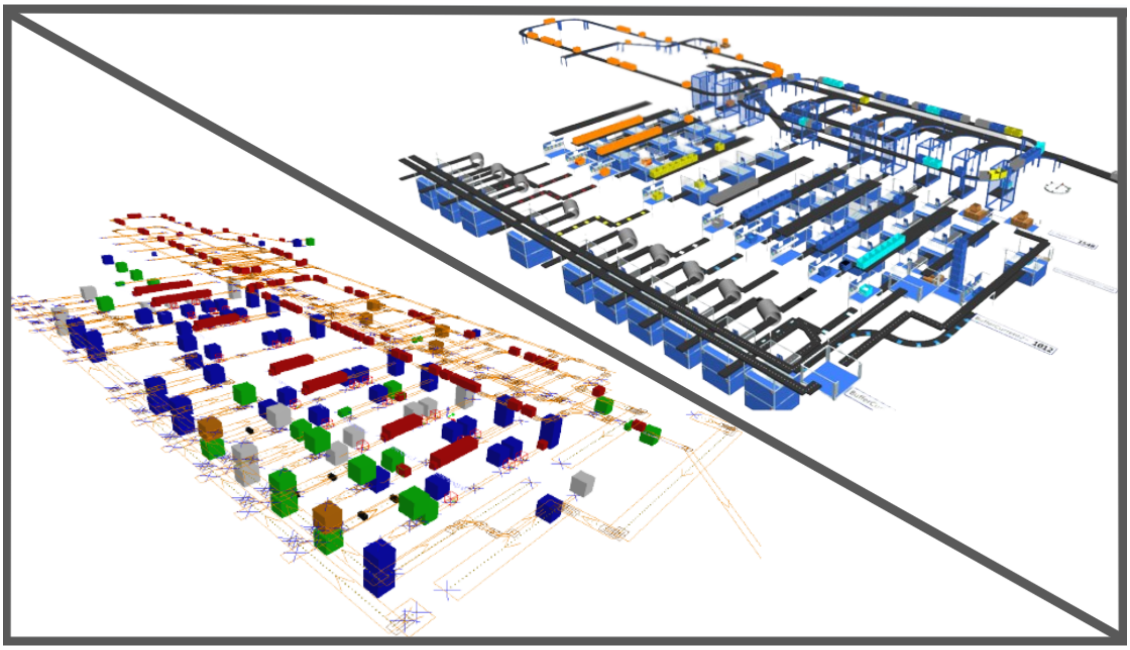
CHALMERS
UNIVERSITY OF TECHNOLOGY



Generalised or specialised simulation?

Comparing types of Discrete-event simulation software and possibilities for increasing the value proposition

Master's thesis in Production Engineering

MARTIN ERIKSSON
ALEXANDER FERNING

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

Master's thesis 2023

Generalised or specialised simulation?

Comparing types of Discrete-event simulation software and possibilities for increasing the value proposition

Martin Eriksson
Alexander Ferning

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

Generalised or specialised simulation?
Comparing types of Discrete-event simulation software and possibilities for increasing the value proposition
Martin Eriksson
Alexander Ferning

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: Use case modelled in AutoMod and Plant Simulation

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Generalised or specialised simulation?
Comparing types of Discrete-event simulation software and possibilities for increasing the value proposition
Martin Eriksson, Alexander Ferning
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Discrete-event simulation has emerged as a powerful tool for analysing dynamic systems. Numerous software companies have developed their own software solutions with different niches, strengths and weaknesses. Users, however, tend to keep using the same software, no matter what they will model. This master's thesis compares and assesses general and specialised DES software on numerous parameters. An industrial use case was modelled, supported by interviews and surveys conducted with participants from academia and industry. An existing model was recreated using specialised software, based on the provided model specifications. The throughput observed in both models was similar, with differences attributed to different underlying model assumptions. Both software types have their advantages and disadvantages, and it is important for users to be aware of these before selecting which one to use. Each software was deemed reliable alternatives for usage in a professional setting. Two different model-building approaches were identified between the software users. A resource-oriented top-down approach was more common in the specialised, while a load-oriented bottom-up approach was more common in the generalised software. Through interviews and a literature review, potential use cases to increase the value proposition of DES were explored and competencies, challenges, enabling technologies and architecture were identified. The interviews revealed some professional scepticism regarding adopting a complete Digital Twin approach, utilizing the DES model as a simulator within the Digital Twin. A promising alternative, Digital Shadow, was identified as an attractive alternative. Potential scenarios, and how they could be implemented were suggested using dynamic production planning.

Keywords: Digital Twin, Discrete-event simulation, Digital Shadow, Comparison of simulation software, Generalised simulation software, Specialised simulation software

Acknowledgements

This master's thesis marks the end of our studies at Chalmers Tekniska Högskola within Production Engineering.

We would like to take this time to express our sincere gratitude to the entire Supply Chain Management department at AFRY Göteborg for allowing us to conduct this research. We are particularly grateful to the simulation team at the department for their invaluable support and guidance throughout the course of this thesis.

We would like to extend a special thanks to Niklas Palm for his persistence, assistance and expertise. We are grateful for his patience, willingness to teach and valuable insights that greatly contributed to the success of this project.

We would also like to acknowledge and thank all interview participants, our academic supervisor Siyuan Chen and the examiner Anders Skoogh at Chalmers. Your feedback and insights into Simulation and Digital Twins have been of great value to the project.

Martin Eriksson, Alexander Ferning, Gothenburg, June 2023

List of Acronyms

API	Application Programming Interface
DES	Discrete-event simulation
DM	Digital Model
DS	Digital Shadow
DT	Digital Twin
IoT	Internet of Things
MES	Manufacturing Execution System
MTTF	Mean time to failure
MTTR	Mean time to repair
MU	Manufacturing Unit
PLC	Programmable logic controller
RQ	Research question
SQ	Sub question

Contents

List of Figures

List of Tables

1
Introduction

This chapter presents the background and context of the research topic, which is focused on the applications of Discrete-event simulation and Digital Twin in an industrial context. The main objectives, research questions and delimitation of the project are outlined. The chapter concludes with the delimitation which establishes the scope and limitations of the project.

1.1 Background

Simulations of production systems have widely gained in popularity since Julian Reitman's somewhat groundbreaking article "Simulation of a manufacturing system" from 1967 where the author explained how to model a semiconductor system with computer simulation techniques [1]. Today, the use of Discrete-event simulation (DES) is regarded as one of the most powerful tools available for decision support, with benefits in planning, designing and improving material flows [2].

The implication of the rising usage of simulation in production systems has transformed the development process; where ideas and changes in production lines can be tested before they are implemented. This yields less investment risk and costs savings [3] for companies successfully implementing simulations that model reality well enough.

Through DES, manufacturing bottlenecks can be identified using strategies such as the utilization detection method and average waiting time detection method [4] which can increase the production system's throughput. In an ever-increasing competitive industrial market, DES has due to its benefits taken place as one of the most popular modelling techniques [5].

This increase in usage has naturally led to competitors within the software industry providing their software, with individual features and interfaces, to satisfy the customer's needs in creating DES models. Today there are several DES software available on the market; both general-purpose DES software (software that is used in a wide-spread of industries) and specialised DES simulation software (software more directly targeted toward one industry).

Software tends to be sticky [6]; meaning that once a user (or company) has learned one software they tend to keep using it. The customer retention rate is therefore high and users might be hesitant to change DES software even though better options for a specific use case might exist. As Abraham Maslow said [7]

> "If the only tool you have is a hammer, everything looks like a nail"

The selection of software for a production system is influenced by several factors such as the availability of object-oriented resources, statistical characteristics, automatic data collection and visual features [8]. The selected software might affect the simulation model's accuracy, utility and building time [9]. Selecting a less appropriate software might therefore increase costs which in turn lowers the profit margin or in the worst cases, even might produce deceptive results. Being able to tackle simulation problems with knowledge of multiple software might therefore be beneficial.

With the increasing demand for simulation software, research has been conducted to create frameworks for choosing the right software for different use cases [10], [8], [11], [12]. These frameworks have in some cases been implemented to evaluate different simulation software against each other [13], [14]; resulting in a better choice of simulation software for the user's particular needs. However, there seems to exist a gap in the literature regarding the comparison between general-purpose simulation software and specialised simulation software.

While software selection is an important consideration, it is also important for companies to be aware of emerging technologies that can transform simulation capabilities. Digital Twins (DT) are described as the next wave in simulation [15]. DTs bridge the gap between real systems data and the DES model; bringing the model from "offline" (based on historical data) to "online" (based on real-time data). Using-real time data can provide more accurate simulations and models that better reflect the current state of the system, leading to improved planning and decision-making.

Research has to some extent explored the integration of DES-models in DTs [16]. The continuous stream of real-time data increases the value proposition of using DES models since it becomes possible to simulate more phases of the system. But, it is still not clear how well DES models perform as a simulator in a DT. It is also to some extent uncertain what competencies and technologies are needed for successfully implementing DTs.

1.2 Purpose

The purpose of this research is twofold. Firstly, it addresses the existing gap in the literature by conducting a comparative analysis between general-purpose simulation software and specialised simulation software. By evaluating both software types through a real-world industrial use case, this study seeks to get insights into the advantages and limitations of each. This analysis will assist users and companies in making informed choices when selecting simulation software.

Secondly, this research explores the potential of using DES models within DTs and the integration of real-time data. Additionally, it aims to identify the competencies and technologies required for successfully implementing DTs. By addressing these aspects, this research aims to contribute to the knowledge of simulation software selection and shed light on the possibilities and limitations of utilizing DES in the context of DTs.

1.3 Project aim

The thesis aims to compare and evaluate specialised and generalised DES software through a real-world use case from the food industry. A factory model will be recreated and programmed in the specialised DES software, while we will leverage our previous experience with the general software. The two models will be compared and rated based on a set of criteria that covers different simulation features. The assessment will enable us to identify the strengths and weaknesses of each software type. Our findings will be supported by interviews and surveys made with professionals from both academia as well as from industry.

Also, this research will investigate the potential of DES as a simulator in DT. To do so, we will examine the necessary competencies, enabling technologies and assess the challenges that may arise through a literature study. Furthermore, the possibility of the software establishing a connection to gather production data and what value this could bring will also be explored.

1.4 Research questions

The research questions (RQ) have served as the project's guiding framework. Each research question has been further refined into detailed sub-questions (SQ) to provide a more comprehensive understanding of the research problem.

RQ 1: How does the model construction and simulation process differ between general and specialised Discrete-event simulation software?

SQ 1.1: How do the user interface and tools in generalised and specialised DES software support the model construction and simulation process?

SQ 1.2: How does the model complexity affect the model construction and simulation process in a generalised and specialised DES software?

SQ 1.3: How does the level of detail in output generated by generalised DES software compare to that of specialised DES software in an industrial use case?

RQ 2: What are the key competencies and technologies for using Discrete-event simulation in the context of Digital Twin technology?

SQ 2.1: How can these be applied to the use case to increase the value proposition of DES?

SQ 2.2: What are the main challenges when implementing Digital Twin technology?

1.5 AFRY

AFRY has a global presence with offices in over 40 counties. The company is one of the market leaders in engineering, design and advisory services in Europe [17]. AFRY is a merger between the former companies ÅF and Pöyry and employs over 19000 employees globally with net sales of about SEK 20,000 Million [18].

Within the Supply Chain Management department at the Gothenburg office, there is an experienced simulation team that to a large extent, although not exclusively, is working with the DES software AutoMod. They help their local and international customers with DES cases within a wide range of industries but especially thrive in simulations of production, warehouse and logistics (internal and external). Customer cases include both greenfield- as well as brownfield projects.

1.6 Delimitation

To investigate the differences between general-purpose and specialised DES software, AutoMod will be used as a representative of general-purpose software [19] [20] and Plant Simulation as a representative of specialised software.

AutoMod was chosen for this study due to its widespread use across a diverse range of industries and sectors, including but not limited to healthcare, logistics, ticket offices and production [21]. Further, it essentially provides the users with a blank slate, allowing for a great deal of flexibility and adaptability to different sectors and use cases. Plant Simulation was selected since it is designed specifically for modelling and simulating production and logistics [22]. Moreover, there exists a library of resources and objects that are to a great extent used in production and logistics [23].

Additionally, the thesis is subject to the following limitations:

- Studies on the physically existing use case will not take place.
- Assumptions will be made if available data is insufficient.
- Sensitive corporate data will not be disclosed.
- The AutoMod model was deemed to be an accurate representation of reality, and therefore, the model in Plant Simulation will be compared to the AutoMod model rather than the actual system.
- The research will not include any optimization or improvement analysis of the use case.
- Version 22.01 of Plant Simulation and Version 12.6 of AutoMod will be used.
- No Digital Twin will be developed for the use case.
- The information presented in the thesis may sometimes be based on subjective observations and analysis of the researchers, and should not be accepted as a universal truth.

1.7 Report outline

The first chapter serves as an introduction to the thesis. This contains background to the topic, why it is of interest, and the goal of the thesis. The research questions together with the sub-questions are outlined. Further, a brief introduction to the company at which the master's thesis is carried out is given. The chapter concludes with delimitation. The second chapter contains the theory of the master's thesis. The literature study and the supporting theoretical background for the thesis work are presented.

The third chapter explains the methods used in the thesis and is divided into six parts. The first part describes how the literature study is done and is used to support the theoretical background. The second part describes how the qualitative study was carried out through interviews. The third part contains the quantitative study and explains the simulation experiments that were carried out in both software for the use case. The fourth part described how Bank's methodology was applied to the use case. The fifth part gives a brief introduction to the use case. The last part provides information regarding how the evaluation of the DES software was carried out through the use of surveys.

The fourth chapter presents the results. The findings from the qualitative, quantitative, and comparative studies are presented. Further, the results from the survey are presented. The fifth chapter consists of a discussion of the results, together with an analysis of the project and used methods. Uncertainties and doubts about the outcomes and methods used are addressed. The sixth chapter concludes the thesis and recommendations for future work are presented.

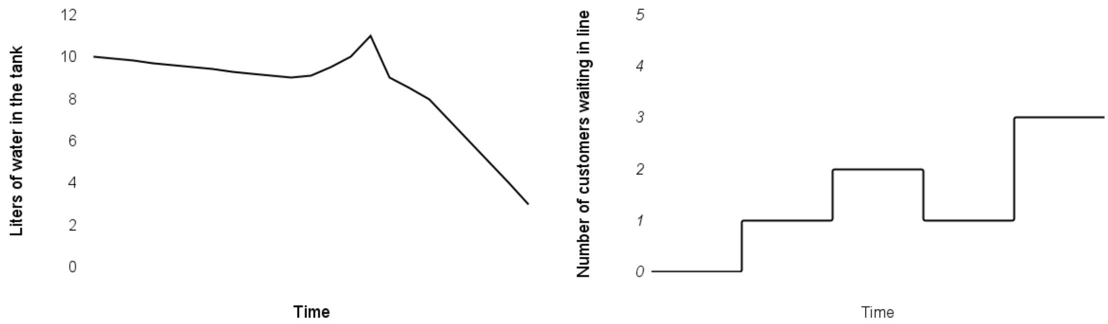
2
Theory

In this chapter, relevant theories, literature and concepts will be reviewed to establish a foundation for the study. This will include theories, models and existing literature relevant to the research questions. The chapter's goal is to help the reader grasp the theoretical basics and connect the thesis to the wider context of the field.

2.1 Discrete-event simulation

Discrete-event simulation (DES) is a powerful technique used to model and analyze complex systems that evolve over time due to specific events [9]. This technique represents a system as a series of discrete transactions or loads which competes for limited resources such as machinery while moving between different locations. In a DES model, events are scheduled to occur at specific time points and represent the progress of transactions within the system [24]. These events can include the arrival of a new load, the completion of a task or the release of a resource.

In contrast to time-continuous simulation models where time flows continuously[25], DES models represent time as a sequence of discrete steps. Each event that occurs and changes the system state is represented as a discrete step, and the system's behaviour changes over time in response to these events. This results in step functions that represent the dynamic nature of the system being modelled. Figure 2.1 illustrates the difference between a time-continuous simulation model and a DES model.



(a) Continous-system state variable **(b)** Discrete-system state variable

Figure 2.1: Visualization of time-continuous models and discrete-event models, freely redrawn [25].

The use of DES allows for the modelling of dynamic systems over time, making it possible to evaluate different scenarios and identify potential bottlenecks or areas for improvement [9]. DES can also be used to estimate the performance of a system under various conditions, such as changes in demand or when investing in new machinery. DES is particularly useful in industries such as production, healthcare, logistics, defence, the service sector and traffic simulations [9].

One of the main advantages of DES is its ability to mimic the randomness of the real world, allowing for a more accurate representation of the system being modelled. However, as with real-world systems, there is an inherent uncertainty in the outcomes of these simulations. This is why DES is often stochastic [26], involving random processes or events. To account for this uncertainty, it is necessary to run these simulations multiple times with the probability distributions of the parameters. One run is simply not enough to draw a reliable conclusion.

The simulation is implemented using a time-stepping algorithm, where the simulation advances in small time increments and the system's state is updated at each time step based on the events that have occurred [9]. Thus, nothing is assumed to have happened between these states. In comparison to standard programming, in DES, the code is not executed in the way it is written, instead, loads can jump and execute different segments of the code. This is implemented with the help of an event scheduler which decides which event to execute next and at what time.

2.1.1 AutoMod

AutoMod is a DES software developed by Applied Materials, Inc. The program allows for a wide range of DES applications, from material handling to healthcare systems. The software has several main capabilities including the ability to reuse model objects [21], which reduces the time required to build a model.

AutoMod has a modular program structure. The central module is the process module, which is then supplemented by additional modules such as AutoStat (a tool for statistical analysis and optimization), AutoView (a tool for animation), Conveyor (Material handling technology) and Kinematics. Further, model communication modules make it possible to connect to third-party software. These capabilities make AutoMod a valuable tool for simulating and analyzing complex systems [27].

