TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Diplomarbeit

Development of a Personalized Real-Time KPI Dashboard Based on a Digital Twin in the TU Wien Industry 4.0 Pilot Factory

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

unter der Leitung von

Univ.-Prof. Dr.-Ing. Dipl. Wirt.-Ing. Prof. eh. Dr. h.c. Wilfried Sihn
(E330 Institut für Managementwissenschaften, Bereich: Betriebstechnik und Systemplanung)

Proj.-Ass. Dipl.-Ing. Werner Kritzinger, BSc
(E330 Institut für Managementwissenschaften, Bereich: Betriebstechnik und Systemplanung, Fraunhofer Austria Research GmbH)

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaften

von

Martin Fekár, BSc

1127099 (066 482)

Ambrova 20

831 01 Bratislava, Slowakei

Bratislava, im März 2019

Martin Fekár

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiters Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Bratislava, im März 2019

Martin Fekár

Danksagung

An dieser Stelle möchte ich mich bei all denen bedanken, die mich beim Verfassen dieser Arbeit, als auch während des früheren Studiums unterstützt haben.

In erster Reihe gilt mein Dank Herrn Prof. Wilfried Sihn, Dipl.-Ing. Werner Kritzinger und Stefan Fruhstuck von Fraunhofer Austria, für die Möglichkeit dieses interessante Thema zu bearbeiten, für ihre Ratschläge und gute Zusammenarbeit, die einen reibungslosen Ablauf meiner Diplomarbeit ermöglicht haben.

Weiters möchte ich mich bei Dr. Leopold Ruppert (ex. IET TU Wien) für die Gelegenheit bedanken, einen ersten Einblick in die akademische Welt gewinnen zu können. Er hat mein Interesse an Python geweckt, das mir bei (nicht nur) dieser Arbeit vieles erleichtert hat.

Für die schöneren Teile des Studiums, Motivation in schwierigen Zeiten und ständige Inspiration danke ich meiner alten Mathe-Lerngruppe: Wei, Christoph und Alessandro, als auch den Little Ducks von Tampere.

Nicht zuletzt gilt mein herzlichster Dank meiner Familie für ihre Unterstützung und Geduld, ohne die es überhaupt nicht möglich gewesen wäre mein Studium erfolgreich zu Ende zu bringen.

Kurzfassung

Im Umfeld aktueller Herausforderungen in der Industrie gewinnen operative Daten als Ressource an Bedeutung. Informationen, die aktuelle Zustände von Assets in Produktion und Logistik beschreiben, können nun dank Entwicklungen in Digitalisierung, Kommunikation und Datenverarbeitung zur Beurteilung deren Leistungsfähigkeit verwendet werden, den Entscheidungsprozess beeinflussen, und als Basis für weitere Optimierung von Produktionssystemen dienen.

Die Ressourcen der Montagelinie in der Pilotfabrik Industrie 4.0 der TU Wien sind mit Sensoren ausgestattet, die operative Daten mit großem Potenzial für ein besseres Verständnis und die Weiterentwicklung der Wertschöpfungsprozesse liefern können. Informationen zusammengefasst zu Kennzahlen (Key Performance Indicators, KPI) sind in diesem Zusammenhang besonders interessant, und sollen für operative Mitarbeiter der Pilotfabrik in Form einer Dashboard-Visualisierung verfügbar gemacht werden.

Als zentrale Informationsquelle soll dabei ein in einem Materialflusssimulationstool aufgebautes Digital Twin-Modell eingesetzt werden, um Daten zu konsolidieren und einen durchgängigen Datenfluss zwischen Assets der Montagelinie und der Dashboard-Anwendung, als auch zurück zum Modell zu realisieren. Dadurch können Vergangenheitswerte zur Vorhersage und Optimierung von zukünftigen Zuständen eingesetzt werden.

In einem ersten Schritt wurde eine Literaturrecherche mit Fokus auf Technologien und Kennzahlen zur Leistungsbeurteilung der Montageprozesse durchgeführt, gefolgt von einer Analyse bestehender Informationsflüsse. Im weiteren Verlauf hat ein Auswahlprozess unter Beachtung von Kennzahlenrelevanz und Verfügbarkeit der Rohdaten zu einem Satz von KPI geführt, der angepasst an einzelne Mitarbeiterrollen visualisiert werden sollte.

Diese Kennzahlen wurden mit der Entwicklung einer personalisierten, echtzeitfähigen Dashboard-Anwendung für Werker und operative Leiter der Pilotfabrik verfügbar gemacht, und somit ihre Fähigkeit Entscheidungen zu treffen verbessert. Des Weiteren ermöglicht die entwickelte Anwendung eine Parametrisierung des Digital Twin mit den gesammelten Vergangenheitsdaten. Dadurch werden realitätsnähere Simulationsergebnisse und schlussendlich optimalere Montageabläufe erwartet.

Abstract

In the light of recent challenges in the manufacturing industry, operational data is becoming an increasingly valuable resource. Enabled by developments in digitalization, communication and processing, information describing the current state of assets in production and logistics can be used to assess performance, aid in the decision-making process and serve as a basis for further optimization of production systems.

Assembly line resources in the TU Wien Industry 4.0 Pilot Factory are equipped with sensors delivering operational data, which holds a great potential for a better understanding and further development of value creating processes in the facility. Information condensed into Key Performance Indicators (KPI) is of special interest in this aspect, and has to be made available for operational employees of the Pilot Factory in the form of a dashboard visualization.

A Digital Twin model based upon a material flow simulation tool has to be used as the central source of information, to consolidate data and to create a continuous information flow from shop floor assets towards the dashboard application, as well as back to the model, where past data can be used for the prediction and optimization of future states.

Following a literature research focusing on enabling technologies and metrics relevant for a performance assessment of the assembly process, an analysis of existing information flows was executed. Taking into account the relevance of performance indicators and the availability of raw data, a selection process led to a set of KPIs to be visualized for individual stakeholder roles.

These metrics were made available to shop floor workers and operational managers of the Pilot Factory with the development of a personalized, real-time dashboard application, improving their decision-making ability. Additionally, the developed application enables a parametrization of the Digital Twin with collected past data, expecting more realistic simulation results, and finally optimized assembly processes.

Table of Contents

1 Introduction

Companies across the manufacturing industry have recently been facing an increasing amount of challenges, as the global markets for industrially produced goods have been transforming in reaction to global megatrends.[1]

Alongside the economic (and geopolitical) situation, demographic changes and urbanization are also quickly becoming important variables influencing the demand for goods and the availability of workforce. The role of environmental issues, climate change and resource availability are likewise gaining significance, creating the need for more efficient and environmentally friendly products. In the meantime, new and innovative technologies are emerging at unprecedented speeds, with the potential to disrupt currently existing business models and change the society as we know it. Inventions such as artificial intelligence, additive manufacturing, virtual reality and ubiquitous computing are reaching commercial availability, while customer expectations are rising in the search for advanced products able to fulfil their individual needs. Moreover, global competition is establishing itself on traditional markets, prompting the currently leading suppliers to strive for more flexibility and efficiency in production, shorter development periods, and more cooperation between supply chain partners.[2,3,4]

Newly developed products can be often characterized by rising complexity and increased use of digital components, extending their functionality and providing added value in comparison to previous generations.[5] Using the automobile as a prime example, the already substantial share of electronic and software components on its overall value is predicted to rise rapidly and reach 50% by 2030, with players from the digital industry finding their ways to participate in this transformation alongside (or in place of) traditional manufacturers.[6] In the near future, the availability of sensor technologies, computing power and networking capabilities will open the way for autonomous, driverless vehicles, which in turn will have major implications for the economy as a whole, changing the concept of mobility and many accompanied services and aspects of everyday life.[7] In a similar way, other revolutionary technologies are poised to become integral parts of various products, opening new opportunities in terms of functionality and quality at the cost of increased complexity and uncertainty for current producers. This is equally true for manufacturing systems,

[1] see Kautzsch et al., 2017, p. 8
[2] see EEA, 2015, p. 4
[3] see Kautzsch et al., 2017, p. 9
[4] see Lasi et al., 2014, p. 239
[5] see Mueller et al., 2018, p. 4
[6] Chitkara et al., 2013, p. 12
[7] see Kautzsch et al., 2017, p. 10

which will inevitably have to adapt to the changing needs and reach new levels of efficiency in order to facilitate the creation of future products.[8]

The primary direction of production transformation is seen in the development of smart, digital factories equipped with extensive IT systems, sensor technologies, data analytics and realistic simulations. In this context, the manufacturing sector is experiencing an application pull prompting for industry-wide transformation, but simultaneously also a technology push responsible for speeding up the implementation of digital solutions.[9]

While computer-based systems employed in various roles (product development, production planning, logistics, etc.) already present core elements of today's industrial world, a digital factory is supposed to achieve more than that – create additional value by data collection, transmission, storage, processing and use. Characterized by a comprehensive network of interconnected digital models, methods and tools, its ultimate goal is the holistic planning, evaluation and improvement of all relevant structures, processes and resources inside a production facility.[10, 11, 12]

1.1 Problem Definition

As the requirements for efficiency and competitiveness of manufacturing enterprises are rising due to current megatrends, real-time operational data is becoming an important resource supporting the management and optimization of production and logistic processes. Information derived from past and current performance values of a system can be employed to better understand its way of operation, to support an information-based decision making process and to discover and realize improvement potentials, creating additional value for equipment operators and their customers. However, this raw data has to be collected, stored and processed before it can be made available to the right stakeholders in the right amount and form.[13]

Through the introduction of technologies and concepts in the scope of Industry 4.0, new possibilities for the processing and use of operational data are becoming feasible. Realistic Digital Twin simulation models of physical objects and systems can be employed to continuously observe the performance of connected assets and provide real-time operational data, as well as to predict future states of physical systems and solve optimization problems, validated with information extracted from said data.[14]

[8] see Kagermann et al., 2013, p. 15
[9] see Lasi et al., 2014, p. 239
[10] see Bracht et al., 2018, p. 1
[11] VDI 4499-1, 2008, p. 3
[12] see Sommarberg, 2016, p. 69
[13] see Tokola et al., 2016, p. 620
[14] see Grieves and Vickers, 2017, p. 3

Assets of the assembly line in the Industry 4.0 Pilot Factory of the TU Wien are able to generate and communicate operational data in various shapes and forms, which holds a great potential to provide insights, support decisions and help optimize the assembly process. In order to make this information available to stakeholders and software tools, a real-time, personalized computation, decision support and visualization tool has to be integrated with the existing Digital Twin model of the production system, creating a continuous data connection between the shop floor assets, the simulation model as a centralized data source, the stakeholders and the simulation model again, now in the role of a stakeholder-like recipient of information.

1.2 Research Questions and Thesis Goals

The problem described in the previous section can now be transformed into the central research question of this thesis, defining the objectives and specifying expected outcomes:

- How can performance indicators based on operational data from production and logistic systems of the Pilot Factory be displayed clearly, taking into account the needs and limitations of various stakeholders?

In order to provide an answer to this question, and to reach the goal of data provisioning for production optimization, three more questions have to be considered as well:

- Which of the operational data and performance indicators gained from the observed assets are relevant for a near-real-time evaluation and optimization of manufacturing processes within the Pilot Factory?
- What is the shape of information flows within the Pilot Factory, and what possibilities are open for the inclusion of additional data sources?
- How can the collected operational data be efficiently and clearly managed, prepared for use within a simulation model, and how can the outputs from this model be stored in a database?

Based on the research questions, a solution-neutral formulation of thesis goals can be constructed, with the purpose of supporting the search for a solution to the presented problem, to describe the expected outcomes and later serve in the evaluation of achieved results. The following work packages have to be completed in order to provide answers to the research questions:

- Investigation of factors relevant for performance assessment of production and assembly processes within the Pilot Factory
- Creation of a data-based decision support tool for operative employees
- Provisioning of fundamental data for production optimization

1.3 Solution Proposal and Thesis Structure

The goals of this thesis can be fulfilled by the creation of a set of software tools, designed to store and process Pilot Factory operational data, carry out numerical operations and communicate their outcomes in real time, while respecting the needs of human and software recipients. Before the actual development can commence, an analysis of information inputs and expected outputs has to be carried out, following an overview of relevant theoretical topics.

In the **Introduction** the motivation, problem definition and goals for this thesis are explained, and a structure of work packages is proposed.

Theoretical Foundations describe and define concepts, terms and technologies important for the practical implementation of solutions to the thesis goals.

KPI Research concerns the search for performance metrics and their input factors relevant for the Pilot Factory use case.

In the practical part, the actual process leading to the creation of a decision support tool and its modules is documented:

Pilot Factory Information Flows serve as a link between the shop floor assets of the Pilot Factory and the researched set of key performance indicators.

The chapter titled **KPI Selection** explains the final choice of performance indicators to be visualized by the decision support tool.

Dashboard Development deals with the task of sourcing, processing and finally displaying data in a clear and efficient manner, taking into account the needs of various stakeholders.

Provisioning of Simulation Parameters documents the transformation of operational data into control variables to be used for production optimization within the Digital Twin simulation model.

Once the practical part of this thesis has been finalized with the development of required software modules, the results of the process are briefly discussed:

Conclusion and **Outlook** summarize the findings and propose directions for future use and development of the created set of software tools.

Finally, the **Appendix** contains additional information and source code complementing and illustrating various parts of this thesis.

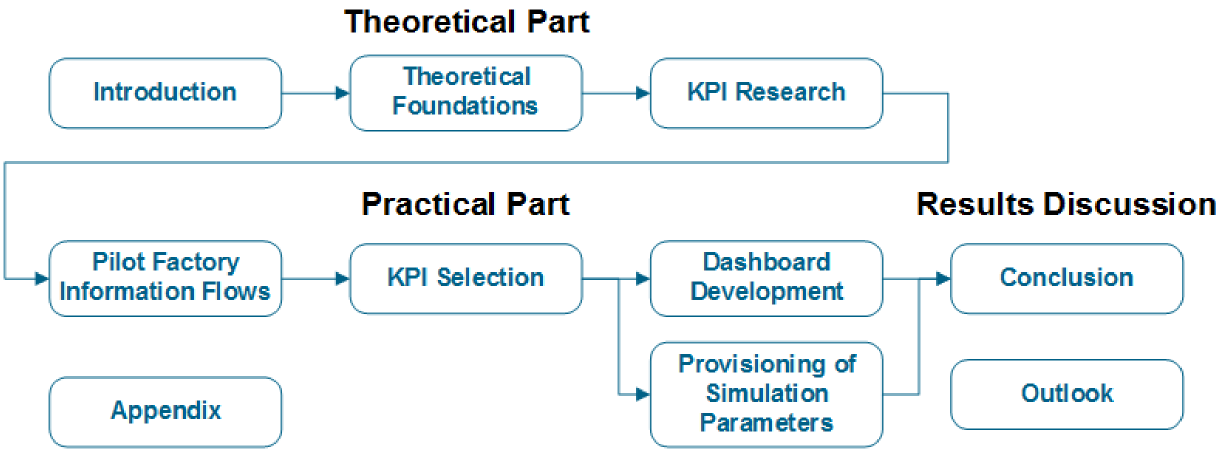


Figure 1: Thesis Structure Visualized

2 Theoretical Foundations

In this chapter, a brief introduction into the topics, concepts and technologies essential for the fulfilment of the thesis goals is provided. Beginning with an overview of Industry 4.0 as the phenomenon reshaping manufacturing, the chapter continues with definitions of key information technologies enabling this digital transformation and the Digital Twin as an innovative concept of information-based value creation. In the central part, the more specific topic of industrial information systems is mentioned, followed by a presentation of the showground for digitalization efforts – the TU Wien Industry 4.0 Pilot Factory. The theoretical part is rounded off with an introduction into Key Performance Indicators and data visualization as the subjects essential for the practical part of this thesis.

2.1 Industry 4.0

In the history of manufacturing, three major milestones (paradigm shifts) occurred over the course of past centuries. Each of them was marked by the introduction of an innovative technology-driven concept, making the way for superior performance and quality of production systems. Because of their significance for humankind, these events are often titled as industrial revolutions, inspired by the widely known (first) Industrial Revolution – the mechanization of manufacturing that begun in Great Britain during the late 18th century, enabled by the introduction of water- and steam powered machinery.

The remaining two equally important milestones were the creation of continuous production lines (early 1900s, enabled by electrification and division of labour), and the programmable automation of manufacturing (thanks to the advancements in electronics and IT in the last third of the 20th century).[15,16]

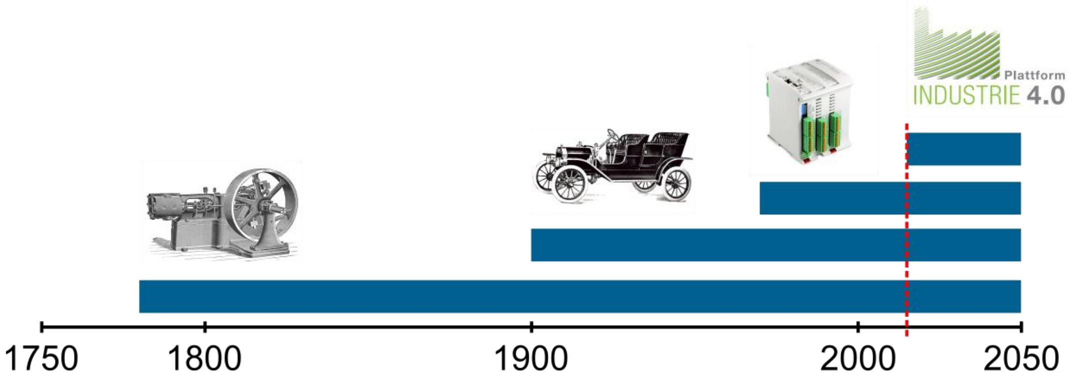


Figure 2: Timeline of Industrial Revolutions[17]

[15] see Drath and Horch, 2014, p. 56
[16] see Kagermann et al., 2013, p. 13
[17] see Kagermann et al., 2013, p. 13

Although the true importance and the revolutionary significance of these events were identified only in retrospect, the signs of an upcoming fourth industrial revolution can already be observed today. In various settings of everyday life, the emergence of internet-based technologies has transformed services, markets and even lifestyles for major parts of the global population, by speeding up communication and simplifying access to information. Such developments are now finding their way into the manufacturing industry, with the potential to further increase its performance and efficiency (saving costs), as well as to disrupt the status quo by introducing new, innovative products and business models (creating value).

Significant cost reductions are achievable through the digitalization of the production process, ranging from warehouse logistics, through the production line to quality control and maintenance (Table 1). In terms of value creation effects of Industry 4.0, the potential in selected branches of the German industry is expected at +23% of gross value added between 2013 and 2025, amounting to €78.77 billion (Table 2).[18]

Area	Cost Saving Potential
Inventory Costs	30% to 40%
Manufacturing Costs	10% to 20%
Logistics Costs	10% to 20%
Complexity Costs	60% to 70%
Quality Costs	10% to 20%
Maintenance Costs	20% to 30%

Table 1: Cost Saving Potentials of Industry 4.0[19]

Industry	Value Creation Potential
Chemical	+30%
Automotive	+20%
Mechanical	+30%
Electrical	+30%
Agriculture	+15%
ICT	+15%

Table 2: Value Creation Potentials of Industry 4.0[20]

[18] see Bauer et al., 2014, p. 36
[19] Bauernhansl et al., 2014, p. 31
[20] Bauer et al., 2014, p. 36

Closely following the appearance of highly innovative concepts in the past, many other serious changes were reshaping the industrial landscape, as well as the society as a whole. Shortly after the mechanization of industry, the newly erected factories in England attracted workforce from rural regions, resulting in a dramatic growth of cities. After the creation of continuous production lines, industrially manufactured goods became widely available for the general population due to sinking prices, as experienced with the first mass produced automobile – the Ford Model T.

Similar wide-reaching effects can be expected following the spread of network connectivity in industrial assets. In the early days of the World Wide Web, hardly anyone would have been able to predict its influence on advertisement, product sales or entertainment. It is now essential to predict the possible outcomes of industry digitalization and prepare accordingly, explaining the early declaration of the Fourth Industrial Revolution. Manufacturers need to analyse their position and define a strategy to unlock its full potential. The important questions to ask are: How is data creating new value? How fast will new business models emerge? How will industry structures change? How do we play in this space? What capabilities do we require to win? And last but not least: How do we get started?[21]

The implications of a potentially revolutionary impact of digitalization on the industrial landscape of Germany have been studied extensively in a cooperation between institutions of the federal government, academic and industrial partners. Their initial findings were published in a report, defining the term Industry 4.0 as the name of a strategic initiative, supposed to secure the world-leading position for Germany's manufacturing industry in the face of upcoming challenges.

Many of the underlying technologies needed for this transformation are already available in other industries, for example in consumer electronics and digital entertainment. The challenge of implementing them in manufacturing lies more in the combination and connection of already existing solutions, than in developing new technology.[22]

The vision of Industry 4.0 is focused on the integration of network-enabled embedded electronics into industrial processes in manufacturing and logistics, with the main benefits not necessarily lying in the technology, but instead in value creation and emergence of new business models, downstream services and work organization concepts. Large potentials of this approach are expected in the technical domain, as well as in social aspects of manufacturing: [23, 24]

---

21 see Russo et al., 2014
22 see Drath and Horch, 2014, p. 57
23 Drath and Horch, 2014, p. 58
24 Kagermann et al., 2013, p. 16

- Individual customer requirements can be observed in all product lifecycle phases, while maintaining profitability. Small batch sizes all the way down to one-off items can be produced with reasonable unit costs.
- The flexibility of production systems is increased, enabling quick reactions to breakdowns and resource shortages thanks to real-time data transparency, while fine-tuning their way of operation to achieve optimal results.
- Efficient use of resources enables maximal output from a given input amount, hand in hand with enhanced environmental friendliness and lower energy consumption.
- New services and value opportunities are created using large quantities of collected operational data. Business-to-business (B2B) offers such as fleet management and maintenance plans are developed by OEMs.
- Improved flexibility and interaction between humans and machines presents a response to demographic change, work-life balance and competition from low-income countries.

The proposed way to achieve similar effects is a long-term, gradual innovation process, where the value of existing manufacturing systems is preserved, leveraging existing technological, economic and human potential of Germany's industry. The implementation of Industry 4.0 is thus based on a dual strategy: the deployment of smart manufacturing technologies across the industry (leading market), and a simultaneous sale of relevant technologies in order to boost the domestic manufacturing equipment producers (leading supplier). By using a comprehensive know-how and technology transfer initiative (publishing pilot applications, best practices, and standardized solutions), small and medium-sized enterprises can also benefit from the potential of smart manufacturing, removing the size-related barriers.[25]

The implementation strategy of Industry 4.0 focuses on three key areas:[26]

- **Vertical integration** of manufacturing systems within businesses, creating uninterrupted digital connections, enabling seamless data flows from shop floor sensors all the way up to the central ERP solution and back
- **Digital end-to-end engineering** integrating real and digital worlds through all phases of the product life cycle.
- **Horizontal integration** creating digital-based cooperation opportunities between companies.

[25] see Kagermann et al., 2013, p. 29
[26] see Kagermann et al., 2013, p. 30

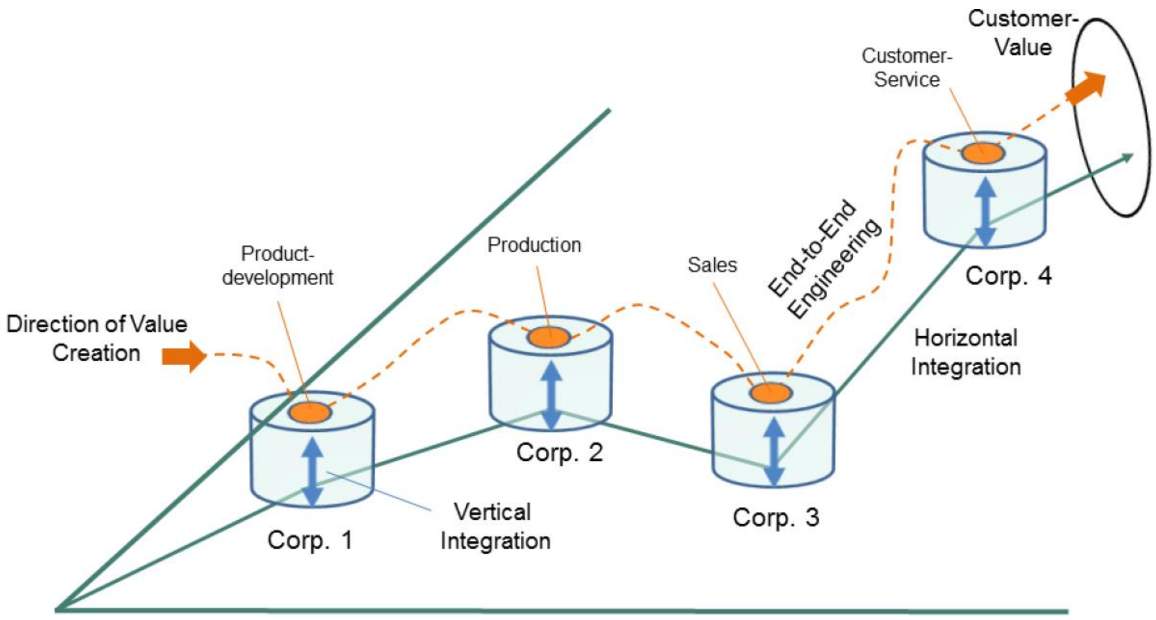


Figure 3: Key Areas of the Industry 4.0 Strategy[27]

Reference Architectural Model Industry 4.0 (RAMI4.0)[28, 29]

Within Industry 4.0, the network-enabled production assets made by various vendors and serving in different roles need to communicate with each other along the whole value chain. To enable a seamless connectivity, a uniform basis is needed, so that all elements are able to understand each other. The Reference Architectural Model Industry 4.0 provides a three-dimensional layer and life cycle structure for object classification and defines the nomenclature for the integration of IT and automation technologies (Figure 4).

The right horizontal axis with hierarchy levels represents functionalities within manufacturing systems. For this purpose, the classic automation pyramid (Figure 13) was expanded to include the work piece (Product), as well as the networking interface (Connected World).

The left horizontal axis symbolizes the life cycle phases of an object, including type development, instance production and usage phases.

Finally, on the vertical axis the functionality and properties of an asset are visualized, broken down into six layers for simplification – asset operation, connectivity, information exchange and its place within the business processes.

---

[27] Schumacher, 2015, p. 18
[28] see Schweichhart, 2016
[29] see Hankel, 2015

The Reference Architectural Model serves to provide a common understanding of requirements within Industry 4.0, aiding in the development of standards across industry sectors and nations.

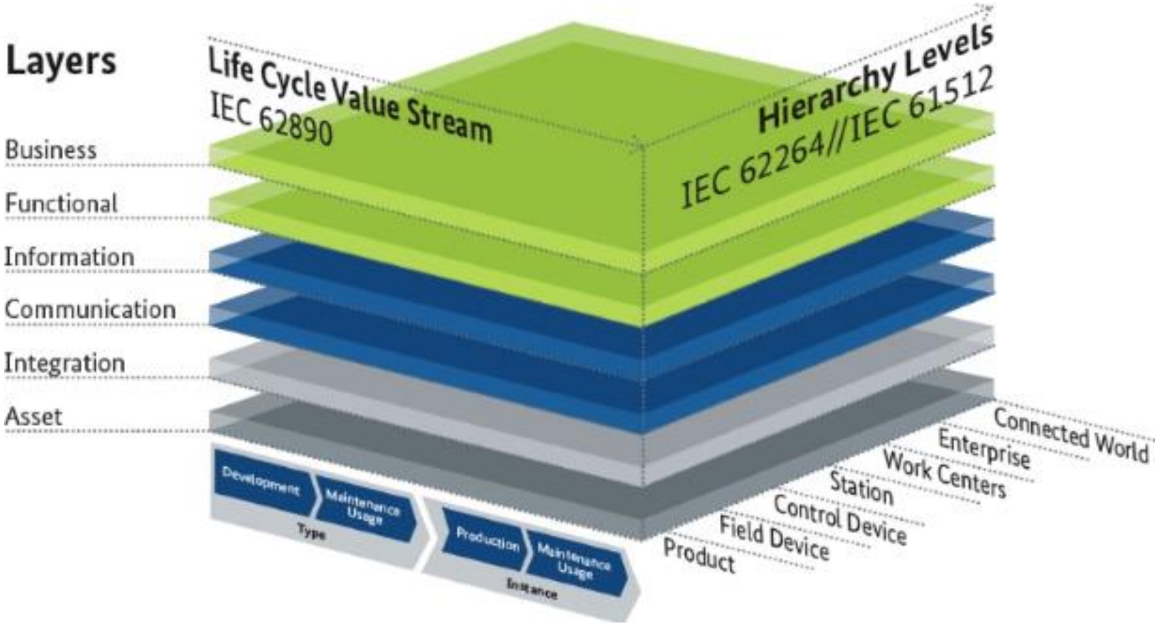


Figure 4: Reference Architectural Model Industry 4.0[30]

Industry 4.0 and the Industrial Internet

The innovation potential and incentives for a transformation of the manufacturing sector are present worldwide. An initiative similar to Industry 4.0 has been proposed by General Electric, now acting within the Industrial Internet Consortium (IIC) – a group of large multinational corporations including AT&T, Cisco, GE, IBM and Intel. Their understanding of interconnected devices is known as the Industrial Internet of Things, and encompasses the digital transformation of a broader spectrum of areas than just the manufacturing industry. In general, any sector of the global economy where an IIC stakeholder is active can be considered for the implementation of Industrial Internet solutions – Energy, Healthcare, Manufacturing, Public Domain or Transportation (Figure 5).

