Universidade de Aveiro
Ano 2021

Departamento de Engenharia Mecânica

Diogo Mário Matos dos Santos

MES Digital Twin

MES Gémeo Digital

Universidade de Aveiro
Ano 2021

Departamento de Engenharia Mecânica

Diogo Mário Matos dos Santos

MES Digital Twin

Relatório de Projeto apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro, e do Engenheiro Joaquim Ribeiro, Engenheiro da Bosch Termotecnologia S.A.

The present study was also developed in the scope of the Project Augmented Humanity [POCI-01-0247-FEDER-046103], financed by Portugal 2020, under the Competitiveness and Internationalization Operational Program, the Lisbon Regional Operational Program, and by the European Regional Development Fund.

Dedico este trabalho à minha família, com especial ênfase para o meu pai e a minha mãe que sempre me apoiaram para que todo este percurso fosse possível.
Uma dedicação especial também para todos aqueles que nunca desistiram e que todos os dias encontram motivação para seguir este percurso notável e acima de tudo recompensante.

o júri

presidente

Prof. Doutor Marco Paulo Soares dos Santos
Professor Auxiliar da Universidade de Aveiro

Prof. Doutora Ana Maria Pinto de Moura (Arguente)
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor José Paulo Oliveira Santos (Orientador)
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Um enorme agradecimento à minha família, em especial pais e irmã por todo o acompanhamento e por sempre acreditarem em mim ao longo desta jornada que certamente foi marcante e sobretudo importante na minha vida. Queria dizer um obrigado também a todos os colegas, amigos e namorada que permitiram que todo este percurso fosse mais agradável, através do companheirismo, lealdade, entreajuda e diversão.

Ao Professor José Paulo de Oliveira Santos, por todo o acompanhamento e ajuda durante este projeto.

À empresa Bosch Termotecnologia S.A por todo o apoio e pela disponibilização de Engenheiros profissionais que em tudo contribuíram para o bom desenvolvimento de todo este projeto. Em especial aos Engenheiros Joaquim Ribeiro e Pedro Rendeiro.

Agradecimento também à instituição, Universidade de Aveiro por ter permitido que tudo isto fosse possível e que procura todos os dias proporcionar as condições para que possamos ter um futuro melhor.

palavras-chave

MES (*Manufacturing Execution Systems*), *Digital Twin*, Indústria 4.0, IoT, Base de dados, Softwares de Simulação do Chão de Fábrica, JSON, Tecnomatix, KPI

resumo

Na atualidade, cada vez mais se procura a necessidade de aplicar as ferramentas da indústria 4.0, procurando essencialmente aumentar a sintonia entre os trabalhadores, máquinas e processos, originando assim o conceito de *Smart Factory*.

Nesta vertente situa-se a digitalização do chão de fábrica, originando o objetivo para este projeto, onde se pretende realizar a simulação de uma linha de produção no ambiente simulado do *Plant Simulation Tecnomatix* e integrar com o *Manufacturing Execution System* (MES) da empresa Bosch, Nexeed MES.

Desta forma, será possível aceder de antemão à linha de produção neste ambiente simulado, permitindo monitorização em tempo real e simular eventuais alterações e melhorias, eliminando custo e tempo excessivo despendido desnecessariamente.

Os MES são um exemplo de solução que a evolução da indústria tem vindo a proporcionar para as empresas. Eles permitem a monitorização e execução de ordens de fabrico em ambiente fabril, a ponte que proporciona a ligação entre o chão de fábrica e a gestão da empresa (ERP).

O *Plant Simulation Tecnomatix* é um software desenvolvido pela Siemens PLM Software que permite a criação de modelos simulados de uma linha de produção ou até mesmo de uma empresa na sua totalidade.

Neste relatório de projeto, foi desenvolvida uma solução que permite a criação de um *Digital Twin* através da integração de uma simulação do MES da empresa Bosch, Nexeed MES, com a simulação da linha de produção um, da Bosch, no *Plant Simulation Tecnomatix* da Siemens.

keywords MES (Manufacturing Execution Systems), Digital Twin, Industry 4.0, IoT, Database, Shop Floor Simulation Software's, JSON, Tecnomatix, KPI

abstract

Nowadays, the need to apply the tools of industry 4.0 is increasingly sought, essentially seeking to increase the harmony between workers, machines, and processes, bringing up the concept of Smart Factory.

Here lies the digitalization of the shop floor, originating the objective for this project, where it is intended to carry out the simulation of a production line in the simulated environment of Plant Simulation Tecnomatix and integrate with the Manufacturing Execution System (MES) of the company Bosch, Nexeed MES.

Therefore, it will be possible to access the production line in advance in this simulated environment, allowing real-time monitoring and simulating any changes and improvements, eliminating costs and excessive time spent unnecessarily.

MES are an example of a solution that the evolving industry has been providing for companies. They allow the execution of manufacturing orders in a factory environment, the bridge that connects the shop floor with the enterprise Level (ERP).

Tecnomatix introduced by Siemens PLM software, allows the creation of simulation models of a production line or even a company in its entirety.

In this project report, a solution was developed that allows the creation of a Digital Twin through the integration of a simulation of the Bosch company MES, Nexeed MES, with a simulation of the production line one of Bosch, in Plant Simulation Tecnomatix environment from Siemens.

Content

List of Tables

List of figures

List of Acronyms

AMQP - Advanced Message Queue Protocol

CPS – Cyber Physical Systems

DB – Data base

DBMS – Database Management System

DT – Digital Twin

ERP – Enterprise Resource Planning

HMI – Human Machine Interface

I4.0 - Industry 4.0

IoT – Internet of Things

KPIs – Key Performance Indicators

MES – Manufacturing Execution System

MOM – Manufacturing Operations Management

MQTT - Message Queuing Telemetry Transport

PLM – Plant Lifecycle Management

SPI – Serial Peripheral interface

SQL – Structured Query Language

SAP - System Analysis Program Development

VMs – Virtual Machines

1. Introduction

In this chapter, it will be presented the problem that led to the writing of this dissertation. First, a brief introduction is done to contextualize the problem, bringing then a close attention to Bosch, to then have a better perspective of what are the objectives, goals and motivations that are acting as a support. Lastly, the structure of this document will be presented and explained.
In the past years, companies have been progressing in a way where they seek to work not faster and harder, but instead smarter. This is where Industry 4.0 situates, a continuous evolution in the industry that aims to find ways to increasingly connect the workers with machines and processes [1].
This was the mindset that worked as a motivation for the development of this project, and most importantly, to build and create something that will elevate and evolve the company in a way where both the company and its collaborators will be able to make more with less effort, money, and time. The goal is to create a production line simulation model that will connect to the manufacturing execution system (MES) that is currently used in Bosch, Nexeed MES, originating the intended Digital Twin. With this implementation, the simulation will be using concrete data of the production line opposed to the theorical or simulated data that is often recured and does not allow the same level of trustworthiness and reflection of the real production line. At the end, it is expected to have a simulated model of a specific production line, stations and processes that will be useful for real time monitoring in a more compact and accurate way, but most importantly, it will create the capability of testing any change in the simulated model, resulting in a better analysis and cost-free trial and error.
Furthermore, while the process of digitalization of a production line usually requires considerable time as all the data needs to be analyzed individually and all the insertion of that data in the simulation model needs to be performed manually, this solution allows a significant reduce in the required time as the data from the production line is being analyzed automatically and the simulation is being updated accordingly.
The most common technologies of the industry 4.0 that characterize CPSs (such Data and Cloud computing as support systems to read big sets of data from the field, store and analyze them and Internet of Things (IoT) to remain connected and extract data) are also the basis for a new simulation approach, which increases immensely the connectivity in production systems to offer a real-time synchronization with the field. This new simulation approach is generally referring to the elaboration of Digital Twins (DT).
It is proposed a Digital Twin of a Bosch production line, line one, integrated with the Nexeed MES, enabling real-time data integration so that, through a computer, it becomes possible to fully interact with the production line.

1.1. Project Goal

After implementing the solution, it is expected to have a functional Digital Twin, fully integrated with the MES. Having this in mind, it is expected that Bosch's Engineers will be able to successfully implement any change or improvement in the Digital Twin, before trying it in a real scenario, preventing from spending money and time unnecessarily. While this information can be currently accessed on Nexeed MES or stored in Excel Sheets, the possibilities of working with it and actually using this information for analysis and testing purposes are limited, as will be explained in a study case. By implementing a simulation on the shop floor simulation software with fully integrated real time data from MES, it opens numerous possibilities for visualization, testing and improvements with concrete and real results that correspond to the real production line status. Various features are developed and explained with emphasis for their applicability and a later comparison to the actual or rather manual methods that are being used at the moment, in order to understand the advantages that these improvements bring to the simulation field in the company.

In order to achieve this goal, it is necessary to build a solid base of knowledge and essential fundaments that are crucial for the development of the solution. The first approach will be studying the concepts for better understanding of the problem and what characteristics are expected to have an integration with this similarity. Along with this, various software's will be explored allowing the creation of the Digital Twin. Also, very important, the understanding of the Bosch production line models and the way they are structured and organized, because only this way it will be possible to achieve an accurate and flexible Digital Twin.

Lastly, it is expected the proposal and implementation of a solution with a grounded architecture that will answer all the expectations and fulfill the needs and requirements. In summary:

- Build a solid ground of fundamental concepts and complete understanding of the problem and the tools required.
- Study of the Bosch production line structure and data model and acknowledgement of the requirements and features that are expected to be implemented in the Digital Twin.
- Proposal of the solution, relying on a clear and reasoned architecture, and most importantly, that meets all the expectation made by the author and the company.
- Implementation of the solution with detailed explanation.
- Development analysis of the implemented solution based on performance tests and analysis of the results.

1.2. Project report structure

This project report will be divided into six chapters. Each of the chapters will be explained to help the contextualization of the document. The first chapter is the introduction of this document. It explains the problem and the contextualization, the goals, and motivations along with a brief introduction to some of the topics that will be studied, in order to help the understanding of the problem.

Chapter two presents the study and investigation made regarding the topics and software's that will be used and explored in this project, while giving some perspective on other options that are not part of the implemented solution. The third chapter is the presentation of the company, Bosch Termotecnologia S.A., along with an explanation and analysis of the production line model that will serve as a base for the creation of the digital twin in Tecnomatix.

In the next chapter, the fourth, a proposed solution is presented for the integration of the MES with the simulation model created with the help of Plant Simulation Tecnomatix. Chapter five is the implementation of the solution. In this chapter, it will be explained the application and integration of the solution described in the previous chapter.

Chapter six is case study analysis in Plant Simulation Tecnomatix to understand its potentiality in production line simulation and explaining the advantages between the previous approach and the new approach that is possible due to the implementation in this project.

Lastly, Chapter seven presenting the conclusions of the project report, describing what aspects did not meet the expectations and the reasons, finishing with the presentation of possible optimizations and future work.

2. State of Art and fundamental concepts

2.1. Industry 4.0

2.1.1. Automation hierarchy

As the industry evolves, there is an inherent increase of complexity and with this, lies the necessity of a hierarchical structure to organize and create a logical flow of information and data. Even though CPS, capable of merging the physical and virtual world by embedded software and hardware, present a nonhierarchical approach in production as explained in the section below, there is still a hierarchical pyramid for automation that is a systematic and centralized solution for the production systems. Some variations of this pyramid can be found but generally, it is composed of five layers or levels, as shown in Figure 2.1.

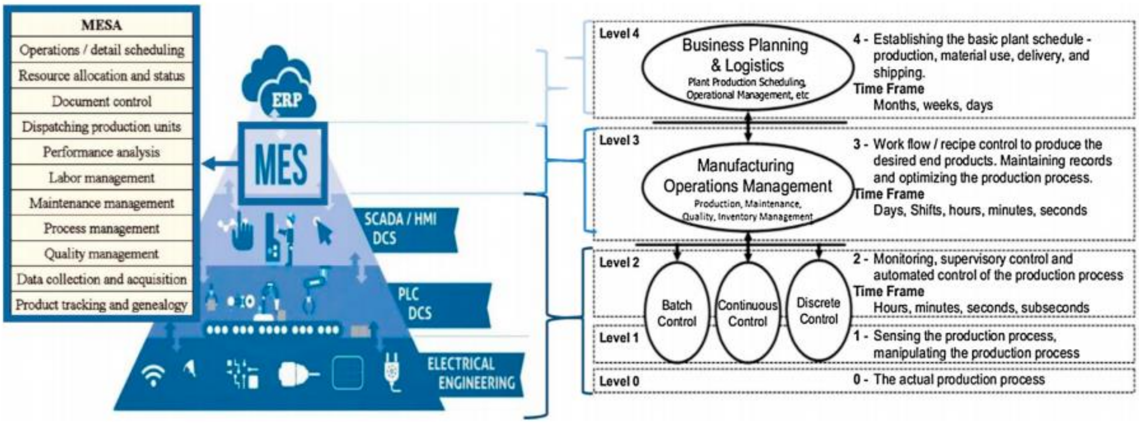

Figure 2.1 – Automation Hierarchy [2]

As in any hierarchy pyramid, the power increases as each level is climbed. For a better understanding of the whole power distribution, each layer is explained, with reference to what is the purpose and the objectives of each level. Overall, the first three layers represent the control levels, and they include PLCS (Programming Logic Controllers), DCS (Distributed Control Systems), electrical engineering layer and HMI/Scada layers [2].

I. The first level, zero, is the lowest in the hierarchy pyramid and is the level where occurs the value-creating production or manufacturing processes.
II. This, level one, is where the highest energy consumption occurs. While the previous level includes the machines, everything inherent to it as for example sensors, devices and actuator are placed on the next level, the level one.

III. The level two is responsible for all the monitoring, supervision, and control. Overall, most of the productions systems use PLCs, and they are used to control the actuators and sensors that are located on the inferior level, level one.
IV. Layer three is the manufacturing executing system (MES). MES is intended to guide the process and its where the Manufacturing Operations and Control (MO&C) such as management of work sequences, productions receipts and part lists, are placed.
V. The top layer, the fourth, is characterized by integrating organizational functions for better customer support and planning. It is often characterized as Enterprise Resource Planning (ERP), and it can be summarized as business planning and logistics. When compared to the previous level, ERP can be said to be more of a farther planning method, operating at weeks or even months, including long term production and/or energy usage [3].

2.1.2. Cyber Physical Systems (CPS)

CPS are generally perceived as the bridge that connects the humans with the machines allowing increased communication and interaction [4], and they are capable of elaborate and communicate real-time data and build a copy of the processes in a digital approach [2].
CPS are characterized by real-time capacity to detect and react to changes in the physical processes, and capacity to recognize and sense pertinent events [5].
As CPS are evolving, with the increasing capacity of merging the digital and physical side, a new architecture is developed. Maximizing the capabilities of the CPS, the automation hierarchy can be decomposed into a more flexible and knowledge-based approach. This means that it is possible to engage all the five levels from any data source of CPS as show in Figure 2.2 [5].

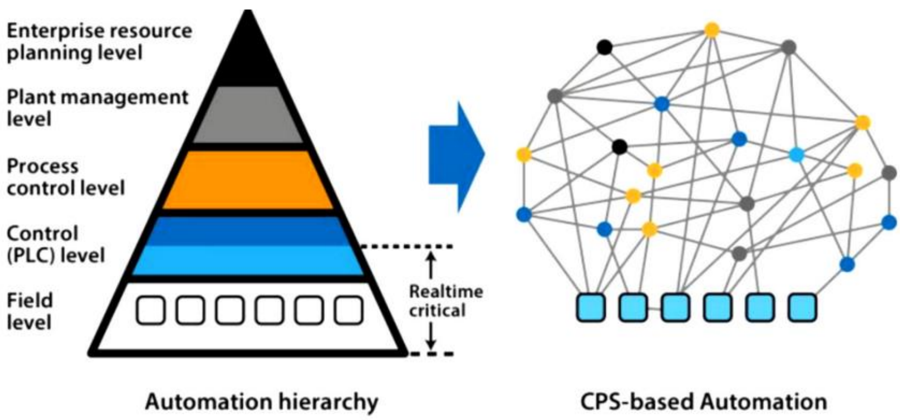

Figure 2.2 – CPS in Automation Hierarchy [5]

2.2. Manufacturing Execution Systems (MES)

In an ongoing evolution of the industry that we live in today, industry 4.0, where it is perceived by its high requirements of automation, smart products and resources, sensors, internet applications with real time capacities in exchanging data, there is an immense concern to continuously seek for ways and solutions to possibly bring something new to add value and quality to the companies [6].

Manufacturing Execution Systems (MES) are a concrete example of an industrial software that fits perfectly in the industry 4.0 paradigm. They allow controlling, and monitoring operations and production, and assisting people and machines in the execution of tasks through the concept of digital manufacturing [7].

MES are crucial for vertical integration in modern automation systems as they provide a connection between the enterprise level and the shop floor. Essentially, with MES, it becomes possible to build the bridge that will result in the perfect synchrony of the decision making at the enterprise level with the shopping floor, thus making the manufacturing system more accurate and reliable [8].

According to International electro technical commission (IEC), Manufacturing operations management (MOM) define the activities of a company that coordinate, manage, and evaluate the resources such as equipment, materials, and workers. The panoply of operations is vast and can be production data collection, maintenance, manage production, etc. MES play a huge role in the achievement and quality of those operations, as they work as an information hub, allowing real time data [7][9].

2.2.1. MES vs ERP

Whether to use MES, use ERP or use both as a complement to each other is a question to the majority of companies when doing a MOM. There are several reasons that must be considered when taking this decision because acquiring these tools require great investment, so it is important to understand when and where each of them is better applicable and necessary [10].

Generally, ERP is defined by its macroscale vision, while MES has more of a microscale vision. Nowadays, ERP systems cover a wide range of modules, which allow numerous operations such as finance, logistics, transportation management, human resources, material use, shipping, customer relationship management [10].

So, what is stopping ERP from matching the MES systems and become totally independent? The major advantage of the MES systems comes when there is a high variability of production processes and local workflows, usually in bigger size companies. The reason is that ERP usually tends to cover most of the business and when the company has too much variety of processes, the scope of the ERP tends not to reach the more specific matters, like supporting local procedures or handling continual changes. This is where MES are mostly necessary, as they are not concerned mainly about storing data, but instead they are mostly about increasing the throughput and efficiency of the production [10]. They are able to do this by showing real-time data, generate logs and notifications for problems and most importantly, involving the job shop workers and

production team managers which are the key for the development and improvement of MES and production itself [11]. and specially in lower time frames than ERP.
Additionally, MES usually operate in considerably lower time frames that ERP system, which often operate between day, weeks or even months. Without MES system, there is a high probability of occurring errors in data insertion as it is usually inserted at the end of the day, and in the positive scenario of no error, there is still the problem of not having real-time and quick data available to handle immediate customization and flexibility in production [11][12][13]. Following, some manufacturing execution systems will be presented (Table 2.1) to help understanding the reason why several big companies are opting for them in order to increase productivity and reduce costs.

Table 2.1 – Example of most used MES systems

Manufacturing Execution System	**Developer**
Nexeed MES	Bosch
Hydra MES	MPDV
SAP MES	SAP
Siemens MES	Siemens

2.2.2. Nexeed MES

There are numerous MES that are currently used throughout different companies, and generally they are acquired from different producers, as generally companies do not have their own MES. However, Bosch built its own vertical integrated MES, the Nexeed (Figure 2.3), covering every level since the sensor and machines until the cloud, and granting the integration with the ERP systems. This is the software used in this project to integrate with the Plant Simulation Tecnomatix.

Figure 2.3 – Nexeed MES used in Bosch. [14]

With Nexeed MES, Bosch intends to guarantee an improved and successful planning, control and optimization of production by creating the bridge between real-time data and the software[15][16].
It features a great number of operations with integration on the most important fields of application, such as operating data, operating equipment management, material

management, planning and process control, information management, quality management & traceability and shopfloor integration.
The benefits of Nexeed MES are notorious, increasing productivity and production performance, up to 15% higher machine availability through predictive maintenance, full transparency between machines, processes and data, short response time due to alarms, possibility of performing a lot of operations remotely, etc [14].

2.2.3. Hydra MES

HYDRA® is a leading, future, and smart-oriented manufacturing execution system developed by the MPDV Group. This group originally started in 1977 and is now a leading provider and supporter of the hydra manufacturing execution system used by several companies, counting with one thousand and four hundred installations and approximately nine hundred thousand daily workers making use of this solution nowadays [17].
This MES is marked by its vertical integration between production and management levels. HYDRA is built of several Manufacturing Apps (mApps) which can be combined with between themselves to build a highly flexible and customized solution for each user [18].

2.2.4. SAP MES

SAP is worldwide German company that was founded in 1972 by five entrepreneurs, focusing on the development of software to support company management. SAP has solutions that cover a wide range since different company sizes to several different areas of applications. Some of its software's are SAP PP (Production Planning), SAP FI (Financial Accounting), SAP QM (Quality Management), SAP MES (Manufacturing Execution System), SAP PM (Plant Maintenance) [19].
SAP MES is used by great industries such as Continental with great feedback on how they are succeeding and improving their productivity, efficiency, and high-quality production with this solution. Some of its features include supplying employees with intuitive interface to support shop floor manufacturing, great scalability possibility provided by the easy and straightforward integration with MES ERP to cover every level of industry [20].

2.2.5. Siemens MES

Siemens PLM developed several products for their portfolio of MOM (Manufacturing Operations Management) aiming to increase the digitalization of industry and among them the Siemens manufacturing execution system.
This software helps with increasing the efficiency and quality of manufacturing and ensures that the harmony between machines and workers is established with maximum optimization. Some of its features include quality operations management, resource allocation and control, process management, production tracking, etc [21].

