Open reference architecture for engineering model spaces

Deliverable 3.1

Review of existing standards and ontologies

Project Coordinator	Olga Kattan, Philips Consumer Lifestyle B.V.		
Start date Project	April 1st 2024	**Duration**	36 months
Version	1.0		
Status	Final		
Date of issue	20-02-2025		
Dissemination level	Public		

Authors' data

Author	Beneficiary	
Klaus Wolf	Fraunhofer SCAI	
Roland Mathijssen	TNO	
Dilara Bayar	Acd Bilgi Islem ltd.sti.	
	Alpata Technology	
Muhammed Oğuz Taş	Inovasyon Muhendislik	
	Philips Consumer Lifestyle B.V.	
	SemLab	
	Thermo Fisher Scientific	
Final editor's address	Olga Kattan Philips Consumer Lifestyle B.V. Oliemolenstraat 5 9203ZN Drachten / Netherlands	

Executive Summary

This document gives an overview of the current or existing standards and ontologies that are related to the 5 use cases in this project. Next to that it lists standards and ontologies that exist and may be of use to the use cases or the generic problem that is being solved in this project.

This document does not have the intention of giving an extensive or complete overview of all the available standards and ontologies. As this document is written in the early phases of the project, we do envision that during this project, the document can become a living document, where applicable standards and ontologies will be added, and less applicable ones might even be removed, while the use cases are worked out further. Depending on the status of this living document, a revised version of this document may be published near the end of this project.

Further, this document mostly lists the applicable standards and ontologies. It is not the intention of this document to give an extensive overview or explanation of them. Only where applicable some brief summary or explanation may be given. The full or more elaborate context should be found at the standards and ontologies themselves. This document will try to provide a rationale why a certain document is useful for a certain use case.

As the use cases are used to get an overview of the standards and ontologies, the structure of this document also follows the use cases. Similar to the standards, the use cases are not described extensively. Only a brief summary will be given for the sake of readability. For the actual content, the documents about the use cases is always guiding.

Table of Contents

Table of Figures

1. Introduction

1.1. Purpose of this document

This document gives an overview of standards and ontologies, as gathered in the beginning of this project, and related to the use cases as described in the project plan. As described in the project plan, this document is defined for the following purpose:

- The required software infrastructure will follow several layers of standardization. Since SmartEM will rather build upon existing solutions than initiate a new standard from scratch, these layers must be ordered and reviewed in T3.1.
- They comprise: domain-specific technological standards (focus in use cases), standards for model descriptions and architectures, be it data models (e.g. ONNX) or physics-based models (e.g. SMILE), model exchange standards (e.g. FMI), model system standards (e.g. SSP) and neutral model formats (e.g. VMAP, STEP, BIM).
- The review identifies opportunities to reuse, extend or align a model-centred digital ecosystem with these standards, as well as numerous de-facto standards, community conventions, industrial guidelines and industry standards from standardisation bodies (ISO, DIN) that imply commonplace terminology and core taxonomies. During the past decades, formal ontologies have gained more popularity with the emergence of CatenaX, SAMM and their derivatives, data space schemata (e.g. IDS ontology) and others.
- T3.1 investigates connection points and alignment strategies. In close cooperation with T6.2, links to external standardisation and engineering communities will be established – the results and further planning are recorded in D6.1.

1.2. Related documents

SmartEM Project plan
- Full Project Proposal Annex SmartEM
- Open reference architecture for smart engineering model spaces
- Edited by: Klaus Wolf, Fraunhofer SCAI
- Date: February 16, 2023
- 22009_SmartEM_FPP_Annex-2023-02-16 1200 Auto Generated Merged FPP

Deliverable D2.1
- SmartEM Use Case Descriptions
- D2.1_Use Case Descriptions_v1.0.pdf
- October 11, 2024
- Submitted

Deliverable D2.1
- SmartEM Use Case Requirements
- D2.2_Use Case Requirements_v1.0.pdf
- October 11, 2024
- Submitted

1.3. Scope

The following image sketches the scope of standards and ontologies, using a fictive example.

