Sensor Interfacing Language for FAIR Sensor Services in Manufacturing

FAIRe Sensordienste in der Produktion auf Basis einer Schnittstellenbeschreibungssprache

Von der Fakultät für Maschinenwesen
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von

Matthias Stefan Bodenbenner

Berichter:

Univ.-Prof. Dr.-Ing. Robert Schmitt
Prof. Dr. rer. pol. York Sure-Vetter

Tag der mündlichen Prüfung: 28.03.2025

Acknowledgement

It’s the job that’s never started as takes longest to finish.

Hamfast "The Gaffer" Gamgee
J.R.R. Tolkien

This thesis represents the quintessence of my work and research at the WZL of RWTH Aachen University, particularly at the Chair of Intelligence in Quality Sensing (IQS). Being relieved and happy to finally finish my dissertation while writing these words, I am very thankful of all the support and supervision during my studies for this thesis and beyond.

Firstly, I would like to thank Prof. Robert Schmitt for his supervision of this thesis, the countless possibilities while working at IQS and his enthusiasm for the research we all do at IQS. I would like to thank Prof. York Sure-Vetter for his great interest in my work and his offer to review my doctoral thesis.

The major research for my thesis have been undertaken within the strong networks of the excellence cluster *Internet of Production* and the *National Research Data Infrastructure for the Engineering Sciences* (NFDI4Ing) which gave me great pleasure and many wonderful memories. This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC-2023 Internet of Production - 390621612 and under Project-ID 432233186 - AIMS.

Special thanks go to my supervisor of my Master’s thesis, colleague, group leader and chief engineer Benjamin for inspiring me to do a doctorate at the WZL and paving the way for me to complete my work successfully. Many thanks to my colleague and friend Mark for countless discussions on the content and his feedback to my thesis. I need to thank Dominik, Dominik, Kilian, Mario, Philipp, and all other colleagues of IQS, for their feedback to and great interest in my work. This has motivated me a lot over the last few years (and still does).

At this point, I would like to acknowledge the contribution of all the students who helped me with the research for this thesis. In particular, Julian and André, who implemented the first prototypes of my ideas in their bachelor theses and who reliably and stoically implemented my countless plans and change requests during their time as student assistants. I would also like to thank Nils and Tim for their excellent work, which laid the foundation for my doctoral presentation.

I would like to express my gratitude to my family for their life-long backing in all concerns. Especially, my parents Hiltrud and Andreas for a peace- and joyful childhood, their encouragement to always strive for the best and for their optimism that everything is possible if you just try hard enough. Moreover, my sister Miriam for being the best sister I could ever imagine and in particular for the small surprises in the very last weeks of this process making me smile every day. I thank my parents-in-law Jutta und Ralf for their support and providing me with a productive environment especially on stressful weekends during my dissertation process. I am very grateful to my best friends Patrick and Dominik for many unforgettable vacations, game days and evenings - of which I hope many more will follow - that reliably made me forget all the stress and that every disagreement is always short-lived.

Most of all I am grateful for the immeasurable support and understanding of my wife Lara. For her patience and sacrifice, especially during the writing of this thesis, I can never thank her enough. I'm looking forward to our future together as a family, and now I can face it with great relief and all my heart.

Matthias Bodenbenner

Aachen, April 2025

Abstract

Data-driven applications are the key technology within *Industry 4.0* to tackle today's most imminent challenges in manufacturing and production, such as sustainable production, massive automation, and resilience against disruptions. The challenges mentioned require an extended usage horizon of the sensor data to months or years, while acquired data is almost immediately consumed in current scenarios. This implies additional complexity to the documentation and management of acquired data because metainformation about the data provenance must be comprehensibly retained to be (re-)usable at a later point in time (by third parties). Due to insufficient data management, these requirements are currently not matched in manufacturing, so sensor data has a large unused value-creation potential.

To leverage the value-creation potential, this thesis proposes *FAIR Sensor Services*: a cyber-physical measuring system that adopts the *FAIR Guiding Principles* for data management, facilitating findable, accessible, interoperable, and reusable sensor data. *FAIR Sensor Services* measure and provide richly and comprehensively described measurement data, boosting long-term reusability and significantly increasing the value-creation potential. Furthermore, this thesis introduces the *Sensor Interfacing Language* (SOIL) to streamline the implementation of a *FAIR Sensor Service* and bridge the gap to the required methods from computer science.

The course of research follows Design Science Research in a three-step approach. First, the requirements promoted by the *FAIR Guiding Principles* are interpreted for the application domain and translated into 17 specific *FAIRness* criteria for that domain. Moreover, a data life-cycle is derived as a reference frame for implementing the defined criteria. Based on these criteria, object-oriented, syntactic data structures and graph-based, semantic data models are developed in the second step to represent sensor data and preserve their provenance. The semantic data model ensures full compliance with *FAIR Guiding Principles* and machine-actionability but bears high complexity. In contrast, the syntactic data structure trades in expressiveness with simplicity but enables low-barrier usage for non-data stewards. The latter is exploited in the third step of research, where a novel domain-specific modeling language - called the *Sensor Interfacing Language* - is proposed to simplify, operationalize, and streamline the definition of the (*FAIR*) semantic data models and development of *FAIR Sensor Services* based on model-driven software engineering.

To validate the developed artifacts, simulated and physical measuring systems at the WZL are modeled with SOIL and implemented as *FAIR Sensor Service*. The acquired data is evaluated for their *FAIRness* using available, automated assessment tools and reveals significant improvement compared to the state-of-the-art. The demonstration in a close-to-industry scenario proves applicability for the manufacturing domain.

Datengetriebene Anwendungen sind die Schlüsseltechnologie von *Industrie 4.0*, um die anstehenden Herausforderungen, wie beispielsweise nachhaltige Produktion, umfassende Automatisierung und Resilienz gegen Störungen, in der Fertigung und Produktion zu bewältigen. Während erfasste Daten in aktuellen Szenarien fast unmittelbar verarbeitet und genutzt werden, verschiebt sich der Nutzungshorizont der Daten zur Bewältigung der genannten Herausforderungen auf Monate oder Jahre. Dies führt zu einer zusätzlichen Komplexität bei der Dokumentation und Verwaltung der erfassten Daten, da Metainformationen über die Datenherkunft nachvollziehbar aufbewahrt werden müssen, um zu einem späteren Zeitpunkt (durch Dritte) (wieder) nutzbar zu sein. Da diese Anforderungen in der Fertigung aufgrund eines unzureichenden Datenmanagements derzeit nicht erfüllt werden, weisen Sensordaten ein großes ungenutztes Wertschöpfungspotenzial auf.

Um dieses Wertschöpfungspotenzial zu heben, führt diese Arbeit *FAIR Sensor Services* ein: ein cyber-physikalisches Messsystem, welches die *FAIR Guiding Principles* aufgreift und auffindbare, zugängliche, interoperable und nachnutzbare Sensordaten bereitstellt. *FAIR Sensor Services* messen und liefern reichhaltig und umfassend beschriebene Messdaten, was die langfristige Nachnutzbarkeit fördert und das Wertschöpfungspotenzial deutlich erhöht. Darüber hinaus wird in dieser Arbeit die *Sensor Interfacing Language* (SOIL) eingeführt, um die Implementierung eines *FAIR Sensor Services* zu vereinfachen und die Lücke zur Anwendung der erforderlichen Methoden aus der Informatik zu schließen.

Die Forschung folgt Design Science Research in einem dreistufigen Ansatz. Zunächst werden die von den *FAIR Guiding Principles* propagierten Anforderungen im Kontext der Anwendungsdomäne interpretiert und in 17 spezifische *FAIRness*-Kriterien für diese Domäne übersetzt. Darüber hinaus wird ein Datenlebenszyklus als Referenzrahmen für die Umsetzung der definierten Kriterien abgeleitet. Basierend auf diesen Kriterien werden im zweiten Schritt objektorientierte, syntaktische Datenstrukturen und Graph-basierte, semantische Datenmodelle entwickelt, um Sensordaten zu repräsentieren und ihre Provenienz zu erhalten. Das semantische Datenmodell gewährleistet die vollständige Einhaltung von der *FAIR Guiding Principles* und Maschinenlesbarkeit, ist aber mit hoher Komplexität verbunden. Im Gegensatz dazu bietet die syntaktische Datenstruktur eine hohe Ausdruckskraft bei gleichzeitiger Einfachheit, wodurch sie auch für Nicht-Data-Stewards nutzbar ist. Letzteres wird im dritten Forschungsschritt ausgenutzt, in dem eine neuartige Domänen-spezifische Beschreibungssprache - genannt *Sensor Interfacing Language* - vorgeschlagen wird, um die Definition der (*FAIRen*) semantischen Datenmodelle und die Entwicklung von *FAIR Sensor Services* basierend auf Modell-basierter Softwareentwicklung zu vereinfachen, zu operationalisieren und zu vereinfachen.

Um die entwickelten Artefakte zu validieren, werden simulierte und physikalische Messsysteme am WZL mit SOIL modelliert und als *FAIR Sensor Service* implementiert. Die gewonnenen Daten werden mit verfügbarer, automatisierter Evaluierungssoftware auf ihre *FAIRness* hin bewertet und zeigen eine deutliche Verbesserung gegenüber dem Stand der Technik. Die Demonstration in einem industrienahen Szenario beweist die Anwendbarkeit in der Produktion und Fertigung.

Table of Contents

List of Figures

List of Tables

List of Listings

1 Introduction to *FAIR manufacturing sensor data*

Sensors and measuring systems are one of the most important data sources for optimizing manufacturing and assembly processes [Owai31; John06; Pand19]. Analysis and utilization of this sensor data in data-driven applications form the foundation for all knowledge generation and exploration activities of the fourth industrial revolution *Industry 4.0* [Sagg18]. Therefore, data-driven applications based on sensor data are the key technology to tackle today's most imminent challenges in manufacturing, such as sustainable production to fight harm of the environment [Ma20; Maje21; Kuma21], massive automation to cope with demographical changes [Hild20; Acem22; Göpp23; Horv19], and to achieve resilience against (market) disruptions [Cava19; Bech22; Sark20]. Considering the rapidly growing market of data-driven applications and services [Owai31; Senn23] and their significance for manufacturing [Li20; Colu14; Libe15; Forr24], sensor data can be considered the root and most valuable resource of *Industry 4.0* [Owai31; John06; Pand19].

But the data-driven applications targeting the mentioned challenges extend the time horizon for sensor data (re-)use from (milli-)seconds and minutes for process control, system monitoring, or quality control to months or even years, e.g., for life-cycle assessments [Schm23]. This extended usage time and that (sensor) data needs to be understood by others than the initial creators impose high standards on the management, documentation, and description of data [Owai31]. As these requirements are usually not matched in industry, the portion of data used beyond immediate process control is small [Kusi19; Li22]. Consequently, there is a large unused value-creation potential due to insufficient data management and a missing long-term usage perspective beyond the initial scope of the application of data. However, to prevent increased resource consumptions (such as energy and hardware) for long-term data management from becoming a limiting factor on its own, the data management itself must be sustainable, resilient, and automatable [Schm23]. This motivates the establishment of data management as a discipline on its own in *Industry 4.0* and underlines the need for low-barrier tools streamlining data acquisition and management processes.

To address the core problems of insufficient and inefficient data management and missing description of sensor data, this thesis proposes *FAIR Sensor Services*, cyber-physical measuring systems (CPMSs) that adopt the *FAIR Guiding Principles* [Wilk16] for data management. *FAIR Sensor Services* measure and provide richly and comprehensively described measurement data, leveraging long-term reusability and significantly increasing the value-creation potential of the measurement data. Furthermore, this thesis introduces the *Sensor Interfacing Language* (SOIL) to streamline the implementation of a *FAIR Sensor Service* and bridge the gap to the required methods from computer science. These two central software artifacts significantly lower the barrier to applying and deploying *FAIR data management* in the manufacturing industry, thereby delivering a significant contribution to sustainable, resilient, and automated manufacturing.

1.1 Motivation of data management for manufacturing based on the *FAIR Guiding Principles*

The manufacturing sector faces several challenges in data management, which stem from technological advancements, increased complexity, and the volume and diversity of sensor data being generated. Despite the large amount of data collected, only a small portion of the data is (re-)used [Kusi19; Beck23; Laba20] as illustrated in Figure 1.1. Data security and privacy concerns often lead to isolated data storage, which is only accessible locally, even within companies [Zhen20a]. Thus, most of the data sinks into data silos which lack accessibility so that the data can hardly be retrieved and (re-)used [Gao20; Khan16]. Moreover, the industry lacks appropriate data management systems that can handle highly heterogeneous data. In particular, brownfield plants are constituted of systems that were not designed to be interconnected but need to be integrated into a unified data infrastructure [Nils18], which is hampered by diverse interfaces, communication protocols, and data formats of the individual devices in place [Libe15]. In addition, if the acquired data can be technically exchanged by common communication protocols in a standardized format, the description of the data does not follow a common standard or is not present at all [Glei20; Gao20; Li22; Khan16]. To unlock the full value-creation potential of sensor data, it is required to make isolated data silos accessible and preserve the provenance and context in which the data was created [Glei20; Gao20; Tanh19; Isel16]. Standardization of these data models and the data exchange is needed to leverage the data (re-)use [Ye18; Kess20; Bosc16; Sawa21].

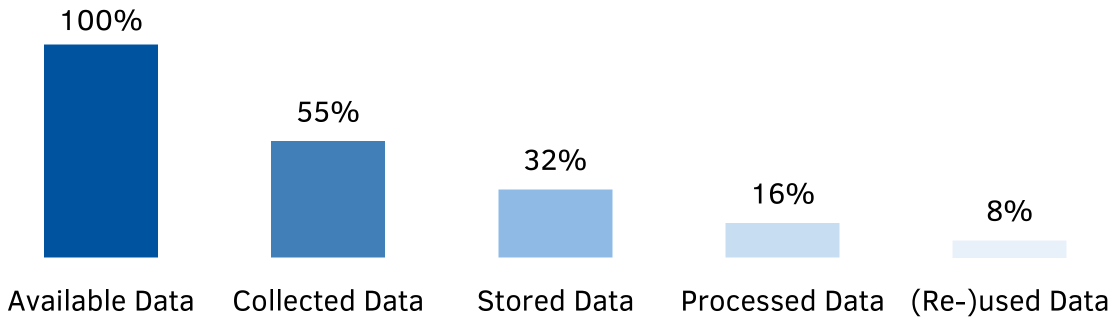


Figure 1.1: Relative amount of data usage in the industry in the DACH region obtained by an interview study by Beckschulte et al. [Beck23]. According to the authors, only 15% of the acquired data is used to carry out actions or make decisions.

Data that is insufficiently described and might be inherently but undiscoverably inconsistent can be considered lost [Tanh19] and can lead to incorrect decisions [Khan16], which motivates the establishment of a dedicated sensor data management for manufacturing. The extended usage horizon and the segregated creation and usage of sensor data require the data to be amply described and documented to persist the information of the context in which the data was created [Kess20; Miel15]. This provenance information is required to understand the data, evaluate if data can be (re-)used for a certain application, and implement the data (re-)use [Grot20; Muse22]. The presence of meaningful metadata and context information contributes significantly to the data quality [Even06; Bert11], required for the implementation of data-driven solutions, such as machine learning approaches [Kess20; Miel15; Gao20]. The extensive documentation and description of sensor data are also required to enable data sharing. Sharing data across company borders is not only needed to establish an Industrial Internet of Things (IIoT) or World-Wide Lab (WWL) [Glei20; Zhen20b; Jesc18], in which companies work jointly together on (foreign) data but also to solve the big challenges of the industry today and in future [Forr24; Otto20]. Some consider sharing data as important as producing it [Nase13; Rodr16]. Moreover, the availability of high-quality and machine-actionable sensor data is a prerequisite for enabling automation [Eich23]. Otherwise, the data can not be processed and analyzed effectively and efficiently [Wang18b], which lowers the automation potential.

The *FAIR Guiding Principles* [Wilk16] have gained increased attention in the scientific community in the last years and have been proven effective in facilitating the machine-actionability and reuse potential of data [Thom21; Sust20; Loga23; Pete22; Jaco20b]. Fostering the findability, accessibility, interoperability, and reusability of data the *FAIR Guiding Principles* leverage sustainable, responsible data management and promote machine-actionable data. The in-depth introduction of the *FAIR Guiding Principles* and discussion of their interpretation in the context of manufacturing is given in Section 2.1

Following the literature, many benefits arise from adhering to the *FAIR Guiding Principles* and designing and implementing the data management accordingly, c.f. Figure 1.2. On a global scale, it is expected that *FAIR* data has significantly increased value-adding potential due to its boosted reusability [Jaco20a] and will thereby reduce costs and save time [Rych18]. Because well-documented and richly described sensor data is more comprehensible [Wang18b; Gao20] it enables data sharing and joint data usage [Wise19; Penn19a; Cui20]. Highly interoperable data widen the scope of application of data analysis and processing tools [Cui20]. Moreover, machine-actionable sensor data based on *Semantic Web* technologies and controlled vocabularies are considered enablers for digital measurement traceability [Eich23].

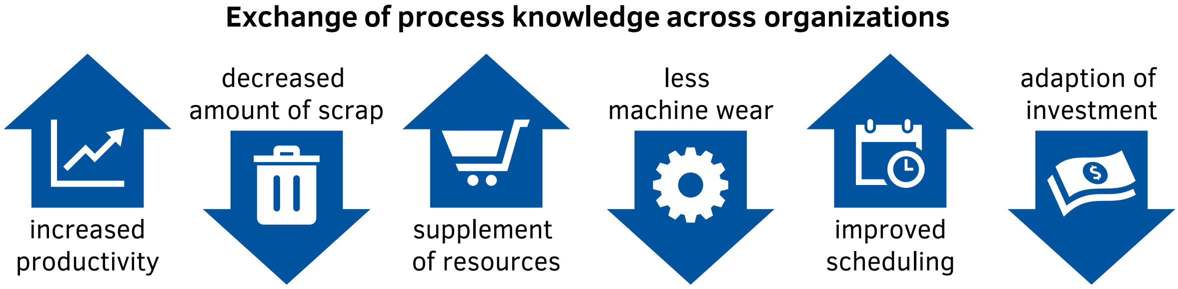


Figure 1.2: Major benefits achievable through data management adhering to the *FAIR Guiding Principles* and intensified data sharing across company boarders according to Tanhua et al. [Tanh19], Pennekamp et al. [Penn19a], Zheng and Cai [Zhen20b], and Jacobsen et al. [Jaco20a]

Consequently, data management adhering to the *FAIR Guiding Principles* is more than beneficial - it is required to solve the predominant challenges of the manufacturing industry. The key to this *FAIR data management* is the provision of high-quality and *FAIR sensor data* from the beginning. For that, it is required to implement measuring systems as cyber-physical measuring systems that can be easily integrated: the *FAIR Sensor Services* producing widely available and reusable *FAIR sensor data*.

1.2 Introduction to *FAIR Sensor Services* and the *Sensor Interfacing Language*

The infrastructure of fully interconnected factories, tailored to maximize data-based value-creation, is backed by cyber-physical systems (CPSs), joining hardware and software in a single system for providing computational intelligence at all places [Sous22b; Ribe17; Give17]. CPS are essential elements of the future industrial IT-infrastructure [Gao20; Geis12] as they are equipped with high-level communication interfaces facilitating a service-oriented architecture (SOA) breaking up monolithic infrastructures. Compared to a monolithic infrastructure, SOAs provide higher flexibility, adaptability, maintainability, and resilience because singular failures do not affect the overall system [Mont23]. These trends and benefits have been adopted by Schmitt and Voigtmann [Schm18a] proposing "Sensor Information as Service" transforming measuring systems to CPMSs as characterized in Definition 1.1 based on the CPS's definition by Geisberger and Broy [Geis12] and Ribeiro [Ribe17].

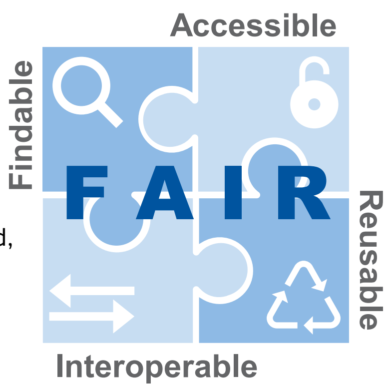
Definition 1.1: Cyber-physical measuring system

A *cyber-physical measuring system* (CPMS) unites software and hardware within a single measuring system for the physical acquisition and provision of digitized measurements. It couples physically sensed data and execution of measuring routines with computing resources (pre-)processing the sensed data and controlling the measuring routines. [Geis12; Ribe17]

Due to this servitization, many challenges are shifted from hardware to software. On the one hand, this makes it possible to develop solutions to previously unsolved problems; on the other hand, existing solutions for processes and systems must be remodeled to enable the integration and interaction of modern systems, leading to an interdisciplinarization of manufacturing. Overcoming the identified deficits requires knowledge and skills that go beyond the core competencies of traditional production engineering and measurement technology. Current developments clearly show that foundational computer science and software development concepts are important building blocks for digitalization in general and the implementation of the *FAIR Guiding Principles*, summarized in Figure 1.3.

Findable
- (meta-) data has globally unique, resolvable, and persistent identifiers
- rich machine-actionable metadata
- (meta-) data is registered or indexed in a searchable resource

Interoperable
- usage of a formal, accessible, shared, and broadly applicable language for knowledge representation
- used vocabularies follow the *FAIR Guiding Principles*
- usage of qualified references



Accessible
- clearly defined access rules
  - access protocol
  - access authorization
  - metadata longevity
- data has a universal resource locator

Reusable
- clearly defined terms and license for (re-) use
- (meta-) data is associated with detailed provenance
- (meta-) data meets domain-relevant community standards

Figure 1.3: An overview of the *FAIR Guiding Principles* proposed by Wilkinson et al. [Wilk16] to facilitate sustainable data management, increasing the value and long-term reusability of data.

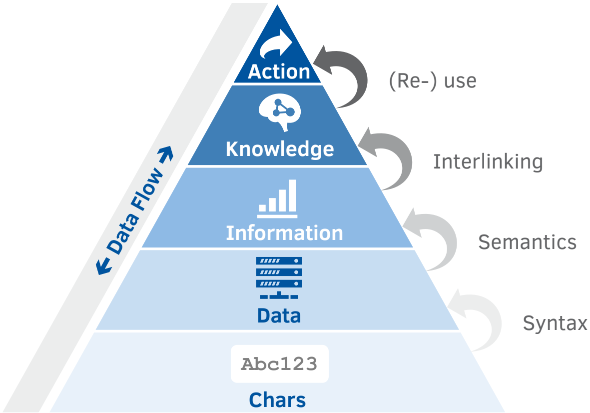


Figure 1.4: The knowledge pyramid models knowledge creation as the reference frame for data management. Illustration based on Schmitt et al. [Schm23; Schm24].

The core of these information technologies is a representation of data in appropriate ways, lifting it to information and knowledge to lastly be turned into action as formalized by the knowledge pyramid [Made23; Schm23; Schm24] depicted in Figure 1.4. The methods for data representation based on formal syntax and semantics are data structures and models, as defined in Definition 1.2 and Definition 1.3.

Definition 1.2: Data Structure

The data structure specifies the syntax for representing data about things. A data structure defines the attributes of represented things by keys naming the attributes and values mapped to these keys.

Definition 1.3: Data Model

The data model specifies the syntactic and semantic rules for representing data about things. A data model defines the meaning, attributes, and relations of represented things.

In summary, CPMSs must be enabled to yield richly described data using unified data models in standardized formats via standardized communication interfaces and protocols. Such a measuring system shall further be denoted *FAIR Sensor Service*, as specified in Definition 1.4, which is built upon a standardized, interoperable data model and uses controlled vocabulary providing the unambiguous meaning of all terms and keys used (see in-depth introduction in Section 2.4.1).

Definition 1.4: *FAIR Sensor Service*

A *FAIR Sensor Service* is a CPMS that yields the *FAIR manufacturing sensor data*. It provides measurement data to clients using unified data models covering a plurality of relevant attributes. The models are based on controlled vocabularies, transmitted using open and standardized communication protocols, and serialized in open and standardized data formats to maximize the reusability of the produced data.

Developing such a *FAIR Sensor Service* requires the modeling mentioned above and expertise from computer science, which is usually not part of the core competencies of production engineers. Therefore, this thesis proposes the *Sensor Interfacing Language* (SOIL) abstracting and encapsulating the required technologies, i.e., standardized communication protocols, data formats, and controlled vocabulary, and making them usable for production engineers and developers of measuring systems. Following the research methodology Design Science Research (DSR) [Hevn10] (cf. Section 1.4), *FAIR Sensor Services*, SOIL and software tools for the light-weight development and implementation of such are the core artifacts of this research. The envisaged main benefits and contributions of these artifacts are illustrated by Figure 1.5. Ultimately, *FAIR Sensor Services* will acquire measurement data of high *FAIRness* for the provision of *FAIR manufacturing sensor data* (cf. Definitions 1.5 to 1.6).

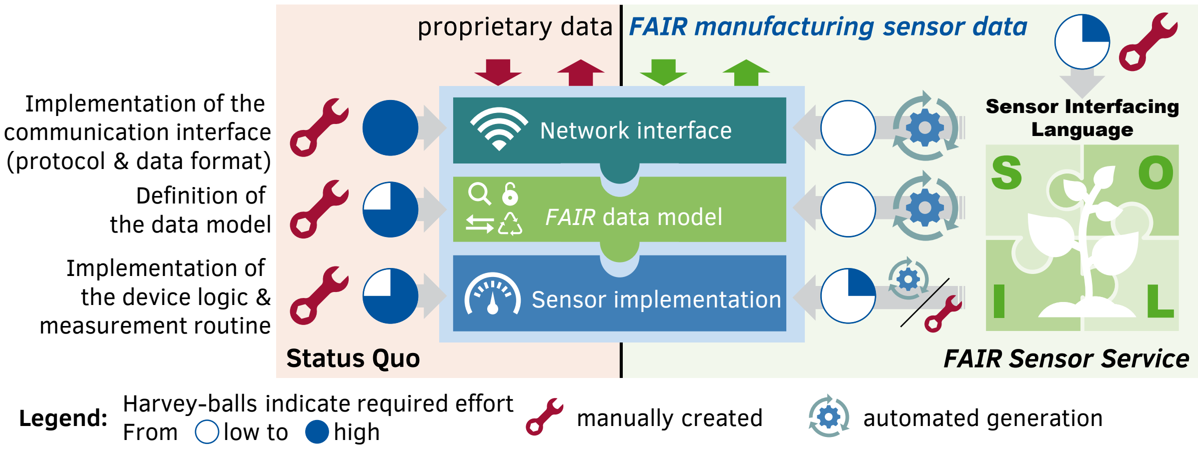


Figure 1.5: The envisaged contribution of *FAIR Sensor Services* and the *Sensor Interfacing Language* (SOIL) for the low-effort provision of *FAIR manufacturing sensor data*. The effortful status quo of developing and deploying measuring systems in manufacturing is shown on the left. The right half shows the provision of *FAIR manufacturing sensor data* and the (partially) automated deployment of a CPMS due to standardized data modeling based on the SOIL.

Definition 1.5: *FAIRness*

The *FAIRness* describes the degree of compliance of a digital object, such as a data set, with the *FAIR Guiding Principles*.

Definition 1.6: *FAIR manufacturing sensor data*

Data represented by a data model adhering to the *FAIR Guiding Principles* and therefore has a high *FAIRness* is denoted as *FAIR data*. *FAIR data* measured and provided by *FAIR Sensor Service* deployed in manufacturing or production is further denoted by *FAIR manufacturing sensor data*.

1.3 Formulation of the research objectives and questions

The main objective of this thesis is to leverage the value-creation potential of manufacturing sensor data by enabling the acquisition and exchange of interpretable data and ensuring its long-term reusability. The *FAIR Guiding Principles* provide guidelines for the design and development of data management facilitating this objective. However, the implementation of the *FAIR Guiding Principles* suffers from the high expertise and effort required to define the complex, interoperable data models [Altu23; Fahe19] and the diversity of measuring systems to be considered. Thus, operationalizing the *FAIR Guiding Principles* in the manufacturing industry requires scalable tools and automation frameworks to implement measuring systems accordingly with low effort [Laba20]. To lower the overall effort required for implementing *FAIR Sensor Services* for the manufacturing industry, this thesis primarily aims for two achievements: (i) the abstraction of the highly sophisticated semantic, interoperable data models to be easily definable (ii) the automated generation of software for running a CPMS reducing the implementation effort. The principal hypothesis to be investigated in the thesis is that these achievements can be reached under the given circumstances by adopting the high abstraction power and the automation potential of model-driven software engineering (MDSE) based on domain-specific languages (DSLs) tailored to this very problem. The conjunction of the *FAIRification* objective using DSLs culminates in the principal research question (PRQ) investigated within this thesis:

Principal Research Question

Can manufacturing sensor data be made *FAIR*, i.e., findable, accessible, interoperable, and reusable, by using domain-specific modeling languages?

Three research guiding questions (RGQs) narrow down the scope and the direction of the research to answer the PRQ. First, the *FAIR Guiding Principles* do not specify definite and measurable criteria that data (sets) must fulfill to be considered compliant with the *FAIR Guiding Principles* [Dors23; Bahi20; Muse22]. Moreover, the *FAIR Guiding Principles* are primed to research data, so the suitability to manufacturing data from industry must be investigated, and adaptions must be made if required. The detailed examination of the *FAIR Guiding Principles*, their interpretation in the context of manufacturing, and the definition of specific criteria are summarized by the first research guiding question (RGQ):

Research Guiding Question 1

What are the criteria manufacturing sensor data must meet to satisfy the requirements defined by the *FAIR Guiding Principles*?

Next, measurement data must be represented by data models that are compliant with these *FAIRness* criteria to leverage the reusability of the data. The data must be stored while preserving its context and provenance to allow reuse after a long time. Considering the large amount of heterogeneous data acquired and also stored during this period, the findability of the particular data of interest (among all the other data) must be guaranteed. It must be ensured that all data still can be understood and interpreted correctly. Thus, the data model must provide an exhaustive and comprehensible description and documentation of the measurement. These challenges are addressed by the second RGQ:

Research Guiding Question 2

How can manufacturing sensor data be described to be findable and interpretable in the applications' context?

Last, the measuring systems must be enabled to produce and provide the sensor data represented as defined by the interoperable data models based on controlled vocabularies. Thus, the CPMS must transform the data accordingly and provide an interface that is seamlessly accessible using a standardized and open communication protocol and data format. However, this task is associated with two main challenges. First, implementing these for each measurement device requires high effort. So, automation of the generalizable implementation of the communication protocol and data serialization is favorable. Second, developing the CPMS based on the device-specific data models as of RGQ 2 requires an interdisciplinary approach with advanced knowledge of semantic modeling, measuring systems, and metrology [Altu23; Voge20; Otto14]. However, the typical skill set of employees and developers in this field reveals a gap in this interdisciplinary knowledge [Li21; Sani23]. For the specific case studied in this thesis, this means that engineers, being experts in metrology and measuring systems, are usually non-experts in semantic modeling and not familiar with controlled vocabularies. In contrast, information scientists - or data stewards in particular - are experts in semantic modeling and controlled vocabularies. However, they typically lack the process and domain knowledge of metrology, so they cannot define the models for specific measuring systems.

Following the principal hypothesis of this work, these two challenges are tackled using domain-specific modeling languages (DSMLs), which effectively decouples the formal semantic modeling process and the definition of a particular sensor model. Moreover, the abstraction power of DSMLs allows to fully generate the source code of the communication interface and data serialization from the models defined for RGQ 2. Consequently, RGQ 3 specifies the investigation of this hypothesis as follows:

Research Guiding Question 3

How can controlled vocabularies be reused systematically by domain-specific modeling languages to provision accessible and interoperable manufacturing sensor data?

Besides the formulated objectives and questions, the following list explicitly defines the aspects that are considered out of scope for this thesis:

- different data (sources) than manufacturing sensor data
- organizational aspects
- addressing legal aspects of data reuse (e.g., which license is appropriate)
- implementation of data security and safety measures (i.e., authorization and authentication)

1.4 Research methodology and structure of this thesis

Research in engineering and production technology is located at the edge of industrial applicability. The motivation and objectives of this thesis originate in the deficits and needs of the manufacturing industry and, therefore, strive for timely adoption. Thus, Design Science Research (DSR) [Hevn07] is employed as research methodology reflecting the application-oriented focus of this thesis. Summarized in Figure 1.6 research according to DSR is embedded into the environment, i.e., the domain of application thoroughly capturing all relevant information, and the knowledge base of applicable methods and the scientific state of the art related to the considered problem. DSR is an iterative approach centered around developing specific artifacts and carried out in three cycles, ensuring proper consideration of the environment and knowledge base.

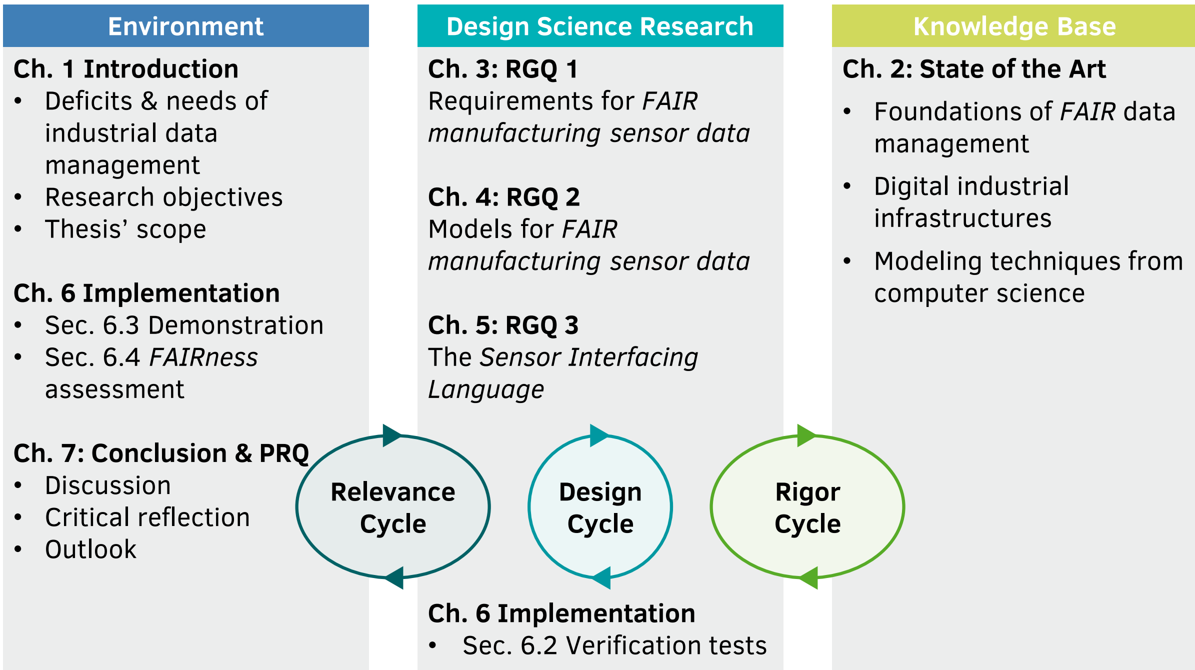


Figure 1.6: The structure and methodology of this research and thesis based on DSR [Hevn07] showing its three cycles and their roles within this thesis.

Design Cylce is the core of DSR. Within this cycle, specific artifacts are implemented to solve open problems and questions about the environment. The artifacts of this scientific work, i.e., the concept and implementation of *FAIR Sensor Services* and SOIL, are iteratively developed, implemented, and evaluated to ensure their envisaged functionality and contribution to the research objectives.

Relevance Cycle ensures the interaction with the environment of the research. The research shall be based on the deficits, needs, and requirements of the application domain to ensure the developed artifacts meet these. Review of the current state of data management for manufacturing in Chapter 1 ensures *FAIR Sensor Services* and SOIL developed in the *Design Cycle* target the current deficits and finally provide *FAIR manufacturing sensor data* to meet the needs.

Rigor Cycle grounds the DSR on the knowledge base, ensuring the current state of the art and best practices are considered. It provides the (scientific) methods and approaches for developing *FAIR Sensor Services* and SOIL in the *Design Cycle*. Furthermore, the *Rigor Cycle* should ensure the newly developed methods and tools are fed back and extend the knowledge base for subsequent research. The *Rigor Cycle* of this work is mainly formed by Chapter 2 reviewing the interdisciplinary state of the art in industrial data management and foundational modeling techniques from computer science and dedicated sections in the Chapters 3 to 5 reviewing the specific state of the art for each RGQ, including *FAIRness* metrics, existing ontologies, and modeling frameworks.

Based on the DSR research methodology and the formulated research objectives and questions, this thesis is structured as summarized in Table 1.1. After the introduction to the topic, the motivation and derivation of the research gap, objectives, and questions in this first chapter, Chapter 2 reviews the interdisciplinary state of the art as the knowledge base of this research. First, the *FAIR Guiding Principles* and their current adoption and importance in the manufacturing domain; second, the status quo of data management in production; and third, the required foundations and methods of computer and information science to implement the *FAIR Guiding Principles*. The Chapters 3 to 5 address the three RGQs one after another. Based on the specific criteria for *FAIR manufacturing sensor data* analyzed and defined in Chapter 3, an interoperable data model is developed in Chapter 4 representing manufacturing sensor data compliant with the *FAIR Guiding Principles*. Chapter 5 describes the development of SOIL for operationalizing the data models derived before and the automation of the implementation procedure. Each RGQ leads to findings and methods that populate the knowledge base for subsequent RGQ and related work. The implementation of *FAIR Sensor Services* as the primary research artifact for the prototypical validation in a close-to-industry scenario is described in Chapter 6. Lastly, this validation serves as the base for evaluating the PRQ in the final Chapter 7, in which the conducted research and the results are critically reflected and summarized in the application's context. It also provides an outlook to subsequent and open research and questions.

The scope and focus of the work are further defined with the help of the heuristic framework by Kubicek [Kubi77]. Therefore, the scope is framed by the reference architecture of the Internet of Production (IoP) [Schu17] as shown in Figure 1.7. The architecture can be simplified to three layers. First, the data acquisition and provision (*Sense*); second, the data structuring, transformation, and processing to gain insights (*Think*); and third, the data used to derive decisions and execute actions (*Act*). This thesis focuses on providing *FAIR manufacturing sensor data* on the level of *Sense* delimited by the interface providing the data to data-driven applications located in the *Think* layer. Assumptions about the design and characteristics of these data-driven applications are deliberately omitted in favor of the broadest possible applicability of the methods and artifacts developed.

Table 1.1: Overview of the structure of this thesis and mapping to the research objectives and questions, the research methodology, and the *FAIR Guiding Principles*.

	Thesis‘ chapter	*FAIR Sensor Services*	Research Cycle	*FAIR Guiding Principles*
1	**Introduction**	Definition of the concepts, objectives, and questions	Relevance	
2	**Interdisciplinary state of the art**	Review of the knowledge base	Rigor	
3	***FAIRness* requirements and criteria**	Investigation of the research guiding question 1	Design	
4	**Data models for *FAIR manufacturing sensor data***	Investigation of the research guiding question *2*	Design	
5	**The *Sensor Interfacing Language***	Investigation of the research guiding question *3*	Design	
6	**Implementation & Validation**	Prototypical software artifacts' realization and validation	Design Relevance	
7	**Discussion & Conclusion**	Answer to the principal research question	Relevance	

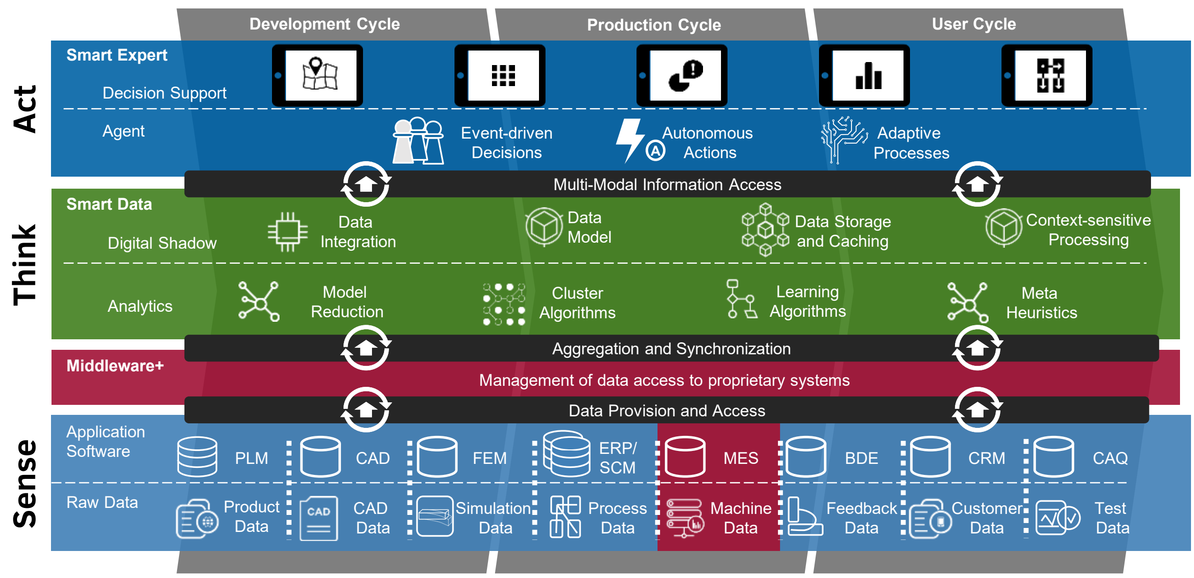


Figure 1.7: The scope and focus of this research within the reference architecture of the Internet of Production [Schu17]. As marked in red, the focus is the acquisition of sensor data and the provision via an appropriate middleware at the edge. The investigation of other types of data (sources) and specific applications based on the *FAIR manufacturing sensor data* are out of the scope of this research.

2 Interdisciplinary state of the art

Prior to the primary research activities and the design cycles, this chapter reflects the rigor cycle of the conducted research by reviewing the interdisciplinary state of the art and related scientific work. Thus, this chapter populates the knowledge base of both domains. *FAIRification* of manufacturing sensor data is a highly interdisciplinary activity of production engineering and computer science so the required foundations in both domains must be investigated. The research deficits in industrial data management and objectives proclaimed in Section 1.1 and Section 1.3 are underpinned by a detailed review of digital industrial infrastructures and scientific approaches to extend these. In the second part of this chapter, the technologies and methods from the computer science domain are introduced, namely the *Semantic Web* and model-driven software engineering (MDSE), that have been adopted in the developed artifacts of this work.

2.1 The *FAIR Guiding Principles* in the context of manufacturing

The *FAIR Guiding Principles* [Wilk16] are designed to guide data producers and publishers in creating an infrastructure that supports the maximum use of data. The origins of these principles were driven by the recognition that large amounts of data were being under-utilized due to the lack of standards that could enable easy sharing and manipulation. The primary purpose of the *FAIR Guiding Principles* is to ensure that data and metadata are well organized and robust enough to facilitate discovery and reuse by humans and machines, thereby fostering scientific innovation and collaboration. Implementing these principles helps overcome barriers to data use, such as inconsistent data formats, limited data access, and poor data integration capabilities. As a result, the adoption of the *FAIR Guiding Principles* is being advocated across scientific disciplines to ensure that data generated by research investments remain accessible and valuable over time, promoting the sustainability of scientific research and increasing the efficiency of scientific inquiry [Loga23]. An overview of the *FAIR Guiding Principles* is given in Figure 1.3. In the following, the *FAIR Guiding Principles* are introduced in detail, and their meaning and implications are investigated for the manufacturing domain.

2.1.1 Findability

The findability of measurement data supports the process of "figuring out what data exists and where it exists." [Laba20, p. 202]. Thus, one can consider findability as the first condition data must be fulfilled because otherwise, it does not matter whether the data is accessible, interoperable, or (re-)usable. Wilkinson et al. [Wilk16] define four guidelines for the findability of data, which are mainly technical (except for guideline **F2**) to circumvent data loss.

Findability guidelines

F1 (meta)data are assigned a globally unique and persistent identifier

F2 data are described with rich metadata (defined by R1 below)

F3 metadata clearly and explicitly include the identifier of the data it describes

F4 (meta)data are registered or indexed in a searchable resource

Findability requires appropriate data infrastructures, supporting efficient discoverability and searchability [Kali17]. Findability can not be treated as isolated from other aspects of the *FAIR Guiding Principles*. In guideline **F2** Wilkinson et al. [Wilk16] explicitly refer to the requirements of reusability, which is also underlined by others [Altu23; Řezn22]. Moreover, findability relies on a consistent and common language of the data description because inconsistently or ambiguously described data is hard to find, as it might not match executed search queries [Řezn22]. In particular, the language used to describe and search for data must match. As the usage of a common language is demanded by guideline **I1**, it can be concluded that data that is not interoperable is also complex to find [Altu23; Řezn22]. Considering the findability of a CPMS and its produced data, respectively, the description of the interface of the CPMS according to **F2** and the unique identification demanded by **F1** must be machine-interpretable, standardized and provide additional information about the measurements it conducts [Weig20; Laba20]. Summarizing these requirements, the findability of manufacturing sensor data is defined by:

Definition 2.1: Findability

The *findability* of data produced or stored by a system is given if the data can be effectively searched - using expressive metadata (**F2**) -, identified (**F1**) and located (**F3**, **F4**) by humans and machines. [Altu23; Wilk16]

2.1.2 Accessibility

According to Wilkinson et al. [Wilk16], the accessibility of data is majorly a technical concern of data provision using tools and approaches that lower the barrier for data access:

Accessibility guidelines

A1 (meta)data are retrievable by their identifier using a standardized communications protocol

A1.1 the protocol is open, free, and universally implementable

A1.2 the protocol allows for an authentication and authorization procedure, where necessary

A2 metadata are accessible, even when the data are no longer available

First, it must be underlined that accessible data does not require the data to be open. The latter implies a loss of data sovereignty and security, which is usually not feasible in industry.

The *FAIR Guiding Principles* explicitly suggest implementing a mechanism for authentication and authorization, if the data should or must not be openly shared, cf. guideline **A1.2**. The only thing demanded by the *FAIR Guiding Principles* to that regard is that the conditions under which the data can be retrieved are transparently communicated [Land20]. Nevertheless, the more open data is, the more accessible it is, as it can be indexed and thus found more easily. In an industrial context, establishing accessible data is mainly hindered by missing incentives for data creators to share their data [Laba20] and the concern that data might be (miss-)used by competitors. Moreover, the strict regulations and often complex decision hierarchies significantly increase the effort to provide data in an accessible way, such that data is stored locally and isolated. Concluding, the accessibility of manufacturing sensor data can be defined by:

Definition 2.2: Accessibility

Data is considered to be *accessible* if the possibilities to reach and retrieve the data are transparently defined, communicated, and implemented.

2.1.3 Interoperability

Over the past decades, a multitude of definitions for interoperability have been specified. Sixteen years ago Ford et al. [Ford07] collected 34 distinct definitions for interoperability and identified 64 different types of interoperability [Gürd18]. It can be expected that since then, many new definitions have been published. The definitions vary between domains, depending on the context and the details. Wilkinson et al. define interoperability as the "the ability of data or tools from non-cooperating resources to integrate or work together with minimal effort" [Wilk16] and formulate three guidelines for achieving interoperability:

Interoperability guidelines

I1 (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.

I2 (meta)data use vocabularies that follow *FAIR* principles

I3 (meta)data include qualified references to other (meta)data

Narrowing down the technical aspects of interoperability considering the guidelines **I1** and **I2**, three levels of interoperability of services or systems communicating with one another can be distinguished on a coarse scale and located in the knowledge pyramid Figure 2.1. At the lowest level of interoperability, systems can communicate and transfer values from one system to another:

Definition 2.3: Technical Interoperability

Technical interoperability is the ability of systems to exchange data using a (common) communication protocol. [Wang09; Turn06; Tolk03]

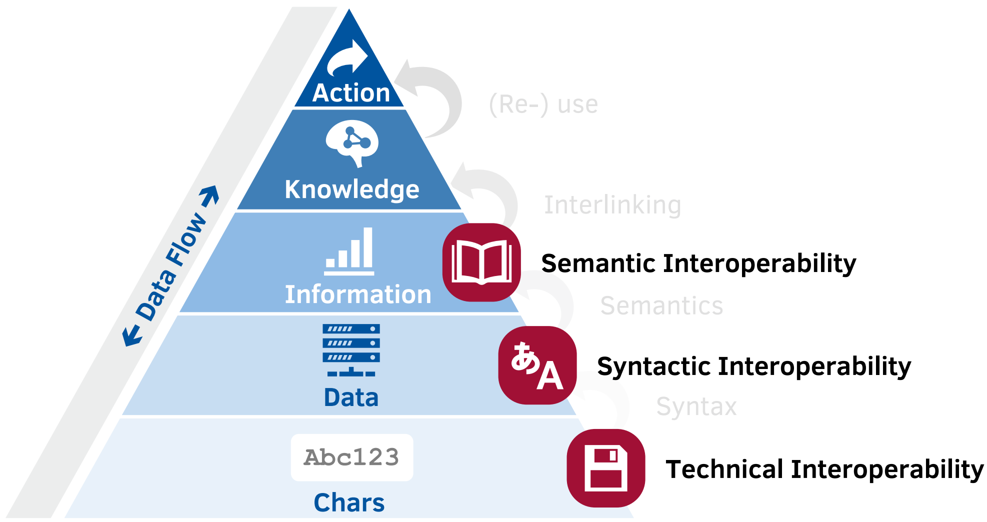


Figure 2.1: The three levels of interoperability mapped to the knowledge pyramid (cf. Figure 1.4).

Precise knowledge of these values is required to process values exchanged by technically interoperable systems. The receiver must know the message structure in advance to use the received values. Otherwise, it must be accessibly documented somewhere. High effort is required to use the exchanged values, which contradicts the interoperability definition by Wilkinson et al. [Wilk16]. They consider this characteristic of data, i.e., the ability of a system offering this data, as accessibility instead. Efforts can be lowered by a common structure serializable to open and standard formats being *syntactically interoperable*.

Definition 2.4: Syntactic Interoperability

Syntactic interoperability is the ability of systems to exchange data in a compatible format using a common structure. [Maci17; Turn06; Tolk03]

Syntactically interoperable data can be processed by machines due to the defined data structure. Such a common structure usually consists of individual elements representing a specific aspect or property stored as key-value pairs. When exchanging such data, the keys and values exchanged must be syntactically interoperable. There must also be an agreement on the meaning of the keys to process the data. However, interpretation can only be done by experts who have domain knowledge of the individual elements of the data structure and are aware of the meaning of the keys within a specific context or application. Therefore, these structures do not adhere to the guideline **I2**. The keys must be taken from a controlled vocabulary to allow application-agnostic and inter-domain interpretation of the exchanged data, which leads to *semantic interoperability*.

Definition 2.5: Semantic Interoperability

Semantic interoperability is the ability of systems to exchange data with an unambiguous and comprehensible meaning using a common description language. [Maci17; Nils18; Tolk03]

Semantically interoperable data can be interpreted by machines. Thus, semantically interoperable data complies with the guideline **I1**. To specify exchanged information utilizing a common description language, definitions of the *Semantic Web* community are usually used [Nils18] following guideline **I2**. Thus, the terms used as keys to identify and describe exchanged values are taken from openly accessible vocabularies or ontologies, although their actual *FAIRness* is currently hard to assess. In addition, information can always only be uniquely and unambiguously interpreted within a given context. That means **I3** must also be fulfilled to ensure semantic interoperability. A summary of the three levels of interoperability and comparison with related work is given in Table 2.1.

2.1.4 Reusability

Following the relevant literature on the *FAIR Guiding Principles* and their implementation, achieving high reusability is the main objective of all data management efforts [Wilk16; Hart24], as the reusability can be directly related to the value-creation potential of the data. In the introductory publication [Wilk16], the reusability of data is characterized by only one primary guideline, which is further detailed by three sub-guidelines:

Reusability guidelines

R1 meta(data) are richly described with a plurality of accurate and relevant attributes

R1.1 (meta)data are released with a clear and accessible data usage license

R1.2 (meta)data are associated with detailed provenance

R1.3 (meta)data meet domain-relevant community standards

The guidelines for reusability imply the most issues to the industrial implementation of the *FAIR Guiding Principles*. First, data licensing, as required by guideline **R1.1**, is uncommon and hardly applicable in the industry due to strong concerns regarding the loss of data sovereignty and intellectual property. The typical way to share data in an industrial context is via negotiating individual access and reuse conditions and contracts [Penn24], as they pretend to maintain better control over the release and use of data [Grab19]. Second, the domain-relevant community standards as of guideline **R1.3** do not exist in a usable fashion for the manufacturing domain, cf. Section 2.2.1. Nonetheless, the reusability of (manufacturing sensor) data can be defined as:

Definition 2.6: Reusability

Data is called *reusable* if all information is available, that is required for deciding whether the data can (re-)used (**R1**, **R1.2**, **R1.3**), and the conditions under which the data (re-)use is permitted are defined clearly (**R1.1**).

In summary, the *FAIR Guiding Principles* can be utilized to improve the value of data in general and not of scientific data only. In particular, the described challenges in the production industry can be addressed by adopting the *FAIR Guiding Principles* for this domain. Related work on implementing the *FAIR Guiding Principles* in an industrial environment is compared in the next section.

Table 2.1: Mapping and comparison of the three levels of interoperability defined by Definitions 2.3 to 2.5 to related work and definitions of interoperable (sensor) data.

	Technical interoperability	Syntactic interoperability	Semantic interoperability
[Cui20]	Non-modeled data	Data modeled without using ontologies	Data modeled using ontologies
[Eich23]	Level 1: Manual metadata management for digital sensor data	Level 2: Structured, machine-readable metadata about the sensor network with modeling languages and SOA considered	Level 4: Structured, machine-readable metadata with content that can be semantically interpreted by software. Implementation of the *FAIR Guiding Principles* for data structures and metrological data
[Czar21]	Level 1: Digital document	Level 2: Machine-readable document	Level 4: Machine-interpretable content
[Wang09]	Level 1: one can observe symbols of communication through a common communication protocol.	Level 2: the structure of data is defined, but not the meaning of the data elements.	Level 3: common reference model allows interacting systems to exchange terms that they can semantically parse.
[Turn06]	Level 1: Technical interoperability: a communication protocol exists	Level 2: Syntactic Interoperability level introduces a common structure to exchange information	Level 3: semantic interoperability: a common information exchange reference model is used
[Tolk03]	Protocol Interoperability: communication protocols are supported	Data/Object Model Interoperability: standardized data element used for the data/information	Information Interoperability: procedures and models used to represent information are mapped
[Ford07]	1) Unstructured data exchange	2) Structured data exchange	4) Seamless sharing of information

2.1.5 *FAIRification* approaches for industrial data

Approaches and literature on *FAIRification* of industrial data are sparse, and a comprehensive solution was not found [vReis20]. A generic *FAIRification* workflow has been proposed by Jacobsen et al. [Jaco20b]. It suggests a general procedure for making existing data *FAIR* by applying tools for assessing the *FAIRness* of existing data, defining metadata models, collecting metadata, and linking those with the data itself. Although this procedure generally applies to all kinds of data, no tools are implemented for the most cumbersome steps. Moreover, the applicability for *FAIRification-by-design*, i.e., the process of making data *FAIR Guiding Principles* at the point of creation, is yet unclear. Prominent tools for *FAIRification*, like *FAIRifier* [Thom20], are only applicable for the *FAIRification* of previously collected data. The open web service *FAIR Data Point* [Kuzn20] provides a server and an API for storing metadata sets. The tool is already widely applied in the Netherlands for storing COVID-19-related data. At the same time, the implemented data storage addresses findability and reusability; regulations on the data models are not defined. Thus, the interoperability of the data is not facilitated and is still a concern of the user.

Gleim et al. [Glei20] proposed *FactDAG*, an approach fully dedicated to the *FAIR Guiding Principles* for describing production data and its provenance. To persistently identify data, keep track of changes, and preserve provenance, each data point is considered an immutable *Fact*, identified by a triple called *FactID* and stored in a directed acyclic graph. Further, the authors did a reference implementation called *FactStack* [Glei21] build upon standardized protocols and data formats following the *FAIR Guiding Principles*. However, the approach lacks a dedicated way of filtering and searching the data by custom domain-specific keys and attributes. This diminishes findability and challenges the retrieval of manufacturing sensor data in practice.

Dorst et al. [Dors23] investigated integrating metrological criteria and the *FAIR Guiding Principles* based on the Hierarchical Data Format (Version 5). The authors used terms provided in ontologies to describe the measurement data and prove notable *FAIRness* assessed using the *FAIR* Data Maturity Model (*FAIR* DMM) [Bahi20] (see Section 3.2.1). However, whether an automated or manual evaluation has been carried out is not evident. Due to the low ripeness of the *FAIR* DMM and the assessment itself, it can be assumed that the evaluation has been done manually by the authors, which tends to yield better results than computationally assessed ones, as already recognized earlier [Bode23]. Thus, the author can not objectively verify the results. Moreover, Dorst et al. describe their individual decisions and do not describe a generalizable or even automatable procedure required to perform *FAIRification* of continuously acquired process data in manufacturing.

2.2 Status quo of data management for manufacturing

Justification for implementing the *FAIR Guiding Principles* for manufacturing requires investigating the current data management practice in the production industry. The reasons for the lack of holistic data life-cycle management [Wang18a] need to be understood precisely to derive specific actions compensating current deficits. Moreover, the challenges that complicate the *FAIRification* process of manufacturing sensor data need to be anticipated as well as possible to avoid shortcomings in the research conducted.

2.2.1 Deficits of industrial data management

In contrast to end-consumer technology, such as USB- or mobile hard drives, which allow “Plug & Play” usage due to technically interoperable and standardized drivers, allowing immediate access and usage of the stored data, these solutions are missing for industrially deployed equipment, such as measuring systems and infrastructure [Drat23; Jirk17]. This lack of interoperability, and ultimately *FAIRness*, is caused by model-related, technical, and legal concerns and limits the value creation potential of manufacturing sensor data as summarized in Figure 2.2. In the following, four deficits are presented, which turned out to be most significant during the conducted studies on the implementation of *FAIRification* processes in manufacturing and have also been reported in the related work.

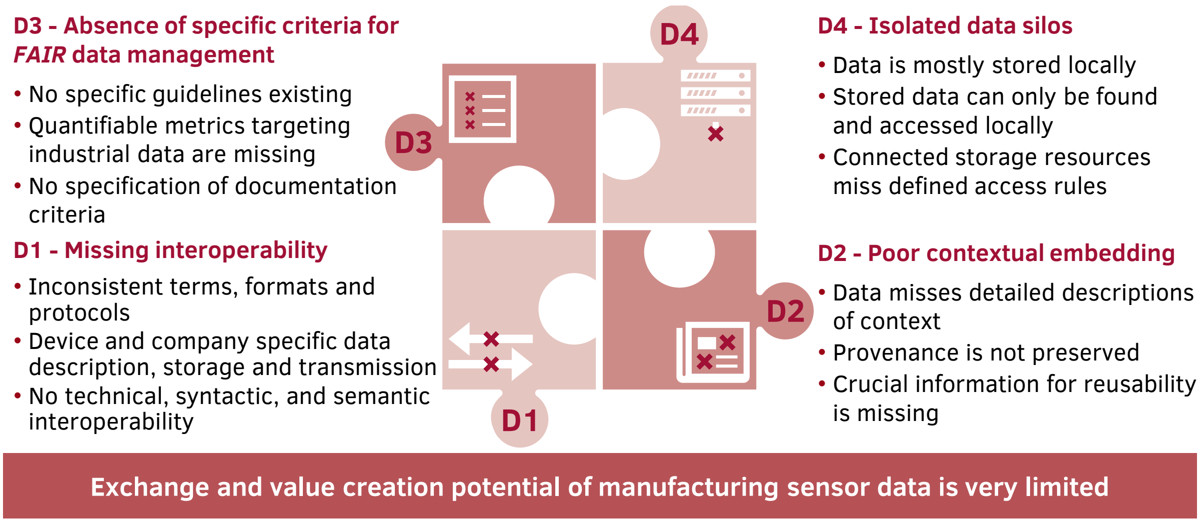


Figure 2.2: Significant deficits of industrial data management inhibiting joint data usage and limiting data-driven value-creation [Glei20; Guiz20; Thom20; vReis20; Zhen20a].

D1 Missing interoperability Manufacturing (sensor) data lacks consistent, agreed terms, formats, and protocols [Penn19b; Huba13]. The data description, storage, and transmission techniques depend on the sensors’ manufacturer and the company that uses them. Thus, data can often not be shared with and understood by a third party [Glei20]. The technical and syntactic interoperability (cf. Definition 2.3 and Definition 2.4) is largely hampered due to proprietary and individual solutions [Eich23; Li24], as devices and systems that have originally not been intended to work together need to be interconnected [Nils18]. The semantic interoperability (cf. Definition 2.5) lacks a controlled vocabulary for the description and documentation of acquired sensor data [Otto17b; Zhan23].

D2 Poor contextual embedding In data-driven and particularly machine learning applications, algorithms initially do not know in advance which data is required to solve the considered problem [Libe15]. If the context information is missing, it can not be evaluated in advance to see if data could be used or if the data was suitable in retrospect. However, collected data usually misses a detailed description of the context within which the data has been created, i.e., the documentation of related entities, such as the process the sensor is integrated into or involved tools and stakeholders [vReis20]. Thus, crucial provenance information regarding the reusability of the data is missing.

D3 Absence of specific criteria for FAIR data management The *FAIR Guiding Principles* are abstract design guidelines only [Dors23] . To implement the *FAIR Guiding Principles* for manufacturing sensor data, specific criteria must be defined to evaluate the *FAIRness* objectively. It must be clarified which exact information is required to make data findable, interoperable, and reusable [Thom20] in manufacturing.

D4 Isolated data silos Data is mostly acquired by and stored on local measuring systems and machines with restricted connectivity so that data can only be found and accessed locally [Laba20; Glei20]. If these devices are connected to a (company-internal) network, such that this data can theoretically be accessed remotely, access rules defining who is permitted to interact with the data are usually not defined [Zhen20a]. Consequently, this isolated data can not be accessed and, therefore, not (re-)used, imposing limitations on the effectiveness of the data-driven manufacturing and decision processes [Li22].

2.2.2 Challenges of the *FAIRification* of manufacturing sensor data

This thesis aims to overcome the abovementioned deficits using the *FAIR Guiding Principles*, cf. Section 1.3. However, efforts to remedy these deficits are hampered by the current heterogeneous circumstances of digital manufacturing [Schl24; Jirk17]. Put simply, heterogeneity can be understood as the counterpart to interoperability and is, therefore, both a cause and an obstacle to suboptimal data usage. The resulting challenges for the *FAIRification* process are summarized in Figure 2.3 and discussed in the following.

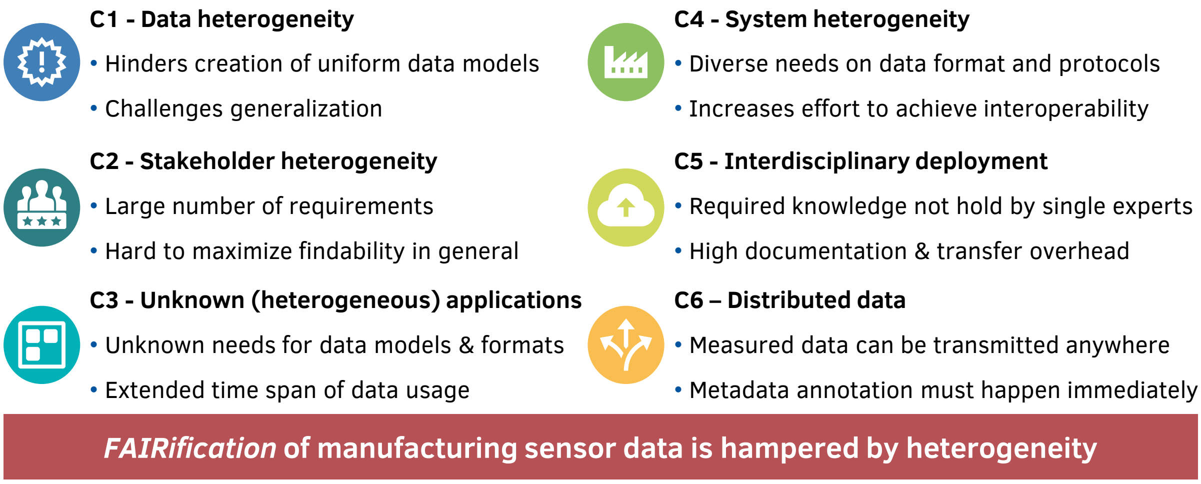


Figure 2.3: Six challenges hamper the *FAIRification* of manufacturing sensor data and impose increased complexity to the tools and solutions for providing *FAIR manufacturing sensor data*.