2.3. Shopfloor Simulation Software's

Economy, nowadays, is growing immensely in a way that is leading to increasing of products complexity, increasing of quality demands, shorter life cycles, competitive pressure, etc. With the necessity of keeping up with all these demands, simulation has been taking place where simpler methods can no longer keep up and provide useful and quick enough results [22].
Although, the outcome of increased quality, less labor and more throughput which are expected have not been reached in many cases. This happens due to the complexity involved in scheduling and controlling manufacturing activities in a dynamic environment on the shop floor. Therefore, the manufacturing planning, scheduling, and control systems should be able to be dynamically adapted to the changing requirements and unforeseen situations in the shop floor.
Simulation is often recurred to get insight into manufacturing systems and is based on the interaction of a series of processes combined with statistical information. While for many years simulation models were being used to design, analyze, and plan changes at long-term and later disposed, nowadays the approach is taking a different direction, aiming for the integration with MES systems in order to provide real-time monitorization and solution evaluation.
MES which are responsible for collecting real-time data from the production lines and execute optimized tasks to increase manufacturing efficiency are the perfect solution for integration in simulation models due to their direct and close approach to the production environment. Some researchers have already referred to information transfer between the simulation models and MES. For instance, Kouiss and Najid presented a combination of a MES and an on-line simulation model for controlling a flexible manufacturing system. Lee and Chan developed an object-oriented simulation for MES. However the relationship between them is yet to be fully discussed [23]. In Table 2.2 there is a list of some of the commonly used software's for shop floor simulation.

Table 2.2 – Different Shop Floor Simulation Software's

Shop Floor Software	Characteristics	Price	Vendor	Complexity
Simio Simio Forward Thinking	- 3D object modelling. - No programming required. - Data integration [24].	No information	Simio	Medium
Simul8 SIMUL8	- Rapid data connections. - Instant process mined models. - Connection to	330€ / month	Simul8	Medium

	live data sources. - APIs. - Scripts enabling coding to control objects [25].			
Anylogic	- Linking GIS and simulation. - Support 2D and 3D animation. - Easy to use libraries [26].	Free	Anylogic	High
Plant Simulation Tecnomatix	- Object-oriented modelling. - Easy to use libraries. - Connection to external sources including databases [27].	500€/month	Siemens	High
Ansys	- Embebed software integration. - Third-party tool integration. - Rapid HMI prototyping [28].	No information	Ansys	High

2.3.1 Tecnomatix Plant simulation

Tecnomatix Plant Simulation is a computer software develop by Siemens PLM Software. It is built for modelling, simulation and exploration and optimization of logistic systems as well as their processes. This software offers numerous advantages when it comes to anticipating the outcome of a production line. These models allow the analysis of materials flow, resource usage and the logistic in all the different levels of the manufacturing system [29].

Combining these tools, plant simulation is powerful software for evaluating individual production plans as well as for comparing different manufacturing productions lines and options, relying on the study of layouts, process control logic, previewing dimensions, analyzing costs, anticipating constraints, etc. [30].

2.3.2. Tecnomatix Plant Simulation applications

Plant Simulation models are utilized to optimize the performance, remove bottlenecks, and minimize the work in process. These models consider internal and external supply chains, production, and business processes, allowing the users to analyze the impact of the various outcomes in production. It is possible to generate extensive statistics and charts that facilitate a dynamic analysis of the performance parameters, including excessive workload on lines, malfunctions, and excessive maintenance times [31].
As a computer software, Tecnomatix Plant Simulation allows great flexibility when it comes to changes, adaptations, and improvements. It is commonly used for the planning phase and analyzing the best options, though the extend of this software is endless [22].

2.3.2.1. Planning phase

- Study and evaluation of planning alternatives.
- Test of capacity, effectiveness, placements.
- Identification of unused potential.

2.3.2.2. Implementation phase

- Performance tests.
- Analysis of problems and future improvements.
- Verification of potential accidents and risks.
- Training new employees in anticipating these events.

2.3.2.3. Operation phase

- Quality assurance and failure management through data analysis.
- Testing of control alternatives.
- Review of emergency strategies and accident programs.

2.3.3. Model visualization

Along with the fact that Plant Simulation's 2D models are highly efficient and useful, the models can be analyzed in a 3D virtual environment, using included libraries or user-assisted computer-aided design (CAD) data. The outcome is impressive 3D virtual models that are always synchronized with their 2D counterparts, allowing customers the flexibility to choose the appropriate visualization method, without compromising the needs of simulation and analysis. Plant Simulation supports the JT data format for 3D modeling, a standard of the International Organization for Standardization (ISO), and direct model technology from Siemens PLM software, which will enable efficient loading and realistic visualization of large 3D simulation models [31].

2.3.4. Characteristics and advantages

Tecnomatix is characterized for its object-oriented hierarchy modelling. Advanced and complex models can be built on various logical layers that are organized by relevance respecting the hierarchy. It uses libraries of objects dedicated to fast and efficient modelling of processes. These objects can be divided in classes and result in a better organizations and transparency of the model.

As for a deep and meticulous evaluation, Tecnomatix includes various models of charts and diagrams, Sankey diagram for monitoring the material flow and Gantt charts for illustrating the evolution of the different steps of a project for instance.

Adding to these, the options for communicating with external sources are also a possibility. Program Access, Oracle, OBDC, SQLite, Excel, SAP, XML, ActiveX are some of the options that can be integrated in the model to allow data transfer and storage [30] [31].

Combining these immense set of tools, Tecnomatix comes with a panoply of benefits that transforms it into a must have for companies that want to take a next step and become more prepared to face the evolution of the industry. Increase in the productivity, reduction of stock and productions times, minimizing the investment on planning new installations, optimizing dimensions of the system, reductions of risks of investment, maximizing the usage of the resources available, live monitoring of all the production line with real-time data, ease on the detecting of possible changes and improvements, these are some of the benefits that a simulation model brings [31].

2.4. Digital Twin

2.4.1. Digital Twin overview

Before the so-called Digital Twins as we see them nowadays, the concept of "twins" was already being explored by NASA in the Apollo program. Using two identical space vehicles, NASA pretended to mirror the condition of the mission on the space vehicle, being the earth vehicle the twin [32].

The term Digital Twin was introduced to the general public for the first time in NASA's integrated technology roadmap under the technology area 11: Modeling, Simulation, Information Technology & Processing (NASA Technology Roadmap, 2010 and 2012). It was outlined as "A Digital Twin is an integrated Multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin"[33].

While in the last decades, simulation was restricted to computer and numeric experts, in the present times, it is a tool accessible to all the engineers that are interested in implementing it and are looking for ways to answer the industry demand. This resulted in the concept of DT to be the new wave in modelling, simulation, and optimization technology. “Communication by simulation” is the new evolution of engineering and is marked as the core concept of model-based systems engineering (MBSE). The evolution of simulation is shown in Figure 2.4 [34].

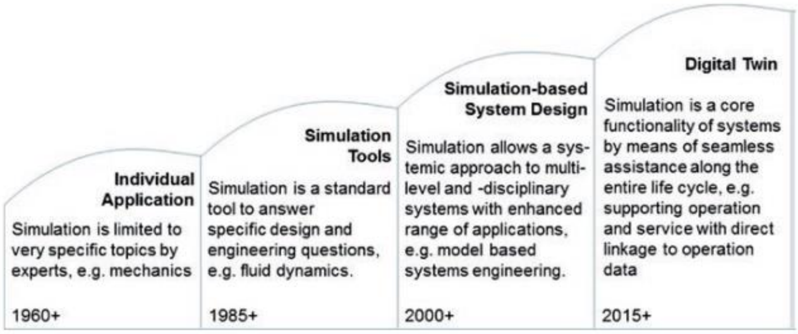

Figure 2.4 – Digital Twin evolution [34]

2.4.2. Digital Twin definition

Digital twin (DT) can be described as a virtual representation of the system that can interact with the physical entities in a bi-directional way [2], a system capable of replicate, control and directly interact with the physical world.
Generally perceived as the “cyber” part of the CPS [5], DTs are a useful way of boosting the merging of the physical and digital world inside the CPS and “*exploits sensed data, mathematical models and real-time data elaboration in order to forecast and optimize the behavior of the production system at each life cycle phase, in real time*” [5]
This representation is a promising replication of the production in real time. By analyzing Figure 2.5 it is possible to observe the fields of integration of the DT and it should be able to guarantee well-defined services to support various activities such as monitoring, maintenance, management, optimization, and security [2].
Due to the evolution of simulation techniques, a DT is not only available of providing a physical device integration and simulation, but also include sensor-based data gathering and algorithms in order to ensure that experiments can be performed in various environments [5].

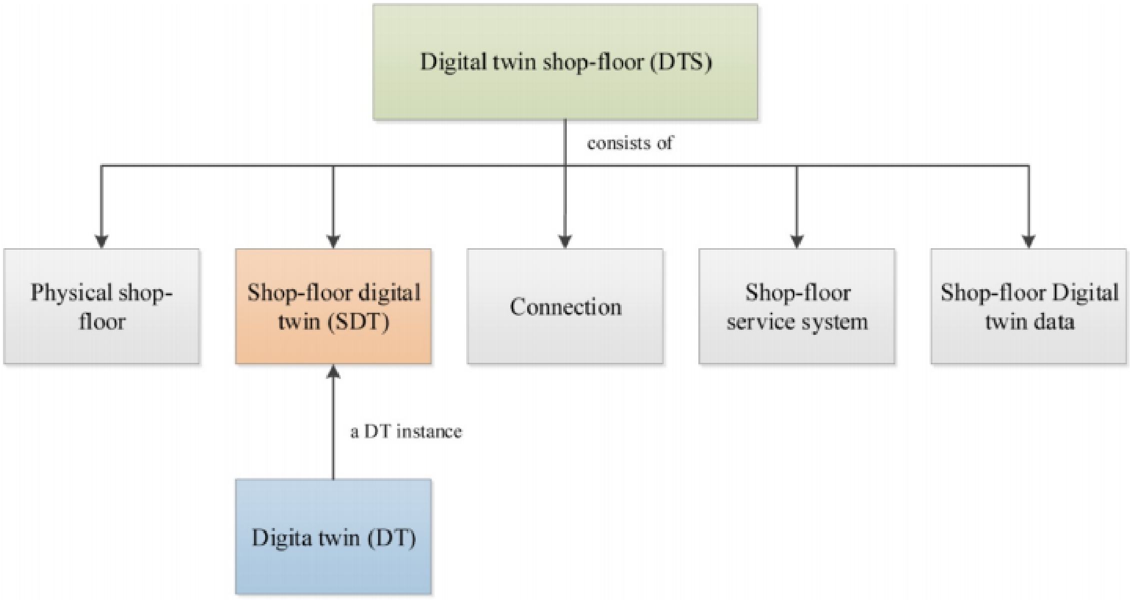

Figure 2.5 – Digital Twin overview [33]

They are often utilized to provide particular analysis, related to the considered system and to its lifecycle, having in mind what services the DT might offer. Generally, it is not necessary to use all of offered services, though their existence can be useful for industrial decision makers, sparing them from making different analysis in separate digital environments. These services can be organized into these categories [2]:

- Real-time monitoring in order to keep the virtual twin updated.
- Energy consumption analysis.
- Predictive maintenance that uses as backup historical data to anticipate and prepare a maintenance plan.
- Smart update and optimization based on analyzing user operation habits and product behaviors data.
- Behavior analysis and user operation guide used to obtain the operations done from the users and provide with user guidance to check up system updates through a user-friendly HMI (Human-Machine Interface).
- Product virtual maintenance and operations; With a 3D environment or software, operations or maintenance strategies to users can be provided.

DT has been growing in popularity the last years, and large corporations like ABB, Siemens, General Electric, Equinor and IBM have found ways and solutions to approach this matter. Below are listed some definitions given by these companies [32].

- **General Electric** – DT is defined as a software representation of physical assets, system and processes seeking to predict, detect, and optimize performance resulting in improved business outcomes. Digital twins consist of three components: a data model, a set of analytics or algorithms, and knowledge [35].

- **Siemens** - DT is a virtual representation of a physical product or process, used to understand and predict the physical counterpart's performance characteristics. By incorporating multi-physics simulation, data analytics, and machine learning capabilities, digital twins simulate the impact of design changes, usage scenarios, environmental conditions, and other endless variables. DT's are used throughout the product lifecycle to simulate, predict, and optimize the product and production system before investing in physical prototypes and assets [36].

- **ABB** - A digital twin is a complete and operational virtual representation of an asset, subsystem, or system, combining digital aspects of how the equipment is built (PLM data, design models, manufacturing data) with real-time aspects of how it is operated and maintained. The capability to refer to data stored in different places from one common digital twin directory enables simulation, diagnostics, prediction, and other advanced use cases [37].

- **IBM** - DT is a virtual representation of a physical object or system across its lifecycle, using real-time data to enable understanding, learning, and reasoning [38].

As for better understanding and looking for improvement ways, an analysis of the current state of the manufacturing industry and a literature review of existing DTs Environment is made aiming specially for the implementation of a Digital Twin with the manufacturing executing system (MES).

2.4.1. Level of integration of Digital Twin

It is important to understand the level of integration of the digital twin with the physical world, because according to Kritzinger et al. not every integration can be defined as a Digital Twin. In order to differentiate the magnitude of integration, three distinct digital representations are explained [5] [2].

- Digital Model (DM)

This is the lowest level of simulation and is characterized by the digital representation of a physical object without any automated data flow between the digital and physical object.

- Digital Shadow (DS)

Digital Shadow is an improvement of DM by implementing the possibility of information exchange between the physical and digital model. However, this information flow is still uni directional as only the digital object changes based on the physical object and not the other way around.

- Digital Twin (DT)

Digital Twin is the highest level of simulation, assuming bidirectional way of information flow between the physical and digital object as shown in Figure 2.6. The data is retrieved from the production system and then processed to be inserted in the Simulation Software, responsible for converting this data in essential metrics that are often object of analysis in the MES and ERP level. Different from the digital shadow, in the DT the digital object also controls the state of the physical objects [39].

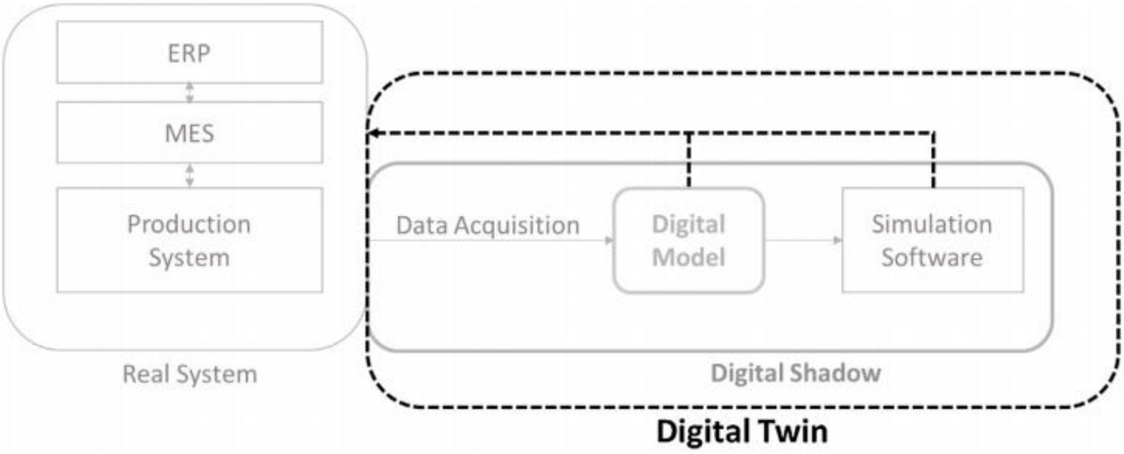

Figure 2.6 – Digital Twin Schema [2]

2.4.2. Integration of MES with Digital Twin

In this section, some frameworks of the integration of a Digital Twin and MES are discussed. MES can be created in two ways, proprietary and/or as open-source system and have the capability of communicating with external software tools. Regarding the protocols used for the communication, the most commonly used in industry is the Ethernet. As from the language standpoint, eXtensible Markup Language (XML), that runs on MT Connect and Structed Query Language (SQL) are some of the most used [5].

2.4.2.1. MT Connect

Enabling data flow and communication between machine to machine and machine to operator, MT Connect can be described as an XML based protocol. The MT Connect standard provides a connotation vocabulary for manufacturing equipment to allow contextualized and structured data with no proprietary format [40].
There are four elements that work in synchronism for the functioning of this protocol. The device which can be various devices varying from PLCs, CNCs, sensors, and they are

connected to an adapter by network, allowing the adapter to gather all the necessary data. The third element is the agent, running usually on a server or on a peripheral hardware. It will aggregate the data from one or more adapters, saving it in a buffer.
The agent, often contemplate an application allowing access and monitoring of the information collected by the adapter [4].

2.6. Bosch IoT Suite

The Bosch IoT Suite is an open source based IoT platform which supports seamless integration into the Bosch IoT eco-system. The Bosch IoT Hub provides devices with various communication interfaces for exchanging data with the Bosch IoT Cloud. The Bosch IoT Suite includes an MQTT message broker referred to as the Bosch IoT Hub.
Bosch IoT Hub is a cloud service of the Bosch IoT Suite that allows the connection of devices through various protocols to IoT applications. Using the cloud service, IoT applications are able to retrieve telemetry data from devices and send command & control messages to devices. Bosh IoT things enables applications to manage digital twins for their IoT device assets in a simple and secure way [41][42].

2.7. Containers and Virtual Machines

Containers are standard units of software that stack code along with its dependencies, so applications can run effortlessly and efficiently between different computer environments.
On the other hand, Virtual Machines (VMs), can be described as an abstraction of physical hardware converting one server into many servers. Hypervisors enable many VMs to run on a single machine, and each VM contains a full copy of an operating system, the application, necessary libraries, and binaries. While for a long time VMs were the common approach for virtualization, recently they have been replaced for more efficient solutions.
As an alternative for the virtual machines (VMs), containers are stepping up, appearing to be a more efficient solution and solving some of the virtual machines limitations, for instance the memory usage and speed [43][44]. Containers, differently from VMs, share the same kernel among them and can be distinguished in terms of resource usage (control groups) and process visibility (namespaces). Now, more than ever containers are increasingly used in automation testing and development of software [45].
The containerization managing process is handled by container technologies such as docker, LxC, runC, etc [46], frameworks that facilitate the package and deployment of containers [47]. Docker is the main adopted technology and Docker containers are stored on the Docker Hub [48] registry which contains millions of images leading currently as the top image repository being used worldwide. Containerization is a lightweight virtualization process to enable fast shipment and deployment of applications. The containers can be described as applications which individually depend on the images. These last are responsible for dictating the configuration and software that is required by the containers [49].

2.7.1. Docker

Docker is a key-enabling technology for microservices, which works as a containerization framework for automatically creating and deploying containers. Docker is composed by the Docker engine and the Docker hub, the two are crucial and they work together to make containerization possible. Docker engine is the platform for creating and working with the containers and Docker hub works as a repository for sharing and managing application stacks as Docker images, as represented in Figure 2.7. Docker uses operating system-level virtualization and linux containers (LCX) to build an environment which isolates the application from the host [50]. Nowadays Docker is available in several operating systems such as linux, windows, OS X, as opposed to the only linux possibility initially when containers appeared. This flexibility that exists at the moment allows easy portability between devices as images can be transported and run in all the supported operating systems.

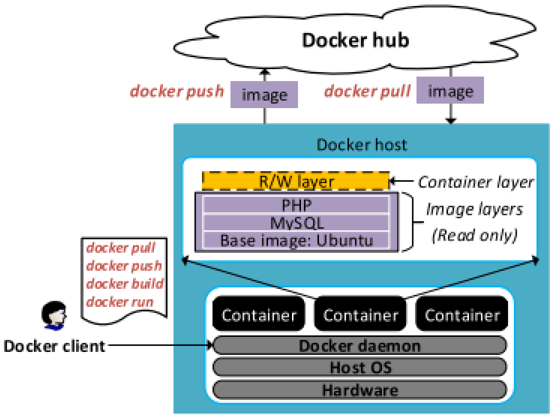

Figure 2.7 – Docker ecosystem [45]

2.7.2. Docker Compose

Several times, running an application requires the use of multiple containers. Docker has this covered up by integrating a solution called Docker compose. Docker compose is a tool to grant blending of multiple services using Docker [51]. This is achieved with the use of YAML (yml) format files that aim for scalability and each of these files contains all the necessary information related to each service that grants that all the containers can work in syntony and run the composed application. An example of application of Docker compose is when running Kafka broker, as it is required the Kafka application and also the zookeeper application. By creating an yml file containing both these services, they will be able to work in harmony allowing Kafka to run.

2.8. Messaging Brokers

As a consequence of the evolution of applications and services, the need for a better and optimized organization of information is imperative. These modern applications and services are often integrated with a huge number of additional entities and the flow of

information becomes too vast resulting in the need of a solution to be able to process all the information and solve the challenges associated with information exchanging.
As a way of solving this heterogeneity between the different applications, message brokers have been given a great development and support over the past years.[52] Message brokers are a software to build a communication bridge between the all the different applications involved in order to bring harmony and common integration supporting cloud native, microservices-based, serverless and hybrid cloud architectures. This integration is commonly a result of the formal messaging protocols that are used to translate the messages between the applications, processing the messages in a way where all the applications are able to understand and grant standardized messaging [53].

2.8.1. Messaging Protocols

Different Internet of Things (IoT) solutions may require different messaging protocols for which the IoT devices use to communicate. Messaging protocols are used to transmit device telemetry (or messages) from the IoT devices to the IoT Messaging Hub (or Broker). There are many protocols currently used and supported but the two most common messaging protocols used for IoT solutions are MQTT and AMQP. These protocols usually communication with the support of TCP.