The Industrial Internet of Things might be described as a collection of opportunities and best practices for the integration of embedded electronics into various technological devices, promoted by their OEMs and focusing on machines, analytics as well as people. In short, Industry 4.0 concentrates on *making* things smartly, while the Industrial Internet focuses on making things *work* smartly.[31, 32, 33]

[30] Schweichhart, 2016, p. 4
[31] see Evans and Annunziata, 2012
[32] see Bledowski, 2015
[33] see Lin et al., 2017, p. 3

Despite the differences in approach and scope, a cooperation between the IIC and the Plattform Industrie 4.0 is taking place, with the goal of enabling interoperability among systems designed in accordance with the respective reference architectures. This can be achieved by mapping common functionalities, layers and communication standards. One of the incentives for such a collaboration is surely the membership of multiple German industrial concerns (Bosch, SAP) in both consortia.[34]

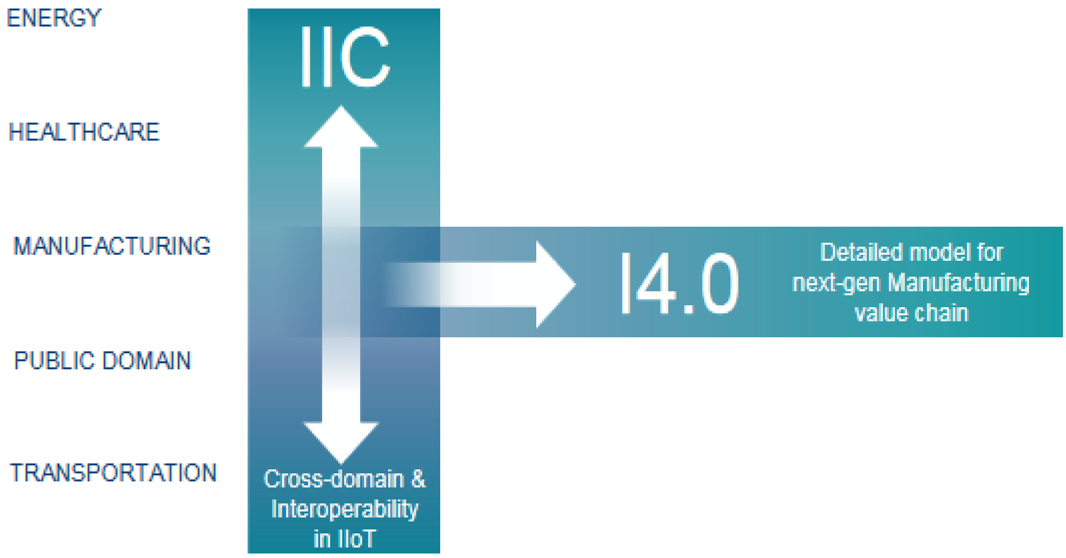


Figure 5: Focus Overlap of Industry 4.0 and the Industrial Internet of Things[35]

2.2 Information Technologies and the Internet of Things

On the technological side of Industry 4.0, network connectivity of physical objects is the central point of digitalization efforts. This widespread trend is currently visible in the consumer sphere, where the transformation of household electronics and appliances into “smart” devices is supposed to bring enhanced functionality, flexibility and comfort for their users – effects directly comparable with the goals of Industry 4.0. The emergence of digital manufacturing is thus going to be a part of a wider smart, networked infrastructure – the Internet of Things (IoT) – including not only consumer electronics, but also concepts such as smart grids, smart health and sustainable mobility transforming many everyday situations.

The core concept for the introduction of internet technologies into previously not connected items is the Cyber-Physical System (CPS) – an addition of digital aspects to a real-life object by means of embedded, network-connected IT solutions. In this

[34] see Lin et al., 2017
[35] Lin et al., 2017, p. 2

sense, a physical object (building, machine, vehicle) can be augmented with two supplementary layers (Figure 6):

- Its digital representation within the network infrastructure (cloud), enabling data exchange with other objects, users and services
- A service and application layer utilizing data gathered from the real object to create added value

With the help of these additional layers, an object is able not only to collect information using sensors, but also to store, analyse and leverage it in real time using (online) software tools, and finally to influence its physical surroundings by means of actors reacting to commands from local (edge) or remote (cloud) computers. Within this environment, a human operator is able to follow and influence the situation using human-machine interfaces such as displays, voice commands or even augmented reality tools.[36]

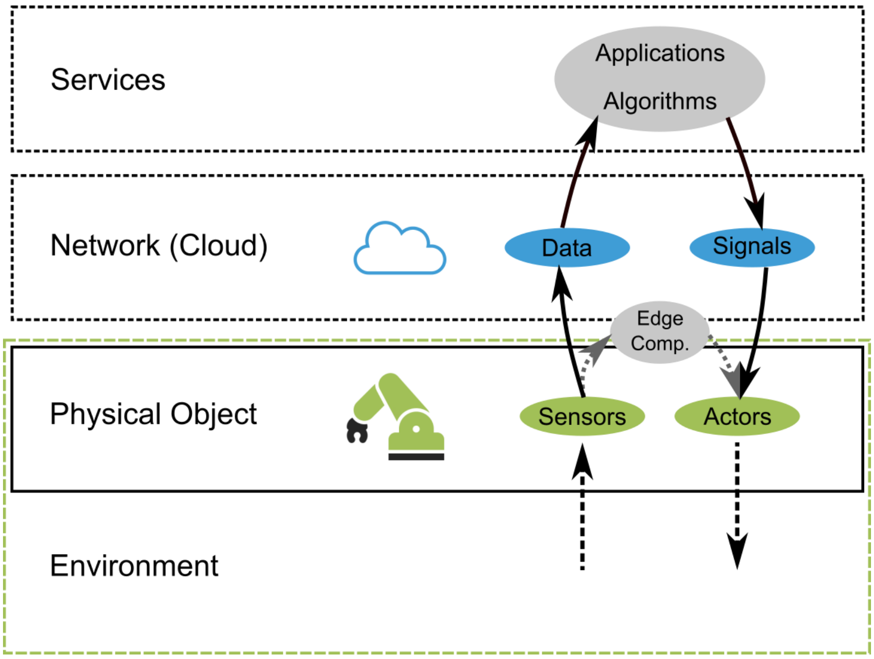


Figure 6: Layers of a Cyber-Physical System[37]

Until recently, the spread of network-enabled devices has been limited by various technological obstacles. For example, a sufficient amount of physical addresses within the Internet Protocol was only provided with the introduction of IPv6 in 2012, which can be seen as one of the decisive moments for the debut of the Internet of Things.[38] Further prerequisites for a widespread IoT implementation are a favourable price-performance ratio of computational power (steadily improving as predicted by Moore's

[36] see Bauernhansl et al., 2014, p. 16
[37] see Drath and Horch, 2014, p. 57
[38] see Kagermann et al., 2013, p. 13

law[39]), as well as the miniaturization and greater energy efficiency of electronics. By removing technological limits, the concept of ubiquitous computing is nearing reality. In this vision, computational power and the accompanying services will be available literally everywhere, enabling for data exchange between users, online services and objects, resulting in optimized operation and additional value creation.

The IoT is not based on a single revolutionary technology, but rather on a combination of developments and concepts in (wireless) communication, sensors and actors, computation, localization and user interfaces.[40] On the following pages, some of the technologies central for data management and information processing within the Internet of Things are introduced.

Sensors and Actors

The whole concept of IoT is based on the use of data to support and facilitate informed decision making in the short term, and to optimize the performance of a system in the longer run. To achieve these goals, data has to be sourced from individual assets in a suitable form, amount and quality. Depending on the individual use case, the variety of possibly monitored variables can be high.

A sensor in the IoT setting might be a traditional probe measuring some physical property during asset operation (Figure 7). Sensors counting event occurrences such as amounts of produced goods might be employed, as well as imaging sensors taking pictures in visible light, infrared or other spectrums. Geographical location, be it on a global scale or within the bounds of a building, can also be monitored using sensors and used within the IoT.

In a broader sense, activities of human actors can also be understood as sensor data. An employee equipped with an input device such as a tablet is able to create relevant information from observations of objects and processes, for example during a planned outage inspection, or by reporting faults.[41]

A network-enabled sensor has to possess some amount of computational power to capture inputs, convert them to a digital signal and forward them for processing using a specified protocol. On the same layer of a Cyber-Physical System as sensors, network-connected actors are able to turn commands received from devices positioned on higher layers into physical actions in the real world, and thus influence their environment.

[39] https://www.intel.com/content/www/us/en/history/museum-gordon-moore-law.html
[40] see Mattern and Floerkemeier, 2010, p. 245
[41] see General Electric Company, 2016, p. 19

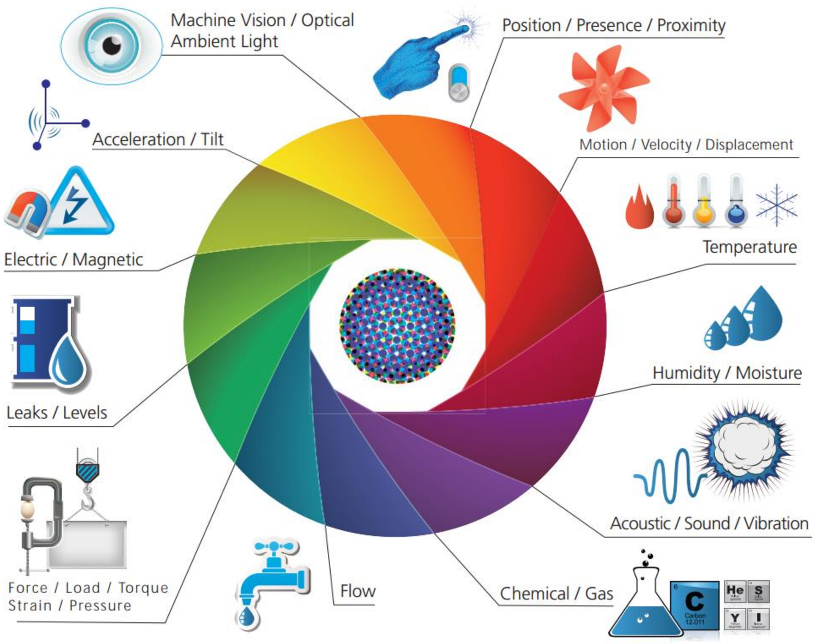


Figure 7: Common Sensor Classes[42]

Network Connectivity[43]

Once information has been collected by sensors, it needs to reach the elements of IoT responsible for processing and decision making. As information processing often takes place in remote devices connected to the network (cloud), the use of network communication technologies is inevitable. Depending on the situation, the aspects of connection speed, reliability and security come into consideration, while boundary conditions such as pre-existing infrastructure and asset mobility can play a role in choosing the right technology for industrial connectivity. Multiple wired and wireless communication standards are available on the market, with widely different capabilities and intended use cases:

- Wired Ethernet
- Passive and active RFID and NFC systems for very short-range identification and tracking of objects
- Short-range data connections, often with focus on low power consumption for battery powered mobile assets (Bluetooth, ZigBee)
- Wireless Local Area Network (WLAN)
- Mobile data connections (3G, 4G, …)

Network-connected elements of the IoT usually utilize the Internet Protocol (e.g. IPv6) as the basis for packet-based end-to-end data transmission across multiple

[42] https://www.postscapes.com/what-exactly-is-the-internet-of-things-infographic/
[43] Xu et al., 2014, p. 2237

networks.[44] On top of the transport layer, the OPC UA (OPC Unified Architecture) communication standard is widely employed, serving as the secure, open and reliable mechanism for transferring information between clients and servers – sensors, machines and enterprise systems.[45]

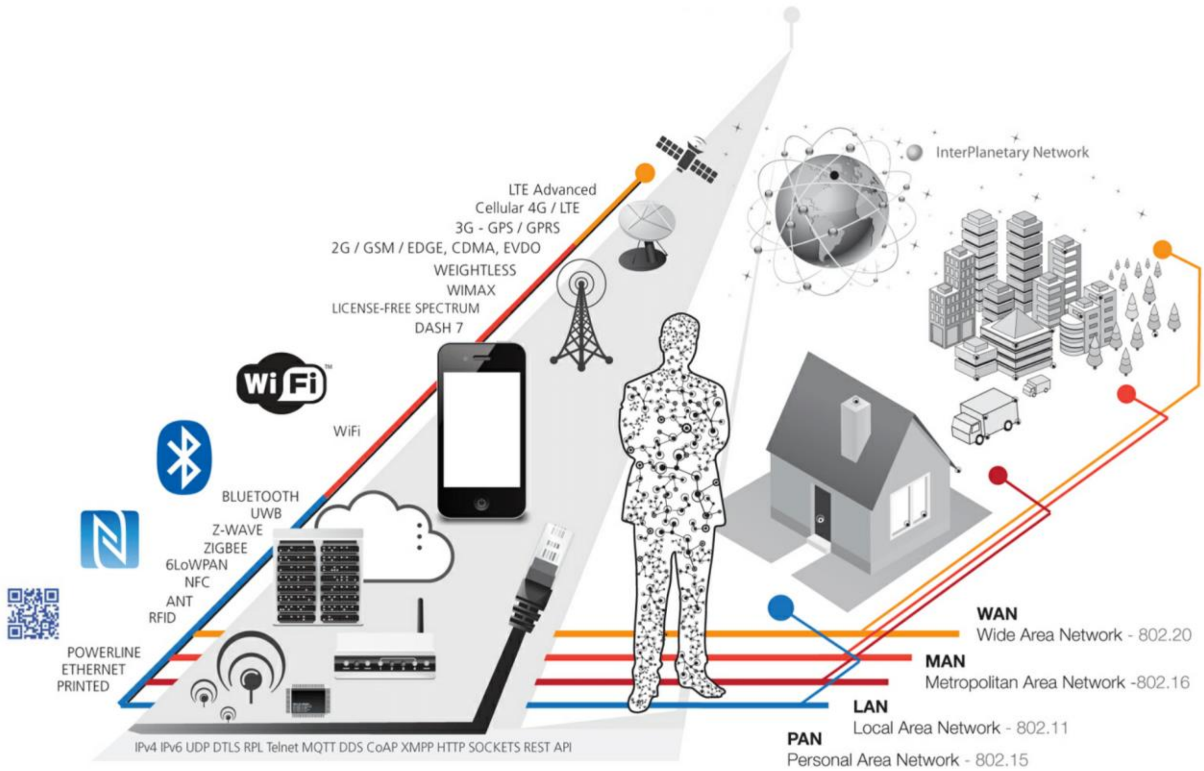


Figure 8: Overview of IoT Connectivity Technologies[46]

Relational Databases[47]

Information created by elements of the IoT has to be stored and made available for further use in analytics, services and applications. The conventional approach to data management relies on the use of relational database systems, which are able to store various information formats in a structured way, including the relationships between data, and enabling automated lookup and manipulation using queries. A database management system (DBMS) acts as an interpreter, providing the necessary capabilities:

- Database structure definition, including relationships between data
- Way to enter, modify, retrieve and delete data, usually using commands written in the Structured Query Language (SQL)
- Information security by user access management

[44] https://www.postscapes.com/internet-of-things-protocols/
[45] see https://www.rtaautomation.com/technologies/opcua/
[46] https://www.postscapes.com/what-exactly-is-the-internet-of-things-infographic/
[47] see Harrington, 2009

```sql
CREATE TABLE IF NOT EXISTS `DATA_PF` (`ID` INTEGER NOT NULL,`ID_Product` TEXT,`OrderNr` INTEGER,`TimeStamp_Start` TEXT,`TimeStamp_End`
 TEXT,`AGV_ID` INTEGER, `AGV_Status` INTEGER, `AGV_Load` TEXT, `StartPos` TEXT, `EndPos` TEXT,`Stoerung` INTEGER, PRIMARY KEY(`ID`));
INSERT INTO `DATA_PF` VALUES (1,1500,1015,'30.08.2018 08:00:09','30.08.2018 08:00:09',1,1,0,'Home','Home',0);
INSERT INTO `DATA_PF` VALUES (2,1500,1015,'30.08.2018 08:00:09','30.08.2018 08:00:31',1,2,0,'Home','Beladestation',0);
INSERT INTO `DATA_PF` VALUES (3,1500,1015,'30.08.2018 08:00:31','30.08.2018 08:04:29',1,1,1,'Beladestation','Beladestation',0);
INSERT INTO `DATA_PF` VALUES (4,1500,1015,'30.08.2018 08:04:29','30.08.2018 08:04:41',1,2,1,'Beladestation','M1',0);
INSERT INTO `DATA_PF` VALUES (5,1500,1015,'30.08.2018 08:04:41','30.08.2018 08:06:39',1,1,1,'M1','M1',0);
INSERT INTO `DATA_PF` VALUES (6,1500,1015,'30.08.2018 08:06:39','30.08.2018 08:06:53',1,2,1,'M1','M2',0);
INSERT INTO `DATA_PF` VALUES (7,1500,1015,'30.08.2018 08:06:53','30.08.2018 08:11:13',1,1,1,'M2','M2',0);
```

Figure 9: Sample SQL code

Multiple DBMS are available on the market, with differences in the offered functionality, intended use (lightweight to enterprise) and licence (open-source to proprietary). The choice of a management system is the first major step in database creation, and is often dictated by the circumstances (requested interoperability with applications and services, licensing, training) of the individual use case environment.

The standard hardware architecture for a relational database is based on the client-server concept. Thanks to the processing power of a client (PC), the server side only needs to process a request for data manipulation and return raw data, which is in turn utilized and formatted by the client. In contrast, SQLite as a lightweight DBMS does not use a server process, and client applications can directly access the database.[48]

The abstract model of a database can be constructed using entities and relationships between them:[49]

- **Entity**: object type described by stored data (e.g. customer, event…).
  - **Attributes**: data describing an entity (name, phone number, date…). Various data types are allowed within databases: strings, integers, floating point numbers, dates, Boolean values, etc.
  - **Instance**: one set of attributes belonging to one object of the entity type. A particular instance can be distinguished from others by a unique identifier (ID number, timestamp…).
- **Relationship**: describing an allowable connection between instances of entities.
  - **One-to-One**: an instance of entity A can be related to exactly zero or one instance of entity B and vice versa.
  - **One-to-Many**: an entity instance A is related to an arbitrary amount of instances B, but B is related to maximally one instance of entity A.
  - **Many-to-Many**: no limitations regarding the amount of related entity instances.

The entity-relationship model of a database directly translates into its structure in the form of tables and columns. One entity usually corresponds to one database table, containing a column for each attribute. Entity instances are represented by rows

[48] see https://www.sqlite.org/docs.html
[49] see Harrington, 2009, p. 51

(tuples) in the table. The most common way of visualizing entity-relationship models is the creation of class diagrams using the Unified Modeling Language (UML).[50]

Multiple challenges have to be dealt with when designing and building a database. Firstly, the entity-relationship model should be designed future-proof, as it can be problematic to subsequently update the database structure and all affected data entries. Such an update proves to be even more difficult when applications and services utilizing the data have to be modified, for example after introducing an additional attribute (column) to a table. As the database fills with data, its physical size on the disk as well as response times needed to process queries will inevitably rise. It is a task for the initial design to minimize the negative impacts of increasing data amounts. Applicable actions include the elimination of duplicate information, as well as an efficient set of primary keys (unique identifiers of a row within a table) and indices.

To guarantee the validity of database operations, a set of conditions summed up in the acronym ACID has to be fulfilled by each and every database transaction:[51]

- **Atomicity**: each database transaction either fully succeeds or fully fails, while the user must be aware of the result at all times.
- **Consistency**: a successful transaction can change the database contents in allowed ways only.
- **Isolation**: events within a transaction must be hidden from other transactions running concurrently.
- **Durability**: the results of a successfully completed transaction must survive any subsequent failures – the user must have a guarantee that the things the system says have happened have actually happened.

Security and resilience of the database against external and internal threats are especially important when dealing with confidential data, as well as when the execution of business tasks and processes depends on the access to stored information. Data has to be available at all times, and its integrity needs to be ensured. Threats compromising database security might range from hardware failures, power outages and faulty code all the way up to intentional abuse. Efficient safety measures rely on access restriction, user rights management, backup solutions and network protection using firewalls and encrypted connections.[52]

Big Data

Due to the amounts of raw data constantly created by elements of the IoT, a conventional relational database is often not a sufficient solution. Instead, a robust specialized IT infrastructure built in accordance with the principles of Big Data can be

[50] http://www.uml.org/
[51] Haerder and Reuter, 1983, p. 289
[52] see Harrington, 2009, p. 323

employed to successfully collect and store information in preparation for further use in analytics, services and applications.

Major attributes of Big Data can be summarized using the 3Vs: [53]

- **Variety**: structured and unstructured information in multiple formats – text, pictures, etc.
- **Volume**: raw amount of information
- **Velocity**: the need for real-time response to emerging situations in the form of analytical outputs and solutions

An emphasis on the performance and scalability of servers, storage and networking equipment is important in order to assure reliable operation, because data as a resource can be mission critical for operating the connected assets. Taking the heterogeneous nature of inputs into account, Big Data can also differ from structured information in conventional business applications.

One of the common software infrastructure solutions to facilitate Big Data management is Apache Hadoop. It is based on a distributed framework for storing, cataloguing, managing and querying of large amounts of potentially unstructured data across a network of servers. The network has the form of a horizontal structure, to enable easy communication scaling and increased resilience against hardware failures.[54]

Alternative mechanisms for data storage can be used in Big Data situations, namely non-relational (NoSQL) databases, replacing tables and their relations with other data structures (objects, documents). The main advantages are faster response times, easier scaling across multiple machines and enhanced availability at the cost of reduced (“eventual”) consistency.[55] NoSQL databases thus do not necessarily comply with the ACID rules described in the previous section, meaning that the user has no guarantee of receiving the most recent version of requested information. While this might be a serious issue for some applications (financial transactions, online ticket purchases), this approach offers significant advantages in performance for many databases.

Cloud Computing

The central prerequisite for the Internet of Things is an array of elements communicating with a central entity, where analytics, decision making, and data storage takes place. Cloud computing is a concept, where IT resources (processing power, data storage, applications, services and network infrastructure) are shared via the network on an on-demand basis. The essential point of a cloud solution is an

---

[53] Juniper Networks, 2012, p. 3
[54] Juniper Networks, 2012, p. 5
[55] see http://nosql-database.org/

automatic, flexible resource assignment based on momentary demand, enabled by pooling available software and hardware capacities that serve multiple services (and possibly clients) simultaneously. Client access is taking place over a network connection, which makes it possible to participate with a variety of client devices (cyber-physical systems, workstations, smartphones, etc.) based on heterogeneous platforms (operating systems).[56]

The cloud infrastructure can be owned and operated by the client company itself, a third-party provider or some combination of both, depending on the particular use case. Its physical location is generally irrelevant thanks to network access, but users might decide to operate from their own premises for security and performance reasons. A private cloud serves one customer only (again, independently of its actual location and ownership), while a community cloud is shared by multiple users, for example within a supply chain or when a vendor provides fleet management to their customers. A public cloud is offered by a provider for general use, and a hybrid one is a combination of the aforementioned options.

The division of responsibilities over hardware and software elements of a cloud may vary to a great extent, and can be characterized by the following models (Table 3): [57]

- **Software as a Service** (SaaS): The provider operates infrastructure including software, the customer utilizes ready-made applications with limited customization options only.
- **Platform as a Service** (PaaS): Infrastructure and platform are managed by the provider; custom applications with greater customization options are deployed by the customer.
- **Infrastructure as a Service** (IaaS): Fundamental computing resources are provider-operated; all software and possibly some hardware components are controlled by the customer.

	Infrastructure	**Platform**	**Applications**
Software as a Service	Provider	Provider	Provider
Platform as a Service	Provider	Provider	Customer
Infrastructure as a Service	Provider	Customer	Customer

Table 3: Overview of Cloud Computing Operation Models

[56] see Mell and Grance, 2011, p. 2
[57] Mell and Grance, 2011, p. 3

Edge Computing[58]

For computing infrastructure existing close to the data source and away from the centralized cloud the term Edge is used (Figure 6). The primary function of such a solution is to gather, pre-process and store data before sending it further on to cloud servers.

A recent trend is utilizing the increasing computational power of edge devices to carry out full-scale data analysis on the spot, reducing the need of communication with the remote cloud. Benefits of edge computing result from a faster response to data input and an elimination of the need for network communication. This way locations with poor connectivity, critical asset elements requiring instant action (such as closed-loop control systems) or security-sensitive assets can benefit from the IoT while respecting their specific constraints.

Because only cloud solutions can offer the massive computational power needed for large-scale data analysis, edge computing can be seen as an enhancement, not a replacement of the cloud concept.

2.3 Digital Twin

During NASA's Apollo program, at least two instances for every space vehicle were built. One supposed to fly into space, the remaining identical twin(s) used on the ground for testing and simulation purposes. Fifty years later, Airbus uses a similar approach to optimize and simulate aircraft systems, with one major difference – more and more parts of their physical twins are being replaced by virtual models – the Digital Twin, providing additional value across the whole life cycle of the designed product.[59]

In its original form, the Digital Twin is described as a digital informational construct of a physical system, created as an entity on its own and linked with the physical system in question. The digital representation should optimally include all information concerning the system (asset), that could be potentially obtained from its thorough inspection in the real world.[60]

This implies a dynamic representation, being constantly updated as the physical system changes during its lifetime. Owing to its origins as a conceptual model for PLM, the Digital Twin has been, ever since its inception, supposed to accompany the physical product starting from the very early design stages, through manufacturing and operational life all the way to the end of its service followed by decommissioning (cradle to grave).

---

[58] see https://www.ge.com/digital/blog/what-edge-computing
[59] see Rosen et al., 2015, p. 2
[60] see Grieves and Vickers, 2017, p. 1

Although the original concept precedes many of the key technologies enabling the creation of Digital Twins within the Internet of Things, it already contains all relevant aspects of the technology: the real space where the physical system exists is connected with the virtual space (digital model) by means of a bi-directional link. Data flow is modelled from the real world towards its digital representation, while information gained from the model is also fed back to the physical system. The set of virtual sub-spaces in the digital model represents the possibility of multiple model instances existing simultaneously. They can be created, used to carry out simulations, modified and destroyed as needed with no consequences to the asset, as opposed to testing physical prototypes (Figure 10).

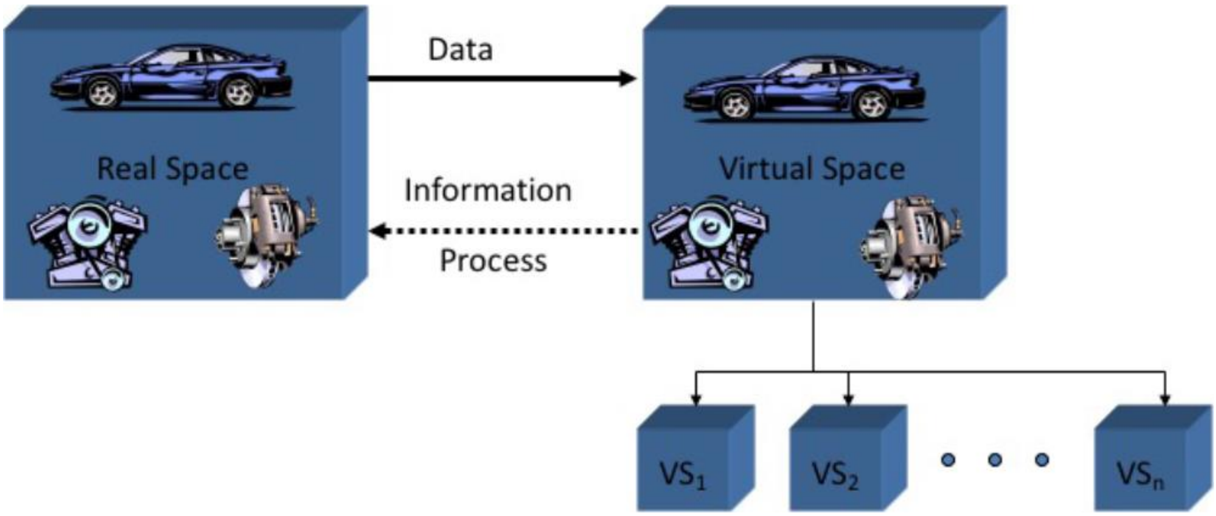


Figure 10: The Digital Twin as a PLM Concept[61]

In its original form, the Digital Twin describes the representation of a product, while the state of the art allows for (manufacturing, power generation etc.) processes to be a subject of virtual space reproduction ("twinning") as well, with the very same benefits as a result.

Industrial Applications

Due to the commonality of CAD and CAE systems, more or less every engineering project currently in development includes a set of geometrical and possibly also physics-based digital models describing the appearance and function of the designed system.[62] Usually these models are created as standalone, pragmatic pieces used for one particular task during the development phase and archived afterwards.

Recent developments in IT allow for a near-real-time transfer and processing of large data amounts, which can be gained from an array of intelligent sensors automatically

[61] Grieves and Vickers, 2017, p. 1
[62] see Grieves, 2014, p. 5

monitoring a product or process during its production as well as day-to-day operation phases. At the same time, increasing competition in many industrial sectors calls for innovative solutions enabling a more efficient operation of systems, reducing costs and increasing safety and quality. New operational strategies can also be created based on observations and predictions.

The Digital Twin approach combines these needs and opportunities to create a dynamic, holistic digital representation of a system all the way down to its component level, used throughout the whole lifecycle of the system from early development until retirement. A system in this sense might stand for either a physical product, or a process**,** in which case a component might symbolise a production machine or any atomic subdivision of the process (Figure 11).

A synchronized set of digital models creates a faithful virtual twin of an asset, which can then be "run" in a digital environment using collected real data to emulate the same operating conditions experienced by its physical counterpart. Simultaneously, model accuracy improves by learning from available data. Additional information and functionality can also be included in the virtual model, such as virtualised control systems, product documentation and metadata.[63]

The existence of an identical virtual copy of a system is a prerequisite for multiple use cases benefiting either its operator, manufacturer or both at the same time:

- Time to market can be reduced by virtual testing, effects of commonality and iterative design.[64]
- Product behaviour can be virtually simulated as a part of acceptance and compliance testing for legal reasons and for technical assurance. This way the system as a whole can be taken into account, not only its particular components.[65]
- Anomalies in system performance can be detected, making sure failures and their consequences are minimised.
- Predictions about future performance can be made, using data collected in the past, enabling predictive maintenance to save on maintenance costs and enhance safety at the same time.
- When twinning a manufacturing process, a virtual replica of a production line is created, providing insight into machinery, work in progress and environmental factors.
- What-if scenarios and their effects on any part of the system can be simulated, making it possible to develop new operation procedures with optimized performance and costs.

[63] see Smogeli, 2017, p. 2
[64] see Parrott and Warshaw, 2017, p. 2
[65] see Smogeli, 2017

- Knowledge from a fleet of identical or similar systems can be used for even more accurate predictions of future states to aid the operator, as well as to deliver data relevant for further development and improvement of products and processes to the manufacturer.
- Closed-loop control systems can be integrated with the Digital Twin, enabling autonomy.

