banks et al., 2010). As previously discussed in section 3.4, it was deemed reasonable to model the process at a bundle level. In reality, a bundle consists of one to several pneumatic tubes.

The tubes may be of different length and diameter. One of the main reason for modeling this process at the bundle level is such that the worker interaction can be included in the process. Since a worker is required to pick tube bundles from the machine, if a bundle is not picked, tied together and placed in the cart, the machine will stop. Moreover, trucks are also modeled as variants that pass through the assembly line. In table 3.3, all levels of variants used in the model and the variant composition can be seen.

Table 3.3: Model variants level representation, model name and variant composition

Variant level name	**Model name**	**Consists of**
Bundle per delivery location	Bundle_D	–
Sequence per delivery location & bundle amount	Sequence_D_A	1-12 x Bundle_D
Sequence per delivery location	Sequence_D	1 x Sequence_D_A
Cart per delivery location	Cart_D	6/12 x Sequence_D
Truck assembly combination	Truck_B	4-14 x Sequence_D
Batched production carrier	Batch_C	–
Bundle per batch line number	Bundle Batch_C	1-3 x Bundle_Batch_C

The model exists on three different assembly levels. The smallest level is the bundle variants, which make up material for both sequence and batching production. A sequence describes the material that is used for one chassis at a delivery location and would in reality correspond to the tubes in slot in the cart using JIS. The batching container is the end product in the model for its use point. However, an amount of sequence variants is further assembled onto a cart for the same delivery location. A truck variant consumes sequences from the cart later down the line.

A denotation of the variable ranges used in the model can be viewed in table 3.4. *A* denotes the number of bundles that belongs to the production of one chassis per delivery location. *B* denotes an indexed truck variant out of the total different realistic combinations. *C* is an indexed batching number and *D* is the particular delivery location that a variant belongs to.

Table 3.4: Denotations in the variant naming convention and their ranges

Variable	**Model representation**	**Value range**
A	Number of bundles	1-12
B	Truck variant number	1-214
C	Batching number	1-5
D	Delivery location	D1-D17

The amount of bundles per chassis number and delivery location were based on historical data and the distribution of the number of bundles per sequence and delivery location. However, the number of bundles is not connected to a specific chassis and is therefore converted into a generic chassis per delivery location sequences for that delivery location. Subsequently, the generic chassis number per delivery location sequences are assembled onto a cart for the same delivery location that can hold a specific amount of tubes per chassis, which corresponds to the amount of chassis per that delivery location. Details about the number of carts for each delivery location and the number of chassis number placed in a cart can be viewed in table 3.5. Most of the delivery locations has four carts, two in the two-bin system at the assembly and two in cutting station buffer.

Table 3.5: Number of carts per delivery location and number of chassis per cart

Delivery location	No. of carts	No. of chassis per cart
D1	4	12
D2	4	6
D3	4	6
D4	4	12
D5	4	12
D6	4	12
D7	4	12
D8	4	12
D9	4	12
D10	4	6
D11	3	6
D12	4	6
D13	1	1
D14	3	3
D15	4	12
D16	4	12
D17	4	6

3.5.2 Model Flow

The model is constructed as to represent the circulation of carts within the plant. Carts can contain different amounts of sequences depending on the delivery location, but constant within each delivery location (see table 3.5. These carts are circulated from the station to the production line through logistics, depleted, and once more sent back to the cutting station. Each cart can contain a predefined number of sequenced material which later is assembled onto trucks. In figure 3.3, the entire model can be viewed. A larger image of the entire model can also be viewed in abstract A.1. Additional to the sequenced production, batched tubes exist in the model and are handled in the same way carts within the tube cutting station.

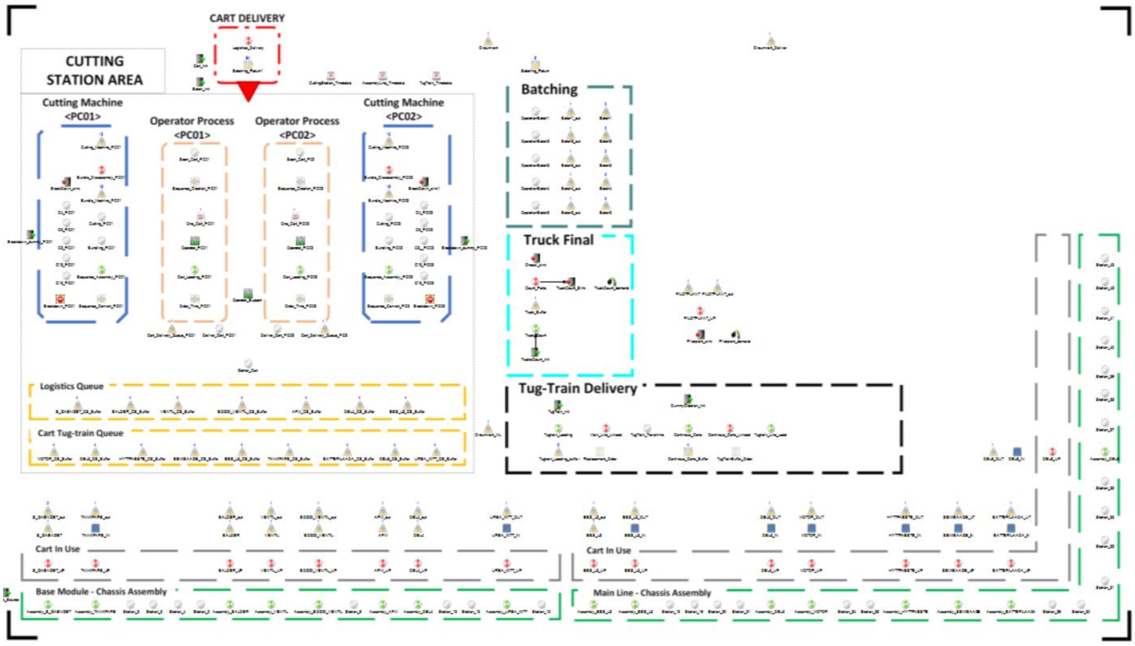

Figure 3.3: A full view of the model in FACTS Analyzer

Operators at the cutting machine were modelled as individual resources, one for each machine and one role for the supporting operator. Furthermore, the model was divided into four separate module areas that were deemed to require representation based on their relevance in reality. These module areas are highlighted in the model through a constructed background image. In the following subsections, each area is described in its purpose and functionality. The areas are listed in order of production sequence and life cycle from that of an empty cart at the cutting station.

3.5.2.1 Cutting Station

The empty cart or batching carrier enters a buffer once dropped off by logistics or fetched by operators at the station. This buffer represents what needs to be produced, but does not have a corresponding buffer slot in reality. The buffer size is set to the total number of carts in the system to make sure there is no choking point during the initialisation of the model. The cutting station area can be seen in figure 3.4. The empty cart can not move on to the initialising scanning operation if another cart is currently being processed. This is controlled through a *MaxWIP* object, which limits the work-in-progress between two points in the model. Once the empty cart can be engaged by the operator, it is first passed through an order creation to make sure the right material gets produced at the right time for the specific cart. The cutting process consists of two key operations, the cutting process and then the gathering of tubes by operators. If a tube bundle has not been picked by an operator the cutting station is blocked.

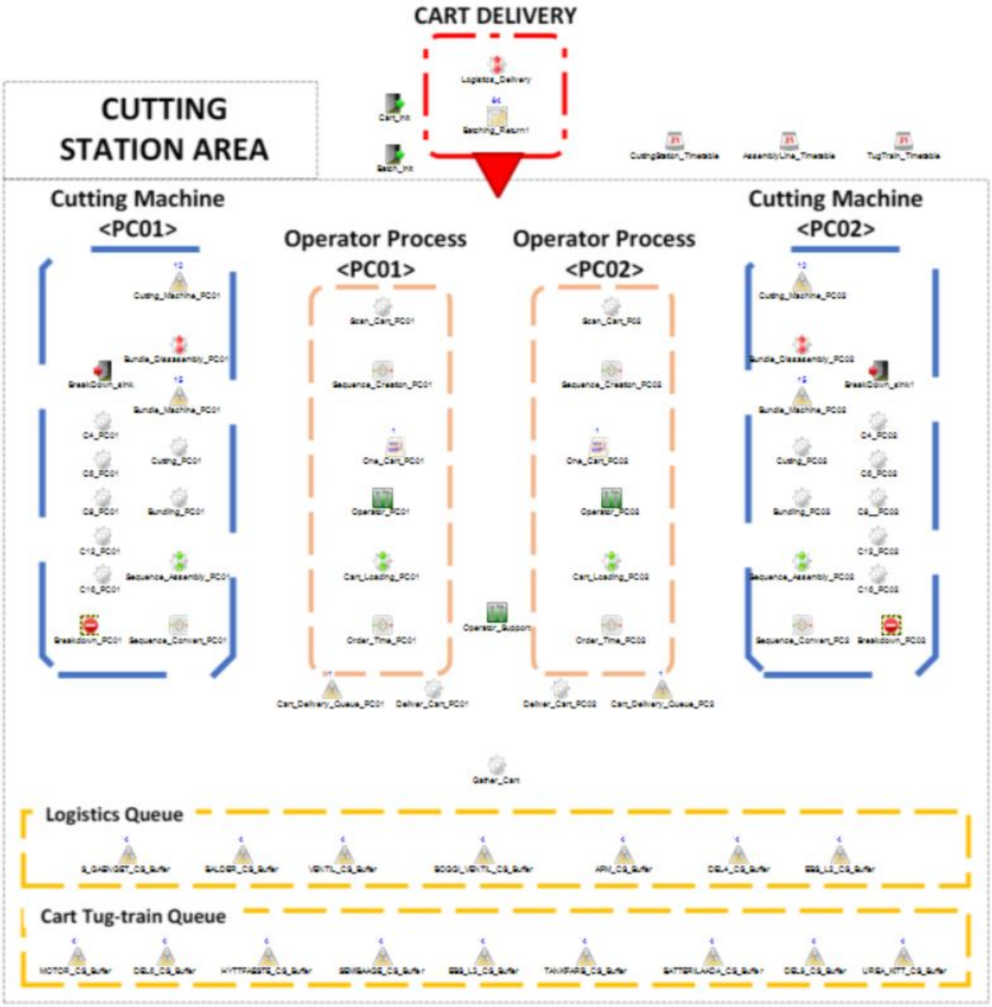

Figure 3.4: The cutting station processes as modelled in FACTS Analyzer

Once the cart has been loaded, all variants are subsequently unloaded and sent to their respective ahead-of-processing location. The sequences are sent into a buffer to await their turn to process their bundles. Following this, one sequence is split at a time such that the bundles are sent to be cut and sequences awaiting their previously split bundles. Once all of the sequence bundles have finished processing, the specific sequence is matched with a generic sequence for that delivery location and the latter is matched with the cart. Once all generic sequences are matched with the cart, the cart is delivered to its individual buffer for its delivery location.