AutoMod can be used to forecast outcomes and run “what-if” scenarios in a simulation environment. This provides a basis for analyzing and improving potential system design ideas. Additionally, current systems can be analyzed for continuous improvements [21]. The AutoMod programming language is based on structured action statements in the English language. As stated by M.Roher in 1997, the AutoMod model logic can be as big or complex as the user want, because AutoMod models are in no way constrained[28].

2.1.2 Plant Simulation

Plant Simulation is a software tool developed by Siemens Industries that focuses on production and logistics systems using DES [13]. The software offers integrated tools for statistical analysis, optimization and interactive animation in both 2D as well as in 3D.

Plant Simulation has a library of pre-built objects, such as conveyors, assembly tables and pick and place robots, which easily can be added to models by drag-and-drop. This reduces the time and effort required to build models [29]. External CAD files can also be imported and exchanged with the pre-built objects. This allows the virtual model to closely resemble the physical system, which leads to increased model interpretability [29].

The programming language used in Plant Simulation is SimTalk 2.0. The software offers the possibility of programming via methods, with available in-built templates to simplify complex logic for the user. Furthermore, the software can be connected with other software through several built-in interfaces [29]. Besides English, Plant Simulation is also available in Chinese, German, Japanese, Hungarian and Russian.

2.1.3 Sustainability

The three categories of sustainability are economic, environmental and social aspects [30] as shown in Figure 2.2. All three categories are working together towards a sustainable society and need to be kept in mind by organisations.

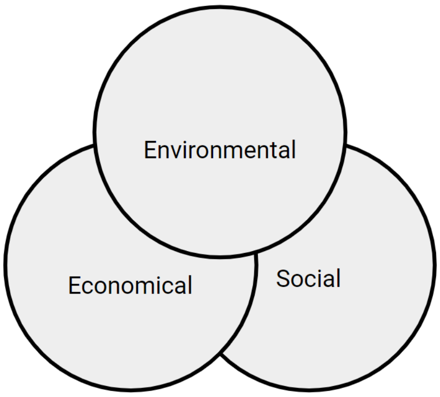
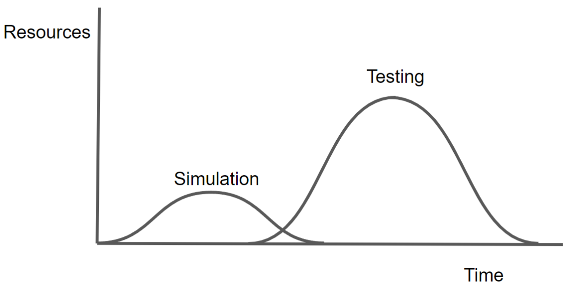


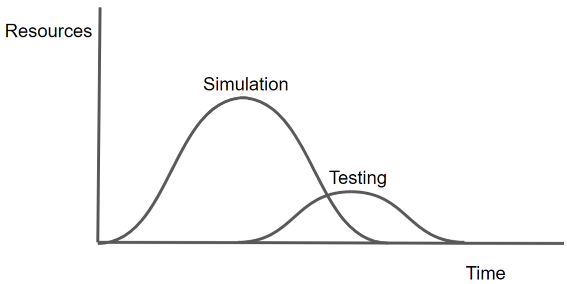
Figure 2.2: Categories of sustainability.

DES can be used to improve all of these aspects in companies through optimization. Companies that implement DES successfully can get higher returns while investing fewer resources [31]. Dabrowska and Grzybowska showed how simulating the supply chain could lead to cost and emission exhaust decreases through consolidation of transportation [32]. Vorst et al. modelled the food supply chain in their use case and compared costs, emissions and energy consumption through DES [33]. The use of DES can make it possible for companies to make informed and data-driven decisions regarding sustainability.

Since changes can be tried and tested virtually before implementing them in the physical world, time and resources can be saved as shown in Figure 2.3. Simulation can help project teams to identify potential issues earlier in the process, which can help to avoid costly errors and save time.



(a) Normal project



(b) Simulation-focused project

Figure 2.3: The benefits of using simulation tools during the development phase in a project, freely redrawn [34].

2.2 Comparison of Discrete-event simulation tools

For evaluating simulation software, Nikoukaran et al. suggested a hierarchical approach [10]. According to their study, there are seven key groups that companies need to keep in mind when choosing between different simulation software. These were the Vendor, Model and Input, Execution, Animation, Testing & Efficiency, Output and User, as shown in Figure 2.4

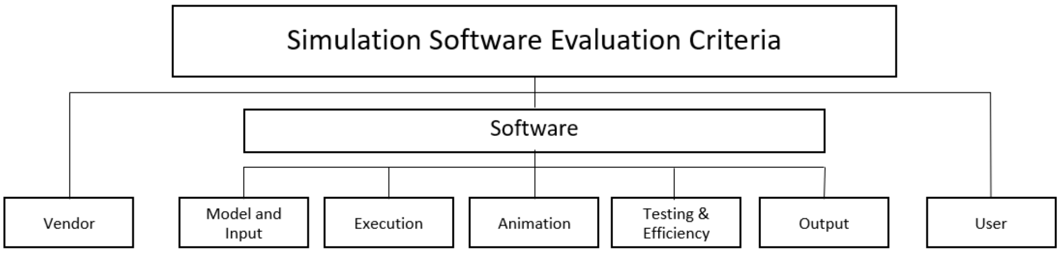


Figure 2.4: Hierarchical simulation software evaluation, freely redrawn [10].

There were divisions of subcategories within each category. The authors recommended selecting the subcategories that were of importance to the user and thereafter weighting the categories according to their perceived value. Further, the author suggested that the hierarchy may provide a better view of software possibilities.

An implementation of the hierarchical model was made by Pezzotta et al. but for evaluation and assessment of two DES software for service engineering [13]. The software Arena and Plant Simulation were compared to determine which was more appropriate for usage in the service setting. The authors discovered that both software systems were quite flexible, but that Plant Simulation's product-orientation focus, attention to statistics, failure management and set-up times made the model building a bit more difficult (and not as relevant in the service environment).

Guimaraes et al. also proposed detailing and weighting components that are important to the specific company [8]. Their implementations also made use of a hierarchical structure for the decision-making process, but they instead made a pairwise comparison of the software. They ranked the software on a scale from 1-10 and then took the individual company's preferences into account for finding suitable software.

Zhong and Zhao choose a somewhat more direct approach in their thesis “A comparison of simulation tools for supply chain management” [14]. They compared two use cases in the simulation tools Arena and AnyLogic focusing on simulation results and building process but with parameters based on earlier written literature. The proposed evaluation parameters were: Capabilities, visualization, simulation efficiency, result accuracy and debugging.

Another way of evaluating DES software is a framework proposed by Banks. The evaluation model contains a set of input, processing, output, environment and cost features[11]. The author proposes to use a scoring model that ranks each parameter from zero to ten, then each score is summed up and normalized to 100 which is the factor weight. The parameters will be ranked once again from zero to one and then multiplied with the factor weight to get the weighted score. The software with the highest total weighted score is considered a potential candidate for the user.

Evaluation criteria can also be classified for each simulation software feature. Hlupic et al. proposed a methodology for selecting simulation software [12]. For each simulation feature, a set of relevant evaluation criteria is investigated and classified. The simulation features presented in the paper are general features, visual aspects, coding, efficiency, modelling assistance, testability, software compatibility, input/output, experimentation facilities, statistical facilities, user support, financial and technical features and pedigree. All criteria were not suitable for a quantitative ranking and thus classifying them was deemed more appropriate. Examples given in the paper are classifying as possible/ not possible or low/medium/high. The user has to be mindful that each criterion and classification is subjective and should fit the specific use case.

2.3 Digital Twin

Digital Twins (DT) is an area of interest that is growing at a rapid pace, both in the academic environment [35] as well as in the industry [36] [37]. It is considered a main building block for smart factories and manufacturing in Industry 4.0 [38]. Rosen et al. stated that DT is the next phase in modelling, simulation and optimization technology [39]. This finding was strengthened by Tao and Zhang who showed that the utilization of DT by those in decision-making positions is an unavoidable trend [40] in the future.

Grieves originally proposed the concept of DT (now known as an early and not sufficiently precise description) in 2002 [41] during one of his presentations regarding product lifecycle management. It consisted of three components: the physical product (real space), the virtual product (virtual space) and the connection of data and information between the spaces, [42], as visualised in Figure 2.5.

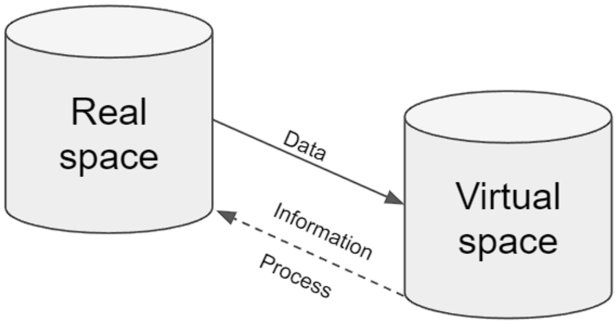


Figure 2.5: Grieves originally proposed Digital Twin, freely redrawn [41].

Singh et al. proposed the following elaborated definition of a Digital Twin:

"A Digital Twin is a dynamic and self-evolving digital/virtual model or simulation of a real-life subject or object (part, machine, process, human, etc.) representing the exact state of its physical twin at any given point of time via exchanging the real-time data as well as keeping the historical data. It is not just the Digital Twin which mimics the physical twin, but any changes in the Digital Twin are mimicked by the physical twin too [43]"

A Digital Twin's operational dynamics are thus predicated on real-time synchronization with its physical counterpart, which is the primary distinction between a DT and a simulation model. Instead of focusing on what might happen (what-if scenarios), which is the case for simulation, DT instead focus on what is happening right now (what-now scenarios). The DT can thus increase the level of confidence in decision-making compared with simulation models.

2.3.1 Digital Model, Digital Shadows and Digital Twins

Kritzinger et al. exemplified the somewhat insufficient description originally made by Grieves in 2002. The authors made distinctions between a Digital Model (DM), Digital Shadow (DS) and Digital Twin (DT). These concepts were oftentimes used interchangeably, even though, there are differences between them [44]. The authors clarified the differences between the three concepts to provide a clearer understanding of their unique features and to provide a common ground when discussing the subject. These concepts have been visualised in Figure 2.6

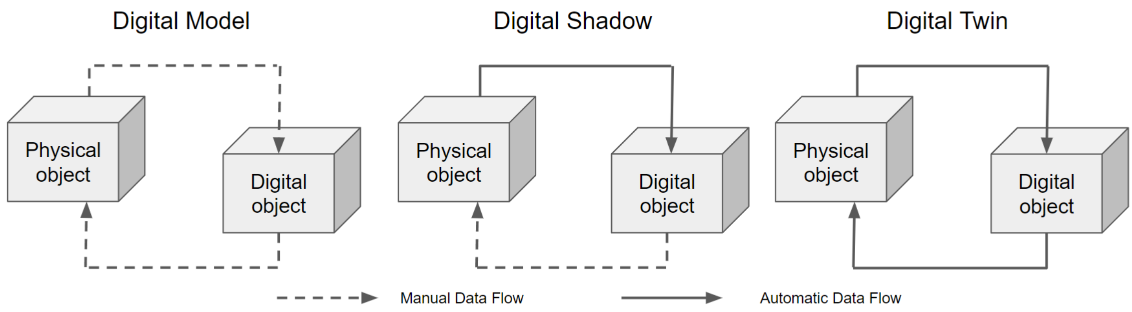


Figure 2.6: Data flow in Digital Model, Digital Shadow, Digital Twin freely redrawn [44].

According to the authors, a DT is a digital representation of an existing physical object, where data flows in both ways and is integrated. Therefore, any change in the physical object would be reflected in the digital object. Similarly, any change in the digital object would also be reflected in the physical object. This can be compared to a DS, which has a one-way data flow from the physical object to the digital object, but not from the digital object to the physical object. Therefore, a change in the physical object would affect the digital object, but a change in the digital object would not affect the physical object.

Lastly, the authors described a DM as a digital representation of a physical object that is either already existing or is planned to exist. However, in contrast to both a DS and DT, an automated data exchange does not occur at all for a DM. Therefore, a change in the physical object would not affect the digital object. Similarly, any change in the digital object would not affect the physical object. This is typically how DES models are built.

2.3.2 Maturity Levels

Another way to classify differences in Digital Twins and to ensure that the same type of DT is discussed is by looking at their abilities and functions. Madni et al. proposed dividing Digital Twins into four different maturity levels [45], dependent on their abilities and level of depth. This has been summarised in Figure 2.7.

The Pre-Digital Twin is a simulation model and the initial level of the four. It can be thought of as a virtual prototype. At this level, there is no data exchange between the physical system and the simulation model. The Pre-Digital Twin supports decision-making at the early stages.

The required software, connection protocols and hardware are implemented at the second level. These are necessary for data transmission between the simulation model and the real system. The simulation model also needs a data link for receiving and transmitting data. Predictive maintenance is a typical instance where the DT is of use. The bi-directional interaction makes it possible to improve performance during real-time operation.

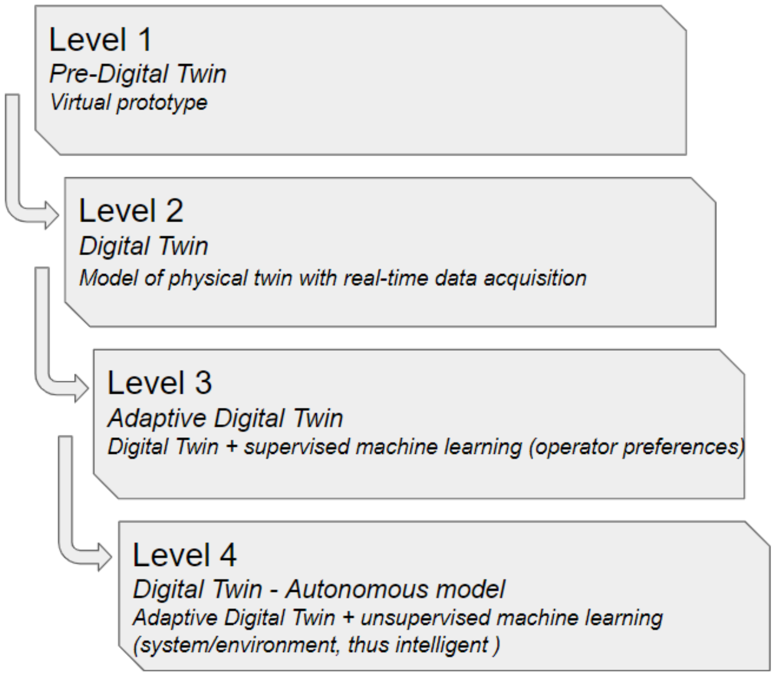


Figure 2.7: Maturity levels of Digital Twins, freely redrawn [45].

In the third level, Adaptive Digital Twin, data is continuously retrieved from the physical model and transmitted in real-time (and vice versa). At this level, the Digital Twin also recognizes operator preferences and priorities. Through machine learning, the Digital Twin adopts its parameters in accordance.

All of the third level's skills are present at the fourth level, named Intelligent Digital Twin, but it also contains unsupervised machine learning capabilities for the operational setting. It may therefore learn to distinguish items and recognize patterns in its surrounding. At this stage, the DT can examine the real-world counterpart's performance, maintenance and health data in more detail. It also has a higher degree of autonomy compared to previous levels.

2.3.3 Architecture and enabling technologies

To connect the DM with the physical world, Internet of Things (IoT) devices are used in factory settings. This enables sensor updates and the exchange of real-time data, which enables the simulation of the behaviour of physical systems in a digital environment [46]. By continuously feeding data into the system, it becomes possible to predict dynamic changes and adapt the physical system based on the simulation results. With the increasing availability and affordability of IoT devices, Digital Twin technology has become more cost-effective to use [45].

The architecture of a Digital Twin consists of several key technologies, which have been illustrated in Figure 2.8. These technologies enable the key feature of DT to exchange data and information [47].

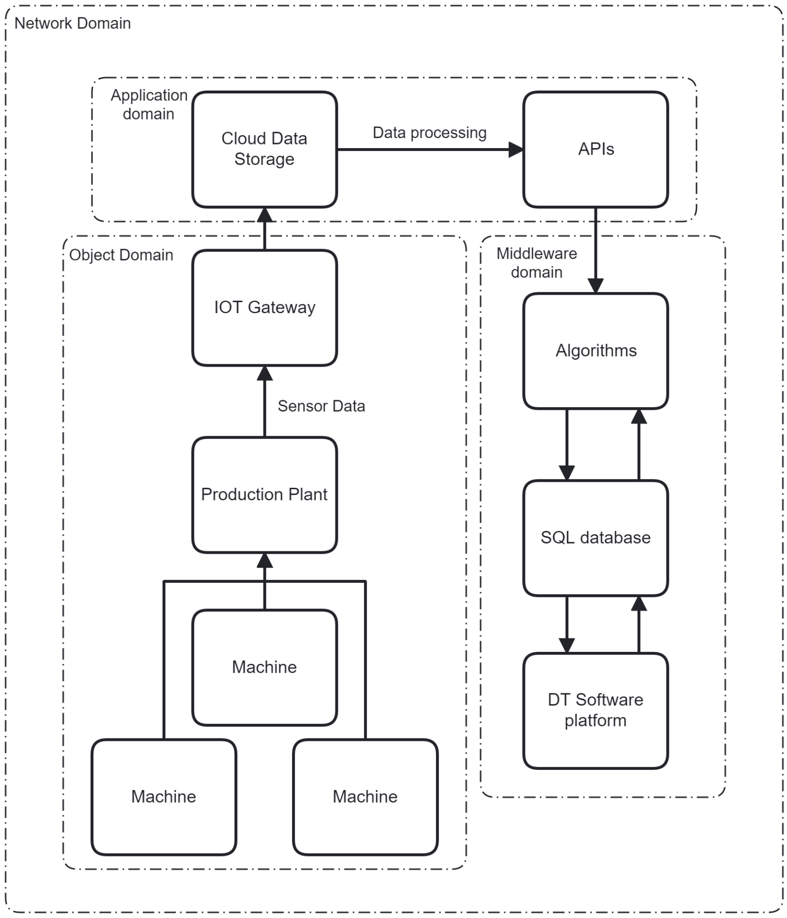


Figure 2.8: Sketch of architecture for enabling technologies, freely redrawn[47].