Looking specifically at the Digital Twin of a manufacturing process, it consists of a model of the process itself (describing the way the product is to be manufactured), a model of the production facility (representing the needed assembly lines and machinery), as well as a model of the automation system supporting the process. A manufacturing Digital Twin offers an opportunity to simulate and optimize the production system, including its logistical aspects, and enables a detailed visualization of the manufacturing process from individual components up to the whole assembly.[66]

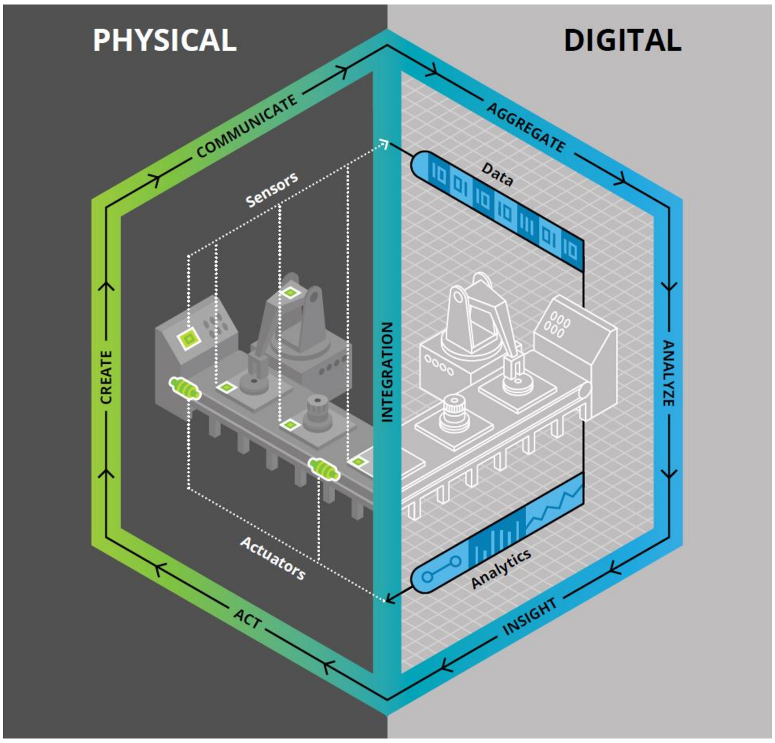


Figure 11: Digital Twin Model of a Manufacturing Process[67]

Figure 11 shows the principle of a manufacturing Digital Twin, integrating the physical and digital spaces. Starting on the left side, data is gathered by sensors and communicated towards the virtual model. There the aggregated data is analysed to gain insights which then influence the physical process by means of network-operated actuators. An identical flow diagram would be applicable to the Digital Twin of any product integrated with its control system, underlining the versatility of the concept.

66 see Siemens PLM Software, 2017, p. 23
67 Parrott and Warshaw, 2017, p. 5

Classification of Digital Twin Technologies[68]

The Digital Twin is a highly advanced and resource-intensive concept. While fully integrated products and systems will become more and more common in the near future, the current situation is characterized by the emergence of similar digitalization concepts with lower levels of aspect integration and automation. As these are simpler to create and maintain, first case studies are available in literature in addition to conceptual definitions. A classification of Digital Twin technologies based on the degree of data flow automation is presented in Figure 12.

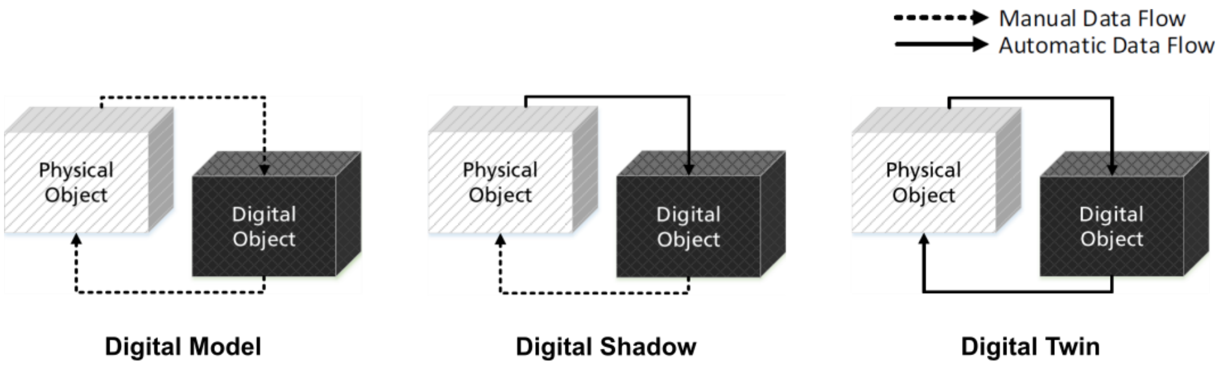


Figure 12: Overview of Digital Twin Technologies[69]

A Digital Model does not employ any form of automated data exchange, all changes have to be transferred manually, and thus no real-time connectivity between the real and virtual domains is present.

In the Digital Shadow, a one-way automated link is available, used to reflect states and properties of the physical object in its digital model.

Finally, the full-scale Digital Twin corresponding with the original concept definition provides a bi-directional real-time connection between the real and the virtual objects, enabling for information transfer towards the digital model, as well as facilitating direct control over the physical system using information provided by its Digital Twin.

2.4 Industrial Information Systems

Enterprise Resource Planning

The planning, management and control of resource use (materials, machines, employees) within an enterprise is a central part of its everyday operation, as the availability of resources, their optimal use in terms of costs and amounts, as well as overall operational flexibility are crucial success factors. [70]

[68] see Kritzinger et al., 2018
[69] Kritzinger et al., 2018, p. 1017
[70] see Osterhage, 2014, p. 5

ERP as a central software solution is tasked with supporting these processes on the business side of operations, but without a clear division from the technical processes in a manufacturing environment. This fact positions ERP on the top of the automation pyramid (Figure 13), where high-level, longer-term, company-wide planning and management tasks are executed. In addition to the software aspect, ERP can be understood as an organizational concept enabled by the centralized availability of relevant information.[71]

Depending on the individual use case and capabilities of the chosen software platform, an ERP system can consist of multiple modules, each of them dealing with a particular aspect or process within the enterprise:[72]

- Order Management, Sales and Distribution
- Procurement and Materials Management
- Production Planning
- Accounting and Controlling
- Quality Management
- Project Management
- Human Relations, etc.

Connections to other information systems are realized through interfaces, with the goal to concentrate as much business-related information as possible within the ERP software. Modern ERP solutions are integral parts of the digital company ecosystem, employing concepts such as Big Data and Cloud Computing (Section 2.2).

Manufacturing Execution Systems

In any modern factory, a fast and flexible handling of manufacturing processes and their optimization in real time is crucial to achieve an efficient and economical production. Standard ERP systems are not ideal for this role, because of their focus on longer-term planning horizons and consequently insufficient level of detail. On the other side, production assets themselves create too much excessively detailed data to be used effectively in short-term production planning.

Thus, in order to enable near-real-time planning of manufacturing processes, ensure process transparency, visualize information and material flows and support the continuous improvement process, specialized Manufacturing Execution Systems (MES) are employed. An MES is a modular software solution positioned on the manufacturing control level – filling the gap between shop floor operations and

[71] see Osterhage, 2014, p. 5
[72] see Osterhage, 2014, p. 26

enterprise-level planning. The main tasks of such a system are to monitor and organize orders, resources and material within the time horizon of up to a few production shifts:[73]

- Manage the resources (machines) based on current availability.
- Decide about the sequence of order processing.
- Assign orders to specific machines and personnel.
- Organize material staging, determine specific material use.
- Manage work-in-progress stocks.

Following the implementation of an MES solution, multiple benefits for its operator can be observed. The forward-planning nature ensures control over the associated production process, including the ability to take preventive steps in order to assure required performance. An MES as a software package is able to reduce employee workloads, especially in routine planning activities and to support the integrated production management through interfaces to other software, hardware and human elements within the manufacturing facility. This is also a prerequisite for the implementation of highly automated production systems.

The previously mentioned facts lead to a variety of positive effects, such as reduced queuing times, improved resource efficiency and higher flexibility, all of them finally resulting in a high level of certainty in the observed production system.[74]

According to VDI 5600, the modular nature of an MES is represented by several optionally available building blocks, dealing with various processes and functions within the manufacturing facility (Figure 13):[75]

- **Order Management** handles the production orders (activity triggers) handed over from ERP and other sources, including all the information required for their processing, which is in turn made available to other function blocks of the MES.
- **Detailed Scheduling and Process Control** assures the execution of queued orders in a real environment with limited resource availability and a possibility of process disruptions by assigning them to available assets.
- **Equipment Management** ensures the availability of production assets, dealing with the conflict between availability and dependability. Historical, current and future aspects of availability are taken into account.
- **Materials Management** deals with the timely and accurate supply of fresh materials, management of work-in-progress items, as well as waste disposal.
- **Human Resources Management** provides personnel for the production process, taking availability and qualification into account using all relevant employee data.

[73] see VDI 5600-1, 2016, p. 10
[74] see VDI 5600-1, 2016, p. 11
[75] see VDI 5600-1, 2016, p. 16

- **Data Acquisition** collects data from a production facility automatically and manually, carrying out pre-processing (input checks, compression…) and opening access to it for other MES modules.
- **Performance Analysis** realises control loops for a short-term correction of performance deviations, as well as long-term optimization processes. This module also deals with the calculation and visualization of performance indicators.
- **Quality Management** supports the achievement of product and process quality goals through quality planning, inspection and complaint management.
- **Information Management** coordinates the integration of other MES tasks and continuously distributes real-time information to production assets and systems, also providing tracking and documentation services.
- **Energy Management** aims to reduce energy consumption by recording, monitoring and optimizing the current state to achieve economic and ecological targets.

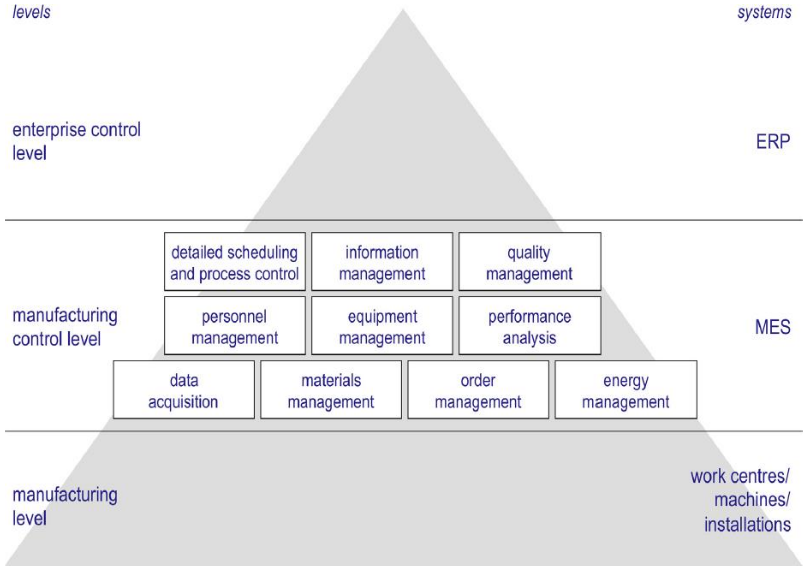


Figure 13: Overview of MES Functionality within the Automation Pyramid[76]

MES modules find application in various manufacturing and business activities within the company, ultimately forming a matrix of functions interfacing with corporate processes. On top of operations planning, production, material handling or maintenance, cross-departmental activities such as the continuous improvement process or controlling can also be supported by an MES solution.

[76] VDI 5600-1, 2016, p. 8

2.5 TU Wien Industry 4.0 Pilot Factory

Aiming to ease the adaptation of Austria's manufacturing industry to future challenges, the concept of Industry 4.0 Pilot Factories was developed by the Austrian Ministry for Transport, Innovation and Technology (BMVIT) as a part of its Production of the Future initiative.[77]

A Pilot Factory is a realistic model of a smart production facility – a laboratory equipped with real machines and systems making real products, but operated without the goal of commercial profitability. It is supposed to offer a research, development, test and demonstration environment to aid in the creation of new manufacturing methods, technologies and processes in a realistic atmosphere, while avoiding the risk of negative influence on commercially operated factories of industrial stakeholders.[78]

With the commissioning of the first Industry 4.0 Pilot Factory, funded, built and equipped in a cooperation between institutes of TU Wien, public institutions and industrial partners, possibilities for gathering hands-on experience and conducting research and development in the fields of smart production and cyber-physical systems were created. The main focus of the facility is the creation of concepts and solutions for flexible and adaptable discrete manufacturing of highly variable, individualized and customized products, with research currently concentrating on four main application fields:[79, 80]

- Adaptive Production Systems/Flexible Manufacturing Cells
- Cyber-Physical Assembly Systems
- Adaptive Logistics Systems
- IT Integration and the Digital Twin

Figure 14: Logo of the TU Wien Industry 4.0 Pilot Factory

The core part of the Pilot Factory is a production system, where components are machined and then utilized alongside externally sourced parts in the assembly of final products – two variants of Fused Deposition Modeling 3D printers. A 3D printer is an especially fitting product for a Pilot Factory, as it is composed of a combination of

[77] see FFG, 2016, p. 8
[78] see FFG, 2016, p. 9
[79] see FFG, 2016, p. 25
[80] see http://pilotfabrik.tuwien.ac.at/inhalte/anwendungsfelder/

mechanical, electronic and software components, with open possibilities for configuration and customization. Manufacturing, logistic and assembly processes can all be simulated within the production system, with additional technologies in the sense of Industry 4.0 introduced and tested on a regular basis.

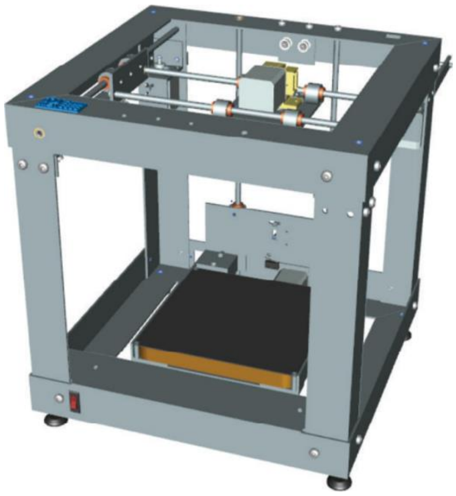


Figure 15: CAD Model of a 3D Printer Produced in the TU Wien Industry 4.0 Pilot Factory[81]

Assembly Line Layout

The Pilot Factory's assembly line consists of multiple assembly stations and their underlying logistic support elements – a storage area and an autonomous ground vehicle (AGV) system (Figure 16). This constellation is based on technologies from multiple vendors, utilizing various sensor types and communicating over an array of interfaces and network protocols.

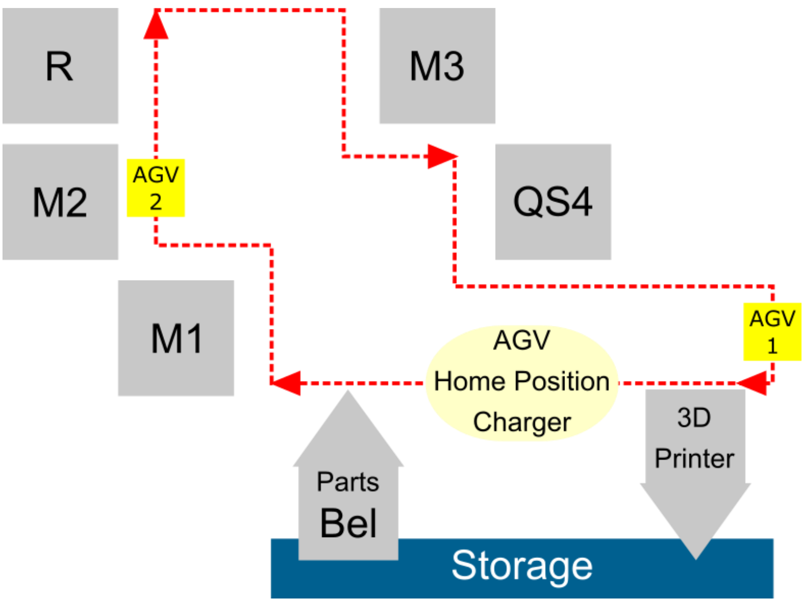


Figure 16: Pilot Factory Assembly Line Layout

[81] Schinagl, 2017, p. 69

As an AGV follows a production order through all stages between its creation and a finished printer, it is a valuable source of operational data and can be used to illustrate the assembly sequence.

Using this approach, an assembly run begins in the AGV home position, where vehicles await new production orders. After receiving an order from the MES, an AGV positions itself to the loading station (Bel), where an employee places components of the printer chassis onto it, and marks the completion of this task in a worker guidance system. Same for all following stations, this is the signal for the vehicle to move on to the next waypoint, approaching assembly stations (M1, M2, M3), a robot station (R) and the final quality check (QS4), where individual production steps are completed in a specified order. Following the final inspection, the 3D printer is removed from the AGV by an employee and placed into storage in the warehouse area. Closing the assembly loop, the vehicle now continues to its home position, where either the next production order is already awaiting, or batteries can be charged while waiting for it.

Neobotix MP-400 Autonomous Ground Vehicle[82]

The MP-400 is a wirelessly communicating automated ground vehicle (mobile robot platform), designed for load carrying in intralogistics. It is able to move on pre-determined, software defined pathways in indoor industrial environments, plan routes between waypoints, as well as to detect and evade obstacles on its course. An MP-400 can navigate autonomously, comparing inputs from a laser scanner and (optional) ultrasonic sensors with an internal map to determine its location, or follow fixed routes. The drive configuration is based around two separately powered wheels positioned on the sides, with smaller castor wheels in the corners maintaining stability. Steering is realized by individual variation of the rotational speed of the drive wheels. Batteries can be recharged in a docking-style automatic charging station, with their full capacity sufficient for approximately 10 hours of continuous operation, or driving distance of 8 km. The AGV is managed by an on-board platform computer, able to communicate operational data and receive commands wirelessly over the network.

[82] see Neobotix, 2017

Figure 17: CAD Model of a Neobotix MP-400 Autonomous Ground Vehicle[83]

2.6 Key Performance Indicators

An organization often needs to compare its day to day performance with longer-term strategic objectives. This way critical points can be discovered, improvement potentials identified and later communicated to employees and other stakeholders. Generally, the insight and explanatory power increases with the amount of available information, but in order to achieve an optimal result the type, frequency and way of performance analysis is crucial. Data collected without a wider understanding of consequences cannot be utilized to its full potential. Misuse and misinterpretation of performance measures might take place, because sadly an improvement of a performance indicator not always equals to a real improvement in the performance of a monitored process.

As not every indicator is equally important for gaining overview of financial, administrative or productive processes within an enterprise, a categorization using measure types is suggested by Parmenter:[84]

- Result Indicators: usually financial metrics, visualizing how well different parts of an organization are working together, without clearly recognizable responsibility for the results.
- Key Result Indicators: summary measures, achieved by multiple stakeholders, generally reported too late to be used for operational management.
- Performance Indicators: non-financial metrics that can be traced back to their source (team, department).
- **Key Performance Indicators (KPI)**: indicators focusing on aspects of performance critical for the current and future success of the organization.

[83] Neobotix, 2017, p. 1
[84] Parmenter, 2015, p. 4

Typical characteristics of KPIs according to Parmenter can be summarized as follows:[85]

- Non-financial nature: looking into processes deeper than the final sales results.
- Timely: monitored frequently, perhaps even in real time, in order to enable swift response and performance improvement.
- Management focus: assuring the metrics are taken seriously by involved employees and keeping the decision makers informed.
- Simple: clear course of action visible to achieve better results.
- Team-based: clear responsibility for the results.
- Significant impact: one KPI influencing multiple performance targets.
- Limited dark side: outcome of measurements and reaction in line with planned goals, limited potential for misinterpretation and abuse.

Because of the commonality of performance measurement in the corporate environment and the diversity of business and productive operations across companies, there is no universally valid consensus on the implementation of Key Performance Indicator measurements. For example, many sources and use cases also use cost-oriented metrics (Result Indicators) alongside Key Performance Indicators.[86]

When working with KPIs, the number of actually reported metrics is an important consideration. Too few indicators result in an incomplete picture, too many are bound to confuse the observer, create redundancies and limit the potential for reaction and process optimization. Different authors recommend varying amounts of KPIs to be measured, with the absolute maximum laying at about 30 metrics, and between 5 and 10 indicators seen as a reasonable bandwidth for most applications.[87] This amount depends on the specific use case and thus the diversity and scope of the observed system or business.

2.7 Data Visualization and Dashboards

Current digitalization efforts in manufacturing rely on a widespread automation of everyday operations, as well as an information-based optimization of production processes. While the delegation of activities and responsibilities to computers offers significant advantages and potentials, the human elements of a manufacturing system still need to be precisely informed about the current situation. In a dynamic industrial environment, it is becoming more and more difficult to recognize critical and actionable information in the large amount of numbers and figures generated by digital assets.

[85] see Parmenter, 2015, p. 11
[86] see Arnold et al., 2008, p. 247
[87] see Parmenter, 2015, pp. 21, 206

As raw data is not especially user friendly, visualization techniques have to be employed to deliver a clear and understandable message to stakeholders. A powerful way of displaying operational as well as strategic data is the dashboard, as defined by Few:[88]

"A dashboard is a visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the information can be monitored at a glance."

Based on his definition, several key characteristics of data visualization through dashboards can be recognized:[89]

- Dashboards are a form of presentation – transforming data into consumable information, helping people to visually identify trends, patterns and anomalies and guiding them towards effective decisions. They can be used for various types of information, and are often found in finance, marketing, as well as manufacturing.
- While graphics present the core part of a visualization, text elements might also be used to convey a message. Quantitative as well as non-quantitative information can be displayed with the goal to increase awareness, transparency, and to draw attention to points requiring action. Ideally, a dashboard should also be able to provide additional information and guidance whenever action is needed.
- All elements of a dashboard should fit on a single screen, preferring display mechanisms that are taking up as little space as possible. As a result, there should be no need for scrolling or otherwise manipulating the display in order to see a particular piece of information. The data can however be split into multiple dashboards in case a greater level of detail is needed.
- A dashboard should be customized for the use by a specific person, group or role in order to optimally deliver its message. A single stakeholder might thus need multiple dashboards to cover all aspects of his position.

[88] Few, 2006, p. 34
[89] see Few, 2006, p. 34

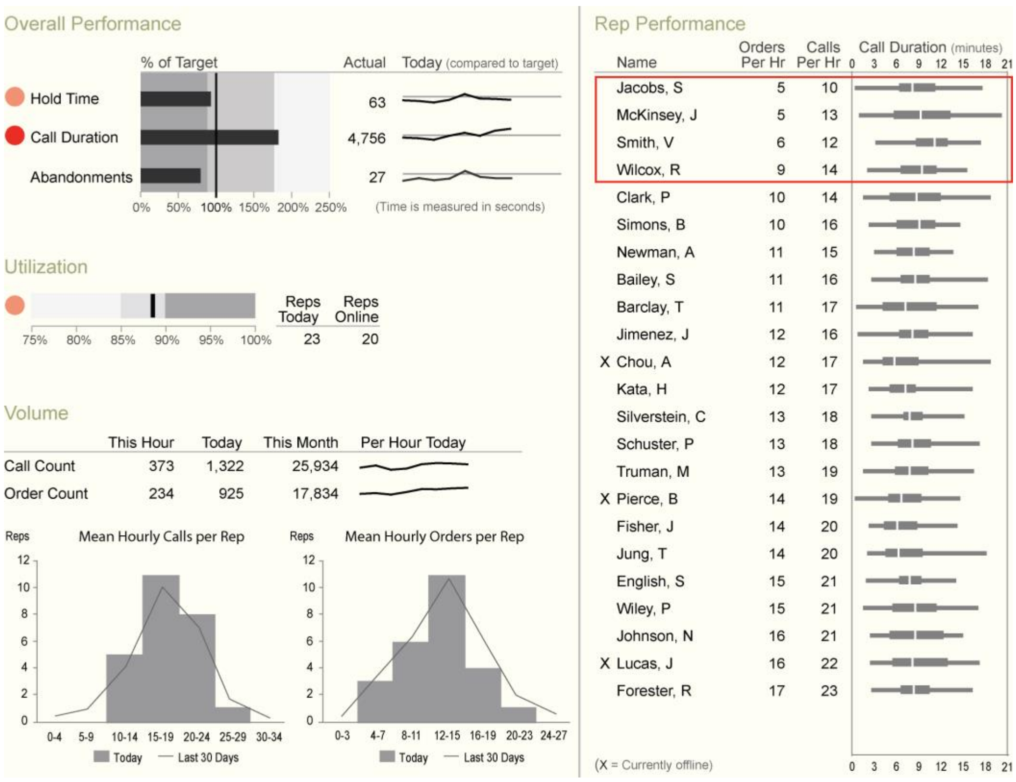


Figure 18: Sample Dashboard[90]

While the implementation of a dashboard rarely presents a technological challenge, its usefulness highly depends on the correct choice of information to be displayed, as well as the way how the information is visualized. The layout and graphical design of a dashboard is thus a major factor deciding about its future value and ease of use. On top of hard knowledge about data analysis, processing and programming, basic understanding of topics such as graphical design, colour theory and cognitive psychology can be beneficial to avoid common design mistakes and unleash the full potential of a dashboard visualization:

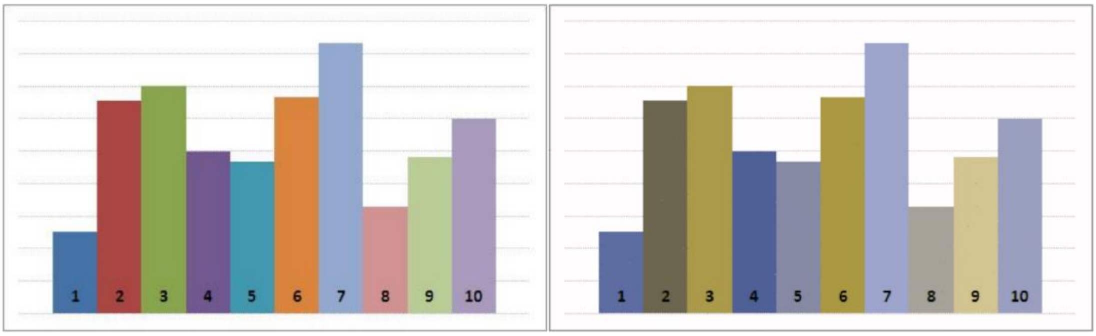
- The same data can be displayed using a multitude of formats with varying informative value. For the application in dashboards, bar charts and box plots are generally preferred over pie charts, because it is much easier for the human brain to judge lengths than areas and angles.[91]
- Some visual elements are more efficient than others in delivering a message to the audience. In examples from the industry, graphical objects in the form of traffic lights and gauges are commonly used. Looking away from their esthetical value, they are generally using up more screen space compared to alternative, more efficient visualization techniques. In Figure 19, the same metric is displayed using a gauge first, and then once more in a bar chart below. The

[90] Few, 2006, p. 199
[91] see Few, 2006, p. 58

contrast in information density is obvious, and additional context is provided with a sparkline chart in the lower example.

Figure 19: Different Space Requirements of Dashboard Elements[92]

- The reliance on contrasting colour to differentiate between two pieces of information is not ideal, as up to 8% of the male population suffer from some form of colour vision deficiency. For them, the typical combination of green (good) and red (bad) might not be distinguishable at all (Figure 20). Instead, scientifically based, colour-blind friendly palettes might be used.[93] Other options include the use of light and dark shades of the same colour, or the placement of additional graphical hints next to measures where action is needed.



Default colours

Deuteranomaly perspective

Figure 20: Colours as Seen by a Colour-Blind Person[94]

- Decorations, background fills and logos all have a negative impact on the clarity of displayed information and cost valuable screen space. Graphical objects needed for navigation and to give context to the visualized data (legends, axis descriptions) should be executed in an unobtrusive way, using lighter and less obvious colours and styles to keep the viewer's attention on the important data.

92 Few, 2007, p. 15
93 https://venngage.com/blog/color-blind-friendly-palette/
94 https://venngage.com/blog/color-blind-friendly-palette/

For these reasons, a flat and minimalistic design should be preferred, a trend also seen in recent graphical user interfaces of major software suppliers. Depending on the particular use case and user preference, the colour scheme might be inverted and a dashboard with dark background and light content created. This form is especially useful when a visualization has to be used in environments with low ambient lighting.

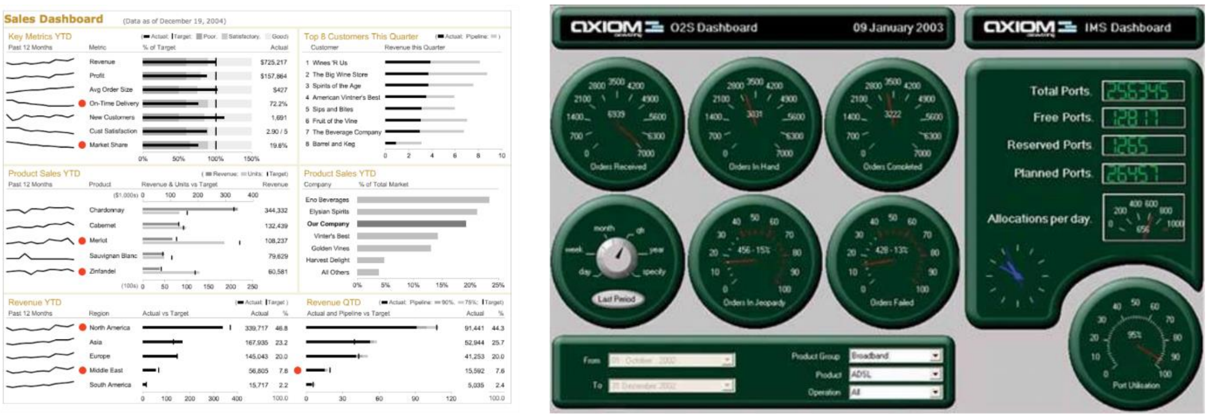


Figure 21: Examples of Good and Bad Practices in Dashboard Design[95,96]

- Data can be put into context by providing comparisons with related measures. This can be achieved by displaying averages, trends, shares and target values alongside the observed measures, providing insight into the current situation and showcasing areas where improvement is needed. Especially suited for trend visualizations are sparkline diagrams – simple line charts, without any axes and markers (bottom left in Figure 19), described as *“data-intense, design-simple, word-size graphics”*.[97] Their informational value is thus lower compared to a full-scale scientific chart, but sufficient to illustrate the direction in which a measure has been developing over time.
- As a speciality of dashboards for operational purposes, the aspect of time plays a greater role compared to presentations visualizing long-term strategic indicators. When immediate reaction to an event is required, the demands for the clarity of displayed data are higher. The level of detail for communicated information has to be chosen carefully to provide insight, but also to avoid overwhelming the observer with unnecessary precision.

[95] Few, 2006, p. 177
[96] Few, 2006, p. 5
[97] Few, 2006, p. 140

3 KPI Research

As a conclusion of the theoretical part, a literature research was carried out to identify commonly used performance indicators, and create a pool of metrics potentially suited for visualization in the dashboard tool. The goal of this work package was to create an extensive system of measures with clear hierarchy and categorization to support a subsequent selection process.

