Digital Platforms for Industrial Metaverse Applications: A Framework to Identify Data Quality Insufficiencies

Master thesis submitted to Delft University of Technology

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in **Management of Technology**

Faculty of Technology, Policy and Management

by

Niklas Biermann

Student number: 5611512

To be defended in public on 07/07/2023

Graduation committee

Chairperson	: Dr. A.Y. Ding, ICT (Engineering Systems and Services)
First Supervisor	: Dr. A.Y. Ding, ICT (Engineering Systems and Services)
Second Supervisor	: Dr.ir. G. Korevaar, E & I (Engineering Systems and Services)

Acknowledgement

As this thesis concludes my master studies at the *Technische Universiteit Delft*, I want to take the time to acknowledge the contribution of everyone who provided support to me during the course of this final research project. I hence want to thank my second supervisor Dr. Gijsbert Korevaar as well as my first supervisor and chairman Dr. Aaron Ding for supervising and guiding me. I especially want to highlight the help of Aaron Ding who was the first of many contacted potential supervisors that saw the potential of this promising research field. His enthusiasm and continuous support in our weekly meetings were not only motivating, but also equipped me with the necessary guidance to look for a "hammer to existing nails" and not the other way around.

I also want to thank all the friends I made in Delft who made my studies in the Netherlands so special as well as my girlfriend Selina, my siblings Lena and Fabian and my parents Kristine and Jens.

Executive Summary

The metaverse is one of the most disruptive technologies to evolve from the digital transformation. While the potential use cases of creating an immersive virtual world are numerous, the vision of an industrial metaverse is only recently emerging as a concept from the technology. In the automotive sector, manufacturers are starting to use simulation, digital twin technology and *Building Information Modelling* (BIM) to build virtual factories in an industrial metaverse. The benefits of this innovation are believed to significantly boost production flexibility and efficiency, which is why manufacturers set up data-driven digital platforms to enable an industrial metaverse that interconnects multiple actors. However, technical barriers still hamper the implementation of such platforms whose dependence on flawless data grows with the number of use cases for an industrial metaverse. Accordingly, quality insufficiencies of spatial data and the absence of automatic quality assessments to identify these insufficiencies are one of the most decisive barriers to a widespread adoption of industrial metaverse applications. This thesis examines this problem and investigates how data quality insufficiencies in an industrial metaverse environment can be identified and overcome at the example of an automotive manufacturer that uses the *Nvidia Omniverse* digital platform to create virtual factory models. A design science approach is pursued to create an extension to the *Omniverse* software that identifies the most critical data quality insufficiencies, derives *key performance indicators* (KPIs) and proposes preventive measures to induce a sustained data quality improvement. Thereby, this thesis lays the groundwork for future research emerging around the concept of an industrial metaverse and the remaining obstacles of digital platforms to enable its applications. The pursued DSRM approach to overcome such barriers is capable to serve as guidance for future research projects that pave the way for a gradual enablement of further industrial metaverse use cases in other industries.

I Table of Contents

Acknowledgement .... **i**

Executive Summary .... **ii**

I Table of Contents .... **iii**

II Abbreviations .... **vi**

III Table of Figures .... **vii**

IV List of Tables .... **ix**

V Source Code Repository .... **x**

1 Introduction **1**

1.1 Motivation and Research Problem 2

1.2 Research Objective 3

1.3 Outline 4

2 Research Design **5**

2.1 Design Science Research Methodology 5

2.2 Methodological Approaches 6

2.2.1 Desk Research 7

2.2.2 Comparative Software Analysis 9

2.2.3 Expert Interviews 9

2.2.4 Failure Modes and Effects Analysis 10

2.2.5 Empirical Software Engineering 10

2.3 Research Flow 11

3 Literature Review **12**

3.1 Digital Manufacturing Platforms 13

3.2 Industrial Metaverse Applications 15

3.2.1 BIM 16

3.2.2 Digital Twin Technology 16

3.2.3 Industrial Process Simulations 17

3.2.4 Virtual Factory 17

3.3 Data Quality 18

3.3.1 The Concept of Data Quality 19

3.3.2 Spatial Data Quality 20

4 Identification and Prioritisation of Data Quality Insufficiencies 21

4.1 Comparative Software Analysis... 21

4.1.1 Enabling Software Architectures.. 21

4.1.2 The *Nvidia Omniverse*.. 24

4.1.3 *Nvidia Omniverse* Terminology 26

4.2 Stakeholder Analysis 27

4.3 Problem Definition.. 29

4.4 Technical Expert Interviews 31

4.5 Software Failure Modes and Effects Analysis 34

4.6 Artefact Objectives 35

5 Design of a Data Quality Gate Software Extension 36

5.1 Identification of Data Quality Insufficiencies... 36

5.1.1 Identification of Positional Deviations of Selected Prims.......................... 37

5.1.2 Identification of Scaling Deviations... 40

5.1.3 Identification of Two-Dimensional Data 43

5.1.4 Identification of Performance-Inhibiting Data 45

5.1.5 Identification of Empty Files.. 47

5.1.6 Identification of Naming Convention Violations....................................... 48

5.2 User Interface.. 50

5.3 Data Analysis.. 52

5.3.1 KPI Definition... 52

5.3.2 Definition of Minimum Data Quality Standards.. 53

6 Prototype Usability Test and Evaluation.. 55

6.1 Software Extension Test 55

6.1.1 Reiteration Cycle 1: Revision of Positional Deviation Identification 56

6.1.2 Reiteration Cycle 2: Revision of Scaling Deviation Identification 59

6.2 Preventive Measures 60

6.2.1 Mitigation of Insufficiencies Caused by Unfamiliarity With the Workflow.............. 61

6.2.2 Mitigation of Insufficiencies Caused by Missing Source Systems Standards............ 62

6.3 Pre-Post-Analysis.. 62

6.4 Communication.. 63

7 Discussion 64

7.1 Limitations ... 64
7.2 Expert Validation ... 66
7.3 Reflections ... 67
7.4 Areas of Future Research ... 67
8 Conclusion ... 69
8.1 Research Outcomes ... 69
8.2 Relevance to the Study Programme ... 71
8.3 Scientific Contribution ... 71
8.4 Societal Contribution ... 72
8.5 Guideline for Framework Application in a Data Management Context ... 73
VI References ... xii
VII Appendix ... xix
A.1 PRISMA Step I – Identification ... xix
A.2 PRISMA Step II - Screening ... xix
A.3 PRISMA Step III – Decision ... xx
A.4 PRISMA Step III – Decision (Forwards/Backwards) ... xxi
A.5 Interview Questionnaire ... xxii
A.6 Interview Summaries ... xxiv
A.7 FMEA Evaluation Table ... xxviii
A.8 Extension Repository READ.ME-File ... xxix
A.9 Test Results ... xxx
A.10 Exhaustive KPI Dashboard ... xxxi
A.11 Simplified Extension UML Diagram ... xxxii
A.12 Duration Comparison of Manual and Automatic Quality Checks ... xxxiii

II Abbreviations

Abbreviation	Description
ACPS	Anthropocentric Cyber-Physical System
API	Application Programming Interface
BIM	Building Information Modelling
CAD	Computer-Aided Design
CPS	Cyber-Physical System
CSV	Comma-Separated Values
DSRM	Design Science Research Method
FMEA	Failure Modes and Effects Analysis
FPS	Frames Per Second
IFC	Industry Foundation Classes
IIRA	Industrial Internet Reference Architecture
IoT	Internet of Things
I4.0	Industry 4.0
KPI	Key Performance Indicator
MOT	Management of Technology
OEE	Overall Equipment Efficiency
OEM	Original Equipment Manufacturer
Prim	Primitive
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
Q1,2,3,4,5	Question 1,2,3,4,5
RAMI	Reference Architectural Model Industry 4.0
RPN	Risk Priority Number
RQ	Research Question
UML	Unified Modelling Language
URL	Uniform Resource Locator
USD	Universal Scene Description
VR	Virtual Reality
3D	Three-dimensional
2D	Two-dimensional

III Table of Figures

Figure 2.1: DSRM Process Model modified from *A design science research methodology for information systems*, by K. Peffers, 2007, p. 54 5

Figure 2.2: PRISMA 2020 flow diagram modified from *The PRISMA 2020 statement: an updated guideline for reporting*, by M. Page et al., 2020, p. 5 8

Figure 2.3: Steps of the FMEA Process 10

Figure 2.4: Thesis Research Flow Diagram 11

Figure 3.1: PRISMA Step IV – Inclusion: Relevant Concepts of the Literature Review 12

Figure 3.2: The Relationship between Industrial Metaverse, Digital Twin and Simulation, modified from *Industrial Metaverse: Connotation, Features, Technologies, Applications and Challenges*, by Z. Zheng et al., 2022, p. 241 15

Figure 3.3: A Digital-Twin-Based Virtual Factory Concept, modified from *Virtual Factory: Digital Twin Based Integrated Factory Simulations*, by E. Yildiz et al., 2020, p. 218 18

Figure 3.4: A Conceptual Data Framework of Data Quality, modified from *Beyond Accuracy: What Data Quality Means to Data Consumers*, by R. Wang & D. Strong, 1996, p. 20 19

Figure 4.1: The DSRM Approach - Chapter 4 21

Figure 4.2: The IFC Data Schema, modified from *The Use of Templates and The Handle for Large-Scale Provision of Security and IoT in the Built Environment*, by P. Kirstein and A. Ruiz-Zafra, 2018, p. 3 22

Figure 4.3: CPS Architecture for Smart Manufacturing, modified from *Evolution of 3C Cyber-Physical Systems Architecture for Industry 4.0*, by A. Ahmadi et al., 2018, p. 452 23

Figure 4.4: Software Architecture of a Metaverse-Enabled Digital Twin Application, modified from *Building Digital Twins of Cyber Physical Systems with Metaverse for Industry 5.0 and Beyond*, by S. Jagatheesaperumal and M. Rahouti, 2022, p. 37 24

Figure 4.5: The *Nvidia Omniverse* Software Architecture at the Example of an Automotive Manufacturer 25

Figure 4.6: Exemplary Excerpt of a Prim Tree in a *Nvidia Omniverse* USD Stage 27

Figure 4.7: Company-Internal Power-Interest-Grid of the *Nvidia Omniverse* Stakeholders 29

Figure 4.8: Action Priority Matrix ... 34

Figure 4.9: The Software Extension as a Virtual Factory Quality Gate ... 35

Figure 5.1: The DSRM Approach - Chapter 5 ... 36

Figure 5.2: Positional Deviation of a Forklift in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022 ... 38

Figure 5.3: Scaling Deviation of a Forklift in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022 ... 41

Figure 5.4: 2D Forklift in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022 ... 44

Figure 5.5: The Data Quality Gate User Interface ... 51

Figure 5.6: KPI Dashboard - Part I ... 53

Figure 5.7: KPI Dashboard – Part II ... 53

Figure 5.8: Approximated Relationship between FPS and Total Mesh Number ... 54

Figure 6.1: The DSRM Approach - Chapter 6 ... 55

Figure 6.2: Result of a *Position-Check-Light* in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022 ... 58

Figure 6.3: KPI Dashboard for Positional Deviations ... 59

Figure 8.1: SCOPUS results for the query *ALL(Industrial Metaverse)* ... 72

IV List of Tables

Table 3.1: Spatial Data Quality Criteria 20

Table 4.1: Company-Internal Stakeholders of the *Nvidia Omniverse* 28

Table 4.2: The Technical Expert Interview Process 32

Table 4.3: Categorised List of Data Quality Insufficiencies within the *Nvidia Omniverse* 33

Table 5.1: Software and Programming Language of Use 37

Table 5.2: Sanity Check of Approaches to Identify Positional Deviations 39

Table 5.3: Sanity Check of Approaches to Identify Scaling Deviations 41

Table 5.4: Sanity Check of Approaches to Identify 2D Data 44

Table 5.5: Sanity Check of Approaches to Identify Performance-Inhibiting Data 46

Table 5.6: Sanity Check of Approaches to Identify Empty Files 48

Table 5.7: Sanity Check of Approaches to Identify Naming Convention Violations 49

Table 7.1: Determination of Extension Accuracy 65

Table 8.1: Research Outputs 70

V Source Code Repository

Source Code 5.1: Identification of Roof Prim ... 40

Source Code 5.2: Identification of Z-Outliers Outside Factory Building Thresholds ... 40

Source Code 5.3: Computation of Bounding Box Volume ... 42

Source Code 5.4: Computation of Mean Bounding Box Volume Confidence Interval ... 43

Source Code 5.5: Identification of Scaling Deviations ... 43

Source Code 5.6: Computation of Bounding Box Extents ... 45

Source Code 5.7: Computation of Vertice & Mesh Count ... 47

Source Code 5.8: Empty File Identification ... 48

Source Code 5.9: Check for Violation of Naming Convention ... 50

Source Code 6.1: Revised Extension: Z-Coordinate Confidence Interval ... 56

Source Code 6.2: Revised Extension: Identification of Positional Deviations ... 57

Source Code 6.3: Revised Extension: Volume of Extended Bounding Box ... 60

1 Introduction

Industry 4.0 (I4.0) is expected to increase the efficiency of manufacturing processes as the concept incorporates for the introduction of *Information and Communication Technologies* (ICT) as well as advanced data analytics into manufacturing (Alcácer & Cruz-Machado, 2019). Consequently, a shift towards knowledge-enabled, data-driven smart manufacturing is induced. To enable the development of smart, fully integrated manufacturing, cyber-physical integration to offer real-time data gathering, transparency and analysis across all parts of a manufacturing process is required. *Cyber-Physical Systems* (CPS) and digital twins are the preferred methods of such integration mentioned in academia and industry, of which smart manufacturing can be considered to be a specialisation (Tao et al., 2019).

Especially in the automotive industry, the market environment is characterised by a high volatility causing manufacturers to explore new methods to maximise production flexibility and efficiency (Llopis-Albert et al., 2021). In order to accommodate for the resulting dynamic factory requirements, *Original Equipment Manufacturers* (OEMs) from the automotive industry aim at the creation of a smart manufacturing environment that allows for collaboration in a fully integrated setting. To pave the way for a smart factory, the preliminary step in achieving I4.0 is to illustrate the physical factory building and all included assets through a network of digital entities. The term *asset* is used to describe the components of a factory, namely products, machines and human workers. This includes digital data visualisation methods to represent factory assets to aid in strategic and informed decision-making (Chandra Sekaran et al., 2021). For that purpose, the advantages of introducing an industrial metaverse are evaluated by OEMs. By applying the metaverse in a manufacturing context, an interactive and collaborative space that allows for smart management of industrial products throughout their life cycle can be generated (Zheng et al., 2022). A possible industrial metaverse use case is the development of a virtual factory model that holds the potential to realistically render a whole production site infrastructure. The resulting *three-dimensional* (3D) environment is believed to allow for the analysis of production systems that are too complex to adequately be described by the means of mathematical or analytical models (Alpala et al., 2022). Thereby, utilisation statistics can be calculated, manufacturing schedules optimised and production bottlenecks identified. Moreover, activities like factory layout planning are believed to become more time- and cost-efficient due to scalability advantages (Kibira & Shao, 2017). In addition, the integration of digital human models allows for the coordination of human-machine-interaction, monitoring work performance and enhancing ergonomics management (Greco et al., 2020).

A preeminent driver of the I4.0 emergence is the *Digital Economy* which is characterised by fundamental changes in the way information is shared and communicated. Digital platforms are part of this newly arising *Digital Economy* as they create opportunities for firms to capture new innovative sources of value across a multitude of industries by facilitating the incorporation of new technologies and applications. Digital platforms furthermore support the formation of supply chain collaborations and hold the potential to form supply networks that can adapt to dynamic production requirements. (Gerrikagoitia et al., 2019; Z. Liu et al., 2022; Reuver et al., 2018) Thus, considering the emergence of the smart manufacturing paradigm, automotive OEMs make use of digital platforms as an enabler of use cases to apply the industrial metaverse and realise a virtual factory. A common software platform that allows for an efficient knowledge exchange across all phases of a product and factory life cycle between a multitude of intra- and interorganisational actors is crucial to ensure interoperability between all virtual factory tools (Tolio et al., 2013).

However, as industrial metaverse platforms are further explored in automotive manufacturing, data sizes as well as the number of actors involved within one ecosystem are growing to unprecedented dimensions. Consequently, ecosystem data quality moves into focus as a critical issue when sharing large amounts of spatial data within an ecosystem such as a digital platform to accurately visualise entire factories. Automotive OEMs with the intention of implementing data-driven digital platforms to create virtual factory representations face data quality challenges to ensure a smooth platform functionality.

1.1 Motivation and Research Problem

Automotive OEMs aim to address dynamic customer and market demands through the use of digital factory tools. The industrial metaverse and its applications in manufacturing to enable a virtual factory model are particularly moving into focus as they prove useful when planning, building and managing complex manufacturing systems and factories. As these activities require heavy investment, design verification is essential to avoid costly mistakes and ensure the achievement of the manufacturing system's intended benefits (H.-T. Park et al., 2010). For industrial metaverse applications, platform solutions prove useful in decentrally integrating the required industrial digital assets and enabling network interactions among actors (Hagiu & Wright, 2015; Julian Marius Müller, 2019; Oliveira & Cortimiglia, 2017). The full potential of a virtual factory unfolds only through the involvement of multiple actors to simultaneously plan and simulate factories. Digital platforms are therefore utilised by automotive OEMs to integrate the necessary digital tools and facilitate collaborative work.

The capabilities of industrial metaverse applications and digital platforms are improving with the growth of computing power capacities (Alt, 2021). On the other hand, virtual factory models to offer maximum physical validity while being accessible and editable by numerous actors hold data volumes that have been previously unmatched. Given the high production pressure in the automotive industry, the seamless and rapid rendering of the virtual factory becomes a critical necessity. Factory planning mistakes are costly which is why data needs to be reliable for it to be trustworthy. This, however, usually requires a significant investment of time and money as increases in factory complexity and spatial data volume make data quality insufficiencies unavoidable (Johnson & Dasu, 2003). An automatic process to identify insufficiencies of spatial data in virtual factory models on digital platforms does not yet exist which is why enterprises that are implementing virtual factory platforms are manually reviewing models for mistakes. Especially large automotive OEMs face the dilemma that such manual data quality screenings are connected to immense costs. Furthermore, the margin of human error associated with such a procedure also grows with factory model complexity and size. The results of interviews with the representatives of an automotive OEM that is currently introducing the *Nvidia Omniverse* as digital platform to create virtual factory models reveal that insufficient data quality has a negative influence on the functionality of the platform, its collaborative value and the general usability of all involved technologies. However, checks to evaluate and quantify this data quality within virtual factory models could not yet be automated by the manufacturer. Currently, manual checks are the only option to identify insufficiencies. The costs as well as non-eliminable margin of error associated with such human checks outweighs the potential benefits of this innovative technology as it hampers the company-internal acceptance and thereby adoption of the *Nvidia Omniverse*. This is the core research problem studied at the example of the aforementioned automotive OEM in this thesis.

1.2 Research Objective

In order to tackle the problem described in section 1.1 and fill the corresponding gap in research, existing data quality barriers of digital platforms for industrial metaverse applications need to become automatically identifiable. Accordingly, this thesis aims at answering the following research question:

Research Question (RQ): *How can quality insufficiencies within spatial data digital platforms automatically be identified to improve industrial metaverse applications in the automotive industry?*

The initial step towards answering the research question is to establish a general understanding of the possible metaverse applications, respectively use cases that already exist or are envisioned in manufacturing. Thereby, the industrial metaverse term and the concept of a virtual factory become more tangible. Moreover, this knowledge is valuable to recognise what technical innovations are hampered by data quality insufficiencies of enabling digital platforms. The first sub-question can be defined as:

Question 1 (Q1)*: What are current and future industrial metaverse applications of digital platforms?*

A key contributing factor to the growing relevance of data quality concerns is the fact that digital platforms employed by larger enterprises such as OEMs are connecting an unprecedented number of actors. As all these actors are involved with the platform and thereby affected by the problem, the key stakeholders and the problem owner need to be identified and distinguished.

Question 2 (Q2)*: What stakeholders of an OEM are affected by data quality insufficiencies of the platform?*

In order to tackle existing spatial data quality barriers, these barriers need to be exhaustively collected and understood which is why the third sub-question is stated as follows:

Question 3 (Q3)*: What spatial data quality barriers exist in an industrial metaverse platform?*

After identifying existing spatial data quality insufficiencies, these insufficiencies need to be adequately prioritised, so that a solution can be specifically designed to target those that are of highest urgency.

Question 4 (Q4)*: How can the identified data quality barriers be prioritised?*

Finally, a software extension to the *Nvidia Omniverse* platform needs to be created that builds on and comprises the results of the previous sub-questions:

Question 5 (Q5): *What software extension fits the stakeholder requirements of Q2 to tackle the insufficiencies prioritised in Q4?*

In the subsequent section, the thesis structure to achieve these objectives and answer the research question as well as its related sub-questions is outlined.

1.3 Outline

This thesis is divided into eight chapters that are organised as follows: Chapter 2 details the research design chosen to tackle the identified research problem. Here, the methodological approaches pursued as well as the corresponding research flow are detailed. Chapter 3 contains a PRISMA literature review of concepts that are relevant to the thesis scope. This includes an introduction to digital manufacturing platforms (chapter 3.1), industrial metaverse applications (chapter 3.2) and data quality (chapter 3.3). Chapter 4 is introduced by an analysis of software comparable to and relevant for an understanding of the *Nvidia Omniverse*. Then, the key stakeholders involved in the introduction of the software at the observed automotive manufacturer are studied. All data quality insufficiencies of knowledge to technical experts among these stakeholders are documented and prioritised through interviews. Finally, the problem is narrowed down and objectives of the software to be designed are defined. Chapter 5 documents the development process of the software whose functionality is demonstrated, evaluated and communicated in chapter 6. Finally, limitations, reflections as well as areas of future research are discussed in chapter 7 before providing a conclusion on the research question and a detailed exposition of the scientific and societal contribution of this work in chapter 8.

2 Research Design

In the following, the research design pursued as well as the methodological approaches entailed within this design are detailed.

2.1 Design Science Research Methodology

This research aims at the creation of a solution to overcome data quality insufficiencies of virtual factory models visualised on the *Nvidia Omniverse* platform. This platform is used by the observed automotive OEM to improve internal processes and thus its status quo. Accordingly, methods of design science research are serving as guidance to the methodology pursued in this research project. The *Design Science Research Method* (DSRM) introduced by Peffers et al. (2007) appears particularly suitable as it incorporates practices specifically applicable to information systems such as digital platforms. It examines the creation of artefacts or innovations required to investigate, design, implement or use information systems. The DSRM entails six methodology activities: (1) *Problem Identification*, (2) *Artefact Objectives*, (3) *Artefact Design & Development*, (4) *Demonstration*, (5) *Evaluation* and (6) *Communication*. The methodology activities are illustrated in Figure 2.1.

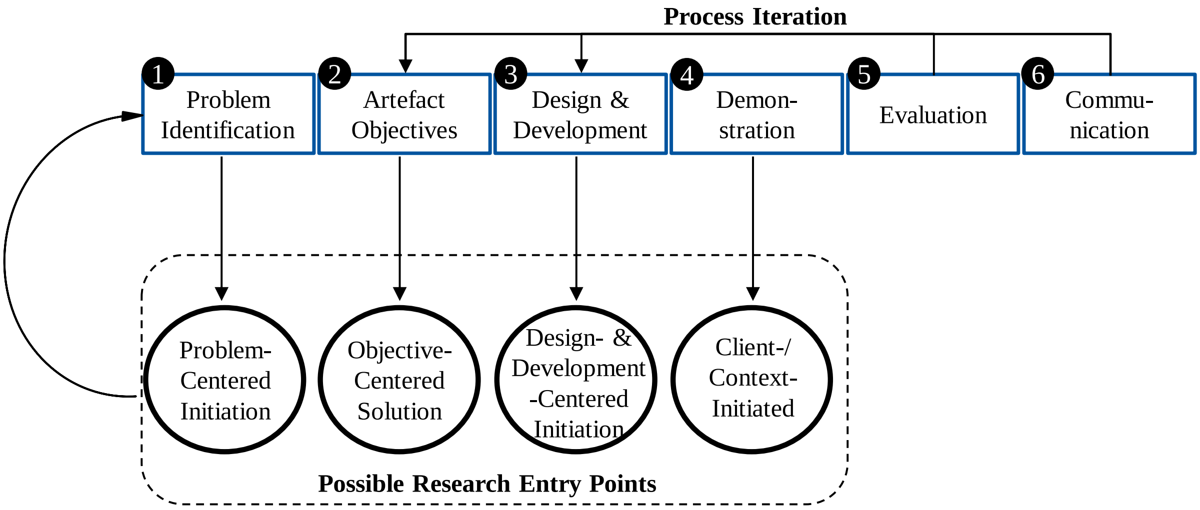


Figure 2.1: DSRM Process Model modified from *A design science research methodology for information systems*, by K. Peffers, 2007, p. 54

The first and second activity to initially identify and define the problem before developing objectives for a solution to this problem are touched upon as part of the introduction. Nonetheless, the problem field as well as corresponding, solution-tailored objectives remain vague and need to be further narrowed down in order to design an adequate solution afterwards. A general understanding of the studied digital platform, the *Nvidia Omniverse* used by an automotive OEM to enable industrial metaverse applications aimed at the creation of virtual factory models, needs to be established. Accordingly, a PRISMA literature review, stakeholder analysis and comparative software analysis of relevant digital platform architectures are presented to comprehend the scientific as well as industrial surroundings and argue the relevance of the problem field. Thereby, the first and second thesis sub-questions are answered. Based on qualitative expert interview data, an FMEA is conducted to assess the occurrence, detectability and severity of insufficiencies within data sets used to visualise virtual factory models. The artefact to be subsequently designed is intended to identify the particularly severe data quality insufficiencies. Based on the stakeholder requirements, concrete solution objectives for the artefact to be designed in activity three are defined and Q3 and Q4 are answerable. The third activity holds as the core of the DSRM by Peffers et al. (2007) and is characterised by initiating the design and development of a solution to the defined problem in which a research contribution is encapsulated. The artefact to be designed is a Python extension to the *Nvidia Omniverse* software. The development of this software extension is described as part of activity three. After designing the software extension, its use to identify the most critical data quality insufficiencies is demonstrated through a test at multiple files within the digital platform. Activity five, *Evaluation*, includes a comparison of the solution objectives to the observable results produced by the extension. Furthermore, it is evaluated whether and if so, which potential actions might be derivable from the software extension output to ultimately improve data quality within virtual factory models. Quantitative or quantifiable measures, such as KPIs, need to be used to perform this evaluation. After activity five, Q5 is answered by presenting a Python extension to the existing digital platform software that is capable of automatically scanning the platform and identifying the prioritised quality insufficiencies. Finally, the created software extension's utility and effectiveness to target the problem is communicated to relevant audiences, especially future users of the studied platform. Iterations and consequential redesigns to improve the software extension functionality are possible based on the provided feedback from testing the solution.

2.2 Methodological Approaches

Before outlining the pursued methodological approaches within the DSRM activities in the subsequent chapters, academic reasoning needs to be provided as to why these methods were chosen:

Firstly, desk research is pursued to arrive at a PRISMA literature review. The purpose of this method is to collect timely research that is of relevance to the studied concepts. The collected research then needs to be synthesised into one cohesive summary to provide the reader with a sufficient overview of the existing knowledge surrounding digital manufacturing platforms, the industrial metaverse vision and data quality. Thereby, a common knowledge foundation is set based on which the findings developed in this thesis can understandably be argued.

Secondly, a comparative analysis complements the results of the literature review in the *Problem Identification* activity. The academic reason for making use of this method is to both provide an overview of the evolution of the unique *Nvidia Omniverse* architecture as well as to analyse how other software compares to the *Nvidia Omniverse*. This provides context and a minimum degree of generalisability before studying the problem only at the specific case of one automotive manufacturer.

The third main method used within this research are interviews with technical experts. The reason to this approach is of pragmatic nature as the project is conducted in collaboration with the studied automotive manufacturer. Therefore, it appears sensible to make use of all knowledge resources providable by this manufacturer to arrive at an ideal result and identify the existing data quality insufficiencies based on found qualitative data.

The use of an FMEA can academically be reasoned as necessary to perform an adequate, scientifically funded prioritisation of the most critical insufficiencies based on the data obtained from the previous interviews. It is a structured approach used in the design phase of a product or software to estimate the impact of failures, root causes and potential mitigation strategies. It therefore perfectly fits the requirements for this research project.

Finally, empirical software engineering holds as the key method of this thesis. The qualitative data collected from expert interviews and analysed through an FMEA is used as empirical input to guide the engineering of a software extension to the *Nvidia Omniverse*. Thereby, the development approach is funded by the results from the previous methods and fulfills the purpose defined as part of the activity *Artefact Objectives* to solve the problem identified as part of the *Problem Identification*.

In combination, the employed methods serve as the ideal, academically reasoned approach to solve the identified research problem.

2.2.1 Desk Research

To establish a basic understanding of the unit of analysis, namely digital platforms used to enable an industrial metaverse, an extensive literature review is required. This literature review is conducted through desk research and its content detailed as part of chapter 3.

The literature search is based on a simplification of the updated guideline for *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (PRISMA) presented by Page et al. (2021). It therefore includes the four phases of *Identification, Screening, Decision* and *Inclusion*. In the *Identification* phase, searches are designed to adequately describe and narrow down the problem area touched upon as part of the introduction. The automotive OEM serving as unit of observation is making use of *Nvidia Omniverse* digital platform to facilitate collaboration and virtually represent its manufacturing systems. Through preliminary talks with involved employees, it was revealed that insufficient data quality has a negative influence on the functionality of the platform and consequently its collaborative value. From the preliminary talks with employees of the observed automotive OEM, a number of keywords was derived and converted into one coherent query (see Appendix A.1). The final search query based on which new studies via databases were identified was defined as: *TITLE(("digital platform*" OR "CPS") AND ("data governance" OR "digital twin*" OR "simulation"))*. The search was limited to results included in the title to only focus on the absolute core papers dealing with concepts surrounding the issue at hand. Two databases, namely *SCOPUS* and *Google Scholar*, were used to arrive at related articles and publications relevant to the mentioned issues. The PRISMA flow diagram depicted in Figure 2.2 is a modified version of the flow diagram by Page et al. (2021). It describes the systematic process of arriving at those studies that were identified to be eligible for inclusion in this research. The search was conducted on the 26th March 2023.

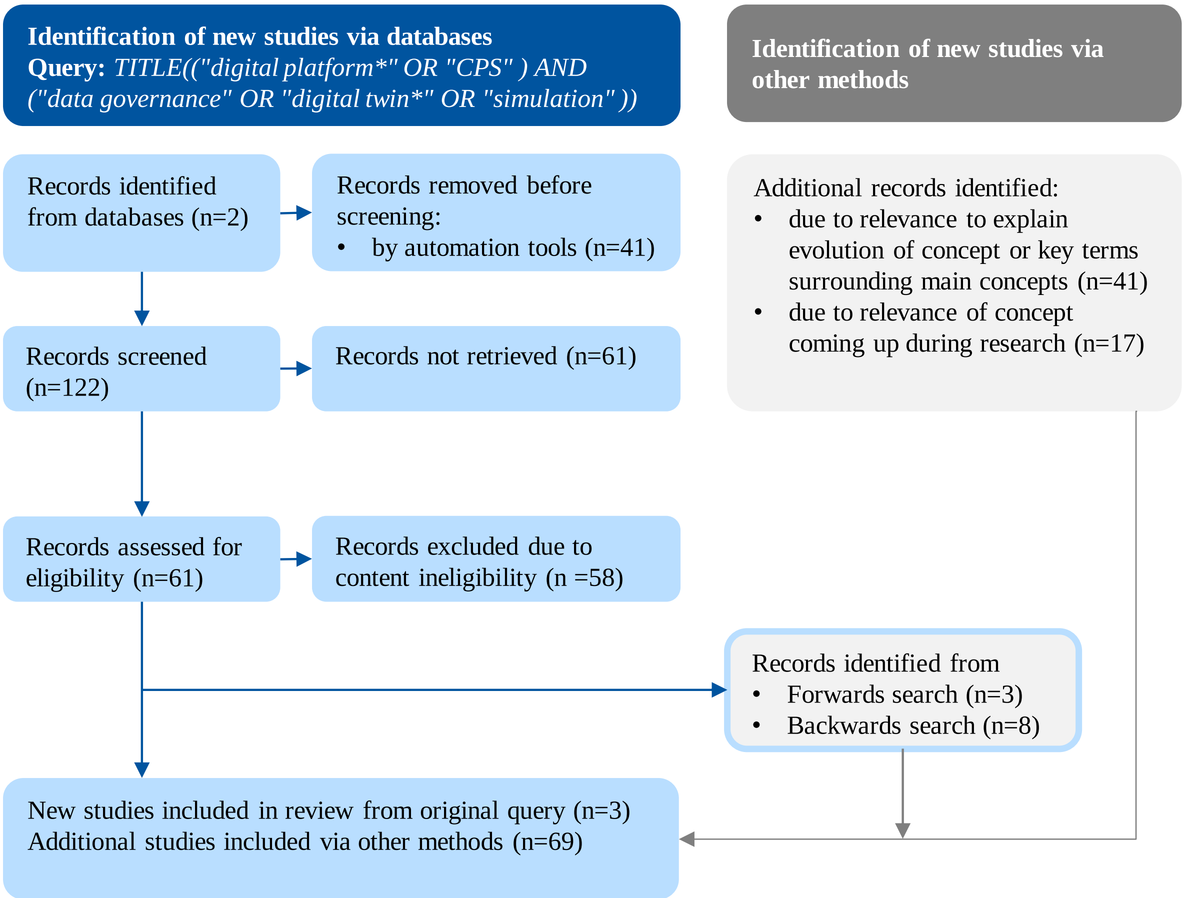


Figure 2.2: PRISMA 2020 flow diagram modified from *The PRISMA 2020 statement: an updated guideline for reporting*, by M. Page et al., 2020, p. 5

The automation tools used in step one filtered out 41 records that were published in 2016 or earlier. 122 records were then screened of which 61 were excluded due to unavailability for open access. From the 61 records assessed for eligibility, only three were included in the final review as the majority was excluded for only limited applicability of the records' content to the field of study. In addition to this, other methods were used to identify further studies of relevance: Firstly, further records necessary to outline the context and evolution of certain key terms and concepts were included. Moreover, records covering two additional concepts, namely the industrial metaverse and spatial data quality, were included as new findings and ongoing interview insights during the course of the research project revealed the relevance of these concepts. Consequently, these concepts were not covered by the original query and related records needed to be identified and screened separately. Lastly, three records were identified through the means of a forwards search and eight papers following a backwards search based on the three studies included from the original search query. In total, 72 eligible studies were included in the simplified PRISMA literature review (see Appendices A.2, A.3, A.4). At the end of PRISMA step four, *Inclusion*, I arrived at a selection of core concepts categorising the studies found to be eligible due to theoretical, societal or practical significance. These concepts are presented and the corresponding literature analysed as part of chapter 3.