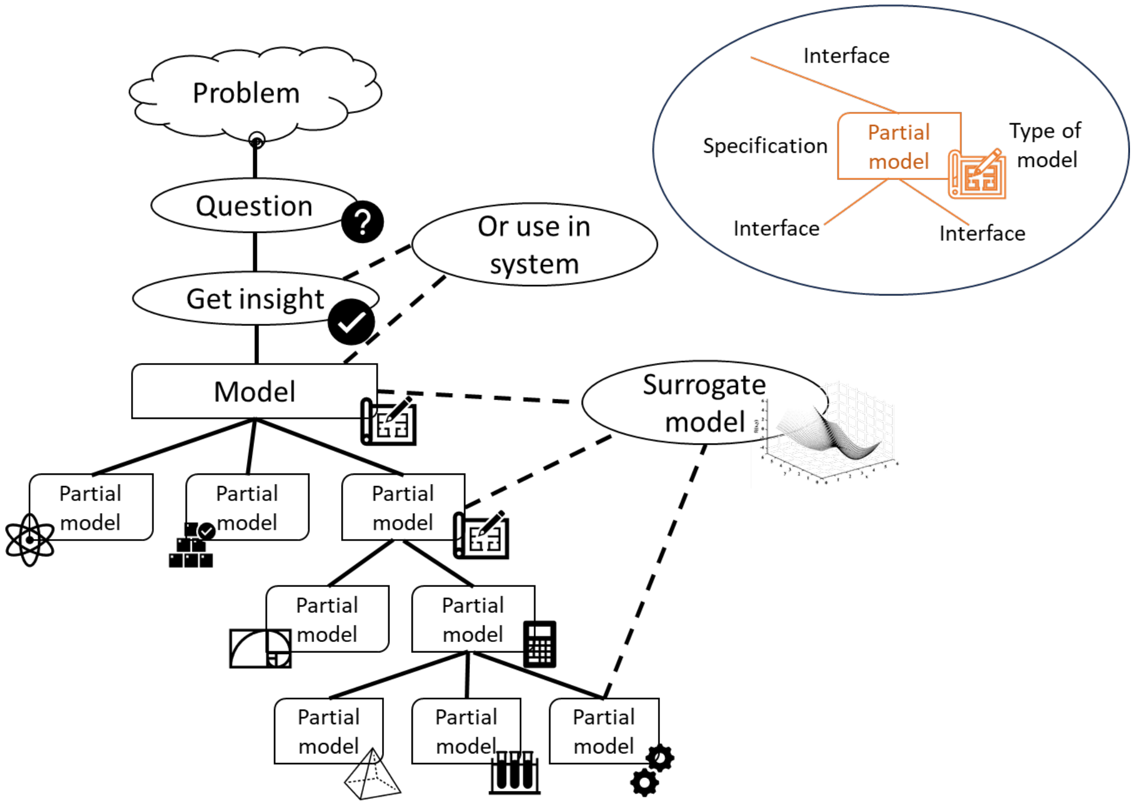


Figure 1 : Fictive example to explain the scope of this document

In this figure we start with a problem that needs to be solved. This problem can be anything, but for this document it is an engineering problem that needs to be solved or understood. The problem will generate a question (or a set of questions), used to get insight or to predict behaviour. In order to get this insight of behaviour, a model is made.

This model can consist of a set of multiple models. As an example, when something drops to the floor, we all know the law of gravity, to determine the force on the object. However to determine the speed, we also need the other Newtonian laws, to derive the acceleration, and from that to derive the speed at a certain moment in time or place. So all together we already need three mathematical or physical models.

However, with only these models, we are not necessarily there yet. The Newtonian laws are only valid in an ideal situation, where there are no other external factors. The moment one begins to also consider e.g. friction or the effect of the atmosphere, the model becomes even more complex, and other partial models need to be added to get a sufficiently correct outcome or prediction. This effect of needing or desiring a multitude of models is sketched in the figure above. But also as certain models or combination of models can become quite complex or hard to compute, surrogate models can be used, that are sufficiently valid in a defined working range.

Another effect that is sketched in the figure is that models, in order to be connected, need interfaces that match one another. In the earlier simple example, the Newtonian laws can interface using the numerical results of e.g. Force, Acceleration, Velocity, Mass, etc. In a more complex problem, these interfaces may not always be so clear. Or even in a computational model, the tools being used to execute two models may not be the same, and that makes interfacing even more challenging.

And that is where we touch upon standards. Standards can help connecting models and interfacing them. And ontologies help in describing the models, so that you know what to expect from models.

A final aspect in this figure to discuss is that the top level model is probably rather specific for a certain domain or product or system. The chance that an off the shelf solution is available is probably not so large. These models will probably be made or composed for one specific use, and trying to find them elsewhere will be hard, if not impossible. The partial models however can be standard models or standardized. Compare this to the software domain, where more and more code snippets are available, e.g. in Java, C++, C#, etc. The code snippets are characterized by properly defined interfaces and description of the function. As a software developer you can quickly add a function to your code. Similarly one can use smaller models, or “model snippets” to construct or compose most of your larger scale model. As long as these snippets are defined in a standardized way.