3.5.2.2 Batching

The batching is handled by batch containers, which include bundles similar to the generic sequences. The batching as it is modelled in FACTS can be seen in figure 3.5. The batches are not consumed at the production line but rather an average depletion time is set based on the historical use of batched tubes. Once the time for depletion runs out, the batch container is returned to the cutting station to be processed once more. The batches holds a lower priority and through a dispatch of

Shortest Processing Time (SPT) is denied entry into the *Scan_Cart_MX* operations if any cart needs processing.

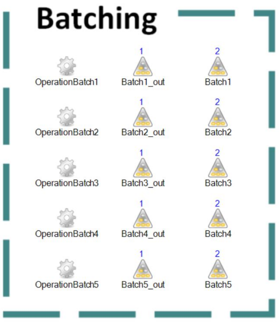

Figure 3.5: The Batching process as modelled in FACTS Analyzer

3.5.2.3 Tug-train Logistics

The tug-train logistics is simulated through a pull delivery, where the signal from a *Facade* object that is empty indicates that a new cart needs to be delivered from the cutting station to the delivery location. This is only done once the empty cart has been re-delivered to the cutting station. The tug-train as it is modelled in FACTS Analyzer can be seen in figure 3.6. All events are instant, part of the *TugTrain_Traveltime* buffer, which is set to the current delivery time. The delivery time is variable such that it can supply the minimum number of chassis per any delivery location to match takt-time.

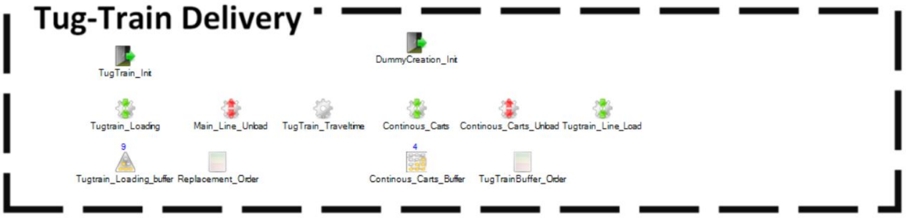

Figure 3.6: The tug-train logistics process

There are two different *Order* objects that manages facade dispatching. One is for the carts that are continuously stored on the tug-train as a buffer and the other one is for carts that get refilled from the cutting station. Finally, the *Circumvent_nr* buffers seen in figure 3.2 exist to redirect the flow arrows for visual clarity.

3.5.2.4 Production Line Assembly

The assembly line is modelled as a way to support the natural depletion rate of chassis from each cart. It stretches for the same number of stations as reality. Stations that consume tubes are modelled as *Assembly* objects. In figure 3.7, a part of the assembly line is depicted and the whole assembly line can be viewed with the full model in appendix A.1. The stations that do not assemble anything are modelled as *Operation* objects. Every assembly and operation has a constant process time which represents the takt-time. See section 3.5.3 for a more detailed explanation of the chassis sequence consumption.

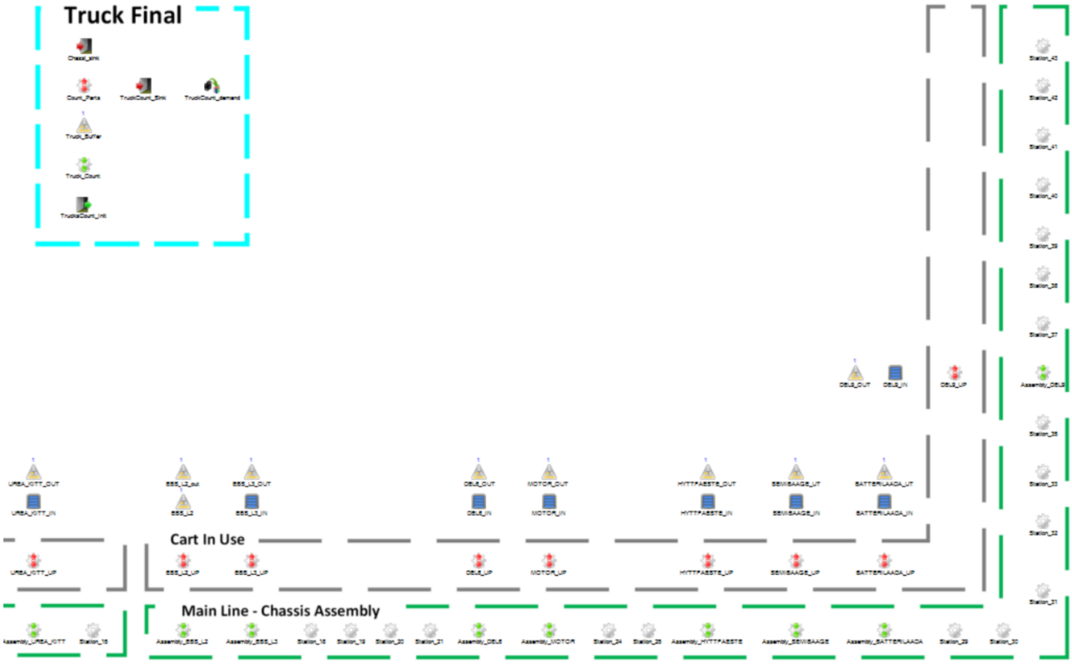

Figure 3.7: A part of the assembly line as modelled in FACTS Analyzer

Each use-point on the assembly line is modelled by four objects: either a queue or a facade, a disassembly, an assembly, and a final outgoing queue. The first queue or facade is the second in line to be consumed in the two-bin system. The disassembly empties the sequences into the assembly operation, but these are only consumed if a truck variant requires assembly from that delivery location. When a cart is empty it moves to the final outgoing queue. This structure of using more amount of stations than is necessary is done for two reasons: It separates consumption locations such that the model more closely resembles the physical process and also creates a structure for future use when adding an additional delivery location.

3.5.3 Material Depletion Rate

The sequences that are consumed from the carts at each station are matched to variants corresponding to trucks. 214 different trucks were identified with unique delivery location combinations. The details about how they were identified can be seen in section 3.6.1. A distribution of different combinations of each delivery location is randomly by proportion selected as the assembly. Through an array of assembly tables, all sequences from the required delivery locations per truck assembly combination are assembled onto the truck at takt-time. This is done to represent a natural depletion rate to support the JIS process. The contents of each cart, the chassis sequences, are consumed at each use point. Once the cart is empty, it is brought back to the cutting station through the same logistic means. As such, the carts are circulating in a loop through the model, from the cutting station to the delivery location and back. Moreover, the batching conducted as parallel operations to the JIT cart deliveries for each use-point is represented as variants in the model.

3.5.4 Simplifications in the Model

To keep the size of the model to a reasonable level it was necessary to make some simplifications and assumptions during the modelling process. In this project, the simplifications consist of:

- Sub-assemblies where the tubes are used are not modelled. Instead it is assumed that these processes happen instantly on the assembly line consuming the tubes.
- Only the tug-train logistical route is modelled. Based on interviews with experts other logistical routes are not a bottleneck in the process, since the assembly line uses a two-bin system.
- The tug-train travels periodically when the least number of chassis prepared in a cart times the takt time has passed. The tug train deliveries of carts are modelled as events and therefore all delivery locations receive carts at the same time.

These assumptions were made after consulting process experts to determine if these simplifications were reasonable. The tug train deliveries were modelled as events, rather than having time pass between different locations due to lack of data. Interviews with process experts confirm that this is a reasonable assumption.

3.6 Data Gathering

This section details the methods used in order to gather and handle the data required to build the simulation model. Data used in this project is divided into three categories. Category A contains available data, meaning data that is readily accessible in different formats. Category B is not available but collectable data. Finally, C is non-available nor collectable data (Robinson & Bhatia, 1995). Table 3.6 summarises the data types used in this project and to which category the data gathered belongs.

Table 3.6: Data acquisition methods used in the project

Data category	Data type
Category A	Historical production
	Machine log
Category B	Calculations
	Time studies
	Work measurements
Category C	Interviews

These data acquisition methods vary in data accuracy, where the lowest accuracy data is in category C (Robinson & Bhatia, 1995). In this project, category C data is only used when no other method of data acquisition was deemed viable within the project timeline. For each of these data sources, a data quality assessment was performed with the data use in mind. When deemed necessary process specialists were consulted to determine the required accuracy of different inputs for the model. All data acquisition methods used in the project will be discussed in more detail in the following sections.

3.6.1 Category A: Historical Production Logs

The historical production log (HPL) contains specifications about the pneumatic tube production for produced vehicles. This includes all tubes produced for all trucks at each delivery location along the production line. To further understand the HPL data collected for the project and identify trends within the data set, an exploratory data analysis was conducted. This was deemed to be of high importance when exploring tube demand. The goal was to identify trends in the tube demand at each delivery location so that the model could more accurately describe the process. During this process, the number of tubes, tube diameters, and tube lengths were considered. The data showed that there is a high variance in the number of tubes at different delivery locations as well as the length. There was also a difference in the tube diameters used at the delivery locations. Based on the results of the exploratory data analysis it was determined that the different delivery locations would have different process times and that they should be distinguished in the model.

The HPL was also used to examine trends with the chassis. This was done in order to identify trends in the use of tubes at different delivery locations. The frequency of tubes being used at a location were explored as well as the combinations of the different delivery locations for the chassis. With 16 different delivery locations on the assembly line and one separate delivery location, the theoretical number of different delivery location combinations can be determined through equation 3.1:

$$Total\ number\ of\ assembly\ combinations = 2^{16} = 65\,536 \tag{3.1}$$

After the data collection was analysed, information about eliminating some delivery location assembly combinations was made. Out of the possible 65 536 delivery locations combinations only 214 were identified in the data collection. For those 214 delivery location combinations the frequency of occurrence was also explored.

3.6.2 Category A: Machine Logs

The tube cutting machines saves a daily production log for each tube that has been cut that day. The data set contains details of the tube production, where of date, length, diameter and delivery location are the most notable for the project. For each tube a start time for the cutting process is logged, but the end time is not logged. Due to this fact, the worker and other interference have a large influence on the total time of each bundle of tubes in the machine log (ML). Therefore, the ML was deemed lacking and was not used to calculate process times.

Rather than using the ML to determine the cutting machine process time, it can be used to validate the process times of carts at the tube cutting station, as the model will contain worker interaction with tubes and the machine. This will be discussed further in section 3.7.2.1. The ML was also used to determine the amount of time required for an operator to gather bundles for different delivery locations.

3.6.3 Category B: Calculations

Calculations were used to determine process times and the rate of breakdowns of the two cutting machines. The calculations for bundle process times per delivery location was based on both historical data collection for over 5000 chassis and the supplier provided cutting machine specifications. To account for the product variants a distribution of process times for bundles in chassis at each delivery location was determined and used within the model.