3.1 Boundary Conditions

Before the actual research, the scope of processes and employee roles considered for dashboard development was defined. These boundary conditions can be translated into a three-dimensional classification matrix, where each potential performance indicator is connected to one or more asset types, employee roles, and assigned to a colour-coded top-level category (Figure 22). As an additional dimension, each of the researched performance indicators was placed on one of six levels (tiers), based on its position within the hierarchy. Simple and directly measurable metrics such as Actual Production Time belong to the lowest tier (T6), while indicators utilizing them as inputs are placed one step higher, converging all the way up in global measures such as Total Effective Equipment Performance, residing on the top tier (T1).

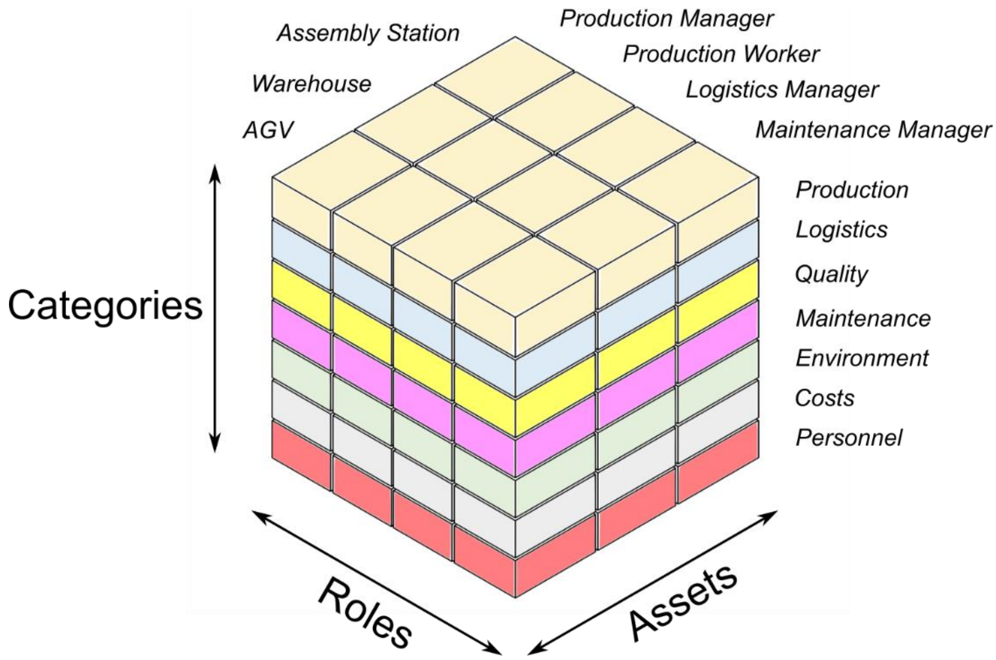


Figure 22: Classification of Performance Indicators

Assets

- Autonomous Ground Vehicle (AGV)
- Assembly Station / Robot
- Warehouse

Three asset types were selected, with the aim of representing manufacturing and logistic processes of the Pilot Factory assembly line.

The assembly station plays a special role in this context, as it was used to represent both a manned workplace, and an industrial robot station. For reasons of simplicity, it has been regarded as an atomic unit, and not further split into individual tools. It was thus assumed, that there are no differences present between the choice of performance indicators relevant for an assembly station and for an industrial robot station, and that all information relevant for their operational assessment can be received from observing the particular work station as a whole.

Employee Roles

- Production Worker
- Production Manager
- Logistics Manager
- Maintenance Manager

Four employee roles belonging to different departments and positioned on various levels of hierarchy were considered for the Pilot Factory dashboard, with the intention to differentiate visualizations and introduce an aspect of information security into the development process. Personalized dashboards were employed to both reflect the varying information scopes required for individual stakeholder roles on the shop floor level and in operational management, and to protect potentially confidential information from unauthorized use.

Top-level Categories

- Production
- Logistics
- Quality
- Maintenance
- Environment
- Costs
- Personnel

The top-level categories, partially corresponding with departments usually found in enterprises, serve the quick sorting of performance measures and enable a balanced choice of indicator categories for the dashboard, resulting in a more informative visualization. On top of categories directly connected with value creation and asset operation, environmental, monetary and personnel metrics were considered in the research phase.

3.2 Research Execution

The main sources used during the literature research were industrial standards, where comprehensive systems of performance indicators including their dependencies and relationships are presented. The most relevant document in this sense is the ISO 22400-2[98] standard (Key performance indicators (KPIs) for manufacturing operations management – Part 2: Definitions and descriptions), alongside with the VDI4400-2[99] (Logistic Indicators for Production) and VDMA 66412-1[100] (Manufacturing Execution Systems (MES) Indicators). Other suitable sources for KPI examples are the Supply Chain Operations Reference Model (SCOR)[101], as well as other publications and papers focusing either directly on performance measurement[102, 103], or more broadly on various aspects of logistics[104, 105], production[106], quality management, etc.

Due to the nature of performance measurement, overlaps in the definitions of measurements between individual publications and authors are present. In case more approaches to a similar indicator were suggested in literature, the one with most relevance for a discrete production facility was chosen. It is also possible to define one and the same indicator from different points of view: warehouse inventory can be reported in pieces, occupied weight, volume, monetary value, etc. For this reason, it is very difficult to create an exhaustive list of performance indicators, and at some point there is little added value potential hidden in incorporating new metrics into the system.

KPI systems actually in use at real enterprises are sometimes available, as is the case with Siemens.[107] While they might offer an interesting perspective into the presumed best practices of renowned companies, their validity for the specific case of the Pilot Factory remains questionable due to expected considerable differences in business processes and the overall way of operation.

Each researched performance indicator was defined by its name, acronym, a short verbal description and original source. Furthermore, its position within the classification

[98] ISO 22400-2, 2014
[99] VDI 4400-2, 2004
[100] VDMA 66412-1, 2009
[101] Supply Chain Council, 2012
[102] Zhu et al., 2017
[103] Hofer, 2015
[104] Arnold et al., 2008
[105] Wannenwetsch, 2014
[106] Kletti and Schumacher, 2014
[107] see Arnold et al., 2008, p. 246

system (asset, employee role, category, tier), a calculation formula (if applicable), unit of measurement and the desired update frequency were provided. The resulting set of performance metrics has the form of a structured table with filtering functionality.

A list of all 149 metrics and calculation inputs originating from 20 literature sources is available in the Appendix, Section 10.1. Diagrams reflecting the final choice of metrics for the dashboard application are provided for each of the observed Pilot Factory assets in Section 5.1.

3.3 Visualization of Results

Thanks to a clear hierarchy of input factors, it was possible to create a tree diagram showing performance indicators in context with their dependencies. In order to automate its creation, a Python script was used in connection with the yUML[108] online tool to transform data from the tabular KPI overview into plain text and finally into a customizable UML-style object diagram.

Due to the amount of researched performance metrics and the need to adapt their theoretical definitions found in literature for the Pilot Factory use case, the overall tree diagram can only be used for illustrative purposes (Figure 23).

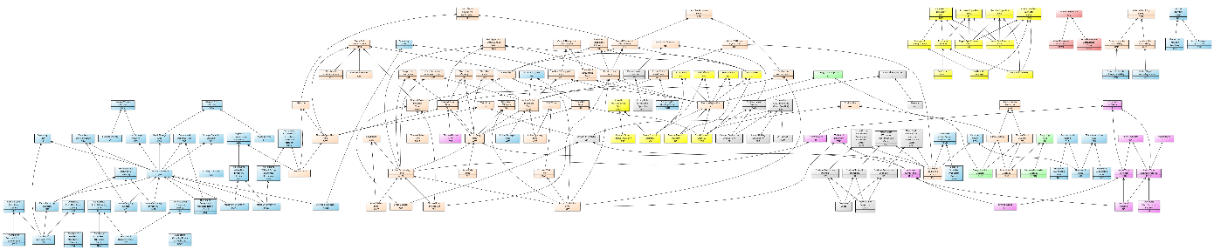


Figure 23: Overview of Researched KPIs and Their Input Factors

[108] see https://yuml.me/

4 Pilot Factory Information Flows

In order to put theory into practice and facilitate the processing of operational data, information flows within the Pilot Factory were analysed, resulting in the identification of sources able to provide values for performance indicator calculation and monitoring. The currently available scope of information, as well as possibilities for the inclusion of additional data sources were investigated.

4.1 Technology Landscape

A wide variety of network-enabled assets is currently operating on the Pilot Factory shop floor. These devices fulfil various roles in production, assembly and logistics, and are able to generate and process operational data.

A multitude of communication protocols and data formats is used to forward these values, and the technology landscape occasionally changes due to the research and development character of the facility. For these reasons, the provisioning of raw data for modules of the dashboard application presents a challenge, as the currently available range of information might not be available in the future, communication protocols might change, or additional data sources become accessible. It is thus beneficial to employ an intermediary system to handle communication with shop floor assets, consolidate the values in accordance with specified rules and serve as an asset-agnostic, single-point source of operational data to assure long-term functionality of the decision support tool.

In the Pilot Factory use case, the Digital Twin simulation model of the assembly line is an ideal candidate for this role, as its primary function relies on the collection of real-time performance values in order to facilitate monitoring of manufacturing processes, make realistic predictions of future states and optimize the manufacturing system. This way, existing interfaces can be utilized, impacts of uncertainty minimized, and the process of information sourcing greatly simplified from the point of view of the dashboard tool.

The general concept of information sourcing is visualized in Figure 24, where the Digital Twin receives raw data from the industrial robot, autonomous vehicles and warehouse assets before saving it into a database accessed by modules of the dashboard application.

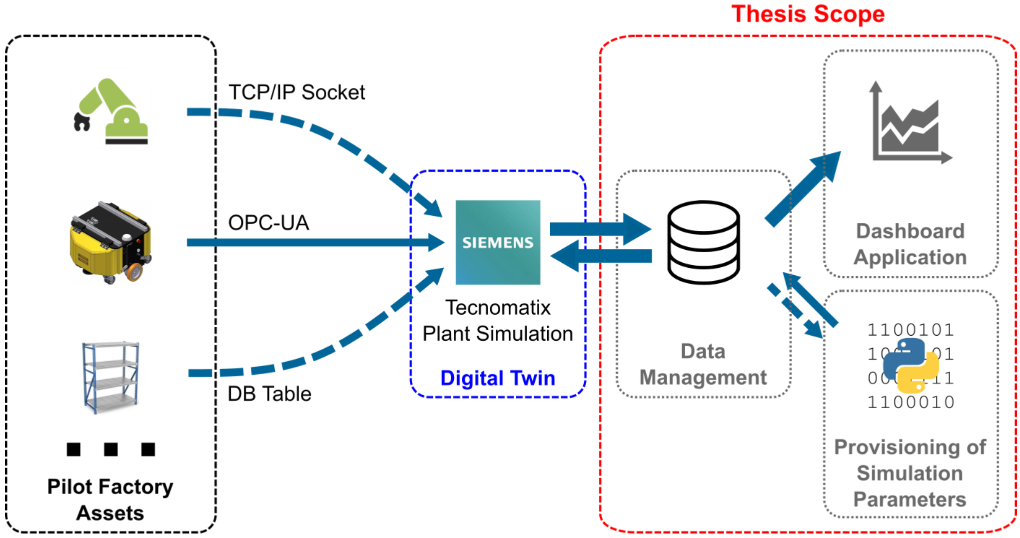


Figure 24: Concept of Information Flows

4.2 Data Availability

Following the draft of communication procedures between data-generating assets and higher-level components of the technology landscape, the extent of currently communicated asset properties and states was investigated.

In this regard, two issues with raw data sourcing were identified, limiting the scope of information available to the Digital Twin, and by extension to the dashboard application:

- **Design of the data interface** for Neobotix autonomous vehicles, where the current position and status of vehicles is not communicated.
- **Non-existence of data interfaces** in the case of assembly station and warehouse assets.

The technically possible range of reported operational data is thus significantly greater than the currently available range, caused by the state of data handling interfaces in use at the Pilot Factory.

An improvement of this situation would require additional research, adaptation and creation of output interfaces for native assets and intermediate information systems, so that the raw data would become visible for the Digital Twin. In addition to the extra effort needed for the inclusion of currently non-existing data sources, the necessary adaptations to existing control logic might present a risk for the regular operation of the Pilot Factory. Because of these reasons, no attempts at extending the scope of available operational data were carried out in the scope of this thesis.

Resulting from this situation, it became clear that the existing data interfaces are not able to provide all the inputs needed for the calculation of complex performance indicators, that are expected to be the most valuable in terms of providing insight into the way of operation of the assembly line.

Data availability thus joined the perceived importance of individual performance metrics as an additional consideration for the selection of KPIs to be visualized in the dashboard application.

4.3 Potentials for Improvement

The current state of communication interfaces between devices in the Pilot Factory is presented in Table 4, with green fields indicating an existing data exchange between two assets. In order to improve the informational value of the decision support tool, additional interfaces have to be created, making the operational states of the respective devices visible for the Digital Twin.

Focusing on the three asset types to be implemented in the dashboard tool, multiple opportunities for sourcing additional data are open:

- AGV System: Improvement of the platform control script to report vehicle state and current position[109], possibly in connection with the local positioning system.
- Assembly Station/Industrial Robot: Creation of an interface with the worker guidance system and the robot controller, in order to receive manual inputs from workers as well as automated status reports concerning production times and breakdowns.
- Warehouse: Making the inventory levels transparent for the Digital Twin, as a replacement for the currently existing manual input.

In a further stage, work on the inclusion of additional asset types can be considered, visualising aspects of quality and enabling a more detailed view of processes at individual assembly stations.

Finally, the visualization of performance indicators might be made accessible through innovative assets such as augmented reality tools, with the goal of increasing the situational awareness of stakeholders.

[109] see Neobotix, 2010, p. 26

Elemente	Nexo 1	AGV 1	WMT115	MC9200	HoloLens	PickGlasses	Tablet QM 1	IMWTablet 1	GlassMontage	HTMMontage	Raspberry Pi Gewicht-QS	Universal Robots UR5	FANUC CR-7iA	PC IMW Montage 1	Server Armbruster	Server Sarissa LPS1	Dell Inspirion RFID	HP Dashboard 4 Display	Sick Inspector Camera	RECA RFID 1	Server Node-RED	Server RoHuC	IMWS Deep-Learn box	Toshiba RFID Laptop	Kinexon NUC Server	SAP
Nexo 1		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
AGV 1	0		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
WMT115	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
MC9200	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
HoloLens	0	0	0	0		0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
PickGlasses	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Tablet QM 1	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
IMWTablet 1	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
GlassMontage	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HTMMontage	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Raspberry Pi Gewicht-QS	0	0	0	0	0	0	0	0	0	0		0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Universal Robots UR5	0	0	0	0	0	0	0	0	0	0	0		0	0	1	0	0	0	1	0	1	0	0	0	0	0
FANUC CR-7iA	0	0	0	0	0	0	0	0	0	0	0	0		0	1	0	0	0	1	0	1	0	0	0	0	0
PC IMW Montage 1	0	0	0	0	0	0	0	0	0	0	0	0	0		1	0	0	0	0	0	1	0	0	0	0	0
Server Armbruster Sequ	1	1	0	0	1	0	0	0	0	0	1	1	1	1		0	0	0	0	0	1	1	0	0	0	0
Server Sarissa LPS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	1	0	0	0	0	0
Dell Inspirion RFID	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
HP Dashboard 4 Display	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
Sick Inspector Camera	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0		0	0	0	0	0	0	0
RECA RFID 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0
Server Node-RED	1	1	0	0	0	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0		1	0	0	0	1
Server RoHuC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1		0	0	0	0
IMWS Deep-Learn box	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
Toshiba RFID Laptop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Kinexon NUC Server	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
SAP	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	

Table 4: Matrix of Existing Interfaces[110]

[110] Provided by Fraunhofer Austria

5 KPI Selection

With the options for sourcing operational information disclosed, a set of performance indicators to be included in the dashboard tool was selected out of the set of metrics researched during the theoretical phase. The importance of individual metrics for performance assessment and optimization of the Pilot Factory assembly line, as well as the availability of input data were taken into account, with the goal to create a picture of the observed system as complete as possible under current circumstances.

The selection was carried out in two steps, and separately for each of the three asset types:

- The number of indicators considered for each asset type was reduced by applying objective criteria intended to remove redundancies and filter out metrics not applicable for the Pilot Factory use case.
- The final selection was carried out by experts at Fraunhofer Austria, who assigned importance points to KPI-employee role pairs in a questionnaire.

In this chapter, an overview of metrics chosen for implementation is followed by definitions of input factors and performance indicators.

5.1 Selection Results

The selection process was influenced by the accessibility of operational data, resulting in an initially unexpected lack of higher-level metrics in the final set. As it has not been investigated which performance indicators would have been selected under optimal conditions, it can be only estimated that the selection would have been different in that case.

At this point, the future extendibility of the dashboard application became a factor for consideration to allow the accommodation of additional metrics as the underlying operational data becomes available. The need for a robust and flexible data handling infrastructure, able to accommodate performance metrics of various types and update frequencies was recognized as a design requirement.

Metrics marked with a green frame are directly visible in the dashboard application, with colour codes for categories adapted from Figure 22.

Autonomous Ground Vehicle Asset Type

The AGV system plays a central role in the performance indicators system, due to the amount of information describing the assembly process it is able to provide. As an autonomous vehicle is accompanying a product through the whole assembly line, order execution times as well as processing times at particular assembly stations can be derived from the data reported by an AGV.

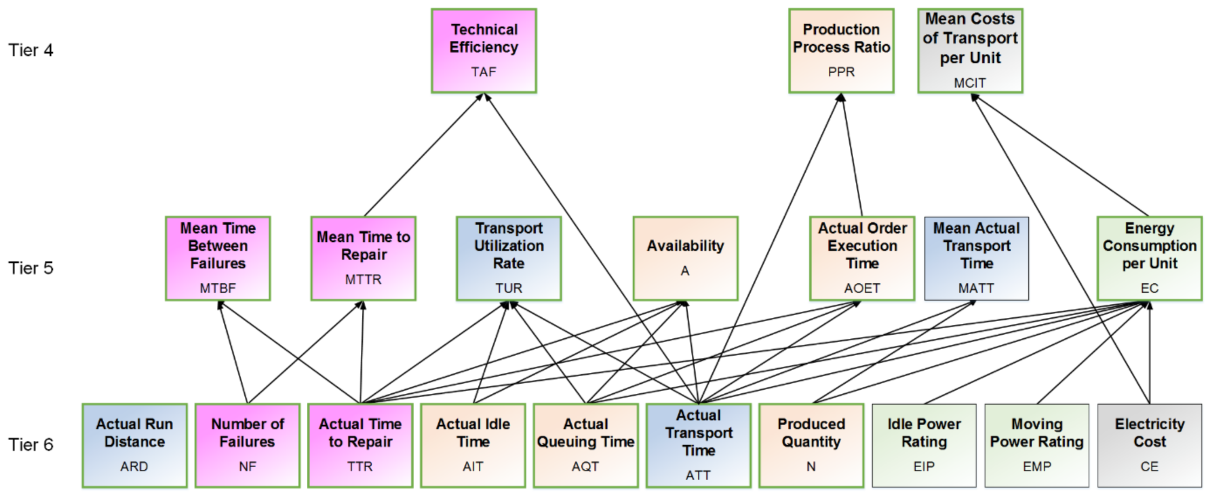


Figure 25: AGV System – Selected KPIs, their Inputs and Relationships

Assembly Station Asset Type

Due to the inaccessibility of operational data generated by assets of the assembly station class (robot, worker assistance system), only values received from AGV assets were used to calculate station-specific performance indicators. Under these circumstances, the goal was to provide a structure for data handling and visualization with an outlook towards a future implementation of richer and more specific data sources.

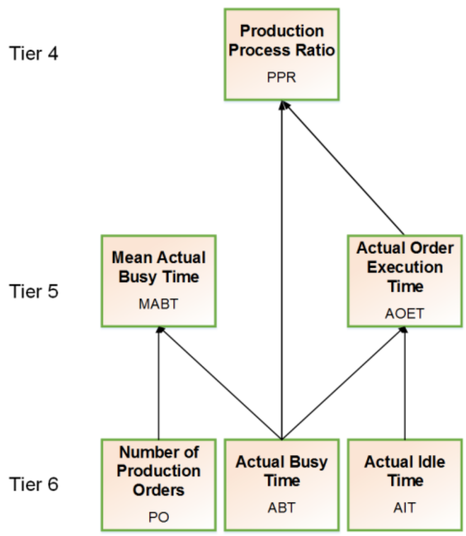


Figure 26: Assembly Station – Selected KPIs, their Inputs and Relationships

Warehouse Asset Type

Logistical metrics were provided on the basis of current inventory levels, available as manual inputs in the Digital Twin, while expecting a full integration with the ERP solution in the future.

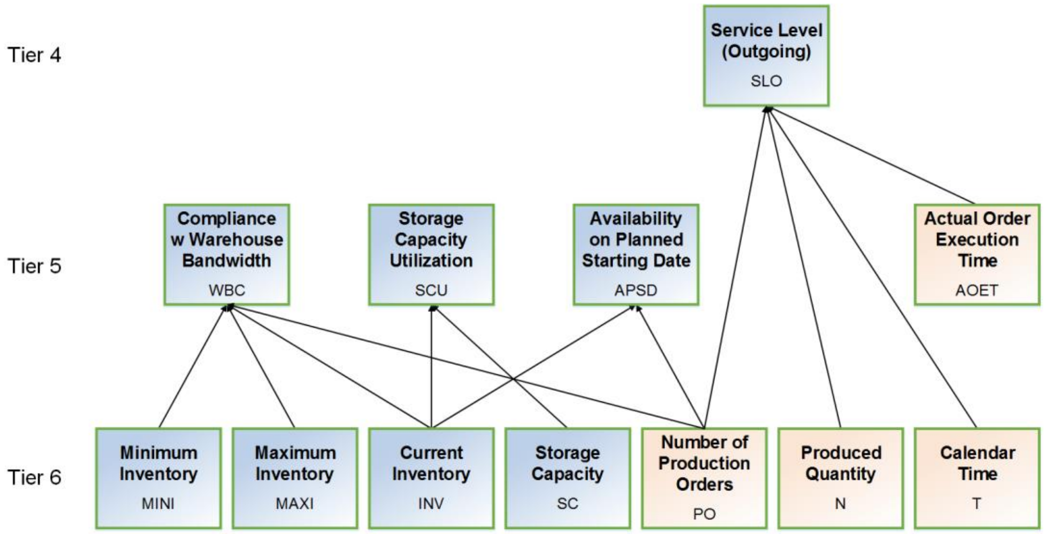


Figure 27: Warehouse – Selected KPIs, their Inputs and Relationships

5.2 Time Model

Many performance indicators utilize time as an input variable, either to mark the occurrence of a discrete event, or to describe the duration of processes and intervals. A comprehensive time model was defined in order to achieve consistency in measurement and enable comparisons between values, using the ISO 22400 standard as a basis, with certain simplifications performed to more accurately represent the use case and better match the available operational information.

The essential concept behind this time model is the breakdown of continuous time segments into intervals fitting for the description of a current asset state. Planned times can be differentiated from actual process durations, and both a bottom-up view can be achieved by summing up the corresponding time segments, as well as a top-down view realized by subtracting individual intervals.

The most significant simplification made for the Pilot Factory dashboard lies in the disregard for planned down times and setup times during the planned operation time of the assembly line. Using this premise, the necessary maintenance and preparation of assets may take place either outside of the planned operation time, or during idle times of the particular work unit. Following this decision, the planned busy time of a work unit equals to its planned operation time, and the actual processing time of an asset is the same as its actual production time (Figure 28).

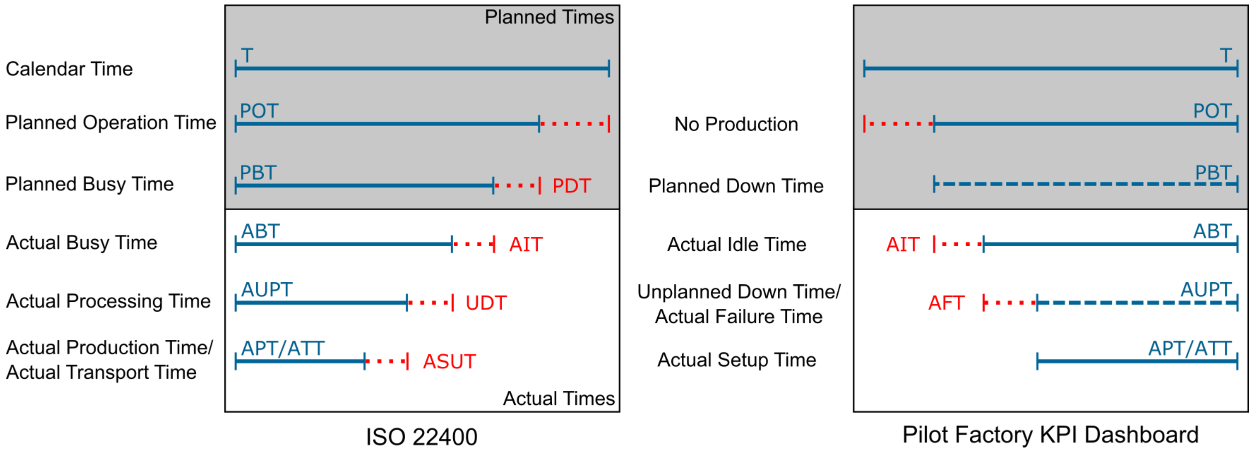


Figure 28: Time Model for Pilot Factory Assets[111]

Calendar Time

Calendar time represents a timeframe for all events and intervals, as well as a dimension to express performance indicators per unit of time. It runs without interruption, independently of the operating state of the Pilot Factory and is identical with the local time at the Pilot Factory's location. The smallest resolution used in the dashboard application is one second.

Planned Operation Time (POT)[112]

The planned operation time is a part of the calendar time scheduled for production and represents the timespan between the beginning and end of a production shift within the Pilot Factory. This interval generally spans from 8:00:00 to 16:00:00 on regular working days (Monday to Friday), excluding weekends and public holidays.[113]

Planned Busy Time (PBT)

Equals to planned operation time in the simplified time model.

Planned Down Time (PDT)

Does not exist in the simplified time model.

Actual Busy Time (ABT)

Equals to the actual production time plus unplanned down time (bottom-up view), or the planned busy time minus the actual idle time (top-down).

---

111 see ISO 22400-2, 2014, p. 11
112 ISO 22400-2, 2014, p. 7
113 https://www.wien.gv.at/ikt/egov/gesetzliche-feiertage.html

Actual Transport Time (ATT)

Corresponds with the duration of AGV movements, excluding breakdowns, idle times and queuing times.

Actual Idle Time (AIT)

Represents the part of planned operation time not used for production, even though the asset is ready. Equivalent with the Actual unit down time of ISO 22400-2.

Actual Queuing Time (AQT)

In the case of autonomous vehicles, idle times spent while waiting at an assembly station are counted separately. Due to the setup of the assembly line, these are inevitable in normal operations and thus need to be differentiated from true idle intervals, when the asset is waiting in its home position and not creating any value.

Unplanned Down Time/Actual Failure Time/Actual Time to Repair (AFT/TTR)

Represents the part of planned operation time needed to restore asset functionality after failures. Equivalent with the Actual unit delay time of ISO 22400-2. It is assumed, that all assets can be repaired, and the failure time is thus equivalent to repair time.

Actual Processing Time (AUPT)

Equals to actual production time in the simplified time model.

Actual Production Time/Actual Transport Time (APT/ATT)

Equals to the share of planned operation time dedicated to value creation, either by performing assembly steps, or by moving between stations in the case of the AGV assets.

Actual Setup Time (ASUT)

Does not exist in the simplified time model.

5.3 Other Calculation Inputs

Number of Failures (NF) [1]

Represent the absolute number of events preventing the machine from functioning as required.

Produced Quantity (N) [1]

Indicates the amount of completed production orders.

Number of Production Orders (PO) [1]

Represents the amount of production orders to be completed.

Actual Run Distance (ARD) [m]

Stands for the total distance covered by an AGV in the assembly process.

Idle Power Rating, Moving Power Rating (EIP, EMP) [W]

Describes the power rating of an AGV while stationary and in movement, respectively.

Electricity Cost (CE) [€/kWh]

Represents the current price for a unit of electrical energy.

Current Inventory (INV) [1]

Indicates the number of components currently stored in inventory. As a simplification, parts needed for the completion of an assembly step at an assembly station are grouped to form a component set, and identified by the name of the particular station.

Minimum Inventory, Maximum Inventory (MINI, MAXI) [1]

Defines the lower and upper desired limits for stored component sets.

Storage Capacity (SC) [1]

Describes the total capacity of the storage area as a number of component sets.

5.4 Key Performance Indicators

The values of desired key performance indicators are calculated using the time model and other input factors. For each of the selected metrics, its description, original source, calculation formula and classification are provided.

Actual Order Execution Time Tier 5, [%]			Production	
Original Source ISO 22400-2, 2014, p.8	*Asset*			
	AGV	**Assembly Station**	Warehouse	
Used Formula $AOET = \sum APT + \sum AIT + \sum TTR$	*Role*			
	Production Manager	**Production Worker**	**Logistics Manager**	**Maintenance Manager**
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Time between the start of the first assembly operation and the end of the last one.				

Availability Tier 5, [%]			Production	
Original Source ISO 22400-2, 2014, p.26	*Asset*			
	AGV	Assembly Station	Warehouse	
Original Formula $A = \frac{APT}{PBT} \cdot 100\%$	*Role*			
	Production Manager	**Production Worker**	**Logistics Manager**	**Maintenance Manager**
Used Formula $A = \frac{APT\ (+AIT)}{APT + AIT + TTR} \cdot 100\%$	*Update Frequency*			
	Real-Time	**Action-Based**	Event-Based	Order-Based
Percentage of productive time vs. total planned time. For an AGV, the time spent at an assembly station is counted as productive.				

Energy Consumption per Unit Tier 5, [Wh]				Environment
Original Source ISO 22400-2, 2014, p.16	*Asset*			
	AGV	Assembly Station	Warehouse	
Used Formula $EC = EMP \cdot \left(\sum ATT_i\right) + EIP \cdot \left(\sum AIT_i + \sum TTR_i\right)$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	**Maintenance Manager**
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Consumption of electrical energy measured for the completion of a particular order.				