The DT architecture framework is proposed by Fuller et al. and is divided into the application domain, middleware domain, networking domain and object domain [48].

The application domain includes model architecture and visualization, software and APIs, data collection and pre-processing. To ensure accuracy and minimize errors, the virtual model must accurately mirror the physical features and behaviour of the physical model [42]. Software and APIs bridge the gap between virtual and physical models, facilitating data exchange and collection. Real-world data collection often comes with noise and missing data, making data pre-processing an essential step to improve data quality, performance and accuracy [49].

The middleware domain is responsible for storage technology and data processing. Through data modelling, data are stored according to specific criteria and logic, which facilitates efficient data processing [42]. The DT must handle a large amount of continuous data, leading to a high demand for computation power and data processing. High-performance data processing helps connect diverse data streams with the DT information model effectively [50]. Data processing methods are used to extract features from the data, reduce noise and select quality data.

The networking domain includes communication technology. Communication networks enable the establishment of DTs. State synchronization between a DT and its physical counterpart relies on bi-directional and real-time data communication [50]. Wireless communication is an effective method of transferring data from IoT devices to data storage, which is later used in data processing.

The object domain consists of the hardware platform and sensor technology. The DT infrastructure requires reliable hardware to handle all integrated software and information flow. To ensure data quality, robust sensors with reliable technology are needed. If sensors repeatedly provide incorrect data, the DT will repeatedly fail to represent the physical system accurately.

2.3.4 Key challenges

Implementing a fully working and reliable DT poses several challenges that need to be addressed. Fuller et al. introduced the main challenges industries face[48].

The first challenge is to establish a reliable IT infrastructure. The simulation model must have a reliable connection with the physical system through the infrastructure. The hardware and software requirements must meet the requirements to ensure smooth system operations. The 5G network, with its higher data transfer rates, lower latency and higher reliability than previous generations [51] can be of great value in the IT infrastructure. However, 5G operates at a higher frequency than 4G, which reduces signal wavelength and limits signal penetration quality [51].

The next challenge is to collect reliable and useful data, as a simulation result is only as trustworthy as its input data quality[52]. Without trustworthy data, the DT may fail to represent the real system and meet its intended purpose. Without proper validation and verification, a DT may behave in unexpected ways, leading to incorrect decisions and negative impacts on organisational outcomes. Therefore, a thorough validation and verification process needs to be established. This ensures that the DT operates as intended and produces reliable data that can be trusted.

In this context, the value of a sensor lies in its proper utilization, as it can become a mere expense otherwise. Raw data has limited meaning on its own. However, when processed and transformed into valuable insights, it gains importance. The true value of data is realized when it is used to achieve the desired outcomes, as illustrated in Figure 2.9.

Uhlemann et al. highlighted how data acquisition in production systems consumes a significant amount of time. Automated data collection is made possible by the digitization of the production system. However, even though time and resources would have been freed, steps towards fully automated data acquisition were not extensive among small and medium enterprises [54].

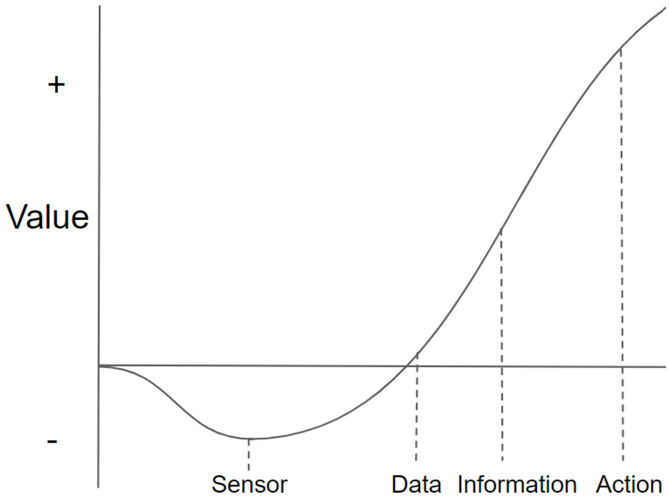


Figure 2.9: Value of Data, freely redrawn [53].

Another challenge with the use of DT technology is regarding security and privacy issues, as a large amount of sensitive data is continuously streamed and processed. This data must be protected through the implementation of robust security protocols and safety barriers to prevent potential leaks or third-party attacks. The manufacturing sector is facing increased cyber threats due to a lack of investment in cyber security [55]. This has resulted in cyber threats evolving from targeting traditional systems such as computers, networks, and smartphones to also targeting the manufacturing sector [55].

Lastly, DT involves various stakeholders and there are challenges with getting their views to align with each other [56]. Transparency, traceability and version control [56] are important to consider when it comes to future-proofing the DT.

2.3.5 Competences

Simulation has developed over the past few decades; from a technology utilized by computer and numerical experts to a tool used by engineers to answer certain design and engineering challenges [39]. This transformation is primarily due to advancements in computer hardware and software, which have made simulation technology more accessible and user-friendly [39].

Similarly, DT technology is undergoing rapid development, and it could as well eventually become a standard tool used by engineers. But given how quickly DT technology is developing, there is a need for continuous learning and competency improvement to fully leverage the use of DT. To properly manage and enhance DT systems, professionals who work with them must stay current on new advances and continuously hone their abilities.

Despite the need for automated data transmission, as described in Chapter 2.3, there still is a need for human expertise that can interpret, handle and maintain the

DT. The key competence required for this is described by Fantini et al. [57]. The authors propose five main competencies.

"Detect" highlights the significance of situational awareness and the ability to identify any problems or anomalies with the operation of the DT. "Determine" refers to the requirement for analytical abilities and the ability to make informed decisions based on the data generated by the DT. "Develop" refers to task execution and technical expertise required to implement adjustments and enhancements to the DT. "Describe" contain competencies for communicating to stakeholders the performance and insights of the DT.

Competencies that may not directly relate to the technical aspects of maintaining a DT but nonetheless are crucial for assuring its success are included under "Other". The requirement for cooperation abilities is one example of this. Building and maintaining a DT involves many different stakeholders, from engineers to data scientists, and effective communication among these groups is essential to ensure the success of the DT.

Kumar and Jagadeeswar, have developed, as they state, lifelong learning materials to improve competencies in the field of DT technology [56]. Their approach involves developing three laboratory materials with concepts and knowledge currently used in the industry, focusing on both technical competencies and cognitive skills. By continuously improving competencies, professionals can ensure that DT technology is effectively implemented and maintained [56].

2.4 DES environment in DT

DES models do not follow a uniform time distribution which complicates the framework of real-time simulation in a Digital Twin [58]. State changes that occur with events are what constitute a discrete model. To achieve real-time monitoring, the simulation time needs to be synchronized to the real-time.

Obermaier et al. provided an example in flight simulations, where one input in the simulation responds as it would in the real world [59]. Aligning the simulation to real-time is done by either slowing down or speeding up the simulation. If the input versus response is too slow or too fast, the simulation will become unrealistic [59]. One approach to synchronise the simulation time to the real-time is to apply precision time protocol (PTA) which can achieve a time accuracy within microseconds [58]. This is especially important when the data is to be transmitted from the physical system to the virtual system and vice versa.

Morabito et al. explained how a DES-based approach could be implemented in DT and verified it through two use cases. A simulator is a prerequisite to building the DT. The simulator must be able to update itself when and if the real system changes autonomously [16]. Due to DES models being based on historical data, they are in their original form not suitable to act as simulators in DT. However, by using the DES model as a simulator and then connecting the DES with its physical counterpart, it was possible to use it during the production phase. Thus, by the implementation of an additional software layer, the DES model could be connected to the physical shop floor.

An automated trigger mechanism detected irregular behavior on the physical shop floor. Data preprocessing identified and extracted the actual functioning of the machinery. Additionally, a simulation was fed and executed through an interface, eliminating the need for human involvement. The implication was that the DT could forecast production performance, and act as a constructive decision-support tool. Further, it could compare scenarios to optimize the configuration of the system.

An article by Sakr et al. suggested an integrated approach and building aspects for a DT based on a DES model where real-time data was implemented [60]. The authors discussed the drawbacks of the conventional DES model (historical data) and emphasized the requirement for real-time data utilization for DES models in order for them to be useful DS. Reconfiguration and state initialization were implemented through an integration layer that had been built into the system. The DT could rebuild the DES model and initiate its state to resume its main function. These features gave the simulation the flexibility and reliability that DT demands.

To create a DT platform of patient paths in a hospital, Karakra et al. proposed integrating DES with IoT devices [61]. The patients were required to wear RFID tags. Then with the use of sensors, and set rules for start and end events each time the tags entered or exited a virtual space, the movement of a patient corresponded to an activity. The DES model's input data corresponds to the beginning and ending events. The status of the DES models could be updated in real-time to be in sync with the outside environment based on these events connected to each action of the patient pathways.

Another framework is proposed by Ricondo, et al. [47], where a DM could be transformed into a DS with the use of Industry 4.0 technologies and the use of DES software. The proposed framework was applied to a railway axles production use case. The states of machines in the real factory could sequentially be updated in a simulation environment. Traditionally the factory had built DM iterative and made analyses based on historical data. The use of a DS improved traceability, current state gathering and model input data estimation. The author's conclusion was that the use of DS improves performance, equipment information and business intelligence capabilities [47].

3
Methods

This chapter describes the different methodologies used to collect data in the thesis, which includes a literature review, qualitative and quantitative studies, and an evaluation of DES in an industrial context. Further, a brief introduction to the use case is given.

3.1 Literature review

Literature from numerous scientific databases was searched using Chalmers discovery services to ensure high reliability of the sources. The methodology used for the literature study can be seen in Table 3.1.

Table 3.1: Literature review methodology.

Literature review methodology	
Database	ScienceDirect, IEEE Xplore, Scopus and Google Scholar
Information source	Scientific articles, case studies and books
Search keywords	"Discrete event simulation", "DES", "Digital twin", "Evaluation of software", "Comparison", "State-of-the-art"
Publishing year	DES: from 1980, Digital Twin: from 2004, State-of-the-art: from 2018
Screening procedure	Determining the content relevance by reading the abstract, introduction & conclusion of the information source. Then proceed to read the content of specific chapters.

The literature selection process was based on comparing and evaluating existing simulation software and state-of-the-art implementations of Discrete-event simulation. The process was carried out by:

1. Reviewing the title together with the abstracts.
2. Examine the conclusion and discussion.
3. Analyzing the literature content.

The paper would be accepted if it fits within the research parameters. Otherwise, it would not be included in the literature review. To ensure that the data obtained was relevant, the literature selection process focused on acquiring peer-reviewed studies that had been published as recently as possible, especially regarding state-of-the-art implementations. However, it was not deemed necessary for general information regarding DES since it is widespread.

3.2 Qualitative study: Interviews

The interview was semi-structured. Thus, there is a set of predetermined questions that will be asked. However, there was room for flexibility regarding topics and follow-up questions [62]. With semi-structured interviews, the answers to the questions tend to be more open-ended and there is room for a deeper elaboration on certain topics that the interviewee find interesting. The questions asked during the interview are presented in Table 3.2. The interviewees were additionally asked to complete a survey with more direct and opinion-based questions (see Chapter 3.6).

Table 3.2: Interview question and goals.

Interview questions	**Goal**
Can you explain your typical model-building approach and what are the key steps for a successful simulation model?	Find what capabilities is key for building a simulation model and identify how the model builder experience might differ in approach and how this might affect outcome.
How do you structure the programming in the simulation software?	Find out similarities, differences, level of ease and structure in Plant Simulation and AutoMod. Ex orderlists, variables, resources.
What kind of analysis or decisions support would you have wanted from a Digital Twin that you feel are lacking in DES?	Find out what benefits a higher level of simulation model can bring.
How do you make sure that the model behaves as intended?	Identify methods the person uses for controlling logic and model behavior. Ex studying graphics, following loads and using the debugger

The goal of the qualitative study is to complement the literature study with human experience and explore perspectives that might be lost in a scientific paper. Interviewing industry experts can lead to perspectives that complement what researchers publish. The following interviewees, presented in Table 3.3, are partaking in the interviews and have experience from either an academic or industry perspective.

Table 3.3: Interviewees description.

Role	**Software**	Experience
Simulation Specialist	AutoMod, Visual Components, Plant Simulation	1-3 years
Simulation Specialist	AutoMod, Visual Components, Plant Simulation, Extend sim, Simul8	3-5 years
Simulation Specialist	Arena, Plant Simulation, AutoMod	<1 year
Simulation Specialist	Plant Simulation, AutoMod	5-7 years
PhD Student	Plant Simulation	1-3 years
Key Partner Manager	Quest, Plant Simulation, FACTS Analyzer, AutoMod	>7 years
PhD Student	Plant Simulation, Facts Analyzer, Visual Components	1-3 years

Qualitative data collected from interviews can tend to be highly subjective to the person interviewed and thus the quality of the data needs to be investigated. This will be done by using normative, descriptive and prescriptive criteria [63]. These criteria can be outlined as follows:

1. Normative criteria: Researcher strives for minimal bias in data collection, unbiased generalisations and neutrality in interpretation. This is ensured by being logically and rationally consistent in decisions.
2. Descriptive criteria: Researcher describes the context of the study and explains the reasoning and thought process.
3. Prescriptive criteria: The study is performed in a structured and careful way which is a scientific approach to decision-making.

To extract valuable information from the interviews, coding principles will be applied. This is used to structure and categorize the answers. Coding principles are used to group and make sense of free text answers[64]. Coding the qualitative data will allow for a (more) objective way of analysing the data since the interviewer follow a set of rules in ways of grouping the answers.

The coding principle used will be deductive coding which is where you apply predetermined code to the qualitative data[64]. The code is constructed such that answers are grouped from themes that are of interest. This approach can save a lot of time but may be prone to bias since the interviewer is looking for answers that fit the predetermined areas of interest rather than exploring the data. The interviewer must therefore be mindful of considering all aspects and not only the ones that fit the area of interest.

3.3 Quantitative study

The study aims to compare the accuracy and performance of two DES models: one built using Plant Simulation and the other previously built by a working professional using AutoMod. The AutoMod model has previously been accepted by a company as a close representation of their actual production. Therefore, the AutoMod model will serve as a benchmark against which the accuracy of the Plant Simulation model will be measured. This comparison will help to identify the extent to which the Plant Simulation model can reproduce the results of the AutoMod model, and the factors that may contribute to any differences observed.

An identical production plan for a full week's production will be run in both Plant Simulation and AutoMod. The production plan contains 27 different orders that are to be produced at 9 active packing lines. The production plan contains, besides what type of order is to be produced at which packing line, also additional information. This can, for example, be the weight of one bag type for that order, how many bags fit in a box, box type, possibilities to support orders with their production and what type of cleaning & maintenance is carried out after the order has finished on the production line.

The production plan is designed to reflect real-world production scenarios and will involve running the simulation model with identical inputs, such as machine configurations, breakdowns and inventory levels. Through the experiment manager, five observations using different random number streams will be used to measure the cumulative throughput of produced boxes each day with and without failure rates in both models. This will provide the average value of the selected parameter sample. Some of the parameters investigated in the quantitative study will act as additional support in the evaluation process.

In addition, we will compare the execution times of the two software programs. Both models will be simulated without animations at maximum speed for the entire 7-day production plan, and the time taken to complete the simulation will be measured. It should be noted that minor variations in execution time between the models are expected and may be attributable to differences in code structure. Noticeable differences in execution time may indicate that one software takes a longer time to run a simulation compared to the other.

3.4 Simulation study

Banks et al. proposed a set of guidelines for constructing a simulation study [9], which are summarised in Figure 3.1. These steps provided a foundation for the development of the model, although only the first seven steps, marked in the figure, were carried out in this project.

Every simulation study should begin with a problem statement that is well-defined and understood by all of the participants. The problem formulation was the same as Research Question 1: How does model construction and simulation process differ from general simulation software compared to specialised simulation software?

The next step is to set objectives and overall project plan. The objectives indicate the questions that the simulation should address. For the case, the objects were the same as the sub-questions related to Research Question 1. The scenario that was assessed to be followed was a production plan provided by the company for one week.

The model conceptualization comes next. This lays the groundwork for conversations and demonstrates the logical connections between model items. The model conceptualization for the simulation can be seen in Chapter 3.5.

The next step is data collection. In a manufacturing environment, common input data are cycle times, resource availability, Mean-time-to-fail (MTTF), Mean-time-to-repair (MTTR) and scrap rates. All necessary data in the project were provided by the company through a model specification.