The process times for each bundle were calculated based on known machine specifications such as tube feeding speed, cutting time, time for marking tube, time to place a tube in the gripper, and time to deliver and release tubes from the gripper. Equation 3.2 shows the formula used in the calculations. To validate the manufacturer's machine specifications, calculations were compared to tubes in the ML and the logged time. Outliers were filtered from the collection based on information from process experts and assumptions made by the modellers. Following this, the absolute error of the tubes was compared and the ML time was investigated. The values for cutting time, time for marking tube, time to place a tube in gripper and time to deliver and release gripper were then explored to minimise the absolute error. Values were adjusted based on the absolute error. The feeding speed values were not changed, due to the fact that that is a value that can be changed in the machine.

$$Bundle\ process\ time = \left(\sum_{0}^{N} TFT\right) + N * (CT + MT + PT) + D * RB \tag{3.2}$$

Where:

TFT = Time to feed tube
N = Number of tubes per bundle
CT = Cutting time
MT = Marking tube time
PT = Time to place tube in gripper
D = Number of deliveries
Rb = Time to release bundle

Equation 3.2 was also used to calculate batch process times. The batches consist of a fixed amount of tubes of the same diameter and length. Therefore the cutting time can be calculated using equation 3.2 and batch tube specifications.

The HPL was also used to determine the rate of disturbances for the two tube cutting machines when coils run out. For the tube cutting machines there are five different types of disturbances. Four are related to coil breakdowns and one is related to machine stops due to cleaning being required for the inkjet printer. For the duration of the cleaning, the machine is stopped and a worker is required to clean. While a coil is being changed the machine can cut another type of tube but will in the end stop. However, this is modelled as a breakdown since the tube cutting machine requires an operator to change the coil and therefore the machine will stop in most if not all cases.

3.6.4 Category B: Time Study

A time study was conducted for two reasons: to collect data on operation times that are not accessible in any other format and to gather data to compare the DES model's throughput time as a way of validating the model's output. The data gathered in this way would according to Robinson and Bhatia (1995) be Category B data, not available but collectable. In this project, time studies were conducted to find the transport time of carts within the cutting station area.

3.6.5 Category C: Interviews

Data that is not available nor collectable was obtained through interviews with experts. Experts with relevant experience in a related area were identified and questioned about the process. Through this method, further information about the process is gained. According to Robinson and Bhatia (1995), this kind of data is classified as Category C, not available nor collectable. Often, experts will provide data of lower accuracy than larger data collections and therefore interviews are a last resort used when no other means of gaining the information is presented (Robinson & Bhatia, 1995). Interviews can also serve as a way to gain knowledge about a process.

Interviews with workers at the station were carried out early in the project. This was done to increase understanding of the work performed at the station. Additionally, the workers are knowledgeable of work disturbances relating to breakdown causes, breakdown times, and breakdown frequencies. The interviews conducted at the start of the project were semi-structured, and the worker was questioned about details in the process, and in some cases to perform specific steps in the process. During the interviews, open questions about different parameters and workflow were asked, so that workers can answer without being influenced by preconceptions from the interviewers or other engineers.

A secondary round of interviews was conducted later in the process to gain a deeper understanding of process times, the disturbance rate of the machines, and repair and setup time. Since the machines were new, limited data was available about breakdowns and worker estimation was useful. The interviews were also used to estimate the accuracy of process times since the machine process times were calculated using data provided by the manufacturer and not previously recorded.

Interviews also serve a secondary purpose, to involve stakeholders in the project and increase confidence in the results (Banks et al., 2010). This will also increase stakeholders knowledge of the simulation and simplify maintenance of the model in the future.

3.6.6 Data Sources in the Model

The data sources in the project have been discussed in detail and a summary of the data sources and the data types acquired from them can be viewed in table 3.7. Throughout the data gathering, data quality was considered since it can have a significant impact on the result of the model. The data quality dimensions are discussed in section 2.3.2.

Table 3.7: Summary of data sources used in the model

Data source	Data type
Historical production log	Cutting machine breakdown rate
Historical production log	Truck variants
Historical production log	Batch consumption rate
Calculations	Cutting machine process times
Time study	Manual cart movement by workers
Interviews	Cutting machine mean time to repair

3.7 Validation

In this section, the validation of different parts of the model are detailed. This includes technical validation, process time validation and operational validation.

3.7.1 Technical Validation

Technical validation involves validating whether the underlying assumption and theories about the process are correct as well as the models' structure, meaning the logical, mathematical and causal relationships fit the purpose of the model and align with the physical process (Sargent, 1981). To validate the conceptual model two approaches were used.

1. Validate through monitoring the process and making necessary adjustments based on the physical process.
2. Validate through interviews with process experts. The main focus was on the detail level and conceptual model from figure 3.2

Monitoring was done on multiple occasions, both in an organised manner and if any unexpected questions or dilemmas arose during the modelling. This involved watching the process without any interactions, and on other occasions workers were interviewed in an open setting without predetermined questions.

Interviews were conducted with process experts as well as different stakeholders in the project. The process experts were asked about details regarding work in the process and provided clarity regarding work processes. The stakeholder interviews were more focused towards the expected outcome and whether this detail level accurately was sufficient to provide all of the required answers. The conceptual model was presented to process experts and stakeholders at the same time, where open dialogue and discussion were encouraged. Based on feedback adjustments to the conceptual model were made.

3.7.2 Data Validation

Data validation is the process of ensuring that the data used in the model is sufficiently correct. The data that should be validated includes values used in the model and modelling tool parameters, as well as the data used for operational validation (Sargent, 1981). Sargent (2010) identifies three steps for data validation:

1. Collecting and maintaining data.
2. Test the quality of the data set collected.
3. Search the data for outliers and determine if those outliers are correct.

3.7.2.1 Process Time Validation

Data validation was focused on the cutting machine process times. This was done as a way of ensuring that the cutting machine process time of each cart in the model coincides with reality. This was done using the logged times in the ML and comparing it to the model output. The ML contains production data with the start time of a tube logged and was used to validate the process times at the cutting machine. However, this data contains breaks during shifts, times when there is no one working at the station and other noise in the data. To clean the MLs, a script was constructed. The process time validation was performed for two reasons:

1. Validate that the model's cutting time of each cart is corresponding to a real distribution.
2. Validate that the model with all components and interactions behaves similarly to historic production.

At first, the ML was combined and cleaned as it contained a lot of noise as discussed in section 3.6.2. During the data collection, the format of the ML was changed such that additional data categories were included. This means that some of the data collected at different times had a varying number of measurements and this had to be considered when combining the data. As the data contained date-time information, a tube cutting time was roughly based on the start of said tube and the start of the next. Equation 3.3 shows how the tube cutting time in the ML was calculated. T_i denotes the time a tube took to process, and t_i denotes the date-time of finished production for a tube. The time for a tube was the date time of the following minus the start of the next. It was then explored how long it took from the first tube being cut for the cart until the last tube was cut. Carts were removed from the data set if any pipe in the cutting process was above the tolerance limit due to any disturbance factor.

$$T_i = t_{i+1} - t_i \tag{3.3}$$

The first validation resulted in too few carts due to narrow and demanding criteria. Thus, an additional validation technique was conducted. Instead of removing any carts that had individual tubes that exceeded the tolerance limit, the values were replaced with interpolated values for that delivery location. The interpolated cutting times per tube were based on the average of all tubes for that delivery location that did not exceed the tolerance limit. Carts were instead removed if they met any of the below listed conditions:

1. More than 4 tubes exceeded the tolerance limit.

2. Fewer chassis than planned were connected to the production of a cart.

3. Too many chassis were connected to the production of a cart.

4. Carts that still lay outside the acceptable estimated cart tolerance limits.

It was thought that too much interference in adding interpolated values might result in poorer quality of the data. Therefore, a limit was added to avoid skewing the data. Also, due to any reason, production can be changed to include additional or fewer chassis than normal. This would be an unequal comparison towards the model and such carts were also removed. However, if a cart was approved, its process time was calculated according to equation 3.4.

$$Cart(DL)_i\ time = (T_{N,i} - T_{1,i}) - Ext + NE * Int(DL) \tag{3.4}$$

Where i denotes the indexed cart number in chronological order and N is the number of tubes belonging to that cart. In table 3.8 all abbreviations are disclosed.

Table 3.8: Explanation of numerical values

Parameter	**Effect**
Delivery location	DL
Time per tube in order per cart	T_i
Date-time per tube in order per cart	t_i
Number of tubes within a cart	N
Summarised exceeded-time	Ext
Number of exceeding instances	NE
Interpolated time per delivery location	Int(DL)

Additionally, the model itself was run for the production of one cart for each delivery location over 300 replications. This was done to get a sizeable sample to measure both the average and standard deviation. Finally, the model output was compared to the ML values and the total average was calculated.

$$Average\ error(DL) = \frac{(Model\ average(DL) - ML\ average(DL))}{Model\ average(DL)} \tag{3.5}$$

3.7.3 Operational Validation

Operational validation is the process of determining whether the model output behaviour is sufficiently accurate for the models‘ intended application (Sargent, 1981). As stated by Sargent (2010) there are multiple ways of conducting operational validation, including but not limited to the following methods used in the project:

- **Animations:** The movement of products through the system was monitored to see if the flow of products in the model corresponds with the flow in the physical system.
- **Face validation:** The user of the model and process expert was consulted and questioned about whether the model logic and assumptions align with his experience of the physical process. This was done repeatedly throughout the process to validate the details that were added. Involving the user in the modelling process also increases the confidence in the model (Banks et al., 2010).
- **Operational graphics:** This is the process of monitoring different performance measurements in the model as it is run. This can include queues, resource usage and the state of operations. In this project, the buffers at the tube cutting station in the model were monitored and compared to the physical process. More specifically how many carts were usually at the station based on the author’s experience.
- **Traces:** Trace is when different variants are tracked through the simulation to see if the logic in the computerised model is correct. In this project traces were most frequently used with events that have a predetermined fixed interval. Then the effects of those events can be monitored. This method was used throughout the validation process because it makes it easy to trace issues related to specific events and to simply monitor the effects of specific events.

These methods were applied throughout the modelling process repeatedly. When details were added to the model they were validated concurrently.

3.8 Experimental Design

Experiments were performed in order to answer the research questions the model should be able to answer. Some scenarios are simple and only require a change to be made to one parameter and connecting flows. Others are more complex and then a multi-objective optimisation was carried out. In total, two different experiments with a set of sub-experiments of varying factors were explored. The experiments were performed separately and compared to the system’s current state.