2.2.2 Comparative Software Analysis

The core deliverable of this thesis project is the design of an extension to the *Nvidia Omniverse* platform software that allows for automatically scanning and identifying a number of data quality insufficiencies within a virtual factory 3D environment. To make sure that the designed software is not only applicable to the use case of observation, generalisability of the project results must me ensured throughout the entire thesis. Therefore, the architectures of comparable, potentially competing software must be reviewed before presenting the software architecture of the *Nvidia Omniverse* to justify the generalisability of the developed software extension.

2.2.3 Expert Interviews

As part of DSRM activity two, concrete solution objectives need to be defined. Therefore, the target state of how a software extension to automatically identify data quality insufficiencies should function must be precisely known. In order to define this target state, data about the existing data quality insufficiencies must be collected and analysed. At the start of the research project, the *Nvidia Omniverse* platform to virtually visualise the automotive OEM's factory can only be manually scanned. This not only holds as the key problem to be tackled by this thesis, but also results in the fact that quantitative data regarding the type, severity, occurrence and detectability of data quality insufficiencies is not obtainable. Therefore, qualitative data in the form of expert knowledge by current users must be collected through interviews to assess these aspects. The objective of the interviews is to collect data that can subsequently be used as input to conduct a software FMEA (see chapter 2.2.4) and analyse this data. The result of the FMEA is an action priority matrix which visualises a prioritisation of the data quality insufficiencies to identify those whose automatic identification is of highest urgency. Identifying those most critical insufficiencies is the objective of the software extension to be designed in the third DSRM activity.

At the time of data collection, there are < 20 people working with the *Nvidia Omniverse* at the OEM. Of these selected individuals, less than ten possess the expert knowledge necessary to provide reliable qualitative data on the types and impact of existing data quality insufficiencies. Only these technical experts are capable of holding as interview partners and giving their qualified rating of the severity, occurrence and detectability of existing data quality insufficiencies. As time constraints are an additional limitation to the thesis project, the time-efficient non-probability sampling technique of judgement sampling is employed to select suitable interviewees.

2.2.4 Failure Modes and Effects Analysis

Following from the interviews, the collected expert knowledge must be analysed to prioritise all spatial data quality insufficiencies within the *Nvidia Omniverse* 3D environment and thereby define the objectives of the software extension designed afterwards. To perform this data analysis, an FMEA is performed. The FMEA framework is a risk management technique to identify potential failure modes of a system and evaluate the effects on the system's performance resulting from these failures. Subsequently, strategies to either fully eliminate the failures or reduce their chance of occurrence and severity are proposed. The FMEA steps pursued within this thesis are guided by the work of Asan & Soyer and illustrated in Figure 2.3 (Asan & Soyer, 2015).

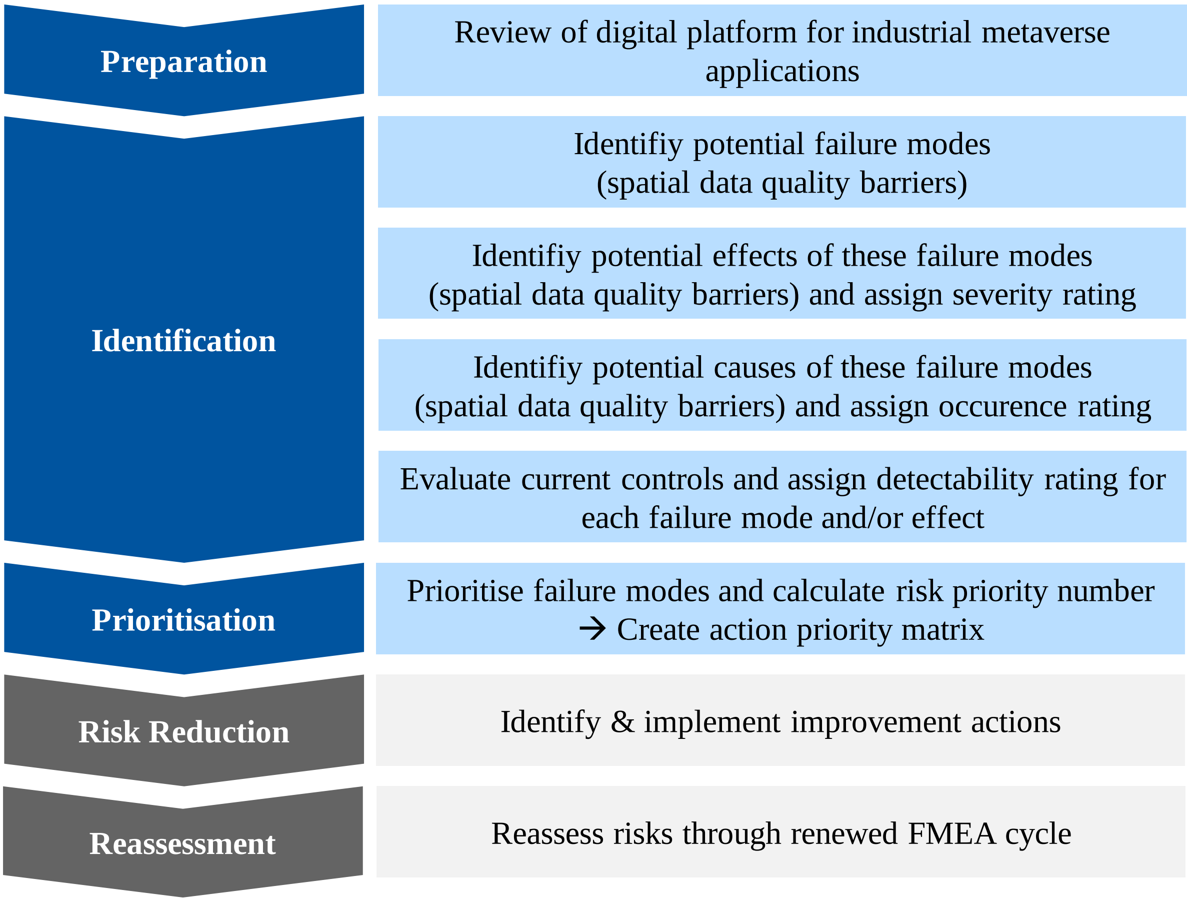


Figure 2.3: Steps of the FMEA Process

As the FMEA is only used as a tool to analyse the interview data and prioritise data quality insufficiencies of the *Nvidia Omniverse*, the steps *Risk Reduction* and *Reassessment* are not part of the thesis scope. It is intended to arrive at an action priority matrix that ranks the insufficiencies. Thereby, the most critical insufficiencies can be selected and targeted with the artefact to be designed in DSRM activity three.

2.2.5 Empirical Software Engineering

Building on the results from the primary and secondary research methods utilised in the problem identification and objectives definition phase, a software extension is designed to tackle this problem and achieve the objectives. The methodology used to actively utilise the knowledge obtained from empirical research methods to design software accordingly is called empirical software engineering and is the prevailing methodological approach pursued as part of DSRM activity three, *Design*.

2.3 Research Flow

Building on the described research design and its entailed methodological approaches, the research flow is visualised in Figure 2.4.

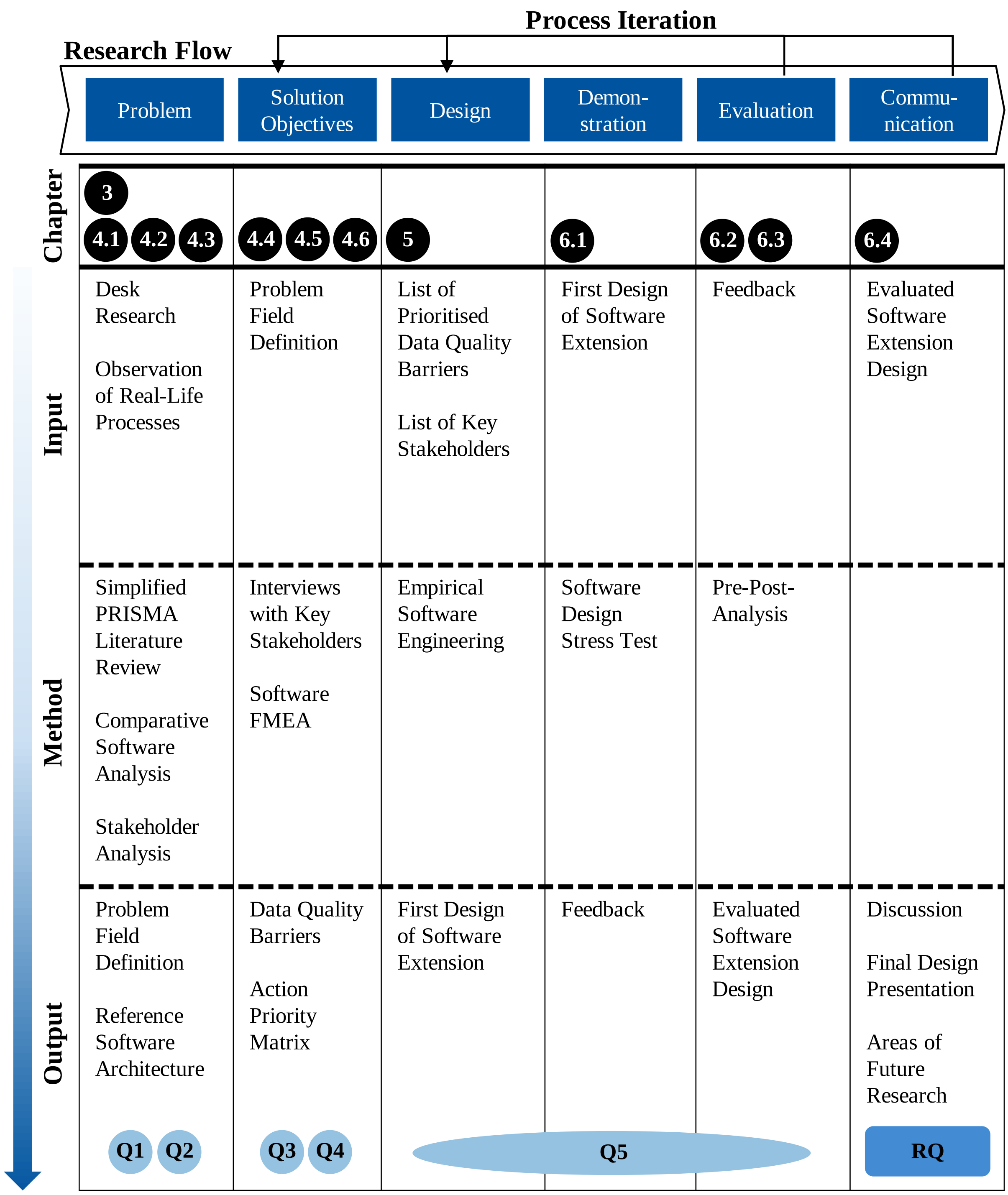


Figure 2.4: Thesis Research Flow Diagram

3 Literature Review

In the following, the current state of literature regarding the concepts to be included is assessed and the findings consequently presented. The concepts that are of theoretical, societal or practical significance are shown in Figure 3.1. These concepts categorise the previously identified, screened and decided upon studies (see chapter 2.2.1) which concludes the final step of the PRISMA literature review.

Practical/Societal/Theoretical Significance		Concepts
Rise of digital platforms as a preeminent driver of I4.0	▶	Digital Manufacturing Platforms
Industrial metaverse applications are a key enabler of smart manufacturing	▶	Industrial Metaverse Applications
Reliability of digital platforms to create a virtual factory depends on a data quality assessment	▶	Data Quality

Figure 3.1: PRISMA Step IV – Inclusion: Relevant Concepts of the Literature Review

I4.0, the fourth industrial revolution, is a paradigm to describe the trend towards automation, data exchange in manufacturing processes and human-machine-interaction. The resulting manufacturing system is characterised by improvements in terms of cost, time and flexibility (Mittal et al., 2020). The concept of I4.0 is perceived to be especially beneficial for larger enterprises (Julian M. Müller & Voigt, 2018; Sahi et al., 2020). Simultaneously, the digital transformation affects a numerous variety of economic and social processes, thereby creating new conditions for the digitalisation of society and the economy. There is a consensus among researchers that digital platforms are an integral part of these changes as they evolve to become a symbol for the formation of the *Digital Economy* and an accelerator of the digital transformation and ultimately I4.0 if used in manufacturing (Dmitrieva, 2020). Accordingly, OEMs have taken steps towards collaborative work and start utilising digital platforms (Constantinides et al., 2018; Huber et al., 2017). Such multi-sided platforms interconnect company departments, such as production planning, with suppliers and customers to enable network interactions between them and are summarised under the concept of *digital manufacturing platforms* (Hagiu & Wright, 2015; Julian Marius Müller, 2019; Oliveira & Cortimiglia, 2017).

Due to the extensive digitalisation of production facilities and its continuously improving infrastructure, the disruptive paradigm of an industrial metaverse moves into the focus of large manufacturing companies and especially OEMs. The industrial metaverse has numerous use cases, including remote monitoring, employee training, preventive maintenance and the prediction of future conditions of manufacturing assets (Chen (2017)). For the purpose of this thesis, industrial metaverse applications enhanced by the capabilities of digital twin technology and industrial process simulations that are aimed at bridging the gap between the physical and virtual world are of particular interest. Thereby, virtual models of entire factories can be recreated in a 3D environment. Especially in the manufacturing sector, a unified framework to develop digital twins has however not yet been defined (Tao et al. (2019)). In factory planning, the digital twin paradigm refers to an enhancement of the existing static processes and models in a CPS context. As this enhancement is not yet sufficiently covered by contemporary literature, background research on simulation and digital twin technology is conducted. Furthermore, BIM, a format to digitally represent architectural characteristics, is introduced. Lastly, the virtual factory concept is investigated as it incorporates for the integration of BIM, digital twin technology and process simulations to create an exhaustive virtual factory model.

The second concept to be reviewed is *data quality*. In manufacturing, computational intelligence and the *Internet of Things* (IoT) pave the way for the application of CPS. Especially the field of data quality generates a renewed interest in research considering the aforementioned developments of I4.0 and the *Digital Economy*. Manufacturing companies with the intention of implementing data-driven digital platforms to set up industrial metaverse applications, such as a collaboratively accessible virtual factory, face challenges to ensure the fulfillment of internal quality requirements when handling increasingly large sizes of spatial data. In order to comprehend these challenges, the subject of data quality needs to be structured and its dimensions reviewed. (Cheong & Chang, 2007; Kravets & Zimmermann, 2012; Sicari et al., 2018)

These concepts are investigated in the following subsections that are similarly structured: Firstly, the concept and all vocabulary is clearly defined and ambiguities in different literature threads regarding related phenomena and variables are identified and adequately explained. Subsequently, relevant literature is integrated, analysed and scholarly critiqued.

3.1 Digital Manufacturing Platforms

The roots of the concept of digital manufacturing platforms can be traced back to the 1960s, when advances in computing capability allowed for advances in manufacturing systems. Ever since then, research was conducted and different groups of scientists have presented their findings containing differing theoretical perspectives and research methods. The literature dealing with the evolution of the paradigm of digital manufacturing platforms, including all related subject phenomena and terms relevant to gain an understanding, is reviewed in this subsection.

The scientific paper of Chryssolouris et al. (2009) highlights characteristics and future trends in digital manufacturing by outlining the influence of the evolution of ICT on manufacturing. Due to digitisation advancements, materials and machines within a manufacturing process are no longer to be viewed separately in digital manufacturing, but as part of a manufacturing system in the need of effective coordination. Chen (2017) was among the first researchers to argue that the implementation of digital, intelligent manufacturing technologies is enabled by platforms. Their work stands in relationship to the solutions companies are adopting to deal with integration issues in manufacturing. Zhong et al. (2017) also provide a review of intelligent manufacturing, but, in contrast to Chen (2017), put their findings in an I4.0 context. This is complemented by the findings of Kusiak (2019) who provides a definition of intelligent manufacturing by referring to it as a concept of manufacturing aimed at optimising production by using advanced manufacturing and information technologies. The findings of Chen (2017), Zhong et al. (2017) and Kusiak (2019) add to the recommendations given by Kagermann et al. (2013). According to the authors, CPS and the IoT allowed for integrated manufacturing systems for the first time.

A slightly varying approach is presented by Alcácer and Cruz-Machado (2019), who, in their research methodology, make use of the smart factory concept to describe the impact of I4.0 on manufacturing systems. Opposed to the most common manufacturing environments, the smart factory concept stresses the core role distributed computing plays for I4.0. Wu et al. (2013) mention cloud manufacturing, an advanced manufacturing model supported by IoT aimed at providing on-demand manufacturing services from the cloud. C. Liu and Xu (2017) extend this literature thread by explaining the importance of joint standards such as the *Reference Architectural Model Industry 4.0* (RAMI 4.0), a layer and life cycle model that holds as a framework for companies on how to approach I4.0 in a structured manner.

De Reuver et al. (2018) are among the first to examine digital platforms and the surrounding ecosystems. The authors recognise the disruptive and transformative impact of digital platforms on organisations. Furthermore, De Reuver et al. (2018) find that platforms change the relationship and power structure between participants and discuss how device manufacturers and service providers strategise in a platform environment. Digital platforms create organisations in which value is co-created by multiple contributing entities that are not isolated. This view is mutually supported by Hein et al. (2020) who point out how digital manufacturing platforms, in contrast to traditional downstream contracting in a supply chain, evolved from a mutual dependency to collaboratively deliver work in a technology-driven environment. Such collaborative platforms allowing for real-time data exchange are presented as a promising technology by Z. Liu et al. (2022), who outline the potential such platforms could have for the I4.0 collaboration of SMEs.

Despite the numerous advantages that digital platforms like *DELMIA* from *Dassault Systèmes* or the *Siemens TeamViewer* introduced to manufacturing, most of these common platforms are prone to one significant disadvantage: Digital manufacturing platforms to this day usually demand for the adaption of a full software stack next to the platform. Functionality of the platform is only given in combination with other software that is also supplied by the platform provider which drastically hampers effortless integrability of digital manufacturing platforms into existing software systems. The *Nvidia Omniverse* is among the first platforms to change that state. One of its most key distinguishing features is the degree to which the platform is interoperable with other software, be it in manufacturing or any other industry. The advantages of such newly arising platform solutions and its capabilities are not thoroughly investigated in literature yet.

3.2 Industrial Metaverse Applications

While the meaning of the term *Metaverse* underwent several adaptations in its meaning due to technological advancements, the core of the concept refers to the creation of and interaction with a digital world that transcends the physical world and generates a symbiosis of virtuality and reality (Ball, 2022; Faraboschi et al., 2022; S.-M. Park & Kim, 2022). The work of J. Lee (1998) who studied the feasibility of an integrated virtual and physical model for remote manufacturing and maintenance can be argued to have laid the foundation for the metaverse concept in an industrial context. The industrial metaverse is a specification of the metaverse that is of highly emergent nature and was only touched upon by researchers recently. It is uniformly described as an industrial ecology that connects production-relevant assets, such as products, resources and human workers and seamlessly integrates them into the physical production through the use of various applications (Zheng et al., 2022). The role of the industrial metaverse as a part of the metaverse paradigm and its interaction with simulation and digital twin technology as the most relevant of these applications are depicted in Figure 3.2.

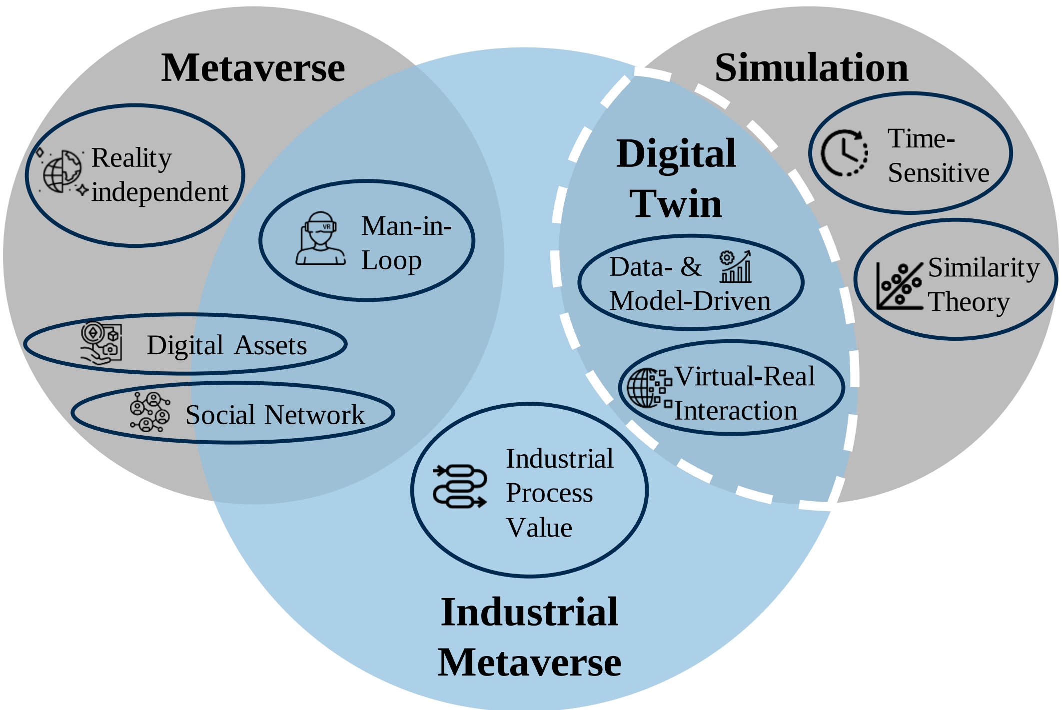


Figure 3.2: The Relationship between Industrial Metaverse, Digital Twin and Simulation, modified from *Industrial Metaverse: Connotation, Features, Technologies, Applications and Challenges*, by Z. Zheng et al., 2022, p. 241

B. Yang et al. (2022) mention the ability to interact between cyberspaces and actual spaces to broaden physical industry operations as the most important feature of industrial metaverses. A significant potential of industrial metaverse applications for OEMs lies in real-time 3D simulation and visualisation of physical entities to design an entire virtual factory in a shared virtual space (J. Kang et al., 2022). The virtual factory incorporates for two key enabling technologies whose capabilities saw advancements in recent years, namely digital twin technology and industrial process simulations. In the following, these technologies, fundamental to the development of a virtual factory, are further reviewed.

3.2.1 BIM

The adaptation of I4.0 ideas like digital representation is perceived as decisive to increase productivity in factory planning (Dallasega, Rauch, & Linder, 2018; Maskuriy, Selamat, Maresova, Krejcar, & David, 2019). For the purpose of creating a virtual factory, the role of digital visualisation methods in the field of factory construction and planning is of particular relevance. Here, the BIM concept is in the focus. BIM can be viewed as a starting point to I4.0 in factory planning as it is a method for representing physical and functional characteristics of buildings and thus the provision of 3D data. First published in a white paper of *Autodesk Building Industry Solutions* (2002), the core idea of BIM remains to be the combination of a technology with a process that defines how to use it (Hardin & McCool, 2015). Research on BIM does not significantly differ from a theoretical perspective. Through the years, scientists only extended the existing findings and added BIM dimensions to identify the type of information that is modelled in more detail (4D: cost, 5D: schedule, etc.) (Dallasega et al., 2020).

3.2.2 Digital Twin Technology

The digital twin concept was defined for the first time by Grieves (2002) in the context of industrial product lifecycle management as a digital informational construct about and linked with a physical system. This supports the notion pushed by subsequent literature describing digital twin technology as the mature evolution of BIM as it builds on the data collected from sources like BIM models that include information about the design and operation of a building and its interior. Digital twins advance this data by offering the opportunity to simulate the production system and create a dynamic virtual representation of the manufacturing process from a component level up to the entire factory (Kritzinger et al., 2018; Sepasgozar et al., 2023). As multiple concepts and solutions of a digital twin emerged over the past years across a multitude of industries, an incomplete and diverse understanding of this concept exists (Rosen et al., 2015; Zhang et al., 2022). Nonetheless, a widely recognised definition of Glaessgen & Stargel describes a digital twin as the "simulation of a complex product […] to mirror the life of its corresponding twin" (Glaessgen & Stargel, 2012, p. 1817). Garetti et al. (2012) and Negri et al. (2017) further specified this definition for the field of manufacturing in consideration of the I4.0 influence. According to the authors, digital twin technology synchronises the virtual and real system with the help of sensed data and connected smart devices to optimise and forecast the behaviour of a production system at each life cycle phase in real time. This perspective is congruent to the role of digital twin technology as an application of the industrial metaverse adapted from Zheng et al. (2022) in Figure 3.2. Digital twin technology allows for simulation, diagnostics and prediction on the digital replica and thereby bridges the gap between real and virtual world (Zheng et al., 2022). Accordingly, simulation is a core part of digital twin technology and plays a vital role in the creation of a virtual factory which is why the most relevant concepts surrounding industrial simulation technology are reviewed in the following section.

3.2.3 Industrial Process Simulations

Simulation applications first appeared in the early 20th century as a reference to deterministic laws to calculate the next state of complete physical mechanisms according to the mechanism's current boundary and condition (Zheng et al., 2022).

Nowadays, industrial process simulations are an important enabler of the industrial metaverse when combined with the capabilities of a digital twin and BIM. Thereby, virtual models that bidirectionally interact with physical entities in real-time can be created. Schroeder et al. (2016) were among the first to present a high-level model to realise interactions for traditional simulation purposes through data exchange between heterogenous systems. IoT middleware was used to store the attributes necessary to exchange data and make these attributes accessible to other systems. Building on this, Tan et al. (2019) proposed a digital-twin-specific solution to allow for real-time data collection and sharing from a physical space via IoT. To allow for the interlinkage of digital twin technology and industrial process simulation within a coherent metaverse environment, Boschert and Rosen (2016) found that multi-physics simulation over different levels of detail, all involved disciplines and several lifecycle phases must be integrated. Accordingly, by compiling specific simulations from the engineering phase together with corresponding digital twin models, existing know-how can be handled and used during the design and execution of the production system. Furthermore, consistency of the operation procedures can be ensured. A simulation model that integrates production line digital twins with its twinning, real production processes was demonstrated by J. Vachálek et al. (2017) who proved that unexpected changes in manufacturing processes can quickly be identified and reacted to through the benefits of real-time interaction between the physical and virtual space. Recent approaches increasingly feature the idea of using simulation and control systems as a complement to digital twins to create smart manufacturing systems that can be monitored and dynamically adjusted (Armendia et al., 2019; Ding et al., 2019; Weyer et al., 2016).

3.2.4 Virtual Factory

The virtual factory is a concept that comprises virtual representations of a real factory as an integrated simulation model of the factory and its subsystems. Thereby, technologically progressive decision support to evaluate and reconfigure existing as well as new smart production systems is provided (Jain et al., 2017; Lin & Fu, 2001). Especially manufacturers from the automotive industry are increasingly employing the virtual factory concept that allows for the integration and simulation of product, process and resource models concurrently and dynamically to support handling highly complex production systems (Jain et al., 2017).

There are several varying definitions of the virtual factory, all evolving around the idea of enabling integrated simulation of major factory components. The virtual factory proposed by Lu et al. (1998) describes a factory life cycle design in a virtual environment that builds on a real-time control system. His insights are confirmed by Souza et al. (2006) as well as Sacco et al. (2010) who suggest that multiple synchronised software tools and technologies are required to support the creation of a virtual manufacturing environment from a real factory. X. Yang et al. (2015) were among the first to connect the concept of the virtual factory with that of digital platforms by introducing the virtual factory as a collaborative analysis and design platform for manufacturing. Finally, Yildiz et al. (2020) integrated these approaches by proposing a digital-twin-based concept of the virtual factory at the intersection of product and production lifecycle which is illustrated in Figure 3.3.

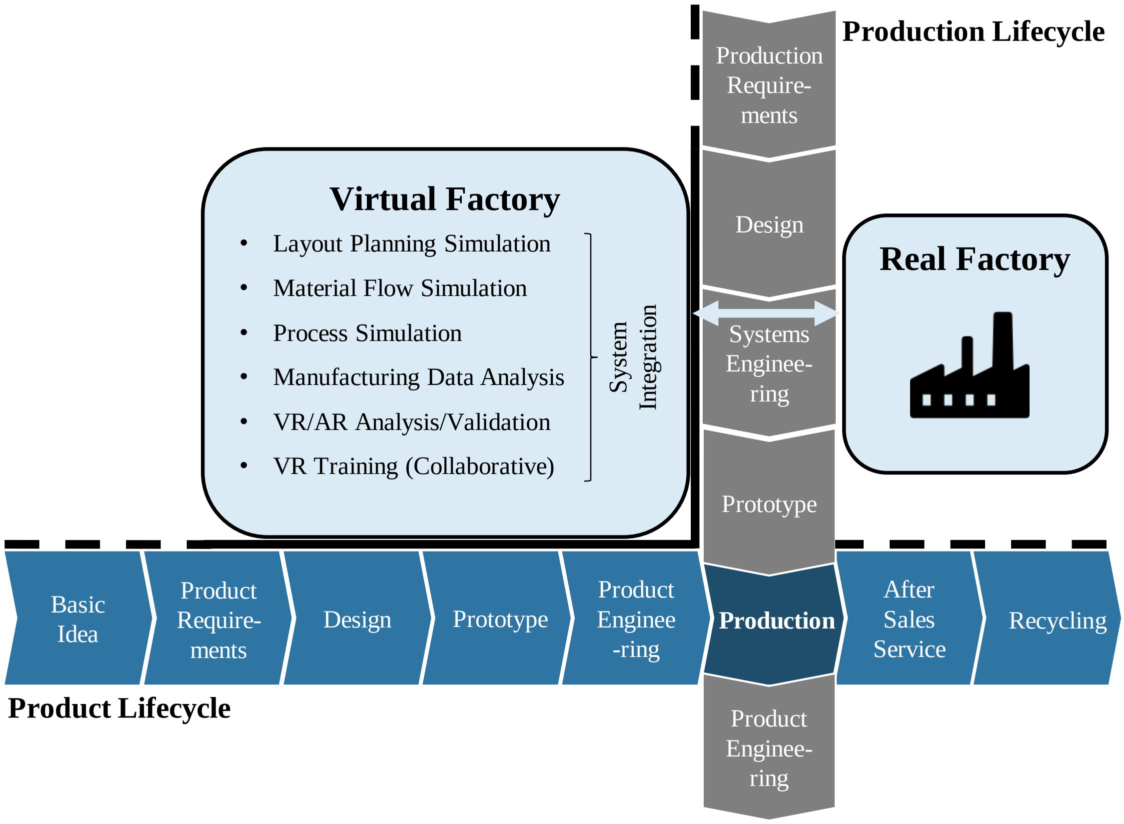


Figure 3.3: A Digital-Twin-Based Virtual Factory Concept, modified from *Virtual Factory: Digital Twin Based Integrated Factory Simulations*, by E. Yildiz et al., 2020, p. 218

3.3 Data Quality

In the following, the concept of data quality is firstly introduced in a general sense before presenting a detailed overview of the literature regarding quality aspects of spatial data relevant for industrial metaverse applications.

3.3.1 The Concept of Data Quality

Khatri and Brown (2010) present five decision domains for data governance: *Data Principles*, *Data Lifecycle*, *Data Access*, *Metadata* and *Data Quality*. *Data Principles* explains how data is linked to a business. *Data Lifecycle* details data processes. *Data Access* describes access rights while *Metadata* entails data about the data. Lastly, *Data Quality* refers to the state of information pieces and holds as the focus of this subsection. The five domains are complemented by Rosenbaum (2010), A. Tiwana et al. (2010) and S. U. Lee et al. (2017).

