

**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**  
**STATUS CONFIRMATION FOR UNDERGRADUATE PROJECT REPORT**  
**IMPLEMENTATION OF A MANUFACTURING EXECUTION SYSTEM**  
**FOR A LEARNING FACTORY**  
**ACADEMIC SESSION: 2022/2023**

I, **MATTIS TOM RITTER**, agree to allow this Undergraduate Project Report to be kept at the Library under the following terms:

1. This Undergraduate Project Report is the property of the Universiti Tun Hussein Onn Malaysia.
2. The library has the right to make copies for educational purposes only.
3. The library is allowed to make copies of this report for educational exchange between higher educational institutions.
4. \*\* Please Mark (✓)

☐

CONFIDENTIAL

(Contains information of high security or of great importance to Malaysia as STIPULATED under the OFFICIAL SECRET ACT 1972)

☐

RESTRICTED

(Contains restricted information as determined by the Organization/institution where research was conducted)

☒

FREE ACCESS

Approved by

Mattis Tom Ritter

Prof. Madya Ir. Dr. Low Cheng Yee

Permanent Address:

19, Jalan Kencana 1A/26,

83300 Pura Kencana

BATU PAHAT,

JOHOR

Date: 04.07.2023

Date: \_\_\_\_\_

NOTE:

\*\* If this Undergraduate Project Report is classified as CONFIDENTIAL or RESTRICTED, please attach the letter from the relevant authority/organization stating reasons and duration for such classifications.

IMPLEMENTATION OF A MANUFACTURING EXECUTION SYSTEM FOR A  
LEARNING FACTORY

MATTIS TOM RITTER

A thesis submitted in partial  
fulfilment of the requirement for the award of the  
Bachelor's Degree of Mechanical Engineering with Honours

Faculty of Mechanical and Manufacturing Engineering  
Universiti Tun Hussein Onn Malaysia

JULY 2023

I hereby declare that the work in this project report is my own except for quotations  
and summaries which have been duly acknowledged

Student : .....  
Mattis Tom Ritter

Date : .....

Supervisor : .....  
Prof. Madya Ir. Dr. Low Cheng Yee

## **ACKNOWLEDGEMENT**

I would like to use this space to give a special acknowledgment for Dr. Low Cheng Yee, who always supported me in every matter.

I would like to also thank all the persons working in the UTHM-Fraunhofer IEM Innovation Lab. It was great working together with very helping people. It was a great experience being part of the project learning factory. Especially one person I like to thank, Moritz Höhnel.

## **ABSTRACT**

In times of the fourth industrial revolution data is the most valuable resource of a company. Manufacturing Execution Systems use digitalised production site information to improve the production processes. This report presents the implementation of such a system in order to provide a demonstrator for small and medium-sized enterprises. The focus is on the data gathering, storage and visualization. The project is conducted by using Arcstone software and shows the data flow from shopfloor level to the Manufacturing Execution System in detail. As the concept is not established in local companies, this report serves as a knowledge transfer platform for their next step of industrial revolution. It will show them the benefits of a more detailed observed production with a learning factory as example. Furthermore, companies can use the step-by-step explanations as a guideline to implement the system in their own production.

## **ABSTRAK**

Pada masa revolusi industri keempat, data adalah sumber yang paling berharga bagi sesebuah syarikat. Sistem Perlaksanaan Pembuatan menggunakan maklumat tapak pengeluaran digital untuk menambah baik proses pengeluaran. Laporan ini membentangkan pelaksanaan sistem sedemikian untuk menyediakan penunjuk perasaan untuk perusahaan kecil dan sederhana. Tumpuan adalah pada pengumpulan data, penyimpanan dan visualisasi. Projek ini dijalankan dengan menggunakan perisian Arcstone dan menunjukkan aliran data dari peringkat shopfloor ke Sistem Perlaksanaan Pembuatan secara terperinci. Memandangkan konsep itu tidak diwujudkan dalam syarikat tempatan, laporan ini berfungsi sebagai platform pemindahan pengetahuan untuk langkah seterusnya revolusi perindustrian mereka. Ia akan menunjukkan kepada mereka faedah pengeluaran yang diperhatikan dengan lebih terperinci dengan kilang pembelajaran sebagai contoh. Tambahan pula, syarikat boleh menggunakan penjelasan langkah demi langkah sebagai garis panduan untuk melaksanakan sistem dalam pengeluaran mereka sendiri.

## CONTENTS

TITLE	i
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
ABSTRAK	v
LIST OF TABLES	vii
LIST OF FIGURES	ix
LIST OF SYMBOLS AND ABBREVIATIONS	xi
LIST OF APPENDICES	xii
CHAPTER 1 INTRODUCTION	1
1.1 Background Study	1
1.2 Problem Statement	2
1.3 Real World Example	3
1.4 Objectives	4
1.4.1 General Description of the Product	4
1.5 Scope of Study	6
1.5.1 Functional Requirements	7
1.5.2 Non-Functional Requirements	9
CHAPTER 2 LITERATURE REVIEW	10
2.1 MES	11
2.1.1 Functionality of a MES	11
2.1.2 MES in the Context of Industry 4.0	14
2.1.3 Arcstone's MES solutions	15

2.2 Artificial Intelligence	17
2.2.1 Neuronal Network in General	17
2.2.2 Convolutional Neuronal Network	20
2.2.3 Recurrent Neuronal Network	22
2.2.4 Summary Neuronal Networks	22
CHAPTER 3 METHODOLOGY	24
3.1 Flow Chart	24
3.2 Previous State of the Learning Factory	27
CHAPTER 4 RESULTS AND DISCUSSION	29
4.1 Dataflow from Automation Level to MES	29
4.1.1 Architecture of the Connection between Automation Software and Server	30
4.1.2 Pull Data from the Server to the MES Database	32
4.1.3 Access Data Inside the MES	36
4.2 Creating the Dashboard	39
4.3 How the AI-Lernfabrik Works	41
4.3.1 Applying the Neuronal Networks	42
CHAPTER 5 CONCUSSION AND RECOMMENDATIONS	49
5.1 MES	49
5.2 AI Lernfabrik	50
REFERENCE	53
APPENDIX	55



**LIST OF TABLES**

Table 1.1: MES Software	7
Table 1.2: Interfaces	8
Table 1.3: LI Learning Factory	8
Table 1.4: Non-Functional Requirements	9
Table 4.1: State of each station and the chutes	40
Table 4.2: Possible active state, code and target of each station	44
Table 4.3: Data	45

## LIST OF FIGURES

Figure 2.1: Architecture of an enterprise with MES	13
Figure 2.2: Arcstone's MES solutions [6]	16
Figure 2.3: Arcstone in the context of the ISA 95 framework	16
Figure 2.4: Model of network [10]	18
Figure 2.5: Mathematical Neuron [10]	18
Figure 2.6: Pattern of number 2 [10]	19
Figure 2.7: CNN working principle [10]	21
Figure 2.8: Recurrent Neuronal Network [11]	22
Figure 3.1: MES Flowchart	25
Figure 3.2: AI-Learning Factory Flowchart	26
Figure 3.3: Fischertechnik Smart Factory [13]	27
Figure 3.4: Schematic Plan of the Learning Factory [15]	28
Figure 4.1: Dataflow from the automation software to the MES	29
Figure 4.2: Architecture examples of the connection	31
Figure 4.3: Used architecture for the learning factory	32
Figure 4.4: General connection setup of arc.quire profile	33
Figure 4.5: Source setup of arc.quire profile	34
Figure 4.6: Destination setup of arc.quire profile	35
Figure 4.7: Microsoft SQL Server Management Studio	36
Figure 4.8: Add new data source to a dashboard	37
Figure 4.9: Dashboard data source wizard	37
Figure 4.10: Bind data to grid	38
Figure 4.11: Calculated field	40
Figure 4.12: Dashboard	41
Figure 4.13: Sample message [15]	43
Figure 4.14: Sorted messages [15]	44

Figure 4.15: Input array (left) and output array (right) [15]	47
Figure 4.16: Multidimensional input array [15]	48
Figure 5.1: Controller overview [16]	50

## LIST OF SYMBOLS AND ABBREVIATIONS

MES	-	Manufacturing Execution System
PLC	-	Programmable Logic Controller
AI	-	Artificial Intelligence
PC	-	Personal Computer
VDI	-	Verein Deutsche Ingenieure, <i>engl.</i> Association of German Engineers
ERP	-	Enterprise Resource Planning
ISA	-	International Society of Automation
CNN	-	Convolutional Neuronal Network
RNN	-	Recurrent Neuronal Network
OPCUA	-	Open Platform Communications United Architecture
URL	-	Uniform Resource Locator
SQL	-	Structured Query Language
JSON	-	JavaScript Object Notation
NFC	-	Near Field Communication
IoT	-	Internet of Things
MQTT	-	Message Queuing Telemetry Transport

**LIST OF APPENDICES**

<b>APPENDIX</b>	<b>TITLE</b>	<b>PAGE</b>
A	AI-CAMPUS COURSE: LERNFABRIK	55
B	SOLUTION EXERCISES STEP BY STEP	152

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 Background Study**

“The fourth industrial revolution is accelerating more and more. The foundations for a completely connected ‘smart factory’ are set. Digitalization, robotics, artificial intelligence [...]. Today, these developments are changing industrial production fundamentally. [...] Physical and digital processes are becoming increasingly intertwined. This also includes networking with customers: Their needs and demands are and remain the guiding principle of all of the company's work.” The Mercedes-Benz Group [1], one of the world’s leading car manufacturers, stating that the current industrialization is having an increasing effect. As three main pillars they name the digitalization, advancing robot technology and artificial intelligence (AI). All these pillars get more and more integrated into production processes, as they have positive effect on production costs.

One goal of Industry 4.0 is to connect workers and machines. Digitalization is providing a platform to collect production data from machines as well as humans. The collected information can be analysed and translated into languages machines and humans understand. Advancing robot technologies are helping to support and take over production steps. Especially those that humans can not handle, as example working in a hot environment. The third pillar, Artificial intelligence, is advancing the processing and analysis of production data.

The goal of this project is to work on a factory simulation that is capable of all the previous requirements for a future factory. The factory is completely operated by robots, to establish one pillar. For this project a Manufacturing Execution System

(MES) should be established. It is used to collect the sensor data, save it to a cloud and then send it to inform the workers and machines that are part of the production. This means that it takes the pillar of Digitalization. The third pillar, the artificial intelligence, would be used to analyse the production data.

Apart from optimizing the production itself, the Mercedes-Benz Group outlines that the need for providing information to the customers also becomes more important. The manufacturing execution system has matching options for this purpose and shall be used to provide production information.

## **1.2 Problem Statement**

While economy leading companies have already implemented important steps that are considered to be part of the fourth industrial industrialization, smaller local companies often lack those realization steps. They are still struggling to get the data as in most cases they are not digitised. But even if the companies have sensorized workstations that collect vast volumes of data, they are not connected to each other. This results in standalone islands, that are not contributing much to the improvement of the overall manufacturing process as the data is not distributed to a manufacturing execution system (MES). In a MES the data has a tremendous operational value. The factory simulation is established to show undigitized companies the advantages of a connected factory as well as giving them input for actual implementation.

This covers one of the pillars of industry 4.0, but there is still more to do for local companies to get most of the data. This is where artificial intelligence comes into fruition. New ways of data processing and analysis, as well as pattern recognition are made possible by neuronal networks, which leads to further improvement and efficiency gains. But the current workforce lacks these skills and has insufficient knowledge on the topic. Furthermore, a traditional mentality towards methodologies and operations, has translated into slow-moving transitions to new technologies. To get rid of these problems they need to be trained with courses that provide an easy hand-on to the field AI.

The learning factory is a convenient way to demonstrate that the path to manufacturing excellence is easy to adopt. It is common that managers are lacking the right mindset to start the process of making their factories ready for industry 4.0

and therefore the learning factory can be used to show that it is achievable for everyone.

### **1.3 Real World Example**

To improve the understanding of the simulated production process, a real-world factory is taken as example. The example is about a rim manufacturer for cars. In their factory they have a warehouse, an oven, a milling machine and after production workpieces get stored depending on colour. The rims can be ordered in colours red, blue and white. In the beginning an order picker collects raw material, puts it into an oven to temper the material. After the hardening is finished, the workpiece needs to be carried to the milling machine, where it is engraved. After that, a person needs to sort it by colour into the right chute for the final storage.

The factory simulation shows how to automate all these work steps. For implementation at first robots are introduced. All the work steps that were executed by a human can also be handled by a robot. That means that one robot can fetch the raw material, a further robot can deliver the workpieces to the factoring sites. Conveyor belts can transport the workpieces between the factoring steps and finally to the chutes. Utilizing a colour detecting sensor, the sorting of workpieces can be executed automatically. After implementing the robots just one worker who overviews all production steps is necessary. As a reduction in total number of laborers is possible, a result is lower production costs.

To make it easier for the worker overviewing the production site production will be digitalised. This includes that data is gathered, analysed and visualized. In a final step, the pillar of artificial intelligence is deployed. An intelligent software is then in charge of supervising the production. The AI can also work with the provided production site data, by applying advanced analysis methods. If all is working perfectly together, there is just a production site manager of the rim producing company necessary to overview the progress of the factory, as execution, monitoring and controlling is done by software tools. Looking at more advantages, the aim is to improve factories process performance. The primary goal of the fourth industrial revolution for local companies should be to reduce failures during production and with that saving money. But with knowing how to establish a more digitalized



production system, companies can also use the other benefits that are coming along. As examples one can take the improved traceability of the workpieces and information chain for the customers.

## **1.4 Objectives**

This chapter provides the specification of the project. Definitions will be given on what is needed to be done to complete the projects. It will be described what the MES and Lernfabrik objectives are.

For the MES project the final goal is to build a platform that enables the UTHM to transfer knowledge. Local companies should be able to understand the concept and the benefits of a Manufacturing Execution System. In detail this means to implement steps from data collection until the creation of a dashboard that gives an overview over the current production steps for the factory simulation. These implementations can then be presented and explained closer with an user guideline.

For the AI-Campus Lernfabrik project the objective is to transfer knowledge for the question: How to setup a factory for the usage of artificial intelligence? To answer the question, documents for a workshop should be created which explain all the steps from data preparation to applying a neuronal network. The workshop will be performed by the University for local companies to help them understand the implementation and the benefits. The workshop must be based on the AI Lernfabrik and includes general explanations as well as follow-along tasks and their solution. Furthermore, an analysis of the simulation factory's hardware should be performed to understand if and how it is possible to use it.