In the experiments, different factors were varied and different responses measured depending on how each scenario is analysed. The responses are measured in two ways. The first metric is the number of trucks produced per shift. This is assuming that all other processes supporting the production line assembly can keep up with the specified takt-time. The theoretically highest capacity of the plant for each shift as a function of the takt-time can be viewed in equation 3.6. The theoretical output depending on the takt-time can be illustrated such as in figure 3.8.

$$Plant\ output\ capacity \left[\frac{trucks}{shift}\right] = \frac{Seconds\ per\ shift}{takttime\ per\ truck} \tag{3.6}$$

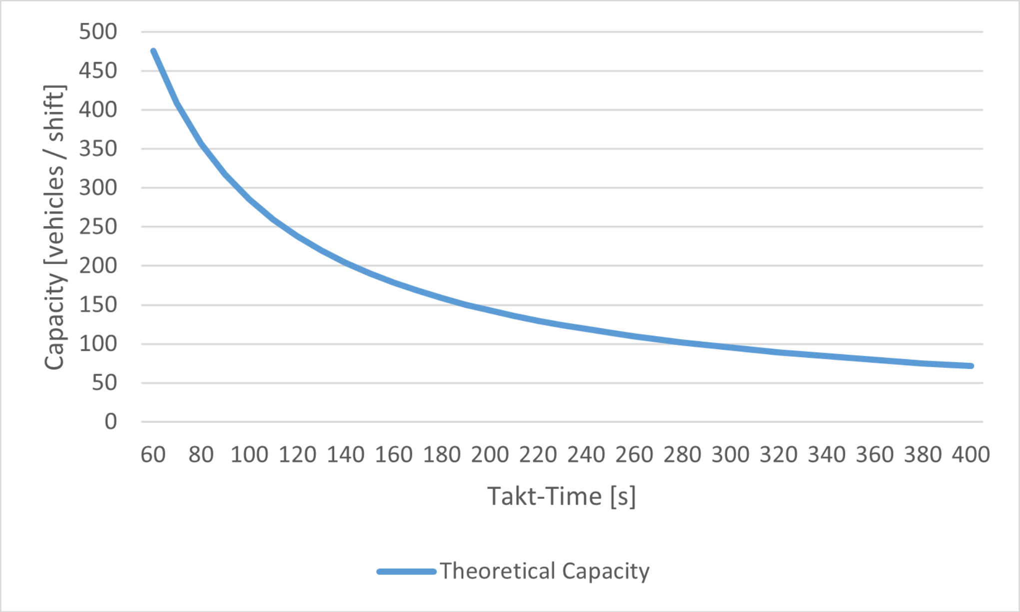

Figure 3.8: The theoretically highest plant capacity as a function of variable takt-time

The model is built in such a way that only the material deliveries from the cutting station can affect the capacity of the plant. If any required material arrives late to an assembly station it will cause a block in the production line. This block would then reduce the measured output. Looking at figure 3.8 it can be seen when the tube cutting station is a bottleneck in the production, and decreases the plant capacity. Therefore, the cutting station is considered to be able to keep up with the changes to takt-time if it does not deviate from the theoretical plant capacity.

The second response measured in these experiment is backlog. Backlog counts the number of trucks that arrive late to the final station on the assembly line. In the model, the *Demand* object at the end of the production line requires a truck to be delivered every takt-time. As previously mentioned, it is assumed that all processes on the production line can keep up with the specified takt-time. The model is built in such a way that the only thing that can lead to backlog is if tubes are not delivered in time from the cutting machine.

For each experiment the simulation initialisation period, length of the simulation runs and the number of replications made for each run need to be decided (Banks et al., 2010). In this project the simulation horizon and warm-up time is standardised in for each experimentation. It is based on simulating a week of production. Since the model does not account for non production hours, this is normalised to be the total time spent in production for seven continuous days. This is equal to 4 days, 15 hours and 4 minutes. The warm-up time is set to three days, giving the model time to reach a steady state. Furthermore, 20 replications for each experiment iteration was done. A summary of the setting detail can be viewed in table 3.9.

Table 3.9: Simulation settings for the experimental study

Start:	2010-01-01 00:00:00
Simulation horizon:	7:15:04:00
Warm-up Time:	3:00:00:00
Replications:	20

3.8.1 Experiment 1: Takt-time Changes

Changing the takt-time would change the rate of demand on the cutting station as material is consumed more or less frequently. The factor in this experiment is the takt-time, i.e., the rate at which trucks move from one station to the next. That is, takt-time determine the process time If the takt-time is reduced, the rate that each cart depletes its tubes is increased. This means that a relatively higher output from the station is required. In contrast, if the takt-time is increased, the demand from the cutting station is instead lowered. The takt-time is changed by connecting all stations along the assembly line to one united variable, and then spanning the value of the variable. The values for takt-time in this experiment is spanned between 60s to 400s, with a step size of 10s. In total, this accounts for 35 measurements with 20 replications internally per measurement. The response from the experimentation run is the capacity of trucks produced per shift which is given by:

$$Capacity\left[\frac{Trucks}{Shift}\right] = Trucks\ throughput\ *\ Shift\ length \tag{3.7}$$

The intention from this experiment is to visualise at what point the cutting station is not able to keep up with the demand of the production line assembly and when backlog starts becoming non-zero.

3.8.2 Experiment 2: Buffer Reduction

The tube cutting station cuts for several chassis ahead of time and then the carts are placed at the assembly line. The assembly line uses a two-bin system, meaning at any point there should be two carts stationed at the cart delivery location. When a cart is empty it will be replaced by a full one, and in the meantime, the second cart acts as a buffer (Jonsson & Mattson, 2009). However, if components are missing from a truck that is scheduled to be produced it may be removed from the line. This means that tubes can stack up while waiting for postponed chassis. By reducing the number of carts in circulation these scenarios will happen less frequently. This also increases available space at the station.

In experiment 2, the number of carts required for each deliver location is investigated. The goal is to identify the minimum amount of carts required so that the tube cutting machine does not slow down the assembly line. Therefore the measured output for this experiment is the backlog of the assembly line demand, which is called TruckCount_Demand in the model.

Currently, all delivery locations except three have four carts in circulation. By reducing the number of carts at specific delivery locations the total amount of tubes in circulation at any time to be reduced. These delivery locations were picked based on which delivery locations were of interest to the process experts. These are locations where the number of carts may be reduced in the future if it is proven that it does not affect the production line. The parameters of the experiments can be viewed in table 3.10.

Table 3.10: Experimental design for experiment 2

Name	**Set**	**Lower bound**	**Upper bound**	**Base**
Cart_Count_D1	{3;4|1}	3	4	1
Cart_Count_D2	{4}			
Cart_Count_D3	{3;4|1}	3	4	1
Cart_Count_D4	{4}			
Cart_Count_D5	{3;4|1}	3	4	1
Cart_Count_D6	{4}			
Cart_Count_D7	{3;4|1}	3	4	1
Cart_Count_D8	{4}			
Cart_Count_D9	{3;4|1}	3	4	1
Cart_Count_D10	{4}			
Cart_Count_D11	{2;3|1}	2	3	1
Cart_Count_D12	{4}			
Cart_Count_D13	{1}			
Cart_Count_D14	{3}			
Cart_Count_D15	{4}			
Cart_Count_D16	{4}			
Cart_Count_D17	{4}			

The chosen optimisation algorithm for this experiment was the Latin Hypercube sampling method. To simplify the process of eliminating solutions that were not an option a constraint was defined, which is depicted as equation 3.8

$$Criteria: \quad TruckCount\_Demand\_Backlog = 0 \tag{3.8}$$

The goal is to minimise the number of carts required to keep up with the demand of the assembly line and to support that the variable total cart was defined as can be seen in equation 3.9.

$$MINIMISE: \ TotalCarts = \sum_{i}^{DL} \sum_{k}^{N} Cart_{i,k} \tag{3.9}$$

Where DL is delivery location and N is the number of carts per a delivery location.

3.9 Model Maintainability

In this project, the aim was to build a simulation model that can be used long-term. Therefore the subject of model maintainability was studied, both from an organisational perspective and a technical perspective.

To explore model maintainability, a literature study was conducted. The literature study did not solely focus on model maintainability. Rather the scope included the concept of maintainability within different areas, mainly within discrete even simulation (Brooks & Tobias, 2000; Piplani & Pauh, 2004; Sargent, 2010; Williams, 2002; Oscarsson & Moris, 2002), software development (Christa et al., 2017; Ebert et al., 2016; Gupta & Gayathri, 2022; Heričko & Šumak, 2023; IBM, n.d.; Robinson & Bhatia, 1995; Schnappinger et al., 2020) and mechanical maintainability (Bidanda, 2023). In addition to the literature study, the authors of the paper had multiple discussions about what was required to maintain the model and what can be done to support model maintainability. This was discussed continuously throughout the project. The authors also included the model's future user and process expert in this discussion, in the form of unstructured interviews. These interviews were conducted in connection to the technical and operational validation (Sargent, 1981, 2010) of the model. Where the user viewed the model and commented on both its accuracy as well as how he understood it. Finally, the authors of this paper had previous experience with DES modelling and used their knowledge and observations during the project to contribute to the maintainability study.

4

Results

This chapter describes the results that were held by the methods carried out, presented in the method chapter 3.

4.1 Model Validation

The validation of the model consists of three parts, technical validation, data validation and operational validation. Technical validation and operation validation were performed in replications. Technical validation was performed by monitoring the stations and making the required adjustments to the model as well as interviews with stakeholders. These interviews were mainly focused on the level of detail and the conceptual model. Based on observations and comments from separate interviews, the model was adjusted until it met the stakeholders' approval. Operational validation was performed periodically as the model was built. As details were added to the model the function of the model was tested through methods discussed in section 3.7.3 until all requirements were fulfilled. Data validation was focused on the tube cutting station process time validations.

4.1.1 Process Time Validation

As discussed in section 2.3.3, the validation of process times was considered an important aspect of the model and was validated through the comparison of historical ML data and the model output. The ML that include timestamps were used and filtered so that unusual activities such as breaks, shift end and lack of carts will not disturb the total cart times. When all delivery location process times that did not fit the pattern had been excluded, the resulting carts were used to find the average and standard deviation.

In table 4.1, a comparison of the model average cart times, standard deviation to the average cart times, as well as average and standard deviation from the filtered ML can be viewed. In addition to the average error of each delivery location, the table includes the number of measurements used from the ML. The results when comparing the model output to the approved times from the ML show a varying average error. A negative percentage value indicates that the model produces faster than the ML suggests, and a positive percentage means the model produces slower. The number of measurements varies between delivery locations, this was deemed to be due to the different frequencies of tube use at the delivery locations as well as internal random disturbances that were removed from the calculation. Since the process has a high variation in terms of production, and the sample from the ML contains roughly a month of data, the low error results are a good indication towards model accuracy.

It should also be addressed that the standard deviation of the model output is based on one cart being produced per replication and 300 replications being performed in total. This was done because FACTS will only depict a comparison of different replications, which may include multiple carts.