Mean Actual Busy Time Tier 5, [s]				Production
Original Source Own definition	*Asset*			
	AGV	**Assembly Station**	Warehouse	
Used Formula $MABT = \frac{\sum ABT}{N}$	*Role*			
	Production Manager	**Production Worker**	Logistics Manager	Maintenance Manager
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Mean time spent for assembly operations, calculated for the completed orders in the current view.				

Mean Actual Transport Time Tier 5, [s]				Logistics
Original Source Own definition	*Asset*			
	AGV	Assembly Station	Warehouse	
Used Formula $MATT = \frac{\sum ATT}{N}$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	Maintenance Manager
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Mean time spent for transport between work units, or to and from inventory storage areas, calculated for the completed orders in the current view.				

Mean Costs of Transport per Unit Tier 4, [€]				Costs
Original Source ISO 22400-2, 2014, p.16	*Asset*			
	AGV	Assembly Station	Warehouse	
Used Formula $MCIT = EC \cdot CE$	*Role*			
	Production Manager	Production Worker	Logistics Manager	Maintenance Manager
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Costs of energy consumed for the completion of a production order.				

Production Process Ratio Tier 4, [%]			Production	
Original Source	*Asset*			
ISO 22400-2, 2014, p.31	**AGV**	**Assembly Station**	Warehouse	
Used Formulas	*Role*			
$PPR = \frac{\sum ATT}{AOET} \cdot 100\%$ $PPR = \frac{\sum ABT}{AOET} \cdot 100\%$	**Production Manager**	**Production Worker**	**Logistics Manager**	Maintenance Manager
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Relationship between the production time or transport time and the whole throughput time of a production order.				

Transport Utilization Rate Tier 5, [%]			Logistics	
Original Source	*Asset*			
Wannenwetsch, 2014, p. 696	**AGV**	Assembly Station	Warehouse	
Used Formula	*Role*			
$TUR = \frac{ATT}{POT} \cdot 100\%$	Production Manager	Production Worker	**Logistics Manager**	Maintenance Manager
	Update Frequency			
	Real-Time	**Action-Based**	Event-Based	Order-Based
Actual vs. theoretical utilization of a transport system.				

Maintenance Indicators

A full availability of production and logistic assets sadly cannot be assumed. By employing best practices in maintenance, and by creating redundancies, the number of unplanned outages can be minimized, and thus the performance of the production system raised. However, these measures are connected with additional costs, and the right strategy has to be chosen to achieve the desired performance and reliability goals. In order to correctly judge the situation, data about the occurrence of failures, outages, their reason and probability are important. Maintenance indicators are thus crucial parts in the KPI setup.

Mean Time Between Failures Tier 5, [s]				Maintenance
Original Source ISO 22400-2, 2014, p.52	*Asset*			
	AGV	Assembly Station	Warehouse	
Used Formula $MTBF = \frac{\sum_{i=1}^{n} TBF_i}{NF + 1}$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	**Maintenance Manager**
	Update Frequency			
	Real-Time	Action-Based	**Event-Based**	Order-Based
Mean timespan between the beginning of one failure episode and the beginning of the following one.				

Mean Time to Repair Tier 5, [s]				Maintenance
Original Source ISO 22400-2, 2014, p.52	*Asset*			
	AGV	Assembly Station	Warehouse	
Used Formula $MTTR = \frac{\sum_{i=1}^{n} TTR_i}{NF + 1}$	*Role*			
	Production Manager	Production Worker	Logistics Manager	**Maintenance Manager**
	Update Frequency			
	Real-Time	Action-Based	**Event-Based**	Order-Based
Average time required to restore the function of a machine after failure. It is assumed, that every asset is repairable, and the repair time starts when the failure event took place and ends when normal operation can be restored.				

Technical Efficiency Tier 4, [%]				Maintenance
Original Source	*Asset*			
ISO 22400-2, 2014, p.30	**AGV**	Assembly Station	Warehouse	
Used Formula $TAF = \frac{APT}{APT + UDT} \cdot 100\%$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	**Maintenance Manager**
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Machine utilization considering unplanned down time (breakdowns) only				

Warehouse Indicators

Metrics concerning the numbers of stored components and their use over time can serve as indicators of resource efficiency and quality of the logistic system.

Availability on the Planned Starting Date Tier 5, [%]				Logistics
Original Source	*Asset*			
VDI4400-2:2004, p. 18	AGV	Assembly Station	**Warehouse**	
Used Formula $APSD = \frac{SPO}{PO} \cdot 100\%$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	Maintenance Manager
	Update Frequency			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Percentage of production orders started on time, with inventory levels sufficient for completion.				

Compliance with Warehouse Bandwidth Tier 5, [%]				Logistics
Original Source VDI4400-2:2004, p. 26	*Asset*			
	AGV	Assembly Station	**Warehouse**	
Original Formula $WBC = \frac{DWI}{T_{days}} \cdot 100\%$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	Maintenance Manager
Used Formula $WBC = \frac{NOWB}{N} \cdot 100\%$	*Update Frequency*			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Percentage of production orders started while inventory levels were within specified limits. $NOWB$ indicates the number of production orders with current inventory lying between the specified minimum and maximum on order start.				

Service Level (Outgoing) Tier 4, [%]				Logistics
Original Source VDI4400-2:2004, p. 26	*Asset*			
	AGV	Assembly Station	**Warehouse**	
Original Formula $SLO = \frac{NOIP}{NOI} \cdot 100\%$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	Maintenance Manager
Used Formula $SLO = \frac{NOIP}{N} \cdot 100\%$	*Update Frequency*			
	Real-Time	Action-Based	Event-Based	**Order-Based**
Percentage of production orders completed on time.				

Storage Capacity Utilization Tier 5, [%]				Logistics
Original Source Werner, 2013, p. 342	*Asset*			
	AGV	Assembly Station	**Warehouse**	
Used Formula $SCU = \frac{INV}{SC} \cdot 100\%$	*Role*			
	Production Manager	Production Worker	**Logistics Manager**	Maintenance Manager
	Update Frequency			
	Real-Time	Action-Based	**Event-Based**	Order-Based
Percentage of warehouse capacity currently in use, measured in absolute units.				

6 Dashboard Development

After it had been decided which performance indicators belong into the set of observed measures, the work on the creation of a data visualization tool could commence.

Following the information flow, operational data on the lowest hierarchy level has to be collected from Pilot Factory assets and made available for the Digital Twin simulation model. After the information has been received and consolidated, the results of this first processing step have to be stored for later use and utilized by the consecutive computational and visualization steps within the dashboard application. The output interface of the Digital Twin marks the boundary of this thesis, serving as the source of data for performance indicator calculations and visualization of their results.

As described in Section 2.7, a dashboard is the preferred solution for visualizing dynamically changing numerical values, their interactions and changes over time. Based on this fact, the expected form of the decision support tool was broadly specified. More detailed conditions result from the requirements for the application, and from technological constraints. The development phase can be divided into individual work packages:

- Formulation of requirements
- Definition of the format and structure of the data received from the Digital Twin
- Creation and management of a database
- Calculation of performance indicators
- Use of a suitable presentation method to make the information accessible and understandable to human observers.

6.1 Dashboard Requirements

On the basis of features expected (must-haves) and preferred (nice-to-haves) to be available in the dashboard application, a set of requirements for the technological solution was composed, leading to the choice of a suitable architecture, methods and tools securing an optimal outcome – a data visualization and analysis tool that is efficient in delivering information, reliable, and pleasant to use.

The ambition was to create more than just a digital, real-time equivalent of a physical KPI board traditionally found in the vicinity of production facilities. By using an appropriate visualization method, and supported by a set of user interface controls, the tool can be used to approach information from multiple angles, enabling for a deeper analysis of past performance and its impact on the current situation. These possibilities, however, must not reduce the clarity of displayed information or create any additional obstacles for the users.

To sum up, the development goal was formulated as a visualization tool able to run in an unattended mode (showing the latest performance values with no human input required), as well as offering the option to roll back to a particular moment in the past, or condense the data from a longer timespan into one view, while offering a multi-user environment.

Based on this vision for the final product, as well as the thesis goals, the following requirements were considered:

- Refresh rates in the order of seconds to achieve a real-time view
- Possibility for multiple users assigned to various employee roles to simultaneously access the dashboard
- Ability to visualize a wide variety of metrics with provisions for future extendibility
- Option to individually select and filter the information by each user

More features with the potential to enhance the usability and value of the final product were identified before and during the development process:

- Compatibility with various device platforms and categories: primarily Windows on PC, with options for Android/iOS on mobile devices, or Linux on the Raspberry Pi
- Compatibility with various screen sizes and resolutions
- Options to export or print a data view
- Integration of an access control solution based on user roles
- Possibility to change parameters, customize the dashboard and implement new layout elements in future

6.2 Choice of Technology

A general arrangement of steps and technologies needed to create the final product had been drafted before the actual start of development. Existing interface options of the Digital Twin model, as well as the specified set of dashboard application requirements were used to provide boundary conditions and simplify the choice. Significant factors for the selection of tools and technologies were presented by the availability of know-how, infrastructure and software licenses.

Data Source

Due to the nature of data flowing through the interface (structured numerical values and text), and the requirement for a simultaneous read/write access, a database solution was identified as a suitable choice. In the case of the existing Digital Twin model, SQLite was used as its primary database technology, with the appropriate know-how existing at Fraunhofer Austria.

Thanks to its server-less and lightweight nature, this database management system is well suited to serve as an information source for the dashboard, and can be accepted without limitations.[114] At this point, the upstream boundary condition for the dashboard development was fixed. For reasons of simplicity and commonality, it was also decided to use SQLite databases for all remaining downstream data management processes within the dashboard.

Data Visualization

With the data source format defined, the remaining top-level decision concerns the visualization technology, which in turn has an influence on the way how calculations with the available data are performed. In general, two approaches with their respective positive and negative aspects were considered for dashboard creation:

- Use of a **ready-made data visualization software**: promising an easier creation and distribution of data visualizations, at the price of reduced flexibility, presence of design constraints and potential licensing issues.
- Development of a **custom dashboard application**: offering ultimate flexibility with greater customization options, leveraged by a higher expected effort and higher uncertainty in the design phase.

Initially, a commercially available software package was considered for data processing and visualization, with the intention to simplify and speed up the development process. The preferred solution in this case was Tableau Desktop[115], a tool for business intelligence previously used at Fraunhofer Austria. Tableau as a software platform offers extensive functionality for data analysis and visualization and is able to work with a wide variety of source formats, including databases, spreadsheets and plain text. Integrated functions for data manipulation and mathematical operations are available, comparable in scope to the built-in functions of Microsoft Excel. Interfaces enabling the execution of external code written in Python and R can be utilized, and dashboard creation is possible within a graphical environment, with easy adjustments to the contents and design of diagrams and the final layout. Finally, online and offline publication methods for visualizations are available, depending on the licence.

While Tableau is generally well suited for the creation of dashboards, there are severe limitations present in a use case involving a periodically changing input dataset, where real-time computations and visualization updates need to be executed frequently:

[114] https://www.sqlite.org/docs.html
[115] https://www.tableau.com/products/desktop

- SQLite databases used by the Digital Twin model are not natively supported, resulting in issues with data interpretation, especially in operations involving date and time values.
- The integrated computational capabilities of Tableau are not sufficient for some of the required data operations. For example, the calculation of time interval durations respecting shift times, weekends and holidays is not practically achievable.
- While an interface to external scripts is available, its philosophy makes it not suitable for dynamically changing data sources. It is only possible to pass whole columns of data to external functions, and the return of such a function is interpreted as an additional data column with exactly one value for each row of the original data source. Moreover, the whole column has to be recalculated after a new data entry is detected, resulting in inefficiency, prolonged calculation times and a reduced responsiveness of the dashboard.
- A periodical refresh of the dashboard view is a feature only available in the Server version of Tableau, connected with significant licensing costs. Additionally, the shortest supported refresh period is 15 minutes, pointing to the fact that real-time visualizations are not supported.
- Without the Server licence, the dashboard cannot be published to the web, requiring users to run a Tableau client on their devices.

Taking these limitations into account, the use of Tableau as a tool for real-time dashboard creation would be possible with extensive workarounds only:

- Running all KPI calculations externally
- Filtering results externally
- Refreshing the visualization by forcing a manual refresh command

Such a solution would only be marginally less difficult to implement than a clean-sheet design, while retaining many of the aforementioned constraints. In the end, Tableau as a business intelligence platform would only be used to plot data values, while more suitable and more efficient software alternatives are available for this task.

Taking these reasons into account, it was decided to develop a custom dashboard application, offering greater flexibility and performance. Python 2.7 was selected as the primary technology, due to its strengths in both data science and web applications, as well as the fact that some of the methods needed for KPI calculations had already been implemented in Python at that point, due to tests of the Tableau-Python interface.

Many Python packages with functions relevant for dashboard development are available online and under open-source licences. This has enabled the use of best practices and proven, highly efficient code, while keeping the complexity of programming efforts as low as possible. The central part of the dashboard application was built around Plotly Dash[116], an open-source Python framework intended for the development of web-based analytical and dashboard applications running in a client's browser, while using analytical and data management capabilities of Python on the server side. Dash is thus able to present the results of KPI calculations in a fully customizable visualization, using various diagram types and other pre-made graphical elements, with support for server-side calculations and multiple clients connected simultaneously.

KPI Calculation

As the last major dashboard function to be specified, the way of executing calculation procedures providing values for visualization was examined. This decision was shifted to the last place on purpose, with the intention to adapt to the more limiting dashboard building blocks discussed in previous paragraphs.

Now that Python as a high-level universal programming language has been chosen for data visualization tasks, it can be used to program the necessary calculation routines as well, taking advantage of the full flexibility it offers in terms of data manipulation and numerical operations.

6.3 Dashboard Application Architecture

Based on the technological choices, the inner structure of the dashboard application with its objects, files and relations was defined. An overview of building blocks and data flows within and around the application is visualized in Figure 29.

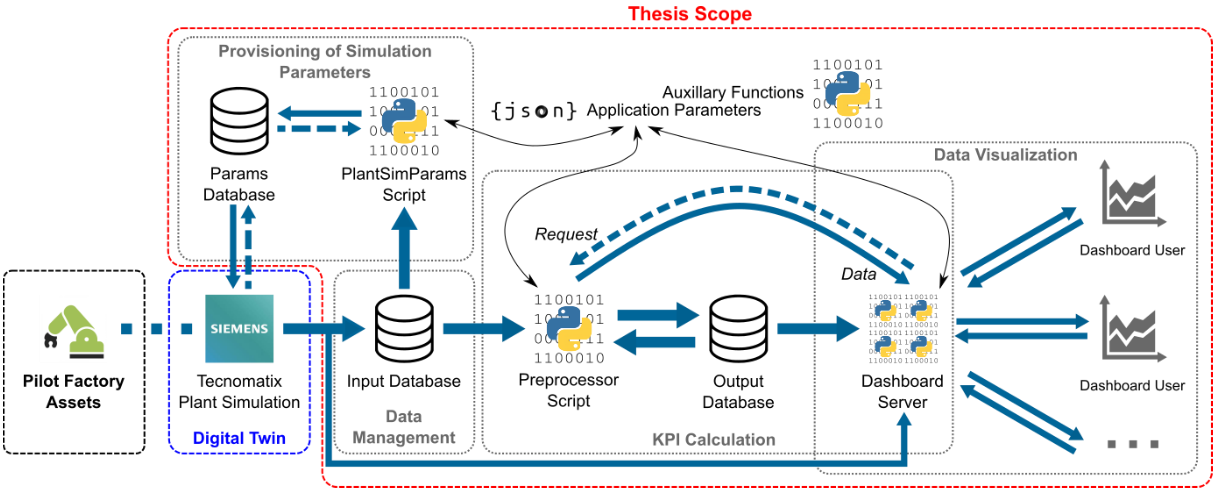


Figure 29: Technology Landscape and Data Flows

Beginning in the bottom left corner of the graphic and outside the scope of this thesis, the communication between assets of the Pilot Factory and the Digital Twin model is depicted. Operational data enters the dashboard ecosystem at the output interface of

[116] https://plot.ly/products/dash/

the Digital Twin, where it is stored within the dashboard input database (Section 6.4). Additional status data that does not need to be processed or stored can bypass the storage and processing modules and continue directly to the dashboard server.

Following the main information flow of the application, data stored in the input database runs through the preprocessor script, which provides the first set of row-based metrics before saving the information in the output database. The same script also contains all algorithms used to calculate KPI values for the dashboard visualizations (Section 6.5).

Requests to execute calculation routines are sent from the dashboard server, which reacts to commands from currently connected users, and displays the resulting information in the form of diagrams and text messages on their devices (Section 6.6). The actual server source code was split into multiple files for increased code readability.

Accessing the input database on behalf of the Digital Twin model, a largely independent module is used to turn data collected from Pilot Factory assets into parameters for future simulations and production optimization – described in Chapter 7.

Additional helper functions and parameter files are employed to assure the functionality of the whole system. All scripts, libraries, graphics and style sheets are stored on and distributed from the dashboard server computer. This approach has two main advantages in comparison to remote loading from online repositories:

- Independence from internet connection
- Guarantee of application stability and reliability by using a fixed and proven version of the underlying software libraries

All modules of the dashboard application are available for download (see Appendix, Section 9810.2).

6.4 Data Management

While the assembly line in the Pilot Factory is running, various devices and systems are continuously generating raw operational data, which can be subsequently used to gain a better understanding of assembly processes, to optimize production with the goal of greater cost and resource efficiency, as well as to enable precise and information-based planning of activities in the manufacturing system. However, this information has to be collected, processed and stored in a specified format before its potential can be utilized by the dashboard application.

As the initial data-related tasks are carried out by assets themselves, systems along the information flow and the Digital Twin (Chapter 4), a database interface is used to connect the dashboard application to operational data (Section 6.2), using a specified data structure to map received values to input variables for calculations.

Furthermore, a file containing global parameters is used to specify the location of data, and also to provide miscellaneous numerical values important for processing and visualizing information:

- Database locations and table names
- IP address of the dashboard server
- Connection to instant messaging
- Distances between assembly stations
- Shift times
- Thresholds for alerts and warnings
- etc.

Information Classes

To reflect the variety of information received from the Pilot Factory, its update triggers and the intended use, a differentiated approach to data processing and storage is needed. Three classes of input information have been proposed for the dashboard, differing in update frequency and flow sequence through the application modules.

Action

The main information carrier used in the dashboard system is an action in the manufacturing system, representing a part of the assembly process with its position in time and space, connection to a specific asset and other values. Examples for an action include a vehicle movement from A to B, or an assembly operation at station C, with each of them described by a row in a table of the input database. The corresponding database entry becomes available as soon as the real-world operation has finished, offering a sufficiently detailed history of movements, events and activities for further analysis.

On the basis of actions, specific events can be filtered out. For example, a failure in the manufacturing system is indicated by a separate status code, and each action interrupted by a failure occurrence is represented by a total of three rows (before – failure – after).

AGV	
ID	Row identifier, used internally for sorting and filtering
ID_Product	Identifier of the production order
OrderNr	Identifier of the sales order
TimeStamp_Start	Starting time of action

TimeStamp_End	Ending time of action
AGV_ID	Vehicle identifier
AGV_Status	Vehicle status (idle, moving)
AGV_Load	Load indicator
StartPos	Starting position of action
EndPos	Ending position of action
Stoerung	Failure indicator and identifier

Table 5: Table Structure for AGV Actions

Due to the limited availability of operational data accessible to the Digital Twin, only information concerning AGV assets can be forwarded to the dashboard application at the current point, with assembly station metrics extrapolated from autonomous vehicle movements. A table structure for station actions has been proposed for future implementation.

Assembly Station	
ID	Row identifier, used internally for sorting and filtering
ID_Product	Identifier of the production order
OrderNr	Identifier of the sales order
TimeStamp_Start	Starting time of action
TimeStamp_End	Ending time of action
Station_ID	Station identifier
Station_Status	Station status (idle, active)
Stoerung	Failure indicator and identifier

Table 6: Table Structure for Assembly Station Actions

Production Order

The list of current and past production orders is accessible in an additional database table, providing information needed to forecast delivery reliability.

Production Order	
ID	Row identifier, used internally for sorting and filtering
OrderNr	Identifier of the production order
ID_Product	Identifier of the sales order
Var	Product variant

TimeStamp_Created	Creation time of production order
TimeStamp_Deadline	Time of promised completion of production order

Table 7: Table Structure for Production Orders

Warehouse Inventory

Inventory levels tied to timestamps are stored in another table, serving as a basis for visualizations of resource use and capacity utilization.

Warehouse Inventory	
ID	Row identifier, used internally for sorting and filtering
ID_Product	Identifier of the production order
OrderNr	Identifier of the sales order
TimeStamp_Real	Timestamp of report
Bel, M1, M2, R, M3	Current inventory levels

Table 8: Table Structure for Warehouse Inventory

Status

The status information class carries value in its original state, is reported periodically and valid until the next update becomes available. These characteristics enable status data to skip storage and processing steps of the dashboard application and to be directly plotted in a visualization. Multiple technologies come into question for handling status data, with database and plain text connections available in the dashboard. Additionally, a network socket interface can be considered for future implementation as a performance boosting measure.

The structure of an asset status report is based on the action information class, except for the omitted start position, and characterized by a single timestamp, as the information describes a point in time instead of a time interval.

Data structure for warehouse status reports is identical to Table 8.

Output Database

Following the data flow, all new information contained in the input database is accessed by the preprocessor module and subsequently stored in an output database for internal use within the dashboard application. This conception serves multiple purposes:

- Enables archiving database entries beyond a single input file
- Reduces the frequency of read/write operations for the input database

- Enables storage of row-based and order-specific KPI values calculated at import (Section 6.5)
- Connects the results with global parameter values valid at the time of calculation

Table structures in the output database are based on input database tables for AGV actions and production orders, with additional columns to accommodate row-based and order specific metrics.

Row-based Metrics, AGV	
Duration	Time difference between start and end timestamps
Distance	Run distance between StartPos and EndPos
TTR	Duration of failed actions

Table 9: Row-based Metrics for AGV Actions

Row-based Metrics, Warehouse Inventory	
CapUtil	Capacity utilization percentage

Table 10: Row-based Metrics for Inventory Levels

Order-specific Metrics	
Planned_Start	Indicates availability of material on order start
As_Promised	Compares planned and real completion times
Warehouse_Bandwidth	Indicates compliance with warehouse bandwidth
Order_Distance	Sum of AGV run distances
Order_Moving_Time, Order_Idle_Time, Order_Failure_Time	Sum of action durations with a particular status
AOET	Actual Order Execution Time
Order_Energy_Cons	Energy consumption based on duration and distance
Order_Cost	Energy cost based on energy consumption
Order_Bel, Order_M1, Order_M2, Order_R, Order_M3, Order_QS4	Duration of processing steps at individual assembly stations

Table 11: Order-specific Metrics

The full workflow dealing with periodical detection of new entries, data management and calculation of row-based and order-specific metrics is illustrated in a flowchart in the Appendix, Section 10.3.

Future Extendibility

The structure of database tables was designed to accommodate all relevant input values for the calculation of currently selected performance indicators, with provisions to include certain additional metrics in the future without the need for column structure alterations.

Preparations have been made to simplify the realization of larger-scale changes by using the definitions of different information classes and the underlying infrastructure for data handling in the dashboard system. This way the inclusion of additional input values for future performance indicator calculations is possible.

However, the perhaps most decisive factor for the dashboard data management is the availability of operational data and ease of its sourcing. The scope of information accessible by the application is largely defined by the native interfaces of individual assets and their ability to forward measured values from internal sensors towards the Digital Twin model (Figure 24).

6.5 KPI Calculation

With operational data available in the database in the right format and structure, the computational steps needed to select, process and condense the information for display can commence. Depending on the character of a particular performance indicator, the conditions for triggering its calculation, the exact procedure, as well as the location and form of results output may vary. Several categories of metrics have been used within the dashboard application, with mechanisms in place to efficiently prepare, store and deliver information to be viewed by a user. A brief overview of the calculation routines is presented in Figure 30.

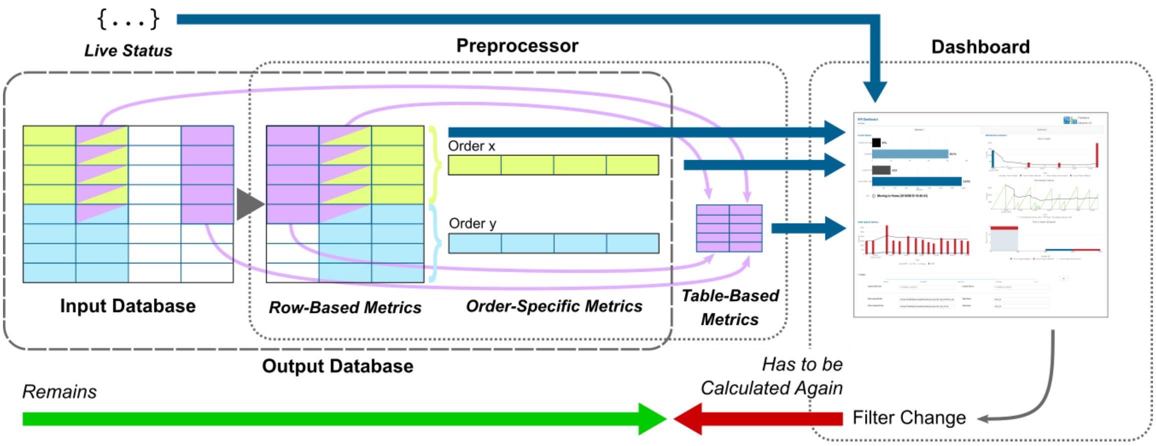


Figure 30: KPI Calculation Workflow

The values of performance indicators displayed on the dashboard are based on the values of operational data received from the Digital Twin, with every piece of information explicitly tied to a particular point of time or time interval. By automatically or manually selecting a timeframe to be visualized (filtering the database by date and time, Section 6.6), only the data falling into this period is taken into account. The source of this information are actions and orders started after the interval starting point and finished before its ending point (Figure 31).

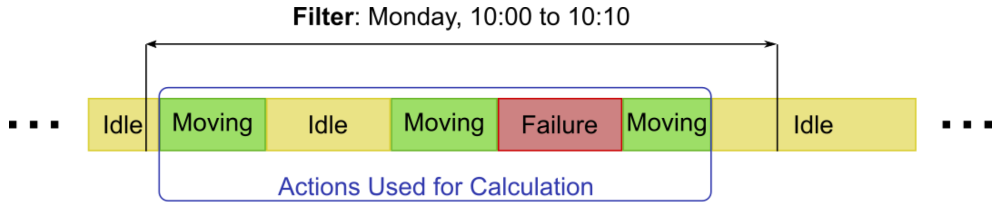


Figure 31: Actions Considered for Calculation

Two factors were considered when designing the infrastructure for KPI calculation and result delivery in a multi-user environment with customizable datasets:

- Not all metrics can be calculated on a one-time basis: many performance indicators take on new values when the observed timeframe shifts.
- Not all calculation results can be stored centrally, as the values are dependent on the individual context, and thus different for individual dashboard users.

In general, when data sourced from more than one database row (representing one action or one production order) is involved, the calculation cannot be executed before a specific request has been received. To reflect the influence of a change in the input dataset on the results of a calculation, a categorization of performance indicators has been introduced:

Row-based Metrics deliver a value resulting from a single action in the manufacturing system (row of the input database), optionally with the help of global parameters. For example, the duration of a movement can be calculated as the time difference between its start and end points, while taking the beginning and end of production shifts into account as parameters. Similarly, the run distance of an AGV is determined by its start and end position in combination with the section length defined as a parameter. These values are not dependent on filter settings, enabling a one-time calculation straight at the time of import, with the result stored directly in the output database to be available for future use.

Order-specific Metrics are based on an aggregation of data entries belonging to one production order, created on order completion. As they are formed as a sum of row-based metrics and only provide meaningful information in the aggregated state, a one-time calculation and storage in the output database is feasible.

Table-based Metrics are computed using information from multiple database entries, and the values of these performance indicators are thus dependent on the selection of source rows that is specified by the currently active filter setting in a dashboard application instance. For this reason, the values of table-based metrics do not stay constant over time, and more importantly might also vary between two dashboard users using the application at the same time. The server-side database cannot be practically used for storing the calculation results, and a different approach is needed. To achieve this functionality, the dashboard instance passes the active filter setting to the preprocessor script before carrying out the necessary calculation and receives the results in the form of a DataFrame – a disposable, database-like Python object containing only the data points needed for visualization.

Category	Storage Location	Calculated	Changes with	
			Filter	Zoom
Row-based	Output DB	Once, at import	No	No
Order-specific	Output DB	Once, on order completion	No	No
Table-based	DataFrame	On dashboard refresh	Yes	No

Table 12: Calculation Types

Various update frequencies have been implemented in order to match the characteristics of individual performance indicators. Some information needs to be reported immediately, some is only able to provide value when summarized and condensed. A balance between informational value and complexity of the technological solution is an important factor as well. Infrastructure for the following update triggers is available in the dashboard application:

- Real-time: updated periodically on a fixed interval (≥1 second)
- Action-based: updated once for every action completed
- Event-based: updated once for every special event (e.g. failure) completed
- Order-based: updated once for every order completed

Common for all calculation routines, the procedure consists of three steps:

- Selection of data from the output database
- Calculation of values (when applicable)
- Provisioning results in the right format

Examples of Calculation Routines

The following example aims to illustrate various types of KPI calculation algorithms on concrete data, as an extension of Figure 30. Table 13 represents an excerpt from the AGV table of a dashboard output database, which stores operational data received from the Digital Twin model, augmented by values of row-based and order-specific metrics calculated by the preprocessor module at the time of import.

Each row of the output database corresponds with an action in the manufacturing system – a movement of an autonomous vehicle in this case. Movements with identical start and end positions represent idle times at the particular station, indicated by a separate status code. Breakdowns can be filtered on values of another code column (Stoerung).

Row-Based Metrics

Two row-based metrics are presented in the example: the duration of actions, determined as a time difference between start and end timestamps, and the Time to Repair, taking on non-zero values in case of failure occurrences. Each failure is represented by a total of three rows (before-during-after). Values of row-based metrics do not change over time, and serve as a basis for further calculations.