### **1.4.1 General Description of the Product**

The following parts will explain more detail what the finished products should look like. Therefore, a summary of the functions, a perspectives analyzation and a user characteristics analyzation will be made. These chapters are necessary to determine how the results looks like. They are also a guideline during development.

#### **1.4.1.1 Product Functions**

For the first part of the project a MES software is used. It is used to digitalize manufacturing processes. The process data should be put together in a dashboard. This dashboard should be usable for different persons. The first group of persons are the factory workers. The second group of persons are those that oversee the production. Both need to know which production step is running and which products are completed.

For the AI-Campus learning factory a set of explanations will be created. This will include a user manual and a workshop in which the functions will be discussed. Furthermore, it will be evaluated what steps need to be done to have the learning factory ready to implement to the software of the Lernfabrik.

#### **1.4.1.2 Product Perspective**

The factory is shown on an exhibition table. The results of the implementation of the MES Software can be accessed via a PC and a screen. The dashboard can be accessed with any PC that is connected to the UTHM network. It is also required to have Arcstone credentials to open the dashboard. For show purposes it is advised to locate the screen right next to the factory, that the prospects are able to compare what they can see happening at the factory and how the dashboard is visualizing it.

The AI-Campus factory learning material will be given digitally. It includes workshop materials. These should consist of two documents. A deeper explanation of every step and software that is used as well as a step-by-step instruction to solve example tasks.

### 1.4.1.3 User Characteristics

The expected users divide into two groups: Workers and Managers.

The workers have expertise for all the steps of the factory. He also expertise will be to use the PLC software in a way that he knows how to start the execution of the factory. He probably has no higher education and won't be able to implement any changes to the PLC software, MES Software nor the AI code. It will also be hard for him to read complex graphs. This means that data shown on the dashboard must be simple.

The manager will have higher education. He has experiences with complex software. He needs to overview that the MES or AI software is running correctly.

## 1.5 Scope of Study

The following chapter is describing the exact tasks that need to be performed. The tasks are collected in tables and are split for each subject. The following list shows abbreviations used specifically for the tables of this chapter:

- ID - Is the identification number of a specification.
- Pr. - Is the priority of a specification. It specifies whether a goal has to be achieved or not.
  - A: Has to be fulfilled (Verification needed).
  - B: Should be fulfilled
  - C: Can be fulfilled

### 1.5.1 Functional Requirements

This chapter gives the functional requirements in table format.

#### 1.5.1.1 MES Software

Table 1.1: MES Software

ID	Name	Description	Verification	Pr.
A.01	Data collection	With arc.require data of the factory must be demanded	Software verifies connection	A
A.02	Database	A database table needs to be created, that is capable to store all important data.	Load data into database	A
A.03	Database	Data from the factory must be stored into the database.	Request list of database	A
A.04	Dashboard Pre-processing	Data has to be filtered, sorted and allocated to a meaning	Understandable data	A
A.05	Dashboard	Data needs to be put into a dashboard. Important information about the manufactory needs to be shown.	Reviewing dashboard on screen	A
A.06	Dashboard	Pictures of the workstations can be added		C

### 1.5.1.2 MES External Interfaces

Table 1.2: Interfaces

ID	Name	Description	Verification	Pr.
<b>B.01</b>	Screen	To access the MES software a display is required.	Plug in display	A
<b>B.02</b>	Connection	All systems need to connect to the UTHM network	Confirmation of established connection	A
<b>B.03</b>	Connection	The factory data needs to be transmitted to the UTHM server	Access data	A
<b>B.04</b>	Connection	The arc.quire software needs to connect with the UTHM server.	Data can be loaded	A

### 1.5.1.3 KI Learning Factory

Table 1.3: LI Learning Factory

ID	Name	Description	Verification	Pr.
<b>C.01</b>	Information gathering	Course of AI learning factory should be attended		B
<b>C.02</b>	Doing tasks	Task during the course should be done, within the possibilities of equipment.	-	B
<b>C.03</b>	Manual	Creating a manual for the use of the AI factory software.	Report finished	A
<b>C.04</b>	Workshop	Prepare documents in which interactively the information is provided	Finished workshop	A
<b>C.05</b>	Connection	Understand if a connection to the factory hardware is possible	-	C
<b>C.06</b>	Data	AI software working principle should be understood and put into words	-	B

### 1.5.2 Non-Functional Requirements

Table 1.4: Non-Functional Requirements

ID	Name	Description	Verification	Pr.
D.01	Software	Usage of Arcstone arc.ops	All PCs need to be controlled to always use the latest version of the software	A
D.02	Software	Usage of Arcstone arc.quire		A
D.03	Software	Usage of Microsoft SQL database manager		A
D.04	Software	Anaconda for python for the AI factory		A
D.05	Server	UTHM server is required to be used	Connection to the server established	A

## **CHAPTER 2**

### **LITERATURE REVIEW**

Keeping pace with digitalization and ever-shorter innovation cycles is a major challenge for companies. Therefore, ongoing networking, knowledge and technology transfer, as well as constant competence development, are becoming increasingly important to remain competitive, especially for small and medium-sized enterprises (SMEs) with limited resources. As the objective of this project is transferring knowledge, a deeper look in the methods and benefits of the state of Baden-Württemberg [2] is taken firstly.

It is pointed out, that the exchange of knowledge and technologies between research institutions, companies, industries and sectors, as well as public administration, represents a great added value for innovative capacity. In particular, where experts from different disciplines meet, disruptive ideas can emerge and horizons can be enlarged.

Technology and knowledge transfer can be used and promoted in a variety of ways. Thanks to the digital transformation, new and more efficient ways of exchanging ideas are also emerging. Overall, a large number of different virtual and physical transfer offerings have developed that share knowledge between a wide variety of players: from tutorials and trainings, continuing education, cooperation exchanges, to science-industry working groups, crowdsourcing offerings, regional cluster initiatives, and topic- and technology-related hubs and labs, to events and trade fairs. The state of Baden-Württemberg considers technology and knowledge transfer to be a key instrument that SMEs can use to optimize processes both within the company and beyond, and to strengthen their competitiveness through new products, services and business models. In this context, the state also provides a wide range of transfer

offers, in which up-to-date knowledge on important technologies, such as artificial intelligence, is imparted for diverse sectors, from industrial production to trade and commerce. In principle, transfer will continue to become increasingly digitalized in the future, so that virtual communication channels, platforms and formats will come more into focus. The digitization of transfer in particular will help to develop new transfer offerings beyond what has been known to date and facilitate access to important expert knowledge in a wide range of industry and technology fields, including for new user groups.

To start with knowledge transfer the theory of the project's topics needs to be explained. This will be covered by the following chapter. The first part will be about Manufacturing Execution Systems. The second part gives the basics for the Artificial Intelligence Lernfabrik.

## **2.1 MES**

MES technology has come a long way since its inception in the 1980s and is now a vital component in the success of manufacturing companies around the world. The following chapter will cover the definition of a MES, its functionalities and how it is embedded into an enterprise. After that the role of MES in the context of Industry 4.0 will be evaluated. Lastly, the used MES for the project is reviewed.

### **2.1.1 Functionality of a MES**

MES is designed to improve operational efficiency and increase overall productivity by giving manufacturers insight into their production processes and providing them with tools to help optimize workflows. This is achieved by monitoring all aspects of the production process in real time. That is why the production process has to be fully digitized before a MES can be applied. In the best case all sensor data and actuator states are available permanently, otherwise additional data has to be provided manually.

Kletti [3] describes the functionality of a nowadays MES by the combination of three functional scopes. Those areas have to work hand in hand to create a high-performance MES and enable a well-timed and effective manufacturing control.



First there is the function group production, where data acquisition plays the most important role. For production, data order and person related times and quantities are recorded, but also machine data is collected to manage machines and other operational resources. Moreover, operational data such as order and personnel timing and amounts are acquired. Other possible applications of a MES in the connection with production are tool and resource management, where tools and other auxiliary materials are managed, material and production logistics, where information about currently circulating material is provided, energy management and many more.

When it comes to human resources a MES should be able to handle time recording, time management, personnel resource planning and wage calculation. Furthermore, short-term manpower planning and escalation management can also be use-cases for a MES.

For the third field quality assurance different mechanisms like production inspection, complaints management, testing and measurement data acquisition can be implemented into a MES.

Lots of associations have developed standards to describe the requirements of a MES. As defined by the VDI (Verein Deutsche Ingenieure, *engl.* Association of German Engineers) in the guideline VDI 5600 [4] the tasks of a MES are the following:

- Detailed planning and detailed scheduling control
- Operating resources management
- Material management
- Personnel management
- Data acquisition and processing
- Interface management
- Performance analysis
- Quality management
- Information management
- Order management
- Energy management

Furthermore, the context of the MES in the complete enterprise is described in these standards, where the attempts in the VDI 5600 and the ISA 95 [5] are similar with three levels stacked on each other.

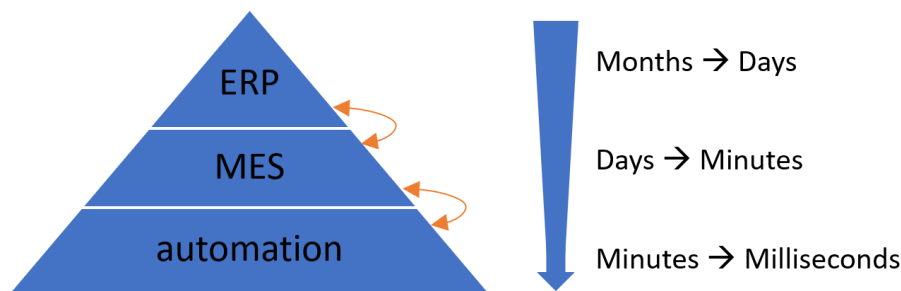


Figure 2.1: Architecture of an enterprise with MES

As shown in figure 2.1 the MES is located in between the enterprise resource planning (ERP) and the automation on shopfloor level. Furthermore, the time horizons of the different stages get significantly shorter from top to bottom. As in the ERP long- or mid-term decisions are made, the time period of interest varies from months to days, whereas for the MES single days to minutes matter. For the shopfloor level, where the actual production happens, it can even depend on milliseconds.

Since the MES has interfaces with lots of other systems, it acts like a hub for the data. The ERP provides the base data such as orders, quality requirements and capacity planning that needs to be saved and processed in the MES. But the MES also has to receive data from the production on shopfloor level including sensor values, process data, machine status, counter ticks or measurements that get processed to business relevant units. It is also possible that the MES provides specific data for the production, for instance process value specifications, target values or recipes. When it is not possible to access these data automatically, workers have to insert them manually. When the MES is seen as a hub, it is important that it is integrated horizontally as well as vertically, which means that the information is distributed on the MES level and also throughout the different levels of the pyramid. When using a MES, a database, either an external one or the MES itself, with high requirements is needed. Data has to be consistent, plausible and complete, which is also relevant for data acquisition. Furthermore, the database has to meet various security aspects so that the production is not endangered in case of power outage and other problems.

### **2.1.2 MES in the Context of Industry 4.0**

The term Industry 4.0 has been circulating in the manufacturing industry for a while, but there are different approaches to revolutionizing industrial production. It is agreed that the use of information technology and increased networking can significantly improve the efficiency of industrial production, potentially increasing productivity by more than 30%. However, achieving productivity gains requires the use of applications that take advantage of these new technologies and changes in organization such as MES.

Many visionaries dream of the ultimate networking of all resources and systems in manufacturing, where each machine is aware of its capabilities and each material knows which product it will become and for which customer it is destined. While this vision has evolved, a more practical approach seeks for each resource to be able to communicate with every other resource, allowing for early error detection and an increase in the flexibility and intelligence of the machines. Decentralized decision-making will become more common, but human experience will always be needed for final decision-making. This vision is known as a Smart Factory.

The concept of a Smart Factory presents new requirements for manufacturing software such as MES. IT systems need to become more flexible in order to adapt to short-term changes in manufacturing processes easily. The increasing interconnectivity generates huge amounts of data that require semantic structuring to become valuable. This "Big Data" must be processed in real-time to derive intelligent and useful insights. Industry 4.0 has to include human involvement and concrete use cases for optimal success.

The basic principles of a MES as defined in VDI guideline 5600 provide a good starting point for the industry 4.0 era. In particular horizontal integration and comprehensive penetration of the company are the most important issues here. The collection of data in production, their aggregation and also the processing and displaying in real time are essential for the success of a smart factory. However, there is much more to it than the processing of data and measured values into meaningful information. In the future modern companies will depend much more on the integration of all functions and modules across all elements involved in production. In addition, the importance of cross-system data exchange will increase, which is therefore another task a MES must fulfill.

In summary, it can be stated that MES systems would be ideally suited as a central information and data hub in a smart factory. But in order to do so perfectly today's systems still need to be further developed. Still the basic approach of the MES idea is already heading in the right direction. The goal must be to synchronize all systems and functions involved in production. Transparent data exchange is the basis for a functioning future scenario. This does not just mean the pure distribution of data, but also application-related preprocessing or aggregation. For example, ERP systems only need final summary values and not comprehensive detailed data from the individual status messages at the machine. An MES system, as the central hub, has the task of providing each user and each system with exactly the data that is expected or required.

The developments and enhancements of the MES 4.0 concept aim to satisfy market and customer demands, such as mass customization and resource efficiency, to ensure manufacturing companies remain competitive in an increasingly globalized market. MES 4.0 provides flexible software tools to support the individualization of mass-produced products and improve efficiency, which is crucial for maintaining and expanding market leadership in a high-wage, highly automated region. The concepts of Industry 4.0 and MES 4.0 will promote sustainable efficiency in manufacturing in the long term.

### **2.1.3 Arcstone's MES solutions**

Arcstone enables all enterprises to unlock the value of data [6] by making it visible and traceable. To do so they provide the MES arc.ops that comes with 20 modules. It features a specific tool for connecting to machines and pull the data named arc.quire, which works with industries standard protocols, including OPCUA and is also capable of manual data input. As a hardware integrator it writes the data into a SQL database inside the MES, from where arc.flow can distribute the data further to an ERP or other top level planning systems. A complete overview of Arcstone's MES solutions is displayed in figure 2.2. On the right side the gateway to the different stakeholders inside an enterprise is shown.