Another aspect to scope this document is that the final model -that will probably not be reused at other companies- can or will be reused in the own company. And in order to be able to reuse or improve it, it needs to also comply with proper specifications and documentation. And here again, standardization will help.

And finally, the sketched surrogate model, that helps in speeding up getting results, will need input data in order to be created. Here too, standardizing data, measurements, etc. is needed in order to generate proper surrogate models. And once this surrogate model is available, it also needs the right (standardized) documentation and interfacing information.

In the next chapter we will dive in more detail into these aspects, using the more concrete use cases of SmartEM.

1.4. Definition of a Model in SmartEM context

The term Model has many different meanings. Sometimes the context clarifies to some extend the intended meaning, but even then, miscommunication is lurking.

The wiki disambiguation page for Model already shows quite some different concepts of a Model

https://en.wikipedia.org/wiki/Model_(disambiguation)

In this project we focus on technical models used in engineering. And even then, many different meanings or “mental models” are possible for the definition of a model. So in SmartEm we specifically consider models used in engineering that can be interpreted or handled by computers and software, and that are executable by the proper software tools.

The actual use of the models can be quite broad, e.g. for gaining insight or understanding, or for predicting behaviour or outcomes, in many domains, such as physics, chemistry, optics, software behaviour, etc. Also the execution time is not discriminating in this project, as some models will e.g. compute in real time or even faster, while other models can take hours or days or weeks to compute.

In the end though, one of the things SmartEM is looking into is how to speed up computing time, by the creation and usage of surrogate models. These are models that not need to have a one-on-one relationship with the physical world, but models that are derived from e.g. the actual world (measurements) or from long running outcomes of more accurate or precise models. This may cause that these surrogate models are less accurate, but perform a lot faster.

Summarizing: when we talk about models in SmartEM, we talk about executable models that can be handled by software tools and that can be used in the systems engineering process.

Do note that -to make matters more complicated- when discussing e.g. the acceptance of models, tools, surrogate modelling, etc., we may describe this process using e.g. social models or mental models. In that case, these are not the SmartEM models that this project is about, but something needed to explain stuff.

2. Use case specific data and models

Based on the information already provided in D2.1 (Use Case Definitions) and D2.2 (Use Case Requirements) we collected a condensed overview of the UC details. The purpose is have specific UC information in the same document as the overview discussion in standards. The focus in this document is on the IT related details for data and model management & exchange – and less an UC specific engineering standards and norms.

The following use cases are considered in this document:

UC1: Engineering Model Space for the design of Hydrogen Turbines
Owner of the use case: Siemens Industry Software (SISW) (Nicolas Lammens)

UC2: Personalization of Consumer Products use case
Owner of the use case: Philips (Olga Kattan)

UC3: Digital Twins for Professional Digital Printers
Owner of the use case: CPP (Roelof Hamberg)

UC4: Thermo Fisher Scientifc Use Cases
Owner of the use case: ThermoFisher Scientific (Tomas Molina)

UC5: Machine Downtime Reduction in Aviation
Owner of the use case: Alpata Technology (Murat Saglam)

3. Standards in Design, Modelling and Characterisation

There is a whole range of standards in the various engineering disciplines and application areas. In this document, we provide an initial overview of the standards relevant to SmartEM, their areas of application and basic concepts, as well as their industrial use (where known)

3.1. Standards on Data Level

3.1.1. QiF for Metrology

QIF is an open CAD format developed by the Dimensional Metrology Standards Consortium (DMSC) that meets the requirements mentioned above for the application of semantic MBD workflows. It was introduced to facilitate the propagation of 21st Century concepts such as digital transformation, digital thread, digital twin, and Industry 4.0 to computer-aided technology and engineering applications. In line with this, it enables interoperability of manufacturing dimensional quality control data between system software components. QIF enables the capture, use, and re-use of metrology-related information throughout the Product Lifecycle Management (PLM) and Product Data Management (PDM) domains.

QIF defines an integrated set of information models which enable the effective exchange of dimensional metrology data throughout the entire manufacturing quality measurement process – from product design to inspection planning to execution to analysis and reporting. Based on the XML Standard, it contains a Library of XML Schema ensuring both data integrity and data interoperability in Model Based Enterprise implementation.