- **MQTT**

MQTT (Message Queueing Telemetry Transport) is the most popular communication protocol for IoT since IoT assets are constrained devices and it is intended to work on low-power machines as a light-weight protocol. It is a publisher/subscriber-based communication protocol, which enables message-based transfer between applications. This protocol involves a main component defined as message broker, which is responsible for distributing messages between the individual applications or the sender (publisher) and receiver (subscriber) of a message. While sending and receiving, all communication devices contact the message broker, which handles the distribution of all the messages. MQTT uses TCP/IP connection to connect the clients to the broker [54][55].
The typical MQTT architecture can be divided into two main components as shown in Figure 2.8 [56].

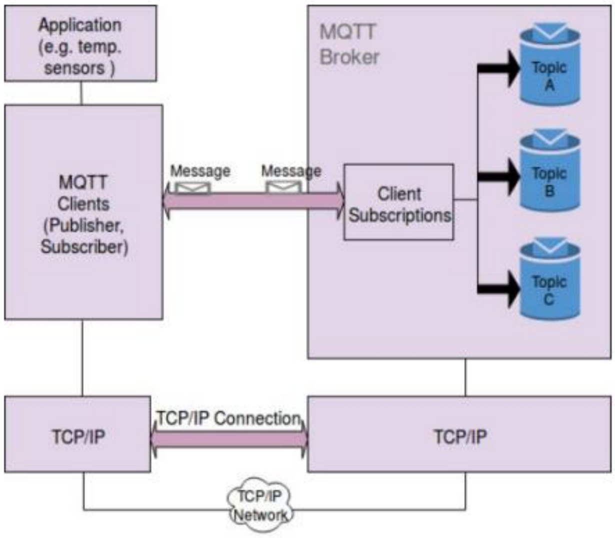

Figure 2.8 – MQTT Protocol Architecture [56]

1) Client: It can be either a Publisher or Subscriber and it always establishes the network connection (TCP/IP) to the Server (Broker). Some of the operations that can be performed are:

- Publish messages.
- Subscribe subjects for receiving messages.
- Unsubscribe to extract from the subscribed subjects.
- Detach from the Broker.

2) Broker: Broker controls the distribution of information and mainly responsible for receiving all messages from publisher, filtering them, decide who is interested in it and then sending the messages to all subscribed clients. It can do the following things:

- Accept Client requests.
- Receives Published messages by Users.
- Processes different requests like Subscribe and Unsubscribe from Users.
- After receiving messages from publisher sends it to the interested Users.

The messages that are published and subscribed to the broker using MQTT protocol are organized by topics. The broker filters incoming messages based on these topics for each connected client. When a publisher sends a message, it always specifies for which topic it is intended. On the other hand, when a subscriber wants to get a message, it indicates which topic is interested in. With the information of the topic, the message broker forwards the message accordingly [56].

- **AMQP**

AMQP (Advanced Message Queue Protocol), is a binary open standard application layer protocol. These protocols appear as a consequence of the contributions of bigger companies that seek a standardized messaging system. It is a protocol designed to support a wide variety of messaging applications and communications patterns. It is not specifically designed for Internet of Things (IoT) solutions, although it employs a decent performance for message communications which include a wide number IoT scenarios [57][52].

- **Apache Kafka**

Apache Kafka is an open-source distributed event streaming platform initially developed by LinkedIn in 2010 and is currently used by many companies to help building high-performance data pipelines, streaming analytics, data integration, and it is suitable for both online and offline message consumption [58][59].
Apache Kafka is based on the publish/subscribe model and it consists of a distributed commit log service designed as distributed system easy to scale out, based on Zookeeper, which enables automatic balancing of consumer/producer/broker [52].
Apache Kafka architecture is a complex but effective system (Figure 2.9). It is composed by cluster which is a combination of brokers. Each broker can contain various topics and for each topic there can be one or more partitions, that work as a safety measure. Along with the partitions it is also possible to define the replication which comes as one by default and minimum. This works in syntony with the partitions to prevent data loss.
Each partition has one server which works as a leader and the other servers work as followers to this leader. The leader is responsible for acquiring and performing all the read/write operations within the server, whereas followers are passive replicas of the leader responsible for replicating the inserted data. Every time the leader server fails, one of the followers out of other servers is set to be the new leader [59].
The flow of messages is very intuitive, a producer produces the messages to the topics which are responsible to store the messages and on the other side there is a consumer to consume the messages that were published before in a given topics [60].

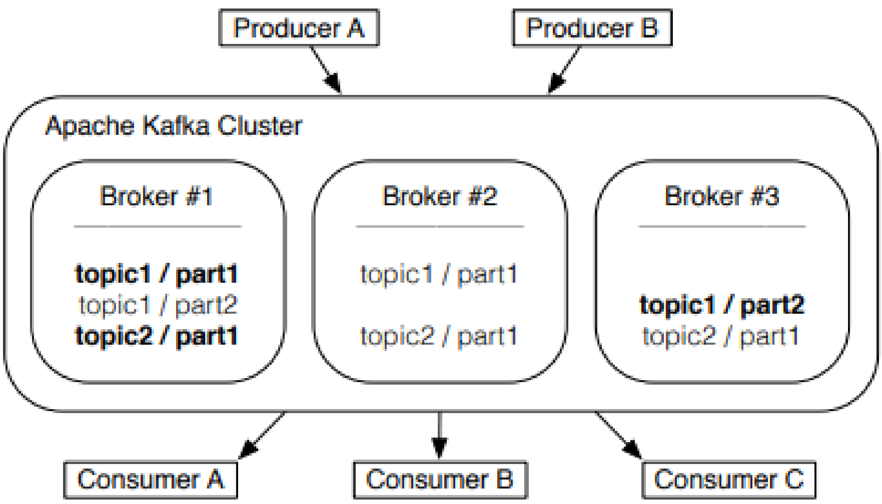

Figure 2.9 – Apache Kafka Architecture [59]

2.9. Databases

In the current software applications, the amount of information flow is vast and sometimes difficult to handle. Databases have been used for many years and they help organize and structure the information, also storing and keeping it, preventing data loss and software failures.
Database is defined as an organized collection of structured data, usually stored in a computer system. Databases are abstract and as a way of facilitating their management, they are controlled by a database management system (DBMS). The information or data and the DBMS in harmony with the applications associated, are characterized as a database system, often abbreviated to just database.
The data commonly follows the same type of organization through most of the database applications, being stored in an undefined number of tables which are arranged in a series of columns and rows. This structure makes it simple to access, modify, send and retrieve the data from the databases and the actions are processed with the help of SQL language, a programming language fully optimized to interacted with databases [61].
There are two types of databases, the relational and non-relation databases. In Table 2.3 there is a list of some of the most common of each group. These two groups are explained the next two sections complemented with an example of each.

Table 2.3 – Examples of relational and non-relational databases

Relational Databases	Non-Relational Databases
Oracle	MongoDB
MySQL	Cassandra
Microsoft SQL server	InfluxDB
SQLite	Redis

2.9.1. SQL Databases

SQL databases, also known as relational databases, have been on use ever since databases appeared. As the name suggest, these databases store the data in normalized related tables, the data being dependent on a primary key as represented in Figure 2.10. Consequently, data needs to follow a structured format in order to get inserted on this tables. These requirements are often reason for losses of performance and speed since there is a huge amount information arriving from multiples sources and rarely it follows the correct structure of the table.
Also, SQL databases are built to run a single machine and this architecture is faulty when compared to the multiple machine approach by the NoSQL databases referred in section 2.9.2. While in SQL database it is required a single and large high-performance machine, in NoSQL databases there are multiples machines forming a cluster and prevent a machine failure from destructing the whole system [62][63].

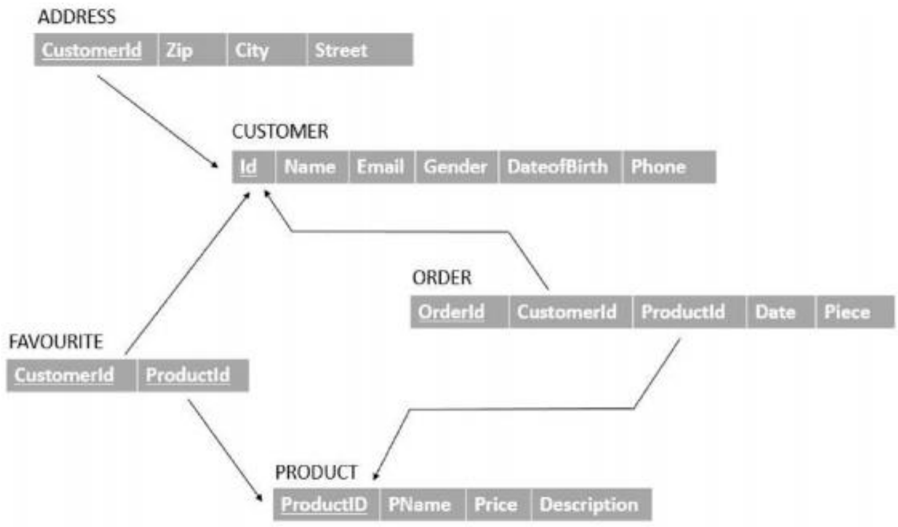

Figure 2.10 – Example of normalized relational approach for SQL databases [63]

2.9.2. NoSQL Databases

While SQL databases are the traditional approach, lately NoSQL databases have been stepping up and they bring a different mark. Also known as non-relation databases, NoSQL databases are claimed to indeed perform better than SQL in some case scenarios.
Due to the increase in assessment of Internet sources and the availability of cheap storage, large amounts of data are stored for a variety of applications, while most of times this data does not come organized neither structured. Big Data is the common reference to this data and working with it requires high performance, speed and flexibility that are mostly common on NoSQL databases as they include flexible schemas and are not centralized [64] [65].
These databases commonly implement different semi structured data models (Figure 2.11) (column stores, key value stores, document databases, graph databases) stepping away from the common related tables and normalized way structure used by SQL databases [63].

BANK DATABASE	
Key	Value
1	ID:1 Joining Date: 15-July-1985 Designation: Cashier
2	ID:2 Joining Date: 19-March-1982 Designation: Manager
3	ID:3 Joining Date: 4-April-1988 Designation: Front Desk Officer

Figure 2.11 – Example of Key Value Database [62]

3. Bosch Group

The company Bosch was founded by a German Engineer Robert Bosch (1861-1942) in Stuttgart in 1886 and is 92% owned by Robert Bosch Stiftung, a charitable foundation. This foundation invests a lot of funds in social and intercultural activities and on medical investigation. [66]

From 1897, Bosch started installing better-designed magneto ignition devices into automobiles and became the only supplier of a truly reliable ignition within the industry. In 1902, the chief engineer at Bosch, Gottlob Honold, unveiled the high-voltage magneto ignition system with spark plug. This product paved the way for Bosch to become a leading automotive supplier.

The first factory was opened by Bosch in Stuttgart in 1901. In 1906, the company produced its 100,000th magneto. In the same year, Bosch introduced the 8-hours day for workers. In 1910, the Feuerbach plant was founded and built close to Stuttgart. In this factory, Bosch started to produce headlights in 1913.

In 1917, Bosch was transformed into a corporation. This corporation kept evolving and today it is placed in four continents and with more than fifty companies. It is mainly connected with the car industry and production of home appliances, but also is connected to electrical tools and security systems.

Bosch is a modern company and aims to captivate the works in order to motivate and also increase their productivity. Some of the values they pass onto their workers with the goal to stimulate their mindset are:

- Responsibility
- Cultural diversification
- Partnership and trust
- Determination and focus
- Credibility and legality

This project will be developed along with the Bosch Termotecnologia S.A. situated in Aveiro. [67] [68]

3.1. Bosh Termotecnologia S.A.

Bosch Termotecnologia S.A (Figure 3.1) has started its activities in Aveiro, more specifically in Cacia in 1977, designated as Vulcano Termodomésticos S.A. The company was started by local entrepreneurs and their operations based on a license contract with Robert Bosch to transfer the technology used by the German company Junkers in the production of water heating systems.
The quality of the products and their selling strategies were a huge factor in the company growth and in 1983 this was recognized by the start of the brand Vulcano, reaching

leadership in water heating systems selling. Five years after, the company was named as Vulcano Termodomésticos S.A., incorporated in group Bosch.
In 1992, Vulcano Termodomésticos becomes the leader in the European market and the third biggest world producer of water heating systems. In 1993, the company created a development and investigation center, allowing numerous innovations. Two years after starts the production of Boilers, along with partnerships with other companies to install the heating systems.
After 1998, Bosch becomes the only actionist, and the company gets the name BBT Termotecnologia Portugal integrating in a Termotecnologia division on group Bosch. Recently in 2008 is named Bosch Termotecnologia S.A.
Nowadays, Bosch has a major impact on the economy of Aveiro, with around 1300 collaborators and leading on water heating technologies. The company goal is essentially development and innovation of new solutions and devices, aiming to become worldwide leaders, while always keeping the costumer satisfied and happy. These solutions include IoT, mobile apps, web, etc.
Bosch Termotecnologia S.A. exports for over than 55 countries. It produces a wide variety of models that are commercialized internationally through the brands of group such as Bosch, Buderus, Junkers, Worcester, Leblanc, Vulcano and secondary brands like Neckar, Zeus.

Figure 3.1 – Bosch Company, Cacia

3.1.1. Bosch Termotecnologia S.A. Organization

Bosch Termotecnologia S.A. is divided in three major groups: Production, Development and Administration. Production section is dedicated to the manufacturing, and they are directly in contact to the shop floor, securing a good production and improving the processes.

Development section is responsible for the innovation, investigation and creating of new and innovative solutions and products. Lastly, there is the administration section that is base of everything and is responsible for the coordination of all the processes.
For the development of this project, the main focus is on the Production group. Inside production, there are many production lines, ones mainly responsible for the production of components, ones more focused on the assembly of a limited set of components to build a part while the rest is more focused on the final assembly of all the components and the realization of safety tests to grant the accuracy of the final product when comparing to what is established in the technical datasheet.
As for the production of water heating systems, there are numerous components that need to be produced or acquired from another companies. Bosch produces the majority of the components since combustion chambers, injectors, ventilator, electronic box, etc. The processes to produce these components include sheet metal bending, welding, CNC machining, etc. For instance, the combustion chambers which are made either by copper or aluminum are produced by machines that take sheet metal and metal tube in rolls and are then bended, stamped and welded.

3.1.2. Bosch Production Lines

The company located in Cacia is responsible for the production of many different electronic products such as heat pumps, water heaters and many more. There are several different lines, ones for producing the components while others are dedicated to the assembly of all the components. The development of this projects does not get affected by the type of production line, as long the layout and the data model follow the same format.
As a result of the advanced standardization of Bosch, all different production lines follow the same structure and the same data used for the example used in this simulation would be completely adequate for a different production line.
The general structure will be explained in this section including the layout, workflow, and the main data necessary for the development of the simulation implemented in this project.

3.1.2.1. Production Line Layout

The layout of a company is the way the production line is organized in terms of location. All the lines in Bosch Cacia follow the same layout and organization to increase standardization.
Starting there is a source of material which will be picked in the first station and then there are a series of stations placed one after another in an "U" layout format. Each station includes a single workstation where only one worker is responsible for performing all the task associated to that station. For the next stations, the process is similar, a worker associated to that station performs the task and then send the part to next station where another worker is waiting to realize its tasks.

3.1.2.2. Production Line Workflow

The workflow is the process of how the work moves inside the production line and how it is organized to provide a fluid and effective sequence. At the start of the production, before the first station, there is a source of material which brings the part in its first state to get picked by the first station. In each station, each worker has an HMI system that is dynamically programmed to provide support by displaying all the required tasks in that station. The worker performs the task and then finalizes the task enabling the part to be moved into the next station through a kart.
The kart is a transporter that circulates through the whole production line guided by a static set of rails. This kart carries the part at all times between each station and it is equipped with a QR code that is read whenever it enters or exits a Station. The kart is locked while the tasks in the station he is located are not completed and is then unlocked once the workers conclude every task.
Whenever the kart completes a sequence in all the stations, the parts is then moved into a final station where some final tests and simulations are performed in other to make sure the part does not present any failure and its ready to be commercialized.

3.1.2.3. Production Line Data Model

While in the whole flow of the production line there is a lot of information and data being processed and sent to Nexeed MES, for the simulation performed in this project even though the amount of data is inferior and its mainly composed by generic data as the simulation is made in a general approach and not specifically about one specific line or processes, all variables were considered.
For this project, the idea is creating a simulation of the layout, the workflow of parts and then calculate some important metrics such as processing time of each station and calculation of key performance indicators (KPIs).
For the first part which is related to the layout and the configuration of the production line, it is necessary to know the numbers of station and their organization. This information is provided in the messages that each station sends to the Nexeed MES, and it includes station number and station location. In the solution section, it is explained how the layout was created based on this information.
Along with this, another crucial and imperative information to consider for the simulation is the time and date at what message is being sent. It will provide the required resources for calculation the processing times of each individual station and consequently the processing time of the whole production line.
By knowing the layout and organization of the production line and by adding the information about processing times along with the information of number of workers per stations is already enough to simulate a production line in a simple and primitive way.
To increase the approximation to the reality, it was studied also the implementation of part information. This will allow the calculation of important metrics such as productivity and efficiency.

Adding to this, worker management information will also be part of this simulation, including data such as the number of workers, their attribution to the different stations, their placement of the production line and the different shifts they perform.
In summary the information that is being considered for the development of the simulation is this project is resumed to the location and identification of stations, the time and date from which every message is sent from the station to the Nexeed MES, the information about the part which is the identifier of the part and its state, successful or with defect and lastly, the number of workers, their association to each station and the shifts setup so that the simulation model can run on the same schedule as the real company.
The production Line Data Model represented in Figure 3.2 will be used as the information flow model for the creation of the simulation model implemented in this project.

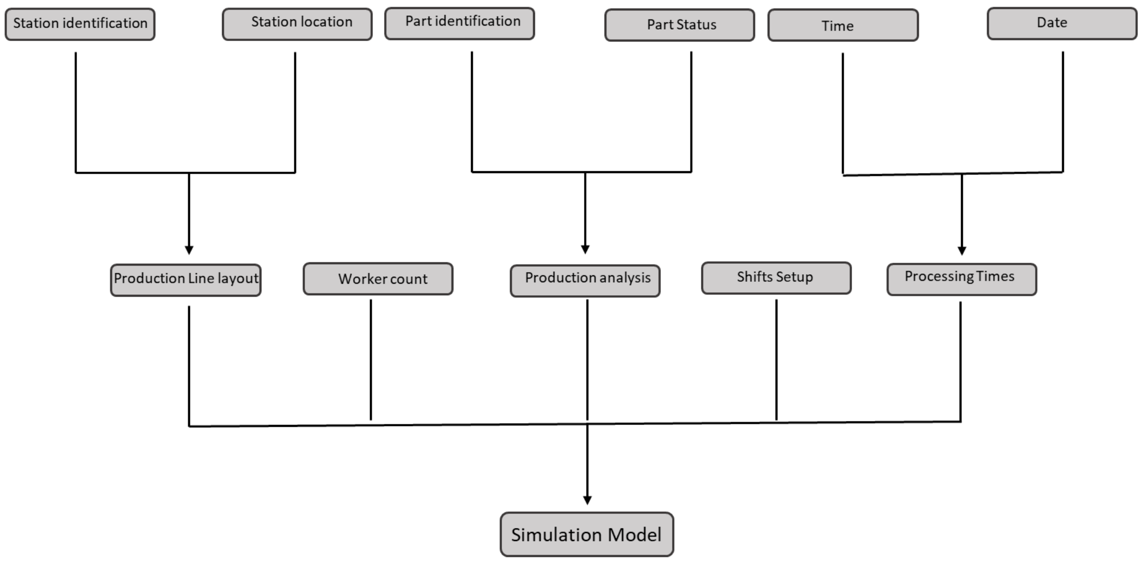

Figure 3.2 – Production Line Data Model

In the following section it will be explained the source of this information and how it was possible to retrieve it and store it and order process it.
It is important to understand that the simulation will be generic following this generic model which gives additional flexibility because if it is required to simulate a different processing line, the data model will be the same and so the simulation will be performed similarly.

4. Proposed Solution

4.1. Solution Architecture

This section intends to present a solution to integrate a Manufacturing Execution System (MES) with a simulation model of a production line. For the purpose of this project, the solution is implemented and tested on the line number one (1), a production line located on the Bosch Termotecnologia S.A. in Cacia, with the integration to a simulation of the Nexeed MES that was developed to substitute the MES software that is currently used on Bosch.
Figure 4.1 represents the proposed architecture for the integration of the MES with the DT.