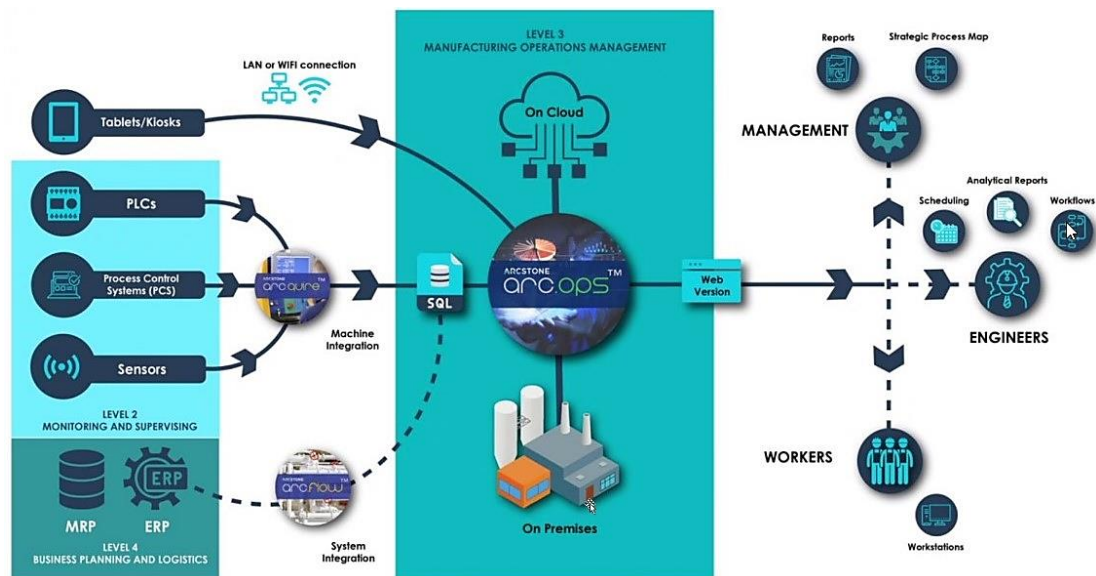


Figure 2.2: Arcstone's MES solutions [6]

When looked at Arcstone's MES solutions in the context of the ISA 95 framework, arc.quire connects the automation level to the MES, arc.ops is the MES itself and arc.flow is the link between ERP and MES. This is shown in figure 2.3.

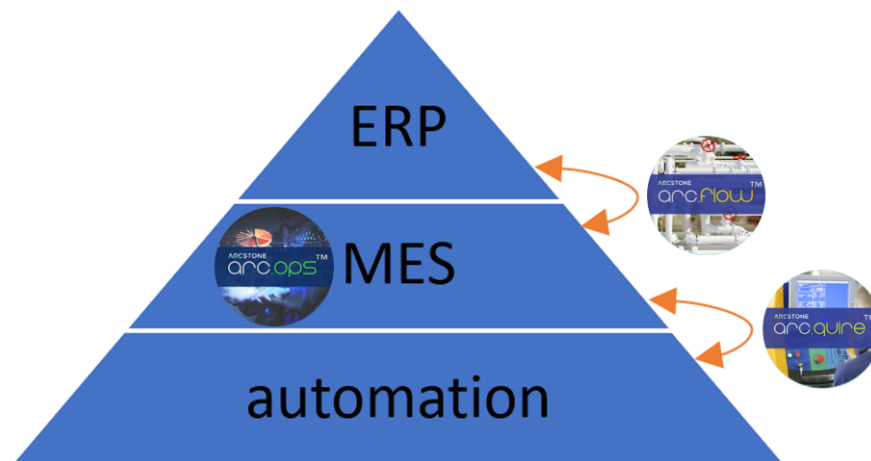


Figure 2.3: Arcstone in the context of the ISA 95 framework

## 2.2 Artificial Intelligence

So far, the factory is controlled by a PLC software. This software allows the developer to set clear instructions what the factory and each machine does. Instructions are worked through precisely. It furthermore needs to be controlled by workmen. The production process is started, viewed and stopped by humans. Furthermore, the quality of products needs to be screened. For a company this always means to pay wages. But how avoid these costs?

With “[the] branch of computer science that is concerned with the automation of intelligent behavior” [7], which is called artificial intelligence (AI). This could help to take over the task of the mentioned steps of work.

The factory is supposed to be monitored and controlled by artificial intelligence. While AI comprehend many ways of implementations, this project is supposed to use neuronal networks. This chapter will review neuronal networks.

### 2.2.1 Neuronal Network in General

When talking about neuronal network it is meant to have an artificial neuronal network. The idea is to create software that works similar like a human brain, a natural neuronal network. This network gives us our intelligence. It helps us to train our abilities, physically and more important mentally. We can adopt to situations and changes.

But how to build such a system? A computer has an architecture not matching with the brain. It is basically a calculator that can execute software which is based on mathematics. The attempt is to analyse human brain and to create models that reflect the brain.

The network consists of many small entities, that are similar like the neurons. They are working like single information storages, powered by electric voltage. As messages reach a neuron, also electrical energy is sent. The voltage level is rising. At some point there is too much energy in a neuron and it will reach out and sends its information. But where to?

“What fires together, wires together” is a quote by Hebb [8] describing that neurons often reach out to each other will connect. The more often they communicate the

stronger their connection will be. For an artificial neuronal network it should work similar. When raw data is put in a new system, there is no connection between the entities. The computer then need to find connections. Hebb is describing the idea of how it's done: Data often colliding, will have some kind of link. This is how the computer can derive clusters. Once these clusters are established it can be used to solve tasks. The more tasks it solves, the more information the system can collect. This information can also lead to an evolvement of the clusters.

Ertel [10] gets deeper into architecture of a neuronal network and provides schematics for illustration on this which are reused here.

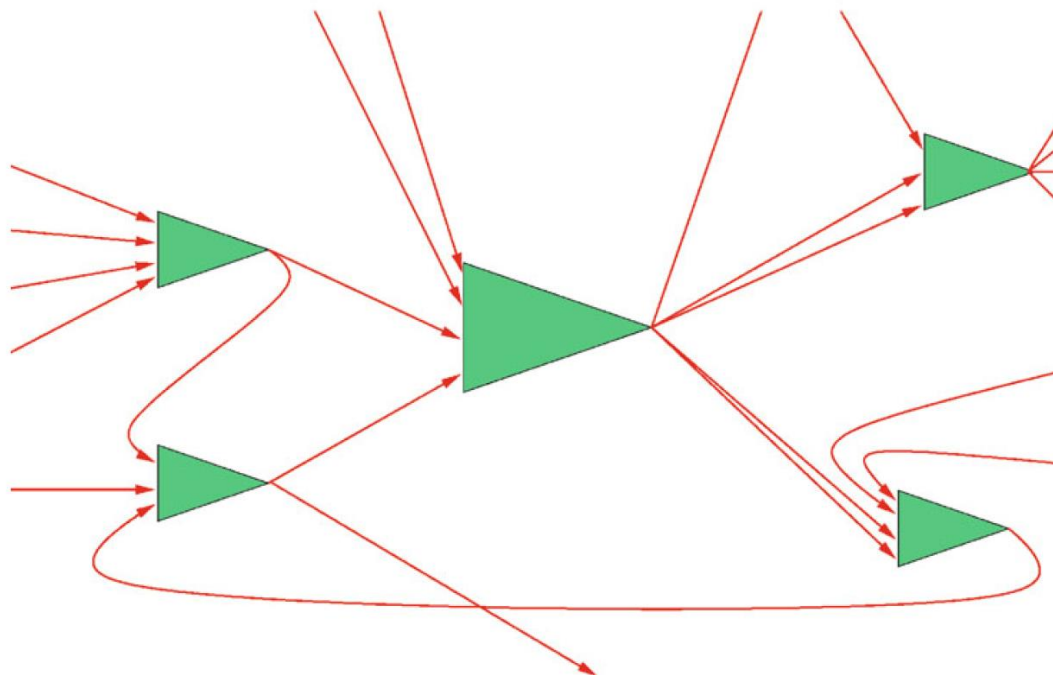


Figure 2.4: Model of network [10]

Figure 2.4 represents the network. The green triangles are the neurons connected by wires. The following shows how this is implemented.

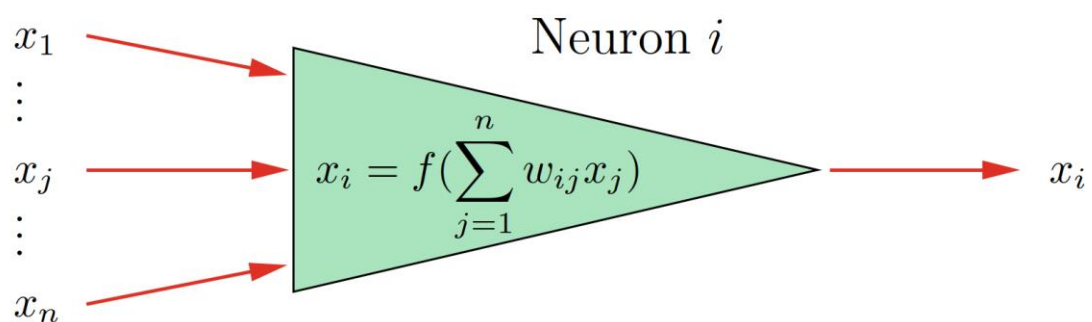


Figure 2.5: Mathematical Neuron [10]

The input voltage of each wire is represented by variables named  $x_j$  in the picture. The higher the voltage value of the human brain is, the higher the value of  $x_j$  is in the mathematical model. The factor  $\omega_{ij}$  is fixed to each  $x_j$  and represents how strong the connection to the respective previous neuron is. The better the connection, the more electrical power will go through. Despite this being a factor, its value can change, if the connection is changing as result of an adapting process. The product of each input and its factor will then be summed up. This is the output value  $x_i$ .

Despite its called learning network in most networks the input data is processed through fixed calculations. In most cases it is a kind of matrix multiplication or applying filters and masks. It is also possible to perform several steps of calculation. That's why an expert is required, who knows what the network is used for. The expert can then create the processing formulas.

What is it the network is learning then? It is the adjustment of the factors  $x_j$ . When saying the network is being trained, it means that the artificial network does the calculation of the factors based on the received input data. With that the network is then able to detect patterns.

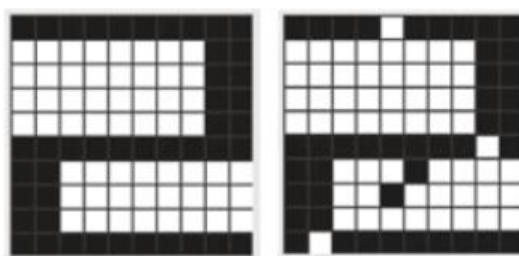


Figure 2.6: Pattern of number 2 [10]

In the above figure Ertel gives an example of pattern image recognition. The network is trained to detect numbers the number 2. The ideal picture of the number is shown in the left side of the figure. The network should then be able to recognize the 2 in other pictures. What makes a neuronal network special, that it is not just able to recognize the same image of the number, but also an image with changes. That is necessary in the real world as there are for example disturbances. This is shown on the right side of the figure. Some pixels of the 2 are not in place. If there was a standard recognition, it would fail to find the 2, but we want to recognize it as a 2. That is where the neuronal network comes in to play. What the network does is applying filters and masks to the picture. This will be done until the pattern of the 2 is recognized.



Another example is face recognition. If you take a picture of a person. The next day you want to recognize the face, it will be hardly possible. Because every day a person looks a bit different. And over the years a person's face can change immensely. The network still needs to be able to recognize the pattern of a human face. For that a series of pictures is required to enable the network to learn. It then can find similarities and use them to recognize a changed face.

The examples just considered image processing. But neuronal networks can be used in different areas. In the human brain, different parts are responsible for different tasks. So, every part is structured differently to match its requirements. For artificial networks it is similar. Different networks suit different tasks. That's why there are many different models that try to attempt to rebuild a part of the brain. Each model has its benefits an application its suits. For this project it will be the convolutional neuronal network and the recurrent neuronal network.

### **2.2.2 Convolutional Neuronal Network**

The issue of most networks is that they have a vast amount of data as input. Processing all of it in just one step would take very long. This is addressed by having several steps of processing. In most networks the first step is to extract attributes. These are often referred as preprocessing layers. They often consist of filters, similar like described above and capable of learning. With every layer different things can be extracted. If you take image processing as an example, there is one layer to recognize edges of the picture and a layer to extract a face. In most cases the more layers you have the more attributes you can extract. In a second step attributes will be interpreted.

In a convolutional neuronal network (CNN) most of the filters are based on the mathematical convolution. The advantage of the convolution is that it reduces the number of the factors. This helps fasten up the calculation operation.

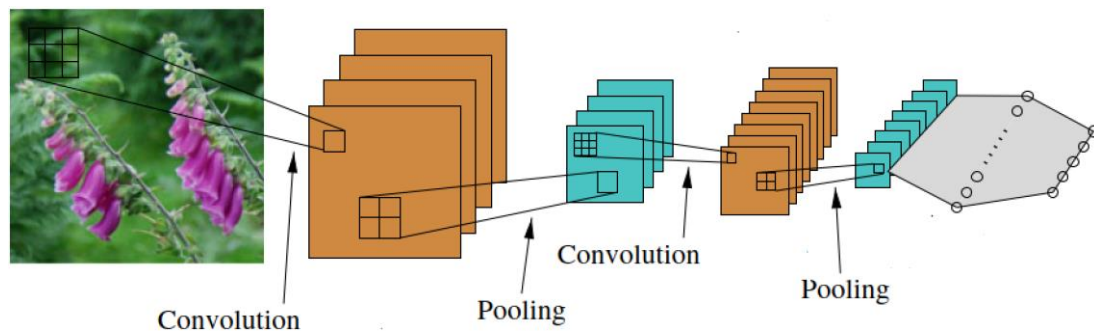


Figure 2.7: CNN working principle [10]

The figure 2.7, Ertel visualizes the working principle of the filters for image processing. The input is a picture of flowers. The picture then will be filtered by using the convolution. The result is a smaller picture with highlighted attributes. The next step is pooling. Information that are similar get bundled here. This data reduction helps to save storage space and ultimately processing time. Repetitions of the two steps will be executed. The goal is to have a smaller set of extracted data.

The whole process is complex. Several mathematical functions need to be used the right way. Additionally, as there are less factors, they need to be more precise. This is leading to long training times. It also requires more validation, to check if the learned factors are precise enough. Otherwise, a too big error in factor value can lead to misinterpretation.

Nowadays, scientist and engineers need to have a huge knowledge about every aspect of the network to make it work correctly. It often requires a lot of experience to know which filters to use and how to order them. Also, on the mathematical side expertise is required. The filters can consist of many different mathematical operations. This can also lead to long development times.

The results of correct implemented convolutional neuronal networks are satisfactory though. For example, face recognition has not been possible with classical computation. Despite it is easier to imagine this kind of network is used for image processing it is also possible to use it in other branches. It just gets more abstract, as the constructor of the network needs to be able to create filters that work for data sets that come as example from a factory.

