Sascha Gotthardt, BSc

Simulation of an RFID-aided and Digitized Milk-run Logistics System within the Context of a Learning Factory

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Software Engineering & Management

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr.techn. Roman Kern

Institute for Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, December 2018

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date | Signature

Abstract

In order to meet the current trends and challenges in the industrial sector, production logistics is one of the focal points in the optimization of assembly systems. In order to increase the efficiency of the internal material supply, milk-run systems were introduced. The milk-run is responsible for the replenishment and transport of parts from the warehouse to the workplaces within a company and is part of the intralogistics system. The aim of this thesis was to digitize such a milk-run system with the help of an RFID system and to test it afterwards. In the course of this digitization a software was developed, which simulates the complete production and logistics process of an assembly line. In order to be able to test this simulation, a suitable institution had to be found where the digitized milk-run system could be implemented and tested in order to generate a meaningful comparative value for the simulator. With the IIM LEAD Factory a suitable learning factory was found in which it was possible to implement the digitized milk-run system. The digitized milk-run system consists of an order management sub-system, which gives the logistics employee an overview of open orders and suggests to him or her where the parts to be picked are located on the shelf. The picking process is completed in connection with a pick-to-light system, that visually shows the employee exactly the compartment in the warehouse that is needed for the active order. In addition, the digitized milk-run system was enhanced by a route calculation, which allows to find the most suitable path from the warehouse to the workplace. One of the tasks of the already mentioned simulator is to simulate real production in such a way that it is possible to make suggestions to the employee for orders that would ideally have been placed in the near future. In order to evaluate that these simulated orders are correct, it was important to compare them with real orders from the learning factory. The result was not only a fully functional digitized milk-run system, but also an evaluation of how well the digitized system works in comparison to the old system and how precise the results of the simulator are. With the completion of this project it is possible to have a digitized milk-run system available, which has been tested and evaluated in a university institution.

Contents

List of Figures

List of Tables

1 Introduction

This chapter gives the reader an overview of this thesis. We start with the motivation, meaning why this topic is interesting at all and what we can expect from it. Followed by the Research Questions which will be answered in the course of this thesis. Thereafter, challenges of this work are briefly addressed in order to understand what problems there are. We then present the LEAD Factory, where the results are tested. Finally, in the last subsection of this chapter the structure of the thesis is explained.

1.1 Motivation

Those who have opened the newspaper or switched on TV in recent years have certainly noticed the topic of digitization. This topic is on everyone's lips all over the world. Some people want to create jobs with it, others simply want to earn money from it.
Whatever the motive behind it, most people agree that it will come. The focus in numerous companies is exactly on that. Producing products as cheaply and effectively as possible with the use of digitization.
But what is actually behind it? What does an application for digitization look like?

In this thesis, the possibilities of digitization are examined more closely, especially in the context of production and industry. Since logistics systems make up a large part of modern production and because there is currently a lack of knowledge regarding digitized milk-run systems, this work concentrates precisely on this topic, the milk-run system, which is responsible for the supply run between workplaces and warehouse. The reader will find in Chapter 2.1 a more detailed explanation on what a milk-run system is.

In order to have a multitude of data available for the purpose of consumption prediction starting right from the beginning, a central aspect of this work will be a simulator, which makes it possible to simulate consumption characteristics, number of workplaces, length of supply routes, etc. of a variety of plants. For validation of the simulator the learning factory of the Institute for Innovation and Industrial Management (IIM) is used.

In a nutshell, this thesis consists of two parts. On the one hand it consists of a simulator which shall generate the required data and on the other hand it consists of a possibility to validate the data from the simulator.

1.2 Research questions

This thesis offers a variety of possible research questions. However, in order not to go beyond the scope of a typical time frame of a thesis, only the following research questions are considered below.

- Can we use the simulator to find improvements in the milk-run system compared to a real-life scenario?

If the answer of the first question is yes:

- Can a digitized milk run system be built in the context of the IIM LEAD Factory?

And if the answer of the second question is yes:

- How precise is the prediction of the simulator compared to a real-life scenario. The time difference is to be recorded, for example in the context of a validation in the learning factory of the IIM Institute.

These questions are the most interesting ones for the author and shall be investigated further.

1.3 Main challenges

To validate the simulator part of the thesis it is necessary to use a real-life scenario. It is planned to use the learning factory of the IIM Institute for this purpose. However, since there is still no functioning digital milk-run system there, it must first be developed and implemented. For this purpose, an actual state analysis of the analog initial state of the learning factory is created in order to then be able to create a concept for the digitization. Furthermore, the necessary material, such as hardware for the visualization and management of supply runs, is acquired. Finally, the software for the learning factory is implemented, which can be used to validate the simulator. The approach for the implementation of all systems is using what is available in the LEAD Factory plus buying additional hardware which is necessary to fulfill the requirements. All hardware engineering is based on a do-it-yourself approach.

1.4 LEAD Factory

Current trends in the industrial sector under the guise of Industry 4.0 mean that more than ever companies are having to think about possible increases in efficiency and productivity in their own operations. Combined with the trend to bring production back to high-wage countries, this results in an increasing demand for distinctive methodological knowledge in the field of process planning and design. With the LEAD Factory [1], as can be seen in Figure 1.1, the IIM Institute [2] pursues two strategies: Targeted training of students, but above all the tailor-made further training of people already working in industry.

The concept of the LEAD Factory is modular, so that different training contents can be combined with each other as desired. The contents of the training courses are based on the competences and many years of experience of the institute in the areas of industrial engineering (workplace design and ergonomics, working time calculation, lean production), logistics management ("just in time" material provision, layout planning) and operational energy efficiency (industrial energy management). In the small-scale learning factory, practical experience can be gained

[1] http://industrie40.tugraz.at (Accessed on: 2018-10-29)
[2] https://www.tugraz.at/institute/iim/home/ (Accessed on: 2018-10-29)

Figure 1.1: The digital state of the IIM LEAD Factory where scooters are assembled. In this state of the learning factory, many implementations of digitization are helping the training participants to optimize the throughput on the production line. These include camera systems for process recognition, cameras to capture facial expressions, gesture control of workplaces and many more.

realistically by assembling the TU Graz scooter, a pedal scooter consisting of approximately 60 individual parts, which can be seen in Figure 1.2. The initial situation is at first a sub-optimal assembly process, which is optimized step by step. Through the continuous measurement of specific indicators such as throughput time or productivity, the improvements achieved are made visible. In the subsequent discussion with the training participants, an attempt will be made to transfer the improvements implemented in the LEAD Factory to the respective company.

1.5 Outline

In the next chapter (Chapter 2), the literature review, the most important topics, which contribute to the understanding of this thesis and shall underline that there is a research gap, are examined and explained in more detail.

Chapter 3 then works towards answering the research questions or filling the gaps in research. For this purpose, the concepts with which the author intends to do

Figure 1.2: The IIM scooter which is assembled in the LEAD Factory exploded in its 60 parts. All these parts need to be assembled by the training participants of the LEAD Factory.

this are presented. Specifically, the RFID system will be presented, which makes this whole undertaking possible, then the already existing logistics systems will be presented and finally it will be shown which steps have been taken for the implementation of the digitization.

Chapter 4 presents both the results for the digitization process and the results of the current analysis before implementation. This provides a good basis for drawing conclusions from the results in the next chapter.

As already mentioned, the results are discussed in chapter 5 and the research questions are answered accordingly. In addition, lessons learned are addressed and the limitations of the developed system are explained.

Finally in chapter 6 the whole thesis is summarized in the conclusion and there will be a outlook on topics that could build on this thesis and also what might be possible if somebody would invest more time into it to implement it.

2 Literature review

In this chapter the reader will find all the relevant background information that is needed in order to understand the content of this thesis.
First, milk-run systems, which make up a crucial part of this thesis, are explained in more detail. Since the implementation takes place in the LEAD Factory, learning factories will also be examined in more detail. Afterwards the principles of RFID will be addressed. Due to the simulation part of the thesis, discrete event and continuous simulations as well as the concrete application of agent-based models will be introduced. In the context of the random number generation the Monte Carlo methods will be presented. Finally, flexible manufacturing and material flow systems are mentioned.

2.1 Milk-run systems

> 'In lean manufacturing, milk run (MR) systems represent route-based, cyclic material-handling systems that are used widely to enable frequent and consistent deliveries of containerised parts on an as-needed basis from a central storage area (the 'supermarket') to multiple line-side deposit points on the factory floor. MR systems generally result in short lead times, low variability, and low line-side inventory levels.' [10]

This definition of a milk-run system by Bozer and Ciemnoczolowski [10] means in simpler words that there is a system which is responsible for the supply of parts between the warehouse (supermarket) and the workplace. The system is operated by a logistics employee who is responsible for picking and transporting (if not fully automated) the parts. The logistics employee drives periodically through the plant and checks the workplaces for empty boxes. The logistics employee takes the empty boxes with him to the supermarket and refills them there. After filling,

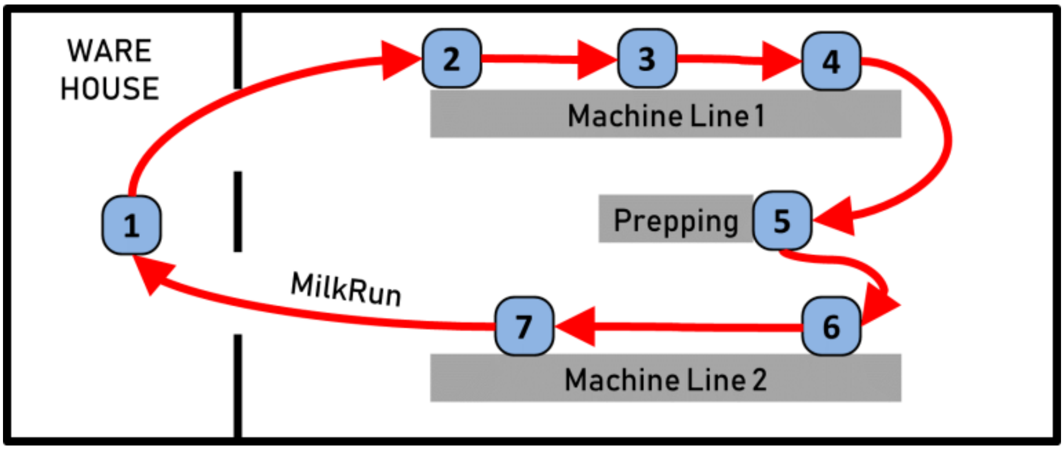

Figure 2.1: The figure shows a simplified milk-run system. The numbering implies the sequence of the process. The logistics employee starts the milk-run at the starting position (1), the warehouse or supermarket. The logistics employee then travels through the workplaces in sequence and checks whether there is material, parts or empty boxes to replace (2, 3, 4). The fifth stopping point (5) implies that the milk-run not only passes the same workplaces, but can also be used flexibly for any type of workplace. After the last two stopping points (6, 7), the logistics employee returns to the starting point (1) and refills the boxes taken along. This process is repeated periodically. This means that on the next trip, the refilled boxes are delivered back to the corresponding workplace and empty boxes are taken back with them. The figure is taken from Christoph Roser at AllAboutLean.com under the free CC-BY-SA 4.0 license.

the logistics employee makes his or her way back through the system and delivers the previously filled boxes again. This ensures constant availability of the parts directly at the workplace with the advantage that no large inventory is required at the workplace itself and everything is stored centrally in the supermarket.

Figure 2.1, which is taken from Christoph Roser [1], shows a typical (simplified) milk-run system which is used in all kind of factories all over the world. The numbering and the red arrows indicate the direction of the route through the shop floor.

Historically, the term milk-run actually comes from delivering milk. In the past, the milk man had to come up with a way to deliver fresh milk to customers as quickly as possible. At that time, a system was developed that runs every morning and sometimes even in the evening after milking the cows, on a fixed route with many

[1] https://www.allaboutlean.com/milk-run-intro (Accessed on: 2018-11-05)

stops to deliver the fresh milk. It became established that the customers put their empty milk bottles in front of the front door if they wanted a full milk bottle. This had the advantage that only empty milk bottles were replaced [59]. This created the requirements for today's milk-run systems which were defined above by Bozer and Ciemnoczolowski [10]. Namely:

- To run a fixed route with several stops.
- To run at fixed times.
- And to replace only the material that was consumed.

In literature there are all kind of synonyms for the milk-run term. In order to keep this thesis readable only the term 'milk-run' will be used throughout this work. Nevertheless the synonyms are listed below for a complete approach:

- Small train [28]
- In-plant milk-run [16]
- In-plant supply [56]
- Tugger train [38]
- Mizusumashi [2]

2.1.1 On-demand milk-run systems

On-demand means that the system eliminates the need to periodically run the route, because a mechanism at the workplace allows the employee to notify when material or parts are consumed and therefore new material or parts are needed to continue the work. Of course, this signal should not come when the employee has used up all parts, but parts should still be available to ensure an incident free assembly. It is therefore necessary to be able to record the parts at the workplace in such a way that at least an approximate estimate of the remaining quantity can be made. A good solution for this is to use several boxes with a certain number of parts per box. For example, if you use four boxes with five parts each, you will know after each box how many parts can still be used. Regardless of the mechanism by which you signal that parts have been consumed, boxes make it possible to reliably emit this signal as soon as a box is empty. This allows you to tell the responsible

[2] http://taegliche-verbesserung.de/die-implementierung-der-lean-tools/ (Accessed on: 2018-11-05)

logistics employee what is needed, or at least that something is needed, without the logistics employee having to visit the workplace beforehand.

Bae et al. [6] experiment in their work with on-demand systems in the milk-run context. They try to optimize the replenishment configurations in factories by simulating different scenarios. They then try to 'make recommendations based on supply train utilization and workplace starvation.' [6]
Another work examines the effects of disturbances on the logistics employees responsible for the milk-run. Such disturbances can also include changes in the system from a periodical to an on-demand system [42].

2.2 Learning factories

> 'Learning factories have been developed to impart substantial knowledge about improvement process concepts and methods to seminar participants within a real-world manufacturing environment. Thus, it is possible to teach the curriculum's contents in a very practice-oriented manner. The advantages of a real-world manufacturing environment can be used for the academic education of university students as well as for the training of industry participants. More and more companies are convinced of this concept, which is why they are implementing learning factories to qualify their specialists from the shop floor to the management level. The principal aim of learning factories is always to convey their complex view of business processes and impart methods, concepts and which provide for the detection of improvement potentials and the implementation of more efficient processes.'[44]

In other words, the biggest advantage of a learning factory is that the participants are actually doing the real thing. In Figure 2.2 the learning pyramid of the NTL Institute for Applied Behavioral Science [3] can be seen. Bih-Show Lou [50] has described in her work that students who are teaching other students have the best retention rate of before gathered knowledge. This makes sense, since they really have to understand the topic in order to teach it to others in a clear way and to answer questions regarding the taught. Even though the work of Lou is concerning teaching chemistry to medical students in China, the principle of learning can also

[3] https://www.ntl.org (Accessed on: 2018-11-06)

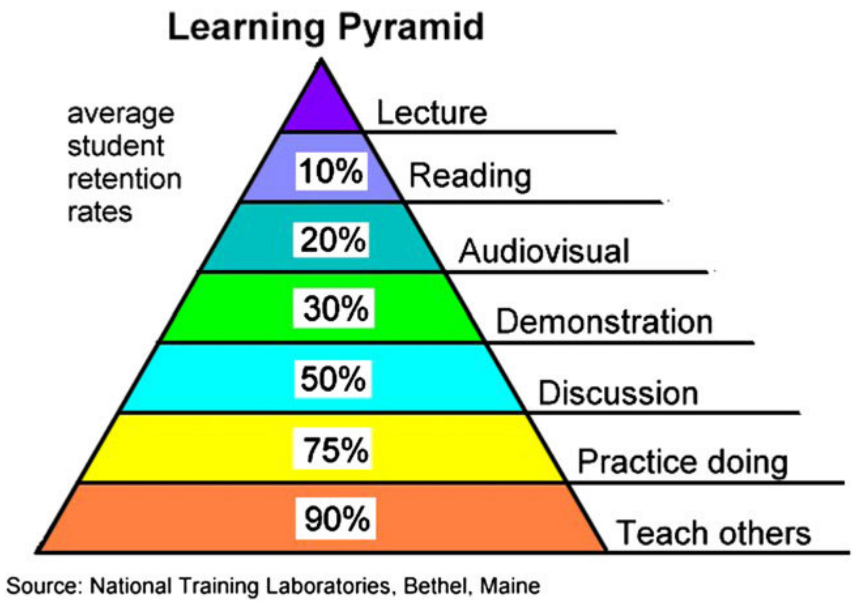

Figure 2.2: The learning pyramid is visually showing the retention rate of an average student. At the top of the pyramid is the worst form for keeping knowledge in the mind, the lecture. And at the bottom of the pyramid is the best form for remembering knowledge, teaching others what was learned before. The principle of a learning factory comes in at the second step of the pyramid, where people are actually doing the practical based on a real-life scenario.

be applied to industry and production trainings.
The next best thing to teaching others is doing a practical work based on the topic. That's where a learning factory becomes involved. By experiencing the real thing in a realistic scenario, the participants of a training can immediately implement what they have learned beforehand and experience the effects of their work. This way the participants keep the gained knowledge longer as when only hearing or talking about theoretical concepts.

It should be mentioned that there is a lively discussion in the scientific community about this learning pyramid. It is pointed out by many scientists [48, 49, 62, 67] that there is no empirical study that gives exact percentage values for the individual types of learning as shown in Figure 2.2. Nevertheless for the purpose of this thesis it is enough to have an approximate classification of the learning types.

2.3 Radio frequency identification

The first Radio frequency identification (RFID) systems were developed during the second world war. They were used to distinguish between friend and foe in aircraft and tanks. Even today, advanced systems are still used by the military for precisely this purpose. The advantage is that, depending on the system, the chip can also be read from a great distance. The decisive factor here is the radio frequency used for the RFID system [64].
Another widely used application for RFID is animal identification. A chip with information stored on it is implanted under the animal's skin and can then be read with an RFID reader [20].
Today, RFID is very widespread and can be used in almost every industry. For example, it is used in retail as anti-theft protection [35]. Another example is vehicle recognition. Since the beginning of 2004, information has been exchanged with trucks on Austria's roads via so-called toll bridges for the purpose of recording costs [37].

Figure 2.3, taken from a bachelor's thesis [29], describes the functionality of RFID. If an electromagnetic field is induced into the coil of the chip by a reader, the chip is supplied with sufficient power to reproduce its stored data. The chip does not generate its own electromagnetic field, but manipulates the reader's field to communicate the information to it. The reader uses its antenna to register the changes in the field and then processes the data [29].

The use of RFID in the manufacturing industry is very well described in the following quote by Zhong et al. [69].

> 'Radio frequency identification (RFID) has been widely used in supporting the logistics management on manufacturing shopfloors where production resources attached with RFID facilities are converted into smart manufacturing objects (SMOs) which are able to sense, interact, and reason to create a ubiquitous environment. Within such environment, enormous data could be collected and used for supporting further decision-makings such as logistics planning and scheduling.' [69]

RFID is therefore not only a technology for monitoring and controlling production [70], but also offers an excellent opportunity to make future improvements

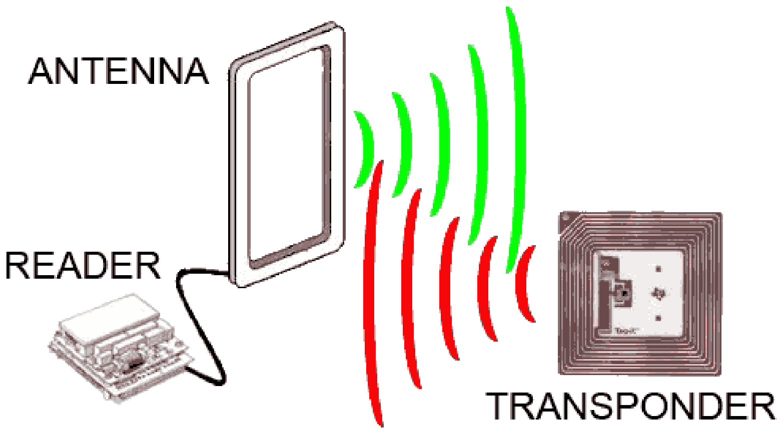

Figure 2.3: This Figure shows a simplified schematic of the functionality of an RFID system. In principle, a system consists of a reader with an antenna and a transponder or chip. The antenna induces an electromagnetic field in the coil of the chip (green). This weak field is sufficient to supply the chip with enough power so that the chip can transmit its stored information. For this purpose, the chip changes the original magnetic field accordingly (red), so that the antenna registers the differences in the field and the reader can interpret these differences as information from the chip.

to the system on the basis of this data due to the amount of information collected [22].