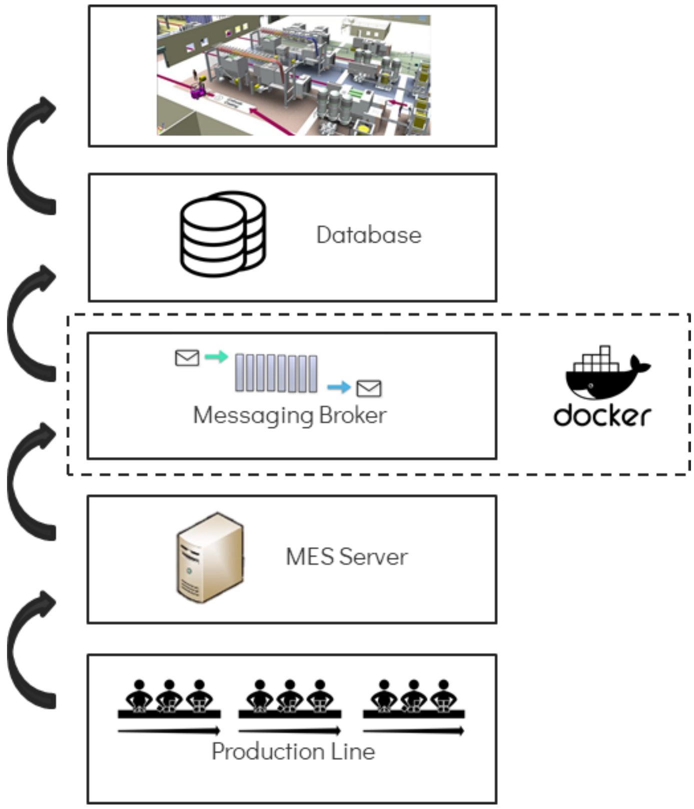

Figure 4.1 – Solution Architecture

The first layer, data acquisition, consists of a windows server, receiving real-time data from the production line which includes machines, sensors, PLCs, etc. This data is received in xml format using the protocols OPCUA or ethernetIP. Each PLC sends an individual message, and this data is further processed in the MES - e.g. stored in Oracle Database, analyzed and answered back to the machine or forward to 3rd party software using Apache Kafka as a broker, being able to produce JSON format models that are the aggregation of all the different messages from the PLCs. These JSON messages are the

source of information that is used in this architecture. This data will be produced to the messaging broker by implementing a simulation of the MES.
The following layer, processing layer is composed by the messaging broker that is receiving the messages, composed by the JSON format data models, and consequently storing the messages and redirecting them. This process is realized by a consumer that will be consuming the messages that were previously produced by the producer and then process the message to place in the third layer.
The consequent layer, third layer, represents the storage layer and it composed by a database whose purpose will be storing the necessary data intended as useful for the creation of the simulation model in the simulation software.
The simulation software grants the possibility of extracting data from various sources and use it to complement the simulation model. Between the sources, it is included the database which is used in this project. The proposed database will be additionally explained in the section 4.1.3.

- Data Acquisition
- Data Processing
- Data Storage
- Simulation Software

4.1.1. Data Acquisition

4.1.1.1. Message processing in Nexeed MES

Before the explanation of the solution, it is necessary to understand what messages arrive from the MES, its format, structure and explain the dissection of the necessary data provided in the messages.
As explained in section 4.1, the data collected from the machines and sensors is sent via OPCUA or Ethernet to the Nexeed MES in an XML format, Figure 4.2. Each PLC that is placed in the production line sends a message whenever an occurrence happens and as a result of this, the numbers of messages arriving to MES, while still being necessary, is vast. Even though MES processes all the messages, only the necessary ones are transmitted.

```xml
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<root>
  <header eventId="1" eventName="partProcessed" version="1.0" eventSwitch="-1" contentType="3" timeStamp="2021-03-
29T06:12:16.8074110+02:00">
    <location lineNo="1" statNo="30" statIdx="1" fuNo="1" workPos="1" toolPos="1" application="PLC" processName="Final Functional Tes
t" processNo="1260" />
  </header>
  <event>
    <partProcessed identifier="557213900848077736501529" />
  </event>
  <body>
    <structArrays>
      <array name="processRealData">
        <structDef>
          <item name="name" dataType="8" />
          <item name="value" dataType="4" />
          <item name="upLim" dataType="4" />
          <item name="loLim" dataType="4" />
          <item name="checkType" dataType="3" />
          <item name="unit" dataType="8" />
          <item name="resultState" dataType="3" />
        </structDef>
        <values>
          <item name="Blr_Supply" value="225.91" upLim="250.000" loLim="220.000" checkType="5" unit="V" resultState="1" />
          <item name="Code_Plug" value="115.5" upLim="1011" loLim="-1" checkType="5" unit="" resultState="1" />
          <item name="Gas_Flow_Min" value="10.85" upLim="10.890" loLim="8.910" checkType="5" unit="L/Min" resultState="1" />
          <item name="Blr_Gas_Pres_Min" value="36.72" upLim="37.500" loLim="36.500" checkType="5" unit="mbar" resultState="1" />
          <item name="Bnr_Gas_Pres_Min" value="-4.72" upLim="5.000" loLim="-5.000" checkType="5" unit="mbar" resultState="1" />
          <item name="CO2_Min" value="10.72" upLim="10.800" loLim="10.300" checkType="5" unit="" resultState="1" />
          <item name="CO_Min" value="36.89" upLim="85.000" loLim="0.100" checkType="5" unit="" resultState="1" />
          <item name="Gas_Flow_Max" value="28.28" upLim="30.124" loLim="26.395" checkType="5" unit="L/min" resultState="1" />
          <item name="Blr_Gas_Pres_Max" value="36.76" upLim="37.500" loLim="36.500" checkType="5" unit="mbar" resultState="1" />
          <item name="Bnr_Gas_Pres_Max" value="-3.71" upLim="5.000" loLim="-5.000" checkType="5" unit="mbar" resultState="1" />
          <item name="CO2_Max" value="11.52" upLim="11.700" loLim="11.300" checkType="5" unit="" resultState="1" />
          <item name="CO_Max" value="76.16" upLim="302.000" loLim="40.000" checkType="5" unit="" resultState="1" />
          <item name="CH_Flow_Rate" value="19.76" upLim="22.000" loLim="12.000" checkType="5" unit="L/min" resultState="1" />
          <item name="Delta_CH_Flow_Temp_Cold" value="-.55" upLim="5.000" loLim="-1" checkType="5" unit="C" resultState="1" />
          <item name="Delta_CH_Flow_Temp_Hot" value=".21" upLim="5.000" loLim="-1" checkType="5" unit="C" resultState="1" />
          <item name="CH_Flow_Press" value="430.26" upLim="999" loLim="2.500" checkType="5" unit="bar" resultState="1" />
          <item name="CH_Flow_NTC_Temp_Rise" value="55.94" upLim="999" loLim="15.000" checkType="5" unit="C" resultState="1" />
        </values>
      </array>
    </structArrays>
    <structs>
      <resHead result="1" typeNo="7736501529" typeVar="" workingCode="0" nioBits="0" cycleTimePrev="34693" />
    </structs>
  </body>
```


Figure 4.2 – Example of XML message sent to MES.

MES is responsible for the aggregation of all the data needed and further conversion to a JSON formatted message, organizing the information in a more compact, accessible, and readable format Figure 4.3.

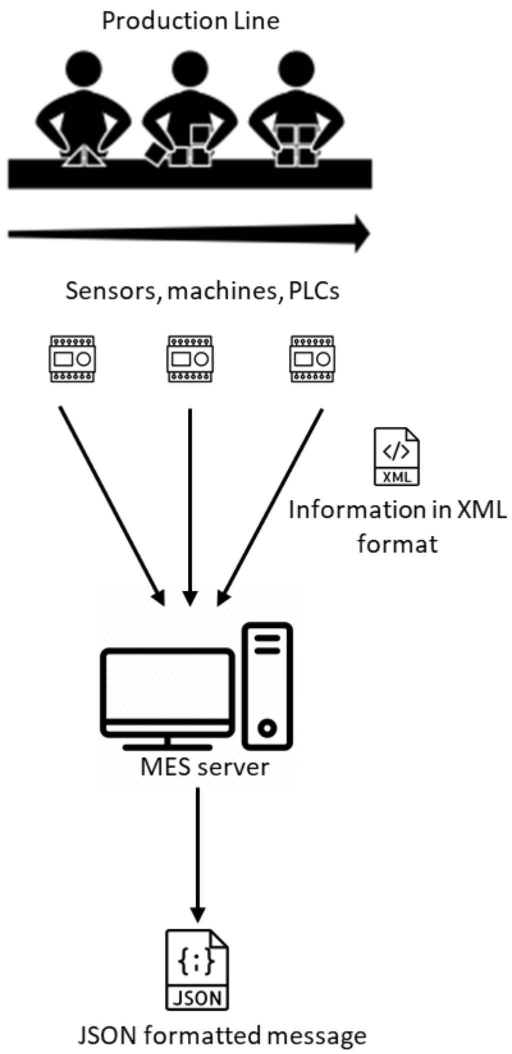

Figure 4.3 – Message Processing in Nexeed MES

In Figure 4.4, there is an example of a single structured JSON message that is sent from the MES. These messages are constantly being sent to the messaging broker as soon as a new part exits a station. The message broker then stores all the messages, which can be consumed by one or more users.

```
// Message Name : line_01_2021.03.05_15.49.42.json
{
    "Application" : "PLC",
    "BodyVariables" : [...
    ],
    "EventId" : 1,
    "EventVariables" : [...
    ],
    "HeaderVariables" : [...
    ],
    "Identifier" : "0377d068-4396-4b4a-923f-26b30988e4fd",
    "Location" : {...
    },
    "Name" : "partProcessed",
    "ProcessName" : "Heat Cell Sealant Leakage Test",
    "ProcessNo": 1170,
    "SourceId": "127.0.0.1"
}
```

Figure 4.4 – Structured JSON message sent from MES.

Starting with the name of each message, it is composed by the identification of the line, line one in this case, followed by the date and time of which the message was sent. The message is composed by three main arrays, “BodyVariables”, “EventVariables” and “HeaderVariables” that contain a huge amount of information regarding the part that just moved out of the associated station. In order to get the information about the location to where that message is associated, there is value name “LocationId” that provides all the information related to the location or even more intuitive it can be analyzed in the location fields, for instance, “StatNo”: 20 which indicates the station from where this message was sent.
The amount of data provided by these messages is vast, containing hundreds of lines of data in each array, although for the development of this project, not everything is necessary, in fact only a small part of it is useful and will be analyzed and processed. From the body variables it will be retrieved only the part state, which dictates whether that part is Ok or not OK, depending on if it is one or zero. From event variables, it is possible to get a value that will complement the state, which is the identifier, responsible for identifying each part that is produced. Also, in header variables, it contains the time stamp, in other words, the time where each message is sent to the MES, and this is also required. Lastly, in the location object, the station number will be retrieved as this value will help organizing the stations in the simulation model and also divide the messages into different stations as they come all together in the beginning. This is later explained with more detail in the section where the consumer is explained, since the consumer is the one responsible for consuming the messages and later extraction of useful information.

4.1.1.2. Data Production

While Nexeed MES is the software used in Bosch for processing all the information associated to the shopfloor and also responsible for sending this information, it is not possible to access this software outside company environment. As a result, it will not be possible to use this software in real time creating a necessity of developing a way to simulate the MES. It can be simulated by implementing a method that allows the flow of messages every certain time. This solution works as a simulation of the MES as it will be producing the messages similar to the one explained in section 4.1.1.1 to the messaging broker Figure 4.5. The solution is explained in detail in section 5.1.2. Using a messaging broker and a topic it is possible to send the messages to the broker and get an approximation to what the MES is performing in real time in the company.
Creating a simulation of MES does not bring any disadvantage because the process of messaging will be exactly the same so there is no necessity for any complex adaptation when this solution is further applied in company environment with the use of the real MES.
It is important to understand that the simulation proposed will not be a simulation of the whole MES software but solely of the messaging process that is processed by MES. Still, the messages that will be sent are real messages used in a real time scenario of a past date, so that the model can work with concrete data and not simulated values. When necessary to implement this project in the Bosch, it is only necessary to change the source of information, keeping the same destiny and same process of message production.

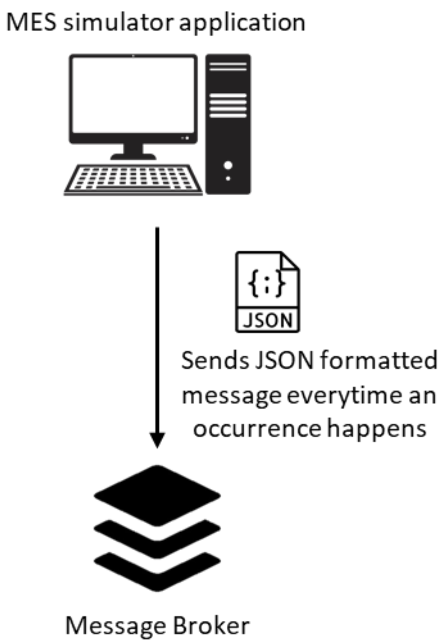

Figure 4.5 – Data production to message broker

4.1.2. Data Processing

This layer, data processing, has a very important role, as it is in this level that the messages are analyzed, and the relevant information is treated and extracted from the messages.
In this layer, its where all the data processing related to the messaging broker architecture will be performed since the production of messages to the broker, until the consumption of these messages and later processing of information and consequent insertion in the database Figure 4.6. Starting, the messages are sent from the application that will be performing as the MES simulator to the messaging broker where they will be stored. At this level, the messages are stored in a topic of the messaging broker and ready to be consumed. An additional application will be responsible for extracting the messages from the broker and it is also responsible for processing the data. As soon as the data is processed the application connects to a database and sends the necessary information storing it in tables.
The messaging broker will be running inside a container, and it will be used Docker to manage the containers. The reason for using containers is the portability of the solution and also the capacity for installing and running the messaging broker. By running the message broker on a container, it is possible to access this container in any other machine and still run the solution in an effortless way.

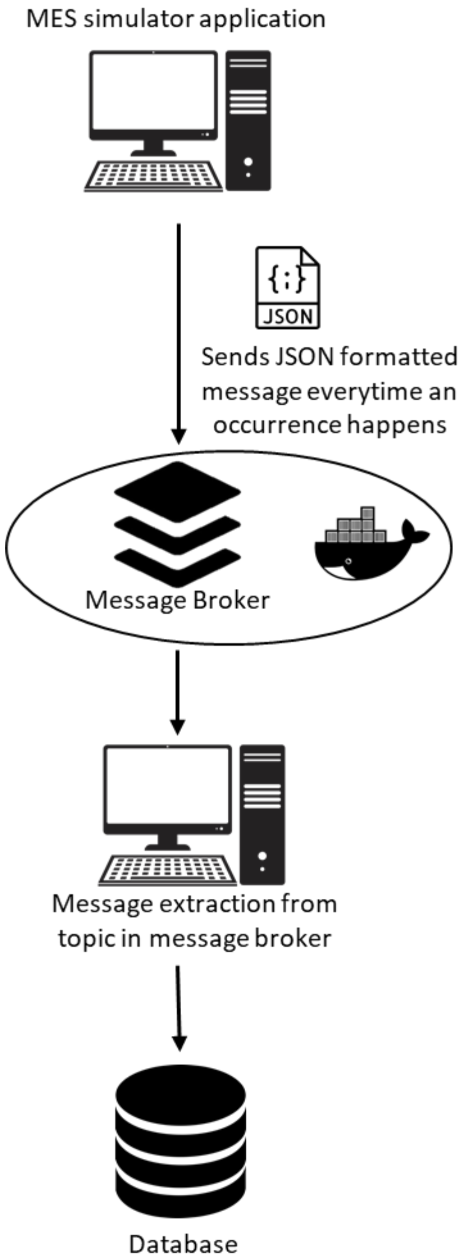

Figure 4.6 – Data Processing Schema

4.1.3. Data Storage

In order to allow Tecnomatix to have real time interaction with the messages that are sent by the MES it is necessary to create a database, so it will be possible to have access to real time data in Tecnomatix as it allows the integration with some database interfaces. In order to create a database, it is important to understand what is expected in Tecnomatix so the data can be added accordingly, and not every value that comes in the message from the MES.

In Tecnomatix it is intended the creation of a single production line where this line is composed by different stations. These number of stations is variable depending on each production line and the goal is to build a simulation model where stations can be added if a new message arrives with data from that new station. Each of these stations include one and only workstation with one worker where each worker performs a given number of tasks defined beforehand.
Based on this information, it is possible to get an overview of the data that happens to be crucial for the creation of this model. The most important, the station number, is being constantly added in the table. The reason is straightforward, with the station number, it will be possible to divide the data by station and consequently each station in Tecnomatix will have access to only the data associated to that station. Additionally, by saving the station, it will be possible to monitor if a new station is added into the table, and if a new station is added, Tecnomatix will be able to automatically add that new station and workplace in the model and work with the data related to that station.
Along with the station number, when working with stations, probably the most important information is the processing time of each individual station as this number will dictate much information about the production line. In order to allow the calculation of this, the time at which each message is sent is also added into the table along with the date. In section five, the implementation, it will be explained how it is possible to calculate the processing time based on the time at which messages are sent.
Also, not directly related to the stations, but more with the output, or in other words, what is being produced, it was decided to add the identifier of each part that is being produced, which is a number composed of 23 digits that represents each single part that is moving in between stations.
Lastly, to complement these last number, it was intended to also add the State value which indicates whether the part that is being produced is ok or not ok and needs rework. The idea was to create a database that was intuitive and flexible so it would facilitate the assessment of data. Table 4.1 represents the table that was created to later integrate with the shop floor simulation software.

Table 4.1 – Production Line data table

ID (Primary Key) (INT) (Auto increment)	**Identifier** (INT) (Not Null)	**State** (INT)	**TimeStamp** (Time) (Not Null)	**DateStamp** (Date) (Not Null)	**Station (INT)** (Not Null)

Identifier – The identifier is a number composed of 23 digits that can be found inside the Event Variables present in the messages that are sent by the MES. This identifier is unique for each part that is being produced so it is used to distinguish among all other parts. The number is implemented in the database as a way of identification for every part that enters or exits a station.

State - State is a single digit number that is found inside the Body Variables. The state value can only assume two numbers, 1 or 0, and based on this numbers it is possible to know whether the part is ok and can move to the next station, or the part is not ok to move on and needs to be put apart for rework.

- 1 - Part Ok (can move to the next station)
- 0 - Part Not Ok (needs rework)

TimeStamp – "TimeStamp", as the name suggest corresponds to the time at which the message is sent to the MES. Inside the Header Variables, there is a value corresponding to the Time Stamp, which actually includes the time and date. In order to create a more intuitive table and for organization reasons in Tecnomatix, the time was separated from the date. The time is represented in the HH:MM:SS.XXXX, "H" for hours, "M" for minutes, "S" for second and "X" for milliseconds.

DateStamp – "DateStamp" corresponds to the date when the message is sent to the MES. This value is also found inside the Header Variables along with the Time and it is represented in the following format, YYYY-MM-DD, "Y" for year, "M" for month and "D" for day.

Station – Station can be found in the location object, and it is composed of a number with two digits, for instance "10" that corresponds to station ten. This field is very important, as it allows the possibility of sending all the messages to a single table and that work with the data relatively to each station solely by iterating through station column.

Also, not related to the production line specifically but that is important to contain in a real time simulation of a production line, the information regarding the working shifts will also be stored in tables in order to be possible to simulate the workflow of the production line in the specific shifts performed by the company in study.
The tables containing this information are not created by the author as they are already created and in use in Bosch. Still, it is importation to mention the tables and explain them for better understanding of the shift management.
This project will contain two different tables regarding the shift, Table 4.2 containing all the information about each shift individually and a second one, Table 4.3, a static table that will not change and solely contains the information regarding the three different types of shifts.
The first table is composed by the Shift ID, starting, and ending times and also the shift type. This information will be used to fill the shift calendar that Plant Simulation Tecnomatix contemplates.
The second table includes the shift type and the shift type description. There are three different types of shifts, 0, 1 and 2. They correspond to early shift, late shift, and night shift.

- "Shift" Table

Table 4.2 – Shift Data table

ID (INT) (Primary Key) (Auto Increment)	**SHIFT_PLAN_SHIFT_UUID** (STR)	**SHIFT_PLAN_ID** (STR)	**SHIFT_START_DAY** (STR)	**SHIFT_START_HOUR** (STR)	**SHIFT_START_MINUTE** (STR)	**SHIFT_END_DAY** (STR)	**SHIFT_END_HOUR** (STR)	**SHIFT_END_MINUTE** (STR)	**START_FLAG** (STR)	**PLANED QUANTITY** (STR)	**TIME_STAMP** (STR)	**SHIFT_ID** (STR)

- “ShiftTypes” Table

Table 4.3 – Shift Types Data table

ID (INT) (Primary Key) (Auto Increment)	**SHIFT_ID** (INT)	**STRING_ACRONYM** (STR)	**TIME_STAMP** (STR)

4.1.4. Tecnomatix

Plant Simulation Tecnomatix is the link between the information source, database, and the production line. As a shopfloor simulator, Plant Simulation Tecnomatix also enables the integration with various information flow objects, objects that act as information sources containing data that can be accessed inside the shopfloor simulation objects. Through these objects, database in concrete, it is possible to connect the real time data provided by the database with the simulation objects, stations, drains, sources, etc. This level of integration creates the intended digital twin, a virtual representation of the production line with real time data flow.

For the simulation of the production line, Tecnomatix appeared to be the best option as it is a modern, capable, intuitive and above all it comes with an enormous number of features which enables the replication of every aspect that is required. Additionally, Tecnomatix is already being used in Bosch which proves that is a capable software. While other shopfloor simulators were analyzed, Tecnomatix includes features that highlight from the others, and it will be more straightforward to integrate in the Bosch company environments as some workers already possess some knowledge and experience using it.

The proposed solution architecture for Tecnomatix is building a production line that will integrate all the stations which are functional in the production line one at the moment of the replication.

The goal is to have a database that will be updated in real time by the consumer, and this database will be integrated in the frame of the simulation model so every object will be capable of accessing all the information. With this, it is intended to have fully operational stations that will be constantly updating their processing times and efficiency based on the information that is arriving from the MES.

Every object that is added into the Tecnomatix frame has multiples fields and each of this fields can be accessed externally by using the object called “Method” which works as a coding script that makes use of the programming language SimTalk, version 2 at the moment of creation of this project. While inside every object, the name of the variable corresponding to each field of information, for instance processing time, can be seen in the SimTalk language, for example “ProcTime” for the processing time field. By attributing a value to this variable in the method using the SimTalk language it is possible to change the value of this field in the object from which it corresponds.
In Figure 4.7, there is an example of the object “Station10” in the “Times” tab.