### 2.2.3 Recurrent Neuronal Network

The second network that this chapter covers is a recurrent neuronal network. Compared to the human brain this is working like the short-term memory. It is capable of remembering data input for a while and even take it in consideration for new learning processes.

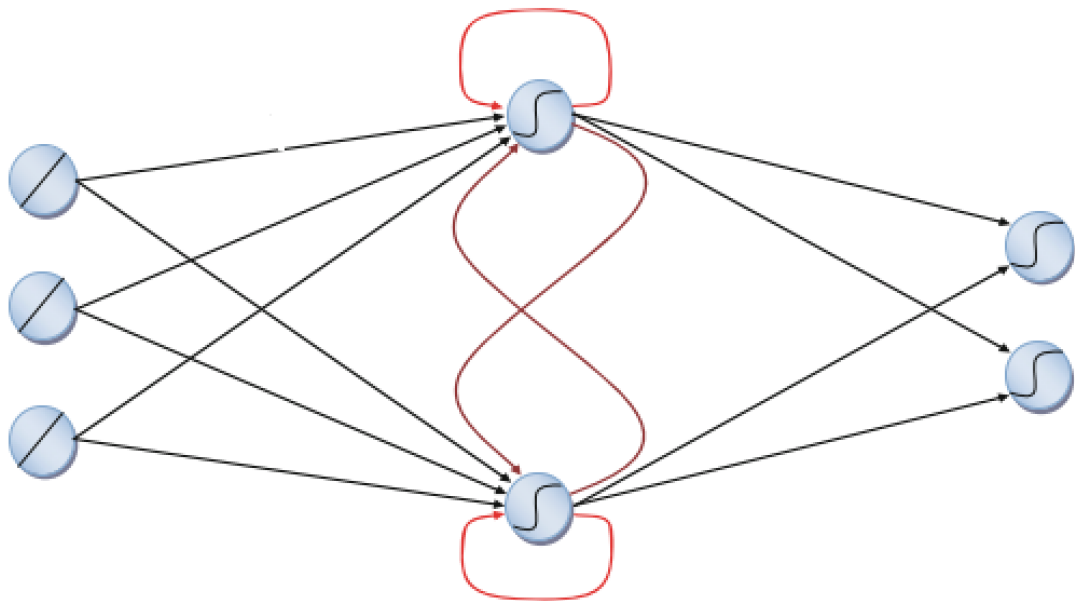


Figure 2.8: Recurrent Neuronal Network [11]

Figure 2.8 from Grum [11] is a simplified visualization of the working principle. The blue points represent the neurons with connections between them. The processing order is from left to right. This kind of network is special, because of the red connections. The dark red connections are between the same layer level, the brighter red connections are to a neuron itself. This can lead to processing of a specific information for several times. It can even be transferred between different neurons of a layer for several times, which makes an information available a longer time. This construction of a network is the counter part of the human's short-term memory.

### 2.2.4 Summary Neuronal Networks

For humans it is often easy to acquire a series of data. The challenge of artificial neuronal networks is to provide the computer with the information. It is also required to process as much raw data as possible, without giving many rule bases. The result

should be software that is capable of learning from data, extract patterns and recognize them on new data.

Neuronal networks still have a lot of potential to be unlocked in future. Scientists researching on the matter for a while and still will. The main issues are that the inspiring model, the human brain, has a huge size with approximately  $10^{11}$  neurons and that its hardly possible to access the human brain. But with ongoing research artificial neuronal networks will grow in power.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Flow Chart**

In this chapter two flow charts will explain the execution of the project, divided into the topics MES (Figure 3.1) and Lernfabrik (Figure 3.2).

After the start of the MES project the objectives and requirements were defined. They serve as a guideline for the following parts of the project. To have basic knowledge about the subject, a literature review was performed, followed by the implementation, with the steps to collect the data from PLC to UPCUA server, fetching the information from the server and writing them into the UTHM database and finally to create a dashboard. After these steps are completed, the fulfilment of the requirements needs to be checked. If a requirement is not fulfilled, the above-named implementation steps will be repeated. As soon as all requirements are met, the report can be created and submitted.

The project AI-Learning-Factory begins with a definition of the project's objectives and requirements, which serve as a guide throughout the project. They also need to be checked at the end of the project as the penultimate step. If the requirements are not fulfilled, all previous steps need to be checked until the requirements are met. Then the report can be written. After the introduction, a literature review about neuronal networks were conducted. This is followed by the implementation of the project. Steps that are performed are named in the following text. The first step here is to understand the learning factory by doing the AI-Campus course. After that several task branches open. The left one is about reviewing and translate the AI-Lernfabrik course. The middle branch includes understanding the Lernfabrik tools,

perform the course tasks and write a documentation for the solutions. The first two branches then combine again as the analysis about the learned is required. The branch on the right side includes the analysis of the simulation hardware and to check if an implementation using the AI-Lernfabrik software is possible. In the final project execution step, the information of all branches is collected and the requirements are checked. As soon as the report is finished, it will be submitted.

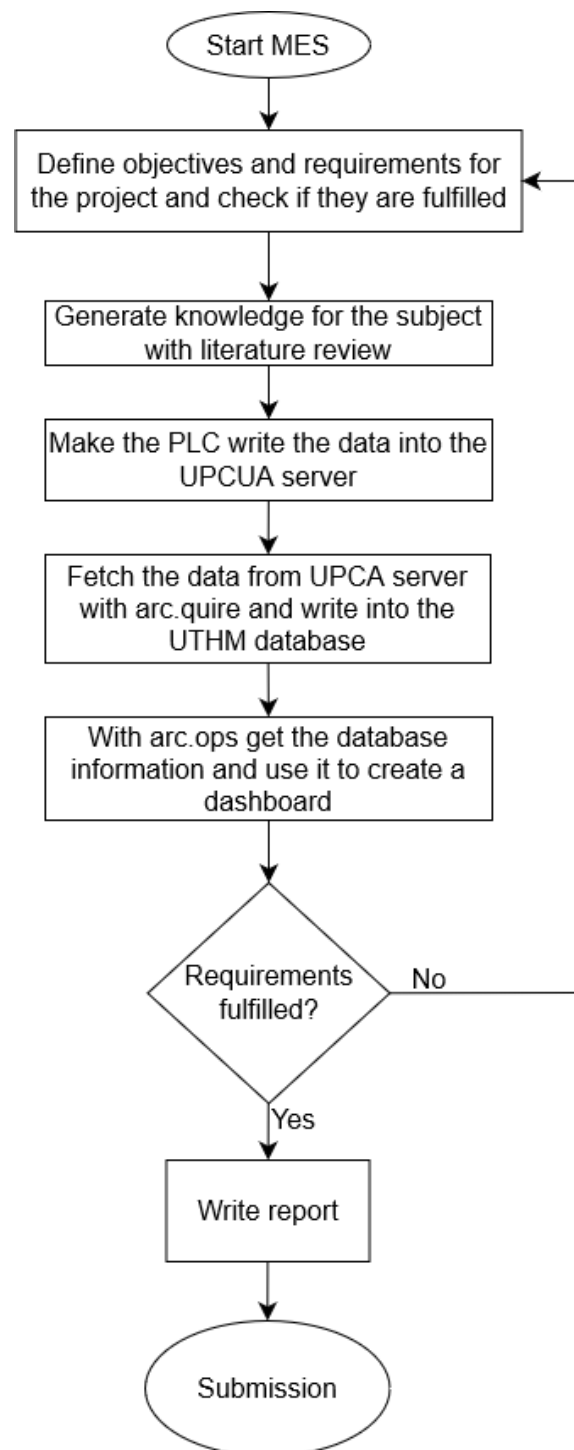


Figure 3.1: MES Flowchart

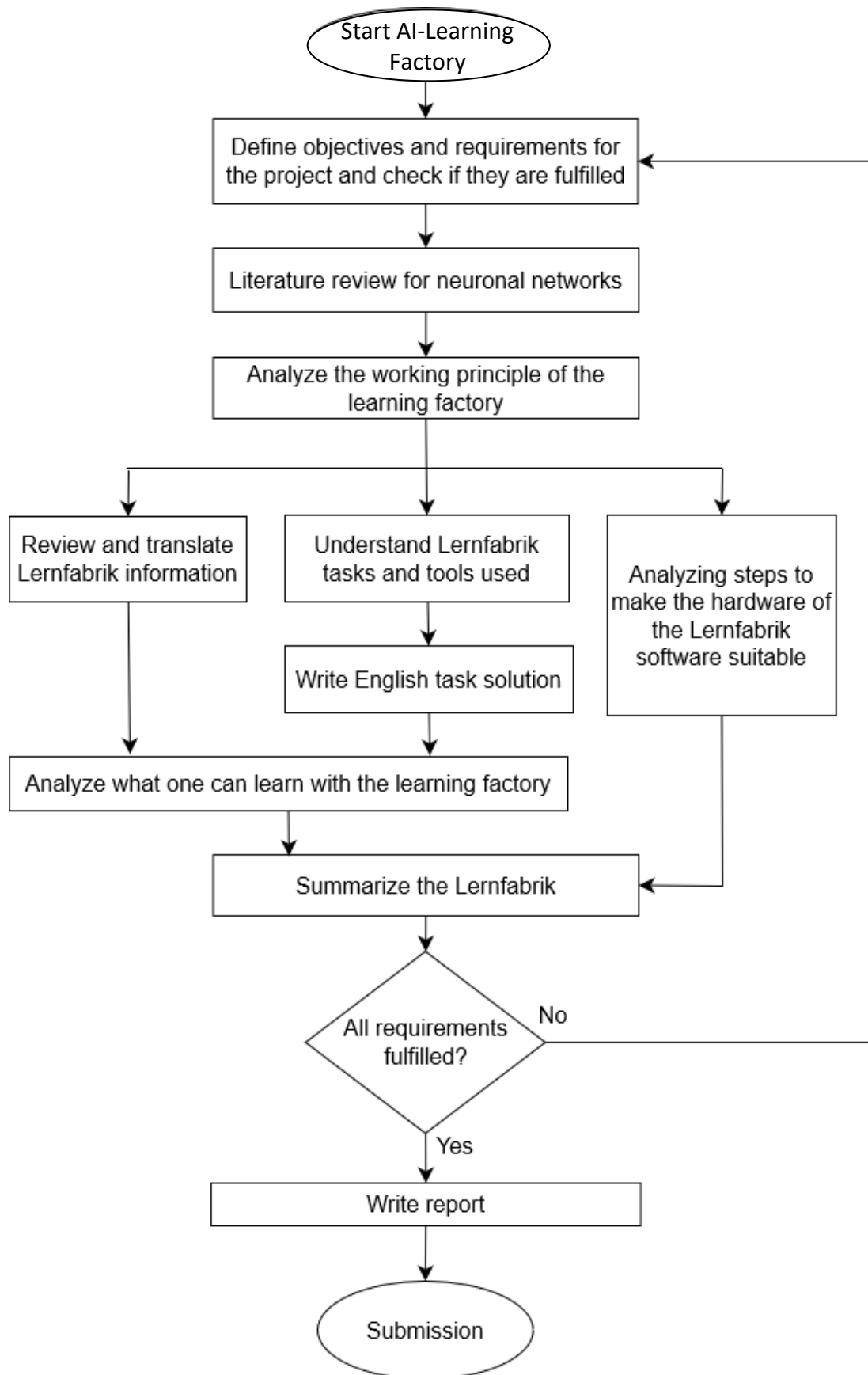


Figure 3.2: AI-Learning Factory Flowchart

### 3.2 Previous State of the Learning Factory

In this chapter the setup of the learning factory in the laboratory is described and it is shown from what basis the project has started. This is done to clarify the existing problems in an already digitised factory.

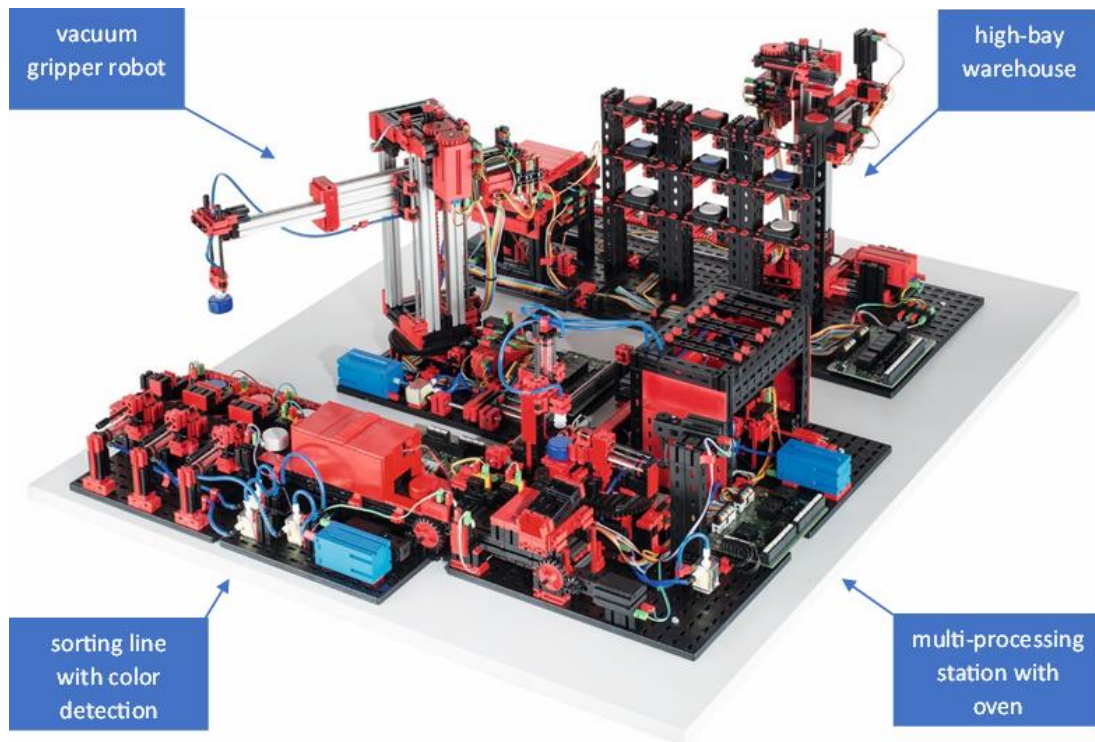


Figure 3.3: Fischertechnik Smart Factory [13]

The Fischertechnik Smart Factory [13] is used in the project as depicted in figure 3.3. It is a learning factory that comes with many features and in the following a typical process is described.