Wang and Strong (1996) provide a widely used definition as they propose four dimensions of data quality. This approach is particularly strong as the authors link the research fields of data and information quality into one comprehensive framework within their research methodology. The four dimensions of data quality by Wang & Strong are depicted in Figure 3.4.

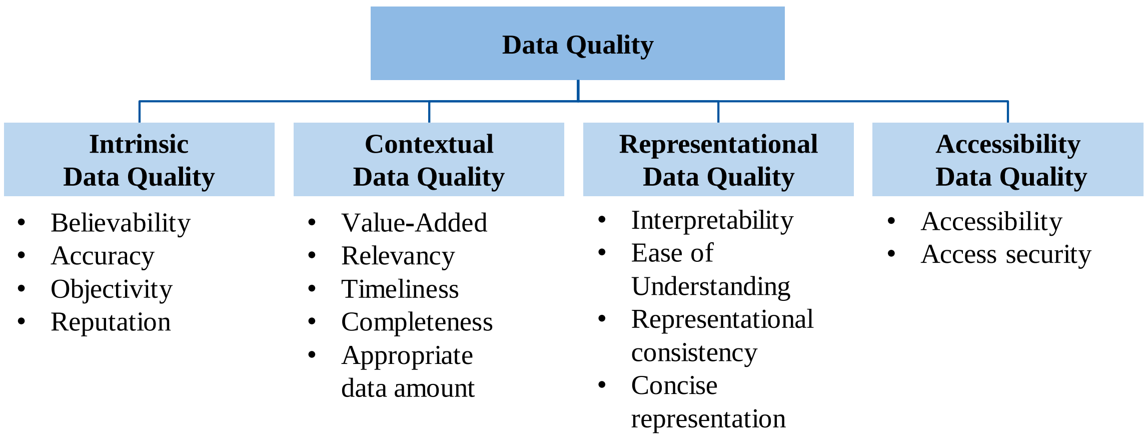


Figure 3.4: A Conceptual Data Framework of Data Quality, modified from *Beyond Accuracy: What Data Quality Means to Data Consumers*, by R. Wang & D. Strong, 1996, p. 20

A different approach was taken by Abate et al. (1998) who argue that a contextual perspective is necessary to sufficiently define data quality requirements in any situation. However, this definition is rather ambiguous which is why modern data quality definitions should be preferred: Hazen et al. (2014) and Gudivada et al. (2017) add to a literature thread based on the quality definition by the *International Standards Organisation* and describe data quality as the rate of which data satisfies its usage requirements. It can be recognised that a user-centric view to data quality is inherent to most authors across this literature thread.

Nokkala et al. (2019) and B. Tiwana et al. (2010) are among the first to examine the influence of data quality on the functionality of digital platforms. However, the work of these authors is not exhaustive neither is it applicable to digital manufacturing. B. Tiwana et al. (2010) present only a generic platform governance framework, Nokkala et al. (2019) aim at investigating the influence of data governance through interviews in the shipyard network, an inappropriate methodology as it is not generalisable to the problem field of interest.

3.3.2 Spatial Data Quality

While originally used for cartographic plotting, the possible applications of datasets in the field of spatial analysis are numerous and especially relevant for the creation of a metaverse environment. While quality limitations are highly significant in the processing of spatial data to precisely represent reality, contemporary researchers struggle to provide a unified definition of standard quality parameters. This is due to the complexity of spatial databases that include 3D geometric descriptions and spatial relationships between data objects (topology). The complexity and resulting high volume of metadata to be stored and organised for geographic information can only be handled by a wide network of users such as an organisation by incorporating for data quality standards when using this data (Devillers et al., 2007).

As mentioned, a unanimous consensus about the terminology surrounding spatial data quality does not exist. In 1987, the *National Committee on Digital Cartographic Data Standards* proposed a definition that entails four quantitative and one qualitative spatial data quality criterion (National Committee for Digital Cartographic Standards [U.S.], 1987). These criteria were extended by and refined by the French *Institut Géographique National* whose definition from 1997 holds as the guideline for the spatial data quality criteria used in this paper that are detailed in Table 3.1 (David & Fasquel, 1997; Morrison, 1995).

Table 3.1: Spatial Data Quality Criteria

Spatial Data Quality Criterion	Description
Lineage (Qualitative)	Acquisition procedures to derive and transform data from its material origin are up to standard
Geometric accuracy (Quantitative)	Deviation in geometric position between nominal ground and spatial data
Semantic accuracy (Quantitative)	Difference between values of non-spatial attributes and real value
Completeness (Quantitative)	Abnormal absence or presence of model, data, objects or attributes
Logical consistency (Quantitative)	Dataset is corresponding with structure of the model used
Temporal accuracy (Quantitative)	Data observation data management and data validity check
Semantic consistency (Quantitative)	Data is relevant and significant to the selected model
Specific quality (Quantitative)	Quality-related information unforeseen by previous criteria

4 Identification and Prioritisation of Data Quality Insufficiencies

In the following sub-chapters, the problem is narrowed down before defining objectives for the software extension to design. In accordance with the research flow (see Figure 4.1), the methodological approaches undertaken in activity one include: A PRISMA literature review, a comparative analysis of relevant software and a stakeholder analysis. The results of the literature review were presented as part of chapter 3 and are now combined with the outcomes of the comparative software analysis and the stakeholder analysis to ultimately define the problem and answer Q1 and Q2. Afterwards, in activity two, expert interviews are conducted before performing an FMEA and finally defining objectives.

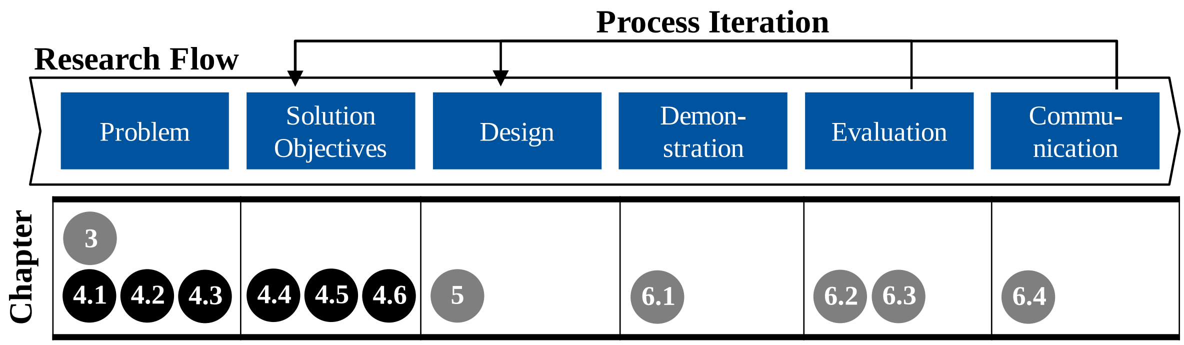


Figure 4.1: The DSRM Approach - Chapter 4

4.1 Comparative Software Analysis

As a means of ensuring a sufficient degree of external validity, the architectures of software similar to the *Nvidia Omniverse* need to be reviewed and analysed. As previously discussed, the use of the *Nvidia Omniverse* as a single platform to run industrial metaverse applications and create a virtual factory is considerably disruptive. As a consequence, similar software structures, for example of competitors to *Nvidia* that are used by other automotive manufacturers than the observed OEM, do not exist yet. Nonetheless, the architectures of comparable software leading up to the *Nvidia Omniverse* are worth to be examined in order to exhaustively comprehend the functionality and capabilities of this digital platform in the context of enabling an industrial metaverse. Furthermore, the *Omniverse* itself is presented more in detail and relevant terminology is introduced.

4.1.1 Enabling Software Architectures

The possibilities of an industrial metaverse in manufacturing are of interest to OEMs that work towards smart, integrated manufacturing systems characterised by a high efficiency due to real-time data analysis capabilities. Introduced in chapter 3.2.1, the BIM concept can be argued to have been the first step towards such a smart manufacturing system. BIM, in combination with the *Industry Foundation Classes* (IFC) open data format specification for building components, holds as the initial standard for a computable building representation (Nawari, 2018). IFC is an open standard *Computer-Aided Design* (CAD) data exchange model compatible to a range of software platforms for a multitude of use cases. The IFC architecture is characterised by the significant advantage of facilitating interoperability and data sharing among a variety of platforms, file types and actors (Temel & Basaga, 2020). It is based on three core pillars: (1) All relevant source systems to deliver the required IFC input, such as geometric data of the physical assets to be visualised, (2) a central database that integrates the input data in the form of a standardised IFC and (3) BIM applications to process and visualise this data. The interaction of these three components, modified from a paper of Kirstein and Ruiz-Zafra (2018), is shown in Figure 4.2.

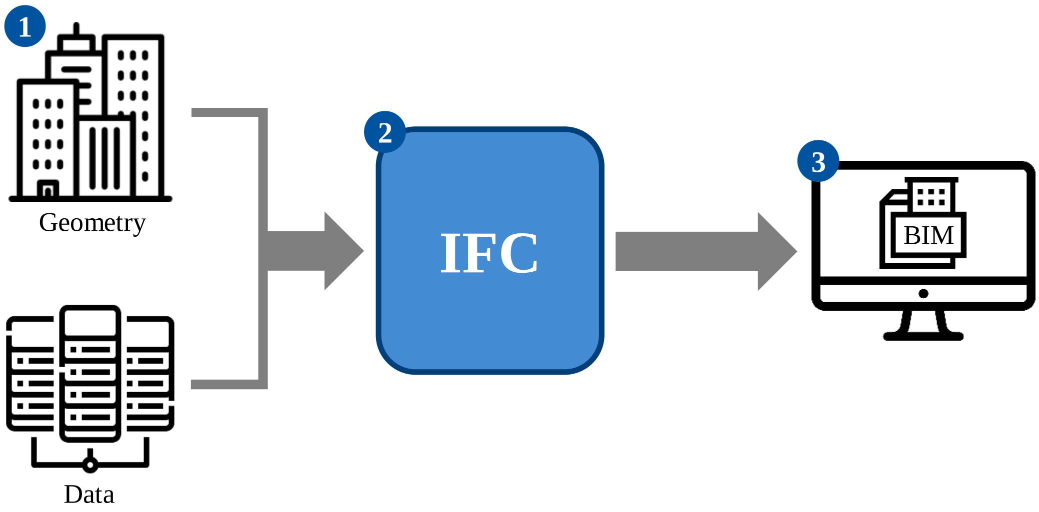


Figure 4.2: The IFC Data Schema, modified from *The Use of Templates and The Handle for Large-Scale Provision of Security and IoT in the Built Environment,* by P. Kirstein and A. Ruiz-Zafra, 2018, p. 3

Next to the IFC data schema, an understanding of the most relevant software architectures surrounding CPSs needs to be established as these systems play a crucial role to achieve the integration of dynamic geometric data from digital twins or simulations.

In literature, three core CPS architectures for smart manufacturing are distinguished: (1) *3C*, (2) *5C* and (3) *8C*. The *3C* CPS architecture integrates the components *Customer*, *Content* and *Coalition*. While the *Customer* facet emphasises the role of the customer within the production process, the *Content* facet contains relevant activities to ensure product traceability. Lastly, the *Coalition* facet covers production and value chain integration between those parties involved in the production process. The *5C* architecture proposed by Bagheri et al. (2015) consists of five levels differing from the *3C* facets. The *5C* levels build upon each other in the following order: *Connection*, *Conversion*, *Cyber*, *Cognition* and *Configuration*. Its focus lays on combining the production actuators and sensors to enable high-level automation and cognitive modules and support intelligent decision-making. The *8C* CPS architecture combines both models by adding the three horizontal facets of *3C* to the vertical *5C* pyramid (Ahmadi et al., 2019; S. K. Jagatheesaperumal & M. Rahouti, 2022).

The *Anthropocentric Cyber-Physical System* (ACPS) reference model exceeds these classical CPS architectures as it explicitly aims at smart factory automation through the integration of human, cyber and physical components (Pirvu et al., 2016). Ahmadi et al. (2019) proposed an enhanced model that combines the *3C*, *5C* and ACPS architectures and thereby includes the horizontal and vertical diversification necessary to achieve cyber-physical integration on one platform while also accounting for the main related interface components. Thereby, this enhanced model is particularly suited to enable cyber-physical integration in an I4.0 context, for example by digitally visualising factory assets. A simplification of this model is illustrated in Figure 4.3. (Ahmadi et al., 2019)

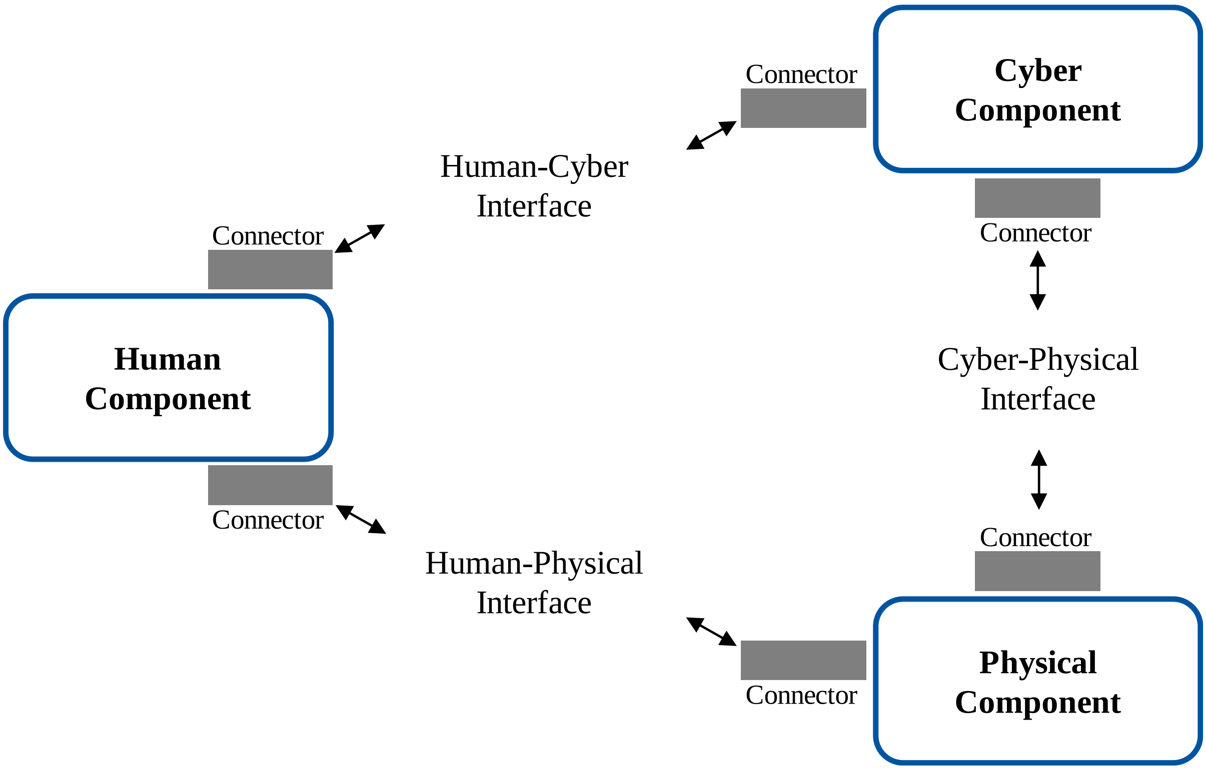


Figure 4.3: CPS Architecture for Smart Manufacturing, modified from *Evolution of 3C Cyber-Physical Systems Architecture for Industry 4.0*, by A. Ahmadi et al., 2018, p. 452

Further enablers of software whose architecture is comparable to that of the *Nvidia Omniverse* are the aforementioned RAMI 4.0 and the *Industrial Internet Reference Architecture* (IIRA). Both of these architectural models need to be introduced: The RAMI 4.0 adapts with earlier standards and simultaneously matures strategies to share information in smart manufacturing. Moreover, it allows for interoperability among different technological tools which is of immense relevance to unify the inputs of multiple source systems necessary to visualise a virtual factory (Yli-Ojanperä et al., 2019). The IIRA adds to this by integrating CPSs and the IoT to facilitate process automation and simulation (Radanliev et al., 2019; Xu et al., 2018).

These CPS architectures hold as the base for platforms that integrate industrial metaverse applications featuring BIM, digital twin technology and simulation. Platforms to comprise industrial metaverse applications and enable smart manufacturing are a phenomenon that is only recently emerging in literature. A comparative analysis of software similar to that of the *Nvidia Omniverse* studied within this thesis is therefore only possible to a limited extent. Nonetheless, S. K. Jagatheesaperumal and M. Rahouti (2022) present a comparable reference architecture that takes into account the horizontal and vertical diversification that comes with integrating cyber and physical assets on one platform. A simplified model of this architecture that solely focuses on the elements required to enable the creation of a virtual factory is presented in Figure 4.4.

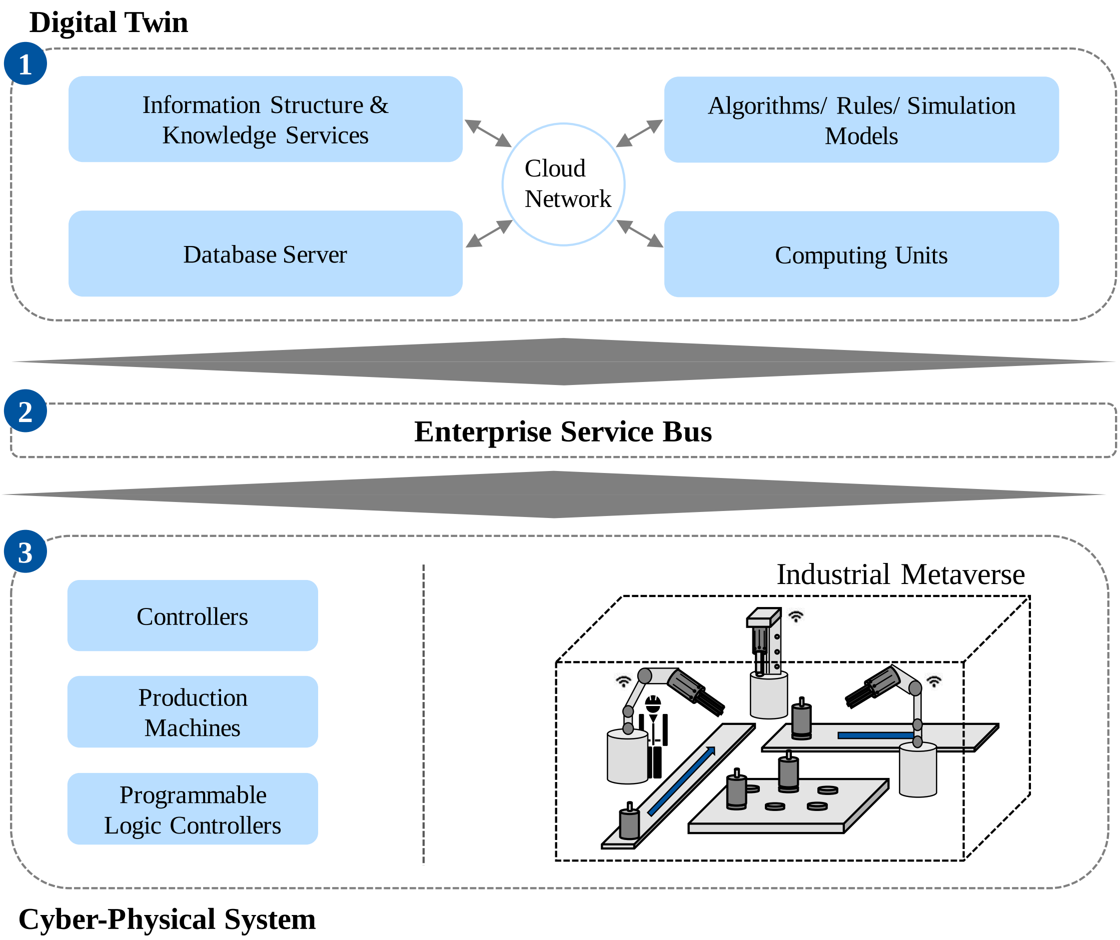


Figure 4.4: Software Architecture of a Metaverse-Enabled Digital Twin Application, modified from *Building Digital Twins of Cyber Physical Systems with Metaverse for Industry 5.0 and Beyond*, by S. Jagatheesaperumal and M. Rahouti, 2022, p. 37

As illustrated above, a metaverse platform that integrates the input of multiple source systems to allows for the digital visualisation of manufacturing systems makes use of and refines the same three key components as the aforementioned IFC architecture: (1) Source systems to deliver the required input, such as simulation data and 3D data of the physical assets to be visualised, (2) a database and collaboration engine that integrates this input data in a unified standard format and (3) the actual visualisation platform connecting the integrated data to create one coherent factory model.

4.1.2 The *Nvidia Omniverse*

Currently, the company *Nvidia* is the considerably dominant supplier of software capable of integrating industrial metaverse applications. The company's *Omniverse* that serves as the unit of observation for this thesis has use cases in multiple industries. These industries include science and research, retail, entertainment and media, construction and architecture and lastly manufacturing. Especially in automotive manufacturing, *Nvidia* has formed collaborations with several OEMs, the most recent one being *Mercedes-Benz* (Nvidia, 2023a). The *Nvidia Omniverse* software architecture is identical for every use case: It includes the three components depicted in Figure 4.2 and Figure 4.4.

As the manufacturing industry in general and the creation of a virtual manufacturing system and ultimately a whole factory in particular serve as the use cases studied within the scope of this thesis, the *Nvidia Omniverse* software architecture tailored to this use case is modelled and illustrated in Figure 4.5.

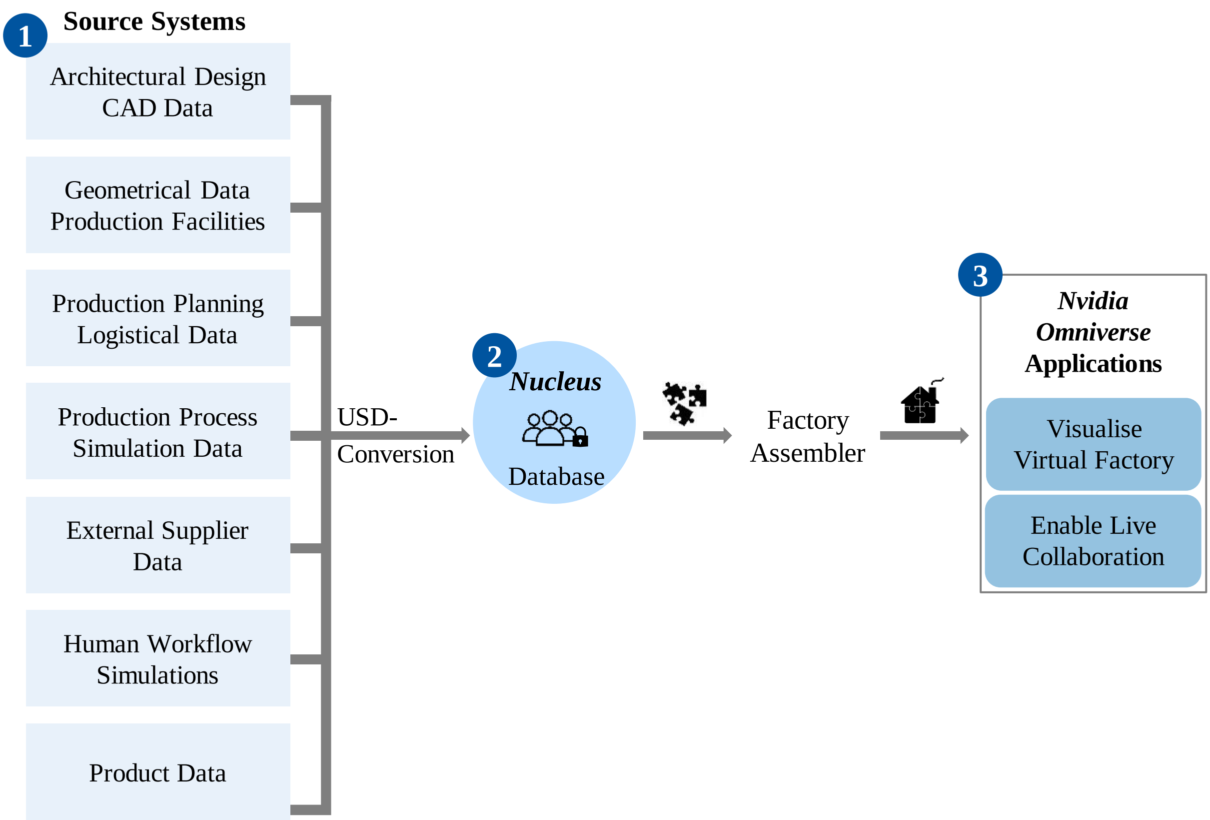


Figure 4.5: The *Nvidia Omniverse* Software Architecture at the Example of an Automotive Manufacturer

The different source systems provide CAD *two-dimensional* (2D) and 3D data in various supported file formats from factory assets such as single bolts up to whole production plants and buildings. Complemented by simulation data, each input data point is then converted to files of the *Universal Scene Description* (USD) format. Originally open-sourced by *Pixar Animation Studios* for the creation of visual film effects, this open scene description allows users to collaboratively create, render and edit virtual models in a 3D environment. Due to its scalability, high resolution virtual models of reality can be enhanced by the capabilities of simulations, digital twin technology and BIM to unlock industrial metaverse applications such as a virtual factory. Its standardising purpose is equivalent to that of IFC in BIM, however, the degree to which this format is interoperable creates a decisive advantage which makes the *Nvidia Omniverse* particularly unique. The provided USD files are then integrated on the

Nvidia Nucleus, the database and therefore second core software component. USD files for all data that is to be included in the later virtual factory model are located on this database. Individualised access rights prevent misuse and provide the necessary degree of confidentiality and security. From the *Nucleus*, the unorganised USD files pass the *Factory Assembler*, a tool to compose all virtual factory parts for different production plants by assigning the plant site and real-life position in this plant to each USD file. Finally, those assembled factory models are visualised in the *Nvidia Omniverse* as digital twins in an immersive 3D environment. The *Nvidia Omniverse* allows for collaborative work within the factory models and can therefore be argued to create an industrial metaverse based on the definitions provided in chapter 3.2.

Live collaboration and editing of the virtual factory are possible within the 3D environment, so that factory layout planners can dynamically move assets and evaluate different arrangements in the virtual models of current and future factories. The key advantage of the *Nvidia Omniverse* compared to all other co-existing digital platforms is its interoperability with a variety of different other software solutions due to the *Pixar* USD format on which the platform is based. Thereby, the *Omniverse* can be integrated into many existing software structures and easily adopted by new companies from different industries.

4.1.3 *Nvidia Omniverse* Terminology

Each virtual factory model is represented as one USD stage in the *Nvidia Omniverse*. A USD stage is defined by a tree of interdependent *primitives* (prims), the basic unit of USD in the *Nvidia Omniverse*. Every asset, light and material from the physical factory, be it a single bolt or the factory floor, can be virtually visualised in the form of one or more prims. To edit the layout of a virtual factory, the relevant prims are moved in the 3D environment. Prims contain and order other prims, hold properties with relevant data and thereby are the only persistent scene graph object in a USD stage's memory. The identity for prims in a USD stage is given by their path. All prim paths form a tree that specifies what prims are children to a superordinated prim, for example a larger object or area within the factory with one root prim holding as parent to all virtual factory prims. Depending on the properties a prim holds, it is differentiated between different types of prims. The two most relevant prim types are Xform prims and mesh prims. Xforms mainly hold location data in the form of 3D coordinates while meshes are polygonal and contain geometric attributes. It is common for mesh prims to be the children to an Xform with both prims representing the same object while holding different sets of attributes that complement each other (Pixar Animation Studios, 2021). A simplified excerpt from a typical prim tree structure is illustrated in Figure 4.6. It shows an exemplary virtual factory building underneath which related factory production line segments are arranged which contain all assets located within this segment in the form of an Xform prim and its corresponding mesh prim.

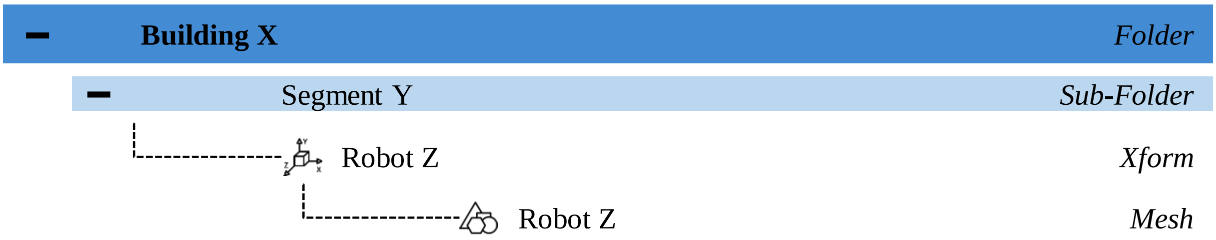


Figure 4.6: Exemplary Excerpt of a Prim Tree in a *Nvidia Omniverse* USD Stage

If the quality of any data involved in the creation of the virtual factory is insufficient, this is only observable in the final step, the virtual visualisation of the coherent factory in an industrial metaverse environment. Here, the virtual factory model is manually reviewed and scanned for data quality insufficiencies. This process is neither time-efficient nor reliable as not all insufficiencies are detectable using manual methods. This can be identified to be a key problem. So as to solve this problem, the data quality assessment of the virtual factory needs to be automated by the means of a software extension to the *Nvidia Omniverse*. The reasons to these data quality insufficiencies can be various, ranging from possibly flawed input data provided by responsible engineers or missing data standards in the source systems in part one to a malfunctioning conversion into the USD format or an unsuccessful integration of this data on the *Nucleus*. However, the reasons leading to the data quality insufficiencies in the virtual factory are not scope of the thesis, therefore not investigated in detail and only touched upon as part of the expert interviews.

4.2 Stakeholder Analysis

In order to design a software extension whose functionality fits the desires of future users and answer Q2, these key users need to be identified through the means of a stakeholder analysis. Furthermore, the stakeholder analysis serves the purpose of mapping out possible participants of the subsequent interviews and to conclude on DSRM activity one by defining the product owner who simultaneously is owner to the problem of study. The data based on which this analysis is conducted is derived from observations of company-internal workflows and relevant processes.

During the creation of this thesis, a first minimum viable product of the software was distributed to a small number (< 20) of users to initiate the creation of virtual factory prototypes at selected factory locations. At this stage, these users hold as the key stakeholders of the tool and are intended to function as intra-organisational early adopters of this innovation. Thereby, they shall drive acceptance to convince an early majority within the company to also adopt the software for purposes like factory layout planning. The key users are distinguished by their respective functional departments within the automotive OEM, which are listed in Table 4.1.

Table 4.1: Company-Internal Stakeholders of the *Nvidia Omniverse*

Stakeholder	Functional Department
General	
A	*Nvidia Omniverse* Software Rollout (≙ **Product Owner**)
B	*Nvidia Omniverse* Data Input (3D Visualisation & Simulation)
C	Factory Planning Process Standards & Methods
Factory A	
D	Virtual Factory Data Management
E	Factory Planning – Layout, Technical & Structural
F	Plant Installation
Factory B	
G	Virtual Factory Data Management
H	Factory Planning – Layout, Technical & Structural
Factory C	
I	Virtual Factory Data Management
J	Factory Planning – Layout, Technical & Structural
Factory D	
K	Virtual Factory Data Management
L	Factory Planning – Layout, Technical & Structural

To adequately address the problem at hand, the stakeholders' power bases in relation to their interests need to be determined. For that purpose, the power versus interest matrix by Eden and Ackermann (1998) proves useful. Here, stakeholders are arrayed on a two-by-two matrix depending on their power to affect the future trajectory of the analysed case and their interest in it. Thereby, four categories of stakeholders can be distinguished: stakeholders with little power and interest (*Crowd*), stakeholders with power but little interest (*Context Setters*), stakeholders with little power but interest (*Subjects*) and stakeholders with significant power and interest (*Players*). (Eden & Ackermann, 1998)

The company-internal power-interest-grid around the *Nvidia Omniverse* introduction to create a virtual factory is illustrated in Figure 4.7. The product owner and main responsible stakeholder is the department ensuring a fluent software roll-out during the early phase. This highly influential player is of particular interest for the interviews in chapter 4.4 as department members are working to solve software functionality barriers such as data quality insufficiencies as part of their occupation leading to particular expert knowledge in that field.