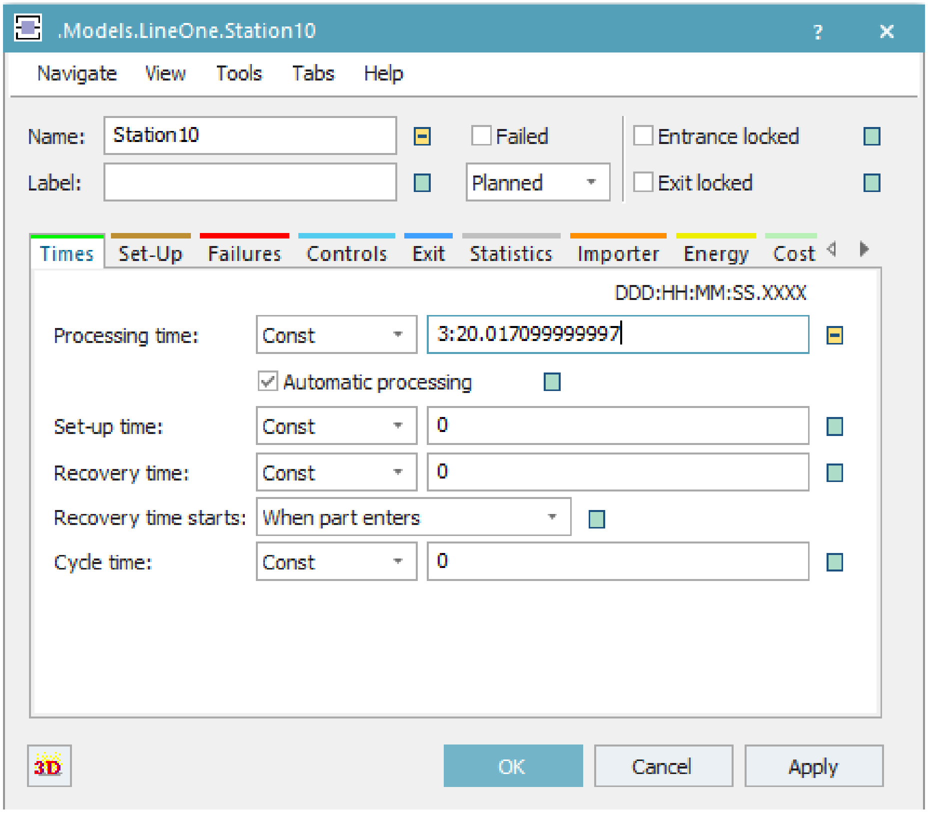

Figure 4.7 – Station10 Object Times tab

For each of those information fields there is a variable corresponding to the SimTalk language as demonstrated in Figure 4.8.

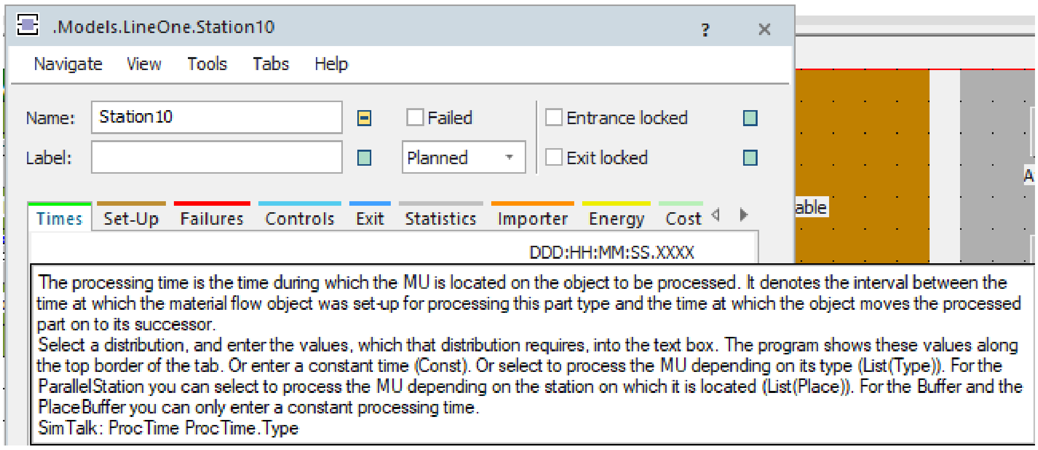

Figure 4.8 – How to access Station object with SimTalk.

By using this capability of Plant Simulation Tecnomatix, it is possible to create models with the help of code which results in a more flexible and versatile model, as most of the operations made in SimTalk could not be performed inside the objects in a dynamic manner.

The database integration is also possible due to the interaction between the database object and various methods. The database object grants the possibility of connecting to a database by giving its directory and attributing a name to it. The method, on the other hand, enables the use of SQL programming by refereeing to the database name beforehand. As a result, from a method it is possible to perform any SQL operation that would be performed in any other programming language. Once the data is retrieved it is then added into a data table object in Tecnomatix, which is nothing else than an Excel sheet, but that facilitates the access to the data afterwards as this object is fully optimized for the integration with the SimTalk language and Tecnomatix objects.

4.1.4.1. Simulation

The simulation proposed for this project is intended to work as a support tool that engineers will be able to use for many different purposes such as monitoring, testing the production lines located in Bosch. Before this data was already possible to access inside Plant Simulation Tecnomatix, although it was inserted manually and that lacks efficiency and scalability. It was not possible to monitor the production line in real time and above all the required time for inserting data for the whole production line would be too severe and not compensating at all.

With the solution proposed in this project that was explained in the sections above, the simulation will have full integration with real time data that is being sent from the production line stations to the MES and from the MES to the database at which the simulation is connected.

The simulation will contain various sections where each one of them will have a different purpose and it is expected that they will work in harmony between themselves. The first requirement in the simulation is the possibility of monitoring the production line in a digital approach and for that, some indicators will be updated in real time where they will show some key performance indicators that are essential for the monitoring of the production line. KPI as the name suggests are indicators that indicate the performance in this case of the production line, and they are an efficient way of monitoring the most important data such as processing times, setup times, efficiency, throughput, etc. These indicators are not included in the data that the MES send so every time they want to be measured it need to be made manually which is not efficient. In Figure 4.9 it is possible to observe the process of monitoring through KPI calculation. The simulation will have methods to make the calculations required in real time and then display them in the simulation. The scalability is the main goal of this approach as every time a new KPI needs to be measured, the process is already made just being necessary the formula adaptation in a new method.

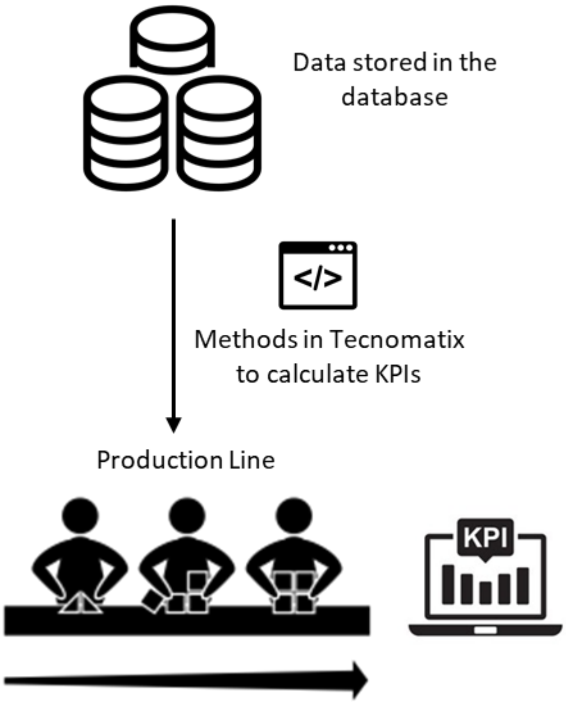

Figure 4.9 – KPI monitoring in Tecnomatix Simulation.

Apart from monitoring, the proposed simulation will also integrate features for the possibility of testing new changes in a dynamic way. Testing a simulation in real time scenario is not a straightforward task and often results in big monetary and time impacts. Still simulating in digital format is also not simple and depending on the level of simulation it can take too much time. To overcome this problem, this solution is intended to integrate a feature for adding simulation objects automatically, based on the information that is arriving in real time. This way the user will not need to make the simulation manually but instead the software will do it automatically by analyzing the data. This feature is essential for helping in the testing of new changes and combined with the monitoring provided by the KPIs it is possible to test the credibility to whether is it an improvement or not.

5. Implementation

In this chapter, the implementation of the solution will be explained. First, an explanation of the diagram and the composition of the layers. Then detailed approach on all of the components, explaining the thought behind every step.
No hardware is used in the implementation of this architecture, yet various software's are involved so a brief explanation of each will also be presented to help in the contextualization of the implementation of the solution.

5.1. Diagram

Figure 5.1 represents the solution that was implemented which is divided in three layers.

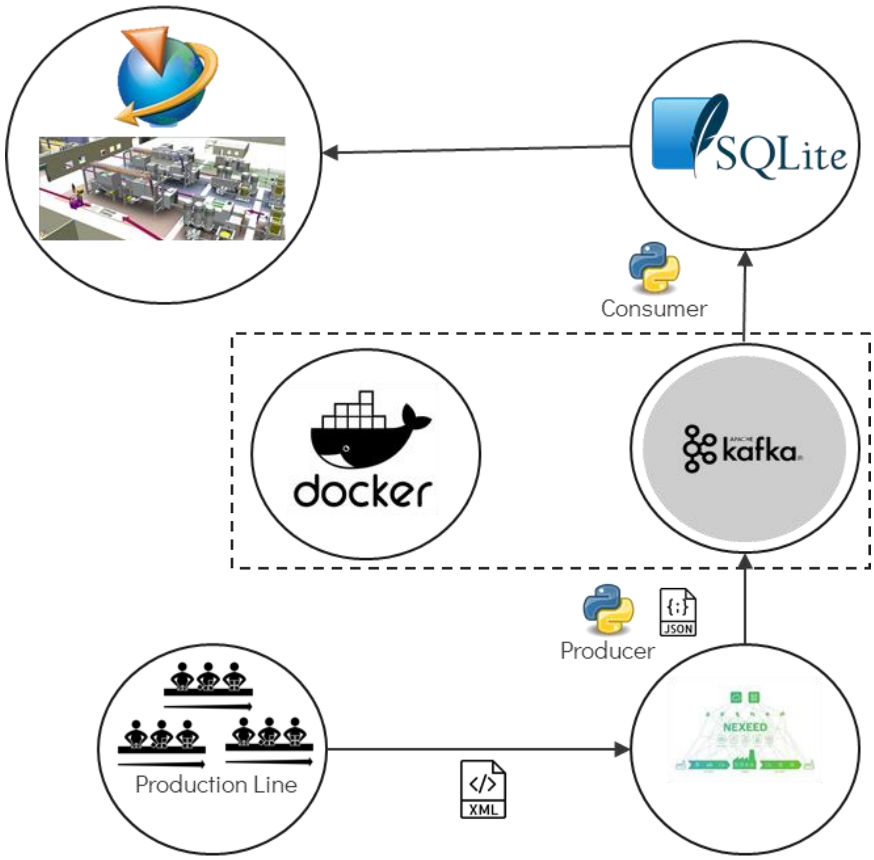

Figure 5.1 – Implementation Diagram

The first layer, data acquisition, will not be focused on this chapter as this layer is already being performed in Bosch company, so we will be focusing the implementation essentially from the point where the data is already acquired and its ready to be send to the message broker Apache Kafka.

The second layer, data processing, is where all the data is being produced, processed, and later consumed to move on to the next layer. While the data in Bosch is being sent to the MES server and later processed and sent via MES to the broker, for the creation of this project, a different approach was made. Since it is not possible to access the real data that is being sent in real time from Bosch, it was imperative the creation of way to simulate the production of this messages in a way that it would still use the original messages of the MES but not in real time. Here is where the producer surges, an application that was created using the Python programming language and that will be responsible for accessing a source containing a large number of saved messages and be capable of producing these messages in the exact way as the MES server does in real time in Bosch.

These messages are being sent to the message broker Apache Kafka which will be responsible for storing and organizing the messages in a Topic. The final step of the data processing consists of another application made in Python called the consumer that will be responsible for consuming the messages that are being stored in the Kafka broker. This application will also work as the connection with the next layer. It is integrated with the database SQLite and the messages are being processed and later being added to the database using SQL queries. The data processing of the messages will be explained later on the explanation of the consumer.

The third layer is the data storage, where there is a database in SQLite. The reason for SQLite is because it is the one that allows the integration with Tecnomatix as it is the database that the version of Tecnomatix being used supports. Additionally, it seemed to be a good choice as it turned out to be an intuitive and simple solution for the database. This database will only store that data that was intended to be necessary for the simulated production line in Tecnomatix, so it will contain limited information and not the whole information that is present in each message.

Lastly, the Tecnomatix simulation model, where the main focus of this thesis will be put. The production line simulation model features integration with the SQLite database enabling real time flow of data. Inside the simulation there are several different sections that will enable the workers to monitor, back test and improve the production line. These sections are explained in detail in section 5.4.

5.2. Data Processing

5.2.1. Producer

Since it is not possible to access the real time flow of messages that is sent from the MES used in the company, Nexeed MES, it is required to overcome this problem by finding a way to simulate this software in a flexible and intuitive way. To perform this simulation, it is important to first understand how the MES processes the messages and how it sends it, in what cadence and what format.

Nexeed MES is a software that is used in the Bosch company to provide support for the shopfloor and as so, it is connected to all the machines that are being operated in the company. Each machine has its own PLC that is responsible for sending the data to the Nexeed MES in XML format by the OPCUA protocol. The aggregation of machines

presented on a specific production line sends a huge number of data provided by all their PLC's, which is processed in the MES. MES is responsible for acquiring all the information in XML format, join it and transform it into a dictionary, JSON format. These messages are explained in section 4, and they are sent every time a part leaves a station.
In Figure 5.2 it is represented the architecture of the MES simulator that was implemented in this project. Starting by parts, in order to replicate the messaging process of MES, a sample of this messages was obtained and stored in a folder. The producer, a Python application created for the simulation of the MES will be accessing the folder containing all this messages and will send them in a sequence ordering by time exactly the same way as the MES server does it. This script also includes the Python Kafka library to produce these messages to the broker Apache Kafka that will be responsible for storing the messages in a topic named Nexeed MES. In order for the messages to be produced, they need to be first converted to string format and later encoded to utf-8 so that Kafka allows producing the messages.

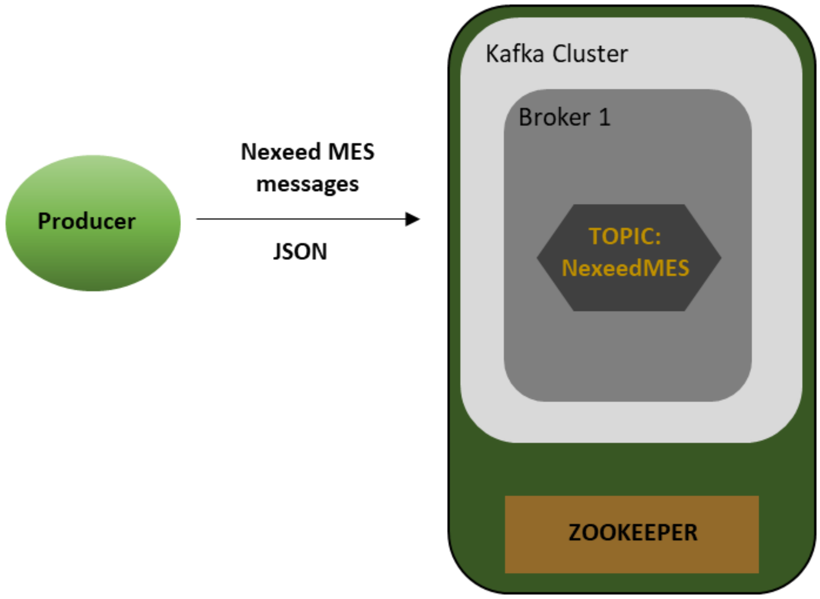

Figure 5.2 – Producer Schema

5.2.2. Consumer

All the messages that are being produced by the producer are being stored in Kafka broker and these messages can be consumed based on the Kafka architecture which relies on a producer/consumer messaging control. A Python application named consumer was created to perform this task. Although the application is not only responsible for consuming messages, in this section the focus will be more on that aspect. The consumer will access the topic which the producer is sending the messages, Nexeed MES, and consume one by one the messages that are arriving, as represented in Figure 5.3. The consumer is called "mygroup" and it listens to the broker waiting for any new message to arrive. Once a message arrives to the topic, the consumer consumes that message and is

then decoded as they come in the encoded (utf-8) format and later converted from string to JSON format to facilitate the iteration of data.
Once the messages are consumed, it is imperative to work with this data and iterate what is in fact necessary, as these messages come with a lot of information that happens not to be everything crucial for the development of this project.
After the string is transformed into the JSON dictionary, it becomes easier to iterate through the objects and get the desires values. For the development of the production line, it was necessary to incorporate the station number, time, date, part status and the identifier. The station number is location on the location object, and it refers to the exact station from which that specific message was sent. This information is crucial on the Tecnomatix model as it is the information that is being used to separate the messages into groups so they can be easily processed. There is a reason for why there is not a table in the database for each station, and it will be explained in the database section.
The time and date are also being iterated from the messages as it is imperative in a production line to calculate and track the times such as processing time. This is probably the most important information on the production line as it dictates which station is taking the most time, where are the bottlenecks being created and much more like the calculation of KPIs. This processing times will be updated in real time as the messages are arriving at the Tecnomatix, this is a complex process and will be explained in detail in the Tecnomatix section.
The identifier is also part of the information retrieved, it is not specifically related to the stations like the previous information, although it is in fact the identification of every part that is being processed in the station. This is very important to keep track of which parts are being produced in the stations and to allow better organization and monitoring.
The last information being iterated is the state of the parts. This value will assume only two values, either “0” or “1”, part not ok and part ok respectively. This information will be complementing the identifier, as there will not just be information on which part is being produced but also if the part is ok and can move on or it is not ok and needs rework.

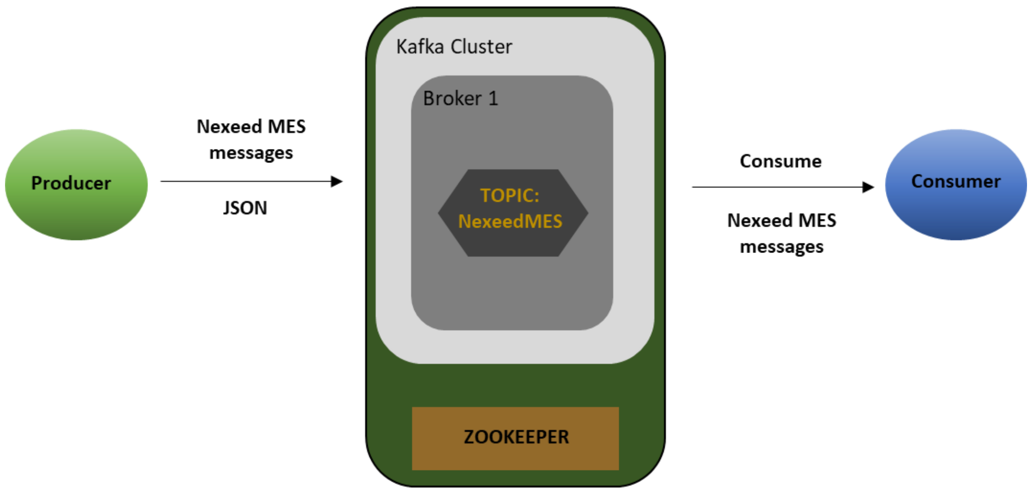

Figure 5.3 – Consumer Schema

5.3. Data Storage

In this section, it is explained the choice of the database and the thinking process behind the creation of the database and corresponding tables. For the implementation of this solution, there were some limiting factors when choosing the database. The main problem was directly related with Tecnomatix as it only allows the connection with few database possibilities. The student version, the one that is used in this project, in fact only allows the integration with SQLite database. Due to this, SQLite is the database that is operating in this implementation and even though it is not the best option for databases, it for sures serves the intention of this project.

SQLite works in file format and is a library in C programming language that incorporates a small, fast and high-reliability SQL database engine. As the database is only a file, it is complemented by a database browser for SQLite that is a graphical UI for monitoring, editing, and working on the database files.