Input Database											Row-Based Metrics	
Order Nr	...	TimeStamp_Start	TimeStamp_End	...	AGV_ Status	Start Pos	End Pos	Stoe rung	...	Duration	TTR
122		2018/12/11 09:00:37	2018/12/11 09:01:00		2	QS4	Home	0		23	0
123		2018/12/11 09:01:00	2018/12/11 09:01:10		2	Home	Bel	0		10	0
123		2018/12/11 09:01:10	2018/12/11 09:04:16		2	Home	Bel	1		186	186
123		2018/12/11 09:04:16	2018/12/11 09:04:31		2	Home	Bel	0		15	0
123		2018/12/11 09:04:31	2018/12/11 09:08:11		1	Bel	Bel	0		220	0
123		2018/12/11 09:08:11	2018/12/11 09:08:27		2	Bel	M1	0		16	0
	...			...					...		
123		2018/12/11 09:36:25	2018/12/11 09:36:49		2	QS4	Home	0		24	0
124		2018/12/11 09:36:49	2018/12/11 09:37:13		2	Home	Bel	0		24	0
	...			...					...		
126		2018/12/11 11:32:44	2018/12/11 11:33:07		2	QS4	Home	0		23	0

Table 13: Extract from the AGV Table of an Output Database

Order-Specific Metrics

Actions belonging to one production order (highlighted in green) can be condensed into order-specific metrics as soon as the particular order has been completed, with the results stored in a separate table of the output database (Table 14). An example of calculation routines using SQL pseudo-code is provided in Code Snippet 1.

```
Order_Moving_Time  = SELECT SUM(Duration) … WHERE OrderNr = 123 AND AGV_Status = 2

Order_Idle_Time    = SELECT SUM(Duration) … WHERE OrderNr = 123 AND AGV_Status = 1

Order_Failure_Time = SELECT SUM(Duration) … WHERE OrderNr = 123 AND Stoerung  != 0

...
```

Code Snippet 1: Example for the Calculation of Order-Specific Metrics

OrderNr	TimeStamp_End	Order_Moving_Time	Order_Idle_Time	Order_Failure_Time	...
123	2018/12/11 09:36:49	195	1371	186	...
...	...	...	...	...	...

Table 14: Extract from the Orders Table of an Output Database

Table-Based Metrics

Highlighted in violet (Table 13), columns of the output database may serve as input values for the calculation of table-based metrics. The results of such calculations depend on the selection of input rows (active filter setting), as the information contained in each row has an influence on the end value of the metric. In the example below, entries considered for the sums are filtered using a time criterion – including actions that started after a specified start point and finished before a specified end point. Information provided by this method can be further processed, illustrated by the calculation of the availability rate of an autonomous vehicle in this case.

```
sum_ttr      = SELECT SUM(TTR) … WHERE TimeStamp_Start >= '2018/12/11 09:00:00'
               AND TimeStamp_End<='2018/12/11 11:35:00'

sum_duration = SELECT SUM(Duration) … WHERE TimeStamp_Start >= '2018/12/11 09:00:00'
        AND TimeStamp_End<='2018/12/11 11:35:00'

...

A (TimeStamp_Start >= '2018/12/11 09:00:00', TimeStamp_End<='2018/12/11 11:35:00') =
      = (sum_duration - sum_ttr) / sum_duration * 100

...
```

Code Snippet 2: Example for the Calculation of Table-Based Metrics

A full example of a Python calculation function responsible for turning database entries into the table-based Availability metric is provided in the Appendix, Section 10.4.

6.6 Data Visualization and Dashboard Functionality

Once a part of the information to be displayed is available in the output database, and the remaining data can be provisioned on demand, the next step consists of presenting it to human stakeholders using a dashboard visualization tool, defining and arranging its layout elements and user interface.

User Access and Role Management

As visualized in Figure 29, the dashboard web application is based on a client-server principle. The server process can be started either directly from the Digital Twin user environment, or by manually executing the main Python script. While the dashboard server is running, users can access the client side by navigating to the server's specified URL (IP address and port number) in a web browser of their choice.

At the start of a session, the user is prompted to log in using a user name and password, which is in turn checked against a list on the server. This way a basic level of cyber security is achieved, as no potentially sensitive data is visualized or transferred before the correct credentials have been entered. Based on the user name, the role of the user within the Pilot Factory is determined, and a personalized set of diagrams and visualizations is loaded in their browser window. In addition to the specified user roles, a master account with access to all available information, metrics and visualizations has been implemented.

Access to Data

Without user intervention, the dashboard application periodically monitors an input database, processes its contents, stores it in an output database and visualizes the results (Figure 29), using "Last Day" as the standard filter setting. The default paths to database files are specified in global parameters of the dashboard and are valid for all users. Using this approach, each user is connected to the active data source and able to receive updates in real time.

In case a user wishes to look into a different dataset, he or she can point the application to another database using the dashboard user interface. Two modes of operation are possible:

- **Live connection:** With an input database specified in the appropriate field, the dashboard application will try to reflect any changes in it in the output database. In case this is not possible due to mismatching identifiers and dates, a backup of the output database will be created before starting over with a clear dataset.
- **Viewing connection:** Removing the pointer to the input database in a dashboard instance disables the processing of additional entries, enabling an analysis of historical data.

Figure 32: Dashboard Client Overview (Master Account)

Dashboard Layout

The dashboard client (Figure 32) is arranged into three rows. On the top a header is located, containing the title as well as displaying the currently applied filter setting. If the particular user role has access to more diagrams than can be displayed on one screen, the controls for switching between dashboard tabs are located right underneath the header. In this case, the most important information is condensed on the front tab, with more detailed visualizations located on subsequent tabs for deeper focus on a particular topic.

Most of the screen area is occupied by visualizations – numerical values displayed as graphics and text, with their position, as well as the use of colours following principles of dashboard design presented in Section 2.7. The use of graphical elements not directly needed for presenting information has been minimized, and contrasting colours applied for increased clarity and usability, respecting also the needs of users with colour vision deficiencies. The visualization area consists of three building blocks arranged in two columns, in order to optimally use the available screen resolution.

Beginning on the top left, the most prominent screen space framed in green is reserved for displaying the current state of assets, which is coincidentally the most frequently updated piece of information. In case there is an issue with the connection between the dashboard and its data sources, appropriate information is also displayed in this frame.

Further down a simple overview of the most important metrics is displayed, with the aim to provide actionable information in one spot. The indicators are graphically divided into two groups – percentages and absolute values. Both groups benefit from a colour-coding mechanism, able to highlight the relation of current values to long-term averages, and percentage values are supplemented with sparklines – small-scale line charts visualizing trends leading to the latest values.[117]

The remaining screen area below and to the right of current values, as well as on following tabs is dedicated to larger-scale diagrams focusing on the individual metrics, providing a greater level of detail and supplementary information. An overview of all visualization blocks is available in Section 6.7. Line charts and bar charts were preferred, because of their efficient use of available space, as well as the high level of clarity and readability they offer. Each of these visualizations can be exported individually and a zooming functionality as well as hovering numerical values are included wherever they can provide additional value. Zooming, in contrast to filtering, has no effect on the values of performance indicators (Figure 33).

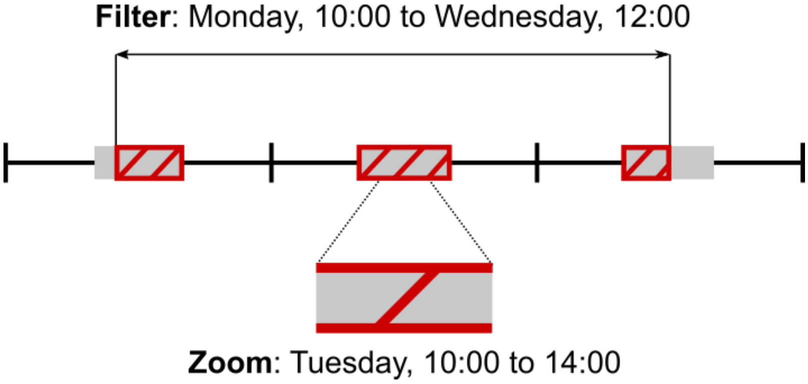


Figure 33: Principle of Filtering and Zooming

As most of the visualized metrics are time-based, the horizontal axis usually serves as the time axis. In case the information to be displayed spans over more than one working day, the visualization automatically switches to a multi-day view, where the non-productive parts of a week are omitted (Figure 34) and thus a higher information density can be achieved.

[117] Few, 2006, p. 140

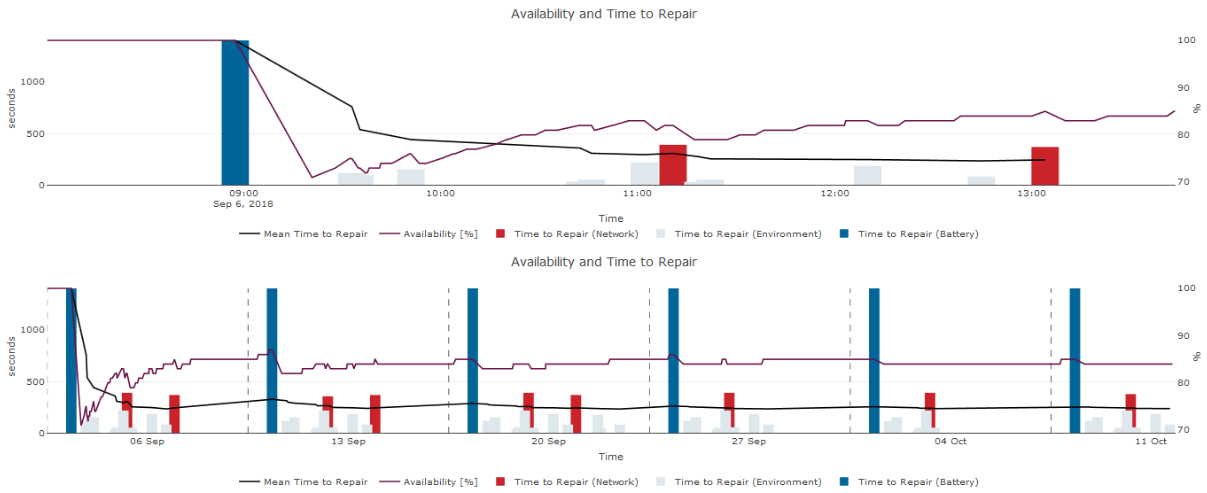


Figure 34: Single-Day and Multi-Day Modes

The third and last row of the dashboard application contains control elements used to change parameters and filter the displayed information. Designed to be hidden by default, the user interface is rendered only on request, in line with the primary use as a visualization tool in an unattended mode. Whenever needed, the filter settings can be changed here, offering pre-defined time intervals (last day, week, month), a custom timespan, as well as the visualization of all the output database contents. Furthermore, the locations of database files can be specified for the current session of the current user. Finally, an option to load the default set of global parameters is provided.

Using the three-row layout, the dashboard header and all diagrams of the selected tab are completely visible when the application is opened in a full-screen window on a 16:9 aspect ratio screen (most current laptops, PC monitors, TV screens and many smartphones). In addition, thanks to the responsive nature of the Dash framework, the individual on-screen elements automatically adapt to a changed resolution and/or aspect ratio, optimally filling all available space.

Alert Messages

As an additional feature meant to increase situational awareness and speed up informed decision making of Pilot Factory employees, the dashboard server is able to send out instant messages using the Telegram platform. While similar to WhatsApp, Telegram supports the use of bots – accounts controlled by third-party applications.[118]

Using a Python interface to control such a bot account, custom messages can be sent out to a group of subscribed users.[119] This way selected stakeholders will receive a notification on their mobile devices as soon as a performance indicator drops below its set threshold, or whenever an event requiring their attention occurs in the Pilot Factory.

[118] https://core.telegram.org/bots
[119] https://python-telegram-bot.org/

By including a link to the dashboard client in the message, the recipients are able to quickly gain a full overview of the situation. The event triggers, alert frequencies, message contents, as well as the list of recipients are fully customizable.

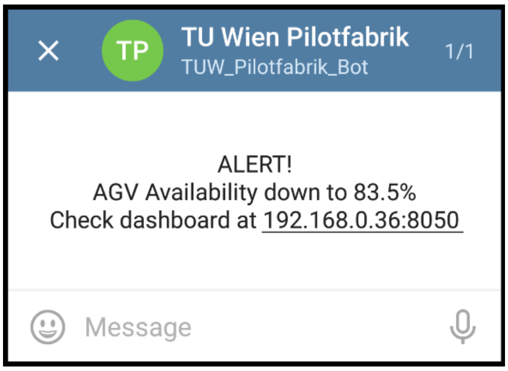


Figure 35: Telegram Alert Message

6.7 Dashboard Elements

In this section individual diagram elements of the dashboard application are presented, using test data from multiple datasets to showcase characteristics and features of individual visualizations. The showcased figures are applicable for the master account, and contain the maximum available amount of information, which can be appropriately scaled down for the needs of Pilot Factory stakeholder roles. An overview of KPI assignment to roles and dashboard elements is available in the Appendix, Section 10.5.

Current Percentages and Sparklines

Focusing on percentage metrics, values in this diagram are supported by colour codes (darker shade equals to lower value), while sparklines on the right can be used to showcase trends in the observed time period. Availability and Transport Utilization Rate are action-based, while the remaining metrics are updated once for every completed order.

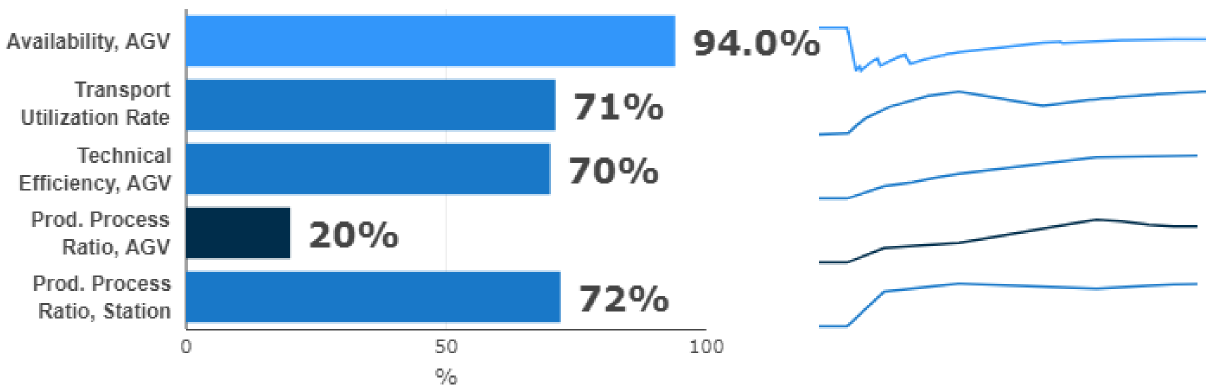


Figure 36: Current Percentages and Sparklines Diagram

Current Percentages (Logistics)

Further percentage values focusing on warehouse logistics can be viewed in a separate diagram, separated from the main element to enable for increased layout flexibility.

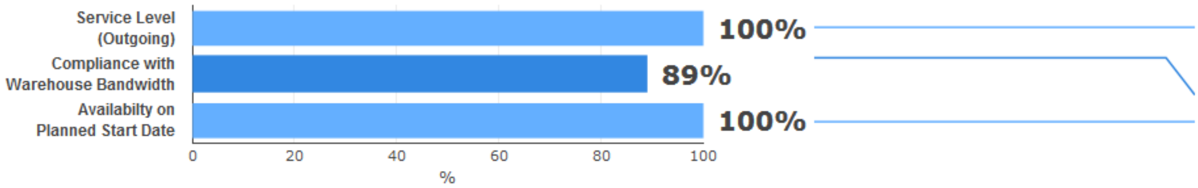


Figure 37: Current Percentages (Logistics) Diagram

Current Values

Other numerical values with the dimension of distance and time can be seen in this visualization. The colour coding mechanism compares current values to long-term averages stored as global parameters. For the metrics with a grey bar, a low value is desired.

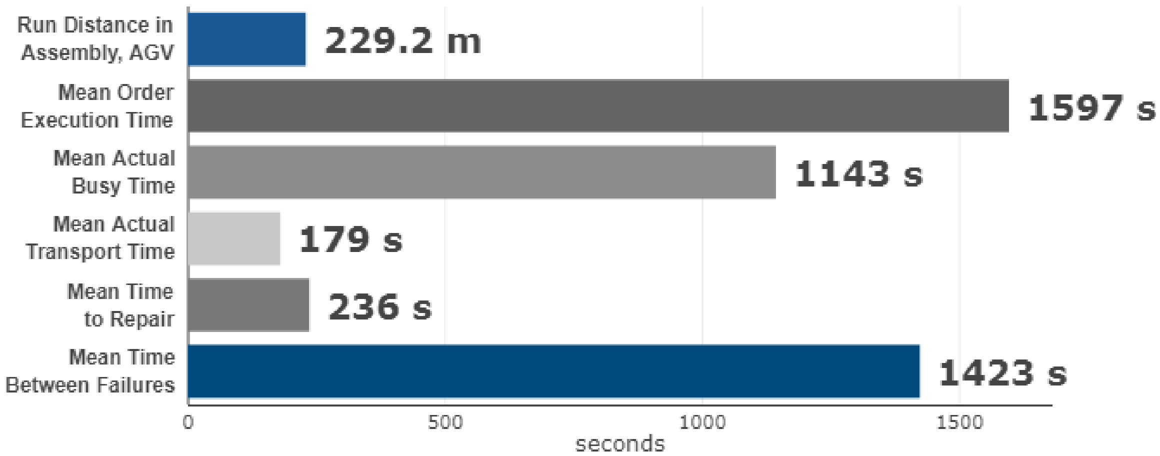


Figure 38: Current Values Diagram

Live Inventory

Current warehouse inventory can be visualized in real time, with the assumption of one component set of each type being used in the assembly of a product. Shades of blue are signalling sufficient amounts, red are values below a specified minimum (red dashed line). The optimal warehouse bandwidth is visualized by two green dotted lines, with the lower limit calculated as a sum of the minimum inventory level and the amount of open production orders. Storage capacity is indicated by a grey dashed line.

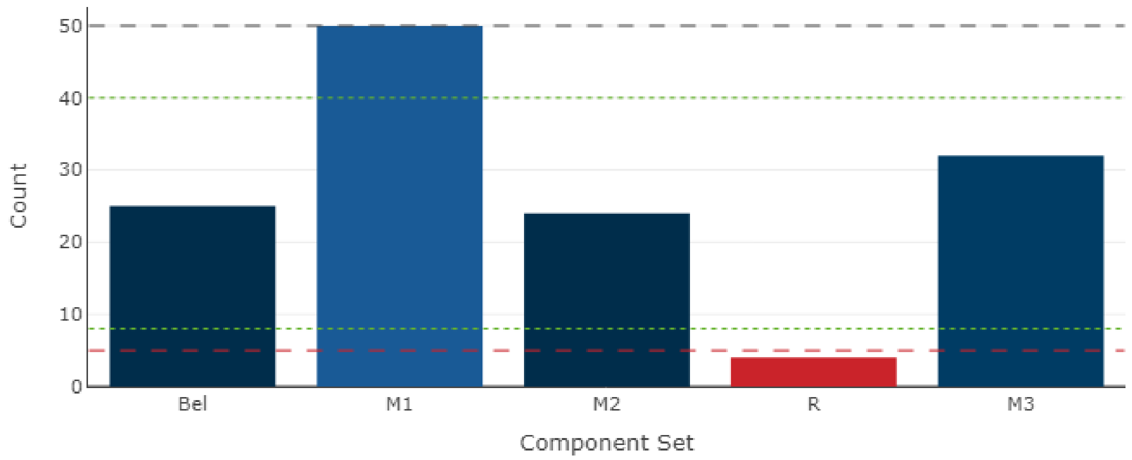


Figure 39: Live Inventory Diagram

Availability and Mean Time to Repair (MTTR)

As the main maintenance indicator for AGV assets, the occurrence of failures can be investigated. Each breakdown is symbolized by a bar, with the x-position symbolizing its end timestamp and height expressing duration. Comparisons with the mean time to repair can be made, and reasons for fluctuations in availability values analysed using the right y-axis.

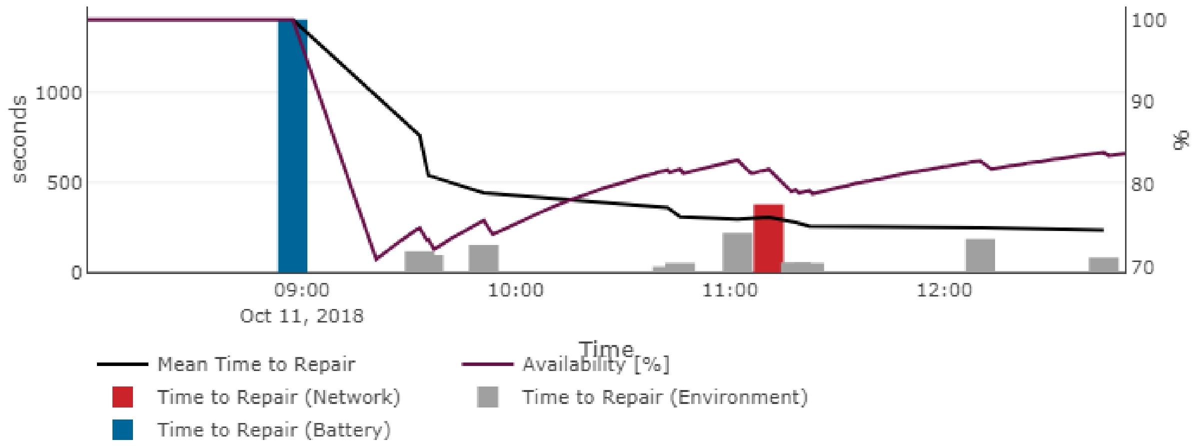


Figure 40: Availability and Mean Time to Repair Diagram

Time to Repair Histogram

In addition to their position in time, the distribution and duration of breakdowns can be visualized as a histogram with colour categories corresponding to breakdown reasons.

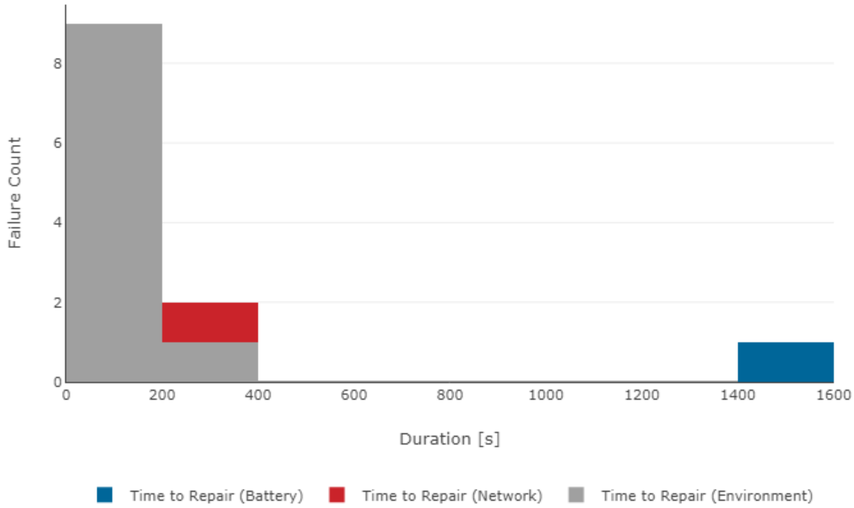


Figure 41: Time to Repair Histogram

Mean Time Between Failures (MTBF)

As a more specialized metric, time between failures can be plotted and also condensed into a mean value.

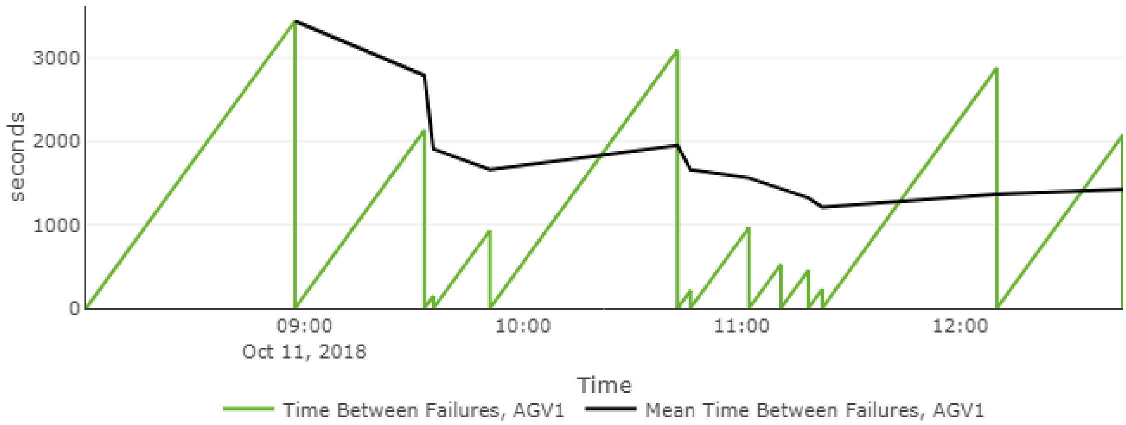


Figure 42: Mean Time Between Failures Diagram

Order-specific Metrics (AGV)

As an autonomous vehicle guides a production order through the whole assembly process, multiple order-specific metrics can be expressed by analysing vehicle movements and states. In this diagram, individual AGV states (queuing, moving, broken) belonging to a production order are summed up to visualize actual order execution times, represented by the height of individual bars. Additionally, idle times spent without an active order are visualized using violet bars. For both cases, the x-position of the bars signifies their position in time. Disregarding the status-based subdivision of orders, a continuous time axis could be created by rotating the bars counter-clockwise by 90 degrees. In their vertical position, a comparison with the mean order execution time, as well as the aggregated production process ratio is possible.

Finally, each order is provided with a value of energy consumption calculated on the basis of operational time.

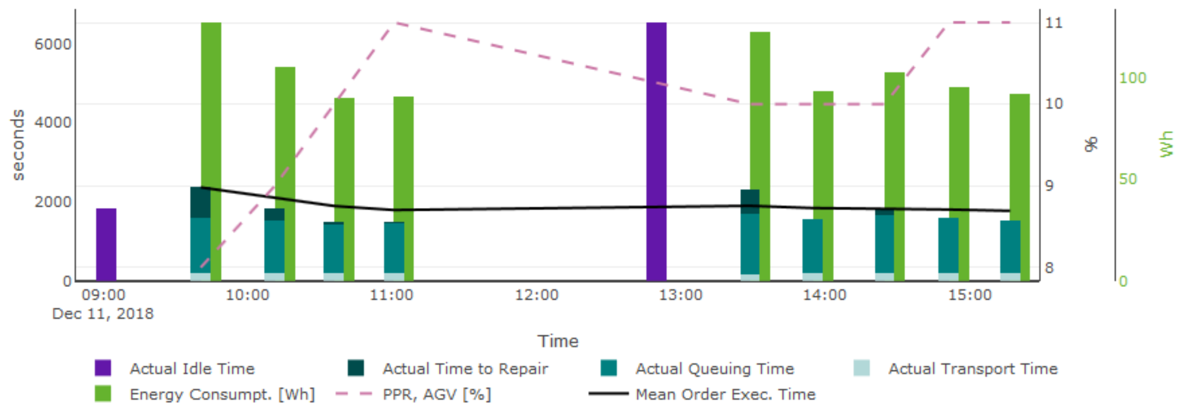


Figure 43: Order-specific Metrics Diagram (AGV)

Order-specific Metrics (Assembly Station)

Performance values concerning assembly station assets are visible in this diagram. In the stacked area chart, the sum of actual busy times at individual stations amounts to the total order-specific busy time. This can be immediately compared with its mean value, and in connection with the actual order execution time it serves the visualization of performance process ratio for the assembly station asset type.

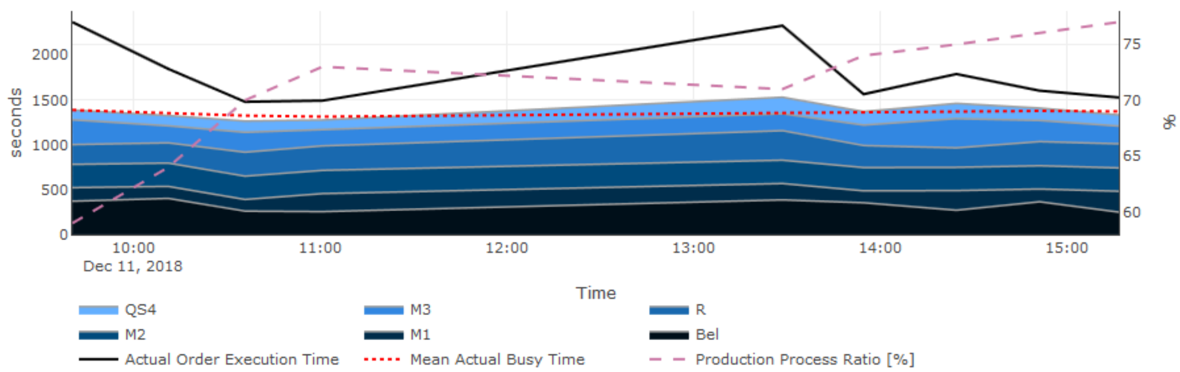


Figure 44: Order-specific Metrics Diagram (Assembly Station)

Overview of Production Orders

The currently open production orders can be viewed including basic information. Additionally, an estimated time of completion is visualized, taking into account the current mean order execution time. In case this estimate exceeds the specified deadline, the particular order is highlighted.

OrderNr	Product ID	Variant	Order Creation	Order Deadline	Predicted Order Completion	OK
1102	1928	1	2018/12/11 10:00:00	2018/12/31 16:00:00	2018/12/11 14:04:16	yes
1102	1635	1	2018/12/11 10:00:00	2018/12/31 16:00:00	2018/12/11 14:39:47	yes
1102	1245	1	2018/12/11 10:00:00	2018/12/31 16:00:00	2018/12/11 15:15:18	yes
1102	1170	1	2018/12/11 10:00:00	2018/12/11 15:00:00	2018/12/11 15:50:49	no
1102	1289	1	2018/12/11 10:00:00	2018/12/11 15:00:00	2018/12/12 08:26:20	no

Figure 45: Production Orders Overview

Inventory Diagram

The development of current inventory levels over time can be viewed and compared to the warehouse capacity in this visualization.