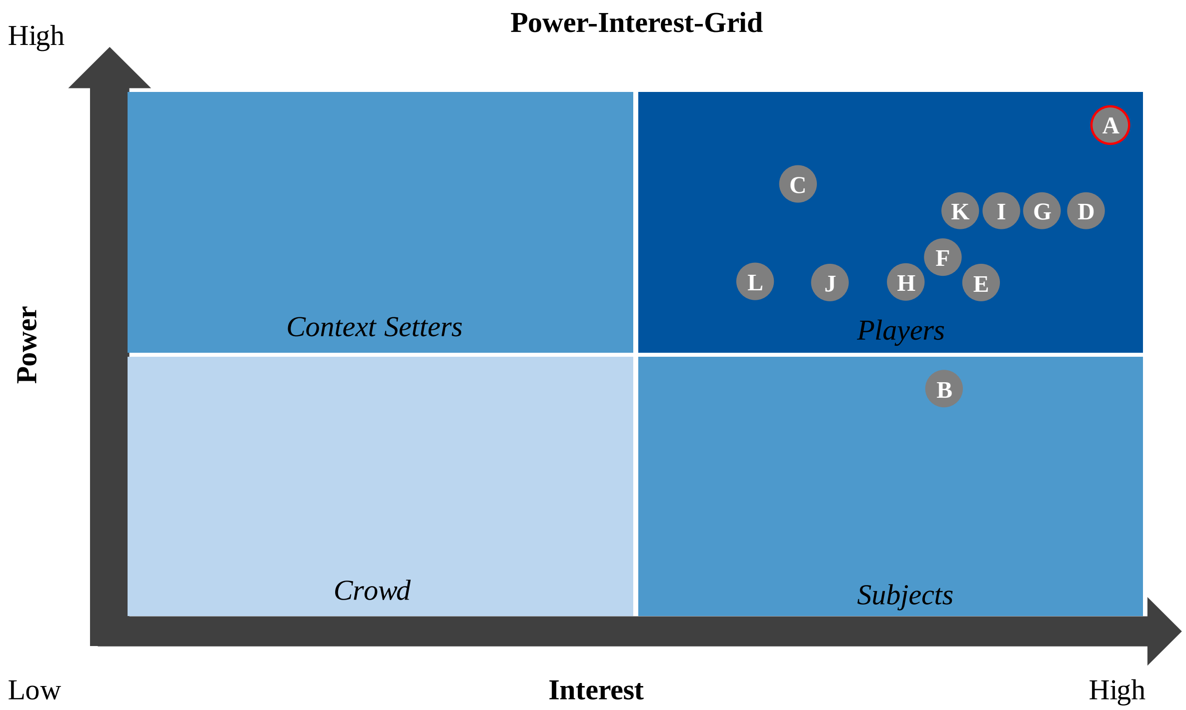


Figure 4.7: Company-Internal Power-Interest-Grid of the *Nvidia Omniverse* Stakeholders

4.3 Problem Definition

Building on the results of the PRISMA literature review, the comparative software analysis and the company-internal power-interest-grid for the *Nvidia Omniverse*, the problem can be narrowed down.

The *Nvidia Omniverse* is a platform-based software that allows for the collaborative development of a USD-based 3D environment. The software's areas of possible application span across multiple industries with the automotive industry holding as unit of observation to this thesis. Automotive manufacturers aim to use it to create virtual models of physical factories that either already exist or that are scheduled to be built within the coming years. From the creation of such virtual factories, manufacturers intend to take advantage of multiple benefits such as worker trainings using augmented reality or ergonomics analyses by integrating human models into the virtual factory. However, the most relevant use case of the *Nvidia Omniverse* in automotive manufacturing is the creation of virtual models for factories whose built-up is agreed upon, yet not initiated so far. Using the *Nvidia Omniverse*, factory planners can plan a future factory's layout collaboratively by converting the data of all relevant factory components from multiple formats and source systems into one coherent USD format and integrating it on one data base, the *Nvidia Nucleus*. From this *Nucleus*, factory planners such as those depicted in Figure 4.7, access the data and assemble it in the *Factory Assembler* to ultimately create a coherent virtual factory in which different layouts can be tested with minimal costs. This not only facilitates organisation-wide collaboration, but also maximises the efficiency of the planning process as all relevant stakeholders of a factory construction project can simultaneously discuss their ideas in real time to arrive at a consensus. The validation of these ideas in a virtual environment that exactly mimics the physical environment thereby not only saves costs, but also time and allows for manufacturers to speed up its overall factory planning and built-up process. Such time savings can make a decisive difference in the highly competitive and volatile automotive market environment.

Nonetheless, these benefits can only be taken advantage of if the *Nvidia Omniverse* renders the virtual factory not only fast, but flawless. Flawless rendering implies a virtual factory model that meets internal data quality requirements and is free of insufficiencies that either make layout planning impossible or hamper the general software functionality. Accordingly, the degree of fulfillment to which the internal data quality requirements are met needs to be obtainable and all possible data quality insufficiencies therefore identifiable. Currently, these data quality insufficiencies can only be identified through manual checks of the virtual factory models when already visualised in the *Nvidia Omniverse*. These manual checks are conducted by factory planners that are capable of identifying insufficiencies related to deviations between the virtual and the real factory as it was planned in the source systems. The costs related to the work time devoted as well as the non-eliminable margin of error of such human checks outweigh the potential benefits described above. As a result, the intra-organisational willingness to adopt the *Nvidia Omniverse* is restricted and the software can neither be used for factory layout planning nor for any other potential use cases that can only be explored if a significant number of employees are open to use the software. In conclusion, the problem to be solved is the non-existence of automatic data checks that identify insufficiencies and serve as a quality gate for the virtual factory models. This problem is to be solved by the outcome of this thesis.

Q1 was targeted at establishing an understanding of the current and potential future industrial metaverse applications of digital platforms. The concepts of digital platforms and the industrial metaverse building on BIM, digital twin and simulation technology were introduced as part of the PRISMA literature review (see chapter 3). Subsequently, a connection between these concepts was established through the means of a comparative software analysis. It is evident, that the *Nvidia Omniverse* currently holds as the only digital platform supporting potential industrial metaverse use cases. These use cases evolve around the concept of a virtual factory. By using 3D models to make decisions regarding a factory's layout, the efficiency of planning processes and ultimately production is optimised. Furthermore, human movement simulations in an industrial metaverse can allow for precise ergonomics analyses. With the results of such analyses, the decline in skilled workers due to an aging society can be absorbed by assigning adequate physical tasks to workers based on their age. This could enable workers in production to be capable of retiring at a higher age as working conditions in production would become more attractive. An additional use case of a virtual factory is the ultimate enablement of a fully simulated production system in which all supply chain actors are interconnected. Any disruptions in the production process would be recognisable in advance as the predictive capabilities of such a smart manufacturing system would reach unprecedented dimensions.

Q2, the analysis of all relevant internal stakeholders affected by data quality insufficiencies, was answered through the creation of a power-interest-grid (see Figure 4.7) that illustrates key platform users at the current stage. It must be noted that this power-interest-grid is projected to grow proportionally to the number of users, nonetheless, the product owner will remain the most important stakeholder.

4.4 Technical Expert Interviews

The studied software is still in an early state and reliable data on the issue of insufficient data quality has not been recorded yet. Therefore, only a selected group of people can provide the information required to adequately identify and prioritise data quality barriers of the *Nvidia Omniverse* platform. The non-probability sampling technique of judgement sampling appears to be adequate to select suitable interviewees to collect qualitative data on the existing data quality insufficiencies. The conducted research has a limited time frame and deals with the emerging concept of a digital platform for industrial metaverse applications. Only a small number of individuals possess the relevant knowledge of interest. Judgement sampling is a strategy to time-efficiently select interview subjects based on their unique expertise and is therefore the most suitable and consequently employed sampling technique within this research. Even though generalisability is limited with this technique, it must be taken into consideration that the results of the interviews do not need to be generalisable to the entire society, but only to comparable use cases, respectively software. In chapter 4.1, it was established that the *Nvidia Omniverse* is the dominant software of this kind and adopted by a variety of differing industries and use cases. The results of this research are hence generalisable to all of these use cases. Therefore, judgement sampling is preferred to any probability sampling techniques.

A selection of future key users that are already working with the tool was presented as part of chapter 4.2. However, among these key users, a distinction between key stakeholders and technical experts has not been made yet as not all key users possess the traits looked for to answer the interview questions, namely expert knowledge about all possibly occurring data quality insufficiencies. The data to fill out the FMEA and thereby prioritise data quality insufficiencies must accordingly only be collected from interviews with technical experts among the key users.

As the number of key users at the time of data collection is < 20 users, only three prerequisites are formulated for the experts to comply to in order to be eligible as an interviewee. These prerequisites are:

- The interviewee is a key user of the digital platform minimum viable product
- The interviewee is significantly involved in setting up the platform for the creation of a virtual factory and thereby qualifies as a technical expert
- The interviewee is willing to provide this expertise

As depicted in the power-interest-grid (see Figure 4.7), the *Nvidia Omniverse* rollout team is the key player. Key users from this department are ensuring the platform functionality during the rollout phase of the minimum viable product. During this phase, a multitude of issues, including data quality insufficiencies, arise and are taken care of by these users. Therefore, users from stakeholder group A (see Table 4.1) make up the majority of interviewees due to their considerably valuable technical expertise. The rest of the interviewees are data managers of selected factories that hold responsibility for the flawless creation of virtual models of their respective factories. The interview process including all relevant information is detailed in Table 4.2.

Table 4.2: The Technical Expert Interview Process

The Interview Process	
Interview Type	Semi-structured
Sampling Technique	Judgement sampling (non-probability sampling)
Input	Interview protocol with semi-structured questions
Duration	30 minutes
Interview Objectives/ Desired Output	1. Creation of an exhaustive list of all existing data quality insufficiencies when visualising virtual factory in *Nvidia Omniverse* (+ free discussion of potential causes and approaches to fix) 2. Assignment of a severity, occurrence and detectability rating for each insufficiency and its effect as input for the software FMEA
Interviewees	Six Technical Experts i. *Nvidia Omniverse* Software Rollout: Visualisation Lead ii. *Nvidia Omniverse* Software Rollout: 3D Data Input & Conversion iii. *Nvidia Omniverse* Software Rollout: Data Provisioning Lead iv. *Nvidia Omniverse* Software Rollout: Plant Built-Up Coordination v. Virtual Factory Data Management: Factory A/B vi. Virtual Factory Data Management: Factory C

The interview questionnaire including the scales used to evaluate the severity, occurrence and detectability rating is provided in Appendix A.5. The collected data quality insufficiencies within the *Nvidia Omniverse* to visualise a virtual factory are categorised based on the spatial data quality criteria presented in Table 3.1. Especially the first part of the interviews was conducted in a semi-structured way to allow room for free discussions. Through these discussions, the root causes for the insufficiencies were explored: All insufficiencies observable in the virtual factory model are either traceable to (1) a workflow issue, so human errors and miscommunication of those providing the data in the source systems, (2) missing standards in the source system causing the conversion of non-standardised data sets or (3) an error within the conversion process to USD. Table 4.3 presents an exhaustive list of all existing data quality insufficiencies. A total of 18 data quality insufficiencies are observable for users of the *Nvidia Omniverse* when visualising virtual factory models.

Appendix A.7 contains an exhaustive list of these data quality insufficiencies including a detailed description of the error pattern as well as the severity, occurrence and detectability ratings assigned by the interviewed technical experts. In case experts differed in their rating of severity, occurrence or detectability, an average was computed for these ratings. Overall, the results of the expert interviews were unanimous and a subsequent best-worst decision-making method therefore found to not be required.

Table 4.3: Categorised List of Data Quality Insufficiencies within the *Nvidia Omniverse*

Nr.	Short Name	Category	Root Cause
1	Positional Error of Selected Assets	Geometric accuracy	Workflow
2	Positional Error of Entire Dataset	Geometric accuracy	Workflow
3	Twisted Geometries	Geometric accuracy	Workflow
4	Scaling Error	Geometric accuracy	Workflow
5	Z-Fighting*	Geometric accuracy	Surface Overlay
6	Kind Problem	Lineage	Flawed Conversion
7	0-Byte-, Versioned File	Lineage	Workflow
8	Unresolvable Origin Path	Lineage	Flawed Conversion
9	Empty File	Completeness	No Source System Standards
10	Missing Geometries	Completeness	Workflow No Source System Standards
11	Additional Geometries	Completeness	Workflow
12	Redundant Geometries	Completeness	Workflow
13	Mesh Boolean Error	Semantic accuracy	No Source System Standards
14	Performance Error	Semantic accuracy	Workflow
15	Violation of Naming Convention	Semantic consistency	Workflow
16	Material Visualisation Deviations	Logical consistency	Flawed Conversion
17	Deviation From Source System	Logical consistency	Flawed Conversion
18	2D Data Within 3D Environment	Logical consistency	Workflow

*: Z-Fighting is a non-eliminable 3D-rendering phenomenon observable when the surfaces of two prims overlay at the exact same coordinates (e.g. floor and floor markings)

During these discussions, experts mentioned that not all of the listed insufficiencies would be identifiable by a software that scans the virtual factory model. Insufficiencies caused by a flawed conversion process as well as *Missing Geometries* are not automatically detectable using a software extension of the *Nvidia Omniverse*. Furthermore, it must be noted that *Z-Fighting* is not remediable as it can not be avoided when two layers exactly overlap. Nonetheless, all insufficiencies are included in the FMEA which is described in chapter 4.5.

4.5 Software Failure Modes and Effects Analysis

Based on the data input provided from the previously conducted interviews, a software FMEA is conducted to identify the most severe data quality insufficiencies occurring within the *Nvidia Omniverse*. For each data quality issue *i*, so failure mode, a respective *Risk Priority Number* (RPN) is calculated based on the following formula:

$$RPN_i = Severity_i * \ Occurrence_i * \ Detectability_i$$

The RPNs are computed using the severity, occurrence and detectability ratings provided by the interviewees. If multiple interviewees mentioned the same data quality issue, an average rating is calculated. The severity rating mostly describes the degree to which the particular data quality insufficiency has a negative effect on the usability of the software. The RPN calculation for the failure modes presented in Table 4.3 is provided in appendix A.7. Based on the work of Lago et al. (2012), an action priority matrix is used to visualise the severity of data quality issue in dependence of the product of their respective occurrence and detectability (see Figure 4.8). Thereby, those insufficiencies holding the highest urgency can be identified as those located within the matrix's top-right, red area.

Nr.	Data Quality Insufficiencies
1	Positional Error of Selected 3D Assets
2	Positional Error of Entire Dataset
3	Twisted Geometries
4	Scaling Error
5	Z-Fighting
6	Kind Problem
7	0-Byte-, Versioned File
8	Unresolvable Origin Path
9	Empty File
10	Missing Geometries
11	Additional Geometries
12	Redundant Geometries
13	Mesh Boolean Error
14	Performance Error
15	Violation of Naming Convention
16	Material Visualisation Deviations
17	Deviation From Source System
18	2D Data Within 3D Environment

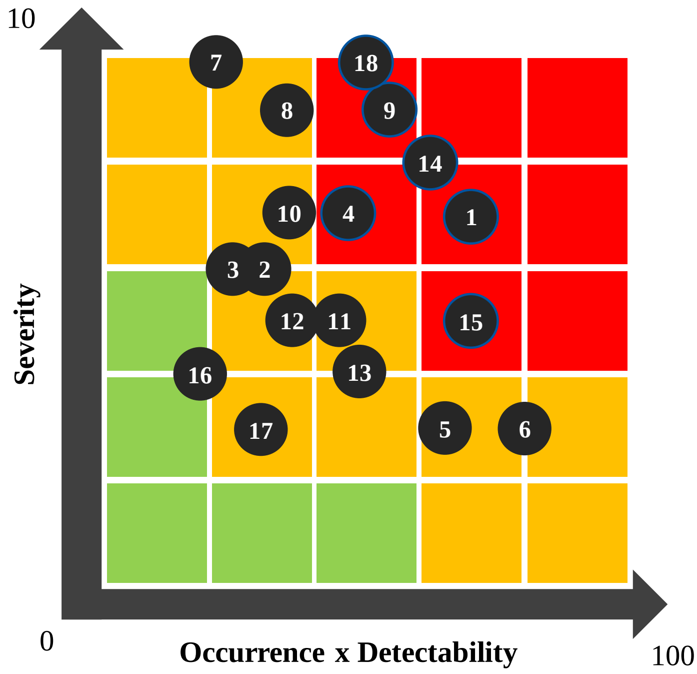


Figure 4.8: Action Priority Matrix

As can be seen in the action priority matrix, the data quality insufficiency types of highest urgency are: *Positional Error of Selected 3D Assets*, *Scaling Error*, *Empty File*, *Performance Error*, *Violation of Naming Convention* and *2D Data Within 3D Environment*. Accordingly, the software extension to be designed should incorporate for functionalities to identify these six data quality insufficiencies.

4.6 Artefact Objectives

From the interviews and the subsequent software FMEA, the most critical data quality insufficiencies were identified, prioritised and Q3 and Q4 thus answered. The problem to be solved is the absence of automatic data quality checks within the *Nvidia Omniverse* to make sure that the platform offers a reliable functionality and virtual factory data fulfills company-internal quality requirements. Accordingly, the objective of the software extension to be designed is to offer a functionality that identifies those insufficiencies that lay within the red area of the action priority matrix (see Figure 4.8). A possible software extension should identify the selected insufficiencies and extract a *comma-separated values* (CSV) file based on which the results of the automatic data quality check can be visualised and analysed, whereby KPIs are to be calculated. From these KPIs, minimum data quality requirements shall be definable to achieve that virtual factory models published for user access within the *Nvidia Omniverse* fulfill a certain standard necessary to assume reliability. For that purpose, the extension can be used to check for the fulfillment of these requirements and thereby serve as a quality gate for data to be visualised as a virtual factory in the industrial metaverse created by the *Nvidia Omniverse*.

In conclusion, the objective of the artefact is to point out existing insufficiencies and derive applicable KPIs to be able to define concrete data quality requirements. The extension is not intended to repair the identified insufficiencies, however, if virtual factory models built in the *Factory Assembler* fail to overcome the quality gate, concrete measures are proposed to achieve a repair of the identified insufficiencies and improve the virtual factory model's data quality. Figure 4.9 illustrates how the extension is integrated into the *Omniverse* architecture. Only if testing reveals that internal data quality requirements are met, virtual factory models pass the quality gate and can be published in the 3D environment.

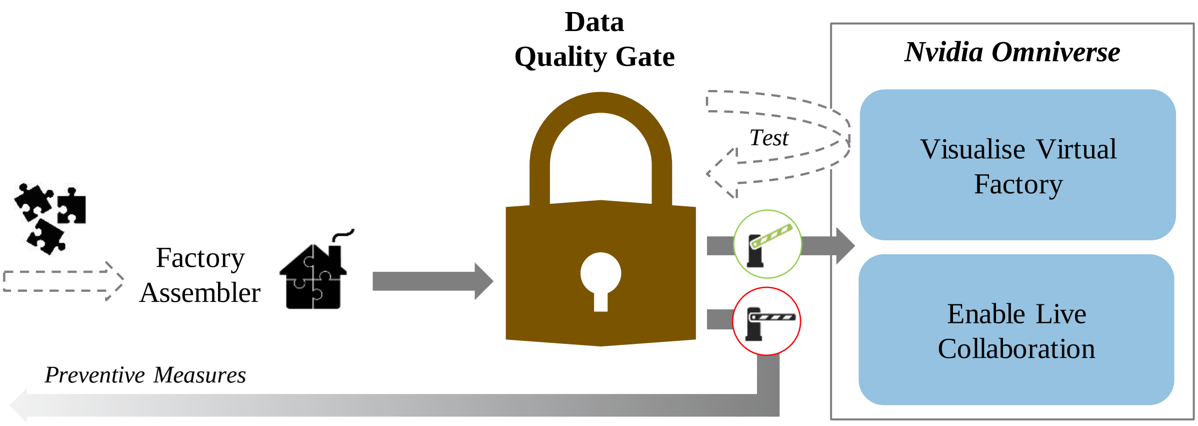


Figure 4.9: The Software Extension as a Virtual Factory Quality Gate

5 Design of a Data Quality Gate Software Extension

As the problem is properly defined and the key objectives to achieve in order to overcome this problem are narrowed down, the DSRM activity *Design & Development* is initiated (see Figure 5.1). This activity serves as the core of the thesis project. It incorporates for the software development of an extension to the existing *Nvidia Omniverse* platform that is capable of automatically scanning a USD stage and identifying selected data quality insufficiencies within the 3D environment in which a virtual factory model is visualised. Thereby, the determined research problem, the necessity to scan these virtual factory models manually, is tackled by a software extension that is able to point out those data quality insufficiencies whose type is of highest criticality and thereby priority (see Figure 4.8).

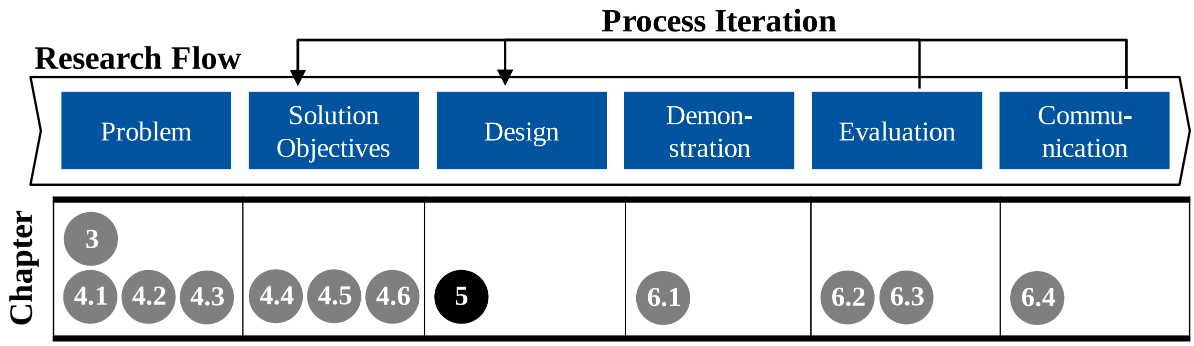


Figure 5.1: The DSRM Approach - Chapter 5

In the following sub-chapters, the empirical software engineering process to step-by-step develop functions that identify each of those insufficiencies on execution is outlined. Different approaches to achieve this identification are presented for each insufficiency and subjected to a sanity check based on their fulfillment of the three criteria: *Code Simplicity*, *Technical Feasibility* and *Minimum Implementation Effort*. The code should not only be functional, but as simple as possible to ensure future maintainability and performant execution. Furthermore, it must be technically feasible and implementable into the *Nvidia Omniverse* software with minimal efforts. Functionality of the presented approaches is implied and hence not included as a standalone sanity check criterion. In case functionality of the selected approach proves to be limited during testing, a reiteration process is induced. Finally, these scripts are integrated into one coherent software extension that fulfills the artefact objectives concretised in chapter 4.6. The deliverable of the software extension is the capability to perform a quantified analysis of any virtual factory data set in terms of the six insufficiencies found to be most critical (see chapter 4.5). Based on this analysis, measures to prevent these insufficiencies shall be derivable.

5.1 Identification of Data Quality Insufficiencies

The software to be designed functions as an extension to the *Nvidia Omniverse* software. The software allows users to create individual extensions, so plug-ins, that can be enabled and disabled by the user and add to the functionality of the *Nvidia Omniverse*. This feature is made use of for the purpose of the software extension to be designed. The *Nvidia Omniverse* contains different applications all of which offer the identical core functionality to visualise a 3D environment with each serving a slightly varying purpose. *Nvidia Omniverse Create* allows for advanced scene composition. Here, users can interactively navigate, modify, and render *Pixar* USD content in a synchronous, live environment. *Nvidia Omniverse Code* is an application tailored to the needs of developers with the intention to test newly created extensions. These two *Nvidia Omniverse* applications are used to develop and test the software extension created as part of this research project. The company-internal enterprise server of *GitHub* is used to continuously version the coding progress throughout the course of the research project. *Visual Studio Code* is used as an internal development environment to write the extension code in the programming language Python. The key library imported to develop within the *Nvidia Omniverse* software environment is *Pixar* USD, which incorporates for *application programming interfaces* (APIs) to collaboratively create, render and edit virtual models in a 3D environment. Originally written in C++, the USD API provides a Python API with bindings to nearly all methods and classes of the original C++ equivalent (Pixar Animation Studios, 2023). Due to its scalability, high resolution virtual models of reality, enhanced by the capabilities of simulations, BIM and digital twin technology, can unlock industrial metaverse applications such as a virtual factory. Furthermore, the *Nvidia*-developed omni.usd API serves a core purpose to develop this extension as it extends the USD API and is more tailored to the *Omniverse* software architecture (Nvidia, 2023b).

The versions of the applied software and programming language are to be found in Table 5.1. The functions to identify data quality insufficiencies, whose development is described in the coming sub-chapters, are all included in the class `DQCheck` of the extension file `quality_checks.py`. Appendix A.8 specifies the process of adding an extension to the *Nvidia Omniverse*, an explanation of the extension folder and file structure as well as a guide on how to use the extension.

Table 5.1: Software and Programming Language of Use

Software/Programming Language	Version
Python	3.7
GitHub Enterprise Server 3.7.8	3.7.8
Nvidia Omniverse Create	2022.3.3
Nvidia Omniverse Code	2022.3.1

5.1.1 Identification of Positional Deviations of Selected Prims

The first data quality insufficiency to identify describes the phenomenon of falsely positioned 3D assets. According to the spatial data quality criteria provided in chapter 3.3.2, this holds as a quantitative spatial error of geometric accuracy. The error describes a deviation between the position held by prims in the virtual factory model compared to the actual position of the visualised asset in the real factory as it exists or is supposed to be built. These geometric deviations can arise in all three dimensions, however, as factory planners have always been used to generating factory layouts solely in 2D, deviations occur most frequently on the z-axis. The error is a result of engineers providing real factory data in the source systems who forget to incorporate or make mistakes when assigning z-coordinates to all factory assets. As a consequence, it appears like the corresponding prims of certain assets are missing from or simply not existing in a factory when navigating through the 3D *Nvidia Omniverse* environment of the virtual factory model. Due to the asset's z-coordinate being flawed and therefore deviant in the virtual model, the particular prim, for example a forklift, is undetectable when solely looking at the virtual factory model, because it might be located several hundred metres above or below its correct z-position in the *Nvidia Omniverse*. The exemplary Figure 5.2 is extracted from a publicly available virtual factory model provided by *Nvidia* for demonstrative purposes (Nvidia, 2022).

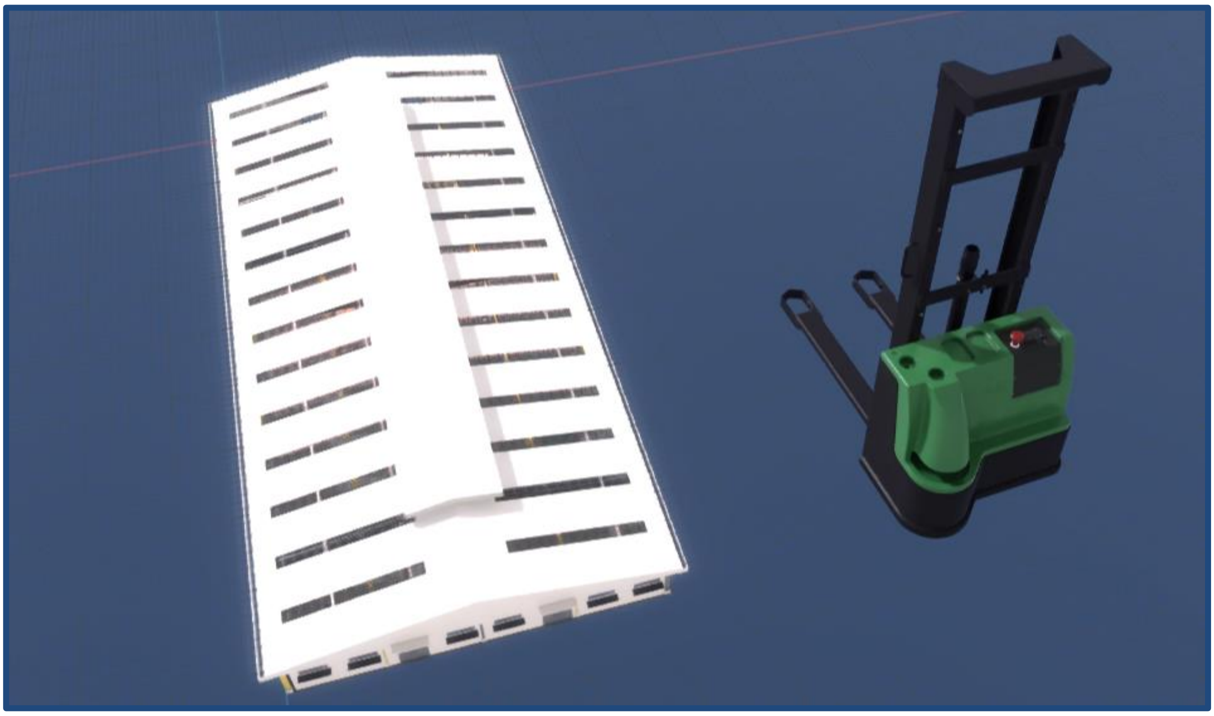


Figure 5.2: Positional Deviation of a Forklift in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022

At worst, factory planners that solely work within the virtual factory and are unaware of any real factory specifications assume a free spot at the misplaced asset's actual position which they then consider to be available for other purposes and reoccupy alternatively.

There are three possible approaches to identify such mislocated prims: Firstly, a comparison of the real position of each asset to the position its corresponding prim holds in the virtual factory might be conceivable. Thereby, deviations between real and virtual coordinates could easily be highlighted. Secondly, to point out prims that are located above or below the actual factory building and therefore must obviously be misplaced, the factory extents can be determined and defined as an upper and lower bound to set an interval outside of which no other prim is supposed to be located. Finally, it might be feasible to identify prims that are of similar shape or whose corresponding asset serves a comparable purpose, for example all robots, to then check and compare their z-positions. The z-coordinates of all assets allocated to the same purpose are expected to differ only minimally. Outliers from this assumption would then be assumed to be positional deviations between the real and virtual factory model. One approach is chosen based on the results from a sanity check provided in Table 5.2.

Table 5.2: Sanity Check of Approaches to Identify Positional Deviations

	Code Simplicity	Technical Feasibility	Minimal Implementation Effort
Comparison real vs. virtual coordinates	○	◐	○
Identify prims above roof or below ground	●	◐	●
Position comparison of similar prims	◐	○	◐

Legend: ○ Not fulfilled ◐ Partially fulfilled ● Completely fulfilled

A coordinate comparison of all prims with their real factory counterparts is likely to be most accurate and reliable. However, this procedure would demand for a reference list of all factory asset coordinates to be accessible from within the *Nvidia Omniverse* software environment. Even though this might be technically feasible, the effort to implement such a solution would be associated with an unreasonably high workload. A position comparison of all similar prims would demand for prims to be comparable by shape using a Python API. This, however, is only partially technically feasible with the API provided by *Nvidia*. Consequently, the attempt to automatically identify such positional deviations features a software extension that scans through all prims within a virtual factory model and identifies prims belonging to the roof as well as the ground. Each function dedicated at identifying data quality insufficiencies of different kinds, in this case positional deviations, receives the parameter `stage` from the user interface (see chapter 5.2). The parameter contains the path of that USD stage, so virtual factory model, a user wishes to check for insufficiencies. Using the command `TraverseAll()` from the USD API, all prims within the prim tree of this virtual factory are traversed and saved as the forward-iterable range parameter `prims`. As a prim tree features the type of factory asset in its corresponding prim's name, the names of all prims are obtained through the USD API command `GetName()`. Then, these names are searched for the terms for roof and ground in the observed manufacturer's native language. Subsequently, variables to represent the minimum and maximum z-coordinates of the USD stage are declared and the z-coordinates of the ground prim respectively roof prim assigned to these variables. The z-coordinates can only be determined for Xform prims (see chapter 4.1) by using the command `Get-LocalTransformations()`. The excerpt of the code to identify the roof prim and assign this prim's z-coordinate to the `max_z` variable is to be seen in Source Code 5.1.

```
'''
Identify roof prim which is considered to be prim with maximum z-coordinate within
virtual factory model.
'''
def position_check(stage):
   prims = stage.TraverseAll()
   for prim in prims:
      if UsdGeom.Xformable(prim):
         if "Dach" in prim.GetName():
            max_z = -float('inf')
            transformations = UsdGeom.Xformable(prim).GetLocalTransformation()
            z = transformations[3][2]
            if z > max_z:
               max_z = z
```

Source Code 5.1: Identification of Roof Prim

Then, it is checked whether there are any other prims located more than three metres above the roof or below the ground. Even though there occasionally are objects positioned slightly above or below the factory architecture, such as water or electricity lines, a tolerance range of three metres is, in coordination with the technical experts interviewed in chapter 4.4, considered to be sufficient to incorporate for such exceptions. As a result, all prims outside of this range are assumed to be falsely positioned. The software code to firstly analyse the z-coordinates of all prims within the USD stage and then point out those prims located outside of the range defined by the roof and ground prim +/- three metres is provided in Source Code 5.2 at the example of the roof prim. In case there are any prims located outside of the predefined range, the unique software-internal paths of the misplaced prims, determined by means of the command `GetPath()`, are, together with these prims' z-coordinates, added to the previously defined dictionary `position_data`.

```
    # Get position of remaining prims in virtual factory model
    prims = stage.TraverseAll()
    for remaining_prims in prims:
        if UsdGeom.Xformable(remaining_prims):
            path = remaining_prims.GetPath()
            position = UsdGeom.Xformable(remaining_prims).GetLocalTransformation()
            # Check if remaining prims are located >= 3 metres above roof prim
            if position[3][2] >= max_z + 3000.0:
                position_data.append({'Prim': path, 'Z': position})
```

Source Code 5.2: Identification of Z-Outliers Outside Factory Building Thresholds

This dictionary is finally exported as a CSV file to a dedicated *Omniverse* folder on the user's local drive which is specified u4nder the parameter `outputpath`.