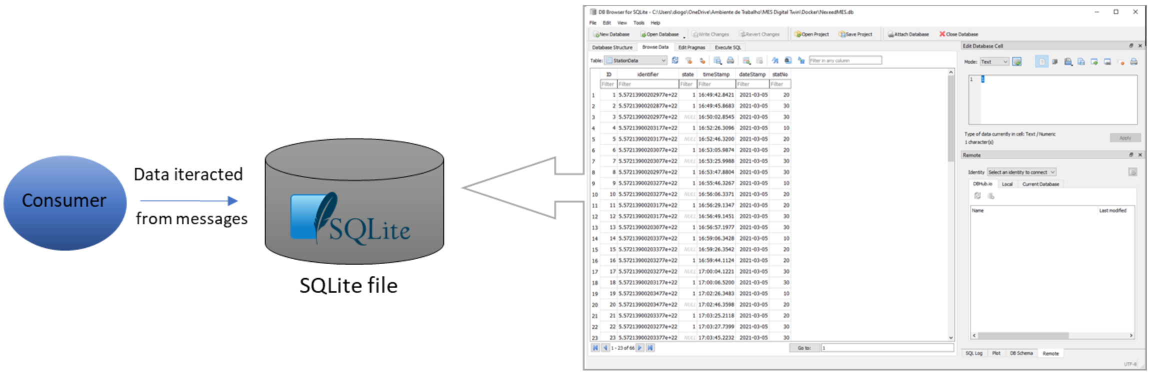

Figure 5.4 – Interaction between consumer and SQLite

As represented in Figure 5.4, the messages are consumed, then the required data is iterated from the messages, finally being stored in the SQLite table. This process is done one message at a time, so each message that arrives goes into the iteration process and then gets inserted in the database and only after that the process repeats.
For the development of this project, only a table is used as it appeared to be the most efficient way to work, with only one table. At first, the approach was different, there was one table for every station of that specifically production line as it was an efficient way to separate the messages. Although that solution did not allow the necessary flexibility that was necessary. It was a good and efficient solution for three stations for example, because it was just required three tables but imagining a different scenario of increasing the production line considerably for an example of fifty stations, it would result in fifty tables, turning it into a poor solution.
The solution for this was simply creating just one table on the database side which contains all the messages that are arriving. This process of separating by stations is obviously still an issue at this point but in Tecnomatix a better and more flexible way is implemented to deal with this situation. It will be explained in depth in the Tecnomatix section.

The table that was created is named “StationData” and it is composed by six columns as displayed in Table 5.1.

Table 5.1 – SQLite “StationData” table

ID (Primary Key) (INT) (Auto increment)	**Identifier** (INT) (Not Null)	**State** (INT)	**Timestamp** (Time) (Not Null)	**Datestamp** (Date) (Not Null)	**Station** (INT) (Not Null)

5.4. Tecnomatix

For the simulation of the production line, Plant Simulation Tecnomatix performed great in the intended replications. Tecnomatix is an object-oriented program and as so it allows the possibility of drag and drop of objects that are graphical friendly and overall save time and make it more efficient. This was imperative for the creation of the production line, where a great part of the objects is added and customize accordingly.
For a better comprehension of all the features there were added and created in the simulation, the section is divided in some topics.

- SQLite in Tecnomatix
- Production Line
- Station Data
- Dynamic Creations
- Worker Management
- Data Analysis
- KPIs

5.4.1. SQLite in Tecnomatix

Tecnomatix allows the integration with various information flow objects and among these objects, the SQLite is one of them. The integration of SQLite in Tecnomatix is very straightforward and is explained in this section.
The process for using SQLite makes use of two different objects, the SQLite object, and the method object. SQLite object is perceived as a connection object, and it is responsible for connecting to the database file that is located in a folder on the computer. It is a simple object which does not allow any SQL operations, so it is a single use object that is static and works as a bridge between SQLite and Plant Simulation Tecnomatix.
In order to work and interact with the database and consequently with the other objects, it is imperative the usage of methods which are scripts of code, but which also allow the possibility of using SQL statements by referring to the database object.
In Figure 5.5, there is an example of a SQL query that is performed for the Nexeed MES database as referred in the beginning.

```
-- SQL query to select the last two rows that arrived for the Station10
NexeedMES.prepare("SELECT * FROM StationData WHERE statNo = 10 ORDER BY timeStamp ASC LIMIT 2")
```

Figure 5.5 – SQLite query in SimTalk language

A great number of features being implemented in this simulation model make use of real time data, data located in the database, so these methods are responsible for retrieving the data and then upgrading the objects data.
As for a simple and interactive way for working with the database, a small section, Figure 5.6, was created in the model which allows connecting to the database and performing the two main simple operations, opening and closing the database. The rest of the methods only work if the database is opened beforehand so it necessary to have these easy-to-access features.

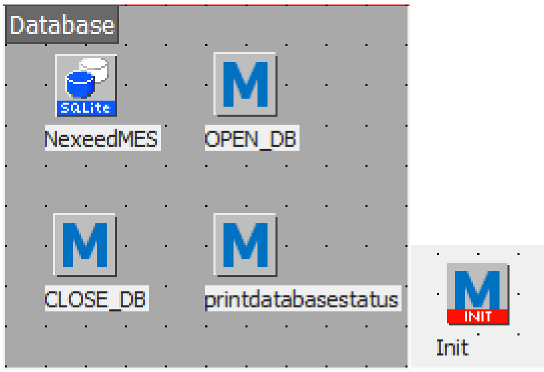

Figure 5.6 – Database integration in Tecnomatix

This is option can be run individually, although, Tecnomatix enables the usage of some special methods which have unique characteristics. One of them is the “INIT” method, that is a differentiated method that runs every time a simulation is started. This method is often used to call methods that are intended to start in the beginning of the simulation as for the example the “OPEN_DB” method, this way every time the simulation is started the database opens automatically without having to do it manually, incurring in the possibility of failure.

5.4.2. Production Line

The simulation model made in Tecnomatix is based on the Line One in Bosch. Overall, it is a line composed of three stations and a single workplace for each of the stations. This means there is only one worker in each station and a single task is performed in each station while the part moves in sequence between stations when the task in each station is finished.
The production line is represented in Tecnomatix with the same configuration as it is represented in real life, though in the simulation it can be very easily changed and upgraded which makes this a flexible approach.
In Figure 5.7, it is possible to observe the production line along with some field of information that refer to the stations.

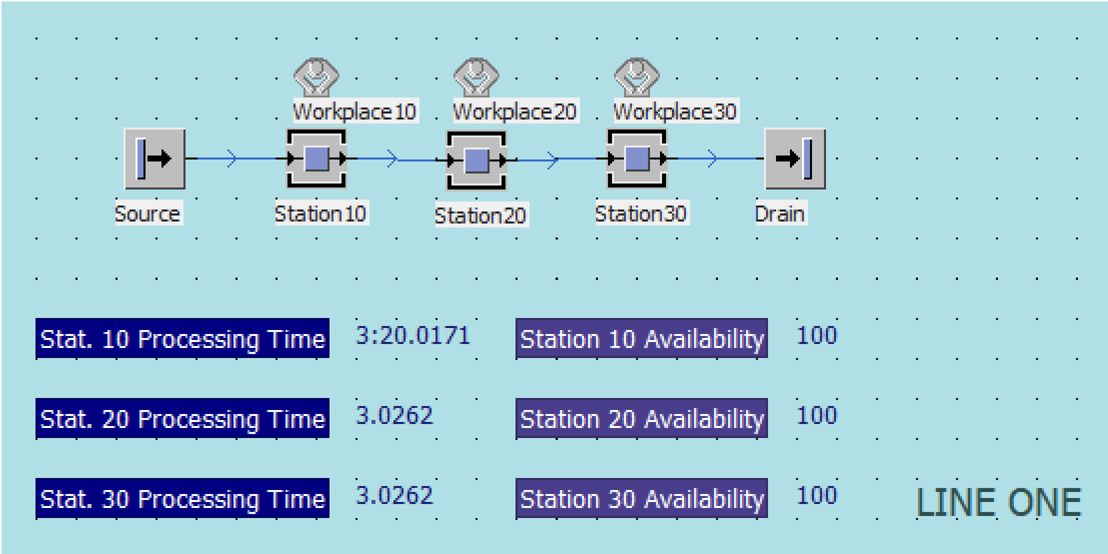

Figure 5.7 – Line One model in Tecnomatix

The production line is composed by a source, from where the parts for manufacturing are arriving, followed by three stations with a workplace attached to each one, which are performing the necessary tasks and preparing the part to move into the next station. Once the part passes the last station, it is prepared and ready to move to the drain.
Based on the data that is arriving from the database, the processing time and availability of each station is being calculated and displayed in the production line area. In the next section, all the calculations and procedures with the data will be explained in depth.

5.4.3. Station Data

Figure 5.8 represents the section that is mainly responsible for the processing of data inside the Tecnomatix. While Tecnomatix allows the integration with SQLite database, it does not allow a great flexibility and intuitive interaction with the data there. To overcome this problem, the data is first introduced in a data table “StationData” with the method “FillTable”, that is an object on Tecnomatix that enables a way simpler way to working with the data.
As for the calculation of the processing time a similar approach is being used although it is quite more complex. Looking back to the way messages are sent, they are sent every time a part leaves a station, and the time stamp associated to each message is the time of which the parts exit a station. In order to calculate the processing time for each station, the solution was dividing the messages into the different stations and working with the times relative to each station individually.
In Tecnomatix a control action was implemented in order to call the function every time a part enters a station. This function will perform a SQL query to the table in SQLite to retrieve the last two messages that were sent from that station and store it in the data table object “UpdateData”. This way it becomes intuitive as it is just necessary to subtract the last message to the previous one in order to calculate the processing time.

As mentioned before, Tecnomatix includes an object-oriented programming language and thanks to this it is possible to see what station is calling the function anytime a part cross there. For example, if a part enters the Station 10, the functions will now assume this and will perform a SQL query to the table and will only retrieve the two last messages correspondent to the Station 10.

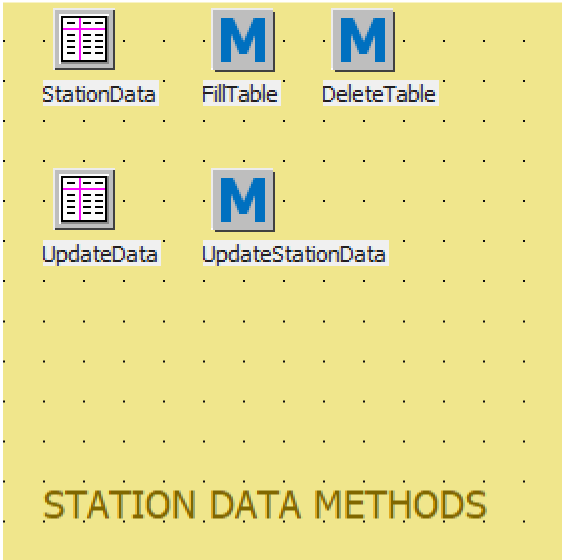

Figure 5.8 – Section for Updating Station Data

The “UpdateStationData” method is then responsible for calculation the processing times by subtracting the last two messages of each station and then converted this information to the formatted time type that is required to update inside the Station 10 object.
This method works perfectly in a real time flow of information because it always recurs to the database to check what are the last two messages of that moment.
In Figure 5.7 it is possible to observe the processing times of each station, they are constantly behind update in real time as a new message is sent from the MES.
Processing time is crucial to know if the stations are working correctly or if there is a problem in any station that is stopping it. By analyzing the model in real time, it is possible to have quick information and then react accordingly.

5.4.4. Dynamic Creations

The process of creating a production line in Tecnomatix every time a simulation needs to be performed is worrying as it is a complex process, and it might take a considerable amount of time to complete it. The worker would need to analyze the data in the database and then build the model accordingly although while in a simple line the data would require simple analysis, in the bigger and complex production line it would be a serious task and it also would not have any flexibility. But what if it was possible to do this automatically, basically Tecnomatix would read the database and then build the model in conformation to the production line data.

Dynamic creations section, Figure 5.9, surges with the intention of fixing the problem and eliminating most of the work while elevating the flexibility considerably. The goal is to create a method that will process the information that is being stored in the database and based on this information it will update the production line model. It will evaluate all the different stations that are sending messages and it will loop through all of them to grant that the simulation model corresponds to the information provided in the table.
The solution implemented is responsible for adding only the stations and workstation to the simulation model. The "DynamicCreation" method performs a SQL query to get all the different stations that are present in all the messages and then stores this station in a table. Then it loops through all the stations in the table and compares to the simulation model to see whether the station is already added or not in the model. If not, the method will add the station and the workplace accordingly to the table which will dictate the correct order of the stations.

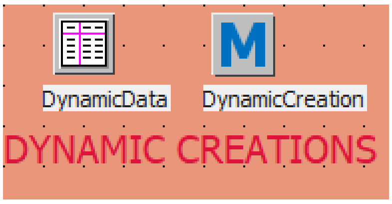

Figure 5.9 – Section for creating objects dynamically.

This method will be running throughout the whole simulation, so it is possible to automatically update the production line if any changes in the data is made. The implementation is prepared for a change in the stations whether it is a different arrangement of the stations, or the number of stations is increased.
For each station, there is a workplace associated and, in each workplace, there is a worker responsible for performing the tasks in that station. So, every time a new station is added or changed, the workplaces also need to be update accordingly so that the numbers of workers is in harmony with the stations.
About the methodology used to create this method, the process is intuitive, but it can be quite complex specially because a change in the production line affect the whole simulation. Similar to the method for updating the station data, this method is run every time a part leaves the station. This way it is possible to check which station is calling the method every time it runs. The first step is comparing the respective station to the order that is dictated by the station order represented in the table "DynamicData". This table is the master, so the production line needs to respect the number of the stations and their respective order. Making this comparison it is possible to check in which place the station needs to be, if in the beginning, in the middle of two stations or at the end of the production line.

5.4.5. Worker Management

The way production lines are displayed in Bosch follow the same methodology and all of them include a workplace and a consequent worker for each individual station. In

summary the number of workers is equal to the number of stations and one station only includes one worker. After each worker performs the task associated to that station, the part moves onto the next station where a new worker will assume and start a new task.
Besides this, additional important information related to workers in the simulation of the production line is the different shifts that the workers perform in order to have an idea of the total working hours and consequent throughput. Bosch does not always have the same shifts through the whole month or year and as so it needs to be presented every week so the workers can follow the schedule.
Nexeed MES already includes tables that contain all the information regarding the different shifts that are set for each week. It includes all the necessary information related to the times of start and end of shifts, the different days it includes and the shift type since it can be early, late and night shift. This information is used to fill the shift calendar, an object of Tecnomatix. A solution is proposed to automate this task and convert the simulation into an even more effortless process. In Figure 5.10 it is represented all the implementation in this section, the worker management. The shift calendar is the object that contain the calendar that represents the different shift. Along with this, the broker and worker poll responsible for the management of the number of workers and their task and lastly the 3 tables and methods used to update the shift calendar.

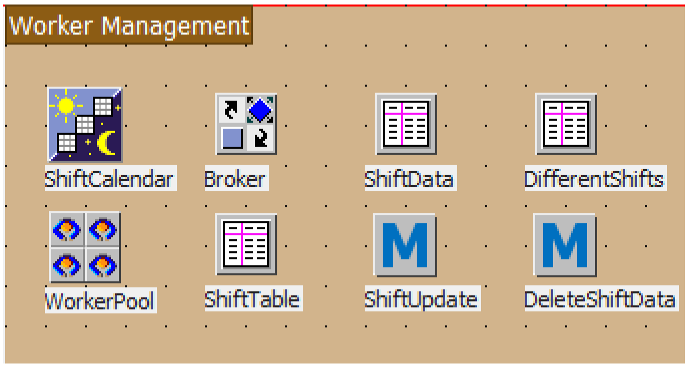

Figure 5.10 – Section for Workers and Shift Management

The method "Shift Update" is responsible for accessing the SQLite table of the shift data and then insert in the data table "ShiftData" in order to facilitate working with this data. It then retrieves the number of different shifts based on the "SHIFT_ID" and stores in the data table "DifferentShifts". In order to edit the shift calendar, the table containing the shift data, Figure 5.11 ,is first copied into a data table," ShiftTable", to facilitate the access to all the different fields. As the information provided in the tables of the Nexeed MES does not correspond exactly to the shift calendar format, it is necessary to process this data beforehand. With the number of different shifts and their schedule and workdays it is finally possible to process everything and fill the "ShiftTable". After the information is updated, the copy created initially is again converted to the format of the shift calendar table. This operation can be performed every time the user required, and it will check the database to see the shift calendar in that week and update in Tecnomatix. The only information field that is not filled is the shift pauses as there is no information in the

Nexeed MES about it. This calendar shift is connected to each workplace so that the workers perform the work during the stipulated time of the shift calendar.

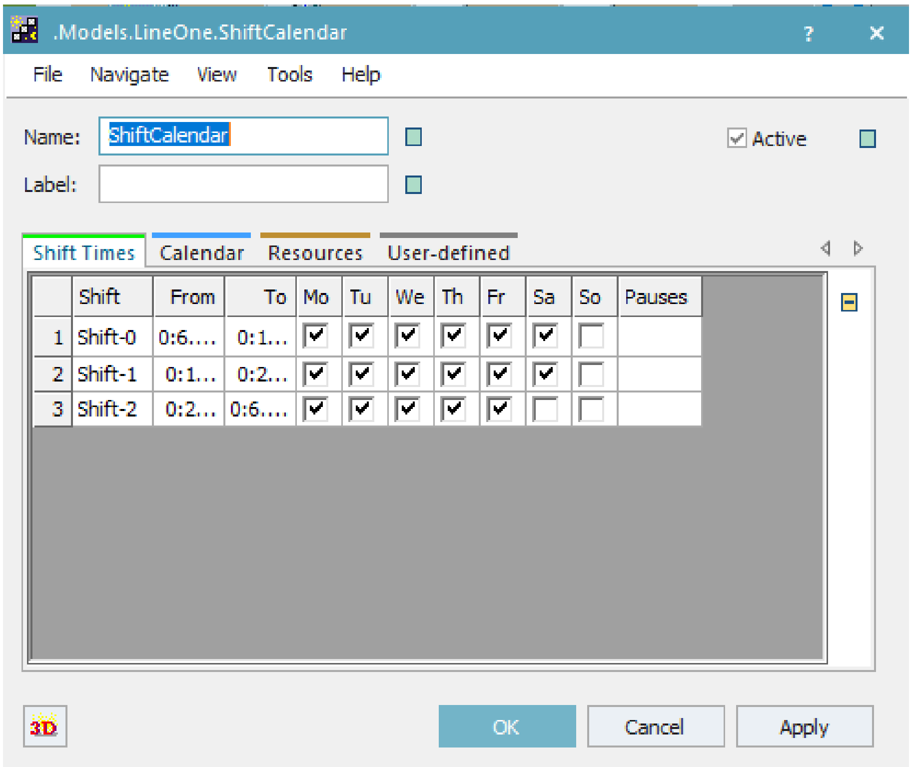

Figure 5.11 – Shift Calendar Table

5.4.6. Data Analysis

Analyzing the performance of the production line is not an intuitive task without the help of any indicator or graphs that include an overview of every object. Tecnomatix has this covered by the integration of some objects that allow the creation of html reports and display of charts.

The chart objects enable the monitorization of various objects and one of them is the stations, which is imperative to monitor as it is the main composition of the simulation model. For the development of this project, the chart object is connected to the stations that are present in the production line and it allows the monitorization of working, set/up, pause, blocked, waiting, stopped, failed and unplanned moment throughout the whole processing flow.

Real time analysis of bottlenecks and production line flow often requires extensive data analysis of data in tools such as Excel, although by implementing this analysis in Tecnomatix it is possible to get this information in a much more efficient way. The data is constantly being updated and so are the chart of the stations. Additionally, at the end of every Simulation an HTML report is made containing a summary of the overall simulation performance. These information sources are essential for the worker to get a wide overview of what is happening in real time in the production line rather that needing to collect all the data and then making the calculations individually and manually.

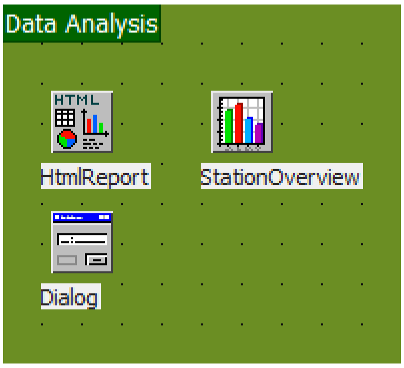

Figure 5.12 – Data Analysis Section

5.4.7. KPIs

This section, Figure 5.13, is created in order to create methods for calculating key performance indicators that can be displayed around the simulation. With the data provided in the databases it is possible to calculate the availability of each station which is their efficiency. This metric is calculated based on the state value that is present on the "StationData" table. This value indicates whether a part is OK if the state is equal to one or if the part is not OK in case the state is equal to zero. This model will be running in real time and whenever it runs, it will retrieve all the data in the database until that moment regarding the state and will calculate the efficiency based on the comparison between the Ok and not Ok parts. This value is not displayed inside this section, but instead it will be available to monitor in real in the section of the production line, along with the processing time.
The second metric being measured is the throughput and this one is not calculated based on the data of the database but instead it is an internal Tecnomatix metric that enables the calculation of the throughput until the moment at which the method in run.

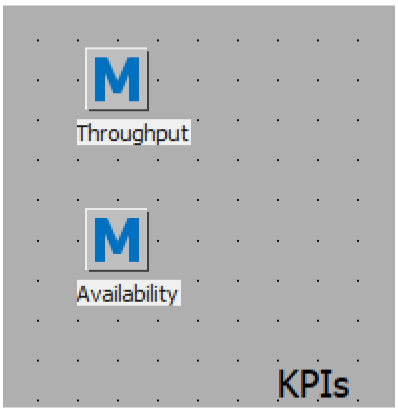

Figure 5.13 – KPIs section

6. Case Study Analysis

As a complement for the implementation proposed, in this chapter it is presented a hypothetical case study that will allow a better understanding of the software advantages in a real time scenario of a company application. Additionally, along the explanation of the case study, the advantages of the implementation will be explained as opposed to the normal, or rather manual method that is implemented for this case study.
An example of a hypothetical production line is represented and explained in the beginning, and even though it does not represent any real line, the model is similar and all the analysis and variables in question are applicable. Next, some graphs are shown, that are a useful method of analysis of the workflow inside the stations and the workers efficiency. These graphs allow the analysis of various problems such as analyzing bottlenecks, excessive waiting times, etc. This will be explained with more detail in the section 6.3 with an analysis of the graphs regarding the case scenario aborded.
Lastly, a representation of some KPIs that are analyzed, important metrics that complement the simulation and allow the possibility of comparison between different scenarios as it will be shown. With all this tools it is possible to build a good foundation of data to monitor any production and compare different case scenarios while evaluating their performance therefore allowing faster improvements with less time dispended.