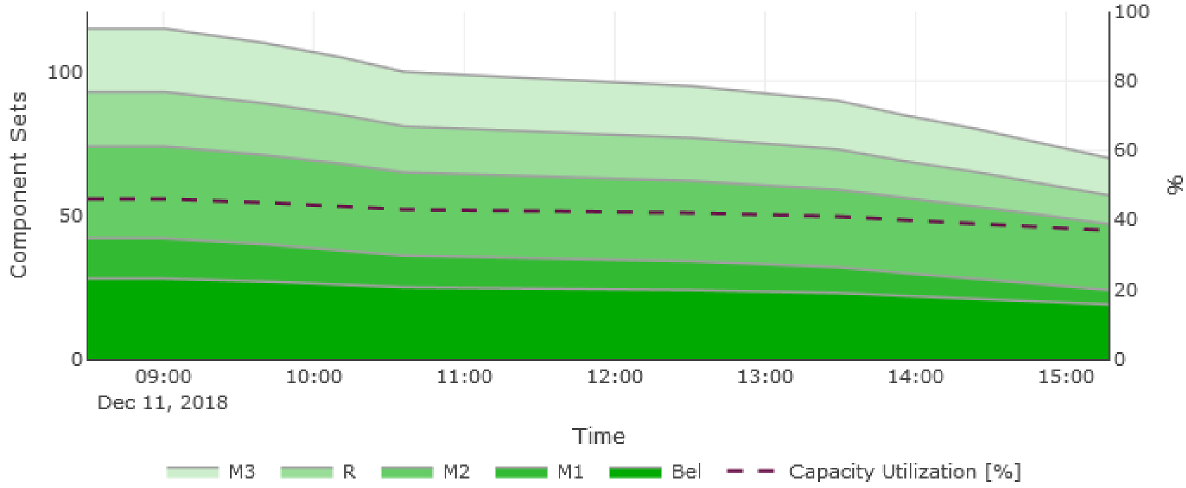


Figure 46: Inventory Diagram

7 Provisioning of Simulation Parameters

A central aspect of a Digital Twin is its ability to utilize real information collected in the past to make realistic assumptions about future states of the observed system. To provide this functionality in the Pilot Factory, operational data from the assembly line has to be analysed and condensed into a few key indicators, serving later to parametrize the simulation model and improve the level of credibility of its outcomes. The process can be compared with a closed-loop control system, in which the simulation parameters act as its control variables (Figure 47). The presence of such a capability makes it possible to project historical and current performance into the near future, and two types of questions can be answered by simulation outputs in this case:

What will the number of finished products be at the end of the day, if we keep up the pace from today's morning?

Will we reach the production target for November, if we produce with the same efficiency as during October? What about last year's November?

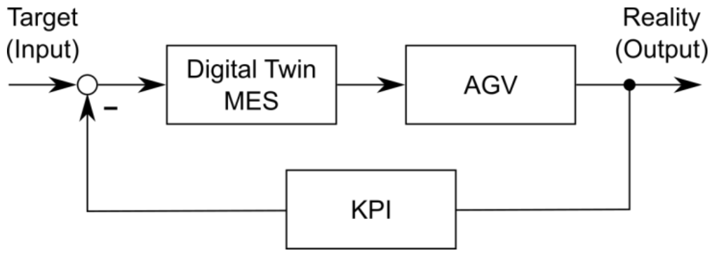


Figure 47: Use of KPIs in a Closed Control Loop

The task of processing past performance into parameters is directly related to the creation of a decision support tool, as both functions are supposed to provide actionable information – once with the Digital Twin as the recipient, and once for operational employees. Similar to the dashboard application, the completion of this side task also begins with a definition of requested outputs and the creation of an interface, enabling seamless communication of the calculation tool with the Digital Twin. Within the dashboard application architecture, the parametrization script has a largely independent position, but is able to benefit from the global parameters and helper functions used by the remaining system parts (Figure 48).

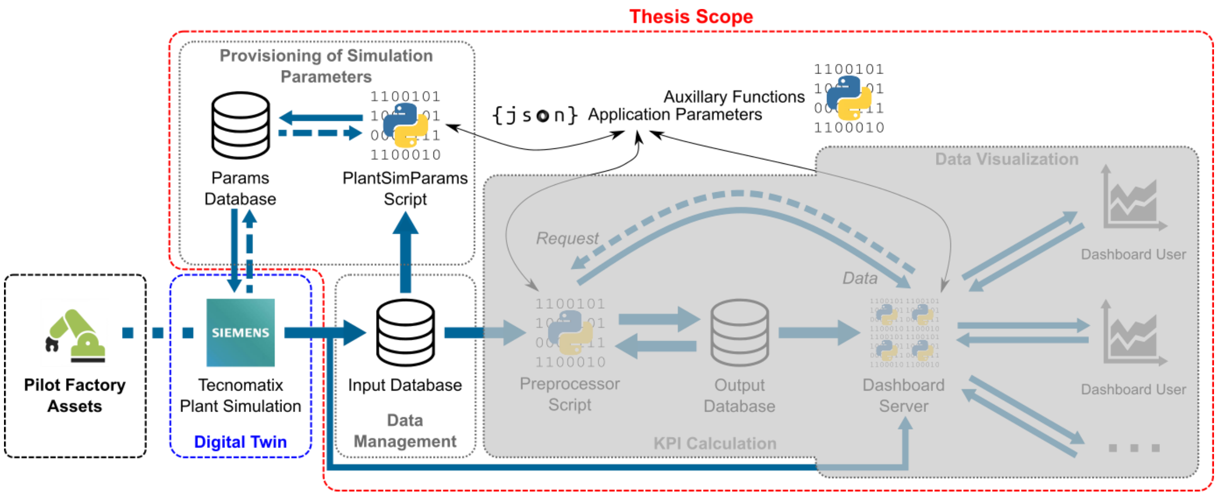


Figure 48: Technology Landscape and Data Flows for the Provisioning of Simulation Parameters

7.1 Choice of Parameters

The choice of individual metrics used for simulation parametrization was dictated by the architecture of the Digital Twin, as the inclusion of additional parameters would require extensive adaptations to the simulation model. The available parametrization variables are:

- AGV Availability
- AGV Velocity
- AGV Mean Time to Repair
  - Network
  - Environment
  - Battery

The values of these performance indicators have to be calculated using the database of Digital Twin outputs (dashboard inputs). To allow for more flexibility when simulating the manufacturing process, two tiers of parameter calculations have been implemented:

- short-term (4 operating hours before a defined timestamp)
- long-term (90 days before a defined timestamp).

A timestamp defining the end of the investigated timeframe can be chosen freely, so that interesting periods from the past, such as days with exceptionally high/low asset availability can be processed.

7.2 Interface with the Digital Twin

The tool for parameter calculation needs to be easily accessible, and ideally controlled directly from the Digital Twin user interface in Plant Simulation. It has been decided to use the existing database interface of the Digital Twin to communicate with the parametrization script. The first reason for this choice is the unification of interfaces with the dashboard application, the second is the possibility of long-term storage of calculation results.

In order to get the desired parameters, three steps have to be carried out on the receiving end (Digital Twin side):

- Requested calculation type and end timestamp specified by writing into the parameters database (Figure 48)
- Calculation script launched using a shell command; completion of the process confirmed when prompted
- Results loaded into the simulation model

7.3 Calculation Procedure

Following the launch of the calculation script, the last entry of the parameters database is loaded, and in case its result fields are empty, the calculation is executed.

Similar to KPI calculations carried out by the dashboard preprocessor (Section 6.5), the procedure consists of three steps:

- Selection of data from the input database
- Calculation of values
- Provisioning of results in the specified format

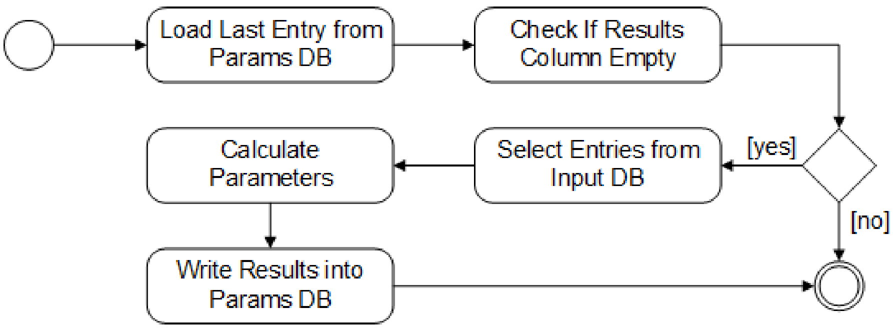


Figure 49: Parameters Calculation Flowchart

After the results have been written to the database, the Digital Twin model is able to use them for parametrization of its Pilot Factory assembly line simulation.

8 Conclusion

In this concluding chapter the fulfilment of thesis goals, the answers to research questions and limitations concerning thesis outcomes are discussed, while possibilities for further development are proposed in the Outlook.

Following an overview of theoretical topics, a list of 149 indicators and input factors relevant for the performance assessment of discrete production facilities was composed, serving as a basis for the choice of metrics to be included in the visualization tool. Additionally, these performance indicators have the potential of being used for the assessment and optimization of similar production lines reaching beyond the dashboard use case.

An analysis of information flows in the Pilot Factory revealed possibilities for the sourcing of operational data, highlighting the role of the Digital Twin model in consolidating data flows, and the benefits of using a single, application-neutral data exchange interface for the dashboard tool. Moreover, the role of the AGV system serving as a progress indicator for production orders has enabled for an extrapolation of additional information. At the same time, the current situation presents a potential for a future inclusion of additional data sources, with the final goal of covering all major aspects of the assembly process by integrating Pilot Factory assets and intermediary information systems with the Digital Twin model.

The selection procedure, consisting of a pre-selection and questionnaire-based final selection, delivered a set of performance indicators to be considered for the use case under the current circumstances. Taking data availability into account, the need for a robust data handling and computational infrastructure, allowing for future upgradability with additional interfaces and metrics was highlighted.

On the basis of selected KPIs, a custom dashboard visualization tool was developed, showcasing not only the real-time performance values, but offering possibilities for the analysis of historical data as well. The application is platform independent, and can be accessed over the network, with an option for multiple users simultaneously viewing information tailored for their roles within the Pilot Factory. The modular layout of the dashboard visualization supports the multi-role approach, and offers potentials for future extensions.

As an additional module, a tool aimed to process operational data into parametrization inputs for the Digital Twin model was created, making it possible to take advantage of values describing past performance of the assembly line, and use them to increase the credibility of predictions made by the simulation model, as well as provide optimization capabilities.

With the development of a dashboard application and simulation parametrization tool, a continuous link between shop floor assets of the Pilot Factory, the Digital Twin model and data visualization was completed. Operational data can now be processed in near-real time before displaying actionable information to human observers, as well as delivering it back to the simulation model to influence future states of the manufacturing system. The contribution to a successful realization of this concept presents the most important outcome of this thesis.

8.1 Answers to Research Questions

Looking back at the research questions formulated in the introduction to this thesis, the achieved results provide appropriate answers:

How can performance indicators based on operational data from production and logistic systems of the Pilot Factory be displayed clearly, taking into account the needs and limitations of various stakeholders?

The performance metrics can be visualized in a dashboard application in the form of several simple diagrams, supported by text information. A clear and easy-to-use layout can be achieved by following best practices for dashboard design. A modular approach to visualization elements in connection with an access control solution enables differentiation between user roles and provides a solid level of data security.

Which of the operational data and performance indicators gained from the observed assets are relevant for a near-real-time evaluation and optimization of manufacturing processes within the Pilot Factory?

Performance indicators to be included in the dashboard tool were selected in a two-step process, taking relevance and data availability into account, with the goal of providing a best possible overview of processes taking place in the assembly line. The choice of metrics provided to the Digital Twin as simulation parameters was dictated by the architecture of the simulation model itself.

What is the shape of information flows within the Pilot Factory, and what possibilities are open for the inclusion of additional data sources?

The Digital Twin model has been used as the sole source of information, consolidating data from various assets of the Pilot Factory before forwarding it to modules of the dashboard application. This way the data flows have been streamlined, and existing interfaces utilized.

The current state and availability of interfaces between shop floor assets, intermediary systems and the simulation model showcases the need for an extension of the scope of accessible operational data. Progress in this area could be achieved through upgrades to existing interfaces, the creation of new ones, and as a last resort through an introduction of additional network-enabled sensors.

How can the collected operational data be efficiently and clearly managed, prepared for use within a simulation model, and how can the outputs from this model be stored in a database?

Raw operational data originating from Pilot Factory assets is processed by the Digital Twin, and made accessible for the dashboard application as either a database entry, or a plain text status update, in accordance with the properties of the respective information class. The created system of database tables is used to store received information as well as calculated values for future use, optimizing performance of the application and supporting simultaneous access by multiple dashboard users. The two-way communication concerning simulation parameters is based around a database table as well.

The existing data handling infrastructure is ready for a future extension, allowing for the implementation of additional data types and metrics in the visualization and parametrization tools.

9 Outlook

With the first version of a calculation and visualization application finalized, information has been made available to selected Pilot Factory stakeholders and the Digital Twin model. Multiple improvement possibilities for this decision support tool concerning usability, extendibility and performance still remain open. Some of these measures aimed to increase the value of the application towards a complete and powerful solution are discussed in this chapter.

In the current state, the set of performance metrics, the procedure for their calculation and visualization, as well as the whole dashboard layout is defined within the application source code. With the introduction of additional configuration files, parameters and an adaptation of existing calculation functions, the layout and contents of dashboard elements might be opened for user customization without the need of programming knowledge. By using a separate software tool (e.g. layout designer), these processes could be greatly simplified, enabling for an ad-hoc inclusion of additional performance indicators. As the data management and calculation infrastructure is already able to accept a variety of calculation types and update triggers, the added value provided by this functionality is expected to be high, with a reasonable effort.

Thanks to the modular structure of the dashboard application, and the single, application-neutral input interface, parts of the software package might be used to provide information in different contexts, such as in connection with a custom Digital Twin model, ideally also based on Python. In this case, the model would be able to directly access the calculation and visualization functions of the dashboard application, resulting in a tighter integration of both systems.

Performance optimization of the dashboard application has not been a development priority, as there were no issues observed during testing, even though the dashboard server has been running on the same dated, consumer-grade hardware as the Digital Twin. However, in case of more than a handful of users simultaneously using the application, the probability of performance-related issues is high. By implementing a caching solution on the server and/or client sides, the amount of data to be computed and then transferred over the network might be reduced to some extent. Further scalability would require the use of more capable server hardware, and at some point also the switch to an enterprise-grade database management system. However, these scenarios are not realistic in the Pilot Factory use case, and performance optimization is thus not a high priority at the moment.

The inclusion of additional information sources would be greatly beneficial for the informational value of the dashboard application. With the communication interfaces of the dashboard tool already able to accept various data formats describing real-time states, actions, as well as whole orders, the main obstacle in this effort lies in the elements between the Pilot Factory shop floor and the dashboard application – the assets themselves, the Digital Twin model and all intermediary information systems. The currently available scope of information that can be sourced from individual assets, processed by the simulation model and made accessible to the dashboard application is smaller than anticipated, resulting in a limited amount of available operational data. An efficient approach to solve this problem could be based around the definition of an optimal set of performance metrics, assuming access to all data describing the manufacturing process, and a subsequent targeted development of necessary interfaces and sensor networks.

The highest development priority in connection with the dashboard application should thus be the configuration of the already accessible and the remaining Pilot Factory assets to increase the amount of reported information, as well as the integration of these data sources with the Digital Twin. This is an extensive task requiring attention from all concerned parties, but its completion is central for the efforts of informed decision making and further optimization of manufacturing and logistic processes within the TU Wien Industry 4.0 Pilot Factory.

10 Appendix

10.1 List of Researched Metrics and Input Factors

Name	Acronym	Source
Actual Busy Time	ABT	ISO 22400-2:2014, p. 8
Actual Idle Time	AIT	ISO 22400-2:2014, p. 8
Actual Order Execution Time	AOET	ISO 22400-2:2014, p. 8
Actual Order Handling Time	AOHT	Kletti and Schumacher, 2014, p. 53
Actual Order Lead Time	AOLT	Kletti and Schumacher, 2014, p. 53
Actual Personnel Attendance Time	APAT	ISO 22400-2:2014, p. 8
Actual Personnel Work Time	APWT	ISO 22400-2:2014, p. 8
Actual Processing Time	AUPT	ISO 22400-2:2014, p. 8
Actual Production Time	APT	ISO 22400-2:2014, p. 8
Actual Queuing Time	AQT	ISO 22400-2:2014, p. 9
Actual Setup Time	ASUT	ISO 22400-2:2014, p. 8
Actual Throughput Rate	AFT	ISO 22400-2:2014, p. 21
Actual to Planned Scrap Rate	APSR	ISO 22400-2:2014, p. 32
Actual Transport Run Time	ATR	Wannenwetsch, 2014, p. 696
Actual Transport Time	ATT	ISO 22400-2:2014, p. 9
Achieved Value Creation	AVC	Kletti and Schumacher, 2014, p. 133; Werner, 2013, p. 344
Allocation Efficiency	AE	ISO 22400-2:2014, p. 22; VDMA 66412-1:2009 p. 17
Allocation Ratio	AR	ISO 22400-2:2014, p. 20; VDMA 66412-1:2009 p. 15
Annual Holding Cost Factor	CHF	Erlach, 2010, p. 68
Availability	A	ISO 22400-2:2014, p. 26
Availability on the Planned Starting Date	APSD	VDI4400-2:2004, p. 18
Average Inventory	AVGI	ISO 22400-2:2014, p. 42
Calendar Time	T	Vorne Industries, 2017
Complaints Rate (Outgoing Items)	CRO	Ossola-Haring, 2006, p. 322

Complaints Rate (Purchase Items)	CRI	Ossola-Haring, 2006, p. 327
Compliance with Minimum Inventory of Source Material	MISM	VDI4400-2:2004, p. 20
Compliance with Warehouse Bandwidth	WBC	VDI4400-2:2004, p. 26
Consumption Deviation	CDE	Werner, 2013, p. 351
Costs of Internal Transport	CIT	VDI4400-2:2004, p. 38
Costs of Production Logistics	CPL	VDI4400-2:2004, p. 36
Costs of Production Planning	CPP	VDI4400-2:2004, p. 40
Current Inventory	INV	Own definition
Customer Demand	CD	Erlach, 2010, p. 48
Empty Run Share	ERS	Own definition
Empty Transport Run Time	ETR	Own definition
Energy Consumption per Unit	EC	Zhu et al., 2017, p. 972; ISO 22400-2:2014, p. 16
Energy Medium Quantity	EMQ	Zhu et al., 2017, p. 971
Equipment Load Ratio	ELR	Hofer, 2015, p. 242; ISO 22400-2:2014, p. 49
Every Part Every Interval	EPEI	Erlach, 2010, p. 72
Excess and Obsolete Ratio	EOR	Werner, 2013, p. 349
Failure Rate	λ	Birolini, 2017, p. 372
Good Product Quantity	NOK	ISO 22400-2:2014, p. 13
Hourly Rate for Machine Operation	HRM	Kletti and Schumacher, 2014, p. 133
Input Quantity	IQ	Zhu et al., 2017, p. 971
Inventory Days of Supply	SRA	Hofer, 2015, p. 250; Kletti and Schumacher, 2014, p. 134
Inventory Ratio	IR	Werner, 2013, p. 342
Inventory Turns	IT	ISO 22400-2:2014, p. 42
Lean Performance Index	LPI	Kletti and Schumacher, 2014, p. 72
Machine Efficiency	ME	Kletti and Schumacher, 2014, p. 133
Material and Equipment Costs	MEC	VDI4400-2:2004, p. 36
Maximum Inventory	MAXI	VDI4400-2:2004, p. 26
Mean Actual Busy Time	MABT	Own definition

Mean Circulating Inventory	MCI	VDI4400-2:2004, p. 43
Mean Costs of Production Logistics per Production Order	MCPL	VDI4400-2:2004, p. 36
Mean Costs of Production Planning per Production Order	MCPP	VDI4400-2:2004, p. 40
Mean Costs of Transport per Production Order	MCIT	VDI4400-2:2004, p. 38
Mean Execution Time Proportion	METP	VDI4400-2:2004, p. 22; Hofer, 2015, p. 248
Mean Incoming Inventory	MII	VDI4400-2:2004, p. 43
Mean Order Execution Time	MAOET	Own definition
Mean Outgoing Inventory	MOI	Hofer, 2015, p. 250
Mean Time Between Failures	MTBF	Feldmann et al., 2014, p. 861; ISO 22400-2:2014, p. 50
Mean Time to Failure	MTTF	Birolini, 2017, p. 372; ISO 22400-2:2014, p. 51
Mean Time to Repair	MTTR	Feldmann et al., 2014, p. 861; ISO 22400-2:2014, p. 52
Mean Transport Distance per Production Order	MTDO	Own definition
Minimum Inventory	MINI	VDI4400-2:2004, p. 20
Net Equipment Effectiveness Index	NEE	ISO 22400-2:2014, p. 25
No. of Days within the Inventory Limits	DWI	VDI4400-2:2004, p. 26
No. of Days Without Falling Below the Inventory	MID	VDI4400-2:2004, p. 20
Number of Automated Picks	NAP	Werner, 2013, p. 343
Number of Complaints (Outgoing Items)	NCO	Ossola-Haring, 2006, p.322
Number of Complaints (Purchase Items)	NCI	Ossola-Haring, 2006, p. 327
Number of Excess and Obsolete Items in Stock	EOI	Werner, 2013, p. 349
Number of Failures	NF	Feldmann et al., 2014, p. 861; ISO 22400-2:2014, p. 10
Number of Outgoing Items	NOI	Werner, 2013, p. 356
Number of Outgoing Items as Promised	NOIP	Werner, 2013, p. 356
Number of Picks	NOP	Werner, 2013, p. 343
Number of Procured Items	NIP	Werner, 2013, p. 342
Number of Procured Items as Promised	NPOP	Werner, 2013, p. 338
Number of Product Variants	VAR	Erlach, 2010, p. 72
Number of Production Orders	PO	VDI4400-2:2004, p. 18

Number of Startable Production Orders	SPO	VDI4400-2:2004, p. 18
Number of Successful Picks	NSP	Werner, 2013, p. 350
Operating Time Between Failures	TBF	ISO 22400-2:2014, p. 10
Optimal Lot Size (Economic Order Quantity)	EOQ	Erlach, 2010, p. 68
Order Fulfillment Cycle Time	OFT	SSC: SCOR, 2012, p. 42
Order Quantity	OQ	ISO 22400-2:2014, p. 7
Overall Equipment Effectiveness	OEE	Nakajima, 1988; ISO 22400-2:2014, p. 24
Overall Throughput Effectiveness	OTE	Muthiah and Huang, 2007, p. 4756; Muchiri and Pintelon, 2008, p. 3520
Performance Rate	P	ISO 22400-2:2014, p. 27; Muchiri and Pintelon, 2008, p. 3520
Personnel Costs	PEC	VDI4400-2:2004, p. 36
Picks Automation Rate	PAR	Werner, 2013, p. 343
Picks per Order	PPO	Werner, 2013, p. 343
Planned Busy Time	PBT	ISO 22400-2:2014, p. 7
Planned Down Time	PDT	Kang et al., 2015, p. 2768
Planned Energy Medium Quantity	PEMQ	Zhu et al., 2017, p. 971
Planned Input Quantity	PIQ	Werner, 2013, p. 351
Planned Operation Time	POT	ISO 22400-2:2014, p. 7
Planned Order Execution Time	POET	ISO 22400-2:2014, p. 7
Planned Production Cost per Unit	CP	Erlach, 2010, p. 68
Planned Raw Material Quantity	PRMQ	Zhu et al., 2017, p. 971
Planned Run Time per Unit	PRU	Kang et al., 2015, p. 2768
Planned Scrap Product Quantity	PNS	ISO 22400-2:2014, p. 32
Planned Setup Time	PSUT	Kang et al., 2015, p. 2768
Planned Value Creation	PVC	Kletti and Schumacher, 2014, p. 133
Point Availability	PA	Feldmann et al., 2014, p. 861
Process Wait Time	PWT	Kletti and Schumacher, 2014, p. 55
Produced Quantity	N	ISO 22400-2:2014, p. 13
Producion Equipment Efficiency	PEE	Raouf, 1994; Muchiri and Pintelon, 2008, p. 3522

Production Process Ratio	PPR	ISO 22400-2:2014, p. 31; Kletti and Schumacher, 2014, p. 129
Production Rate	PR	Zhu et al., 2017, p. 972; Hofer, 2015, p. 245
Productivity	FGR	Zhu et al., 2017, p. 972; ISO 22400-2:2014, p. 16
Quality Rate	Q	Nakajima, 1988; ISO 22400-2:2014, p. 28
Raw Material Quantity	RMQ	Zhu et al., 2017, p. 971
Recycled Material Quantity	REMQ	Werner, 2013, p. 344
Resource Amount	RES	Erlach, 2010, p. 72
Revenue	RE	Werner, 2013, p. 344
Rework Product Quantity	NR	ISO 22400-2:2014, p. 13
Rework Rate	RR	ISO 22400-2:2014, p. 35
Scrap Product Quantity	NS	ISO 22400-2:2014, p. 13
Scrap Rate	SR	ISO 22400-2:2014, p. 34
Service Level (Internal)	SLI	Werner, 2013, p. 350
Service Level (Outgoing Items)	SLO	Werner, 2013, p. 356; VDI4400-2:2004, p. 24
Service Level (Purchase Items)	SLP	Werner, 2013, p. 338
Set-Up Cost	CSU	Erlach, 2010, p. 68
Setup Rate	SUR	ISO 22400-2:2014, p. 29
Storage Area	SA	Wannenwetsch, 2014, p. 696
Storage Area Utilization	SAU	Wannenwetsch, 2014, p. 696
Storage Capacity	SC	Werner, 2013, p. 342
Storage Capacity Utilization	SCU	Werner, 2013, p. 342
Storage Volume	SV	Werner, 2013, p. 345
Storage Volume Utilization	SVU	Werner, 2013, p. 345
Takt Time	KT	Erlach, 2010, p. 48
Technical Efficiency	TAF	ISO 22400-2:2014, p. 30
Theoretical Throughput Rate	TFT	ISO 22400-2:2014, p. 21
Theoretical Transport Run Time	TTRT	Wannenwetsch, 2014, p. 696
Time to Failure	TTF	ISO 22400-2:2014, p. 10

Time to Repair	TTR	Feldmann et al., 2014, p. 861; ISO 22400-2:2014, p. 10
Total Effective Equipment Performance	TEEP	Vorne Industries, 2017
Transport Distance	TD	Own definition
Transport Utilization Rate	TUR	Wannenwetsch, 2014, p. 696
Turnover Rate of the Total Inventory	TRTI	VDI4400-2:2004, p. 43
Unplanned Down Time	UDT	ISO 22400-2:2014, p. 8
Used Storage Area	USA	Wannenwetsch, 2014, p. 696
Used Storage Volume	USV	Werner, 2013, p. 345
Utilization	U	Vorne Industries, 2017
Utilization Efficiency	UE	ISO 22400-2:2014, p. 23
Vertical Integration	VI	Werner, 2013, p. 344
Wait Time	WT	Kletti and Schumacher, 2014, p. 55
Weighting Factors	k1, k2, k3	Raouf, 1994; Muchiri and Pintelon, 2008, p. 3522
Work in Progress	WIP	Kropik, 2009, p. 172
Worker Efficiency	WE	ISO 22400-2:2014, p. 19

Table 15: List of Researched Metrics and Input Factors

10.2 Source Codes and File Structure

The dashboard application consists of multiple Python scripts, data sources and supporting files. The entire dashboard application is available for download in an online repository.[120]

Data Sources	
PF_DB_Data[…].db	Input database
sek.json	Live asset states
params.json	Global parameters
Data Handling and KPI Calculation	
util.py	Helper functions supporting data calculation and visualization
preprocessor_dash.py	Database update and KPI calculation routines

[120] https://drive.google.com/drive/folders/1pmyMLAwnqPNI_SF2ndWPDN_p6SaGh2M6?usp=sharing

Output Database	
PF_DB_OUT[…].db	Output database
Dashboard Server	
dash-live.py	Dashboard server
dash_elements.py	Layout and content definition of default dashboard elements
dash_elements_pro.py dash_elements_wor.py dash_elements_log.py dash_elements_mnt.py	Layout and content definition of dashboard elements adapted for specific stakeholder roles
dash_layouts.py	Definition of role-based element arrangements
style.css	Stylesheet for dashboard appearance
Provisioning of simulation parameters	
PF_DB_PARAMS.db	Source of calculation requests and location for results storage
plantsim_params_cli.py	Data handling and parameter calculation

Table 16: File Structure of the Dashboard Application

10.3 Database Update Procedure

To provide the dashboard visualizations with most recent information, outputs of the Digital Twin have to be periodically compared with the visualized dataset.

In the first step, connections to both databases are established, confirming the existence of tables in the output database and checking their contents. Empty tables can be created if necessary, and the compatibility of both datasets is investigated, making sure that two rows with matching identifiers actually contain the same data. In case incompatibility is detected, the old output database is archived, and replaced by an empty one.

The differences between input and output databases are then determined, and production orders followed by actions transferred towards the output database and complemented by results of row-based and order-specific KPI calculations.

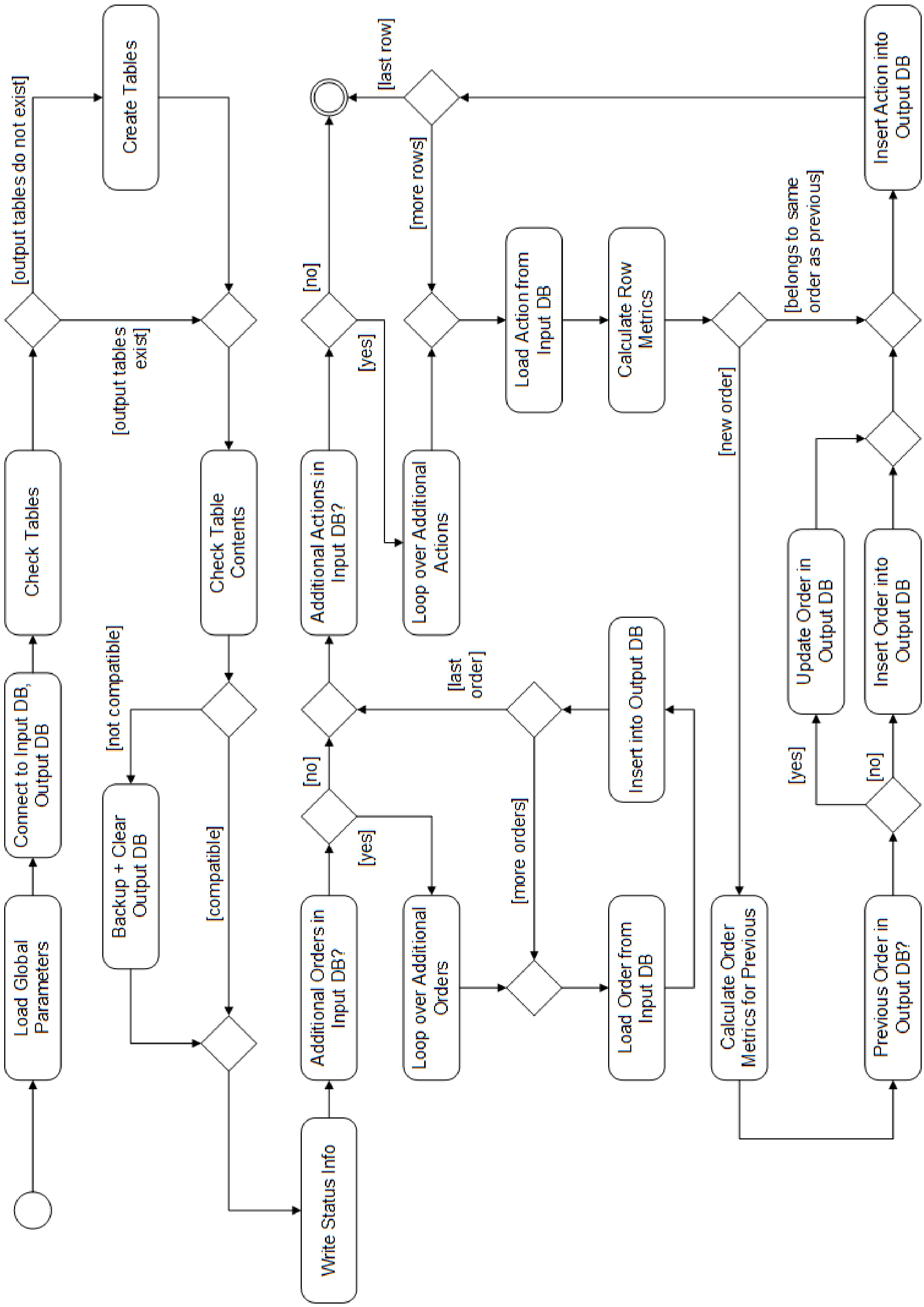


Figure 50: Database Update Flowchart

10.4 Sample KPI Calculation Function

The full code used to calculate and communicate values of AGV availability is presented below. The function takes several arguments and returns data in the requested extent and form – documented at the beginning of the code snippet.