When the process is started, the transport arm of the high-bay warehouse (HBW) moves to the storage system, picks up the wanted workpiece and places it in the output station. There it is conveyed to the pick-up position of the vacuum gripper robot.

The vacuum gripper robot (VGR) picks up the workpiece from the output station of the high-bay warehouse (HBW) and places it on the slide of the oven of the multi-processing station with oven (MPO). There the workpiece is carried in, fired and moved out again. The transport carriage with vacuum suction pad then transports the workpiece to the milling machine. There, the workpiece is placed on the rotary table and is milled.



After the milling process, the workpiece is pneumatically pushed onto the conveyor belt of the sorting line with colour detection (SLD). On the conveyor belt, the workpiece passes through a colour recognition system. Depending on the colour detected, the workpiece is pneumatically pushed onto the corresponding chute. Figure 3.4 illustrates the schematic plan of the learning factory.

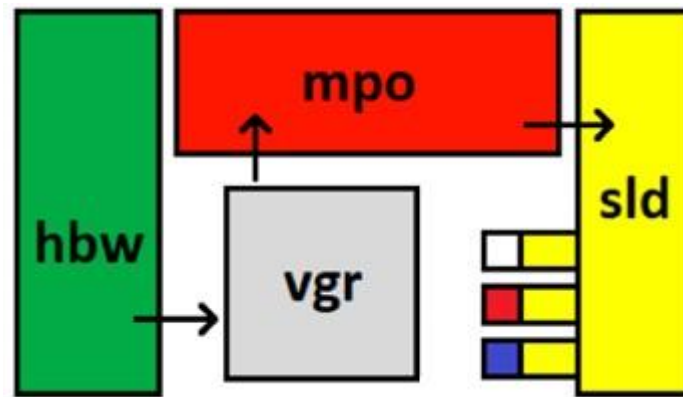


Figure 3.4: Schematic Plan of the Learning Factory [15]

As for the start of the project the learning factory is connected to a programmable logic controller (PLC) from Beckhoff and controlled with their software TwinCAT 3. This does not fulfil all the requirements of a factory in the context of Industry 4.0 as described in the previous chapters. Therefore, the projects deal with making the learning factory ready for the future.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Dataflow from Automation Level to MES

In this chapter the data flow from the automation at shopfloor level to the MES software is described. For that purpose all machines need to be digitized and sensorized in order for a PLC to control the production steps and read the input and output data. With this done the data can be written into a server, which is then also accessed by the MES, where it can be processed, analysed and displayed. Figure 4.1 illustrates the dataflow from the automation software TwinCAT to the MES arc.ops and where the data is stored during this process.



Figure 4.1: Dataflow from the automation software to the MES

In the following subchapters each connection step is evaluated in more detail. First, the connection of the PLC with the server is described. Then the linking of the server data to the MES using arc.quire is discussed and lastly, the access of the data inside the MES is elaborated.

#### **4.1.1 Architecture of the Connection between Automation Software and Server**

One of industries standard protocol that is widely used in automation is OPCUA, featuring flexible data transport across platforms and making machine data readable for other applications. Furthermore, it allows the user to easily write data into an OPCUA server.

When it comes to connecting the PLC to the OPCUA server there are three components that must be arranged and connected correctly. The TwinCAT environment controls the PLC and the input and output data of the PLC is written into the OPCUA server. The OPCUA expert can be utilized to test the connection and view the data. On the device that hosts the OPCUA server, TwinCAT XAR (runtime only) or TwinCAT XAE (runtime and engineering) has to be installed as well as the function TF6100 OPCUA [14]. On each PC where a TwinCAT software product is used, the license TF6100 has to be activated. When setting up an OPCUA server authentication can be implemented by defining an username and password.

Figure 4.2 illustrates possible architectures of this system. Firstly, it is possible to run all components on one PC, which is obviously the easiest method. If it is wished to distribute the components on more than one PC, there are several ways to do so. For example, the TwinCAT environment with the PLC and the OPCUA server can be run on one PC, while the server data is accessed from another PC. Basically, every combination of the elements is possible, but it is notable, that a functional TwinCAT environment supporting the operation of the PLC is always on one device.

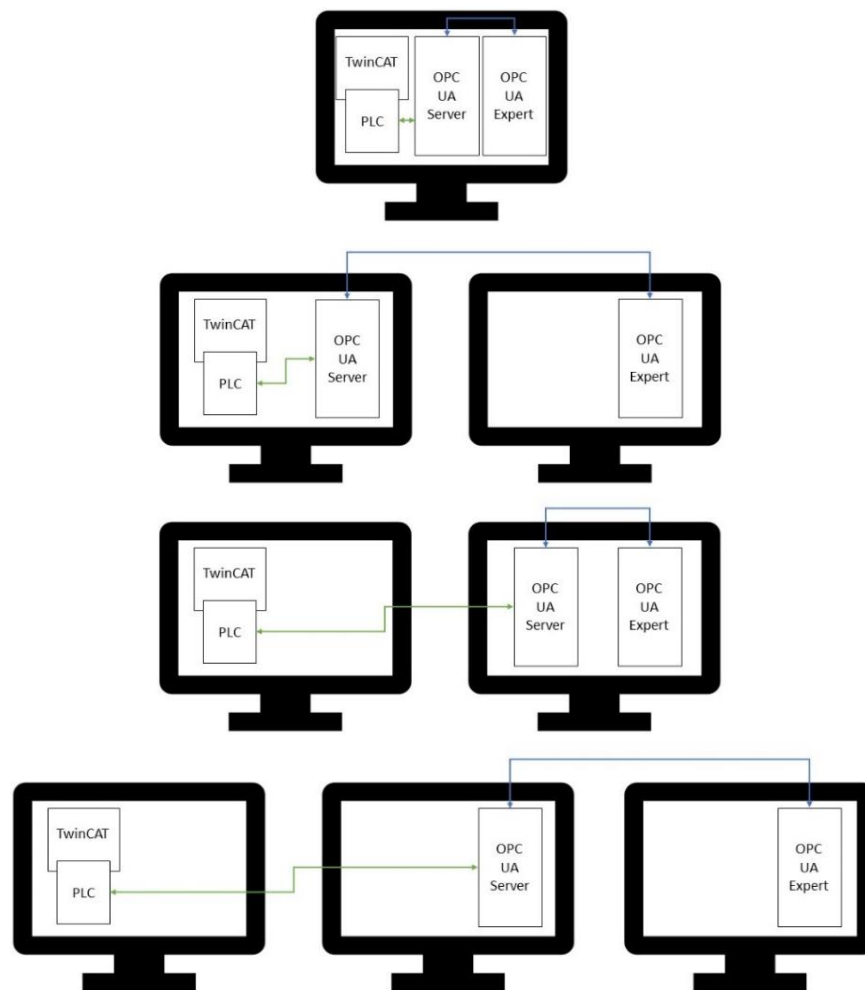


Figure 4.2: Architecture examples of the connection

Another way of connecting the components, when they are distributed on two PCs is shown in figure 4.3. The left PC is the engineering computer, where the PLC program for the server is developed and the right PC is an industrial computer from Beckhoff. Here TwinCAT also has to be installed, but functions only as a runtime for the PLC. Furthermore, the OPCUA server is hosted on the industrial PC. The access to the server data happens from the engineering computer with the OPCUA expert. In this architecture it is important that a valid OPCUA license is activated in each TwinCAT software product.

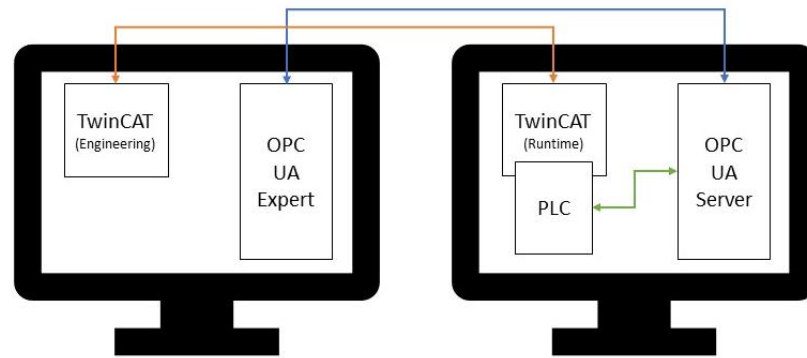


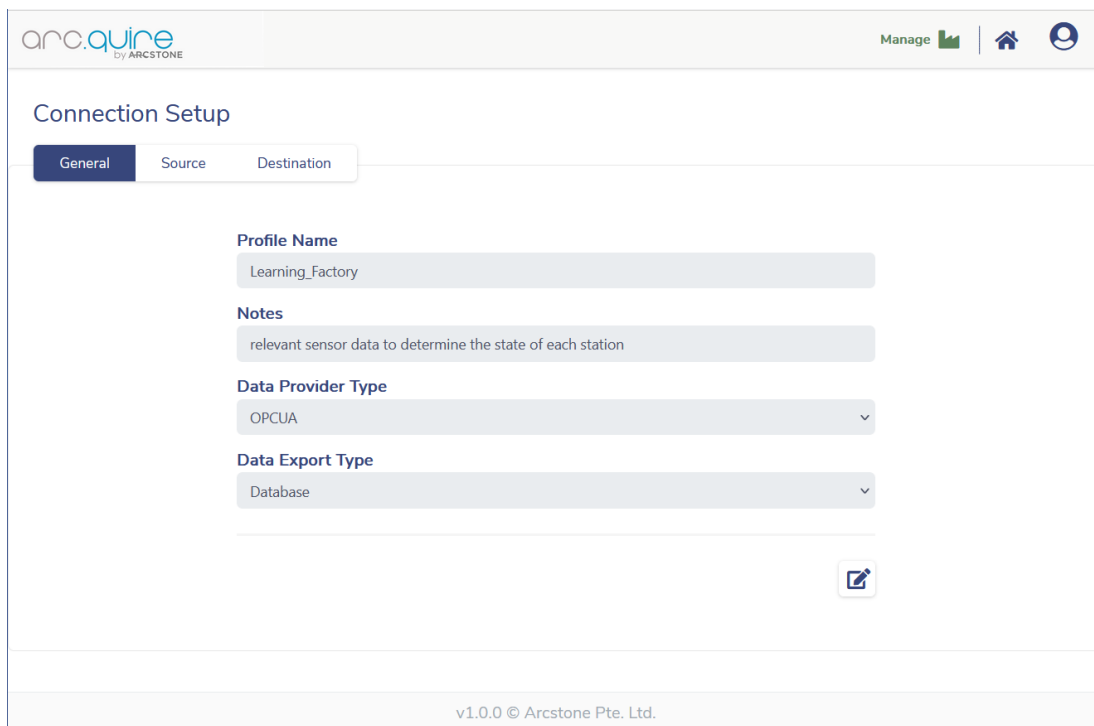
Figure 4.3: Used architecture for the learning factory

#### 4.1.2 Pull Data from the Server to the MES Database

Once the OPCUA server is set up, the data can be utilized in the MES. A new arc.quire profile needs to be set up in order to pull the data to the MES. A meaningful name is chosen and a description of the profile can be added in the notes. Next, the data provider and export types have to be selected, whereby OPCUA is the provider and Database is the export for the learning factory. Refer to Figure 4.4 for the example settings.

In the following step the data source has to be defined as illustrated in Figure 4.5. The OPCUA server URL has to be inserted as well as the publish interval and the authentication credentials if required. Afterwards, add the data that is wanted for the MES by giving each a meaningful name and inserting the corresponding node ID. If a test connection is running, the current data value is displayed in the right column.

Before the profile is ready to run, the data destination has to be specified, which is shown in Figure 4.6. Most of the fields here are predefined and there is no need to change them. But the connection string has to be modified to match the username and password of the Arcstone server.






The screenshot displays the 'arc.quire' web interface, specifically the 'Connection Setup' page. The header includes the 'arc.quire by ARCSTONE' logo on the left and 'Manage' with icons for a bar chart, home, and user profile on the right. The 'Connection Setup' section has three tabs: 'General' (selected), 'Source', and 'Destination'. The 'General' tab contains the following fields:

- Profile Name:** A text input field containing 'Learning\_Factory'.
- Notes:** A text area containing 'relevant sensor data to determine the state of each station'.
- Data Provider Type:** A dropdown menu with 'OPCUA' selected.
- Data Export Type:** A dropdown menu with 'Database' selected.

Below these fields is a small icon of a document with a pencil. At the bottom of the page, the version 'v1.0.0 © Arcstone Pte. Ltd.' is displayed.

Figure 4.4: General connection setup of arc.quire profile

arc.quire  
by ARCSTONE

Manage   

### Connection Setup

General


Source

Destination

Server URL

opc.tcp://10.61.44.184:4840

Publish Interval

1000 








☒ Requires Authentication


User Name

123456

Password

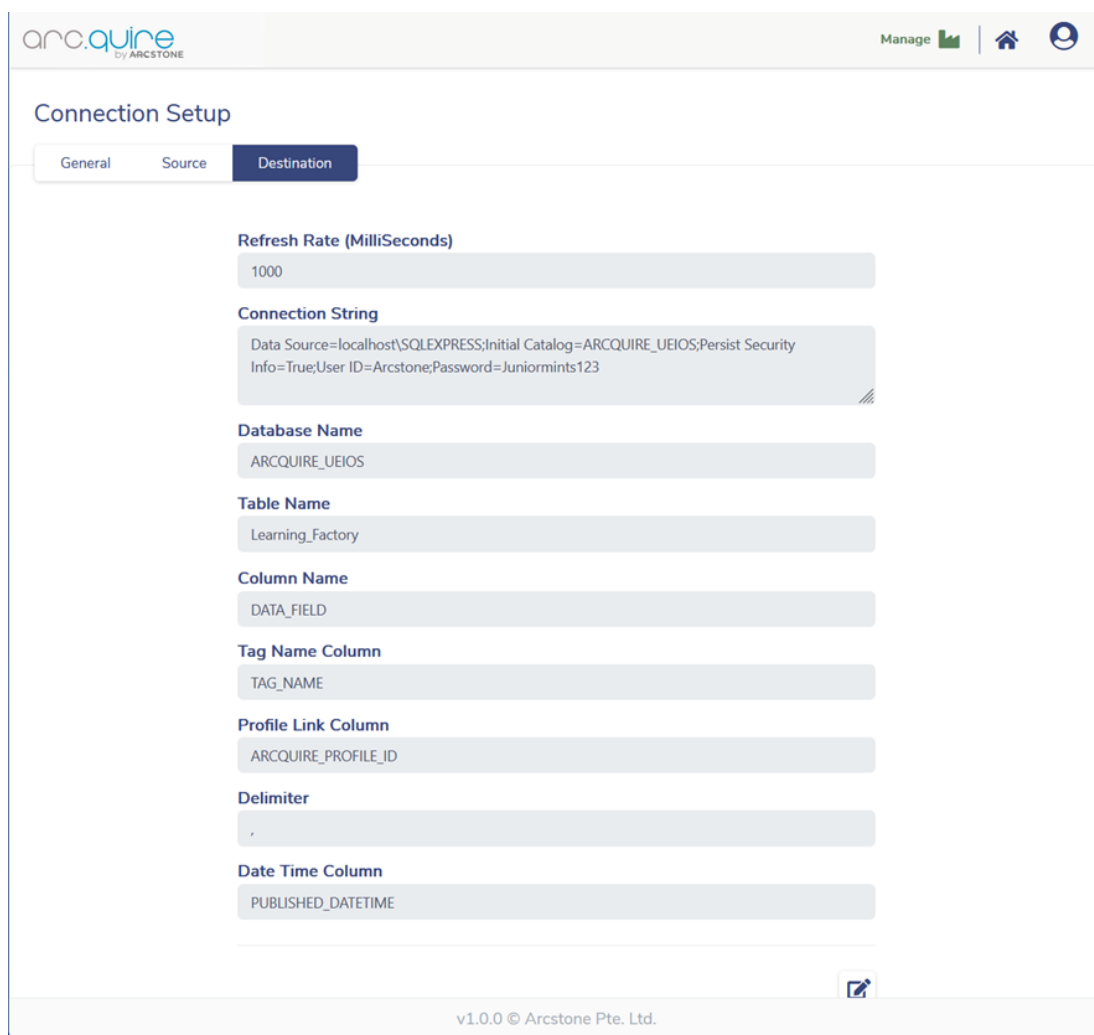
.....