C1 Data heterogeneity Industrial (sensor) data is characterized by high quantity and high heterogeneity. Thus, the metadata associated with the data also has high heterogeneity. The high data heterogeneity significantly increases the complexity of standardizing the data underlying models [Liu14; Maci17; Baum17] and makes it practically impossible to define one standardized model covering all aspects of the data, which are required to achieve a detailed description as demanded by the *FAIR Guiding Principles* [Glei20]. Beyond that, the standardization of the data transmission and exchange is complex because of the different companies' proprietary data formats and communication protocols for their individual devices and systems [Schl24; Cui20]. Ergo, the developed methods and tools for *FAIRification* of manufacturing sensor data must be flexible, adaptable, and interoperable with existing standards.

C2 Stakeholder heterogeneity The group of stakeholders of manufacturing (sensor) data is manifold [Stea96]. To maximize findability for all of these stakeholders, the metadata must contain appropriate information in an indexable format, which they (or algorithms) will most probably use when searching for the data [Azzo13]. Moreover, the requirements for reusability differ between use cases and, therefore, between different stakeholders, which means the metadata collected and stored must be sufficiently broad to address distinct stakeholders but also specific enough so that a stakeholder can evaluate the reusability in their situation.

C3 Unknown (heterogeneous) applications At the time of data creation, i.e., when the measurement is taken, the group of applications potentially reusing the data is unknown in its entirety. Generally, one has applications in mind for which to use data, but one cannot imagine every possible application the measured data might be used for in the future. For example, activities such as the maintenance or disassembly of products after potentially years of usage require data about the production and usage of that product, which are currently either not available or not usable, as this usage of the data has not been anticipated at the time of production. Hence, the metadata and the format required for such applications are also unknown and demand the collection of all related metadata in standardized and open formats.

C4 System heterogeneity The information model, system architecture, and network setup differ from company to company and often even from site to site [Glei20; Schl24; Cui20]. Hence, the models and tools developed for a sensor will require manual adaptation at each deployment to be technically, syntactically, and semantically interoperable with the existing architectures and models. Moreover, technologies and systems deployed in manufacturing infrastructures tend to have extended lifetimes [Saut10]. So, one has to deal with many legacy systems, which will probably also (co-)exist with novel systems in the future. This implies costly, time-consuming, and error-prone adaptations. Thus, the *FAIRification* procedure must be system agnostic.

C5 Interdisciplinary deployment Deployment requires different expertise and knowledge from at least two domains [Jaco20b]: One has to be familiar with the sensor and the process, which should be *FAIRified*, and with the requirements and process of *FAIRification* itself. Because the expertise in *FAIRification* is usually not the core business of production companies, external experts must be consulted [Baum17]. This becomes even more complex when the deployment covers more issues than *FAIRification*. Therefore, the development, implementation, and deployment processes of *FAIR Sensor Service* must decouple these adverse activities.

C6 Distributed data In Industry 4.0 scenarios, a freshly acquired measurement might be untraceably distributed within an interconnected network and to innumerable consumers or storages [Pete20]. Furthermore, data consumption might occur immediately, e.g., within control loops, or later (up to several months or years). Thus, the *FAIRification* of the measurement must happen at the edge level before any transmission into a network takes place. Consequently, manual curation and *FAIRification* of data is not applicable due to the high effort associated with the manual process [vVlij20]. Consequently, the *FAIRification* of sensor data must be automatable.

2.3 Digital manufacturing infrastructure

A plethora of technologies exist for handling, representing, storing, and exchanging data in digital infrastructures. Based on the three aspects of interoperability defined in Section 2.1.3, the technological solutions currently implemented or developed in manufacturing are reviewed. By analyzing existing (i) data models, addressing semantic interoperability, (ii) data formats, addressing syntactic interoperability, and (iii) communication protocols, addressing technical interoperability, the current state-of-the-art in digital industrial manufacturing is investigated to substantiate the deficits and challenges identified in Section 2.2. Moreover, related scientific work aiming at standardizing the digital infrastructure for measuring systems is reviewed.

2.3.1 Industrially used technologies

Figure 2.4 gives an overview of technologies and related work considered and indicates their classification regarding the coverage of the three interoperability aspects. The details of each depicted approach are summarized and discussed in the following.

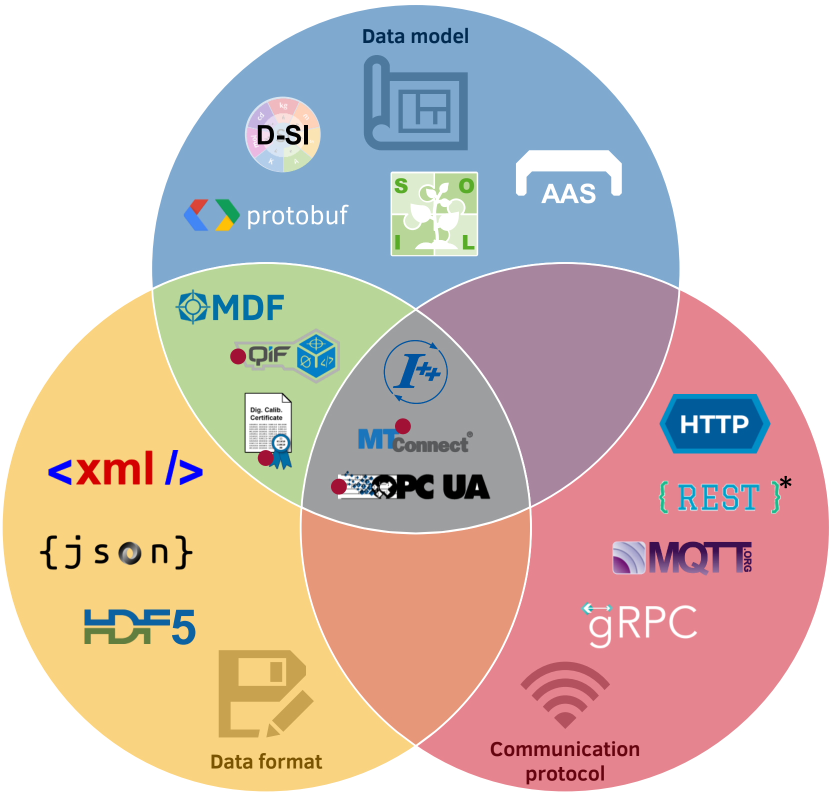


Figure 2.4: Classificiation of IIoT frameworks and technologies in the categories *data model*, *data format*, and *communication protocol*. The frameworks marked with a red circle do not define a custom data format but enforce the usage of XML only. *Although strictly REST is a design pattern and not communication protocol, it is often considered as such because REST describes how the communication interface should be designed. The classification of SOIL and the underlying data model developed in Chapters 4 to 5 is anticipated here.

Asset Administration Shell

Some consider the Asset Administration Shell (AAS) a fundamental pillar in the Industry 4.0 landscape. The AAS enables digitally and dynamically representing physical and non-physical entities. Therefore, Drath et al. [Drat23] even proclaim the AAS the technical realization of the Digital Twin (DT). The AAS encapsulates the vendor-specific interface of an asset employing a vendor- and protocol-independent model of the asset, which contains all information about the asset, e.g., properties, capabilities, measurements, parameters, status, and condition. The model has a hierarchical structure that can be easily extended and combined with other AASs. To be interoperable, the AAS defines a domain-agnostic base model of which domain-, company-, use-case- or even device-specific submodels can and have been derived [Heid19]. Several open protocols and data formats are proposed and recommended for the implementation of the AAS, but in principle, the AAS can be implemented with any format and protocol. In addition, all information contained in a device's AAS can be serialized and thus easily exchanged between different partners. The AAS is also intended to uniquely identify the asset and its data in the digital world [Schl24]. Thus, the AAS significantly covers the *FAIR Guiding Principles*. Moreover, the lifetime of a specific AAS exceeds the lifetime of the asset it represents, requiring long-term usability of the contained data. However, the long-term validity and usability of the (sub-)models created are not explicitly addressed. In addition, the applicability of the AAS is not unlimited, which requires it to be accompanied by other technologies and frameworks for addressing all relevant use cases in manufacturing [Drat23].

Open Platform Communications Unified Architecture

One standard for a general data model and system architecture is Open Platform Communications Unified Architecture (OPC UA) [Damm09]. The strengths of OPC UA are the service-oriented architecture (SOA), which makes it possible to integrate previously independent, isolated parts of OPC into a framework, and the platform independence of the implementation. The SOA also makes it possible to break up monolithic structures, as independent modules can be implemented as individual services, and the coupling of the individual services can be minimized. This makes the system easier to expand, increases flexibility, and reduces maintenance costs [Schl24]. The versatile and flexible architecture of OPC UA is shown in Figure 2.5 and ensures interoperability but also specificity of devices. The core information model, i.e., the OPC UA metamodel, is generic and can be used for all devices and sensors. It is supplemented by Companion Specifications (CSs) which define specific nodes for certain domains, device groups (such as machine tools, robots, ...) [VDW22; Fieb19], or use-cases, e.g. Machine Vision from the German *Mechanical Engineering Industry Association* (German: *Verband Deutscher Maschinen- und Anlagenbau e.V. (VDMA)*) [VDMA19; VDMA24a]. Whereas the metamodel mainly defines the syntax of OPC UA, the CSs provide the semantics of concepts [Drat23]. Over 100 CSs have been published in recent years or are still in the draft phase [OPC 24]. CSs are either developed directly by the OPC Foundation, in cooperation with the OPC Foundation, or by external working groups.

An OPC UA-based information model of a measuring system consists of nodes representing the system's components, methods, variables, and edges connecting these nodes. To ensure interoperability, each node and edge should follow standardized definitions in the CSs. Thus, OPC UA explicitly contributes to the interoperability of manufacturing sensor data. Moreover, the CSs can be considered the establishment of the domain-relevant community standards demanded by guideline **R1.3**. Nevertheless, they are only valid and applicable within the OPC UA framework. All nodes are uniquely identified within the address space of the OPC UA server implementing the OPC UA-based CPMS [Ye18], which implicitly adheres to guideline **F1**. OPC UA has already been widely adopted in industry [Li24; Coit20; Eich20]. Nevertheless, developing an OPC UA model for a specific device requires high effort and expertise [Li24], which makes retrofitting legacy systems costly.

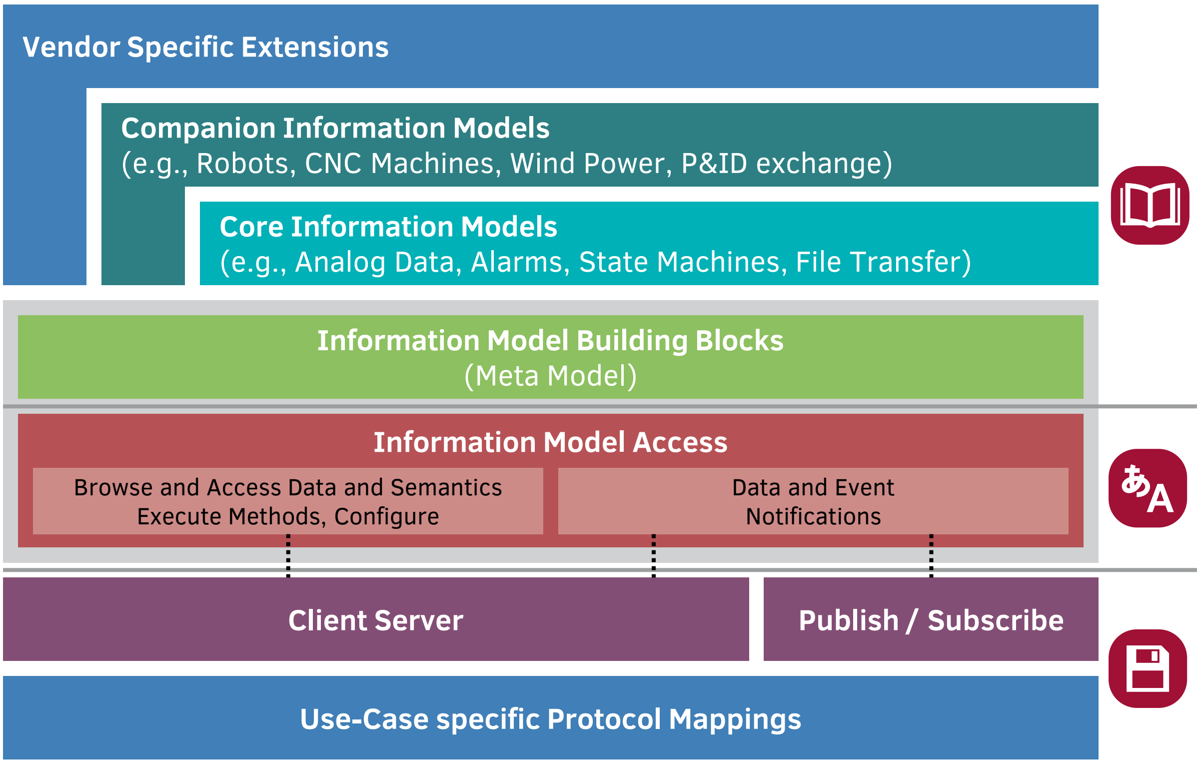


Figure 2.5: The software architecture of OPC UA as depicted at the webpage of the Open Platform Communications (OPC) Foundation [OPC 19]. The architecture ensures high flexibility and interoperability by definition of a generic metamodel as the core of the specification and is complemented by domain- or use-case-specific companion models. The extendible and versatile structure promotes the SOA of OPC UA and captures all three levels of interoperability.

MTConnect

Another standard for digital modeling and interfacing devices and sensors is MTConnect, primarily used in the USA. Like OPC UA, MTConnect relies on standardized data formats (Extensible Markup Language (XML)) and transport protocols (Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP)). This contributes to guideline **A1**. MTConnect is proven to provide real-time data acquisition [Shar22; Nguy22]. In contrast to OPC UA, the MTConnect information model contains building blocks for specific machine components, such as “LINEAR” for linear axes or “PNEUMATIC” for pneumatic systems [AMT 23]. This makes it easier to create a model using MTConnect for machine tools, as many components are already provided and do not have to be created manually by the model developer based on the metamodel. Thus, the standard also adheres to guideline **I1** because it ensures that two components of the same type have the same meaning [Shar22], even if they are contained in different models. With OPC UA, this cannot generally be guaranteed, as the type for a node in the OPC UA model can be defined by the model developer. However, this means that MTConnect is less flexible compared to OPC UA in terms of interaction [Push20] and domain coverage because the standard cannot be used in domains whose concepts are missing from the MTConnect information model. MTConnect even lacks full support for all types of data relevant for machine tools [Libe15]. Another significant difference is that MTConnect only allows read-only access to the encapsulated device. It is impossible to control or configure a device using MTConnect, which can also be considered a security feature [Shar22]. To ensure interoperability there is an OPC UA CS for MTConnect [Liu19; Sobe19].

Inspection Plus Plus Dimensional Measurement Equipment

Inspection Plus Plus Dimensional Measurement Equipment (I++ DME) was initially developed around 2000 by an initiative of several car manufacturers to overcome the burden of highly individual and proprietary interfaces of coordinate-measuring machines (CMMs) requiring costly modifications if the same measuring routine needs to be executed on machines from different CMM providers [Töpf07]. I++ DME defines a standardized data structure for the components of a CMM and a set of manufacturer-agnostic commands for executing measuring procedures and retrieving the results [Proc07]. Moreover, I++ DME defines a line-based communication protocol for a server-client architecture. Over the years, I++ DME has emerged as the de-facto standard for interfacing CMMs [Sous22a], but was also not updated for several years and is not compatible with IIoT protocols. Lately, the development of I++ DME has been reignited by a VDMA working group mainly consisting of CMM manufacturers and led to the publication of the new version 2.5 of the standard [VDMA24b]. Nevertheless, the data structure, data format, and communication protocol are technically outdated, inefficient, and error-prone compared to more actively maintained or recent frameworks, such as OPC UA or RESTful application programming interfaces (APIs).

The Digital Calibration Certificate

The measuring system or sensor's calibration status and documentation is an essential part of recording measurement data. This documentation can be done using the Digital Calibration Certificate (DCC) [Wied19; Must20]. The DCC is divided into four layers, see Figure 2.6. The top mandatory layer contains administrative information, such as unique identifiers. The results of the calibration are recorded in the second layer. To maintain interoperability, the Digital System of Units (D-SI) [Hutz20] is prescribed for specifying the measurement results of the calibration. Individual information about the calibration can be specified in the non-regulated part of the second layer. Layers three and four are used to store additional information. The third layer can be interpreted as a kind of commentary layer in which additional tables and graphics can be stored in freely selectable data formats. The customer's wishes for the calibration can be incorporated here. The bottom layer is optional and can contain a human-readable version of the information from layers one to three.

The DCC is stored and exchanged using the XML-based Calibration Data Exchange Format (CDE) [VDI22]. As the XML format is an open standard and can be signed using cryptographic algorithms, the *FAIR Guiding Principles* a are partially fulfilled for the implementation of the DCC by the CDE. Due to a mandatory, fixed representation required for unique identification in the first layer of the DCC, the data can be found more easily. By using the open XML format for storage in the CDE, a significant part of the accessibility requirements is fulfilled. Even if no protocol is proposed for data exchange, using XML means that almost any common Internet protocol, such as HTTP, can be used. The data is, therefore, also technically interoperable. However, there is no semantic description of the individual fields of the DCC. The DCC is reusable because legal requirements relating to the DCC are taken into account [Nika20], and the origin and authenticity of the data can be ensured by signing the CDE [Must20].

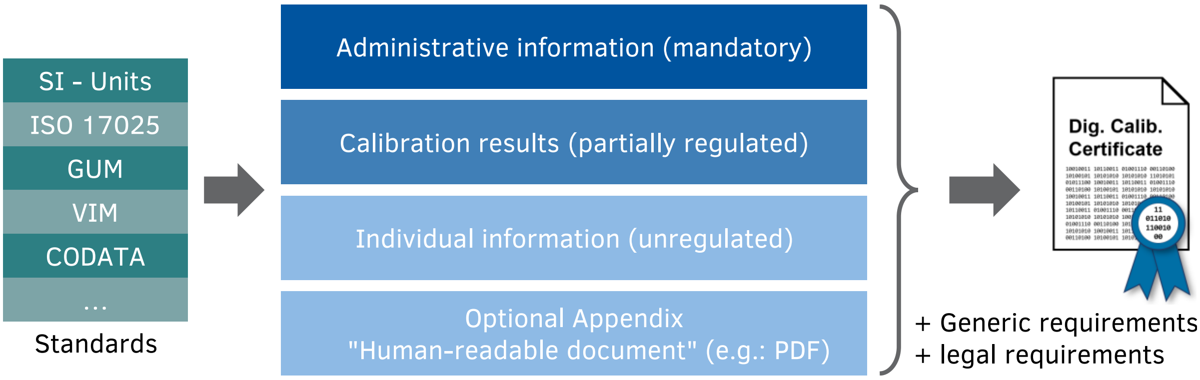


Figure 2.6: Overview of the structure of the Digital Calibration Certificate (DCC). Figure adapted from Montavon and Bodenbenner [Mont23]

Message Queuing Telemetry Transport

Message Queuing Telemetry Transport (MQTT) has been under development since 1999 and was standardized in 2013. MQTT implements the publish/subscribe principle for devices and applications using a broker as middleware [Al-F15], whereby the messages are organized into topics. The protocol enables one-to-one, many-to-one, and many-to-many communication with three levels of quality of service. It is based on TCP and its security mechanisms and has a low overhead, as the message header size is limited to two bytes. The (de-)serialization of messages is not part of the definition of MQTT and, thus, is agnostic of the data models and formats used. For networks that are not based on TCP/IP, such as for devices with limited resources, a variant of the protocol known as MQTT-SN (Sensor Networks) can be used.

Hypertext Transfer Protocol and Representational State Transfer

The origin of Representational State Transfer (REST) is usually attributed to Fielding's dissertation in 2000 but goes back to the HTTP Object Model of 1994. REST can be seen as an architectural design pattern for client-server systems that communicate using the request/response scheme via HTTP. Apart from representing the system state of the individual resources in the transmitted messages, the state is not saved within the system [Fiel00]. This favors horizontal scalability, tolerance of errors, multi-tenancy, portability, and the ability to cache results. Resources are organized and addressed using Uniform Resource Locators (URLs), but the messages sent are self-describing. A fixed set of HTTP verbs is used to specify the interaction, defined fields in the HTTP header, and typed message content, for example, serialized in JavaScript Object Notation (JSON) or XML. RFC 7252 defines the Constrained Application Protocol (CoAP) as a subset of REST, which was specially developed for embedded systems with low memory capacity and low computing power in restricted networks [Shel16].

Quality Information Framework

The Quality Information Framework (QIF) provides metamodels for measurement and quality data related to a product and its relevant features over the complete product life-cycle [Zhao12; Ram21]. The Digital Metrology Standards Consortium (DMSC) that maintains and standardizes QIF aims at providing neutral, vendor-independent solutions for handling measurement and quality data with QIF [Sous22b]. The models define the semantics of product characteristics, attributes of measurement data, and relations between those. Thus, QIF provides a standardized way to define sensor data models. Moreover, QIF explicitly considers the metrological traceability [Sous22b]. The metamodels are based on XML-schema so that the XML data format is used for storing and exchanging the measurement data. QIF has been standardized by the International Organization for Standardization (ISO) [ISO20]. Software libraries for creating and interacting with QIF models for measuring systems and measurement data are provided open-source [Zhao12; Camp24]. So, QIF implicitly adheres to guidelines **I1**, and **R1**. However, the measurement data is product-related, so that QIF can not link it to a process, measuring system, or manufacturing resource, e.g., a machine tool.

Measurement Data Format

The Measurement Data Format (MDF) defines a binary format for storing and exchanging measurement and calibration information. It has been developed for and is widely adopted in the automotive sector [Paul14]. In 2009, MDF became a standard of the Association for Standardization of Automation and Measuring Systems (ASAM). The underlying model allows for storage of the measurement data, including the corresponding metainformation. Due to its binary format, the storage and data exchange are efficient and support hardware with minimal transmission, storage, and computing resources. The specification of the standard is freely available up to Version 3.3 [Vect24]. Since the specification became an ASAM standard in 2009, the specification is only accessible to members of ASAM, which requires an annual fee [Vect24; Asso19]. Thus, the most current version of the standard is not freely and openly accessible, contradicting the *FAIR Guiding Principles* **A1.1** and **I1**.

Hierarchical Data Format (Version 5)

The Hierarchical Data Format (Version 5) (HDF5) is a format for storing highly dimensional and complex data [Häne20]. It is primarily applied in the scientific domain, as it natively stores the measurement data and its metadata in a single file. HDF5 is well suited for coherent data storage and exchange but less for large data analysis such as processing using Machine Learning [Magu22]. HDF5 can handle data of arbitrary types and dimensions and is therefore widely adopted such that standard libraries for many programming languages exist, making the format easy to use [Pelo23]. Dorst et al. [Dors23] already illustrated the usefulness of HDF5 for storing data and its metainformation in a *FAIR* manner. However, HDF5 is a file format and, thus, naturally splits data into batches. On the one hand, the format supports timely bound data acquisition, such as in most scientific experiments. On the other hand, the format is less suitable for continuous data streaming, e.g., in assembly or manufacturing processes, where the continuously acquired sensor data can not easily be split into logical batches. Moreover, the redundantly stored metainformation, which only changes occasionally, significantly increases the overhead.

Table 2.2: The characteristics of industrial technologies for data modeling, serialization, and transmission about the deficits and challenges identified in Section 2.2. The Harvey balls indicate how much the individual technology addresses or resolves the deficits and challenges.

Technology	D1	D2	D3	D4	C1	C2	C3	C4	C5	C6
AAS	◕	●	○	◔	●	◑	●	◑	◑	◕
OPC UA	◕	◕	○	◔	●	◑	●	○	◑	◕
MTConnect	◕	◑	○	◔	◕	◔	◕	○	◑	◕
I++ DME	◕	◔	○	○	◑	○	○	◔	◔	○
DCC	◕	◑	○	○	◕	◔	◕	○	○	◔
MQTT	◔	○	○	◔	○	○	○	◕	◑	○
HTTP	◔	○	○	◔	○	○	○	◕	◑	○
QIF	◕	◑	○	○	◑	◔	◔	○	◔	◔
MDF	◔	○	○	○	◑	○	○	○	◔	◑
HDF5	◑	◕	○	○	◑	◑	◑	◑	◔	◔

Summarized in Table 2.2, none of the reviewed technologies or frameworks alone can deal with all deficits and challenges identified in Section 2.2. The most significant gap is the long-term reusability of acquired manufacturing sensor data, namely **D3 Absence of specific criteria for FAIR data management**. The technologies mostly have a system-centric viewpoint, i.e., the requirements and the viewpoint from the data usage perspective are insufficiently considered, leading to significant deficits regarding **C2 Stakeholder heterogeneity** and **C3 Unknown (heterogeneous) applications**. Missing tools supporting the implementation and deployment of measuring systems reveal gaps about **C5 Interdisciplinary deployment**. The isolated data silos are caused by the absence of clearly defined access rules and the missing support to implement these in some of the technologies. The "all-in-one" technologies defining the data model, format, and communication protocol, such as AAS, OPC UA, and MTConnect, already address most deficits and challenges at least partially. The technologies, providing either a data model, format, or communication protocol, only partially address the deficits and challenges. Thus, the solution for achieving *FAIRness* of manufacturing sensor data will require a combination of different standards and technologies to handle the diverse use cases and systems of today's production industry [Drat23]. Hence, it will be required to provide manufacturing sensor data compliant with different (and partially competing or conflicting) standards and data models, such as OPC UA and AAS [Neub23].

2.3.2 Related scientific work

First approaches to resolve these compatibility conflicts exist [Weis23; Cava20]. Still, they usually focus on either making OPC UA compliant with AAS or vice-versa. They do not widen the scope for finding the smallest common core, enabling interoperability with other technologies and frameworks. In addition, the explicit consideration of long-term reusability is mainly at the back of interoperability concerns.

In the advent of *International Data Spaces (IDS)* (formerly *Industrial Data Spaces*) [Otto17b; Otto17a], the interest in the development of *connectors* for devices and micro-services securely providing data has been aroused. Nast et al. [Nast20] proposed a connector based on the OGC SensorThings API and a RESTful API using HTTP and MQTT. Another tool is the *Eclipse Data Space Connector (EDSC)* [Ecli22] for the connectivity of systems compliant with the regulations of *Gaia-X*. Neubauer et al. [Neub23] investigated the combination of AAS, OPC UA and EDSC and proposed a general-purpose architecture based on these three technologies. However, the authors highlight limitations due to the conceptual differences of OPC UA and AAS. This small subset of tools in development shows a fundamental issue of current developments: The representation of data according to standardized data models differs between different approaches. Thus, the interoperability between these technologies is still limited and requires a manual mapping between these models.

Sousa et al. [Sous22b; Sous22a] propose an architecture of CPSs based on OPC UA and QIF for the standardized provision of measurement data and vertical integration of measuring systems into an IIoT. Zhang et al. [Zhan23] developed a wireless machine-to-machine data exchange system based on OPC UA and 5G. Their prototype has proven the envisaged functionality, and the authors plan to extend their system by an AAS-compliant interface.

2.4 Modeling techniques and methods from computer science

Representing manufacturing sensor data in a machine-actionable way and compliant with the *FAIR Guiding Principles* requires a deep understanding of data modeling techniques and methods. This section introduces the basics of the *Semantic Web* and model-driven software engineering (MDSE) as knowledge base from computer science for the conducted research.

2.4.1 The *Semantic Web*

Semantic Web technologies and standards were ideated by Berners-Lee et al. [Bern01] to enable computers and machines to work with and understand data effectively. Three concepts and methods form the base for the *Semantic Web*: (i) the Resource Description Framework (RDF) [Cyga14] to structure information, (ii) persistent and unique identification and localization of resources via persistent URLs (PURLs) [Weib95], and (iii) a controlled vocabulary for consistent data description based on ontologies [Shad06].

Using RDF, a graph is constructed, where each object or concept of interest is stored as a node, and the relations are encoded as directed edges between these nodes. RDF organizes the modeled information in triples, where the source node of an edge is called the subject, the edge itself is called the predicate, and the target node is called the object. This simple but powerful approach allows for the management and representation of data in machine-actionable models. The unique and persistent identification and localization of resources, i.e., the nodes and edges of the RDF triple-graph, guarantees that a concept is always retrievable under the same address - however, the actual location might change. Persistency is usually technically realized using dedicated platforms that maintain look-up and routing tables pointing from a persistent URL to the resource's current location.

According to *Semantic Web* standards, the terms and keywords for data description in the IIoT must be formally defined to ensure data is interoperable and universally understandable [Schl24]. For that, a group of domain experts must agree on the meaning of these terms. One speaks of *controlled vocabulary* if such an agreement and formalization of the definitions exist [Harp13]. According to the Oxford English Dictionary, the term *controlled vocabulary* is in the domain of information science defined as:

Definition 2.7: Controlled Vocabulary

A *controlled vocabulary* is "a prescribed set of terms which may be used in abstracting, indexing, etc., and which is recognized within that system as having a predetermined set of meanings." [Oxfo23a]

Definition 2.8: Controlled Term

A *controlled term* is a term of which the meaning (in a specific context) is formally defined as part of a *controlled vocabulary*.

In the *Semantic Web* context, formalization requires the controlled term to be retrievable by a unique resource identifier and locator and represented using machine-actionable data formats. This formalization is usually done and documented by ontologies using RDF. A key feature of controlled terminology is the unique identification and accessibility via PURL.

While the technical requirements are comparably easy to realize, achieving an agreement on a term's meaning is complex, making developing a formal ontology tedious. This is also because the sense of a term might differ between the domains or even specific applications within a domain. Moreover, if very specific terms are formally defined as a controlled vocabulary, the interoperability is often limited as their scope of usage is too specific [Berm06]. One can avoid introducing new terms by formulating constraints on the relationships and properties of (meta-)data sets to address this lack of specific terms.

2.4.2 Model-driven software engineering

Developing and implementing software (for running production systems) is complex and requires advanced software engineering and programming skills and expertise. To ensure the developed software meets the expectations of all stakeholders, the software is usually abstracted to descriptive models for capturing the software's purpose or structure more conveniently for non-software engineers [Bram12]. MDSE is a sub-discipline of software engineering that makes maximal use of such models by directly translating these models into software source code. Among others, the benefits of software developed with MDSE are higher maintainability, increased modularity, and reduced complexity [Mern05]. The foundation of MDSE is the definition of metamodels for describing the application domain and models for representing individual applications or use cases. The relation between metamodels and models and the resulting metamodel hierarchy is shown in Figure 2.7.

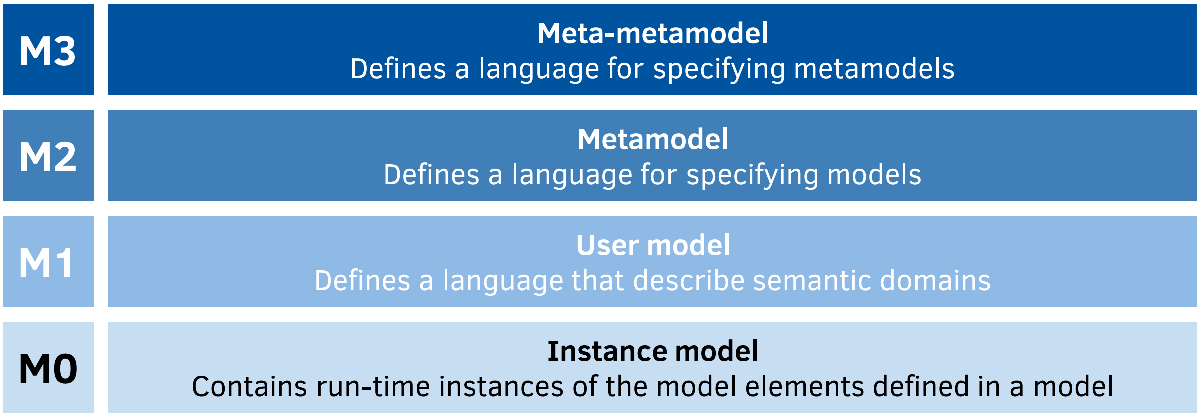


Figure 2.7: The hierarchy of metamodeling as defined by Fettke [Fett05] and the OCUP 2 Examination Team [OCUP17]. Each layer is an instance of the layer above, i.e., a model is an instance of a metamodel.

The central aspect of MDSE is the definition of models for automatic source code generation, called domain-specific language (DSL) [Fowl11], cf. Figure 2.8. Such languages are characterized by high expressiveness and concise applicability. These languages have a high level of abstraction and reduced complexity compared to general-purpose languages (GPLs), such as C++, Python, or Java, which results in increased productivity and reduced maintenance costs for the applications based on these DSLs [Mern05]. DSLs usually define a specific syntax and semantics derived from the application domain [Rodr15]. The concepts and relations of the application domain are picked up by the *abstract syntax* of a DSL. The *abstract syntax* specifies how these concepts are captured and thus forms the metamodel (M2) for the definition of a specific application model (M1) using the DSL. The *abstract syntax* is formalized by the grammar of the DSL. Based on the *abstract syntax*, the *concrete syntax* of a DSL defines how these captured concepts are represented and notated. The *concrete syntax* mainly determines how easily the DSL can be learned and used [Rodr15]. Thus, the *concrete syntax* should be tailored to the domain experts, i.e., the users of the DSL, and mirror familiar terms and ways of representation. The *concrete syntax* can be of different shapes, such as textual, visual, tabular, or a combination of them. To address experts from domains other than software engineering, it is advantageous to design a visual representation that reduces perceived complexity and simplifies usage [Rodr19]. It is also possible to define more than one *concrete syntax* for representing models based on the *abstract syntax*. Compared to GPLs, DSL usually do not have compilers for direct translation into machine-actionable commands. Instead, a model based on DSL is translated to a GPL relying on existing compilers and software ecosystems. Lastly, the *semantics* of a DSL define the rules for the validity and correctness of the models, which can not be expressed by the *concrete syntax*. The *semantics* encode the logic of a well-defined model [Kirc22], which is usually defined using natural language in terms of additional explanations to the DSL's users. Several specialized software tools exist to support developers and users of DSLs. These tools are usually referred to as *language frameworks* or *language workbenches*. Examples of such tools are *JetBrains MPS* [JetB24], *Xtext* [Ecli24], the *Eclipse Modeling Framework* [Stei08], or *MontiCore* [Rump21].

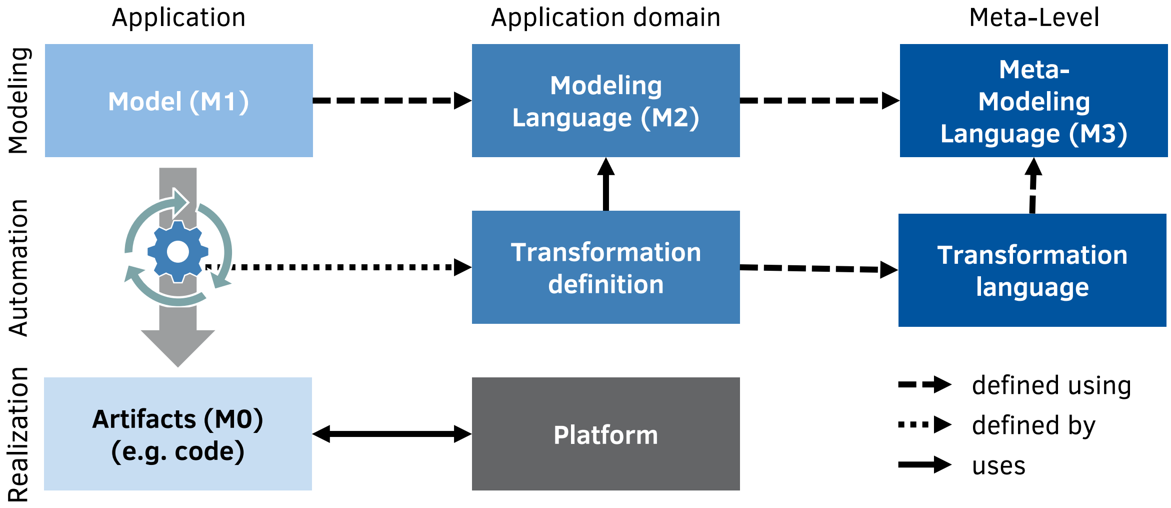


Figure 2.8: Overview over the software development employing model-driven software engineering. Figure has been adopted from Brambilla et al. [Bram12].

DSLs can further be divided into subclasses. The subclasses relevant to this thesis are:

Domain-specific programming languages (DSPLs) are used to develop executable source code for a dedicated purpose with a high level of abstraction. DSPLs contain syntax and semantics for defining processes, tasks, or actions translated into algorithms. A popular example of a DSPL is Simulink [The 24].

Domain-specific modeling languages (DSMLs) are primarily intended to define the structures or models. In contrast to DSPLs, they do not provide an executable logic of procedures. The Unified Modeling Languages (UMLs) are examples of such DSMLs.

Data definition or data description languages (DDLs) are designed to represent data structures, tables, or databases efficiently. Examples for DDLs are SQL, XML, and JSON.

Interface description languages (IDLs) provide concepts for the definition of the functionality, inputs, and outputs of the interface of (micro-)services. IDLs provide translation mechanisms for generating data serialization and transmission routines such that the service developer must not implement the communication interface and logic itself. IDLs are widely used for the development of distributed systems and SOAs, such as FlatBuffers, OpenAPI, Protocol Buffers (protobuf), and the Web Services Description Language (WSDL) [Chin07].

All the mentioned examples belong to category M2 of the metamodel hierarchy, cf. Figure 2.7, and are used to define an application- or device-specific model. Among all these, IDLs promise the functionality required to solve the formulated research questions.

2.5 Intermediate summary

The role of the *FAIR Guiding Principles* for the long-term management of manufacturing sensor data has been analyzed, and four significant deficits of industrial data management have been identified, which underpin the need to implement the *FAIR Guiding Principles* for the manufacturing domain and justify the research objectives and questions formulated in Chapter 1. Moreover, six challenges have been determined that impose hurdles to implementing the *FAIR Guiding Principles* for manufacturing. The current state-of-the-art industrially used data models, data formats, and communication protocols have been reviewed to prepare for developing methods to provide *FAIR manufacturing sensor data* and implement research artifacts, particularly the *FAIR Sensor Service*. The advantages and disadvantages of these frameworks and related scientific approaches have been analyzed concerning implementing the *FAIR Guiding Principles*. To overcome the identified shortcomings, based on *Semantic Web* technologies and MDSE, the foundations of these topics from the computer science discipline have been shortly introduced and reviewed. The reviewed technologies from the production industry and methods from computer science lay the foundation for the investigation of the formulated research question in the remainder of this thesis.

3 *FAIRness* requirements and criteria for manufacturing sensor data

This chapter is dedicated to the first research guiding question, which deals with transferring and concretizing the *FAIR Guiding Principles* for the manufacturing domain. The *FAIR Guiding Principles* in their pure form are very generic. This has the advantage that the principles are broadly applicable to data from various domains. However, the principles themselves are vague, so the realization requires a further definition of specific criteria, which data has to fulfill to be compliant with the *FAIR Guiding Principles* [Dors23; Bahi20; Muse22]. Such specifications have already been proposed and published (e.g., [Wilk18b; Rese20; Deva21]), of which some are available as ready-to-use tools already. But the *FAIR Guiding Principles* and the specifications published so far originate from science and research and, thus, are primarily tailored to scientific data and workflows, which requires a detailed review of published approaches and their applicability to manufacturing sensor data as characterized in Section 3.1. Based on this review, requirements and criteria are defined specifically for measurement data from manufacturing, which must be fulfilled by such to be considered *FAIR*, to answer the first research guiding question:

Research Guiding Question 1

What are the criteria manufacturing sensor data must meet to satisfy the requirements defined by the *FAIR Guiding Principles*?

3.1 Characterization of manufacturing sensor data

Two aspects need to be addressed to characterize sensor data from manufacturing. First, the term *sensor data* needs to be defined; second, the unique features of manufacturing sensor data need to be identified. According to the International vocabulary of metrology (VIM) [BIPM12], a sensor is a piece of hardware capable of perceiving changes in physical quantities. The extent of these changes can be quantified by a *measuring instrument*. This

The author has published parts of the work described in this chapter in

Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. “Model-Driven Development of Interoperable Communication Interfaces for FAIR Sensor Services”. In: *Measurement: Sensors* 24 (2022), p. 100442. ISSN: 26659174. DOI: 10.1016/j.measen.2022.100442

Matthias Bodenbenner, Jan Pennekamp, Benjamin Montavon, Klaus Wehrle, and Robert H. Schmitt. “FAIR Sensor Ecosystem: Long-Term (Re-)Usability of FAIR Sensor Data through Contextualization”. In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. July 2023, pp. 1–8. DOI: 10.1109/INDIN51400.2023.10218149

Robert H. Schmitt, Matthias Bodenbenner, Tobias Hamann, Mark P. Sanders, Mario Moser, and Anas Abdelrazeq. “Leveraging Measurement Data Quality by Adoption of the FAIR Guiding Principles”. In: *tm - Technisches Messen* (July 9, 2024). DOI: 10.1515/teme-2024-0040 perception and quantification procedure is called *measuring*. A device inheriting multiple measuring instruments is called a *measuring system*, which also consists of one of multiple sensor(s). However, thinking application-oriented, it is often not only the bare physical quantity technically measured, which is of interest, but an aggregated property. Considering a barcode reader, one is not interested in the differences in brightness of the reflected light but in the extracted string, which is a non-quantifiable, nominal property. This functional, application-oriented, instead of physical perspective on sensors, is required for achieving interoperability [Mont21]. From the user's perspective, the sensor is not only the hardware but also includes its software digitizing the measurement result, i.e., the CPMS. This perspective leads to Definition 3.1 of sensor data used in this thesis.

Definition 3.1: Sensor data

The digitized values and related metadata produced by a CPMS based on measurements of observed physical quantities of observable properties are further called *sensor data*.

As for almost all industries, the manufacturing sensor data can be considered as *Big Data*. In the literature, *Big Data* is often characterized by what is commonly known as the "V's" of *Big Data*. These characterizations describe *Big Data* with different terms, all starting with "V", such as *Volume*, *Velocity*, *Veracity*, *Value*, *Variety*, and others [Lomo14]. In the considered literature, the number of terms and characteristics vary strongly between different frameworks from initially three [Ziko12] up to 17 [Kapi16]. Among all these aspects, the most prominent ones are the mentioned five: *Volume*, *Velocity*, *Veracity*, *Value*, *Variety* [Lomo14]. In the following, the aspects of *Big Data* are generally analyzed considering the "5 V's" in the context of manufacturing sensor data. Their impact on the deficits and challenges identified in Section 2.2 is summarized in Figure 3.1 and underlined further to tailor the *FAIR Guiding Principles* to the needs of manufacturing sensor data later.

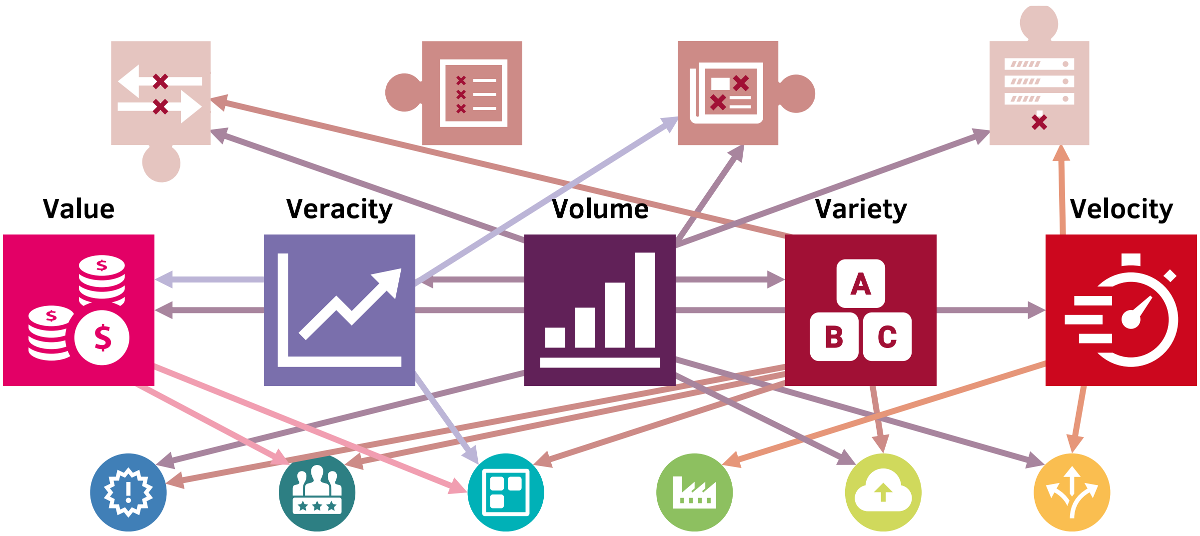


Figure 3.1: The influence of the "V's" characteristics of *Big Data* considerung manufacturing sensor data to the deficits and challenges identified in Section 2.2 (cf. Figure 2.2 and Figure 2.3). *Volume* and *Variety* contribute most to the identified challenges.

Volume

The *Volume* of *Big Data* refers to the exponentially increasing amount of data generated, processed, and stored [Sadi23; Stav23; Moor15]. Statista estimates the globally created *Volume* for 2026 to 230 Zettabytes [Stat23], which is almost one thousand times the overall existing *Volume* estimated for 2016. The increasing *Volume* of manufacturing sensor data imposes data handling, storing, and processing challenges. In particular, the issues induced by other *Big Data* characteristics, such as *Variety*, *Veracity*, and *Value*, are intensified with increasing *Volume*, e.g., decreasing *Value* of individual data sets,[Kais13]. In addition, a high amount of manufacturing sensor data forms a continuous data stream with no beginning and no end in contrast to an enclosed data set [Kais13] and, therefore, heavily impacts all data management and *FAIRification* measures (cf. Figure 3.1). Consequently, handling the large *Volume* of generated data can only be carried out automatically.

Velocity

The *Velocity* refers to the high and unpredictable speed of data generation, transmission, and processing [Sadi23; Owai31; Kais13; Li15]. Increasingly powerful transmission hard- and software (such as fibers and 5G) enable the dissemination of large amounts of sensor data in a short amount of time, causing the communications middleware to be the bottleneck of the data transmission [Kais13], which favors the creation of data silos to reduce the amount of potentially unnecessarily transmitted data. Moreover, the *Velocity* can be associated with the need for immediate data consumption to enable real-time applications [Moor15] which highly impacts **C6 Distributed data**. Thus, handling the highly and largely generated amount of data requires large computation power. Pulling computationally cheap operations, such as the accumulation of metadata and context information to the edge or even into the CPMS could relieve the central computation middlewares as the comparably small and slowly changing amount of metainformation only slightly increases the amount of transmitted data.

Veracity

The *Veracity* is influenced by the varying quality, trustfulness, and authenticity of *Big Data* [Sadi23; Krot15; Moor15]. To ensure authenticity and trustfulness, the preservation of data provenance is crucial [Kais13], implying the need for data documentation and contextualization. The harsh environment of manufacturing yields noisy values and often decreased accuracy and correctness [Wang22; Li15]. Moreover, biased and imbalanced data influence the *Veracity* [Owai31], Consequently, the quantification of the reusability in terms of the measurement uncertainty [Luko13] and the completeness [Reim19] is crucial to indicate the sensor data quality. However, the first is restrained due to missing or not integrated methods and tools for uncertainty determination [Jin15; Gao20]. Thus, the *Veracity* of manufacturing sensor data strongly influences the reusability. Nevertheless, automation of the data acquisition, i.e., of the measurement processes and data transmission, increases the authenticity of the data and thus positively influences the *Veracity* [Moor15].

Variety

The *Variety* describes the heterogeneity of *Big Data*, which includes highly varying kinds, sources, types, descriptions, and representation of the data and is the dominant aspect of manufacturing sensor data [Gao20; Wang22; Stav23; Li22; Kess20]. Many approaches distinguish between structured, semi-structured and unstructured data [Cui20; Gao20; Moor15; Stav23] (cf. Section 2.1.3). Manufacturing sensor data can be considered unstructured, as they often do not explicitly include the structure within the transmitted data packages or sets. Moreover, the structures or models of proprietary data are not always provided by the manufacturers [Libe15], or the structures differ from manufacturer to manufacturer [Sadi23], and sometimes even from device to device as there are often no standards even within companies [Khan16]. This causes significant challenges to the data management and processing [Kais13] and requires specific interoperability criteria to unify structures and terminology. The multitude of different signals sensed and quantified results in a wide plethora of individual measuring systems, which constitute a highly heterogeneous set of data sources and consequently data formats [Sadi23; Wang22; Wang18a]. Also the processes or systems, this data originates from are highly diverse [Libe15; Wang22; Li15]. Lastly, the *Variety* of *Big Data* can also be associated with a variance within the data itself, yielding the information content [Moor15], making the data valuable in the first place.

Value

The *Value* of *Big Data* denotes the worth and profit that can be generated out of the data acquired, stored, and analyzed [Sadi23; Stav23; Kais13]. In contrast to the other characteristics described, the *Value* of data is hard to quantify and depends on multiple factors [Stav23]. Due to the imbalance of data generated in manufacturing (or industrial processes in general) [Wang22], only a smart part of the generated data can be reasonably used [Li20]. Next, long-term applications are often out of focus at the time of data generation. The *Value* of the data for stakeholders interested in long-term applications, such as global optimization, predictive maintenance or quality, or recyclability assessments, is often not considered because the return of investment by the primary short-term usage is considered more important [Moor15]. Thus, usability beyond short-term usage is not often considered, and the acquired data is neither documented nor contextualized. Therefore, it can be regarded as lost [Tanh19]. Last, the *Value* of data is compromised by the high economic and ecologic cost of storing enormous amounts of data [Thiy15].

Conclusion

Several challenges arise when analyzing the “5 V” of manufacturing sensor data concerning the *FAIR Guiding Principles*. The *Volume* of data complicates both findability and accessibility, as massive datasets require robust indexing and efficient retrieval systems. The high *Velocity* of data generation hinders real-time interoperability unless data processing can be streamlined (cf. challenge **C6 Distributed data**). Ensuring sensor data’s *Veracity*, especially under uncertain conditions, affects its reusability since poor data quality can lead to misleading insights. With its diverse formats and structures, the *Variety* of sensor data poses a major obstacle to interoperability without standardized approaches. Finally, data’s *Value* is often underutilized, especially for long-term purposes, because insufficient documentation and contextualization limit its future reusability. Thus, manufacturing sensor data characterized as *Big Data* requires domain-specific criteria for standardized data management seamlessly integrating with the data life-cycle to ensure high quality and valuable data.

3.2 Criteria for *FAIR manufacturing sensor data*

This section defines domain-specific criteria for the *FAIRness* of manufacturing sensor data, based on the characterization of manufacturing sensor data to complement and extend domain-agnostic requirements and metrics reviewed [Wilk18b]. The main motivation for these criteria is to define how the information and data quality must be preserved to allow for the reusability which must be evaluated for each application individually [Kess20]. Enabling comprehensible and traceable data (re-)use requires the identification and coordinated, known, accessible and unambiguous description of all information sources [Miel15] based on these specific criteria.

3.2.1 Available metrics to quantify the *FAIRness* of data and metadata

In the scientific domain, the *FAIR Guiding Principles* are turned into objectively measurable metrics. These are often based on intermediate concretization into dedicated requirements [Deva21]. Thus, metrics represent even more specific requirements, which are investigated to derive the domain-specific *FAIRness* criteria of manufacturing sensor data.

Since the publication of the *FAIR Guiding Principles*, many projects have been conducted and working groups or organizations have been formed in order to foster the aggregation and management of data compliant with the *FAIR Guiding Principles*. Some of these resulted in the specification of explicit rules and metrics for assessment of the *FAIRness* of data. When writing, the author is unaware of any publication of domain-specific *FAIRness* metrics or criteria for the manufacturing domain. Published generic requirements have been reviewed to define specific *FAIRness* requirements for the manufacturing domain. All metrics designed for a dedicated domain unrelated to production are not included in the evaluation, as those approaches are not expected to be suitable for the considered domain. Four different approaches have been investigated and compared in detail: (i) the *FAIR* Metrics first introduced by Wilkinson et al. [Wilk18b] and later extended in [Wilk19b], (ii) the *FAIR* Data Maturity Model (*FAIR* DMM) developed by the Research Data Alliance (RDA) [Rese20], (iii) the FAIRsFAIR metrics proposed by Devaraju and Huber [Deva21], and (iv) the metrics from the base rubrics of FAIRshake [Clar19].

Putting together all four frameworks in Table 3.1, 147 metrics are defined (42 targeting findability, 31 targeting accessibility, 35 targeting interoperability, and 39 targeting reusability). The approaches have been compared and the individual metrics have been mapped to retrieve a set of distinct metrics and criteria. 61 metrics remained after this step (15 targeting findability, 12 targeting accessibility, 14 targeting interoperability, and 20 targeting reusability). In the third step, 15 metrics specific to different domains are excluded to finalize the basis for the definition of the domain-specific *FAIRness* criteria in Section 3.2.2. The four reviewed frameworks are analyzed in detail in the following.

Table 3.1: Overview of the number of metrics identified in the state-of-the-art scientific literature [Wilk19a; Deva22; Rese20; Clar19]. The **x* rows contain the number of metrics for each of the four base principles, which have not been assigned to a dedicated guideline. From left to right the columns show the number of metrics, before any filtering and mapping, after mapping an elimination of redundant metrics, and after the filtering of unsuitable ones. The summary of all 46 distinct, applicable metrics is given in Appendix A, namely Tables A.1 to A.4.

	Unfiltered metrics	Distinct metrics	Applicable metrics
F1	14	4	4
F2	8	3	3
F3	7	2	2
F4	6	2	1
Fx	7	4	1
Sum for F	42	15	11
A1	11	6	6
A1.1	7	2	2
A1.2	6	2	2
A2	5	1	1
Ax	2	1	0
Sum or A	31	12	11
I1	15	4	4
I2	9	3	3
I3	10	6	6
Ix	1	1	0
Sum for I	35	14	13
R1	6	1	1
R1.1	6	3	3
R1.2	3	2	2
R1.3	7	4	4
Rx	17	10	1
Sum for R	39	20	11
Overall Sum	**147**	**61**	**46**

FAIR Metrics Framework

One problem when evaluating *FAIRness* of data is that it might differ between different communities or domains. The *FAIR* metrics framework by Wilkinson et al. [Wilk18b] provides domain-agnostic metrics that can be used by a semi-automated process to evaluate any digital resource. These metrics can also be used as templates by communities to derive new metrics that are tailored specifically to the community's domain [Wilk18b]. The first version of the framework provided 14 universal metrics, which have later been extended to 22 metrics [Wilk19b] and published on GitHub [Wilk19a]. To fully utilize the results of the framework, all *FAIRness* evaluation results also adhere to the *FAIR Guiding Principles*.

Research Data Alliance *FAIR* Data Maturity Model

The *FAIR* Data Maturity Model (*FAIR* DMM) framework [Rese20] developed by the Research Data Alliance (RDA) provides a checklist-like approach to evaluate the *FAIRness* of data manually. Manual evaluation enables the evaluator to take the individual requirements and needs for the *FAIR* evaluation into account [Rese20]. However, a fully objective assessment can not be guaranteed if the *FAIRness* can not be accessed autonomously. For the evaluation process, the *FAIR* DMM framework provides a set of indicators, each measuring a specific aspect of the *FAIR Guiding Principles*. Thus there can be multiple indicators for a single guideline. The *FAIR* DMM defines separate metrics and indicators for the data and metadata if a *FAIR Guiding Principles* addresses both.

Additionally, the indicators are prioritized on a three-level scale representing the indicator's importance for achieving *FAIRness*. Without fulfilling *essential* indicators (meta-)data can not be considered *FAIR*. *Important* indicators are not as crucial but substantially increase the *FAIRness*. *Useful* indicators are not strictly required to achieve high *FAIRness* but help achieve the highest compliance with the *FAIR Guiding Principles*.

The *FAIR* DMM defines 41 indicators of which 19 are duplicates of metrics of the *FAIR* Metrics Framework. These duplicate metrics are mostly categorized as *essential* or *important*. 14 indicators do not share strong similarities with metrics from other approaches because of the high degree of granularity of these indicators.

FAIRsFAIR Metrics

Based on early versions of *FAIR* Metrics and *FAIR* DMM the FAIRsFAIR Metrics have been developed by Devaraju and Huber [Deva21]. The most current version 0.5 of the FAIRsFAIRs Metrics was published in 2022 [Deva22] and consists of 17 domain-agnostic metrics for evaluation of the *FAIRness*. To evaluate the metrics they are split into multiple practical tests for evaluation which can also be executed autonomously. The FAIRsFAIR Metrics do not add any distinct metrics on top of the *FAIR* Metrics and the *FAIR* DMM and is thus fully interoperable and compliant with the other approaches. The main contribution of Devaraju and Huber [Deva21] is the development and proposal of the *F-UJI* tool for autonomous evaluation of data *FAIRness*.

FAIRshake Metrics

Clarke et al. [Clar19] introduced the *FAIRshake toolkit* for facilitating the community-specific development of *FAIRness* metrics, and for assessing the *FAIRness* of digital resources based on these community-specific metrics. This section focuses on the first, while the latter is elaborated in detail in Chapter 6. Clarke et al. [Clar19] describe how to define metrics using their toolkit and associating a metric with the *FAIR Guiding Principles*. Moreover, *FAIRness* metrics can be assigned to *rubrics* that should represent a set of metrics suitable for a dedicated domain or project. Thus, single metrics can be easily and individually reused by others. Metrics created and added to the FAIRshake toolkit are evaluated manually using an ordinal scale, which is quantified to a numerical scale from zero to one, where one corresponds to the complete fulfillment of a metric.

The developers of FAIRshake specify three rubrics for assessing the *FAIRness* of datasets[1], repositories[2], and tools[3]. The metrics of these three rubrics are more specific compared to the three domain-agnostic approaches described above. Therefore, some of the metrics are too specific to be reused for deriving *FAIRness* criteria of *FAIR manufacturing sensor data*.

Conclusion

The majority of tools for *FAIRness* assessment originate from the life sciences. Moreover, the tools are dedicated to evaluating research data and rely on common tools, repositories, and data descriptions used in research. Thus, some specific criteria of the measurable metrics are not suited for assessing the *FAIRness* of sensor data from industrial production. The majority of metrics and frameworks published so far are domain-agnostic, such as the approaches reviewed above. This becomes evident by considering that only the FAIRsFAIR Metrics and the Research Data Alliance *FAIR* Data Maturity Model formulate metrics or indicators, respectively, for guideline **R1.3** demanding “domain-relevant community standards” [Wilk16]. However, due to the domain-independence, the metrics are not more specific on how these community standards need to be expressed. Moreover, many of the metrics are hardly more specific than the *FAIR Guiding Principles* themselves in terms of guiding the data stewards in creating *FAIR* data. Thus, to provide *FAIR manufacturing sensor data* the domain-agnostic metrics need to be checked for applicability in the manufacturing domain. Moreover, these metrics must be extended by related domain-specific requirements. Furthermore, specific requirements must be defined addressing **R1.3** in particular.

Nonetheless, the formal establishment of widely applicable community or domain standards is out of the scope of this dissertation. Within this work, it is investigated how widely known standards and those that are already formally defined can be implemented to acquire *FAIR manufacturing sensor data*. For that, some of the reviewed metrics, indicators, or practical tests need to be relaxed into less strict criteria as illustated in the following: Many metrics measure the fulfillment of the *FAIR Guiding Principles* by looking for specific controlled terms from selected vocabularies or ontologies (cf. metric “Data identifier explicitly in metadata” [Emon20]) in the assessed dataset. On the one hand, this fosters interoperability by

[1] `https://fairshake.cloud/rubric/8/`
[2] `https://fairshake.cloud/rubric/9/`
[3] `https://fairshake.cloud/rubric/7/`

forcing to use of a small set of terms only, which are then reused on a large scale instead of having everyone using different terms. On the other hand, this limits flexibility and prohibits the usage of other, but fitting terms. Finally, the 46 distinct, suitable metrics identified in related work are used as basis for the derivation of specific *FAIRness* criteria for *FAIR manufacturing sensor data*.

3.2.2 Derivation of *FAIRness* criteria for manufacturing

The domain-specific criteria of *FAIR* for the manufacturing domain do not necessarily need to be based on measurable metrics, which allow automatic and objective evaluation of the *FAIRness* of manufacturing sensor data. Nevertheless, defining criteria that can be turned into metrics is favorable for developing automatable compliance tests later. For the design of *FAIRness* metrics, Wilkinson et al. [Wilk18b] defined five criteria a "good metric" [Wilk18b, p. 2] should fulfill: clear, realistic, discriminating, measurable, and universal. A clear metric is easy to understand, and the purpose of the metrics is comprehensible. A metric is realistic if a digital resource can fulfill it without overly complex overheads. A metric is discriminating when the degree of fulfillment can be quantified and if it can provide information and hints on how to increase the degree of fulfillment. Machines can objectively and autonomously quantify measurable metrics. A universal metric can be applied to all types of digital resources.

This dissertation defines criteria that set the bounding conditions for the model for *FAIR manufacturing sensor data*. The summary of all 17 criteria defined is depicted in Figure 3.2. Some criteria target the data management system instead of the data itself. The criteria follow the requirements proposed by Wilkinson et al. [Wilk18b]. Only the need to be measurable has not been considered, as the purpose of the criteria is not to provide a framework for objectively quantifying the *FAIRness* of manufacturing sensor data.

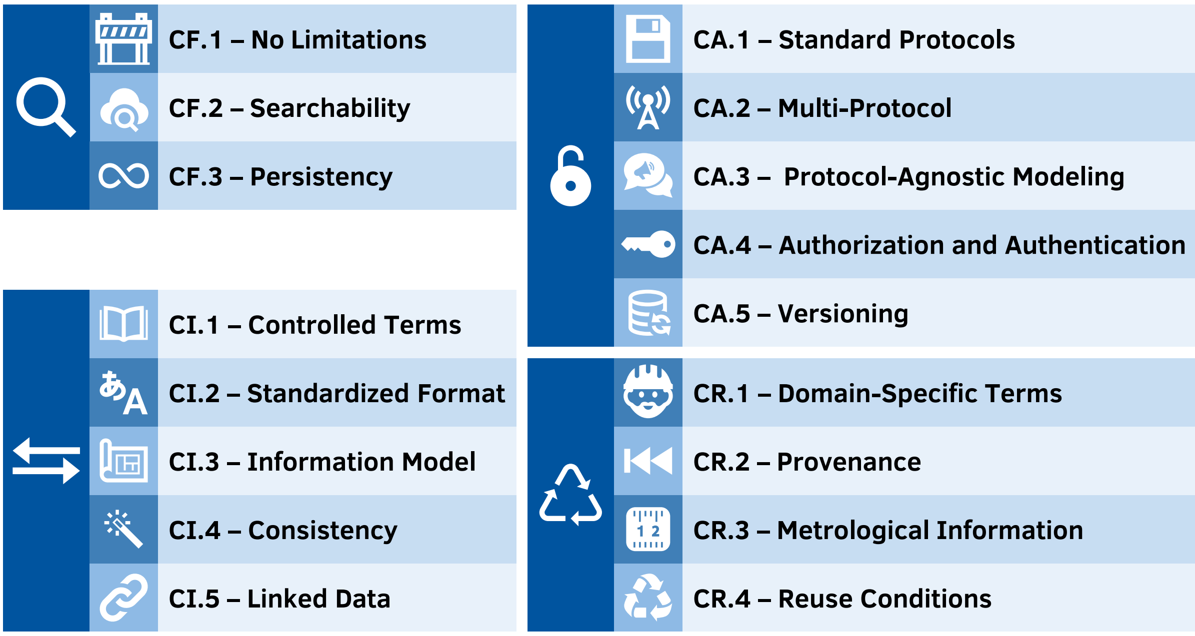


Figure 3.2: Summary of the 17 *FAIRness* criteria defined in this thesis focusing on accessibility and interoperability. The mapping of the individual criteria to the *FAIR Guiding Principles* is done in Table 3.2.

Findability

CF.1 No Limitations According to guideline **F2** data must be "richly" [Wilk16] described with metadata. Whereas the notion of "richness" in this context is hard to quantify [Muse22], it does imply that there should be no limits for the metadata used to describe the data. Thus, instead of requiring a certain amount of metadata to be provided, this criterion enforces that no conceptual and technological limits are introduced. This concerns the amount, type, and kind of metadata. Besides that, data and metadata must be modeled and described according to criteria **CI.1 Controlled Terms**, **CI.3 Information Model**, and **CR.1 Domain-Specific Terms**; there should be no further restrictions.