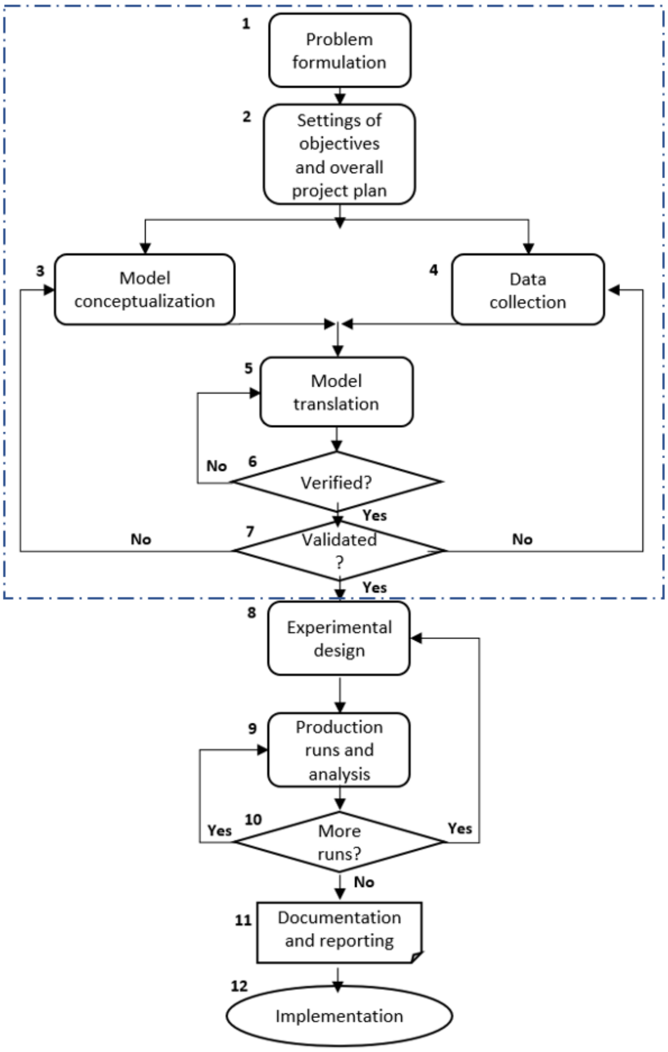


Figure 3.1: Banks proposed main steps for building a simulation model, freely redrawn [9].

After the data has been collected, the real-world system has to be translated into a computer-recognizable format (Model translation). This step includes choosing suitable software and coding the simulation model. This took the majority of the time in this project and ensured that we could answer the set problem statement together with the objectives.

Verified? Is used to ensure that each element in the model behaves properly. The verification techniques used in this project were cross-checking the code, studying the model animation, utilizing the debugger and allowing others to look at the code.

Validated? Is a comparison and quality check performed to ensure that the model (including its data) corresponds to the real world to an acceptable level. Validation took place through a walk-through and discussion of the model and results with the simulation specialist that built the previous model in AutoMod.

3.5 Introduction to use case

The thesis use case is an industrial customer case within the food industry. The use case has previously been simulated by AFRY in the software AutoMod. An overview of this model can be seen in Figure 3.2. Worth noting is that some areas of the factory that were deemed irrelevant (activities to the left of the packing lines) will not be included in the Plant Simulation model and the corresponding analysis.

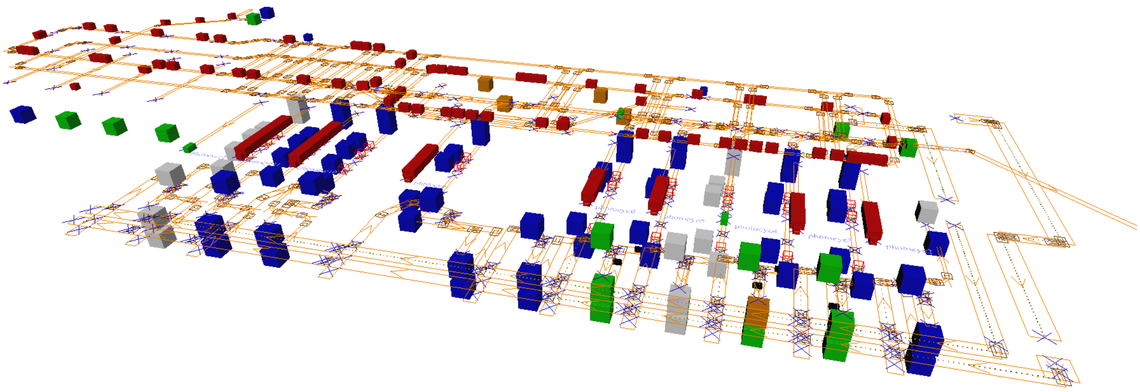


Figure 3.2: AutoMod reference model (previously built).

A conceptual model was created to visualize the material flow and logical relations between model entities based on the existing simulation model and the provided model specification. This has been visualised in Figure 3.3.

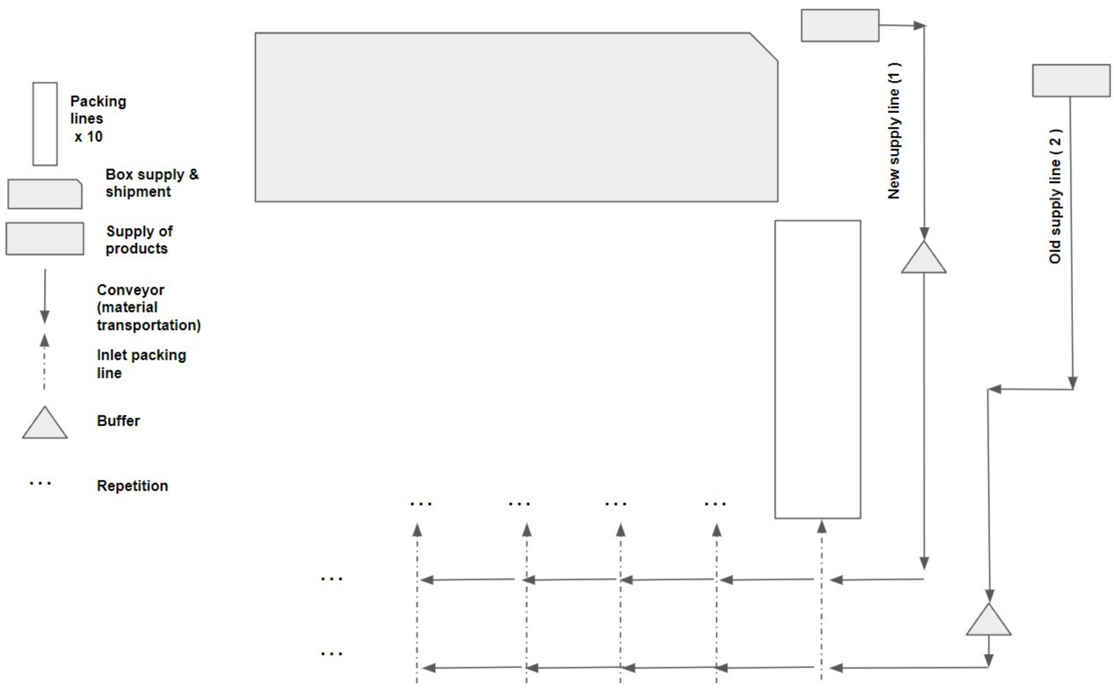


Figure 3.3: Conceptual model of factory and material flow.

The system illustrated in Figure 3.3 can be viewed as a black box, where the input and output of the model are simplified as the flow of products. Each activity consists of several sub-activities to be performed. The area within the packing lines in Figure 3.3 is of special interest. Figure 3.4 shows the activities in the packing lines.

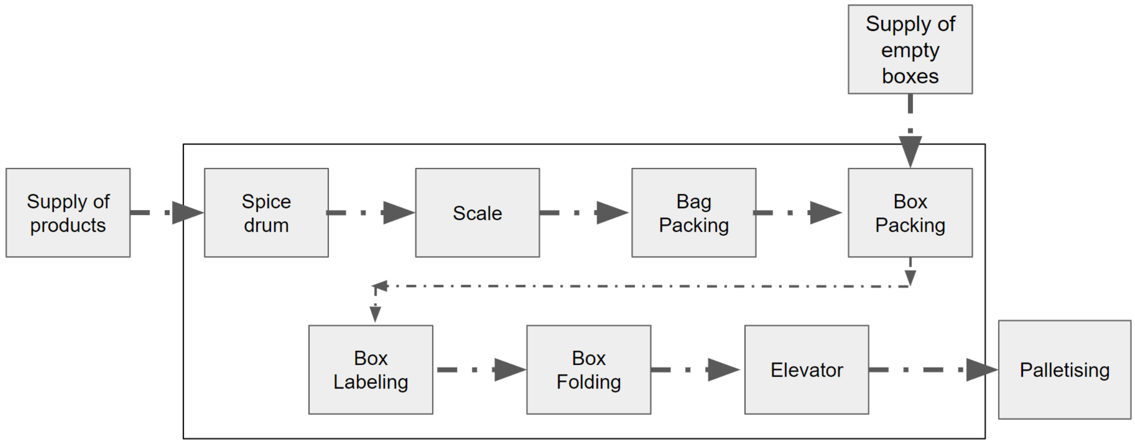


Figure 3.4: Activities in packing lines.

3.6 Evaluation of DES software

The literature review conducted in Chapter 2.2 revealed several evaluation criteria that are commonly used to assess simulation software. The following criteria have been selected to highlight potential differences between general and specialised simulation software.

- Model building
- Visualisation
- Accuracy
- In-built objects and pre-existing code templates
- Tutorials, documentation and model examples
- Coding and syntax
- Possibilities to integrate the model with external software
- Time and ease to learn
- Debugger

To determine the relative importance and ratings of these criteria, a complimentary survey to the interview was conducted. This survey was conducted with the same industry experts and researchers that participated in the interviews (which were described in Chapter 3.2).

The anonymous survey consisted of a set of questions asking the respondent to select what parameters that would be of special importance to look at when choosing a new simulation software. Further, they were asked to answer what parameters the software they are currently using is exceeding; and asked to rate each criterion on a scale of 1 to 5, where 1 represented "poor" and 5 represented "very good".

Based on the survey results, a weighting scheme for the evaluation criteria was developed using Analytical Hierarchal Process(AHP) [65], which is a common approach for decision-making. The multi-factor decisions making process is facilitated by weighting each factor and rating them to gain a collective evaluation of the different parameters. The weights represent the relative importance ratings provided by the survey respondents. A higher weight equals a larger impact on the evaluation, and therefore of more importance. The weights are then used to calculate an overall weighted total score for each simulation software, based on its performance across the different evaluation criteria.

The aim is to provide a comprehensive and objective evaluation of simulation tools, taking into account the perspectives and priorities of different user groups.

4
Results

This chapter presents the findings from the qualitative and quantitative studies, as well as the comparison and evaluation of the software. This includes a discussion of the main themes and patterns identified through the model-building process as well as any significant differences or similarities observed. The chapter continues by presenting the results of the software evaluation.

4.1 Qualitative study

The results from the qualitative study have been summarised in Figure 4.1

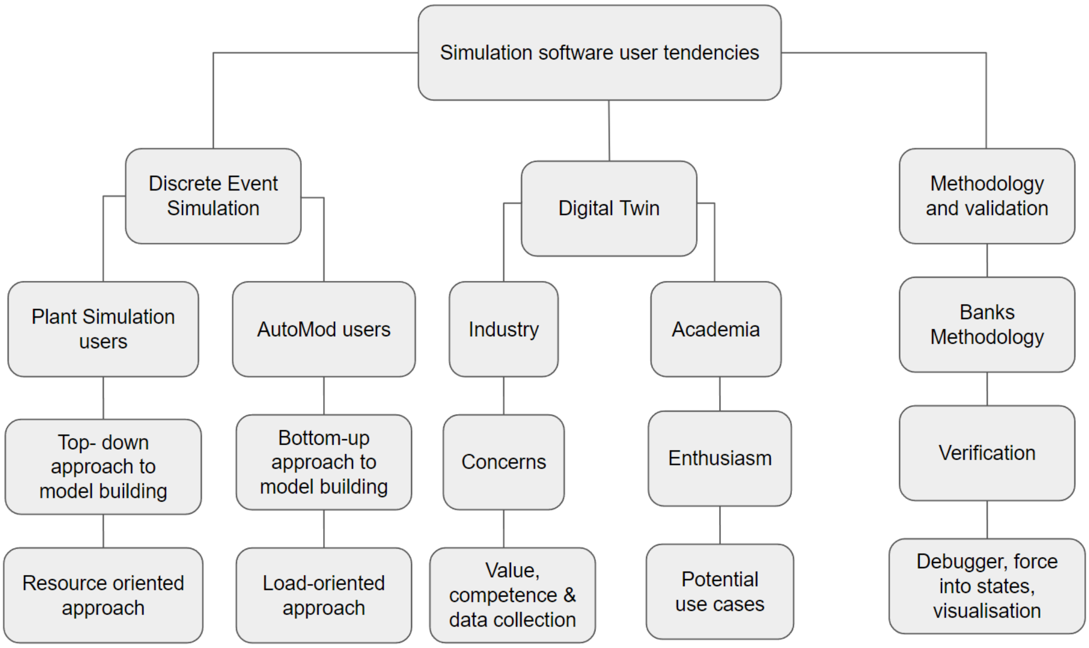


Figure 4.1: Summary of Qualitative study.

Based on the findings from the interview it was observed that there seemed to be a difference in the model-building approach for the two software. AutoMod users tended to use a bottom-up approach, breaking the model down into subsystems and programming, designing and testing each subsystem before integrating them into the overall system to form the model. *"I break it down into subsystems"*.

"I usually like to take a subsystem and complete it, such as an automated guided vehicle (AGV) system and then build it separately. Then, you can build the machine, conveyors or other systems. Smaller modules to build a larger one".

This could be compared to Plant Simulation users who described that they usually started with modelling the whole layout including material flow, and after that started programming logic at the different resources when it was needed.
"I just make sure that the production line and flow work as expected and then I will start to build some logic".

Further, what stood out from the interview was that AutoMod users tended to be more load oriented *"Loads determine everything"*.
They brought up that loads were seen as carriers of information. Multiple interviewees mentioned that they could design processes where a load only spins and only deals with information. *"I always assume that the function is to send loads and that they are the carriers of information"*.

All interview objects, no matter if they were from academia or industry, shared similarities when it came to ensuring that the model logic behaved as expected. The main tools used were the debugger, the printer function and visually making sure the loads moved as expected. One of the interview objects also described rubber-duck debugging and forcing the model into different states to be able to make sure to find potential errors.

During the interviews conducted with professionals working with simulation, a noticeable pattern emerged where the majority expressed scepticism regarding the use of DES in DT. *"It would be very case specific, I think. In many cases, there may not be much interest"*. The main concerns seemed to centre around the need for a compelling customer case that justifies the extra costs of implementing DT technology. *"It may not be cost-effective"*.*"I am not sure this step from the DES model to the Digital Twin is useful"*.

Furthermore, interviewees expressed doubts about the feasibility of obtaining real-time data, as data collection is already a time-consuming process *"Customers always say they have the data, but when you ask for it, it usually takes time for them and is the wrong data"*. Further, production systems were described as generally not well-connected. *"It places demands on production systems, which are generally not well-connected. So that aspect is partially problematic."*

One interviewee expressed that they lack the necessary competence to give a proper answer *" Can I pass on that question? I'm a little too unfamiliar with a Digital Twin to answer that"*.

One professional suggested that a middle ground could be achieved by importing current data from the factory after a major breakdown then running the simulation model, optimizing it, and obtaining actionable results. Potential use cases for Digital Twin technology in production operations and planning were identified by the researchers.

"It provides different types of decision support, perhaps more focused on production operations. You have the ability to get decision support in real-time, perhaps during a shift change or planning, and to act on unforeseen events. You can use a tool to plan the nearest period, not 3-6 months in advance when you introduce a new product. This doesn't mean it's more important than other things, which may be more profitable, but there are other possibilities there.

4.2 Plant Simulation model

The use case (briefly introduced in Chapter 3.5) was built by modelling several subsystems in different frames and then merging these together into the overall model. The use case (modelled in Plant Simulation) can be seen in Figure 4.2.

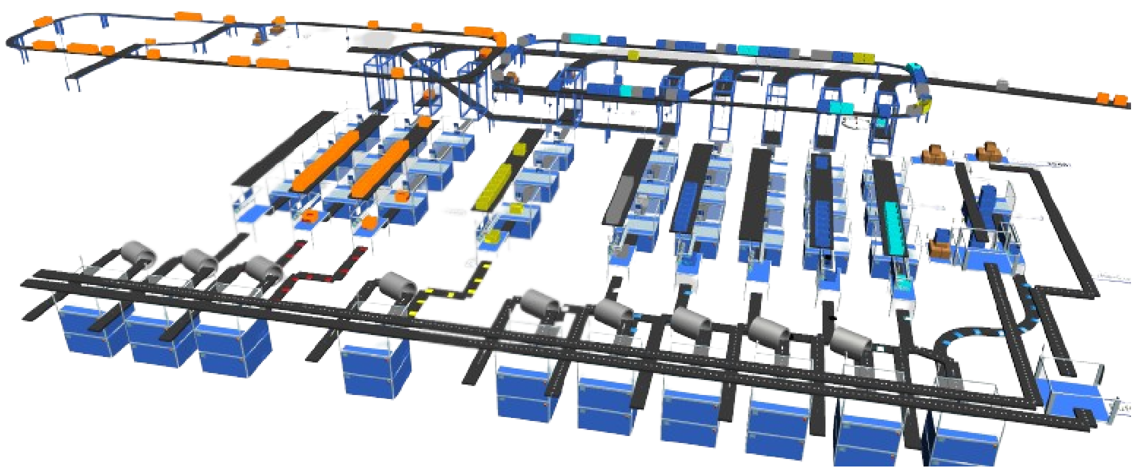


Figure 4.2: Plant Simulation model of the factory.

The model consists of three subsystems and each of these subsystems fulfils an important role in order to produce the finished products. Each system, its functionality and modelling assumptions are explained in the following sub-chapters.