ID	Display Name	Node ID	Value
1	S2_I2	ns=4;s=POU.S2_I2	
2	S1_I1	ns=4;s=POU.S1_I1	
3	S4_I5	ns=4;s=POU.I5	
4	S4_I4	ns=4;s=POU.I4	
5	S3_I1	ns=4;s=POU.S3_I1	
6	S3_I6	ns=4;s=POU.S3_I6	
7	S4_I6	ns=4;s=POU.I6	



v1.0.0 © Arcstone Pte. Ltd.

Figure 4.5: Source setup of arc.quire profile



arc.quire  
by ARCSTONE

Manage | Home | Profile

### Connection Setup

General | Source | **Destination**

**Refresh Rate (MilliSeconds)**  
1000

**Connection String**  
Data Source=localhost\SQLEXPRESS;Initial Catalog=ARCQUIRE\_UEIOS;Persist Security Info=True;User ID=Arcstone;Password=Juniormints123

**Database Name**  
ARCQUIRE\_UEIOS

**Table Name**  
Learning\_Factory

**Column Name**  
DATA\_FIELD

**Tag Name Column**  
TAG\_NAME

**Profile Link Column**  
ARCQUIRE\_PROFILE\_ID

**Delimiter**  
,

**Date Time Column**  
PUBLISHED\_DATETIME

v1.0.0 © Arcstone Pte. Ltd.

Figure 4.6: Destination setup of arc.quire profile

With that done, the profile is ready to be started by clicking on the play button and once it is running the selected data is automatically read from the OPCUA server and written into the SQL database. To view the database in the Arcstone server of the UTHM Microsoft SQL Server Management Studio can be used. Figure 4.7 contains a screenshot of the Microsoft SQL Server Management Studio where the first 1000 elements of the database are viewed.



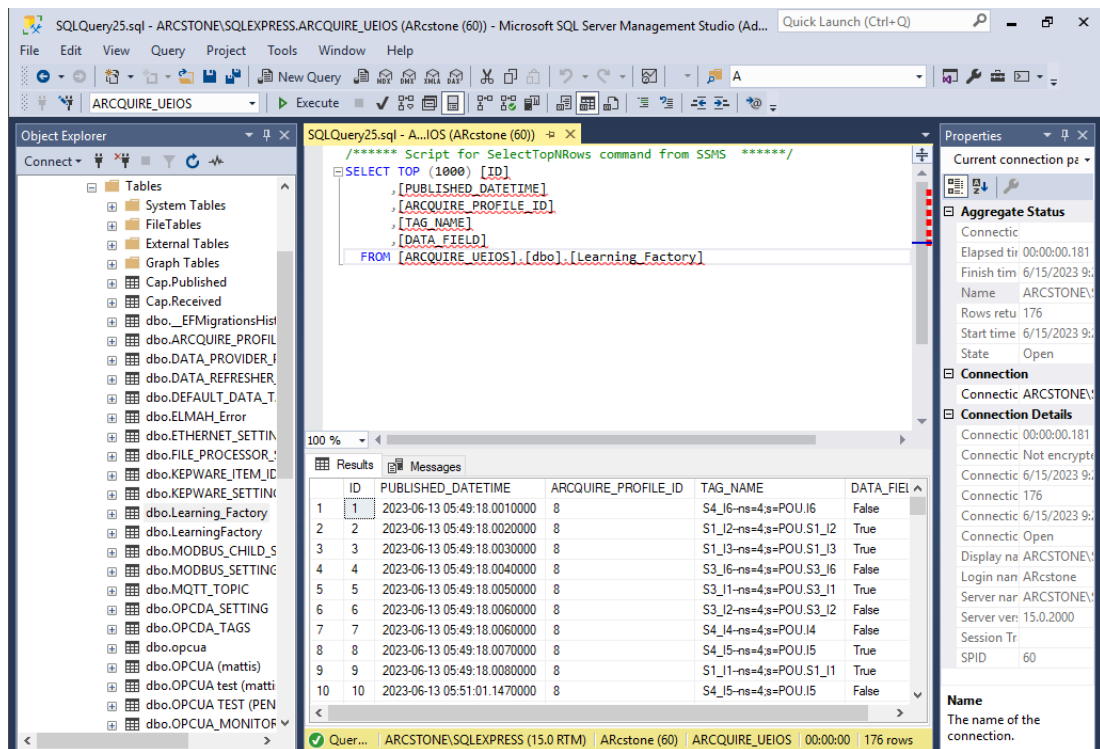


Figure 4.7: Microsoft SQL Server Management Studio

### 4.1.3 Access Data Inside the MES

As the data is now available for Arcstone's MES arc.quire a dashboard for visualization is created. When a new dashboard is made, the first thing to do is adding a data source as shown in figure 4.8. Insert a name, a short description and the connection string from the destination of the arc.quire profile that was created earlier.

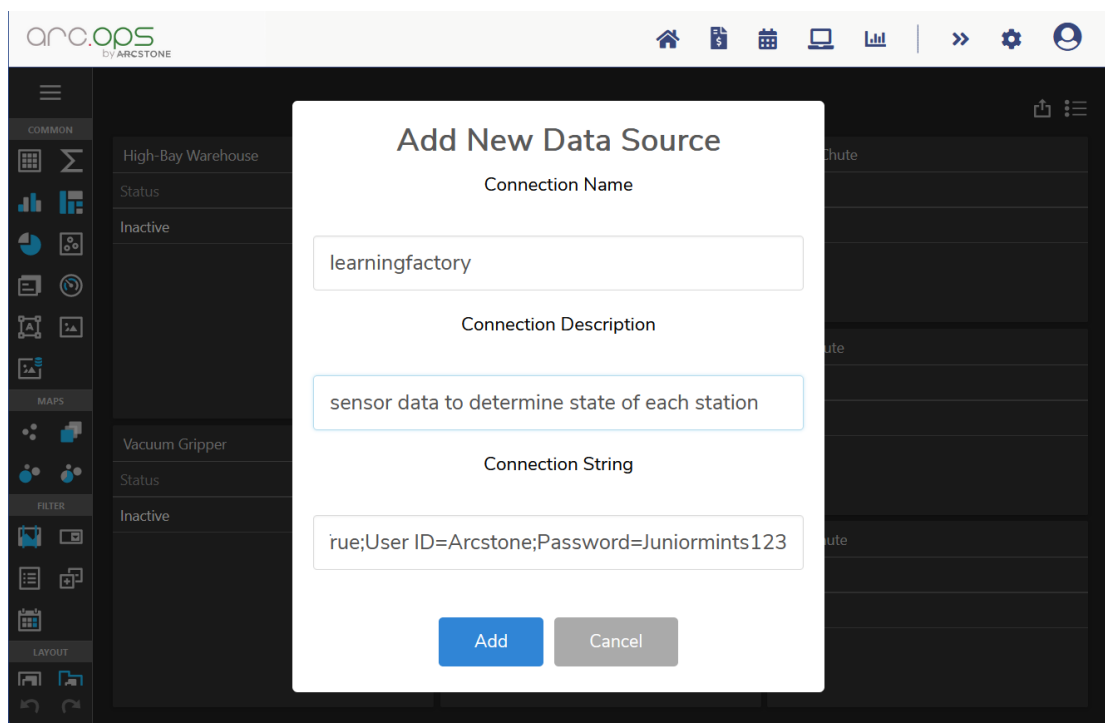


Figure 4.8: Add new data source to a dashboard

Navigate to data sources, which opens the dashboard data source wizard and create a new source. The data is stored in a database, so this source type is selected. Choose the previously added data source as connection. Figure 4.9 shows the window where a query has to be created. The query builder can be utilized to simplify the process.

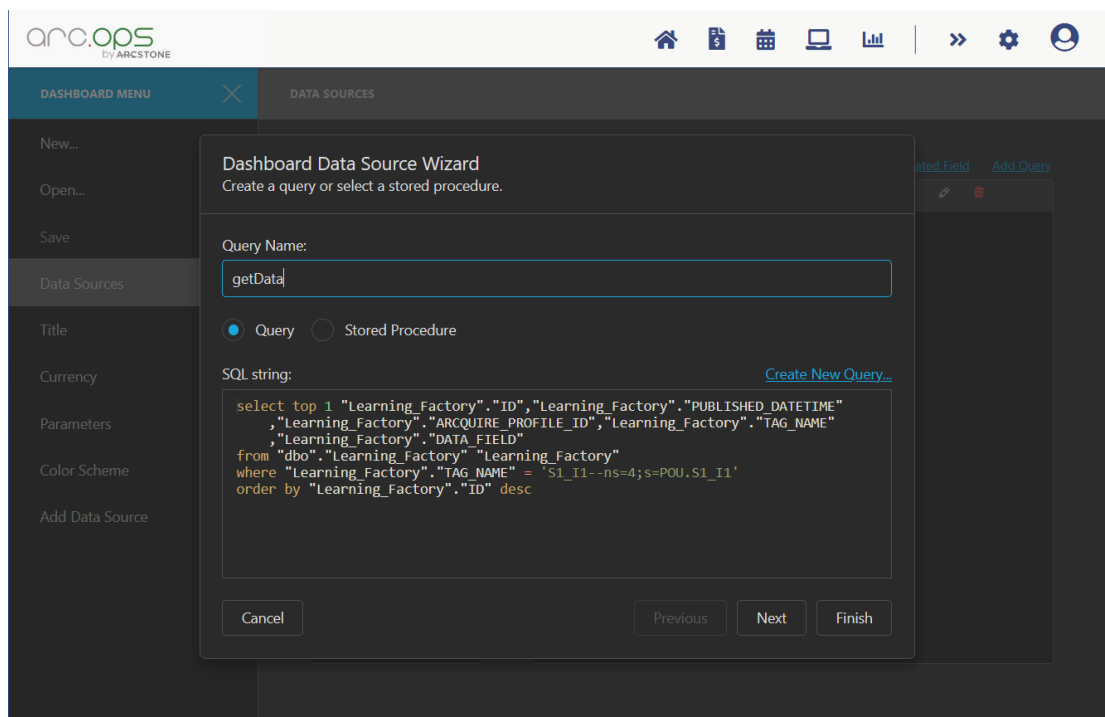


Figure 4.9: Dashboard data source wizard

By using drag and drop the required table can be selected as well as the columns. The SQL string can then be modified to fulfil the users demands. In the shown example only the newest value of S1\_I1 is selected by filtering for the TAG\_NAME and putting the ID in a descending order. Other methods of an SQL query can also be used here. Now the data is ready to be displayed on the dashboard.

For that purpose, Arcstone offers a variety of options. The most common and easiest way of displaying data is the grid, which is also used for the Learning Factory, but there are also elements like the bar graph or the pie chart. The grid is basically a table, where data from the data sources can be added in the columns. Figure 4.10 shows the settings for a grid. The data either be directly displayed or processed further by adding a calculated field. This is useful if the output needs to be normalized.

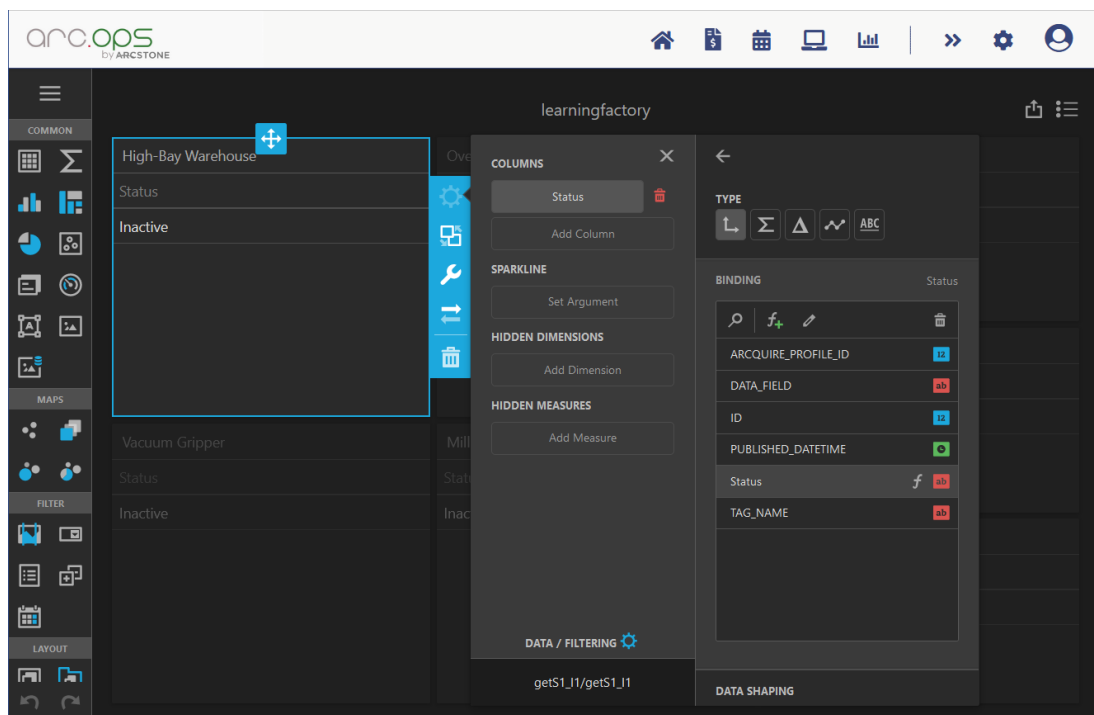


Figure 4.10: Bind data to grid

## 4.2 Creating the Dashboard

The dashboard should clearly visualize which station is currently running. To achieve this, appropriate sensors that indicate the status of each station have to be selected.