CF.2 Searchability of stored data is critical to ensure the retrievability of the data [Muse22]. To fulfill guideline **F4**, data must be stored in a way so that it can be filtered by any field. This search procedure must allow using any key from ontologies or terminologies as the search query term. Thus, the search must also include the metadata fields, which can store any type as per criteria **CF.1 No Limitations** and **CI.4 Consistency**. This criterion is required to cope with **C2 Stakeholder heterogeneity**.

CF.3 Persistency Upon the guidelines **F1** and **F3**, in case the configuration, setup, or relations between objects change, the identifiers of data must be persisted. Thus, the assignment of an identifier should be decoupled from its name and description, which might change over time.

Accessibility

CA.1 Standard Protocols To disseminate the data from one participant to another and unlock data silos (see **D4 Isolated data silos**), standardized - and ideally open - communication protocols must be used. These protocols must be suitable for use in the considered manufacturing domain and should also have notable recognition.

CA.2 Multi-Protocol To be accessible across all layers of the automation pyramid and by different systems or applications, data provision via one fixed protocol is not sufficient [Derh17a; Cånd10]. The high fluctuation and velocity of communication and data management technology developments will yield new standards that need to be adopted. So, multiple different protocols must be supported to ensure the accessibility of the sensor data across various systems (cf. **C4 System heterogeneity**).

CA.3 Protocol-Agnostic Modeling To assure accessibility, the description, and annotation of data and metadata according to interoperability criteria must be agnostic of the protocols used to provide and distribute the data. Although not explicitly stated by the *FAIR Guiding Principles* and the reviewed metrics (see Section 3.2.1), a coupling of description and communication limits the accessibility and impedes **C5 Interdisciplinary deployment**. Primarily, if clients try to access the data but cannot implement the intended communication protocol due to technological restrictions, this is a direct requirement for **CA.2 Multi-Protocol**.

CA.4 Authorization and authentication Preserving the sovereignty of acquired data is essential in industrial data management. Thus, the used communication protocols must implement proper authentication and authorization mechanisms to overcome **D4 Isolated data silos**.

CA.5 Versioning To fulfill guideline **A2** metadata must not be deletable. Instead, a new metadata revision should be linked to the older version. Thus, the data management system must implement a versioning mechanism for metadata.

Interoperability

CI.1 Controlled Terms Terms used as keywords and field descriptors in the data model must be thoroughly defined by available and accessible vocabularies or ontologies using *Semantic Web* standards to fulfill guideline **I1** and diminish **C1 Data heterogeneity**.

CI.2 Standardized Format For representing the data, a standardized, open data format must be used to mitigate **C1 Data heterogeneity**.

CI.3 Information Model Data must be defined using a common, accessible information model so that data can be easily exchanged and unambiguously interpreted [Otto17b]. This information must be defined according to criterion **CI.1 Controlled Terms** and exchanged according to criteria **CA.1 Standard Protocols** and **CI.2 Standardized Format**.

CI.4 Consistency Data of the same type must always be modeled equally to prevent ambiguity and preserve interoperability, i.e., two models describing the same thing must have the exact attributes and use the same terms for describing these attributes. The definition of consistent models allows the implementation of automated compliance checks to guarantee interoperability [Miel15]. Consistency is particularly relevant to face **C1 Data heterogeneity**, **C2 Stakeholder heterogeneity**, and **C3 Unknown (heterogeneous) applications**.

CI.5 Linked Data Data must refer to other data or metadata that are related to it. This related (meta-)data can be data about individual components of an assembled product, data from suppliers, or context data about the process or used resources [Otto17b]. Following guideline **I3**, these references must be qualified, i.e., the reference must be defined appropriately using terms, usually verbs, for relations from ontologies or vocabularies. Linking data with other data effectively tackles **D2 Poor contextual embedding**.

Reusability

CR.1 Domain-Specific Terms According to criterion **CI.1 Controlled Terms**, the definition of terms must originate in the domain in question [Muse22], i.e., the manufacturing domain, to meet guideline **R1.3**. Even more precise, terms related to measurements and their characteristics must originate in production metrology.

CR.2 Provenance To ensure an unambiguous interpretation of the data and tackle **D2 Poor contextual embedding**, the provenance and context of the data must be captured [Kess20]. For manufacturing sensor data, this context can be given by the product produced, the processes executed, and the resources used when the measurement took place [Pint13; Schl09]. This criterion is closely related to **CI.5 Linked Data**.

CR.3 Metrological Information Domain-specific *FAIRness* requirements from metrology are required to ensure reusability from a quality management perspective. Measurement uncertainty is one of the most important quality criteria for measured values. Thus, to ensure conformance with metrological standards and evaluate measured values regarding their reuse potential, measured values must always have a quantified specification of their uncertainty [Eich23; Libe15; BIPM12]. Further information required are the physical unit and the type of the quantity observed (cf. [Ever15; BIPM12; Hutz20]). This criterion formalizes the most important domain-specific requirement concerning **D2 Poor contextual embedding** and **D3 Absence of specific criteria for FAIR data management**.

CR.4 Reuse Conditions Data must be annotated with reuse conditions to enable reuse and ensure data sovereignty [Kess20]. Expressing a usage license and usage conditions is required to establish *International Data Spaces* [Otto17b].

These 17 criteria for *FAIR manufacturing sensor data* cover all aspects of the *FAIR Guiding Principles* and formulate further requirements not directly implied by the *FAIR Guiding Principles* as summarized in Table 3.2. The criteria substantiate the *FAIR Guiding Principles* by considering specific aspects relevant to the implementation and their interpretation in the manufacturing domain. In particular, the accessibility guidelines are extended by defining concrete criteria for implementing the communication interface to exchange manufacturing sensor data, namely **CA.2 Multi-Protocol** and **CA.3 Protocol-Agnostic Modeling**. Criterion **CR.3 Metrological Information** defines "domain-relevant community standards" [Wilk16] as required by **R1.3** for the domain of production metrology. Moreover, the derived criteria address the deficits and challenges identified in Section 2.2 and, therefore, guide the subsequent course of research to implement the *FAIR Guiding Principles* for the manufacturing domain and lift the value-creation potential of manufacturing sensor data. Next, a harmonized data life-cycle for managing manufacturing sensor data is introduced to set up a reference frame for the implementation of the defined *FAIRness* criteria.

3.3 A harmonized data life-cycle for managing *FAIR manufacturing sensor data*

Data life-cycles (DLCs) serve as a conceptual model to structure the handling and management of data [Pouc16]. However, the design and the steps of DLCs depend on various factors. First, the design depends on the domain, such as research and science, compared to industry. Second, the applications of the data in question and the purpose of the DLC itself impact the DLC's design, i.e., which overarching use-case motivated the definition of the DLC, such as data curation, usage of machine-learning, long-term reusability of the data. Third, the perspective, e.g., organizational, conceptual, or technological, influences the DLC's layout. To ensure the provision of *FAIR manufacturing sensor data* from the beginning, the *FAIRification* steps and procedures must be embedded into the industrial data flow. Even more, *FAIRification* must be an integral part of the life-cycle of manufacturing sensor data [Jaco20b]. Therefore, 17 published DLCs have been reviewed to identify the steps that must be included in the harmonized DLC for managing *FAIR manufacturing sensor data*.

Table 3.2: Systematic mapping of the derived *FAIRness* criteria for *FAIR manufacturing sensor data* to the *FAIR Guiding Principles* showing full coverage. Whereas most criteria are directly related to the *FAIR Guiding Principles* (indicated using checkmarks), some define additional requirements in the context of manufacturing that are not formulated by the *FAIR Guiding Principles* (marked with the plus sign).

		Findability				Accessibility				Interoperability			Reusability			
		F1	F2	F3	F4	A1	A1.1	A1.2	A2	I1	I2	I3	R1	R1.1	R1.2	R1.3
Findability	CF.1		✓													
	CF.2				✓											
	CF.3	✓		+												
Accessibility	CA.1					✓	✓									
	CA.2					+	+									
	CA.3						+									
	CA.4							✓								
	CA.5								✓							
Interoperability	CI.1									✓	+					
	CI.2									✓						
	CI.3									✓						
	CI.4			✓						+			✓			
	CI.5											✓				
Reusability	CR.1			✓												✓
	CR.2												✓		✓	
	CR.3			✓												✓
	CR.4													✓		

3.3.1 Review of existing data life-cycles

The individual steps of all reviewed DLCs have been compared and mapped to mitigate conflicts and ensure consistent definition and naming. Based on the explanations of the steps, the set of steps has been reduced to twelve activities, which cover almost all steps defined by the reviewed publications. Steps only described by a singular DLC having no relation to the steps of any other DLC have been removed as they are considered irrelevant. Table 3.3 maps the steps of all reviewed DLCs to the twelve identified data-centered activities. The definitions of the twelve activities are as follows.

Table 3.3: Mapping and coverage analysis of the reviewed DLC models. Checkmark means at least one dedicated step primarily addresses this topic in this framework. Nevertheless, this step might secondarily also cover other steps. Tilde means this topic is addressed, but not at least one step is primarily attributed to this topic alone. Cross means the respective DLC does not explicitly address this topic. The number of mentions of the individual activities is given, excluding the life-cycle proposed by Bodenbenner et al. [Bode22a], which is also part of this doctoral thesis.

	Framework	Planning	**Creation**	**Analysis**	**Description**	**Storage**	**Access**	**Reuse**	Integration	Preprocessing	Visualization	Application	Feedback
Science	ANDS [Burt09]	×	✓	×	✓	✓	✓	✓	×	×	×	×	×
	DataONe [Mich12]	✓	✓	✓	✓	✓	✓	×	✓	✓	×	×	×
	UK Data Archive [UK D24; Ball12]	✓	✓	✓	~	✓	✓	✓	×	~	×	×	×
	DDI 3.3 [Data20; Ball12]	✓	✓	✓	×	✓	✓	✓	×	×	×	×	×
	RDMKit [ELIX21]	✓	✓	✓	~	✓	✓	✓	×	✓	×	×	×
	Schmitz et al. [Schm18b]	✓	✓	✓	~	✓	✓	✓	×	×	×	×	×
	Research 360 [McKe12]	✓	✓	✓	~	✓	✓	✓	×	×	×	×	×
	Curation Lifecycle [Higg08]	✓	✓	×	✓	✓	~	~	✓	×	×	×	×
	Wing [Wing19]	×	✓	✓	✓	✓	×	✓	×	×	✓	✓	×
	Berman et al [Berm18]	×	✓	~	×	✓	✓	~	×	✓	~	~	×
Industry	Gleim et al. [Glei21]	×	~	~	×	✓	✓	✓	×	✓	×	×	×
	Tao et al. [Tao18]	×	✓	✓	×	✓	✓	~	~	×	✓	✓	×
	Zhang et al. [Zhan22]	×	✓	~	✓	~	~	~	×	×	×	×	×
	Ma et al. [Ma22]	×	✓	~	×	✓	×	~	✓	✓	✓	✓	×
	Sebastion-Coleman [Seba22]	✓	✓	×	~	✓	✓	~	×	~	×	×	✓
Misc	Olawoyin et al. [Olaw23]	✓	✓	✓	×	×	×	✓	×	✓	×	×	×
	Möller [Möll13]	✓	✓	×	✓	✓	✓	✓	×	×	×	×	✓
	Addressed by	10	**17**	**13**	**11**	**16**	**14**	**16**	4	8	4	5	2
	Bodenbenner et al. [Bode22a]	×	✓	✓	✓	✓	✓	✓	×	×	~	~	×

Planning In the planning phase, the intended usage, application, and management of the data are designed. The requirements of data and their annotation are determined, and the bounding conditions are examined. It is primarily included in research DLCs and less often in industrial ones, which mainly focus on the actual data flow, which starts with data creation. If planning is included in a DLC, it is always the first step.

Creation The creation process covers all activities which produce data. Sensor data creation represents the measurement process. The step usually contains the digitization of the measured values and sometimes includes a first description with metadata [Schm18b].

Analysis is included in most of the DLCs. Within the analysis step, the data is processed (either manually or with software), and knowledge is derived, or accumulated data is created. Those DLCs that do not include the analysis focus on the curation and maintenance of the data and not on its usage. However, the position within the DLC differs and is not always clearly separated from data reuse, particularly in the industrial DLCs. Moreover, it is not always clearly separated from data preprocessing.

Description of the collected data with metainformation for increasing findability, interpretability, and reusability. This step is one of the crucial activities for achieving *FAIRness* of the collected data [Wilk16]. However, approximately only half of the DLCs mentioning this activity do have dedicated steps for this. On the one hand, the description with appropriate metadata should be tightly coupled with the **Creation**, **Analysis**, and **Access**, making the description activity indistinguishable from the others. On the other hand, the core activity, naming a step composed of multiple activities, tends to be interpreted as the most critical activity. Thus, omitting a dedicated description step in a DLC might cause it to be skipped or ignored.

Storage deals with the persistent preservation of the data in repositories and databases. It also often called “Archiving” [Ball12; Möll13] or “Preservation” [Mich12; ELIX21; McKe12; Glei21; Berm18]. If there is no dedicated step for data **Description** with context and metainformation, storage sometimes implicitly covers this. However, storage is not consistently located. Most industrial DLCs consider storage prior to **Analysis** [Seba22; Ma22; Zhan22]. Also, some of the research DLCs consider storage prior to **Analysis** [Wing19; Mich12], but some do not [ELIX21; Schm18b; Berm18; McKe12].

Access always takes place after **Storage** and includes all activities of sharing and publication, i.e., enabling others to access the stored (meta-)data. According to the *FAIR Guiding Principles*, it does not enforce openly available data. *Access* can briefly be summarized as making others aware of the data and providing an endpoint to retrieve it directly or request retrieval. While almost all scientific DLCs separate access from **Storage** (except for [Higg08]), industrial DLCs are less strict [Zhan22; Seba22].

Reuse of data might be carried out by the original data creator or by others. In most DLCs, reuse builds upon accessing the stored data without direct temporal relation to the original creation. Moreover, the data might be reused for a different purpose than initially created. Not all DLCs separate reuse from **Analysis** [Zhan22; Higg08; Ma22].

Integration includes the alignment of differently formatted or structured data. In contrast to **Description**, this activity does not operate on the conceptual and terminology level but on the technological level. The activity also includes the transformation of data from one data format to another or the integration into a different storage or data management system (such as Enterprise Resource Planning (ERP) or Manufacturing Execution System (MES)). In particular, it does not cover the data models’ extension, modification, or mapping, i.e., it does not touch the terms used to describe the data. The activity is rarely included in the reviewed DLCs.

Preprocessing covers multiple different activities, which can broadly be summarized under this term, such as data filtering, cleaning, or quality checking.

Visualization of the data to be human-actionable is mostly not considered as an individual activity of the DLC. Nevertheless, those DLCs with a solid technological focus [Tao18; Ma22; Wing19] distinguish the visualization from the **Analysis** and **Reuse**, implicitly introducing a separation of manual and automatic data processing.

Application is a dedicated step of a few DLCs only. It covers the usage or interpretation of analyzed data. Within the application activity, explicit actions are derived and executed based on the data, such as discarding defective parts. Although it can not always be sharply distinguished from **Reuse**, the application of analyzed data usually has a direct temporal relation to the **Creation** and **Analysis** of the data.

Feedback Only two of the reviewed DLCs consider collecting feedback on the data itself [Möll13; Seba22] to improve the description and thus, the *FAIRness* of the data.

The identified activities describe distinguishable steps of a harmonized DLC (cf. Schmitt et al. [Schm24]) and depicted in Figure 3.3. The steps **Integration** and **Visualization** are often coupled with the steps **Storage** and **Application**, respectively. At least two-thirds of the reviewed models mention six of these activities, namely **Creation**, **Analysis**, **Description**, **Storage**, **Access**, and **Reuse**. In the following, these six are considered the core activities and form the base of the life-cycle of *FAIR manufacturing sensor data* (*FAIR* DLC). However, the activities **Description** and **Reuse** dealing with crucial principles of the *FAIR Guiding Principles* are often not explicitly considered (cf. Table 3.3). Significantly, the need to extend the description of the data provenance when fed back into **Storage** is often neglected. Moreover, the reviewed DLCs usually lack the technological perspective of data management following the data flow. They do not consider the boundaries of the individual services producing, processing, storing, or consuming the data, e.g., the step **Creation** where the sensor data is acquired is usually carried out with CPMS inline or at the edge, while **Storage** is handled by a cloud-based middleware. Thus, the **Access** must already be considered at edge level, i.e., between these two steps. Finally, the complexity of the harmonized DLC lifts the barrier for operationalization of the *FAIR Guiding Principles*. The DLC must focus on the aspects of the *FAIR Guiding Principles* and the *FAIRness* criteria and reflect the actual data flow, considering the interfaces of the individual software modules to facilitate implementation.

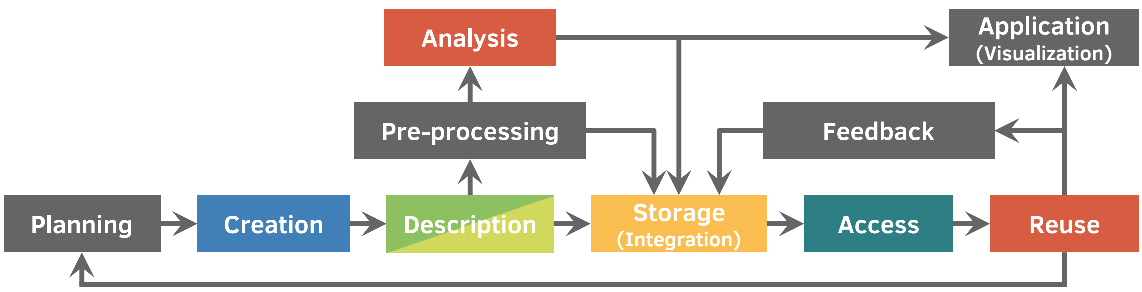


Figure 3.3: A harmonized data life-cycle as proposed by the author in Schmitt et al. [Schm24]. The life-cycle resembles the twelve steps identified. The color scheme maps each step to the life-cycle of *FAIR manufacturing sensor data* proposed in Section 3.3.2 and depicted in Figure 3.4.

3.3.2 The life-cycle of *FAIR manufacturing sensor data*

The life-cycle of *FAIR manufacturing sensor data* (*FAIR* DLC) consists of the six core activities of DLCs identified in the literature and adopts a technological perspective to facilitate operationalization and interconnection of the tools supporting the individual steps. As depicted in Figure 3.4, the *FAIR* DLC is split into two parts to distinguish between the steps that can or must be executed decentralized, by or close to the CPMS, and the steps executed centralized by a middleware at edge or cloud layer. The *FAIR* DLC focuses on the initial data creation and transmission up to the ingestion and preservation in a storage system. Because the **Description** step is divided into one decentralized and one centralized step, the *FAIR* DLC consists of seven steps, described in the following.

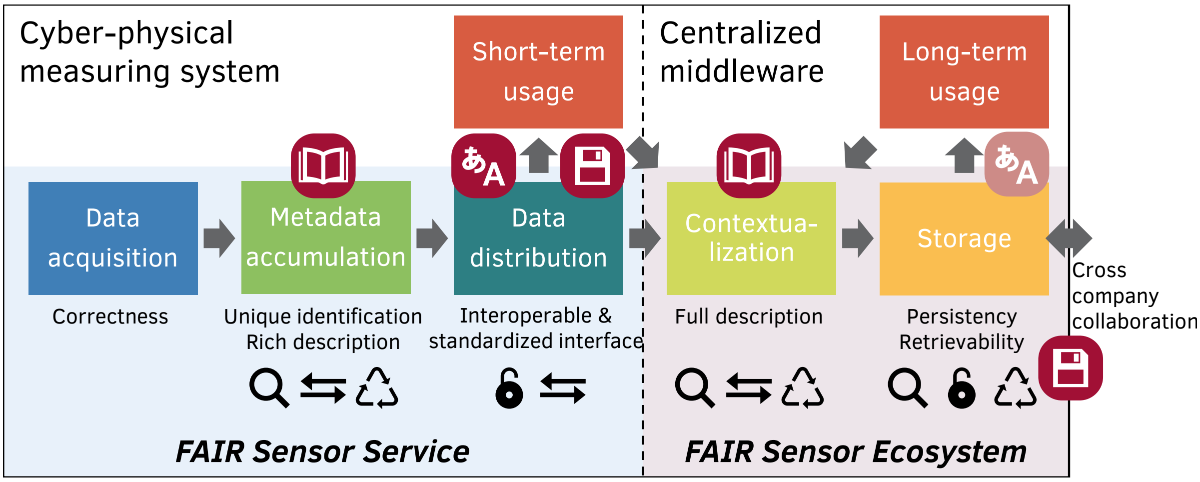


Figure 3.4: The life-cycle of *FAIR manufacturing sensor data*. The relevant data management concerns are stated below each step. The main *FAIR Guiding Principles* (cf. Figure 1.3) and the levels of interoperability (cf. Definitions 2.3 to 2.5) are mapped to each step. The steps at the device and edge level are covered by the *FAIR Sensor Services* and *FAIR Sensor Ecosystem*, respectively.

Data acquisition The life-cycle starts with the **Creation**, called *Acquisition*, focusing on the physical measurement and digitization of observable properties and quantities.

Metadata accumulation Following the servitization principle and **C6 Distributed data**, the **Description** with metainformation of the measuring system itself must be carried out by the CPMS itself to ensure findability and reusability. This metainformation also includes the parametrization of the device. The measurement value's metainformation is accumulated and attached to the measured value. A standardized data description model based on controlled terms must be employed to ensure interoperability. Most importantly, provenance information must be collected to foster reusability.

Data distribution targets the **Access** activity in the decentralized part of the *FAIR* DLC. In this step, richly described measurement data is provided via an interoperable communication interface using standardized and open data format and communication protocols so that accessibility can be guaranteed. The destination of provided data can not be determined beforehand. The data could be directly used in arbitrary applications or transferred into the centralized middleware.

Short-term usage of acquired, described, and distributed data can not be comprehensively classified or determined beforehand. The data can be used to control a process, check the quality of a produced part, monitor a system or process, or do other things. It can be seen as part of the **Analysis** activity but does not solely represent it.

Contextualization completes the **Description** activity at the centralized middleware layer by linking the distributed (and already used) measurement data to the metainformation of the entities related to the measuring system. The entirety of related entities can hardly be covered in general, but examples of major entities of the context are (i) the product the measured quantity is associated with, (ii) a tool or resource that interacts with the product directly before or at the time of the measurement, (iii) the overall process, the sensor, product, and tool are integrated in, and (iv) if the process is not fully automated, workers either operate the sensor, perform the measuring process, or perform a production step. This context data enhances the findability and reusability of the measurement data. As for **Metadata accumulation**, controlled terms are required to ensure interoperability of the context data.

Storage All measurement and context data is ingested into a persistent database. The data is appropriately registered to be findable and accessible for **Long-term usage**. The step also includes the **Access** activity of the centralized middleware by employing proper communication protocols and data formats. Assigning unique and persistent identifiers is required to assure findability.

Long-term usage corresponds to the **Reuse** activity. It covers all data usages that do not necessarily have a direct temporal relation with the initial data **Data acquisition**, **Data distribution**, and **Short-term usage**. The wholeness of long-term applications is not known at the time of **Data acquisition** (c.f. **C3 Unknown (heterogeneous) applications**). Thus, the contextualization of the measurement data is crucial. Consequently, long-term applications only retrieve data from the central storage system. Before computed results of long-term usage are fed back into the storage system, it is again contextualized to ensure the provenance is preserved and the traceability of the data is guaranteed. Long-term usage includes the **Application** as defined above, i.e., the interpretation of the data, derivation decisions, and the execution of actions accordingly. Moreover, **Long-term usage** usually includes **Visualization** enabling these applications.

The **Planning** stage is excluded, as this takes part once before the deployment of a sensor and is thus not part of the data flow itself. Nevertheless, proper planning, i.e., definition and maintenance of the sensor and contextualization models, is required to enable correct and *FAIR* **Metadata accumulation** and **Contextualization**. In some sense, the research and developments for this doctoral thesis can be considered part of the **Planning** activity. The specific execution of the **Integration** activity depends on the storage systems chosen and the applications in which the data is used. Specialized analysis software might require individual data structures to which *FAIR manufacturing sensor data* must be transformed. Analogously, the data **Preprocessing** depends on the applications and, therefore, must not be carried out generically in beforehand. Any modification before the data **Storage** of the measurement data is a kind of **Short-term usage** and might affect or restrict **Long-term usage**. The **Contextualization** and **Storage** of results of **Long-term usage** could be seen as **Feedback** but does not fully align with a definition from the literature.

3.4 Intermediate summary

For adopting the *FAIR Guiding Principles* for the manufacturing domain, sensor data from that domain has been defined and characterized. By that, the need to adapt some specific requirements and formulate additional ones has been emphasized. Based on a review of existing domain-agnostic approaches for measuring the *FAIRness* of scientific data, 17 criteria have been derived for manufacturing by considering the particular characteristics of sensor data and data management systems of this domain. Moreover, the *FAIR* DLC has been developed to provide a reference frame for implementing the *FAIR Guiding Principles* for manufacturing. These two research artifacts concerning RGQ 1 shown in Figure 3.5 lead to the preliminary response of this RGQ. The two steps **Metadata accumulation** and **Contextualization** of the *FAIR* DLC serve as entry points for the definition of standardized, interoperable data models for *FAIR manufacturing sensor data* in the next chapter.

Preliminary answer to the Research Guiding Question 1

What are the criteria manufacturing sensor data must meet to satisfy the requirements defined by the *FAIR Guiding Principles*?

Definite and universal criteria all manufacturing sensor data must fulfill to be *FAIR* are hard to define. In Section 3.2.2, a set of *FAIRness* criteria for the data itself but also for the data management system has been introduced. To objectively evaluate the applicability and validity of these criteria, data models that reflect these criteria need to be defined first, which can then be used to represent acquired manufacturing sensor data accordingly. The concluding validation of the complete artifacts in Section 6.2 and the *FAIRness* assessment of the acquired measurement data in Section 6.4 will reveal the validity of the defined criteria.

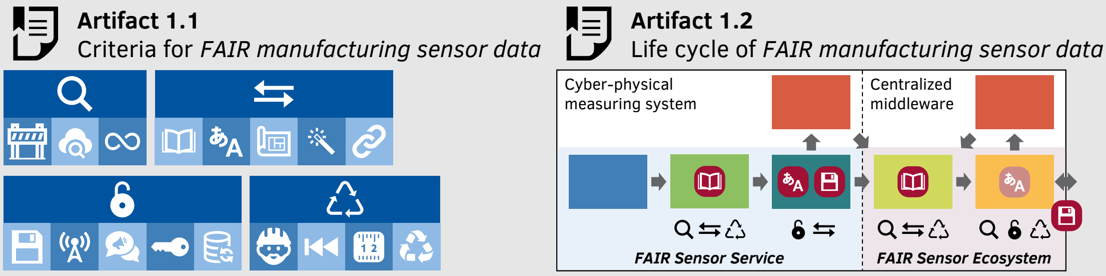


Figure 3.5: 17 defined *FAIRness* criteria and the *FAIR* DLC are the main research artifacts of the research activities to answer the first research guiding question, culminating in a preliminary positive response to that question.

4 Data models for *FAIR manufacturing sensor data*

Interoperable, standardized data models are required for the exchange and joint usage of manufacturing sensor data [Otto17b; Neub23]. The primary concern of the data models is to ensure the interoperability of data from different data sources, i.e., measuring systems, using *Semantic Web* technologies and controlled vocabulary provided by ontologies [Cui20; Eich23; Glei20; Wang18b]. The definition of these models is challenging. On the one hand, the models must achieve high specificity to adequately describe the sensor (meta-)data of a specific measuring system. Still, on the other hand, the models must be sufficiently generic to preserve interoperability with other models [Libe15; Muse22; Dors23]. Thus, a suitable approach is to define highly abstract metamodels leveraging the interoperability of specific data models for individual measuring systems, which are created based on the metamodels [Schl24; Muse22]. All these boundary conditions must be considered to define proper data models ensuring comprehensibility and unambiguous interpretability [Libe15] of *FAIR manufacturing sensor data* as summarized in the second RGQ.

Research Guiding Question 2

How can manufacturing sensor data be described to be findable and interpretable in the applications' context?

As depicted in Figure 4.1, two separate metamodels are developed to tackle the steps **Metadata accumulation** and **Contextualization**, as defined by the *FAIR* DLC. First, available ontologies are investigated for proper domain-relevant controlled terminology to ensure the compliance of the data models with criterion **CR.1 Domain-Specific Terms**. Second, the sensor data model is developed. In preparation for the definition and implementation of SOIL in Chapter 5, a syntactic data structure is defined, which is then extended to a full semantic data model. This later allows the automatic generation of the model from the structure. The semantic data

The author has published parts and preliminary results of the work described in this chapter in the following conference proceedings

Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon, and Robert H. Schmitt. "Domain-Specific Language for Sensors in the Internet of Production". In: *Production at the Leading Edge of Technology*. Ed. by Bernd-Arno Behrens, Alexander Brosius, Wolfgang Hintze, Steffen Ihlenfeldt, and Jens Peter Wulfsberg. Lecture Notes in Production Engineering. Berlin, Heidelberg: Springer, 2021, pp. 448–456. ISBN: 978-3-662-62138-7. DOI: 10.1007/978-3-662-62138-7_45

Matthias Bodenbenner, Jan Pennekamp, Benjamin Montavon, Klaus Wehrle, and Robert H. Schmitt. "FAIR Sensor Ecosystem: Long-Term (Re-)Usability of FAIR Sensor Data through Contextualization". In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. July 2023, pp. 1–8. DOI: 10.1109/INDIN51400.2023.10218149

Matthias Bodenbenner, Dominik Wolfschläger, and Robert H. Schmitt. *Providing FAIR Sensor Data Models Using Semantic Web Technologies and Ontologies*. Apr. 15, 2024. DOI: 10.13140/RG.2.2.27882.73928. Pre-published models are built upon the ontologies reviewed before. Third and last, the contextualization model is developed to link manufacturing sensor data represented with the sensor data model to related entities and assets to address criterion **CI.5 Linked Data**.

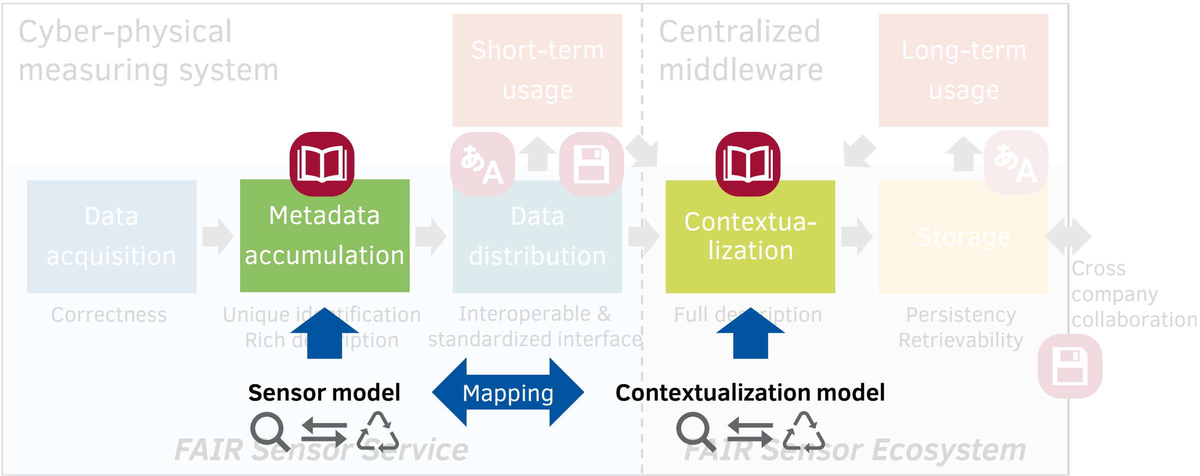


Figure 4.1: The data models developed with respect to RGQ 2 provide the methodological realization of the steps **Metadata accumulation** and **Contextualization** of the *FAIR* DLC defined in Section 3.3.

4.1 Review of available data models for sensor data

The defined models must be applicable in the manufacturing industry, meaning they must be aligned with industrially used data models. On the one hand, this covers standardized and widely applied models such as OPC UA. The data models used in the manufacturing industry have already been reviewed in Section 2.3. On the other hand, it includes data models most recently developed in research targeting industrial applications, such as the *Protocol Agnostic Device Interface* [Mont21]. To comply with the *FAIR Guiding Principles* and the criteria for *FAIR manufacturing sensor data* defined in Section 3.2.2, the data models must use controlled vocabulary that is available via ontologies using *Semantic Web* standards (cf. **CI.1 Controlled Terms**). The terms and ontologies used for modeling should be widely known and applied to ensure interoperability. So, a thorough review of existing ontologies for modeling sensor data is carried out in Section 4.2.2 to identify the terms for the semantic sensor data model.

Data structures and models already used in industry for capturing the characteristics of manufacturing data and its relation to relevant assets or processes have already been investigated in Section 2.3. A small set of the closest related scientific approaches recently published are reviewed here.

Cramer et al. [Cram21] propose a comprehensive object-oriented *Meta-Model for Production Data (MMPD)* which can capture the context in which a measurement is acquired. The authors propose a step-by-step procedure for deriving specific data structures for individual use cases and applications from the MMPD. However, the MMPD does not build on controlled vocabulary and *Semantic Web* technologies. Thus, the relations between the concepts described are not compliant to **CI.1 Controlled Terms**. Moreover, crucial criteria for reusability, such as **CR.4 Reuse Conditions**, are not covered.

Another model for industrial data and so-called Digital Shadow (DS) has been published by Becker et al. [Beck21]. The model is well formalized by qualifying and quantifying the relations between the modeled entities. But as with the FactDAG model (cf. Section 2.1) and the MMPD, the conceptual model for DSs is missing an investigation on a minimal set of measurement-related metadata, which is required in any case to ensure interoperability. In particular, specific metadata and terminology as required by **CR.1 Domain-Specific Terms** and **CR.3 Metrological Information** are missing, and reuse conditions (cf. **CR.4 Reuse Conditions**) are not considered.

Mayer et al. [Maye17] propose the *Open Semantic Framework (OSF)* which combines domain-agnostic ontologies with domain-specific "knowledge packs" to provide semantic models for dedicated use-cases. By that, the OSF is a knowledge management tool with a simplified user interface (UI) leveraging the usability of semantic modeling approaches for engineers. The aim of the OSF is to provide machine-actionable information to improve machine-to-machine communication with a focus on ensuring safety regulations in the industry. The approach focuses on interoperability but neglects the other aspects of the *FAIR Guiding Principles*. Criteria, such as **CA.5 Versioning** in case of changing configurations or situations, **CF.3 Persistency**, **CR.3 Metrological Information**, and **CR.4 Reuse Conditions** are not considered. Lastly, the primary use case for the OSF is engineering complete plants instead of single systems.

To dismantle the monolithic architecture of industrial data infrastructures, Schmitt and Voigtmann [Schm18a] proposed *Sensor Information as a Service*, where each measuring system is realized as an application-independent micro-service. Montavon [Mont21] extended this concept by *Coordinates as a Service* and defined a data model for the protocol-agnostic definition of the data structure and communication interface of a measuring system with a focus on large-scale metrology (LSM) systems. The model has strong concordance to OPC UA. However, the structure misses controlled terminology and is therefore not machine-actionable, as required by **CI.1 Controlled Terms** and **CR.1 Domain-Specific Terms**. In addition, the **CI.5 Linked Data** and **CR.4 Reuse Conditions** are not addressed.

In summary, the most recent related work mainly focuses on domain-agnostic and syntactic interoperability only. Additionally, they often lack explicit reusability measures, so the value of the manufacturing sensor data tends to degree rapidly. Therefore, the sensor and contextualization models must account for that.

4.2 A data structure and a data model for manufacturing sensor data

The sensor model is the base for the unambiguous and interpretable representation of the information a sensor or measuring system acquires and provides the main contribution to criterion **CI.3 Information Model**. Thus, the model must cover all relevant attributes for the data's long-term reusability. Moreover, the model serves as the base for the definition of SOIL in Chapter 5 and the subsequent generation of the interoperable communication interface. Therefore, the model must reflect all characteristics required for interaction with the measuring system. For maximizing the abstraction power of SOIL, first, a syntactic data structure is defined as the groundline for SOIL. Second, the semantic data model is derived based on existing ontologies and fully aligned with the requirements for *FAIR manufacturing sensor data* as defined in Section 3.2.2.

4.2.1 Syntactic data structure for sensors

The syntactic sensor data structure builds upon the *Protocol Agnostic Device Interface* proposed by Montavon (et al.) [Mont19; Mont21] and facilitates a functional viewpoint for enabling interoperability. Standardization and reusability can be amplified by modeling the sensor's capabilities as a set of resources with defined actions encapsulated as service embracing a functional rather than a physical point of view. The description of different measuring systems must rely on a common meta structure to ensure their interoperability and consistency (cf. criterion **CI.4 Consistency**). This standardized, generalized meta structure is then the base for specific sensor data structures containing all information for one specific device as shown in Figure 4.2. The meta structure (an M2 metamodel) originates from a functional perspective for sensor services and consists of four basic building *Elements*: *Components*, *Functions*, *Parameters*, and *Measurements*. A device-specific structure (an M1 user model) defined using the meta structure forms the sensor service's static description and has a tree-like shape due to a recursive composition scheme. Figure 4.3 illustrates the object-oriented data structure. The definition and attributes of the four *Elements* are introduced in the following.

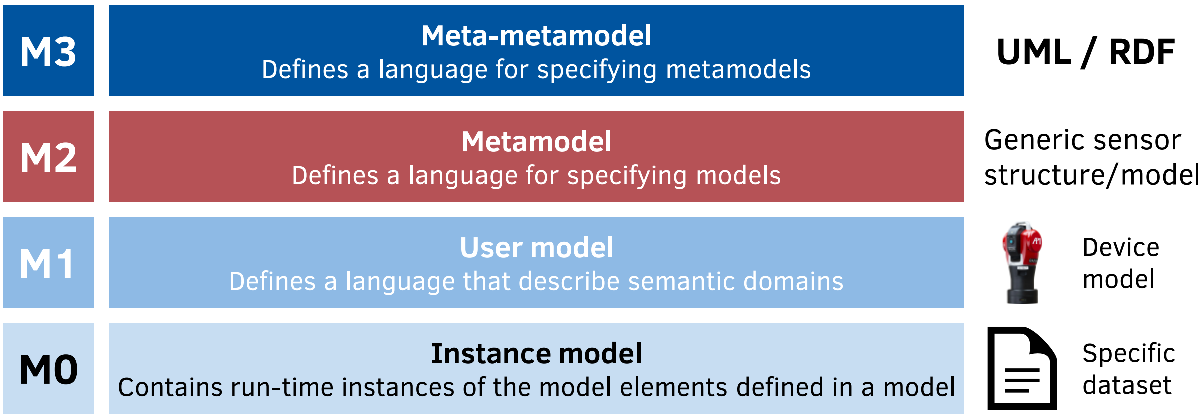


Figure 4.2: Localisation of the developed data structure and model in the model hierarchy (cf. Figure 2.7) highlighted in red color.

Element

Each *Element* has a unique identifier, a human-readable name, and a short description of its meaning to lay the base for the fulfillment of criterion **CR.2 Provenance**. Identifiers are locally unique and become globally unique by prepending the parents' identifiers successively, as introduced by Montavon [Mont21]. In addition to these three mandatory fields, the field "ControlledTerm" is introduced, extending the optional "Ontology" field of the model of Montavon [Mont21] addressing criteria **CI.1 Controlled Terms** and **CR.1 Domain-Specific Terms**. This controlled term allows the structure to be mapped to a full semantic sensor model based on the semantic sensor data metamodel described in Section 4.2.3.

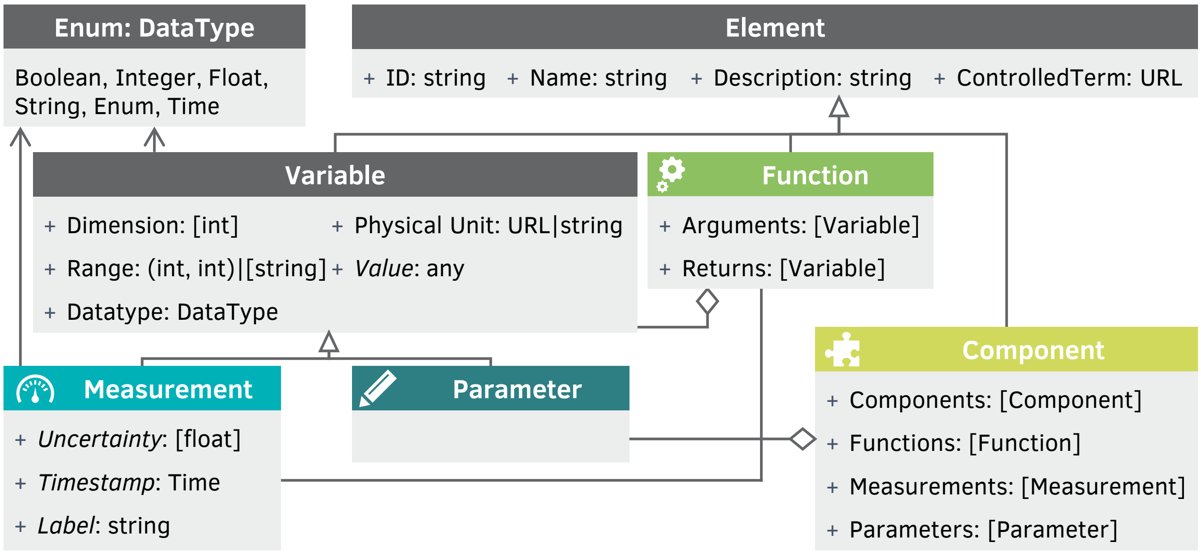


Figure 4.3: UML diagram of the objected-oriented, syntactic data structure for sensors and measuring systems. The colors and icons indicate the four types of *Elements*. All properties in italics are relevant at runtime only and are not specified in the static model definition. By selecting a controlled term for each *Element*, which refers to some object or concept restricted by the model, a device-specific sensor data structure can automatically be translated into a semantic sensor data model based on the semantic sensor metamodel depicted in Figure 4.6.

Measurement

The sensor structure must represent and include all values produced and measured by the measuring system. These *Measurements* are formally defined in Definition 4.1.[1]

Definition 4.1: Measurement

Measurements are observed, captured, and digitized expressions of certain characteristics of the real world. Measurements represent all characteristics physically sensed by the sensors of the measuring system, i.e., all data generated by and originating from the measuring system. The value of a measurement can be nominal or quantifiable.

These characteristics include the intended purpose, like temperature or distances, and attributes, like battery level or signal strength. In contrast to the VIM [BIPM12], the definition also allows *Measurements* to be nominal properties. In the context of metrology, measurement results, i.e., values of a *Measurement*, always require the measurement uncertainty to be quantified [BIPM12] (cf. criterion **CR.3 Metrological Information**). Data integrity and traceability are promoted by specifying a label and a timestamp [Eich23; Wang18b]. A manually specified value of a *Measurement* has no meaning, as it does not capture an expression of real-world characteristics. Thus, the inherited value cannot be altered externally compared to *Parameters*. The *Measurement* shares all other relevant attributes with *Parameters* and, therefore, are encapsulated in the common base class *Variable*.

[1] *Measurements* correspond to the *Variables* of the model proposed by Montavon [Mont21]. The renaming facilitates clear semantic distinction between *Parameters* and *Measurements*. To emphasize the similarities and adhere to best practice modeling techniques, *Variables* within this thesis refer to both *Measurements* and *Parameters*.

Parameter

Parameters cover all data not directly measured by the device but required for operation or user interaction, as specified by Definition 4.2. Examples are generic settings, addresses for internal communication, or metainformation. *Measurements* and *Parameters* share many attributes [Mont21]. Therefore, *Variables* are introduced, including all attributes required to describe *Measurements* and *Parameters*.

Definition 4.2: Parameter

Parameters express properties of the measuring system. The value of a parameter can be nominal or quantifiable.

Variable

The *Variable* is realized as an abstract base class for *Measurements* and *Parameters* that defines all attributes that these two *Elements* have in common and are required to adequately characterize all values of or produced by the measuring system. Following Evertz and Epple [Ever15], the physical meaning of the values is persisted by defining the datatype, the allowed range, the mathematical dimension, and the physical unit (cf. Figure 4.3). Each *Variable* is of one of the following primitive datatypes: Boolean, Integer, Double, String, Enum, or Time. The physical unit is specified using controlled terminology to ensure interoperability, i.e., selecting the Uniform Resource Identifier (URI) of a term defined by an ontology, such as QUDT (see Section 4.2.2). Dorst et al. [Dors23] emphasize the importance of specifying the physical quantity. The field `controlledTerm` is intended to hold the URI of the controlled term defining a physical quantity. However, from the functional modeling perspective, the *Variable* might also describe a nominal value. Thus, the field might also contain a URI to another term than a physical quantity.

Definition 4.3: Variable

A variable characterizes any nominal or quantifiable value through the metainformation describing this value.

Function

Functions trigger complex tasks or change the inner state of the sensor. A *Function* generally accepts an ordered set of argument *Variables* and returns an ordered set of return *Variables*. The logic of the tasks executed is not further specified in the model. This highly abstracted and simplified representation of an executable procedure is required to fulfill the criteria **CA.1 Standard Protocols** and **CA.3 Protocol-Agnostic Modeling**.

Definition 4.4: Function

A function of a measuring system executes a non-atomic procedure that might consist of multiple tasks. A function might change the inner state of the measuring system, modify *Parameters*, or capture *Measurements*.

Component

Components are structural *Elements* of the sensor data structure. Each *Component* contains an arbitrary number of children *Elements*, so the overall model has a tree-like shape. Cyclic *Component* relationships are forbidden to avoid infinite composition cycles. Semantically, the *Component* of a measuring system is defined in Definition 4.5. *Components* group logically or functionally coherent characteristics, as *Parameters* and *Measurements*, or subsystems represented by *Components* of the measuring systems to obtain a browsable and understandable structure.

Definition 4.5: Component

A component is a distinguishable part of a measuring system that has properties, offers specific *Functions*, and might contain sensors. A component can be constituted of other components, itself.

Overall Structure

In the following, the device-specific M2 data structure (cf. Figure 4.2) of an OptiTrack™ motion capture system (MCS)[2] is illustrated to explain the usage of this device-agnostic M1 meta structure (see Figure 4.3). An excerpt of the complete data structure is given in Figure 4.4, showing the core *Elements* of the model. The OptiTrack™ MCS can detect reflective spherical markers attached to physical assets (for more details on the OptiTrack™ MCS see Section 6.1). Multiple markers attached to the same asset are grouped to rigid bodies (*Component* `RigidBody`) so that the MCS can localize the ridig body, i.e., the physical asset, if it is reliably tracked (*Parameter* `TrackingState`). The systems use multiple distributed cameras (*Component* `Camera`) with overlapping fields of view for that. For reconstructing the three-dimensional position and orientation of the ridig body, the exact knowledge of the position (*Measurement* `Position`) and orientation of the individual cameras are crucial. Moreover, the camera's name (*Parameter* `Name`) is required to allow users to interact with individual cameras precisely. The complete structure with 30 *Elements* is available online [Bode24e].

This model can be mapped to other models such as OPC UA and a RESTful HTTP interface as already proven by Montavon [Mont21] and, therefore, adheres to the criteria **CA.1 Standard Protocols**, **CA.2 Multi-Protocol**, and **CA.3 Protocol-Agnostic Modeling**. The relatively simple structure and minimal set of *Elements* suffice to represent the functionality of sensor and measuring systems [Mont21]. Due to the very few concepts that need to be understood, it is easy to learn and use. Providing a short introduction and explanation complemented with real examples, people who are no data stewards or information scientists can define data structures for measuring systems using this model. Nevertheless, this structure only fulfills Definition 2.4 of *syntactic interoperability*. Moreover, the keys of all attributes are no controlled terms, which do not comply with the criteria **CI.1 Controlled Terms** and **CR.1 Domain-Specific Terms** defined in Section 3.2.2. Hence, based on a review in the next section of existing ontologies providing controlled terms, an extension of the data structure to a semantically interoperable data model (see Definition 2.5) is defined in Section 4.2.3.

[2] A detailed introduction to the system is given in Section 6.1.

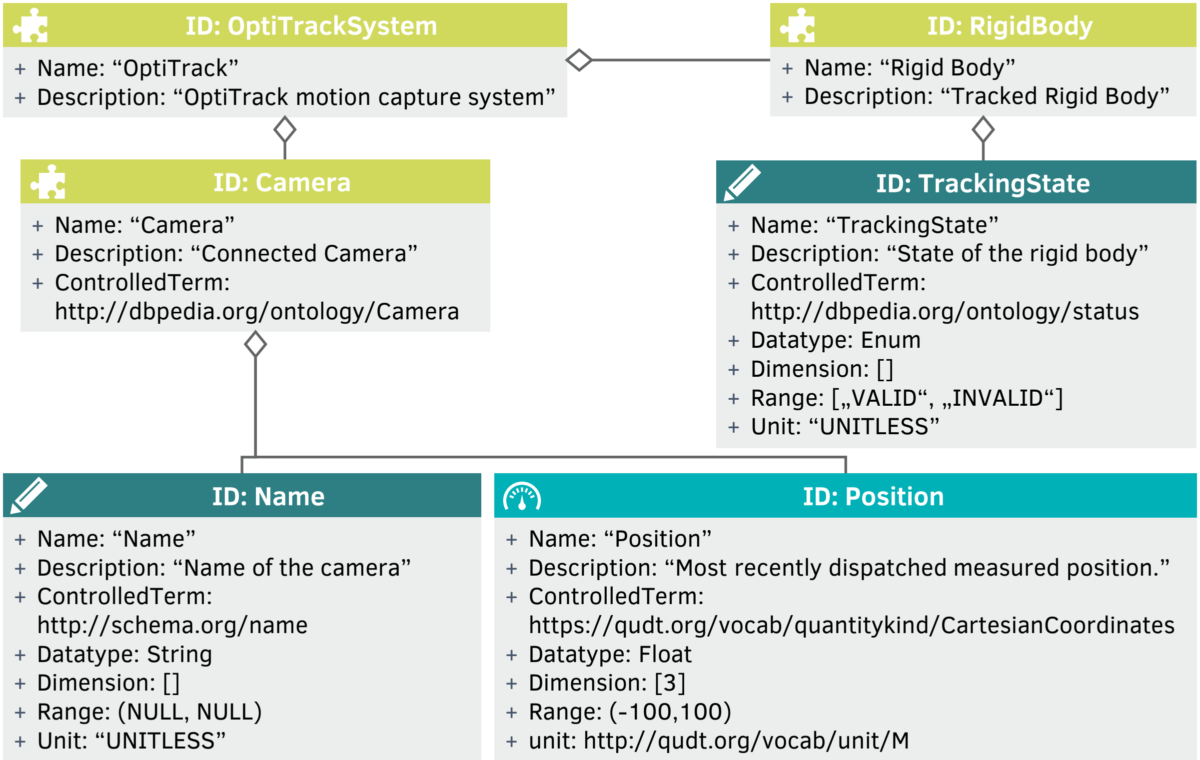


Figure 4.4: Visualization of six *Elements* of the data structure of an OptiTrack™ MCS. The overall structure contains 30 *Elements* and cannot be fully shown. The serialization of the complete structure in JSON is published online [Bode24e].

4.2.2 Discussion of existing ontologies and controlled vocabularies

The reuse of widely applied and acknowledged controlled vocabularies, e.g., provided by ontologies, is required to ensure interoperability [Eich23]. Therefore, existing ontologies have been reviewed to evaluate their reuse potential for describing manufacturing sensor data. The selection of appropriate ontologies and terms within the ontologies can hardly be based on measurable and fully objective criteria and is therefore carried out based on the following four qualitative criteria:

Specificity According to the Oxford English Dictionary, *specificity* is “[t]he quality or fact of being specific in operation or effect.”[Oxfo23c], whereas *specific* is defined as: “Of qualities, properties, effects, etc.: Specially or peculiarly pertaining to a certain thing or class of things and constituting one of the characteristic features of this.”[Oxfo23b]. In other words, a controlled term is *specific* if it describes the thing of interest, excluding non-similar things, without additions restricting the correct usage. A term is a *little specific* if a very broad range of concepts are described and *highly specific* if the range of concepts described is strongly restricted.

Documentation To be interpretable by humans, the controlled terms must have a human-readable definition and description. Ideally, the definition of the term is completed with intuitive usage examples. Consequently, ontologies that only define the relations between the terms but do not provide a textual definition are excluded, as the Specificity can hardly be evaluated.

Persistency The ontology should be persistent with regard to the technical identification of the terms included, e.g., using PURLs, but also concerning the permanence of the defined meaning of the terms. So, the definitions should only be slightly adapted over time, if at all. Renaming or removing the controlled terms should be avoided. Otherwise, data models built on these terms are no longer machine-actionable.

Dissemination The amount of reuse by other ontologies is an implicit quality characteristic. If two ontologies are equally applicable, the ontology being reused more often is adopted. The amount of reuse has been taken from Linked Open Vocabularies [Onto24][3] and is summarized in Table 4.1.

For building the semantic models domain-agnostic, general-purpose ontologies such as RDF [Cyga14], the XML Schema Definition Language (XSD) [Fall04], the Shapes Constraint Language (SHACL) [Knub17], and the Web Ontology Language (OWL) [W3C 12], are reused for generic attributes, such as identifiers, descriptions, and license information and the structure of the model itself. In the following, the specialized ontologies building the core structure and most relevant attributes of the sensor and contextualization model are briefly introduced. It is described how these are used to describe manufacturing sensor data in a machine-actionable and *FAIR* way. Those reviewed ontologies that have not been used are summarized in Appendix B.

Table 4.1: Number of reuses of the inspected ontologies according to Linked Open Vocabularies [Onto24]. The ontologies used for the data models in this thesis are bold.

Name	Acronym	Common prefix	Reuses
Sensor, Observation, Sample, and Actuator	SOSA	sosa	14
Semantic Sensor Network Ontology	SSN	ssn	9
DCMI Metadata Terms	DC-Terms	dcterms	730
Quantities, Units, Dimensions and Types	QUDT	qudt	18
Metadata4Ing	M4I	m4i	1
Provenance Ontology	PROV-O	prov	80
Records in Contexts-Ontology	RiC-O	rico	0
Schema.org	-	schema	99
IoT-Lite Ontology	-	iot-lite	2
Ontology of units of Measure	OM	oum	0
Thing Description Ontology	TD	td	0
Digital System of Units	D-SI	si	not listed

[3]Numbers extracted from website on 2024-01-20

Sensor, Observation, Sample, and Actuator and Semantic Sensor Network Ontology

Sensor, Observation, Sample, and Actuator (SOSA) forms the core module of an ontology pair completed by Semantic Sensor Network Ontology (SSN) [Comp12; Hall18]. SOSA defines the relations between measurements, the measuring system, the observed property, and the measurement procedure in detail. That facilitates the interoperability of measuring systems, sensors, and measurements. To reduce the complexity and foster reuse, the developers of SOSA avoided using subclasses and subproperties. On the one hand, the simple structure makes the ontology easier to understand. On the other hand, the expressiveness is limited. Complete axiomatization is added by SSN to overcome the latter. By defining a common vocabulary and structure for modeling measurement data, the combination of SOSA and SSN enables the combination, reuse, and exchange of measurement data from multiple sources. In the *FAIR manufacturing sensor data* model, the classes and properties defined in SOSA and SSN are used to capture the four elements (i.e., *Measurement*, *Parameter*, *Function*, and *Component*) of the data structure and their relationships machine-actionably. Thus, the data represented using this model are interoperable with foreign measurement data that also rely on these two ontologies. Next, the attributes of the elements must be semantically defined. For the generic information of all elements DCMI Metadata Terms is used.

DCMI Metadata Terms

First introduced in 2000 as Dublin Core, DCMI Metadata Terms (DC-Terms) emerged as the most reused vocabulary according to Linked Open Vocabulary (LOV) [Onto24], setting the base for defining essential meta attributes of any resource [DCMI20]. In total, 87% of the vocabularies indexed by LOV reuse DC-Terms. The core of DC-Terms consists of fifteen properties defining attributes such as *description*, *identifiers*, or *title*. These properties are supplemented by properties and classes specifying them further by restricting the range, domain, and other attributes. Because DC-Terms can be considered the de facto standard for modeling essential attributes of resources, the sensor and contextualization models introduced in this dissertation reuse definitions from DC-Terms for capturing basic attributes, such as the *description*, *identifier*, and *title* of data points. For the representation of the metrological information and attributes, the specialized QUDT ontology is used.

Quantities, Units, Dimensions and Data Types Ontologies

Quantities, Units, Dimensions and Data Types Ontologies (QUDT) [QUDT23] provides a set of vocabularies to describe physical quantities, units, dimensions, and datatypes. QUDT defines 950 physical quantities and 1711 physical units, which allows semantically describing almost every common measurement. The physical relations between quantities and units, such as the velocity dependency on distance and time, are fully described. The units and quantities are detailed, and additional informative resources are provided. Mappings to other unit encoding schemes, such as UNECE [UNEC21], are included, ensuring maximum interoperability of QUDT. Beyond classes for these, QUDT defines a large set of relations to associate a physical quantity to applicable units. Therefore, QUDT is employed for describing all attributes of variables. But, QUDT misses concepts for modeling the uncertainty of a measurement result. This gap is closed by combining QUDT with the Metadata4Ing ontology.

Metadata4Ing

A special interest group has developed the Metadata4Ing (M4I) ontology [Igle24] within the consortium *National Research Data Infrastructure for the Engineering Sciences (NFDI4Ing)*. The ontology defines how scientific experiments create datasets and links these datasets to the responsible authors, the applied methods, and the tools used. It details the attributes associated with a measurement value, its physical quantity, and its unit for which QUDT is used. Moreover, M4I defines the measurement uncertainty of a measured value using D-SI (cf. Appendix B). This modeling of measurements has been reused by the author but slightly adapted for developing the sensor metamodel described in Section 4.2. M4I provides a sophisticated controlled vocabulary for representing measurement uncertainty and linking this information to the actual measurement value based on the D-SI, which is used for that purpose in the models developed in this thesis (cf. Figure 4.6, `soil:MeasurementResult`).

Records in Contexts-Ontology

An expert group of the International Council on Archives (ICA) has developed the Records in Contexts Conceptual Model (RiC-CM) [Clau21] to describe (archived) *records*. The model defines relations between different records and their contexts, i.e., interactions and people dealing with records. Because the metainformation of measurement data, such as records, should also be long-term reusable and findable, the model is well suitable for the contextualization model. The model defines dedicated relations for composition, which can be used to model hierarchical structures between things. This feature has been exploited to construct the relations between measuring systems and related assets (e.g., machines, processes, tools). The expert group defined the Records in Contexts-Ontology (RiC-O) as a semantic representation of the model, making it suitable for the definition of *FAIR* data models.

Provenance Ontology

The Provenance Ontology (PROV-O) [Lebo13] is the OWL-compliant representation of the Provenance Data Model (PROV-DM) [More13] describing the provenance and origin of things in a generic, domain-independent manner [Butt20]. The description model defines how *entities*, e.g., measured data or written documents, were generated by an *activity* attributed to an *agent*. Moreover, the model specifies terms to handle revisions of *entities*, which enables describing and linking metadata sets. PROV-O is used in the contextualization model described in Section 4.3 to address **CA.5 Versioning** and ensures that revisions of metadata sets are linked, browsable, and must not be deleted.

Schema.org Vocabulary

In contrast to most other ontologies, Schema.org [Sche23] is a community project. The community approach reduces the effort to adapt and extend the vocabulary. Based on the definition of a generic *Thing*, the scope of the 805 defined types with over one thousand properties can hardly be described concisely. Examples of more specific *Things* according to Schema.org are *Events*, *Persons*, *Organizations*, or *CreativeWork*. The properties defined by Schema.org include *name*, *description*, and *identifier*. Due to their generality, the terms defined by Schema.org are widely reused to define semantic data models. Schema.org is used to model generic attributes and time-related information of the contextualization model.

4.2.3 Semantic data model for measuring system interfaces

Achieving *FAIR manufacturing sensor data* requires the definition of a sensor model fully relying on controlled terminology and *Semantic Web* standards, cf. criterion **CI.1 Controlled Terms**. Thus, a model is defined using RDF and suitable ontologies, reviewed in Section 4.2.2. However, one issue is the lack of specific terms precisely describing the meaning of the concepts and data points of the sensor data model [Grön23]. To overcome this issue and also not introduce new semantic definitions, restrictions on the properties and relations between the different nodes of a sensor data model have been defined as metadata profiles (MPs). MPs, often also called *Application Profiles* [Coyl17], can achieve high specificity while maintaining interoperability, modularity, and reusability [Berm06]. With MPs, specific restrictions are defined based on a generic controlled term broadly describing the thing in question. For example, instead of defining a new specific controlled term, "temperature sensor", one defines a generic "sensor", which is restricted to measure a value having the physical quantity "temperature". This MP can entirely rely on existing generic controlled vocabularies, as illustrated in Figure 4.5.

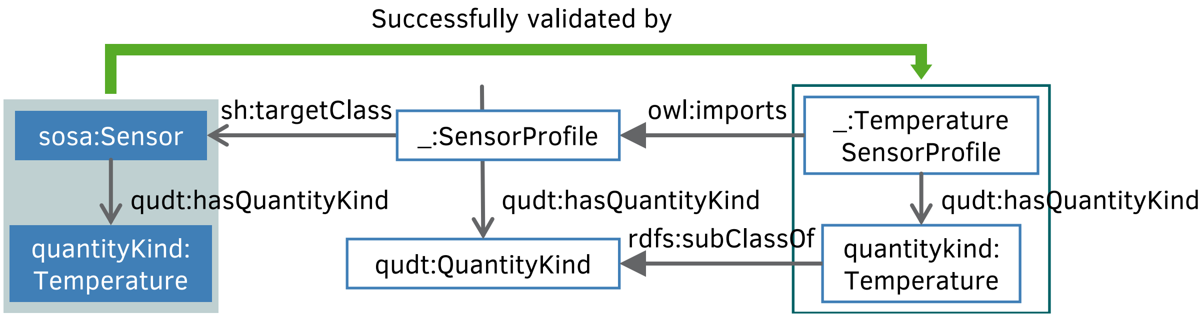


Figure 4.5: Illustration of a metadata profile (blue bordered boxes) based on available domain-agnostic vocabularies and the metadata set (blue filled boxes), which is validated by this MP. The metadata set does not need additional specialized terminology for "temperature sensor", as the metadata profile defines restrictions accordingly.

SHACL has been used to define MPs for all nodes of a sensor model. For the type of all nodes available, controlled terms have been used only. For the definition of the metadata profiles, the namespace `soil` has been introduced, for which a PURL has been claimed, namely `http://purl.org/soil/` to ensure **CF.3 Persistency** is fulfilled.

The core of the MP, i.e., the semantically interoperable sensor metamodel, is shown in Figure 4.6. As the complexity of the sensor model is much higher compared to the sensor structure, a mapping between the semantic model and the syntactic structure, as defined in Section 4.2.1, must be possible to allow the automatic derivation of the sensor model from the sensor structure using the IDL to be developed in the next chapter. Thus, the model needs to reflect the *Elements* of the sensor structure. To ensure interoperability, all specific metadata and data graphs of sensors need to be compliant with this MP. Even more specific MPs are automatically generated for a specific sensor based on a specific sensor structure based on the meta structure defined in Section 4.2.1. All nodes of a sensor model are identified and linked to each other using SSN and SOSA. The properties of the nodes are majorly modeled using QUDT and DC-Terms. In the following, the parts or sub-profiles of the generic sensor MP are explained.