Table 4.1: The error results from the model compared to the Machine Log output

Delivery Location	**Model AVG** [Min]	**Model STD** [Min]	**ML AVG** [Min]	**ML STD** [Min]	**Average Error** [%]	**Measurements** [-]
D1	5,38	0,71	-	-	-	0
D2	1,73	0,33	1,75	0,54186	-1%	63
D3	1,42	0,22	-	-	-	0
D4	3,08	0,03	3,15	0,46975	-2%	5
D5	2,18	0,11	1,94	0,22193	11%	11
D6	3,07	0,46	2,28	0,51415	26%	2
D7	2,77	0,38	2,29	0,57157	17%	27
D8	2,72	0,44	-	-	-	0
D9	3,60	0,53	4,39	0,65435	-22%	4
D10	15,03	0,93	15,68	1,399	-4%	223
D11	7,15	0,71	7,05	0,60208	1%	224
D12	2,83	0,59	-	-	-	0
D13	5,92	1,11	6,27	6,1153	-6%	3
D14	1,23	0,27	-	-	-	0
D15	7,02	1,28	-	-	-	0
D16	8,05	0,94	8,16	0,92919	-1%	9
D17	7,33	1,61	7,98	1,5821	-9%	3

As Law (2007) states, a validation of a complex system such as with great variation like the system presented in this project, can never be fully validated. The error values are for the most part small, but most are hard to argue for being correct since few measurements exist. Some delivery locations contained zero measurements and were of course disregarded. In other cases the delivery location had too few measurements to be considered valid. The average error for delivery locations with over 50 measurements is between -4% and 1%. This was deemed to be accurate enough for the purpose of the model. With a larger data collection, more extensive validation might have been possible, but for this project, a larger data collection was not available.

The cart process times were confirmed by a process expert. Although delivery location D10 is according to the process expert considered to be the most important production to validate, it had a low but negative (-4%) average error from the

223 complete carts measured from the ML. However, it is considered better that a simulation model underestimates the total production time, such that it is more credible. The negative values are thus less desired as they would lead to a higher output than might be true.

4.2 Experimental Study

In this section, the results of the experimental study are discussed. The experiments displayed in this thesis are examples of possibilities of analysis provided by the model. Experiments were limited to what was of immediate interest from the model, based on the stakeholder analysis and continuous meetings with the end user of the model. In addition to the experiments, the model is built to house room for analysis of interactions and changed process parameters that were established to be of other interest as well.

4.2.1 Experiment 1: Changes to Takt-time

The results of the experiment regarding the peak of capacity when changing takt-time can be seen in figure 4.1. The model output suggests that it can keep up with the demand from the assembly line at takt-time until a certain point. Here it is visualised that the cutting station is able to supply material for roughly 113 chassis per shift. At around a takt-time of 250 seconds, the model output diverges from the theoretical output. The horizontal asymptote shows that the cutting station process is unable to supply the main line assembly when going below a takt-time of about 250 seconds.

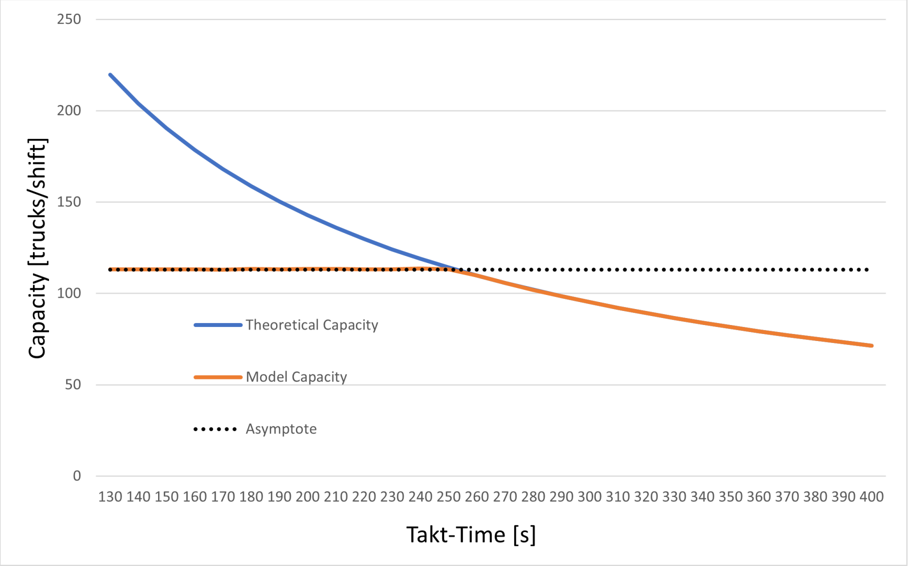

Figure 4.1: The capacity of the cutting station as a function of takt-time

However, looking at the backlog at around the same interval of takt-time in which the capacity diverges from the theoretical output. In figure 4.2, a close up interval between 250 and 262 and the model output on backlog is shown. It can be seen that a backlog starts becoming non-zero at around 260 seconds takt-time. The backlog indicates that vehicles arrive late due to disturbances in the assembly processes due to lack of supplied material by the cutting station. The capacity at 260 seconds takt-time is around 110 chassis per shift.

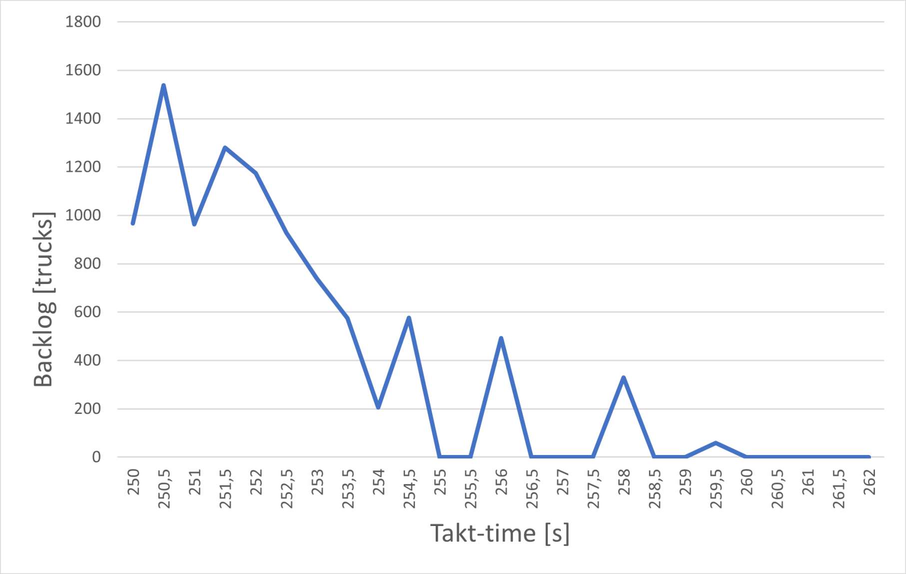

Figure 4.2: The backlog as result of changed takt-time

From these experiment it can be deducted that the current state capacity of the pneumatic tube cutting station is around 110 to 113 chassis/shift at around 255 to 260 seconds takt-time.

4.2.2 Experiment 2: Varying Buffer Levels

The result of the experiment with a varying number of carts can be viewed in figure 4.3. It can be seen that many of the experiments result in zero backlog despite the total cart count being reduced from the original number, 64 carts. This indicates that the current number of carts in the system is higher than required using the current takt-time at the assembly line. Based on this the experiments with the zero backlog and the lowest cart count were further explored and the results can be viewed in table 4.2.

In the experiment, the minimum number of carts is 57, that is when all parameters are set to the lower bound of the experiment. Table 4.2 shows three results from the experiment with zero backlog and the minimum number of total carts. However, it can also be seen in table 4.2, that in some cases there is backlog with 58, 59 and 60 carts in circulation. This contradicts that the number of carts at all delivery locations can be reduced. Therefore, the result from this experiment can be questioned and should be further explored in the future. Using backlog as a measurement in this model can also lead to the results being challenging to read.

As the model is built in such a way that one truck being late means that that truck and all the following will be logged late and backlog is registered. This means that a backlog of 200 can mean 200 trucks were late or that 1 truck was late in the beginning of the experiment.

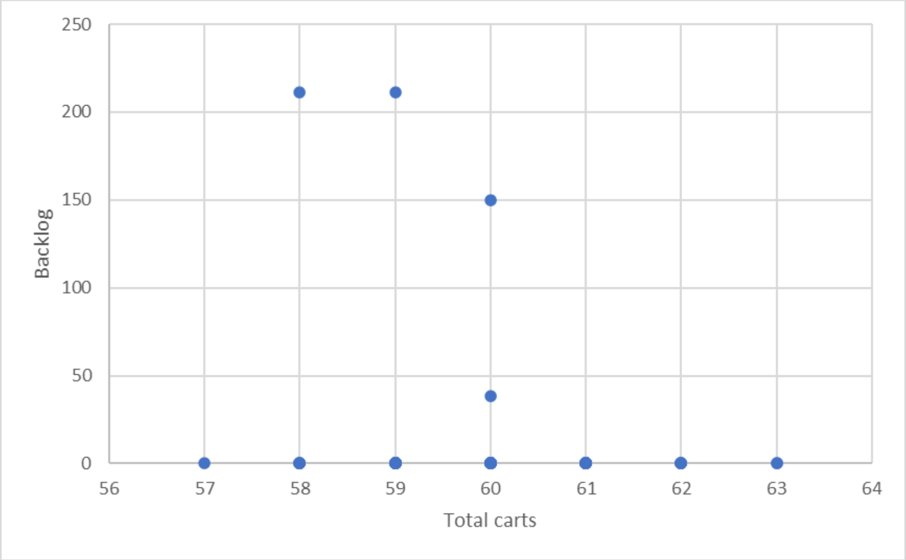

Figure 4.3: Backlog as result from changes made to the number of carts

Table 4.2: Results from experiment 2 with zero backlog and the lowest cart count

Delivery location	**Result 1**	**Result 2**	**Result 3**
D1	3	3	3
D2	4	4	4
D3	3	3	3
D4	4	4	4
D5	3	3	3
D6	4	4	4
D7	3	3	3
D8	4	4	4
D9	3	3	3
D10	4	4	4
D11	2	3	2
D12	4	4	4
D13	1	1	1
D14	3	3	4
D15	4	4	4
D16	4	4	4
D17	4	4	4
Total Carts	57	58	58

Based on the result of this experiment, it is advised that if the number of carts will be reduced it should be done in steps and monitored closely. The results indicated that the number of carts can be reduced for all the delivery locations that were investigated. The results from the experiment show that the lowest number of carts is 57.

4.3 Maintainability

Throughout the model-building process, the concept of building a maintainable DES model was a high priority. To support this, the topic of maintainability was researched within different areas. As discussed in section 3.9, the literature study was mainly conducted within the area of DES model, software and machine maintainability. In addition to the literature study, the authors' own experience through the project was used, as well as unstructured interviews with the model user. This results in the definition of two different criteria: A criteria for technical support for a maintainable DES model and another for criteria for the organisational support of a maintainable DES model. Throughout the project, these criteria were applied if possible.

Furthermore, it should be noted that these criteria are mainly defined for DES models in a simulation environment. It was deemed that a framework developed for DES using a general-purpose programming language or simulation programming language would contain different criteria and in this case the simulation software FACTS Analyzer was used, which is a simulation environment (Banks et al., 2010).