Organization, and association quality information, including measurement plans, results, part geometry and product manufacturing information (PMI), measurement templates, resources, statistical analysis, etc. can be enabled by QIF.

An important advantage driven by QIF from a metrological point of view is the ability to store all necessary information -- from design/tolerances to measurements, results, and statistics.

The continuous lifecycle of a QIF model is shown in the figure. The starting point is the QIF derivative, which is validated to fully represent the authority model. The QIF plans are used to design the measurement process based on the features to be measured and how this will take place. This process is sometimes referred to as inspection planning or control planning. QIF also has other attributes, such as the ability to assign resources and rules related to the measurement process, which are outside the scope of the current application, thus will not be covered in detail.

For upstream metrology applications, QIF serves as a container for the results and maps these back to the original model. The results can have different forms, originating from different measurements processes (manual, CMM, etc.). With this approach, the measurement results remain linked to the original model (CAD+PMI), and all ambiguities are eliminated. The results can be presented together with the 3D model, further statistical operations can be performed, and they can be used as factual data to iteratively improve tolerance simulations.

3.1.2. Material Characterisation Information

Data formats used to store and manage materials characterization information vary depending on the type of data, the analytical techniques, and the software tools involved.

Here's a breakdown of commonly used formats:

1. Generic Data Storage Formats
    a. CSV (Comma-Separated Values): Simple text-based format ideal for tabular data.
    b. XLS/XLSX: Excel formats for structured tables, including metadata and graphs.
    c. JSON: Lightweight, flexible format for hierarchical or nested data
    d. XML: Structured and extensible format, often with schemas tailored to materials data
    e. HDF5 (Hierarchical Data Format): Binary format for handling large datasets with hierarchical organization
    f. TXT: Plain text format for basic data logging.
2. Specialized Formats for Materials Characterization
    a. CIF (Crystallographic Information File): Standard for crystallographic structure data.
    b. VASP Files (e.g., POSCAR, CONTCAR, OUTCAR): For material simulations in computational tools.
    c. EBSD Formats (e.g., .ang, .ctf): For electron backscatter diffraction data.
    d. XDATCAR: For molecular dynamics simulation trajectories in VASP.
    e. XSF (XCrySDen Structure File): For visualization of molecular and crystallographic structures.
3. Imaging and Spectroscopy Data Formats
    a. TIFF: Lossless format suitable for high-resolution images (e.g., microscopy).
    b. PNG/JPEG: Common image formats, though lossy for data storage.
    c. DAT: Generic format often used for spectroscopy data (e.g., Raman, FTIR)
    d. RAW: Sensor-level data from imaging devices.
    e. HIS (Hyperspectral Image File): For spectral imaging.
4. Computational Data Formats
    a. NetCDF (Network Common Data Form): For multi-dimensional scientific data, especially simulation outputs.
    b. FHI-aims Files: Specific to atomistic simulation outputs from the FHI-aims code.
    c. LAMMPS Dump Files: For molecular dynamics data storage.
5. Standardized Exchange Formats
    a. OPeNDAP: For distributing and sharing scientific datasets over the web.
    b. Materials Project JSON: Standardized output from Materials Project databases

3.1.3. Geometric properties and deviations

These standards help ensure that geometric data is interoperable, machine-readable, and capable of accurately representing shapes, tolerances, and deviations.

1. ISO Standards
    a. ISO 10303 (STEP):
        i. A widely adopted standard for the exchange of product data, including geometric shapes and their properties.
        ii. STEP-Tolerances (Part 242): Specifically covers the representation of geometric tolerances (e.g., flatness, roundness, parallelism) and deviations.
        iii. Enables integration of CAD (Computer-Aided Design), CAE (Computer-Aided Engineering), and CAM (Computer-Aided Manufacturing) data.
        iv. AP203 and AP214 are earlier implementations, while AP242 focuses on modern applications.
    b. ISO 1101: Standard for Geometrical Product Specifications (GPS), covering the representation of tolerances and deviations for features.
    c. ISO 14649 (STEP-NC): Focuses on manufacturing data but incorporates geometric features and deviations for NC (Numerical Control) programming.
2. File Formats for Geometric Data
    a. STL (Stereolithography):
        i. Common in 3D printing and CAD, used for representing the surface geometry of objects.
        ii. Does not inherently manage tolerances or deviations but can be combined with metadata.
    b. OBJ (Wavefront):
        i. A geometry-focused format that stores 3D model data, including vertices, edges, and surfaces. Can include deviations as annotations.
    c. PLY (Polygon File Format):
        i. Stores geometric data and associated properties (e.g., color, normal vectors) and can be used for point clouds representing deviations.
    d. AMF (Additive Manufacturing Format):
        i. Enhances STL by including information about material properties, tolerances, and deviations.
3. Standards for Deviations in Measurement
    a. GD&T (Geometric Dimensioning and Tolerancing):
        i. Not a data format but a set of symbolic language rules (ANSI Y14.5, ISO 1101) for defining allowable deviations from ideal geometry.
        ii. Can be implemented in software that adheres to STEP, QIF, or CAD.
    b. Point Cloud Data Formats (e.g., LAS, E57):
        i. Used in laser scanning and metrology to record deviations from ideal geometries.
        ii. Often combined with statistical or spatial analysis tools to assess deviations.
    c. FARO and Leica Formats: Proprietary formats from metrology equipment.