Further information on RFID in the industrial context and why RFID was chosen for this milk-run approach can be found in chapter 3.5.1.

2.4 Discrete event simulation and continuous simulation

> 'A Discrete Event System is a system where state changes (events) happen at discrete instances in time, and events take zero time to happen. It is assumed that nothing (i.e. nothing interesting) happens between two consecutive events, that is, no state change takes place in the system between the events (in contrast to continuous systems where state changes are continuous). Those systems that can be viewed as Discrete Event Systems can be modeled using Discrete Event Simulation. (Other systems can be modelled e.g. with continuous simulation models.)' [65]

An often used example for a discrete event system takes place in a supermarket. Consider that there is a number of customers in the queue for the register. This number can only be a whole (real) number like 5 or 8 and not 5.2 or 8.6, since a human is always counted as 1 and can not be divided. The number increases if a customer joins the queue and it will decrease if a customer is done at the register and is leaving the store. So there is a concrete time when an event occurs, namely the time when a customer steps into the queue and when a customer leaves the store. This event is used to then update the simulation [46].

In Figure 2.4, which is taken from Wikipedia [4], two charts can be seen. The upper chart shows the result of a sales time-line done with a discrete event simulation and the lower shows the same done with a continuous simulation. As can be seen in the upper chart, in this example the discrete event simulation is only changing at fixed intervals. This intervals can be any time-frame, like minutes, days, weeks, months or even years. This means there is a certain real number of sales in a certain time-frame [24] However, this is just true for this example because it shows the

[4] https://en.wikipedia.org/wiki/Continuous_simulation (Accessed on: 2018-11-08)

number of sales in a certain time-frame. An event can occur at any time, which means that there is no fixed interval or time for an event to necessarily occur. Figure 2.5, which is based on a drawing from A. I. Sivakumar [5], illustrates the meaning of a non-fixed event occurrence.

The continuous simulation on the other hand is not event driven and uses mostly differential equations to determine the exact value correlating to a certain time. This can mean that the value of sales at a particular minute is 1/4 of a product. That may not be possible, because e.g. a quarter of a mobile phone can not be sold, but it is nevertheless a accurate value for that point in time [46].

The use of simulators (most of them are discrete event simulators since processes in the industry are event driven and do seldom rely on physical equations) is also widely spread in industry for all kind of applications. It is used for performance improvements of the sales department [30], for production planning [14] and supply chain management [63] to name just a few. For this purpose there are lots of different languages to develop a simulator that fits every specific need [27, 9, 11, 15, 18, 53].

In a nutshell we can say that the simulation of discrete events is about modeling a system that evolves over time through a representation in which state variables change momentarily with the event. Whereas the continuous simulation comprises the modelling over time by a representation in which state variables change continuously depending on the time [13].

.

2.5 Agent-based models for simulation

Agent-based modelling is a unique, individual-based method of computer-aided modelling and simulation [31].
In contradistinction to different types of modelling, in agent-based modelling many small entities (agents) have possibilities for decision-making as well as for action. The system behavior derives from the behavior of the different agents and therefore is not predefined at the system level. If this leads to effects on the system level that are not directly derivable from the algorithms of the agents, one speaks of

[5] http://www.ntu.edu.sg/home/msiva (Accessed on: 2018-11-15)

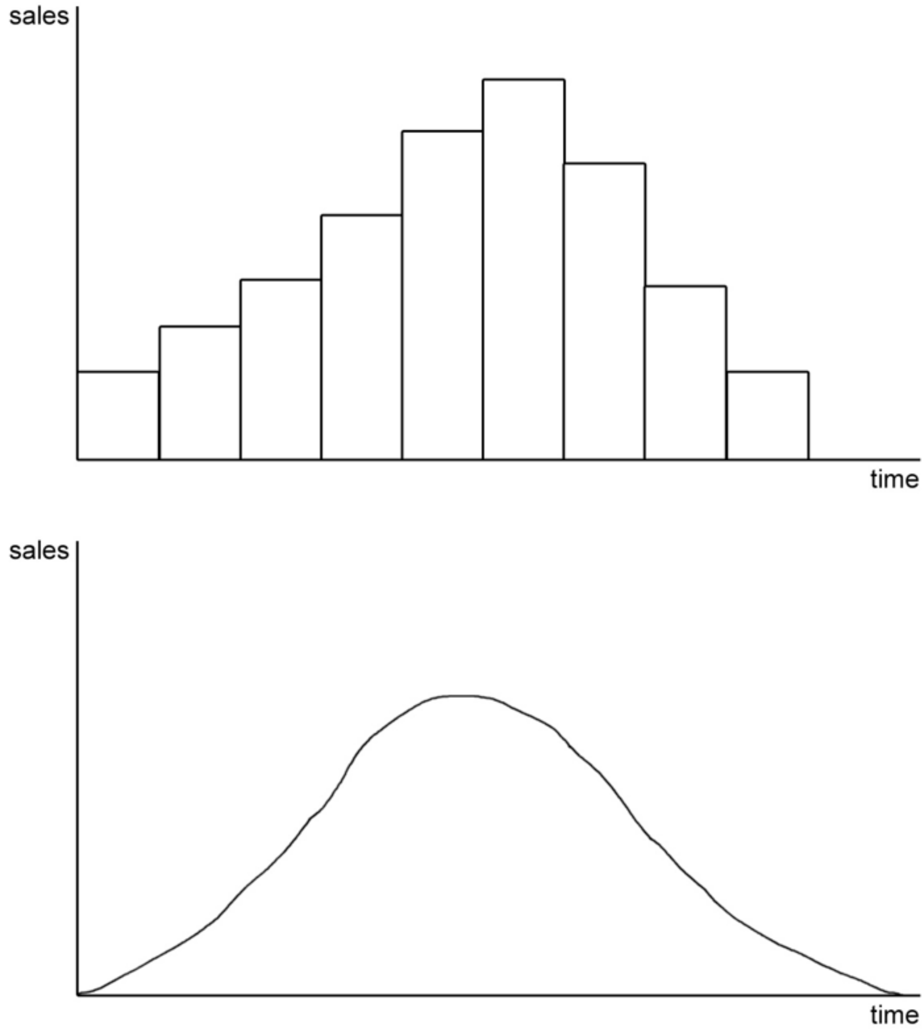

Figure 2.4: In this example of a sales distribution over time with a discrete event simulation (the upper chart) and a continuous simulation (the lower chart), the difference between these simulation methods can be seen. The discrete event simulation is using predefined events to update the simulation accordingly. As can be seen in the upper chart, the amount of sales is measured at fixes time intervals. This intervals can be anything from minutes, hours, days and months to years. Thus, the basic assertion can be e.g. 'we sold 300 products in a time-frame of 2 months'. A continuous simulation on the other hand is suitable to simulate the exact number of sales at any given moment of time, even if that means that the result is an impossible one. E.g. 'We sold 1/4 products in the last minute'. Since selling a quarter of a product is not possible, one could say that the simulation is not correct, but then again it is correct if we look at a time-frame of 4 minutes. In a nutshell, the continuous simulation can make an exact prediction at any given moment while the discrete event simulation is event driven and can provide predictions in between events.

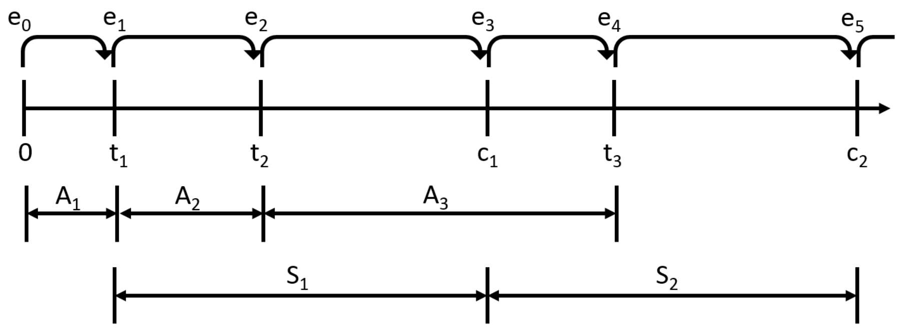

Figure 2.5: This Figure illustrates the random occurrence of events (e), the time between arrivals (A) and the time it takes the costumer to finish their task (S) on a time-line. The time-line starts at 0 or with the 0th event (e0), which indicates the start of the simulation. The next events are the arrival of the customers (e1 and e2 or t1 and t2). The third event (e3) occurs because the first customer is done (c1) and e4 indicates the arrival of customer number three at t3. A1 (0 to t1), A2 (t1 to t2) and A3 (t2 to t3) are the times that have passed between the arrival of customers. Respectively S1 (t1 to c1) and S2 (c1 to c2) are the times the customers need to finish their task. What is important is, that events can occur at any time and therefore do not depend on a fixed schedule. However, it is still possible that an event of one kind needs another event to happen before, e.g. the finishing of a task can only occur when the task was started in the first place.

Emergence [8]. Furthermore, a system behavior independent of the individual decision-making can be realized.
Two crucial characteristics of agent-based modeling are the ability to explicitly map a heterogeneous behavior and the dependencies on other agents [40].
This form of modelling is used primarily when the focus of a particular question is not the stability of an equilibrium or the assumption that a process returns to equilibrium, but rather the question of how a system can adapt to changing conditions [5]. Thereby, the insight is taken into account that more complex problems demand a thorough understanding of the micro-level, i.e. the decision-making of individuals, their diversity and their interactions.

Macal and North characterize the behaviour of Agents as 'often described by simple rules, and interactions with other agents, which in turn influence their behaviours' [52]. In Figure 2.6, also taken from Macal and North [52], a typical schematic model of an agent with its attributes, methods and interactions can be seen.

Agent-based models are used in many different areas because they can easily be

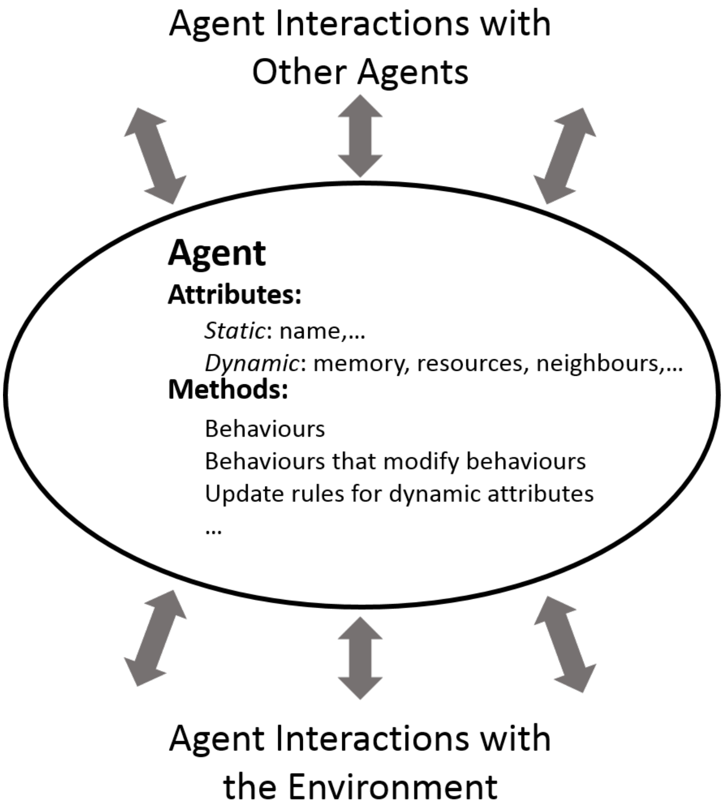

Figure 2.6: In this figure, the schematic model of an agent can be seen. An agent consists typically of a set of attributes and methods. The attributes can be anything from its own name to certain characteristics. The behavior of an agent and all sorts of instructions of how to react to a certain situation are defined in the method section of the agent. An agent always interacts with other agents and its environment. This is indicated by the sets of the bidirectional arrows.

adapted to any conceivable situation. For example, there is scientific work on the use of agent-based models on the stock exchange to find out how the stock price could be realistically predicted [4]. There are models for the simulation of supply chains in industry [51], for the prediction of epidemic spread [7] and in this context also for possible scenarios of a biological weapon attacks [12]. There are also applications that simulate the human immune system [25] and applications to better understand the purchasing behavior of customers [57]. Then there are applications to understand why past civilizations collapsed and disappeared [41], as well as simulations to predict the outcome of land [54] and sea battles [34]. Thus, the possibilities of agent-based models are almost unlimited.

2.6 Monte Carlo method

> 'The idea of using computers to carry out statistical sampling dates back to the very beginning of electronic computing. Stanislav Ulam and John Von Neumann pioneered this approach with the aim to study the behavior of neutron chain reactions. Nicholas Metropolis suggested the name Monte Carlo for this methodology, in reference to Ulam's fondness of games of chance.' [45]

To understand how this works, consider that you want to travel from point A to point B. In order to generate as realistic a time as possible for this problem, certain circumstances must be taken into account. E.g. the time of day and the time of year. Whether it rains or the sun shines, whether the fuel is empty or whether there is a traffic jam or an accident on the road. Countless different possibilities can shorten and extend the journey time. In order to achieve a reliable result all these possibilities are randomly triggered by a certain probability and calculated in a large number of calculations (simulations). If you calculate an average result after each simulation, you approach the most probable result gradually. There's also the chance to get the exact result at the first simulation, but there is no possibility to know it before a large number of calculations is averaged or the result is already known. Thus, one can be lucky and only need a hand full of calculations or one can be unlucky and needs billions of calculations [21].

The Monte Carlo method is therefore a method that can be used to generate random numbers, variables or events (sampling) in order to predict some outcome. The method is for instance used to simulate complex processes that cannot be analyzed directly, such as e.g. production processes in a manufacturing company, to uncover bottlenecks and opportunities in production [36].

2.6.1 Hit-or-Miss Monte Carlo method

With the Hit-or-Miss Monte Carlo method it is possible to approximate the solution or even solve intractable problems that would otherwise take much more effort to solve them in a traditional way. In many cases it is enough to make a close call than to calculate the exact result. Lets take for instance Figure 2.7, which is taken from ScratchaPixel.com [6]. On the left side of the figure there can be seen a graph with a rectangle and a black shape in it. Now consider that you want to know the area of this black shape. With the help of the Hit-or-Miss method, all you need is the size of the rectangle and some points that are uniformly and randomly 'thrown' into the rectangle. Now all you have to do is count the points that are inside the black shape and divide it by the number of total points. Multiplied by the area of the rectangle will leave you with the approximate area of the black shape. What this means in form of an equation is shown in Equation 2.1.

$$A_{shape} \approx \frac{N_{hits}}{N_{total}} \times A_{rectangle} \tag{2.1}$$

On the right side of Figure 2.7 you can see the reason why uniformly distributed points are essential for this method. If there is no uniform distribution but a non-uniform one, then this is called a bias. If a bias occurs, then it is no longer possible to make a reliable estimate using the Hit-or-Miss method. If, however, it is known that a bias exists, then these effects can be compensated with a number of appropriate techniques [32].

2.7 Flexible manufacturing systems

> 'A flexible manufacturing system (FMS) is an arrangement of machines … interconnected by a transport system. The transporter carries work to

[6] https://www.scratchapixel.com (Accessed on: 2018-11-08)

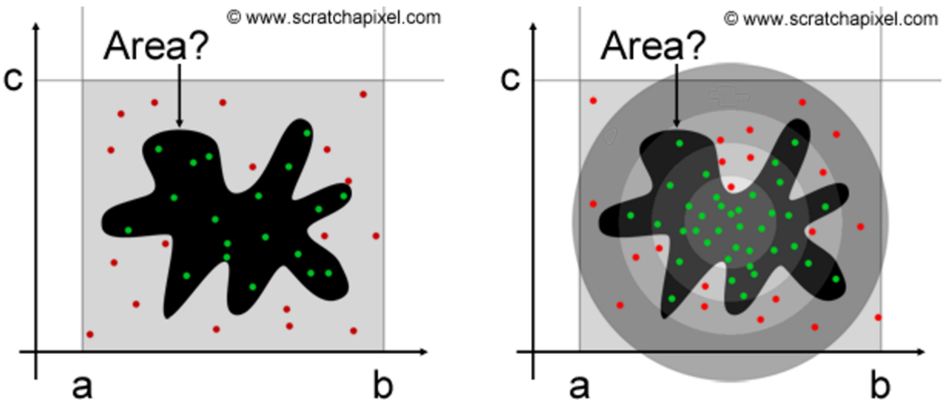

Figure 2.7: On the left we see a grey rectangular shape with the area of ab x ac and a black shape. If we want to know the area of the black shape we can approximate it with the help of the Hit-or-Miss Monte Carlo method. To do so, we need to throw uniformly, independently and randomly placed points on the rectangle and count how many of the thrown points are inside the black shape (hits). The hits divided by the total number of throws (points), multiplied by the area of the rectangle will give us a approximation of the black shaped area. The higher the number of thrown points, the more accurate the approximation becomes. It is very important that the points are uniformly distributed, otherwise result will no longer be accurate. If this happens we speak of a bias. Such a bias can be seen on the right side of the figure. In the middle of the graph we find a higher density of points than at the periphery. However, if such a bias is known, it can be countered with appropriate techniques.

> the machine on pallets or other interface units so that work-machine registration is accurate, rapid and automatic. A central computer controls both machines and transport system.' [60]

In order for a company to be able to satisfy its customers, be as cost efficient as possible and react very quickly to market situations, you need a system that can achieve all this. The FMS was developed to do all the before mentioned. The buzzword for this approach that can be heard on every corner today is 'agility'.

In Figure 2.8, which is taken from MechLecture.com [7], the typical components of a FMS are shown. The main components are

- Workplaces
- Material handling
- And computer control

Workplaces are all those locations where machines are automated and do their work without being controlled by employees. This criterion usually includes CNC milling machines, CNC drills and the like, because they can be programmed to do a specific task over and over again.
There are various sub-components for material handling. The loading or unloading of material or parts into the corresponding machine is carried out automatically by robots at loading and unloading stations provided for this purpose. In the same way, the entire transport of the material or parts is automated, e.g. with assembly lines. Finally, there is also the automated storage and retrieval system, which is important for the completion of the product flow process. Thus, the entire process from fetching the individual parts from the warehouse to the final storage of the finished part is fully automated.
The third main component is the computer control, which makes this whole FMS possible by controlling all sub-systems. The computer control is thus the central starting point for the entire process and enables a flexible and optimized processing of all tasks with the help of the real time control system [43].