5.1.2 Identification of Scaling Deviations

Similar to positional errors of selected prims, scaling insufficiencies are mostly caused by human mistakes of those responsible to provide and handle data in the source systems. Scaling deviations describe the occurrence of one or more prims in the virtual factory whose size significantly exceeds or falls below the size of its corresponding factory asset in reality. Figure 5.3 shows a forklift whose scaling visibly exceeds realistic dimensions.

Figure 5.3: Scaling Deviation of a Forklift in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022

A software extension could identify such falsely scaled prims by taking a statistical approach and computing the mean volume of all factory prims to check for prim volumes that exceed a certain upper confidence interval threshold from this mean. This would indicate a scaling deviation between the prim and its real factory counterpart. Moreover, the size of each prim in the *Nvidia Omniverse* 3D environment might be comparable to the reference size of its corresponding asset to spot deviations. Additionally, an extension could scan through all prims, and check for a match between the volumes of all prims with the exact same shape. Thus, for instance, all bolts, forklifts or robots of the same type could be compared and outliers pointed out. The three approaches are evaluated as part of Table 5.3.

Table 5.3: Sanity Check of Approaches to Identify Scaling Deviations

	Code Simplicity	Technical Feasibility	Minimal Implementation Effort
Check for mean prim volume outliers	◐	●	●
Comparison of prim sizes to reference sizes	○	◐	○

Size comparison of similar-shaped prims	◐	○	◐

Legend: ○ Not fulfilled ◐ Partially fulfilled ● Completely fulfilled

Analogous to the sanity check performed in chapter 5.1.1, a comparison of prims with reference values from reality might be feasible yet complex and an identification of similar-shaped prims not technically feasible. Hence, the first approach, a check for outliers from the mean prim volume, is pursued (see Source Code 5.3).

```
# Compute volume of mesh bounding box
boxcache = UsdGeom.BBoxCache(Usd.TimeCode.Default(),
                             includedPurposes=[UsdGeom.Tokens.default_],
                             useExtentsHint=True)
prims = stage.TraverseAll()
for prim in prims:
    if UsdGeom.Mesh(prim):
        path = prim.GetPath()
        box = boxcache.ComputeWorldBound(prim)
        volume = int(box.GetVolume())
        scaling_data.append({'Prim': path, 'Volume': volume})
```

Source Code 5.3: Computation of Bounding Box Volume

It is hardly possible to determine the exact volume of each prim using the Python API in the *Nvidia Omniverse* as the current versions of neither the USD nor the omni.usd API contain a command to compute a prim's exact shape. Instead, the bounding box to each mesh prim is determined to subsequently calculate the volume of this box. A prim's bounding box is a cuboidal block that matches the extents of the prim and thereby fully contains it. As the software must reiterate through all prims within a USD stage to calculate each prim's bounding box, a bounding box cache is created using the command `UsdGeom.BBoxCache()` to handle and store these computations efficiently. A specific time as well as the purposes of the examined prims whose bounding boxes are to be computed are not specified, because the time component is not relevant at the current degree of maturity of the virtual factory models and all prims are to be included. Declaring the `useExtentsHint` as `True` leads to a bounding box computation for a prim only if this prim is visible. Thereby, only those prims are included into the computation that are visible to a user. Only polygonal mesh prims are complex enough to hold the geometric attributes necessary to determine the bounding box. Consequently, only mesh prims are examined. The bounding box for each prim is calculated by applying the command `ComputeWorldPrim()` using the previously defined cache. Subsequently, the volume of this bounding box is computed as an integer value through the command `int(box.GetVolume())` for each prim to then be added to a dictionary together with the prim's path.

From the entire sample of prim volumes, a mean and standard deviation are calculated using the commands `statistics.mean()` and `statistics.stdev()`. While developing the software extension piece, it became obvious that the standard deviation of the prim volumes is usually comparatively small. This is due to the fact that the number of total prims, so sample size, for an exhaustive factory model is high ($n > 1000$) and the majority of prims is similarly sized. To not mistakenly exclude large volumes of virtual factory prims from the confidence interval that are however not falsely scaled, a one-sided 99.99% confidence interval is considered as sufficient to make sure that only wrongly scaled outliers are within the area of rejection. Hence, as depicted in Source Code 5.4, scaling deviations are identified by firstly calculating a one-sided 99.99% confidence interval around the mean volume which is achieved by adding the sample standard deviation multiplied by the factor 3.5 to the sample mean (Miller & Childers, 2004).

```
# Compute average volume of all meshes and one-sided 99.99% confidence interval
volume_values = [item['Volume'] for item in scaling_data]
std_volume = statistics.stdev(volume_values)
mean_volume = statistics.mean(volume_values)
v_upper = mean_volume + 3.5 * (std_volume)
```

Source Code 5.4: Computation of Mean Bounding Box Volume Confidence Interval

Volumes from the sample that fall into the area of rejection are identified and a corresponding boolean value added to the dictionary (see Source Code 5.5).

```
# Error message for prims exceeding average volume by factor 3.5 (or more)
for item in scaling_data:
            if item['Volume'] >= v_upper:
                item['Scaling'] = False
            else:
                item['Scaling'] = True
```

Source Code 5.5: Identification of Scaling Deviations

This dictionary is finally exported as a CSV file for further data analysis.

5.1.3 Identification of Two-Dimensional Data

The *Nvidia Omniverse* is designed to allow for the creation of 3D workflows in a virtual environment. Users in an automotive manufacturing context must be able to rely on the accurate 3D visualisation of factories to allow for a more efficient layout planning process. Nonetheless, occasionally, 2D objects that only span on the x-axis and y-axis are converted from source systems. This insufficiency is due to workflow errors, as experienced engineers are used to plan a factory's layout using 2D tools that do not require a z-coordinate assignment to factory assets. As a consequence to this habit, the assignment of a z-coordinate to every factory asset is sometimes overlooked causing their corresponding prims in the virtual factory to be visualised in 2D rather than 3D. Thereby, these prims might be overlooked by factory planners unaware of the real factory asset's existence in reality. Thus, 2D prims need to be identifiable to be able to repair the source system data. In Figure 5.4, the falsely scaled forklift from the previous sub-chapter (see chapter 5.1.2) is to be seen in 2D as an example for this insufficiency type.

Figure 5.4: 2D Forklift in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022

Different approaches are imaginable to identify 2D data within the 3D *Omniverse* environment: Firstly, 2D prims might be identifiable by their attributes. Each prim holds selected attributes which could be extracted using a software extension. This approach assumes that there are definite attributes based on which 2D prims can be distinguished from 3D prims. Secondly, the geometry and shape of each prim could be analysed to determine all prims that are level and do only span on the x-axis and y-axis.

Table 5.4: Sanity Check of Approaches to Identify 2D Data

	Code Simplicity	Technical Feasibility	Minimal Implementation Effort
Check prim attributes for 2D indicators	●	○	◐
Check prim shape for 2D geometry	●	●	●

Legend: ○ Not fulfilled ◐ Partially fulfilled ● Completely fulfilled

All approaches would be executable using a relatively simple and maintainable code. The implementation effort would be higher with approach one as this procedure would demand for an initial identification of prim attributes that indicate two-dimensionality. Nonetheless, such indicators do not coherently exist within all prims of a USD stage and are therefore not reliable and technically feasible. Thus, the second approach, a check for prims with non-existent z-extents, is followed.

To identify those prims that only span on the xy-plane and thereby are 2D, the extents of each prim on the z-axis must be computable to then highlight all prims whose extents in z-direction equal zero. To compute the extents of any given prim within the *Nvidia Omniverse* environment, USD offers the functionality to determine the bounding box for a prim as well as the children it may contain. As previously established, a prim's bounding box matches the prim extents. Thus, a calculation of the bounding box extents in z-direction corresponds to a calculation of the related prim's z-extents. For each mesh, the bounding box, a 3D floating point range, is computed. The range minima and maxima are added to the list `max_extents` (see Source Code 5.6).

```
# Compute bounding box of every prim
box = boxcache.ComputeWorldBound(prim)
b_range = box.GetRange()
# Compute extents of bounding box
max_extents = [*b_range.GetMin(), *b_range.GetMax()]
# Compute z extents (Assumption: x,y,z-coordinate system)
height = round(float(max_extents[5] - max_extents[2]), 1)
```

Source Code 5.6: Computation of Bounding Box Extents

From this list, the extents in z-direction are extracted as a float value, rounded to one decimal and subtracted from each other to arrive at the mesh prim's extents on the z-axis. Indices two and five of the list `max_extents` represent the minimum and maximum z-extent values of a prim. It must be noted that a coordinate system whose z-axis points upwards is assumed to be the prevalent coordinate system for all virtual factory models. The prim path as well as the affiliated z-extents are added to a dictionary. This dictionary is then exported as a CSV file. From this CSV file, those prims with z-extents equal to 0.0 are identifiable in chapter 5.3.

5.1.4 Identification of Performance-Inhibiting Data

A data quality insufficiency that hampers the usability of the software unrelated to deviations between real and virtual factory data is detectable within so-called performance-inhibiting data. Experts mentioned that in selected virtual factory models, a significant downgrade in the *frames per second* (FPS) rate is observable within the 3D environment. However, up until now, this performance decline could not be allocated to specific prims. While it is assumed to be a workflow, so man-made issue, a pattern as to what the insufficiency is specifically caused by could not be made out. It is therefore crucial to initially find the reason for the FPS drop within certain data sets before being able to identify the insufficiency automatically within a virtual factory model.

This can be achieved by pursuing different approaches depending on what causes the FPS rate to decline: The performance impact of each individual prim could be measured using Python profiling tools such as modules to compute the time to render each prim. Furthermore, it could be checked whether the render settings for selected prims differ from those of the remaining prims in a way that causes the FPS to go down. Lastly, it must be checked whether certain parts of the CAD data to describe the factory architecture or production facility geometries, which is supplied from source systems, have been converted into a too complex virtual model with an unreasonably high number of prims and vertices. It is expected that such a too complex virtual factory model causes the visualisation performance to decline. Based on several tests with different data sets, the following table could be filled out to hold as a sanity check for the function to be coded to identify performance-inhibiting data (see Table 5.5).

Table 5.5: Sanity Check of Approaches to Identify Performance-Inhibiting Data

	Code Simplicity	Technical Feasibility	Minimal Implementation Effort
Compute render time of each prim	◐	○	●
Check for render setting differences	◐	○	◐
Check for total number of prims and vertices	●	●	●

Legend: ○ Not fulfilled ◐ Partially fulfilled ● Completely fulfilled

The computation of the rendering time of each prim is hardly doable with Python modules accessible from the *Nvidia Omniverse* software environment and would not be technically feasible as this individual calculation for each prim is expected to delay the code execution at the time of initial scene loading. A check for render setting differences is not technically feasible as these are globally defined for a USD stage and do not vary among different prims. Furthermore, render settings with an influence on the FPS rate would firstly need to be identified causing the implementation effort to be moderately high. After several manual checks and a knowledge exchange with the interviewed technical experts, it can be concluded that the rendering performance expressed by the FPS is mainly influenced by the total number of mesh prims within a virtual factory data set. The relationship between the total number of meshes in a virtual factory and the FPS when visualising this factory in the *Nvidia Omniverse* 3D environment appears to be linearly inversely proportional (see Figure 5.8). Hence, the extension should allow for a functionality to identify the number of total mesh prims to later be able to define an upper bound virtual factory models should not exceed to pass the quality gate and be eligible for visualisation and publication to multiple users. As the number of vertices of a prim also minorly affects the speed in which this prim can be rendered, the vertice count of each prim is also to be determined.

Analogous to the previous bounding box computation (see chapters 5.1.2 and 5.1.3), the calculation of geometric attributes of each prim is only feasible with mesh prims. A mesh counter is initiated to count the number of scanned meshes and thereby determine their total amount. Simultaneously, the previously defined function `get_vertice_count()` aimed at calculating a mesh prim's number of vertices is called. Core to this function is the command `GetFaceVertexCountsAttr()` that provides the number of vertices in each face of a mesh. Source Code 5.7 shows the core functionality of the `performance_check(stage)` function.

```
meshes = 0
# Get vertice count of all meshes and add up total number using mesh counter
prims = stage.TraverseAll()
for prim in prims:
    if UsdGeom.Mesh(prim):
        vertice_count = get_vertice_count(prim)
        path = prim.GetPath()
        meshes += 1
        # Add number and vertice count of valid meshes to list
        if vertice_count != 0:
            performance_list.append({'Prim': path, 'Vertices': vertice_count,
                                       'Number': meshes})
```

Source Code 5.7: Computation of Vertice & Mesh Count

This count, the assigned mesh counter and the path of each scanned prim is then added to a dictionary and exported in the form of a CSV file.

5.1.5 Identification of Empty Files

Occasionally, source systems supply input files for conversion that do not contain any content. There are no defined standards that data outputs from the source system need to fulfill in order to be passed on for conversion to USD. As a result, such empty files are not recognisable up until visualisation of the virtual factory where they appear as invisible prims.

The usually pursued approach to identify such files within the *Nvidia Omniverse* would be the scan of all individual prim units and identification of those prims with a size equal to zero bytes. Another coding strategy could involve the check of all prims for children or non-metadata properties. A prim with neither children nor any properties that are not metadata can be assumed to be a prim without content. However, during the course of this research project, the conversion process was adjusted by the product owner, so that it still converts empty files, but now adds a small text label that is visualised as a placeholder for the empty prim in the 3D environment. Furthermore, the converter changes the name of each empty file appearing as a prim in the *Nvidia Omniverse* to *EMPTY_FILE* to make it more easily detectable. As one virtual factory model can contain more than 100,000 prims, those empty prims remain to be moderately hard to spot within a data set and thus still represent a severe data quality insufficiency worth to be identified through this software extension. The approaches to identify empty prim units are presented and compared in Table 5.6.

Table 5.6: Sanity Check of Approaches to Identify Empty Files

	Code Simplicity	Technical Feasibility	Minimal Implementation Effort
Check for prims that use zero storage	◐	◐	●
Check prim children and properties	◐	◐	◐
Check for empty-file-indicating prim name	●	●	●

Legend: ○ Not fulfilled ◐ Partially fulfilled ● Completely fulfilled

All three approaches are pursuable, yet the simple check of prim names fulfills the sanity check criteria to the highest extent. Due to the explicit naming of empty prims, the extension is designed to scan through all prim names and thereby identify the prims whose name contains *EMPTY_FILE* as an invalid set of characters. This is achieved by firstly defining the invalid set of characters used to relabel empty files in the conversion process. Then, `re.compile()` is applied to compile this set of characters into a pattern object that can subsequently be used to search for a match between the invalid characters and all prim names using `re.search()` (see Source Code 5.8). This operation could also be achieved using a string comparison, however, the chosen approach proved to be more performant.

```
empty_file_ident = 'EMPTY_FILE'
invalidCharactersRegex = re.compile(empty_file_ident)
# Check prim names for the defined characters and add empty prim paths to dict.
    prims = stage.TraverseAll()
    for prim in prims:
        name = prim.GetName()
        path = prim.GetPath()
        if re.search(invalidCharactersRegex, name):
            empty_files_data.append({'Prim': path})
```

Source Code 5.8: Empty File Identification

In case matches are identified, the paths of these prims are added to a dictionary and subsequently exported in CSV form.

5.1.6 Identification of Naming Convention Violations

The insufficiency to be identified by the function outlined in this chapter refers to the violation of a naming convention that relates to the *Factory Assembler*. In the *Factory Assembler*, the converted outputs of the source systems that have been integrated on the *Nvidia Nucleus* are put together. Thereby, a coherent virtual factory model can be visualised in the 3D environment of the *Nvidia Omniverse*. The build-up-algorithm of the *Factory Assembler* utilises a functionality that categorises all factory assets.

Thereby, each asset to be visualised as a prim is assigned to a building, a segment of the production line within this building, a particular production station and lastly, a machine or larger object this prim is a part of in the factory as it is built or planned. This categorisation is already manifested as part of each file's name handled in the *Factory Assembler* and then documented and structured in the *Nvidia Omniverse* prim tree (see Figure 4.6). Due to this company-internal naming convention, each asset's later purpose in the virtual factory is already obtainable from the source system data and converted from there to the *Nucleus* before passing the *Factory Assembler* to be visualised in the 3D *Nvidia Omniverse* environment. In case this naming convention is violated and not in alignment with a prim's position in the factory as it is categorised in the *Factory Assembler*, the flawed prim names appear at the correct position in the *Nvidia Omniverse* prim tree. However, the software only visualises prims whose name aligns with that of its parent sub-folder indicating the actual segment this prim is assigned to in the virtual factory model. Accordingly, all prims violating the naming convention are not visible in the 3D environment.

There are two possible approaches to check for violations of the prim naming convention: Firstly, a software extension could scan all prim names in the virtual factory and compare the assigned positions from the *Factory Assembler* with each prim's name to check for accordance with the naming convention. Secondly, it is imaginable to compare prim names to their individual prim path. As the prim path contains the sub-folder name this prim is subordinated under and the sub-folder name gives away the correct categorisation of each subordinated prim, such a comparison is expected to uncover naming convention violations. Table 5.7 illustrates a sanity check of both procedures.

Table 5.7: Sanity Check of Approaches to Identify Naming Convention Violations

	Code Simplicity	Technical Feasibility	Minimal Implementation Effort
Comparison of name and assigned position	○	○	○
Comparison of name and path	●	◐	●

Legend: ○ Not fulfilled ◐ Partially fulfilled ● Completely fulfilled

It is clearly concludable that a comparison of the prim name and its assigned position is hardly simple nor technically feasible. Even if the reference position of each prim were obtainable, a prim's correct coordinates would not automatically reveal the prim's categorisation in the prim tree naming hierarchy. If this was to be achievable, the implementation effort would be immense as the range of coordinates of each factory segment would need to be defined first to then determine the correct segment for each prim based on its coordinates. Thus, prim names and paths are to be compared to identify violations of the naming convention.

The software extension function to identify prims violating the naming convention adds those prims whose name decisively differs from the path of its superordinated sub-folder to a list. To extract these prims, the name of every prim as well as the path indicating the prim's position in the prim tree are split up into their respective parts. To differentiate these parts, the name has to be split at each slash-symbol, while the path has to be split at each underscore-symbol. The condition `IsLoaded()` ensures that only those nodes of the prim tree are checked, whose data is supplied from the *Factory Assembler*, which can be assumed to be the whole prim tree in case of a functioning process as depicted in Figure 4.5. In order to be in coherence with the naming convention, the first and second component of the prim name have to equal the path's fifth and sixth component. The first component of the prim name and the fifth component of the path contain information on whether the prim represents an asset from the factory architecture or interior production facilities. The second name component and sixth path component specify the factory segment. In case one of these two conditions is not met, a naming violation is occurrent in the prim's name and the individual prim path as well as a corresponding boolean value are added to a dictionary that is subsequently exported to CSV (see Source Code 5.9).

```
# Split prim names and check for accordance with the naming convention
if not prim.IsLoaded():
    name = prim.GetName()
    path = prim.GetPath().pathString
    path_parts = path.split('/')
    path_parts = [part for part in path_parts if part != '']
    name_parts = name.split('_')
    # Check for naming convention violation
    if name_parts[0] != path_parts[4] and name_parts[5] != path_parts[4]:
        violation_check = False
    else:
        violation_check = True
    naming_data.append({'Prim': path, 'Naming Convention':
                         violation_check})
```

Source Code 5.9: Check for Violation of Naming Convention

5.2 User Interface

As previously mentioned, the roll-out of the *Nvidia Omniverse* software at the automotive OEM of observation is still in its early stages during the scope of this research project. Hence, the current number of users is limited. At this stage, it is crucial for the software to be as easy to handle for new users as possible to increase user acceptance and ultimately transform first-time users into adopters. Usability of the software and all its extensions is therefore key. A clearly arranged, understandable user interface ensures this usability as it is the single point of interaction between users and the software and mainly determines the users' impression.

The data quality gate extension designed as part of this thesis is activatable from within the *Nvidia Omniverse*. Upon activation, the extension user interface pops up in the bottom right corner of the screen to catch the eye of the user and allow for immediate use yet not cause user annoyance. The appearance of the user interface is presented in Figure 5.5.

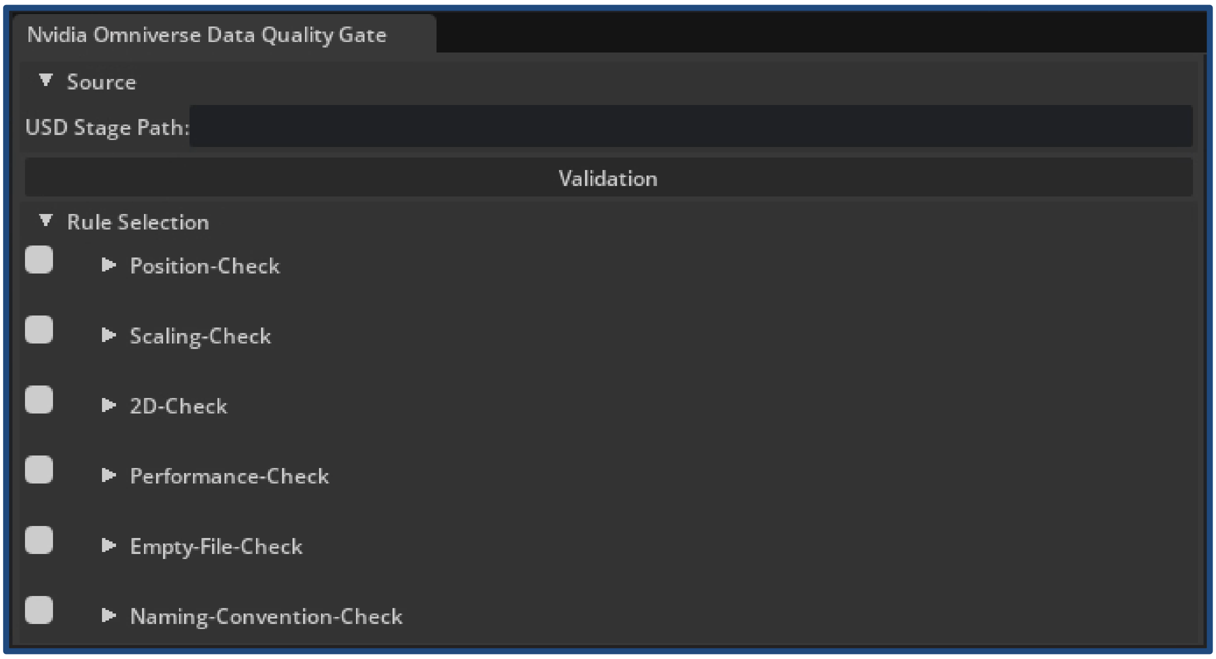


Figure 5.5: The Data Quality Gate User Interface

The widget is labelled as *Nvidia Omniverse Data Quality Gate* to reveal the purpose of the extension to new users. At the top, the extension user interface features a collapsable frame labelled as *Source*. It contains an empty text field with the name *USD Stage Path* in which the path to the USD stage to scan through, so the *Uniform Resource Locator* (URL) link of the virtual factory to examine, is to be inserted. This is easily achievable as the *Nvidia Omniverse* allows for uncomplicated copying of software-internal file URLs. Once a user has inserted the URL of the virtual factory model to assess regarding its data quality, the interface offers the option to select which data quality insufficiencies to check for by activating the respective checkboxes. It might be the case that a user already assumes the occurrence of a particular insufficiency in a virtual factory model and is therefore solely interested in identifying the degree to which this insufficiency, regardless of any other insufficiencies, appears in the model. Furthermore, the selection of only a share of functions reduces code execution time and further increases user friendliness and acceptance. At the top of the window, a click on the *Validation*-button initiates the execution of those functions selected before. The exported CSV files are written and saved in a dedicated *Omniverse* folder on the user's local drive. To run the extension, the virtual factory model does not have to be opened for visualisation in the *Nvidia Omniverse*. This represents a significant technical advantage of the extension as it thus could be automatically run in the backend. This feature is further touched upon as part of chapter 7.4.

5.3 Data Analysis

The automatic identification of those data quality insufficiencies evaluated as most critical is the first of two key functionalities of the software extension. The second functionality is the ability to analyse the extracted CSV files to derive quantified KPIs based on which minimum data quality requirements are definable that need to be met by data to allow for publication of a virtual factory model to the entirety of users. This data analysis functionality of the extension is key for it to be capable of acting as a quality gate between *Factory Assembler* and *Nvidia Omniverse*, an objective formulated in chapter 4.6.

5.3.1 KPI Definition

In coordination with the product owner, members of the functional department responsible for the company-internal *Nvidia Omniverse* software roll-out, the data analysis leads to KPIs whose form differs for every type of insufficiency. These KPIs are calculated automatically for each insufficiency depending on whether the respective check has been initiated and a CSV file has been exported before. The KPIs obtained from each data analysis are then displayed to the user in a lucid dashboard that is part of the software extension interface. The developed user interface of the extension displays the KPIs derived from each data analysis in a collapsable frame underneath the checkbox to select what insufficiency to check for.

1. Following from the first check for positional deviations between real and virtual factory models, the percentage of prims located at a position outside of the z-coordinate thresholds set by the factory ground and roof coordinates is calculated. Thereby, users can immediately determine whether there is a possibility for factory layout planning errors due to positional deviations of selected prims that are mislocated in the virtual factory. If the percentage of positional deviations on the z-axis is unequal to zero, the corresponding CSV file should be checked for those prims causing the error message and appropriate measures should be initiated. These measures are detailed in chapter 6.2.
2. The second insufficiency, scaling deviations, are identified by highlighting prims whose volume does not fall inside a 99.99% confidence interval from the mean volume. The KPI to this error is expressed in the form of a percentage value of these assumed scaling errors within the analysed entire virtual factory model.
3. Analogous, the results of the check for 2D prims are also presented as a percentage-based share of 2D prims.

KPIs whose unit is a percentage are illustrated in the form of a process bar. Figure 5.6 representatively shows the bars concluded from an exemplary check for positional and scaling deviations as well as 2D prims. In the analysed model, a total of 15.36% of all prims in the 3D environment only span in x- and y-direction, 0.38% are located outside the factory building z-coordinate thresholds and 0.37% exceed the mean volume confidence interval.

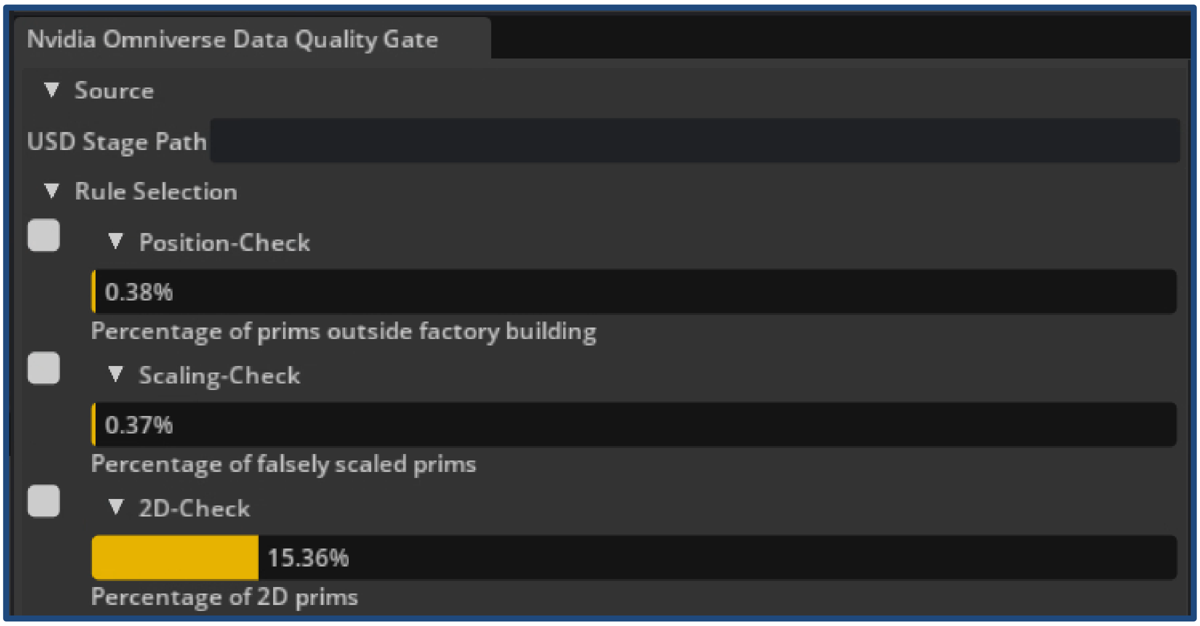


Figure 5.6: KPI Dashboard - Part I

4. Performance insufficiencies and a general decline of the FPS rate within a virtual factory model were found to be traceable to the total number of mesh type prims within a USD stage. Hence, this total count holds as the KPI following from the check for performance-inhibiting data.
5. The number of empty files within a USD stage is extracted and shown in the user interface in the form of an integer value as well.
6. Finally, naming convention violations are also documented as a total number in the dashboard.

Figure 5.7 shows the KPIs derived from a data quality check of a virtual factory model that contains 7767 prims, no empty prims and ten naming convention violations.

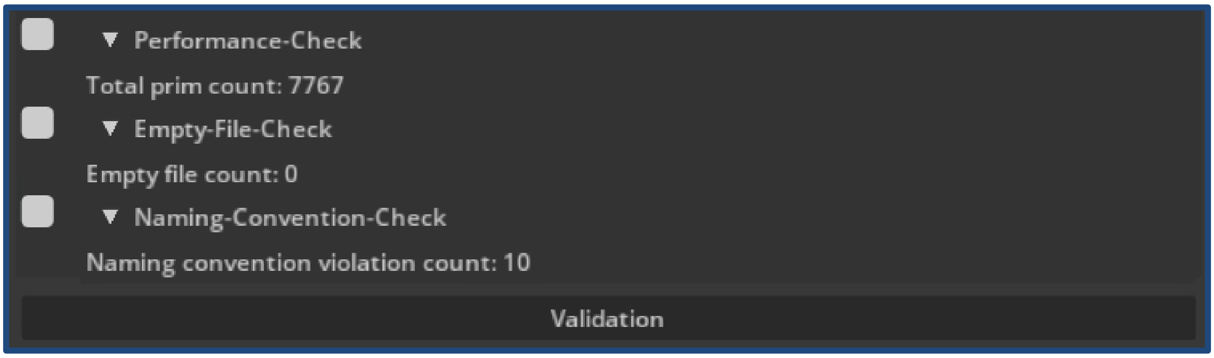


Figure 5.7: KPI Dashboard – Part II

5.3.2 Definition of Minimum Data Quality Standards

As the project remains to be at an early stage, the definition of data quality standards can only be based on the severity evaluation provided by technical experts and illustrated in Figure 4.8. However, experts mentioned that the occurrence of any data quality insufficiency should generally be avoided. Thus, if any of the checks other than the *Performance-Check* reveal the occurrence of insufficiencies, the data quality gate should not be passed. Regarding the *Performance-Check*, it became obvious from testing that an FPS rate of 40 holds as the minimum to still be able to smoothly work in the *Nvidia Omniverse* 3D environment. This rate is reached at a mesh number greater than 50,000. Hence, the mesh count of a virtual factory model must be lower than 50,000 in order to pass the data quality gate. The relationship between the FPS rate and the total number of prims is visualised in Figure 5.8.

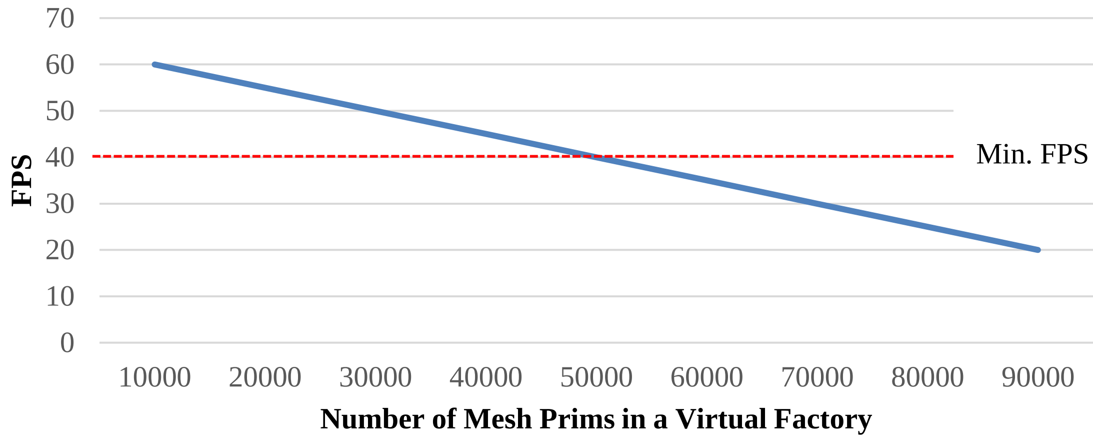


Figure 5.8: Approximated Relationship between FPS and Total Mesh Number

To allow for the future definition of a more flexible and unified minimum standard, an overall severity score is provided at the bottom of the user interface. The computation of this score is based on the severity score of each insufficiency and the total number of insufficiencies of that particular type that were identified when running the extension. The corresponding formular can be defined as:

$$Total\ Severity\ Score = \sum_{i=1}^{6} n_i s_i$$

n represents the total number of detected insufficiencies of each type i. This number is multiplied with the severity score s for that insufficiency. To obtain n_1, so the number of positional deviations, prims mislocated on the x-, y- and z-axis are summed up. This score of total severity can be used to define a maximum threshold that is not to be exceeded to pass the quality gate.