4.2.1 Material supply

The first step in the model is the creation and supply of manufacturing units (MUs). The model simplifies the entire process, resulting in a continuous outflow of products with time gaps between the creation of these MUs. If the packing lines are not able to keep up with production lines MUs are stored in the buffers. As the number of loads stored in the buffers reaches a predefined threshold, the time gap between load creation increases, leading to a reduction in material supply capacity. When the buffers are completely filled, the creation of MUs is paused until the buffer level decreases to a specified level, at which point the creation process resumes. Worth noting is that the change in load creation time has a delay and is not executed immediately. There are two separate production lines that supply the packing lines, as seen in Figure 4.3.

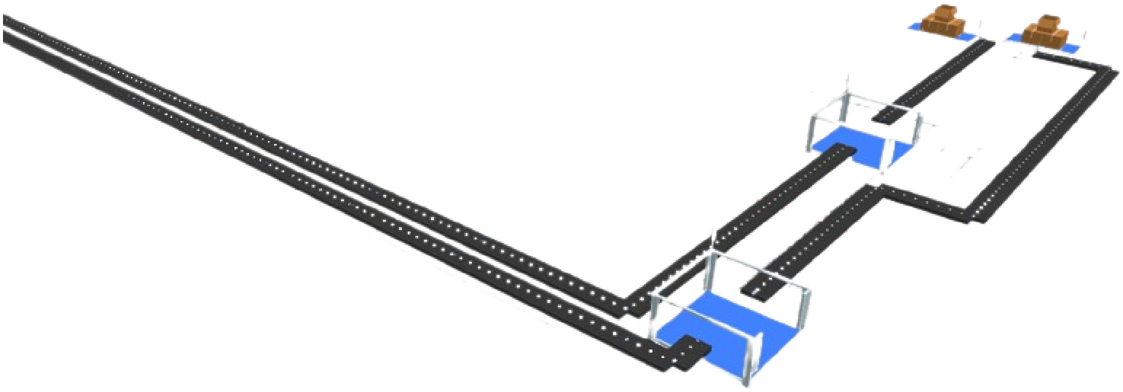


Figure 4.3: Plant Simulation model of the material supply.

The flow of material is modelled as a push flow, where the highest priority of distribution is downstream. Each MU is evaluating statements at each packing line to determine if an order is currently active on that packing line, if products are needed on that line and if there is a packing line downstream with higher prioritization that requires products. The MU then either moves into the packing line or continues to move down along the production line to the next packing line, where the same control is performed again. This decision-making process allows for the dynamic allocation of MUs based on the current operational demands and available resources.

The packing lines, described in Chapter 4.2.3, can be supported by both production lines. The model strives in keeping as equal distribution as possible between the production lines. Controls are performed every 5 minutes to calculate the demanded number of loads on the packing lines and compare it to the status in the buffers. If the demand on one of the production lines is higher and the buffer level is higher in either of the buffers, one packing machine will change the preferred supplied production line. The model changes the packing machine which results in the least possible difference in the demanded amount of products.

4.2.2 Box supply

Boxes are produced by 6 machines modelled as sources, as seen in Figure 4.4. The type of order currently processed on each packaging line determines what box type that needs to be produced. The boxes circulate on two conveyor systems until there is space available for the boxes on the packing lines. The model strives to keep a set number of boxes at the packing lines at all times to support continuous production. Since there are more packing lines than box-producing machines the machines have the capability to switch the type of box that is produced to match the requirements. These changes come with a corresponding set-up time.

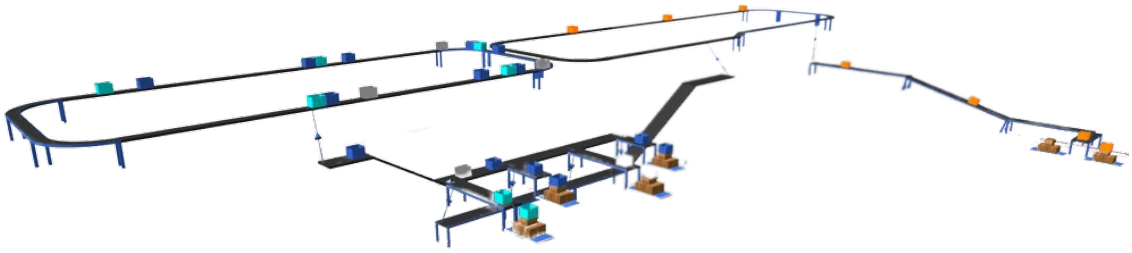


Figure 4.4: Plant Simulation model of the box supply.

The model also accounts for a maximum limit of allowed boxes on the conveyor system to prevent it from becoming full. The machines are also modelled with individual failure rates and processing time. Once a box type is not required by any of the packing lines, it is deleted to leave space for new box types. A complete box of bags is the final shipped product.

4.2.3 Packaging

The product (MUs) from the material supply, described in Chapter 4.2.1, has a weight of 500g and travels on the packing lines conveyors to a scale. The scale measures the weight needed to produce one bag of this order type and drops it into the next machine. This machine closes provided bags with the product inside. Since the product from the material supply has a higher weight than for all types of different bags, multiple bags can be created from one product. The bag weight determines the processing time for the machine.

The weight of the bags does not always work out correctly with the weight of the loads for all order types. Thus, the model also keeps track of how much weight is left over from the supplied product till the next product enters the scale. A 2 % scrap rate has also been modelled in the Packing lines. This means that each packing line needs to produce more bags for every order than what was originally planned, in order to complete an order.

The process for the whole Packing line has been visualised in Figure 4.5. Once the bag is sealed, it is transported on conveyors and packed into a box from the box supply (as described in Chapter 4.2.2). When the box has been filled with the correct amount of bags it passes through the remaining machines, enters an elevator and is transported to the palletizing where the model ends. Each machine has its individual failure rate and process time.

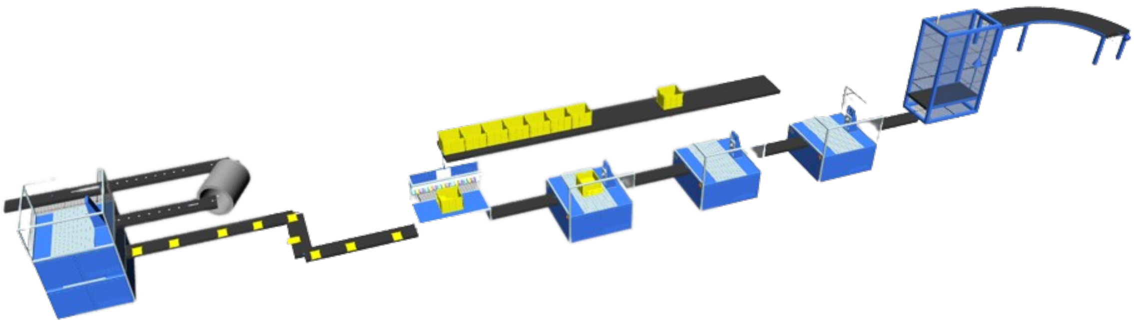


Figure 4.5: Plant Simulation model of the packaging line.

Once the correct number of boxes has been sent to the palletizing the model makes a control. It checks the production plan to see if any orders of the same type are currently active on any other packing line. If this is the case, and that order can be supported, this packing line supports the other packing line with its order. If not, cleaning and maintenance take place, according to the specified time in the production plan. The model then continues to produce the next order on the packing line until all planned orders are complete.

4.3 Quantitative study

As stated in Chapter 3.3 an identical weekly production plan with 27 different orders on 9 active packing lines was simulated in both AutoMod and Plant Simulation. The randomness of stochastic modelling makes comparison on a day-to-day basis for a single run (one random number variant) a bit misleading since the result is not constant between different runs, but rather there exists a range of values that is possible.

Therefore, the models were run with 5 different random number variants, also known as 5 observations, for the full week's production. In the first experiment, both models were run without any failure rates.

The results showed that the Plant Simulation model could reproduce the results of the AutoMod model with high precision. The difference between the average accumulated throughput of boxes varies between 0.46% and 0.2% on each day during the production week between the two models. The difference between the two models' average accumulated produced boxes each day is shown in Figure 4.6.

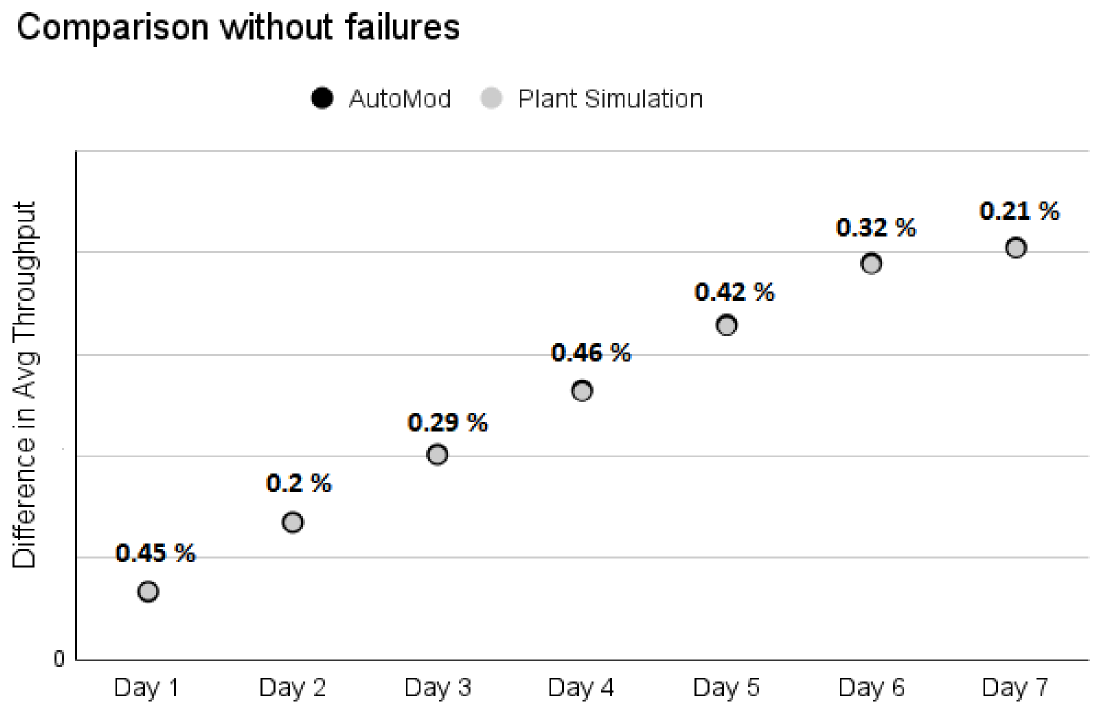


Figure 4.6: Comparison of the two models' accumulated average difference in output for each day, without failure profiles active (5 observations).

The reason for the difference could be related to the way the two models handle supporting other lines. If packing lines produced the same order type, and one of the lines was done with its order, it was for some of the orders to assist the other line. In Plant Simulation, a threshold of at least 100 boxes left to produce on order was set as a requirement to be able to be assisted with an active order. This was done to prevent the order from being split indefinitely between two production lines. A different solution could have been implemented by the AutoMod modeller.

Further, the modelling of the distribution of material flow and events is coded according to the logic of the provided model specification was interpreted. The AutoMod modeller may have used a different approach, other rules, or thresholds to distribute the material to the packing lines, which could have led to differences in the simulation results.

The second experiment included running the model with the failure rates active which increased the difference between the two models. The difference between the average accumulated throughput of boxes is now between 3.24% and 5.61% each day. The difference between the two model's average accumulated produced boxes, with failure rates active, are shown in Figure 4.7

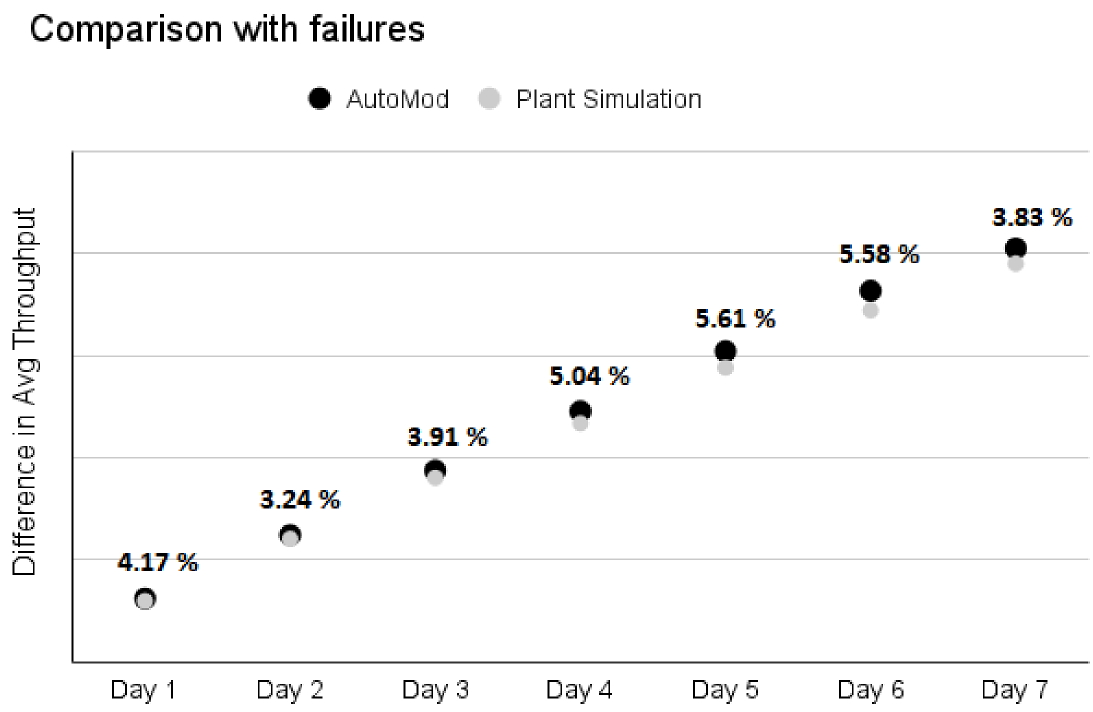


Figure 4.7: Comparison of the two models' accumulated average difference in output for each day, with failure profiles active (5 observations).

The main reason for this difference is how the two models model machine failures. Both modelled machine breakdowns were based on simulation time and the same failure data. The failure data for all machines were exponential distributions of Mean Time Between Failures (MTBF) and Mean Time to Repair (MTTR) metrics, as this was the data provided in the model specification. However, in the AutoMod model, if the machine was idle (empty), the breakdown did not occur. This can be compared to the Plant Simulation model which assumed that the breakdown could occur at any time, regardless of whether the machine was processing products or not.

While this difference leads to a higher failure rate in the Plant Simulation model, as breakdowns can occur during cleaning time and when all orders have already been processed, this approach also captures breakdowns that potentially could occur in the small gaps between MUs entering machines. For one random number variant, this seemed especially unfavourable which led to lower average throughput and increased the possible range of values.

In Table 4.1 the difference in the machine breakdown portion is illustrated, where the difference in downtime of the AutoMod model compared to the Plant Simulation can be noticed during a simulation period of four days for one machine type.

As can be seen, Plant Simulation has consistently higher failure rates. Worth noting, however, is that it is "artificially" high for Machine 1 since the packing line already finished all of its orders on day 4 (for all observations), but the machine can still continue to break down (in the Plant Simulation model).

Table 4.1: Comparison of average model failure rates of one machine type after a simulation of 4 days (5 observations).

Failure rates	AutoMod	STD	Plant Simulation	STD
Machine 1	4,25%	0,77%	8,27%	1,25%
Machine 2	5,45%	0,70%	7,08%	0,99%
Machine 3	7,68%	1,12%	8,24%	1,37%
Machine 4	-	-		
Machine 5	6,51%	0,85%	7,62%	0,41%
Machine 6	7,62%	0,88%	8,14%	0,98%
Machine 7	7,48%	1,71%	9,38%	1,16%
Machine 10	4,69%	1,43%	7,33%	1,01%
Machine 11	5,59%	0,53%	6,89%	1,28%
Machine 12	-	-	-	-

Both failure modelling approaches have their drawbacks and benefits. Basing the machine breakdown on production cycles instead of simulation time would have been a closer representation of reality for both models. A further discussion on the topic can be found in Chapter 5.2.

An additional study aimed to compare the timing of events between the two models. To achieve this, the time at which each order was finished on each packing line were saved and plotted against each other. It should be stressed that these results are only valid for one run (random number variant). The plot, as can be seen in Figure 4.8 shows that the models display an alternating but somewhat following pattern in the completion times. As time progresses, the failure rates have a larger impact, whereas the Plant Simulation model completes orders slower than in AutoMod.

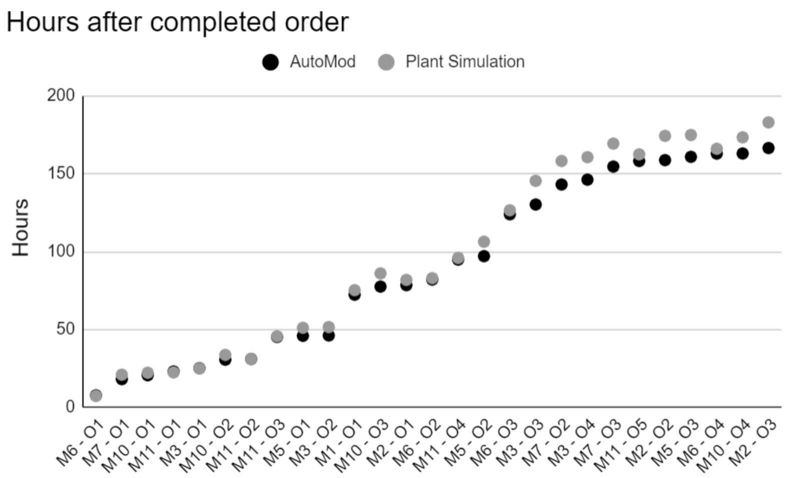


Figure 4.8: Comparison of when orders are completed, where M stands for machine and O for order.