[7] `www.mechlectures.com` (Accessed on: 2018-11-12)

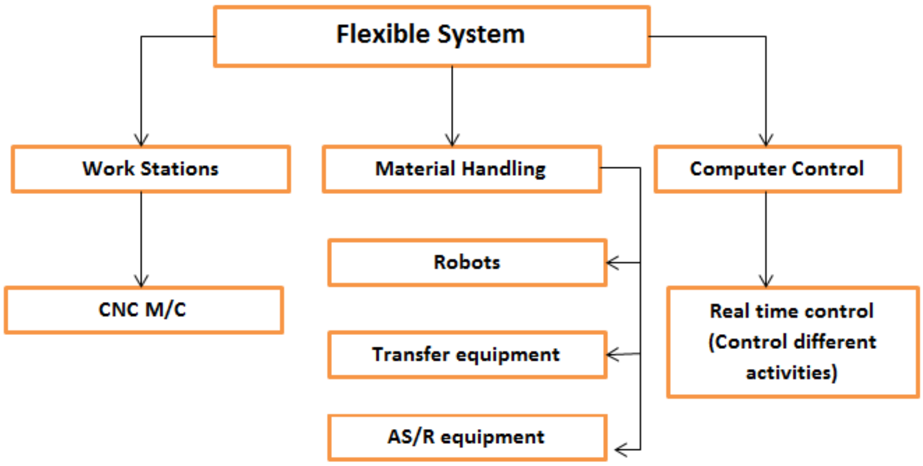

Figure 2.8: The flexible manufacturing system is more or less defined by its three main components. These are workplaces, material handling and a computer control. A workplace is any type of station on which a machine can work automatically. These include CNC milling machines and CNC drills to name a few. Material handling is the entire process from picking up the material in the warehouse, through transport to the workplace where the machine waits for the material, through loading and unloading stations where the material is loaded by robots, to automated storage and retrieval systems which ensure that the finished product arrives at the right place at the final storage location. The third and crucial component is the computer control, which takes over the control of all sub-systems and thus ensures that all systems work together smoothly and efficiently.

2.8 Material flow systems

The individual steps of production and distribution are linked to each other in a sequence of processes according to certain rules. Under simplifying assumptions, such a sequence of processes represents a production or logistics process as a whole or in parts as a flow process. Material is a generic term that can be used for all possible products in production. In addition, the use of the word flow must not hide the fact that the material is at rest most of the time in the process. Apart from unavoidable waiting times due to process technical reasons, the aim is to achieve the shortest possible throughput time for the material flow [2].

The simplest flow process is a serial one, i.e. one where one station comes after the other in a line. However, this process is usually not very economical and was therefore quickly replaced by so-called networks. The network structure has the advantage that processes not only run serially, but can also run in parallel. This has the advantage that waiting times can be shortened and that different process steps for the same kind of material can be realized.
Within the framework of material flow systems, highly complex processes are simplified in order to ensure the best possible overview of the processes. In addition, it is much easier to find possible problems in the process. However, networks also make the process more complicated. For example, a network process needs priority control for every type of material and buffer zones for waiting sections. At this point at the latest, it becomes apparent that the physical aspects of the material flow system are no longer sufficient, and the flow of information is just as important for making the right decisions. The localization and identification of material is crucial, i.e. where which material is located. This information can then be used to properly control the material flow system [3].

Figure 2.9, taken from Arnold and Furmans [2], shows the two types of material flow systems that were described before.

2.9 Summary

In this chapter, a literature review on all the necessary topics was done. Included were the basics of milk-run systems as well as some explanations on learning factories in general. Also a short introduction of the RFID technology was done.

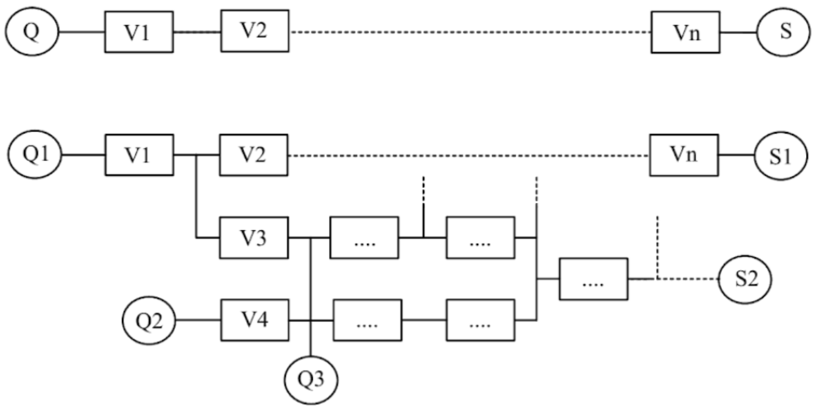

Figure 2.9: This figure shows the two types of material flow systems. The upper process is called a flow-line structure and is characterized by its simple single line (serial) architecture with a source (Q), a sink (S) and processes (V) in between. The lower process is called a network structure. Its key characteristic is that there are not only serial processes but also parallel ones. The advantage of this structure is that it is more flexible and can process similar material in different ways. To be able to operate a network structure some knowledge of the information flow is also needed. That means, the information of which material is where at what time is needed.

Then the direction was set on simulation basics. We discussed the differences between discrete event simulation and continuous simulation and also what an agent-based model is in this context. After that the principles of Monte Carlo methods were introduced to learn more of random numbers and their applications. Finally, we learned about flexible manufacturing and material flow systems in industry.

In summary, it can be said that there is a great deal of theoretical as well as practical research on these general topics. However, in the course of the research it became apparent that there is a research gap on digitized milk-run systems. Therefore, the next chapter will deal with this issue and try to shed some light on this gap and will also try to fill it.

3 Methods

Although there is a lot of scientific literature on inbound and outbound logistics, the subject of milk-run systems is unfortunately not as scientifically investigated [39]. Even fewer literature is available on digitized milk-run systems. Some papers on milk-run delivery systems, such as Bae et al. [6] mention on-demand systems in their work. These are systems that wait for a signal and are not designed to periodically check the workplaces to see if there is a need for a new parts delivery. Since this thesis presents work to realize an on-demand system, the papers related to it are the most interesting. Especially on-demand systems in combination with RFID technology should be mentioned here. However, current literature mentions RFID systems almost exclusively in combination with the traceability of materials and parts in an industrial context [26].

An agent-based model is used for the simulator in this thesis. One of the reasons for this decision was that Parunak et al. writes that agent-based simulations are better suited for modelling information processes, whereas equation-based simulations are better suited for modelling physical laws [58]. In order to find a suitable simulator for this project, the paper by Abar et al. is also used, which provides a clear decision-making aid for the selection [1].

3.1 Milk-run concept

An RFID system is used for the production system in the LEAD Factory, which knows at all times which employees are working on which product at which workplace. To achieve this, an RFID chip is mounted on each product and since each chip has a unique identifier, all data gathered from the product, e.g. its route through the assembly line and the time it needed to finish can be traced in the system. That is why it is possible to have full control over the assembly with the

collected data stored centrally on a server. The RFID system is not only there for production control, but is also used to detect faulty parts during the assembly process. For each group of parts in the assembly, there is an RFID chip that tells the system that there is a problem with the part when it is read by the antenna. With this data it is possible to generate information about the material used in the assembly process and its quality.
A much more detailed description of the RFID system and its functions in the LEAD Factory can be found in Chapter 3.5.1, where the concrete implementation of the system is described.

But that's not all the production system can be used for. The whole digitized milk-run idea is based on the fact that the RFID system can be uses to collect information about consumed materials or parts and thus make the milk-run more efficient.
The idea is that each box for the parts has an RFID chip and if the box is empty, the employee pushes it through under the table as before to have it replaced by a replenished box at the next milk-run. What is changing with the digitization is that the RFID antenna under the table now reliably recognizes the box and forwards the data (e.g. which part it is, the time and at which station it was read) to the central server.

The warehouse has a control station that sends a request every few seconds to the central server with the RFID data and asks if there is a new replenishment entry or order. If there is an order, it is automatically displayed on the logistics employee's screen as in Figure 3.3. If there is no order, the logistics employee sees an empty interface on the screen (Figure 3.2).
Initially, it was intended that the orders should be listed on the screen, as shown in Figure 3.1. However, this concept was discarded after further considerations, because you had to actively select the orders you wanted to pick next. So it was possible that some orders were never selected and the corresponding workplace was condemned to starvation. Furthermore, the lean principles say that a system should be as simple as possible. Therefore every move or click done by an employee that is not adding value to the product is a type of waste and needs to be thought over.
Thus, this concept was replaced by the one in Figure 3.3, which automatically selects the next job.

Since, the LEAD Factory only has 27 different parts to be replenished in the warehouse, the idea of using a grid came up. Thus, this concept now displays data in a

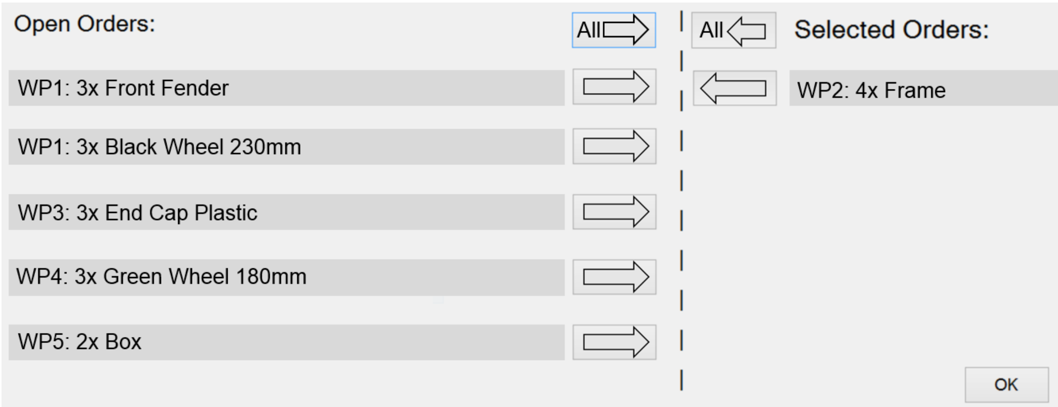

Figure 3.1: Early concept for a graphical user interface for the order handling as a list. This concept was discarded, because it was possible to make a simpler concept where the employee does not have to actively select orders. Nevertheless, with some modifications this concept would be suitable for larger factories that more frequent or a larger amount of orders.

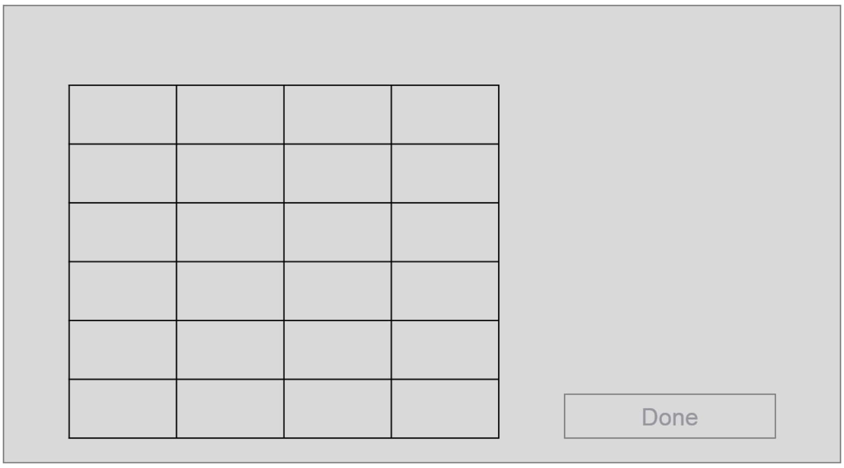

Figure 3.2: Concept for a graphical user interface for the order handling when it is empty. The logistics employee thus sees at first glance that there are currently no open orders, because there are no highlighted areas and the button is disabled. The System will automatically update the interface once an order is made.

Figure 3.3: A concept of how the graphical user interface of the order handling will look like. The grid on the left side of the figure represents the individual compartments on the shelf of the LEAD Factory warehouse. If the part is displayed in the top right corner of the grid, it means that the part in the warehouse is also located in the top right corner. It is therefore a first indication for the logistics employee to help him find the part. The number 3 in the green (active) grid means that three pieces of this part are to be picked. This number is determined by a Kanban system and stored in the database for each part. On the right side of the figure you can see which part needs to be picked. In this example a 'Black Wheel 180mm' should be picked. When the employee is finished, he or she clicks on the 'Done' button to inform the system that he or she wants to start the next milk-run.

grid format, which is a direct indicator of where the part to pick is on the shelf. If the part is in the upper right area, as shown in Figure 3.3, then this part is also on the shelf in the upper right area. The number 3 means that the logistics employee has to take three parts from the shelf. The respective number of parts was determined with a Kanban system and is also stored in the database for each part.

3.2 Pick-to-light concept

The idea behind a pick-to-light system is to make it as easy as possible for the logistics employee to pick the required part as quickly as possible. In order to make this process possible, a concept was developed that is based on visual perception. It is an ancient mechanism deeply anchored in the human being that draws our attention to light impulses and changes in the field of vision or perception of our environment [19].

As soon as the logistics employee has the current order on the screen, as already mentioned in Chapter 3.1 and as can be seen in Figure 3.3, the position of the part to be picked is displayed on the screen. A lamp is also attached to the shelf in the warehouse at every position from which picks can be made. When a lamp lights up, the logistics employee knows that he or she has to pick the amount of parts displayed on the screen at this point in the shelf.
The use of lamps on the shelf also enables a rapid increase in employee efficiency. Well trained employees may not need the pick-to-light system as much because they know every part and its position on the shelf by heart after a short time, but if we think of job starters, people with learning disabilities or those who find it difficult to remember such positions, a pick-to-light system is an optimal prerequisite for these employees to be able to deliver efficient and good work.
Once the employee has done this, he or she confirms the process on the screen and waits for further instructions.

3.2.1 Picking strategies

There are different strategies for picking, which are listed below. We will pick one of the below for implementation in the LEAD Factory.

1. The box with the RFID chip is picked up at the workplace and then refilled in the warehouse. This way you have the right material in the right box, which already has the right RFID chip on it.
2. There is a box in the warehouse for each box from each workplace with the appropriate RFID chip. The right box must be found and filled by picking. The empty box from the workplace is returned to the warehouse and stored there for the next order. One problem is that the employee first has to find the empty boxes in the warehouse, which is inefficient or generates more work.
3. There is a collection of boxes in the warehouse with the appropriate RFID chip. The boxes are already filled with the right amount of parts and only need to be taken and transported. The empty box is returned from the workplace when the logistics employee delivers the full box. One problem is that someone has to refill the empty boxes and then group them.
4. The required parts are picked in the warehouse and placed in temporary (enumerated) boxes. At the workplace, the correct boxes with the RFID chips are then refilled and the temporary boxes are returned to the warehouse. But it doesn't have to be a temporary box, it can also be the plain area on the milk wagon itself. We have 5 levels on the wagon for five different orders.

3.3 Route concept

Choosing the right route is a crucial factor for the effectiveness of the milk-run. No matter how good the concept of order management and picking is, if you waste the time you have gained when choosing a route again, you have no advantage.
It is therefore part of this work to calculate the best possible route for each milk-run so that the logistics employee can work as efficiently as possible on each tour. The value of the logistics employee is generated when picking and delivering the parts. Every second that the employee does not travel on the fastest route is wasted money for the company and may endanger the employee's job over a longer period of time.

The aim of the route computation is therefore to generate a route that is as fast and economical as possible. In order to get a comparison between an good route and a bad route, it is part of this thesis that a baseline method is implemented. This baseline method is, so to speak, an extremely bad variant to get from A to B. With this baseline method it is possible to compare the implemented algorithm to find a fast route and to highlight time differences between these two.

Figure 3.4 shows a possible representation of a route. The logistics employee or milkman (red) starts directly next to the warehouse (blue) and has the order to bring the required parts to workplace two or WP2 (green). Light grey field can be freely traveled whereas dark grey fields are obstacles of any kind (tables, machines, wall, etc.). If you assume that each field is the same size or need the same time for crossing, the best route is via WP4 and WP3 and not via WP5 and WP1 as indicated by the arrows. Back the same route would be reasonable, because it is the shortest path and there are no further orders to complete.

3.3.1 Baseline route concept

The baseline concept is intended to compare the result of the fast route finding with the baseline in order to highlight a meaningful result.
In concrete terms, the concept states:

- That the algorithm should visit every possible field until it has completed all orders.
- If a field offers several possibilities to go on, the next field to be visited should be decided randomly.
- Remember which fields have already been visited.
- If there are no more free surrounding fields, but still open orders, go back the path you came until at least one free field borders again.
- Once all orders have been completed, you go back the same route you came.

3.3.2 Route strategies

A placement of the parts on the wagon according to the order in which a fast route is taken in the factory can only be guaranteed if the orders are all in the system at the same time. If the orders come in individually and the logistics employee picks them immediately, then he or she has a certain sequence of parts already on the wagon, which he or she no longer wants to rearrange in order to enable correct arrangement for the best route. In other words, it is not always possible that the

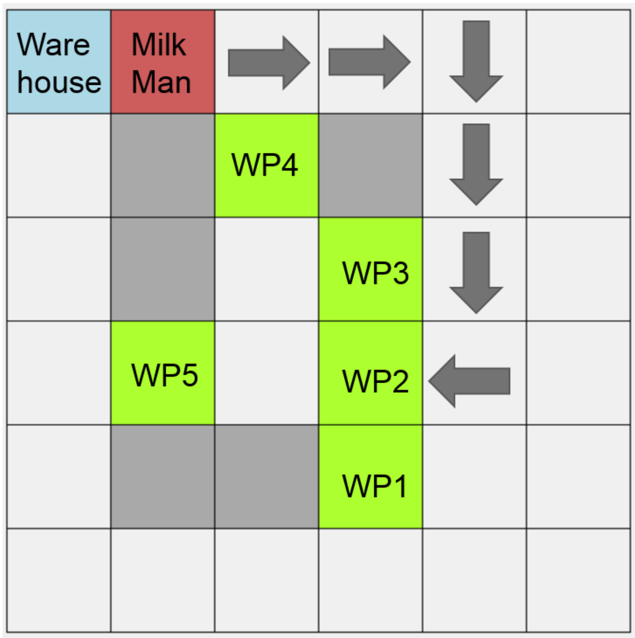

Figure 3.4: Concept for a visualization of the route through a factory. This visualization shows the LEAD Factory which is rather small compared to normal factories. The logistics employee or milkman (red) has the order to deliver parts from the warehouse (blue) to workplace two or WP2 (green). Light grey fields symbolize walkable fields whereas dark grey fields symbolize an obstacle like a table, a machine or a wall. Since the employee starts right next to the warehouse where he or she picked the parts, the best route is via WP4 and WP3, because the distance is less then via WP5 and WP1. Each field has exactly the same size. Therefore we can simply count the fields to evaluate the claim that this route is shorter. Since there are no more orders to fulfill the employee can take the same way back again to the warehouse where he or she waits for the next order.

parts for the first station are also on the first level of the wagon. It may be that these parts are on level four and the parts for the last station are on level two. That's why we have to differentiate between the case that all orders are known and picked according to the system and the case that it is almost coincidence which part is where on the wagon, because the orders come into the system one after the other and are picked immediately without waiting for further orders.

Here you can decide if you want to wait for a minimum number of orders until you start picking. Or if you wait for all five orders. So different strategies are possible. Below is a list of different strategies of which one should be used in the LEAD Factory:

1. Do not wait for further orders, but start picking as soon as the order is in the system. If there is no further order after that, start the milk-run immediately.
2. Do not wait for further orders, but start picking as soon as the order is in the system. Wait for the next order and pick immediately as well. Only start with the milk-run when the minimum number has been reached.
3. Do not wait for further orders, but start picking as soon as the order is in the system. Wait for the next order and pick immediately as well. Only start with the milk-run when the maximum number has been reached.
4. Wait for further orders until the minimum number of orders in the system is reached. Then immediately start the milk-run.
5. Wait for other orders until a minimum number of orders is reached in the system. Only start with the milk-run when the maximum number has been reached.
6. Wait for further orders until the maximum number of orders in the system is reached. Then immediately start the milk-run.

Alternatively, the minimum and maximum values can also be replaced by time values. This means that the logistics employee waits for example one minute after picking the first order before starting the milk-run.

In the case of the LEAD Factory, Strategy 2 is probably the most useful, because the factory is relatively small and you can wait for a minimum number of orders without having to wait a long time or being on the run for a long time. With a maximum of five orders it doesn't matter where the parts are on the wagon. The refilling at the workplaces is not more difficult through this.