6 Prototype Usability Test and Evaluation

The following chapter contains a description of all actions undertaken as part of the DSRM activities *Demonstration*, *Evaluation* and *Communication* (see Figure 6.1). Thus, the process of testing the developed software extension is described. In case tests fail or flaws within the software extension are observable, a reiteration cycle is initiated to revise the extension. Afterwards, potential preventive measures to mitigate the future occurrence of the most critical data quality insufficiencies are evaluated before conducting a final pre-post analysis aimed at ascertaining whether the extension serves its purpose and fulfills those artefact objectives defined in chapter 4.6. Finally, the developed software extension is adequately communicated to involved stakeholders.

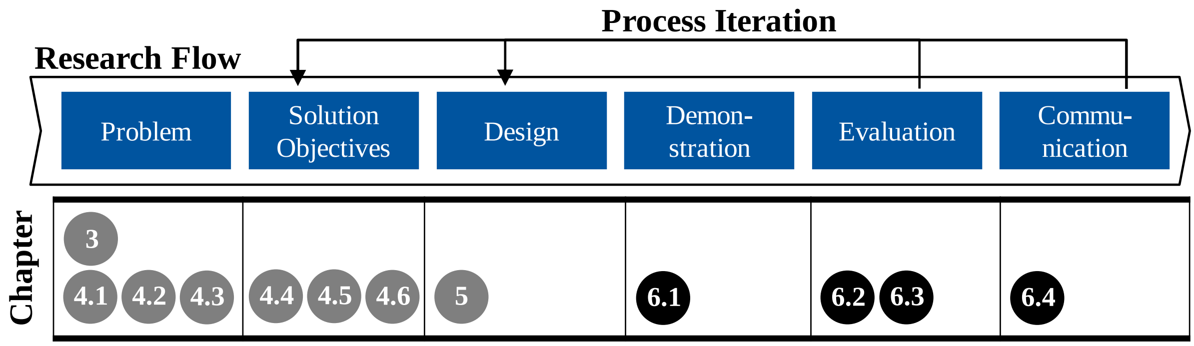


Figure 6.1: The DSRM Approach - Chapter 6

In order to test the functionality of the developed software extension, selected virtual factories are scanned by the software extension in the *Nvidia Omniverse* 3D environment. The data sets at which the extension is tested must deviate from the data set used to develop the extension to ensure that a coincidental functionality of the software in the developing stage of the extension can be excluded. Testing for the six insufficiencies is performed using eight sets of data. The eight data sets contain virtual factory models of production sites for which such models already exist. The testing data sets were manually found to each be containing data from at least one of the six most critical insufficiency types. It is furthermore provided that each insufficiency occurs in at least two of the data sets. Hence, it is made sure that the designed artefact fulfills the minimal generalisability requirements and is functionable with all virtual factory models. All test results are included in appendix A.9.

6.1 Software Extension Test

The first extension function is aimed at identifying all prims whose position deviates from that of its real factory counterpart asset on the z-axis. For that purpose, an extension was developed that identifies the roof and ground of a virtual factory model by scanning all prim names for the automotive manufacturer's native words for roof and ground. The z-coordinates of the thereby identified prims are determined and a z-interval of acceptance is defined that ranges from three metres below the factory ground up to three metres above the factory roof. All prims whose z-coordinates are not within this interval are declared as falsely positioned.

When testing the extension, it quickly became evident that a command that identifies the z-extents of a virtual factory model by searching for certain keywords among the prim names is highly prone to error as the manufacturer operates a number of international production sites in countries whose native language differs from the language spoken in the manufacturer's country of origin. Another limitation to this approach is that some production sites contain multiple roofs that are located on different heights. This leads to unreliable results when testing the extension as the approach considers the z-coordinate of the highest roof prim to be the upper threshold for the whole factory model. Accordingly, a reiteration cycle was initiated the outcomes of which are detailed in the following chapter.

6.1.1 Reiteration Cycle 1: Revision of Positional Deviation Identification

To revise the portion of the software extension aimed at identifying falsely positioned prims within a USD stage, a completely new approach is pursued:

To avoid misfunctioning of the extension due to language barriers or potential future prim relabelling, the reiterated extension piece's functionality should not be affected by the name of a prim. To define a z-interval that represents the acceptable area within which a prim's location is acceptable, a statistical approach is taken instead and a 99.99% confidence interval for the z-coordinates of all prims within one virtual factory is calculated. For that purpose, the z-coordinates of all prims within a USD stage are extracted and summarised in one dictionary that contains the prim's path and its corresponding z-coordinates. From this dictionary, the sample mean and the standard deviation for the z-coordinates of all prims are calculated. Then, the confidence interval is to be determined for which the upper and lower interval bound are to be calculated. The standard deviation is, similar to the approach pursued in chapter 5.1.2, multiplied by the factor 3.89 and subtracted from, respectively added to, the sample mean to calculate the lower and upper bound that make up the threshold values of the confidence interval (see Source Code 6.1). Here, the factor 3.89 is chosen as it is the statistically defined factor to obtain the upper and lower bound of a two-sided confidence interval, whereas in chapter 5.1.2, only a one-sided confidence interval was computed (Miller & Childers, 2004).

```
z_values = [x['Z'] for x in position_data]
z_mean = statistics.mean(z_values)
z_stddev = statistics.stdev(z_values)

# Compute 99.99% mean confidence interval
z_lower = z_mean - 3.89 * z_stddev
z_upper = z_mean + 3.89 * z_stddev
```

Source Code 6.1: Revised Extension: Z-Coordinate Confidence Interval

Using the lower and upper bound, a condition can be defined in Python that compares the z-coordinates recorded in the priorly composed dictionary to the lower and upper bound to check each prim whether it is located within the confidence interval of acceptance or not. Within this if-statement, the dictionary is extended by a third category that contains data for every prim indicating whether a prim's z-coordinate lies within the lower and upper bound and therefore inside the confidence interval or not. Source Code 6.2 shows the corresponding code excerpt to this particular operation.

```
for prim in position_data:
    if prim['Z'] < z_lower or prim['Z'] > z_upper:
        prim['Z-Axis Confidence Interval'] = False
    else:
        prim['Z-Axis Confidence Interval'] = True
```

Source Code 6.2: Revised Extension: Identification of Positional Deviations

Finally, the dictionary with all prim paths, their z-coordinates and the information whether a prim is located within the 99.9% confidence interval is exported in the form of a CSV file.

Analogous to this check for z-coordinate deviations, this revised code is extended to also identify falsely positioned prims on the x- and y-axis. The x- and y-coordinates as well as the information whether a prim belongs to the 99.99% x-coordinate mean confidence interval, respectively y-coordinate mean confidence interval, are added to the CSV file in the form of additional rows.

When testing the reiterated software extension in coordination with the key stakeholders, another suggestion for improvement came up: Even though the revised check for positional deviations is capable of identifying the percentage of prims that are located outside of the defined acceptance intervals on all three coordinate axes, the information of what concrete position these mislocated prims hold remains unclear to the user of the extension as for that purpose, the CSV would still need to be manually screened. To implement this feedback, a *Position-Check Light* is added to the extension. This pre-check to the actual check for positional deviations offers the functionality to visualise the bounding box of an entire USD stage before conducting the actual check for positional deviations of selected prims. The core functions included in the classes `DrawUtil` and `ViewportScene` to draw lines between defined points in the 3D environment are adopted from a company-internal extension and adapted to the needs of the *Position-Check-Light*. To initiate the bounding box drawing, the user opens a USD stage, selects all prims from within this stage by clicking on its root prim at the top of the prim tree and then pushes a button that is conveniently integrated into the user interface. This calls the function `_on_position_check_light()`, which is, other than the core quality checks, part of the class *WidgetWindow*. By firstly computing and then drawing a bounding box around the extents of the entire virtual factory model, users are able to quickly oversee whether this bounding box shows any distortions and if so, approximately where in the virtual factory prims must be mislocated to have caused the bounding box to take this distorted shape. An example of a *Position-Check-Light* for the same positional deviation of a forklift analysed in chapter 5.1.1 is to be seen in Figure 6.2. It is evident that the forklift prim's position both deviates on the z- as well as the x- or y-axis.

Figure 6.2: Result of a *Position-Check-Light* in the *Nvidia Omniverse*, modified from *USD Physics Sample Pack*, by *Nvidia*, 2022

While the main function to identify prims outside a 99.99% confidence interval from the coordinate mean performs an extraction and export of all relevant position data to a CSV file, the *Position-Check Light* demands less computational effort while still conveying an analysis of the extreme coordinate values of and thereby potential deviations in a virtual factory. Users are able to quickly see a comprehendible result which benefits understandability, usability and thereby ultimately acceptance of the developed extension and the entire *Nvidia Omniverse* software.

Another addition that was made during the first reiteration cycle is the complementation of the KPI dashboard in the user interface with a histogram that visualises the z-coordinate distribution of all prims by showing the number of prims per z-coordinate value. As previously explained, deviations between prim positions in the virtual factory and their corresponding assets in reality occur most commonly on the z-axis. The visualisation of this histogram allows it to users to not only see the bounding box as well as the percentage of mislocated prims, but easily obtain an overview over the number and rough position of those prims mislocated on the z-axis. Testers of the software extension mentioned this histogram to make a highly valuable addition to the KPI dashboard.

Figure 6.3 shows the complete KPI dashboard following a check for positional deviations of a virtual factory model. It is visible that only 0.23% of the prims are mislocated on the x- and y-axis, which represents the forklift illustrated in Figure 5.2 and Figure 6.2.

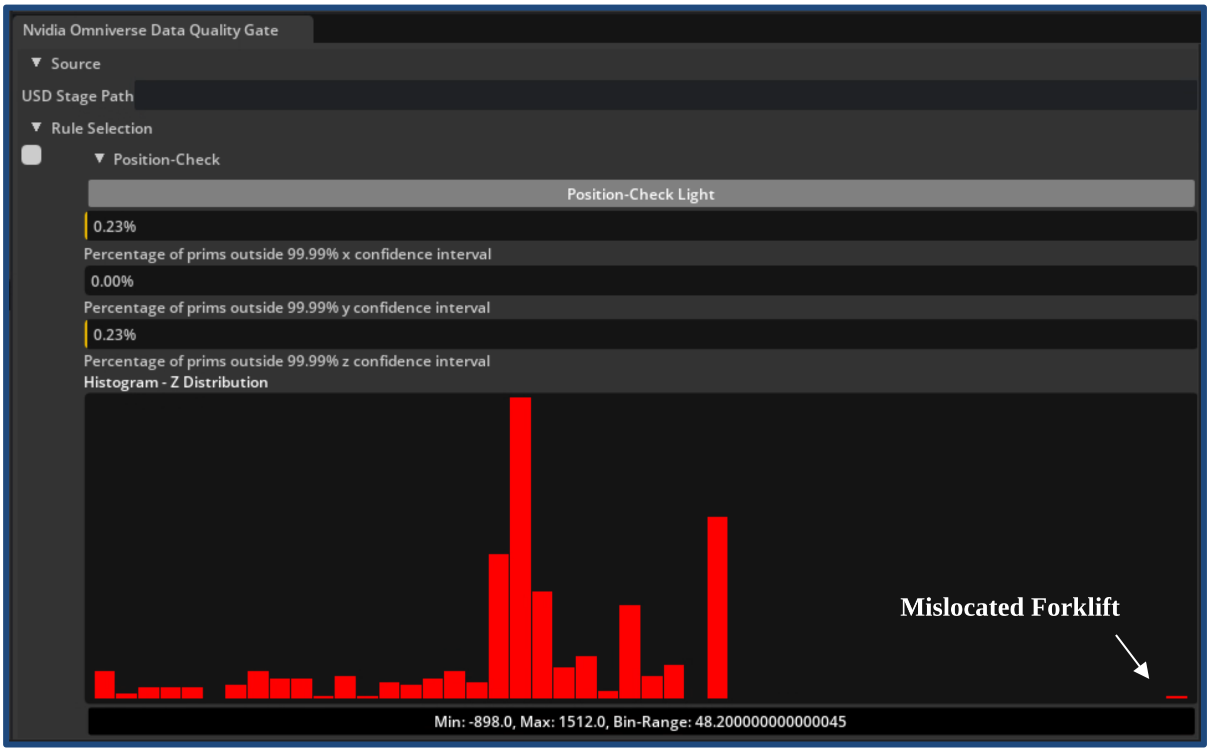


Figure 6.3: KPI Dashboard for Positional Deviations

6.1.2 Reiteration Cycle 2: Revision of Scaling Deviation Identification

As testing proceeded after the first reiteration cycle, it became obvious that the intended approach to identify scaling deviations, as chosen and detailed in chapter 5.1.2, does not reliably identify all prims whose size significantly exceeds that of its corresponding factory asset as it was planned or built in reality. Tests with different virtual factory model data sets revealed that the identification of such prims through their volume does not work sufficiently for prims whose extents are minimal in one or more directions, for example thin walls. Even if such a prim is falsely scaled and significantly larger in the virtual factory compared to the factory as built or as planned in the source systems, its volume might still be small enough to remain underneath the upper threshold of the one-sided 99.99% confidence interval of the mean sample volume. 7However, it was concluded from testing that the only prims that are falsely scaled yet fail to be recognised by the developed extension are either representing factory walls or pillars. Therefore, these prims are not identifiable by their volume, but by their disproportionally large extents in z-direction. Thus, a second condition is introduced that checks the z-extents of each prim's bounding box (see Source Code 6.3). These extents are computed in the exact same fashion as depicted in Source Code 5.6. Subsequently, the one-sided confidence interval is also computed for these z-extents by determining the mean volume, standard deviation and upper threshold of the confidence interval to the mean z-extents of all prims. The factor by which the standard deviation has to be multiplied in order to arrive at a 99.99% one-sided confidence interval is 3.5 (Miller & Childers, 2004).

```
# What prims exceed mean volume and z-extents by factor 3.5
        for item in scaling_data:
            if item['Volume'] >= v_upper or item['Z-Extents'] >= z_ext_upper:
                 item['Scaling'] = False
            else:
                 item['Scaling'] = True
```

Source Code 6.3: Revised Extension: Volume of Extended Bounding Box

After making the described adjustments to the extension, functionality could be ensured in all data sets. Additionally, testers wished for the addition of a histogram to the KPI dashboard illustrating the number of prims with a matching bounding box volume to gain an overview of how these volumes are generally distributed throughout a virtual factory and whether significant differences are observable when comparing multiple factories.

Following from the second reiteration, testing was proceeded and completed without further needs for reiteration. 2D prims, performance-inhibiting data sets, empty files and naming violations could successfully be identified throughout all data sets included in testing. A table containing all testing data and results as well as the final KPI dashboard are presented in appendices A.9 and A.10.

6.2 Preventive Measures

After successfully concluding on the testing and reiteration process of the software extension, possible mitigation strategies to reduce or ideally avoid the future occurrence of the most critical data quality insufficiencies need to be addressed. To come up with applicable measures that serve this purpose, root causes to the six insufficiencies targeted with the developed extension need to be known. During open discussions as part of the expert interviews, these causes were established (see Table 4.3).

The problem of empty files being transferred from source systems all the way to visualisation in the *Nvidia Omniverse* is caused by missing standards at exit points of the source systems. If applicable standards were in place, files without content would not leave source systems for conversion into USD.

The other five of the six most critical insufficiencies are entirely caused by workflow issues, so human errors, of those engineers providing data in the source systems. Positional deviations of selected prims can be traced to the assignment of faulty coordinates to real factory assets in the source systems, especially on the z-axis. As previous to the *Nvidia Omniverse* introduction, engineers mostly used 2D layout sketches for factory planning, the additional task to assign z-coordinates to every asset is only now moving into the responsibility field of those engineers. The unfamiliar new workflow consequently leads to errors. This unfamiliarity with the new tools in combination with adjusted processes of work also causes mistakes when allocating size parameters such as extent values to each asset, causing scaling deviations and 2D prims appearing in virtual factory models. Performance-inhibited virtual factory models are traceable to engineers that put too much emphasis on detail when creating reoccurring assets in source systems leading to a significantly high number of total mesh prims in the virtual factory. Such assets, for example a robot that appears more than 20 times within the same factory, often contain a high number of individual parts. Ideally, these parts should be merged to only be visualised as one single robot mesh prim in the virtual factory. However, occasionally, engineers fail to perform this merge before reduplicating the asset. This causes the number of total mesh prims within a virtual factory model to exponentially increase and the performance and FPS rate to decline as a consequence. Finally, violations of the naming convention are also caused by human error, namely false labelling of assets created in source systems.

Hence, there are two kinds of preventive measures imaginable to induce a sustained data quality improvement and mitigate the risk of critical insufficiencies appearing in the future: Firstly, strategies to mitigate insufficiencies caused by an unfamiliarity with the new workflow and secondly, strategies to mitigate insufficiencies caused by an absence of appropriate standards in the source systems. Both types are presented in the following two subsections.

6.2.1 Mitigation of Insufficiencies Caused by Unfamiliarity With the Workflow

To tackle human errors, familiarity with the *Nvidia Omniverse* software as a tool to support multiple activities in and around factory planning must be increased. All involved stakeholders need to be fully trained to adapt their workflows and avoid errors in the process of creating virtual factory models. In case data sets supplied by the *Factory Assembler* fail to pass the quality gate represented by the developed extension, because the minimum quality requirements regarding one of the five workflow-related insufficiencies are not met, an applicable employee training should be proposed. Even though it would be technically feasible, it is not recommended to directly assign this training only to the one engineer responsible for causing the insufficiency to not individually blame anyone and avoid privacy-related complaints. Instead, the production line segment affected by insufficiencies should be identified and the training be assigned to all contributors from this particular segment. The segment from which an insufficiency arose is easily identifiable from a prim's path which contains this information, as previously mentioned. Next to assigning appropriate trainings, all involved stakeholders should be transparently included into the roll-out process of the *Nvidia Omniverse*. It is of utmost importance that the stakeholders are aware of the company-internal trajectory and goals pursued with the introduction of the *Nvidia Omniverse*. Thereby, acceptance is increased which boosts openness to get used to the tool. Consequently, stakeholders become familiar with all workflow standards whose fulfillment is needed to create virtual factory models that pass the requirements set by the data quality gate software extension developed as part of this research project.

Both the automatic assignment of trainings to company employees responsible for data handling in the virtual factory production line segment containing flawed data as well as the continuous inclusion of all stakeholders are not exclusively achievable through a software extension of the *Nvidia Omniverse*. Hence, this task is to be enforced by the product owner, the department responsible for the company-internal *Nvidia Omniverse* roll-out, based on the recommendations provided in this chapter.

6.2.2 Mitigation of Insufficiencies Caused by Missing Source Systems Standards

Insufficiencies caused by missing source system standards are to be handled by creating appropriate standards at the exit points of all source systems that prevent flawed data from being converted to the USD format. In case files do not contain any information and are therefore empty, such a standard is easily to be set by only allowing data with a file size greater than zero bytes to exit the source systems for conversion and throwing an appropriate error message to engineers whose data input does not fulfill that standard. Thereby, the empty files can be removed or fixed in time, only a flawless data set without empty files is converted to USD and the risk of empty files being visualised in the *Nvidia Omniverse* 3D environment is eliminated. The creation of such a source system check is not part of the thesis scope, but should be approached based on the recommendations given in this chapter.

6.3 Pre-Post-Analysis

To conclude on stages three, four and five of the DSRM approach, the completed data quality gate needs to be evaluated by determining the degree to which the artefact objectives defined in stage two (see chapter 4.6) have been achieved. This evaluation is performed by comparing the situation previous to the introduction of the data quality gate to the situation afterwards.

Before the initiation of this research project, data quality assessments of virtual factory models visualised in the *Nvidia Omniverse* 3D environment had to be conducted fully manually. This procedure was time-consuming and prone to errors, causing the benefits of the *Nvidia Omniverse* in the automotive manufacturing sector to be outweighed by the costs of ensuring sufficient data quality. Accordingly, the artefact objectives were defined as follows: A software extension was to be designed that is capable of automatically identifying the most critical insufficiencies within a virtual factory model and performing an analysis of the results to compute adequate KPIs. Based on these KPIs, minimum data quality requirements were to be defined that need to be met by virtual factory models coming from the *Factory Assembler* to be published for visualisation in the *Nvidia Omniverse*. Thereby, a data quality gate is created that ensures a certain quality standard of data in the *Nvidia Omniverse*. Furthermore, mitigation strategies were to be proposed to accomplish a sustained elimination of the identified data quality insufficiencies from future virtual factory models.

Comparing these objectives to the results described, it can be summarised that the development of a *Nvidia Omniverse* software extension that serves as a data quality gate for virtual factory models was a success as all artefact objectives were completely fulfilled. After two reiterations, the extension is now capable of flawlessly identifying all of the six most critical insufficiencies, perform an applicable analysis of the data obtained from this identification and derive concrete KPIs. Using these KPIs, requirements were defined based on which virtual factory models supplied from the *Factory Assembler* can be evaluated. If the defined minimum standards to meet are not achieved by a virtual factory model, the data quality gate is not passed and the model is not to be publishable in the *Nvidia Omniverse* 3D environment. Depending on the type of data insufficiencies causing the virtual factory model to not pass the quality gate, two different mitigation strategies are conceivable to tackle the root causes and achieve a sustained data quality improvement. The developed software extension to automatically identify insufficiencies and derive KPIs to be able to define concrete data quality requirements in combination with the proposal of concrete measures to achieve a repair of the identified insufficiencies eliminates the need for manual data quality checks and hence is an answer to Q5. The *Nvidia Omniverse* can thereby unfold its full potential and the observed automotive manufacturer is able to take full advantage of the benefits held by the industrial metaverse applications enabled through this software.

6.4 Communication

The completed software extension is finally communicated to all involved stakeholders. As the full project progress was continuously documented on the company-internal *Git* server, sharing of the software extension is uncomplicated. Appendix A.8 contains the guide on how to enable and use the extension within the *Nvidia Omniverse* in the same form as it is also communicated to users. Finally, an abstracted *Unified Modelling Language* (UML) diagram is added in appendix A.11 to further facilitate understanding. As part of this project, only the classes *WidgetWindow* and *DQCheck* were newly created. The class *DQCheck* contains all functionalities relevant for the identification of data quality insufficiencies as described in chapter 5.1 while *WidgetWindow* creates the user interface as explained in chapters 5.2 and 5.3.

A class to provide the main code to integrate the extension into the *Omniverse* architecture is provided by *Nvidia*. All classes required to draw lines and ultimately enable a *Position-Check-Light* (see chapter 6.1.1) as well as existing company-internal standards on how the extension should be callable from the main platform are only minorly adjusted and imported from company-internal developers.

7 Discussion

As a result of the DSRM approach by Peffers et al. (2007), an extension to the *Nvidia Omniverse* software was developed that is capable of serving as a data quality gate and inducing a sustained improvement of data quality while eliminating the need for manual data quality checks. In the following, limitations and reflections on this outcome are discussed before highlighting areas of future research.

7.1 Limitations

During the course of the research project, a selection of limitations was encountered. The limitations as well as the measures undertaken to overcome these obstacles need to be presented.

First of all, one of the biggest risks associated with analysing the broad industrial metaverse research field at the example of only one case is the limited generalisability of this approach. The extent to which the developed solution is transmittable to similar manufacturers or software solutions outside the specific scope of this study holds as a significant limitation that needed to be incorporated for in the research design. To increase this external validity, an analysis of comparable and enabling software was conducted. As a part of this analysis, it was established that the *Nvidia Omniverse*, at the time of thesis creation, is the only software capable of visualising a 3D environment that can enable applications associated with the concept of an industrial metaverse as explained in chapter 3.2. Furthermore, it was described that this software is used by organisations from multiple industries, including several automotive manufacturers. It can be assumed that these industry players, especially those from the automotive sector, are dealing with similar questions as those dealt with in this project.

Another significant limitation of the research project arises from the time constraint imposed by the scope of this thesis, which restricted the project to the early stages of the company-internal roll-out of the *Nvidia Omniverse*. This resulted in two negative effects: Firstly, the significance of the results is deemed limited in comparison to conducting the study as a longitudinal project spanning over an extended duration. Secondly, only a limited number of early adopters were starting to gain familiarity with the software at the time of the project. From the limited number of users, only a portion were considerable as technical experts eligible to be interviewed, leading to a total of only six interviewees (see chapter 4.4). A small sample size generally reduces the degree of capacity and reliability of the qualitative data collected (Blackford, 2017). However, even though the total number of interviewees was small, the sample size in relation to the total population can still be assumed as sufficient (Guest et al., 2006). The population is equal to the total number of eligible interviewees, so the number of technical experts within the automotive manufacturer, namely ten. Hence, the sample size was greater than half of the entire population with the interview results being mostly unanimous across all interviews. Consequently, the collected qualitative data is considered to be reliable.

Another set of limitations is related to technical constraints set by the *Nvidia Omniverse* software environment, into which the extension developed as part of this research project is integrated. One of the most significant technical constraints emerging during the development process was the unavailability of appropriate documentation of Python libraries specific to the *Nvidia Omniverse*. Relevant modules and related commands required to develop an extension that identifies data quality insufficiencies, analyses the resulting data and finally visualises the results in an *Omniverse*-specific user interface are to this date not comprehensively summarised in an accessible document. Hence, the development process required extensive research in combination with a trial-and-error approach. If the command for a specific intended operation, such as computing a prim's bounding box, could not be located despite extensive research, an expert from *Nvidia* was contacted through a dedicated developer's forum. Thereby, the coding progress was hampered in early stages of the process. However, due to a steep learning curve, functionality of the code was ensured step-by-step throughout the duration of this thesis.

Another limitation is posed by the fact that two of the six identification procedures to spot insufficiencies, namely both positional as well as scaling deviations, are based on statistical calculations. Thus, only prims outside a 99.99% confidence interval from the mean are declared to be falsely positioned, respectively scaled. If a prim mistakenly falls into the rejection area, this produces a statistical type I error, while only minor deviations between the virtual model and the factory as built or as planned are not identifiable by the developed software extension. Such prims that wrongly fall into the confidence interval fall are called statistical type II errors. Type I errors are only known to occur in the case of scaling deviations. Here, the extension occasionally mislabels prims that are scaled correctly, but erroneously fall outside the confidence interval, as oversized. This leads to an average inaccuracy of up to 2% for each automatic scaling check with the extension. On the other hand, type II errors as of now exclusively occurred in the context of positional deviations with technical experts being aware of only one virtual factory model that features slightly mislocated prims leading to a potential type II error. To quantify the degree to which this error leads to the extension failing to identify certain insufficiencies, the extension is tested at this particular virtual factory model. Table 7.1 contains the results of this test and reveals that the extension missed around 0.05% of slightly mislocated prims due to a type II error.

Table 7.1: Determination of Extension Accuracy

Size	**Data Quality Insufficiencies**						
	Position (found)	Position (not found)	Scaling	2D	Perfor-mance	Empty-File	Naming
480603 Prims	x: 0% y: 0% z: 0%	x: 0% y: 0% z: 0.05%	1.47%	5.67%	Yes	0	0

All in all, the extent to which the extension accurately spots all current insufficiencies is thus **97.95%**. Apart from the mentioned statistical errors, the developed software extension is suitable to be applied for all data sets.

Lastly, the concept of the industrial metaverse and its applications around which the whole thesis evolves has only recently appeared in contemporary literature and is not uniformly defined yet. Therefore, an understanding of the concept needed to be established first in order to properly explain use cases of the metaverse in automotive manufacturing summarised under the virtual factory term. This was accomplished by properly reviewing literature evolving around BIM, digital twin technology, simulation as well as existing industrial metaverse definitions. By demonstrating relationships between the concepts and analysing the argumentation of different scientists elaborating on the subject, a comprehensive explanation of the industrial metaverse was provided and developed within this thesis. The potential use cases, so applications of the industrial metaverse in automotive manufacturing, could then be laid out to simultaneously answer Q1.

7.2 Expert Validation

After discussing limitations to the presented research approach, the success of the research project and the functionality of the developed solution need to be evaluated by the product owner. As the key problem identified was the time intensity and error susceptibility of manual virtual factory data quality checks, the usefulness of the tool is quantified by comparing the results of a manual data quality check to the results of applying the software extension for a set of exemplary virtual factory models.

The time to spot a data quality insufficiency of any sort can hardly be broken down to a specific detection time per prim as some insufficiencies are more obvious than others. Therefore, in coordination with the technical experts, the duration of a full manual data quality check of each of the eight virtual factory models used for testing purposes in chapter 6 is timed. Furthermore, a successful software extension execution is timed and added up for each automatic quality check. Based on these metrics, a quantified comparison of the time savings induced by the introduction of the developed software extension can be performed. The results of this comparison for the eight virtual factory models are detailed in Appendix A.12. It can be concluded that using automatic data quality checks instead of manual screening techniques reduces the average quality check time by about 99%. It must be noted that this procedure is prone to the key limitation that during the manual data quality checks, not all insufficiencies spotted by the software extension could be identified. Moreover, the technical experts identified further, less critical insufficiency types that the extension is not programmed to identify.

In conclusion, the key stakeholders of the product and especially the product owner were highly satisfied with the output produced in this research project. While the key benefit lies in the immense time savings from automating quality checks, these checks are also less likely to miss any insufficiencies and produce a comfortably analysable output in the form of adequate KPIs. In addition, company stakeholders mentioned the capabilities of the extension to be run in the backend as well as the proposed mitigation strategies as particularly helpful to introduce the extension as a quality gate between *Factory Assembler* and *Nvidia Omniverse* into the software architecture.

7.3 Reflections

The research outcome of this thesis must be critically reflected on due to the unique circumstances and limitations surrounding the studied concepts. Even though the results are exposed to the risk of suffering from a limited external validity, I am convinced that the produced output is of significant practical and theoretical relevance. In my opinion, the uniqueness of the studied case is not causing the results to be less meaningful or generalisable, but is laying the groundwork for future research emerging around the concept of an industrial metaverse in general and the remaining obstacles of digital platforms to enable its applications in particular. Thus, the presented DSRM approach to overcome such barriers is capable to serve as a guidance for future research projects that pave the way for a gradual enablement of further industrial metaverse use cases in other industries.

7.4 Areas of Future Research

This thesis can be argued to be a kick-starter to the scientific discussion of the potential applications to an industrial metaverse hosted on digital platforms. The potential future fields of study evolving from the results of this thesis are numerous.

Generally, the concept of data quality is only one relevant data-related aspect to mind as the importance of digital platforms in manufacturing is on the rise. Next to data quality, data privacy and governance are two topics that have not been sufficiently covered by literature yet. This thesis touches upon the subject of data governance as part of the preventive measures discussed to mitigate the risk of future data quality insufficiencies (see chapter 6.2). Here, it is proposed that appropriate trainings should be assigned to those employees handling data from a specific virtual factory production line segment. Nonetheless, as the relevance and size of company-internal data rises with the introduction of the *Nvidia Omniverse*, standards and data policies that apply to how data is gathered, processed and stored need to be set. **A data governance framework needs to be introduced that clearly defines all roles and responsibilities associated with those tasks.**

The result of this thesis is a software extension that can identify data quality insufficiencies and visualise appropriate KPIs in a user-friendly interface. Based on the derived KPIs, preventive measures are proposed to improve the data quality. However, as touched upon in chapter 6.2, this thesis only recommends a set of strategies to pursue in order to mitigate future insufficiencies. It does not focus on the implementation of those strategies. The implementation process requires addressing the root causes of each data quality insufficiency. In case an insufficiency is caused by a workflow error, the responsible set of employees should follow a training to increase their familiarity with the *Nvidia Omniverse* and the work process to follow to visualise a flawless virtual factory. Future work could cover the **development of a software-overarching script that identifies the production line segment of the factory in which faulty data is located and assigns a training measure to the affected employees**. Another cause for data quality insufficiencies is the absence of quality standards in source systems. Addressing this issue requires initiating future projects aimed at **improving the source systems and aligning them with the standards established by the data quality gate** developed in this thesis.

As the extension is already executable without having to visualise a virtual factory in the 3D environment, future work could also contain an integration of the extension into the *Factory Assembler* that blocks the publishment of insufficient virtual factories to the *Omniverse* by automatically checking for alignment with the requirements in the backend. If these standards are not met, the gate does not open and measures need to be taken to improve data quality (see Figure 4.9). To induce a sustained improvement of data quality, the exported CSV files could additionally be fed to **machine learning algorithms to identify patterns among the insufficient data sets**. The supervised machine learning approach *Classificati*on is imaginable to use the developed data quality gate as an anomaly detector during the conversion process. Virtual factory models classified as insufficient should then be routed to staging for review before being published. Data managers would review the staged data and fix the detected insufficiencies. A machine learning algorithm could then learn from the reviews to identify, auto-fix and keep track of insufficiency patterns in an accessible data base.