To compare the execution time of the two simulation models, each software runs a simulation of seven days at maximum speed, while keeping the animations off. The experiment is performed under the same circumstances. Thus the same hardware and settings are used. The results have been summarised in Table 4.2.

Table 4.2: The execution times for both software during a simulation duration of seven days for 3 runs.

	Run1	Run2	Run3	Average
Plant Simulation	**8m, 38s**	**8m, 12s**	**8m, 6s**	**8m, 19s**
AutoMod	**5m, 8s**	**5m, 4s**	**4 m, 58s**	**5m, 3s**

The factors that affect the execution time of a model are for example the amount of code, objects and events. Naturally, a larger model with more code to run takes a long time to execute. Further, code structure has an effect on execution time. There is a risk that code was written in a less efficient way in the Plant Simulation model. A further discussion on the results can be found in Chapter 5.2.

4.4 Comparative study

Based on the knowledge gained from building the use case, in combination with information extracted from the interviews, a comparison between Plant Simulation and AutoMod is carried out for different simulation features.

4.4.1 Model building

During the interviews, two different model-building approaches were identified for the software types. In Plant Simulation, a top-down approach was commonly used, where users first designed the system layout, including resources, components, and material flow, and then added control logic to replicate real-world behaviour. In contrast, AutoMod tended to use a bottom-up approach, where users modelled a single resource or sub-system, then gradually built up the model to the final layout.

This corresponds with our observations from the use case. We believe this difference is due to the simplicity of using pre-built objects which can be easily dragged and dropped from the extensive library of objects in Plant Simulation. Additionally, in Plant Simulation, material flow can be modelled by connecting the objects with connectors, and basic process logic/control is already built into standard resources, as shown in Figure 4.9.

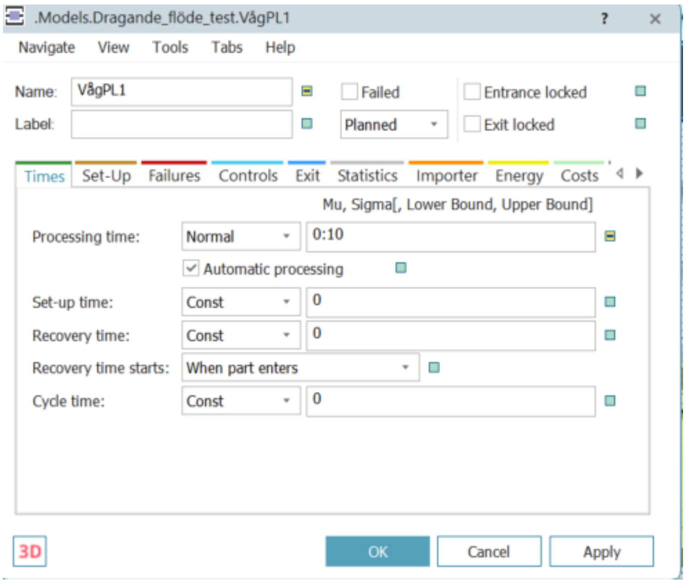


Figure 4.9: Categories of settings for a station in Plant Simulation.

This can be compared to AutoMod where for example creating a conveyor involves three components: sections, transfer and stations. This process includes multiple subfunctions to specify the conveyor's details.

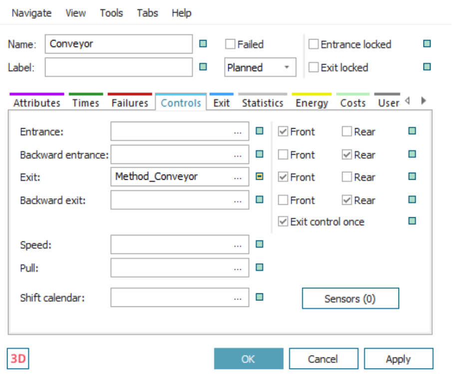
However, we argue that even though users may tend to favour one approach over the other, the opposite approach is still possible in both software. For example, if the simulation model is not that complex, a top-down approach would be suitable in AutoMod as well, where resources can be placed and simple coding using move statements can be used to create material flow. Similarly, in Plant Simulation, a bottom-up approach, such as combining multiple frames for different subsystems, does make sense for complex models.

Another finding during the interviews was that Plant Simulation users tend to be more resource-oriented and AutoMod users tend to be more load-oriented. This is likely because, in Plant Simulation, load-trigger events can be placed as a method of an exit or entry strategy in the resource settings. This means that a method is called every time a MU exits or enters that specific object. Further, as shown in Figure 4.9 object settings can be manually set directly in the resource by the users. This can be compared to AutoMod, where users tend to be more load-oriented, a load typically contains all the information, and discrete events are based on how the load triggers the events.

But, we want to stress that orientation towards load or resource may depend on various factors. For example, the background of the user, the type of simulation being performed, and the specific requirements and objectives of the simulation can influence the orientation toward load or resource.

4.4.2 Coding

When using Plant Simulation, the user writes code in files called "Methods". These methods can be triggered by an object, sensor or another method, and the code is executed when triggered by an event such as a load entering or exiting a conveyor, as shown in Figure 4.10a. The methods also contain pre-built templates of codes for different categories. These templates can be used to insert general code snippets, reducing the time and effort to create logic.



(a) Exit strategy for conveyor, using a method.

```
-- Method for controlling a load on a conveyor
if @.location= Conveyor
    Counter_Loads+=1
    @.move
end
```

(b) The method for controlling the action when a load exit the conveyor.

Figure 4.10: Exit control using a method in Plant Simulation.

The most common control structures are provided as templates. Although Plant Simulation offers the possibility to avoid coding when creating simple models it is likely that coding will be necessary when the model complexity increases.

On the other hand, in AutoMod, the simulation model is executed by running different source files in a defined order. This allows the model to specify what events should be simulated and when loads and resources are generated. Loads are then sent to processes that trigger the code containing commands for the load, making it easy to control the flow of commands in the code by sending loads to different processes. Processes are triggered by loads and perform actions stated within the process. These processes control the behaviour of a model. The user, therefore, has to code no matter the complexity of the model. See Figure 4.11 for an example of controlling loads on the conveyor.

(a) Conveyor system in AutoMod.

```
begin P_start arriving procedure
    move into conv.sta_in
    travel to nextof(conv.sta_out1, conv.sta_out2)
    send to P_next
end
```

(b) Coding for controlling flow of load on conveyor.

Figure 4.11: Conveyor system in AutoMod.

Another identified difference is how the software continuously checks for statements. In AutoMod, it is common to have while loops running to observe statements triggered by discrete events, A dummy load can be used for this purpose, allowing the loop to keep checking for the condition during the simulation. Instead in Plant Simulation observers can be used and trigger methods when the observed variable or cell changes or reaches a certain value (although while-loops are possible to implement through dummy MUs triggering methods).

The Data Table in Plant Simulation is a valuable tool that allows users to store large amounts of data generated during a simulation run. Further, the user is able to format and index columns and rows. This can be used to reference cells with strings instead of numbers. This makes the code easily understandable for others. An example of this can be seen in Figure 4.12.

	string 1	weight[kg] 2
string	Part Type	Part Weight
1	Part A	5
2	Part B	3
3		

(a) DataTable in Plant Simulation

```
if DataTable["Part Type",1] = "Part A"
    // Perform action A
else
    // Perform action B
end
```

(b) Method for referencing cell 1,1

Figure 4.12: Data Table and reference in Plant Simulation.

Regarding coding, the two software are rather similar but there are differences in syntax and code structure. If you are used to one software, it's important to be aware of features that may not exist in the other program. This can help make your coding more efficient and you can avoid brute forcing your way to a solution.

4.4.3 Collaboration

In AutoMod, code files are stored locally in a folder that can be executed by the software, and these files can be shared easily with other users via a sharing platform. Additionally, the layout, resources, and objects in AutoMod can be translated into coordinates and shared, facilitating collaboration among users. Load-oriented programming improves collaboration possibilities, as long as all participants follow the same coding principles.

Similarly, in Plant Simulation the methods used to build a model can be exported and shared as text files with other users. However, working on the same model simultaneously is challenging, as it requires a significant amount of manual work and coordination among participants to ensure that changes are properly integrated. This approach carries the risk of conflicting ways of structuring the model building and naming of variables. Since methods in Plant Simulation are often triggered by entry and exit methods, it is important for participants to understand when and where methods should be executed.

There are a few ways around this, with different drawbacks and limitations. One way around this problem is to keep a master model where any changes or bug fixes are implemented directly when done. This worked relatively well in this project. However, if the number of participants in the model increases this solution quite fast becomes problematic. Another approach is version control, however, the drawbacks are if changes are carried out simultaneously. Both of these solutions place demands on communication among the collaborators. Another possible solution is that Plant Simulation provides the option to work in different frames or subsystems that can be connected to a main frame, allowing for different coding and model-building principles to be applied by users. Thus, it should be possible to work with different processes and then later merge these. However, communication among the participants is still needed to avoid conflicts.

Another approach is to use Teamcenter, which is a product life-cycle management platform that can be used to facilitate collaboration and file sharing in the context of product or production data. Teamcenter can lock a file when a user is working on it, and therefore the need for communication or risk of conflicting changes to the same file is reduced. Other users can still access the file, but they cannot make any changes to it until the original users allow it.

4.4.4 Execution time & visualisation

Despite Plant Simulation's superior visual aesthetics, the appearance of the simulation is often of secondary importance. During the simulation execution, the user's attention is focused on monitoring the simulation output, rather than the visual representation of the model. Both software provides the option to run simulations without graphical objects, which can be particularly useful for simulations with many graphics and movable objects to save computational resources and time. Additionally, both software provides the ability to execute simulations in either two or three dimensions, with the latter requiring more rendering time.

However, having a visually pleasing simulation can have its benefits in terms of sales. A simulation that is aesthetically appealing might make the model more easily understood and accepted by those who do not have that much experience in simulation. Nevertheless, the primary focus of a simulation is to generate accurate results and output.

With a potential increased usage of the DES model as a simulator in a Digital Twin, we expect that execution time will increase in importance moving forward. For the software to accurately represent the system or process being modelled, the software must be able to execute the model quickly enough to keep up with the rate of change in the real system, as presented in Chapter 2.4. Implementing supervised and unsupervised machine learning algorithms, as shown in Figure 2.7, will require substantial data preparation, feature selection and model training. This will lead to an increased number of lines of code that must be executed.

The study, as shown in Chapter 4.3, indicated that Plant Simulation had a higher execution time when compared with AutoMod for the two models.

4.4.5 Integration with other software

Plant Simulation provides several inter-process communication interfaces to access and exchange data with other applications and servers. The wide range of built-in interfaces can be embedded with the software and allows for flexible integration with other software, which is useful for specific purposes. Further, if your manufacturing execution system (MES) can support Excel, it is possible to import them as Data Tables into Plant Simulation. This approach would instead be carried out through reading and importing data as text files in AutoMod based on our findings.

This feature lays the groundwork for a Digital Shadow, where data can be imported to the model whenever a critical event occurs in the factory and needs to be simulated. Potential scenarios and how they could be implemented have been explored in Chapter 4.6. The use of simulation runs with an updated model can give additional decision support to flexible production planning.

Plant Simulation has the ability to import STEP files. This means that users are not limited to specific CAD software when they want to modify the appearance of loads or resources. By supporting STEP files, Plant Simulation offers users the freedom to experiment with different design options and explore various configurations to increase interpretability. In AutoMod the user can either use ACE, which is a basic graphic editor to create cell file graphics (.cel files) to represent entities in a model, or AutoMod Graphic Viewer, which allows the user to display, create and edit more detailed graphics for use in a simulation.

Open platform communications (OPC) is available in both Plant Simulation and AutoMod. This protocol allows the user to communicate with an OPC server during a simulation run. This functionality is often used to replace hardware with a simulation model in order to test PLC code[66]. This is an important step in emulation and virtual commissioning. To create custom interfaces, windows Activex control can be integrated with AutoMod and Plant Simulation in order to allow applications that support this technology to execute code through the external application. For example, starting or stopping the simulation, call functions and to set variables.

Both software provides adequate built-in interfaces that allow for third-party communication to facilitate data sharing and simulation control. However, Plant Simulation has a wider range of built-in interfaces that allows the user greater flexibility in designing a custom interface tailored to the specific application. This advantage is due in part to the fact that Plant Simulation is created by Siemens, which offers a broad range of digital tools. The integration of these tools enables communication between different Siemens products, potentially simplifying the work with multiple applications.

It is important to consider the possibility of becoming locked into either supplier's products. If the user chooses the software for one aspect of operations, it may limit the user's ability to integrate other tools from different vendors. In Table 4.3 Interprocess communications and interfaces in Plant Simulation and AutoMod have been summarised.

Table 4.3: Interprocess communication and interfaces in Plant Simulation and AutoMod.

Interface	Plant Simulation	AutoMod
ActiveX	X	X
OPC& OPC UA	X	X
XML	X	
SQLite	X	
ODBC	X	X
COM	X	
DDE	X	
HTML	X	X
SIMIT	X	
C	X	
PLCSIM	X	
Socket	X	X
Teamcenter	X	
CAD	X	X

4.4.6 Learning Curve

Measuring the time and ease of learning a new simulation software can be difficult, as it can vary depending on the user's prior experience. However, based on our experience, we have noticed some differences between AutoMod and Plant Simulation. Getting started with Plant Simulation and building a simple model is relatively straightforward due to the availability of pre-built templates for code and setting for objects. Additionally, the presence of helpful guides and an active support forum provides assistance when needed. However, as the model complexity and size increase, the user still has to learn the syntax and in-built functions, which can lengthen the learning curve.

In contrast, we noticed that learning to build a simple model and program basic actions in AutoMod can be initially challenging, as it requires learning the syntax and programming concepts. However, once the user becomes familiar with the methods used to control the logic, it can become easier to provide solutions to more complex problems. As a result, we believe that the learning curve for AutoMod may be steeper at the beginning, but it can become less challenging with experience. As the model complexity increases, the prior knowledge of the syntax and programming concepts learned earlier can be an advantage.

The tutorials and documentation for both AutoMod and Plant Simulation provide good aid in learning the basic knowledge of working in the software. Both software has built-in help documentation which works in similar ways. Plant Simulation has a tool called "help on word", where the user can highlight a command to get information about syntax and attributes regarding the specific command or object. In AutoMod, the user has to search the documentation for commands and information. If using the documentation does not solve the experienced problem, there is a community forum for Plant Simulation, where users can ask questions and receive support from other users.

4.5 Evaluation of DES software

A set of evaluation criteria, see Chapter 3.6, is used to compare the two software for the specific use case. The higher frequency of answers, the more important parameter. These answers act as support when weighing the parameters according to importance when selecting DES software.

The frequency of answers to the most important simulation parameters (that were selected by the seven interviewees) when selecting a new DES software has been summarized in Table 4.4.

Table 4.4: Survey ranking of what parameters are considered most important.

Parameter	Frequency
Model building	5
Visulisation	1
Accuracy	6
In-built objects & functions	2
Tutorials, Examples & Documentation	3
Syntax/Coding	4
Integration with other software	4
Time and ease to learn	3
Debugger	4

To analyse how each parameter is weighted against each other, the AHP method, described in Chapter 3.6 is used. Each parameter is pairwise compared and scored from 1-6, where the relative score is based on the frequency of answers (Table 4.4).

Each row is then summed up as a weight in percentage, as shown in Table 4.5.

Table 4.5: Analytical hierarchal process for weighting the simulation parameters.

Criteria	Model building	Visulisation	Accuracy	In-built objects &functions	Tutorials Documentation & examples	Syntax &Coding	Integration with other software	Time and ease to learn	Debugger	Weight
Model building	1,00	5,00	0,50	4,00	3,00	2,00	2,00	3,00	2,00	18,01%
Visulisation	0,20	1,00	0,17	0,50	0,33	0,25	0,25	0,33	0,25	2,63%
Accuracy	2,00	6,00	1,00	5,00	4,00	3,00	3,00	4,00	3,00	24,82%
In-built objects &functions	0,25	2,00	0,20	1,00	0,50	0,33	0,33	0,50	0,33	4,36%
Tutorials, Documentation and examples	0,33	3,00	0,25	2,00	1,00	0,50	0,50	1,00	0,50	7,27%
Syntax & Coding	0,50	4,00	0,33	3,00	2,00	1,00	1,00	2,00	1,00	11,88%
Integration with other software	0,50	4,00	0,33	3,00	2,00	1,00	1,00	2,00	1,00	11,88%
Time and ease to learn	0,33	3,00	0,25	2,00	1,00	0,50	0,50	1,00	0,50	7,27%
Debugger	0,50	4,00	0,33	3,00	2,00	1,00	1,00	2,00	1,00	11,88%

In the survey, the participants were asked to rate the simulation parameters in either AutoMod or Plant Simulation from 1-5. 5 corresponded to the best possible rating, while 1 was the worst possible rating. The average score for each parameter is presented in Table 4.6.

Table 4.6: Average rating from survey study.

Parameters	Avarage score Plant Simulation	Avarage score AutoMod
Model building	4,40	4,00
Visulisation	4,20	2,00
Accuracy	4,20	5,00
In-built objects & functions	4,40	3,33
Tutorials, Examples & Documentation	4,20	3,83
Syntax/Coding	3,00	4,00
Integration with other software	2,80	2,00
Time and ease to learn	3,80	3,67
Debugger	3,20	4,17
Total Score:	34,20	32,00

The goal function y is the total score for the software evaluation. A higher score yields a better rating, where variable W is the weight factor and variable P is the parameter score, as shown in Equation 4.1.

$$y = \sum_{i=1}^{n} (w_i) * (P_i) \tag{4.1}$$

The average score from the survey (Table 4.6) is multiplied by the weight from the AHP matrix (Table 4.5). In total, AutoMod scored 3.912 and Plant Simulation 3.788. The results have been summarised in Table 4.7.