In larger factories a different strategy can be more reasonable. But if one waits for the maximum, a possibility for an emergency order would be advisable. In the event that a workplace needs a part immediately and production would otherwise come to a halt.

3.4 Simulator concept

The simulator shall be realized as an agent-based model. This means that each workplace or worker is its own agent, which acts independently of all others but in connection to its environment and other agents. More details can be found in Chapter 2.5 about agent-based models in simulation.

It is supposed to be possible to give the simulator a time frame in which it simulates the complete milk-run. This results in a certain list of which parts will be the next orders to get into the system and be picked.

The feature of the simulator should be switchable in the main application. This means that the logistics employee can choose whether he or she wants suggestions for the next order from the simulator or not.
In addition, it should be possible to assign different profiles to the agents. For example, one should be able to say that agent number three, who works at workplace three, is a trainee and therefore does not know the procedures and processes in the assembly and requires accordingly longer to do his or her work. Thus different results emerge which can be compared with each other, or situations which can be tested without jeopardizing the work in the real factory.

The simulator should not be a deterministic process, but should always lead to a different result by probabilities within a certain frame. The probabilities required for the calculations can be stored locally or in a database. For example, there can be probabilities for a defect of any part or any tool or machine. Likewise, there are probabilities which indicate in what time frame the assembly per workplace moves.

3.5 Infrastructure implementation

In this chapter a further deepening of the concepts is planned. In the form of concrete implementations for these concepts the practical content of this thesis is highlighted.
In the respective chapters the practical implementation in the LEAD Factory as well as applied algorithms and methods will be explained in more detail.

3.5.1 RFID implementation

Figure 3.5 shows the conceptual process of the scooter assembly in the Digital State. Basically, five workplaces are installed, which are operated by four workers. Three workplaces (WP1, WP2, and WP5) are so-called jumper workplaces. The tasks there are carried out by two employees who constantly rotate between the three workplaces. Work stations WP3 and WP4 are stationary work stations. This means that the employees at these workplaces always remain at the same workplace and are only responsible for this workplace. The assembly process takes place in a U-cell arrangement using the line principle. The scooter assembly starts at WP1 and then continues to WP5, where the scooter is finally packed and placed on a pallet. The individual workplaces have already been precisely synchronized and the assembly process has been optimized according to Lean principles. The material is supplied to the workplace via slide racks. The employee pushes the empty parts boxes back under the shelf where an RFID antenna such as the left one in Figure 3.6 is placed. There they are picked up by the logistics employee who received the order and is refilling the boxes accordingly. At each workplace there is an additional RFID antenna such as in Figure 3.6 (the right device) which can be used to read in the individual tags on the scooters or the material boxes. Each scooter is provided with a tag at workplace WP1 and thus receives a unique identification number. Each event is then stored on this identification number in the subsequent assembly steps and made permanently available. The RFID tag remains permanently on the scooter.

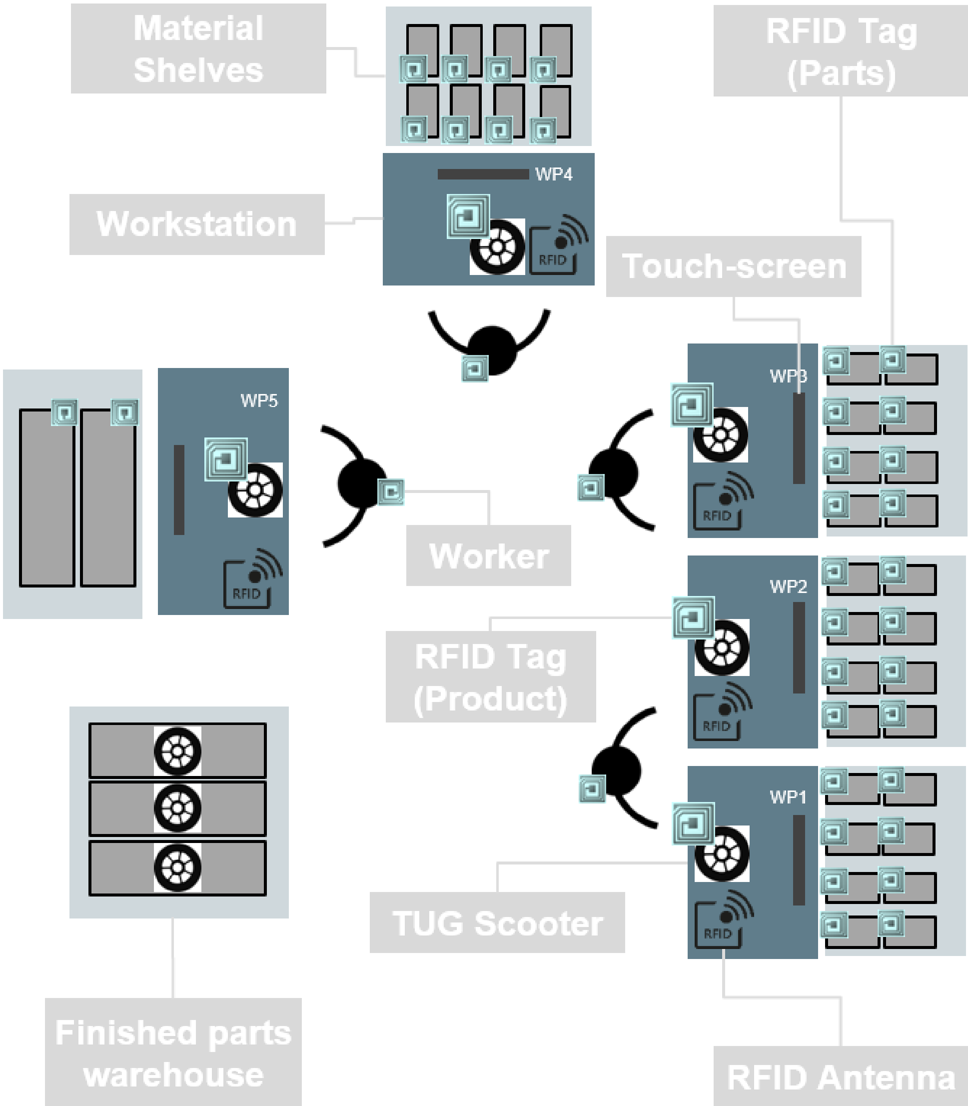

Figure 3.5: This figure shows the process of the digital state in the LEAD Factory. At five workplaces four employees are doing their job in a U-cell arrangement. The process starts at WP1 where the scooter is registered in the system with an RFID chip which is placed on a main part of the scooter. From WP1, the scooter continues its assembly process across all other workplaces until it is completed at WP5 and ready for packing. On its way, the scooter is registered at each station by the RFID system. Since WP1 determines which variant of the scooter is involved, all subsequent stations at which the scooter is checked in also know exactly which variant is involved. All the necessary data for the employees is displayed at the monitors at the workplaces. In addition, there are extra RFID antennas under each workplace, which are exclusively responsible for detecting used boxes that are tagged with RFID chips when they are pushed under the table.

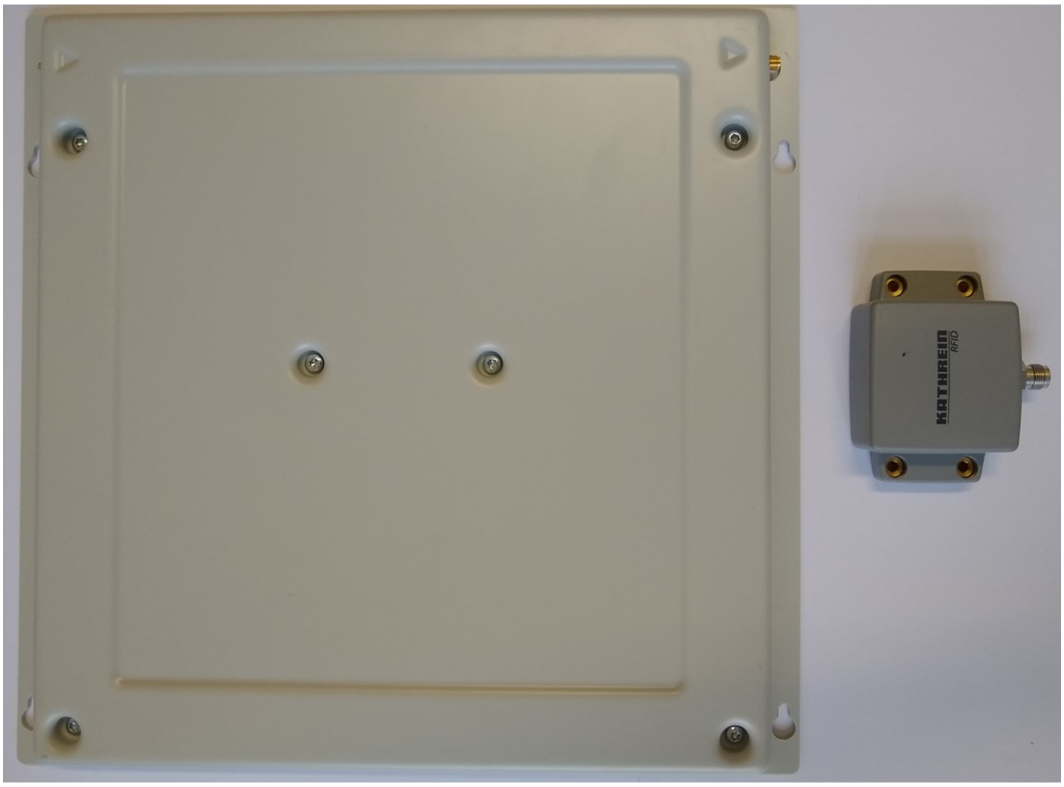

Figure 3.6: The left RFID antenna is an area antenna that scans a very focused rectangular area. It is used to detect the used boxes that are pushed under the workplace. In order for the antenna not to scan the wrong boxes, it was important that the scanning surface was very focused. To the right you can see the antenna that is used to detect the product RFID chips. This antenna has a semicircular detection area. The difference in size of these two antennas is clearly visible.

RFID functions

The RFID system in the LEAD Factory can be used in many different ways. The implemented functions include among others

- product tracking
- production control
- time recording
- production planning (Heijunka)
- the registration of damaged parts
- the replenishment entry

Most importantly for this thesis is the last item, the replenishment of parts.
The data that is stored in the database for the replenishment entry has the following structure.

- 'ID' - the ID of this entry in the database
- 'ProductID' - the ID of the scanned product or part
- 'ReadPointID' - the ID of the read point or workplace
- 'TimeStampEvent' - the time when the scanned product or part
- 'EndTime' - the time till the scanned product or part is needed
- 'Status' - the status of the scanned product or part
- 'ReadPointName' - the name of the read point or workplace
- 'ProductName' - the name of the scanned product or part
- 'ArticelNumber' - the article number of the scanned product or part

For this thesis, only the 'TimeStampEvent', the 'ReadPointName' and the 'ProductName' are important and therefore used. The remaining parts may be useful for future work but are not considered during the course of this thesis.

Why RFID was chosen

Some systems have limitations or advantages over other systems. Optical systems such as barcodes have the decisive disadvantage that the label with the code must always be clearly visible. Bar codes are therefore ruled out for larger quantities because it is cumbersome to place each part correctly for the bar code scan. In addition, neither the label nor the scanner lens must be dirty, as the scanner then cannot read the code correctly. Thus, the application is not suitable for dirty or

dusty industrial companies, as for example in the ore or mining industry. Another problem is that the scanner has to keep a relatively close distance to the barcode in order to guarantee optimal and reliable reading. However, this is not always possible and is often associated with some restrictions.
Optical systems are therefore not suitable for every industry and every location. Non-optical systems, on the other hand, have proven themselves in almost every industry.
The main advantages over optical systems are that the tag (which contains the code) does not have to be visible because the system works over radio waves and the tag can also be read through objects and obstacles. However, there are limitations in readability when reading through metal and liquids, because the electromagnetic waves are attenuated and shielded. Another advantage over optical systems is that several tags can be read simultaneously. On the hardware side, there is almost no upper limit for simultaneous reading. The limit for this is set in the software (in the reader firmware), since it has to resolve the signals of the individual transponders (anti-collision). In addition, depending on the system, it is also possible to read the tag from a great distance. The decisive factor here is which radio frequency is used for the RFID system [22]. In order to be able to offer a showcase system to as many industry representatives as possible and above all to enable the flexibility of the system required from an industry point of view, the underlying technology has been chosen for RFID.

In addition, it was probably also decisive that there was already a functioning RFID system implemented in the LEAD Factory at the start of this thesis and it was therefore very economical to use it. Using a different system would have meant that additional development work would have had to be put into this task.

3.5.2 Milk wagon

In order to make it possible for the logistics employee to transport the parts from the warehouse to the workplace as efficiently and without much effort as possible, he or she can use a wagon. Used names for such a wagon vary from company to company. In this work such a wagon will be called milk wagon, since it is used in the milk-run.

Figure 3.7 shows the milk wagon which is used in the LEAD Factory. As can be seen in the figure, the milk trolley has five storing spaces in which boxes can be

transported from the warehouse to the workplaces.
Since there are five levels on the milk wagon and the maximum number of orders is also five, it is planned that each order is placed on one level of the milk wagon. Previous training's have shown that the training participants mostly use the first (top) level of the wagon, because this level is at a comfortable height and offers a free area where the boxes can be placed. In order to get more data about the usability of the individual levels, it is important that the training participants are forced to use all levels and not only those which are most comfortable to reach. With the help of the acquired data, it is then possible to improve the design of the milk wagon, but this is not part of this thesis.
A milk wagon can support the principles of lean manufacturing by reducing certain types of waste. Namely:

- The waste of transportation, since the employee can transport more boxes on the wagon than if he or she would have to carry them.
- The waste of waiting time, since the transport with a wagon is faster and simultaneous deliveries of boxes saves time.
- And the waste from inventory, since you don't have to keep so many parts in inventory at the workplace, because the delivery is made on schedule by the milk wagon.

No adaptations or improvements were made to the milk wagon for this thesis. It should only be clarified what the task of such a wagon is and what it looks like. However, improvements are not excluded and some concepts can be found in Chapter 6.2 where future work is mentioned.

3.5.3 Supermarket

A supermarket is a warehouse in which the empty boxes of the workplaces are refilled or where the required parts are picked. In the LEAD Factory a simple shelf with five levels is used for this purpose. Altogether the supermarket contains 27 different parts, which are ready for picking. 23 parts are stored centrally on the shelf, one part is on the left and three on the right. The reason for storing parts left and right of the shelf is simply because they are too big for it. In Figure 3.8 you can see the supermarket as it is used in the LEAD Factory.

Figure 3.7: The milk wagon as used in the LEAD Factory is a device to transport material of all kind or boxes through the factory. It is supposed to support the logistics employee in his or her work. There are five levels on which the boxes can be placed. This makes it possible to transport a large number of boxes at the same time, which saves time and reduces costs. Milk wagons support lean manufacturing by reducing the seven types of waste. In this case it is mainly waste of transportation, waste from waiting time and waste from inventory.

Figure 3.8: The supermarket or warehouse is the central supply station for the LEAD Factory. It contains 27 different parts which are needed for the milk-run. There are 23 different parts directly on the shelf. One part is on the left side and 3 more are on the right side of the shelf. The reason for this is that some parts are too big for the shelf.

The following list shows the assignment to the respective workplaces as well as the sequence of assembly.

- Workplace 1
  - Front Fork Assembly
  - Front Fender - Plastic
  - Std Wheel (Black) - 230mm
  - Std Wheel (Green) - 230mm
  - Headtube
  - M24 Headset Nut
  - Slide Cylinder
  - Slide Bracket
- Workplace 2
  - Frame
  - Rear Brake
  - Brake Spring
  - Frame Plate
- Workplace 3
  - T-Bar
  - T-Bar Bushing - Plastic
  - Skewer Clip
  - Button Cover - Plastic
  - T-Bar Cap - Plastic
  - Steering Tube
  - Skewer Handle
  - Headtube Clamp
- Workplace 4
  - Handlebar Cap w/ Cord
  - Handle Bar w/ Grip
  - Handlebar Cap
  - Std Wheel (Black) - 180mm
  - Std Wheel (Green) - 180mm
  - Kickstand Assembly
- Workplace 5

- Box - Bold Wheel

For this thesis it is important to know which parts there are and in which order they are assembled in the assembly line, because it is the job of the simulator to recreate this sequence.
The supermarket itself becomes important for this thesis in combination with the pick-to-light system, which is further described in Chapter 3.7.

3.6 Milk-run implementation

The implementation of the graphical user interface (GUI) of the order handling system is very close to the concept described in Chapter 3.1. As already mentioned there, the orders originate directly from the employee at the workplace when he or she moves the empty box over the RFID antenna. The data is send to the central database where all RFID data is stored. The milk-run system in the warehouse of the LEAD Factory periodically checks the database for orders and processes them accordingly if there are any.

The orders are placed in a grid format on the display which has the same layout as the shelf in the warehouse of the LEAD Factory. This means that the part that is displayed in the second row on the right side of the screen can also be found on the shelf at exactly this position.

If there is no order in the system, the GUI is empty except for the grid and a disabled button as shown in Figure 3.9. This means that the logistics employee knows at a first glance that there are no orders at hand.

As soon as an order is received by the system, it is displayed as a button in the grid at the corresponding position. There are two types of buttons. Always only one active button that is colored (green) and greyed out buttons that stand for orders that are in the queue for picking. Each button shows the number of parts to pick, which has been defined in advance in the database by a Kanban system.
Once the logistics employee has picked the active (green) order, he or she has to click on the corresponding button on the GUI to confirm the execution to the system. The system automatically activates the next order, if there is one. This happens until the maximum number of orders has been picked, or the logistics

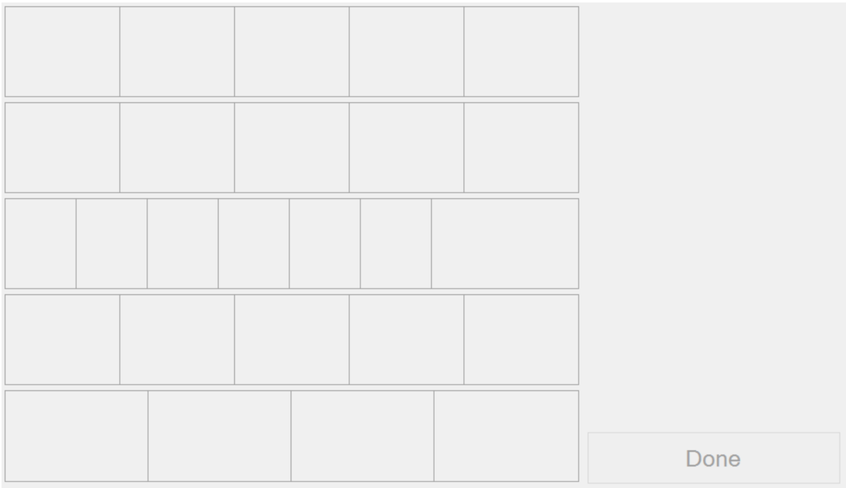

Figure 3.9: Graphical user interface for the order handling in the LEAD Factory when it is empty. The logistics employee knows at first glance that there is nothing to do because no order is highlighted on the display and the button for starting the milk-run is deactivated.

employee clicks the 'Done' button, which informs the system that the milk-run is about to start and that it should now provide a route for delivering the parts.

Figure 3.10 shows the GUI with an active order (green) and several orders in the queue (grey).

3.7 Pick-to-light implementation

The GUI was introduced in the last chapter. This chapter now deals with the pick-to-light system, which is controlled by the GUI whenever there is an active job in the system. This means that every time an order in the GUI becomes active, this order will be displayed in exactly the same position on the shelf by a light signal. This light signal is a light-emitting diode or LED.