3.1.4. CAD based Design

Depending on end-user preferences and software availability, any number of CAD software tools may be employed during the design process. Each tool will generally make use of their own proprietary file format to save file data. However, geometry import and export in a range of commonly used formats is also widely maintained. Here, this allows interoperability to be maintained between the various upstream and downstream stages in the digital workflow (Process Map).

STEP CAD input: STEP, IGES, STL to mention a few of the most common formats. Other formats are also supported.

STEP file, or a Standard for the Exchange of Product model data file, is a standardized CAD file format used for exchanging 3D data between different computer-aided design (CAD) software applications. It is widely supported by various CAD software platforms.

Here are some key features about STEP files:

- Neutral Format: STEP files are considered neutral file formats as they are designed to be platform independent and facilitate interoperability between different CAD software programs. They provide a common standard for exchanging 3D models, ensuring that data can be easily shared and accessed by users with different CAD systems.
- Geometry and Attributes: A STEP file contains information about the 3D geometry of a model, including its shape, dimensions, and structure. It can also include additional attributes such as material properties, annotations, assembly relationships, and other relevant metadata.
- Compatibility: Most CAD software applications can import and export STEP files. This compatibility allows users to exchange 3D model data easily, even if they are working with different software platforms. It also enables collaboration and communication between designers, engineers, and manufacturers using different CAD tools.
- Loss of Information: While STEP files provide a means of sharing and transferring 3D models, it’s important to note that some advanced features and specific design details may be lost or simplified during the conversion process. This loss of information can occur due to the differences in the capabilities and functionalities of different CAD software applications.
- File Size: The size of a STEP file can vary depending on the complexity of the model. In general, STEP files tend to be larger than native CAD file formats due to their comprehensive nature and inclusion of additional data.
- Visual Representation: To view and work with STEP files, CAD software or specialized STEP file viewers are required. These applications provide tools for viewing, analysing, and modifying the 3D model within the STEP file.

.

3.1.5. VMAP CAE format for physics based simulation and experimental data

The VMAP Standard focusses on gaining a common understanding and interoperable definitions for the modelling of materials and manufacturing processes and generating universal concepts and open software interface specifications for the exchange of simulation results information in CAE workflows. It is an interface standard for integrating multi-disciplinary and multi-software simulation processes in the manufacturing industry along with I/O routines, which can be integrated in any CAE Software. VMAP is a vendor-neutral standard for CAE data storage and transfer to enhance interoperability in virtual engineering workflows. The VMAP interface and transfer file relies on the HDF5 technology. The Hierarchical Data Format (HDF) implements a model for managing and storing data.