Table 4.7: Weighted ranking of the average parameter rating score, where the total score is the goal function y.

Parameters	Weight	Weighted Ranking Plant Simulation	Weighted Ranking AutoMod
Model building	18,01%	0,793	0,721
Visulisation	2,63%	0,110	0,053
Accuracy	24,82%	1,042	1,241
In-built objects & functions	4,36%	0,192	0,145
Tutorials, examples & Documentation	7,27%	0,305	0,279
Syntax/Coding	11,88%	0,356	0,475
Integration with other software	11,88%	0,333	0,238
Time and ease to learn	7,27%	0,276	0,267
Debugger	11,88%	0,380	0,495
Total Score:		3,788	3,912

For this specific weighting of parameters, accuracy and model building had high relevance. Worth noting is that the participants from academia rated Plant Simulation and the participants from industry rated AutoMod, which could give a skewed result toward one or the other. For a different case it might be different parameters that have higher weight and the outcome of the evaluation will be different.

By weighting the parameters according to the academic opinions solely, a different outcome of the evaluation can be achieved, see Table 4.8. In this case, documentation and tutorials were most important, and Plant Simulation scored slightly higher than AutoMod. This variation resulted in a higher total score for Plant Simulation.

Table 4.8: Ranking based solely on academia weighting.

Parameters	Weight	Weighted Ranking Plant simulation	Weighted Ranking AutoMod
Model building	8,83%	0,37	0,33
Visulisation	8,83%	0,35	0,17
Accuracy	14,69%	0,62	0,73
In-built objects & functions	4,38%	0,19	0,15
Tutorials, Examples & Documentation	22,26%	0,93	0,85
Syntax/Coding	14,69%	0,44	0,59
Integration with other software	14,69%	0,41	0,29
Time and ease to learn	8,31%	0,32	0,30
Debugger	4,38%	0,14	0,18
Total:		3,77	3,60

This can be compared with Table 4.9 where parameters were ranked according to the opinions of working professionals. Model building and accuracy were weighted highest once again. This resulted in a higher total score for AutoMod.

Table 4.9: Ranking based solely on working professionals weighting.

Parameters	Weight	Weighted Ranking Plant Simulation	Weighted Ranking AutoMod
Model building	22,41%	0,99	0,90
Visulisation	4,41%	0,19	0,09
Accuracy	22,41%	0,94	1,12
In-built objects & functions	7,30%	0,32	0,24
Tutorials, examples & Documentation	7,30%	0,31	0,28
Syntax/Coding	7,30%	0,22	0,29
Integration with other software	7,30%	0,20	0,15
Time and ease to learn	7,30%	0,28	0,27
Debugger	14,26%	0,46	0,59
Total score:		3,90	3,93

4.6 Dynamic production schedule

To investigate the opportunity to implement a Digital Shadow (DS) in Plant Simulation in the use case, a proposed framework for the implementation has been visualised and can be seen in Figure 4.13.

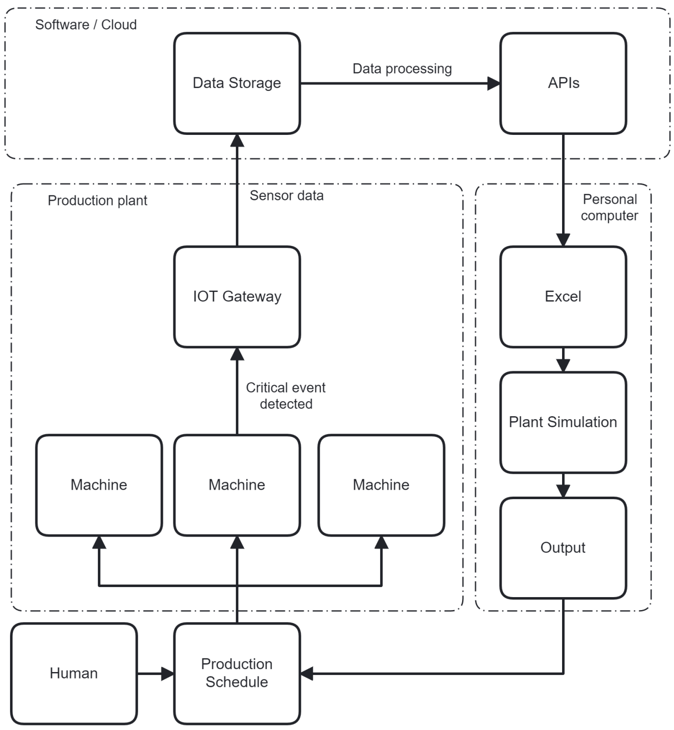


Figure 4.13: Proposed architecture for a DS in DES software.

A scenario where this could be implemented is presented. Firstly, the production schedule is manually set by the production planner. The throughput of the production schedule is then validated by running the simulation model. However, a critical event, outside of what was included in the simulation model occurs. This could for example be a major breakdown where spare parts that should have been available, in fact, were not available. Production data could then be extracted through a data storage system or the company's MES and necessary parameters would be inserted into an Excel file.

This data would is then imported into the DES model into a data table and run with the new updated specific scenario that needs to be tested. Based on the output of the new simulation run, an action can be made to reschedule orders. While the data exchange is not automated and requires human assistance to import/export into the model, the process is facilitated by the framework and can save model-building time. This allows for a quicker analysis of unexpected problems and the result can act as important decision support on what the best course of action is to take.

Using Excel and data tables is only possible if the Plant Simulation model is adapted to setting parameters by reading the data from a data table. A method to import data from Excel into a Data Table in Plant Simulation can be seen in Figure 4.14. The production facility is required to have the enabling Industry 4.0 technology to collect, transmit and process the required data to the personal computer, as described in Chapter 2.3.3.

```
PackLinjeSupplyPlanning.readExcelFile("C:\Users\    \Desktop\Production_Plan.xlsx","Plan")
Machine_parameters.readExcelFile("C:\Users\    \Desktop\Production_Plan.xlsx","Machine_Parameters")
Failure_table.readExcelFile("C:\Users\    \Desktop\Production_Plan.xlsx","Failures")
```

Figure 4.14: Code to import data from Excel into data tables in Plant Simulation.

To simulate a scenario where the conditions change with time, the simulation could be run on the default settings until the time of the critical event is detected. The settings can then be updated by altering the parameters using the Excel sheets and continuing to run the simulation. This allows us to study the effect of the event with respect to what has already been produced. This alternation can be effective addition in decision support in whether or not to change the production plan.

Two imagined scenarios were implemented in the Plant Simulation model, as described in Chapter 4.2, to illustrate the potential of the Digital Shadow.

Case 1: A machine is finished before the scheduled time
If a machine due to dynamic events is in a state of "Unplanned" (thus done with all its orders) and other machines are behind schedule, a signal on the shop floor can send the data to the simulation model to run scenarios to optimize the production based on the new situation.

For the originally set production schedule, the last order would be finished after 7 days, 9 hours, 59 minutes and 12 seconds in machine two. After extracting the data from the production facility, a simulation run can be performed to analyse the effect of moving the last order on machine two to another machine that is currently in a state of Unplanned. The result would be that the last order is finished after 7 days, 1 hour, 17 minutes and 27 seconds, in machine two. This is a reduction in the production time of more than 8 hours.

Case 2: A machine breakdown is taking longer than expected
If a machine is not able to be fixed due to unforeseen reasons, all the scheduled orders will be delayed on that machine. To test how this affects the production system, the data can once again be exported into an Excel document and imported into the simulation model to be analyzed.

On day three of production, machine 5 is down and there are no spare parts or maintenance workers available. The impact is that it takes a whole day to repair the machine. The simulation is then run to the point of failure and then the parameters are updated so that the failure rate is correlating to the time in the real machine.

The final order finishing time is then shifted to 7 days 8 hours, 46 minutes and 17 seconds, with a total downtime of 27,67 hours in machine 5. Since machine seven is done with all its orders first, we investigate the opportunity to assist with the final order on machine 5. The same scenario is run with the new production plan to analyze the difference. With the changed production planning, machine five is finished after 7 days, 22 minutes and 12 seconds, while machine seven (and thus all orders) is finished after 7 days, 49 minutes and 31 seconds.

5
Discussion

The chapter discusses the project content and the results from the different studies. The discussion also addresses any doubts about the methodology used to come up with the results.

5.1 Qualitative study

The qualitative study conducted in this thesis included interviews with both academic and industry experts to gather insight on topics related to DES, data quality, and Digital Twins. A noticeable difference in the model construction was identified between the users of general and specialised DES software users, indicating that there are differences in model construction and simulation processes between general and specialised DES software (RQ1). The interviewees using specialised DES software described a resource-oriented top-down approach while the general-purpose DES software users described a load-oriented bottom-up approach. The reasons for this difference could be related to how the software types are structured with user interface, objects and tools. Another explanation for this difference could be related to model complexity, where a bottom-up approach could be more common in more complex models.

The interviews also revealed professionals hesitations about using DES in Digital Twin, with concerns focusing on the cost- benefits of the implementation and the current state of data collection in production systems. While the concerns related to data collection align with previous research made by Uhlemann et al. [54], the doubts raised about the cost-effectiveness of real-time data integration with simulation models was somewhat unexpected. The progress of IOT devices, which has made it more cost-effective to implement and support Digital Twins as described by Madni et al. [45] and that Digital Twins are viewed as an avoidable trend as described by Tao and Zhang [40], may indicate a potential gap between academia and industry. These are challenges that need to be addressed before fully implementing a Digital Twin (SQ 2.2). However, potentially cost-effective solutions were identified during the interviews as dynamic production planning (SQ 2.1) through implementing a Digital Shadow.

The study's sample size was small, which has its advantages and disadvantages. On one hand, a small sample size allows for greater depth and detail in the analysis, as each response can be thoroughly examined. On the other hand, a larger sample size would have provided a more diverse perspective from multiple industries and companies. Further, it is worth noting that the majority of the interviewees from the industry had similar experiences. This could have influenced the outcome and introduced bias to the study. To mitigate this risk, a larger or more diverse sample size would have been ideal.

Another limitation of the qualitative study is the risk of bias in data collection. The outcome of the study will depend on the actors performing the study, and this can introduce a subjective interpretation of the data. Additionally, the quality of questions and how they were formulated can significantly impact the quality of answers. The study used open-ended questions, which allowed interviewees to interpret the questions according to their own experiences. However, this approach may also lead to different interpretations of the same questions, making it more challenging to compare responses across different interviewees. There is also a risk of leading the interviewee to a certain answer when clarifying any question.

5.2 Quantitative study

Multiple simulation runs of the full production schedule were carried out under equal conditions to gain statistical confidence in the cumulative throughput of boxes on each day for both models. The difference between the two models, without the failure rate included, varied between 0.46% and 0.2%, indicating that the level of detail in output generated by the specialised DES software compared to the generalised DES software for the industrial use case is at a sufficient level (SQ 1.3). Further, the execution time for both models was measured, where the specialised DES software had a 64,8% increase in execution time for the use case (SQ 1.2).

The different methods of modelling machine breakdown were noticed to have an impact on the results. The model in the general DES software made use of an approach where breakdowns only could occur while the machine was active, i.e. processing products. If the machine did not register as processing at the time of breakdown, the machine would not go down, and a new MTBF counter would begin, disregarding the previous demand. However, it was still based on the simulation time. While in the specialised software, breakdowns were similarly modelled to occur based on simulation time, but it was regardless of whether the machine was active or idle. As a result, a machine could go down even if it was not processing any loads.

This difference in the modelling approach resulted in a higher failure rate for the model in the specialised DES software since breakdowns could occur during cleaning and order changes. But it would also catch breakdowns that would occur in the small gaps between MU:s entering resources. Nonetheless, it is worth noting that both approaches have their drawbacks, and a more accurate representation of the real factory setting could have been achieved with breakdown data based on the number of cycles, i.e. data regarding how many products can be processed before a breakdown occurs.

It is also important to consider what type of breakdown when deciding on the appropriate data to use. For example, breakdowns of tools do not happen when the machine is not processing any products. Therefore, for these types of breakdowns, basing the breakdown on simulation time does not make sense. Instead, basing the breakdown on processing time would have been more appropriate. Worth noting, however, is that the machine still corrodes, even if it is not working, and this type of wear and tear should also be considered in the modelling approach. Therefore a deeper breakdown of data would have been beneficial in the use case to be able to model reality better.

A fairer comparison could have been made for the execution time where the two software executes the exact same code. The two simulation models were built by different people with different levels of expertise and experience. This naturally leads to differences in the way the models are optimized and implemented. However, a 64.8% higher average execution time for the specialised software could indicate that the generalised software has a faster execution time.

5.3 Comparative Study

The comparative study showed that there are both similarities and differences between the two software types in model building, coding, collaboration, execution time & visualisation, integration with other software and learning curve (SQ 1.1). If you are used to one of the software, it is important to be aware of features that may not exist in the other type of software.

The choice of using AutoMod to represent the generalised DES software and Plant Simulation as the specialised DES software could have potentially influenced the outcomes of the study. For example, if a different software program had been used to represent the generalised software, it might have affected the ease of representing complex systems or integrating various industry-specific processes into the simulation. Similarly, if another DES software had been selected to represent the specialised software, the findings, such as the top-down approach, might have been different. Naturally, including more DES programs would have been beneficial to strengthen the findings regarding different software types. However, due to the limited time frame of a master's thesis, it is challenging to fully explore and learn all the details of multiple DES software options, let alone include more than two DES software in the analysis.

Furthermore, since only one use case was investigated, a limited number of processes were modelled and examined. Modelling different systems and processes for a different use case could have potentially resulted in slight variations in the findings. For instance, if use cases within the processing industry were modelled, different findings might have emerged, even when using the same software.

Therefore, one might argue that an extended time frame would have been needed to master both software with their unique features and capabilities to be able to compare both software fairly in depth. Even though you might have worked with a particular software for many years, there may still be details or features that you are not aware of. This research might therefore be viewed as only an introduction to differences that exist between the software types, and what implications they might have when choosing software.

5.4 Evaluation of DES software

The interviewees from both academia and the industry answered an anonymous survey, where they answered what simulation parameters they considered the most important (multi-choice) when choosing new software. They also rated the software they had the most experience with on a set of simulation features (SQ 1.2) from 1 to 5. AutoMod had a weighted total score of 3.912, which can be compared to Plant Simulation which had a weighted total score of 3.789.

If the weighting were solely based on the weighting of participants from academia Plant Simulation had a total score of 3.77 and AutoMod had a weighted total score of 3.6. If the weighting were solely based on working professionals AutoMod had a weighted total score of 3.93 and Plant Simulation of 3.90. These results verify earlier research by Guimares et al. [8] by indicating that different users have different preferences when it comes to DES software, and that the detailing and weighting components should be made according to the user or specific companies preferences. The result of the evaluation can give an indication of what software is suitable for the user and act as an additional decision support when selecting DES software. The choice of evaluation parameters and the number of criteria used to compare the two software, as well as the number of participants in the survey and interview, impacts the study's outcome. With other participants, the results could have a wide range of outcomes.

However, this study is not intended to serve as a guide for selecting a particular software or to suggest that one is superior to the other. Instead, the goal is to evaluate simulation parameters and apply a methodology to determine which simulation capabilities are important for the organization and the specific use case. From the results, it is clear that both general and specialised DES software are viable alternatives when building simulation models, and the choice may ultimately come down to individual and organizational factors which were not included in the evaluation.

5.5 RQ1

Research Question 1 was mainly answered by experience gained from building the use case in the specialised DES software while leveraging our previous experience with generalised DES software. The findings were supported by the available documentation in the two software. The quantitative and qualitative studies, as well as the survey, acted as support and provided external insights from working professionals in the industry and academia.

The interfaces and tools that exist to support the user, differ from each software. The specialised DES software is more up to date in terms of a user-friendly interface and design, while the generalised DES software has an older user interface and is mostly text based. Both software types have their strengths and weaknesses when it comes to the certain simulation features investigated in the thesis. However, the level of detail in terms of output, and thus, accuracy, for the specialised Discrete-event simulation software is comparable to the generalised software for the specific use case.

Additionally, the effect of model complexity has been touched upon, where the specialised software might be the preferred option if the user does not have a lot of experience in simulation or coding and would want to build a simple model. However, as the model complexity increases there are benefits and drawbacks of both software, making the choice less distinct and more related to personal preference. One might favor the generalised DES software due to the increased control of flow and another might favor the specialised DES software due to the ease of modelling.

5.6 RQ2

Research Question 2 was mainly answered through the literature study. This literature review aimed to identify key competencies and enabling technologies for DES in DT applications and challenges that exists in moving towards a fully functioning DT. Within the described key technologies, the need for competent people to develop and maintain is needed to support the DT.

The findings from the literature study, inputs from the interview and the experience from our model building made us identify an interesting area of implementation for the use case. This would be to move from a Digital Model to a more accurate and dynamic Digital Shadow. This could be implemented in the use case by utilizing dynamic production planning. The idea behind this approach is that before the start of the week, the orders the company is to produce the following week are simulated to obtain the most optimal production schedule.

In reality, the production process is prone to uncertainties such as prolonged machine breakdowns due to missing spare parts or sudden changes in demand, which can affect the production schedule. Meaning that what was optimal at the beginning of the week, might no longer be the most optimal going forward. Therefore, by setting a predefined rule, such as after a major breakdown, the production data is imported into the simulation, and the simulation is run once again to optimize the production schedule based on the latest information.