In principle, the pick-to-light system works as follows. The database, the main software with the GUI and the pick-to-light controller are connected to the same network to exchange data. When the GUI activates a job, it automatically sends a signal to the pick-to-light controller with the information which LED's should light.

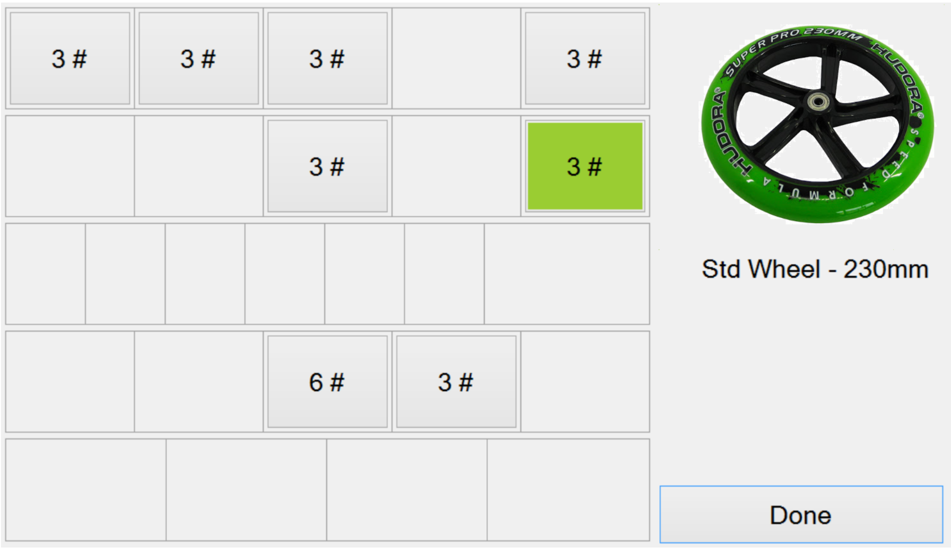

Figure 3.10: Graphical user interface for the order handling in the LEAD Factory. The grid in the left section of the figure shows exactly the same grid as can be found in the warehouse on the shelf. As soon as there is an order in the system, a button is created on the grid with the number of parts to be picked on which the logistics employee has to click to inform the system that this part has been picked. The green field is the active order which is also highlighted by the pick-to-light system on the shelf and is also specified in the right section of the figure with a picture and the name. Orders that are in the queue are displayed as grey buttons and will get active one after another. If the maximum number of orders has been picked or the logistics employee clicks on the 'Done' button, the system knows that the milk-run is supposed to start and switches to the next window, the proposed route.

To be exact, the controller is a small Wi-Fi (wireless network) micro-controller on which a server is installed and which listens for a specific HTTP (hypertext transfer protocol) request or command in the network. The specific command is the IP (internet protocol) address of the controller which was reserved in advance at the router, with the following keyword LED and two numbers, the start LED and the end LED. The LED's are controlled by the FastLED library [1], which allows each LED to be controlled individually with just a few lines of simple code.

For example: `http://192.168.0.105/led/1-5`
When this command is sent from the GUI system and received by the controller, it activates the LED's from 1 to 5 accordingly. Any combination of numbers is allowed as long as the second number is greater than or equal to the first number. If you want only one LED to light up, enter `5-5`. If you want to deactivate the LED's again, send the command OFF.

Summarized there are three different kind of commands that the controller is able to process:

- `http://192.168.0.105/led/5-5` activates LED number five
- `http://192.168.0.105/led/1-5` activates LED's one to five
- `http://192.168.0.105/led/OFF` deactivates all LED's

Table 3.1 lists all the corresponding addresses of the parts on the shelf. This information is stored in the database and is retrieved by the milk-run system as soon as the system is online. This information is essential to ensure the best possible picking. Although this information is so important, there is no requirement to store it locally. Because if, for whatever reason, there is no access to the database, there will be no access to the orders as well.

The controller that was used in this thesis is a NodeMCU 1.0 (ESP-12E module) as shown in Figure 3.11, which is taken from Amazon India [2]. The advantage of this series is that the Wi-Fi chip is integrated and the software can be programmed with the Arduino IDE [3] (integrated development environment). Numerous software examples are available to facilitate the usage of the device. The choice for this

[1] `http://fastled.io/` (Accessed on: 2018-11-30)
[2] `https://www.amazon.in` (Accessed on: 2018-11-30)
[3] `https://www.arduino.cc/en/Main/Software` (Accessed on: 2018-11-30)

Table 3.1: The table lists all parts on the shelf and the corresponding LED's that need to be addressed by the controller, sorted by the LED's addresses or the position on the shelf.

Part	**LED address**
Frame	1 - 20
Steering Tube	25 - 40
Slide Cylinder	65 - 70
Handlebar Cap w/ Cord	80 - 85
M24 Headset Nut	90 - 95
T-Bar	100 - 115
Handle Bar w/ Grip	120 - 135
T-Bar Bushing - Plastic	137 - 142
Button Cover - Plastic	144 - 149
Handlebar Cap	151 - 156
Skewer Clip	158 - 163
Skewer Handle	165 - 170
Headtube Clamp	172 - 177
Front Fork Assembly	180 - 195
Rear Brake	197 - 212
Brake Spring	214 - 219
Frame Plate	221 - 226
Std Wheel (Black) - 230mm	228 - 239
Headtube	240 - 255
Front Fender - Plastic	260 - 270
T-Bar Cap - Plastic	272 - 277
Kickstand Assembly	280 - 285
Slide Bracket	287 - 299

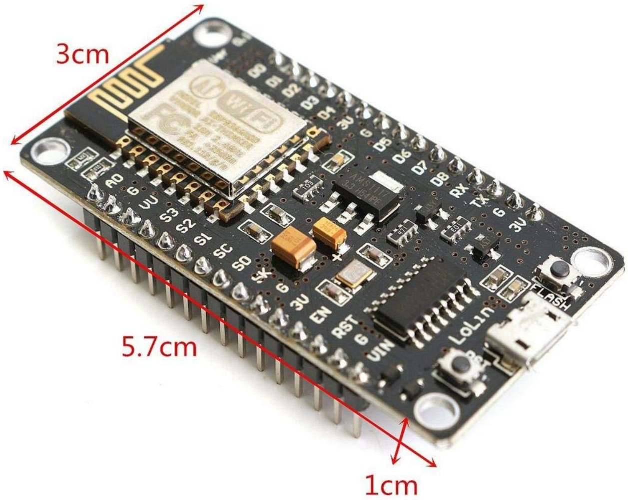

Figure 3.11: The controller of the pick-to-light system is a NodeMCU 1.0 (ESP-12E module). It is a small and compact controller with an integrated Wi-Fi chip, which makes it ideal for this application. In addition, the software can be flashed with the Arduino IDE and there are many sample programs ready to be tested, which makes using this controller very easy. A further criterion for the choice of this controller and an advantage is that it is comparatively cheap.

controller resulted from an excellent combination of price and performance and the recommendation of MakerAdvidor.com [4].

The used LED's listen to the 'WS2812B' protocol and are housed in a 'SMD 5050' package. They are delivered as strips and only need to be cut into the right size. The LED strip has three connectors. A positive connector (5 volts DC), a negative connector and a data connector which is responsible for the individual control of each LED. For this thesis five meters (one strip) or 300 LED's were used in the LEAD Factory. Figure 3.12, which is taken from Amazon [5], shows the LED strip that is used for the pick-to-light system.

The LED's are housed in aluminum bars, which provide protection against damage. Magnets are glued to the back of the bars to allow unfixed attachment to the shelf.

[4] https://makeradvisor.com (Accessed on: 2018-11-30)
[5] https://www.amazon.de (Accessed on: 2018-11-30)

Figure 3.12: The LED's used for the pick-to-light system. These are supplied as strips and only need to be cut to the required length. For use in the pick-to-light system, five strips with a length of one meter were required. The strip has three connections. A positive and negative connector for the power supply (5 volts DC) and a data connector through which the individual LED's are controlled. The LED's listen to the WS2812B protocol, which is responsible for the correct processing of the data from the data port. The LED's are housed in a SMD 5050 package.

This solution was chosen because it has the advantage over a fixed mounting that the bars can be easily removed for training purposes.

In order to supply the controller and the LED's with sufficient power, a suitable power supply was purchased, which meets all requirements. Since every LED needs 0.3 watts of power and in the worst case all 300 LED's have to light up, the power supply must have a minimum power of 90 watts. In order to be able to also supply the controller with sufficient power, a device with 100 watts has been purchased and can be seen in Figure 3.13, which is taken from Amazon [6]. This device has an output voltage of about five volts, whereby the exact value can be adjusted with a screw.

The schematic in Figure 3.14 shows how the different devices are connected to each other. All devices in the pick-to-light system are powered by one power source. The data pin can be assigned in the software to any free pin of the controller. In this case the data pin is pin 1 on the controller. It is also important that if you use another

[6] `https://www.amazon.de` (Accessed on: 2018-11-30)

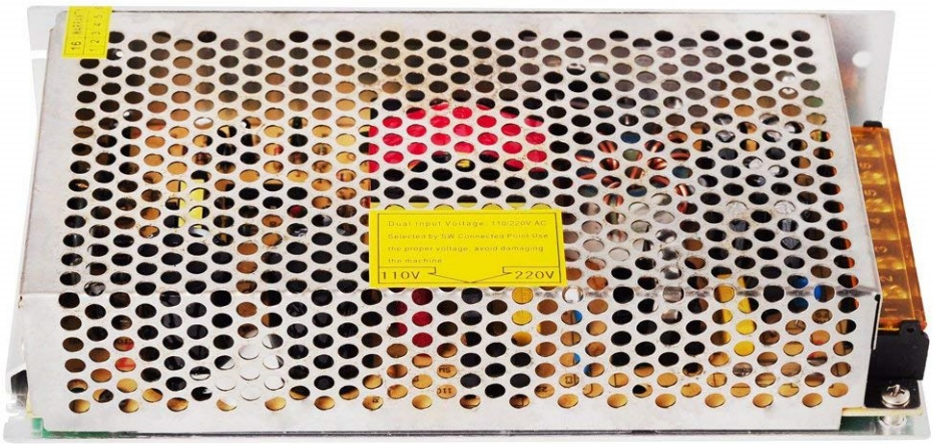

Figure 3.13: The power supply for the pick-to-light system. A version with 100 watt power was chosen, because in the worst case all 300 LED's with a total consumption of 90 watt (0.3 watt each) have to be supplied and additionally the controller has to be supplied with power. The output voltage is around 5 volts, whereby the exact output voltage can be adjusted with a screw.

power source for the controller, such as a USB cable, you have to connect the GND connector of the controller to the GND connector of the LED strip, otherwise the signal will be disturbed.

3.8 Route implementation

The previous chapter explained how to pick the parts efficiently. After the picking comes the part of the process that is no less important. The time saved by picking should not be lost during the transportation of the parts. Thus, it is important to always find the best possible route through a factory or to the appropriate workplaces. An experienced logistics employee in a small factory may have found a quick route after a short time, but what if it is a factory with hundreds of workplaces spread all over the place? In order to ensure that it does not matter whether the logistics employee knows the routes like the back of his hand, or whether he or she has his or her first day on the job, the route should always be as efficient as

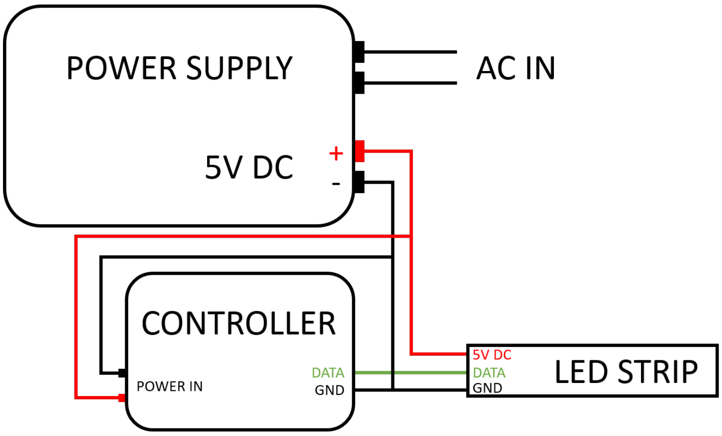

Figure 3.14: The schematic of all the wiring of the pick-to-light system. Both the controller and the LED's are powered from the same power source. The data connector is connected to a pin previously defined in the software, from which the information for controlling the respective LED's comes.

possible, regardless of the knowledge of the employee.
To ensure this, the milk-run system suggests a calculated route for the logistics employee at the start of each milk-run. This is achieved by calculating the route with the A* (A-star) algorithm. The A* algorithm was first published by Peter Hart et al. [33] at the Stanford Research Institute in 1968. This algorithm can be seen as a further development of Dijkstra's algorithm published in 1956 by Edsger Dijkstra [17].
The advantage of A* over the Dijkstra algorithm is that A* assumes that the shortest route from A to B is a direct one and therefore tries to get closer to the target as quickly as possible in a direct path. Only when A* is present at an obstacle does the algorithm continue to search for the path in breadth. This makes the A* algorithm faster and more efficient in most cases [68]. There are cases in which the Dijkstra algorithm or similar algorithms perform better [47], but for the purposes of this thesis special cases should not be of further importance.
A* was primarily developed for the autonomous navigation of mobile robots, but has conquered numerous areas of application where it is important to find a fast and efficient route due to its superior performance and simplicity [23, 66, 55, 61].

Figure 3.15 shows an example of a small factory with a warehouse (blue) in the upper left corner and the waiting position of the logistics employee (red) next to it. As already described in the route concept, the light grey fields are free fields and

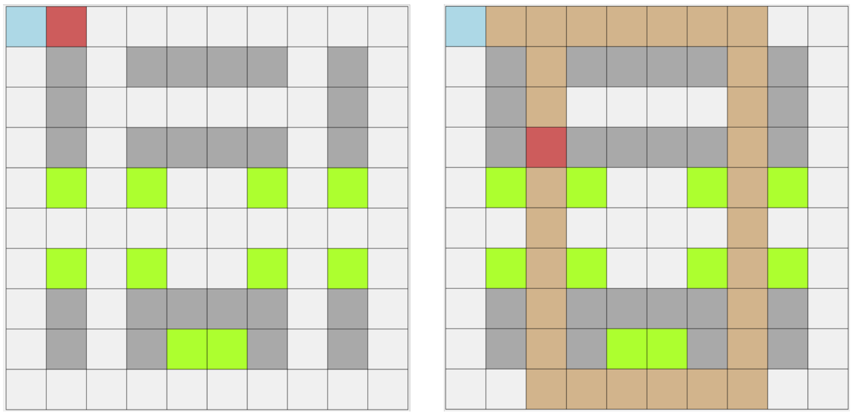

Figure 3.15: Example of a route using the A* algorithm on a small factory. As in the route concept, the colours describe various functions in the factory. The blue field is the warehouse. The red field is the logistics employee who carries out the milk-run. Light grey fields are free fields, whereas dark grey fields are obstacles. The green fields symbolize workplaces and the brown path the way from the start to the destination and back again. The left part shows the factory in its initial state and the right part shows the path calculated by the A* algorithm. In total there are 30 fields to be covered for this example.

the dark grey fields are obstacles of any kind. The green fields are symbolic for the workplaces in the factory and mark the many destinations of the milk-run. The example on the left shows the initial state of the factory and the example on the right shows the path calculated by the A* algorithm, when it has to go to all the stations in one trip.
As long as the free fields do not change, A* will always suggest the exact same route. The route from start to finish and back is a circle and requires only 30 fields and is therefore very effective.

The route calculation does also work with only one destination. However, if several destinations are selected, as is usually the case in a factory, as it would not be very economical for the logistics employee to supply only one workplace at a time, the A* algorithm behaves in such a way that, using the starting point, it always targets the closest destination. Alternatively, two other variants would be possible, namely that the execution would take place according to the numbering of the workplaces (for example, WP1 to WP3 to WP5) and that the execution would take place according to the chronological order of the order entry time. This means that the first new order in the system is also brought to the workplace first.

Although the laurels for the A* algorithm are numerous, it is good to be able to convince yourself of its performance. A baseline algorithm has been implemented to provide a useful comparison. This baseline algorithm is very simple and shows how A* performs in comparison. The baseline algorithm behaves as follows:

- It looks at all free direct neighboring fields and randomly selects one of them. It moves forward to the selected field and then searches for free neighboring fields again until all points of the previously defined route have been visited.
- If all points were visited, the algorithm goes back its way exactly as it came until it is at the beginning and the route calculation is completed.
- If a field has no free direct neighbours, the algorithm will go back until it encounters at least one free field and will start the random selection again.

Figure 3.16 finally shows the use of the baseline algorithm in the same scenario as the A* algorithm with the same conditions. Since the baseline algorithm relies heavily on randomness, there are both relatively good and very bad results and everything in between. As an example, the left part shows a relatively good path (brown fields) in which the algorithm has to travel 96 fields (keep in mind that the A* needed 30 fields). The right part shows a bad path, here the algorithm needs 308 steps to get to all the green workplaces and back again.

3.9 Simulator implementation

The implemented simulator is a non-deterministic procedure due to its processing of probabilities. This means that the result of the simulation differs from the previous one for each run. Randomness determines how long a workplace needs for the work to be performed (within a certain time frame) and whether there are any complications with the workpiece or with parts, etc., resulting in further delays.

Each workplace is regarded by the simulator as an independent agent. The properties of an agent require it to interact with its environment and with other agents. In the context of the LEAD Factory, for example, it is the case that certain dependencies exist on other workplaces. For example, workplace 3 has to wait for the intermediate product of workplace 2 to start work and workplace 4 has to wait for workplace 3 etc. This results in a process chain that stands still if there is a problem somewhere.

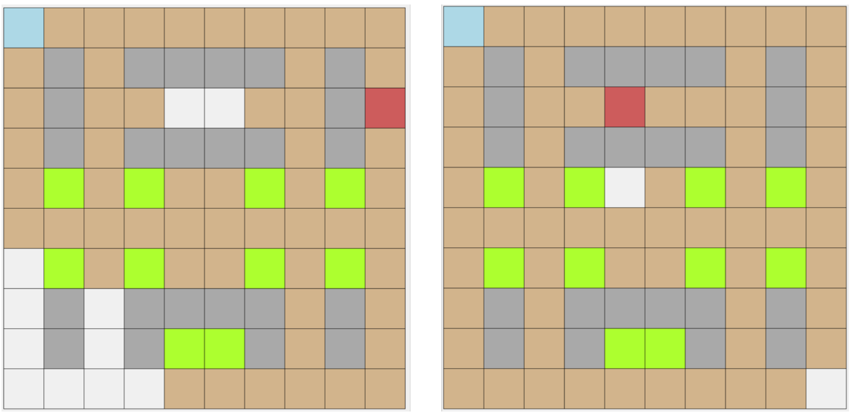

Figure 3.16: Example of a route using the baseline algorithm in a small factory. The brown fields describe the path from the start (blue field) to all the workplaces (green fields) and back, calculated by the algorithm. Since the baseline algorithm works strongly with randomness, both relatively good results and very bad results can be obtained. For comparison, the left part shows a calculated path of 96 fields and the right part a path with 308 fields. The complexity of the path can increase with the number of free fields and destinations.

Although there are countless frameworks for agent-based simulators, a separate simulation system was developed in this thesis. The reason for this decision is that most frameworks are oversized and it would take the same time to become familiar with these frameworks as to build a system of your own. Another reason for this decision is that the author with a self-development has full control and a deep understanding of the system. In addition, the system is adapted to the special requirements of the LEAD Factory and is designed for resource conservation and low-budget hardware, since the software should run on any Windows system.

In principle, the simulator can be activated in the GUI to simulate a certain time frame. Then the simulator delivers a list of upcoming orders and compares it with the already requested orders. If a simulated order has not yet been created in reality, the system proposes it to the employee and flags it as such. Should the real order be received during the proposal of the simulated order, the proposal will simply be overwritten by the real order.