4.3.1 Observations and Interviews

In this section, we will discuss the results of our interviews and observations made during the project. Based on interviews with stakeholders at the beginning of the process, it was clear that the results of the model should be able to communicate results to people at all levels within the company. The interest in the model is diverse, where some will want to understand how it was built and its data sources, while for others the interest lies in how accurately the model represents the physical system and what information the model can provide. Therefore it was deemed necessary to build a model visually similar to the physical system. This is also important because a DES model is often used to visualise problems and showcase different scenarios in a process. Not only is the summary data from the simulation of value but also being able to visualise the flow of the process helps establish the credibility of the solution.

Throughout the project, different people interacted with the model and it was clear that different competencies are required in a project of this magnitude. More specifically a lot of process knowledge is required and having a process expert to support the model building supports model credibility and decreases the risk of the model being inaccurate. Another qualification that is required to maintain the model is discrete event modelling knowledge. With this and software experience the maintainability of the model is less time consuming and the model is less prone to mistakes.

Some observations were made during the model building phase in the project. The simulation software used in this project is FACTS Analyzer and it has a feature allowing model data to be read from an input data file. This was a method that was frequently used during the model building and was considered an easier method to keep track of input data. From a maintainability perspective, the option of updating values in an output document can be useful since often it can be easier to navigate and does not require much simulation knowledge. Using an automatic data input also decreases the risk of errors being made by the modeller when the data in the model is updated.

At the beginning of the project, the simulation software had already been chosen for the project. Throughout the project, the authors would tend to compare previous DES modelling experience and discuss the suitability of different simulation software. The idea that a different software would result in the project looking very different arose, as the properties of the software will have a big effect on a project. Therefore, it was determined that to make a model more maintainable, the software chosen for the project should be carefully considered at the beginning of any project and picked based on the scope of the project.

The documentation that was provided alongside the DES model contains the current state description and function of the system. If this changes in the future, it is important that the documentation changes with it. If the changes are made to the model but are subsequently not added or changed in the external documentation, the documentation becomes obsolete. More importantly, this also brings the fact that no proper documentation exists of the updated model. In regards to this, knowing where to look and what to change when making updates to the objects of a model is essential knowledge to possess. An increasing amount of flows, variants and combinations thereof, similar to what is described cyclomatic complexity by Heričko and Šumak (2023), increases the challenge of understanding the model. This underlines the importance of a well-documented DES model that is changed in parallel with the documentation.

4.3.2 Organisational Maintainability Framework

The organisational maintainability framework was built based on the literature study, see section 2.6.1, as well as the authors' experience during the case study. The following criteria were identified:

- Documentation of the model
    - Documentation at several levels for different stakeholders
- Schedule for model review
    - Decide a frequency of review that fits the model purpose
    - Decide a plan for what to review
- In-house DES modelling knowledge
- In-house process knowledge

Documentation is important for the maintenance of the model in the future. Assumptions and decisions made during the modelling process are documented for future users. This means that less time will be spent navigating the software and understanding the model. Documentation may be done at different detail levels depending on who it is intended for. As Heričko and Šumak (2023) discuss in their paper on software maintainability index, documentation plays a key role in deciding the quantitative measurement of the software's maintainability.

Similarly to how maintenance is done for machines, a DES model has to be updated regularly. Never updating a DES model in between uses would gradually render it useless as the system or process is not represented by it accurately anymore. Providing an unintended failure replacement approach, as described by (Bidanda, 2023) when maintaining a DES model would not prove useful as there are no indicators as to when the model has become obsolete. Similar to what is stated by Sargent (2010) about machine PM, a scheduled revisit to the model and checking that it is still valid is necessary would prove useful.

It was observed during the project that to maintain a DES model the right knowledge is required. Both DES modelling knowledge and process knowledge are required if any changes are to be made to the model. This is not only to increase maintainability but also so that the changes made to the model are correct.

4.3.3 Technical Maintainability Framework

The technical maintainability framework was built based on the literature study, see section 2.6.2 and the authors' experience during the project. The framework is listed here:

- Understandability and clear visualisation
- Appropriate detail level in the model
- Good modelling practices
    - Re-use components if possible
    - Clear internal documents
    - Minimising the use of embedded constants
- A suitable simulation software
- Automated data input
    - Input data document
    - Automatic Data handling

Understandability of a DES models in a simulation environment is not clearly defined in the literature found in this thesis. The criteria of understandability in this project refers to the ease with which a model can be understood. In general, a model can be better understood if it accurately represents the process that is being modelled. In software development, the understandability of programming is of high importance (Schnappinger et al., 2020). This can also be applied to DES models.

Choosing an appropriate level of detail that fits the purpose of the model, such that it can answer all relevant questions. Adding unnecessary detail to the model complicates the model leading to more complicated maintenance activities later on. Therefore, setting an appropriate level of detail early on in the project and not exceeding that contributes to a more maintainable model was deemed to be an important aspect of building a maintainable model.

Using good modelling practices is also beneficial for model maintainability. This includes re-using constructs within the model. By re-using the same structures the model can be more easily understood and therefore more easily updated and debugged. Making clear comments throughout the model can help another user in the future and makes the model easier to navigate. With clear comments throughout the model, it is easier to understand the model in a shorter time, while decreasing the risk of misunderstandings. Less time will be spent on maintenance activities and the risk of creating a bug in the model is decreased. The use of embedded constants within the model should be avoided, especially variables that can be of interest to the modeller. In the case where constants are used the source of data and reason for use should be clearly documented. By introducing variables instead the user of the model has the option of interacting with the variable and changing its value. Then the effect of those changes can be explored to gain further insight into the process.

As discussed in section 4.3.1 the authors discussed how different software can impact how a model will look. Therefore it is argued here that choosing the right software that fits the problem can simplify the model and lead to more credible results. Since the software can have objects more suited to the project or be able to convey results differently. Some will suit the project better than others.

Automating the data input leads to a smaller risk of issues arising when the model data is updated. This minimises the risk of the modeller inputting the wrong data, leading to the wrong result from the model.

4.3.4 The Maintainable Framework

In the previous two sections, frameworks from two different domains are highlighted to support the maintainability of a model. In figure 4.4, the two proposed frameworks can be viewed. However, this thesis does not address the perks of combining these frameworks or how different criteria within these frameworks may compliment each other.

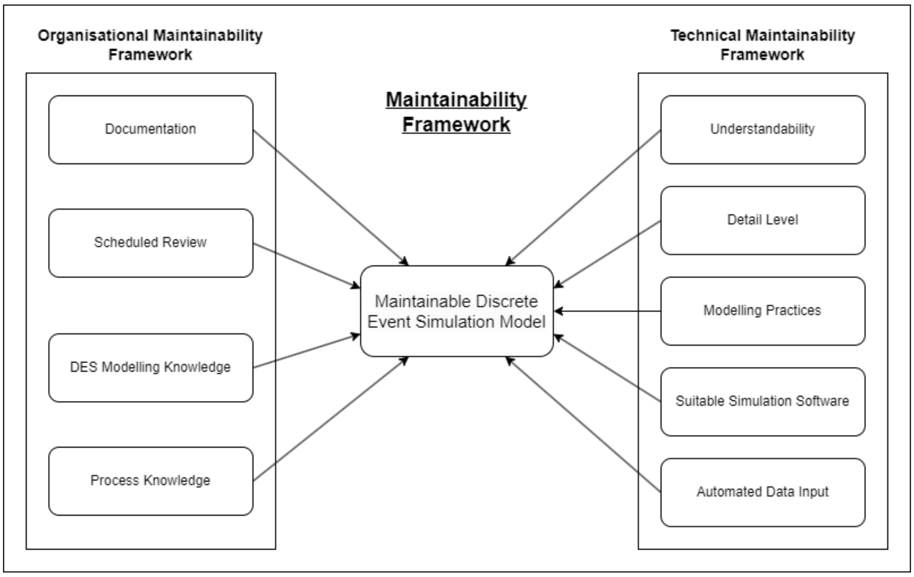

Figure 4.4: The maintainability framework with a combination two criteria frameworks

4.4 Application of the Maintainability Framework

In this section, the model built in this project will be evaluated based on the framework established in section 4.3.

4.4.1 Evaluation Based on Organisational Criteria

The identified criteria for organisational maintainability was considered during the building of the simulation model in this process. When possible, arrangements were made to fit the criteria, but some are not within the control of the authors. For example the criteria of in-house process knowledge and in-house DES knowledge cannot be controlled by the project or the authors and will therefore not be addressed. Suggestions can be made regarding scheduling regular model reviews but it is not within the scope of the project to follow up on the models future state.

The input data dimension of the organisational maintainability framework was addressed in the project. An Excel file is used to manage data in the model. The information that is provided to the model in this project has some important changeable aspects. Most assembly tables in the file are automatically generated based on the HPL in the standard format at the company. However, other data such as process time distribution requires more effort and time to update. The data file requires pre-processing before relevant information can be extracted. This applies to all data acquired using calculations. However, it also contains information that requires automatic input, e.g., the numerous assembly tables depending on the assembly combination and delivery location. It should also be addressed that drastic changes like adding a delivery location will require some changes to be made to the function that generates these assembly tables. This also requires the modeller to add corresponding objects, flows, and product variants to the model. This provides a challenge for the maintainability of the model as it puts limitations on adding new delivery locations.

4.4.1.1 Documentation

The documentation dimension in the organisational maintainability framework was also addressed. Here it was deemed important to document the models at different levels. The importance of documentation at different level was discussed by Oscarsson and Moris (2002). This was also highlighted by one of the project's stakeholders: the model should be documented so that the project can be presented and thought to be credible. It should also be documented with more detail so that future modellers can navigate it.

For low level users and developers, a guide was created due to the high abstraction level in facts. This includes flow charts and descriptions of processes, specifically how the physical process was translated to facts. The guide also documents the detail level of the model, assumptions made, how data was gathered and how the results can be read (Oscarsson & Moris, 2002). At the end of the project, this guide was written for the model, describing the function in detail for different objects: sources of data, assumptions made about the process, and the naming system used in the model. In addition to this the user and developer of the model were involved throughout the project and the final model was presented to them in great detail.

For stakeholders who have a higher level of interest in the project, a presentation with the highlights of the project was conducted. The results of the model were presented to stakeholders in a formal presentation. Each part of interest in the model was discussed and how it translates to reality. To increase belief in the model a 2D animation was shown and explained. However, FACTS Analyzer is not a very visual program, and therefore, flow diagrams were depicted to add details to the presentation. The results from different experiments were presented and the meaning of those results discussed.

4.4.2 Evaluation Based on Technical Criteria

During the model building process, the technical criteria was kept in mind, so that when possible arrangements were made to fulfill these criteria in the model. However, the dimension of choosing a suitable simulations software will not be discussed, as the software for the project was not chosen by the authors.

The model was built to represent the process closely. The detail level was chosen partially such that the model can be more easily understood. Especially the result of the model as was discussed in section 3.4. Choosing the bundle chassis number detail level also means that the model can be more easily validated, which is important to establish credibility for the model. In addition, the number of products circulating the system decreases and the model is greatly simplified when the highest detail level in the model is bundles not tubes.