The first station is the high-bay warehouse. Here the end-switch S1\_I1 of the transportation arm is utilized to determine whether the station is running or inactive. If the switch is pressed the sensor value equals true and the station is inactive. As soon as the transportation arm starts to move to the storage system the switch gets released and the sensor value turns to false. When a workpiece has been taken out of the storage system and is put on the output station the process is finished and the end-switch is pressed again.

The vacuum gripper robot then picks up the workpiece from the output station. To reach the pick-up position the robot needs to extend its arm and the end-switch S2\_I2 is released. After putting the workpiece on the slide of the oven the arm returns to its initial position and the end-switch is triggered again. Just like for the first station a sensor value that equals false means that the robot is running.

The multi-processing station is divided in two separate stations for the visualization. Starting with the oven, which is running as long as the end-switch of the slide S3\_I6 is pressed. In contrast to the previous stations the logic is inverted here, meaning that a true sensor value stands for a running oven. When the workpiece leaves the oven, it is transported to the milling machine. The end-switch of the rotary table S3\_I1 indicates the state of this station. If the milling process is running the switch is released, thus its sensor value is false during this time.

The state detection of the sorting line with colour detection differs from the previous stations as it is not determined if it is running or inactive. Instead, the state of each chute is derived from the sensor values and the dashboard should show if it is empty or full. The light barriers S4\_I4, S4\_I5 and S4\_I6 return true if the chute is empty and true if the workpiece is sorted.

A summary of the sensor selection and their meaning for the states of the stations and the chutes can be found in the following table 4.1.

Table 4.1: State of each station and the chutes

Station	Sensor	State	Sensor value
High-bay warehouse	S1_I1	Running	false
		Inactive	true
Vacuum gripper robot	S2_I2	Running	false
		Inactive	true
Oven	S3_I6	Running	true
		Inactive	false
Milling machine	S3_I1	Running	false
		Inactive	true
White chute	S4_I4	Full	false
		Empty	true
Red chute	S4_I5	Full	false
		Empty	true
Blue chute	S4_I6	Full	false
		Empty	true

As not all sensors have the same meaning the sensor value needs to be modified, which is achieved by adding a calculated field with a simple if-query as shown in figure 4.11. The first input in the brackets of the if-query is the statement that is checked. If the statement is true, the second input is returned and else the third input is returned. The example shows the calculation for S1\_I1.

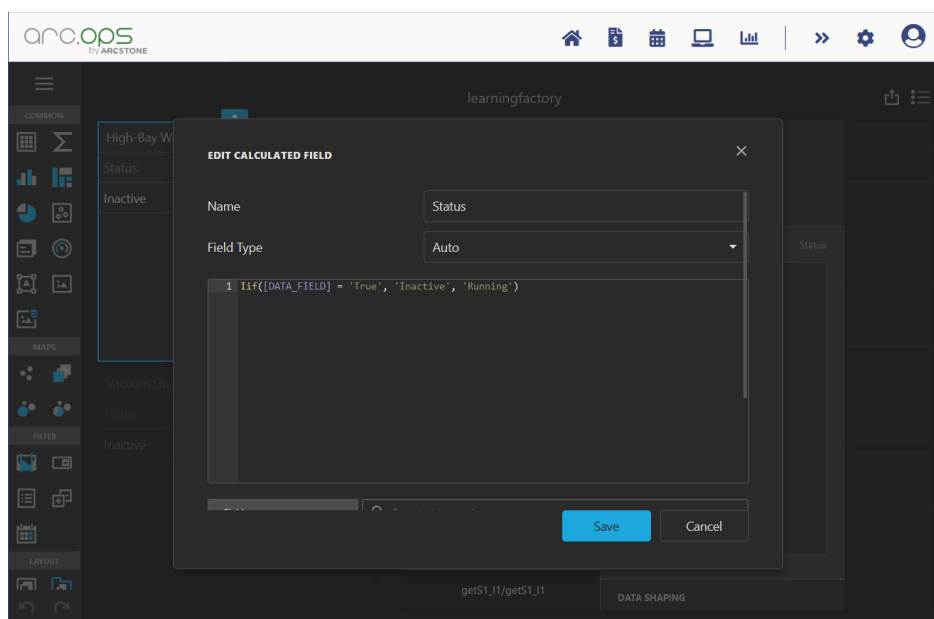


Figure 4.11: Calculated field

For each of the seven sensors a grid is added into the dashboard and to make the dashboard easier to understand an image of each station is added below the grid. The final dashboard is shown in Figure 4.12 below. In the top left corner is the high-bay warehouse, below that the vacuum gripper is shown. In the middle column the oven and milling machine are shown and on the right side are the chutes of the sorting line. In the screenshot the oven is running and only the white chute is full.

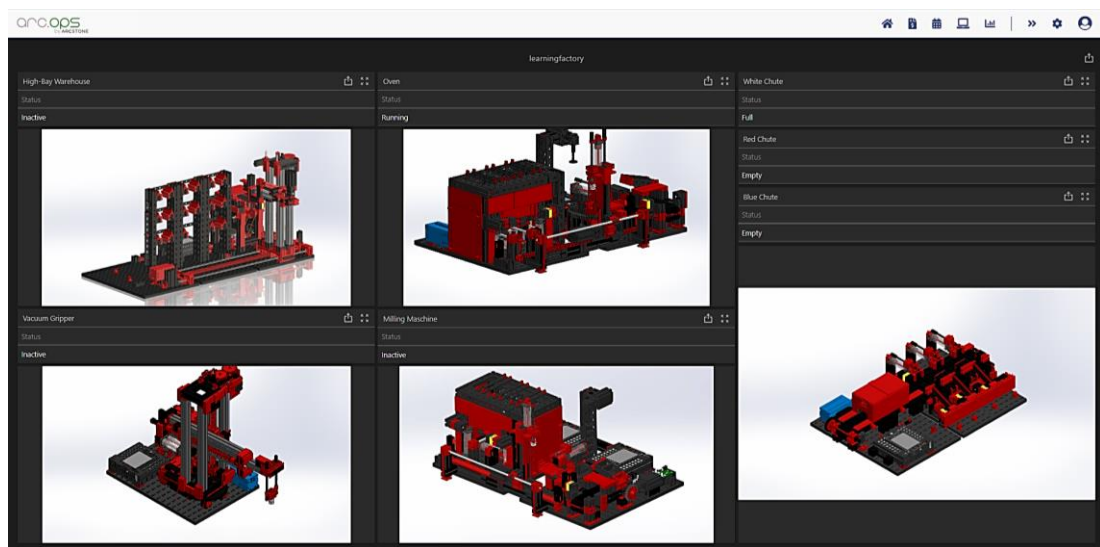


Figure 4.12: Dashboard

### 4.3 How the AI-Lernfabrik Works

This chapter will evaluate what one can learn from university Albstadt-Sigmaringen's AI-Campus course [15] of the learning factory. Notice that the Lernfabrik differs from the learning factory that is used in the laboratory, which makes it hard to apply on the learning factory. Therefore, this should only provide information on what may be possible with the model in the laboratory, rather than actually make it work. More about this can be found in chapter 5.

The focus will only be on the software tools and their application. The overall goal is to set up a system that is capable of monitoring and controlling the factory. The course is showing all the steps which are necessary for implementation. In the following paragraphs each step will be summarized and analysed. The full course and the task solutions can be reviewed in Appendix A and B.

To monitor the factory, the artificial intelligence system needs to fetch the available data and store them into a cloud. Most of the available data are sensor signals. The

data communication is done by MQTT. The first thing that is taught is a way to read the incoming data and write it into a database. For that purpose, the software tool Apache Nifi is used. The programming is done by a kind of block programming. The course is providing a sample code. As user the task is to control and change the source URI. The sample code enables the user to understand the modus operandi of the tool.

For the next step a python script is introduced. Its purpose is to request current states and to place orders, that get executed by the factory. The script will be accessed via Jupyter Hub. Basic understanding about python scripts and the python programming language are of advantage. The user should also be familiar with command language as the execution is performed via the Python Hub.

As a next step, sensor data is put into graphs. This step is not essential for the factory control, but interesting for factory analysis. Therefore, the data is transferred into the Amazon cloud. This enables the user to use the AWS Redshift tool. The course is accessing to the Redshift tool via another python script.

A second possibility of communication is introduced in the next step. The tool to establish the communication is Node-RED. The communication standard is OPCUA. Node-RED is a tool that is using block coding. The course is providing sample code. The tool is used to read from the factory, put the data into the Microsoft SQL database and control the learning factory. With that the course introduced the two most common used communication standards and useful tools to work with.

In the penultimate step the preparation for the artificial intelligence is done. Therefore, data editing is performed to have it neuronal network ready. The course is teaching how to use Node-RED and Python for that purpose. In the final step the neuronal network is fitted to the purpose of the factory. The course is also using a Python script to accomplish this. The next subchapter describes the application of the two used neuronal networks in detail.

#### **4.3.1 Applying the Neuronal Networks**

The university Albstadt-Sigmaringen is using two neuronal networks to detect the current state of the Lernfabrik during a typical ordering process. To make this possible it is necessary to preprocess all data in a way that a neuronal network is

applicable on it. How to preprocess the data in that manner and how to train and test the neuronal networks is described in the following.

#### 4.3.1.1 Data Preprocessing

To begin with the data preprocessing, an ordering process is launched in order for the Lernfabrik to provide real data for the neuronal networks. First of all, the data is acquired by receiving messages from each topic, which are the different stations of the learning factory. Each message comes with several information, such as the time stamp, the station, an acknowledge code, an active or inactive state and an optional description and target. To get a set of data to work with, the ordering process is conducted once and the raw data is written into a JSON file and saved. Therefore, this data can be used in all of the following steps and it is not necessary to repeat an ordering process more often. A sample message from the high-bay warehouse is shown in the following figure 4.13.

```
{  
  "topic": "f/i/state/hbw",  
  "message": {  
    "ts": "2022-04-06T18:43:06.453Z",  
    "station": "hbw",  
    "code": 1,  
    "description": "",  
    "active": false,  
    "target": ""  
  }  
}
```

Figure 4.13: Sample message [15]

The raw data is not sorted correctly corresponding to the time steps, so the first step for data preparation is to run a simple sorting algorithm. The reason for the data to be in the wrong order is that the distance between some recorded time stamps is less than a millisecond and during transmission they can get mixed up. Data is read from the JSON file and written into a list, then the time stamps are extracted from the messages and a list with two columns is created, where the time stamps are in the first column and the message in the second. Now the list is sorted by the time stamps, with the newest being on top, and the messages can be written in a new JSON file. An example is given in the figure 4.14 below.



23	1601-01-01T00:00:00.000Z	{'topic': 'f/i/alert', 'message': {'code': 0, ...
4	2022-04-06T13:30:16.819Z	{'topic': 'f/i/state/dso', 'message': {'ts': '...
1	2022-04-06T13:30:16.829Z	{'topic': 'f/i/state/sld', 'message': {'ts': '...
5	2022-04-06T13:30:16.829Z	{'topic': 'f/i/state/dsi', 'message': {'ts': '...
6	2022-04-06T13:30:17.069Z	{'topic': 'f/i/order', 'message': {'ts': '2022...
...	...	...

Figure 4.14: Sorted messages [15]

Now the raw data needs to be converted into a form that allows the models of the neuronal networks to work with it. The relevant data are the code, the active state and the target, which must be values from the same data type and range in order for the model to work efficient. Since the codes are either 0, 1 or 2 and the active state is either 0 or 1 it makes sense to also convert the target to small integer values. In this case 0 to 3 are used to represent the different targets. With that done there are only 13 inputs with small integer values left. The starting value of each input is set to minus 1 and updated whenever a new message is received, then the integer values are written into a new file. The initial state is reached as soon as there is a message from each station.

Table 4.2: Possible active state, code and target of each station

station	active	code	target
vacuum gripper robot	0, 1	1, 2	0, 1, 2, 3
sorting line with colour detection	0, 1	1, 2	
multi-processing station with oven	0, 1	1, 2	
high-bay warehouse	0, 1	1, 2	
input station	0, 1	0, 1	
output station	0, 1	0, 1	

The data is provided as in the following table 4.3. It can be observed that every value is -1 at the beginning and one station after the other gets updated to the initial state.

Table 4.3: Data

hbwstate	hbwcode	vgrstate	vgrcode	vgrtarget	mpostate	mpocode	sldstate	sldcode	dsistate	dsicode	dsistate	dsocode
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	1
-1	-1	-1	-1	-1	-1	-1	0	1	-1	-1	0	1
-1	-1	-1	-1	-1	-1	-1	0	1	0	1	0	1
-1	-1	-1	-1	-1	0	1	0	1	0	1	0	1
-1	-1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	0	1	0	1	0	1	0	1
0	1	0	1	0	0	1	0	1	0	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...

Next, every state that is possible during an ordering process has to be determined. This is achieved by sorting out all identical states, but only the consecutive ones, because one state can appear twice during the process. For example, the initial state (0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1) is found twice in that list because it occurs at the beginning as well as at the end of an ordering process. Next, all states that have a -1 in it are removed as they only occur before the stations are initialized. Then the remaining states have to be labelled manually, because otherwise the model would not know what they mean. This leads to the following data, where the state is added at the end of the values.