This approach helps to ensure that the simulation remains accurate and up-to-date, allowing for more cost-effective decision-making and increasing the value proposition of a DES. Two situations where this approach could be useful were when a major breakdown occurs in the factory and when a machine is not being utilised.

5.7 General discussion

Understanding the differences between generalised and specialised simulation software can increase the understanding of specific requirements and functionalities necessary for using Discrete-event simulation within the Digital Twin framework. By looking at the model construction and simulation process in both types of software (RQ 1), insights can be gained regarding flexibility and industry-specific features. This knowledge is then linked to RQ 2, which gave insights into skills and knowledge areas that are needed to effectively utilise the simulation capabilities. Integrating these research questions allows for an exploration of the interaction between simulation software types and the competencies required for successful application within the Digital Twin, contributing to a deeper understanding of simulation technologies and their potential in real-world applications.

Sakr et al. [60], Morabito et al. [16] and Karakra et al. [61] show use cases where there have been successful implementations of DES in DT environments using real-time data. However, real-time synchronization and having a reliable IT infrastructure remain a challenge, and therefore, it is essential to identify an application where the benefits outweigh the extra investment and maintenance cost of a Digital Twin. Transitioning to a Digital Shadow may be a possible alternative that does not rely on the real-time bi-directional continuous flow of data. One could argue that the specialised DES software might be more future-proof for this shift since the software has more capabilities regarding the connection of the model with external applications. On the other hand, the faster execution time could be an argument in favor of the generalised DES software, making the choice not clear-cut.

6
Conclusion & Future work

By building the model in the specialised software, performing experiments and comparing the results to an existing model in the generalised software, we could conclude that DES software is at its core the same, no matter if it's specialised or general. The differences in results can be explained by the underlying methods and assumptions made. Systems and processes that are possible to model in a general Discrete-event simulation software are also possible to model in a specialised software, and vice versa. The difference lies in the methodology to reach the goal. Both types of software have their advantages and disadvantages and are both viable options to use in a professional setting.

While our study revealed differences between the types of software in model building, it is important to note that the choice of software ultimately depends on the user's specific requirements and objectives. Factors such as the user's experience, preferences, and interpretations can significantly affect the outcome of the model-building process and results. To strengthen these findings, more software for generalised and specialised could be looked into.

Selecting the most effective simulation tool for a production process depends on the specific requirements and constraints of the project. Understanding the differences between the two types of software can help researchers and users select the appropriate software and develop effective and efficient models that achieve the desired results.

Throughout this study, an interesting gap between academia and industry was identified. Although a connected Digital Twin can enhance decision-making capabilities, a relevant question is whether the additional investment and maintenance costs justify its implementation for companies. In some instances, using DES might be sufficient to attain the desired outcome. Before companies decide to invest in a DT, numerous factors and challenges must be evaluated, and the technology may not be appropriate for all scenarios. It is essential to analyze the capabilities and expected outcome of the DT before committing to the investment, much like selecting the right software. Therefore, exploring cost-effective steps to implement DT should be a priority for future research.

There exist different definitions of a DT, both in terms of to what extent the data flow, but also in terms of abilities and depths, complicating the matter. Reaching a common ground in both academia as well as industry regarding what a Digital Twin is, and what it should be able to do, would be beneficial.

While a fully implemented DT seems to be facing some resistance from the industry at the moment due to described challenges and cost versus benefits doubts, our results show indications that moving from a DM to a DS could be an attractive alternative for increasing the value proposition of DES. Future work within the subject could investigate the possibility to implement the proposed framework and apply it to a use case. Another possibility for future work is to find other applications than production planning where a DS can increase the value proposition of DES.

Another recommendation for future work is to utilize DES as a platform for training various machine learning algorithms. Reinforcement learning (RL) algorithms use agents to explore the environment to find optimal paths and configurations. The use of DES as an environment in RL could be a powerful tool for the agent to optimize the simulation model towards a goal function. An example is to optimize AGV systems where many carriers are in motion simultaneously and each has a target destination with multiple paths. The goal function could then be to reach the destination in the shortest time or path.

Bibliography

[1] Julian Reitman. “Simulation of a manufacturing system”. en. In: *Simulation* 8.6 (1967), pp. 311–317.

[2] Anders Skoogh, Björn Johansson, and Johan Stahre. “Automated input data management: evaluation of a concept for reduced time consumption in discrete event simulation”. en. In: *Simulation* 88.11 (2012), pp. 1279–1293.

[3] Faezeh Hosseinpour and H. Hajihosseini. “Importance of Simulation in Manufacturing”. In: 51 (Jan. 2009).

[4] P Faget, U Eriksson, and F Herrmann. “Applying discrete event simulation and an automated bottleneck analysis as an aid to detect running production constraints”. In: *Proceedings of the Winter Simulation Conference, 2005.* IEEE, 2006, 7 pp.–.

[5] S Robinson. “Discrete-event simulation: from the pioneers to the present, what next?” In: *J. Oper. Res. Soc.* 56.6 (2005), pp. 619–629.

[6] *Customer satisfaction: retention rates by industry worldwide 2018*. en. `https://www.statista.com/statistics/1041645/customer-retention-rates-by-industry-worldwide/`. Accessed: 2023-2-8.

[7] Abraham Harold Maslow. *Psychology of Science: A Reconnaissance.* Regnery Publishing, 1966.

[8] Alexandre Magno Castañon Guimarães, José Eugenio Leal, and Paulo Mendes. “Discrete-event simulation software selection for manufacturing based on the maturity model”. en. In: *Comput. Ind.* 103 (2018), pp. 14–27.

[9] Jerry Banks et al. *Discrete-Event System Simulation: International Edition.* en. 5th ed. Upper Saddle River, NJ: Pearson, 2010.

[10] Jalal Nikoukaran, Vlatka Hlupic, and Ray J Paul. “A hierarchical framework for evaluating simulation software”. en. In: *Simul. Pr. Theory* 7.3 (1999), pp. 219–231.

[11] J Banks. “Selecting simulation software”. en. In: *1991 Winter Simulation Conference Proceedings.* IEEE, 2002.

[12] V Hlupic, Z Irani, and R J Paul. “Evaluation Framework for Simulation Software”. In: *Int. J. Adv. Manuf. Technol.* 15.5 (1999), pp. 366–382.

[13] Giuditta Pezzotta et al. “Evaluation and assessment of two simulation software for service engineering”. en. In: *Serviceology for Services.* Tokyo: Springer Japan, 2014, pp. 263–272.

[14] Shishengxiong Zhong, Jinzhe Zhao, and Sabri Pllana. "A comparison of simulation tools for supply chain management". MA thesis. Linnéuniversitetet, July 2020.

[15] Erik Flores-García et al. "Analyzing the Characteristics of Digital Twin and Discrete Event Simulation in Cyber Physical Systems". In: Aug. 2020, pp. 238–244. ISBN: 978-3-030-57996-8. DOI: 10.1007/978-3-030-57997-5_28.

[16] Lucrezia Morabito et al. "A Discrete Event Simulation Based Approach for Digital Twin Implementation". In: *IFAC-PapersOnLine* 54.1 (2021). 17th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2021, pp. 414–419. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2021.08.164. URL: https://www.sciencedirect.com/science/article/pii/S2405896321009332.

[17] *About us*. en. https://afry.com/en/about-us. Accessed: 2023-1-30.

[18] AFRY. *AFRY AB (PUBL) YEAR-END REPORT JANUARY–DECEMBER 2021*. https://afry.com/sites/default/files/wkr0006-4175031_6.pdf. Accessed: 2023-1-30.

[19] Matthew W Rohrer. "AutoMod tutorial". In: *Proceedings of the 29th conference on Winter simulation - WSC '97*. New York, New York, USA: ACM Press, 1997.

[20] Kajsa Björnham and Camilla Marklund. "Simulering av produktionskapacitet ledtider och PIA vid kiselkretstillverkningen på Ericsson Components AB". en. MA thesis. Luleå Tekniska Universitet, 1997.

[21] http://www.automod.se/documents/automod_key_features.pdf. Accessed: 2023-1-30.

[22] Gabriel Fedorko et al. "Research on Using the Tecnomatix Plant Simulation for Simulation and Visualization of Traffic Processes at the Traffic Node". In: *Applied Sciences* 12.23 (2022). ISSN: 2076-3417. DOI: 10.3390/app122312131. URL: https://www.mdpi.com/2076-3417/12/23/12131.

[23] Mike Rouman. *Tecnomatix Plant Simulation 15 - what's new?* en. https://blogs.sw.siemens.com/tecnomatix/tecnomatix-plant-simulation-15-whats-new/. Accessed: 2023-2-8. May 2019.

[24] Thomas J Schriber and Daniel T Brunner. "Inside discrete-event simulation software: How it works and why it matters". In: *Proceedings of the 29th conference on Winter simulation - WSC '97*. New York, New York, USA: ACM Press, 1997.

[25] Norman Matloff. "Introduction to discrete-event simulation and the simpy language". In: *Davis, CA. Dept of Computer Science* 2 (Jan. 2008).

[26] Satyajith Amaran et al. "Simulation optimization: a review of algorithms and applications". en. In: *4OR* 12.4 (2014), pp. 301–333.

[27] Dan Muller. "Automod™ - providing simulation solutions for over 25 years". In: *Proceedings of the 2011 Winter Simulation Conference (WSC)*. IEEE, 2011, pp. 39–51.

[28] Matthew W Rohrer. "AutoMod tutorial". In: *Proceedings of the 29th conference on Winter simulation - WSC '97*. New York, New York, USA: ACM Press, 1997.

[29] *Tutorial - chapter 1: Introduction to simulation and*. de. `https://plant-simulation.de/schulungen/tutorial/tutorial-chapter1/`. Accessed: 2023-1-30. Dec. 2018.

[30] Flavio Hourneaux Jr, Marcelo Luiz da Silva Gabriel, and Dolores Amalia Gallardo-Vázquez. "Triple bottom line and sustainable performance measurement in industrial companies". en. In: *Rev. Gest.* 25.4 (2018), pp. 413–429.

[31] Britta Ottesjö and Sophia Wetterberg. "Enabling Digitalization with Emulation Prepared Simulation". en. MA thesis. Göteborg: Chalmers University of Technology, 2021.

[32] Patrycja Hoffa-Dabrowska and Katarzyna Grzybowska. "Simulation Modeling of the Sustainable Supply Chain". In: *Sustainability* 12.15 (2020).

[33] Jack G.A.J. van der Vorst, Seth-Oscar Tromp, and Durk-Jouke van der Zee. "Simulation modelling for food supply chain redesign; integrated decision making on product quality, sustainability and logistics". In: *International Journal of Production Research* 47.23 (2009), pp. 6611–6631. DOI: `10.1080/00207540802356747`. URL: `https://doi.org/10.1080/00207540802356747`.

[34] Marcus Frantzén. *MPR 271 - Simulation of Production Systems: Theoretical basics of DES [PowerPoint Slides]*. en. Nov. 2021.

[35] Maulshree Singh et al. "Digital Twin: Origin to Future". In: *Applied System Innovation* 4.2 (2021). ISSN: 2571-5577. DOI: `10.3390/asi4020036`. URL: `https://www.mdpi.com/2571-5577/4/2/36`.

[36] *Top 10 strategic technology trends for 2019*. en. `https://www.gartner.com/en/newsroom/press-releases/2018-10-15-gartner-identifies-the-top-10-strategic-technology-trends-for-2019`. Accessed: 2023-2-6.

[37] Bernard Marr. *These 25 technology trends will define the next decade*. en. `https://www.forbes.com/sites/bernardmarr/2020/04/20/these-25-technology-trends-will-define-the-next-decade/?sh=50f0051929e3`. Accessed: 2023-2-6. Apr. 2020.

[38] Haifan Jiang et al. "How to model and implement connections between physical and virtual models for digital twin application". In: *Journal of Manufacturing Systems* 58 (2021). Digital Twin towards Smart Manufacturing and Industry 4.0, pp. 36–51. ISSN: 0278-6125. DOI: `https://doi.org/10.1016/j.jmsy.2020.05.012`. URL: `https://www.sciencedirect.com/science/article/pii/S0278612520300777`.

[39] Roland Rosen et al. "About The Importance of Autonomy and Digital Twins for the Future of Manufacturing". In: *IFAC-PapersOnLine* 48.3 (2015). 15th IFAC Symposium onInformation Control Problems inManufacturing, pp. 567–572. ISSN: 2405-8963. DOI: `https://doi.org/10.1016/j.ifacol.2015.06.141`. URL: `https://www.sciencedirect.com/science/article/pii/S2405896315003808`.

[40] Fei Tao and Meng Zhang. "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing". In: *IEEE Access* 5 (2017), pp. 20418–20427. DOI: 10.1109/ACCESS.2017.2756069.

[41] Michael Grieves. "Digital Twin: Manufacturing Excellence through Virtual Factory Replication". In: (Mar. 2015).

[42] Fei Tao et al. "Digital Twin in Industry: State-of-the-Art". In: *IEEE Transactions on Industrial Informatics* 15.4 (2019), pp. 2405–2415. DOI: 10.1109/TII.2018.2873186.

[43] Maulshree Singh et al. "Digital Twin: Origin to future". en. In: *Appl. Syst. Innov.* 4.2 (2021), p. 36.

[44] Werner Kritzinger et al. "Digital Twin in manufacturing: A categorical literature review and classification". In: *IFAC-PapersOnLine* 51.11 (2018). 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018, pp. 1016–1022. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2018.08.474. URL: https://www.sciencedirect.com/science/article/pii/S2405896318316021.

[45] Azad M. Madni, Carla C. Madni, and Scott D. Lucero. "Leveraging Digital Twin Technology in Model-Based Systems Engineering". In: *Systems* 7.1 (2019). ISSN: 2079-8954. DOI: 10.3390/systems7010007. URL: https://www.mdpi.com/2079-8954/7/1/7.

[46] K Agalianos et al. "Discrete event simulation and digital twins: Review and challenges for logistics". en. In: *Procedia Manuf.* 51 (2020), pp. 1636–1641.

[47] Itziar Ricondo, Alain Porto, and Miriam Ugarte. "A digital twin framework for the simulation and optimization of production systems". en. In: *Procedia CIRP* 104 (2021), pp. 762–767.

[48] Aidan Fuller et al. "Digital Twin: Enabling technologies, challenges and open research". In: (2019). eprint: 1911.01276.

[49] Amal Tawakuli, Daniel Kaiser, and Thomas Engel. "Synchronized Preprocessing of Sensor Data". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020.

[50] Yuqian Lu et al. "Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues". en. In: *Robot. Comput. Integr. Manuf.* 61.101837 (2020), p. 101837.

[51] Eoin O'Connell, Denis Moore, and Thomas Newe. "Challenges Associated with Implementing 5G in Manufacturing". In: *Telecom* 1.1 (2020), pp. 48–67. ISSN: 2673-4001. DOI: 10.3390/telecom1010005. URL: https://www.mdpi.com/2673-4001/1/1/5.

[52] Marlon Dumas et al. *Fundamentals of business process management*. en. Berlin, Germany: Springer, 2019.

[53] Henric Widén. *PPU231 - Production and product service system. Smart Maintenance for master students [PowerPoint Slides]*. en. Jan. 2022.

[54] Thomas H-J Uhlemann et al. "The digital twin: Demonstrating the potential of real time data acquisition in production systems". en. In: *Procedia Manuf.* 9 (2017), pp. 113–120.

[55] Dazhong Wu et al. "Cybersecurity for Digital Manufacturing". In: *Journal of Manufacturing Systems* (Apr. 2018). DOI: 10.1016/j.jmsy.2018.03.006.

[56] Alalvanan Lingam Ashok Kumar and Rahul Rao Jagadeeswar. "Life-long learning materials enabling the growth of Digital Twin competences". en. MA thesis. Göteborg: Chalmers University of Technology, 2023.

[57] Paola Fantini, Marta Pinzone, and Marco Taisch. "Placing the operator at the centre of Industry 4.0 design: Modelling and assessing human activities within cyber-physical systems". en. In: *Comput. Ind. Eng.* 139.105058 (2020), p. 105058.

[58] Yassine Qamsane et al. "A unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems". In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE).* IEEE, 2019.

[59] Christina Obermaier et al. "Measuring the realtime capability of parallel-Discrete-Event-Simulations". en. In: *Electronics (Basel)* 10.6 (2021), p. 636.

[60] Ahmed H Sakr et al. "Building discrete-event simulation for digital twin applications in production systems". In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).* IEEE, 2021, pp. 01–08.

[61] Abdallah Karakra et al. "A discrete event simulation-based methodology for building a digital twin of patient pathways in the hospital for near real-time monitoring and predictive simulation". In: *Digital Twin* 2 (Jan. 2022), p. 1. DOI: 10.12688/digitaltwin.17454.1.

[62] Martyn Denscombe. *The good research guide: For small-scale social research projects.* 4th ed. Buckingham, England: Open University Press, 2010.

[63] Ralph L Keeney. "On the foundations of prescriptive decision analysis". In: *Utility Theories: Measurements and Applications.* Dordrecht: Springer Netherlands, 1992, pp. 57–72.

[64] Alyona Medelyan. *Coding qualitative data: How to code qualitative research (2023).* https://getthematic.com/insights/coding-qualitative-data/. Accessed: 2023-2-15. Oct. 2019.

[65] Rosaria de F.S.M. Russo and Roberto Camanho. "Criteria in AHP: A Systematic Review of Literature". In: *Procedia Computer Science* 55 (2015). 3rd International Conference on Information Technology and Quantitative Management, ITQM 2015, pp. 1123–1132. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.07.081. URL: https://www.sciencedirect.com/science/article/pii/S1877050915015562.

[66] Jerry Banks. *Getting Started with AutoMod, second edition.* en. Brooks Automation, Inc., May 2004.

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

CHALMERS
UNIVERSITY OF TECHNOLOGY