6.1. Production Line Scenario

In Figure 6.1 there is an example of a hypothetical production line that was created for this case study. The most important to notice in this example is the displacement of the production line and its interaction with the workers.
It starts from the bottom left, with the source of material, which basically is the place where the part is located in its initial state. Following, from the left to the right, there is a series of stations and buffers that are responsible for mounting all the different components in the part until its final state. Once the part is finally mounted there is a station in the end, the bigger one, that is responsible for making some performance tests to analyze its reliability before moving to the packing phase.
The top part is composed also by a series of stations and buffers that are responsible for packing the product and preparing it before moving them to the drain.
Important to understand that each of the stations has one workstation and while each station only has one worker at a time, this worker can work in more than one station. For example, the station of the bottom and top that are close, they can share the same worker, but this will be explained in more detail in the graph analysis.

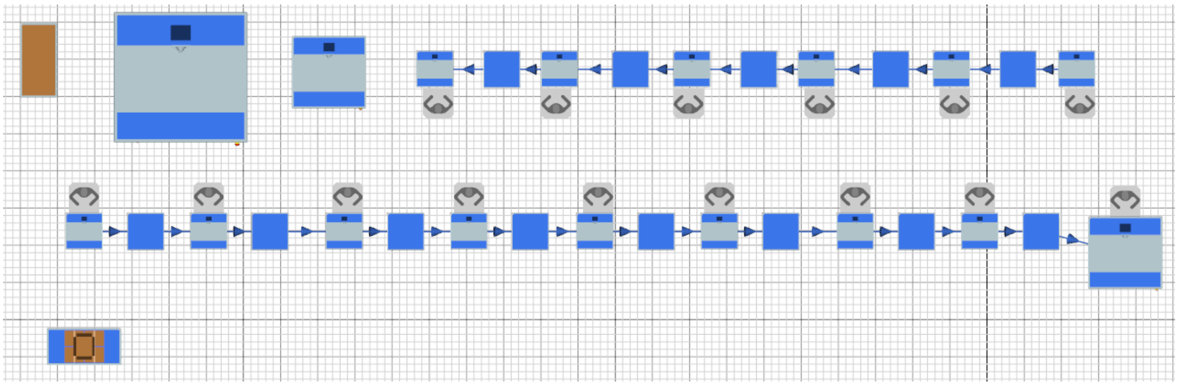

Figure 6.1 – Hypothetical example of a production line

- Stations in serie.
- U layout format.
- One workstation per station.
- One worker can operate in multiple stations.

Here we can see the first advantage of the implementation made in this project. As the case study is made manually, in case we wanted to add new stations or workstations, we would need to do this manually, also needing to add the information about the stations to the simulation data. Although with the implementation made, the model is capable of analyzing the information prevenient from the database and add the station and workstations, accordingly, saving considerable time and turning this process effortlessly.

6.2. Station Data

In order to make it possible to analyze the model, it was imperative to have some data to work with. This process was again done manually, by using a station data object inside Plant Simulation Tecnomatix that works basically as an excel sheet. Although this process is not efficient as depending on how much data we want to add to the simulation it might take an excessive number of hours. Here we can notice another big advantage of the implemented solution, as we are able to do this process automatically, by retrieving the data in real time directly from the Nexeed MES and instantly filling the station data.
Still, for this case study, it was added information regarding the set-up and processing times of each station as these values will be indispensable for the calculation of KPIs and the analysis of the production line efficiency. These times are divided in two different scenarios, the mono and the split. The mono scenario assumes that all the stations produce the same part without any changes being necessary in the production flow. On the other hand, the split scenario assumes that some stations will have to perform a different variation of the task, as it is possible to observe in Table 6.1, that some processing times differ between the two scenarios, increasing the processing time in the split scenario.

Table 6.1 – Processing time different in the two scenarios

Stations	**Processing Time (Mono)**	**Processing Time (Split)**
Station (Line 1)	3.4850	6.5850
Station (Line 7)	46.0000	1:15.4640

Furthermore, some additional information was added regarding the different workers and their attribution to the different workstations and stations. This simulation will be running for one week and the working hours respect the shifts that the Bosch company makes. In the case of this case analysis, the shifts are set manually, and the calendar of shifts needs to be updated weekly whenever a new shift calendar is available on MES. Thanks to the implementation made in this project, this process got drastically improved as it is now possible to make this process automatically by retrieving the table of the shifts weekly from the MES and automatically updating the shift calendar inside Tecnomatix without needing to add any value manually.

object 0	string 1	time 2	time 3	time 4	boolean 5	string 6	object 7	boolean 8	real 9
Station	Label	ProcTime_Mono	ProcTime_Split	SetupTime	Worker	Workplace	WorkplaceObject	Walking	FPY
station_kIB1QOnj0MnVCxEtBF1Bgj0MnVC	[illegible]	3.4850	6.5850	0.0000	true	1	*.Models.MidAutomation.Workplace1	true	100.00
station_kIB1QOnj0MnVCxVvBF1Bgj0MnVC	[illegible]	1:02.3400	1:02.3400	0.0000	true	1	*.Models.MidAutomation.Workplace2	true	100.00
station_kIB1QOnj0MnVCxwhBF1Bgj0MnVC	[illegible]	1:03.2240	1:03.2240	0.0000	true	2	*.Models.MidAutomation.Workplace3	false	100.00
station_kIB1QOnj0MnVCBFjBF1Bgj0MnVC	[illegible]	1:05.1800	1:05.1800	0.0000	true	3	*.Models.MidAutomation.Workplace4	false	100.00
station_kIB1QOnj0MnVCBZIBF1Bgj0MnVC	[illegible]	1:00.6960	1:00.6960	0.0000	true	4	*.Models.MidAutomation.Workplace5	false	100.00
station_kIB1QOnj0MnVCBtnBF1Bgj0MnVC	[illegible]	1:01.4900	1:01.4900	0.0000	true	5	*.Models.MidAutomation.Workplace6	false	100.00
station_kIB1QOnj0MnVCRBpBF1Bgj0MnVC	[illegible]	46.0000	1:15.4640	0.0000	true	6	*.Models.MidAutomation.Workplace7	false	100.00
station_kIB1QOnj0MnVCRVrBF1Bgj0MnVC	[illegible]	1:05.2600	1:05.2600	0.0000	true	7	*.Models.MidAutomation.Workplace8	false	100.00
station_kIB1QOnj0MnVCRptBF1Bgj0MnVC	[illegible]	1:03.0660	1:03.0660	0.0000	true	8	*.Models.MidAutomation.Workplace9	false	100.00
station_kIB1QOnj0MnVCR9vBF1Bgj0MnVC	[illegible]	40.3600	40.3600	0.0000	true	9	*.Models.MidAutomation.Workplace10	false	100.00
station_kIB1QOnj0MnVChVhBF1Bgj0MnVC	[illegible]	1:05.1520	1:05.1520	0.0000	true	10	*.Models.MidAutomation.Workplace11	false	100.00
station_kIB1QOnj0MnVChpjBF1Bgj0MnVC	[illegible]	55.2000	55.2000	0.0000	true	11	*.Models.MidAutomation.Workplace12	false	100.00
station_kIB1QOnj0MnVCh9lBF1Bgj0MnVC	[illegible]	45.0800	45.0800	0.0000	true	12	*.Models.MidAutomation.Workplace13	true	100.00
station_kIB1QOnj0MnVCxRnBF1Bgj0MnVC	[illegible]	35.0000	35.0000	25.4200	true	13	*.Models.MidAutomation.Workplace14	true	100.00
station_kIB1QOnj0MnVCxlpBF1Bgj0MnVC	[illegible]	39.2400	39.2400	28.7600	true	13	*.Models.MidAutomation.Workplace15	true	94.00
station_kIB1QOnj0MnVC7XnBVlKj0MnVC	[illegible]	21.7800	21.7800	0.0000	true	12	*.Models.MidAutomation.Workplace16	true	100.00
station_kIB1QOnj0MnVCx5rBF1Bgj0MnVC	[illegible]	1:01.0800	1:01.0800	0.0000	true	14	*.Models.MidAutomation.Workplace17	false	100.00
station_kIB1QOnj0MnVCBOtBF1Bgj0MnVC	[illegible]	1:04.3460	1:04.3460	0.0000	true	15	*.Models.MidAutomation.Workplace18	false	100.00
station_kIB1QOnj0MnVCBivBF1Bgj0MnVC	[illegible]	1:00.1040	1:00.1040	0.0000	true	16	*.Models.MidAutomation.Workplace19	false	100.00
station_kIB1QOnj0MnVCB6hBF1Bgj0MnVC	[illegible]	1:05.1400	1:05.1400	0.0000	true	17	*.Models.MidAutomation.Workplace20	false	100.00
station_kIB1QOnj0MnVClTpBF5Brj0MnVC	[illegible]	15.0000	15.0000	0.0000	false		*.Models.MidAutomation.Workplace21	false	100.00
station_kIB1QOnj0MnVCROjBF1Bgj0MnVC	[illegible]	50.0000	50.0000	0.0000	false		*.Models.MidAutomation.Workplace22	false	100.00
station_kIB1QOnj0MnVCRilBF1Bgj0MnVC	[illegible]	15.6600	15.6600	0.0000	true	11	*.Models.MidAutomation.Workplace23	false	100.00
station_kIB1QOnj0MnVCR2nBF1Bgj0MnVC	[illegible]	1:00.0000	1:00.0000	0.0000	false		*.Models.MidAutomation.Workplace24	false	100.00
station_kIB1QOnj0MnVChKpBF1Bgj0MnVC	[illegible]	12.0000	12.0000	0.0000	true	9	*.Models.MidAutomation.Workplace25	false	100.00

Figure 6.2 – Hypothetical data for the production line

6.3. Graph Analysis

Plant Simulation Tecnomatix allows the visualization of charts, and these charts can be connected the objects inside the simulation such as stations, workers, etc. In this chapter it is analyzed the chart of the stations and the chart of the workers of two different scenarios, a first one without failures and a second one, more realistic, with failures. These charts are a perfect solution for analyzing the workflow of the production line as they dictate the times of every activity throughout the simulation, for example working times, pauses, blocked times, etc. By combining this chart with the chart of the workers there are many conclusions that can be taken in order to verify important information such as where the bottlenecks are occurring and ways of eliminating unwanted stopped or blocked times, as well as increasing the working time of the workers.

In Figure 6.3 there is an example of the two charts side by side after a simulation that ran the production of one entire week, and in this simulation, it simulates a hypothetical case of no failures in the station of performance tests. Starting with the chart of the station statistics we can see the different colors that correspond to different occurrences during the simulation. Starting with the green color it represents the time where the station is

working, the yellow corresponds to the blocked time, the dark blue corresponds to the pauses of each shift, the brown corresponds to the set-up times, the red to fail occurrences and lastly the blue to unplanned time.
The pauses represented in dark blue, are constant through all the station as it cannot be avoided. These are small breaks that occur during the shifts, and they are planned and expected. The brown bars correspond to the stations that make performance test and these stations do not require the worker to operate during the whole task, but on the other hand they require an initial and end set-up time to prepare the test.
By analyzing the worker statistics chart, it is also possible to verify the same displacement of bars that have the same representation by colors. The worker thirteen is represented essentially by a brown bar dictating that his only task is to set-up the station. Based on this is it possible to assure that this worker is the one associated to the performance tests station as the other stations do not required set-up times.
By analyzing the charts, it is possible to notice some peculiarities. For example, in the stations chart it is possible to see that until station A there is always blocked time and this time ends in station A. On the other hand, station A has a lot of waiting time which does not make sense because if it is waiting so much time why is it blocking the other stations before. The conclusion for this is related to workers, by looking at the worker statistics it is noticeable that the worker twelve which is responsible for operating in the station A is the most occupied and also it has some time where he is moving. This moving time happens because this worker is operating in two different stations and due to the time he takes in the other station and the time he takes to then walk until station A, it leaves this station with a lot of waiting time and therefore blocking all the stations beforehand. The same occurs with all the stations before D that have blocking time and the station D has a lot of waiting time. Station D also shares a worker with another station and this process of walking between stations leaves this station waiting, blocking the previous ones.

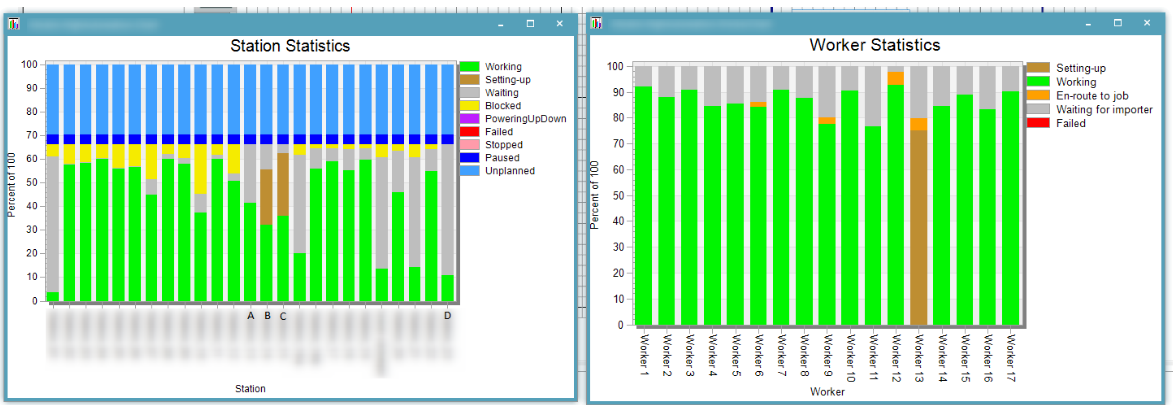

Figure 6.3 – Relation between station statistics and worker statistics without failures

In Figure 6.4 there is an example of the same graphs but in a different scenario of simulation, whereas in this case the station responsible for the performance tests does not have one hundred per cent efficiency, therefore producing failures.
The red bar, representing the failures, corresponds to the station of performance tests that is placed at the end of the production line. Whenever this station fails a performance

test, it needs to repeat the test again taking additional time. Comparing these charts to the previous one we can see some differences. When looking at the blocked stations in the beginning it is possible to observe that now it is blocking until the A station and not only until the station before A. This happens because now with the failures, the workers take more time to complete the tasks in the testing station, therefore originating blocks also in the previous station, A.
Another important conclusion that can be taken from this chart is that even though there is some failure time, the waiting time of station B is still higher than these failures meaning that even with the station failing, it stills need to wait for parts to arrive. This means that the impact of this failures in the productivity will not be very noticeable as these failures will not occupy the entirety of the waiting time.

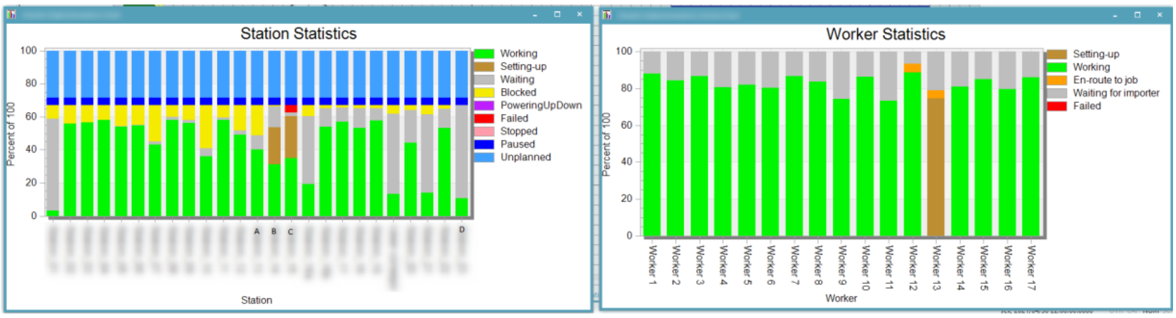

Figure 6.4 – Relation between station statistics and workers statistics with failures

6.4. KPI Analysis

For a better analysis of the production line efficiency and in order to have a foundation to compare between different scenarios, it was implemented the calculation and listing of some KPIs that enunciate an overview of the whole simulation. In this section, two different lists of KPI will be analyzed, one for a hypothetical case scenario of 100% efficiency of the station responsible for the performance tests and one of an optimistic scenario of 97% efficiency of the same station. This could have been tested for more scenarios such as 94% efficiency which would be a more realist one or even 90% efficiency imagining a pessimist case scenario. Still with these two cases analyzed, it is possible to understand the impact of the efficiency in the results after a week of simulating the production.
Before comparing the two different scenarios it is important to understand the meaning of each metric that was calculated and analyzed.

- **Planned Cycle Time**

Planned Cycle Time represents the expected time that each part exits the production line. This time is the perfect case scenario although it is not the real time as we need to do the adjust of the first part that takes more time.

- **First Part Time**

This is the time represent the throughput time of the first time. This time is faster than the throughput time because in the first part there is not blocking in the production line yet.

- **Throughput Time**

The throughput time is the time the part takes since the first station to the final station.

- **DT [time/pc.] (Ramp-Up-Adjusted)**

This time represents the planned cycle time but in the real scenario of the simulation. It is calculated by the time of simulation diving by the number of parts.
Ramp adjustment -> (Simulation Time – First Part Time) / Number of parts. This adjust needs to be made because just for the production of the first part there was a waiting time longer than the next ones.

- **OEE [%] (Ramp-Up-Adjusted)**

OEE (Overall Equipment Efficiency) indicated as the name suggest the global efficiency. It is the relation between the cycle time and the simulation time. The simulation time is also ramp adjusted.

- **Part Good [pcs.]**

This metric indicates the number of Ok parts produced at the end of the simulation.

- **TEK [pcs. /Hour] (Ramp-Up-Adjusted)**

TEK (*Technische Kapazität*, technical capacity) indicates the number of parts per hour with the adjust of the first part. It is calculated by number of parts / (Simulation time – Time of the first part) in hours.

- **Balance Efficiency [%]**

Balance Efficiency indicates the mean of occupation of the workers.

- **Line Efficiency [%]**

Line Efficiency represents the mean of occupation of the stations.

After comparing both scenarios analyzed, the hypothetical one with 100% efficiency in Figure 6.5 and the optimistic one with 97% efficiency in Figure 6.6 it is possible to notice some disparity in the results, as expected.
For example, there is an increase in the throughput time and a decrease in the values of efficiency in the 97% scenario when comparing to the 100%, and also it is possible to see that with just 3% of difference in the efficiency it resulted in almost 200 hundred less parts. From this we can conclude that the efficiency of the station of performance tests

has to be the main focus on the production line, because with a minor change it affects the results of productivity heavily.
Another important factor to analyze is the first part time and the influence it has on the disparity of the planned cycle time and the DT. While the planned cycle time dictates the cadence of production, this time of one minute and five seconds is hypothetical because this time is only possible to achieve without any blocking in the stations and only after the first part. The first part time is twenty-two minutes which means that to get the first part done it is necessary twenty-two minutes as opposed to the following parts that it is just required one minute and five seconds. The DT is the planned cycle time but in the real time scenario of the simulation with station blockings and this value needs to be ramp-up-adjusted by subtracting the first part time to the simulation time.
In order to have a quick global result of the efficiency of the production line it is possible by analyzing the Overall Equipment Efficiency (OEE). For instance, if various adjusts are being made and it is necessary a fast way of analyzing their result, this can be done by looking at the OEE and have a look at the global efficiency of the production and see whether the changes are improving or instead reducing the efficiency.