0,1,0,1,0,0,1,0,1,0,1,Resting  
0,1,0,2,0,0,1,0,1,0,1,StorageOut  
0,2,0,2,0,0,1,0,1,0,1,StorageOut  
1,2,0,2,0,0,1,0,1,0,1,StorageOut  
1,1,0,2,0,0,1,0,1,0,1,StorageOut  
1,1,0,2,0,0,2,0,1,0,1,Processing  
0,2,0,2,0,0,2,0,1,0,1,Processing  
0,2,1,2,1,0,2,0,1,0,1,TransportToMPO  
0,2,1,2,1,1,2,0,1,0,1,TransportToMPO  
0,2,0,2,0,1,2,0,1,0,1,Processing  
0,1,0,2,0,1,2,0,1,0,1,Processing  
0,1,0,1,0,1,2,0,1,0,1,Processing

0,1,0,1,0,0,2,0,1,0,1,0,1,Processing  
 0,1,0,1,0,0,2,1,2,0,1,0,1,Sorting  
 0,1,0,1,0,0,1,1,2,0,1,0,1,Sorting  
 0,1,0,2,0,0,1,1,2,0,1,0,1,Sorting  
 0,1,0,2,0,0,1,1,1,0,1,0,1,Transport  
 0,1,0,2,0,0,1,0,1,0,1,0,1,Transport  
 0,1,1,2,3,0,1,0,1,0,1,0,1,TransportToDSO  
 0,1,1,2,3,0,1,0,1,0,1,0,0,OutputStation  
 0,1,1,2,3,0,1,0,1,0,1,1,0,OutputStation  
 0,1,1,2,3,0,1,0,1,0,1,0,1,TransportToDSO  
 0,1,1,2,3,0,1,0,1,1,0,0,1,TransportToDSO  
 0,1,0,2,0,0,1,0,1,1,0,0,1,Transport  
 0,2,0,2,0,0,1,0,1,1,0,0,1,Transport  
 0,2,1,2,2,0,1,0,1,1,0,0,1,TransportToHBW  
 0,2,1,2,2,0,1,0,1,0,1,0,1,TransportToHBW  
 0,1,1,2,2,0,1,0,1,0,1,0,1,TransportToHBW  
 1,2,1,2,2,0,1,0,1,0,1,0,1,TransportToHBW  
 1,2,0,2,0,0,1,0,1,0,1,0,1,StorageIn  
 1,2,0,1,0,0,1,0,1,0,1,0,1,StorageIn  
 0,2,0,1,0,0,1,0,1,0,1,0,1,StorageIn  
 1,2,0,1,0,0,1,0,1,0,1,0,1,StorageIn  
 0,2,0,1,0,0,1,0,1,0,1,0,1,StorageIn  
 0,1,0,1,0,0,1,0,1,0,1,0,1,Resting

If two or more states are represented by the same data, they have to be summarized under one name. This happened twice in the previous labelling, so it has to be changed to a meaningful common name. Now there are 10 states left, which are easy to process in the model. The states are StorageOut, TransportToMPO, Processing, Sorting, OutputStation, Transport, TransportToDSO, TransportToHBW, StorageIn and Resting

#### 4.3.1.2 Create and Test the Models

When all the data preparation and preprocessing is done, they can finally be loaded into the model of the neuronal networks. At first the convolutional neuronal network (CNN) is used and then a recurrent neuronal network (RNN) is applied.

For the CNN the data has to be separated into input and output data. As input the 13 values of the stations are used. The 10 different output states, have to be encoded into an output array. First the names need to be encoded into integer values. The simple list then needs to be transferred into an array, where the number determines which position is a 1 and the other values are 0. For example, 'Resting' gets assigned with the number 9, which will be (0, 0, 0, 0, 0, 0, 0, 0, 0, 1). Figure 4.15 shows the final input and output array.

```
array([[0, 1, 0, ..., 1, 0, 1],
       [0, 1, 0, ..., 1, 0, 1],
       [0, 1, 0, ..., 1, 0, 1],
       ...,
       [0, 1, 0, ..., 1, 0, 1],
       [0, 1, 0, ..., 1, 0, 1],
       [0, 1, 0, ..., 1, 0, 1]])
```

```
array([[0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.]])
```

Figure 4.15: Input array (left) and output array (right) [15]

The following step is to set the definitions for the sequential model of the convolutional neuronal network. The model consists of 13 input neurons and 1 bias neuron that fires random values to ensure the model will not specify too much. A hidden layer with 10 neurons is added to the model as well as the 10 output neurons and the activation functions are defined. When the model is fully defined it has to be configured using a compile function, where the standard optimizer is chosen.

When training the model with the input and output data, the results shows that it gets better and better for each epoch. After 100 epochs the accuracy is >99% and the mean absolute error is <2%. After 200 epochs the accuracy reaches 100% and the mean absolute error is near 0% and from then on, the improvements are neglectable. When the model is now tested with sample input data it is expected that it always gives the correct result.

For the RNN the previous states are also put into consideration, leading to a faster and more precise result. The output data is the same as for the CNN, but the input

data has to be processed further as a 1D-array is not enough for the RNN. A multidimensional array with as many previous states as wanted is created. In this case only one previous state is used.

```
[[[0 1 0 ... 1 0 1]
  [0 1 0 ... 1 0 1]]

[[[0 1 0 ... 1 0 1]
  [0 1 0 ... 1 0 1]]

[[[0 1 0 ... 1 0 1]
  [0 1 0 ... 1 0 1]]
```

Figure 4.16: Multidimensional input array [15]

For model definition a different input layer called long short-term memory is required, which contains the 13 neurons, the dias neuron plus the previous states.

The model training follows the same procedure as for the CNN, but the results are worse this time, because after 500 epochs the accuracy is about 98% and the mean absolute error 2%. It seems like the RNN does not suit the application as well as the CNN does. When tested with the same sample data the results are also correct which indicates that the RNN is still good, but there are some issues with transitions between the output states.

## **CHAPTER 5**

### **CONCUSSION AND RECOMMENDATIONS**

#### **5.1 MES**

As for the end of this project the MES has been successfully connected to the learning factory and a dashboard that visualises the current state of the stations of the model has also been created. This is a significant improvement of the initial state of the learning factory, as data is now available to the MES and displayed on a dashboard, but it does not cover all the functionalities and capabilities of the MES. This chapter will discuss what can be done to enhance the usage of the MES in the future in order to get the most out of the learning factory.

When taking the guideline VDI 5600 into account, it can be seen that some of the points, like data acquisition and processing or information management are met. But still there is a lot more that a MES is capable of. At the moment only a dashboard that keeps track of the production process is used. Other features of Arcstone's MES arc.ops are not implemented yet. For example, orders can be tracked, jobs can be scheduled and viewed, workflows can be created or the personnel can be managed within the arc.ops software. As the dataflow from the automation level is established, these steps should be achievable in future projects with the learning factory.

The MES can also be seen as a data hub that connects all systems within an enterprise. By the end of this project the link to the MES is running by using arc.quire. The next step would be to connect the MES to an ERP, which can be done with arc.flow.

Moreover, there are improvements that can be done to the dashboard. Right now, it is designed to be easily understandable and to fulfil the purpose of showing the current

state in the most convenient way. Only the data of the sensors that are crucial to determine the state is used, but other sensor data may also be shown on a dashboard. With that it would be possible to show the production process in more detail or provide more information on the dashboard like the runtime of the stations.

The focus of this project was to show how to integrate a MES into a factory and how an enterprise can benefit from it. The dashboard is a good solution to demonstrate these points. Now it is up to others to take the usage of the MES to the next level.

## 5.2 AI Lernfabrik

This chapter concludes the project AI-Learning factory and gives recommendations. To analyse if the project was conducted successfully all requirements must be fulfilled. After the literature review the AI-Lernfabrik course was attended and the task were performed. Based on that the workshop a manual and task solutions were created. In appendix A the complete course can be found and the step-by-step solutions for the exercises are given in appendix B. Then the working principle was summarized to understand what one can learn from the course.

The remainder of this chapter will analyse what needs to be done in future to use the learning factory in full extent. It will be shown which hardware needs to be acquired, and what software is missing.

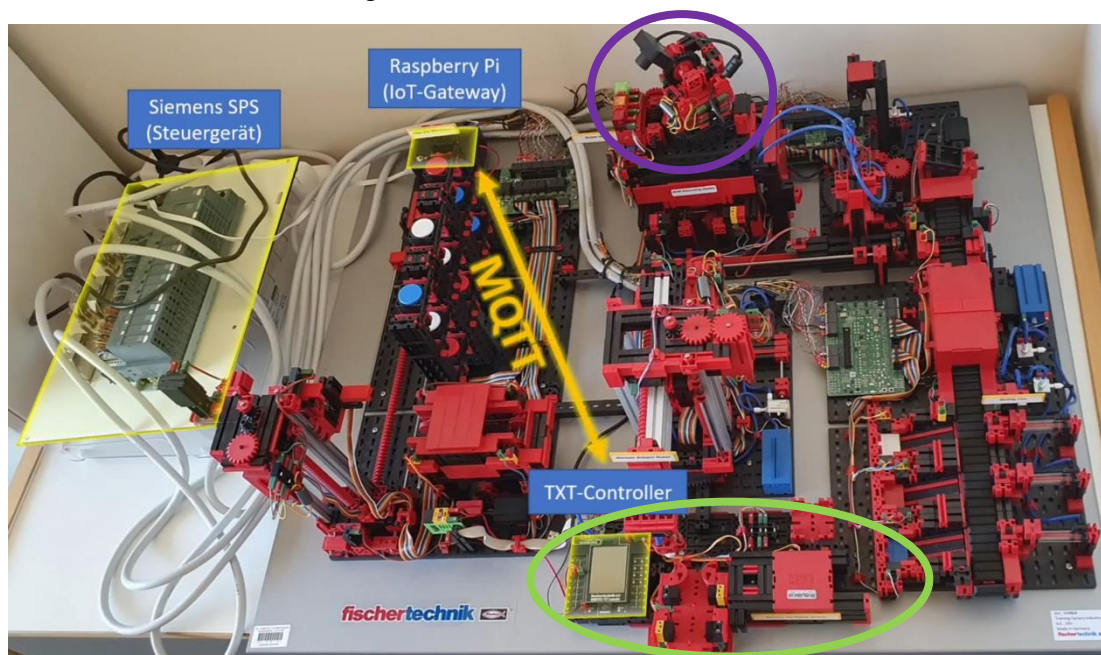


Figure 5.1: Controller overview [16]

The figure above shows the prototype of the factory. All the parts not highlighted can be found in the learning factory of the Innovation-Lab. The other elements are discussed in the following.

The purple highlighted parts are a combined sensor station. It is located on top of the oven, which is already part of the laboratory's factory. The biggest sensor of the station is a camera. It is capable of 360° filming. This is used for a web based remote control. The camera is not integrated in the software that controls the factory. It is meant for enabling persons far away from the factory to take a look. Environment sensors make up the other sensors of the station. The built in sensors are for measuring:

- Brightness
- Temperature
- Humidity
- Air quality

These values of the sensors get collected and analysed. They help to monitor the factory. With them, one can make sure that the production process is always within limits.

Highlighted by the green circle is the delivery station. The whole station is not present in the Innovationslab. It is used for incoming as well as outgoing goods. It consists of several elements. At the end of the production the gripper puts the workpiece into the delivery space. A light barrier is activated hereby. This information is collected by the controlling script. The information is then processed with all the other station data through the preparation steps to analyse it with the neuronal network. The missing information will be seen as an interruption to the production and a danger of a hang-up of the software.

In the following step the gripper puts the workpiece onto a colour sensor. It detects the colour of the product and this information is also stored. As soon as the workpiece is back into the storage, the software knows which colour it has. During ordering a customer can select the wished colour. The software is able to get a matching product.

In the final step of the delivery station a NFC reader and writer tags the workpiece with the production information. This is important that the workpiece "knows" what it is. Thus, this step is used as a simulation to real world factories, the NFC chip of



the workpiece is not read again. Meaning it is a nice to have feature for the factory simulation, but not necessary for the neuronal network.

Attached to the delivery station is the Fischertechnik txt controller. Its purpose is to collect all sensor information. After the data collection it is send to the IoT-Gateway via MQTT standard. There is no specification which chip has been used. There is also no information about the used software. This makes it hardly possible to rebuild this element.

The IoT-Gateway receives the data from the txt controller. The hardware used is a Raspberry PI, which has two purposes. The first purpose is to control the ordering process. The utilised software is the python script that was introduced in the course. The second purpose is to transmit the factory data to the cloud. This is handled with the OPCUA protocol. In the cloud the source code for the neuronal network is running. So, this connection is crucial to hand the production information to the neuronal network.

As important hardware is missing, the course of university Albstadt-Sigmaringen can not be applied to the learning factory in the Innovationslab. But the major takeaways are pointed out and the accumulated knowledge of the course can be used to create own neuronal networks that supervises the factory's state.

## REFERENCES

1. Ministry for Economy, Labour and Tourism (2023). *Technology and Knowledge Transfer*. Retrieved on June 30, 2023, from <https://www.wirtschaft-digital-bw.de>
2. Mercedes Benz Group (2023). *Industry 4.0 and the networked factory*. Retrieved on May 20, 2023, from <https://group.mercedes-benz.com>
3. Kletti, J.. *MES – Manufacturing Execution System*. 2<sup>nd</sup> Ed. Mosbach. Springer Vieweg. 2015
4. Verein Deutscher Ingenieure. *Manufacturing execution systems (MES)*. Düsseldorf. VDI 5600. 2016
5. International Society of Automation. *Enterprise-Control System Integration*. Research Triangle Park. ISA95. 2014
6. Arcstone (2022). *Unlocking the Value of Data*. Retrieved on April 30, 2023, from <https://www.arcstone.co>
7. Luger G. F. & Stubblefield W. A.. *Artificial intelligence*. 2<sup>nd</sup> Ed. Albuquerque. Benjamin/Cummings Pub. Co.. 1993
8. Hebb, D. O.. *The Organization of Behavior*. 1<sup>st</sup> Ed. New York. Psychology Press. 2002
9. Picton, P.. *Introduction to Neural Networks*. 1<sup>st</sup> Ed. London. Red Globe Press London. 1994
10. Ertel, W.. *Grundkurs Künstliche Inelligenz*. 5<sup>th</sup> Ed. Weingarten. Springer. 2021
11. Grum, M.. *Construction of a Concept of Neuronal Modeling*. 1<sup>st</sup> Ed. Berlin. Springer. 2022
12. Sonnet, D.. *Neuronale Netzwerke kompakt*. 1<sup>st</sup> Ed. Hamburg. Springer Vieweg. 2022
13. Fischertechnik (2023). *Lernfabrik 4.0*. Retrieved on May 5, 2023, from <https://www.fischertechnik.de>
14. Beckhoff (2023). *TF6100 / TwinCAT 3 OPC UA*. Retrieved on May 20, 2023, from <https://infosys.beckhoff.com>

15. KI Campus (2022). *Lernfabrik 4.0 – Steuerung, Monitoring und NN-Modell*. Retrieved on June 25, 2023, from <https://learn.ki-campus.org>
16. Klein, M. (2022). *Osteraktion 12 Weiterbildung*. Retrieved on May 26, 2023, from <https://www.hs-albsig.de>