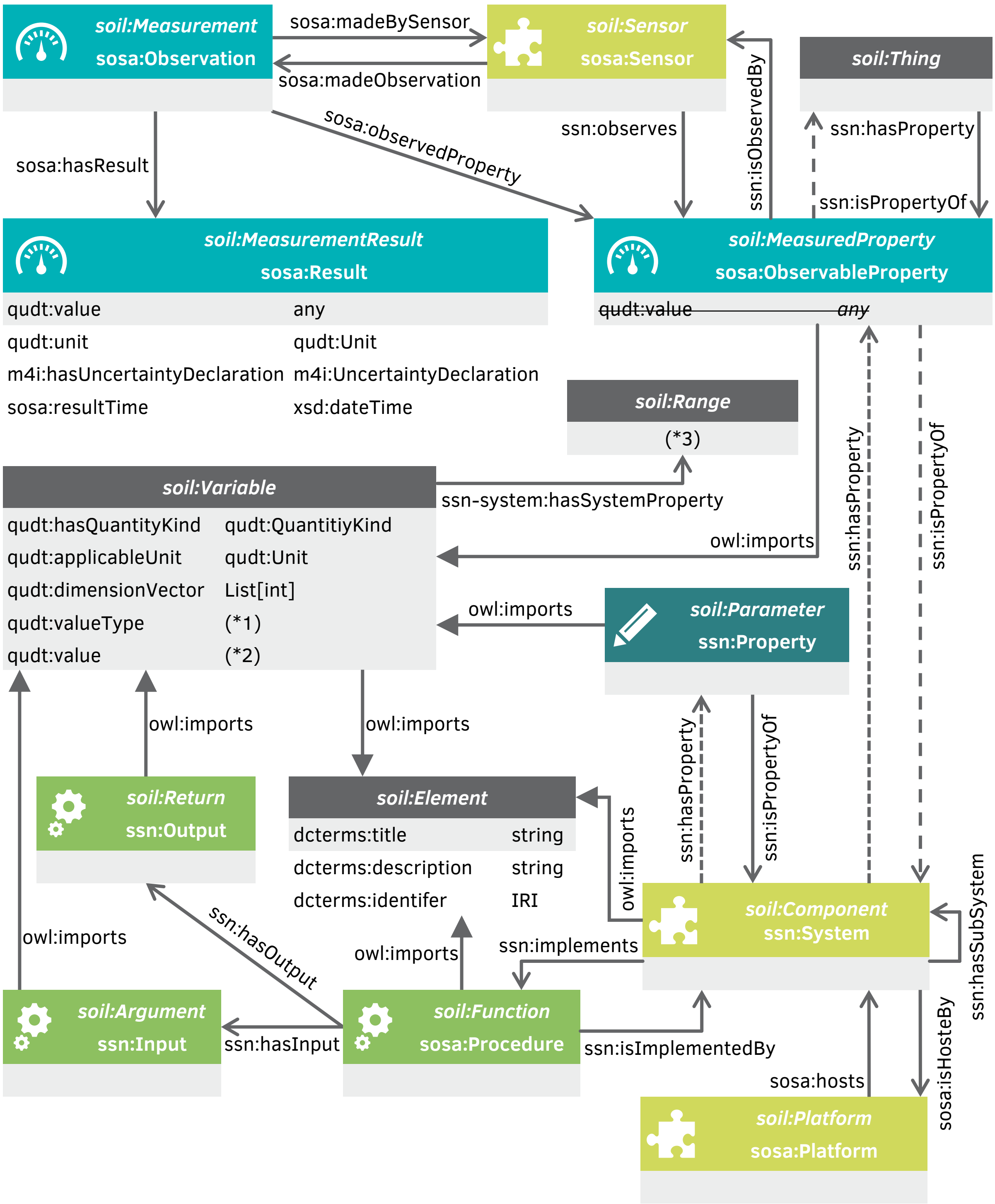


Figure 4.6: The semantically interoperable sensor model for the provision of *FAIR manufacturing sensor data*. The mapping to the syntactic data structure, as depicted in Figure 4.3, is indicated using the same colors and icons. The identifiers of the SHACL shapes defining the restrictions according to the data model are written in italics, while the target classes are in normal font. Equally dashed arrows state exclusive alternatives, e.g., a `soil:Measurable` is either a property of a `soil:Component` or of a `soil:Thing`. (i) The value type is one of the six base types, as described for *Variables* (ii) The value is restricted to fit the value type, either as a scalar or a list. (iii) The range is either given by an (infinite) interval with minimal and maximum inclusive numbers or as a set of allowed strings.

Element

To adequately cover the sensor structure in the model and retain maximum modularity and reusability, an MP for *Elements* is introduced. A `soil:Element` has two mandatory properties: `dcterms:title` and `dcterms:description`. The identifier of an *Element* is given by the RDF node name. The profile does not target a dedicated class to retain interoperability.

Variable

For maximum interoperability of *Measurements* and *Parameters* of *Components*, as well as arguments and returns of *Functions* of the semantic sensor model, a common base profile `soil:Variable` is introduced, which semantically defines the restrictions for the properties. To cover all attributes of a *Variable* according to the description in Section 4.2.1, at least the following properties are expected for a *Variable*:

1. A *Variable* may have one or multiple `qudt:value` which type is one of `xsd:boolean`, `xsd:integer`, `xsd:float`, `xsd:dateTime`, or `xsd:string`. Whereof the value of a *Variable* of datatype enum also has the type `xsd:string`.
2. The range of permissible values is given using `ssn-system:hasSystemProperty`, which points to a dedicate node `soil:Range` as defined by Definition 4.6.
3. A *Variable* requires at exactly one `qudt:applicableUnit`, which defines the physical unit in case the value is a quantity, i.e., the datatype of the *Variable* is `xsd:integer` or `xsd:float`. The unit must be given using QUDT. Thus, it must be a `qudt:Unit`.
4. Same as with the unit, a *Variable* can have at most one `qudt:hasQuantityKind` property pointing to a `qudt:QuantityKind` specifying the physical quantity of the *Variable*.
5. The specification of the *Variable's* `qudt:dimensionVector` by a list of integer values is mandatory.

Definition 4.6: soil:Range

The range restricts permissible values either by specifying an interval based on optional, inclusive lower and upper limits or a set of explicitly defined values using an enumeration.

The profiles of *Measurements*, *Parameters*, arguments and returns import the *Variable* profile to restrict their properties and introduce new restrictions to link the profiles to the other parts of the sensor model.

Parameter

Parameters are represented using `ssn:Property`. To reflect its definition in the sensor model, a `ssn:Property` is restricted to be a property of a *Component* of the sensor model. This relation is modeled with the property `ssn:isPropertyOf`.

Measurement

For a *Measurement* of the data structure, the metainformation and measurement value must be handled separately to accurately model it following the SOSA ontology. The metainformation of the *Measurement* is modeled using `sosa:ObservableProperty`. A `sosa:ObservableProperty` inherits all restrictions of a *Variable* but does not allow the property `qudt:value` to be present. Thus, it only covers the static metainformation of the *Measurement* and not the dynamic data. The MP `soil:Measurable` defining these attributes and restrictions is depicted in Listing 4.1.

The dynamic measurement data itself is represented using `sosa:Result`, where the result carries the actual measured value, with quantification of its uncertainty, the physical unit, and the time of acquisition of the value, as shown in Figure 4.6. While the VIM [BIPM12] defines terminology for unambiguously describing measurement uncertainty, devices, and methods in detail, it does not comply with *Semantic Web* standards [Eich22]. Thus, the VIM can not be used for modeling the measurement uncertainty. Nevertheless, the VIM has influenced the development of ontologies and semantic vocabulary for the definition of the measurement uncertainty, such as Metadata4Ing [Igle24] and Digital System of Units [Hutz20]. Thus, these two ontologies are used to describe the measurement uncertainty compliant with the criteria for *FAIR manufacturing sensor data*. Following SOSA, the link between the measured data and the metainformation, i.e., the observed property and the sensor that acquired the result, is established using `sosa:Observation`.

Function

Functions are realized using `sosa:Procedure`, for which the property `ssn:hasInput` is restricted to relate to a node of class `ssn:Input` used for all arguments of the *Function*. Analogously, `ssn:hasOutput` and `ssn:Output` are used for the returns of *Functions*. The MPs `soil:Argument` and `soil:Return` which target these inputs and outputs import the MP `soil:Variable` defining restrictions for *Variables*.

Component

Three core classes of the SOSA and SSN ontologies are used to model the complete characteristics of a *Component*. First, a *Component* with a *Parameter* or *Function* has characteristics of an `ssn:System` which has `ssn:Property` or implements a `sosa:Procedure`, respectively. Second, if a *Component* contains another *Component*, it is also a `sosa:Platfrom` which hosts another `ssn:System`. Third, if a *Component* has a *Measurement*, it must fulfill all restrictions of a `sosa:Sensor`, which observes a `sosa:Observation`. If a controlled term is specified for a *Component*, the term is used as the type of the metadata node.

More detailed examples and a proof-of-concept of the semantic data model are given in Section 4.4. The relation between the syntactic data structure and the semantic data model is also illustrated there based on a specific example.

```
1  @prefix sh:      <http://www.w3.org/ns/shacl#> .
2  @prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
3  @prefix dcterms: <http://purl.org/dc/terms/> .
4  @prefix owl:     <http://www.w3.org/2002/07/owl#> .
5  @prefix ssn:     <http://www.w3.org/ns/ssn/> .
6  @prefix sosa:    <http://www.w3.org/ns/sosa/> .
7  @prefix qudt:    <http://qudt.org/schema/qudt/> .
8
9  @prefix soil:     <https://purl.org/soil/> .
10
11 soil:Measurable
12     a                   sh:NodeShape ;
13     dcterms:title       "Measurable Property" ;
14     sh:targetClass      sosa:ObservableProperty ;
15     dcterms:description "A property observed by a sensor, i.e. the sensor
    measures values which quantify this property. Hold the static
    metainformation of a SOIL measurement." ;
16     dcterms:creator     <https://orcid.org/0000-0002-9413-1874> ;
17     dcterms:license     <https://spdx.org/licenses/CC0-1.0.html> ;
18     dcterms:created     "2023-12-04"^^xsd:date ;
19
20     owl:imports         soil:Variable ;
21     sh:node             soil:Variable ;
22
23     sh:property [ sh:path     qudt:value ;
24                   sh:maxCount 0 ; # a static default value of a measured
    property does not have any meaning, so static values are forbidden
25                 ] ;
26
27     sh:property [ sh:path sosa:isObservedBy ;
28                   sh:name "is observed by" ;
29                   sh:description  "Component being the sensor observing this
    property." ;
30                   sh:minCount 1;
31                   sh:maxCount 1;
32                   sh:class sosa:Sensor ;
33                 ] ;
34
35     sh:property [ sh:name        "is property of"@en ;
36                   sh:description "Relation between the measured property and
    the entity it belongs to."@en ;
37                   sh:path        ssn:isPropertyOf ;
38                   sh:minCount    1 ;
39                   sh:maxCount    1 ;
40                   sh:node soil:Thing ;
41                 ] ;
42 .
```

Listing 4.1: Metadata profile “Measurable” of a *Measurement* of the semantic sensor data model restricting the expected attributes of the observed property of some assets. The definition of the MP can be roughly divided into four parts. First, the used prefixes, i.e., namespaces, are defined (lines 1-9). After that, the definition of the MP starts with the metadata of the MP itself, such as the creator and the usage license (lines 13-18). Third, the MP `soil:Measurable` extends the `soil:Variable` and applies all restrictions of this MP and restricts one property even further (lines 20-25). Last, the MP `soil:Measurable` defines two restrictions on the metadata of the observed property. Namely, it enforces the existence of exactly one link to the sensor observing the property and another link to the asset of which it is a property (lines 27-41).

4.3 Contextualization model

Based on the device-specific metainformation represented using the data model proposed in Section 4.2.3, the relation of measurement and sensor data to relevant assets[4] must be captured to address **CI.5 Linked Data**, **CR.2 Provenance**, and **CF.1 No Limitations**. This contextualization of the manufacturing sensor data is crucial to find, interpret, and decide on the reuse of existing data in the scope of a specific use-case or application [Gao20]. The definition of a widely applicable contextualization model is particularly complex due to the high application heterogeneity (see **C3 Unknown (heterogeneous) applications**).

Because of that and criteria **CF.1 No Limitations**, the author proposes a simple but generic contextualization model shown in Figure 4.7, which defines relations between measurement data and relevant assets simplified to a generic composition scheme, where each relation is associated with an interval of temporal validity to ensure **CF.3 Persistency**. Each asset of the physical world is modeled as a component, which can have arbitrarily many subcomponents but only one parent component that matches the structure of the sensor data model. The resulting hierarchical tree-like structure might not fully express the usually complex connections of assets in production but gives a basic structure that allows to flexibly relate measurement data to superordinate assets. The assumption that the relationships between the assets change frequently over time is reflected in the model by storing a validity interval for all relations defined by timestamps of each relation's beginning and end.

The configuration or parametrization of a single device, i.e., component, might change, whereas the component itself does not change. This information must be decoupled in the data model. To store the metadata, the entity `ComponentInformation` is introduced, which is linked to the component it describes and to its preceding and following versions of the metadata record by using PROV-O [Lebo13] fulfilling **CA.5 Versioning**. The structure of the metadata itself is not further restricted except for being serializable to JSON (and JavaScript Object Notation for Linked Data (JSON-LD)). Thus, users can store theoretically arbitrary (semi) structured data compliant to **CF.1 No Limitations**.

Ingested measurement data is always (indirectly) linked to at least two components to fulfill guideline **I3**. First, to the `ComponentInformation` (and vice versa) of the component of the measuring system. Thus, the metadata includes a reference to the data they describe so that guideline **F3** is satisfied. Second, to the component representing the asset of which a property is measured. Both together fulfill **CI.5 Linked Data**. Considering that the asset of interest is not necessarily the parent component of the measuring system (e.g., with large-scale metrology devices), measurements are explicitly linked via a dedicated relation to the component of interest. All data points are identified by using an identifier according to UUID4. Because the identifier will never change, **CF.3 Persistency** is fulfilled. Lastly, to fulfill **CI.4 Consistency**, all component information and all measurements must conform to a specific type definition to which the data is linked. In general, all relations are bidirectional for easy browsability and discoverability. By adding a mandatory license field to each entity of the model, criteria **CR.4 Reuse Conditions** is addressed.

[4] In the following, the term "asset" refers to both a full physical object and physically inseparable, but logically distinguishable, sub-parts of this object.

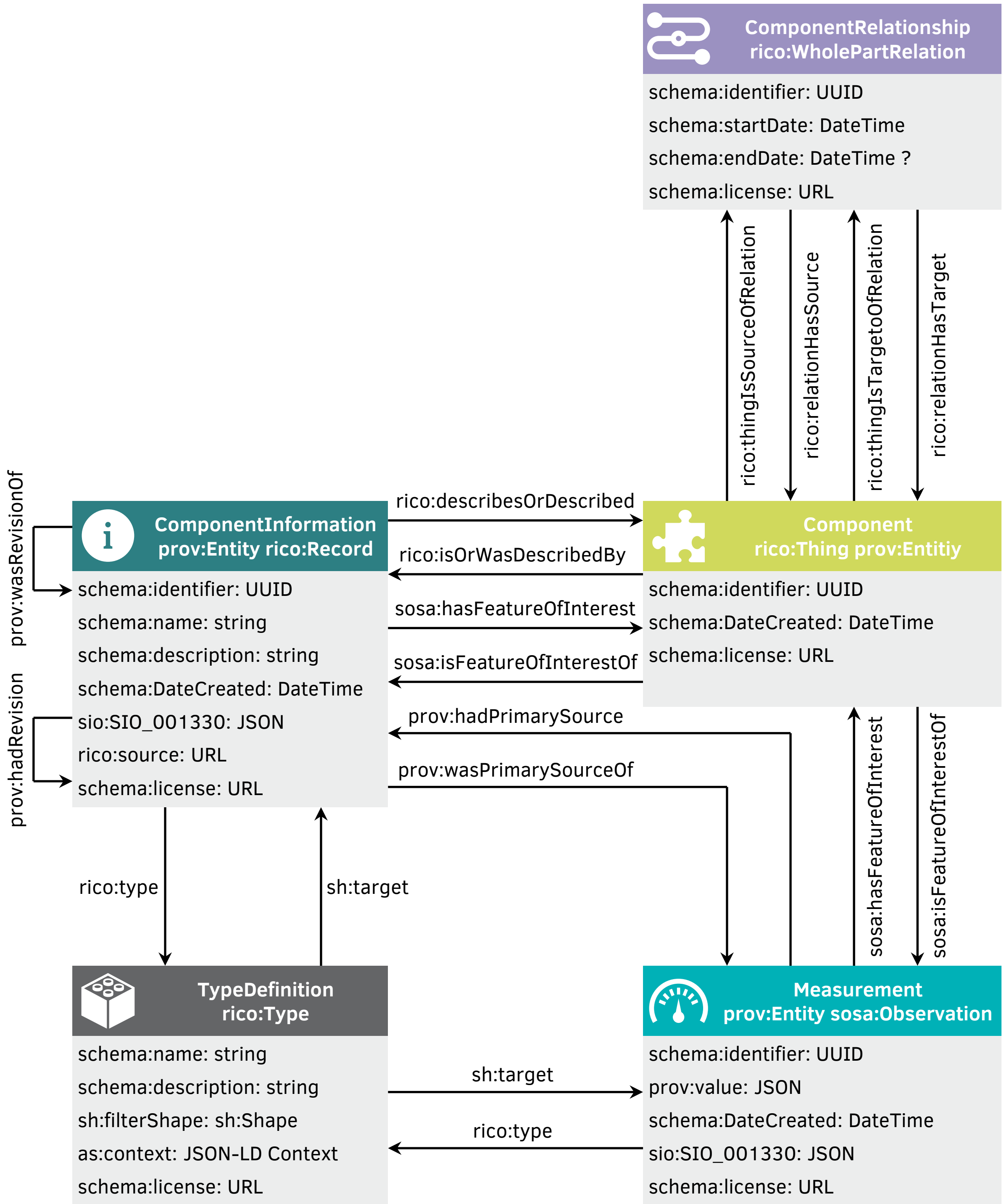


Figure 4.7: The contextualization model linking assets, represented by `Components`, their metadata, represented by `ComponentInformation`, and measurement data, represented by `Measurements`. All the data and metadata are always associated with a `TypeDefinition`. `Measurements` are linked to the system they have been measured with, the component of interest, and the configuration of both at the time of measuring. For increased findability, all relations are bidirectional.

4.4 Proof-of-concept

To demonstrate the applicability of the developed data structures and models, three simulated measuring systems, called *Dummies*, have been defined for a proof-of-concept. For this artificial validation scenario, a laser tracker is considered, which tracks the position of a mobile robot located on some shopfloor. A distributed sensor system monitors the shopfloor's environmental conditions (temperature, pressure, humidity). The *laser tracker Dummy* uses the same SOIL model as an actual device and stresses the modularity of SOIL, the context-sensitive override of names, descriptions, and default values, and the streaming and event features. The *mobile robot Dummy* is a simplified and fully artificial SOIL model to test the correctness of the extension feature for *Components*. The *environmental monitoring Dummy* consists of an arbitrary number of identical, distributed sensors and verifies the correct implementation of dynamic *Components*.

All three systems have later been implemented as *FAIR Sensor Services* based on a SOIL model (see Section 6.2). The three *Dummies* are designed to capture all features and aspects of the defined meta structure and metamodel. While the complete data structures and models of all three *Dummies* can be viewed in Bodenbenner [Bode24c], this description of the proof-of-concept focuses on the mobile robot to illustrate the mapping between the data structure and the data model.

The elements of the data structure (cf. Section 4.2.1) of the simulated mobile robot are shown in Figure 4.8 and the serialization of the measurement TCP with all attributes as required by the meta structure is shown in Listing 4.2. The complete data structure has been published at [Bode24c]. The core nodes of the RDF metadata graph compliant with the metamodel (cf. Section 4.2.3) are depicted in Figure 4.9. The mapping between the data structure and the data model becomes visible when comparing Figures 4.8 to 4.9.

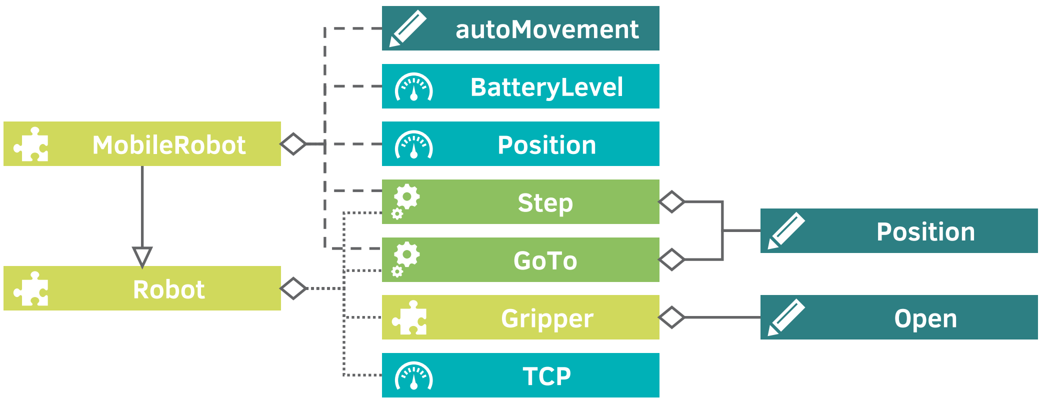


Figure 4.8: Overview of the data structure of the mobile robot that consists of the mobile platform and the six-axis robot on top modeled as individual components. The lines to the components of the Robot and MobileRobot are differently dashed to be distinguishable.

The creation of the data structure and model has been done in synergy with the validation and verification described in Chapter 6. The reason for that is the immense complexity and size of the entire data structure and data model. The serialization of complete static metadata graph to RDF-Turtle consists of fifteen nodes split into fifteen files and over 400 lines of code, of which many appear repeatedly. For comparison, the SOIL model consists of only one

```
{
    "uuid": "MEA-Tcp",
    "name": "Tool Center Point",
    "description": "Tool center point of a six-joint robot.",
    "profile": "RobotTcp",
    "datatype": "float",
    "unit": "unit:M",
    "dimension": [3],
    "range": [-0.5, 0.5],
    "value": [0, 0, 0]
}
```

Listing 4.2: The data structure of the variable "TCP" of the simulated mobile robot serialized to JSON.

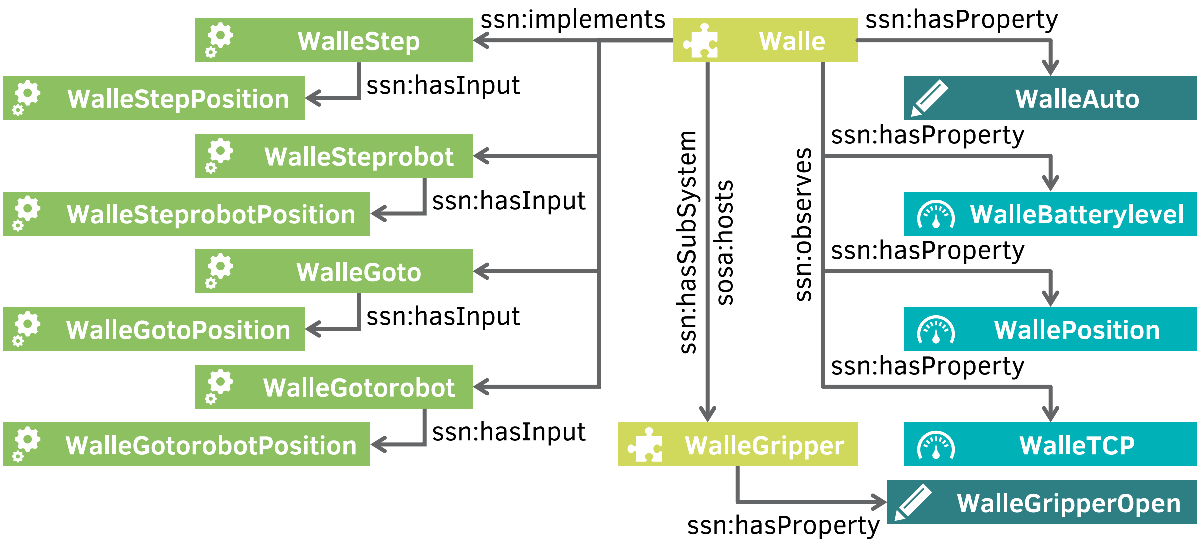


Figure 4.9: Overview of the core nodes of the RDF metadata graph of the mobile robot. For the sake of readability, the bidirectional relations are omitted. The automatically derived names of the nodes ensure uniqueness yet reflect the IDs of the data structure shown in Figure 4.8.

file with 112 lines. This complexity introduces manifold potential sources for mistakes and errors in the manual definition. Creating the bidirectional relations of the data model by hand is error-prone, as one direction might be easily forgotten. Moreover, the manual definition of the model and structure is tedious and repetitive. As can be seen in the data model in Figure 4.9, the variable *Position* is required five times with minor differences only. While most attributes, such as the dimension or datatype, are identical, the name and description might differ slightly. Thus, the manual creation might be subject to "copy-paste errors".

Therefore, the structures and models here have not been created by hand. Following the DSR methodology, a SOIL model has been designed for each *Dummy* using the syntax and semantics later described in Section 5.4.1. The implemented *SOIL Workbench*, see Section 5.4.2 has been executed to generate the data structure and model shown here. Errors and insufficiencies of the generated models have been mitigated by fixing errors in the implementation, correcting mistakes in the meta structure and model, and extending the capabilities of SOIL. Thus, this proof-of-concept does not only prove the suitability of the meta structure and metamodel. It is also directly embedded in the *Design Cycle* of SOIL described in the next chapter.

4.5 Intermediate summary

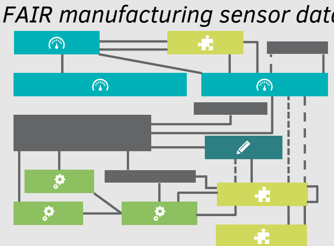
The definition of *FAIR* data models for manufacturing sensor data has been initialized by an in-depth investigation of available ontologies providing the controlled terminology required for adequate machine-actionable and interoperable representation of manufacturing sensor data. Domain-relevant controlled terminology has been identified and used to define metamodels based on RDF and *Semantic Web* technologies for the semantic sensor data and contextualization models. These two models are the main research artifacts (shown in Figure 4.10) of the design cycle of the second RGQ and lead to the preliminary answer to that question.

Preliminary answer to the Research Guiding Question 2

How can manufacturing sensor data be described to be findable and interpretable in the applications' context?

To provide *FAIR manufacturing sensor data*, the sensor data itself and its context are generally defined by dedicated metamodels. The sensor model describes the measurement data and all characteristics of its origin in a unified way. The contextualization model describes the relation to relevant assets (process, resource, product) and their metadata. This preserves the measurement data's provenance, ensuring their compatibility and linking between measurement data from different sources. To achieve a *FAIR* data description according to the previously defined *FAIRness* criteria, controlled terms from available ontologies are used and slightly extended if insufficiently defined.

Artifact 2.1 Sensor model for *FAIR manufacturing sensor data*



Artifact 2.2 Context model for *FAIR manufacturing sensor data*

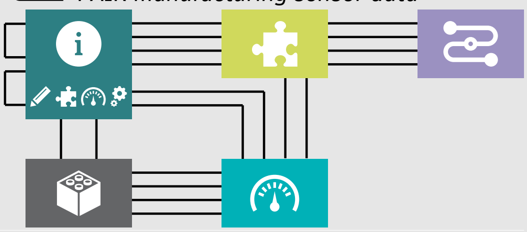


Figure 4.10: The data and contextualization model form the two research artifacts of the second research guiding question. Based on *Semantic Web* technologies, such as ontologies and controlled vocabulary, the data models provide findable and interpretable manufacturing sensor data.

In addition, a lightweight data structure based on key-value pairs has been developed for each model, simplifying the representation of the sensor and context data. On the one hand, these data structures can be serialized to plain JSON and XML format and are therefore easily deployable to industrial infrastructures. On the other hand, the structure for sensor data serves as the foundation of the IDL for the low-effort definition of interoperable interfaces for measuring systems developed in the next chapter, which is then used to generate the communication interface and the full semantic data model of the measuring system.

The *Sensor Interfacing Language* for the development of *FAIR Sensor Services* 5

The *FAIR Sensor Service* integrated into an ecosystem must realize diametrically opposed actions: On the one hand, highly device-specific but ecosystem-agnostic tasks, such as execution of measurement routines, and on the other hand, tasks which are device-agnostic but ecosystem-specific, like providing its data in an appropriate format and transmitting it with a defined protocol to other entities in the ecosystem. This imposes high complexity on the *FAIR Sensor Service* and bears the risk of an entangled, highly coupled, but hardly maintainable implementation. Therefore, implementing these two aspects (device-logic and communication interface) should be decoupled to tackle this risk. While implementing the device-specific logic requires individual work for each measuring system due to the very high heterogeneity, the communication interface can be generalized by using standardized protocols (such as HTTP, MQTT, or OPC UA) and open-source libraries for the implementation. These two independent tasks of which one can be automated (the implementation of the communication interface), while the other remains manual and individual work (the implementation of the device-specific logic), motivating the usage of model-driven software engineering (MDSE) and domain-specific languages (DSLs). Moreover, according to **CA.2 Multi-Protocol**, it is not sufficient to choose a single communication protocol and data format for the CPMS for targeting all use-cases of the IIoT [Derh17a; Când10]. Because every single protocol required significantly increases the implementation effort, employing MDSE is crucial to automate the implementation of the communication interface. A third aspect of implementing a *FAIR Sensor Service* is the device-specific instantiation of the *FAIR data models* developed in Chapter 4 to acquire uniformly described and structured sensor data. However, the knowledge base of information science to create semantic data models as of Section 4.2.3, network expertise of the communication protocols, and production engineering familiar with the measurement routines usually do not overlap. Thus, joining these competencies is required for this highly interdisciplinary development of CPMS [Ye18; Gürd16]. Finally, how these three aspects can

The author has published intermediate parts of the work described in this chapter in

Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon, and Robert H. Schmitt. "Domain-Specific Language for Sensors in the Internet of Production". In: *Production at the Leading Edge of Technology*. Ed. by Bernd-Arno Behrens, Alexander Brosius, Wolfgang Hintze, Steffen Ihlenfeldt, and Jens Peter Wulfsberg. Lecture Notes in Production Engineering. Berlin, Heidelberg: Springer, 2021, pp. 448–456. ISBN: 978-3-662-62138-7. DOI: 10.1007/978-3-662-62138-7_45

Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. "FAIR Sensor Services - Towards Sustainable Sensor Data Management". In: *Measurement: Sensors* 18 (2021), p. 100206. ISSN: 26659174. DOI: 10.1016/j.measen.2021.100206

Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. "Model-Driven Development of Interoperable Communication Interfaces for FAIR Sensor Services". In: *Measurement: Sensors* 24 (2022), p. 100442. ISSN: 26659174. DOI: 10.1016/j.measen.2022.100442

Matthias Bodenbenner. *Description and Manual of the SensOr Interfacing Language*. Zenodo, May 22, 2024. DOI: 10.5281/zenodo.11239479 be jointly addressed by MDSE and DSMLs is formulated as the third research guiding question and investigated in this chapter to deliver step three of the *FAIR* DLC as shown in Figure 5.1.

Research Guiding Question 3

How can controlled vocabularies be reused systematically by domain-specific modeling languages to provision accessible and interoperable manufacturing sensor data?

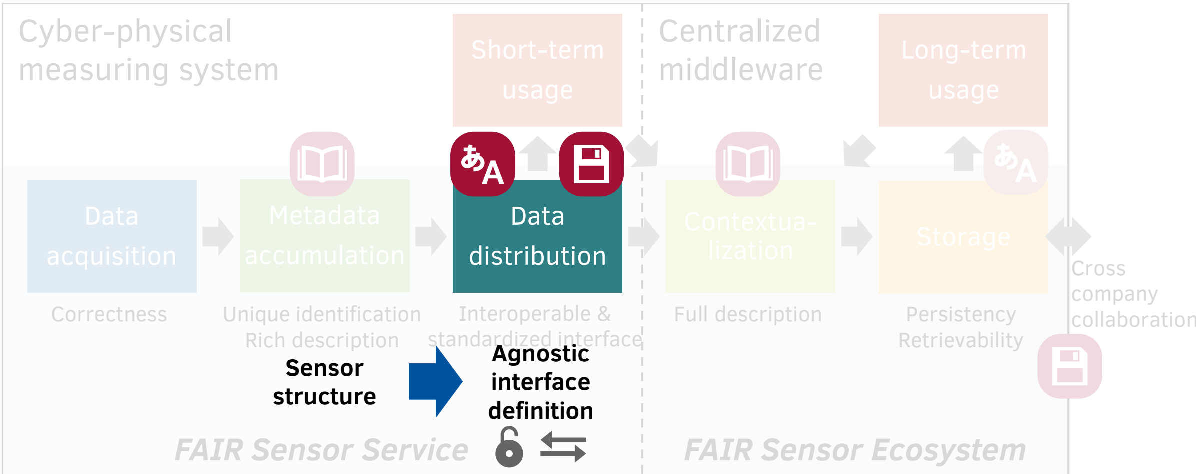


Figure 5.1: The approach and results described in this chapter towards answering the third RGQ delivers the realization of the third step of the *FAIR* DLC: The interoperable communication interface for providing accessible, and ultimately *FAIR*, *manufacturing sensor data*.

The basic idea of the solution of this third RGQ is depicted in Figure 5.2, which is first grounded to the knowledge base by a review of the specific state-of-the-art of the development of interfaces for CPMS and scientific approaches. After that, the envisaged development procedure using SOIL is explained, and the features of this IDL are defined. After the syntax and semantics of SOIL have been developed, SOIL is validated by implementing the *SOIL Workbench* and *SOIL Editor* and executing translation tests. This chapter closes with an intermediate conclusion of the third RGQ.

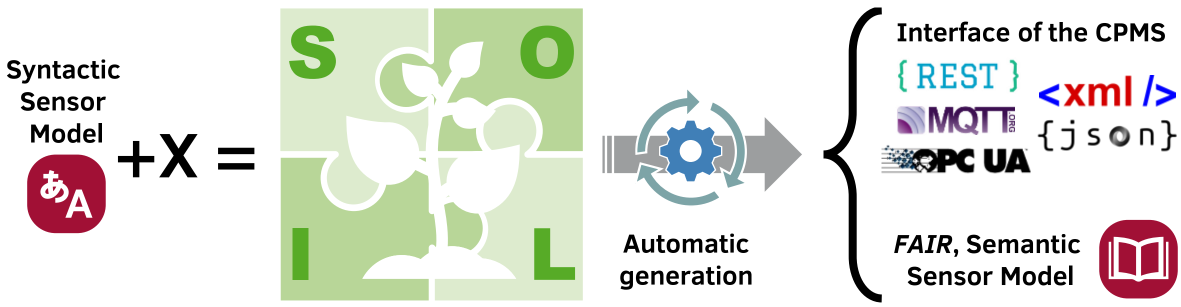


Figure 5.2: The basic idea of the answer to the third RGQ is the definition of the syntax and semantics of a IDL, namely SOIL, based on the sensor meta structure (cf. Section 4.2.1). Based on the specific SOIL model of a particular CPMS, the semantic data model and the communication interface can be automatically generated.

5.1 Development workflow of a *FAIR Sensor Service* using the *Sensor Interfacing Language*

The central hypothesis of this doctoral thesis is that the implementation of the *FAIR Guiding Principles* for manufacturing sensor data can be operationalized using MDSE and a IDL, namely the *Sensor Interfacing Language*. Therefore, this section will detailly motivate SOIL and illustrate this approach's benefits by analyzing the development and implementation process of a CPMS. For the provision of *FAIR manufacturing sensor data*, the SOIL-based CPMS is implemented by a three-layer software architecture depicted in Figure 5.3 conceptualizing the *FAIR Sensor Service* with the following layers:

1. The execution of the inner device logic, acquisition, and digitization of measurements based on physical principles using the sensor hardware.
2. The *FAIR* representation of the measurement data and annotation with metainformation according to the data model described in Section 4.2.3 to fulfill the *FAIRness* criteria **CF.2 Searchability**, **CI.1 Controlled Terms**, **CI.3 Information Model**, **CI.4 Consistency**, **CR.1 Domain-Specific Terms**, and **CR.3 Metrological Information**.
3. The provision and transmission of the annotated measurement data in interoperable data formats using a standardized communication protocol, addressing **CA.1 Standard Protocols**, **CA.4 Authorization and authentication**, and **CI.2 Standardized Format**.

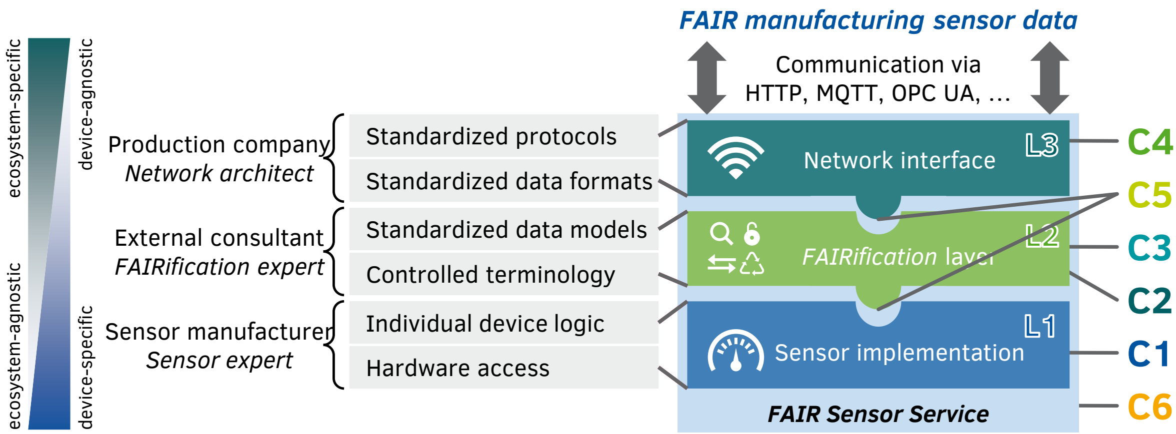


Figure 5.3: Three-layer architecture of a *FAIR Sensor Service*. The envisaged capabilities of the system are mapped to each layer together with the responsible stakeholders. The stakeholders might also be one or different persons in the same company. Furthermore, it is shown how the layers address the challenges of the *FAIRification* process, cf. Figure 2.3.

The development of these layers can be decoupled by employing MDSE and SOIL, see Figure 5.4. Based on the meta structure defined in Section 4.2.1, SOIL is used to define a device-specific data model of the CPMS. After checking this device-specific SOIL model for correctness, the generalizable layers are automatically translated to:

1. A template for implementing the inner device logic (Layer 1).
2. The full semantic data model, for providing the *FAIR* measurement data (Layer 2).
3. A mapping to endpoints of the network interface for request/response and publish/ subscribe interaction with the CPMS (Layer 3).

Loading pre-implemented libraries for providing specific protocols and data formats, the third layer is fully implemented based on the protocol-agnostic second layer as required by **CA.3 Protocol-Agnostic Modeling** equally supporting **CA.2 Multi-Protocol** (cf. Section 6.2.2). The second layer has also already been completed by the automatic generation of the full semantic data model. The first layer must be finished by hand as the device-specific logic and measurement procedures can hardly be generalized due to the immense heterogeneity.

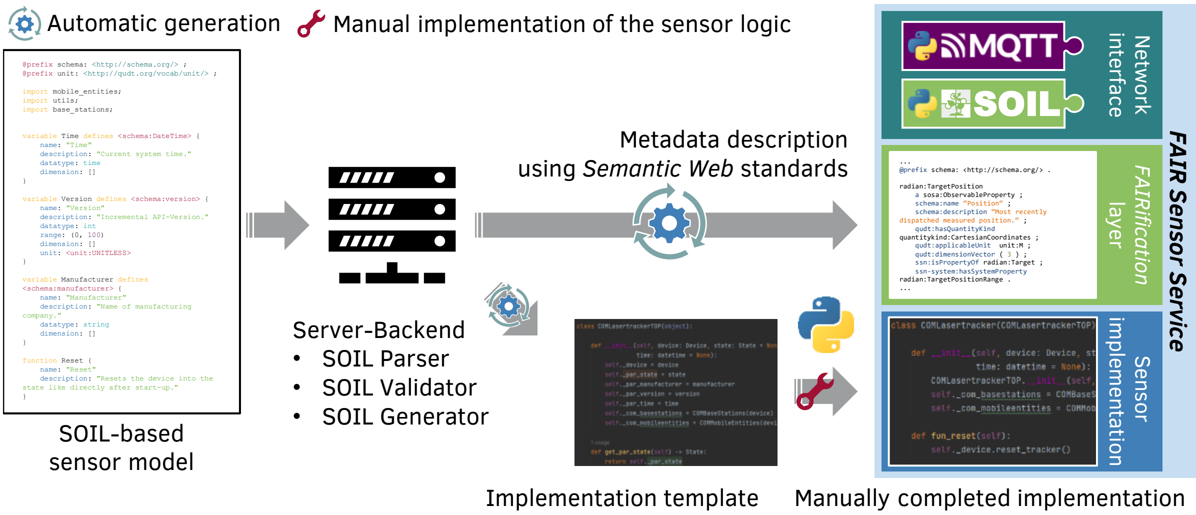


Figure 5.4: Development workflow of a *FAIR Sensor Service* using SOIL. Based on a SOIL model, the *SOIL Workbench* implemented in Section 6.2 parses and validates the model and then translates it into the semantic data model and a template for implementing the sensor logic. The network interface is completed by importing pre-implemented libraries, see Section 6.2.

Thus, the purpose and contribution of SOIL is to streamline the development process of CPMS, providing long-term reusable sensor data by decoupling the implementation of the individual layers of the CPMS' software and leaving only the device-specific non-generalizable implementation of the CPMS' measurement logic to the domain-expert. In particular, SOIL should meet the following functional and non-functional requirements to facilitate adoption in an industrial application:

- As fewer individual and distinct elements as possible to minimize the learning effort,
- compatibility with state-of-the-art data models (e.g., OPC UA, AAS), so that such that these models can be generated based on a SOIL model,
- textual and visual representation to address programmers and non-programmers,
- capable of modeling the characteristics and features of sensors and CPMSs,
- translatable to a full semantic data model, but hide semantic concepts to be usable by developers, which are no data stewards,
- serializable using open data formats for high *FAIRness* of the SOIL models itself,
- modularity and flexibility to allow reuse of (foreign) SOIL models or parts of these,
- the textual version should have minimal validation and translation overhead, i.e., only the *SOIL Workbench* (and the *SOIL Editor*) (see Section 5.4.2) should be required, and
- the language itself must be *FAIR*, i.e., all terms and keywords must be defined properly.

5.2 IIoT communication protocols and data formats

To provide an interoperable and accessible interface for *FAIR Sensor Services*, the service must implement standardized and commonly used communication protocols and data formats. An initial review of technologies currently used in the manufacturing domain has already been done in Section 2.3 with a focus on the identified deficits of industrial data management and the challenges of the *FAIRification* process. This section highlights the capabilities of communication protocols and data formats by briefly recapitulating the studies of Montavon [Mont21] as the foundation of this thesis.

Montavon [Mont21, p. 48ff] identified eight capabilities communication protocols must have to provide all the ways of interaction with a CPMS required from the client's perspective. Table 5.1 reviews the findings of Montavon and extends it by an evaluation of I++ DME. Based on these findings, this thesis implements *FAIR Sensor Services* with an OPC UA interface and a combination of an HTTP and MQTT interface. Besides data serialization, all capabilities are provided by combining HTTP and MQTT. This combination provides higher flexibility and maintainability, easing the development process compared to OPC UA as they focus on the communication aspect only and are agnostic of the data model and format (cf. Section 2.3). The data serialization can be seamlessly integrated using additional open-source libraries. XML and JSON are widely used data formats for serializing data structures and exchanging them via HTTP and MQTT (cf. Section 2.3). Both are natively designed to represent syntactically interoperable data by key-value pairs. In addition, extensions exist for both formats to represent semantically interoperable graph data based on RDF, namely JSON-LD [Kell20] and RDF/XML [Gand14]. Thus, JSON and XML are best suited for implementing the serialization of (*FAIR*) manufacturing sensor data. OPC UA enforces the usage of XML.

Table 5.1: Evaluation of the capabilities of IIoT protocols as identified by Montavon [Mont21, Table 4.1, p. 49]. In this thesis I++ DME has been added. The evaluation is done on the following scale: ○ not implementable, ◔ must be manually realized with high effort, ◑ can be manually implemented with medium effort, ◕ implementable with low overhead effort, ● natively included in the standard.

	Publish/ Subscribe	Request/ Response	Serialization Model	Authentication/ Authorization	Structured Identifiers	Model Browsing	CRUD Data View	Function Invocation
OPC UA	●	●	●	●	●	●	◑	●
MQTT	●	○	○	●	●	○	◔	◔
REST via HTTP	○	●	◔	◕	●	●	●	◕
gRPC	◔	●	●	◑	○	○	○	●
MTConnect	○	●	●	○	●	●	◔	○
I++ DME	◑	●	●	○	●	●	○	●

5.3 Tools and languages for the model-based definition and generation of communication interfaces

This section investigates state-of-the-art approaches and methods for developing and generating service and system interfaces using MDSE and the definition of semantic data models. The tools reviewed cover a wide range of applications, from industrial infrastructures to research data management.

gRPC Remote Procedure Calls and Protocol Buffers

gRPC Remote Procedure Calls (gRPC) is a framework developed and maintained by Google that implements the communication protocol Remote Procedure Call (RPC). The protocol is mainly used to establish connections and communication between microservices based on a minimal description of its interface, based on executable functions [Goog24a]. Based on Google's IDL, called protobuf, the exchangeable messages, callable functions, and all their arguments and returns (expressed as messages) can be defined [Goog24b]. The description of the interface using protobuf is independent of the programming language for implementing the microservice. Based on an interface definition in Protocol Buffers (protobuf), the interface implementation can be generated for multiple GPLs, e.g., Python, C++, Java, C#, and JavaScript [Goog24a]. gRPC implements the Request/Response communication pattern via HTTP and does not support Publish/Subsribe. Thus, gRPC and protobuf do not provide syntax and semantics to represent Components. Moreover, protobuf misses the possibility of integrating *Semantic Web* technologies and controlled vocabularies.

Web Services Description Language

The Web Services Description Language (WSDL) is an IDL based on XML and XSD to define the interface of web services [Chin07]. The concepts are comparable to protobuf and, therefore, come with the same deficits. There exists an extension of WSDL, called Web Service Semantics (WSDL-S) [Akki05] that introduces semantics for WSDL so that controlled vocabularies can be used. WSDL-based services have been majorly replaced by RESTful services, as they are easier to maintain and decoupled from the client interacting with the service [Ahma23].

MontiThings and MontiGem

Another approach for developing Internet of Things (IoT) applications to handle measurement data has been published by Kirchhof et al. [Kirc22]. The *MontiThings* modeling infrastructure enables the definition of IoT devices, the parameters and data they provide, and how the data is processed. The authors apply MDSE and generative software implementation for that. However, the approach focuses on the data flow between distributed data processing entities. Suitability for long-term storage and data management following the *FAIR Guiding Principles* is not considered. Service interfaces developed with MontiThings can be easily integrated into a larger Digital Twin (DT) of a process or product using MontiGem [Mich21]. MontiGem is another DSL for the definition of DTs and the specification of UI elements providing a graphical representation of the DT.

NFDI4Ing Metadata Profile Service

The NFDI4Ing Metadata Profile Service (MPS) is a visual web application for the definition of MPs [Fuhr24], see Figure 5.5. The application targets researchers with no data modeling experience and supports them by providing building blocks for MPs defined by the community [Grön23]. The creation effort and complexity of an MP compliant with *Semantic Web* standards and the *FAIR Guiding Principles* is significantly lowered using the MPS. But, the UI still contains many features only relevant to advanced users, so beginners might be overwhelmed by the possibilities, making it hard to identify and concentrate on the basics. Moreover, the generated MPs need to be manually integrated into data generation and processing software, leaving significant effort to the researcher, i.e., the implementation of the communication interface and the internal systems logic of the CPMS.

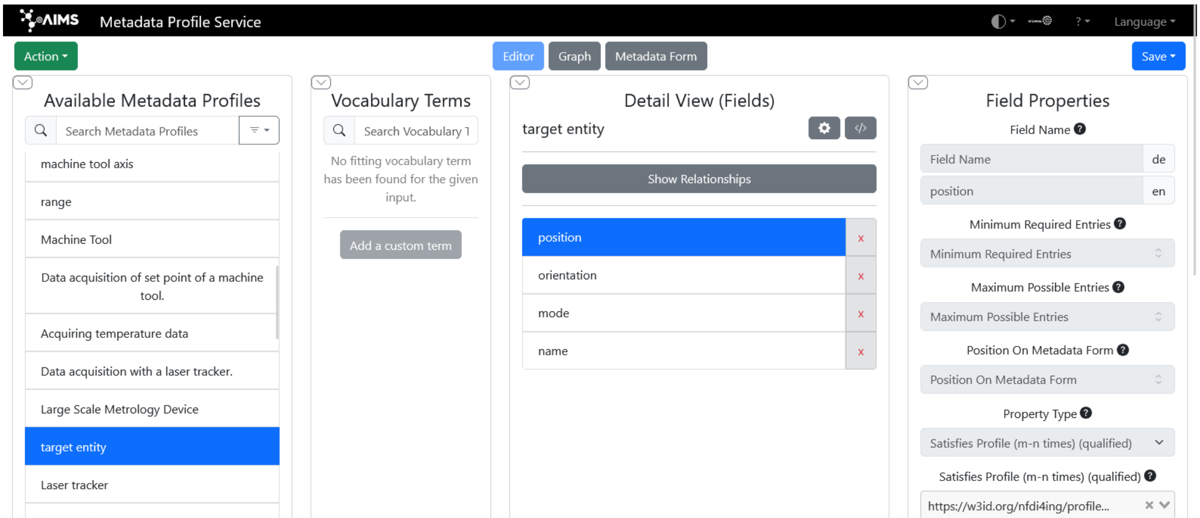


Figure 5.5: Screenshot of the NFDI4Ing Metadata Profile Service showing the MP of the target of an API Radian™ Lasertracker created by the author.

CEDAR Workbench

The CEDAR Workbench is a web application for managing MPs and creating these for scientific datasets [Muse15]. The tool is also designed as a library for metadata sets based on the MPs created with CEDAR. The tool explicitly addresses the *FAIR Guiding Principles* and evaluates the *FAIRness* of created MPs and metadata sets [Muse22]. To strictly follow guideline **I1**, CEDAR automatically searches for matching controlled terms if the user does not provide such and suggests found terms for replacing the non-controlled term. The tool is comparable with the MPS of NFDI4Ing and also relies on a community-driven approach for the creation, share, and reuse of MPs. However, the integration of the MPs created with CEDAR into data acquisition and processing workflows is not addressed.

Related scientific work and approaches

Besides all these frameworks and tools, several other approaches can be found in scientific literature. DSLs for interfacing and interconnecting devices in industrial context have been published under the names IoTDSL [Amra17] and GIMLE [Toml17]. While the physical capabilities of the considered devices can be modeled in greater detail, metrological concerns are not covered. Generally, most approaches, such as SEAL [Elst13], GIMLE, or Midgar, have a strong focus on usability and reduction of complexity but missing central concepts for ensuring data integrity and traceability, which are essential for metrological applications [Hack18]. Thus, criteria **CR.3 Metrological Information** and **CR.1 Domain-Specific Terms** remain open. The generation of OPC UA models has also been investigated in general [Gold12; Busb24] and in particular based on UML diagrams [Pauk18] or using the *Eclipse Modeling Framework* [Frie20]. However, OPC UA lacks the integration of controlled terminology based on semantic-web standards, which can not be compensated by the CSs, so that criterion **CI.1 Controlled Terms** is unfulfilled. Moreover, focusing on OPC UA only means that criteria **CA.2 Multi-Protocol** and **CA.3 Protocol-Agnostic Modeling** are not addressed. To compensate for the lack of knowledge in implementing CPS Francalanza et al. [Fran17] introduced a software tool assisting developers with an intuitive UI. The tool should help the authors to understand the consequences of design decisions and mitigate potential issues based on the decision. However, the tool does not provide any automation mechanism.

5.4 The *Sensor Interfacing Language*

Based on the protocol-agnostic interface by Montavon [Mont21], the author developed an early visual version of an IDL [Bode21b]. SOIL has then been strongly formalized as a textual IDL using the concepts of this early visual representation as described in Section 5.4.1. The *SOIL Workbench* has been implemented for parsing, validating, and translating textual SOIL models according to the defined syntax and semantics. The *SOIL Editor* has been implemented in the second step, providing a visual representation using standard UI elements. Both tools are described in Section 5.4.2. The correctness and functionality of SOIL have been validated by unit tests and the development of models for the three simulated measuring systems, called *Dummies* (cf. Section 4.4), covering all features of SOIL in Section 5.4.3.

5.4.1 Syntax and semantics of the *Sensor Interfacing Language*

The basic syntax of the textual language is inspired by JSON, Python, and Turtle to mirror familiar syntax patterns, increasing usability and reducing learning effort. SOIL builds upon the syntactic data structure for sensors defined in Section 4.2.1. Thus, the language formalizes the definition of *Components*, *Functions*, *Parameters*, and *Measurements*. Beyond the static metainformation specification of the four *Elements*, the language provides dedicated mechanisms to maximize modularity and reusability of already defined *Elements*. For that, the language distinguishes between *types* and *instances*. To specify the attributes of an *Element*, a dedicated *type* is declared and defined, e.g., specifying the attributes, like name, description, datatype, etc., of a *Variable type* “Temperature”, as shown in lines 4-11 of Listing 5.1. The definition of an element starts with a dedicated keyword for each of the for *Elements* followed by a locally unique name, which can be used to refer to that specific *Element type* at other positions in the model. An *Element type* can be *instantiated* as the child of a *Component* by using this unique name as the reference to the *type* (see line 17 of Listing 5.1). By this decoupling of type definition and type instantiation, a high degree of modularity can be achieved because types can be easily reused and updated. The instantiation syntax and mechanism are reused from common GPLs.

```
1  @prefix quantitykind: <https://qudt.org/vocab/quantitykind/>;
2  @prefix unit:  <https://qudt.org/vocab/unit/>;
3
4  variable Temperature defines <quantitykind:Temperature> {
5      name: "Temperature"
6      description: "The current temperature."
7      datatype: float
8      range: (-20, 50)
9      dimension: []
10     unit: <unit:DEG_C>
11 }
12
13 component TemperatureSensor {
14     name: "Temperature sensor"
15     description: "A simple temperature sensor measuring ambient temperature."
16     measurements:
17         Temperature temperature
18
19     streams:
20         temperature: fixed(5)
21
22     if temperature < 0: warning("The ambient temperature is low.")
23 }
24
25 interface TemperatureSensor SensorOne{}
```

Listing 5.1: Exemplary definition (lines 4-11) and instantiation (line 17) of a *Variable type* “Temperature”. It is a floating point with a scalar value of -20 to 50 °C. Using the keyword `defines`, the type is mapped to a term defined in a semantic namespace provided by an available ontology. The same mechanism is used to define the physical unit unambiguously.

A SOIL model can be split into multiple files, of which each file starts with importing other files and defining the prefixes of controlled vocabularies to be used throughout the file. Because the model can be split into multiple files, a high degree of modularity and flexibility, improved readability, and high reusability are achieved. After that, the types of all *Parts* of the SOIL model are defined. The five different definable *Part types* are summarized in Table 5.2. The order of definition does not matter, i.e., *Part types* can be instantiated before they are defined. Because SOIL only defines the static metainformation of the measuring system, the actual measurement value of *Variables*, their uncertainty, and timestamp are not part of the syntax and semantics of SOIL. All attributes of the three *Element types* are listed in Table 5.3. Moreover, SOIL implements all six datatypes of *Variables* as summarized in Table 5.4

Modifiers

SOIL defines four *modifiers* for assigning specific meaning or behaviour to children of *Components*. There is one *modifier* for each of the four *Elements* of SOIL, namely:

1. *Measurements* can be marked as `internal`. With this *modifier*, the developer can indicate that this *Measurement* is a characteristic of the CPMS itself, such as a mobile unit’s battery level or signal strength. The *modifier* adjusts the value of the `ssn:isPropertyOf` attribute of the `soil:MeasuredProperty` in the generated semantic data model. This *modifier* directly targets **CI.5 Linked Data**.

Table 5.2: All five different *Part types* of a SOIL model, directly mapping the syntactic data structure defined in Section 4.2.1.

Part	Description
`variable`	Defines the *Variables* of a SOIL model according to Section 4.2.1.
`function`	Defines the *Functions* of a SOIL model according to Section 4.2.1.
`component`	Defines the *Components* of a SOIL model according to Section 4.2.1. Also, implements an inheritance mechanism. If a *Component type* `extends` another *Component type*, it contains all children as its own (in contrast to composition).
`enum`	An enum contains a list of names which restrict the permissible values of a *Variable* referencing this enum datatype.
`interface`	Instantiates and specifies the *root Component* of the tree-shape model. It is handled as the declaration of the main method in Python. To generate the semantic model, metadata profiles, and the interface of a *FAIR Sensor Service*, the model must contain exactly one interface in the primary file. Interface definitions in imported files are ignored.

Table 5.3: Attributes of the individual *Element types* of a SOIL model. **Legend:** ✓ mandatory, ∽ optional, ✗ forbidden.

Attribute	Component	Function	Variable	Description
`name`	✓	✓	✓	Human-readable short title of the *Element*.
`description`	✓	✓	✓	Human-readable description of the *Element*. Can be extensive and context-sensitive.
`components`	∽	✗	✗	List of all child *Components*.
`functions`	∽	✗	✗	List of all child *Functions*.
`parameters`	∽	✗	✗	List of all child *Parameters*.
`measurements`	∽	✗	✗	List of all child *Measurements*.
`arguments`	✗	∽	✗	List of all arguments of the *Function*.
`returns`	✗	∽	✗	List of all returns of the *Function*.
`datatype`	✗	✗	✓	Datatype of the value of the *Variable* (cf. Table 5.4).
`dimension`	✗	✗	✓	Mathematical dimension of the value of the *Variable*.
`range`	✗	✗	✓ \ ∽ \ ✗	Range of permissible values. Specification and necessity depends on the datatype (cf. Table 5.4)
`unit`	✗	✗	✓ \ ✗	Physical unit of the value of the *Variable*. Necessity depends on the datatype (cf. Table 5.4)
`streams`	∽	✗	✗	List of all streams for child *Measurements*.
`events`	∽	✗	✗	List of all events. Keyword not part of the syntax.
`overrides`	∽	∽	✗	List of all context-sensitive overrides of names and descriptions of child *Elements*. Keyword not part of the syntax.

Table 5.4: Specification of the six datatypes for *Variables* in SOIL. For each datatype, whether the unit and range attributes are mandatory is specified. **Legend:** ✓ mandatory, ~ optional, ✗ forbidden. A string's range refers to the string's allowed length with the natural lower limit of zero. The range of an enum value is defined by referencing an enum type. The datatype in the SOIL model does not imply a specific accuracy or implementation, e.g., in terms of the number of digits for int or float.

Datatype	Description	Unit	Range	Examples
`bool`	A Boolean *Variable* distinguishes between true and false.	✗	✗	powered on/off, open/closed
`int`	An integer value.	✓	✓	signal strength, counter, version number
`float`	A floating point number.	✓	✓	temperature, position, length
`string`	An unregulated string of characters.	✗	~	manufacturer, tags, addresses
`enum`	If a *Variable* has datatype *enum*, the range of this *Variable* defines a fixed set of permissible values.	✗	✓	state, type, selections
`time`	Time consists of date, time of day, and time zone information. The precision should be as high as possible (ideally up to milliseconds or nanoseconds). A time value is specified according to RFC9557 [Shar24].	✗	~	uptime, system time

2. *Parameters* can be defined to be `constant` so that the default value provided in the SOIL model can not be changed by the client during runtime.
3. *Functions* can have `streaming` behavior, which is particularly useful for complex measurement routines where the same outputs are measured multiple times. A streaming *Function* transmits its outputs via a publish/subscribe protocol, i.e., usually split into multiple consecutive messages. In contrast, a *Function* without this *modifier* provides the outputs once as the direct response to the request triggering the *Function*.
4. *Components* can be modified to be `dynamic`. If a *Component* is dynamic, instances of these *Components* can be created and removed during runtime of the CPMS. For example, this is required by LSM-Systems, such as a motion capture system tracking a varying number of rigid bodies or a laser tracker measuring multiple different targets. The behavior is implemented by generating additional endpoints for the request/response interface to create and remove the child *Components*.

Data streaming and event handling

SOIL includes syntax and semantics to abstract the publish/subscribe communication pattern so that the developer of the measuring system does not need to deal with the protocols explicitly. Another benefit of this abstraction is that the model and implementation are independent of a specific protocol or library. A *Component type* can have streams to send *Measurement* values. The three different stream types of SOIL are explained in Table 5.5. Moreover, a *Component type* can have events to send dedicated messages in case a *Variable's* value fulfills a defined condition. Analogously to OPC UA, SOIL distinguishes different levels of severity of events, defined in Table 5.6. Examples of both features are given in Listing 5.1.

Table 5.5: Definition of three stream types in SOIL for automated transmission of *Measurements*.

Stream	Argument	Description & Example
`fixed`	float value	The *Measurement* (including the measured value) is published at the fixed interval specified in seconds in the SOIL model. Example: `pos: fixed(10)`
`dynamic`	sibling *Parameter*	The measurement value is regularly published at a given interval. The interval is specified by a sibling *Parameter* of the published *Measurement*. Thus, it can be updated during runtime of the *FAIR Sensor Service*. The *Parameter* must be a scalar of datatype `int` or `float` and greater than zero. Example: `pos: dynamic(interval)`
`update`	n.a.	If the measured value changes, the *Measurement* is published immediately. Example: `pos: update`

Table 5.6: Definition of different severity levels for SOIL events. The severity levels are linearly mapped to OPC Severity as suggested by the OPC information model [OPC 21].

Severity	Description	OPC UA severity
`debug`	Should only be given to events used to monitor the behavior during development.	200
`info`	Used to give the client feedback that something has changed, but this change has no major implication to the device's functioning.	400
`warning`	Something has happened that should grab the operators' attention, but an intervention of the operator is not necessarily required.	600
`error`	Something went wrong. The intervention of the operator is required.	800
`critical`	Something happened that imposes a severe serious to the system. The operator's intervention is required as soon as possible.	1000

Integration of controlled terminology

As described, SOIL shall reduce the modeling effort and the developers' expertise for creating data models based on *Semantic Web* standards to ensure criterion **CI.1 Controlled Terms** and support criterion **CR.1 Domain-Specific Terms**. However, automatically determining appropriate controlled terms based on uncontrolled strings is error-prone and often not unambiguously possible as it depends on the information provided by the user and can usually only be done syntactically [Hein24; Fang06]. The syntax of Turtle is employed to incorporate controlled terminology in SOIL. At the beginning of the document, one defines the prefixes used throughout the current file. A defined *Element type* can associated with a controlled term based on the prefixes defined by using the keyword `defines`. Moreover, the unit of *Variables* can be specified using controlled terms. Examples of both are given in Listing 5.1 and Listing 5.2. By that, the required expertise on *Semantic Web* technologies is reduced to be able to search for appropriate controlled terminology using specialized search engines, such as Zazuko[1], LOV[2], or the Terminology Service[3].

[1] https://prefix.zazuko.com/
[2] https://lov.linkeddata.es/dataset/lov/
[3] https://terminology.tib.eu/ts

Context-sensitive metainformation

The mandatory attributes `name` and `description` of an *Element type* store crucial information for the unambiguous interpretation of the *Element* and are essential to ensure criterion **CR.2 Provenance** is met. Thus, this information must be adaptable depending on the *Element's* context. Listing 5.2 illustrates such a case, where a *Variable* "Position" is once used to provide the coordinate of the platform of a mobile robot and once to give the position of the end effector of the robotic arm mounted on the mobile platform. Similarly, the default values of *Parameters* may be context-sensitive. So, a default value of a *Parameter* can be overwritten when the *Component type* is instantiated employing the Python syntax for calling the constructor method of a class, as shown in line 27 of Listing 5.2.

```
1  @prefix quantitykind: <https://qudt.org/vocab/quantitykind/>;
2  @prefix unit: <https://qudt.org/vocab/unit/>;
3
4  variable Position defines <quantitykind:CartesianCoordinates> {
5      name: "Position"
6      description: "A 3D position characterized by three axes (abscissa, ordinate, and applicate) and usually denoted by (x,y,z)."
7      datatype: float
8      dimension: [3]
9      range: (-100, 100)
10     unit: <unit:M>
11 }
12
13 component Robot {
14     name: "Robot"
15     description: "A six-axis robotic arm."
16     parameters:
17         Position end_effector_position = (0,0,0)
18
19     end_effector_position.name = "End effector position"
20     end_effector_position.description = "The 3D position of the end effector of the robot, denoted by (x,y,z)."
21 }
22
23 component MobileRobot {
24     name: "Mobile Robot"
25     description: "An automated guided vehicle with a robotic arm attached."
26     components:
27         Robot robot_arm(end_effector_position=(0,0,0.5))
28     parameters:
29         Position coordinate
30
31     coordinate.name = "Coordinate"
32     coordinate.description = "The 3D position of the mobile platform, denoted by (x,y,z), with respect to a fixed global reference point."
33 }
```

Listing 5.2: Definition of a generic *Variable* to provide a Cartesian coordinate and instantiation of this *Variable* twice in different contexts with individual name and description of the *Measurement*.