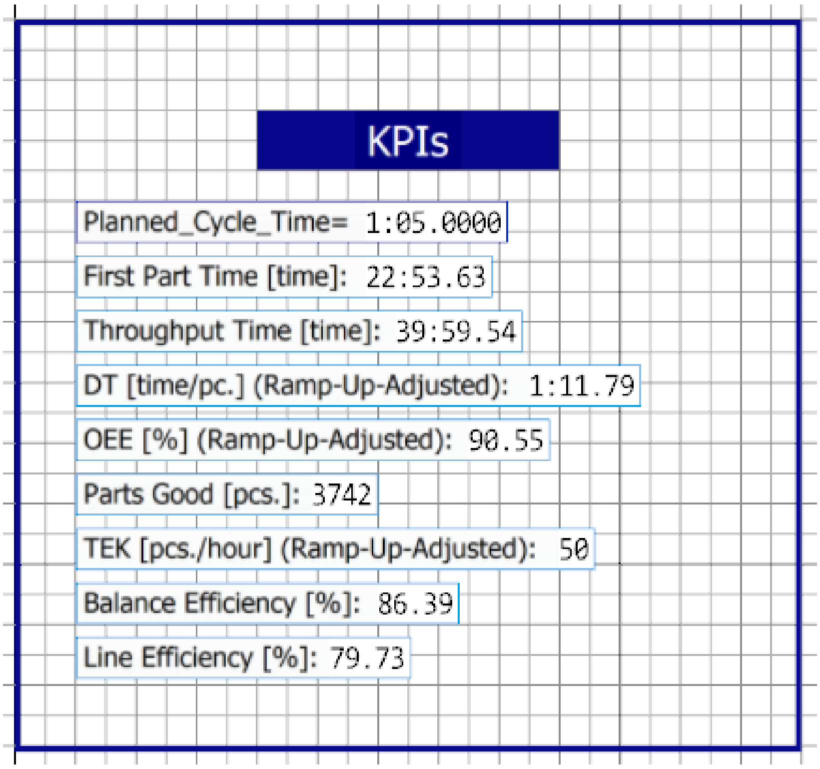

Figure 6.5 – KPI analysis for 100% efficiency

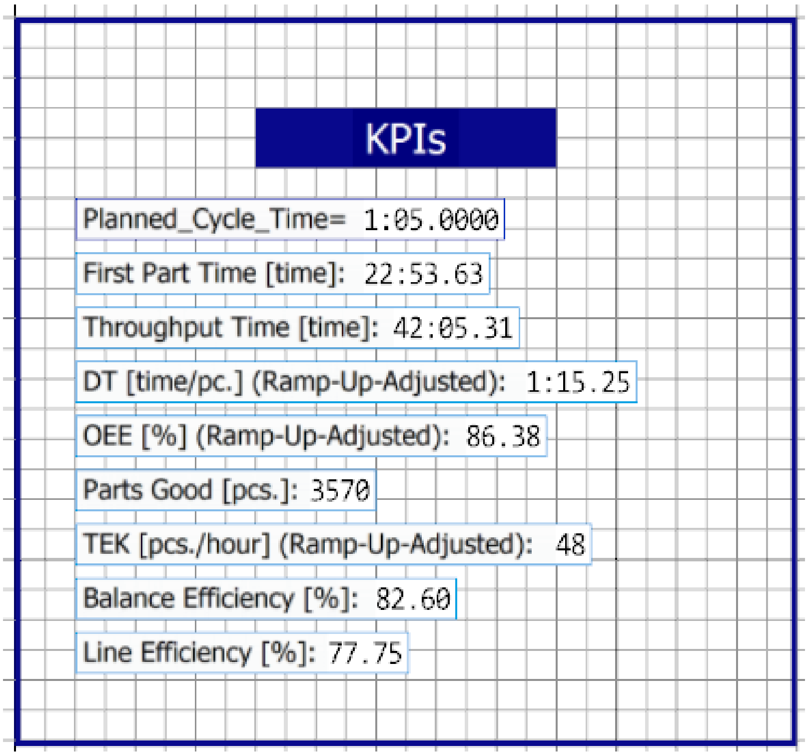

Figure 6.6 – KPI Analysis for 97% efficiency

In summary, it is possible to see the applications of Plant Simulation Tecnomatix and how we are able to monitor and test different scenarios inside the simulation of this case study. These results will enable engineers to analyze the result and decide where there is margin for improvement and where the biggest failures are occurring.
Although as we can see, the process of creating this simulation and specially transferring the real data to the model can be often an excessive time-wasting process, and this is where the implementation of this thesis brings huge improvements and a big leap in the simulation field.
This process of inserting data into the simulation, and most importantly real time data from the production line, is now possible to be done automatically saving hours of unnecessary work. Adding to this, there is also the advantage of being able to calculate this KPIs in real time as the data is being added, instead of needing to calculate them manually every time we want to add new values. Also, the feature implemented for the dynamic creation enables the possibility of creating objects automatically by analyzing data that is being retrieved from the databased as opposed to the way that was made in this case study which is implementing them manually requiring the extensive analysis of excel sheets or a big data set for that. By complementing the implementation created with this case scenario, it is possible to have the same or better results in more efficiency and fast approach.

7. Conclusion

7.1. Conclusions

Throughout the development of this project, it was possible to get a deep understanding on the evolution of Industry 4.0 and how simulation might affect this concept. The concept of Digital Twin is complex and to be able to fully create one is not an easy task, as it involves a huge number of factors. Yet more and more companies are making that effort and investing their time in creating and exploring this concept.

Every company that aims for evolution and increased digitalization proposes the standardization of all the processes and machines because this opens doors for a straightforward integration and grants full harmony between processes, machines, and humans. Still everyday a new idea surges and sooner than expected engineers notice themselves coming up with new and innovative ideas that might or not be an improvement to the actual state of things. However, this leads to a problem which is how it will be possible to test this idea because it involves changing the whole ongoing process resulting in losses in production, stopped machines and unwanted waste of money and time.

This is where this project takes place, a solution for the digitalization of the real world with the intention of solving the problem mentioned above and proportionate additional flexibility and possibility for engineers to test their ideas in an easier and cost-free approach.

It was proposed an integration between a production line simulation model and the shop floor support software, the Nexeed MES. This integration intends to create the Digital Twin, the ultimate goal for virtual digitalization, a simulation with real time data flow, in simple words, a virtual twin of the production line. The advantages are clear, real time monitorization of the production line, easy accessibility information, immense flexibility and most importantly the absence of costs associated to all these operations.

This project created a significant foundation for engineers to keep developing this simulation and increase its efficiency for maximum utility. The main focus of this dissertation was the integration of the simulation model of line one with the real time data from Nexeed MES and that is successfully implemented along with many other features.

A data model was created with essential information for the creation of a simulation of the production line. The production line, more especially line one, is successfully implemented in Plant Simulation Tecnomatix as well as the integration with the real time data flow from the Nexeed MES. This last one was not possible to access due to privacy reasons, although this problem was solved with the creation of a solution that was building an application to perform the same way as the MES. This is not a simulation of the whole MES, but only of the message processing between the MES server and the Apache Kafka message broker, and it is important to understand the simulating application is sending the same messages exactly as the MES, in other words, the concrete data of the shop floor and not some simulated values. This made a great improvement in the simulation as before engineers were using excel sheets to store the data and then transferring it manually to the simulation model, whereas now engineers already have the bridge between the simulation model and the real time data source, Nexeed MES, not needing to waste the unnecessary and excessive time because it will be done automatically.

The solution is fully working, and it features integration with a database that is storing MES data in real time. This feature is the base of the Digital Twin concept as it grants the access to information in real time inside the simulation. It is not possible to directly integrated MES with Tecnomatix, although using a database is an efficient and intuitive solution that works well in the desired conditions. As a consequence of this integration, several other features were developed in order to increase virtualization of the production line. Additionally, an interface was developed for the simulation model allowing it to be capable of accessing all the information stored in the database tables and transfer them into local tables inside Tecnomatix, which facilitates the visualization and data manipulation inside Tecnomatix. The production line represented by a source, stations, workplaces, and a Drain is being updated in real time with some important information regarding production indicators (KPIs), that are measurement values of a production line. These include the processing time of each station and the efficiency of each station.

Along with stations, this model also includes worker management, including shift adaptations and changes. It is possible to automatically define the shift calendar and update it weekly whenever it changes, thanks to the integration with the database, so that the workers and the stations works in conformity with the number of hours of labor of Bosch in this case.

In the present model is also possible to automatically add different objects such as stations and workstations without any need of contact with the simulation model if the data in the database is updated with new information regarding the addition of a new station/s. This feature is essential for the automation of the model. It also adapts everything related to this change since the number of workers to the inclusion of this object values in calculation of Key performance indicators that are directly related to the stations.

In the case study analysis, it is possible to verify the impact that this implementation has as opposed to the traditional manual method of inserting and working with the data. While previously, these simulations were performed with the help of Excel and inserting the data individually inside Tecnomatix and taking excessive time, it is now possible to perform this automatically providing great scalability to the simulation models and saving considerable time for engineers. Also, KPIs in the implemented solution are being calculated automatically in real time without needing to change the values manually every time we want to change the data of the simulation, and before this process was not possible, reducing the simulations flexibility.

In summary, this project was developed with maximum focus on the integration of the simulation model of line one with the real time data source, Nexeed MES, as well as real time monitoring of indicators. These two aspects are successfully implemented and working and the door for scalability is open, which means that from this standing point, the difficulty in evolving this model in another direction such as performing real time simulations of shop floor scenarios just got drastically reduced, as well as the necessary time for simulating them.

7.2. Future Work

At this point there is a great development in the project, but there is always room for improving and upgrades that can be developed, as Leonardo da Vinci once said, "the art is never finished, just abandoned". By upgrading this model in the future, it is possible to have a positive impact in the efficiency and reliability of the simulations.
The first next step that must be considered is moving the model from the student version to the enterprise version, so all the capabilities of the software are available and its possible to get maximum benefit of it, including the unlimited number of objects per simulation as opposed to the limited eighty objects of the student version. This include transferring the database from the SQLite to the Oracle that is currently implemented in Bosch and therefore have a significant increase and optimization in the information source. This option was not available during this project because the student license only allows the integration with the SQLite database, still there was a huge effort put into making this process of transition the smoothest possible and therefore the process is very simply not requiring any big changes in the adaptation. Also, there are some other features such as socket objects that allow real time alarms inside Plant Simulation Tecnomatix and that was also not possible to implement in the student version.
Also, in the feature developed in the simulation model for dynamic creation of objects based on real time data, the layout of the production line, the "U" layout, is not fully accomplished when adding new stations and workstations, and the future step here is including more information about the location of each station, so it is not only possible to see if the station comes after of before when comparing to another but also if it comes on the top, bottom, etc., of the production line. Furthermore, another interesting feature is the automatic deletion of objects, basically the simulation model is automatically adding the objects accordingly although it is not optimized to delete the objects from the model if by any case the information from that object disappears from the database tables.
The future work will be essentially increasing the complexity of the simulation model and aim to approximate the maximum as possible from the real production line. There is still a lot that can be done since the calculation of different key performance indicators, monetarize many different data values (such as different times for instance), etc.
Along with this, it is also interesting the future integration of machine and process information as now the simulation implemented is a more general approach, in other words simulating not only the layout and station configuration but also bring the detail down to the process level and open doors also for process improvement and monitorization.
Lastly, the dissertation provided a significant foundation for engineers. Bottleneck analysis in real time, analysis of a potential improvements in a production line and other continuously improvement projects and use cases can be done to test the performance of what was developed.

References

[1] L. Barreto, A. Amaral, and T. Pereira, “Industry 4.0 implications in logistics: an overview,” *Procedia Manuf.*, vol. 13, pp. 1245–1252, 2017, doi: 10.1016/j.promfg.2017.09.045.

[2] C. Cimino, E. Negri, and L. Fumagalli, “Review of digital twin applications in manufacturing,” *Comput. Ind.*, vol. 113, p. 103130, 2019, doi: 10.1016/j.compind.2019.103130.

[3] M. F. Körner *et al.*, “Extending the automation pyramid for industrial demand response,” *Procedia CIRP*, vol. 81, pp. 998–1003, 2019, doi: 10.1016/j.procir.2019.03.241.

[4] P. D. Urbina Coronado, R. Lynn, W. Louhichi, M. Parto, E. Wescoat, and T. Kurfess, “Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system,” *J. Manuf. Syst.*, vol. 48, pp. 25–33, 2018, doi: 10.1016/j.jmsy.2018.02.002.

[5] E. Negri, S. Berardi, L. Fumagalli, and M. Macchi, “MES-integrated digital twin frameworks,” *J. Manuf. Syst.*, vol. 56, no. April, pp. 58–71, 2020, doi: 10.1016/j.jmsy.2020.05.007.

[6] E. Arica and D. J. Powell, “Status and future of manufacturing execution systems,” *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2017-Decem, pp. 2000–2004, 2018, doi: 10.1109/IEEM.2017.8290242.

[7] S. Mantravadi and C. Møller, “An overview of next-generation manufacturing execution systems: How important is MES for industry 4.0?,” *Procedia Manuf.*, vol. 30, pp. 588–595, 2019, doi: 10.1016/j.promfg.2019.02.083.

[8] A. Bratukhin and T. Sauter, “Execution System Distribution,” *Ieee Trans. Ind. Informatics*, vol. 7, no. 4, pp. 740–749, 2011.

[9] O. Brauckmann *et al.*, *Manufacturing Execution Systems – MES*, 2007th ed. Mosbach: Springer, 2007.

[10] D. Brandl, “MES vs . ERP for manufacturing,” *IT Eng.*, pp. 1–15, 2011.

[11] S. M. Lee, S. G. Hong, P. Katerattanakul, and N. R. Kim, “Successful implementations of MES in Korean manufacturing SMEs: An empirical study,” *Int. J. Prod. Res.*, vol. 50, no. 7, pp. 1942–1954, 2012, doi: 10.1080/00207543.2011.561374.

[12] Z. Technologies, “What Is The Difference Between MES and ERP For a Life Sciences Manufacturer? By Keith Bowen,” 2014. https://www.pharmaceutical-technology.com/contractors/process_automation/zenith-technologies/pressreleases/pressmes-erp-manufacturer/ (accessed Mar. 03, 2021).

[13] H. Santos, “A diferença entre o MES e o ERP,” 2016. https://flowtech.pt/pt/blog/diferenca-mes-erp/ (accessed Mar. 03, 2021).

[14] Robert Bosch GmbH, “NEXEED - Software and Services for Production and Logistics,” pp. 1–17, 2018, [Online]. Available: https://assets.bosch.com/media/en/global/products_and_solutions/connected_products_and_services/industy_40/bosch-connected-industry-brochure.pdf.

[15] Robert Bosch GmbH, “No Title,” 2018. https://www.bosch-india-software.com/en/products-and-services/digital/factories-of-the-future/nexeed-mes/ (accessed Mar. 22, 2021).

[16] Robert Bosch GmbH, “No Title,” 2018. https://www.bosch.com/stories/nexeed-production-performance-manager/ (accessed Mar. 22, 2021).

[17] mpdv, “HYDRA X,” p. 17.

[18] “MPDV.” https://www.mpdv.com/en/ (accessed May 27, 2021).

[19] E. Industrial, “José Nuno Machado da Silva Ferreira PLANEAMENTO E CONTROLO DE PRODUÇÃO NUM SISTEMA SAP R / 3 José Nuno Machado da Silva Ferreira PLANEAMENTO E CONTROLO DE PRODUÇÃO NUM,” 2020.

[20] “SAP MES.” https://www.sap.com/products/execution-mes/features.html (accessed May 27, 2021).

[21] “Siemens MES.” https://www.plm.automation.siemens.com/global/pt/products/manufacturing-operations/manufacturing-execution-system.html (accessed May 27, 2021).

[22] S. Bangsow, *Manufacturing Simulation with Plant Simulation and Simtalk*. Munich: Springer, 2010.

[23] Y. Rao, F. He, X. Shao, and C. Zhang, “On-Line simulation for shop floor control in manufacturing execution system,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5315 LNAI, no. PART 2, pp. 141–150, 2008, doi: 10.1007/978-3-540-88518-4_16.

[24] “Simio.” https://www.simio.com/software/simulation-software.php (accessed May 13, 2021).

[25] “Simul8.” https://www.simul8.com/software/ (accessed May 13, 2021).

[26] “Anylogic.” https://www.anylogic.com/.

[27] “Plant Simulation Tecnomatix.” https://www.plm.automation.siemens.com/global/pt/products/manufacturing-planning/plant-simulation-throughput-optimization.html (accessed May 13, 2021).

[28] “Ansys.” https://www.ansys.com/products/digital-twin/ansys-twin-builder (accessed Jun. 15, 2021).

[29] J. Hagemann Snabe, “Siemens,” 2021. https://www.plm.automation.siemens.com/global/pt/products/manufacturing-planning/plant-simulation-throughput-optimization.html (accessed Mar. 03, 2021).

[30] M. Kliment, P. Trebuňa, and M. Straka, “Tecnomatix Plant Simulation, Its Features and Its Integration into Business Processes in Logistics Systems,” *Am. J. Mech. Eng.*, vol. 2, no. 7, pp. 286–289, 2014, doi: 10.12691/ajme-2-7-24.

[31] “Siemens Tecnomatix Plant Simulation.” https://www.engusa.com/pt_br/product/siemens-tecnomatix-plant-simulation (accessed Mar. 03, 2021).

[32] H. Arnarson, “Visual Components,” The arthic University of Norway, 2019.

[33] C. Zhuang, T. Miao, J. Liu, and H. Xiong, “The connotation of digital twin, and the construction and application method of shop-floor digital twin,” *Robot. Comput. Integr. Manuf.*, vol. 68, no. November 2020, p. 102075, 2021, doi: 10.1016/j.rcim.2020.102075.

[34] R. Rosen, G. Von Wichert, G. Lo, and K. D. Bettenhausen, “About the importance of

autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 28, no. 3, pp. 567–572, 2015, doi: 10.1016/j.ifacol.2015.06.141.

[35] "GE Digital." https://www.ge.com/digital/innovation/discover-ge-digital (accessed May 12, 2021).

[36] "Siemens." https://www.plm.automation.siemens.com/global/en/our-story/glossary/digital-twin/24465 (accessed May 12, 2021).

[37] "ABB." https://new.abb.com/control-systems/features/digital-twin-applications (accessed May 12, 2021).

[38] "IBM." https://www.ibm.com/topics/what-is-a-digital-twin (accessed May 12, 2021).

[39] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018, doi: 10.1016/j.ifacol.2018.08.474.

[40] "Getting Started — MTConnect." https://www.mtconnect.org/getting-started (accessed Mar. 10, 2021).

[41] "BOSCH DOCUMENTATION." https://developer.bosch-iot-suite.com/documentation/ (accessed Apr. 20, 2021).

[42] I. Communication, "Manual | EN," 2021.

[43] K. Matthias and S. P. Kane, *No Title*. Sebastopol: O'Reilly Media, 2015.

[44] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," *ISPASS 2015 - IEEE Int. Symp. Perform. Anal. Syst. Softw.*, pp. 171–172, 2015, doi: 10.1109/ISPASS.2015.7095802.

[45] N. Zhao *et al.*, "Large-Scale Analysis of Docker Images and Performance Implications for Container Storage Systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 4, pp. 918–930, 2021, doi: 10.1109/TPDS.2020.3034517.

[46] "No Title." https://www.docker.com/company (accessed Apr. 28, 2021).

[47] J. Bhimani *et al.*, "Understanding performance of I/O intensive containerized applications for NVMe SSDs," *2016 IEEE 35th Int. Perform. Comput. Commun. Conf. IPCCC 2016*, 2017, doi: 10.1109/PCCC.2016.7820650.

[48] "No Title." https://hub.docker.com/ (accessed Apr. 28, 2021).

[49] J. Simonsson, L. Zhang, B. Morin, B. Baudry, and M. Monperrus, "Observability and chaos engineering on system calls for containerized applications in docker," *arXiv*, vol. 122, pp. 117–129, 2019, doi: 10.1016/j.future.2021.04.001.

[50] M. Sollfrank, F. Loch, S. Denteneer, and B. Vogel-Heuser, "Evaluating Docker for Lightweight Virtualization of Distributed and Time-Sensitive Applications in Industrial Automation," *IEEE Trans. Ind. Informatics*, vol. 17, no. 5, pp. 3566–3576, 2021, doi: 10.1109/TII.2020.3022843.

[51] B. Piedade, J. P. Dias, and F. F. Correia, "An empirical study on visual programming docker compose configurations," *Proc. - 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Model. 2020 - Companion Proc.*, pp. 403–412, 2020, doi: 10.1145/3417990.3420194.

[52] L. Magnoni, "Modern messaging for distributed sytems," *J. Phys. Conf. Ser.*, vol. 608, no. 1, 2015, doi: 10.1088/1742-6596/608/1/012038.

[53] IBM, "Message Brokers," *23 January*, 2020. https://www.ibm.com/cloud/learn/message-brokers (accessed May 03, 2021).

[54] "MQTT." https://mqtt.org/ (accessed May 04, 2021).

[55] Beckhoff, "TC3 IoT Communication (MQTT)," pp. 1–128, 2018.

[56] D. Soni and A. Makwana, "A survey on mqtt: a protocol of internet of things(IoT)," *Int. Conf. Telecommun. Power Anal. Comput. Tech. (Ictpact - 2017)*, no. April, pp. 0–5, 2017, [Online]. Available: https://www.researchgate.net/publication/316018571_A_SURVEY_ON_MQTT_A_PROTOCOL_OF_INTERNET_OF_THINGSIOT.

[57] "AMQP." https://www.amqp.org/ (accessed May 04, 2021).

[58] "Apache Kafka." https://kafka.apache.org/.

[59] G. Hesse, C. Matthies, and M. Uflacker, "How fast can we insert? an empirical performance evaluation of apache kafka," *Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS*, vol. 2020-Decem, pp. 641–648, 2020, doi: 10.1109/ICPADS51040.2020.00089.

[60] G. M. D'Silva, A. Khan, Gaurav, and S. Bari, "Real-time processing of IoT events with historic data using Apache Kafka and Apache Spark with dashing framework," *RTEICT 2017 - 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc.*, vol. 2018-Janua, pp. 1804–1809, 2017, doi: 10.1109/RTEICT.2017.8256910.

[61] "Oracle." https://www.oracle.com/database/what-is-database/ (accessed May 12, 2021).

[62] M. Dave, "International Journal of Advanced Research in SQL and NoSQL Databases," *Int. J. Adv. Res. Comput. Sci. Softw. Eng. Res.*, vol. II, no. 8, 2016.

[63] Á. Vathy-Fogarassy and T. Hugyák, "Uniform data access platform for SQL and NoSQL database systems," *Inf. Syst.*, vol. 69, pp. 93–105, 2017, doi: 10.1016/j.is.2017.04.002.

[64] Y. Li and S. Manoharan, "A performance comparison of SQL and NoSQL databases," *IEEE Pacific RIM Conf. Commun. Comput. Signal Process. - Proc.*, pp. 15–19, 2013, doi: 10.1109/PACRIM.2013.6625441.

[65] M. Stonebraker, "SQL databases v. NoSQL databases," *Commun. ACM*, vol. 53, no. 4, pp. 10–11, 2010, doi: 10.1145/1721654.1721659.

[66] "Robert Bosch GmbH." https://en.wikipedia.org/wiki/Robert_Bosch_GmbH (accessed Apr. 15, 2021).

[67] S. F. Alves and G. B. Termotecnologia, "Sara Filipa Alves Pina Da Silva Integração E Adaptação Do Sap E Indústria 4.0 Na Área De Manutenção," 2017.

[68] A. C. B. Fernandes, "Gestão do desempenho dos colaboradores : O caso da Bosch," pp. 1–131, 2013.