The availability metric can be reported both as the latest recorded value, and also as a line chart visualizing past developments. An alert function is also included, sending out instant messages in case the current value falls below a specified limit. A detailed overview of the processes within the calculation function is visualized in Figure 51.

A significant part of the code deals with adaptations needed for visualization in the multi-day mode (Figure 34). The Dash framework is able to automatically create a time axis based on timestamp values from the database, which is the default format used for single-day operations. In case data spanning across multiple days has to be plotted, the uninteresting parts such as weekends and nights have to be cut out from the timeline. This has been realized by a conversion of timestamp values to integers, so that every second in the visualized interval is assigned a numerical value between 1 and n. Furthermore, a correction is applied in order to account for the time between the start of the interval and the first data point. This enables the Dash library to understand the position of values, but the format is not suited for human users. For this reason, the original date values are also part of the function output and are displayed as labels in the chart to preserve clarity of information.

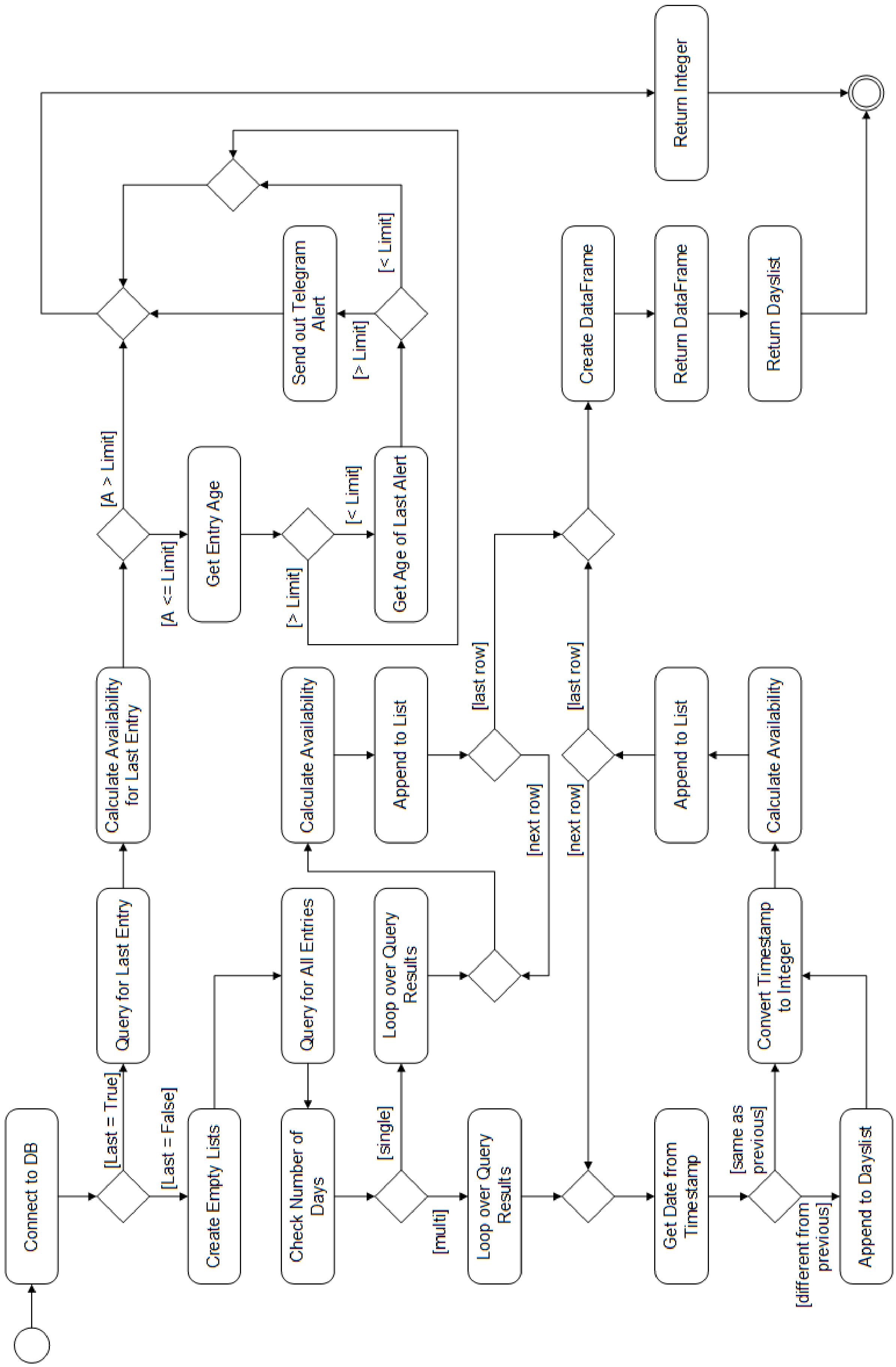


Figure 51: Flow Chart for the Availability Function

```
def availability(db_out_file, db_out_table, qfilter, agvid, otstart, otend,
                 last = True, alert_a = 0.0, alert_a_sensitivity = 1800):
    """
    A = SUM([Working] + [Idle]) / SUM([Working] + [Idle] + [Failure])

    Args:
    db_out_file  [String]  Path to Output database file
    db_out_table [String]  Table name in the Output database
    qfilter      [String]  Curr. filter setting in the dashboard session
    agvid        [Integer] Asset ID, or 0 for all assets
    otstart      [Integer] Shift start in the Pilot Factory, format H (e.g. 8)
    otend        [Integer] Shift end in the Pilot Factory, format H (e.g. 16)
    last         [Boolean] Only the most recent value returned if True
    alert_a      [Float]   Threshold for alert message

    Returns:
    last == True:
        availability            [Integer]
    last == False:
        availability(...)[0]    [Dataframe]
            Format: [[x-axis position], [value], [label for hover info]]
        availability(...)[1]    [Dataframe]
            Format: [[date[%b %d]], [x-axis correction[s]]]
    """
    #-- Create a connection to database
    db_out_conn = sqlite3.connect(db_out_file)
    db_out_curs = db_out_conn.cursor()

    #-- Filter for a particular asset ID when requested
    if agvid == 0:                          # All asset IDs
        qaddition = ''
    else:                                   # Filter for a particular asset only
        qaddition = ' AND AGV_ID = "{agvid}"'.format(agvid = agvid)

    #-- Calculate the most current value only --used for the bar chart
    if last == True:
        query_last = 'SELECT SUM(Duration), SUM(TTR) FROM {table} WHERE '.\
                     format(table = db_out_table)
        db_out_curs.execute(query_last + qfilter + qaddition + ';') # Full query
        row = db_out_curs.fetchone()                # Get data from the database
        ot, ft = row[0], row[1]

        query_age = 'SELECT TimeStamp_Start FROM {table} WHERE '.\
                    format(table = db_out_table)
        db_out_curs.execute(query_age + qfilter + qaddition +\
                            'ORDER BY ID DESC LIMIT 1;')
        timestamp_last = db_out_curs.fetchone()[0]
        db_out_conn.close()
        a = round((ot - ft) / float(ot) * 100, 1)   # Calculate the latest value

        #-- Send out a Telegram alert message when value sinks below limit
```

```
    if a < alert_a:
        timestamp_now = datetime.strftime(datetime.now(),
                                          "%Y/%m/%d %H:%M:%S")
        try:
            age_last = util.duration(timestamp_last, timestamp_now,
                                     otstart, otend)
        except:
            age_last = 999999
        if age_last < alert_a_sensitivity:  # If last data newer than xx sec
            util.telegram_message('ALERT!\nAGV Availability down to ' +\
                                  str(a) +'%', link = True)
    return a

#-- Calculate values for every dataset entry --used for the line chart
else:
    a_list, dayslist = [], []          # Prepare lists for values and dates
    query_all  = 'SELECT TimeStamp_Start, Duration, TTR FROM {table} WHERE '.\
                 format(table = db_out_table)
    db_out_curs.execute(query_all + qfilter + qaddition + ';') # Full query
    data = db_out_curs.fetchall()      # Get data from the database
    tss, duration, ttr = zip(*data)
    db_out_conn.close()

    firstday = tss[0][:10].replace('/', '')     # Date of the first row
    lastday  = tss[-1][:10].replace('/', '')    # Date of the last row
    start    = datetime.strptime(tss[0], "%Y/%m/%d %H:%M:%S")
    #-- Find out if we are about to plot data from multiple days
    if firstday != lastday:                    # More than one day
        multiday = True
        a_list.append([0, 100.0, tss[0]])      # Start with 100%, pos = 0
    else:
        multiday = False                       # One day only
        a_list.append([tss[0], 100.0, tss[0]]) # 100%, pos = DB timestamp

    currentday = firstday
    # Write down the day we are working on
    # (+ time correction if operation started later than at "otstart"-time)
    dayslist.append([
        datetime.strftime(datetime.strptime(currentday, "%Y%m%d"), "%b %d"),
        util.correction(tss[0], otstart, otend)])
    # Prepare variables for KPI calculation
    sum_duration, sum_ttr, delta = duration[0], ttr[0], 0
    # Loop over every row in the dataset...
    for i in range(1, len(tss)):
        # ...check if the row lies within operation time, skip if it doesn't
        if util.is_in_ot(tss[i], otstart, otend):
            if multiday == True:   # Multiple days - x-pos is an integer
                # Handle data for each additional day
                if tss[i][:10].replace('/', '') != currentday:
                    newday = tss[i][:10].replace('/', '')
                    #-- Time intervals outside of factory operating hours,
                    # and working days inside the interval with no entries
```

```
                curr = datetime.strptime(currentday, "%Y%m%d")
                new  = datetime.strptime(newday, "%Y%m%d")
                delta += ((new - curr).days - 1) * 24 * 3600 +\
                         ((24 - (int(otend) - int(otstart))) * 3600)

                dayslist.append([
                    datetime.strftime(
                        datetime.strptime(newday, "%Y%m%d"), "%b %d"),
                    delta])          # Date, correction
                currentday = newday  # Set the add. day as current
            #-- Convert timestamp from database to an integer
            timenew = datetime.strptime(tss[i], "%Y/%m/%d %H:%M:%S")
            timestamp = (timenew - start).total_seconds() - delta
        else:            # Plotting one day - no date conversion needed
            timestamp = tss[i]    # X-axis position = timestamp from DB
        #-- Calculate availability, make list entry
        if sum_duration == 0:     # A= 100% when duration of first row= 0
            a_list.append([timestamp, 100.0, tss[i]])
        else:                          # Calculate availability
            a_list.append([
                timestamp,
                round((sum_duration - sum_ttr) /
                    float(sum_duration) * 100, 1),
                tss[i]])
        sum_duration += duration[i]     # Increment the totals
        sum_ttr += ttr[i]
#-- Convert Python list to Pandas dataframe, return dataframe
df_a = pandas.DataFrame.from_records(a_list,
            columns = ['TimeStamp_Start', 'Availability', 'Label'])
df_a.set_index('TimeStamp_Start')
return df_a, dayslist
```

Code Snippet 3: Source Code of the Availability Function

Exemplary return values, single-day mode:

```
TimeStamp_Start, Availability, Label
2018/09/06 08:55:53, 100.0, 2018/09/06 08:55:53
2018/09/06 08:57:29, 100.0, 2018/09/06 08:57:29
2018/09/06 09:20:53, 7.1,   2018/09/06 09:20:53
2018/09/06 09:27:43, 26.9,  2018/09/06 09:27:43
2018/09/06 09:27:57, 27.4,  2018/09/06 09:27:57
2018/09/06 09:32:17, 36.0,  2018/09/06 09:32:17
2018/09/06 09:32:25, 36.3,  2018/09/06 09:32:25
2018/09/06 09:33:02, 37.3,  2018/09/06 09:33:02
2018/09/06 09:34:59, 35.5,  2018/09/06 09:34:59
...

[['Sep 06', 3342.0]]
```

Exemplary return values, multi-day mode:

```
TimeStamp_Start, Availability, Label
0.0,    100.0, 2018/09/06 08:55:42
11.0,   100.0, 2018/09/06 08:55:53
107.0,  100.0, 2018/09/06 08:57:29
1511.0, 7.1,   2018/09/06 09:20:53
1921.0, 26.9,  2018/09/06 09:27:43
1935.0, 27.4,  2018/09/06 09:27:57
2195.0, 36.0,  2018/09/06 09:32:17
2203.0, 36.3,  2018/09/06 09:32:25
2240.0, 37.3,  2018/09/06 09:33:02
2357.0, 35.5,  2018/09/06 09:34:59
...

[['Sep 06', 3342.0], ['Sep 13', 576000], ['Sep 20', 1152000], ['Sep 27', 1728000],
['Oct 04', 2304000], ['Oct 11', 2880000]]
```

10.5 Assignment of Metrics to Roles and Elements

An overview of KPI availability for individual stakeholder roles is presented in Table 17, including the visualization element(s) where the metrics have been implemented.

			Role			
Performance Indicator	**Acr-onym**	**Basis**	**Production Manager**	**Production Worker**	**Logistics Manager**	**Maintenance Manager**
Actual Busy Time, Assembly Station	ABT	Order	Order (Station)	Order (Station)		
Actual Idle Time, Actual Queuing Time, Actual Time to Repair, Actual Transport Time	AIT, AQT, TTR, ATT	Order	Order (AGV)		Order (AGV)	Order (AGV)
Actual Order Execution Time	AOET	Live, Order	Live, Order (Station, AGV)	Order (Station)	Order (AGV)	Order (AGV)
Actual Run Distance	ARD	Action			Current Values	Current Values
Availability	A	Action	Percentages, MTTR	Percentages, MTTR	Percentages, MTTR	Percentages, MTTR
Availability on Planned Starting Date	APSD	Order	Percentages (Logistics)		Percentages (Logistics)	
Compliance with Warehouse Bandwidth	WBC	Order			Percentages (Logistics)	
Current Inventory	INV	Live	Inventory	Live	Live, Inventory	
Energy Consumption per Unit	EC	Order			Order (AGV)	Order (AGV)
Mean Actual Busy Time	MABT	Order	Current Values, Order (Station)	Current Values, Order (Station)		
Mean Actual Transport Time	MATT	Order	Current Values		Current Values	
Mean Order Execution Time	MAOET	Order	Current Values, Order (AGV)		Order (AGV)	Order (AGV)
Mean Costs of Transport per Unit	MCIT	Order	Order (AGV)			
Mean Time Between Failures	MTBF	Failure Event			Current Values, MTBF	Current Values, MTBF
Mean Time to Repair	MTTR	Failure Event	MTTR	MTTR	Current Values, MTTR	Current Values, MTTR, Histogram
Number of Production Orders	PO	Order	Orders Table	Orders Table	Orders Table	
Production Process Ratio, AGV	PPR	Order	Percentages, Order (AGV)	Percentages	Order(AGV)	Order (AGV)
Production Process Ratio, Assembly St.	PPR	Order	Percentages, Order (Station)	Percentages, Order (Station)		

Service Level (Outgoing)	SLO	Order	Percentages (Logistics)		Percentages (Logistics)	
Storage Capacity Utilization	SCU	Order	Inventory		Inventory	
Technical Efficiency	TAF	Order			Percentages	Percentages
Transport Utilization Rate	TUR	Action			Percentages	

Table 17: Overview of Available KPI-Role-Element Combinations

11 Bibliography

Arnold, D., Isermann, H., Kuhn, A., Tempelmeier, H. and Furmans, K.(eds): Handbuch Logistik, Springer, Berlin, Heidelberg, 2008

Bauer, W., Schlund, S., Marrenbach, D. and Ganschar, O.: Industrie 4.0 – Volkswirtschaftliches Potenzial für Deutschland, BITKOM, Fraunhofer IAO, 2014

Bauernhansl, T., Ten Hompel, M. and Vogel-Heuser, B.(eds): Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien, Migration, 1st Edition, Springer Vieweg, Wiesbaden, 2014

Bledowski, K.: The Internet of Things: Industrie 4.0 vs. the Industrial Internet, MAPI Foundation, Arlington, 2015

Bracht, U., Geckler, D. and Wenzel, S.: Digitale Fabrik: Methoden und Praxisbeispiele, 2nd Edition, Springer Vieweg, Berlin, 2018

Chitkara, R., Ballhaus, W., Kliem, B., Berings, S. and Weiss, B.: Spotlight on Automotive, PwC, 2013

Drath, R. and Horch, A.: Industrie 4.0: Hit or Hype? [Industry Forum], *In: IEEE Industrial Electronics Magazine*, Volume 8, Issue 2, 2014, p. 56–58

EEA: The European Environment: State and Outlook 2015: Assessment of Global Megatrends, Publications Office of the EU, Luxembourg, 2015

Evans, P. C. and Annunziata, M.: Industrial Internet: Pushing the Boundaries of Minds and Machines, General Electric, 2012

Few, S.: Information Dashboard Design: The Effective Visual Communication of Data, 1st Edition, O'Reilly, Beijing; Cambride (MA), 2006

Few, S.: Dashboard Design for Real-Time Situation Awareness, Perceptual Edge, 2007

FFG: Ausschreibungsleitfaden für Pilotfabriken Industrie 4.0, Österreichische Forschungsförderungsgesellschaft, Wien, 2016

General Electric Company: GE Digital Twin: Analytic Engine for the Digital Power Plant, 2016

Grieves, M.: Digital Twin: Manufacturing Excellence through Virtual Factory Replication, 2014

Grieves, M. and Vickers, J.: Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems, *In: Kahlen, Franz-Josef, Flumerfelt, Shannon and Alves, Anabela (Eds)*: *Transdisciplinary Perspectives on Complex Systems*, Cham, 2017, p. 85–113

Haerder, T. and Reuter, A.: Principles of transaction-oriented database recovery, *In: ACM Computing Surveys*, Volume 15, Issue 4, 02.12.1983, p. 287–317

Hankel, M.: The Reference Architectural Model Industrie 4.0 (RAMI 4.0), ZVEI, Frankfurt/Main, 2015

Harrington, J. L.: Relational Database Design and Implementation: Clearly Explained, 3rd Edition, Morgan Kaufmann/Elsevier, Amsterdam; Boston, 2009

Hofer, P.: Produktionscontrolling - Kennzahlen in der Produktion*, In: Eisl, Christoph, Engelbrechtsmüller, Christian and Losbichler, Heimo (Eds)*: *Handbuch der betriebswirtschaftlichen Kennzahlen: Key Performance Indicators für die erfolgreiche Steuerung von Unternehmen*, Wien, 2015

ISO 22400-2: Key performance indicators (KPIs) for manufacturing operations management - Part 2: Definitions and descriptions, ISO/TC 184 Automation systems and integration, 2014

Juniper Networks: Introduction to Big Data: Infrastructure and Networking Considerations, 2012

Kagermann, H., Helbig, J., Hellinger, A. and Wahlster, W.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group, Plattform Industrie 4.0, Frankfurt/Main, 2013

Kautzsch, T., Kronenwett, D. and Thibault, G.: Megatrends and the Future of Industry*, In: Perspectives on Manufacturing Industries*, Volume 12, 2017

Kletti, J. and Schumacher, J.: Die perfekte Produktion, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014

Kritzinger, W., Karner, M., Traar, G., Henjes, J. and Sihn, W.: Digital Twin in manufacturing: A categorical literature review and classification*, In: IFAC-PapersOnLine*, Volume 51, Issue 11, 2018, p. 1016–1022

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T. and Hoffmann, M.: Industry 4.0*, In: Business & Information Systems Engineering*, Volume 6, Issue 4, 08.2014, p. 239–242

Lin, S.-W., Murphy, B., Clauer, E., Loewen, U., Neubert, R., Bachmann, G., Madhusudan, P. and Hankel, M.: Architecture Alignment and Interoperability, An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper, 2017

Mattern, F. and Floerkemeier, C.: From the Internet of Computers to the Internet of Things*, In: Sachs, Kai, Petrov, Ilia and Guerrero, Pablo (Eds)*: *From active data management to event-based systems and more*, Heidelberg, 2010, p. 242–259

Mell, P. and Grance, T.: The NIST Definition of Cloud Computing, National Institute of Standards and Technology, 2011

Mueller, M., Eddy, S., Geissbauer, R. and Jaruzelski, B.: Industrial Manufacturing Trends 2018–19, PwC, 2018

Neobotix: MP-400 Operating Manual, 2017

Neobotix: PlatformCtrl Programmer’s Guide, 2010

Osterhage, W. W.: ERP-Kompendium, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014

Parmenter, D.: Key performance indicators: developing, implementing, and using winning KPIs, Third edition, Wiley, Hoboken, New Jersey, 2015

Parrott, A. and Warshaw, L.: Industry 4.0 and the digital twin: Manufacturing meets its match, Deloitte University Press, 2017

Rosen, R., von Wichert, G., Lo, G. and Bettenhausen, K. D.: About The Importance of Autonomy and Digital Twins for the Future of Manufacturing, *In: IFAC-PapersOnLine*, Volume 48, Issue 3, 2015, p. 567–572

Russo, M., Helm, R. and McGabe, G.: The Industrial Internet: Six Critical Questions for Equipment Suppliers, The Boston Consulting Group, 2014

Schinagl, F.: Konzeption und Umsetzung des Auftragsabwicklungsprozesses der TU Wien Pilotfabrik Industrie 4.0, TU Wien, Wien, 2017

Schumacher, A.: Development of a Maturity Model for Assessing the Industry 4.0 Maturity of Industrial Enterprises, TU Wien, Wien, 2015

Schweichhart, K.: Reference Architectural Model Industrie 4.0 (RAMI 4.0), Plattform Industrie 4.0, 2016

Siemens PLM Software: MindSphere: The cloud-based, open IoT operating system for digital transformation, 2017

Smogeli, Ø.: Digital Twins at Work in Maritime and Energy, DNV GL Digital Solutions, Høvik, 2017

Sommarberg, M.: Digitalization as a Paradigm Changer in Machine-Building Industry, Tampere University of Technology, Tampere, 2016

Supply Chain Council: SCOR: Supply Chain Operations Reference Model, v11.0, The Supply Chain Council, Inc., 2012

Tokola, H., Gröger, C., Järvenpää, E. and Niemi, E.: Designing Manufacturing Dashboards on the Basis of a Key Performance Indicator Survey, *In: Procedia CIRP*, Volume 57, 2016, p. 619–624

VDI 4400-2: Logistikkennzahlen für die Produktion, VDI-Gesellschaft Produktion und Logistik, 2004

VDI 4499-1: Digitale Fabrik Grundlagen, Verein Deutscher Ingenieure, Düsseldorf, 2008

VDI 5600-1: Manufacturing Execution Systems, Verein Deutscher Ingenieure, Düsseldorf, 2016

VDMA 66412-1: Manufacturing Execution Systems (MES) Kennzahlen, Verband Deutscher Maschinen- und Anlagenbau e.V., 2009

Wannenwetsch, H.: Integrierte Materialwirtschaft, Logistik und Beschaffung, 5., neu bearb. Aufl, Springer Vieweg, Berlin, 2014

Xu, L. D., He, W. and Li, S.: Internet of Things in Industries: A Survey*, In: IEEE Transactions on Industrial Informatics*, Volume 10, Issue 4, 11.2014, p. 2233–2243

Zhu, L., Johnsson, C., Mejvik, J., Varisco, M. and Schiraldi, M.: Key performance indicators for manufacturing operations management in the process industry*, In:* , 2017, p. 969–973

Online Sources

Edlich, S.: NoSQL Database, http://nosql-database.org/, accessed 07.09.2018

Fekár, M.: KPI Dashboard Files, https://drive.google.com/drive/folders/1pmyMLAwnqPNI_SF2ndWPDN_p6SaGh2M6?usp=sharing, accessed 20.02.2019

GE Digital: What is Edge Computing, https://www.ge.com/digital/blog/what-edge-computing, accessed 09.09.2018

Intel: Moore’s Law and Intel Innovation, https://www.intel.com/content/www/us/en/history/museum-gordon-moore-law.html, accessed 30.08.2018

Jager, T.: How to Optimize Charts for Color Blind Readers Using Color Blind Friendly Palettes, https://venngage.com/blog/color-blind-friendly-palette/, accessed 07.09.2018

Object Management Group: Unified Modeling Language, http://www.uml.org/, accessed 03.09.2018

Plotly: Dash, https://plot.ly/products/dash/, accessed 03.10.2018

Postscapes: IoT Standards and Protocols, https://www.postscapes.com/internet-of-things-protocols/, accessed 07.12.2018

Postscapes and Harbor Research: What Is The ‘Internet of Things’?, https://www.postscapes.com/what-exactly-is-the-internet-of-things-infographic/, accessed 22.11.2018

python-telegram-bot: python-telegram-bot, https://python-telegram-bot.org/, accessed 13.12.2018

Real Time Automation: OPC UA Overview, https://www.rtaautomation.com/technologies/opcua/, accessed 29.12.2018

SQLite: SQLite, https://www.sqlite.org/docs.html, accessed 19.11.2018

Stadt Wien: Gesetzliche Feiertage, https://www.wien.gv.at/ikt/egov/gesetzliche-feiertage.html, accessed 11.07.2018

Tableau: Tableau Desktop, https://www.tableau.com/products/desktop, accessed 03.10.2018

Telegram: Bots: An introduction for developers, https://core.telegram.org/bots, accessed 13.12.2018

TU Wien: Pilotfabrik Industrie 4.0: Anwendungsfelder, http://pilotfabrik.tuwien.ac.at/inhalte/anwendungsfelder/, accessed 12.09.2018

yUML, https://yuml.me/, accessed 12.04.2018

Additional Sources of KPI Research

Birolini, A.: Reliability Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2017

Erlach, K.: Wertstromdesign, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010

Feldmann, K., Schöppner, V. and Spur, G.(eds): Handbuch Fügen, Handhaben, Montieren, 2., vollständig neu bearbeitete Auflage, Hanser, München, 2014

Kang, N., Zhao, C., Li, J. and Horst, J. A.: Analysis of key operation performance data in manufacturing systems*, In:* , 2015, p. 2767–2770

Kropik, M.: Produktionsleitsysteme in der Automobilfertigung, Springer, Berlin, 2009

Muchiri, P. and Pintelon, L.: Performance measurement using overall equipment effectiveness (OEE): literature review and practical application discussion*, In: International Journal of Production Research*, Volume 46, Issue 13, 07.2008, p. 3517–3535

Muthiah, K. M. N. and Huang, S. H.: Overall throughput effectiveness (OTE) metric for factory-level performance monitoring and bottleneck detection*, In: International Journal of Production Research*, Volume 45, Issue 20, 15.10.2007, p. 4753–4769

Nakajima, S.: Introduction to TPM: total productive maintenance, Productivity Press, Cambridge, Mass, 1988

Ossola-Haring, C.(ed): Handbuch Kennzahlen zur Unternehmensführung: Kennzahlen richtig verstehen, verknüpfen und interpretieren, 3., aktualisierte und erw. Aufl, mi-Fachverl., REDLINE, Landsberg am Lech, 2006

Raouf, A.: Improving Capital Productivity through Maintenance*, In: International Journal of Operations & Production Management*, Volume 14, Issue 7, 07.1994, p. 44–52

Vorne Industries: Calculate TEEP – Measure Utilization and Capacity (OEE.com), 2017

Werner, H.: Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling, 5., überarb. und erw. Aufl, Springer-Gabler, Wiesbaden, 2013

12 List of Figures

13 List of Tables

14 List of Code Snippets

15 List of Abbreviations

3D	Three-dimensional
ACID	Atomicity, Consistency, Isolation, Durability
AGV(S)	Autonomous Ground Vehicle (System)
AI	Artificial Intelligence
B2B	Business-to-Business
BMVIT	Austrian Ministry for Transport, Innovation and Technology
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CPS	Cyber-Physical System
DBMS	Database Management System
€	Euro
EEA	European Environment Agency
e.g.	for example
ERM	Entity-Relationship Model
ERP	Enterprise Resource Planning
etc.	et cetera
EU	European Union
FDM	Fused Deposition Modelling
FFG	Austrian Research Promotion Agency
GE	General Electric
h	Hour
I(C)T	Information (and Communications) Technology
ID	Identifier
IIC	Industrial Internet Consortium
IMW	Institute of Management Science (TU Wien)
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
km	Kilometre
KPI	Key Performance Indicator
min.	minimum
max.	maximum
NASA	National Aeronautics and Space Administration (USA)
NFC	Near-Field Communication
OEM	Original Equipment Manufacturer

OPC	Object Linking and Embedding for Process Control
PC	Personal Computer
PLM	Product Lifecycle Management
RFID	Radio-Frequency Identification
s	second
SCOR	Supply Chain Operations Reference (Model)
SME	Small and Medium-sized Enterprises
SQL	Structured Query Language
TCP	Transmission Control Protocol
TU	Technical University
UML	Unified Modeling Language
URL	Uniform Resource Locator
US	United States (of America)
WLAN	Wireless Local Area Network