Complete examples employing all features of SOIL have been created for validation (cf. Chapter 6) and can be found in the Appendix: a complete SOIL model for an API Radian™ laser tracker (cf. Appendix D), a partial model of a Leitz PMM-C Precision™ encapsulated by an I++ DME server (cf. Appendix E), and an OptiTrack™ MCS (cf. Appendix F).

5.4.2 Implementation of the *SOIL Workbench* and *SOIL Editor*

For operationalization, the *SOIL Workbench* has been implemented. The purpose of this software is to (i) parse SOIL models using the syntax explained in Section 5.4, (ii) verify the model by checking if the model is valid considering the defined semantics and (iii) translate the model into a full semantic data model and an implementation template for the *FAIR Sensor Service* if the model could be parsed and validated successfully. The *SOIL Workbench* has been implemented using MDSE based on the *MontiCore Language Workbench* provided by Rumpe et al. [Rump21] and is therefore written in Java. An overview of the software architecture of the *SOIL Workbench* is given in Figure 5.6.

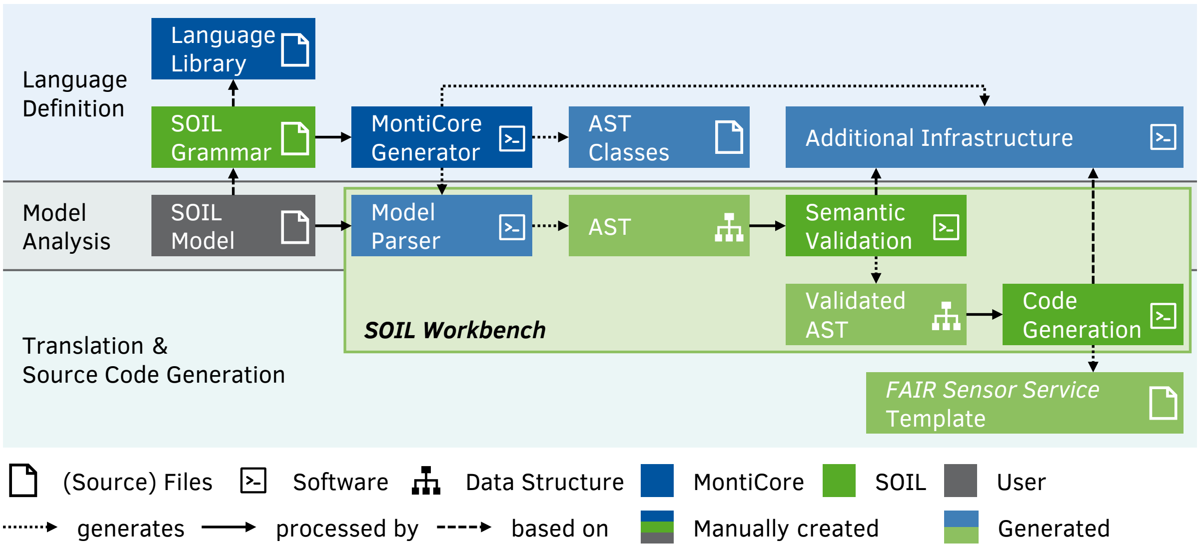


Figure 5.6: The software architecture and execution workflow of the *SOIL Workbench* for the development of *FAIR Sensor Services*. Based on the *MontiCore Language Workbench* (blue), the grammar of SOIL and the SOIL-specific validation and generation logic has been implemented by the author (green). When implementing a *FAIR Sensor Service*, the user defines the device-specific SOIL model (grey), which is parsed, validated, and translated into the base implementation of the *FAIR Sensor Service* using the *SOIL Workbench*. Illustration based on Rumpe [Rump].

The grammar and syntax for SOIL have been defined using the *MontiCore* grammar for language definition. This definition is automatically translated into the core software modules of the *SOIL Workbench* using *MontiCore*, which provides a fully implemented parser for SOIL models and templates to implement the validation of the model's semantic, the model translation and source code generation. Using the generation-gap pattern [Fowl11, p. 571ff] for software development and *MontiCore's* TOP-mechanism [Rump21, p. 96f], additional functionality has been added to the classes of the Abstract Syntax Tree (AST) and the rules for checking the semantic validity of the model has been implemented, called context conditions (CoCos). Among others, these CoCos check if all instantiated *element types* have also been defined, whether the *Component* structure is cycle-free, and if the defined prefixes and used controlled terminology exist (by querying the online resources referenced via the prefix URL). Suppose the model parsing and validation checks have been executed without errors. In that case, the model is translated into source code and/or a full semantic data model, depending on the choice of the developer. For the transformation, *MontiCore* uses *Freemarker* so that *Freemarker* templates have been created accordingly by the author.

The translation into implementation templates for the CPMS has been developed for Python and C++, covering widely used programming languages [Stac23] relevant for the implementation of CPSs. Given a SOIL model, the translation into a Python implementation template as shown in Listing 5.3 is defined as follows:

- every *Component* is translated into a Python class,
- every *Function* is translated into a method of a class,
- every *Measurement* of a *Component* is translated into an attribute of a Python class and a method for retrieving the value of this attribute, and
- every *Parameter* of a *Component* is translated into an attribute of a Python class and a getter method for retrieving the value of this attribute. If the *Parameter* is not modified to be `constant`, a method for writing the value is also generated.

```python
class COMTemperatureSensor(object):
    def __init__(self, device: Device ):
        self._device = device
        self._mea_temperature = 0

    def get_mea_temperature(self) -> Tuple[float, Any]:
        return self._mea_temperature, None

    def streams(self, fqid: str) -> List[Job]:
        schedule = []
        schedule += [AdvertisementJob(fqid)]
        schedule += [FixedJob(f"{fqid}/MEA-Temperature", 5, self.get_mea_temperature)]
        schedule += [EventJob(f"{fqid}/MEA-Temperature", 10, self.get_mea_temperature, Event(EventSeverity.WARNING, EventTrigger.SMALLER, "float", 0, "The ambient temperature is low."))]
        return schedule
```

Listing 5.3: The Python class is generated from the SOIL model in Listing 5.1. This class serves as template for implementing retrieval of the physically measured temperature data which is then provided together with the measurement uncertainty via the getter method `get_mea_temperature`.

A dedicated class exists for each stream type and events (cf. Figure 6.5), so that for each stream and event defined for a *Component* an object of the particular class is created, instantiated in the Python class of the *Component*. If a *Component* is modified to be `dynamic`, methods for creating and deleting objects of the corresponding class during runtime are generated for the composite class (of the parent *Component*).

The translation to C++ is based on the libraries developed by Montavon [Mont21] and has been implemented for the basic *Component* structure of SOIL. At the time of writing, events and streams are not automatically translated to C++ but can be implemented manually. Besides, the translation is similar to Python. Still, each method, *Measurement*, and *Parameter* is additionally encapsulated by a class, which is then instantiated as an attribute of the C++

class of the parent *Component*. The translation to other models has been prototypically developed for OPC UA and a generation of an OPC UA server in Python:

- every *Component* is translated into a component node,
- every *Function* is translated into a function node,
- every *Measurement* of a *Component* is translated into a read-only variable node, and
- every *Parameter* of a *Component* is translated into a variable node, which can be read and written. Parameters marked as `constant` can not be written.

To provide the (meta) data compliant with the semantic data model described in Chapter 4, the soil model is translated into an RDF-based metadata graph and a MP. While the first provides semantically interoperable sensor data, the latter restricts the provided data sets and can be used to parametrize algorithms processing the sensor data or validate or filter a data set. All partial MPs of the measuring system are extensions of the generic MPs defined in Chapter 4. All RDF-based metadata graphs are compliant to the specific MP of that element of the SOIL and, thus, also compliant to the generic MPS. An example of such a generated RDF-based metadata graph is shown in Listing 5.4.

```
sensorone:SensoroneTemperature
    a sosa:ObservableProperty ;
    ssn:isPropertyOf schema:Thing ;
    dcterms:title "Temperature" ;
    dcterms:description "The current temperature." ;
    dcterms:identifier sensorone:SensoroneTemperature;
    qudt:hasDimensionVector() ;
    qudt:dataType xsd:float ;
    sosa:isObservedBy sensorone:Sensorone ;
    ssn-system:hasSystemProperty sensorone:SensoroneTemperatureRange ;
    qudt:hasQuantityKind quantitykind:Temperature ;
    qudt:applicableUnit unit:DEG_C ; .

sensorone:SensoroneTemperatureRange
    schema:minValue "-20"^^xsd:float ;
    schema:maxValue "50"^^xsd:float ; .
```

Listing 5.4: The node of the RDF-based metadata graph serialized to the Turtle format representing the static metainformation of temperature measurement as defined in the SOIL model in Listing 5.1. The definitions of the required prefixes have been omitted for the sake of brevity.

The *SOIL Workbench* provides a command line interface for parsing, validating, and translating SOIL models in a textual representation according to the syntax described. A web-based editor has been implemented to lower the usage barrier even further, the *SOIL Editor* shown in Figure 5.7. The *SOIL Editor* allows to organize multiple SOIL models and provides the explained textual representation and an additional visualization using typical UI elements. The two kinds of representations are fully equivalent so that the users can switch between both representations easily. The web-based frontend is publicly hosted and can be used from any client device without needing to install any software, mainly because the *SOIL Editor* communicates with a backend server encapsulating the *SOIL Workbench*. The command line interface of the *SOIL Workbench* is wrapped by a RESTful API of the server backend. The REST endpoints for parsing, validating, and translating SOIL models created with the *SOIL Editor* can be easily queried using dedicated buttons in the fronted.

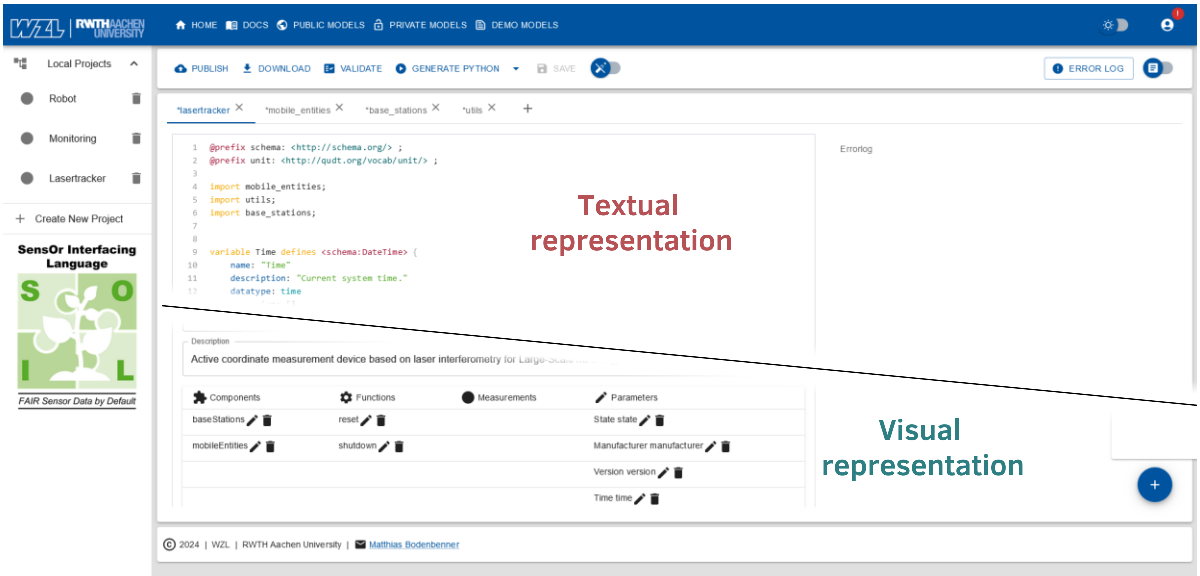


Figure 5.7: Web-based user interface of the *SOIL Editor* showing a mash-up of the textual and visual representation. The textual representation implements syntax highlighting, also used throughout this thesis to increase the readability and comprehensibility of the models. The visual representation uses common UI elements, such as text fields, drop-down lists, switches, and checkboxes, providing strong guidance to the users in creating the model. Thereby, possible sources of errors - usually detected in the validation of the textual representation - can already be prevented during the model definition using the visual representation.

5.4.3 Verification of the functional correctness of the *Sensor Interfacing Language*

Based on the source code generation features of SOIL based on MDSE, a template for implementing *FAIR Sensor Service* can be generated only. The provision of *FAIR manufacturing sensor data* requires full implementation of all required artifacts. Thus, the full evaluation and verification of the effectiveness of SOIL for providing accessible and interoperable data can only be done together with all other developed and implemented artifacts in Chapter 6. Thus, in the first step, the correctness of the semantic validation (i.e., the CoCos) and generation features of the implemented *SOIL Workbench* have been verified to give a preliminary answer to the third RGQ. The validation of the developed IDL has been conducted in two steps: (i) unit tests have been implemented to test the parsing and semantic validation of SOIL and (ii) models of three simulated measuring systems have been defined to check the correctness of the translation and source code generation.

The unit tests for the CoCos have been implemented in Java using the JUnit test-suite [JUni24]. An exemplary CoCo and JUnit tests for a CoCo are shown in Figure 5.8. At least two test cases are defined for every non-trivial CoCo to identify false positives and false negatives of the semantic validation and fix these issues in case a false positive or negative is detected. Each unit test consists of a minimal SOIL model containing only the features validated by the tested CoCo. To verify the correct validation of a well-defined model, the test case parses the model, executes the tested CoCo only, and checks if the list of caught exceptions is empty. The unit test checks the list of caught exceptions for the expected errors to verify that erroneous SOIL models are correctly validated. The unit test succeeds if and only if all expected exceptions and no additional exceptions have been caught. 176 unit tests have been defined for the CoCos, of which 53 test for correct validation of valid SOIL models and 123 test for the correct validation of invalid SOIL models. After iteratively fixing detected issues, all unit tests were executed successfully.

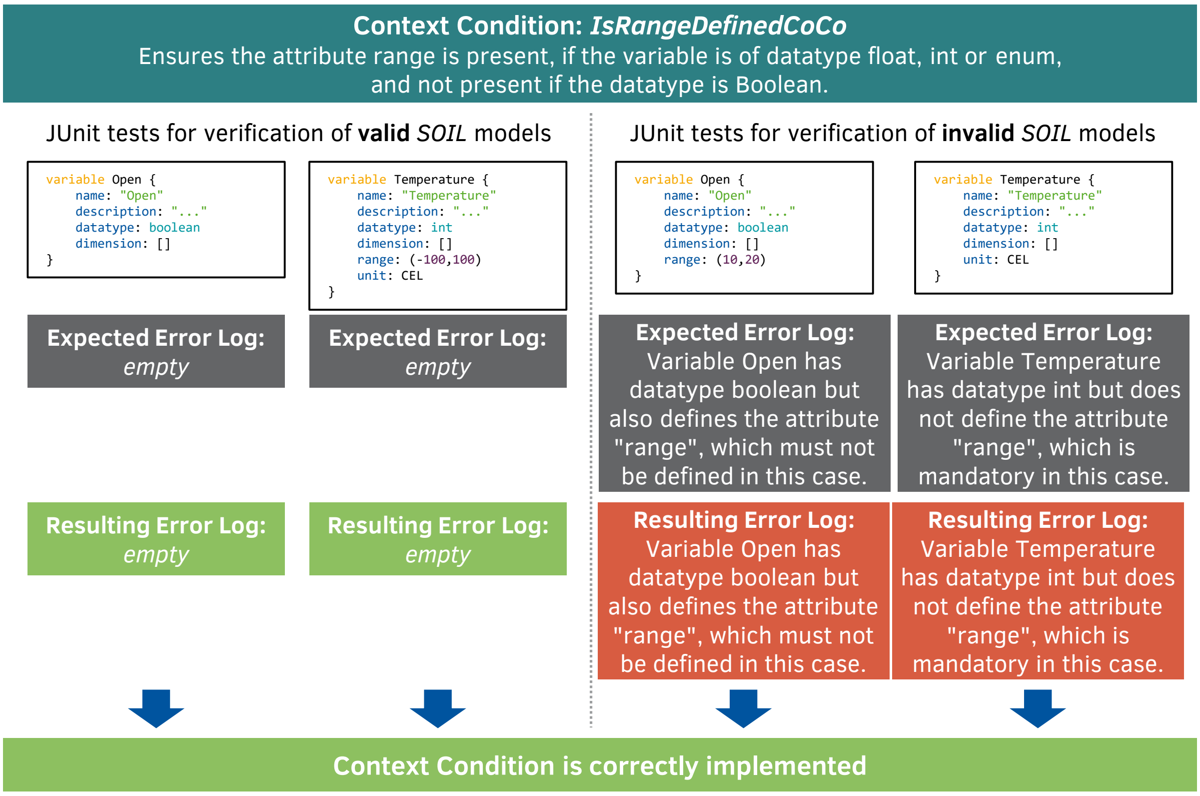


Figure 5.8: JUnit tests for the context condition (CoCo) "IsRangeDefinedCoCo" to check whether the range attribute of a *Variable* is defined depending on the datatype of this *Variable*. The CoCo specifies exceptions to be thrown in case the range is not defined, although it should and vice versa. The JUnit tests verify the correct behavior by checking if expected exceptions are thrown (or not).

The manual definition of all the SOIL models and errors to be caught for the validation required high effort but is necessary to ensure even rare corner cases are correctly handled. The manual definition of SOIL models and their corresponding translation and generation steps results require even more effort, as a SOIL model consisting of very few elements only already produces multiple source files for the Python or C++ implementation. Thus, a more pragmatic approach has been chosen. The implementation of the translation and generation is tested based on fully defined SOIL models of the three *Dummies* (see Section 4.4). These models are parsed, validated, and translated into the implementation template of the *FAIR Sensor Service* in Python, which are executable when importing the libraries developed in Chapter 6. The successful execution of the generated source code verifies the correct translation and generation.

5.5 Intermediate summary

To answer the third RGQ, the development and implementation process of a *FAIR Sensor Service* has been investigated. MDSE has been employed to streamline the process and achieve a scalable and flexible approach, enabling engineers without deep knowledge of data modeling to define semantic data models. Generating interfaces for different communication protocols and data formats significantly increases the accessibility of the provided measurement data [Derh17b]. Moreover, developers of CPMS do not need to deal with the complexity of the semantic sensor model, as they never are in direct contact with it [Muse22].

Preliminary answer to the Research Guiding Question 3

How can controlled vocabularies be reused systematically by domain-specific modeling languages to provision accessible and interoperable manufacturing sensor data?

The *Sensor Interfacing Language* (SOIL) has been defined as IDL - a specialized type of DSMLs for defining interfaces - for the development of interfaces of *FAIR Sensor Services*. SOIL supports the inclusion of controlled vocabularies by design yet abstracts the complexity of the definition of full semantic data models. It has been explained in detail how the *FAIRness* criteria specified in Chapter 3 as the answer to RGQ 1 have been considered in the development process of SOIL. Because the underlying data structure can be mapped to IIoT protocols and data formats SOIL can be used to generate implementations accordingly as explained in Section 5.4.2. Based on the theoretical evaluation in Section 5.4.3, the third RGQ can be positively confirmed with SOIL, the *SOIL Workbench*, and the *SOIL Editor* as primary artifacts of the design cycle with regard to RGQ 3 (see Figure 5.9).

Artifact 3.1 *Sensor Interfacing Language* & *Workbench*
Grammar, Parser, Validator & Generator

Artifact 3.2 *SOIL Editor*

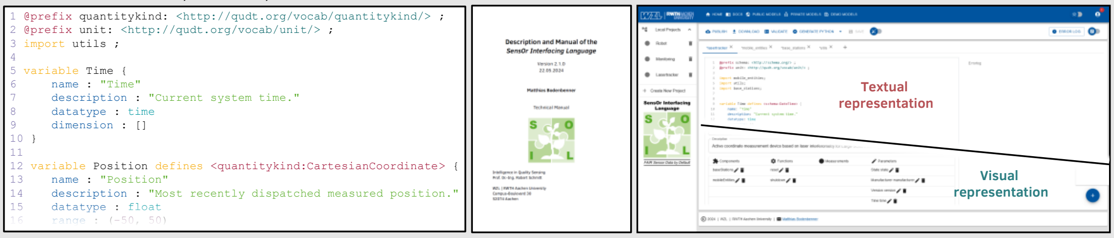


Figure 5.9: The first research artifact of the third research guiding question are the *Sensor Interfacing Language*, namely its syntax and semantics, and the implemented software tool, the *SOIL Workbench*, for validating a SOIL model and translating it into source code. The second artifact is the *SOIL Web Editor* complementing the *SOIL Workbench* with a user-friendly UI for the textual and visual definition of SOIL models.

This answer finalizes the investigation of all three RGQs. While all three have been positively confirmed based on a preliminary qualitative evaluation, the implementation of the complete framework and the verification and validation are yet to be done and described in the following Chapter 6.

6 Implementation and validation in a close-to-industry scenario

The implementation and validation of all methods developed in Chapters 3 to 5 is executed following the *V-Model* [Rook86], shown in Figure 6.1. The research on the RGQ described in the Chapters 3 to 5 form the requirements, design, and development phases of the left, down-stream side of the *V-Model*. The implementation and unit tests of SOIL in Sections 5.4.2 to 5.4.3 initialized the right, upstream side of the *V-Model* verifying and validating the correctness of all artifacts of the down-stream side. In this chapter, three industrially used measuring systems are described and implemented as *FAIR Sensor Service* to demonstrate the applicability of SOIL. The integration tests are performed by implementing the *FAIR Sensor Services* and *FAIR Sensor Ecosystem* and using the SOIL in a machine tool calibration use case. Finally, the acceptance test is conducted in Section 6.4 by qualitative and quantitative evaluations of the *FAIRness* of the data acquired with the *FAIR Sensor Services* of the three simulated measuring systems described in Section 4.4.

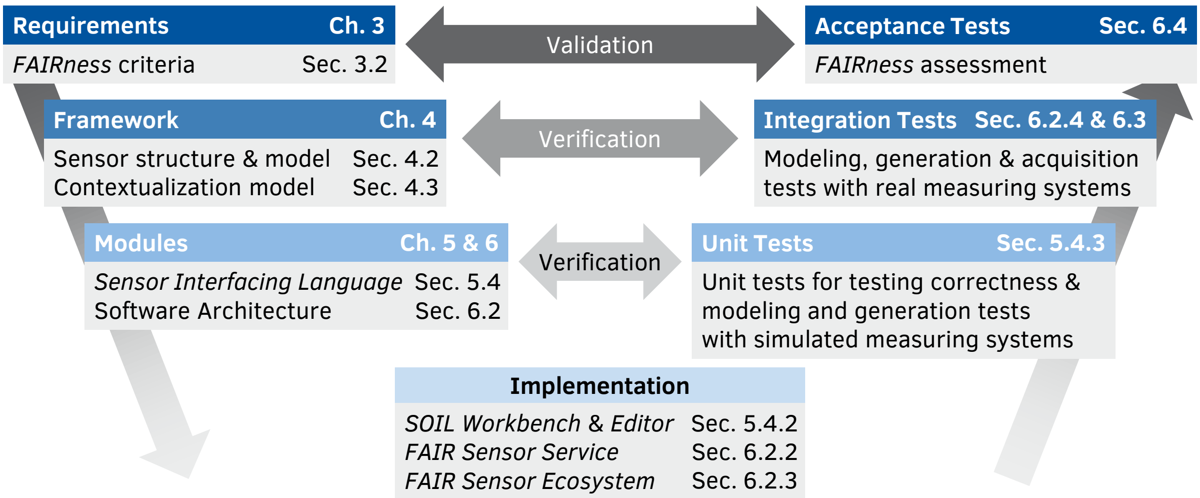


Figure 6.1: The prototypical implementation and its validation of the methods and the framework developed in this thesis is based on the *V-Model* [Rook86] consisting of three individual steps.

The author has published parts of the work described in this chapter in

Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. "Model-Driven Development of Interoperable Communication Interfaces for FAIR Sensor Services". In: *Measurement: Sensors* 24 (2022), p. 100442. ISSN: 26659174. DOI: 10.1016/j.measen.2022.100442

Matthias Bodenbenner, Jan Pennekamp, Benjamin Montavon, Klaus Wehrle, and Robert H. Schmitt. "FAIR Sensor Ecosystem: Long-Term (Re-)Usability of FAIR Sensor Data through Contextualization". In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. July 2023, pp. 1–8. DOI: 10.1109/INDIN51400.2023.10218149

Matthias Bodenbenner, Dominik Wolfschläger, and Robert H. Schmitt. *Providing FAIR Sensor Data Models Using Semantic Web Technologies and Ontologies*. Apr. 15, 2024. DOI: 10.13140/RG.2.2.27882.73928. Pre-published

6.1 Description of the close-to-industry validation scenario

The suitability and effectiveness of the developed data models, SOIL, and the implemented software tools are validated in a close-to-industry scenario. The laboratory of WZL possesses state-of-the-art and industrially used LSM equipment that is very well suited to validate the developed artifacts. Three-dimensional measuring systems for acquiring cartesian coordinates have been implemented as *FAIR Sensor Services* using SOIL and the implemented libraries: (i) an API Radian™ laser tracker, (ii) a Leitz PMM-C Precision™ coordinate-measuring machine, encapsulated by an I++ DME interface, and (iii) an OptiTrack™ photogrammetry-based motion capture system.The combination of these three measuring systems covers the whole range of measurement accuracy and range of LSM-systems (cf. Figure 6.2). The definition of a SOIL model and the development of *FAIR Sensor Services* for these three measuring systems will reveal the applicability of the overall approach.

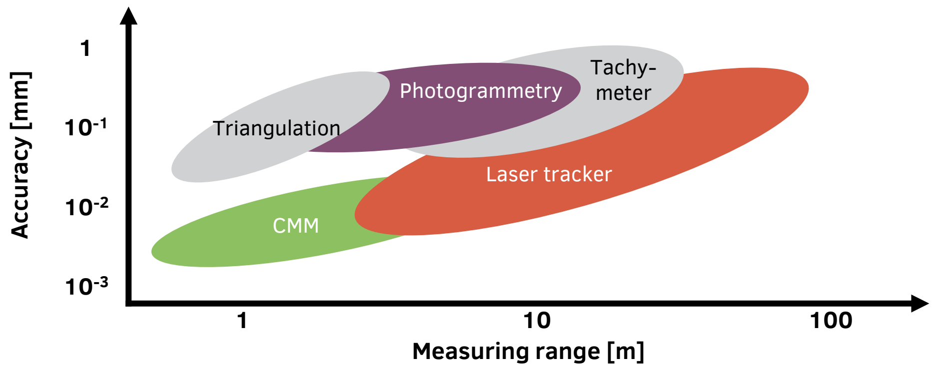


Figure 6.2: Overview over the measuring range and accuracy of LSM measuring systems adopted from Cuypers et al. [Cuyp09]. The three systems used to validate this thesis cover the spectrum in both dimensions, measuring range and accuracy.

API Radian™ laser tracker

A *laser tracker* is a device used to measure the position of special targets, so-called Spherically Mounted Retroreflectors (SMRs). Generally, laser trackers have outstanding measurement precision and a high working volume of up to 100 metres [Cuyp09]. The measurement uncertainty is usually smaller than 0.5 mm in the complete working volume. But, it is not possible to track two or more points simultaneously with one tracker [Wang11]. Typical fields of application for laser trackers are robot calibration and the assisted assembly of large products like airplanes, ships, or cars. The WZL possesses two API Radian™ 3, having 20 and 40 meters measuring radius, respectively, and one Hexagon AT960 with 40 meters measuring radius. To validate the developed tools, a *FAIR Sensor Service* of an API Radian™ 3 is implemented by extending the software artifacts provided by Montavon [Mont21]. The simulated *FAIR Sensor Service* of a laser tracker (cf. Section 4.4) implemented in Python (cf. Appendix D) is based on the implementation of an API Radian™ 3 in C++ by Montavon [Mont21]. Key features of SOIL stressed by the implementation of the laser tracker are dynamic components, `update` streams of measurements, and events in case the base station loses line of sight to the mobile target.

OptiTrack™ motion capture system

In recent years, motion capture systems (MCSs) have gained notable attraction for determining and tracking the position, orientation, and movements of assets or humans. Typical use cases are animations in movie production or game development and the tracking of drones or mobile robots. For tracking objects like mobile robots, multiple markers are attached to the robot and then grouped into a rigid body so that the position and orientation of the robot can be determined with high reliability. The laboratory of WZL is equipped with an OptiTrack™ MCS consisting of 38 Prime^{x}41 cameras covering a volume of ca. 25×15×8 m (L×W×H) for tracking mobile robots. The OptiTrack™ MCS is implemented as *FAIR Sensor Service*. The high data rate of the system and the wide variety of measurements and parameters of the system are well suited to test the modularity and flexibility of modeling using SOIL. Rigid bodies might be added and removed dynamically at runtime if the system regularly loses and restores the line of sight to the ridig bodies.

Leitz PMM-C Precision™ coordinate-measuring machine

Coordinate-measuring machines (CMMs) are standard equipment for the quality inspection of manufactured parts with very low tolerances in the micro- and nano-meter range. In contrast to the laser tracker and MCS, a CMM runs part-specific measuring routines, which require knowledge of the nominal shape of the part and individual configuration and parametrization to determine deviations from the nominal shape. Environmental conditions such as temperature, humidity, and pressure must be quasi-static and stable to achieve the high accuracy required. The WZL has an industry-grade K1 measurement room with a Leitz PMM-C Precision™. The device is equipped with an I++ DME interface. The command-based interface has been realized as SOIL model that makes use of functions with varying arguments and returns by encapsulating the I++ DME interface of the machine by a *FAIR Sensor Service* based on SOIL.

6.2 Implementation of the software artifacts and verification of their functionality

This section describes the implementation of all the software required to provide the *FAIR Sensor Service* of the three presented measuring systems and the *FAIR Sensor Ecosystem*. Moreover, the integration tests are executed as a combination of (i) the definition of the SOIL models of the three simulated and three real measuring systems, translation into full semantic data models, and implementation of their hardware layers, (ii) the development of the *FAIR Sensor Ecosystem* as the realization of the contextualization model, (iii) the data acquisition using the *FAIR Sensor Services* and ingestion into the *FAIR Sensor Ecosystem*. The major implemented software artifacts are depicted in Figure 6.3. The software libraries providing the communication interface of the *FAIR Sensor Service* via HTTP, MQTT, and OPC UA have been developed. Based on these and the SOIL modeling framework for the model definition and software generation presented in Section 5.4.2, the six *FAIR Sensor Services* have been implemented.

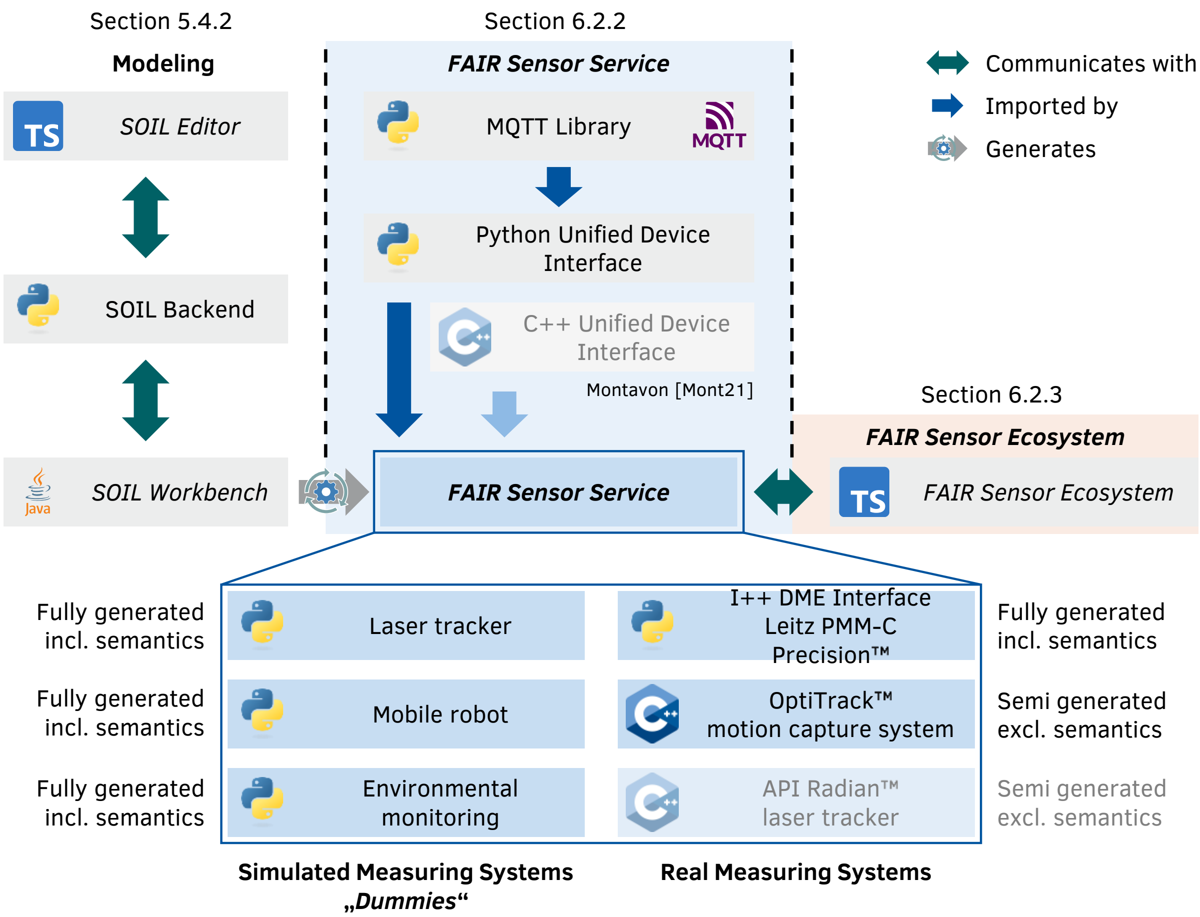


Figure 6.3: The major software artifacts developed during research and their relations. Evaluation scripts and discarded intermediate software are excluded. The C++ Unified Device Interface implemented by Montavon [Mont21] has been reused, and the interface of the API Radian™ laser tracker by Montavon [Mont21] has been re-generated for validation purposes.

6.2.1 System and service architecture

The foundation of implementing all software tools is the architecture of the *FAIR Sensor Infrastructure* mirroring the *FAIR* DLC depicted in Figure 6.4. Each step of the *FAIR* DLC is realized by a dedicated layer of the *FAIR Sensor Infrastructure*, such that the step-specific issues and tasks can be tackled individually following the *separation-of-concerns* principle [Hürs95]. The data produced by the *FAIR Sensor Service* is transmitted to the *FAIR Sensor Ecosystem* via a message broker and might be directly used by short-term applications, e.g., process control or quality inspections, prior to validation and storage. Long-term applications such as recyclability assessments, job scheduling, or global process optimization access the data persisted in the *FAIR Sensor Ecosystem* via an API. The architecture of the infrastructure for *FAIR manufacturing sensor data* can be mapped to the Reference Architecture Model Industry 4.0 (RAMI 4.0) [Adol15] and covers almost all layers and hierarchy levels of the production step of the life-cycle and value stream as shown in Figure G.1.

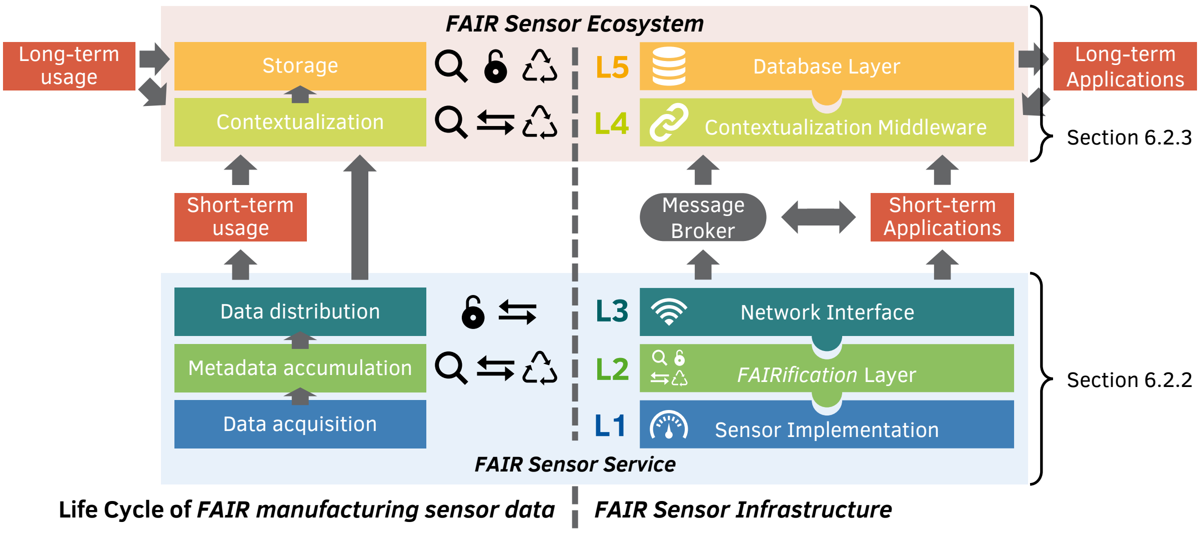


Figure 6.4: The system architecture of the *FAIR Sensor Infrastructure* on the right mirrors the *FAIR* DLC (cf. Figure 3.4) on the left and shows the interplay between the *FAIR Sensor Services* and the *FAIR Sensor Ecosystem*.

6.2.2 Software libraries of the *FAIR Sensor Service*

Following the approach of generating generalizable components of the *FAIR Sensor Service*, the communication layer is implemented as a software library, which is imported into an implemented service. For Python, the author implemented the Python Unified Device Interface (PUDI) [Bode24b]. The central component of the library is a RESTful HTTP server, which provides a resource for every *Element* of a SOIL model. Given the implementation of a measuring system based on the templates and class structure generated from a SOIL model via the *SOIL Workbench*, the resources of the REST-API are extracted from the serialized model description and inflated by a mapping of each endpoint to a dedicated method of the classes of the sensor implementation. Figure 6.5 shows the structure of the PUDI library. The advantage of this design and implementation is the separation between the HTTP server and MQTT-Publisher and the underlying internal system logic. As long as the internal sensor logic is based on a SOIL model, the PUDI library can be used without restrictions. Furthermore, by outsourcing the MQTT publisher to an extra Python library, the protocol and its implementation for publishing can be changed without adapting the remaining components of the *FAIR Sensor Service*. Moreover, by including several different communication protocols in PUDI **C4 System heterogeneity** can be solved effectively.

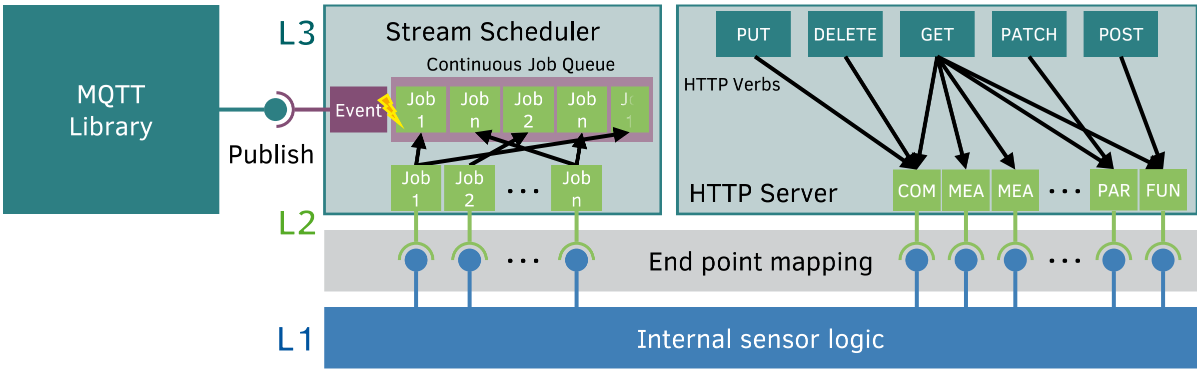


Figure 6.5: The software architecture of the Python Unified Device Interface [Bode22a]. The software framework implements the communication interface and the *FAIRification* layer of the *FAIR Sensor Service*. Due to the standardized endpoint mapping, the library offers communication protocols, which can be easily replaced. Whereas the Stream Scheduler and HTTP Server are central components of the library, the internal sensor logic is implemented by the sensor expert based on the implementation templates generated with the *SOIL Workbench*. The MQTT library is is implemented separately.

Using PUDI the *FAIR Sensor Service* can provide the system's (meta-)data represented as syntactic data structure according to the meta structure developed in Section 4.2.1 and as semantic data model according the the metamodel developed in Section 4.2.3. By default, the endpoint of each *Element* returns the data structure serialized to JSON. All responses of the *FAIR Sensor Service* can be parametrized using *query strings* [Bern05] to configure the representation and data format of the system's (meta-)data. Using `format=xml`, the data is returned after serialization to XML. Moreover, the semantic data model can be requested via `semantic=metadata` and the corresponding MP using `semantic=profile`. The two query parameters can be combined to retrieve the RDF node graph of the semantic representation in the desired format, either JSON-LD, XML or Turtle. These endpoints provide the first part of *FAIRification* layer. The second part is implemented by adding additional endpoints to resolve the identifiers of the nodes of the semantic data model. The resolution depends on the system-specific MPs and static metadata set of the device. Due to security reasons, only the simulated measuring systems have been deployed on central servers and are openly accessible. By registering a namespace for each simulated measuring system individually at `https://purl.archive.org/` the globally unique and persistent identifiers of the endpoints to the semantic (meta-)data models can be used to query the semantic data. The endpoints are redundant to the default endpoints parametrized either with `semantic=metadata` or `semantic=profile`, if the metadata set or the MP is requested, respectively.

Based on these libraries the simulated measuring systems and the I++ DME server of the PMM-C Precision™ have been implemented as *FAIR Sensor Service*. The software of all three simulated measuring systems is openly accessible at Zenodo [Bode24c]. The source code of the *FAIR Sensor Service* of the PMM-C Precision™ interacts with a closed-source reference implementation of I++ DME and can therefore not be published. Nevertheless, the SOIL model based on I++ DME is listed in Appendix E.

6.2.3 Implementation of the *FAIR Sensor Ecosystem*

The implementation of the *FAIR Sensor Ecosystem* is completely based on open-source technologies and libraries. The system is implemented in TypeScript and translated into JavaScript so NodeJS libraries can be used without restrictions. As shown in Figure 6.6, the architecture consists of four software components.

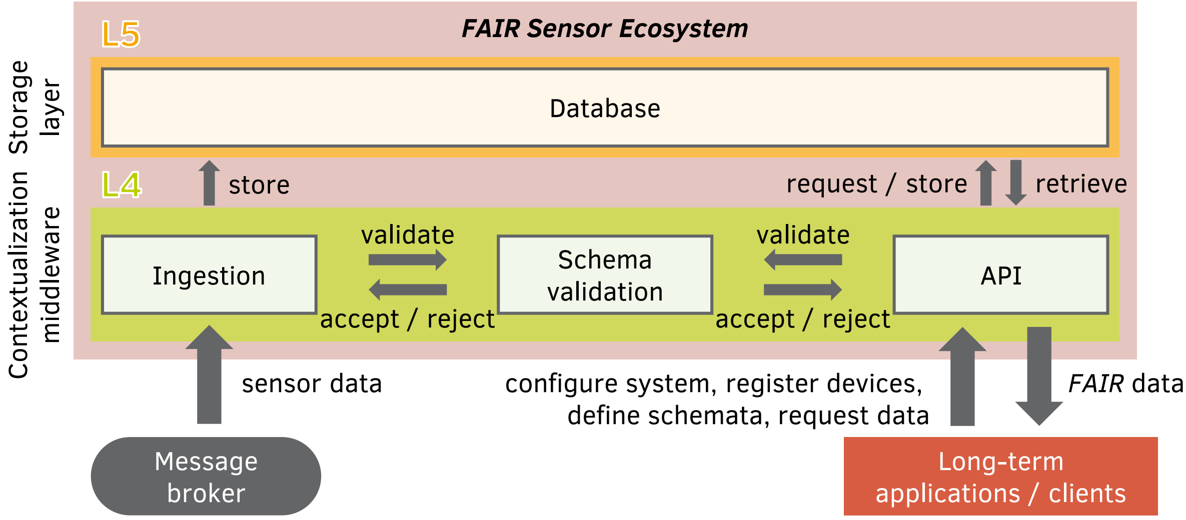


Figure 6.6: Architecture of the *FAIR Sensor Ecosystem* for processing and storing measurement data according to the contextualization model defined in Section 4.3. While the assets and their relations are manually created via the API, measurement data is automatically received from a message broker. If the incoming data packages are compliant with schemata of registered assets, the data is persisted in the database.

According to criterion **CI.4 Consistency**, data ingested into the *FAIR Sensor Ecosystem* must be validated against defined schemata or metadata profiles. Storage in the persistent database includes updates of links and references between related resources. The ecosystem provides endpoints for creating and maintaining the relationships between modeled assets, data, and metadata schemata via an API. The API provides endpoints for creating virtual assets, representing real-life assets such as machines or sensors, relations between them, and metainformation about the assets according to the contextualization model presented in Section 4.3. The relations and metainformation can further be updated via the API according to **CA.5 Versioning** and **CI.5 Linked Data**. Furthermore, schemata for data and metadata, such as provided by the *FAIR Sensor Services*, can be registered as required by **CI.4 Consistency**, which are used to validate incoming data and metadata via the *Schema Validation Service*. To fulfill **CA.5 Versioning**, the API does not offer an endpoint to delete data. The API is realized via a REST API using the HTTP protocol, enriched by the HTTP Memento protocol [vdSomp09] to provide the time-based filtering technique. Measurement data is exchanged in JSON format and sent via the MQTT protocol. JSON is fully compatible with JSON-LD used to define and describe the context of the data and the type schemata.

Measurement data sent from *FAIR Sensor Services* via a publish/subscribe protocol is handled by the *Ingestion Service*. According to the stored contextualization model, specified topics are subscribed by the *Ingestion Service*, which thereby receives corresponding data. The data is checked for the validity of its type against the MPs for *FAIR manufacturing sensor data* defined in Section 4.2.3 via the *Schema Validation Service* and rejected if the type is invalid. Otherwise, the data is stored in the *Database* (following **F4**) and linked to the currently valid metainformation of the assets that produced the data.

To achieve scalability, the ingestion service has been split into three subcomponents: ingress, buffering, and ingestion. By that, incoming measurement data can be validated, even if the amount of data is very high. For that, RabittMQ [Broa24] has been used, as the quality of service feature of RabbitMQ ensures that no data buffered is lost. This way, potential bottlenecks in the validation of incoming data and writing accepted data to the database can be mitigated. The influence of the latter should have been further reduced by using PostgreSQL with the TimescaleDB extension, which splits the table into smaller chunks, so-called *hypertables*. However, hypertables do not support foreign keys [Vujc18], which conflicts with the contextualization models that require a foreign key from `Measurement` to the other data points. Thus, the TimeScaleDB extension has not been used, which theoretically reduces the insertion rate of the database if the table contains large amounts of data [Booz24]. The data model is implemented using TypeORM[1], which integrates well with the other tools used and reduces implementation and maintenance effort due to its easy-to-use API. To deal with multiple unrelated object trees in the database, a hidden root node is introduced, which can not be queried but serves as the parent node for all assets that do not have a parent.

According to **CF.2 Searchability**, searching and filtering the database by any field in the data model and the stored metadata must be possible. This allows users to retrieve the data by setting the context in which the data was created. This context can primarily be described by specifying an interval in the past during which the data was produced, the types of the data itself, and the related metadata and values of attributes of related assets. So, the following filtering techniques capable of utilizing a multitude of filter arguments are introduced:

Date Filtering To retrieve data and metadata of a certain interval, two arguments can be supplied. If the argument `from` is used, only data that is younger than the specified date is returned. If the argument `to` is given, only data that is older than the specified date is returned.

Associated Type Filtering This technique allows filtering data by a given metadata or value type. Furthermore, it allows retrieving all measurement values that are associated with specific component information or produced by a specific component.

Custom Filter Expressions The ecosystem has to support filtering by any arbitrary attribute or field of the model and of the stored metadata. Thus, it must enable users to write custom filtering expressions, which address the a priori unknown properties of the metadata stored. Otherwise, these fields are neither filterable nor searchable.

All data in the database is stored in JSONB. Thus Structured Query Language (SQL) operators and functions, both defined in ISO/IEC 19075-6:2021 [ISO21], are utilized to implement the *Associated Type Filtering* and *Custom Filter Expressions*. Non-functional but essential system features regarding the *FAIR Guiding Principles* are the usage of standardized, open communication protocols and data formats, as required by guideline **A1.1**. In addition, the metainformation must use publicly defined terms following guideline **I2**.

[1] `https://typeorm.io/`

6.2.4 Integration test

For the integration test of the *FAIR Sensor Services* and the *FAIR Sensor Ecosystem*, the implemented prototype has been integrated into the IIoT infrastructure of the chair (see Figure 6.7). The measurements produced by measuring systems are represented and distributed according to the syntactic data structure developed in Section 4.2.1. For ingestion in the implemented *FAIR Sensor Ecosystem*, the definitions of measurements according to SOIL are mapped to `TypeDefinitions` of the contextualization models. The test setup consisted of the API Radian™ 3 laser tracker producing high-frequency position data of multiple targets measured and attached to several machines or parts. By that, the setup and configuration were modified between different measurement runs. Additionally, environmental data collected by multiple different sensors have been included in the test. To approve the system design and implementation concerning the *FAIRness* criteria defined in Section 3.2.2, four steps have been taken: (i) ingestion of valid real data, (ii) ingestion of real data modified to be invalid, (iii) retrieval of data using the custom filter functions, (iv) reconstruction of the setup and configuration at a given point in time. The technical validation was carried out successfully and showed that the concept and the implementation worked as desired. The high-frequency data could be ingested without data loss, and invalid data was rejected without exceptions. Moreover, all different setups and configurations could be fully reconstructed. At the time of writing, the *Schema Validation Service* has only been implemented for checking the conformance with data sent in the JSON format and represented according to the data structure of Section 4.2.1 based on JSON schemata. Nevertheless, the validation of fully semantic data using MPSs has been tested separately and does also work [Bode24d] [Bode24f]. From that, it can be concluded that a technical integration will yield the desired functionality.

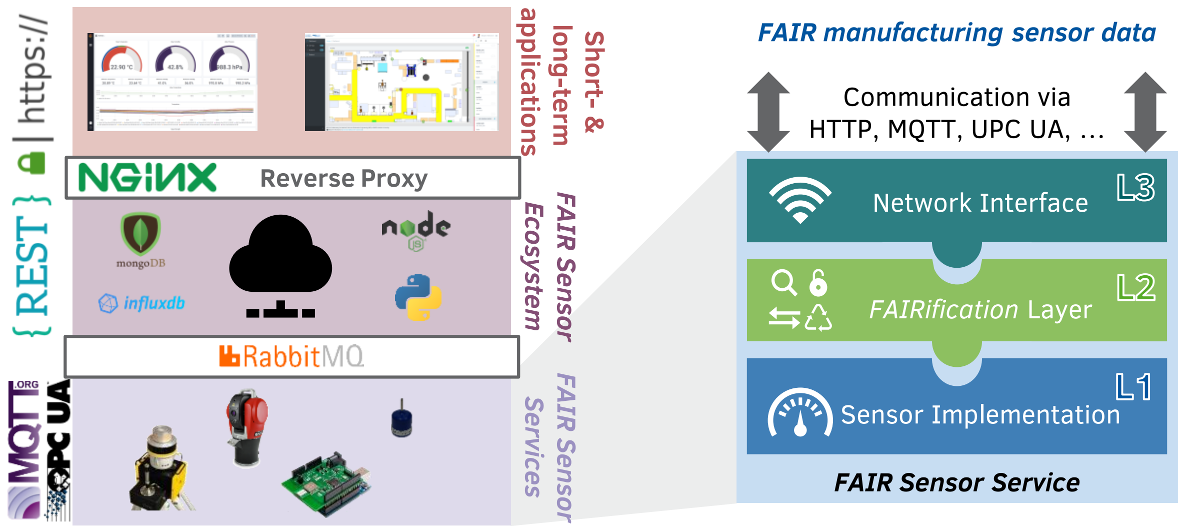


Figure 6.7: Conceptual depiction of the IIoT infrastructure at WZL in which the *FAIR Sensor Services* and the *FAIR Sensor Ecosystem* have been deployed for the integration tests. Figure adapted from Montavon and Bodenbenner [Mont23].

In a separate integration test, the compliance of the semantic data provided by the *FAIR Sensor Services* with the MPs of the semantic sensor data model has been verified. The correct generation of the MPs and full semantic data models based on device-specific SOIL models is verified via SHACL validation. Compliance with the MPs of the generated metadata has been validated by retrieving each measuring system's complete metadata set, requesting all endpoints of the *FAIR Sensor Service*, and merging the retrieved files into a complete graph. Then, the full MP defined is retrieved from the web at `https://purl.org/soil/` to mirror an actual application of the MP. Based on the MP, the metadata graphs of all three systems downloaded before are validated using the pySHACL [Somm23] library. All metadata were validated successfully. The source code for validation and the results are openly accessible [Bode24c; Bode24d].

Consequently, the integration test verifies the correct functionality of the envisaged behavior and features of the *FAIR Sensor Service*, *FAIR Sensor Ecosystem*, and the overall infrastructure. Although the validation via the MPs has been conducted separately, integration into the *FAIR Sensor Ecosystem* can be expected to be successful based on the achieved results. In the second phase, the integration tests are concluded by applying SOIL and the *FAIR Sensor Service* in another active research activity described in the next section.

6.3 Validation of applicability and demonstration in an industrial use-case

The data structure and SOIL form the base for a data infrastructure at the WZL, which streamlines the data acquisition, (pre-)processing, storage, and visualization based on the standardized and interoperable data description and transmission. The infrastructure depicted in Figure 6.8 implements three individual *FAIR Sensor Services* and a simplified version of the *FAIR Sensor Ecosystem* based on a time-series database. The contextualization is implemented by the *MQTT2Influx-Bridge* [Sand23a] that does not provide full contextualization but validates the compliance of incoming messages with the syntactic data structure (cf. Section 4.2.1) and stores all valid messages in an automatically configured InfluxDB. All components of the *FAIR Sensor Ecosystem* and the web-based dashboards for data visualization have been containerized, deployed, and accessibly hosted on servers at WZL.

Colleagues of the author are using this infrastructure to speed up the configuration and deployment of experiments. One example is the validation of the calibration research and *Virtual Climatization* [Emon21] of machine tools to compensate for thermally induced errors of machine tools without active cooling. For that, the condition of the machine tool must be monitored closely using a system of distributed temperature sensors, which are used to predict the deviation of the machine tool's structure depending on the current temperatures. The training and validation of this prediction and compensation rely on heterogeneous data, such as the nominal position of the tool center point provided by the machine tool and its actual position measured with a laser tracker. In addition, frequent adaptions of the experimental setup, shown in Figure 6.9, are required [Sand23b]. These activities profit from the standardized data representation and automated data documentation as data can also be easily (re-)used later by other researchers than the original experimentalist. Moreover, the streamlined infrastructure and software architectures ease the development of data processing and evaluation tools for machine tool calibration and thermal compensation. The implemented libraries and infrastructure have already been reused by other colleagues and in the work for Bachelor and Master theses, which significantly simplified the implementation and deployment effort of new measuring systems.

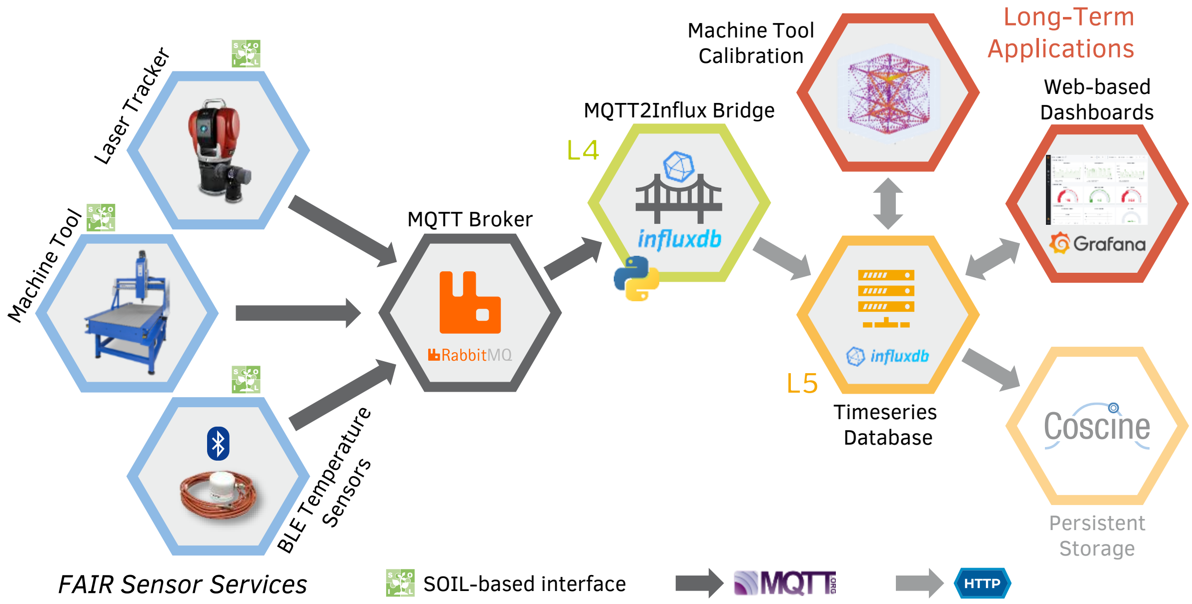


Figure 6.8: The data infrastructure for the streamlined acquisition, (pre-)processing, and visualization of the experimental measurement data. All data sources have SOIL based interfaces and publish the acquired data via MQTT. A small middleware implemented in Python, the *MQTT2Influx-Bridge* [Sand23a], validates the data packages and stores it in automatically configured time series databases. A connection to *Coscine* [Lang24] for persistent data archiving is planned.

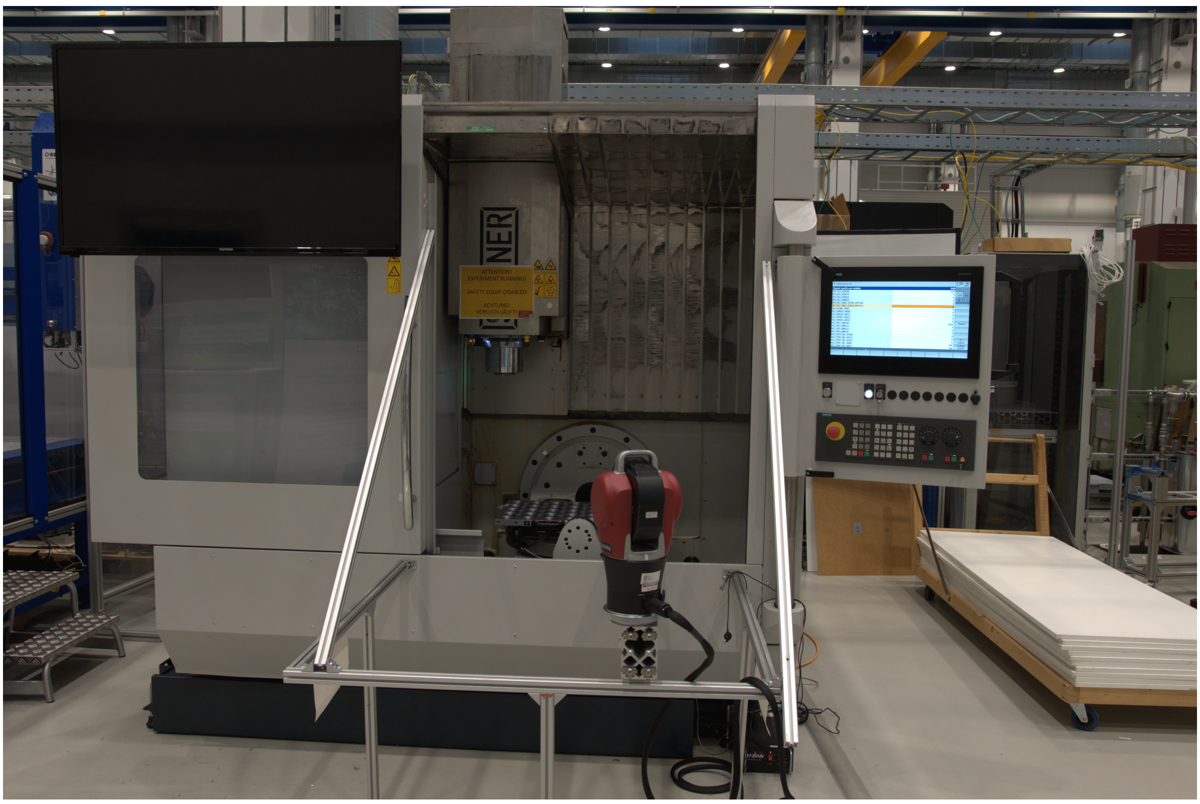


Figure 6.9: The experimental setup for validation of the *Virtual Climatization* of machine tools consisting of a machine tool, a laser tracker, and multiple hidden temperature sensors. All systems are equipped with a SOIL compliant interface and provide the data via MQTT (cf. Figure 6.8).

6.4 Evaluation of the *FAIRness* of the acquired data

The acceptance test of the developed methods is done by assessing the *FAIRness* of the manufacturing sensor data acquired by the implemented simulated *FAIR Sensor Service*, described in Section 5.4.3 and Section 6.2 respectively. The acceptance test is executed in two steps. First, a qualitative evaluation is done by a manual check and justification, if the developed methods and implemented artifacts fulfill the *FAIRness* criteria specified in Chapter 3. Second, the *FAIRness* of the acquired manufacturing sensor data is determined using *FAIRness* assessment tools and metrics. The assessments are carried out with automated assessment tools only to achieve an objective evaluation.

6.4.1 Qualitative evaluation of the compliance with the *FAIRness* criteria

The qualitative evaluation of the compliance of the implemented software artifacts with the defined *FAIRness* criteria is required to reflect if the methods and software are adequately developed. Moreover, it will reveal if the *FAIRness* criteria are implementable. The evaluation is based on an ordinal scale with five levels visualized by *Harvey balls*. The individual levels are defined as follows:

- ○ The criterion is not considered at all. The described methods and developed tools do not cover the requirements formulated by this criterion.
- ◔ The criterion is only theoretically considered or partially realized in the author's data models and software artifacts but not implemented. Moreover, additional research must be conducted to develop methods required for full implementation.
- ◑ The criterion is only theoretically considered in the author's data models and software artifacts but not implemented. But in contrast to ◔, all required methods exist already, i.e., the implementation can be done using pre-existing libraries or tools.
- ◕ The criterion is fully considered, but there are minor deficits either in the validation or implementation or aspects that are out of the author's control. For example, if the complete fulfillment of the criterion also depends on developers' decisions of device-specific SOIL models, that can not be reliably influenced by the tools implemented by the author.
- ● The criterion is fully considered and implemented in the models and software artifacts. The author's solution sufficiently implements the criterion completely.

The described scale is not quantifiable, i.e., a *Harvey ball* that is three-quarters full does not mean that the criterion is 75% fulfilled or implemented. Table 6.1 contains the qualitative evaluation by the author according to the described procedure. The rating for each criterion is briefly justified. According to the author's evaluation, only two criteria have more significant deficits in the implementation. The implementation of **CA.4 Authorization and authentication** can be solved using ready-to-use features of the adopted protocols. Implementing this does not yield novel scientific insights and has therefore been excluded from the scope of

Table 6.1: The qualitative evaluation of whether the developed models and tools (cf. Chapters 4 to 6) sufficiently address the defined *FAIRness* criteria defined in Section 3.2.2.