To conclude from the above paragraphs, future research should look into:

1. Introducing a data governance framework that guides large corporations in handling the unprecedented data sizes associated with the industrial metaverse
2. Implementing the mitigation strategies proposed in this thesis, namely
    a. Automatic training assignments
    b. Aligning source system standards with the set requirements
3. Applying machine learning algorithms to insufficient virtual factory data sets to independently identify, track and auto-fix data quality insufficiency patterns

8 Conclusion

In order to draw an adequate conclusion of the research project and evaluate on whether the outcomes of the project successfully fulfill the research objective, the answers to each sub-question as well as the overall research question are summarised. The research project evolves around the problem of how to automatically identify insufficient data in an industrial metaverse platform. This subject was studied at the case of an automotive manufacturer that uses the *Nvidia Omniverse*.

8.1 Research Outcomes

The key objective to achieve with this research was to come up with a solution that allows for automatic quality checks of virtual factory models in the *Nvidia Omniverse* platform as the current necessity to perform these checks manually is time-consuming and hence costly. The overall research question is accordingly directed at investigating how quality insufficiencies within spatial data digital platforms can automatically be identified to improve industrial metaverse applications in the automotive industry. The research question was answered by developing a software extension to the *Nvidia Omniverse* that is capable of identifying the most critical data quality insufficiencies apparent in the 3D environment created by the platform. To sustainingly improve data quality in those data sets failing to fulfill certain minimum requirements, adequate preventive measures were moreover defined. Thereby, the output of this thesis allows for both the identification of existing insufficiencies as well as the mitigation of future insufficiencies and thus improvement of overall data quality. This advances the potential of industrial metaverse applications in the automotive industry as the benefits of the technology can now be fully taken advantage of.

Leading up to answering this overall research question, a total of five sub-questions were answered. Firstly, sub-question one demanded for current and future industrial metaverse applications of digital platforms to be outlined. Therefore, relevant concepts surrounding digital manufacturing platforms as well as the industrial metaverse were analysed by the means of a PRISMA literature review. Furthermore, software architectures of platform ecosystems that support or might enable an industrial metaverse were presented as part of chapter 4.1. It was established that as of now, companies in an industrial context are only starting to explore the possibilities of such a metaverse. Therefore, most applications of the industrial metaverse still remain to be projections of the future. The most significant use-case, especially for manufacturing industry players, is the concept of building a virtual factory.

To answer the second sub-question, the question of what stakeholders of an automotive manufacturer are affected by data quality insufficiencies of an industrial metaverse platform, a stakeholder analysis was conducted at the example of the observed automotive OEM. It can be concluded that the degree to which a stakeholder is affected correlates with the degree to which this stakeholder uses the platform in the workplace. Company-internal product owner and therefore most affected by data quality insufficiencies is the department responsible to introduce and roll-out the software to a wider base of internal users.

Subsequently, interviews with technical experts were performed to firstly identify all data quality insufficiencies observable in a 3D industrial metaverse environment, before evaluating these insufficiencies in terms of severity, occurrence and detectability based on the collected interview data. An FMEA was conducted to arrive at an action priority matrix that visualises a criticality ranking of all insufficiencies. The matrix revealed that six types of data quality insufficiencies are particularly critical and therefore are most important to be automatically identifiable. Thereby, sub-question three (*What spatial data quality barriers exist in an industrial metaverse platform?*) and sub-question four (*How can the identified data quality barriers be prioritised?*) were answered.

Following from the insufficiency prioritisation, sub-question five (*What software extension fits the stakeholder requirements of Q2 to tackle the insufficiencies prioritised in Q4?*) was answered and a software extension to the *Nvidia Omniverse* was developed that automatically identifies the six most critical insufficiencies. To comply with stakeholder requirements, the extension was designed in a way to function as a data quality gate between *Factory Assembler* and *Omniverse* which needs to be passed by virtual factory data in order to be eligible for visualisation in the 3D *Omniverse* environment.

An overview of the research outputs is provided as part of Table 8.1.

Table 8.1: Research Outputs

Type	Research Question	Short Answer
RQ	*How can quality insufficiencies within spatial data digital platforms automatically be identified to improve industrial metaverse applications in the automotive industry?*	• Identification Software Extension • Derivation of KPIs • Mitigation Strategies
Q1	*What are current and future industrial metaverse applications of digital platforms?*	• Smart Manufacturing (Virtual Factory, Predictive Supply & Maintenance)
Q2	*What stakeholders of an OEM are affected by data quality insufficiencies of the platform?*	• Platform Users • Product Owner: Roll-Out Team
Q3	*What spatial data quality barriers exist in an industrial metaverse platform?*	• 18 Total Insufficiencies (6 highly critical)
Q4	*How can the identified data quality barriers be prioritised?*	• FMEA (Action Priority Matrix)
Q5	*What software extension fits the stakeholder requirements of Q2 to tackle the insufficiencies prioritised in Q4?*	• Identification + KPI Dashboard • Quality Gate for Virtual Factory Data to be Published

8.2 Relevance to the Study Programme

The *Management of Technology* (MOT) master programme introduces students to the relevance of technology as a corporate resource. In that context, students learn how companies make use of technology to develop products that improve productivity and thus ensure profitability. Moreover, the programme puts an emphasis on educating engineers that are capable of consulting firms on how to successfully implement innovative technological solutions into their organisational context.

I am convinced that the research field dealt with as part of my thesis ideally fits these conditions and is hence highly applicable to the MOT study programme. At first, the relevance of data-based digital platforms and industrial metaverse applications for corporations from the automotive sector was investigated and understood. As part of this thesis, the introduction of the *Nvidia Omniverse* platform to visualise virtual factories in an immersive 3D environment for the purpose of factory layout planning was studied. If successfully integrated, the platform technology offers the potential to revolutionise existing, rigid planning processes. Thereby, overall productivity and profitability are expected to be decisively enhanced. Subsequently, the research problem was defined by revealing the obstacles automotive manufacturers still face to introduce the technology. Then a solution to overcome these obstacles, namely the inability to automatically identify data quality insufficiencies, was developed. To facilitate a smooth implementation of the solution, it was fully adapted to the organisational requirements of the studied firm. Taking these aspects into consideration, the thesis scope exhaustively covers the key learning goals of the MOT programme and consequently makes for a perfect final project to conclude on my master studies.

8.3 Scientific Contribution

This research project touches upon several research fields that only recently emerged in literature, all of which hold a significant scientific relevance. Firstly, the results of this thesis serve as the first instance in literature that analyses the subject of data quality in the context and dimensions of digital manufacturing platforms to digitally represent entire production systems. Following from developments associated with the digital transformation and I4.0, manufacturers are making use of data-driven digital platforms whose business relevance grows with the number of use cases for such platforms. Analogously, data sizes are constantly increasing causing the manufacturers' dependence on flawless data to move into focus. This thesis contributes to the scientific discussion by investigating the importance of high-quality data at the case of the interoperable *Nvidia Omniverse* that handles large sizes of spatial data.

Last but not least, this thesis lays the groundwork for the industrial metaverse concept in manufacturing in general and automotive manufacturing in particular. The rising relevance of the industrial metaverse concept in literature is shown in Figure 8.1. The related search was conducted on the 26th of May 2023.

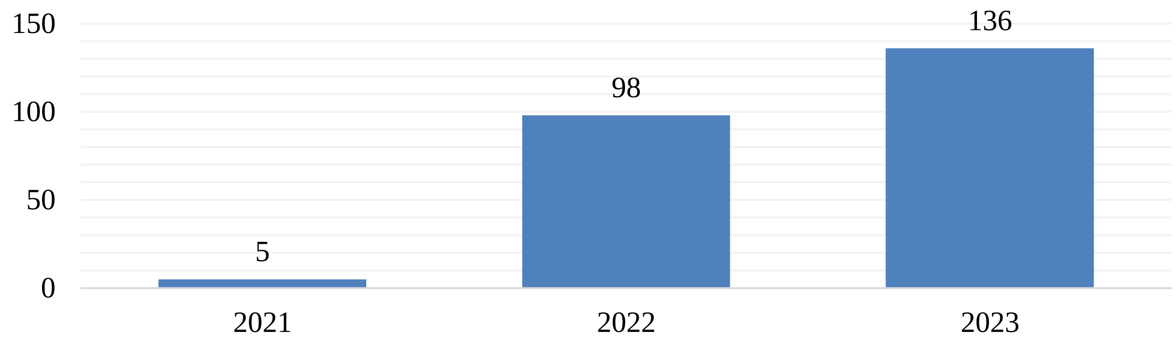


Figure 8.1: SCOPUS results for the query *ALL(Industrial Metaverse)*

Even though the industrial metaverse concept is discussed by multiple researchers, industrial metaverse applications enhanced by the capabilities of BIM, digital twin technology and industrial process simulations have not been specified in literature. The idea of a virtual factory, an industrial metaverse application that is integrated on a digital platform, is highly innovative. This research project introduces this concept to the scientific community and thereby undoubtedly makes a substantial contribution.

Furthermore, this research makes a significant addition to the scientific state-of-the-art in the field of data management in general. Research specifically directed at studying the role of data quality in digital platforms in a data-heavy I4.0 context was still missing from contemporary literature. The presented research framework closes this corresponding knowledge gap. Additionally, this research holds as the first instance to examine solutions to enhance the data quality within a large-scale digital platform whose architecture is based on integrating and converting the input of several source systems in one database. Achieving a high degree of data quality in such an interoperable platform is highly sophisticated and was not yet included into the academic debate. The approach pursued in this research can guide the data management community to sustainingly improve the data quality within comparable platform structures as detailed in chapter 8.5. Thus, a method was developed that goes beyond the state-of-the-art and advances the scientific discussion among researchers of that field.

8.4 Societal Contribution

The digital transformation affects a numerous variety of social processes, thereby creating new conditions for further societal digitalisation. One of the most recent disruptive technologies to emerge in that context is the metaverse concept, which is rapidly gaining popularity in multiple areas of society. The potential use cases of creating an immersive virtual world are numerous, however, technical barriers are still hampering the implementation of metaverse technology. A particularly impactful transformation is expected to be induced by the industrial metaverse in automotive manufacturing. Here, the market environment is characterised by a high volatility, which is why manufacturers are exploring industrial metaverse applications to work towards smart manufacturing and maximise production flexibility. Based on simulation, BIM and digital twin technology, automotive manufacturers are building virtual factories in an industrial metaverse to, in a first step, maximise efficiency of their planning processes and ultimately production. In addition, complete simulations of human movement in an industrial metaverse can allow for precise ergonomics analyses and optimisations. Especially considering the demographic development of aging societies in industrial countries, such analyses could allow for the specific assignment of age-adequate physical tasks to workers. The observable decline in skilled workers could thus be absorbed by allowing for more comfortable working conditions in which production workers can flourish even at a higher age. This thesis investigates how one of the most decisive technical barriers to innovative metaverse technology, namely data quality insufficiencies, can be overcome in the manufacturing sector. Hence, it is of immense societal relevance as it advances the implementation efforts of the metaverse in general and thereby contributes to enabling a world in which society can benefit from metaverse applications of all sorts.

8.5 Guideline for Framework Application in a Data Management Context

As depicted in chapter 8.3, the research approach pursued in this thesis makes a non-negotiable scientific contribution to the data management community. In order for the developed framework to be usable in a more general data management context outside of manufacturing, the process of how to generally apply the research approach must be described.

At first, it is essential to fully comprehend the architecture of the digital platform structure of analysis. This research provides an overview of how the architectures of platforms such as the *Nvidia Omniverse* that are interoperable with other software, be it in manufacturing or any other industry, are generally structured. These results should be used as a base to gain an understanding of other comparable platform structures. Next, the types of data quality insufficiencies observable within the platform need to be identified, categorised and prioritised. Here, it is crucial to utilise the opinion of key stakeholders such as technical experts that are highly familiar with the platform. An FMEA proves useful to analyse all insufficiencies and perform a funded evaluation of their negative impact. These stakeholders then need to be integrated into the development process of a solution that fits their specific needs. The empirical software engineering method is recommended to take into consideration the previously collected data and use it as input to iteratively design a software extension to the digital platform that identifies data quality insufficiencies. Large-scale digital platforms are characterised by a significant number of interconnected actors. To make sure that only data of sufficient quality is shared among such a large user base, minimum quality requirements should be determined by data managers. These requirements can be either adjusted to the specific use case or oriented at the severity score calculation described in chapter 5.3.2. To sustainingly improve the data quality of those data sets failing to fulfill the minimum requirements, adequate preventive measures are to be defined furthermore. Thereby, the developed output of this thesis guides data managers in both the identification of existing insufficiencies as well as the mitigation of future insufficiencies and thus facilitates the improvement of data quality in large-scale interoperable digital platforms of all sorts.

VI References

Abate, M. L., Diegert, K. V., & Allen, H. (1998). A hierarchical approach to improving data quality. *Data Quality*, *4*(1), 365–369.

Ahmadi, A., Sodhro, A. H., Cherifi, C., Cheutet, V., & Ouzrout, Y. (2019). Evolution of 3C Cyber-Physical Systems Architecture for Industry 4.0. In T. Borangiu, D. Trentesaux, A. Thomas, & S. Cavalieri (Eds.), *Service Orientation in Holonic and Multi-Agent Manufacturing* (pp. 448–459). Springer International Publishing.

Alcácer, V., & Cruz-Machado, V. (2019). Scanning the industry 4.0: A literature review on technologies for manufacturing systems. *Engineering Science and Technology, an International Journal*, *22*(3), 899–919.

Alpala, L. O., Quiroga-Parra, D. J., Torres, J. C., & Peluffo-Ordóñez, D. H. (2022). Smart Factory Using Virtual Reality and Online Multi-User: Towards a Metaverse for Experimental Frameworks. *Applied Sciences*, *12*(12). https://doi.org/10.3390/app12126258

Alt, R. (2021). *Electronic Markets on digital platforms and AI*. Springer. *Electronic Markets*, *31*.

Armendia, M., Ghassempouri, M., Ozturk, E., & Peysson, F. (2019). *Twin-Control A Digital Twin Approach to Improve Machine Tools Lifecycle: A Digital Twin Approach to Improve Machine Tools Lifecycle.* https://doi.org/10.1007/978-3-030-02203-7

Asan, U., & Soyer, A. (2015). Failure Mode and Effects Analysis Under Uncertainty: A Literature Review and Tutorial. In (Vol. 97). https://doi.org/10.1007/978-3-319-24499-0_10

Autodesk Building Industry Solutions. (2002). *Building Information Modeling*. http://www.laiserin.com/features/bim/autodesk_bim.pdf

Bagheri, B., Yang, S [Shanhu], Kao, H.-A., & Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. *IFAC-PapersOnLine*, *48*(3), 1622–1627.

Ball, M. (2022). *The metaverse: and how it will revolutionize everything*. Liveright Publishing.

Blackford, J. U. (2017). Leveraging Statistical Methods to Improve Validity and Reproducibility of Research Findings. *JAMA Psychiatry*, *74*(2), 119–120. https://doi.org/10.1001/jamapsychiatry.2016.3730

Boschert, S., & Rosen, R. (2016). Digital Twin—The Simulation Aspect. In P. Hehenberger & D. Bradley (Eds.), *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers* (pp. 59–74). Springer International Publishing. https://doi.org/10.1007/978-3-319-32156-1_5

Chandra Sekaran, S., Yap, H. J., Musa, S. N., Liew, K. E., Tan, C. H., & Aman, A. (2021). The implementation of virtual reality in digital factory—a comprehensive review. *The International Journal of Advanced Manufacturing Technology*, *115*(5), 1349–1366. https://doi.org/10.1007/s00170-021-07240-x

Chen, Y. (2017). Integrated and intelligent manufacturing: perspectives and enablers. *Engineering*, *3*(5), 588–595.

Cheong, L. K., & Chang, V. (2007). The need for data governance: a case study. *ACIS 2007 Proceedings*, 100.

Chryssolouris, G., Mavrikios, D., Papakostas, N., Mourtzis, D., Michalos, G., & Georgoulias, K. (2009). Digital manufacturing: history, perspectives, and outlook. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *223*(5), 451–462.

Constantinides, P., Henfridsson, O., & Parker, G. G. (2018). *Introduction—platforms and infrastructures in the digital age* (No. 2). Informs. *Information Systems Research*, *29*.

Dallasega, P., Revolti, A., Sauer, P. C., Schulze, F., & Rauch, E. (2020). BIM, Augmented and Virtual Reality empowering Lean Construction Management: a project simulation game. *Procedia Manufacturing*, *45*, 49–54.

David, B., & Fasquel, P. (1997). *Qualité d'une base de données géographique: concepts et terminologie*. Institut géographique national.

Devillers, R., Bédard, Y., Jeansoulin, R., & Moulin, B. (2007). Towards spatial data quality information analysis tools for experts assessing the fitness for use of spatial data. *International Journal of Geographical Information Science*, *21*(3), 261–282.

Ding, K., Chan, F., Zhang, X., Zhou, G., & Zhang, F. (2019). Defining a Digital Twin-based Cyber-Physical Production System for Autonomous Manufacturing in Smart Shop Floors. *International Journal of Production Research*, *1*, 1–23. https://doi.org/10.1080/00207543.2019.1566661

Dmitrieva, E. I. (2020). Digital Platforms as a Base for Forming a Digital Economy. In *2nd International Scientific and Practical Conference on Digital Economy (ISCDE 2020)* (pp. 553–558). Atlantis Press.

Eden, C., & Ackermann, F. (1998). Stakeholder analysis and management. *Making Strategy: The Journey of Strategic Management*, 113–135.

Faraboschi, P., Frachtenberg, E., Laplante, P., Milojicic, D., & Saracco, R. (2022). Virtual worlds (Metaverse): From skepticism, to fear, to immersive opportunities. *Computer*, *55*(10), 100–106.

Garetti, M., Rosa, P., & Terzi, S. (2012). Life Cycle Simulation for the design of Product–Service Systems. *Computers in Industry*, *63*(4), 361–369. https://doi.org/10.1016/j.compind.2012.02.007

Gerrikagoitia, J. K., Unamuno, G., Urkia, E., & Serna, A. (2019). Digital manufacturing platforms in the industry 4.0 from private and public perspectives. *Applied Sciences*, *9*(14), 2934.

Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future NASA and US Air Force vehicles. *Paper for the 53rd Structures, Structural Dynamics, and Materials Conference: Special Session on the Digital Twin*, 1818.

Greco, A., Caterino, M., Fera, M., & Gerbino, S. (2020). Digital twin for monitoring ergonomics during manufacturing production. *Applied Sciences*, *10*(21), 7758.

Grieves, M. (2002). Conceptual ideal for PLM. *Presentation for the Product Lifecycle Management (PLM) Center, University of Michigan*.

Gudivada, V., Apon, A., & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, *10*(1), 1–20.

Guest, G., Bunce, A., & Johnson, L. (2006). How Many Interviews Are Enough? An Experiment with Data Saturation and Variability. *Field Methods*, *18*(1), 59–82. https://doi.org/10.1177/1525822X05279903

Hagiu, A., & Wright, J. (2015). Multi-sided platforms. *International Journal of Industrial Organization*, *43*, 162–174.

Hardin, B., & McCool, D. (2015). *BIM and construction management: proven tools, methods, and workflows*. John Wiley & Sons.

Hazen, B. T., Boone, C. A., Ezell, J. D., & Jones-Farmer, L. A. (2014). Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. *International Journal of Production Economics*, *154*, 72–80.

Hein, A., Schreieck, M., Riasanow, T., Setzke, D. S., Wiesche, M., Böhm, M., & Krcmar, H. (2020). Digital platform ecosystems. *Electronic Markets*, *30*, 87–98.

Huber, T. L., Kude, T., & Dibbern, J. (2017). Governance practices in platform ecosystems: Navigating tensions between cocreated value and governance costs. *Information Systems Research*, *28*(3), 563–584.

J. Kang, D. Ye, J. Nie, J. Xiao, X. Deng, S. Wang, Z. Xiong, R. Yu, & D. Niyato (2022). Blockchain-based Federated Learning for Industrial Metaverses: Incentive Scheme with Optimal AoI. In *2022 IEEE International Conference on Blockchain (Blockchain)*.

J. Vachálek, L. Bartalský, O. Rovný, D. Šišmišová, M. Morháč, & M. Lokšík (2017). The digital twin of an industrial production line within the industry 4.0 concept. In *2017 21st International Conference on Process Control (PC)*.

Jain, S., Shao, G., & Shin, S.-J. (2017). Manufacturing data analytics using a virtual factory representation. *International Journal of Production Research*, *55*(18), 5450–5464.

Johnson, T., & Dasu, T. (2003). T3: Data Quality and Data Cleaning: An Overview.

Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0. *Final Report of the Industrie 4.0 Working Group*, *2013*, 82.

Khatri, V., & Brown, C. V. (2010). Designing data governance. *Communications of the ACM*, *53*(1), 148–152.

Kibira, D., & Shao, G. (2017). Virtual Factory Framework for Supporting Production Planning and Control. *IFIP Advances in Information and Communication Technology*, *APMS 2016*, 291–298. https://doi.org/10.1007/978-3-319-51133-7_35

Kirstein, P. T., & Ruiz-Zafra, A. (2018). *Use of Templates and The Handle for Large-Scale Provision of Security and IoT in the Built Environment*. https://doi.org/10.1049/cp.2018.0029

Kravets, J., & Zimmermann, K. (2012). Inter-organizational information alignment: A conceptual model of structure and governance for cooperations.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, *51*(11), 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474

Kusiak, A. (2019). Intelligent manufacturing: bridging two centuries. *Journal of Intelligent Manufacturing*, *30*, 1–2.

Lago, P., Bizzarri, G., Scalzotto, F., Parpaiola, A., Amigoni, A., Putoto, G., & Perilongo, G. (2012). Use of FMEA analysis to reduce risk of errors in prescribing and administering drugs in paediatric wards: A quality improvement report. *BMJ Open*, *2*. https://doi.org/10.1136/bmjopen-2012-001249

Lee, J. (1998). Teleservice engineering in manufacturing: challenges and opportunities. *International Journal of Machine Tools and Manufacture*, *38*(8), 901–910. https://doi.org/10.1016/S0890-6955(97)00135-1

Lee, S. U., Zhu, L., & Jeffery, R. (2017). Data governance for platform ecosystems: Critical factors and the state of practice. *ArXiv Preprint ArXiv:1705.03509*.

Lin, M.-H., & Fu, L.-C. (2001). A virtual factory based approach to on-line simulation and scheduling for an FMS and a case study. *Journal of Intelligent Manufacturing*, *12*, 269–279.

Liu, C., & Xu, X. (2017). Cyber-physical machine tool–the era of machine tool 4.0. *Procedia CIRP*, *63*, 70–75.

Liu, Z., Sampaio, P., Pishchulov, G., Mehandjiev, N., Cisneros-Cabrera, S., Schirrmann, A., Jiru, F., & Bnouhanna, N. (2022). The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration. *Computers in Industry*, *138*, 103623.

Llopis-Albert, C., Rubio, F., & Valero, F. (2021). Impact of digital transformation on the automotive industry. *Technological Forecasting and Social Change*, *162*, 120343.

Lu, C. J. J., Tsai, K. H., Yang, J. C. S., & Wang, Y. (1998). A virtual testbed for the life-cycle design of automated manufacturing facilities. *The International Journal of Advanced Manufacturing Technology*, *14*(8), 608–615. https://doi.org/10.1007/BF01301704

Miller, S. L., & Childers, D. (Eds.). (2004). *Probability and Random Processes*. Academic Press.

Mittal, S., Khan, M. A., Purohit, J. K., Menon, K., Romero, D., & Wuest, T. (2020). A smart manufacturing adoption framework for SMEs. *International Journal of Production Research*, *58*(5), 1555–1573.

Morrison, J. L. (1995). Spatial data quality. *Elements of Spatial Data Quality*, *202*, 1–12.

Müller, J. M [Julian M.], & Voigt, K.-I. (2018). Sustainable industrial value creation in SMEs: A comparison between industry 4.0 and made in China 2025. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *5*, 659–670.

Müller, J. M [Julian Marius] (2019). Antecedents to digital platform usage in Industry 4.0 by established manufacturers. *Sustainability*, *11*(4), 1121.

National Committee for Digital Cartographic Standards. (1987). *A draft proposed standard for digital cartographic data [microform] / Harold Moellering, editor*. *Issues in digital cartographic data standards: rept. # 8*. U.S. Geological Survey : [Books and Open-File Reports Section, distributor.

Nawari, N. O. (2018). *Building information modeling: Automated code checking and compliance processes*. CRC Press.

Negri, E., Fumagalli, L., & Macchi, M. (2017). A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing*, *11*, 939–948. https://doi.org/10.1016/j.promfg.2017.07.198

Nokkala, T., Salmela, H., & Toivonen, J. (2019). Data governance in digital platforms.

Nvidia. (2022). *USD Physics Sample Pack*. https://www.nvidia.com/en-us/on-demand/session/omniverse2020-om1565/

Nvidia. (2023a). *Manufactured in the Metaverse: Mercedes-Benz Assembles Next-Gen Factories With NVIDIA Omniverse*. https://blogs.nvidia.com/blog/2023/01/03/mercedes-benz-next-gen-factories-omniverse/

Nvidia. (2023b). *omni.usd API*. https://docs.omniverse.nvidia.com/kit/docs/omni.usd/latest/Overview.html

Oliveira, D. T. de, & Cortimiglia, M. N. (2017). Value co-creation in web-based multisided platforms: A conceptual framework and implications for business model design. *Business Horizons*, *60*(6), 747–758.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T [Tianjing], Loder, E. W., Mayo-Wilson, E., McDonald, S., . . . Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, *372*, n71. https://doi.org/10.1136/bmj.n71

Park, H.-T., Kwak, J.-G., Wang, G.-N., & Park, S. C. (2010). Plant model generation for PLC simulation. *International Journal of Production Research*, *48*(5), 1517–1529.

Park, S.-M., & Kim, Y.-G. (2022). A metaverse: taxonomy, components, applications, and open challenges. *Ieee Access*, *10*, 4209–4251.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Pirvu, B.-C., Zamfirescu, C.-B., & Gorecky, D. (2016). Engineering insights from an anthropocentric cyber-physical system: A case study for an assembly station. *Mechatronics*, *34*, 147–159.

Pixar Animation Studios. (2021). *USD Glossary*. https://openusd.org/release/glossary.html#usdglossary-prim

Pixar Animation Studios. (2023). *Pixar USD API*. https://openusd.org/dev/api/

Radanliev, P., Roure, D. de, Nicolescu, R., & Huth, M. (2019). A reference architecture for integrating the Industrial Internet of Things in the Industry 4.0. *University of Oxford Combined Working Papers and Project Reports Prepared for the PETRAS National Centre of Excellence and the Cisco Research Centre. Https://doi. Org/10.13140/RG*, *2*(26854.47686).

Reuver, M. de, Sørensen, C., & Basole, R. C. (2018). The digital platform: a research agenda. *Journal of Information Technology*, *33*(2), 124–135.

Rosen, R., Wichert, G. von, Lo, G., & Bettenhausen, K. D. (2015). About the importance of autonomy and digital twins for the future of manufacturing. *IFAC-PapersOnLine*, *48*(3), 567–572.

Rosenbaum, S. (2010). Data governance and stewardship: designing data stewardship entities and advancing data access. *Health Services Research*, *45*(5p2), 1442–1455.

S. K. Jagatheesaperumal, & M. Rahouti (2022). Building Digital Twins of Cyber Physical Systems With Metaverse for Industry 5.0 and Beyond. *IT Professional*, *24*(6), 34–40. https://doi.org/10.1109/MITP.2022.3225064

Sacco, M., Pedrazzoli, P., & Terkaj, W. (2010). VFF: virtual factory framework. In *2010 IEEE International Technology Management Conference (ICE)* (pp. 1–8). IEEE.

Sahi, G. K., Gupta, M. C., & Cheng, T. C. (2020). The effects of strategic orientation on operational ambidexterity: A study of indian SMEs in the industry 4.0 era. *International Journal of Production Economics*, *220*, 107395.

Schroeder, G. N., Steinmetz, C., Pereira, C. E., & Espindola, D. B. (2016). Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine*, *49*(30), 12–17.

Sepasgozar, S. M. E., Khan, A. A., Smith, K., Romero, J. G., Shen, X., Shirowzhan, S., Li, H., & Tahmasebinia, F. (2023). BIM and Digital Twin for Developing Convergence Technologies as Future of Digital Construction. *Buildings*, *13*(2). https://doi.org/10.3390/buildings13020441

Sicari, S., Rizzardi, A., Cappiello, C., Miorandi, D., & Coen-Porisini, A. (2018). Toward data governance in the internet of things. *New Advances in the Internet of Things*, 59–74.

Souza, M. C. F., Sacco, M., & Porto, A. J. V. (2006). Virtual manufacturing as a way for the factory of the future. *Journal of Intelligent Manufacturing*, *17*, 725–735.

Tan, Y., Yang, W., Yoshida, K., & Takakuwa, S. (2019). Application of IoT-aided simulation to manufacturing systems in cyber-physical system. *Machines*, *7*(1), 2.

Tao, F., Qi, Q., Wang, L., & Nee, A. (2019). Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering*, *5*(4), 653–661. https://doi.org/10.1016/j.eng.2019.01.014

Temel, B., & Basaga, H. (2020). Investigation of IFC file format for BIM based automated code compliance checking. *J. Constr. Eng*, *3*, 113–130.

Tiwana, A., Konsynski, B., & Bush, A. A. (2010). Research commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, *21*(4), 675–687.

Tiwana, B., Balakrishnan, M., Aguilera, M. K., Ballani, H., & Mao, Z. M. (2010). Location, location, location! modeling data proximity in the cloud. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*.

Tolio, T., Sacco, M., Terkaj, W., & Urgo, M. (2013). Virtual Factory: An Integrated Framework for Manufacturing Systems Design and Analysis. *Procedia CIRP*, *7*, 25–30. https://doi.org/10.1016/j.procir.2013.05.005

Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, *12*(4), 5–33.

Weyer, S., Meyer, T., Ohmer, M., Gorecky, D., & Zühlke, D. (2016). Future Modeling and Simulation of CPS-based Factories: an Example from the Automotive Industry. *IFAC-PapersOnLine*, *49*(31), 97–102. https://doi.org/10.1016/j.ifacol.2016.12.168

Wu, D., Greer, M. J., Rosen, D. W., & Schaefer, D. (2013). Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems*, *32*(4), 564–579.

Xu, H., Yu, W., Griffith, D., & Golmie, N. (2018). A survey on industrial Internet of Things: A cyber-physical systems perspective. *Ieee Access*, *6*, 78238–78259.

Yang, B., Yang, S [Shuang], Lv, Z., Wang, F., & Olofsson, T. (2022). Application of Digital Twins and Metaverse in the Field of Fluid Machinery Pumps and Fans: A Review. *Sensors*, *22*(23), 9294.

Yang, X., Malak, R. C., Lauer, C., Weidig, C., Hagen, H., Hamann, B., Aurich, J. C., & Kreylos, O. (2015). Manufacturing system design with virtual factory tools. *International Journal of Computer Integrated Manufacturing*, *28*(1), 25–40.

Yildiz, E., Møller, C., & Bilberg, A. (2020). Virtual factory: digital twin based integrated factory simulations. *Procedia CIRP*, *93*, 216–221.

Yli-Ojanperä, M., Sierla, S., Papakonstantinou, N., & Vyatkin, V. (2019). Adapting an agile manufacturing concept to the reference architecture model industry 4.0: A survey and case study. *Journal of Industrial Information Integration*, *15*, 147–160.

Zhang, H., Qi, Q., & Tao, F. (2022). A multi-scale modeling method for digital twin shop-floor. *Journal of Manufacturing Systems*, *62*, 417–428. https://doi.org/10.1016/j.jmsy.2021.12.011

Zheng, Z., Li, T [Tan], Li, B., Chai, X., Song, W., Chen, N., Zhou, Y., Lin, Y., & Li, R. (2022). Industrial Metaverse: Connotation, Features, Technologies, Applications and Challenges. In W. Fan, L. Zhang, N. Li, & X. Song (Eds.), *Methods and Applications for Modeling and Simulation of Complex Systems* (pp. 239–263). Springer Nature Singapore.

Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of industry 4.0: a review. *Engineering*, *3*(5), 616–630.

VII Appendix

A.1 PRISMA Step I – Identification

I. Identification

Keywords of the Intital Search

1st Field of Research	
Variable	**Keywords**
digital platform*	digital platform digital platforms
CPS	Cyber-Physical Systems

2nd Field of Research	
Variable	**Keywords**
data governance	data governance data quality
digital twin*	digital twin digital twins
simulation	simulation

Search Query before specifying Field Tags and Language
TITLE(("digital platform*" OR "CPS") AND ("data governance" OR "digital twin*" OR "simulation"))

A.2 PRISMA Step II - Screening

II. Screening

Two Databases were chosen for the systematic literature review: *SCOPUS* and *Google Scholar*

		English Language	*26.03.2023			
Database	**Link**	**Query**	**Total hits**	**Open Access**	**2017 -2023**	**Open Access & 2017 - 2023**
SCOPUS	https://www.scopus.com/	TITLE(("digital platform*" OR "CPS") AND ("data governance" OR "digital twin*" OR "simulation"))	87	25	64	23
SCOPUS	https://scholar.google.com	ALLINTITLE(("digital platform*" OR "CPS") AND ("data governance" OR "digital twin*" OR "simulation"))	76	53	58	38
		Sum:	**163**	**78**	**122**	**61**

A.3 PRISMA Step III – Decision

III. Decision

The resulting set of 61 papers are documented and further analysed according to two selection criteria:
1) Focus on topics dealing with the relationship between digital platforms, data governance (and/or quality) and digital twin technology and/or simulation
2) Relevance to understand concepts surrounding the problem field of study

Both the selection criteria and the full-text analysis (FTA) lead to a final set of 3 papers from the detailed screening.