The system works in such a way that every agent (workplace) can start working immediately when the simulator starts. So it is predefined that there is a workpiece that can be worked on, that all tools are available and that the parts are completely

filled according to the Kanban system. This is the default setting and can be changed at any time in any direction for any part.
At some point the time comes when a box for a certain part is empty and then an order is placed for that part. In the best case, the workplace always has additional boxes to be able to continue with the work. However, there are also cases where workplaces have only one box and if this box is empty, they have to wait for the logistics employee to refill it. Like the workplaces, the logistics employee is his or her own agent who exclusively takes care of the replenishment of the parts. He or she waits in the warehouse until an order arrives and he or she can set off with it. Each interaction, task and waiting time takes up the time of the corresponding agent and is tracked individually to ensure flawless interaction between the agents.

As already mentioned, there are possible dependencies on other agents. In the LEAD Factory, for example, only workplace 1 is free of dependencies on other workplaces because the workpiece is started here. If a problem occurs at this workplace and there is no finished workpiece in the buffer, the next workplace must also wait. This chain extends to the last station and can be demonstrated with the help of the simulator without affecting the real operation.

The probabilities for problems with the respective parts, which cause the working time to differ per simulation run, are listed in Table 3.2. The minimum probability is 0.1% and can be up to 5.0% for some parts, depending on how often there have been problems with that part in the LEAD Factory. The data was taken from the database and corrected from errors.

Probabilities for defects in machines and tools have been implemented, but are not considered in this thesis, because problems with them are so limited or non-existent in the LEAD Factory that it makes no sense to cover them.
What does matter, however, is the reliability of the RFID system. Since the entire digitized milk-run system depends on the RFID system, it must also be taken into account in the simulation. In addition, it is not uncommon for there to be problems with the RFID system. The most common of these is that at workplace 1 a problem occurs with the initial reading of the RFID chip on the product. Occasionally, there are also problems with the network via which the RFID system sends its data and requires a restart. These aspects are to be considered in the simulation and are listed in Table 3.3 with the corresponding probabilities.

In addition, the simulator makes it possible to simulate different employee profiles. A trained long-term employee will do his or her job faster than an employee who

Table 3.2: The table lists all parts in the process and the corresponding probability of a defect or a mishandling of the part by the employee, sorted by the occurrence in the process.

Part	**Defect probability in %**
Front Fork Assembly	0.1%
Front Fender - Plastic	0.5%
Std Wheel (Black) - 230mm	1.0%
Std Wheel (Green) - 230mm	1.0%
Headtube	0.2%
M24 Headset Nut	2.0%
Slide Cylinder	4.0%
Slide Bracket	1.0%
Frame	0.1%
Rear Brake	1.0%
Brake Spring	5.0%
Frame Plate	0.5%
T-Bar	0.1%
T-Bar Bushing - Plastic	1.0%
Skewer Clip	0.5%
Button Cover - Plastic	0.2%
T-Bar Cap - Plastic	0.5%
Steering Tube	1.0%
Skewer Handle	2.0%
Headtube Clamp	0.5%
Handlebar Cap w/ Cord	0.5%
Handle Bar w/ Grip	0.5%
Handlebar Cap	0.5%
Std Wheel (Black) - 180mm	1.0%
Std Wheel (Green) - 180mm	1.0%
Kickstand Assembly	0.5%
Box - Bold Wheel	2.0%

Table 3.3: The table lists occasional problems with the infrastructure and the corresponding probability, sorted by their probability.

Problem	Probability in %
Workplace 1 RFID reading	15.0%
Workplace 2 RFID reading	5.0%
Workplace 3 RFID reading	1.0%
Workplace 4 RFID reading	1.0%
RFID system is not responding	0.5%
Network is not responding	0.5%

has the first day of work and does not know his or her way around. In the context of the LEAD Factory, this means that a training participant has a different profile than an employee.

The times for the simulation of the work steps result from the respective probabilities of problems that can occur and the given time frame per part. The time frame has been recorded in the LEAD Factory by the author himself through repeated assembly.
Table 3.4 lists the time frame of each part in which a normal assembly should happen in the LEAD Factory.

3.10 Additional hardware implementation

The implementation part of the project was done in Microsoft Visual Studio 2015 [7] with C# 6.0 [8] and was developed on a HP Elite Book 840 with a 64bit Windows 8.1 Enterprise edition. The project runs smoothly on the computer it was developed on with an Intel Core i5-5200U CPU @ 2.20GHz and 8GB of RAM.
The LEAD Factory uses an Acer Switch 3 notebook, which is equipped with an Intel Pentium N4200 @ 1.10GHz and 4GB RAM and a 64bit Windows 10 edition. Thanks to the resource-saving development method, the project also runs without problems on this low-budget convertible notebook.

[7] https://visualstudio.microsoft.com/de/ (Accessed on: 2018-11-30)
[8] https://docs.microsoft.com/dotnet/csharp/ (Accessed on: 2018-11-30)

Table 3.4: The table lists all parts in the process and the time frame in which a normal assembly should happen, sorted by the occurrence in the process.

Part	Minimum time	Maximum time
Front Fork Assembly	2.5 seconds	3.0 seconds
Front Fender - Plastic	29.0 seconds	35.1 seconds
Std Wheel (Black) - 230mm	27.0 seconds	30.6 seconds
Std Wheel (Green) - 230mm	27.0 seconds	30.6 seconds
Headtube	5.2 seconds	6.2 seconds
M24 Headset Nut	31.6 seconds	41.0 seconds
Slide Cylinder	13.8 seconds	17.0 seconds
Slide Bracket	23.5 seconds	25.8 seconds
Frame	3.5 seconds	3.6 seconds
Rear Brake	3.1 seconds	3.3 seconds
Brake Spring	60.4 seconds	86.2 seconds
Frame Plate	40.8 seconds	42.8 seconds
T-Bar	3.2 seconds	3.3 seconds
T-Bar Bushing - Plastic	2.8 seconds	3.4 seconds
Skewer Clip	1.8 seconds	2.4 seconds
Button Cover - Plastic	7.8 seconds	8.9 seconds
T-Bar Cap - Plastic	12.3 seconds	17.9 seconds
Steering Tube	12.2 seconds	14.9 seconds
Skewer Handle	13.9 seconds	21.8 seconds
Headtube Clamp	24.4 seconds	24.6 seconds
Handlebar Cap w/ Cord	2.4 seconds	2.8 seconds
Handle Bar w/ Grip	7.5 seconds	9.4 seconds
Handlebar Cap	46.7 seconds	66.3 seconds
Std Wheel (Black) - 180mm	48.4 seconds	51.6 seconds
Std Wheel (Green) - 180mm	48.4 seconds	51.6 seconds
Kickstand Assembly	33.5 seconds	34.4 seconds
Box - Bold Wheel	25.0 seconds	25.4 seconds

3.11 Summary

In this chapter, which is by far the most comprehensive in this thesis, all relevant concepts and implementations were discussed and documented.

Starting with the milk-run concept on a large scale, in which the functionality and appearance of the software has been largely defined and delimited, through the pick-to-light concept, which has clarified the purpose and advantages of visual assistance for the logistics employee and thus also for the process itself, and which strategies there are for picking in general and in the context of the LEAD Factory. Furthermore, the concept of a route calculation, the idea behind the use of a baseline route algorithm and different strategies for transport and waiting on the start were presented. Finally, a simulation concept was presented, in which the basic principles of the implementation are already laid down.

The second part of this chapter deals with the concrete implementations, which were realized within the framework of this thesis and which strongly follow and adhere to the previously mentioned concepts. When implementing concepts one sometimes comes across better ideas or insights that it doesn't work as expected.

In the implementation of the milk-run one thing must not be forgotten: the infrastructure that makes this whole process possible in the first place. In other words, the RFID system. That's why an extensive part of this section is dedicated to explaining to the reader the functionality and features of the RFID system in the LEAD Factory. The process behind it as well as the antennas and functions are introduced and it is explained why this system was the chosen one. After that, things are discussed that do not seem significant at first glance, but in reality play a very large role in the feasibility of this thesis. The milk wagon or the transport device and the supermarket or the warehouse for the parts in the LEAD Factory should be mentioned for that. They represent the backbone of the LEAD Factory and their importance is to be emphasized here once again, since the pick-to-light system was implemented in the warehouse, for example. Furthermore, the interaction of the picking software, the controller that establishes the connection with the pick-to-light system and the calculated route was explained in more detail. In addition, the different algorithms used for the route were examined more closely and insights were given into their respective functions. Finally, the implementation of the simulator made clear which role randomness plays in the calculation of the results and how this can be influenced by the employee. The standard times and probabilities for failures, defects, etc. were listed.

Despite all this complexity and many decisive factors, a suitable project was set up in this chapter that hopefully meets the requirements and can deliver results. These results are now set out in Chapter 4 hereafter.

4 Results

After the previous chapters discussed how the simulator and its components are designed and how they work, this chapter now presents the results. Specifically, data for route calculation are quantified by the different algorithms, pick-to-light and refill data are compared and finally the results of the simulation are listed.

4.1 Route calculation results

The results in Table 4.1 list the calculated path in the LEAD Factory from the warehouse to each individual workplace and back. The algorithm used is the baseline algorithm that has been implemented to provide a meaningful comparison to the A* algorithm. Since the baseline algorithm is based heavily on chance, different numbers of runs have been made in order to smooth the result. As shown in the table, up to one million calculations were made and the average was calculated to obtain a meaningful value. The resultant numbers indicate the number of fields the logistics employee has to travel from the start to the goal and back again. One field equals one meter in the LEAD Factory. The average speed in the LEAD Factory is 0.72 m/s, equivalent to 2.59 km/h. Or to put it another way, to cover one meter (equivalent to one field in the layout), the logistics employee needs 1.388 seconds with the transportation device. Figure 4.1 shows the layout of the LEAD Factory as it is used to calculate the path.

The results in Table 4.2 list the calculated path computed with the A* algorithm under the same conditions as before with the baseline algorithms. The same starting point and destination were calculated and the same number of iterations were performed.

Table 4.1: The table lists the results of the baseline algorithm in the context of the LEAD Factory. The algorithm is executed once, 10,000 times, 100,000 times and 1 million times respectively and the average is calculated. The Time column contains the time it takes to get to the destination and return at an average speed of 0.72 m/s based on the result of one million runs.

Route to	1x	10.000x	100.000x	1.000.000x	Time
Workplace 1	22	19	19	19	26.4 seconds
Workplace 2	26	19	18	18	25.0 seconds
Workplace 3	14	17	16	16	22.2 seconds
Workplace 4	6	11	8	8	11.1 seconds
Workplace 5	10	10	12	12	16.7 seconds

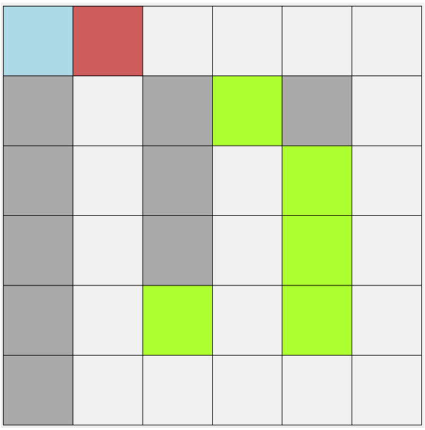

Figure 4.1: Layout of the LEAD Factory used to calculate the route. The different coloured fields describe various functions in the factory. The blue field represents the warehouse. The red field represents the logistics employee who carries out the milk-run. Light grey fields represent free fields on which the logistics employee is allowed to move, whereas dark grey fields represent obstacles of any kind, like walls, tables or similar objects. The green fields are the single workplaces to which the logistics employee has to travel in order to fulfill a replenishment order.

Table 4.2: The table lists the results of the A* algorithm in the context of the LEAD Factory. The algorithm is executed once, 10,000 times, 100,000 times and 1 million times respectively and the average is calculated. The Time column contains the time it takes to get to the destination and return at an average speed of 0.72 m/s.

Route to	**1x**	**10.000x**	**100.000x**	**1.000.000x**	**Time**
Workplace 1	16	16	16	16	22.2 seconds
Workplace 2	14	14	14	14	19.4 seconds
Workplace 3	12	12	12	12	16.7 seconds
Workplace 4	4	4	4	4	5.6 seconds
Workplace 5	8	8	8	8	11.1 seconds

4.2 Time measurement results

Different time measurements are needed for the evaluation of the simulator. This includes, for example, the time required to pick the parts in the warehouse, as well as the time required to refill the parts at the respective workplaces. Therefore, the times required for this are listed below.

4.2.1 Pick-to-light results

In order to evaluate the performance of the simulator we need to measure the time it takes to pick every part. Since we have different scenarios in the LEAD Factory, it is necessary to take this into account in the measurements as well. Therefore, the required picking times of an expert and an amateur were recorded.

Table 4.3 lists the results of the picking process done by an expert who knows already where the parts are placed on the shelf. The picking times are measured once with the pick-to-light deactivated and once with the pick-to-light activated.

As a comparison, Table 4.4 lists the results of the picking process done by an amateur who does the picking for the first time and does not know there the parts are placed or how they look. The picking times are measured once with the pick-to-light deactivated and once with the pick-to-light activated, same as with the expert.

Table 4.3: The table lists all parts in the process and the corresponding picking time an expert needs with the pick-to-light system deactivated and activated, sorted by the occurrence in the process.

Part	**Exp. picking time**	**Exp. pick-to-light time**
Front Fork Assembly	6.6 seconds	7.2 seconds
Front Fender - Plastic	5.2 seconds	5.5 seconds
Std Wheel (Black) - 230mm	9.1 seconds	6.5 seconds
Std Wheel (Green) - 230mm	9.1 seconds	6.5 seconds
Headtube	7.0 seconds	7.4 seconds
M24 Headset Nut	6.8 seconds	6.2 seconds
Slide Cylinder	7.4 seconds	6.8 seconds
Slide Bracket	9.2 seconds	5.9 seconds
Frame	7.6 seconds	9.8 seconds
Rear Brake	6.7 seconds	6.0 seconds
Brake Spring	5.4 seconds	5.8 seconds
Frame Plate	7.7 seconds	7.5 seconds
T-Bar	6.9 seconds	7.7 seconds
T-Bar Bushing - Plastic	6.9 seconds	5.6 seconds
Skewer Clip	7.8 seconds	6.0 seconds
Button Cover - Plastic	5.9 seconds	5.8 seconds
T-Bar Cap - Plastic	5.8 seconds	6.6 seconds
Steering Tube	7.4 seconds	7.3 seconds
Skewer Handle	6.2 seconds	4.8 seconds
Headtube Clamp	5.8 seconds	6.0 seconds
Handlebar Cap w/ Cord	7.3 seconds	9.5 seconds
Handle Bar w/ Grip	11.4 seconds	10.4 seconds
Handlebar Cap	5.6 seconds	5.5 seconds
Std Wheel (Black) - 180mm	7.3 seconds	7.5 seconds
Std Wheel (Green) - 180mm	7.3 seconds	7.5 seconds
Kickstand Assembly	6.0 seconds	6.3 seconds
Box - Bold Wheel	8.4 seconds	6.2 seconds

Table 4.4: The table lists all parts in the process and the corresponding picking time an amateur (beginner) needs with the pick-to-light system deactivated and activated, sorted by the occurrence in the process.

Part	**Beg. picking time**	**Beg. pick-to-light time**
Front Fork Assembly	7.2 seconds	6.9 seconds
Front Fender - Plastic	8.2 seconds	5.4 seconds
Std Wheel (Black) - 230mm	12.0 seconds	8.0 seconds
Std Wheel (Green) - 230mm	12.0 seconds	8.0 seconds
Headtube	26.4 seconds	7.1 seconds
M24 Headset Nut	13.5 seconds	6.0 seconds
Slide Cylinder	17.9 seconds	7.0 seconds
Slide Bracket	9.6 seconds	6.8 seconds
Frame	12.6 seconds	8.0 seconds
Rear Brake	18.6 seconds	6.6 seconds
Brake Spring	4.9 seconds	6.0 seconds
Frame Plate	4.8 seconds	7.7 seconds
T-Bar	30.0 seconds	7.3 seconds
T-Bar Bushing - Plastic	8.9 seconds	6.0 seconds
Skewer Clip	6.4 seconds	7.1 seconds
Button Cover - Plastic	7.9 seconds	6.1 seconds
T-Bar Cap - Plastic	8.6 seconds	7.0 seconds
Steering Tube	9.3 seconds	8.0 seconds
Skewer Handle	11.7 seconds	5.3 seconds
Headtube Clamp	15.4 seconds	6.2 seconds
Handlebar Cap w/ Cord	16.9 seconds	8.9 seconds
Handle Bar w/ Grip	11.3 seconds	10.5 seconds
Handlebar Cap	4.1 seconds	5.2 seconds
Std Wheel (Black) - 180mm	7.0 seconds	7.0 seconds
Std Wheel (Green) - 180mm	7.0 seconds	7.0 seconds
Kickstand Assembly	7.8 seconds	6.3 seconds
Box - Bold Wheel	22.9 seconds	7.2 seconds

4.2.2 Refill results

The milk-run process consists not only of picking the parts and transporting them, but also of refilling the box at the workplace. It is therefore important to measure the times of the refilling process and take them into account when evaluating the simulator.

Table 4.5 lists the results of these time measurements. The times were measured as an expert who knew where each part belonged replenished the parts, and the times were measured as an amateur who did not know where the parts belonged replenished them.

4.2.3 Box retrieval results

Retrieving the empty boxes from under the table of the workplace may only be a small part of the total time, but to ensure a holistic evaluation, this part must still be considered in the process. For this reason, the average times of five individual runs in the LEAD Factory are listed in Table 4.6 hereafter.

4.3 Simulation results

Finally, we come to the centerpiece of this thesis, the results of the milk-run simulator.
In order to produce comparable results, the simulator has been set so that it only has the part to be measured in a small number in stock and thus runs out within a few minutes in the simulation. This ensures that only one part runs out at a time and the logistics employee can set off immediately without waiting for further parts. In addition, there was no waiting time, as the next order did not arrive until the logistics employee was back in the warehouse. In other words, the minimum and maximum number of orders was one.

The result of the milk-run simulator consists of the time needed to pick the part (2 to 5 seconds for a maximum of 3 parts and 4 to 10 seconds for more parts), the time to get from the warehouse to the workplace (3 to 10 seconds), the time to pick up the empty box from under the workplace (2 to 4 seconds), the refilling of this box

Table 4.5: The table lists all parts in the process and the corresponding time an expert and an amateur (beginner) need to refill the part at the workplace, sorted by the occurrence in the process.

Part	Exp. refill time	Beg. refill time
Front Fork Assembly	5.0 seconds	9.5 seconds
Front Fender - Plastic	3.4 seconds	9.4 seconds
Std Wheel (Black) - 230mm	3.4 seconds	8.2 seconds
Std Wheel (Green) - 230mm	3.4 seconds	8.2 seconds
Headtube	5.8 seconds	9.4 seconds
M24 Headset Nut	3.9 seconds	8.8 seconds
Slide Cylinder	3.1 seconds	6.6 seconds
Slide Bracket	5.5 seconds	7.6 seconds
Frame	12.8 seconds	19.0 seconds
Rear Brake	3.9 seconds	6.2 seconds
Brake Spring	4.2 seconds	6.6 seconds
Frame Plate	4.2 seconds	6.4 seconds
T-Bar	5.8 seconds	6.6 seconds
T-Bar Bushing - Plastic	4.7 seconds	7.6 seconds
Skewer Clip	4.2 seconds	7.3 seconds
Button Cover - Plastic	4.1 seconds	7.0 seconds
T-Bar Cap - Plastic	3.9 seconds	8.2 seconds
Steering Tube	11.8 seconds	22.2 seconds
Skewer Handle	4.6 seconds	8.1 seconds
Headtube Clamp	3.6 seconds	7.4 seconds
Handlebar Cap w/ Cord	4.4 seconds	8.2 seconds
Handle Bar w/ Grip	4.3 seconds	6.3 seconds
Handlebar Cap	3.6 seconds	6.9 seconds
Std Wheel (Black) - 180mm	3.4 seconds	8.4 seconds
Std Wheel (Green) - 180mm	3.4 seconds	8.4 seconds
Kickstand Assembly	4.4 seconds	6.7 seconds
Box - Bold Wheel	10.1 seconds	18.6 seconds

Table 4.6: The table lists the average time it takes an employee at the LEAD Factory to retrieve the empty boxes from under the workplace, sorted by the time required.