	Criteria		Justification
Findability	CF.1	●	The proposed metamodels represent the manufacturing sensor (meta-)data in a standardized way. The developed solutions do not restrict which controlled terms can be used, how many elements can be defined in a SOIL model, and how many components can be related in the contextualization model.
	CF.2	◕	The *FAIR Sensor Ecosystem* allows to filter the stored data by all fields of the contextualization model. The searchability could be improved by implementing a graph database instead of a relational one and using the device-specific MPs as search filters.
	CF.3	●	The unique identification of the persisted *FAIR manufacturing sensor data* is decoupled from the content, configurations, and relations and thus fulfilled.
Accessibility	CA.1	●	All implemented artifacts are based on standardized and open communication protocols already used in the manufacturing domain.
	CA.2	●	Based on the protocol-agnostic data models and SOIL, the generation of multiple different communication interfaces (HTTP, MQTT, OPC UA) has been demonstrated.
	CA.3	●	The definition of the data models and IDL is independent of the communication protocol, which can be freely chosen while maintaining the same data description.
	CA.4	◑	Authentication and authorization procedures have not been implemented within the research for the thesis. However, the used protocols either directly allow the inclusion of such or specific implementations of the protocols, which allow defining authentication and authorization mechanisms.
	CA.5	●	While the *FAIR Sensor Service* does not implement versioning within the CPMS and keeps the most recent configuration only, the *FAIR Sensor Ecosystem* implements the versioning of all ingested data and never deletes any of the data, which also includes the configuration of the deployed *FAIR Sensor Service*.
Interoperability	CI.1	●	The full semantic sensor data model realized as MPs and the contextualization model are based completely on controlled terminology. Controlled terminology is technically enforced by validating all ingested measurement data for compliance with the MPs.
	CI.2	●	The implemented *FAIR Sensor Service* and *FAIR Sensor Ecosystem* provide data in open and standardized data formats, namely JSON, XML, and Turtle.
	CI.3	●	The data models defined in Chapter 4 provide the required common and accessible information model. The compatibility to industrially used models has been validated for OPC UA.
	CI.4	◕	The modeling consistency is facilitated by the modularity and reusability features of SOIL and the usage of controlled terminology. However, it can never be fully ensured that two independent developers will always define consistent models.
	CI.5	◕	The semantic sensor model enables linking the manufacturing sensor data to related assets via the contextualization model. A full validation of the seamless interplay of both models and implemented systems is yet to be done.
Reusability	CF.1	◔	All controlled terms used within the defined metamodels and structures are domain-agnostic or originate from the manufacturing domain. However, the suitability of controlled terms used in device-specific SOIL models can not be enforced. Moreover, the manufacturing domain largely lacks machine-actionably defined terminology.
	CR.2	●	The contextualization model ensures the measurement data can be linked to all related entities of the application's context.
	CR.3	◕	The measurement uncertainty is a mandatory attribute of all acquired data. In case it can not be quantified and provided by the measuring system, it is explicitly stated. The physical unit and quantity of the measurement are mandatory attributes.
	CR.4	●	All data and metadata provided by a *FAIR Sensor Service* includes the machine-actionable specification of the usage license.

the research (cf. Section 1.4). Using appropriate, domain-specific controlled terminology as required by criterion **CR.1 Domain-Specific Terms** can not be guaranteed based on the methods and tools developed in this thesis. The data models and SOIL facilitate the usage of these controlled terms, but enforcing proper usage lacks available controlled terminology and methods to automatically and objectively assess the suitability of a specific term. Thus, to fulfill criterion **CR.1 Domain-Specific Terms**, additional scientific work is required, which is briefly anticipated in Section 7.3. Nevertheless, 15 of the 17 criteria are almost fully addressed by the developed methods and implemented tools, and eleven were rated with the highest score. Thus, the artifacts created in the Chapters 4 to 6 almost entirely and therefore satisfactorily address and fulfill the *FAIRness* criteria defined in Chapter 3.

6.4.2 Quantitative evaluation using automated *FAIRness* assessment tools

The assessment of the *FAIRness* of the manufacturing sensor data acquired with the implemented *FAIR Sensor Services* is based on quantifiable metrics. This enables an objective, automatically determinable *FAIRness* score so that a potential bias of a manual evaluation can be avoided. Besides the *FAIRness* of the data itself, the quantitative assessment also yields an estimation if the *FAIRness* criteria defined in Chapter 3 are appropriate and sufficient. If the *FAIRness* assessments reveal good results and high scores, the *FAIRness* criteria are well-defined. Suppose the *FAIRness* assessments reveal bad results and low scores. In that case, it must be analyzed whether the gaps between the implementation and the criteria (cf. Section 6.4.1) or deficits of the criteria itself are the cause of the low scores.

FAIRness assessment tools

The evaluation of the *FAIRness* of the (meta-)data and the MPs provided by the *FAIR Sensor Services* is executed using available *FAIRness* assessment tools. Based on the review of Sun et al. [Sun22] and Peters-von Gehlen et al. [Pete22], six *FAIRness* assessment tools have been considered for the evaluation. The functionality, advantages, and disadvantages of each tool have been reviewed based on available literature and summarized below. Although all developers state their tool as a work in progress, the *FAIRness* metrics and procedure provided so far will yield a first assessment of the actual *FAIRness* of the data acquired and produced with the *FAIR Sensor Services*. The metrics implemented in the tools differ regarding coverage of the *FAIR Guiding Principles* and the scale on which the individual metrics are quantified.

***F-UJI* [Deva21]** is based on the FAIRsFAIR Metrics [Deva22] described in Section 3.2.1. The automated *FAIRness* assessment with *F-UJI* can either be executed using the publicly hosted web tool[2] and API or with a self-hosted version based on the source code provided as Docker container [Deva23]. By providing a valid persistent identifier, such as a PURL or Digital Object Identifier (DOI), *F-UJI* assesses the *FAIRness* of the digital object identified. It queries the digital object but also harvests external resources such as generic and domain-specific repositories for the identifier [Pete22].

[2] `https://www.f-uji.net/index.php?action=test`

***FAIR Checker* [Gaig23]** assesses the *FAIRness* of digital objects based on the *FAIR* Metrics Framework defined by Wilkinson et al. [Wilk19b]. The tool offers a user-friendly web-based UI and an API for automating the *FAIRness* assessments. The results are returned in the JSON-LD format.

***FAIR Evaluation Services* [Wilk19b]** are developed by Wilkinson et al. and provided as web service[3]. For automatically assessing the *FAIRness* of a digital object, its persistent identifier, a title for the evaluation, and the unique identifier of the evaluator must be provided. This ensures that the results are openly retrievable to increase transparency. The *FAIR Evaluation Services* provides multiple metrics and frameworks for assessing the *FAIRness*. Users can also add their own metrics and criteria. The developers of the *FAIR Evaluation Services* integrated the metrics of the *FAIR* Metrics Framework [Wilk19a] into the web service. All metrics are evaluated on a binary yes/no scale.

***FAIRshake* [Clar19]** is a web tool for semi-automated *FAIRness* assessment. It follows the same approach as the *FAIR Evaluation Services*. Users can add their own *FAIRness* criteria and metrics, and the results of an assessment are stored and openly accessible. The upload of *FAIRness* metrics created by the community enables the development and usage of domain-specific metrics. The assessment of the *FAIRness* of data is executed based on their persistent identifiers organized in projects to group related data sets. The developers themself provided the FAIRshake Metrics for assessing the *FAIRness* of datasets, repositories, and tools (cf. Section 3.2.1). The dataset rubric[4] ranks the fulfillment of a metric on a four-step scale (0%, 25%, 75%, 100%).

***FAIR enough* [Emon24]** works similar to the *FAIR Evaluation Services* and *FAIRshake*. Metrics and test suites can be registered, and the *FAIRness* of digital objects can be assessed. The results are automatically shared with other users. However, only four predefined collections of several metrics can be used to evaluate the *FAIRness* of a digital resource. All collections are directly based on the *FAIR* Metrics Framework by Wilkinson et al. [Wilk18a]. *FAIR enough* offers a web-based UI, a accessible API and a Docker image for automated self-hosted local *FAIRness* assessment.

***FAIRware Workbench* [Muse22]** follows a different approach. Based on generalized templates instantiated for a specific digital resource, the *FAIRness* of this resource can be assessed. The *FAIRware Workbench* is embedded into the *CEDAR* platform. Although the automated *FAIRness* evaluation is possible via an API, creating the *CEDAR* template before the evaluation requires high effort and is counter-intuitive. This significantly reduces the usability of the tool. But, the *FAIRware Workbench* provides additional insights to users and gives suggestions on how the *FAIRness* of the assessed data set or digital object can be increased [Muse22].

The authors and developers of the assessment tools used disclaim that their tools are early versions and under active development, so the assessments should be used and evaluated cautiously. Nevertheless, following the reviews by Peters-von Gehlen et al. [Pete22] and Sun et al. [Sun22], these are the most sophisticated ones that allow for automatic and computational evaluation at the time of writing and, thus, are the only frameworks that guarantee a reliable and objective evaluation.

[3] `https://fairsharing.github.io/FAIR-Evaluator-FrontEnd/`
[4] `https://fairshake.cloud/rubric/8/`

The input is always a publicly accessible, persistent, and globally unique identifier. Due to security and privacy concerns, only the interfaces of the *Dummies* could be made publicly available. So, the assessment is only executed for these three *FAIR Sensor Services*. However, the kind of the *FAIR Sensor Service* (simulated vs. real) does not influence the data model and format. Therefore, the assessment yields generalizable results that cover all *FAIR Sensor Services* implemented. The author tested the six tools presented for automated *FAIRness* assessment of the manufacturing sensor data produced by the implemented *FAIR Sensor Services*. Only the three tools *FAIR enough*, *F-UJI*, and *FAIR Checker* turned out to be usable. The other tools did not respond, raised internal server errors, or required effortful preparation, which is only sparsely documented. As *FAIR enough* offers three collections of metrics for evaluation, all three have been used to acquire a detailed assessment. However, the tests of the collection *fair-evaluator-maturity-indicators* implementing the *FAIR* Metrics Framework seem to be implemented incorrectly by the developers because all metrics analyzing the data set retrieved under the requested identifier raise a parsing error. Thus, this collection has been excluded from the evaluation. Nevertheless, the *FAIR* Metrics Framework are also implemented in *FAIR Checker* and are therefore also used for the *FAIRness* assessment.

Achieved *FAIRness* scores

A relative score has been computed compared to the maximum possible score for all metrics to align the scales to a percentage scale. The resulting automatically and computationally retrieved *FAIRness* scores are depicted in Figure 6.10. The scores of the individual metrics are available in the results published online [Bode24d]. As shown in Figure 6.10, *FAIR Checker* yields higher *FAIRness* scores on average compared to the other two tools. The interoperability score slightly differs, while the accessibility score reveals a significant deviation between the highest and the lowest score. For findability, all four assessments deviate. For the other three *FAIRness* aspects, there are always at least two assessments that yield a comparable score. Considering the reusability only *FAIR Checker* strongly deviates. In general, the achieved score reveals the lowest reusability score, which can be attributed to guideline **R1.3**. Due to the nature of this guideline, generic, domain-agnostic *FAIRness* metrics and tools can obviously not assess this guideline. In contrast, findability, accessibility, and interoperability can largely be evaluated using generic metrics and tools.

Reference *FAIRness* values of other data sets - further considered baseline - have only been found for the assessment using the *FAIR enough* framework. The comparison of the average *FAIRness* of the baseline data sets and data produced with the *FAIR Sensor Services* is shown in Table 6.2. Based on an overall of thousand assessments retrievable, the assessments using the *fair-enough-data collection* the metadata and the MPs provided by the *FAIR Sensor Services* yield an absolute increase of the *FAIRness* score of 13.7%, which corresponds to a relative improvement of 36.0%, compared to the baseline. Based on the *fair-enough-metadata collection FAIR Sensor Services* improve the *FAIRness* score by 29.4% in total, which is approx 70% higher than the baseline. Publicly available reference values obtained with *F-UJI* or *FAIR Checker* could not be found during the studies.

Thus, the results indicate a significant improvement in the *FAIRness* of the acquired measurement data compared to the baseline and justify the methods and models developed. However, the low scores, especially on the assessment with *F-UJI*, also reveal improvement potentials

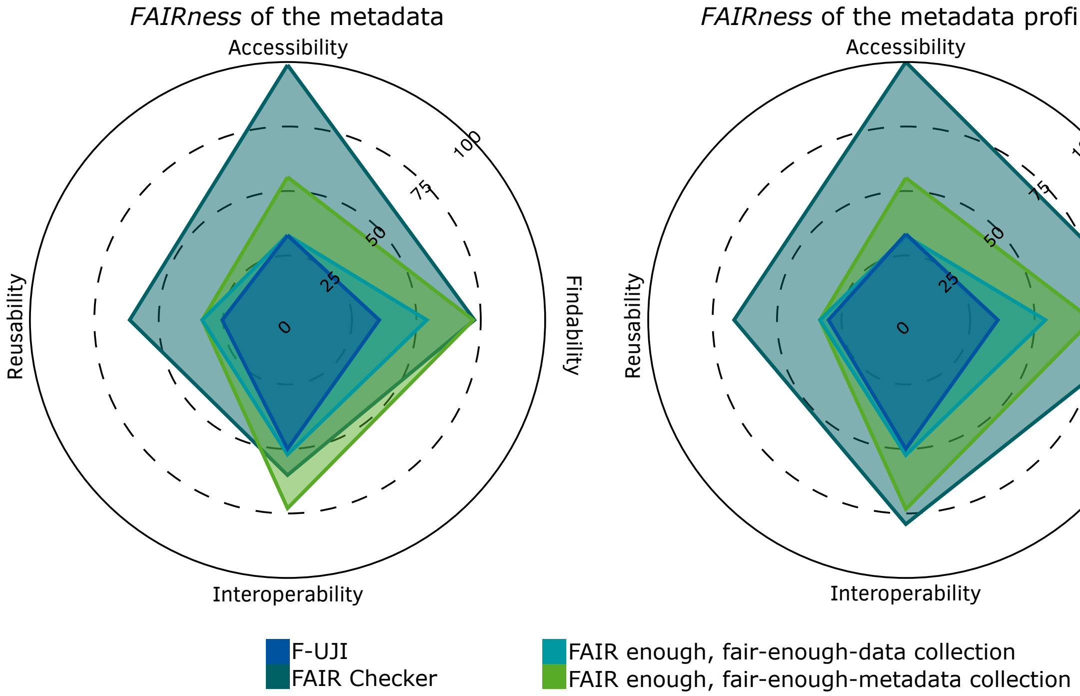


Figure 6.10: The *FAIRness* scores of the *FAIR manufacturing sensor data* produced by the *FAIR Sensor Service* assessed with automated *FAIRness* assessment tools. The plots show the results obtained using the three different frameworks and *FAIRness* metrics: *F-UJI* [Deva23]; *FAIR Checker* [Gaig23]; *FAIR enough*, *fair-enough-data* collection; *FAIR enough*, *fair-enough-metadata* collection. The score depends heavily on the metrics applied and shows promising results for all *FAIR Guiding Principles*.

Table 6.2: Comparison of the average *FAIRness* of data acquired and provided by the *FAIR Sensor Services* developed in this thesis and publicly available evaluations of foreign researchers' data sets via the *FAIR enough* framework.

	Baseline		*FAIR Sensor Services*	
Collection	Nr. of datasets	Avg. *FAIRness*	Nr. of data points	Avg. *FAIRness*
fair-enough-data	686	38.1%	218	51.8%
fair-enough-metadata	182	42.2%	449	71.6%

of the author's solution. According to the *FAIRness* assessments, the primary deficit of the novel sensor model is the missing description of the data provenance. Moreover, there is no explicit link to a dedicated address where the data can be found, and links to external references are lacking. Following a servitisation approach, only the measurement and system metadata are provided by the *FAIR Sensor Service*. The contextualization of the (meta-)data introducing external links and provenance attribution is done by the *FAIR Sensor Ecosystem*, which is aware of the application context of the measuring system [Bode23]. With this contextualization step, the *FAIRness* of the measurement data is expected to increase even further. A preliminary evaluation of the *FAIRness* of data stored in the *FAIR Sensor Ecosystem* using the data structure for contextualization shown in Figure C.1 has been executed in previous work of the author (cf. Bodenbenner et al. [Bode23] and Appendix C). However, a full validation is yet to be done as the implementation of the full semantic contextualization model is not finished for the *FAIR Sensor Ecosystem*. Nevertheless, based on the results of the data contextualized and stored by *FAIR Sensor Ecosystem* in the previous study and the scores achieved for the data acquired by the *FAIR Sensor Services* developed and implemented in this thesis, it can be expected that data provided by a *FAIR Sensor Service* and then contextualized by the *FAIR Sensor Ecosystem* using the semantic contextualization model introduced in Section 4.3 will achieve even higher *FAIRness* scores.

6.5 Intermediate summary

The implementation of all software artifacts and their validation according to the *V-Model* successfully concludes the *Design Cycle* of the research activity for this doctoral thesis. The architecture of the overall infrastructure for producing, managing, and storing *FAIR manufacturing sensor data* has been introduced based on the *FAIR* DLC and implemented through the *FAIR Sensor Services* and *FAIR Sensor Ecosystem*. Six *FAIR Sensor Services* have been implemented, of which three are simulated measuring systems, and the other three are real systems. The implementation is based on open-source libraries and commonly used programming languages. The open-access publication of all software artifacts facilitates the awareness and re-use of the developed methods and contributes to the knowledge base. Therefore, this thesis and the research carried out for it are a salient example of responsible and sustainable research data management. The integration tests ensured the implemented software's correct functionality without any compromises.

Finally, the acceptance test by a qualitative and quantitative evaluation of the *FAIRness* of the data acquired with the six *FAIR Sensor Services* and stored in the *FAIR Sensor Ecosystem* revealed outstanding results. The manual, qualitative evaluation in Section 6.4.1 revealed high coverage and compliance of the developed tools with the defined *FAIRness* criteria. The automated, quantitive *FAIRness* assessment yields remarkable *FAIRness* scores compared to state of the art (cf. Table 6.2). Thus, it can be concluded that the defined *FAIRness* criteria facilitate the implementation of the *FAIR Guiding Principles* for manufacturing and that the developed data models, SOIL and the software tools successfully implement these criteria. This proves the effectiveness of the developed methods and software tools concerning the formulated research questions and implementation of the *FAIR Guiding Principles* within the scope defined for this thesis (cf. Section 1.3). This leads to the positive confirmation of the principal research question and the reflection of the research in the next chapter.

7 Discussion of the results, conclusion and outlook

Based on the conducted research and the developed, implemented, and validated artifacts, this thesis concludes with an answer to the principal research question. It is analyzed how the *Sensor Interfacing Language*, *FAIR Sensor Services*, and the *FAIR Sensor Ecosystem* overcome the deficits and challenges of industrial data management and the limited value-creation potential of measurement data in the manufacturing domain. The methods and results of the research are critically reflected. Lastly, the thesis finishes with an outlook on the subsequent research potential and open questions.

7.1 Conclusion and response to the principal research question

This thesis proposed an approach for implementing the *FAIR Guiding Principles* in the manufacturing domain. Based on the concept of *Distributed Sensor Services*[Pete20; Berg20, p. 158-180] this thesis introduced *FAIR Sensor Services* as the concrete implementation of cyber-physical measuring systems, providing *FAIR manufacturing sensor data*, and the *Sensor Interfacing Language* (SOIL) for the streamlined development of *FAIR Sensor Services*. This proves that, using domain-specific modeling languages, manufacturing sensor data can be acquired such that the resulting data sets adhere to the *FAIR Guiding Principles*, as shown in Figure 7.1.

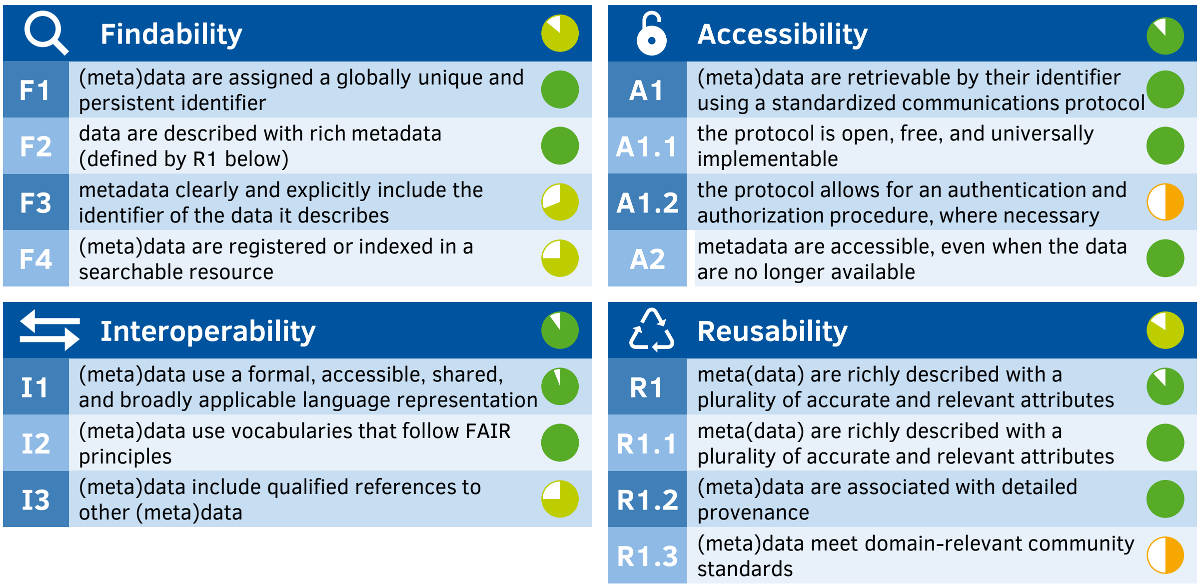


Figure 7.1: Mapping of the qualitative evaluation of the fulfillment of the defined 17 *FAIRness* criteria by SOIL, *FAIR Sensor Services*, and the *FAIR Sensor Ecosystem* to the *FAIR Guiding Principles* using Table 3.2 and Table 6.1. The detailed mapping is shown in Table H.1.

This positively confirms the principal research question as summarized by Table 7.1. Based on an abstraction of semantically interoperable data models to simpler data structures, SOIL has been developed and implemented to generate models for *FAIR manufacturing sensor data*. The unique feature of SOIL is the usage of controlled and open vocabularies, e.g., ontologies, ensuring interoperability and interpretability, combined with the provision of the *SOIL Workbench* and *SOIL Editor*, allowing engineers to define simple, abstracted sensor models. For the latter, the Python Unified Device Interface has been developed as the specific realization of *FAIR Sensor Services*.

Table 7.1: Based on the preliminary responses to each individual RGQ in Chapters 3 to 5 and the overall validation in Chapter 6, the principal research question can be confirmed positvely. The checkmarks indicate that an effective solution has been found for each individual research question (see Section 3.4, Section 4.5, Section 5.5, and Section 6.5).

RGQ1	What are the criteria manufacturing sensor data must meet to satisfy the requirements defined by the *FAIR Guiding Principles*?	✓
RGQ2	How can manufacturing sensor data be described to be findable and interpretable in the applications' context?	✓
RGQ3	How can controlled vocabularies be reused systematically by domain-specific modeling languages to provision accessible and interoperable manufacturing sensor data?	✓
PRQ	**Can manufacturing sensor data be made *FAIR*, i.e., findable, accessible, interoperable, and reusable, by using domain-specific modeling languages?**	✓

Utilizing SOIL, *FAIR Sensor Services*, and the *FAIR Sensor Ecosystem*, the four deficits of industrial data management limiting (re-)use and value-creation potential of manufacturing sensor data have been successfully addressed as summarized in Figure 7.2. Deficit **D3 Absence of specific criteria for FAIR data management** has been resolved by formulating 17 specific criteria for *FAIR* data management. Moreover, the *FAIR* DLC and the system architecture derived from that help to implement the *FAIR Guiding Principles*. The proposal of a protocol-agnostic, syntactic data structure as the base of SOIL and the developed semantic sensor data model based on controlled vocabularies provide a method for compensating for deficit **D1 Missing interoperability**. Using SOIL, company- and device-specific data description and transmission approaches are replaced by unified data models and open and standardized communication protocols and data formats. The usage of MQTT and OPC UA expedite the industrial applicability of *FAIR Sensor Services*. The standardized protocols also facilitate the connection of former isolated data silos into an interconnected Internet of Production [Schu17], as both the *FAIR Sensor Services* and the *FAIR Sensor Ecosystem* have increased accessibility compared to state-of-the-art solutions. Once authentication and authorization mechanisms are implemented, the prevailing compartmentalization of data sources and databases, i.e., deficit **D4 Isolated data silos**, can be overcome. Lastly, this work provides the *SOIL Editor* and *SOIL Workbench* for allowing the detailed description of manufacturing sensor data with metadata. Moreover, SOIL lowers the required development effort and modeling expertise for implementing a *FAIR Sensor Service*.

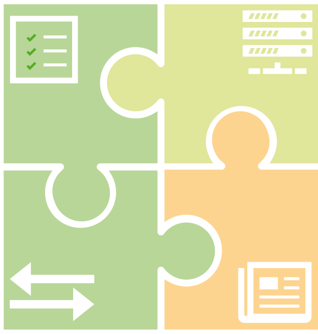
The *FAIR Sensor Ecosystem* brings the technical possibility to link manufacturing sensor (meta-)data to (meta-)data of related assets. Using the novel contextualization model and the *FAIR Sensor Ecosystem*, the provenance of the manufacturing sensor data can be preserved. Moreover, all data contains licensing information to explicitly allow (or disallow) (re)-use. However, the *FAIR Sensor Ecosystem* has been mainly conceptualized but implemented and validated based on a preliminary data structure only (see Appendix C). Thus, the approach to solve deficit **D2 Poor contextual embedding** is only validated for syntactic interoperability (cf. Definition 2.4). The contextualization model proposed in Section 4.3 still required practical proof. Furthermore, the challenges associated with operationalizing the developed methods were overcome by employing MDSE and developing SOIL. In summary, SOIL and *FAIR Sensor Services* (and all related, developed methods and tools) solve the identified deficits. Thus, the produced *FAIR manufacturing sensor data* have increased exchange and value creation potential. Nevertheless, the effectiveness and positive impact of *FAIR manufacturing sensor data* to the amount of data (re-)use must be demonstrated in industrial practice. It cannot be conclusively validated and answered in this thesis.

Absence of specific criteria for *FAIR* data management

- Defined17 specific *FAIRness* criteria
- Defined FAIR-DLC as reference frame for implementation

Missing interoperability

- Provided two-level modeling approach
  - Simple syntactic data structure
  - Sophisticated semantic data model
- Simplified and enforced usage of controlled terminology



Isolated data silos

- Protocol-agnostic data description
- Automated interface generation
- Accessible data interface

Poor contextual embedding

- Domain-specific reusability information are provided (measurement uncertainty, …)
- Defined contextualization model (full implementation remains)

Exchange and value creation potential is increased by *FAIR manufacturing sensor data*

Figure 7.2: The major deficits of industrial data management as identified in Figure 2.2 have been addressed by the conducted research and could be largely resolved within the scope of this thesis specified in Section 1.3. The contextualization model as part of the solution for **D2 Poor contextual embedding**, defined in Section 4.3, has not been fully validated while the methods and tools to compensate for the deficits **D1 Missing interoperability**, **D3 Absence of specific criteria for FAIR data management**, **D4 Isolated data silos** have been fully implemented and validated

7.2 Critical reflection of the research approach and the achieved results

All research is subject to finite resources, such as time or available hardware and software, limiting the achievable. To pro-actively credit for that, the objectives and the scope of this thesis have been precisely defined based on formal research methodologies. Nevertheless, the work and results indicated missing ends and room for improvement, although the formulated research questions could be positively answered and the goals be achieved.

Limits and deficits of the results and artifacts

A high degree of complexity and effort in the research for this thesis has been induced by the multitude of features and aspects of the developed models and SOIL and the requirement to demonstrate the compatibility with and translatability to multiple different data formats of communication protocols, and programming languages. The continuous development and improvement of the data models and SOIL cause the need for continuous updates of all developed tools and artifacts based on these, which applies to all software libraries developed during the research for this thesis. Many of these activities are repeated execution and implementation of the same aspects or tasks for different programming languages and communications protocols. It has been ensured that all features are implemented at least once in any of the libraries for the *FAIR Sensor Service* to validate all aspects completely. Realizing the remaining (advanced) features for all libraries and considered technologies can be considered implementation only and most likely will not reveal new scientific insights. The basic features have been implemented in all libraries to validate the scalability and transferability of the method to other or new communication protocols, data formats, and programming languages. Table 7.2 shows which features are implemented in which library.

Table 7.2: The matrix shows which implemented library - namely the Python Unified Device Interface (PUDI) [Bode24b], the C++ Unified Device Interface (CUDI) [Mont21], and the SOIL-based OPC UA Python library [Ens24] - implements the features and characteristics of which developed method. As the translation into the different protocols, data formats, and programming languages could be proven in general, the overall objective is considered to be achieved. The remaining implementation is a repetition of the things that are already done. Therefore, it can be assumed that this can be done successfully.
Legend: ○ not implemented, ◔ partially included and not automatically generated, ◑ partially included, but automatically generated, ◕ full implementation and generation based on an intermediate version of SOIL, ● fully implementation and generation based on the current version of SOIL.

	Artifact		
Features	**PUDI**	**CUDI**	**OPC UA**
Elements	●	◕	◕
Streams	●	◔	◕
Events	●	◔	◕
Semantics	●	○	○
Modifiers	●	◑	◑
Overriding Context	●	◕	◕

Another aspect of incompleteness concerns the coverage of the *FAIR Guiding Principles* and the defined *FAIRness* criteria for manufacturing sensor data. The implementation of the *FAIR Guiding Principles* for the manufacturing domain cannot be entirely provided by one dissertation due to the complexity of the topic. While some aspects have been left out intentionally, such as the legal and security aspects, others have been investigated in theory but not in practice. The inclusion of licensing and usage metadata has been technically realized by a dedicated field for the usage license, but it has not been investigated which licenses are appropriate. Similarly, all implemented communication protocols support authentication and authorization procedures but have not been implemented in this thesis, as they do not yield scientifically relevant new findings. However, both are crucial for industrial applicability and must, therefore, be addressed and implemented to realize the industrial application of the results of this thesis. Appropriate controlled terminology is essential for unambiguously interpreting the acquired manufacturing sensor data. The developed data and contextualization model and SOIL provide the technical solution to enable developers with minimal modeling expertise to utilize controlled terminology. However, the selection of appropriate terms can not be guaranteed. To maintain flexibility and address **CF.1 No Limitations**, restricting specific ontologies or vocabularies is unreasonable, which might lead to inappropriate usage. For example, it can not be ensured that the term used in the `unit` actually defines a unit. Furthermore, validating whether a specified prefix and the used controlled term can be resolved is error-prone because interfaces for providing controlled terminology are often insufficiently implemented.

From a modeling perspective, simplifications have been made (cf. Section 4.2.3) that are not strictly compatible with the used ontologies. Although this does not affect the *FAIRness* of measurement data, it could cause issues when executing strict semantic reasoning on the data. However, the proposed models significantly increase the overall *FAIRness* and the long-term reusability of measurement data. Thus, the models fully serve the intended purpose and scope with minor drawbacks only.

Industrial applicability

Despite small gaps to be closed by additional engineering and implementation soon, (aspects of) SOIL have already been applied in practice and adopted by others. Göppert et al. [Göpp23] and Mathews et al. [Math23] adopted an early version of the data structure defined in Section 4.2.1 to develop an ontology-based *Digital Twin Pipeline* [Göpp23] for the streamlined definition and automatic generation of use-case specific DTs. Göppert et al. [Göpp23] slightly extended the data structure by merging it with key concepts of the AAS, which also demonstrates the compatibility of the models developed within this thesis with the AAS. In a similar approach, the data structures have also been applied in multiple subsequent industrial projects to define the information model for a new and digitalized assembly line of a German automotive manufacturer.[1] Translating SOIL models into OPC UA node sets and generating an OPC UA server has been demonstrated by a prototypical implementation. However, the semantic interoperability of OPC UA servers must be ensured by using the node types pre-defined in the CSs of OPC UA, which is not reflected in a plain SOIL model. Thus, the industrial applicability of SOIL to develop OPC UA servers requires additional effort to ensure semantic interoperability (see Section 7.3).

Furthermore, data licensing is not well applicable in an industrial context. Negotiating individual usage permissions and rights would be more suitable as they provide more fine-grained control over the data release and (re-)use. Linking SOIL and *FAIR Sensor Service* with developments in the area of data spaces, such as the Eclipse Data Space Connector for *Gaia-X*, could help here. The structure of SOIL allows the development of an extension of the current syntax and semantics that would enable the configuration of specific access and reuse regulations up to individual variables. This configuration would offer more specificity for access and reuse conditions than the status quo.

[1] The details are under a non-disclosure agreement and can therefore not be described here.

Reflection of the research methodology

The conducted research actions have been organized and formalized with Design Science Research (DSR) [Hevn07]. Starting from an initial idea derived from open questions of previous work by Montavon (et al.) [Mont19; Mont21], the specific needs of the manufacturing industry have been investigated, and a prototype of a visual syntax of SOIL has been developed and published [Bode21b]. With additional studies on the environment, the deficits of industrial data management have been analyzed and identified in detail. Moreover, a review of related work and literature from the knowledge base revealed the *FAIR Guiding Principles* as a promising approach for solving the identified deficits [Bode21a]. Based on the formulation of the research objectives and questions, the formal methods, data models, and SOIL have been iteratively developed and improved in parallel. The timely implementation of roughly sketched methods helped identify and correct the weaknesses. Intermediate artifacts and solutions to individual questions have already been published in journals [Bode22a; Bode22b] and presented at conferences [Bode21a; Bode23; Bode24f] to feed the findings back into the knowledge base, facilitate the exchange with other researchers and retrieve additional input to improve the outcome of the research. The software artifacts implementing SOIL, *FAIR Sensor Services*, and the *FAIR Sensor Ecosystem* have been provided to colleagues within WZL to promote early usage. Later, the artifacts were published open-access so that the results were already usable by others before the publication of this thesis, which increases the *FAIRness* of the conducted research itself. Thus, DSR framed the conducted research successfully and positively confirmed the research questions. Moreover, the identified deficits in industry could be addressed. Following the guidelines for good scientific practice and research data management, all results and artifacts have been published open-access by the author (see the data availability statement)[2]. Thus, this thesis, the related research, and artifacts can be considered a meta-artifact of exemplary research data management.

7.3 Subsequent research questions and outlook

This thesis only provides a few “puzzle-pieces” to the implementation of the *FAIR Guiding Principles* for manufacturing. The identified core deficits were largely resolved, and all formulated research questions could be answered positively. However, some points have been left out intentionally to limit the scope of this thesis, and other questions arose during the author’s research. Four open challenges and needs are briefly presented in the following.

Connectability to industrially employed information models

The translation from SOIL models to an OPC UA information model and the generation of the server implementation have been implemented and validated successfully in Sections 5.4.2 to 5.4.3. This shows that the development and implementation effort and complexity of an OPC UA server can be reduced significantly. However, the semantic interoperability of specific OPC UA information models is provided by the Companion Specifications (CSs). To ensure that an underlying SOIL model is compliant with CSs, one can define an inverse mapping to automatically translate nodes defined in CSs into default elements for SOIL. The general

[2]Some artifacts can not be published due to restrictive licenses of required dependencies or non-disclosure agreements. All these artifacts are listed and shortly described in the data availability statement.

feasibility of such an approach has already been demonstrated by Steindl and Kastner [Stei21]. An exemplary manual translation of the node type *RotaryTableType* from the CS *Geometric Measuring Systems* [VDMA23] into a SOIL model is depicted in Listing 7.1, which illustrates the basic idea and indicates the feasibility. Other industrially used information models, such as AAS or QIF, have been discussed, but a mapping to these models has not been defined and implemented. Such definition and implementation are reasonable next steps to facilitate the industrial operationalization and connectability of the developed methods to initiatives such as *Gaia-X* and *International Data Spaces*.

```
import opcuaInformationModel;
import opcuaMachinery;

variable IsIntegrated {
  name: "is integrated"
  description: "IsIntegrated is defined as True if the RotaryTable is
    integrated into the GMS and cannot be removed."
  datatype: boolean
  dimension: []
}

variable NrOfAxes {
  name: "Number of axes"
  description: "NumberOfAxes defines the number of axes (DegreesOfFreedom) the
    RotaryTable can be configured for."
  datatype: int
  dimension: []
  range: (0,)
}

component RotaryTable extends opcuaInformationModel.BaseObject {
  name: "Rotary Table"
  description: "The rotary table used as an additional degree of freedom in
    some geometric measuring system."
  components:
    opcuaMachinery.MachineryItemIdentification Identification
  parameters:
    NrOfAxes nrOfAxes
    IsIntegrated isIntegrated
}
```

Listing 7.1: Exemplary definition of the *RotaryTableType* of the OPC UA CS *Geometric measuring systems* [VDMA23, p. 39] (cf. Table I.1) as component of a SOIL model including the required variables.

Remaining *FAIR Guiding Principles*

Following the discussion in Section 7.2, some aspects of the *FAIR Guiding Principles* need further investigation to provide manufacturing sensor data in an even more *FAIR* way. First, the investigation of suitable alternatives for specifying a license must be found to properly implement **R1.1** for the industrial domain. The specification of individually negotiable reuse conditions and contracts is more suited. Moreover, providing globally unique and persistent identifiers and ensuring the longevity of the metadata as required by **F1** and **A2** is an organizational challenge. In the case of manufacturing data, companies want to keep control of their own data since it could reveal corporate secrets about manufacturing techniques [Penn19b].

However, if a company shuts systems offline, the data can be unobtainable if not archived on an external system. Thus, methods must be developed which, on the one hand, ensure the owner's control over data, but on the other hand, guarantee the (meta-)data remains uniquely, persistently identifiable, and available in case the data owner no longer exists.

Domain-specific controlled terminology

During the author's research, the lack of domain-specific controlled terminology compliant with *Semantic Web* standards was the biggest obstacle to the definition of the data models, resulting in highly increased modeling complexity. The same issue has also motivated the research project *Applying Interoperable Metadata Standards (AIMS)* the author participated in [Grön23]. Extensive standardized vocabularies are already available, such as the VIM [BIPM12], the "CIRP Encyclopedia of Production Engineering" [The 16], or the "Dictionary of Production Engineering" [CIRP11]. However, these vocabularies are not compliant with *Semantic Web* standards and are not machine-actionable. These can not be used to describe and document manufacturing sensor data compliant with the *FAIR Guiding Principles* (in particular **R1.3**). The conversion into controlled vocabularies is crucial to lower the modeling complexity and increase the acceptance of SOIL and the *FAIR Guiding Principles*.

FAIR and FACT Data Processing Services

This thesis focuses on the creation of *FAIR manufacturing sensor data* but does not explicitly address the **Short-term usage** and **Long-term usage** of sensor data as sketched by the lifecycle of *FAIR manufacturing sensor data* (cf. Figure 3.4). Given *FAIR manufacturing sensor data* based on SOIL, the logical next step is developing algorithms and software making maximum use of the unique characteristics of *FAIR* data. One possible approach is the definition of interoperable interfaces for data processing services based on SOIL. Using the same language for data producers and consumers enables them to automatically detect algorithms that can process certain data or, vice-versa, find data that a particular algorithm can process. Even more, generic *metadata-based algorithms* could be automatically parametrized or configured based on the specific metadata of processed sensor data. The basic idea behind the concept is to lift former highly specific logic into more abstract processing steps while encoding the specific logic in the metadata. For example, instead of implementing a specific sensor fusion algorithm for individual sensors (types), one implements a generic fusion algorithm, and the information on how to process the data for specific sensors can be obtained from the metadata of the sensor. However, developing such algorithms raises additional questions, such as how this required generalization can be formalized. It needs to capture the (physical) relations between sensor and assets in more detail than the contextualization model proposed in Section 4.3 allows. Such data processing capabilities would extend the work of this thesis by *FACT*-based, i.e., fair (in terms of unbiased), accurate, confidential, and transparent, algorithms [vdAals17] that pave the way from sustainable data management to responsible data usage and, ultimately, accountable and resilient decision making.

Acronyms

AAS	Asset Administration Shell
AIMS	Applying Interoperable Metadata Standards
API	application programming interface
ASAM	Association for Standardization of Automation and Measuring Systems
AST	Abstract Syntax Tree
CDE	Calibration Data Exchange Format
CMM	coordinate-measuring machine
CoAP	Constrained Application Protocol
CoCo	context condition
CODATA	Committee on Data for Science and Technology
CPMS	cyber-physical measuring system
CPS	cyber-physical system
CS	Companion Specification
CUDI	C++ Unified Device Interface
DCC	Digital Calibration Certificate
DC-Terms	DCMI Metadata Terms
DDL	data description language
DLC	data life-cycle
DOI	digital object identifier
DMSC	Digital Metrology Standards Consortium
DOI	Digital Object Identifier
DS	Digital Shadow
DSL	domain-specific language
DSPL	domain-specific programming language
DSML	domain-specific modeling language
DSR	Design Science Research
DT	Digital Twin
EDSC	Eclipse Data Space Connector
ERP	Enterprise Resource Planning
***FAIR* DLC**	life-cycle of *FAIR manufacturing sensor data*
***FAIR* DMM**	*FAIR* Data Maturity Model

GPL	general-purpose language
gRPC	gRPC Remote Procedure Calls
GUM	Guide to the expression of uncertainty in measurement
HDF5	Hierarchical Data Format (Version 5)
HTTP	Hypertext Transfer Protocol
ICA	International Council on Archives
IDL	interface description language
IDS	International Data Spaces
IIoT	Industrial Internet of Things
I++ DME	Inspection Plus Plus Dimensional Measurement Equipment
IoP	Internet of Production
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
LOV	Linked Open Vocabulary
LSM	large-scale metrology
M4I	Metadata4Ing
MCS	motion capture system
MDSE	model-driven software engineering
MDF	Measurement Data Format
MES	Manufacturing Execution System
MMPD	Meta-Model for Production Data
MP	metadata profile
MPS	NFDI4Ing Metadata Profile Service
MQTT	Message Queuing Telemetry Transport
NDA	non-disclosure agreement
NFDI4Ing	National Research Data Infrastructure for the Engineering Sciences
OM	Ontology of units of Measure
OPC	Open Platform Communications
OPC UA	Open Platform Communications Unified Architecture
OSF	Open Semantic Framework
OWL	Web Ontology Language
protobuf	Protocol Buffers
PROV-O	Provenance Ontology
PROV-DM	Provenance Data Model
PRQ	principal research question
PUDI	Python Unified Device Interface
PURL	persistent URL

PyPi	Python Package Index
QIF	Quality Information Framework
QUDT	Quantities, Units, Dimensions and Data Types Ontologies
RAMI 4.0	Reference Architecture Model Industry 4.0
RDA	Research Data Alliance
RDF	Resource Description Framework
RGQ	research guiding question
REST	Representational State Transfer
RiC-O	Records in Contexts-Ontology
RiC-CM	Records in Contexts Conceptual Model
RPC	Remote Procedure Call
SHACL	Shapes Constraint Language
D-SI	Digital System of Units
SMR	Spherically Mounted Retroreflector
SOA	service-oriented architecture
SOIL	*Sensor Interfacing Language*
SOSA	Sensor, Observation, Sample, and Actuator
SQL	Structured Query Language
SSN	Semantic Sensor Network Ontology
TCP	Transmission Control Protocol
TD	Thing Description Ontology
UI	user interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
VIM	International vocabulary of metrology
WoT	Web of Things
WSDL	Web Services Description Language
WSDL-S	Web Service Semantics
WWL	World-Wide Lab
WZL	Laboratory for Machine Tools and Production Engineering
XML	Extensible Markup Language
XSD	XML Schema Definition Language

References

[Acem22] Daron Acemoglu and Pascual Restrepo. “Demographics and Automation”. In: *The Review of Economic Studies* 89.1 (Jan. 1, 2022), pp. 1–44. ISSN: 0034-6527. DOI: 10.1093/restud/rdab031.

[Adol15] Peter Adolphs, Heinz Bedenbender, Dagmar Dirzus, et al. *Reference Architecture Model Industrie 4.0 (RAMI4.0)*. VEI – German Electrical, Electronic Manufacturers’ Association, and VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, July 2015. URL: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report__Reference_Archtitecture_Model_Industrie_4.0__RAMI_4.0_/GMA-Status-Report-RAMI-40-July-2015.pdf (visited on Oct. 15, 2024).

[Ahma23] Saeed Ahmad, Sehrish Ali, Nazish Waqar, Naila Samar Naz, and Muhammad Hassaan Mehmood. “Comparative Evaluation of the Maintainability of RESTful and SOAP-WSDL Web Services”. In: *2023 International Conference on Business Analytics for Technology and Security (ICBATS)*. 2023 International Conference on Business Analytics for Technology and Security (ICBATS). Mar. 2023, pp. 1–9. DOI: 10.1109/ICBATS57792.2023.10111436.

[Akki05] Rama Akkiraju, Joel Farrell, John Miller, et al. *Web Service Semantics - WSDL-S*. Apr. 2005. URL: https://corescholar.libraries.wright.edu/knoesis/69 (visited on Oct. 15, 2024).

[Altu23] Osman Altun, Pooya Oladazimi, Max Leo Wawer, et al. “Enhanced Findability and Reusability of Engineering Data by Contextual Metadata”. In: *Proceedings of the Design Society* 3 (July 2023), pp. 1635–1644. ISSN: 2732-527X. DOI: 10.1017/pds.2023.164.

[Amra17] Moussa Amrani, Fabian Gilson, Abdelmounaim Debieche, and Vincent Englebert. “Towards User-centric DSLs to Manage IoT Systems”. In: *MODELSWARD 2017*. Ed. by Luís Ferreira Pires, Slimane Hammoudi, and Bran Selic. Setúbal: SCITEPRESS - Science and Technology Publications Lda, 2017, pp. 569–576. ISBN: 978-989-758-210-3. DOI: 10.5220/0006285405690576.

[AMT 23] AMT - The Association for Manufacturing Technolog. *MTConnect Standard, Part 2.0 – Device Information Model*. Aug. 8, 2023. URL: https://docs.mtconnect.org/MBSD_MTConnect_Part_2_2-2-0.pdf (visited on May 7, 2024).

[Asso19] Association for Standardization of Automation and Measuring Systems. *Regulation of Fees*. Aug. 29, 2019. URL: https://www.asam.net/index.php?eID=dumpFile&t=f&f=673&token=1321eb0a4e6cfb99112f61648e411520f83a8d01 (visited on May 9, 2024).

[Azzo13] L. Azzopardi, C. Wilkie, and Tony Russell-Rose. "Towards Measures and Models of Findability". In: 2013. URL: https://www.semanticscholar.org/paper/Towards-Measures-and-Models-of-Findability-Azzopardi-Wilkie/f3e5b285f2773f9fc63e9fd66b23b5583aa8c102 (visited on June 8, 2024).

[Bahi20] Christophe Bahim, Carlos Casorrán-Amilburu, Makx Dekkers, et al. "The FAIR Data Maturity Model: An Approach to Harmonise FAIR Assessments". In: *Data Science Journal* 19.1 (2020). ISSN: 1683-1470. DOI: 10.5334/dsj-2020-041.

[Ball12] Alexander Ball. *Review of Data Management Lifecycle Models*. Bath: University of Bath, Feb. 13, 2012, p. 14. URL: https://researchportal.bath.ac.uk/en/publications/review-of-data-management-lifecycle-models (visited on Mar. 27, 2024).

[Baum17] Felix W. Baumann, Uwe Breitenbücher, Michael Falkenthal, Gerd Grünert, and Sebastian Hudert. "Industrial Data Sharing with Data Access Policy". In: *Cooperative Design, Visualization, and Engineering*. Ed. by Yuhua Luo. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 215–219. ISBN: 978-3-319-66805-5.

[Bech22] Dimitrios Bechtsis, Naoum Tsolakis, Eleftherios Iakovou, and Dimitrios Vlachos. "Data-Driven Secure, Resilient and Sustainable Supply Chains: Gaps, Opportunities, and a New Generalised Data Sharing and Data Monetisation Framework". In: *International Journal of Production Research* 60.14 (July 18, 2022), pp. 4397–4417. ISSN: 0020-7543. DOI: 10.1080/00207543.2021.1957506.

[Beck21] Fabian Becker, Pascal Bibow, Manuela Dalibor, et al. "A Conceptual Model for Digital Shadows in Industry and Its Application". In: *Conceptual Modeling*. Ed. by Aditya Ghose, Jennifer Horkoff, Vítor E. Silva Souza, Jeffrey Parsons, and Joerg Evermann. Lecture Notes in Computer Science. Cham: Springer, 2021, pp. 271–281. ISBN: 978-3-030-89022-3. DOI: 10.1007/978-3-030-89022-3_22.

[Beck23] Sebastian Beckschulte, Marcos Padrón Hinrichs, Lorenzo Pirrone, et al., eds. *Manufacturing Data Analytics Study 2023: Empirical Industry Study*. 1. Auflage. Aachen: Apprimus Verlag, 2023. 101 pp. ISBN: 978-3-9855518-1-1.

[Berg20] Thomas Bergs, Christian Brecher, Robert Schmitt, and Günther Schuh, eds. *Internet of Production - Turning Data into Value*. Aachen: Fraunhofer-Gesellschaft, 2020. URL: https://publica.fraunhofer.de/entities/publication/7593c7bf-6ae9-4d4a-99b1-794fb32b4f66 (visited on June 20, 2024).

[Berm18] Francine Berman, Rob Rutenbar, Brent Hailpern, et al. "Realizing the Potential of Data Science". In: *Communications of the ACM* 61.4 (Mar. 26, 2018), pp. 67–72. ISSN: 0001-0782. DOI: 10.1145/3188721.

[Berm06] Luis Bermudez and Michael Piasecki. “Metadata Community Profiles for the Semantic Web”. In: *GeoInformatica* 10.2 (June 1, 2006), pp. 159–176. ISSN: 1573-7624. DOI: 10.1007/s10707-006-7577-2. URL: https://doi.org/10.1007/s10707-006-7577-2 (visited on Mar. 11, 2024).

[Bern05] Tim Berners-Lee, Roy Thomas Fielding, and Larry M. Masinter. *RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax*. Internet Standard. Jan. 2005. URL: https://datatracker.ietf.org/doc/html/rfc3986 (visited on July 3, 2024).

[Bern01] Tim Berners-Lee, James Hendler, and Ora Lassila. “The Semantic Web”. In: *Scientific American Magazine* 284.5 (May 2001), pp. 34–43. URL: https://www.scientificamerican.com/article/the-semantic-web/.

[Bert11] Leopoldo Bertossi, Flavio Rizzolo, and Lei Jiang. “Data Quality Is Context Dependent”. In: *Enabling Real-Time Business Intelligence*. Ed. by Malu Castellanos, Umeshwar Dayal, and Volker Markl. Berlin, Heidelberg: Springer, 2011, pp. 52–67. ISBN: 978-3-642-22970-1. DOI: 10.1007/978-3-642-22970-1_5.

[BIPM19] BIPM. *SI Brochure: The International System of Units (SI)*. Version 9. 2019. URL: https://www.bipm.org/documents/20126/41483022/SI-Brochure-9-EN.pdf (visited on Oct. 15, 2024).

[BIPM08] BIPM, IEC, IFCC, et al. *Evaluation of measurement data — Guide to the expression of uncertainty in measurement*. Joint Committee for Guides in Metrology, JCGM 100:2008. 2008. URL: https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf (visited on Oct. 15, 2024).

[BIPM12] BIPM, IEC, IFCC, et al. *International vocabulary of metrology — Basic and general concepts and associated terms (VIM)*. Joint Committee for Guides in Metrology, JCGM 200:2012. (3rd edition). 2012. URL: https://www.bipm.org/documents/20126/2071204/JCGM_200_2012.pdf (visited on Oct. 15, 2024).

[Bode24a] Matthias Bodenbenner. *Description and Manual of the SensOr Interfacing Language*. Zenodo, May 22, 2024. DOI: 10.5281/zenodo.11239479.

[Bode24b] Matthias Bodenbenner. *Python Unified Device Interface*. Version 10.1.1. Aachen: RWTH Aachen University, Aug. 12, 2024. DOI: 10.5281/zenodo.13922866.

[Bode24c] Matthias Bodenbenner. *SOIL Dummies*. Version 1.0.0. Aachen: RWTH Aachen University, Apr. 8, 2024. DOI: 10.5281/zenodo.10940075.

[Bode24d] Matthias Bodenbenner. *Validation and FAIRness Assessment Results of FAIR Sensor Services Based on the SensOr Interfacing Language*. Zenodo, Apr. 12, 2024. DOI: 10.5281/zenodo.10966080.

[Bode24e] Matthias Bodenbenner and Tim Heß. *Data Model Based on the Sensor Interfacing Language (SOIL) for an OptiTrack Motion Capture System*. Version 1.0.0. Zenodo, Sept. 25, 2024. DOI: 10.5281/zenodo.12592388.

[Bode21a] Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. “FAIR Sensor Services - Towards Sustainable Sensor Data Management”. In: *Measurement: Sensors* 18 (2021), p. 100206. ISSN: 26659174. DOI: 10.1016/j.measen.2021.100206.

[Bode22a] Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. "Model-Driven Development of Interoperable Communication Interfaces for FAIR Sensor Services". In: *Measurement: Sensors* 24 (2022), p. 100442. ISSN: 26659174. DOI: 10.1016/j.measen.2022.100442.

[Bode23] Matthias Bodenbenner, Jan Pennekamp, Benjamin Montavon, Klaus Wehrle, and Robert H. Schmitt. "FAIR Sensor Ecosystem: Long-Term (Re-)Usability of FAIR Sensor Data through Contextualization". In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. July 2023, pp. 1–8. DOI: 10.1109/INDIN51400.2023.10218149.

[Bode21b] Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon, and Robert H. Schmitt. "Domain-Specific Language for Sensors in the Internet of Production". In: *Production at the Leading Edge of Technology*. Ed. by Bernd-Arno Behrens, Alexander Brosius, Wolfgang Hintze, Steffen Ihlenfeldt, and Jens Peter Wulfsberg. Lecture Notes in Production Engineering. Berlin, Heidelberg: Springer, 2021, pp. 448–456. ISBN: 978-3-662-62138-7. DOI: 10.1007/978-3-662-62138-7_45.

[Bode22b] Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon, and Robert H. Schmitt. "Multiprotokollfähiges Meta-Datenmodell". In: *wt Werkstattstechnik online* 112.11-12 (2022). ISSN: 1436-4980. DOI: 10.37544/1436-4980-2022-11-12.

[Bode24f] Matthias Bodenbenner, Dominik Wolfschläger, and Robert H. Schmitt. *Providing FAIR Sensor Data Models Using Semantic Web Technologies and Ontologies*. Apr. 15, 2024. DOI: 10.13140/RG.2.2.27882.73928. Pre-published.

[Booz24] Ryan Booz. *PostgreSQL + TimescaleDB: 1,000x Faster Queries, 90 % Data Compression, and Much More*. Timescale Blog. Feb. 16, 2024. URL: https://www.timescale.com/blog/postgresql-timescaledb-1000x-faster-queries-90-data-compression-and-much-more/ (visited on June 30, 2024).

[Bosc16] Ann Bosche, David Crawford, Michael Schallehn, and Paul Smith. *How Providers Can Succeed in the Internet of Things*. Bain & Company. Aug. 28, 2016. URL: https://www.bain.com/insights/how-providers-can-succeed-in-the-internet-of-things/ (visited on Oct. 31, 2023).

[Bram12] Marco Brambilla, Jordi Cabot, and Manuel Wimmer. *Model-Driven Software Engineering in Practice*. Synthesis Lectures on Software Engineering 1. San Rafael, Calif.: Morgan & Claypool, 2012. 166 pp. ISBN: 978-1-60845-883-7 978-1-60845-882-0. DOI: 10.2200//S00441ED1V01Y201208SWE001.

[Broa24] Broadcom. *RabbitMQ 4.0 Documentation*. July 21, 2024. URL: https://www.rabbitmq.com/docs (visited on June 30, 2024).

[Burt09] Adrian Burton and Andrew Treloar. "Designing for Discovery and Re-Use: The 'ANDS Data Sharing Verbs' Approach to Service Decomposition". In: *International Journal of Digital Curation* 4.3 (3 Dec. 7, 2009), pp. 44–56. ISSN: 1746-8256. DOI: 10.2218/ijdc.v4i3.124.

[Busb24] Axel Busboom. “Automated Generation of OPC UA Information Models — A Review and Outlook”. In: *Journal of Industrial Information Integration* 39 (May 1, 2024), p. 100602. ISSN: 2452-414X. DOI: 10.1016/j.jii.2024.100602.

[Butt20] Anila Sahar Butt, Nicholas Car, and Peter Fitch. “Towards Ontology Driven Provenance in Scientific Workflow Engine”. In: 8th International Conference on Model-Driven Engineering and Software Development. 2020, pp. 105–115. ISBN: 978-989-758-400-8. DOI: 10.5220/0008963701050115.

[CIRP11] C.I.R.P., ed. *Dictionary of Production Engineering/Wörterbuch Der Fertigungstechnik/Dictionnaire Des Techniques de Production Mechanique Vol IV*. Berlin, Heidelberg: Springer, 2011. ISBN: 978-3-642-12006-0. DOI: 10.1007/978-3-642-12007-7.

[Camp24] Daniel Campbell, Bob Stone, and Thomas Kramer. *Qif-Community Repository*. Version 3.0. Mar. 14, 2024. URL: https://github.com/QualityInformationFramework/qif-community (visited on Oct. 8, 2024).

[Cånd10] Gonçalo Cândido, François Jammes, José Barata de Oliveira, and Armando W. Colombo. “SOA at Device Level in the Industrial Domain: Assessment of OPC UA and DPWS Specifications”. In: *2010 8th IEEE International Conference on Industrial Informatics*. 2010 8th IEEE International Conference on Industrial Informatics. July 2010, pp. 598–603. DOI: 10.1109/INDIN.2010.5549676.

[Cava19] Ian M. Cavalcante, Enzo M. Frazzon, Fernando A. Forcellini, and Dmitry Ivanov. “A Supervised Machine Learning Approach to Data-Driven Simulation of Resilient Supplier Selection in Digital Manufacturing”. In: *International Journal of Information Management* 49 (Dec. 1, 2019), pp. 86–97. ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2019.03.004.

[Cava20] Salvatore Cavalieri and Marco Giuseppe Salafia. “Insights into Mapping Solutions Based on OPC UA Information Model Applied to the Industry 4.0 Asset Administration Shell”. In: *Computers* 9.2 (2 June 2020), p. 28. ISSN: 2073-431X. DOI: 10.3390/computers9020028.

[Chin07] Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, and Sanjiva Weerawarana. *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*. June 26, 2007. URL: https://www.w3.org/TR/wsdl20/ (visited on June 2, 2024).

[Cimm23] Andrea Cimmino, Farshid Tavakolizadeh, Kunihiko Toumura, and Michael McCool. *Web of Things (WoT) Discovery: Candidate Recommendation*. Web of Things (WoT) Discovery: Candidate Recommendation. Dec. 5, 2023. URL: https://www.w3.org/TR/wot-discovery/ (visited on Oct. 15, 2024).

[Clar19] Daniel J. B. Clarke, Lily Wang, Alex Jones, et al. “FAIRshake: Toolkit to Evaluate the FAIRness of Research Digital Resources”. In: *Cell Systems* 9.5 (2019), pp. 417–421. ISSN: 2405-4712. DOI: 10.1016/j.cels.2019.09.011.

[Clau21] Florence Clauvad, Anton Rubinstein, Daniel Pitti, et al. *International Council on Archives Records in Contexts Ontology (ICA RiC-O) Version 1.0*. International Council on Archives Expert Group on Archival Description (ICA EGAD), Feb. 12, 2021. URL: https://www.ica.org/standards/RiC/RiC-O_v0-2.html (visited on Jan. 26, 2024).

[Coit20] Tiago Coito, Miguel S. E. Martins, Joaquim L. Viegas, et al. "A Middleware Platform for Intelligent Automation: An Industrial Prototype Implementation". In: *Computers in Industry* 123 (Dec. 1, 2020), p. 103329. ISSN: 0166-3615. DOI: 10.1016/j.compind.2020.103329.

[Colu14] Louis Columbus. *84% Of Enterprises See Big Data Analytics Changing Their Industries' Competitive Landscapes In The Next Year*. Forbes. Oct. 21, 2014. URL: https://www.forbes.com/sites/louiscolumbus/2014/10/19/84-of-enterprises-see-big-data-analytics-changing-their-industries-competitive-landscapes-in-the-next-year/ (visited on Oct. 31, 2023).

[Comp12] Michael Compton, Payam Barnaghi, Luis Bermudez, et al. "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group". In: *Journal of Web Semantics* 17 (2012), pp. 25–32. ISSN: 1570-8268. DOI: 10.1016/j.websem.2012.05.003.

[Coyl17] Karen Coyle. "Application Profiles: An Overview". In: *Developing Metadata Application Profiles*. 2017, p. 15. ISBN: 978-1-5225-2221-8.

[Cram21] Simon Cramer, Max Hoffmann, Peter Schlegel, Marco Kemmerling, and Robert H. Schmitt. "Towards a Flexible Process-Independent Meta-Model for Production Data". In: *Procedia CIRP* 99 (2021), pp. 586–591. ISSN: 2212-8271. DOI: 10.1016/j.procir.2021.03.112.

[Cui20] Yesheng Cui, Sami Kara, and Ka C. Chan. "Manufacturing Big Data Ecosystem: A Systematic Literature Review". In: *Robotics and Computer-Integrated Manufacturing* 62 (2020), p. 101861. ISSN: 07365845. DOI: 10.1016/j.rcim.2019.101861.

[Cuyp09] W. Cuypers, N. Van Gestel, A. Voet, et al. "Optical Measurement Techniques for Mobile and Large-Scale Dimensional Metrology". In: *Optics and Lasers in Engineering*. Optical Measurements 47.3 (Mar. 1, 2009), pp. 292–300. ISSN: 0143-8166. DOI: 10.1016/j.optlaseng.2008.03.013.

[Cyga14] Richard Cyganiak, David Wood, Markus Lanthaler, et al. *RDF 1.1 Concepts and Abstract Syntax*. RDF 1.1 Concepts and Abstract Syntax. Feb. 25, 2014. URL: https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ (visited on Mar. 24, 2024).

[Czar21] Damian A. Czarny, Johannes Diemer, Mario Schacht, et al. *Scenarios for Digitizing Standardization and Standards*. June 2021. URL: https://www.din.de/resource/blob/801106/0251eb1280a9a97e53285d42d3bf1fea/whitepaper-idis-en-data.pdf (visited on Nov. 1, 2023).

[Damm09] Matthias Damm, Stefan-Helmut Leitner, and Wolfgang Mahnke. *OPC Unified Architecture*. Springer Berlin, Heidelberg, Mar. 19, 2009. 339 pp. ISBN: 978-3-540-68898-3.

[Data20] Data Documentation Initiative. *DDI Lifecycle 3.3*. Apr. 15, 2020. URL: https://ddialliance.org/Specification/DDI-Lifecycle/3.3/ (visited on Mar. 27, 2024).

[DCMI20] DCMI Usage Board. *DCMI Metadata Terms*. Jan. 20, 2020. URL: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/ (visited on Nov. 1, 2023).

[Derh17a] Hasan Derhamy, Jens Eliasson, and Jerker Delsing. "IoT Interoperability—On-Demand and Low Latency Transparent Multiprotocol Translator". In: *IEEE Internet of Things Journal* 4.5 (Oct. 2017), pp. 1754–1763. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2697718.

[Derh17b] Hasan Derhamy, Jesper Rönnholm, Jerker Delsing, Jens Eliasson, and Jan van Deventer. "Protocol Interoperability of OPC UA in Service Oriented Architectures". In: *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). July 2017, pp. 44–50. DOI: 10.1109/INDIN.2017.8104744.

[Deva21] Anusuriya Devaraju and Robert Huber. "An Automated Solution for Measuring the Progress toward FAIR Research Data". In: *Patterns (New York, N.Y.)* 2.11 (2021), p. 100370. DOI: 10.1016/j.patter.2021.100370.

[Deva23] Anusuriya Devaraju and Robert Huber. *F-UJI - An Automated FAIR Data Assessment Tool*. Version v3.1.0. Nov. 10, 2023. DOI: 10.5281/ZENODO.6361400.

[Deva22] Anusuriya Devaraju, Robert Huber, Mustapha Mokrane, et al. "FAIRsFAIR Data Object Assessment Metrics". In: (Apr. 14, 2022). DOI: 10.5281/zenodo.6461229.

[Dors23] Tanja Dorst, Maximilian Gruber, Anupam P. Vedurmudi, et al. "A Case Study on Providing FAIR and Metrologically Traceable Data Sets". In: *Acta IMEKO* 12.1 (1 Mar. 28, 2023), pp. 1–6. ISSN: 2221-870X. DOI: 10.21014/actaimeko.v12i1.1401.