As mentioned in section 3.5.4, some simplifications were made in the model. A detail that was deemed unnecessary in the model was to include sub-assemblies that are separate from the assembly line but consume tubes. It was assumed that the sub-assemblies produce at takt-time and are never late. To confirm this, process experts were consulted and it was confirmed that this detail was unnecessary. By excluding unnecessary details from the model it is less complicated, and according to Piplani and Pauh (2004) a model containing more detail is more difficult to maintain.

Another method used in this project to increase the understandability of the model was to divide the model into different modular areas based on the function of the each, as previously discussed in section 3.5.2. The modules are clearly marked within the model with a background image. This makes the model easier to navigate. Constructs within the model were placed in the same order as in reality. In addition to this the naming convention for product variants is clear and organised to represent the physical process. Names of structures are also descriptive of the physical process. Templates in the model were re-used as much as possible and all templates follow the same naming convention and have a similar flow if possible.

The software used in the project has the option of uploading data from an Excel file. All process times and most assembly tables used in the model are included in that Excel file, such that changes can be easily managed within the model. Making changes in Excel allows the user to navigate the data faster and changing values becomes easier. This makes it easier to maintain the model. When changes are made to the model, the user has to have a good idea about how the data in the current state model is connected and what has to be updated to implement these changes. Some process time distributions in the model are determined based on historical data. In the case of the process time of cutting bundles, changes in tube demand will have an effect on the time distribution. When these values are updated it is done manually. The distributions have to be changed manually in the input data file, and that is only after a new process time distribution has been determined. This is far from being an automatic process and could be improved.

5

Discussion

In this chapter, the results from the method of the project is highlighted and suggestions into further work are presented.

5.1 Maintainability

Although DES has been widely researched, the area of DES model maintainability is an area that has fairly little literature available, although some examples exist (Williams, 2002). The subject simulation project methodologies and of what to include in a simulation model has been addressed in literature (Banks et al., 2010; Law, 2007). Simplification methods for DES models have also been addressed (Piplani & Pauh, 2004; Brooks & Tobias, 2000). However, no clear commonly accepted and established methodology for building a DES model for maintainability was found in the literature study conducted during this thesis. This thesis attempts to create a framework with organisational and technical criteria for modellers wishing to build a model for maintainability.

The frameworks for maintainability were developed based on the authors' experience during the project and information from a literature study. However, a limitation was applied to these maintainability frameworks. The frameworks were designed for simulation environments rather than all types of simulation softwares. This was because the authors determined that a framework for covering three types of simulation software: simulation environments, simulation programming language and general-purpose programming languages, would not be as useful in any of the cases. A simulation environment was chosen since this project is conducted using FACTS Analyzer which is a simulation environment. Therefore, the authors have better insight into simulation environment softwares. In the case where a general-purpose programming language or a simulation programming language is used, a different framework may prove more useful. In the case of a general-purpose programming language being used for DES, a framework for software maintainability may be a better fit.

Early on in the project, it was discussed that the goal was to build a model that can be maintained as the system changes. The authors had previous experience building a DES model. Both authors had previous experience building a DES model in a simulation environment using a software called AutoMod, and one of the authors also had experience using Anylogic (Banks et al., 2010). Therefore, some ideas about the outcome of the project were formed early on and some criteria were not surprising to the authors. For example, the scoping of the model was done with great detail from the start of the project to limit the complexity of the model. The model started out with a lower detail level and as the project progressed further detail was added. As stated in section 4.3.3 choosing an appropriate detail level in the model ended up being one of the criteria in the technical maintainability framework. The same can be said for the criteria for using good modelling practices, although the sub-criteria of this is largely based on the literature study. However, some criteria developed during the project were not expected. From the technical maintainability framework, the criteria for choosing a suitable simulation software was one of those. During the project, the authors debated if the software used in this project fit the project and compared this to their previous experience. During the project, the authors also saw different use cases for FACTS Analyzer where the system in question was much more suited to FACTS Analyzer.

In the organisational maintainability framework, two criteria address the in-house knowledge required to maintain the model. The maintainability framework argues for both a need for In-house DES knowledge and In-house process knowledge. However, the impact of these two criteria combined is uncertain, i.e. whether this consists of one or two individuals. It is unclear what would be the ideal combination for this, but both are required to maintain a model. The authors reflected on the fact that a process without DES knowledge might lead to the model being discontinued, while a model expert without process expertise might not be able to model accurately.

With this project focusing on a building technical and organisational framework for maintainability, the focus is not on when a model should be built with maintainability in mind. As mentioned before, the subject of software maintainability is not a highly researched subject. This can be because DES models are often used to solve specific problems within manufacturing, such as improving manufacturing performance, optimising transportation systems, and layout planning (Banks et al., 2010). If a DES model is built to solve a specific problem it may not be beneficial to design the model for maintainability. Rather making the model maintainable may only make the modelling process more time consuming without adding any value. However, if there is value in repeatedly analysing a system it should be considered that building a DES model is time consuming and expensive for a company (Johansson et al., 2003). An alternative to building a new model could then be to build a model intended for long-term use with maintainability in mind. However, this is something that has to be decided for each case.

5.2 The Simulation Model

Throughout the project, a lot of decisions were made regarding the model and here some of these decisions will be discussed. As discussed in section 3.5.2.4 the assembly line was built in the model. It consists of 43 assembly stations, which is a lot of content to add to the model. Since the focus of the model is the tube cutting station. This was done so that the delivery locations could be modelled at the assembly line. However, this requires adding over 50 objects to the model and the data associated with those objects. For example, all the different delivery locations have 214 assembly tables, one for each chassis combination. There either a dummy variant or Sequence_Delivery location variant are assembled onto the truck variant. A dummy load is required so that the assembly line within the model keeps moving. This feature adds a lot of detail to the model and makes the animations more understandable. However, it can be that this makes the model less maintainable, as it complicates the model. Piplani and Pauh (2004) stated that a model containing more detail is more difficult to maintain. This choice can be criticised but was modelled so that the simulation model could more closely resemble the physical process.

The process time distributions of the cutting machines are determined based on historical data, the process can be viewed in more detail in section 3.6.3. One criteria of the technical maintainability framework is automatic data input, which the process times distributions cannot be. Using the input data file values in the model can be updated but the process times distributions have to be determined separately. If the same method is used again the historical data requires some processing before a function can be run to find the summary statistics. The data also has to be examined to determine what distribution is appropriate. This is something that could be further explored in other projects, but in this case, it was deemed outside of the scope due to the time constraint.

One criteria of the organisational maintainability framework is to choose a software that fits the project. In this project that was deemed outside the scope since the software had already been chosen for the project. However, the effect of using FACTS Analyzer can be debated. During this project the FACTS Analyzer Professional version was used, not the developer edition. This version does not offer the option of coding and limits the model to the predetermined objects within the software. In some cases, this led to an increased number of objects and the complexity of the model. For example, to facilitate a pull production connected to the number of bundles per sequence or sequences per cart, objects and variants had to be matched within a template using assembly tables. This had to be repeated for each delivery location.

Furthermore, the breakdowns representing the material changes at the tube cutting station are modelled in a sub-optimal way. This was because during the experiments some runs would result in a very high failed portion for the machine. The sum of different breakdown states activity in a machine would not tally up to the reported amount for the simulation. The authors of this report are unsure why this happens, as it is not the intention. On a similar note, the breakdowns are caused by a need for material change is calculated according to current capacity. However, since the coil breakdown is based on the current state of chassis per shift, if the takt-time changes, production of chassis increases and as such the frequency of need for coil changing does as well. This was not able to be accurately modelled through variable distribution as these distributions are kept in their current state distribution.

5.3 Sustainability

Sustainability is often discussed from three different angles: economics, environment and social. This is a long-standing practice and allows a discussion of sustainability from multiple dimensions (Purvis, Mao, & Robinson, 2019). Here it will discuss how this project can support sustainability. Both the model itself and the two maintainability frameworks.

DES models are used within many different industries, including the manufacturing and automotive industry. It is a flexible tool that can be used to solve a wide array of problems, including bottleneck analysis layout planning, and optimisation of manufacturing and transportation systems. The model can consider many different measurements and provide a detailed analysis of the system being modelled. With a DES model, experiments can be performed on a computerised model instead of the physical model. This can benefit a company both from an economical perspective as well as saving time (Banks et al., 2010; Law, 2007).

The model in this project was built with multiple parameters as well as supporting functions for the tube cutting station and the assembly line. This was so that it can be used for optimisation and process identification strategies within several different improvement areas. The project also experimented with reducing the number of carts circulating the system as discussed in section 4.2.2. By reducing the number of carts the tube inventory is decreased and reduces the tied-up capital that could be reallocated elsewhere. Another perk of the optimisation is that by minimising the number of carts in circulation the number of tubes being produced ahead for a chassis is reduced. By reducing the number of tubes, the risk of tubes being mishandled or scrapped if changes are made at the line are decreased.

In this project, two frameworks for model maintainability are proposed. These frameworks should support a model to be maintainable long term. Based on the authors' perception, DES models are commonly built for one-time use and as mentioned previously, creating a DES model is both time-consuming and costly. If one can build and maintain a DES model instead of creating one for ad hoc usages, better utilisation could result from the spent resources. In relation to this, a maintainable model can extend its credibility long-term when changes to the surrounding model are made. However, as discussed in 5.1, there needs to be a trade-off between the cost of maintaining a model and its construction. In some cases, the cost of maintaining a model might be considerably higher than the cost of creating and scrapping a model for a single purpose.

5.4 Future Research

While this report describes a framework for maintainability, it does not address the question of when it is advised to create a maintainable model, and when it should be designed and used for single use. Here the trade-off between the cost of building a new model later and the cost of building a model to be maintainable has to be considered. It is therefore suggested that future research should be to explore when a model should be constructed for upkeep and when to construct it for a single time usage.

One of the criteria in the technical maintainability framework is automated data handling. The model built in this project does not have fully automated data handling or an automated way to update the model if any changes are made to the production. FACTS Analyzer allows modellers to read values from data files to update values within the program. However, to use this feature to automate data handling in the model, the data file has to be built in such a way that it can be automatically updated. This is of particular interest with the process times distributions and the truck product variants circulating within the system. The data handling in this project was simplified with the use of a data input file but the values within that file have to be updated manually in some cases. This argues for further exploration into processes to automatically update the distributions within DES models or input data files.

6

Conclusion

In this thesis, a case study of model maintainability was conducted. A DES model of a pneumatic tube cutting station and processes that affect the cutting station was made. The model is built in such a way that different parameters of interest can be experimented with. The model includes the tube cutting station, the assembly line, logistical routes, and batch consumption area. At the assembly line, 17 different delivery locations are modelled and corresponding product variants were created.