								Selection Criteria		
Titel	**Year**	**Relevance**	**Authors**	**Journal / Publisher**	**Method**	**Abstract**	**Database**	**1**	**2**	**Full text check**
A Review of the Roles of Digital Twin in CPS-based Production Systems	**2017**	5	Negri, E., Fumagalli, L., & Macchi, M.	Procedia Manufactu	Literature Review	The Digital Twin (DT) is one of the main concepts associated to the Industry 4.0 wave. This term is more and more used in industry and research initiatives; however, the scientific literature does not provide a unique definition of this concept. The paper aims at analyzing the definitions of the DT concept in scientific literature, retracing it from the initial conceptualization in the aerospace field, to the most recent interpretations in the manufacturing domain and more specifically in Industry 4.0 and smart manufacturing research. DT provides virtual representations of systems along their lifecycle. Optimizations and decisions making would then rely on the same data that are updated in real-time with the physical system, through synchronization enabled by sensors. The paper also proposes the definition of DT for Industry 4.0 manufacturing, elaborated by the European H2020 project MAYA, as a contribution to the research discussion about DT concept.	SCOPUS	**x**	**x**	**x**
The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration	**2022**	5	Liu, Z., Sampaio, P., Pishchulov, G., Mehandjiev, N., Cisneros-Cabrera, S., Schirrmann, A., Jiru, F., & Bnouhanna, N.	Computers in Indust	Design Science	This paper presents the architectural design and implementation of DIGICOR — a collaborative Industry 4.0 (I4.0) platform aimed at enabling SMEs to dynamically form supply-chain collaborations so as to pool production capacities and capabilities and jointly address complex supply chain requests. The DIGICOR architecture builds on the event-driven service-oriented architecture (EDSOA) model to support the collaboration between SMEs, dynamic modelling of their systems and services, and their integration in the supply chains of large OEMs, enforcing digital platform governance rules for knowledge protection and security. In contrast to the extant platforms assessed through our systematic review, the proposed architecture supports the entire lifecycle of I4.0 collaborations, from creation of viable teams to deployment and operation. The architecture provides an open and extensible solution for (i) creating a marketplace for the collaboration partners, (ii) providing services for planning and controlling the collaborative production, logistics, and risk management, while supporting APIs for third parties to provide complementary services such as advanced analytics, simulation, and optimization; and (iii) seamless connectivity to automation solutions, smart objects and real-time data sources. We report on the design of the architecture and its innovative artefacts such as the component model description and the semantic model constructs created for meaningful event exchanges between architectural end-points. We also describe a running use case demonstrating implementation scenarios.	SCOPUS	**x**	**x**	**x**
Data governance in digital platforms	**2019**	5	Nokkala, T., Salmela, H., & Toivonen, J.	25th Americas Conference on Information Systems, Cancun	Literature Review	As data is becoming an increasingly valuable asset for organizations, pressure to improve its governance is increasing. Prior research provides a thorough view on data governance within a single organization. Although data sharing with partner organisations in data platforms provides a means to add data's value, empirical studies addressing data governance in shared data platforms are still few. We propose a literature based preliminary framework for platform data governance with five domains: original data quality; ownership and access; stewardship; platform data quality and value of data usage. We report experiences from a European shipbuilding network, which is seeking ways to share data collected during ship's operations. While the device manufacturers already process sensor data within their device-specific platforms, they are knowledgeable about potential benefits of sharing their data with others in the network. Based on interviews within the shipyard network, we report views along the five platform data governance domains.	Google Scholar	**x**	**x**	**x**

A.4 PRISMA Step III – Decision (Forwards/Backwards)

III. Decision - Forwards/Backwards Search

The resulting set of 3 papers is documented and further analysed according through a fowards and backwards search featuring the following criteria:
1) Focus on topics dealing with the relationship between digital platforms, data governance (and/or quality) and digital twin technology and/or simulation
2) Relevance to understand concepts surrounding the problem field of study

This leads to a final set of **11** papers from the forwards and backwards analysis.

Titel	Year	Relevance	Authors	Journal / Publisher	Method	Abstract	Identification2	1	2	Full text check
Life Cycle Simulation for the design of Product–Service Systems	**2012**	5	Garetti, Marco; Rosa, Paolo; Terzi, Sergio	Computers in Industry, 63	Literature Review	The present paper discusses Life Cycle Simulation (LCS) as a new approach for supporting the design of Product–Service Systems. The increased relevance	Backwards Search	x	x	x
The digital twin paradigm for future NASA and US Air Force vehicles	**2012**	5	Glaessgen, E., & Stargel, D.	Paper for the 53rd Structures, Structural Dynamics, and Materials Conference: Special Session on the Digital Twin	Literature Review	Future generations of NASA and U.S. Air Force vehicles will require lighter mass while being subjected to higher loads and more extreme service conditions	Backwards Search	x	x	x
About the importance of autonomy and digital twins for the future of manufacturing	**2015**	5	Rosen, R., Wichert, G. von, Lo, G., & Bettenhausen, K. D.	IFAC-PapersOnLine, 48(3)	Literature Review	Industrie 4.0 - the "brand" name of the German initiative driving the future of manufacturing - is one of several initiatives around the globe emphasizing the	Backwards Search	x	x	x
Digital twin data modeling with automationml and a communication methodology for data exchange	**2016**	5	Schroeder, G. N., Steinmetz, C., Pereira, C. E., & Espindola, D. B.	IFAC-PapersOnLine, 49(30)	Design Science	In the context of the Cyber Physical Systems toward the realization of a Digital Twin system for future manufacturing and product service systems we	Backwards Search	x	x	x
VFF: virtual factory framework	**2010**	5	Sacco, M., Pedrazzoli, P., & Terkaj, W.	2010 IEEE International Technology Management Conference (ICE)	Literature Review	The current complex market highlights the need of software tools supporting product engineering and manufacturing during the various stages of	Backwards Search	x	x	x
Designing data governance	**2010**	5	Khatri, V., & Brown, C. V.	Communications of the ACM	Literature Review	Organizations are becoming increasingly serious about the notion of "data as an asset" as they face increasing pressure for reporting a "single version of the	Backwards Search	x	x	x
Research commentary—Platform evolution: Coevolution of platform architecture, governance, and environmental	**2010**	5	Tiwana, A., Konsynski, B., & Bush, A. A.	Information Systems Research, 21(4)	Literature Review	The emergence of software-based platforms is shifting competition toward platform-centric ecosystems, although this phenomenon has not received	Backwards Search	x	x	x
Data governance for platform ecosystems: Critical factors and the state of practice	**2017**	5	Lee, S. U., Zhu, L., & Jeffery, R.	21st Pacific Asia Conference on Information Systems, Langkawi	Literature Review	Recently, "platform ecosystem" has received attention as a key business concept. Sustainable growth of platform ecosystems is enabled by platform users	Backwards Search	x	x	x
Digital Twin in manufacturing: A categorical literature review and classification	**2018**	5	Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W.	IFAC-PapersOnLine, 51(11)	Literature Review	The Digital Twin (DT) is commonly known as a key enabler for the digital transformation, however, in literature is no common understanding	Forwards Search	x	x	x
A multi-scale modeling method for digital twin shop-floor	**2022**	5	Zhang, H., Qi, Q., & Tao, F.	Journal of Manufacturing Systems, 62	Design Science	Digital twin has attracted more and more attentions in the past few years. To put digital twin into practice, modeling is one of the most important foundations.	Forwards Search	x	x	x
Application of IoT-aided simulation to manufacturing systems in cyber-physical system	**2019**	5	Tan, Y., Yang, W., Yoshida, K., & Takakuwa, S.	Machines, 7(1)	Design Science	With the rapid development of mobile and wireless networking technologies, data has become more ubiquitous and the IoT (Internet of Things) is attracting	Forwards Search	x	x	x

A.5 Interview Questionnaire

Technical Expert Interview - Protocol

1. **Research Question and Goal of the Interview (1 min)**

   How can quality insufficiencies of spatial data digital platforms automatically be identified to improve industrial metaverse applications in the automotive industry?

   → List of all existing data quality barriers

   → Prioritisation of the barriers using a Failure Modes and Effects and Analysis

2. **Introduction (2 min)**

- Explain Situation:
  - Roll-out of *Nvidia Omniverse* is in an early phase
  - Production pressure is high and factory planning mistakes are costly
    - ➔ Rendering of spatial factory data must be fast and flawless
    - ➔ Data quality must be ensured
  - **BUT:** Only manual data quality assessments of spatial data possible
  - Benefits of platform outweighed by costs for manual data scans & human margin of error
- Explain Scope and Duration of Interview

3. **Introductory Questions (2 min)**

- What is your job function respectively role within the *Nvidia Omniverse* roll-out project?
- How long have you been working in this position?

4. **Key Questions (10 - 15 min)**

- What data quality insufficiencies do you encounter when visualising the virtual factory in one of the apps from the *Nvidia* toolkit?

For each of these insufficiencies:

- How would you rate the severity of this insufficiency on the following scale?

Effect	Criterion	Rating
Very Strong Negative Effect on Software Functionality	Basic functions fail - Platform lock-up or irresponsiveness	10
		9
Strong Negative Effect on Software Functionality	User experiences serious functionality loss/restrictions	8
		7
Moderate Negative Effect on Software Functionality	User perceives annyoing performance degredation	6
		5
Weak Negative Effect on Software Functionality	User perceives but is not annoyed by failure mode	4
		3
Very Weak Negative Effect on Software Functionality	User does not perceive failure or is only insignificantly affected	2
		1

- How would you rate the occurrence of this insufficiency on the following scale?

Probability of Occurrence	Criterion	Rating
Very High	1:2	10
	1:10	9
High	1:20	8
	1:100	7
Moderate	1:200	6
	1:1000	5
Low	1:2000	4
	1:10,000	3
Very Low	1:20,000	2
	<20,000	1

- How would you rate the detectability of this insufficiency on the following scale?

Control = Manual Inspection	Detection	Criterion	Rating
Manual Inspection	Almost Impossible	Non-detectable or not checked	10
Manual Inspection	Very Low	Control achieved with indirect or random checks	9
Manual Inspection	Low	Control achieved with thorough visual inspection only	8
Manual Inspection	Moderately Low	Control achieved with multiple visual checks only	7
Not applicable	Moderate	Control achieved with charting methods, such as statistical process control	6
Not applicable		Control based on variable gauging after delivery or 100% go-no go gauging after	5
Not applicable	Moderately High	Error detection in subsequent operations; gauging performed at service delivery	4
Not applicable	High	Error detection at delivery or in subsequent operations	3
Not applicable	Very High	Error detection in delivery, such as automatic gauging with automatic stop feature; failure	2
Not applicable	Almost Certain	Error proofed–failure cannot occur	1

5. Summary and Outlook

- Information regarding processing of collected data

A.6 Interview Summaries

Monday, 3rd April 2023 – Virtual Factory Software Rollout: Visualisation Lead

1. **Positional Deviations of Selected 3D Assets**
    a. Severity: 8
    b. Occurrence: 9
    c. Detection: 7
2. **Performance Error**
    a. Severity: 9
    b. Occurrence: 8
    c. Detection: 9
3. **Missing Geometries**
    a. Severity: 6
    b. Occurrence: 7
    c. Detection: 5
4. **Wrong Scaling**
    a. Severity: 10
    b. Occurrence: 8
    c. Detection: 8
5. **Empty File**
    a. Detection: 7
    b. Occurrence: 7
    c. Severity: 8
6. **Violation of Naming Convention**
    a. Severity: 4
    b. Occurrence: 6
    c. Detection:9
7. **Redundant Geometries**
    a. Severity: 4
    b. Occurrence: 6
    c. Detection: 7
8. **Additional Geometries**
    a. Detection: 7
    b. Occurrence: 7
    c. Severity: 5
9. **Positional Error of Entire Dataset**
    a. Detection: 5
    b. Occurrence: 7
    c. Severity: 5
10. **Mesh Boolean Error**
    a. Severity: 3
    b. Occurrence: 7
    c. Detection: 7
11. **Unresolvable Origin Path**
    a. Severity: 8
    b. Occurrence: 6
    c. Detection: 7
12. **0-Byte-/Versioned-File**
    a. Detection: 7
    b. Occurrence: 2
    c. Severity: 10
13. **Material Visualisation Deviations**
    a. Detection: 7
    b. Occurrence: 4
    c. Severity: 4
14. **Deviation From Source System**
    a. Detection: 6
    b. Occurrence: 4
    c. Severity: 4
15. **2D Data**
    a. Severity: 10
    b. Occurrence: 8
    c. Detection: 6

16. **Kind Problem**
    a. Severity: 3
    b. Occurrence: 9
    c. Detection: 9
17. **Z-Fighting**
    a. Detection: 8
    b. Occurrence: 9
    c. Severity: 3
18. **Twisted Geometries**
    a. Detection: 4
    b. Occurrence: 7
    c. Severity: 5

Tuesday, 4th April 2023 –Virtual Factory Software Rollout

1. **Positional Error of Selected Assets**
    a. Detection: 9
    b. Severity: 7
    c. Occurence: 8
2. **Performance Error**
    a. Detection: 8
    b. Severity: 7
    c. Occurence: 8
3. **Positional Error of Entire Dataset**
    a. Severity: 5
    b. Occurrence: 7
    c. Detection: 4
4. **Missing Geometries**
    a. Severity: 7
    b. Occurrence: 6
    c. Detection: 5
5. **Deviation From Source System**
    a. Severity: 2
    b. Occurrence: 6
    c. Detection: 6
6. **Additional Geometries**
    a. Severity: 6
    b. Occurrence: 8
    c. Detection: 7
7. **Z-Fighting**
    a. Detection: 7
    b. Occurrence: 9
    c. Severity: 4

Tuesday, 4th April 2023 –Virtual Factory Data Management Plant A

1. **Positional Error of Selected Assets**
    a. Detection: 7
    b. Severity: 8
    c. Occurence: 8-9, due to new workflow

2. **Performance Error**
    a. Detection: 9
    b. Severity: 8
    c. Occurence: 7

3. **Missing Geometries**
    a. Detection: 7
    b. Severity: 7
    c. Occurence: 7

4. **Z-Fighting**

a. Occurrence: 10
b. Detection: 8
c. Severity: 3

Wednesday, 5th April 2023 – Virtual Factory Data Management Plant D

1. **Redundant Geometries**
   a. Detection: 7
   b. Severity: 5
   c. Occurence: 4

2. **Additional Geometries**
   a. Detection: 7
   b. Severity: 5
   c. Occurrence: 8

3. **Wrong Scaling**
   a. Severity: 7
   b. Occurrence: 4
   c. Detection: 7

Tuesday, 2nd May 2023 – Virtual Factory Software Rollout: Data Provisioning Lead

1. **Redundant Data**
   - Detection: 7
   - Severity: 4 - 5
   - Occurence: 6/7
2. **Positional Error of Selected Assets**
   - Detection: 10
   - Severity: 6
   - Occurence: 8
3. **Performance Error**
   - Detection: 10
   - Severity: 7
   - Occurence: 7

Tuesday, 2nd May 2023 –Virtual Factory Software Rollout: Plant Built-Up/Coordination

1. **Positional Error of Selected Assets**
   a. Detection: 8
   b. Severity: 8
   c. Occurence: 9-

2. **Violation of Naming Convention**
   a. Detection: 10
   b. Severity: 5
   c. Occurence: 7

3. **Empty File**
   a. Detection: 8
   b. Severity: 9
   c. Occurence: 9

4. **Additional Geometries (Halbkugel wird mit angezeigt, Standardbezeichnungen)**
   a. Detection: 7
   b. Severity: 5
   c. Occurence: 6

5. **Performance Error**
   a. Detection: 9
   b. Severity: 10

    c. Occurence: 7

6. **0-Byte-/ Versioned File**
    a. Detection: 7
    b. Severity: 10
    c. Occurence: 2

7. **Unresolvable Origin Path**
    a. Detection: 7
    b. Severity: 7
    c. Occurence: 4

8. **Scaling**
    a. Detection: 8
    b. Severity: 6
    c. Occurence: 5

9. **2D Data Within 3D Environment**
    a. Detection: 8
    b. Severity: 10
    c. Occurence: 6

Free Discussion

- Root Causes: Workflow, Missing Standard Source Systems, Conversion
- Completeness:
    - Missing Geometries
        - Causes: Workflow (intentionally or accidentally hidden levels?), Missing standards source system (MicroStation)
        - Fix: Training, Standard in source system: Comparison 3D elements source system ←→ Omniverse applications
- Positional:
    - Wrong Position – Coordinates of whole dataset
        - Causes: Workflow
    - Positional Error of Selected Assets
        - Causes: Workflow
        - Fix: Training
    - Scaling Error
        - Causes: Workflow, Wrong translation American → Metric system
    - Rotation Error
- Lineage
    - Performance Error
        - Causes: Workflow
        - Fix: Training

A.7 FMEA Evaluation Table

Category	Name	Failure Mode(s)	Effect(s) of Failure	Severity Average	Potential Cause(s)/ Mechanism(s) of Failure	Cause Details	Occurrence Average	Current Design Controls	Detection Average	Occurrence X Detection	RPN
Lineage (Qualitative)	Kind Problem	Single mesh files declared as kind "subcomponent" instead of "component"	Suffering user experience	3	**Flawed Conversion**		10	Manual Quality Check	8	**80**	**240**
Geometric accuracy (Quantitative)	Positional Error of Selected 3D Assets	Positional deviations real/virtual factory	Layout Planning Flawed/Impossible	7	**Workflow**		9	Manual Quality Check	8	**75**	**513**
Semantic consistency (Quantitative)	Violation of Naming Convention	Prim names violate defined naming convention	Layout Planning Flawed/Impossible	5	**Workflow**	False labelling	7	Manual Quality Check	10	**70**	**350**
Geometric accuracy (Quantitative)	Z-Fighting	Z-Fighting of assets with the same z-value	UX/Performance suffers	3	-	Conversion of non-existent layers	9	Manual Quality Check	7	**63**	**210**
Semantic accuracy (Quantitative)	Performance Error	Number of mesh prims > 50,000	UX/Performance suffers	8	**Workflow**	Individuals do not reference correctly when providing data in source systems.	7	Manual Quality Check	9	**59**	**483**
Completeness (Quantitative)	Empty File	File has no content	Inability to visualise asset in virtual factory	9	**Missing Source System Standards**	Flawed DGN file in Microstation causes inability of Converter to convert the file to USD	7	Manual Quality Check	8	**53**	**473**
Logical Consisteny (Quantitative)	2D Data Within 3D Environment	Appearance of 2D data in 3D environment	Inability to visualise virtual factory	10	**Workflow**	Microstation files created in 2D	7	Manual Quality Check	7	**49**	**490**
Semantic accuracy (Quantitative)	Mesh Boolean Error	Files from library are badly visualised	UX/Performance suffers	4	**Missing Source System Standards**	Insufficient data quality in Microstation source library ("Digitaler Planungstisch")	7	Manual Quality Check	7	**49**	**196**
Completeness (Quantitative)	Additional Geometries	Interfering contours are permanently visualised	Layout Planning Flawed/Impossible	5	**Workflow**	Geometries that are not to be visualised are not moved into default layer	7	Manual Quality Check	7	**47**	**236**
Geometric accuracy (Quantitative)	Wrong Scaling	Geometrically flawed proportions of assets	Layout Planning Flawed/Impossible	7	**Workflow**	Microstation file falsely sized & Conversion metrical/imperial system fails	5	Manual Quality Check	8	**44**	**296**
Completeness (Quantitative)	Redundant Geometries	Identical geometries appear twice (or more)	Layout Planning Flawed/Impossible	5	**Workflow**		5	Manual Quality Check	7	**37**	**187**
Completeness (Quantitative)	Missing Geometries	Factory assets are not visualised	Layout Planning Flawed/Impossible	7	1. **Workflow** 2. **Missing Source System Standards**	User accidentally moves geometries to visualise into default layer	7	Manual Quality Check	5	**36**	**237**
Lineage (Qualitative)	Unresolvable Origin Path	Unresolvable heritage path of Nucleus files	Inability to visualise asset in virtual factory	9	**Flawed Conversion**		5	Manual Quality Check	7	**35**	**315**
Geometric accuracy (Quantitative)	Twisted Geometries	Buildings have an angular tilt in virtual factory	Layout Planning Flawed/Impossible	6	**Workflow**		8	Manual Quality Check	4	**32**	**192**
Geometric accuracy (Quantitative)	Positional Error of Entire Dataset	Entire dataset has flawed coordinates	Layout Planning Flawed/Impossible	6	**Workflow**	Differing coordinate systems & Assets converted in "geo" instead of "plant position"	8	Manual Quality Check	4	**30**	**165**
Logical Consisteny (Quantitative)	Deviation From Source System	Conversion changes prim composition	Layout Planning Flawed/Impossible	3	**Flawed Conversion**		5	Manual Quality Check	6	**30**	**90**
Semantic accuracy (Quantitative)	Material Visualisation Deviations	Material is falsely visualised or deviates from reality	UX/Performance suffers	4	?		4	Manual Quality Check	7	**28**	**112**
Lineage (Qualitative)	0-Byte-/ Versioned File	New prim version created, but old version of the file is still referenced	Inability to visualise virtual factory	10	**Workflow**	Converter unable to process new file versions if relabelled	3	Manual Quality Check	7	**21**	**210**

A.8 Extension Repository READ.ME-File

Nvidia Omniverse Data Quality Gate

This project contains the source code to an Nvidia Omniverse extension capable of flawlessly identifying the six most critical data quality insufficiencies within the 3D environment of an automitve manufacturer's virtual factory models, perform an applicable analysis of the data obtained from this identification and derive concrete KPIs.

Project Description

Problem: Data quality assessments of virtual factory models visualised in the Nvidia Omniverse 3D environment have to be conducted fully manually. This procedure is time-consuming and prone to errors, causing the benefits of the Nvidia Omniverse in the automotive manufacturing sector to be outweighed by the costs of ensuring sufficient data quality.

Objective: This software extension to the Nvidia Omniverse is capable of automatically identifying the most critical insufficiencies within a virtual factory model and performing an analysis of the results to compute adequate KPIs. Based on these KPIs, minimum data quality requirements are to be defined that need to be met by virtual factory models coming from the Factory Assembler to be published for visualisation in the Nvidia Omniverse. Thereby, a data quality gate is created that ensures a certain quality standard of data in the Nvidia Omniverse.

Get Started

1. In the Omniverse App open extension manager: Window → Extensions.
2. In the Extension Manager Window open a settings page, with a small gear button in the top left bar.
3. In the settings page there is a list of Extension Search Paths. Add file path to `exts` subfolder there as another search path
4. Now you can find omniverse.quality.gate extension in the top left search bar. Select and enable it.
5. "Nvidia Omniverse Data Quality Gate" window will pop up in bottom right corner. Extension Manager watches for any file changes and updates extension following a hotreload.

Table of Contents

All relevant code is part of the `exts` -folder. The content of any other folder is provided by Nvidia and not created by the software extension creator. The only relevant files created or edited as part of this project are detailed below:

Key Source Code

The relevant functionalities are stored in: \`exts/omniverse.quality.gate/omniverse/quality/gate´

Functionalities:

`quality_checks.py` : Identification functions for each of the six data quality insufficiency checks (+ relevant reoccuring functions enbaling these checks)

`ui_window.py` : All UI specifications including window set-up and automatic data analysis to create KPI dashboard

`draw_util.py` : Function to draw bounding box for Position-Check-Light

`extension.py` : Basic functions to enable start-up and destroyment of extension window in Nvidia Omniverse + get viewport specifications for bounding box drawing in Position-Check-Light

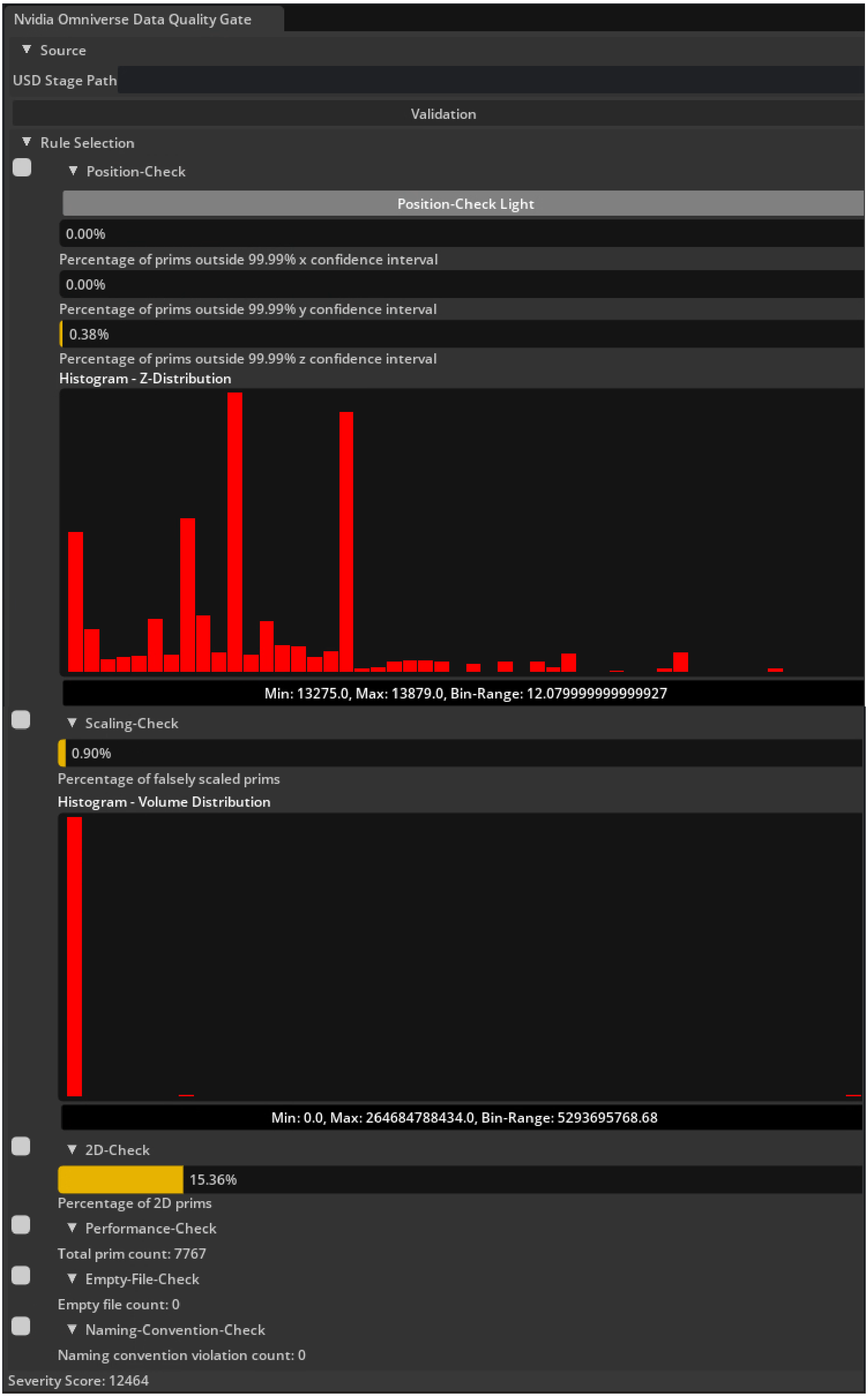
Technical Information

`extension.toml` under the file path `exts/omniverse.quality.gate/config/` contains relevant metadata, such as the project author, title, version, description and logo

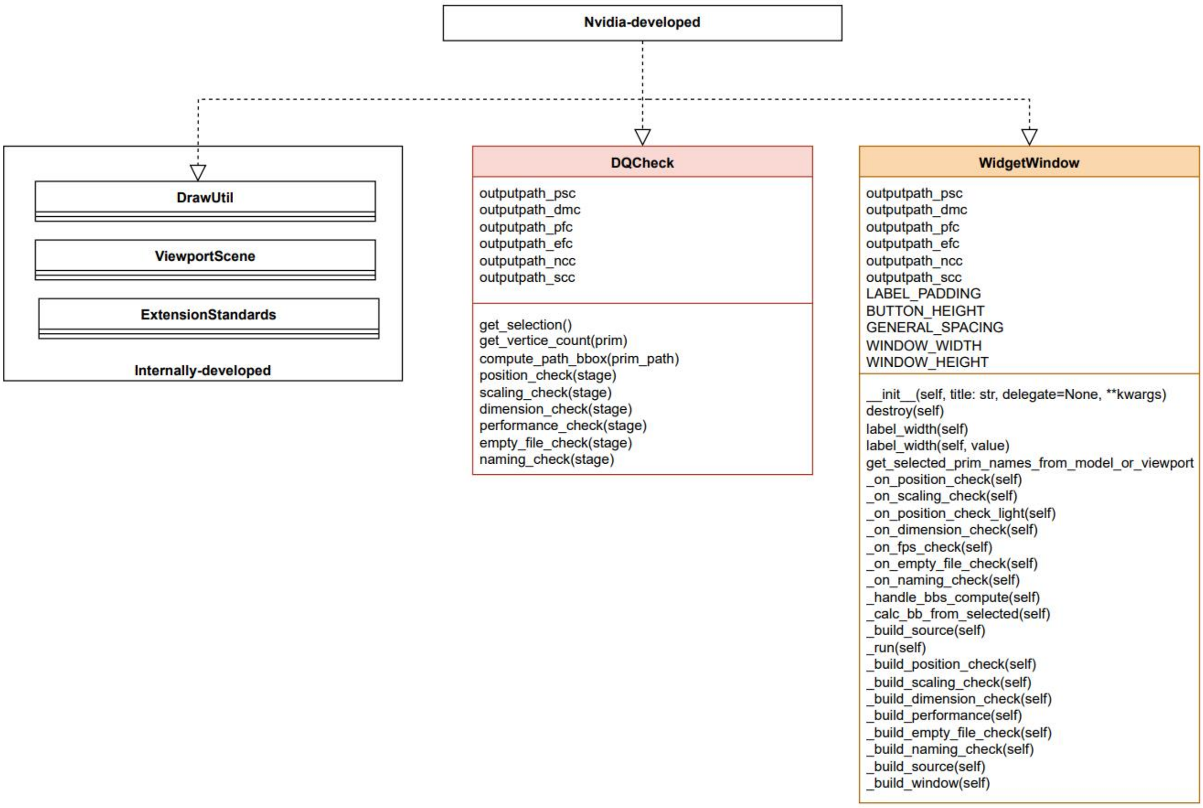
A.9 Test Results

Data Set	Size	Data Quality Insufficiencies						
		Position	Sca-ling	2D	Perfor-mance	Empty-File	Naming	Severity
1	825,325 Prims	x: 0% y: 0% z: 0%	2.47%	15.51%	Yes	0	35	8024835
2	203,564 Prims	x: 0.9% y: 0.3% z: 3.1%	5.36%	15.52%	Yes	12	0	203564
3	258,026 Prims	x: 0% y: 0% z: 0.31%	1.9%	23.98%	Yes	0	0	2717099
4	38,308 Prims	x: 0% y: 0% z: 0%	0.04%	23.24%	No	4	1	89832
5	13,364 Prims	x: 0% y: 0.03% z: 0.78%	0.52%	6.2%	No	0	0	8842
6	1,387 Prims	x: 0.34% y: 0% z: 1.36%	2.16%	2.02%	No	0	3	525
7	10,831 Prims	x: 0% y: 0% z: 0%	3.41%	0.07%	No	1	0	766
8	1,374 Prims	x: 0% y: 0% z: 0%	2.04%	9.02%	No	0	0	1436

A.10 Exhaustive KPI Dashboard



A.11 Simplified Extension UML Diagram



A.12 Duration Comparison of Manual and Automatic Quality Checks

Model	**Check**	**% of Identified Data Quality Insufficiencies**						
		Position	Scaling	2D	Perfor-mance	Empty-File	Naming	Total Time
#1	Manual							14,400 sec
	Auto	56 sec	122 sec	102 sec	24 sec	23 sec	23 sec	350 sec
#2	Manual							3600 sec
	Auto	7 sec	30 sec	22 sec	6 sec	6 sec	6 sec	77 sec
#3	Manual							5400 sec
	Auto	12 sec	38 sec	30 sec	8 sec	8 sec	8 sec	104 sec
#4	Manual							1800 sec
	Auto	2 sec	5 sec	5 sec	2 sec	2 sec	2 sec	18 sec
#5	Manual							1800 sec
	Auto	4 sec	6 sec	5 sec	2 sec	2 sec	2 sec	21 sec
#6	Manual							1800 sec
	Auto	2 sec	6 sec	5 sec	2 sec	2 sec	2 sec	19 sec
#7	Manual							2700 sec
	Auto	1 sec	3 sec	3 sec	2 sec	1 sec	2 sec	12 sec
#8	Manual							900 sec
	Auto	1 sec	2 sec	2 sec	1 sec	1 sec	1 sec	8 sec