[Drat23] Rainer Drath, Christoph Mosch, Stefan Hoppe, et al. *Diskussionspapier - Interoperabilität mit der Verwaltungsschale, OPC UA und AutomationML*. Apr. 11, 2023, p. 22. URL: https://www.automationml.org/wp-content/uploads/2023/04/Diskussionspapier-Zielbild-und-Handlungsempfehlungen-fuer-industrielle-Interoperabilitaet-5.3.pdf (visited on May 6, 2024).

[Ecli22] Eclipse Foundation. *Eclipse Data Space Connector*. 2022. URL: https://github.com/eclipse-edc/Connector (visited on July 9, 2024).

[Ecli24] Eclipse Foundation. *Xtext - Language Engineering Made Easy!* 2024. URL: https://eclipse.dev/Xtext/ (visited on June 6, 2024).

[Eich20] S. Eichstädt and B. Ludwig. "Metrology for Heterogeneous Sensor Networks and Industry 4.0". In: *at - Automatisierungstechnik* 68.6 (June 1, 2020), pp. 459–464. ISSN: 2196-677X. DOI: 10.1515/auto-2020-0059.

[Eich23] Sascha Eichstädt, Maximilian Gruber, and Anupam Prasad Vedurmudi. “Integrating Metrological Principles into the Internet of Things: A Digital Maturity Model for Sensor Network Metrology”. In: *tm - Technisches Messen* (Oct. 16, 2023). ISSN: 2196-7113. DOI: 10.1515/teme-2023-0103.

[Eich22] Sascha Eichstädt, Daniel Hutzschenreuter, J. Niederhausen, and J. Neumann. “The Quality Infrastructure in the Digital Age: Beyond Machine-Readable Documents”. In: *Proceedings of IMEKO M4Dconf2022, Berlin, Germany* 10 (2022). DOI: 10.21014/tc6-2022.042.

[ELIX21] ELIXIR. *Research Data Management Kit. A Deliverable from the EU-funded ELIXIR-CONVERGE Project (Grant Agreement 871075)*. Research Data Management Kit. A deliverable from the EU-funded ELIXIR-CONVERGE project (grant agreement 871075). 2021. URL: https://rdmkit.elixir-europe.org (visited on Feb. 18, 2024).

[Elst13] Atis Elsts, Janis Judvaitis, and Leo Selavo. “SEAL: A Domain-Specific Language for Novice Wireless Sensor Network Programmers”. In: *2013 39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2013)*. Ed. by Onur Demirors. Piscataway, NJ: IEEE, 2013, pp. 220–227. ISBN: 978-0-7695-5091-6. DOI: 10.1109/SEAA.2013.16.

[Emon20] Vincent Emonet. *Data Identifier Explicitly in Metadata*. 2020. URL: https://metrics.api.fair-enough.semanticscience.org/tests/f3-data-identifier-in-metadata (visited on Oct. 6, 2024).

[Emon24] Vincent Emonet and Michel Dumontier. *FAIR Enough*. Maastricht University IDS, May 8, 2024. URL: https://github.com/MaastrichtU-IDS/fair-enough (visited on June 29, 2024).

[Emon21] Dominik Emonts, Philipp Dahlem, Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt. “Virtuelle Klimatisierung/Virtual Climatization”. In: *wt Werkstattstechnik online* 111.11-12 (Nov. 30, 2021), pp. 887–892. ISSN: 1436-4980. DOI: 10.37544/1436-4980-2021-11-12-117.

[Ens24] Davis Ens, Oliver Rostig, Susanna Weber, and Tobias Westphal. *SOIL2OPC UA Translator*. Version 1.0.0. Aachen: RWTH Aachen University, Oct. 12, 2024. DOI: 10.5281/zenodo.13923657.

[Even06] A. Even, G. Shankaranarayanan, and S. Watts. “Enhancing Decision Making with Process Metadata: Theoretical Framework, Research Tool, and Exploratory Examination”. In: *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS’06)*. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS’06). Vol. 8. Jan. 2006, 209a–209a. DOI: 10.1109/HICSS.2006.152.

[Ever15] Lars Evertz and Ulrich Epple. “Semi-Automatic Development of Service Adaptors from Property-Based Service Descriptions”. In: *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*. Piscataway, NJ: IEEE, 2015, pp. 1–5. ISBN: 978-1-4673-7929-8. DOI: 10.1109/ETFA.2015.7301544.

[Fahe19] Muhammad Rehan Faheem, Tayyaba Anees, and Muzammil Hussain. "The Web of Things: Findability Taxonomy and Challenges". In: *IEEE Access* 7 (2019), pp. 185028–185041. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2960446.

[Fall04] David C. Fallside and Priscilla Walmsley. *XML Schema Part 0: Primer Second Edition*. XML Schema Part 0: Primer Second Edition. Oct. 28, 2004. URL: https://www.w3.org/TR/xmlschema-0/ (visited on Oct. 12, 2024).

[Fang06] Hui Fang and ChengXiang Zhai. "Semantic Term Matching in Axiomatic Approaches to Information Retrieval". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06. New York, NY, USA: Association for Computing Machinery, Aug. 6, 2006, pp. 115–122. ISBN: 978-1-59593-369-0. DOI: 10.1145/1148170.1148193.

[Fett05] Peter Fettke. "Unified Modeling Language". In: *Encyclopedia of Information Science and Technology, First Edition*. IGI Global, 2005, pp. 2921–2928. ISBN: 978-1-59140-553-5. DOI: 10.4018/978-1-59140-553-5.ch520.

[Fieb19] Bernd Fiebiger. *OPC 400010-1 - OPC UA for Robotics - Vertical Integration*. Version 1.0. Frankfurt am Main, May 13, 2019. URL: https://opcfoundation.org/developer-tools/documents/view/218 (visited on June 9, 2024).

[Fiel00] Roy Thomas Fielding. "Architectural Styles and the Design of Network-based Software Architectures". Irvine: University of California, 2000. 180 pp. URL: https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf (visited on June 2, 2024).

[Ford07] Thomas C. Ford, John M. Colombi, Scott R. Graham, and David R. Jacques. *Survey on Interoperability Measurement*. June 1, 2007. URL: https://apps.dtic.mil/sti/citations/ADA481314 (visited on Nov. 2, 2023).

[Forr24] Forrester Consulting. *Advanced Manufacturing Report*. Hexagon, Mar. 7, 2024, p. 21. URL: https://hexagon.com/advanced-manufacturing-report (visited on June 8, 2024).

[Fowl11] Martin Fowler and Rebecca Parsons. *Domain-Specific Languages*. A Martin Fowler Signature Book. Upper Saddle River, NJ et al.: Addison-Wesley, 2011. ISBN: 0-321-71294-3.

[Fran17] E. Francalanza, J. Borg, and C. Constantinescu. "A Knowledge-Based Tool for Designing Cyber Physical Production Systems". In: *Computers in Industry* 84 (Jan. 1, 2017), pp. 39–58. ISSN: 0166-3615. DOI: 10.1016/j.compind.2016.08.001.

[Frie20] Sebastian Friedl, Christian von Arnim, Armin Lechler, and Alexander Verl. "Generation of OPC UA Companion Specification with Eclipse Modeling Framework". In: *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*. 2020 16th IEEE International Conference on Factory Communication Systems (WFCS). Apr. 2020, pp. 1–7. DOI: 10.1109/WFCS47810.2020.9114448.

[Fuhr24] Marc Fuhrmans and Benedikt Heinrichs. *About - AIMS Documentation*. Apr. 2024. URL: https://aims.pages.rwth-aachen.de/public/documentation/user-guide/platform/about/ (visited on June 2, 2024).

[Al-F15] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2347–2376. DOI: 10.1109/COMST.2015.2444095.

[Gaig23] Alban Gaignard, Thomas Rosnet, Frédéric De Lamotte, Vincent Lefort, and Marie-Dominique Devignes. "FAIR-Checker: Supporting Digital Resource Findability and Reuse with Knowledge Graphs and Semantic Web Standards". In: *Journal of Biomedical Semantics* 14.1 (July 1, 2023), p. 7. ISSN: 2041-1480. DOI: 10.1186/s13326-023-00289-5.

[Gand14] Fabien Gandon, Guus Schreiber, and Dave Becket. *RDF 1.1 XML Syntax*. RDF 1.1 XML Syntax. Feb. 25, 2014. URL: https://www.w3.org/TR/rdf-syntax-grammar/ (visited on Sept. 22, 2024).

[Gao20] Robert X. Gao, Lihui Wang, Moneer Helu, and Roberto Teti. "Big Data Analytics for Smart Factories of the Future". In: *CIRP Annals* 69.2 (Jan. 1, 2020), pp. 668–692. ISSN: 0007-8506. DOI: 10.1016/j.cirp.2020.05.002.

[Geis12] Eva Geisberger and Manfred Broy. *AgendaCPS: Integrierte Forschungsagenda ; Cyber-Physical Systems*. Vol. 1. Acatech Studie. Berlin and Heidelberg: Springer, 2012. ISBN: 978-3-642-29098-5. DOI: 10.1007/978-3-642-29099-2.

[Give17] Omid Givehchi, Klaus Landsdorf, Pieter Simoens, and Armando Walter Colombo. "Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems". In: *IEEE Transactions on Industrial Informatics* 13.6 (Dec. 2017), pp. 3370–3378. ISSN: 1941-0050. DOI: 10.1109/TII.2017.2740434.

[Glei20] Lars Gleim, Jan Pennekamp, Martin Liebenberg, et al. "FactDAG: Formalizing Data Interoperability in an Internet of Production". In: *IEEE Internet of Things Journal* 7.4 (2020), pp. 3243–3253. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.2966402.

[Glei21] Lars Gleim, Liam Tirpitz, and Stefan Decker. "FactStack: Interoperable Data Management and Preservation for the Web and Industry 4.0". In: (2021). ISSN: 1617-5468. DOI: 10.18420/BTW2021-20.

[Gold12] Thomas Goldschmidt and Wolfgang Mahnke. "Evaluating Domain-Specific Languages for the Development of OPC UA Based Applications". In: *IFAC Proceedings Volumes*. 7th Vienna International Conference on Mathematical Modelling 45.2 (Jan. 1, 2012), pp. 860–865. ISSN: 1474-6670. DOI: 10.3182/20120215-3-AT-3016.00152.

[Goog24a] Google. *Introduction to gRPC*. gRPC. 2024. URL: https://grpc.io/docs/what-is-grpc/introduction/ (visited on June 2, 2024).

[Goog24b] Google. *Protocol Buffers - Overview*. 2024. URL: https://protobuf.dev/overview/ (visited on June 2, 2024).

[Göpp23] Amon Göppert, Lea Grahn, Jonas Rachner, et al. “Pipeline for Ontology-Based Modeling and Automated Deployment of Digital Twins for Planning and Control of Manufacturing Systems”. In: *Journal of Intelligent Manufacturing* 34.5 (June 2023), pp. 2133–2152. ISSN: 0956-5515, 1572-8145. DOI: 10.1007/s10845-021-01860-6.

[Grab19] Sam Grabus and Jane Greenberg. “The Landscape of Rights and Licensing Initiatives for Data Sharing”. In: *Data Science Journal* 18.1 (July 4, 2019), p. 29. ISSN: 1683-1470. DOI: 10.5334/dsj-2019-029.

[Grön23] Matthias Grönewald and Nils Preuß. “Mit maßgeschneiderten Metadatenprofilen zu validierten und nachhaltigen Forschungsdaten”. In: *E-Science-Tage 2023: Empower Your Research – Preserve Your Data*. E-Science-Tage 2023: Empower Your Research – Preserve Your Data. Heidelberg: heiBOOKS, 2023, pp. 220–227. DOI: 10.11588/heibooks.1288.c18080.

[Grot20] Paul Groth, Helena Cousijn, Tim Clark, and Carole Goble. “FAIR Data Reuse – the Path through Data Citation”. In: *Data Intelligence* 2.1-2 (2020), pp. 78–86. DOI: 10.1162/dint_a_00030.

[Guiz20] Giancarlo Guizzardi. “Ontology, Ontologies and the “I” of FAIR”. In: *Data Intelligence* 2.1-2 (2020), pp. 181–191. DOI: 10.1162/dint_a_00040.

[Gürd18] Didem Gürdür and Fredrik Asplund. “A Systematic Review to Merge Discourses: Interoperability, Integration and Cyber-Physical Systems”. In: *Journal of Industrial Information Integration* 9 (2018), pp. 14–23. ISSN: 2452-414X. DOI: 10.1016/j.jii.2017.12.001.

[Gürd16] Didem Gürdür, Fredrik Asplund, and Jad El-khoury. “Measuring Tool Chain Interoperability in Cyber-Physical Systems”. In: *2016 11th System of Systems Engineering Conference (SoSE)*. 2016 11th System of Systems Engineering Conference (SoSE). June 2016, pp. 1–4. DOI: 10.1109/SYSOSE.2016.7542964.

[Hack18] Siegfried Gustav Hackel, Frank Härtig, Julia Hornig, and Thomas Wiedenhöfer. “The Digital Calibration Certificate: 399916 Bytes / PTB-Mitteilungen. Volume 127 (2017), Issue 4, Page 75 to 81. ISSN 0030-834X”. In: (2018). DOI: 10.7795/310.20170403.

[Hall18] Armin Haller, Krzysztof Janowicz, Simon J.D. Cox, et al. “The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation”. In: *Semantic Web* 10.1 (2018), pp. 9–32. ISSN: 15700844. DOI: 10.3233/SW-180320.

[Häne20] Albrecht Hänel, Thorben Schnellhardt, Eric Wenkler, et al. “The Development of a Digital Twin for Machining Processes for the Application in Aerospace Industry”. In: *Procedia CIRP*. 53rd CIRP Conference on Manufacturing Systems 2020 93 (Jan. 1, 2020), pp. 1399–1404. ISSN: 2212-8271. DOI: 10.1016/j.procir.2020.04.017.

[Harp13] Patricia Harpring and Murtha Baca. *Introduction to Controlled Vocabularies: Terminology for Art, Architecture, and Other Cultural Works*. Updated edition = Second edition. Introduction To. Los Angeles, California: Getty Research Institute, 2013. 247 pp. ISBN: 978-1-60606-150-3.

[Hart24] Stefanie Hartl. *Statement Zum Eckpunktepapier Des Bundesministeriums Für Bildung Und Forschung (BMBF) Zum Forschungsdatengesetz*. Apr. 16, 2024. URL: https://www.nfdi.de/statement-zum-eckpunktepapier-des-bundesministeriums-fuer-bildung-und-forschung-bmbf-zum-forschungsdatengesetz/ (visited on June 8, 2024).

[Heid19] Roland Heidel, Michael Hoffmeister, Martin Hankel, and Udo Döbrich. *Industrie 4.0: the reference architecture model RAMI 4.0 and the Industrie 4.0 component*. 1st edition. Berlin Wien Zürich: Beuth Verlag GmbH, 2019. 150 pp. ISBN: 978-3-410-28919-7.

[Hein24] Benedikt Paul Anton Heinrichs, Matthias S. Müller, and Thomas Stäcker. "Asynchronous Tracking and Description of Research Data Changes in Distributed Systems with Interoperable Metadata". Apprimus Verlag, 2024. ISBN: 9783985552146.

[Hevn10] Alan Hevner and Samir Chatterjee. "Design Science Research in Information Systems". In: *Design Research in Information Systems*. Ed. by Alan Hevner and Samir Chatterjee. Integrated Series in Information Systems. Boston, MA: Springer Science+Business Media LLC, 2010, pp. 9–22. ISBN: 978-1-4419-5653-8. DOI: 10.1007/978-1-4419-5653-8_2.

[Hevn07] Alan R. Hevner. "A Three Cycle View of Design Science Research". In: *Scandinavian journal of information systems* 19.2 (2007), p. 4. URL: https://aisel.aisnet.org/sjis/vol19/iss2/4.

[Higg08] Sarah Higgins. "The DCC Curation Lifecycle Model". In: *International Journal of Digital Curation* 3.1 (1 Dec. 2, 2008), pp. 134–140. ISSN: 1746-8256. DOI: 10.2218/ijdc.v3i1.48.

[Hild20] Constantin Hildebrandt, Aljosha Kocher, Christof Kustner, et al. "Ontology Building for Cyber–Physical Systems: Application in the Manufacturing Domain". In: *IEEE Transactions on Automation Science and Engineering* 17.3 (2020), pp. 1266–1282. ISSN: 1545-5955. DOI: 10.1109/TASE.2020.2991777.

[Horv19] Djerdj Horvat, Henning Kroll, and Angela Jäger. "Researching the Effects of Automation and Digitalization on Manufacturing Companies' Productivity in the Early Stage of Industry 4.0". In: *Procedia Manufacturing*. 25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing August 9-14, 2019 | Chicago, Illinois (USA) 39 (Jan. 1, 2019), pp. 886–893. ISSN: 2351-9789. DOI: 10.1016/j.promfg.2020.01.401.

[Huba13] Thomas Hubauer, Steffen Lamparter, Mikhail Roshchin, Nina Solomakhina, and Stuart Watson. "Analysis of Data Quality Issues in Real-World Industrial Data". In: *Annual Conference of the PHM Society* 5.1 (2013). ISSN: 2325-0178. DOI: 10.36001/phmconf.2013.v5i1.2198.

[Hürs95] Walter L. Hürsch and Cristina Videira Lopes. "Separation of Concerns". In: (Feb. 1995). URL: ftp://ftp.ccs.neu.edu/pub/people/lieber/crista/techrep95/index.html (visited on June 30, 2024).

[Hutz20] Daniel Hutzschenreuter, Frank Härtig, Heeren Wiebke, et al. “SmartCom Digital System of Units (D-SI) Guide for the Use of the Metadata-Format Used in Metrology for the Easy-to-Use, Safe, Harmonised and Unambiguous Digital Transfer of Metrological Data - Second Edition”. Version D-SI 1.3.0-2. In: (July 1, 2020). DOI: 10.5281/ZENODO.3816686.

[Igle24] Dorothea Iglezakis, Giacomo Lanza, Susanne Arndt, et al. *Metadata4Ing: An Ontology for Describing the Generation of Research Data within a Scientific Activity.* Version 1.3.0. Sept. 20, 2024. DOI: 10.5281/zenodo.5957103.

[Isel16] Robert Isele and Natanael Arndt. “Mit Semantischer Datenverwaltung Big Data in Den Griff Bekommen”. In: *Wirtschaftsinformatik & Management* 8.4 (2016), pp. 56–63. ISSN: 1867-5913. DOI: 10.1007/s35764-016-0065-z.

[ISO20] ISO. *ISO 23952:2020 - Automation Systems and Integration Quality Information Framework (QIF)*. Version 2020. July 2020. URL: https://www.iso.org/standard/77461.html. Published.

[ISO21] ISO. *ISO/IEC 19075-6:2021 - Information Technology — Guidance for the Use of Database Language SQL, Part 6: Support for JSON*. Version 1. Aug. 2021. URL: https://www.iso.org/standard/78937.html (visited on June 30, 2024). Published.

[ISO22] ISO. *ISO 80000-1:2022, Quantities and Units, Part 1: General*. Version 2. Dec. 2022. URL: https://www.iso.org/standard/76921.html. Published.

[Jaco20a] Annika Jacobsen, Ricardo de Miranda Azevedo, Nick Juty, et al. “FAIR Principles: Interpretations and Implementation Considerations”. In: *Data Intelligence* 2.1-2 (Jan. 1, 2020), pp. 10–29. ISSN: 2641-435X. DOI: 10.1162/dint_r_00024.

[Jaco20b] Annika Jacobsen, Rajaram Kaliyaperumal, Luiz Olavo Bonino da Silva Santos, et al. “A Generic Workflow for the Data FAIRification Process”. In: *Data Intelligence* 2.1-2 (Jan. 1, 2020), pp. 56–65. ISSN: 2641-435X. DOI: 10.1162/dint_a_00028.

[Jesc18] Sabina Jeschke, Christian Brecher, Houbing Song, and Danda B. Rawat, eds. *Industrial Internet of Things: Cybermanufacturing Systems*. Softcover reprint of the original 1st edition 2017. Springer Series in Wireless Technology. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-42558-0. DOI: 10.1007/978-3-319-42559-7_30.

[JetB24] JetBrains. *MPS: Entwickler für domänenspezifische Sprachen von JetBrains*. JetBrains. 2024. URL: https://www.jetbrains.com/de-de/mps/ (visited on June 6, 2024).

[Jin15] X. Jin, B.W. Wah, X. Cheng, and Y. Wang. “Significance and Challenges of Big Data Research”. In: *Big Data Research* 2.2 (2015), pp. 59–64. ISSN: 2214-5796. DOI: 10.1016/j.bdr.2015.01.006.

[Jirk17] Vaclav Jirkovsky, Marek Obitko, and Vladimir Marik. “Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration”. In: *IEEE Transactions on Industrial Informatics* 13.2 (2017), pp. 660–667. ISSN: 1551-3203. DOI: 10.1109/TII.2016.2596101.

[John06] Timothy L. Johnson and Mark E. Dausch. “SENSOR INFORMATICS FOR MANUFACTURING”. In: *IFAC Proceedings Volumes*. 12th IFAC Symposium on Information Control Problems in Manufacturing 39.3 (Jan. 1, 2006), pp. 125–130. ISSN: 1474-6670. DOI: 10.3182/20060517-3-FR-2903.00078.

[JUni24] JUnit Team. *JUnit 5*. 2024. URL: https://junit.org/junit5/ (visited on June 16, 2024).

[Kais13] Stephen Kaisler, Frank Armour, J. Alberto Espinosa, and William Money. “Big Data: Issues and Challenges Moving Forward”. In: *2013 46th Hawaii International Conference on System Sciences*. 2013 46th Hawaii International Conference on System Sciences. Jan. 2013, pp. 995–1004. DOI: 10.1109/HICSS.2013.645.

[Kali17] Danguolė Kalinauskaitė. “To Be Findable, Accessible, Interoperable and Reusable: Language Data and Technology Infrastructure for Supporting the FAIR Data Approach”. In: *16130073*. ICYRIME 2017. Vol. ICYRIME 2017 : proceedings of the symposium for young researchers in informatics, mathematics and engineering. Kaunas, Lithuania, Apr. 28, 2017, pp. 21–26. DOI: 20.500.12259/35989.

[Kapi16] Gayatri Kapil, Alka Agrawal, and R. A. Khan. “A Study of Big Data Characteristics”. In: *2016 International Conference on Communication and Electronics Systems (ICCES)*. 2016 International Conference on Communication and Electronics Systems (ICCES). Oct. 2016, pp. 1–4. DOI: 10.1109/CESYS.2016.7889917.

[Kell20] Gregg Kellog, Pierre-Antoine Champin, Dave Longley, et al. *JSON-LD 1.1 - A JSON-based Serialization for Linked Data*. JSON-LD 1.1 - A JSON-based Serialization for Linked Data. July 16, 2020. URL: https://www.w3.org/TR/json-ld11/ (visited on Sept. 22, 2024).

[Kess20] René Kessler and Jorge Marx Gómez. “Implikationen von Machine Learning Auf Das Datenmanagement in Unternehmen”. In: *HMD Praxis der Wirtschaftsinformatik* 57.1 (2020), pp. 89–105. ISSN: 2198-2775. DOI: 10.1365/s40702-020-00585-z.

[Khan16] Ateeq Khan and Klaus Turowski. “A Survey of Current Challenges in Manufacturing Industry and Preparation for Industry 4.0”. In: *Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’16)*. Ed. by Ajith Abraham, Sergey Kovalev, Valery Tarassov, and Václav Snášel. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2016, pp. 15–26. ISBN: 978-3-319-33609-1. DOI: 10.1007/978-3-319-33609-1_2.

[Kirc22] Jörg Christian Kirchhof, Bernhard Rumpe, David Schmalzing, and Andreas Wortmann. “MontiThings: Model-Driven Development and Deployment of Reliable IoT Applications”. In: *Journal of Systems and Software* 183 (2022), p. 111087. ISSN: 01641212. DOI: 10.1016/j.jss.2021.111087.

[Knub17] Holger Knublauch and Dimitris Kontokostas. *Shapes Constraint Language (SHACL)*. July 20, 2017. URL: https://www.w3.org/TR/shacl/ (visited on Oct. 12, 2024).

[Krot15] Marina Krotofil, Jason Larsen, and Dieter Gollmann. “The Process Matters: Ensuring Data Veracity in Cyber-Physical Systems”. In: *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ASIA CCS ’15. New York, NY, USA: Association for Computing Machinery, Apr. 14, 2015, pp. 133–144. ISBN: 978-1-4503-3245-3. DOI: 10.1145/2714576.2714599.

[Kubi77] Herbert Kubicek. “Heuristische Bezugsrahmen und heuristisch angelegte Forschungsdesign als Elemente einer Konstruktionsstrategie empirischer Forschung”. In: *Empirische und handlungstheoretische Forschungskonzeptionen in der Betriebswirtschaftslehre*. Ed. by Köhler, Richard. Stuttgart: C. E. Poeschel, 1977, pp. 3–36. ISBN: 978-3-88398-016-4.

[Kuma21] Narender Kumar, Girish Kumar, and Rajesh Kumar Singh. “Big Data Analytics Application for Sustainable Manufacturing Operations: Analysis of Strategic Factors”. In: *Clean Technologies and Environmental Policy* 23.3 (Apr. 1, 2021), pp. 965–989. ISSN: 1618-9558. DOI: 10.1007/s10098-020-02008-5.

[Kusi19] Andrew Kusiak. “Fundamentals of Smart Manufacturing: A Multi-Thread Perspective”. In: *Annual Reviews in Control* 47 (2019), pp. 214–220. ISSN: 13675788. DOI: 10.1016/j.arcontrol.2019.02.001.

[Kuzn20] Arnold Kuzniar, Rajaram Kaliyaperumal, Carlos Martinez-Ortiz, and Cunliang Geng. *FAIR Data Point*. Zenodo, 2020. DOI: 10.5281/zenodo.3894151.

[Laba20] Clément Labadie, Christine Legner, Markus Eurich, and Martin Fadler. “FAIR Enough? Enhancing the Usage of Enterprise Data with Data Catalogs”. In: *2020 IEEE 22nd Conference on Business Informatics (CBI)*. 2020 IEEE 22nd Conference on Business Informatics (CBI). Vol. 1. June 2020, pp. 201–210. DOI: 10.1109/CBI49978.2020.00029.

[Land20] Annalisa Landi, Mark Thompson, Viviana Giannuzzi, et al. “The “A” of FAIR – As Open as Possible, as Closed as Necessary”. In: *Data Intelligence* 2.1-2 (2020), pp. 47–55. DOI: 10.1162/DINT_A_00027.

[Lang24] Ilona Lang, Marcel Nellesen, and Marius Politze. “RDM Platform Coscine - FAIR Play Integrated Right from the Start”. In: *ing.grid* 2.1 (1 Apr. 30, 2024). ISSN: 2941-1300. DOI: 10.48694/inggrid.3952.

[Lebo13] Timothy Lebo, Satya Sahoo, Deborah McGuinness, et al. *PROV-O: The PROV Ontology*. Apr. 30, 2013. URL: https://www.w3.org/TR/prov-o/ (visited on Oct. 15, 2024).

[Li22] Chunquan Li, Yaqiong Chen, and Yuling Shang. “A Review of Industrial Big Data for Decision Making in Intelligent Manufacturing”. In: *Engineering Science and Technology, an International Journal* 29 (2022), p. 101021. ISSN: 2215-0986. DOI: 10.1016/j.jestch.2021.06.001.

[Li21] Guoyan Li, Chenxi Yuan, Sagar Kamarthi, Mohsen Moghaddam, and Xiaoning Jin. “Data Science Skills and Domain Knowledge Requirements in the Manufacturing Industry: A Gap Analysis”. In: *Journal of Manufacturing Systems* 60 (July 1, 2021), pp. 692–706. ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2021.07.007.

[Li15] Jingran Li, Fei Tao, Ying Cheng, and Liangjin Zhao. “Big Data in Product Lifecycle Management”. In: *The International Journal of Advanced Manufacturing Technology* 81.1 (Oct. 1, 2015), pp. 667–684. ISSN: 1433-3015. DOI: 10.1007/s00170-015-7151-x.

[Li20] Jun Li, Jun-Jiang Qiu, Yong Zhou, et al. “Study on the Reference Architecture and Assessment Framework of Industrial Internet Platform”. In: *IEEE Access* 8 (2020), pp. 164950–164971. DOI: 10.1109/ACCESS.2020.3021719.

[Li24] Xiaobin Li, Shucheng Zhang, Pei Jiang, et al. “Knowledge Graph Based OPC UA Information Model Automatic Construction Method for Heterogeneous Devices Integration”. In: *Robotics and Computer-Integrated Manufacturing* 88 (Aug. 1, 2024), p. 102736. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2024.102736.

[Lian16] Steve Liang, Chih-Yuan Huang, and Tania Khalafbeigi, eds. *OGC SensorThings API Part 1: Sensing, Version 1.0*. Open Geospatial Consortium, 2016. DOI: 10.25607/OBP-455.

[Libe15] Don Libes, Seungjun Shin, and Jungyub Woo. “Considerations and Recommendations for Data Availability for Data Analytics for Manufacturing”. In: *2015 IEEE International Conference on Big Data (Big Data)*. 2015 IEEE International Conference on Big Data (Big Data). Oct. 2015, pp. 68–75. DOI: 10.1109/BigData.2015.7363743.

[Liu19] Chao Liu, Hrishikesh Vengayil, Yuqian Lu, and Xun Xu. “A Cyber-Physical Machine Tools Platform Using OPC UA and MTConnect”. In: *Journal of Manufacturing Systems* 51 (Apr. 1, 2019), pp. 61–74. ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2019.04.006.

[Liu14] Shixiong Liu, Weizi Li, and Kecheng Liu. “Pragmatic Oriented Data Interoperability for Smart Healthcare Information Systems”. In: *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. May 2014, pp. 811–818. DOI: 10.1109/CCGrid.2014.38.

[Loga23] Kevin T. Logan, Marvin M. Meck, Nils Preuß, Philipp Wetterich, and Peter F. Pelz. “Data Management as an Enabler of Sustainability – Discussion Using the Example of a Digital Data Sheet; 2nd Revised Edition”. In: *Fluid power: digital* reliable (2023), pages 813–827. DOI: 10.18154/RWTH-2023-04624.

[Lomo14] Richard K. Lomotey and Ralph Deters. “Towards Knowledge Discovery in Big Data”. In: *2014 IEEE 8th International Symposium on Service Oriented System Engineering*. 2014 IEEE 8th International Symposium on Service Oriented System Engineering. Apr. 2014, pp. 181–191. DOI: 10.1109/SOSE.2014.25.

[Luko13] Tatiana Lukoianova and Victoria L. Rubin. “Veracity Roadmap: Is Big Data Objective, Truthful and Credible?” In: *Advances in Classification Research Online* (2013), pp. 4–15. ISSN: 2324-9773. DOI: 10.7152/acro.v24i1.14671.

[Ma20] Shuaiyin Ma, Yingfeng Zhang, Yang Liu, et al. “Data-Driven Sustainable Intelligent Manufacturing Based on Demand Response for Energy-Intensive Industries”. In: *Journal of Cleaner Production* 274 (Nov. 20, 2020), p. 123155. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2020.123155.

[Ma22] Shuaiyin Ma, Yingfeng Zhang, Jingxiang Lv, et al. “Data-Driven Cleaner Production Strategy for Energy-Intensive Manufacturing Industries: Case Studies from Southern and Northern China”. In: *Advanced Engineering Informatics* 53 (Aug. 1, 2022), p. 101684. ISSN: 1474-0346. DOI: 10.1016/j.aei.2022.101684.

[Maci17] Rita Suzana Maciel, José Maria David, Daniela Claro, and Regina Braga. “Full Interoperability: Challenges and Opportunities for Future Information Systems”. In: *I GranDSI-BR: Grandes Desafios Da Pesquisa Em Sistemas de Informação No Brasil Para o Período de 2016 a 2026*. SBC, 2017, pp. 107–118. DOI: 10.5753/sbc.2884.0.9.

[Made23] Isabella Mader. “Die Welt des Wissens in Organisationen im Wandel”. In: *Wissensmanagement erfolgreich umsetzen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023, pp. 9–44. ISBN: 978-3-662-66762-0. DOI: 10.1007/978-3-662-66763-7_2.

[Magu22] Conor Maguire, Abraham George, Matthew Kelly, et al. “An Information Model with Voxelated Data from Computer Aided Systems and the Additive Manufacturing Workflow”. In: *Procedia CIRP*. 15th CIRP Conference on Intelligent Computation in ManufacturingEngineering, 14-16 July 2021 112 (Jan. 1, 2022), pp. 382–387. ISSN: 2212-8271. DOI: 10.1016/j.procir.2022.09.013.

[Maje21] Arfan Majeed, Yingfeng Zhang, Shan Ren, et al. “A Big Data-Driven Framework for Sustainable and Smart Additive Manufacturing”. In: *Robotics and Computer-Integrated Manufacturing* 67 (Feb. 1, 2021), p. 102026. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2020.102026.

[Math23] Julius B. Mathews, Jonas Rachner, Lea Kaven, et al. “Industrial Applications of a Modular Software Architecture for Line-Less Assembly Systems Based on Interoperable Digital Twins”. In: *Frontiers in Mechanical Engineering* 9 (Feb. 28, 2023). ISSN: 2297-3079. DOI: 10.3389/fmech.2023.1113933.

[Maye17] Simon Mayer, Jack Hodges, Dan Yu, Mareike Kritzler, and Florian Michahelles. “An Open Semantic Framework for the Industrial Internet of Things”. In: *IEEE Intelligent Systems* 32.1 (Jan. 2017), pp. 96–101. ISSN: 1941-1294. DOI: 10.1109/MIS.2017.9.

[McCo23] Michael McCool, Michael Lagally, Tomoaki Mizushima, et al. *Web of Things (WoT) Profile*. Jan. 18, 2023. URL: https://www.w3.org/TR/wot-profile/ (visited on Oct. 16, 2024).

[McKe12] Katy McKen, Catherine Pink, Liz Lyon, and Matthew Davidson. *Research360: Data in the Research Lifecycle*. June 11, 2012. URL: https://researchportal.bath.ac.uk/en/publications/research360-data-in-the-research-lifecycle (visited on Oct. 16, 2024).

[Mern05] Marjan Mernik, Jan Heering, and Anthony M. Sloane. “When and How to Develop Domain-Specific Languages”. In: *ACM Computing Surveys* 37.4 (2005), pp. 316–344. ISSN: 03600300. DOI: 10.1145/1118890.1118892.

[Mich21] Judith Michael and Andreas Wortmann. “Towards Development Platforms for Digital Twins: A Model-Driven Low-Code Approach”. In: *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*. Ed. by Alexandre Dolgui, Alain Bernard, David Lemoine, Gregor von Cieminski, and David Romero. Cham: Springer International Publishing, 2021, pp. 333–341. ISBN: 978-3-030-85874-2. DOI: 10.1007/978-3-030-85874-2_35.

[Mich12] William K. Michener and Matthew B. Jones. “Ecoinformatics: Supporting Ecology as a Data-Intensive Science”. In: *Trends in Ecology & Evolution*. Ecological and Evolutionary Informatics 27.2 (Feb. 1, 2012), pp. 85–93. ISSN: 0169-5347. DOI: 10.1016/j.tree.2011.11.016.

[Miel15] Michael Mielke. “Prinzipien Erfolgreichen Informationsqualitätsmanagements Im Lichte von Industrie 4.0”. In: *Daten- Und Informationsqualität*. Ed. by Knut Hildebrand, Marcus Gebauer, Holger Hinrichs, and Michael Mielke. Wiesbaden: Springer Vieweg, 2015, pp. 313–323. ISBN: 978-3-658-09214-6. DOI: 10.1007/978-3-658-09214-6âĂŮ19.

[Mohr15] Peter J. Mohr, David B. Newell, and Barry N. Taylor. *CODATA Recommended Values of the Fundamental Physical Constants: 2014*. Zenodo, Aug. 4, 2015. DOI: 10.5281/zenodo.22826.

[Möll13] Knud Möller. “Lifecycle Models of Data-Centric Systems and Domains”. In: *Semantic Web* 4.1 (Jan. 1, 2013), pp. 67–88. ISSN: 1570-0844. DOI: 10.3233/SW-2012-0060.

[Mont21] Benjamin Montavon. *Virtual Reference Frame Based on Distributed Large-Scale Metrology Providing Coordinates as a Service*. Aachen: Apprimus Verlag, 2021. ISBN: 978-3-9855501-0-4.

[Mont23] Benjamin Montavon and Matthias Bodenbenner. “Messdaten verarbeiten, verteilen, nutzen - aktuelle Entwicklungen”. In: *Handbuch Messtechnik in der industriellen Produktion : valide Messergebnisse planen, erhalten, auswerten und verteilen*. 1st ed. Carl Hanser Verlag GmbH & Co. KG, Aug. 2023, pp. 673–690. ISBN: 978-3-446-46559-6.

[Mont19] Benjamin Montavon, Martin Peterek, and Robert H. Schmitt. “Model-Based Interfacing of Large-Scale Metrology Instruments”. In: *Multimodal Sensing: Technologies and Applications*. Ed. by Ettore Stella. Proceedings of SPIE. 5200-. Bellingham, Washington: SPIE, 2019, p. 11. ISBN: 978-1-5106-2797-0. DOI: 10.1117/12.2527461.

[Moor15] Janakiraman Moorthy, Rangin Lahiri, Neelanjan Biswas, et al. “Big Data: Prospects and Challenges”. In: *Vikalpa* 40.1 (Mar. 1, 2015), pp. 74–96. ISSN: 0256-0909. DOI: 10.1177/0256090915575450.

[More13] Luc Moreau and Paolo Missier. *PROV-DM: The PROV Data Model*. Apr. 30, 2013. URL: https://www.w3.org/TR/prov-dm/ (visited on Jan. 24, 2024).

[Muse15] Mark A. Musen, Carol A. Bean, Kei-Hoi Cheung, et al. “The Center for Expanded Data Annotation and Retrieval”. In: *Journal of the American Medical Informatics Association: JAMIA* 22.6 (Nov. 2015), pp. 1148–1152. ISSN: 1527-974X. DOI: 10.1093/jamia/ocv048.

[Muse22] Mark A. Musen, Martin J. O’Connor, Erik Schultes, et al. “Modeling Community Standards for Metadata as Templates Makes Data FAIR”. In: *Scientific Data* 9.1 (2022), p. 696. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01815-3.

[Must20] Tuukka Mustapaa, Juuso Autiosalo, Pekka Nikander, Joshua E. Siegel, and Raine Viitala. “Digital Metrology for the Internet of Things”. In: *GIoTS, Global IoT Summit*. Piscataway, NJ: IEEE, 2020, pp. 1–6. ISBN: 978-1-72816-728-2. DOI: 10.1109/GIOTS49054.2020.9119603.

[Nase13] Aisha Naseer, Loredana Laera, and Takahide Matsutsuka. “Enterprise Big-Graph”. In: *2013 46th Hawaii International Conference on System Sciences*. 2013 46th Hawaii International Conference on System Sciences. Jan. 2013, pp. 1005–1014. DOI: 10.1109/HICSS.2013.202.

[Nast20] Michael Nast, Benjamin Rother, Frank Golatowski, et al. “Work-in-Progress: Towards an International Data Spaces Connector for the Internet of Things”. In: *2020 16th IEEE International Conference on Factory Communication Systems (WFCS)*. 2020 16th IEEE International Conference on Factory Communication Systems (WFCS). Apr. 2020, pp. 1–4. DOI: 10.1109/WFCS47810.2020.9114503.

[Neub23] Michael Neubauer, Lukas Steinle, Colin Reiff, et al. “Architecture for Manufacturing-X: Bringing Asset Administration Shell, Eclipse Dataspace Connector and OPC UA Together”. In: *Manufacturing Letters* 37 (Sept. 1, 2023), pp. 1–6. ISSN: 2213-8463. DOI: 10.1016/j.mfglet.2023.05.002.

[Nguy22] Tu N. Nguyen, Sherali Zeadally, and Abhijith B. Vuduthala. “Cyber-Physical Cloud Manufacturing Systems With Digital Twins”. In: *IEEE Internet Computing*. IEEE Internet Computing. Vol. 26. May 2022, pp. 15–21. DOI: 10.1109/MIC.2021.3058921.

[Nika20] Pekka Nikander, Tommi Elo, Tuukka Mustapää, et al. *Document Specifying Rules for the Secure Use of DCC Covering Legal Aspects of Metrology*. Zenodo, Feb. 12, 2020. DOI: 10.5281/zenodo.3664211.

[Nils18] Jacob Nilsson and Fredrik Sandin. “Semantic Interoperability in Industry 4.0: Survey of Recent Developments and Outlook”. In: *Proceedings IEEE 16th International Conference on Industrial Informatics (INDIN)*. Piscataway, NJ: IEEE, 2018, pp. 127–132. ISBN: 978-1-5386-4829-2. DOI: 10.1109/INDIN.2018.8471971.

[OCUP17] OCUP 2 Examination Team. *Meta-Modeling and the OMG Meta Object Facility (MOF)*. Mar. 2017. URL: https://www.omg.org/ocup-2/documents/Meta-ModelingAndtheMOF.pdf (visited on June 4, 2024).

[Olaw23] Anifat M. Olawoyin, Carson K. Leung, Connor C. J. Hryhoruk, and Alfredo Cuzzocrea. “Big Data Management for Machine Learning from Big Data”. In: *Advanced Information Networking and Applications*. Ed. by Leonard Barolli. Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2023, pp. 393–405. ISBN: 978-3-031-29056-5. DOI: 10.1007/978-3-031-29056-5_35.

[Onto24] Ontology Engineering Group. *Linked Open Vocabularies (LOV)*. Jan. 23, 2024. URL: https://lov.linkeddata.es/dataset/lov/ (visited on Jan. 23, 2024).

[OPC 19] OPC Foundation. *Unified Architecture*. Sept. 2019. URL: https://opcfoundation.org/about/opc-technologies/opc-ua/ (visited on May 8, 2024).

[OPC 21] OPC Foundation. *DIN EN IEC 62541-5:2021-08, OPC Unified Architecture - Part 5: Information Model (IEC_62541-5:2020); Englische Fassung EN IEC 62541-5:2020*. Version 2021. Aug. 2021. DOI: 10.31030/3264496.

[OPC 24] OPC Foundation. *OPC UA - List of Documents*. June 9, 2024. URL: https://opcfoundation.org/developer-tools/documents (visited on June 9, 2024).

[Otto17a] Boris Otto and Steffen Lohmann. *Reference Architecture Model for the Industrial Data Space*. Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V., 2017. URL: https://www.fraunhofer.de/content/dam/zv/de/Forschungsfelder/industrial-data-space/Industrial-Data-Space_Reference-Architecture-Model-2017.pdf (visited on Nov. 15, 2023).

[Otto20] Boris Otto, Niko Mohr, Matthias Roggendorf, and Tobias Guggenberger. *Data Sharing in Industrial Ecosystems*. McKinsey & Company, Apr. 2020, p. 9. URL: https://www.mckinsey.de/~/media/mckinsey/locations/europe%20and%20middle%20east/deutschland/publikationen/data%20sharing%20in%20industrial%20ecosystems/mckinsey_article_data_sharing_in_industrial_ecosystems.pdf (visited on June 8, 2024).

[Otto17b] Boris Otto, Michael ten Hompel, and Stefan Wrobel. “Industrial Data Space”. In: *Digitalisierung*. Ed. by Reimund Neugebauer. Berlin, Heidelberg: Vieweg, 2017, pp. 113–133. ISBN: 978-3-662-55890-4. DOI: 10.1007/978-3-662-55890-4_8.

[Otto14] Jens Otto, Steffen Henning, and Oliver Niggemann. “Why Cyber-physical Production Systems Need a Descriptive Engineering Approach – A Case Study in Plug & Produce”. In: *Procedia Technology*. 2nd International Conference on System-Integrated Intelligence: Challenges for Product and Production Engineering 15 (Jan. 1, 2014), pp. 295–302. ISSN: 2212-0173. DOI: 10.1016/j.protcy.2014.09.083.

[Owai31] Suhail Sami Owais and Nada Sael Hussein. “Extract Five Categories CPIVW from the 9V’s Characteristics of the Big Data”. In: *International Journal of Advanced Computer Science and Applications (ijacsa)* 7.3 (3 2016/20/31). ISSN: 2156-5570. DOI: 10.14569/IJACSA.2016.070337.

[Oxfo23a] Oxford English Dictionary. *Controlled Vocabulary (n)*. In: *Oxford English Dictionary*. Oxford: Oxford University Press, Sept. 2023. DOI: 10.1093/OED/4086827516.

[Oxfo23b] Oxford English Dictionary. *Specific (Adj. & n.)* In: *Oxford English Dictionary*. Oxford: Oxford University Press, July 2023. DOI: 10.1093/OED/4780393804.

[Oxfo23c] Oxford English Dictionary. *Specificity (n.)* In: *Oxford English Dictionary*. Oxford: Oxford University Press, Sept. 2023. DOI: 10.1093/OED/4215072792.

[Pand19] Mohit Pandey, Shreyansh Tatiya, Shantanu Bhattacharya, and Shailendra Singh. "Sensors in Assembly Shop in Automobile Manufacturing". In: *Sensors for Automotive and Aerospace Applications*. Ed. by Shantanu Bhattacharya, Avinash Kumar Agarwal, Om Prakash, and Shailendra Singh. Singapore: Springer, 2019, pp. 193–207. ISBN: 978-981-13-3290-6. DOI: 10.1007/978-981-13-3290-6_10.

[Pauk18] Florian Pauker, Sabine Wolny, Solmaz Mansour Fallah, and Manuel Wimmer. "UML2OPC-UATransforming UML Class Diagrams to OPC UA Information Models". In: *Procedia CIRP*. 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 19-21 July 2017, Gulf of Naples, Italy 67 (Jan. 1, 2018), pp. 128–133. ISSN: 2212-8271. DOI: 10.1016/j.procir.2017.12.188.

[Paul14] Michael Paulweber and Klaus Lebert. "Softwaresicht – Prüfstand". In: *Mess- und Prüfstandstechnik: Antriebsstrangentwicklung · Hybridisierung · Elektrifizierung*. Ed. by Michael Paulweber and Klaus Lebert. Wiesbaden: Springer Fachmedien, 2014, pp. 273–376. ISBN: 978-3-658-04453-4. DOI: 10.1007/978-3-658-04453-4_4.

[Pelo23] E. Peloquin and J. Habermehl. "Adopting a Universal File Format for the Non-destructive Testing and Examination Industry". In: *Journal of Non-Destructive Testing and Evaluation (JNDE)* 20.4 (4 Dec. 7, 2023), pp. 51–53. URL: https://jnde.isnt.in/index.php/JNDE/article/view/65 (visited on May 9, 2024).

[Penn24] Jan Pennekamp. *Secure Collaborations for the Industrial Internet of Things*. 1. Auflage. Düren: Shaker, Apr. 2024. ISBN: 978-3-8440-9467-1.

[Penn19a] Jan Pennekamp, Markus Dahlmanns, Lars Gleim, Stefan Decker, and Klaus Wehrle. "Security Considerations for Collaborations in an Industrial IoT-based Lab of Labs". In: *2019 IEEE Global Conference on Internet of Things (GCIoT)*. 2019, pp. 1–7. DOI: 10.1109/GCIoT47977.2019.9058413.

[Penn19b] Jan Pennekamp, Rene Glebke, Martin Henze, et al. "Towards an Infrastructure Enabling the Internet of Production". In: *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2019, pp. 31–37. ISBN: 978-1-5386-8500-6.

[Perg19] Robert Pergl, Rob Hooft, Marek Suchánek, Vojtěch Knaisl, and Jan Slifka. ""Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning". In: *Data Science Journal* 18 (2019). ISSN: 1683-1470. DOI: 10.5334/dsj-2019-059.

[Pete20] Martin Peterek and Benjamin Montavon. “Prototype for Dual Digital Traceability of Metrology Data Using X.509 and IOTA”. In: *CIRP Annals* 69.1 (2020), pp. 449–452. ISSN: 0007-8506. DOI: 10.1016/j.cirp.2020.04.104.

[Pete22] Karsten Peters-von Gehlen, Heinke Höck, Andrej Fast, et al. “Recommendations for Discipline-Specific FAIRness Evaluation Derived from Applying an Ensemble of Evaluation Tools”. In: *CODATA Data Science Journal* 21.1 (1 Mar. 24, 2022), p. 7. DOI: 10.5334/dsj-2022-007.

[Pint13] G. Pintzos, M. Matsas, N. Papakostas, and G. Chryssolouris. “Production Data Handling Using a Manufacturing Indicators’ Knowledge Model”. In: *Procedia CIRP*. Forty Sixth CIRP Conference on Manufacturing Systems 2013 7 (Jan. 1, 2013), pp. 199–204. ISSN: 2212-8271. DOI: 10.1016/j.procir.2013.05.034.

[Pouc16] Line Pouchard. “Revisiting the Data Lifecycle with Big Data Curation”. In: *International Journal of Digital Curation* 10.2 (May 27, 2016), pp. 176–192. DOI: 10.2218/ijdc.v10i2.342.

[Proc07] Fred Proctor, Bill Rippey, John Horst, Joe Falco, and Tom Kramer. “Interoperability Testing for Shop Floor Measurement”. In: *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*. PerMIS ’07. New York, NY, USA: Association for Computing Machinery, Aug. 28, 2007, pp. 275–279. ISBN: 978-1-59593-854-1. DOI: 10.1145/1660877.1660917.

[Push20] R. L. Pushkov, L. I. Martinova, and S. V. Evstafieva. “Extending Functionality of the Control System Using MTConnect Data Collection Standard”. In: *IOP Conference Series: Materials Science and Engineering* 971.4 (Nov. 2020), p. 042080. DOI: 10.1088/1757-899X/971/4/042080.

[QUDT23] QUDT.org. *QUDT CATALOG - Quantities, Units, Dimensions and Data Types Ontologies*. Oct. 19, 2023. URL: https://www.qudt.org/2.1/catalog/qudt-catalog.html (visited on Nov. 1, 2023).

[Ram21] Peethani Sai Ram and K. Deepak Lawrence. “Implementation of Quality Information Framework (QIF): Towards Automatic Generation of Inspection Plan from Model-Based Definition (MBD) of Parts”. In: *Industry 4.0 and Advanced Manufacturing*. Ed. by Amaresh Chakrabarti and Manish Arora. Singapore: Springer, 2021, pp. 127–138. ISBN: 978-981-15-5689-0. DOI: 10.1007/978-981-15-5689-0_12.

[Reim19] Andrew P. Reimer and Elizabeth A. Madigan. “Veracity in Big Data: How Good Is Good Enough”. In: *Health Informatics Journal* 25.4 (Dec. 2019), pp. 1290–1298. ISSN: 1741-2811. DOI: 10.1177/1460458217744369.

[Rese20] Research Data Alliance FAIR Data Maturity Model Working Group. *FAIR Data Maturity Model: Specification and Guidelines*. Research Data Alliance, 2020. DOI: 10.15497/RDA00050.

[Řezn22] Tomáš Řezník, Lieven Raes, Andrew Stott, et al. “Improving the Documentation and Findability of Data Services and Repositories: A Review of (Meta)Data Management Approaches”. In: *Computers & Geosciences* 169 (Dec. 1, 2022), p. 105194. ISSN: 0098-3004. DOI: 10.1016/j.cageo.2022.105194.

[Ribe17] Luis Ribeiro. “Cyber-Physical Production Systems’ Design Challenges”. In: *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. 2017, pp. 1189–1194. ISBN: 2163-5145. DOI: 10.1109/ISIE.2017.8001414.

[Rijg13] Hajo Rijgersberg, Mark van Assem, and Jan Top. “Ontology of Units of Measure and Related Concepts”. In: *Semantic Web* 4.1 (2013), pp. 3–13. ISSN: 15700844. DOI: 10.3233/SW-2012-0069.

[Rodr15] Alberto Rodrigues Da Silva. “Model-Driven Engineering: A Survey Supported by the Unified Conceptual Model”. In: *Computer Languages, Systems & Structures* 43 (Oct. 2015), pp. 139–155. ISSN: 14778424. DOI: 10.1016/j.cl.2015.06.001.

[Rodr19] Jose Maria Rodriguez Corral, Ivan Ruiz-Rube, Anton Civit Balcells, et al. “A Study on the Suitability of Visual Languages for Non-Expert Robot Programmers”. In: *IEEE Access* 7 (2019), pp. 17535–17550. DOI: 10.1109/ACCESS.2019.2895913.

[Rodr16] Lisbeth Rodríguez-Mazahua, Cristian-Aarón Rodríguez-Enríquez, José Luis Sánchez-Cervantes, et al. “A General Perspective of Big Data: Applications, Tools, Challenges and Trends”. In: *The Journal of Supercomputing* 72.8 (Aug. 1, 2016), pp. 3073–3113. DOI: 10.1007/s11227-015-1501-1.

[Rook86] Paul Rook. “Controlling Software Projects”. In: *Software Engineering Journal* 1.1 (Jan. 1, 1986), pp. 7–16. ISSN: 2053-910X. DOI: 10.1049/sej.1986.0003.

[Rump] Bernhard Rumpe. *Domain-Specific Languages (DSLs)*. URL: https://rumpe.github.io/research/Domain-Specific-Languages/ (visited on June 14, 2024).

[Rump21] Bernhard Rumpe, Katrin Hölldobler, and Oliver Kautz. *MontiCore Language Workbench and Library Handbook*. Aachener Informatik-Berichte, Software-Engineering Band 48. Düren: Shaker Verlag, 2021. 481 pp. ISBN: 978-3-8440-8010-0.

[Rych18] Michael Rychlik, Giovanna Zappa, Larraitz Añorga, et al. “Ensuring Food Integrity by Metrology and FAIR Data Principles”. In: *Frontiers in Chemistry* 6 (2018), p. 49. ISSN: 2296-2646. DOI: 10.3389/fchem.2018.00049.

[Sadi23] Matthew N. O. Sadiku, Abayomi J. Ajayi-Majebi, and Philip O. Adebo. “Big Data in Manufacturing”. In: *Emerging Technologies in Manufacturing*. Ed. by Matthew N. O. Sadiku, Abayomi J. Ajayi-Majebi, and Philip O. Adebo. Cham: Springer International Publishing, 2023, pp. 95–107. ISBN: 978-3-031-23156-8. DOI: 10.1007/978-3-031-23156-8_7.

[Sagg18] Mandeep Kaur Saggi and Sushma Jain. “A Survey towards an Integration of Big Data Analytics to Big Insights for Value-Creation”. In: *Information Processing & Management*. In (Big) Data We Trust: Value Creation in Knowledge Organizations 54.5 (Sept. 1, 2018), pp. 758–790. DOI: 10.1016/j.ipm.2018.01.010.

[Sand23a] Mark P. Sanders and Matthias Bodenbenner. *MQTT2Influx-Bridge*. Version 1.0.0. Aachen: RWTH Aachen University, Aug. 14, 2023. DOI: 10.5281/zenodo.8247484.

[Sand23b] Mark P. Sanders, Matthias Bodenbenner, Philipp Dahlem, et al. “Laser Tracker-Based on-the-Fly Machine Tool Calibration without Real-Time Synchronization”. In: *Journal of Manufacturing and Materials Processing* 7.2 (2 Apr. 2023), p. 60. ISSN: 2504-4494. DOI: 10.3390/jmmp7020060.

[Sani23] Sebastian Saniuk, Dagmar Caganova, and Anna Saniuk. “Knowledge and Skills of Industrial Employees and Managerial Staff for the Industry 4.0 Implementation”. In: *Mobile Networks and Applications* 28.1 (Feb. 1, 2023), pp. 220–230. ISSN: 1572-8153. DOI: 10.1007/s11036-021-01788-4. URL: https://doi.org/10.1007/s11036-021-01788-4 (visited on Sept. 28, 2024).

[Sark20] Md Nazirul Islam Sarker, Yang Peng, Cheng Yiran, and Roger C. Shouse. “Disaster Resilience through Big Data: Way to Environmental Sustainability”. In: *International Journal of Disaster Risk Reduction* 51 (2020), p. 101769. DOI: 10.1016/j.ijdrr.2020.101769.

[Saut10] Thilo Sauter. “The Three Generations of Field-Level Networks—Evolution and Compatibility Issues”. In: *IEEE Transactions on Industrial Electronics* 57.11 (Nov. 2010), pp. 3585–3595. DOI: 10.1109/TIE.2010.2062473.

[Sawa21] Pegdwendé Sawadogo and Jérôme Darmont. “On Data Lake Architectures and Metadata Management”. In: *Journal of Intelligent Information Systems* 56.1 (2021), pp. 97–120. DOI: 10.1007/s10844-020-00608-7.

[Sche23] Schema.org Community Group. *Schema.Org*. Oct. 17, 2023. URL: https://schema.org/ (visited on Nov. 1, 2023).

[Schl09] Miriam Schleipen and Rainer Drath. “Three-View-Concept for Modeling Process or Manufacturing Plants with AutomationML”. In: *2009 IEEE Conference on Emerging Technologies & Factory Automation*. 2009 IEEE Conference on Emerging Technologies & Factory Automation. Sept. 2009, pp. 1–4. DOI: 10.1109/ETFA.2009.5347260.

[Schl24] Alexandra Schlemitz and Vitaliy Mezhuyev. “Approaches for Data Collection and Process Standardization in Smart Manufacturing: Systematic Literature Review”. In: *Journal of Industrial Information Integration* 38 (Mar. 1, 2024), p. 100578. ISSN: 2452-414X. DOI: 10.1016/j.jii.2024.100578. URL: https://www.sciencedirect.com/science/article/pii/S2452414X24000220 (visited on May 4, 2024).

[Schm23] Robert H. Schmitt, Matthias Bodenbenner, Hanna Brings, and Benjamin Montavon. “Datenstrukturen für eine resiliente Life Cycle Sustainability”. In: *Empower Green Production*. Aachener Werkzeugmaschinen-Kolloquium 2023. Aachen, May 4, 2023. DOI: 10.24406/publica-958.

[Schm24] Robert H. Schmitt, Matthias Bodenbenner, Tobias Hamann, et al. “Leveraging Measurement Data Quality by Adoption of the FAIR Guiding Principles”. In: *tm - Technisches Messen* (July 9, 2024). DOI: 10.1515/teme-2024-0040.

[Schm18a] Robert H. Schmitt and Christoph Voigtmann. “Sensor Information as a Service – Component of Networked Production”. In: *Journal of Sensors and Sensor Systems* 7.1 (2018), pp. 389–402. DOI: 10.5194/jsss-7-389-2018.

[Schm18b] Dominik Schmitz and Marius Politze. “Forschungsdaten managen – Bausteine für eine dezentrale, forschungsnahe Unterstützung”. In: *o-bib. Das offene Bibliotheksjournal / Herausgeber VDB* 5.3 (3 Sept. 28, 2018), pp. 76–91. ISSN: 2363-9814. DOI: 10.5282/o-bib/2018H3S76-91. URL: https://www.o-bib.de/bib/article/view/5339 (visited on Feb. 14, 2024).

[Schu17] Günther Schuh, Jan Reschke, G. Tücks, et al. “Change Request im Produktionsbetrieb”. In: *Internet of Production für agile Unternehmen*. 29. Aachener Werkzeugmaschinen-Kolloquium. Aachen: Apprimus Verlag, 2017, pp. 109–131. ISBN: 978-3-86359-512-8.

[Seba22] Laura Sebastian-Coleman. “Chapter 11 - Data Life Cycle Processes”. In: *Meeting the Challenges of Data Quality Management*. Ed. by Laura Sebastian-Coleman. Academic Press, Jan. 1, 2022, pp. 257–280. ISBN: 978-0-12-821737-5. DOI: 10.1016/B978-0-12-821737-5.00011-0.

[Senn23] Leonie Senn-Kalb and Dev Metha. *Artificial Intelligence: In-Depth Market Analysis*. Market Insights report. Statista, Apr. 2023, p. 211. URL: https://de.statista.com/statistik/studie/id/50489/dokument/in-depth-report-artificial-intelligence/ (visited on June 8, 2024).

[Shad06] N. Shadbolt, T. Berners-Lee, and W. Hall. “The Semantic Web Revisited”. In: *IEEE Intelligent Systems* 21.3 (Jan. 2006), pp. 96–101. DOI: 10.1109/MIS.2006.62.

[Shar22] Fathi M. Sharadah, Shawki Al-Dubaee, and George Weir. “MTConnect and Digital Twin Applications and Future Perspectives”. In: *Digital Twins for Digital Transformation: Innovation in Industry*. Ed. by Aboul Ella Hassanien, Ashraf Darwish, and Vaclav Snasel. Cham: Springer International Publishing, 2022, pp. 87–98. ISBN: 978-3-030-96802-1. DOI: 10.1007/978-3-030-96802-1_5.

[Shar24] Ujjwal Sharma and Carsten Bormann. *RFC 9557 - Date and Time on the Internet: Timestamps with Additional Information*. Apr. 2024. URL: https://datatracker.ietf.org/doc/html/rfc9557 (visited on July 11, 2024).

[Shel16] Zach Shelby, Klaus Hartke, and Carsten Bormann. *RFC 7252 - The Constrained Application Protocol (CoAP)*. June 2016. URL: https://datatracker.ietf.org/doc/html/rfc7252 (visited on June 2, 2024).

[Sobe19] William Sobel, Randy Armstrong, John Turner, Russel Waddell, and Shaurabh Singh. *OPC 30070-1 MTConnect Devices*. Version 2.0. June 5, 2019. URL: https://opcfoundation.org/developer-tools/documents/view/211 (visited on June 9, 2024).

[Somm23] Ashley Sommer and Nicholas Car. *pySHACL*. Version 0.25.0. Nov. 23, 2023. DOI: 10.5281/ZENODO.10199105.

[Sous22a] João Sousa, João Pedro Mendonça, and José Machado. “A Generic Interface and a Framework Designed for Industrial Metrology Integration for the Internet of Things”. In: *Computers in Industry* 138 (June 1, 2022), p. 103632. ISSN: 0166-3615. DOI: 10.1016/j.compind.2022.103632.

[Sous22b] João Sousa, João Silva, and José Machado. “Metrology Information in Cyber-Physical Systems”. In: *Innovations in Mechatronics Engineering*. Ed. by José Machado, Filomena Soares, Justyna Trojanowska, and Sahin Yildirim. Cham: Springer International Publishing, 2022, pp. 285–292. ISBN: 978-3-030-79168-1. DOI: 10.1007/978-3-030-79168-1_26.

[Stac23] Stack Exchange. *Stack Overflow Developer Survey 2023*. Stack Overflow. 2023. URL: https://survey.stackoverflow.co/2023/#technology-most-popular-technologies (visited on June 3, 2024).

[Stat23] Statista. *Big Data - Statistik-Report zum Thema Big Data*. Statista, Nov. 3, 2023. URL: https://de.statista.com/statistik/studie/id/16989/dokument/big-data--statista-dossier/ (visited on Oct. 16, 2024).

[Stav23] Panagiotis Stavropoulos, Alexios Papacharalampopoulos, and Kyriakos Sabatakakis. “Data Attributes in Quality Monitoring of Manufacturing Processes: The Welding Case”. In: *Applied Sciences* 13.19 (Jan. 2023), p. 10580. ISSN: 2076-3417. DOI: 10.3390/app131910580.

[Stea96] Mark Steadman, Tom Albright, and Kimberly Dunn. “Stakeholder Group Interest in the New Manufacturing Environment”. In: *Managerial Auditing Journal* 11.2 (Jan. 1, 1996), pp. 4–9. ISSN: 0268-6902. DOI: 10.1108/02686909610107915.

[Stei08] Dave Steinberg, Frank Budinsky, Marcelo Paternostro, and Ed Merks, eds. *Eclipse Modeling Framework: EMF*. 2. ed., rev. and updated, 2. printing. The Eclipse Series. Upper Saddle River, NJ Munich: Addison-Wesley, Dec. 16, 2008. 704 pp. ISBN: 978-0-321-33188-5.

[Stei21] Gernot Steindl and Wolfgang Kastner. “Transforming OPC UA Information Models into Domain-Specific Ontologies”. In: *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*. 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS). May 2021, pp. 191–196. DOI: 10.1109/ICPS49255.2021.9468254.

[Sun22] Chang Sun, Vincent Emonet, and Michel Dumontier. “A Comprehensive Comparison of Automated FAIRness Evaluation Tools”. In: *Semantic Web Applications and Tools for Health Care and Life Sciences*. 13th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences. Vol. 3127. CEUR Workshop Proceedings. Jan. 1, 2022, pp. 44–53. URL: https://cris.maastrichtuniversity.nl/en/publications/a-comprehensive-comparison-of-automated-fairness-evaluation-tools (visited on Oct. 16, 2024).