At the start of this thesis, four research questions are posed and divided into two categories: model and DES model maintainability research questions. For the two research questions belonging to the model, experiments are designed and carried out. The model is used to analyse the current capacity at the station as well as the minimum number of carts possible in circulation for certain delivery locations, such that the station can still meet the assembly line demand.

Furthermore, for the two research questions on DES maintainability, this thesis also aimed to investigate support that can be applied to a DES model to enable upkeep of the model. To answer research questions about maintainability, a literature study was conducted as well as unstructured interviews during the project. Observations made during the project also influenced the results. Two maintainability frameworks were developed: A technical maintainability framework and organisational maintainability framework, each containing several criteria. The organisational maintainability framework highlights external support that can enable maintainability in a simulation model. This includes documentation of the model, having a schedule for model review, and in-house knowledge both in terms of DES modelling and process knowledge. The technical maintainability framework describes internal support that can be applied throughout the modelling process to support maintainability of that model in the future. The criteria in this framework are: The model should be understandable and clearly visualise the physical process, built at an appropriate detail level to answer all questions while not including unnecessary details, choosing a simulation software that fits the problem, using automated data input as well as good modelling practices throughout the model. These two frameworks are not interchangeable and to achieve the highest level of maintainability both should be applied.

References

Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2010). *Discrete event system simulation* (5th ed.). Pearson Education Inc.

Bidanda, B. M. (2023). *Maynard's industrial and systems engineering handbook* (6th ed.). McGraw-Hill.

Bokrantz, J., Skoogh, A., Lämkull, D., Atieh, H., & Perera, T. (2018). Data quality problems in discrete event simulation of manufacturing operations. *Simulation: Transactions of the Society for Modeling and Simulation Internationa*, *94*(11), 1-17.

Boysen, N., Emde, S., Hoeck, M., & Kauderer, M. (2015). Part logistics in the automotive industry: Decision problems, literature review and research agenda. *European Journal of Operational Research*, *242*(1), 107-120.

Brooks, R. J., & Tobias, A. M. (2000). Simplification in the simulation of manufacturing systems. *International Journal of Production Research*, *38*(5), 1009-1027. doi: 10.1080/002075400188997

Bryson, J. M. (2004). What to do when stakeholders matter. *Public Management Review*, *6*(1), 21–53. doi: DOI:10.1080/14719030410001675722

Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data science journal*, *14*. doi: http://doi.org/10.5334/dsj-2015-002

Christa, S., Madhusudhan, V., Suma, V., & Rao, J. J. (2017). Software maintenance: From the perspective of effort and cost requirement. In S. C. Satapathy, V. Bhateja, & A. Joshi (Eds.), *Proceedings of the international conference on data engineering and communication technology* (pp. 759–768). Singapore: Springer Singapore.

Ebert, C., Cain, J., Antoniol, G., Counsell, S., & Laplante, P. (2016). Cyclomatic complexity. *IEEE Software*, *33*(6), 27-29. doi: 10.1109/MS.2016.147

Gupta, S., & Gayathri, N. (2022). Study of the software development life cycle and the function of testing. In *2022 international interdisciplinary humanitarian conference for sustainability (iihc)* (p. 1270-1275). doi: 10.1109/IIHC55949.2022.10060231

Han, J., Pei, J., & Tong, H. (2011). *Data mining: concepts and techniques.* Morgan kaufmann.

Heričko, T., & Šumak, B. (2023). Exploring maintainability index variants for software maintainability measurement in object-oriented systems. *Applied Sciences*, *13*(5). doi: 10.3390/app13052972

IBM. (n.d.). *Cyclomatic complexity.* Retrieved from `https://www.ibm.com/docs/en/raa/6.1?topic=metrics-cyclomatic-complexity`

Ieee standard glossary of software engineering terminology. (1990). *IEEE Std 610.12-1990*, 1-84. doi: 10.1109/IEEESTD.1990.101064

Jepsen, A. L., & Eskerod, P. (2009). Stakeholder analysis in projects: Challenges in using current guidelines in the real world. *International Journal of Project Management*, *27*(4), 335-343. doi: https://doi.org/10.1016/j.ijproman.2008.04.002

Johansson, B., Johansson, J., & Kinnander, A. (2003). Information structure to support discrete event simulation in manufacturing systems. *Proceedings of the 2003 Winter Simulation Conference*, *2*, 1290-1295. doi: 10.1109/WSC.2003.1261564

Jonsson, P., & Mattson, S.-A. (2009). *Manufacturing planning and control.* McGraw-Hill.

Law, A. M. (2007). *Simulation modeling & analysis* (4th ed.). McGraw-Hill.

Mahanti, R. (2019). *Data quality : Dimensions, measurement, strategy, management, and governance.* ASQ Quality Press.

Morgenthaler, S. (2009). Exploratory data analysis. *WIREs Computational Statistics*, *1*. doi: https://doi.org/10.1002/wics.2

Nance, R. E. (1996). *A history of discrete event simulation programming languages.* New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/234286.1057822`

Oscarsson, J., & Moris, M. (2002). *Documentation of discrete event simulation models for manufacturing system life cycle simulation* (Vol. 2). doi: 10.1109/WSC.2002.1166359

Perera, T., & Liyanage, K. (2000). Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. *Simulation Practice and Theory*, *7*(7), 645 - 656. doi: https://doi.org/10.1016/S0928-4869(99)00020-8

Piplani, R., & Pauh, S. A. (2004). *Simplification strategies for simulation models of semiconductor facilities.* doi: https://doi.org/10.1108/17410380410555862

Purvis, B., Mao, Y., & Robinson, D. (2019). Three pillars of sustainability: in search of conceptual origins. *Sustainability science*, *14*, 681–695.

Robinson, S., & Bhatia, V. (1995). Secrets of successful simulation projects. *Proceedings of the 27th Conference on Winter Simulation*, 61–67. Retrieved from `https://doi.org/10.1145/224401.224424` doi: 10.1145/224401.224424

Sargent, R. G. (1981). *An assessment procedure and a set of criteria for use in the evaluation of 'computerized models and computer-based modelling tools'* (Tech. Rep.). SYRACUSE UNIV NY DEPT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH. Retrieved from `https://apps.dtic.mil/sti/pdfs/ADA098785.pdf`

Sargent, R. G. (2010). Verification and validation of simulation models. *Proceedings of the 2010 Winter Simulation Conference*, 166-183. doi: 10.1109/WSC.2010.5679166

Schnappinger, M., Fietzke, A., & Pretschner, A. (2020). *Defining a software maintainability dataset: Collecting, aggregating and analysing expert evaluations of software maintainability.* 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME). doi: 10.1109/ICSME46990.2020.00035

Skoogh, A., & Johansson, B. (2007). *Time-consumption analysis of input data activities in discrete event simulation projects.*

Wang, J., Ge, J., & Lu, Q. (2012). *A review of stakeholder analysis* (Vol. 2). International Conference on System Science, Engineering Design and Manufacturing Informatization. doi: 10.1109/ICSSEM.2012.6340802

Williams, E. (2002, 01). *Achieving maintainability of process simulation software models.*

A

Appendix A

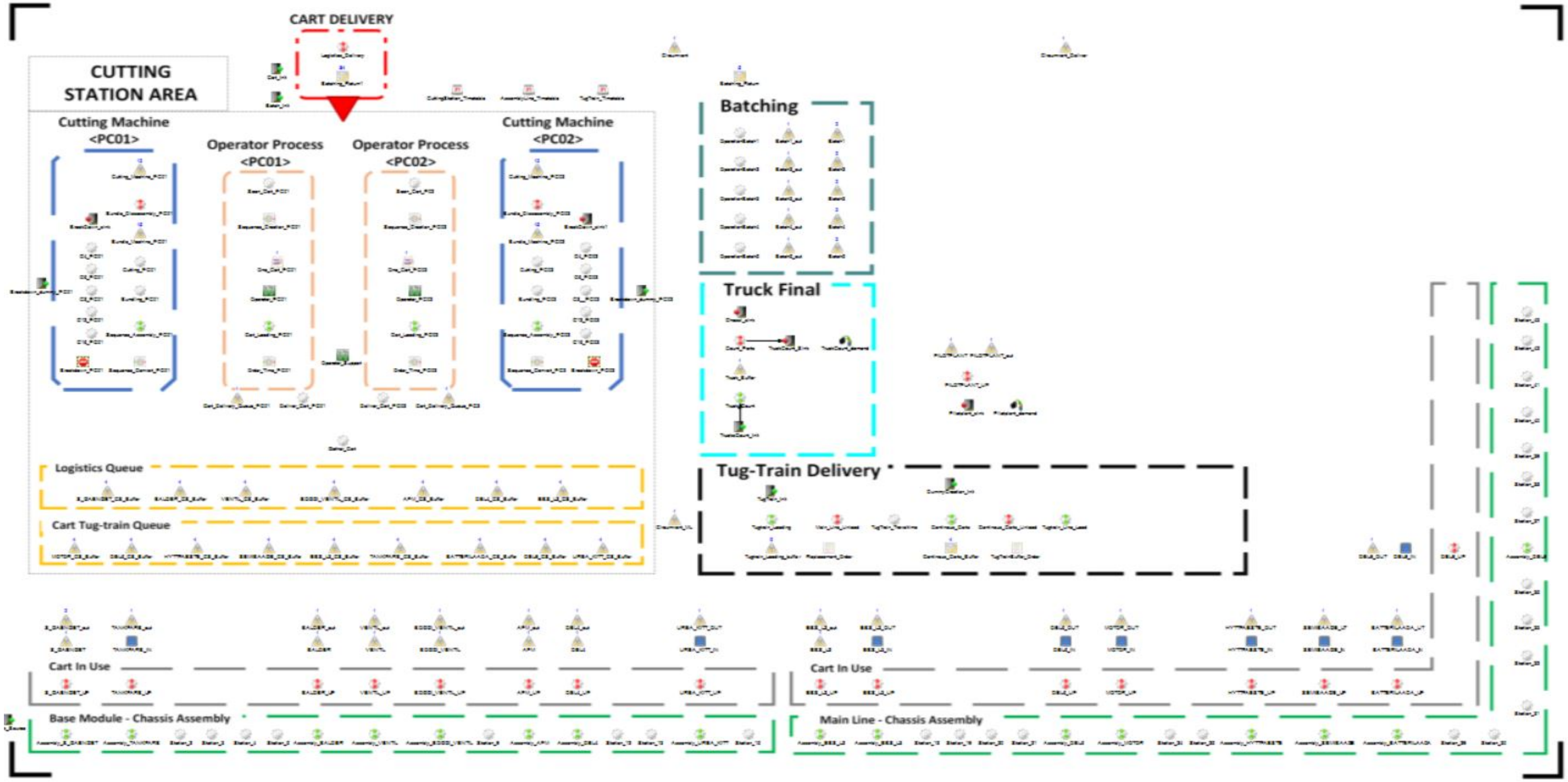

Figure A.1: The whole model as modelled in FACTS Analyzer

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se