Run number	**Retrieval time**
Run number 1	2.8 seconds
Run number 2	3.0 seconds
Run number 3	3.2 seconds
Run number 4	3.2 seconds
Run number 5	3.4 seconds

(3 to 6 seconds) and the way back to the warehouse (3 to 10 seconds). The times are chosen randomly in the respective time frame.

Table 4.7 lists the results of the simulator from the randomly selected times within the previously described time frame. In addition, you can find the respective times that an expert and an amateur need for the milk-run without using the digitized milk-run system. It is important to note that these times are calculated without the periodic waiting time normally used by a milk-run system to travel the workplaces at fixed intervals. The times for the real measurements consist of the distance from the warehouse to the workplace to get the box, the retrieval of the box from underneath the workplace, the picking process, the transport of the full box back to the workplace, the refilling of the workplace and the way back to the warehouse.

Table 4.8 lists the same results of the simulator as in 4.7, which are from the randomly selected times within the previously described time frame. In addition, you will find the respective times that an expert and an amateur need for the milk-run using the digitized milk-run system. The times for the real measurements are composed of the picking process in the warehouse, the transport of the parts to the workplace, the retrieval of the box from under the workplace, the refilling of the box and the way back to the warehouse.

Table 4.7: The table lists all parts in the process and the corresponding time an expert, an amateur (beginner) and the simulation need to fulfill the order without using the digitized milk-run system, sorted by the occurrence of the part in the process.

Part	**Sim. time**	**Exp. time**	**Beg. time**
Front Fork Assembly	22.0 seconds	66.2 seconds	71.4 seconds
Front Fender - Plastic	17.2 seconds	63.2 seconds	72.3 seconds
Std Wheel (Black) - 230mm	17.6 seconds	67.2 seconds	75.3 seconds
Std Wheel (Green) - 230mm	18.1 seconds	67.2 seconds	75.3 seconds
Headtube	24.7 seconds	67.5 seconds	90.5 seconds
M24 Headset Nut	20.5 seconds	65.4 seconds	77.0 seconds
Slide Cylinder	17.3 seconds	65.2 seconds	79.2 seconds
Slide Bracket	21.3 seconds	69.4 seconds	71.9 seconds
Frame	21.8 seconds	75.1 seconds	86.2 seconds
Rear Brake	19.0 seconds	65.3 seconds	79.5 seconds
Brake Spring	20.4 seconds	64.3 seconds	66.2 seconds
Frame Plate	21.5 seconds	66.6 seconds	65.8 seconds
T-Bar	14.0 seconds	67.4 seconds	91.3 seconds
T-Bar Bushing - Plastic	19.3 seconds	66.3 seconds	71.2 seconds
Skewer Clip	17.7 seconds	66.7 seconds	68.4 seconds
Button Cover - Plastic	21.8 seconds	64.7 seconds	69.6 seconds
T-Bar Cap - Plastic	22.3 seconds	64.4 seconds	71.5 seconds
Steering Tube	20.0 seconds	73.9 seconds	86.2 seconds
Skewer Handle	20.6 seconds	65.4 seconds	74.5 seconds
Headtube Clamp	19.5 seconds	64.1 seconds	77.5 seconds
Handlebar Cap w/ Cord	13.7 seconds	66.4 seconds	79.9 seconds
Handle Bar w/ Grip	16.9 seconds	70.4 seconds	72.3 seconds
Handlebar Cap	16.7 seconds	63.8 seconds	65.7 seconds
Std Wheel (Black) - 180mm	20.3 seconds	65.4 seconds	70.1 seconds
Std Wheel (Green) - 180mm	15.4 seconds	65.4 seconds	70.1 seconds
Kickstand Assembly	21.3 seconds	65.1 seconds	69.3 seconds

Table 4.8: The table lists all parts in the process and the corresponding time an expert, an amateur (beginner) and the simulation need to fulfill the order using the digitized milk-run system, sorted by the occurrence of the part in the process.

Part	Sim. time	Exp. time	Beg. time
Front Fork Assembly	22.0 seconds	37.5 seconds	41.7 seconds
Front Fender - Plastic	17.2 seconds	34.1 seconds	40.0 seconds
Std Wheel (Black) - 230mm	17.6 seconds	35.2 seconds	41.9 seconds
Std Wheel (Green) - 230mm	18.1 seconds	35.2 seconds	41.9 seconds
Headtube	24.7 seconds	38.5 seconds	41.9 seconds
M24 Headset Nut	20.5 seconds	35.4 seconds	40.0 seconds
Slide Cylinder	17.3 seconds	35.3 seconds	38.9 seconds
Slide Bracket	21.3 seconds	36.7 seconds	39.7 seconds
Frame	21.8 seconds	45.2 seconds	49.2 seconds
Rear Brake	19.0 seconds	32.4 seconds	35.3 seconds
Brake Spring	20.4 seconds	32.5 seconds	35.1 seconds
Frame Plate	21.5 seconds	34.1 seconds	36.6 seconds
T-Bar	14.0 seconds	33.3 seconds	33.7 seconds
T-Bar Bushing - Plastic	19.3 seconds	30.0 seconds	33.4 seconds
Skewer Clip	17.7 seconds	30.0 seconds	34.2 seconds
Button Cover - Plastic	21.8 seconds	29.7 seconds	32.9 seconds
T-Bar Cap - Plastic	22.3 seconds	30.3 seconds	35.0 seconds
Steering Tube	20.0 seconds	38.9 seconds	50.0 seconds
Skewer Handle	20.6 seconds	29.1 seconds	33.2 seconds
Headtube Clamp	19.5 seconds	29.4 seconds	33.4 seconds
Handlebar Cap w/ Cord	13.7 seconds	22.6 seconds	25.8 seconds
Handle Bar w/ Grip	16.9 seconds	23.4 seconds	25.6 seconds
Handlebar Cap	16.7 seconds	17.8 seconds	20.8 seconds
Std Wheel (Black) - 180mm	20.3 seconds	19.6 seconds	24.0 seconds
Std Wheel (Green) - 180mm	15.4 seconds	19.6 seconds	24.0 seconds
Kickstand Assembly	21.3 seconds	19.3 seconds	21.7 seconds

4.4 Summary

This chapter has listed all the results needed to answer the research question, which will be answered in Chapter 5 afterwards. Concrete time measurements of the route in the LEAD Factory were carried out as well as general time measurements for the pick-to-light system or picking without this system and refilling the workplaces with the boxes. These times were then used to determine the complete order fulfillment time for each part and to compare this time with the predicted or simulated time of the simulator.

5 Discussion

In this chapter, the three research questions are addressed using the results from Chapter 4 and the results are interpreted and evaluated in detail.
Then the lessons that were learned in the course of this thesis will be discussed and the limitations of the system will be explained, as this thesis does not claim to be complete. In the course of the execution of the project there were countless changes, ideas for improvement and insights as to how the milk-run system cannot and should not work. These considerations will also exist in the future, since a project on this scale and in the context of the digitization of industrial processes will probably never manage without improvements. Therefore, only the research questions will be answered and there will be an attempt not to deviate too generally and too far from the topic at hand.

5.1 Answering the research questions

On the following pages, the research questions will be discussed and then answered in order to bring this project to a conclusion. Although a project of this kind is never really finished and there is still something to do, it is good to have a certain limitation in the form of these research questions.

5.1.1 Research question 1

Research question number one is the following:

- Can we use the simulator to find improvements in the milk-run system compared to a real-life scenario?

This question is essential and was chosen because it is important for the thesis to know whether the simulator or the simulation of a digitized milk-run system has any advantage over the old, non-digitized milk-run system. If this is not the case and the non-digitized system functions better or more efficiently than the digitized system, then the further pursuit of the project would be completely irrelevant and would not satisfy the scientific claim to improve the system.
That's why it was important not only to build a simulator, but also to make a comparison available to see how good the promised performance of the simulator is. And that's where the LEAD Factory comes in. With the help of this model learning factory, it is possible to measure exactly this process, namely the non-digitized milk-run, which was already an important part of the learning factory because it gave the training participants a direct experience of the lean principles.

Since the range of training participants is very large, it was important to have not only one comparison value to the simulator result, but two. Namely the value of an expert who knows the LEAD Factory like the back of his hand and the value of an amateur who has never done a single thing in the LEAD Factory before and is therefore supposed to be representative for all training participants, since the amateur class accounts for more than 95%.

Figure 5.1 shows the values from Table 4.7 combined again in a conclusive and concise way. The results show that the simulator takes an average of 19.5 seconds to complete an order. The expert takes 66.6 seconds, the second shortest, and the amateur 75.1 seconds, the longest.
As a result, the simulator promises to improve performance by using a digitized milk-run system by an average of 47.1 seconds compared to the expert using the non-digitized milk-run system and 55.6 seconds compared to the amateur under the same conditions as the expert.

With these strong results in hand, we now move on to research question number 2, which answers the question of whether such a system can be implemented.

5.1.2 Research question 2

After the first research question could be answered positively and thus nothing stands in the way of a further pursuit of the project, we continue with research question number two, which asks the following question:

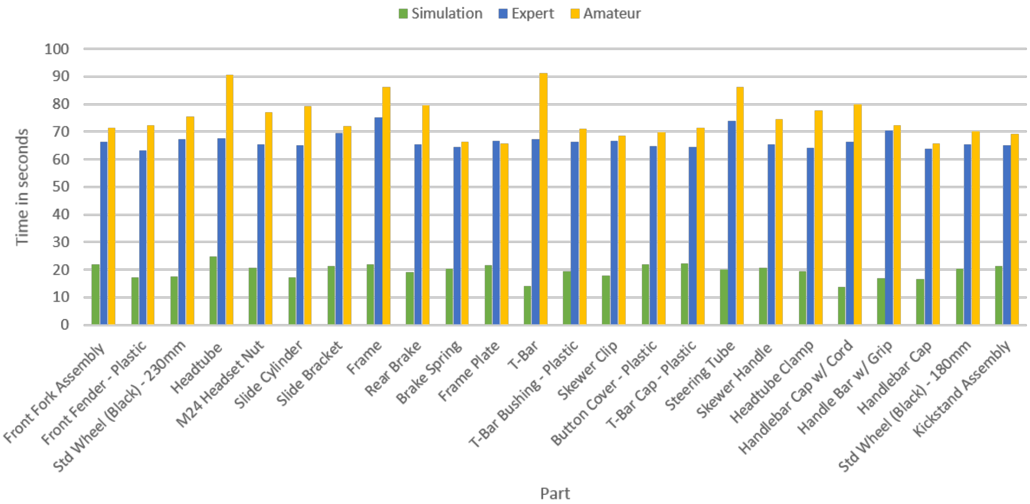

Figure 5.1: The results of the simulation and evaluation in the LEAD Factory without the digitized milk-run system are shown in this chart. On the x-axis all parts relevant for the milk-run are shown in the order of their appearance in the process from left to right. The y-axis shows the time in seconds that the respective part takes from order entry to order completion. Or to be more precise, the time it takes the logistics employee to pick up the empty box of the part at the workplace, refill it, deliver it and return to the starting point. Shown is the time the simulator claims to take when using the digitized milk-run system in green, the time an expert in the LEAD Factory needs for the non-digitized milk-run in blue and the time an amateur needs under the same conditions in yellow. As can be seen clearly, the simulator promises significantly better results. On average 47.1 seconds compared to the expert and 55.6 seconds compared to the amateur.

- Can a digitized milk run system be built in the context of the IIM LEAD Factory?

I do not want to lose many words here, but rather speak directly of the answer. Yes, it is possible to design and implement a digitized milk-run system in the LEAD Factory. Chapter 3 describes on about 35 pages how this was realized and what resulted from it.
In a nutshell, the digitized milk-run system consists of an automatic order processing system that works with RFID, a GUI that manages the order management, a pick-to-light system to pick the parts as efficiently as possible and a route calculation that suggests an efficient path to the logistics employee.

This means that two research questions have already been answered positively, which means that we can approach the last research question, which will be clarified in the following section.

5.1.3 Research question 3

As already mentioned, the first two research questions were answered positively. That is why it is now also possible to answer the third and last research question, which reads as follows:

- How precise is the prediction of the simulator compared to a real-life scenario. The time difference is to be recorded, for example in the context of a validation in the learning factory of the IIM Institute.

In order to answer this question, the digitized milk-run system was implemented and tested in the LEAD Factory, as described in detail in Chapter 3. The results were written down in Table 4.8 and are now presented in Figure 5.2 in such a way that the reader has an immediate, good overview of what arose out of this.
The results of the simulator are in the figure green, those of the expert blue and those of the amateur yellow. As can be seen, the promises of the simulator with an average time of 19.5 seconds are still below the real results for most parts. Nevertheless, it must be said that the digitized milk-run has significantly improved the results. Especially for those parts that are used at workplaces near the warehouse, as can be clearly seen in the right section of Figure 5.2, the gap between the simulation value and the actually measured value closes completely. The average time taken

to complete an order is now 31.3 seconds for an expert and 35.2 seconds for an amateur. This means that the average difference from the simulator result to the expert is only 11.8 seconds and the difference to the amateur is only 15.7 seconds. Although this is still a wide gap for most of the parts, it can certainly be reduced or completely closed with future work and improvements on the simulator. The closing gap suggests that there may be a crucial issue with the transport time used in the simulator.

5.2 Lessons learned

At the beginning of the thesis, when it was important to find a suitable technology for the simulator, the focus was quickly put on an agent-based model. This has its justification and is in my opinion the only reasonable way to solve this problem adequately, but afterwards a lot of time was wasted to get to know frameworks, to try them out and to compare them with each other. This time could have been saved, because as it soon turned out, the best idea was to design and build an agent-based model simulator by oneself. On the one hand most frameworks are oversized for the needs of this project and on the other hand you learn the most when you take things into your own hands. So that was definitely a lessons learned moment.

After building the simulator, the challenge was to implement the digitized milk-run in the LEAD Factory. Since I am a software developer at heart, I had a limited idea of hardware development and the use of electronics in an industrial context. My estimates for the effort and the necessary functions of the hardware were much higher than it was in reality in the end. Through valuable tips and pushes in the right direction from friends and supporters, I soon realized that everything was not as insanely complicated as I thought. I was very grateful for this and was able to complete the project without any major incidents. Therefore, lesson learned.

In the course of the thesis there were always moments in which great ideas came to me, but I was so naive and thought that I could certainly remember them for later. Only late in the course of the work did I get used to always carrying a block with me when I was working on the thesis in order to be able to write down ideas immediately. But not only ideas, but also wild thought constructs, which seemed very clever and fantastic in the mind, but then written down on paper, promptly looked different than expected. When writing down such constructs, one quickly

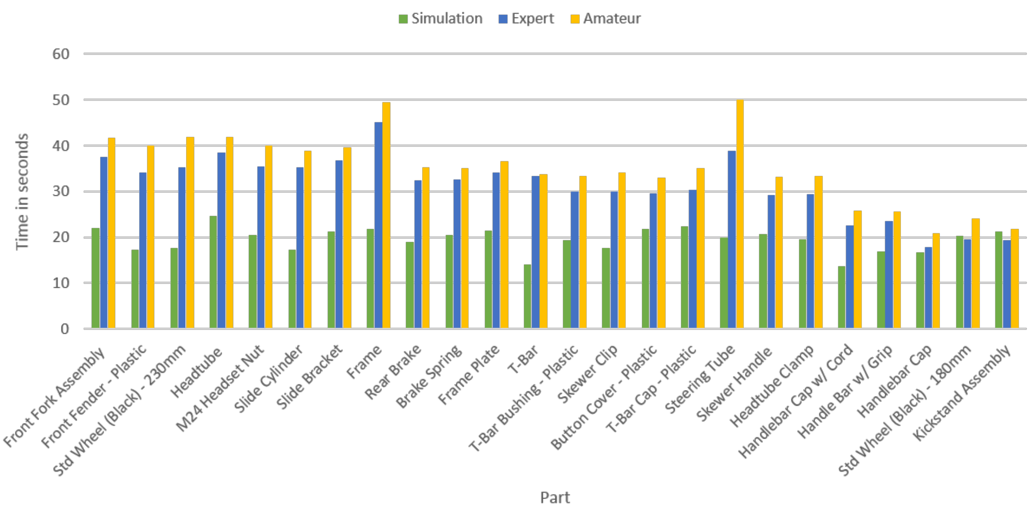

Figure 5.2: The results of the simulation and evaluation in the LEAD Factory with the help of the newly implemented digitized milk-run system are shown in this chart. On the x-axis all parts relevant for the milk-run are shown in the order of their appearance in the process from left to right. The y-axis shows the time in seconds that the respective part takes from order entry to order completion. More specifically, the time it takes the logistics employee to pick the part in the warehouse using the pick-to-light system, transport the part to the workplace via the recommended route, take the empty box and refill it, and finally return to the starting point. Shown is the time the simulator claims to take when using the digitized milk-run system in green, the time an expert in the LEAD Factory needs when using the digitized milk-run in blue and the time an amateur needs under the same conditions in yellow. As can be seen clearly, the simulator promises significantly better results. On average 11.8 seconds compared to the expert and 15.7 seconds compared to the amateur. It also shows that the smaller the distance between the workplace and the warehouse, the closer the simulation times and the actually measured times become.

becomes aware of gaps that allow everything to collapse as soon as one engages in them. A very valuable experience, which is very general, but in my opinion must nevertheless make it into this chapter.

5.3 Limitations

Since the entire digitized milk-run system is based on the fact that the orders are received via the RFID system, it is mandatory that all involved devices are connected to the same network. That means the workplaces, the RFID system with the production control, the database with the RFID data, the order management system in the warehouse and the pick-to-light system, must all be constantly connected to perform their work properly. Depending on the size of a factory, it may be necessary to reconsider this circumstance and look for another, more suitable solution.

Once an order has been received in the order management system, it is displayed on the GUI according to its position on the shelf in the LEAD Factory warehouse. Here you can already see the limitation. The layout is explicitly adapted to that of the LEAD Factory to make it as easy as possible for the logistics employee. However, this also means that the software cannot be used out of the box in another factory or with another layout, but that the GUI must first be adapted accordingly.
Finally, the first concept for the GUI is to be mentioned here, which provided that the orders were displayed in a list form, but was discarded for the simpler layout variant. However, this list form can make more sense for larger factories than a layout-based form.
It should also be noted that in the course of the GUI adaptation, a pick-to-light adaptation will probably also have to be made if a layout form is chosen.

As the process of the milk-run progresses, one cannot help but notice that the maximum number of orders was set at five. There are several reasons for this and will be explained in more detail here.
One reason is that the baseline algorithm that was implemented for the route calculation is very dumb. Because it relies heavily on chance and has no directional mechanism, it can quickly happen that even with small factory layouts with few fields, this algorithm finds a path that spans thousands of fields. The higher the number of orders or workplaces that the algorithm has to process, the greater the

risk that the route becomes too large.
Another reason for limiting the number of orders is the milk wagon itself, which is used to deliver the parts or boxes to the workplace. The wagon is relatively small and has exactly five levels on which parts can be placed. So the maximum was also chosen not to overfill this wagon.
But probably the most compelling reason for the limitation of orders is simply that the LEAD Factory is very small and primarily a learning factory, in which training participants should get to know the lean principles. In a training scenario, there is always one participant who is responsible for the milk-run. The trainee should therefore have a task that is more than delivering twenty boxes at the end of the training. The participant should be on the move all the time and should ensure supplies and, if possible, also get stressed because the maximum number of orders is too low. These are valuable experiences that are precious and desirable in a hands-on teaching style. The participants should therefore above all learn something from it and question this maximum amount.