[Sust20] Hana Pergl Sustkova, Kristina Maria Hettne, Peter Wittenburg, et al. “FAIR Convergence Matrix: Optimizing the Reuse of Existing FAIR-Related Resources”. In: *Data Intelligence* 2.1-2 (Jan. 1, 2020), pp. 158–170. ISSN: 2641-435X. DOI: 10.1162/dint_a_00038.

[Tanh19] Toste Tanhua, Sylvie Pouliquen, Jessica Hausman, et al. “Ocean FAIR Data Services”. In: *Frontiers in Marine Science* 6 (2019). DOI: 10.3389/fmars.2019.00440.

[Tao18] Fei Tao, Qinglin Qi, Ang Liu, and Andrew Kusiak. “Data-Driven Smart Manufacturing”. In: *Journal of Manufacturing Systems*. Special Issue on Smart Manufacturing 48 (July 1, 2018), pp. 157–169. ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2018.01.006.

[The 16] The International Academy For Production Engineering. *CIRP Encyclopedia of Production Engineering*. Ed. by Luc Laperrière and Gunther Reinhart. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. 1500 pp. ISBN: 978-3-642-35950-7. DOI: 10.1007/978-3-642-35950-7.

[The 24] The MathWorks. *Simulink – Simulation und Model-Based Design*. 2024. URL: https://de.mathworks.com/products/simulink.html (visited on June 5, 2024).

[Thiy15] V.S. Thiyagarajan and K. Venkatachalapathy. “Isolating Values from Big Data with the Help of Four V’s”. In: *International Journal of Research in Engineering and Technology* 04.01 (Jan. 25, 2015), pp. 132–135. ISSN: 23217308, 23191163. DOI: 10.15623/ijret.2015.0401022.

[Thom21] Spencer A. Thomas and Frederic Brochu. “A Framework for Traceble Storage and Curation of Measurement Data”. In: *Measurement: Sensors* 18 (Dec. 1, 2021), p. 100201. ISSN: 2665-9174. DOI: 10.1016/j.measen.2021.100201.

[Thom20] Mark Thompson, Kees Burger, Rajaram Kaliyaperumal, Marco Roos, and Luiz Olavo Bonino da Silva Santos. “Making FAIR Easy with FAIR Tools: From Creolization to Convergence”. In: *Data Intelligence* 2 (2020), pp. 87–95. DOI: 10.1162/DINT_A_00031.

[Tolk03] Andreas Tolk. “Introducing a Reference Model for Measures of Merit for Coalition Interoperability”. In: 2003 Command and Control Research and Technology Symposium. Washington DC, June 2003. URL: https://www.researchgate.net/publication/228582744_Introducing_a_Reference_Model_for_Measures_of_Merit_for_Coalition_Interoperability (visited on Oct. 16, 2024).

[Toml17] Matúš Tomlein and Kaj Grønbæk. “A Visual Programming Approach Based on Domain Ontologies for Configuring Industrial IoT Installations”. In: *Proceedings of the Seventh International Conference on the Internet of Things*. Ed. by Unknown. New York, NY: ACM, 2017, pp. 1–9. ISBN: 978-1-4503-5318-2. DOI: 10.1145/3131542.3131552.

[Töpf07] Susanne Töpfer, Gerhard Linss, Maik Rosenberger, Uwe Nehse, and Karina Weißensee. “Automatic Execution of Inspection Plans with the I++ DME Interface for Industrial Coordinate Measurements”. In: *Metrology and Measurement Systems* 14 (2007), pp. 71–87. URL: https://api.semanticscholar.org/CorpusID:109876590 (visited on June 9, 2024).

[Turn06] Charles Turnitsa and Andreas Tolk. “Battle Management Language: A Triangle with Five Sides”. In: *Proceedings of the Simulation Interoperability Standards Organization (SISO) Spring Simulation Interoperability Workshop (SIW)*. Vol. 27. 2006. URL: https://www.academia.edu/17073015/Battle_management_language_a_triangle_with_five_sides (visited on Oct. 16, 2024).

[UK D24] UK Data Service. *Research Data Management*. UK Data Service. 2024. URL: https://ukdataservice.ac.uk/learning-hub/research-data-management/ (visited on Mar. 27, 2024).

[UNEC21] UNECE. *UN/CEFACT-Rec20 | UNECE*. UNECE, June 9, 2021. URL: https://unece.org/trade/documents/2021/06/uncefact-rec20-0 (visited on Jan. 31, 2024).

[vReis20] Mirjam van Reisen, Mia Stokmans, Mariam Basajja, et al. "Towards the Tipping Point for FAIR Implementation". In: *Data Intelligence* 2.1-2 (2020), pp. 264–275. DOI: 10.1162/DINT_A_00049.

[vVlij20] Herman van Vlijmen, Albert Mons, Arne Waalkens, et al. "The Need of Industry to Go FAIR". In: *Data Intelligence* 2.1-2 (2020), pp. 276–284. DOI: 10.1162/dint_a_00050.

[vdSomp09] Herbert van de Sompel, Michael L. Nelson, Robert Sanderson, et al. *Memento: Time Travel for the Web*. Nov. 5, 2009. URL: https://arxiv.org/pdf/0911.1112.

[vdAals17] Wil M. P. van der Aalst, Martin Bichler, and Armin Heinzl. "Responsible Data Science". In: *Business & Information Systems Engineering* 59.5 (Oct. 1, 2017), pp. 311–313. ISSN: 1867-0202. DOI: 10.1007/s12599-017-0487-z.

[VDI22] VDI. *VDI/VDE 2623:2022-03 - Format für den Austausch von Daten im Prüfmittelmanagement - Definition des Calibration-Data-Exchange-Datenformats (CDE-Datenformat)*. Mar. 2022. URL: https://www.dinmedia.de/de/technische-regel/vdi-vde-2623/345941956 (visited on Oct. 16, 2024).

[VDMA19] VDMA. *OPC 40100-1 - OPC UA for Machine Vision - Part 1: Control, Configuration Management, Recipe Management, Result Management*. Version 1.0. Frankfurt am Main, Aug. 5, 2019. URL: https://opcfoundation.org/developer-tools/documents/view/217 (visited on June 9, 2024).

[VDMA23] VDMA. *OPC 40210 Geometric Measuring Systems*. Version 1.0. Frankfurt am Main, May 24, 2023. URL: https://opcfoundation.org/developer-tools/documents/view/269 (visited on Sept. 20, 2024).

[VDMA24a] VDMA. *OPC 40100-2 Machine Vision - Asset Mgmt and Condition Monitoring*. Version 1.0. Frankfurt am Main, May 20, 2024. URL: https://opcfoundation.org/developer-tools/documents/view/316 (visited on June 9, 2024).

[VDMA24b] VDMA. *VDMA 8722:2024-04 - I++ DME – Dimensional Measurement Equipment Interface (Version 2.5)*. Berlin, Apr. 1, 2024. URL: https://www.dinmedia.de/de/technische-regel/vdma-8722/378593245 (visited on June 9, 2024).

[VDW22] VDW. *OPC 40501-1 - OPC UA for Machine Tools - Machine Monitoring and Job Overview*. Version 1.01.1. Frankfurt am Main, July 4, 2022. URL: https://opcfoundation.org/developer-tools/documents/view/215 (visited on June 9, 2024).

[Vect24] Vector Informatik GmbH. *Messdatenformat MDF*. Vector Informatik GmbH. 2024. URL: https://www.vector.com/at/de/produkte/anwendungsgebiete/steuergeraete-kalibrierung/messen/mdf/ (visited on May 9, 2024).

[Voge20] Birgit Vogel-Heuser, Markus Böhm, Felix Brodeck, et al. “Interdisciplinary Engineering of Cyber-Physical Production Systems: Highlighting the Benefits of a Combined Interdisciplinary Modelling Approach on the Basis of an Industrial Case”. In: *Design Science* 6 (Jan. 2020), e5. ISSN: 2053-4701. DOI: 10.1017/dsj.2020.2.

[Vujc18] Josip Vujcic. *Foreign Key to Hypertable · Issue #498 · Timescale/Timescaledb*. GitHub. Apr. 13, 2018. URL: https://github.com/timescale/timescaledb/issues/498 (visited on June 30, 2024).

[W3C 12] W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Dec. 11, 2012. URL: https://www.w3.org/TR/owl2-overview/ (visited on Oct. 12, 2024).

[Wang22] Junliang Wang, Chuqiao Xu, Jie Zhang, and Ray Zhong. “Big Data Analytics for Intelligent Manufacturing Systems: A Review”. In: *Journal of Manufacturing Systems* 62 (Jan. 1, 2022), pp. 738–752. ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2021.03.005.

[Wang18a] JunPing Wang, WenSheng Zhang, YouKang Shi, ShiHui Duan, and Jin Liu. *Industrial Big Data Analytics: Challenges, Methodologies, and Applications*. 2018. URL: https://arxiv.org/pdf/1807.01016.

[Wang18b] Shiyong Wang, Jiafu Wan, Di Li, and Chengliang Liu. “Knowledge Reasoning with Semantic Data for Real-Time Data Processing in Smart Factory”. In: *Sensors* 18.2 (2 Feb. 2018), p. 471. ISSN: 1424-8220. DOI: 10.3390/s18020471.

[Wang09] W. Wang, A. Tolk, and W. Wang. “The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S”. In: Spring Simulation Multiconference 2009 - Co-located with the 2009 SISO Spring Simulation Interoperability Workshop. 2009. URL: https://arxiv.org/pdf/0908.0191 (visited on Oct. 16, 2024).

[Wang11] Zheng Wang, Luca Mastrogiacomo, Fiorenzo Franceschini, and Paul Maropoulos. “Experimental Comparison of Dynamic Tracking Performance of iGPS and Laser Tracker”. In: *The International Journal of Advanced Manufacturing Technology* 56.1-4 (Sept. 2011), pp. 205–213. ISSN: 0268-3768, 1433-3015. DOI: 10.1007/s00170-011-3166-0.

[Weib95] Stuart L. Weibel and Erik Jul. “PURLs to Improve Access to Internet”. In: *OCLC Newsletter* 2018 (Dec. 1, 1995), p. 19. URL: https://library.oclc.org/digital/collection/p267701coll28/id/1839 (visited on Mar. 24, 2024).

[Weig20] Tobias Weigel, Ulrich Schwardmann, Jens Klump, Sofiane Bendoukha, and Robert Quick. “Making Data and Workflows Findable for Machines”. In: *Data Intelligence* 2.1-2 (2020), pp. 40–46. DOI: 10.1162/dint_a_00026.

[Weis23] Arno Weiss and Dirk Reichelt. “Reusing OPC UA Information Models in the Asset Administration Shell”. In: *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*. 2023 IEEE 21st International Conference on Industrial Informatics (INDIN). July 2023, pp. 1–6. DOI: 10.1109/INDIN51400.2023.10218292.

[Wied19] Thomas Wiedenhöfer, Daniel Hutzschenreuter, Ian Smith, and Clifford Brown. "Document Describing a Universal and Flexible Structure for Digital Calibration Certificates (DCC)". In: (Nov. 11, 2019). DOI: 10.5281/zenodo.3696567.

[Wilk18a] Mark Wilkinson, Erik Schultes, Luiz Olavo Bonino, et al. *FAIRMetrics/Metrics: FAIR Metrics, Evaluation Results, and Initial Release of Automated Evaluator Code*. Version v1.0.3. Zenodo, July 4, 2018. DOI: 10.5281/zenodo.1305060.

[Wilk19a] Mark D. Wilkinson. *FAIR Metrics - Gen2*. May 13, 2019. URL: https://github.com/FAIRMetrics/Metrics/tree/master/MaturityIndicators/Gen2 (visited on Feb. 21, 2024).

[Wilk16] Mark D. Wilkinson, Michel Dumontier, I. Jsbrand Jan Aalbersberg, et al. "The FAIR Guiding Principles for Scientific Data Management and Stewardship". In: *Scientific data* 3 (2016). DOI: 10.1038/sdata.2016.18.

[Wilk19b] Mark D. Wilkinson, Michel Dumontier, Susanna-Assunta Sansone, et al. "Evaluating FAIR Maturity through a Scalable, Automated, Community-Governed Framework". In: *Scientific Data* 6.1 (2019), p. 174. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0184-5.

[Wilk18b] Mark D. Wilkinson, Susanna-Assunta Sansone, Erik Schultes, et al. "A Design Framework and Exemplar Metrics for FAIRness". In: *Scientific Data* 5.1 (2018), p. 180118. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.118.

[Wing19] Jeannette M. Wing. "The Data Life Cycle". In: *Harvard Data Science Review* 1.1 (July 3, 2019). DOI: 10.1162/99608f92.e26845b4.

[Wise19] John Wise, Alexandra Grebe de Barron, Andrea Splendiani, et al. "Implementation and Relevance of FAIR Data Principles in Biopharmaceutical R&D". In: *Drug Discovery Today* 24.4 (2019), pp. 933–938. ISSN: 1359-6446. DOI: 10.1016/j.drudis.2019.01.008.

[Ye18] Xun Ye, Tae Yang Park, Seung Ho Hong, Yuemin Ding, and Aidong Xu. "Implementation of a Production-Control System Using Integrated Automation ML and OPC UA". In: *2018 Workshop on Metrology for Industry 4.0 and IoT*. 2018 Workshop on Metrology for Industry 4.0 and IoT. Apr. 2018, pp. 1–6. DOI: 10.1109/METROI4.2018.8428310.

[Zhan22] Joe Zhang, Joshua Symons, Paul Agapow, et al. "Best Practices in the Real-World Data Life Cycle". In: *PLOS Digital Health* 1.1 (Jan. 18, 2022), e0000003. ISSN: 2767-3170. DOI: 10.1371/journal.pdig.0000003.

[Zhan23] Xiongfeng Zhang, Seonjo Lim, Changdae Lee, et al. "Integration of 5G and OPC UA for Smart Manufacturing of the Future". In: *2023 IEEE/SICE International Symposium on System Integration (SII)*. 2023 IEEE/SICE International Symposium on System Integration (SII). Jan. 2023, pp. 1–6. DOI: 10.1109/SII55687.2023.10039191.

[Zhao12] Yaoyao Fiona Zhao, John A Horst, Thomas R. Kramer, William Rippey, and Robert J. Brown. "Quality Information Framework – Integrating Metrology Processes". In: *IFAC Proceedings Volumes*. 14th IFAC Symposium on Information Control Problems in Manufacturing 45.6 (May 23, 2012), pp. 1301–1308. ISSN: 1474-6670. DOI: 10.3182/20120523-3-RO-2023.00113.

[Zhen20a] Xiaochen Zheng, Jinzhi Lu, Shengjing Sun, and Dimitris Kiritsis. “Decentralized Industrial IoT Data Management Based on Blockchain and IPFS”. In: *Advances in Production Management Systems. Towards Smart and Digital Manufacturing*. Ed. by Bojan Lalic. Vol. 592. IFIP Advances in Information and Communication Technology Ser. Cham: Springer International Publishing AG, 2020, pp. 222–229. ISBN: 978-3-030-57996-8. DOI: 10.1007/978-3-030-57997-5_26.

[Zhen20b] Xu Zheng and Zhipeng Cai. “Privacy-Preserved Data Sharing Towards Multiple Parties in Industrial IoTs: IEEE Journal on Selected Areas in Communications, 38(5), 968-979”. In: *IEEE Journal on Selected Areas in Communications* 38.5 (2020), pp. 968–979. ISSN: 0733-8716. DOI: 10.1109/JSAC.2020.2980802.

[Ziko12] Paul Zikopoulos. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. New York: McGraw-Hill, 2012. 141 pp. ISBN: 978-0-07-179053-6.

Data availability statement

Following the guidelines of responsible and transparent research, all technical artifacts and all research data acquired to evaluate the conducted research have been archived and published, if possible. In particular, the author published the following supplementary material, which this thesis is based on during the studies:

Matthias Bodenbenner
Metadata Profile for FAIR Sensor Data Based on the Sensor Interfacing Language
Metadata profiles of the sensor data model shown in Figure 4.5 in SHACL. The most current machine-actionable MPs are available at `https://purl.org/soil/`.
Dataset, Zenodo, 2024
`https://doi.org/10.5281/zenodo.10965203`

Matthias Bodenbenner
Validation and FAIRness assessment results of FAIR Sensor Services based on the SensOr Interfacing Language
This dataset contains the scripts and the raw output of the automated *FAIRness* assessment described in Section 6.4.2. Figure 6.10 is created based in this data.
Dataset & Software, Zenodo, 2024
`https://doi.org/10.5281/zenodo.13835237`

Matthias Bodenbenner
Python Unified Device Interface
The implemented Python library for running a *FAIR Sensor Service*. It can directly be installed via the Python Package Index (PyPi) using `pip wzl-udi`.
Software, V 10.1.1, Zenodo, 2024
`https://doi.org/10.5281/zenodo.13922866`

Matthias Bodenbenner
SOIL Dummies
The SOIL models, handwritten source code, and the generated MPs and static metadata sets of the three simulated measuring systems implemented as *FAIR Sensor Services* used for validation of the developed models, SOIL and the implemented libraries.
Software, 1.0.0, Zenodo, 2024
`https://doi.org/10.5281/zenodo.13831871`

Matthias Bodenbenner, André Fugmann, Julian Giesen
SOIL Workbench
Language workbench implementing the parser, validator, and generator for SOIL based on the *MontiCore* language workbench developed by Rumpe et al. [Rump21].
Software, V2.3.1, Zenodo, 2024
https://doi.org/10.5281/zenodo.13922911

Matthias Bodenbenner, André Fugmann, Julian Giesen
SOIL Editor
Web-based UI for intuitively defining interface descriptions of *FAIR Sensor Services* using SOIL.
Software, V1.0.0, Zenodo, 2024
https://doi.org/10.5281/zenodo.13922935

Matthias Bodenbenner, André Fugmann, Julian Giesen
SOIL Backend
FastAPI-based Python server encapsulating the command line interface of the *SOIL Workbench*. Primarily used as the backend in conjunction with the *SOIL Editor* providing a web-based UI for developing interfaces for *FAIR Sensor Services* based on the SOIL.
Software, V1.0.0, Zenodo, 2024
https://doi.org/10.5281/zenodo.12513293

Jonas Schlabertz, Matthias Bodenbenner, Jan Pennekamp
FAIR Sensor Ecosystem
Implementation of the FAIR Sensor Ecosystem as a web service in TypeScript.
Software, V1.0.0, Zenodo, 2024
https://doi.org/10.5281/zenodo.13923574

Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon
Python-MQTT Library
Minimal, lightweight MQTT-Client for Python.
Software, V2.5.4, Zenodo, 2024
https://doi.org/10.5281/zenodo.8192493

Benjamin Montavon, Matthias Bodenbenner, Tim Heß
C++ Unified Device Interface
Implementation of the dynamic linked libraries in C++ for provision of *FAIR Sensor Services*.
Software, V1.0.0, Zenodo, 2024
https://doi.org/10.5281/zenodo.13923642

Matthias Bodenbenner, Davis Ens, Oliver Rostig, Susanna Weber, Tobias Westphal
SOIL2OPC UA Translator
Implementation of a Python script to generate OPC UA-based *FAIR Sensor Service* from a SOIL model.
Software, V1.0.0, Zenodo, 2024
https://doi.org/10.5281/zenodo.13923657

Nevertheless, some artifacts, especially source code, can not be published due to copyright regulations or non-disclosure agreements:

- The implementation of the *FAIR Sensor Service* of the API Radian™ laser tracker, due to the usage of the closed-source API provided by the device manufacturer.
- The implementation of the *FAIR Sensor Service* of the I++ DME interface of the Leitz PMM-C Precision™, because the reference implementation used for validation is closed source. The reference implementation was kindly and exclusively provided for validation of the results of this thesis.
- Implementation of the *FAIR Sensor Service* of the OptiTrack™ MCS in C++.

Publications

Publications with a direct contribution of the author

The following list contains all publications with the direct contribution of the author which are directly related to this thesis.

Robert H. Schmitt, **Matthias Bodenbenner**, Tobias Hamann, Mark P. Sanders, Mario Moser, and Anas Abdelrazeq.
"Leveraging Measurement Data Quality by Adoption of the FAIR Guiding Principles".
In: tm - Technisches Messen (July 9, 2024).
https://dx.doi.org/10.1515/teme-2024-0040

Matthias Bodenbenner, Dominik Wolfschläger, and Robert H. Schmitt.
"Providing FAIR Sensor Data Models Using Semantic Web Technologies and Ontologies".
Pre-print, Apr. 15, 2024.
https://dx.doi.org/10.13140/RG.2.2.27882.73928.

Matthias Bodenbenner.
"Description and Manual of the SensOr Interfacing Language".
Whitepaper, May 22, 2024.
https://dx.doi.org/10.5281/zenodo.11239479

Robert H. Schmitt, **Matthias Bodenbenner**, Hanna Brings, and Benjamin Montavon.
"Datenstrukturen für eine resiliente Life Cycle Sustainability".
In: Empower Green Production. Aachener Werkzeugmaschinen-Kolloquium 2023. Aachen, May 4, 2023.
https://dx.doi.org/10.24406/publica-958

Mark P. Sanders, **Matthias Bodenbenner**, Philipp Dahlem, Dominik Emonts, Benjamin Montavon, and Robert H. Schmitt.
"Laser Tracker-Based on-the-Fly Machine Tool Calibration without Real-Time Synchronization".
In: Journal of Manufacturing and Materials Processing 7.2 (2 Apr. 2023), p. 60.
https://dx.doi.org/10.3390/jmmp7020060

Nils Preuß, **Matthias Bodenbenner**, Benedikt Heinrichs, Jürgen Windeck, Mario Moser, and Marc Fuhrmans.
"Creating Application-Specific Metadata Profiles While Improving Interoperability and Consistency of Research Data for the Engineering Sciences".
Report. Darmstadt: Universitäts- und Landesbibliothek Darmstadt, Sept. 27, 2023.
`https://dx.doi.org/10.26083/tuprints-00024573`

Jan Pennekamp, Anastasiia Belova, Thomas Bergs, **Matthias Bodenbenner**, Andreas Bührig-Polaczek, Markus Dahlmanns, Ike Kunze, Moritz Kröger, Sandra Geisler, Martin Henze, Daniel Lütticke, Benjamin Montavon, Philipp Niemietz, Lucia Ortjohann, Maximilian Rudack, Robert H. Schmitt, Uwe Vroomen, Klaus Wehrle, and Michael Zeng.
"Evolving the Digital Industrial Infrastructure for Production: Steps Taken and the Road Ahead".
In: Internet of Production: Fundamentals, Applications and Proceedings. Ed. by Christian Brecher, Günther Schuh, Wil van der Aalst, Matthias Jarke, Frank T. Piller, and Melanie Padberg. Cham: Springer International Publishing, 2023, pp. 1-25.
`https://dx.doi.org/10.1007/978-3-030-98062-7_2-1`

Matthias Bodenbenner, Jan Pennekamp, Benjamin Montavon, Klaus Wehrle, and Robert H. Schmitt.
"FAIR Sensor Ecosystem: Long-Term (Re-)Usability of FAIR Sensor Data through Contextualization".
In: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN). July 2023, pp. 1-8
`https://dx.doi.org/10.1109/INDIN51400.2023.10218149`

Benjamin Montavon and **Matthias Bodenbenner**.
"Messdaten verarbeiten, verteilen, nutzen- aktuelle Entwicklungen".
In: Handbuch Messtechnik in der industriellen Produktion: valide Messergebnisse planen, erhalten, auswerten und verteilen. 1st ed. Carl Hanser Verlag GmbH & Co. KG, Aug. 2023, pp. 673-690

Matthias Grönewald, Patrick Mund, **Matthias Bodenbenner**, Marc Fuhrmans, Benedikt Heinrichs, Matthias S. Müller, Peter F. Pelz, Marius Politze, Nils Preuß, Robert H. Schmitt, and Thomas Stäcker.
"Mit AIMS zu einem Metadatenmanagement 4.0: FAIRe Forschungsdaten benötigen interoperable Metadaten".
heiBOOKS, Apr. 21, 2022.
`https://dx.doi.org/10.11588/heibooks.979`

Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon, and Robert H. Schmitt.
"Multiprotokollfähiges Meta-Datenmodell".
In: wt Werkstattstechnik online 112.11-12 (2022).
`https://dx.doi.org/10.37544/1436-4980-2022-11-12`

Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt.
“Model-Driven Development of Interoperable Communication Interfaces for FAIR Sensor Services”.
In: Measurement: Sensors 24 (2022), p. 100442.
https://dx.doi.org/10.1016/j.measen.2022.100442

Dominik Emonts, Philipp Dahlem, **Matthias Bodenbenner**, Benjamin Montavon, and Robert H. Schmitt.
“Virtuelle Klimatisierung/Virtual Climatization”.
In: wt Werkstattstechnik online 111.11-12 (Nov. 30, 2021), pp. 887-892.
https://dx.doi.org/10.37544/1436-4980-2021-11-12-117

Matthias Bodenbenner, Friedhelm Bösser, Sascha Dessel, André Martin, Benjamin Montavon, Ulrich Neuschaefer-Rube, Dietrich Imkamp, Niels Rabbe, Torsten Ring, and Robert Schmitt.
“Fertigungsmesstechnik in der digitalen Produktion: Modelle für Messsysteme und deren Anwendung”.
Ed. by VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, Nov. 2021

Matthias Bodenbenner, Benjamin Montavon, and Robert H. Schmitt.
“FAIR Sensor Services - Towards Sustainable Sensor Data Management”.
In: Measurement: Sensors 18 (2021), p. 100206.
https://dx.doi.org/10.1016/j.measen.2021.100206

Matthias Bodenbenner, Mark P. Sanders, Benjamin Montavon, and Robert H. Schmitt.
“Domain-Specific Language for Sensors in the Internet of Production”.
In: Production at the Leading Edge of Technology. Ed. by Bernd-Arno Behrens, Alexander Brosius, Wolfgang Hintze, Steffen Ihlenfeldt, and Jens Peter Wulfsberg. Lecture Notes in Production Engineering. Berlin, Heidelberg: Springer, 2021, pp. 448-456.
https://dx.doi.org/10.1007/978-3-662-62138-7_45

Thesis with a contribution of the author to the supervision

Tim Heß
Untersuchung der Performance eines Motion-Capture- und Photogrammetriesystems für die Referenzierung von mobilen Robotern in der flexiblen Montage
Master Thesis, 2024
The SOIL-model of the OptiTrack™ MCS (cf. Appendix F) represents joint work in the context of this thesis.

Nils Bender
Empirische Untersuchung der Messunsicherheit eines OptiTrack-Photogrammetriesystems in großen Volumen
Bachelor Thesis, 2023

Colin Klein
FAIR Sensor Health Monitoring of Flight Test Data
Master Thesis, 2023

Julian Giesen
Extending the SensOr Interfacing Language using Semantic Modeling Concepts
Bachelor Thesis, 2022
The syntax for the integration of controlled terminology in SOIL (cf. Section 5.4.1) represents joint work in the context of this thesis.

Jonas Schlabertz
Implementing FAIR Principles for Long-Term Sensor Data Storage in the Internet of Production
Master Thesis, 2022
The contextualization structure (cf. Appendix C), the architecture, and the implementation of the *FAIR Sensor Ecosystem* (cf. Section 6.2.3) represent joint work in the context of this thesis.

Serhat Combas
Analyse und Vergleich verschiedener Algorithmen zur Kalibrierung eines Kamera-Trackers
Bachelor Thesis, 2021

Yavuz Kalem
Untersuchung von FAIRness Metriken zur FAIRness-Bewertung industrieller Sensordaten
Bachelor Thesis, 2021
The review of available *FAIRness* metrics (see Section 3.2.1) has been initialized in this thesis.

André Fugmann
Modeling of sensor interfaces with a textual domain-specific language
Bachelor Thesis, 2021
The first, later reworked, prototype of the *SOIL Workbench* represents joint work within the context of this thesis.

Appendix

A Summary of the consolidated *FAIRness* metrics

Tables A.1 to A.4 summarize the results of the investigation of the state of the art of metrics for assessing the *FAIRness* of a data (sets). For each metric, the addressed guideline of the *FAIR Guiding Principles*, the framework the metric is contained in, and the number of redundant metrics (of other frameworks) is given. Moreover, the short explanation of the authors of the metric is repeated. Each metric is given a unique ID in ascending order for unambiguous identification.

Table A.1: Applicable and distinct *FAIRness* metrics for findability identified in literature [Wilk18b; Rese20; Deva21; Clar19].

ID	Guideline	Framework	Redundancy	Description
MF.1	**F1**	*FAIR* Metrics	3	Identifier Uniqueness of the Metadata.
MF.2	**F1**	*FAIR* Metrics	3	Identifier of the metadata is persistent.
MF.3	**F1**	*FAIR* Metrics	3	Identifier of the data is persistent.
MF.4	**F1**	*FAIR* DMM	2	Data is identified by a globally unique identifier.
MF.5	**F2**	*FAIR* Metrics	2	Metadata must be structured (e.g., RDFa, JSON, JSON-LD, RDF-Turtle, . . .).
MF.6	**F2**	*FAIR* Metrics	3	Namespaces of the Metadata can be resolved.
MF.7	**F2**	*FAIR* DMM	1	Rich metadata is provided to allow discovery.
MF.8	**F3**	*FAIR* Metrics	3	Metadata contains the unique identifier to the data.
MF.9	**F3**	*FAIR* Metrics	2	Metadata contains the unique identifier to the metadata itself.
MF.10	**F4**	*FAIR* Metrics	5	A machine can discover the resource by search (using Microsoft Bing).
MF.11	**F**	FAIRshake dataset	0	The dataset is hosted in an established data repository, if a relevant repository exists.

Table A.2: Applicable and distinct *FAIRness* metrics for accessibility identified in literature [Wilk18b; Rese20; Deva21; Clar19].

ID	Guideline	Framework	Redundancy	Description
MA.1	**A1**	*FAIR* DMM	1	Metadata contains information to enable the user to get access to the data.
MA.2	**A1**	*FAIR* DMM	0	Metadata can be accessed manually (i.e. with human intervention).
MA.3	**A1**	*FAIR* DMM	0	Data can be accessed manually (i.e. with human intervention).
MA.4	**A1**	*FAIR* DMM	0	Metadata identifier resolves to a metadata record.
MA.5	**A1**	*FAIR* DMM	0	Data identifier resolves to a digital object.
MA.6	**A1**	*FAIR* DMM	0	Data can be accessed automatically (i.e. by a computer program).
MA.7	**A1.1**	*FAIR* Metrics	5	Data may be retrieved by an open and free protocol.
MA.8	**A1.1**	*FAIR* Metrics	5	Metadata may be retrieved by an open and free protocol.
MA.9	**A1.2**	*FAIR* Metrics	2	Tests data GUID for the ability to implement authentication and authorization in its resolution protocol.
MA.10	**A1.2**	*FAIR* Metrics	2	Tests metadata GUID for the ability to implement authentication and authorization in its resolution protocol.
MA.11	**A2**	*FAIR* Metrics	4	Metadata contains a persistence policy.

Table A.3: Applicable and distinct *FAIRness* metrics for interoperability identified in literature [Wilk18b; Rese20; Deva21; Clar19].

ID	Guideline	Framework	Redundancy	Description
MI.1	**I1**	*FAIR* Metrics	4	Metadata uses a formal language broadly applicable for knowledge representation (anything that can be represented as structured data will be accepted).
MI.2	**I1**	*FAIR* Metrics	4	Metadata uses a formal language broadly applicable for knowledge representation (Any form of RDF will pass this test).
MI.3	**I1**	*FAIR* Metrics	2	Maturity Indicator to test if the data uses a formal language broadly applicable for knowledge representation. This particular test takes a broad view of what defines a "knowledge representation language"; in this evaluation, a knowledge representation language is interpreted as one in which terms are semantically grounded in ontologies. Any form of structured data will pass this test.
MI.4	**I1**	*FAIR* Metrics	2	Maturity Indicator to test if the data uses a formal language broadly applicable for knowledge representation. This particular test takes a broad view of what defines a "knowledge representation language"; in this evaluation, a knowledge representation language is interpreted as one in which terms are semantically grounded in ontologies. Any form of ontologically grounded linked data will pass this test.
MI.5	**I2**	*FAIR* Metrics	2	Maturity Indicator to test if the linked data metadata uses terms that resolve. This tests only if they resolve, not if they resolve to *FAIR* data; therefore, it is a somewhat weak test.
MI.6	**I2**	*FAIR* Metrics	3	Maturity Indicator to test if the linked data metadata uses terms that resolve to linked (*FAIR*) data.
MI.7	**I2**	*FAIR* DMM	0	Data uses *FAIR*-compliant vocabularies.
MI.8	**I3**	*FAIR* Metrics	4	Maturity Indicator to test if the metadata links outward to third-party resources. It only tests metadata that can be represented as Linked Data.
MI.9	**I3**	*FAIR* DMM	0	Metadata includes references to other metadata.
MI.10	**I3**	*FAIR* DMM	0	Data includes references to other data.
MI.11	**I3**	*FAIR* DMM	0	Metadata includes references to other data.
MI.12	**I3**	*FAIR* DMM	0	Data includes qualified references to other data.
MI.13	**I3**	*FAIR* DMM	0	Metadata include qualified references to other data.

Table A.4: Applicable and distinct *FAIRness* metrics for reusability identified in literature [Wilk18b; Rese20; Deva21; Clar19].

ID	Guideline	Framework	Redundancy	Description
MR.1	**R1**	*FAIR* DMM	1	Plurality of accurate and relevant attributes are provided to allow reuse.
MR.2	**R1.1**	*FAIR* Metrics	3	Maturity Indicator to test if the linked data metadata contains an explicit pointer to the license. Tests: xhtml, dvia, dcterms, cc, data.gov.au, and Schema license predicates in linked data and validates the value of those properties.
MR.3	**R1.1**	*FAIR* Metrics	7	Maturity Indicator to test if the metadata contains an explicit pointer to the license. This 'weak' test will use a case-insensitive regular expression and scan both key/value style metadata and linked data metadata. Tests: xhtml, dvia, dcterms, cc, data.gov.au, and Schema license predicates in linked data and validates the value of those properties.
MR.4	**R1.1**	*FAIR* DMM	0	Metadata includes information about the license under which the data can be reused.
MR.5	**R1.2**	*FAIR* DMM	1	Metadata includes provenance information according to community-specific standards.
MR.6	**R1.2**	*FAIR* DMM	0	Metadata includes provenance information according to a cross-community language.
MR.7	**R1.3**	*FAIR* DMM	1	Metadata complies with a community standard.
MR.8	**R1.3**	*FAIR* DMM	1	Data complies with a community standard.
MR.9	**R1.3**	*FAIR* DMM	0	Metadata is expressed in compliance with a machine-understandable community standard.
MR.10	**R1.3**	*FAIR* DMM	1	Data is expressed in compliance with a machine-understandable community standard.
MR.11	**R**	FAIRshake dataset	0	Information is provided on the experimental methods used to generate the data.

B Ontologies considered but not used

Not all ontologies that have been investigated have been used for the sensor and contextualization models. This section summarizes these ontologies.

Digital System of Units

The Digital System of Units (D-SI) [Hutz20] is a framework for the machine-actionable representation of measurement data based on standardized conventions and guidelines for human-actionable (manual) formalization of measurement results, such as the VIM [BIPM12], Guide to the expression of uncertainty in measurement (GUM) [BIPM08], ISO:80000, Part 1 [ISO22], CODATA [Mohr15] and the SI-Brochure [BIPM19]. The document defines a controlled terminology for representing measurement values and required metainformation. The definitions are strongly influenced by metrological requirements. However, it lacks the provision of an easily accessible and automatically retrievable machine-actionable document of the definitions and terminology. It is the only reviewed ontology not listed at LOV.

OGC SensorThings API

The *OGC SensorThings API* [Lian16] provides a similar set of terms like SSN for the definition of data structures for measurement data. Moreover, it offers a REST API for accessing and manipulating data from devices for which a data structure is defined according to the SensorThings API's data model. The API also supports exchanging sensor data in various formats, including JSON and XML. Due to the API, it is possible to alter the data model and metainformation of a measuring system after deployment and in use.

Ontology of units of Measure

The Ontology of units of Measure (OM) [Rijg13] defines concepts for modeling physical quantities, units, and measurements. Based on a detailed domain analysis, OM defines a controlled vocabulary for the mentioned concepts and relations between these. Initially, it has been more expressive than QUDT, but QUDT has been largely extended and is adopted more often now.

Web of Things – Thing Description Ontology

A more general approach is the Web of Things (WoT)-Thing Description Ontology (TD), which does not only target measurement data and devices. It provides a data model describing the characteristics, capabilities, and behavior of IoT devices. Based on RDF and using JSON and JSON-LD as data formats, it paves the way for interoperable and compatible devices. On top of the description, there exist drafts for WoT *Profiles* [McCo23] and WoT *Discovery* [Cimm23], which deal with the provision of a RESTful HTTP API and the discovery and exploration of connected devices, respectively, based on a *Thing description*, i.e., a corresponding data model.

C Data structure for sensor data contextualization

Figure C.1 shows the data structure for sensor data contextualization developed, implemented, and validated in previous work [Bode23].

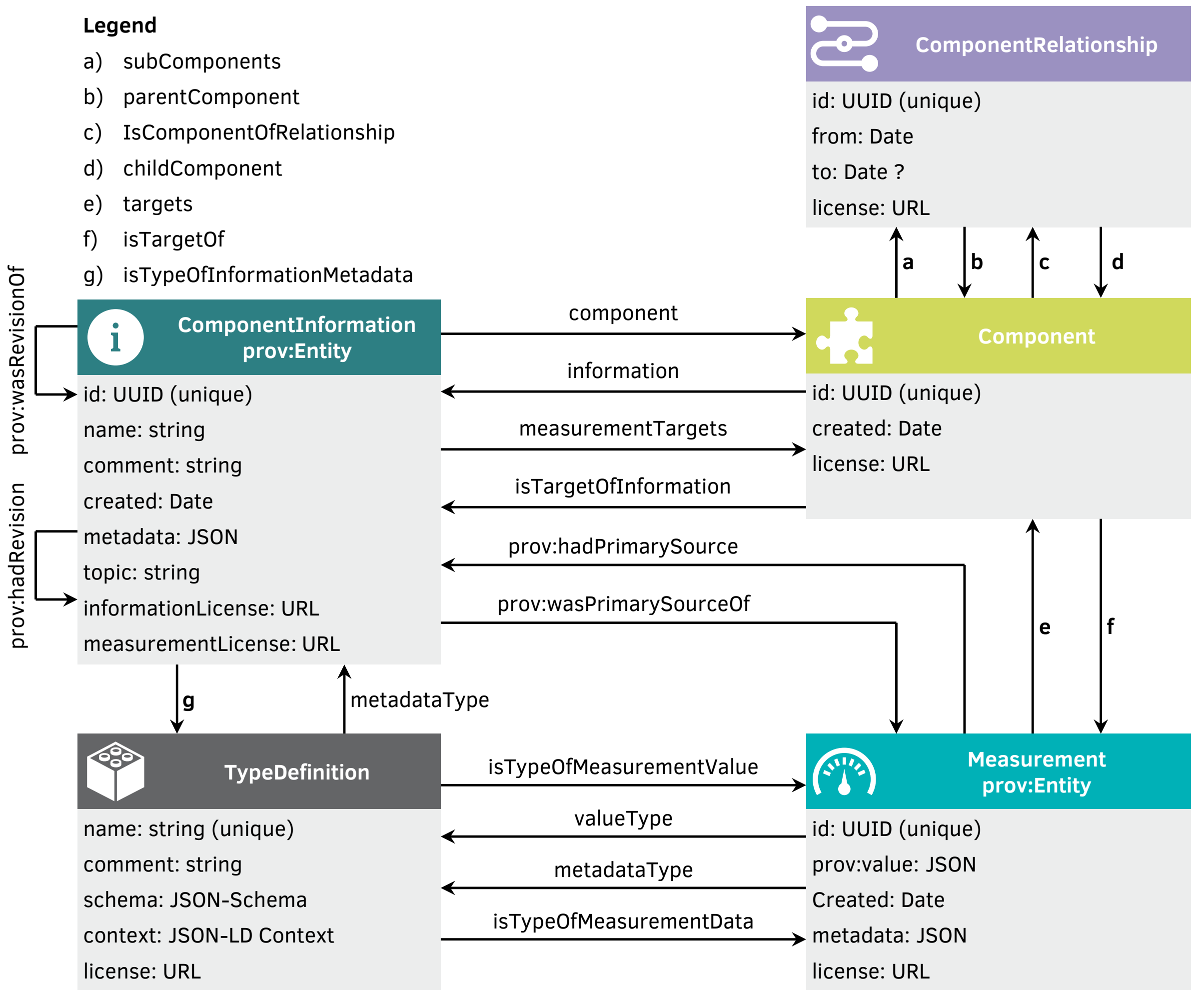


Figure C.1: The meta structure for the contextualization of the manufacturing sensor data provided by *FAIR Sensor Services*. The evaluation of the *FAIRness* of the data stored and provided by the *FAIR Sensor Ecosystem* is based on this structure.

The assessment of the *FAIRness* achieved with this data structure and the implemented *FAIR Sensor Ecosystem* has been executed with manual and (semi-)automated *FAIRness* assessment frameworks in spring 2022. Namely, the *FAIR Metrics* proposed by Wilkinson et al. [Wilk19b], the *RDA FAIR Data Maturity Model* [Bahi20] and the *Data Stewardship Wizard* [Perg19]. To evaluate the overall *FAIRness*, the individual results of all considered frameworks have been mapped to a percentage scale to be comparable. The outcome is shown in Table C.1.

Table C.1: Consolidated assessment of the *FAIRness* of data stored in the proposed *FAIR Sensor Ecosystem* using the data structure for contextualization depicted in Figure C.1 given in percent. Whereas the overall *FAIRness* ratings differ significantly from framework to framework, the average ratings for the four base principles are comparable.

Framework	F	A	I	R	**Average**
FAIR Metrics (manual) [Wilk19b]	100	100	60	100	**90**
FAIR Metrics (automatic) [Wilk19b]	25	20	71	100	**54**
Maturity Model [Bahi20]	100	100	20	0	**55**
Stewardship Wizard [Perg19]	80	83	90	84	**84**
Average	**76**	**76**	**60**	**71**	**71**

D SOIL model of an API Radian™ 3 laser tracker

The complete model of the API Radian™ laser tracker is redefined based on the dissertation of Montavon [Mont21] and consists of four files. The validation and generation tools implemented during this research have been verified by implementing a virtual, simulated API Radian™ laser tracker in Python.

```
@prefix schema: <http://schema.org/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;

import mobile_entities;
import utils;
import base_stations;


variable Time defines <schema:DateTime> {
    name: "Time"
    description: "Current system time."
    datatype: time
    dimension: []
}

variable Version defines <schema:version> {
    name: "Version"
    description: "Incremental API-Version."
    datatype: int
    range: (0, 100)
    dimension: []
    unit: <unit:UNITLESS>
}

variable Manufacturer defines <schema:manufacturer> {
    name: "Manufacturer"
    description: "Name of manufacturing company."
    datatype: string
    dimension: []
}

function Reset {
    name: "Reset"
    description: "Resets the device into the state like directly after start-up."
}

function Shutdown {
    name: "Shutdown"
    description: "Gracefully shutdown the device."
}

component Lasertracker {
    name: "Lasertracker"
    description: "Active coordinate measurement device based on laser interferometry for Large-Scale metrology applications."
    components:
        base_stations.BaseStations baseStations
```

```
        mobile_entities.MobileEntities mobileEntities
    functions:
        Reset reset
        Shutdown shutdown
    parameters:
        utils.State state = OK
        constant Manufacturer manufacturer = "API"
        constant Version version = 1
        Time time

    if state == ERROR: error("An error occured!")
}

interface Lasertracker APIRadian {}
```

Listing D.1: The main file of the SOIL model of the API Radian™ laser tracker. It defines the root component type `Lasertracker` and the interface and imports the three other files of the model.

```
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;

import utils;

variable Azimuth  defines <quantitykind:Angle> {
    name: "Azimuth"
    description: "Current position of azimuth rotation encoder in Radian."
    datatype: float
    dimension: []
    unit: <unit:RAD>
    range: (-5.59, 5.59) # ~ (-320, 320) degrees
}

variable Elevation defines <quantitykind:Angle> {
    name: "Elevation"
    description: "Current position of elevation rotation encoder in Radian."
    datatype: float
    dimension: []
    unit: <unit:RAD>
    range: (-1.03, 1.38) # ~ (-59, 79) degrees
}

variable Distance  defines <quantitykind:Distance> {
    name: "Distance"
    description: "Measured distance to the currently activate target."
    datatype: float
    dimension: []
    unit: <unit:M>
    range: (0, 100)
}

variable Interval defines <quantitykind:Time> {
    name: "Interval"
    description: "Interval in seconds."
    datatype: float
    dimension: []
    unit: <unit:SEC>
```

```
    range: (0, 360)
}

function Jog {
    name: "Jog"
    description: "Jogs the tracker head by the given angles for the azimuth and elevation."
    arguments:
        Azimuth azimuth
        Elevation elevation
}

function PointTo {
    name: "Point to"
    description: "Moves the tracker head so that the laser points to the specified position."
    arguments:
        utils.Position position
}

component Base {
    name: "Base"
    description: "Represents a base station in a distributed system."
    functions:
        Jog jog
        PointTo point_to
    parameters:
        utils.State state = OK
        Interval interval = 10
    measurements:
        internal utils.Position position
        internal utils.Quaternion quaternion
        internal Azimuth azimuth
        internal Elevation elevation
        internal Distance distance

    streams:
        position: update
        quaternion: update

    if state == ERROR: error("An error of the trackers base occured.")
    if distance > 50.0: warning("Distance is very high. Measurements have high uncertainty.")

    position.description = "Position of the base of the tracker within a global reference frame."
    quaternion.description = "Orientation of the base in relation to the orientation of a global reference frame."
}

component BaseStations {
    name: "Base Stations"
    description: "Object acting as a list of base stations of the metrology system."
    components:
        Base base
```

```
}
```

Listing D.2: Definition of the base station of the API Radian™ laser tracker with all its specific parameters and measurements.

```
import utils;
@prefix unit: <http://qudt.org/vocab/unit/> ;

enum Mode {
    CONTINUOUS
    TRIGGERED
    EXTERNAL
    IDLE
}

variable Mode {
    name: "Mode"
    description: "Current state of the entity. In CONTINUOUS mode, values are
    dispatched as fast as possible. In TRIGGERED mode, values are only
    dispatches after a software trigger. In EXTERNAL mode, values are
    dispatched in accordance to an external trigger, e.g. probe or TTL. IDLE
    means the entity is currently not used."
    datatype: enum
    range: Mode
    dimension: []
}

variable Type {
    name: "Type"
    description: "System specific identifier of the target Type, e.g. SMR or
    Active SMR."
    datatype: string
    dimension: []
}

variable Locked {
    name: "Locked"
    description: "Boolean flag specifying whether the target is locked in."
    datatype: boolean
    dimension: []
}

function Reset {
    name: "Reset"
    description: "Starts the search routine around the current direction."
}

variable Counter {
    name: "Counter"
    description: "Natural numeric value."
    datatype: int
    dimension: []
    range: (1,1000)
    unit: <unit:UNITLESS>
}

```

```
variable Label {
    name: "Label"
    description: "A string serving as comment."
    datatype: string
    dimension: []
}

function Trigger {
    name: "Trigger"
    description: "Trigger count variables and set the resulting label. This
    function is only allowed in triggered acquisition mode."
    arguments:
        Counter counter = 100
        Label label = "test"
    returns:
        utils.Position position
        Label outputLabel
}

component Target {
    name: "Target"
    description: "Represents an individual mobile entity."
    functions:
        Reset reset
        streaming Trigger trigger
    measurements:
        utils.Position position
        utils.Quaternion quaternion
        internal Locked locked
    parameters:
        utils.State state = OK
        Mode mode = CONTINUOUS
        Type type = "SMR"

    streams:
        position: update
        quaternion: update

    state.description = "Reflects the current state of the target. If logged in
    : OK. If not stable: WARNING. If lost: ERROR."
}

component MobileEntities {
    name: "Mobile Entities"
    description: "Object acting as a list of mobile entities in the metrology
    system."
    components:
        dynamic Target target
}
```

Listing D.3: Definition of the mobile entities, i.e. the SMR targets tracked by the API Radian™ laser tracker. As the tracker might track and measure multiple different targets (successively), the mobile entities' component is modified to be dynamic.

```
@prefix dbo: <http://dbpedia.org/ontology/> ;
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> ;
```

```
@prefix unit: <http://qudt.org/vocab/unit/> ;

enum StateRange {
    OK
    WARNING
    ERROR
    MAINTENANCE
}

variable State defines <dbo:status> {
    name: "State"
    description: "The current state of the device."
    datatype: enum
    range: StateRange
    dimension: []
}

variable Position defines <quantitykind:CartesianCoordinates> {
    name: "Position"
    description: "Most recently dispatched measured position."
    datatype: float
    dimension: [3]
    range: (-100, 100)
    unit: <unit:M>
}

variable Quaternion defines <quantitykind:Angle> {
    name: "Quaternion"
    description: "Most recently dispatched measured orientation as quaternion,
    if available."
    datatype: float
    dimension: [4]
    range: (0, 1)
    unit: <unit:UNITLESS>
}
```

Listing D.4: The utils file of the SOIL model of the API Radian™ laser tracker. It defines utility variables and enums, referenced by the mobile entities and the base station of the laser tracker.

E SOIL model of a Leitz PMM-C Precision™ wrapped by an I++ DME interface

The partial model of a CMM encapsulated by an I++ DME server consists of five files. The verification of the validation and generation tools implemented during this research has been carried out by wrapping a reference implementation of an I++ DME server in Python.

```
import cartCMM;

function AbortE {
    name: "AbortE"
    description: "Aborts all pending transactions and stops the machine from moving."
}

function ClearAllErrors {
    name: "ClearAllErrors"
    description: "Is called to recover from an error."
}

component Ipp {
    name: "I++"
    description: "The top-level component of the Server."
    components:
        cartCMM.CartCMM cmm
    functions:
        AbortE abortE
        ClearAllErrors clearAllErrors
}

interface Ipp IppServer {}
```

Listing E.1: The main file of the SOIL model of the I++ DME reference server. It defines the root component type Ipp and the interface.

```
import tool;
import toolChanger;
import part;

enum CoordinateSystems {
    MachineCsy
    MovableMachineCsy
    MultipleArmCsy
    RotaryTableVarCsy
    PartCsy
}

variable ActiveCsy {
    name: "ActiveCsy"
    description: "The currently active coordinate system."
    datatype: enum
    dimension: []
    range: CoordinateSystems
}

```

```
component CartCMM {
    name: "CartCMM"
    description: "Cartesian Coordinate Measuring Machine"
    components:
        toolChanger.ToolChanger toolChanger
        part.Part part
    parameters:
        tool.Id activeTool
}
```

Listing E.2: The partial SOIL model of the class “CartCMM” of I++ DME.

```
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;

variable Name {
    name: "Name"
    description: "Name of a Tool"
    datatype: string
    dimension: []
}

variable Id {
    name: "Id"
    description: "Id of a Tool"
    datatype: string
    dimension: []
}

variable Position defines <quantitykind:CartesianCoordinates> {
    name: "Position"
    description: "Current position of the tool, consisting of the I++ X, Y and
    Z values."
    datatype: float
    dimension: [3]
    range: (0,1000)
    unit: <unit:MilliM>
}

function ScanOnCurve {
    name: "ScanOnCurve"
    description: "Function that issues a ScanOnCurve command on the I++ Server"
    arguments:
        List Position posL
}

function PtMeas {
    name: "PtMeas"
    description: "Function that issues a PtMeas command on the I++ Server"
    arguments:
        Position pos
}

component Tool {
    name: "Tool"
    description: "The currently active Tool."
```

```
    measurements:
        Position pos
    parameters:
        constant Name name = "Tool"
        constant Id id = "T1"
        List Position posL
    functions:
        PtMeas ptMeas
        ScanOnCurve scanOnCurve
}
```

Listing E.3: The SOIL model of those properties and commands of the class “Tool” of I++ DME implemented in the reference implementation.

```
import tool;

function EnumTools {
    name: "EnumTools"
    description: "Function that issues an EnumTools command on the I++ server."
}

function ChangeTool {
    name: "EnumTools"
    description: "Function that issues an ChangeTool command on the I++ server.
    "
    arguments:
        tool.Id tool_id
}

component ToolChanger {
    name: "ToolChanger"
    description: "ToolChanger component"
    components:
        dynamic tool.Tool tools
    functions:
        EnumTools enumTools
        ChangeTool changeTool
}
```

Listing E.4: The SOIL model of those properties and commands of the class “ToolChanger” of I++ DME implemented in the reference implementation.

```
@prefix quantitykind: <http://qudt.org/vocab/quantitykind/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;

variable Temperature defines <quantitykind:Temperature> {
    name: "Temperature"
    description: "Current temperature of the part."
    datatype: float
    dimension: []
    range: (-50, 200)
    unit: <unit:DEG_C>
}

variable Approach defines <quantitykind:Distance> {
    name: "Approach"
    description: "Currently specified approach distance of the part."
```

```
    datatype: float
    dimension: []
    range: (0, 500)
    unit: <unit:MilliM>
}

component Part {
    name: "Part"
    description: "Soil representation of a Part."
    measurements:
        Temperature temperature
    parameters:
        Approach approach
}
```

Listing E.5: The SOIL model of those properties and commands of the class “Part” of I++ DME implemented in the reference implementation.

F SOIL model of an OptiTrack™ motion capture system

The SOIL model contains the characteristics, measurements, and parameters of an OptiTack MCS relevant for using the system as the global reference tracking system of mobile assets. The model reuses the definition of basic attributes from the API Radian™ laser tracker model. For that, the model imports the file *utils.soil* shown in Listing D.4.

```
@prefix schema: <http://schema.org/> ;

import utils;
import Camera;
import Marker;
import Frame;
import RigidBody;

variable Time defines <schema:DateTime> {
    name: "Time"
    description: "Current UTC system time."
    datatype: time
    dimension: []
}

variable APIVersion  defines <schema:version> {
    name: "API Version"
    description: "NatNet SDK Version"
    datatype: string
    dimension: []
}

variable Manufacturer defines <schema:manufacturer> {
    name: "Manufacturer"
    description: "Name of the manufacturer of the device."
    datatype: string
    dimension: []
}

component OptiTrackSystem {
    name: "OptiTrack"
    description: "OptiTrack motion capture system"
    components:
        Frame.Frame frame
        Camera.Cameras cameras
        Marker.Markers markers
        RigidBody.RigidBodys rigidBodys

    parameters:
        utils.State state = OK
        constant Manufacturer manufacturer = "NaturalPoint Inc."
        APIVersion version
        Time time
}

interface OptiTrackSystem OptiTrack {}
```

Listing F.1: The main file of the OptiTrack™ model. It defines the root component type `OptiTrackSystem` and the interface.

```
@prefix quantitykind: <https://qudt.org/vocab/quantitykind/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;
@prefix schema: <http://schema.org/> ;
@prefix dbo: <http://dbpedia.org/ontology/> ;

import utils;

enum MarkerTypes {
    ACTIVE
    UNLABELED
    LABELED
}

variable Type defines <schema:type> {
    name: "Type"
    description: "Type of the marker"
    datatype: enum
    range: MarkerTypes
    dimension: []
}

variable ID defines <schema:identifier> {
    name: "Marker ID"
    description: "ID of the marker"
    datatype: string
    dimension: []
}

variable Name defines <schema:name> {
    name: "Marker Name"
    description: "Name of the marker"
    datatype: string
    dimension: []
}

variable Size defines <quantitykind:Diameter> {
    name: "Marker Size"
    description: "Size of the marker"
    datatype: float
    range: (0.0,)
    dimension: []
    unit: <unit:M>
}


component Marker {
    name: "Marker"
    description: "Detected Marker"

    parameters:
        Name name
        ID id #readonly

    measurements:
        utils.Position position
        Size size
```

```
    streams:
        position: update
        size: update

}

component Markers {
    name: "Markers"
    description: "List of detected Markers"

    components:
        dynamic Marker marker
}
```

Listing F.2: The SOIL model for the individual markers being detected and tracked by the OptiTrack™ MCS.

```
@prefix quantitykind: <https://qudt.org/vocab/quantitykind/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;
@prefix qudt: <http://qudt.org/schema/qudt/>;
@prefix schema: <http://schema.org/> ;
@prefix ebucore: <http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#> ;

variable Latency defines <quantitykind:Time> {
    name: "Latency"
    description: "Measured latency"
    datatype: float
    range: (0.0,)
    dimension: []
    unit: <unit:MilliSEC>
}

variable Timestamp defines <qudt:DATETIME> {
    name: "Timestamp"
    description: "Timestamp of the current frame"
    datatype: float
    range: (0.0,)
    dimension: []
    unit: <unit:MilliSEC>
}


variable FrameID defines <schema:identifier> {
    name: "Frame ID"
    description: "ID of the current frame"
    datatype: int
    range: (0,)
    dimension: []
    unit: <unit:UNITLESS>
}

component Frame defines <ebucore:Keyframe>{
    name: "Frame"
    description: "Information about the current frame"

```

```
    measurements:
        internal FrameID frameID
        internal Timestamp timestamp
        internal Latency softwareLatency
        internal Latency systemLatency
        internal Latency transitLatency
        internal Latency clientLatency

    streams:
        frameID: update
        timestamp: update
        softwareLatency: update
        systemLatency: update
        transitLatency: update
        clientLatency: update

    softwareLatency.name = "Software Latency"
    systemLatency.name = "System Latency"
    transitLatency.name = "Transit Latency"
    clientLatency.name = "Client Latency"
}
```

Listing F.3: The SOIL model of an individual frame acquired by the OptiTrack™ MCS provides information about the systems latency.

```
@prefix schema: <http://schema.org/> ;
@prefix dbo: <http://dbpedia.org/ontology/> ;

import utils;

variable CameraName defines <schema:name> {
    name: "Name"
    description: "Name of the camera"
    datatype: string
    dimension: []
}

component Camera defines <dbo:Camera> {
    name: "Camera"
    description: "Connected Camera"

    parameters:
        CameraName name
        # Weiter Werte mit MotiveAPI
    measurements:
        internal utils.Position position
        internal utils.Quaternion orientation
}

component Cameras {
    name: "Cameras"
    description: "List of connected Cameras"

    components:
        dynamic Camera camera
```

```
}
```

Listing F.4: The SOIL model of all required characteristics of a single camera of the OptiTrack™ MCS.

```
@prefix quantitykind: <https://qudt.org/vocab/quantitykind/> ;
@prefix unit: <http://qudt.org/vocab/unit/> ;
@prefix schema: <http://schema.org/> ;
@prefix dbo: <http://dbpedia.org/ontology/> ;

import utils;

variable Name defines <schema:name> {
    name: "Name"
    description: "Name of the rigid body"
    datatype: string
    dimension: []
}

enum TrackingStates {
    VALID
    NOTVALID
}

variable TrackingState defines <dbo:status> {
    name: "State"
    description: "State of the rigid body"
    datatype: enum
    range: TrackingStates
    dimension: []
}

variable MeanError defines <quantitykind:Distance> {
    name: "Mean Error"
    description: "Mean Error of the ridid body"
    datatype: float
    range: (0.0,)
    dimension: []
    unit: <unit:M>
}


component RigidBody {
    name: "Rigid Body"
    description: "Tracked Rigid Body"

    parameters:
        Name name
        TrackingState state
    measurements:
        utils.Position position
        utils.Quaternion orientation
        MeanError meanError
    streams:
        position: update
        orientation: update
        meanError: update
```

```
}

component RigidBodys {
    name: "RigidBodys"
    description: "List of detected RigidBodys"

    components:
        dynamic RigidBody rigidBody
}
```

Listing F.5: The SOIL model of a rigid body representing a mobile asset, tracked and located by the OptiTrack™ MCS. The rigid body is constituted of a set of at least three markers, which have a fixed position related to each other marker of the rigid body.

G Architecture of the infrastructure mapped to RAMI 4.0

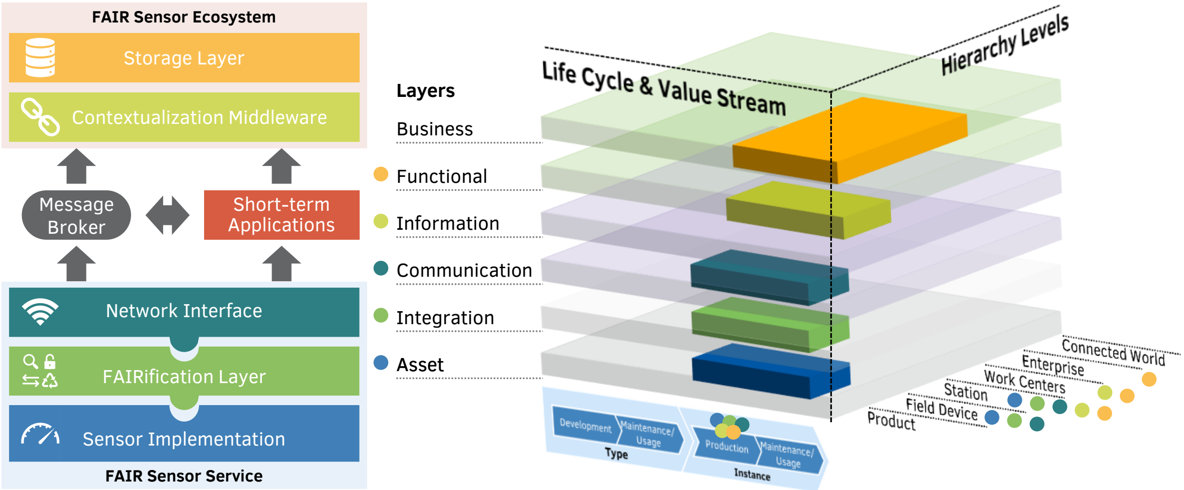


Figure G.1: The architecture of the infrastructure for *FAIR manufacturing sensor data* mapped to the RAMI 4.0 [Adol15]. It covers almost all layers and hierarchy levels of the production step of the life-cycle and value stream.

H Compliance with the *FAIR Guiding Principles*

Table H.1: Mapping of the qualitative evaluation of the 17 *FAIRness* criteria (cf. Table 6.1) to the *FAIR Guiding Principles* by Wilkinson [Wilk19a] based on Table 3.2. If a single guideline of the *FAIR Guiding Principles* is addressed by more than criteria, the average score is computed on a scale from 0 to 1. The total score of each of the four base principles, i.e., findability, accessibility, interoperability, and reusability, is computed as the average of the respective guidelines. The last two columns are the base for Figure 7.1.

		Findability				Accessibility				Interoperability			Reusability			
		F1	F2	F3	F4	A1	A1.1	A1.2	A2	I1	I2	I3	R1	R1.1	R1.2	R.13
Findability	CF.1		●													
	CF.2				◕											
	CF.3	●		●												
Accessibility	CA.1					●	●									
	CA.2					●	●									
	CA.3						●									
	CA.4							◑								
	CA.5								●							
Interoperability	CI.1									●	●					
	CI.2									●						
	CI.3									●						
	CI.4			◕						◕			◕			
	CI.5											◕				
Reusability	CR.1			◔												◔
	CR.2												●		●	
	CR.3			◕												◕
	CR.4													●		
Average		●	●	◕	◕	●	●	◑	●	◕	●	◕	◕	●	●	◑
Total		◕				◕				◕			◕			

I Definition of the OPC UA NodeType *RotaryTableType*

The following Table I.1 and descriptions are word by word taken from the OPC UA Companion Specification (CS) *Geometric Measuring System* [VDMA23, p. 39]. The table and the descriptions exemplary illustrate the translation from an OPC UA node type to a SOIL model (cf. Listing 7.1).

Table I.1: Definition of the *RotaryTableType* from the OPC UA CS *Geometric Measuring Systems*, as given in [VDMA23, Table 32, p. 39]

Attribute	Value				
BrowseName	RotaryTableType				
IsAbstract	False				
References	Node Class	BrowseName	DataType	TypeDefinition	Other
Subtype of the 0:BaseObjectType defined in OPC 10000-5 i.e., inheriting the InstanceDeclarations of that Node.					
0:HasAddIn	Object	2:Identification		3:MachineryItemIdentificationType	0:Mandatory
0:HasProperty	Property	IsIntegrated	0:Boolean	0:PropertyType	0:Mandatory
0:HasProperty	Property	NumberOfAxes	0:Byte	0:PropertyType	0:Mandatory
Conformance Units					
GMS RotaryTableType					

Identification is performed according to MachineryIdentificationType.

IsIntegrated is defined as True if the RotaryTable is integrated into the GMS and cannot be removed.

NumberOfAxes defines the number of axes (DegreesOfFreedom) the RotaryTable can be configured for.