Finally, it should be mentioned that the entire project was implemented in C#. Therefore it can only be used on computers running Microsoft Windows. For the LEAD Factory this is no problem, because all systems run on Windows anyway. However, if this milk-run system is to be used on other operating systems, a certain amount of effort is required to adapt the software.

5.4 Summary

In this chapter the importance of the results were discussed in further detail. The research questions, which were determined at the beginning of the thesis, were explained and answered one after the other. Afterwards, a short session was held about learned lessons that were important for the author during this project. Finally, the limitations of the digitized milk-run system in the context of the LEAD Factory were shown. Thus only a summary of this complete thesis remains, which will be done in Chapter 6 below.

6 Conclusion

In the course of this thesis, many aspects in general on the topic of digitization in industry and in particular on the topic of digitization of a milk-run system were addressed. The thesis can be used as a decision aid to get an overview of the scope of a digitization process. Since the topic is very versatile, this chapter will once again summarize the most important points and also give an overview of what can be improved or revised in the future to make this topic even more holistic. Of course a final completion of the work is not possible, because there is always something to accomplish. With the answer of the research questions, however, the work is finished for the author.

6.1 Review

Chapter 1 introduced and defined the three research questions that set the pace for this thesis. Since it was very important to be able to evaluate the results of the software, a suitable factory was sought in which the simulated system could be tested. This is usually not an easy task, as there is hardly a company that can respond to such a request. Fortunately it was possible to use the LEAD Factory of the IIM institute for this undertaking. That is why part of the first chapter is devoted to the idea of a learning factory as it was made available to the author for his work.

Chapter 2, which was about the literature review contained important information on the background of the work. Starting with general information about milk-run systems in the industry and their origin and use. When talking about milk-run systems, it is also necessary to know what they are and under what other terms they are known.
Afterwards it was explained what a learning factory is and for what purpose it

is used. A learning factory is hence a powerful instrument for passing on and receiving knowledge. The hands-on mentality of this learning style is very well suited, thanks to various factors, to gain an understanding of what has been learned in a short period of time.
A no less important part of this thesis is the use of the RFID system, so it is imperative that the reader be familiarized with RFID and learn about its operation and history as well as alternative applications.
In the context of the simulator, the literature review talks about the difference between a discrete-event simulation and a continuous simulation. It is important to note that the discrete-event simulation method was used in this thesis, as it is better suited for use as a milk-run simulator.
Additionally it was explained what an agent-based model is and why it is the right choice for the use in a simulator in this thesis.
Since some aspects of the software rely on chance for its calculations, it was also relentless to learn more about random processes. That's why the Monte Carlo method of finding random numbers was discussed in detail and how important it is that the numbers are actually randomly distributed.
In order not to neglect the topic of industry, which constitutes a major part of this thesis, flexible manufacturing systems were discussed. It was explained what it is and how it works. This is important because parts of the LEAD Factory can also be seen as a flexible manufacturing system and the digitization of the processes is ultimately aimed at maintaining a fully automated system at some point.
Finally, a second industry topic was addressed, namely the material flow system. The idea is that each part, product and sub-assembly should be subject to rules that say where a part is to go next, what steps are still required to complete it, etc. In other words, each component is part of a flow that brings it ever closer to completion.

In Chapter 3, which is the most comprehensive chapter in this thesis, we talked about the conception and subsequent implementation of the digitized milk-run system in the LEAD Factory. The approach taken was to ensure that the reader moves on from the general to the special. In other words, in the beginning there was a general discussion about the concept of a milk-run and what this could look like. Then on to the pick-to-light system and the strategies that can exist when picking in the setting of the learning factory. Afterwards it was important to understand what the route planning actually has to do in the context of this thesis and how it should behave. In order to measure the efficiency of the calculated route, a baseline

algorithm was considered. Finally, the simulator and its functions were presented in the concepts.
In the implementation part of this chapter, special attention was given to what was actually realized. Since it is not possible to implement such a project without a certain infrastructure and RFID is an important aspect of this infrastructure, a lot of energy was put into bringing the reader closer to the RFID system in the LEAD Factory. In addition, the milk wagon and the supermarket were also mentioned and described in the context of infrastructure. Then it was shown which system is used as an order management system and to what extent it is structured and how it forms a unit with the implemented pick-to-light system. The implementation of the route finding was also described in more detail and it was shown how the baseline algorithm looks like to compare the results. Finally much was written about the simulator, because it is a crucial part of this thesis. Functionality and all important parameters were presented to create a deep understanding of the simulator.

Chapter 4 is devoted exclusively to results. All results necessary to answer the research questions are listed in this chapter. This includes the results of the route calculation and all time measurements that are important to have a comparison value to the simulator result. This includes pick-to-light times and refilling times at the workplaces. Finally, the simulator results were presented and listed with the actually measured times in the LEAD Factory.

Finally, Chapter 5 discussed the results of Chapter 4 and answered the three research questions. In addition there was talk about learned lessons and limitations of the system, which are related to this thesis.

6.2 Future work

The future work section is a tricky topic, because in my opinion there is always something to improve and you always come up with new ideas on how to make the project or the software even more impressive. The following ideas are therefore only a few that, according to the author, make the most sense.

As already mentioned in the corresponding chapter, the order management system for this project is linked to the layout of the warehouse in the LEAD Factory. This means that you have to adapt the GUI every time you want to use the system somewhere else. To counteract this, it would make sense, as noted in the concept, to create a way to switch the GUI from layout-based to list-based. This would ensure that other factories would be able to use the software without modifications by simply switching the GUI to the list form.

The pick-to-light system can also be given a meaningful upgrade. At present, the confirmation that a part has been picked is made on the screen. In the future, however, it would make sense to either attach a confirmation button directly to the shelf or to attach a sensor that detects from which compartment a part has just been removed. This can be done, for example, by a light sensor or by a scale which records how many parts have been removed.

It can also make sense to improve route calculation by implementing even more algorithms that can suggest an alternative solution. After all, the baseline algorithm is very dumb and only intended as a negative comparison. The A* algorithm, on the other hand, is very good and always returns the same result as long as the layout doesn't change. However, there could also be other algorithms that allow a further comparison of the efficiency of route calculation.

One aspect that the digitized milk-run does not cover is the one directly at the workplace, because a lot of time is wasted putting the refilled boxes in the right place at the workplace. Since the correct place is only described with the name of the part, an amateur who does not know the parts is at a disadvantage compared to an expert. In this process step a lot of time is lost for the amateur, because he or she can't do it as fast as the expert, who knows exactly where the parts belong, despite the digitized milk-run.

Another improvement concerns the simulator. At the moment it only simulates strictly according to the rules and is not able to learn by himself. Exactly there an improvement would be possible, by making the simulator able to learn. This would allow the simulator to react automatically to changes or disturbances in the process and still deliver good results.

The last improvement is not really an improvement, but an approach to make the milk-run system more accessible. The entire project was developed in C# and can therefore only run on Microsoft Windows systems. By adapting to other systems, it would be possible to create a much larger user base. This does not rule out development as an app, as the entire software system can be made mobile without much effort.

6.3 Summary

This chapter provided an overview of the entire thesis and what was concretely implemented in the LEAD Factory. This includes the introduction of the research questions, the conception and implementation of the digitized milk-run system as well as the documentation, discussion and evaluation of the results. Finally, future improvements and modifications were discussed, which could be useful in the context of this thesis.

Bibliography

[1] S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. O'Hare. Agent based modelling and simulation tools: A review of the state-of-art software. In *Computer Science Review. Volume 24*, pages 13–33, May 2017.

[2] D. Arnold and K. Furmans. *Materialfluss in Logistiksystemen*, volume 6. Springer, 2005.

[3] D. Arnold, K. Furmans, H. Isermann, A. Kuhn, and H. Tempelmeier. *Handbuch logistik*, volume 3. Springer, 2008.

[4] B. W. Arthur, S. Durlauf, and D. Lane. The economy as an evolving complex system ii. In *Proceedings*, volume 27, 01 1997.

[5] W. B. Arthur. Out-of-equilibrium economics and agent-based modeling. In *Handbook of Computational Economics*, volume 2, 2006.

[6] K.-H. G. Bae, L. A. Evans, and A. Summers. Lean design and analysis of a milk-run delivery system: case study. In *Proceedings of the 2016 Winter Simulation Conference (WSC '16)*, pages 2855–2866, Piscataway, NJ, USA, 2016. IEEE Press.

[7] R. Bagni, R. Berchi, and P. Cariello. A comparison of simulation models applied to epidemics. In *J Artif Soc Social Simul*, volume 5, June 2002.

[8] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences*, volume 99, pages 7280–7287, 2002.

[9] V. Bos and J. J. T. Kleijn. Structured operational semantics of chi. *Computing*, 99:01, 1999.

[10] Y. A. Bozer and D. D. Ciemnoczolowski. Performance evaluation of small-batch container delivery systems used in lean manufacturing – part 1: system stability and distribution of container starts. In *International Journal of Production Research*, volume 51, pages 555–567, 2013.

[11] J. F. Broenink. Modelling, simulation and analysis with 20-sim. *Journal A*, 38:22–25, 1997.

[12] K. M. Carley. Biowar: Scalable agent-based model of bioattacks. In *IEEE Trans Syst Man Cybernet*, volume 36, page 252–265, Chicago, IL, USA, 2006.

[13] C. S. Chong, A. I. Sivakumar, and R. Gay. Dynamic scheduling i: simulation-based scheduling for dynamic discrete manufacturing. In *Proceedings of the 35th conference on Winter simulation: driving innovation*, pages 1465–1473. Winter Simulation Conference, 2003.

[14] A. Crespo-Márquez, R. R. Usano, and R. D. Aznar. Continuous and discrete simulation in a production planning system. a comparative study. In *Proceedings of International System Dynamics Conference, Cancun, Mexico, The System Dynamics Society*, volume 58, 1993.

[15] A. Deshpande, A. Gollu, and L. Semenzato. The shift programming language for dynamic networks of hybrid automata. *IEEE Transactions on automatic control*, 43:584–587, 1998.

[16] M. Dewitz, S. Galka, and W. A. Günthner. Drive-thru loading concept for in–plant milk runs. In *Proceedings of the XX International Conference MHCL'12*, volume 20, 2012.

[17] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1:269–271, 1959.

[18] H. Elmqvist. Dymola - dynamic modeling language - user's manual. *Dynasim AB, Lund, Sweden*, 1994.

[19] A. K. Engel. Prinzipien der wahrnehmung: das visuelle system. *Kopf-Arbeit-Funktionen und kognitive Leistungen*, pages 181–207, 1996.

[20] W. J. Eradus and M. B. Jansen. Animal identification and monitoring. In *Computers and Electronics in Agriculture*, volume 24, pages 91–98, 1999.

[21] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99:10143–10162, 1994.

[22] K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and near-Field Communication.* Wiley, 2010.

[23] E. R. Firmansyah, S. U. Masruroh, and F. Fahrianto. Comparative analysis of a* and basic theta* algorithm in android-based pathfinding games. In *Information and Communication Technology for The Muslim World (ICT4M), 2016 6th International Conference on*, pages 275–280. IEEE, 2016.

[24] G. S. Fishman. *Principles of discrete event simulation.* John Wiley and Sons, New York, NY, 1978.

[25] V. A. Folcik, G. C. An, and C. G. Orosz. The basic immune simulator: An agent-based model to study the interactions between innate and adaptive immunity. In *Theoret Biol Med Model*, volume 4, 2007.

[26] A. Freitas, A. Maio, P. Maia, N. Gomes, A. Nogueira, J. Fernandes, M. Carvalho, A. Alves, A. Costa, P. Afonso, P. Silva, D. Barbosa, and S. Machado. Savings in internal logistics using a rfid-based software system in a lean context. In *Proceedings of International Conference on Computers and Industrial Engineering*, CIE, 2017.

[27] P. Fritzson and P. Bunus. Modelica-a general object-oriented language for continuous and discrete-event system modeling and simulation. In *Simulation Symposium Proceedings. 35th Annual*, pages 365–380. IEEE, 2002.

[28] P. Gorecki and P. R. Pautsch. *Lean Management.* Carl Hanser Verlag GmbH Co KG, 2015.

[29] S. Gotthardt. *Maschinenberechtigungsprüfung in öffentlich zugänglichen Werkstätten.* Graz University of Technology, 2018.

[30] A. Greasley. Using system dynamics in a discrete-event simulation study of a manufacturing plant. *International Journal of Operations & Production Management*, 25:534–548, 2005.

[31] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis. A standard protocol for describing individual-based and agent-based models. In *Ecological Modelling*, volume 198, pages 115–126, September 2006.

[32] J. Hammersley. *Monte carlo methods*. Springer Science & Business Media, 2013.

[33] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4:100–107, 1968.

[34] R. R. Hill, R. G. Carl, and L. E. Champagne. Using agent simulation models to examine and investigate search theory against a historical case study. In *J Simul*, volume 1, pages 29–38, 2006.

[35] G. Jayendra, S. Kumarawadu, and L. Meegahapola. Rfid-based anti-theft auto security system with an immobilizer. In *International Conference on Industrial and Information Systems*, volume 2, pages 441–446, 2007.

[36] D. Johnson and B. Thorn. Capacity and resource planning for an engineering technology department. In *American Society for Engineering Education*. American Society for Engineering Education, 2009.

[37] K. Kamarulazizi and W. Ismail. Electronic toll collection system using passive rfid technology. In *Journal of Theoretical and Applied Information Technology*, volume 22, pages 70–76, 2010.

[38] C. Keuntje, F. Hormes, and J. Fottner. Considering technical details in the planning of tugger train systems. In *HP3C*, 2018.

[39] H. Kilic, M. Durmusoglu, and M. Baskak. Classification and modeling for in-plant milk-run distribution systems. In *The International Journal of Advanced Manufacturing Technology*, volume 62, page 1135–1146. Springer, October 2012.

[40] F. Klügl and A. L. C. Bazzan. Agent-based modeling and simulation. In *AI magazine*, volume 33, 2012.

[41] T. A. Kohler, G. J. Gumerman, and R. G. Reynolds. Simulating ancient societies. In *Scient Amer*, volume 293, pages 77–84, 2005.

[42] P. Korytkowski and R. Karkoszka. Simulation-based efficiency analysis of an in-plant milk-run operator under disturbances. *The International Journal of Advanced Manufacturing Technology*, 82:827–837, 2016.

[43] P. Kostal and K. Velisek. Flexible manufacturing system. *World Academy of Science, Engineering and Technology*, 53:825–829, 2010.

[44] D. Kreimeier, F. Morlock, C. Prinz, B. Krückhans, D. C. Bakir, and H. Meier. Holistic learning factories – a concept to train lean management, resource efficiency as well as management and organization improvement skills. In *Procedia CIRP*, volume 17, pages 184–188, 2014.

[45] D. P. Kroese and R. Y. Rubinstein. Monte carlo methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4:48–58, 2012.

[46] A. M. Law and W. D. Kelton. *Simulation modeling and analysis*, volume 2. McGraw-Hill New York, 1991.

[47] J. Lerner, D. Wagner, and K. Zweig. *Algorithmics of large and complex networks: design, analysis, and simulation*, volume 5515. Springer, 2009.

[48] K. Letrud. A rebuttal of ntl institute's learning pyramid. In *Education*, volume 133, pages 117–124, 2012.

[49] K. Letrud and S. Hernes. The diffusion of the learning pyramid myths in academia: an exploratory study. In *Journal of Curriculum Studies*, volume 48, pages 291–302, 2016.

[50] B.-S. Lou. Applying principles from "scientific foundations for future physicians" to teaching chemistry in the department of medicine at chang gung university. In *Kaohsiung Journal of Medical Sciences*, volume 28, pages 36–40, 2012.

[51] C. M. Macal. Emergent structures from trust relationships in supply chains. In *Proceedings of Agent 2004: Conference on Social Dynamics: Interaction, Reflexivity and Emergence*, pages 743–760, Chicago, IL, USA, 2004.

[52] C. M. Macal and M. J. North. Tutorial on agent-based modelling and simulation. In *Journal of Simulation*, volume 4, pages 151–162, 2010.

[53] E. E. L. Mitchell and J. S. Gauthier. Advanced continuous simulation language (acsl). *Simulation*, 26:72–78, 1976.

[54] J. Moffat, J. Smith, and S. Witty. Emergent behaviour: theory and experimentation using the mana model. In *J Appl Math Decis Sci*, volume 10, pages 1–13, 2006.

[55] B. Mufida. Implementation method a* (a-star) for npc enemies in 3d game vocabulary learning arabic. *IEESE International Journal of Science and Technology*, 5, 2016.

[56] V. Mácsay and T. Bányai. Toyota production system in milkrun based in-plant supply. In *Journal of Production Engineering*, volume 20, pages 141–146, 2017.

[57] M. J. North, C. M. Macal, J. S. Aubin, P. Thimmapuram, M. Bragen, J. Hahn, J. Karr, N. Brigham, M. E. Lacy, and D. Hampton. Multi-scale agent-based consumer market modeling. In *Complexity*, volume 15, pages 37–47, 2009.

[58] H. V. D. Parunak, R. Savit, and R. Riolo. Agent-based modeling vs. equation-based modeling: A case study and users' guide. In *Sichman J.S., Conte R., Gilbert N. (eds) Multi-Agent Systems and Agent-Based Simulation. MABS 1998. Lecture Notes in Computer Science*, volume 1534, pages 10–25. Springer, 1998.

[59] C. Roser. *"Faster, Better, Cheaper" in the History of Manufacturing: From the Stone Age to Lean Manufacturing and Beyond*. Productivity Press, 2016.

[60] H. K. Shivanand. *Flexible manufacturing system*. New Age International, 2006.

[61] B. Siregar, D. Gunawan, U. Andayani, E. S. Lubis, and F. Fahmi. Food delivery system with the utilization of vehicle using geographical information system (gis) and a star algorithm. In *Journal of Physics: Conference Series*, volume 801, pages 12–38. IOP Publishing, 2017.

[62] D. P. Subramony, M. Molenda, A. K. Betrus, and W. Thalheimer. Previous attempts to debunk the mythical retention chart and corrupted dale's cone. In *Educational Technology*, volume 54, pages 17–21. Educational Technology Publications, Inc., 2014.

[63] A. A. Tako and S. Robinson. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decision support systems*, 52:802–815, 2012.

[64] A. S. Tanenbaum, M. R. Rieback, and B. Crispo. The evolution of rfid security. In *IEEE Pervasive Computing*, volume 5, pages 62–69, 2006.

[65] A. Varga. Discrete event simulation system. In *Proc. of the European Simulation Multiconference (ESM'2001)*, 2001.

[66] H. Wang, J. Zhou, G. Zheng, and Y. Liang. Has: Hierarchical a-star algorithm for big map navigation in special areas. In *2014 5th international conference on digital home (icdh)*, pages 222–225. IEEE, 2014.

[67] E. J. Wood. Problem-based learning: Exploiting knowledge of how people learn to promote effective learning. In *Bioscience Education*, volume 3, pages 1–12, 2004.

[68] W. Zeng and R. L. Church. Finding shortest paths on real road networks: the case for a. *International journal of geographical information science*, 23:531–543, 2009.

[69] R. Y. Zhong, G. Q. Huang, S. Lan, Q. Dai, C. Xu, and T. Zhang. A big data approach for logistics trajectory discovery from rfid-enabled production data. In *International Journal of Production Economics*, volume 165, pages 260–272, 2015.

[70] S. Zhou, W. Ling, and Z. Peng. An rfid-based remote monitoring system for enterprise internal production management. In *The International Journal of Advanced Manufacturing Technology*, volume 33, pages 837–844, 2007.

Appendix