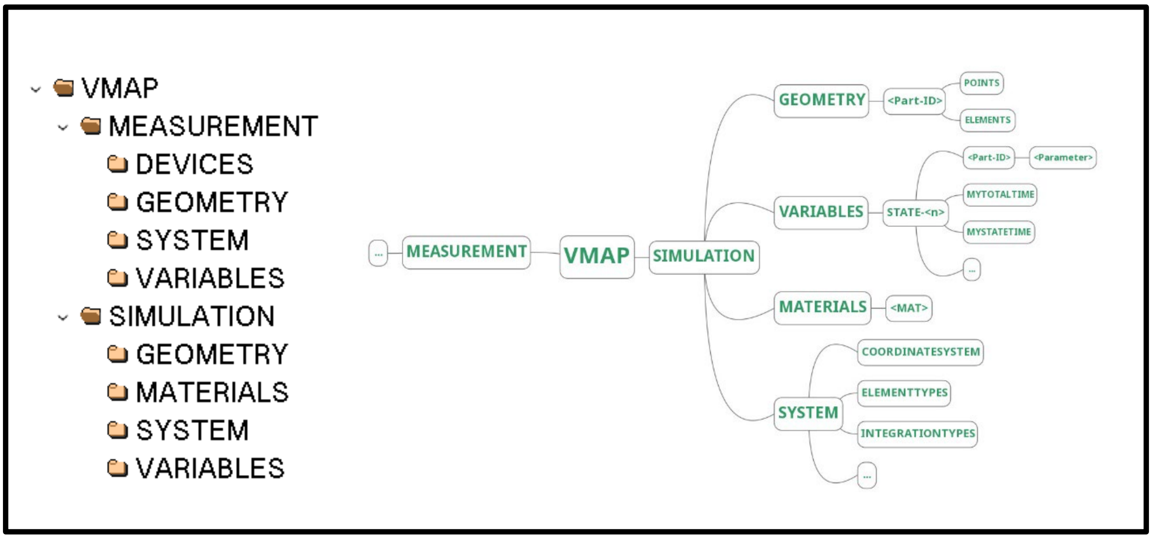


Figure 2 VMAP Data Hierarchy.

CAE Simulation Data

Many of the commonly used CAE tools can already store data in VMAP Standard format. Some of those include BETA CAE Ansa, Abaqus, OpenFoam, ANSYS Mechanical, e-Xstream Digimat. MpCCI Mapper, OpenFOAM, etc.

Within SmartEM, we see that the simulations which are being carried out using typical FEA or CFD codes can be stored in the VMAP Standard format, and the data can then be transferred for further steps like for visualisation purposes.

Data from physical Measurement and Machine Monitoring

The VMAP sensor data storage group aims to be able to standardize the storage of measured and experimental data within the manufacturing industry. The group consists of industrial partners with use-cases coming from plastic and steel industry, and aerospace domain. A clear application from the blow moulding domain, where the stereography and thermography data need to be incorporated into the validation process along with the simulation data. This use case needs a standard format to store both test and simulation data, to carryout validation process without any loss of information. This kind of requirement is also seen in the Pioneer project, where there is a testing data coming from (1) curing validation using DSC (Dynamic Scanning Calorimetry), (2) material composition analysis using TGA (Thermogravimetry), and (3) material degradation temperature obtention also with TGA etc. For all these data, .csv file is generally the storage format, however, it is not very well organized, and it is difficult for others (non-domain experts) to read and understand the data without proper guidelines.

Full Model Descriptions

The full model group aims to store boundary conditions in the VMAP file so that, in the future, a simulation can be directly initiated using the VMAP file. This group consists of partners from ISV, automotive, aero-industry and research. One of the applications from this working group requires storage of numerical data, like boundary conditions, in the VMAP Standard. The jet engine design requires coupling of various tool, standardized data exchange and support for multi-fidelity data. Integration of all such data components into the VMAP Standard will provide, such complex use cases, a comprehensive data exchange format. Even though applications from this group cannot be directly seen in the Pioneer project right now, we would take the use cases of Pioneer project as examples for this group. This would provide a larger perspective to the working groups for further extensions to the standard.

3.1.6. Standards and Protocols in Production

Data standards are rules and protocols that ensure data conforms to a specific format or structure for consistency, compatibility, and shareability.

- **ISO 9001**: This standard is a general standard for quality management systems. It ensures that data such as production data, process documentation, quality records, etc., are recorded in a specific format and presented consistently.
- **OPC-UA** (Open Platform Communications - Unified Architecture): This is a communication protocol used in industrial automation and control systems. OPC-UA enables different production systems to communicate with each other and exchange data.
- **MTConnect**: This is an open communication protocol used for data sharing between manufacturing machines. It enables the transfer of information such as machine status, operational data, sensor data, etc., in a standard format.
- **ISO 15926**: This standard provides a data model and definition standards for information management of industrial plants. It ensures that different data types and objects are represented and linked in a standardized manner.
- **STEP** (Standard for the Exchange of Product Model Data): STEP is a standard that enables the standardization of data used in product design and manufacturing processes. It ensures that data such as geometry, material properties, machining instructions, assembly information, etc., are shared in a consistent format.

These data standards facilitate data integration, reduce errors, and improve data quality by ensuring the standardization of data used across various industries and business processes. Adherence to appropriate data standards is particularly crucial in complex systems like digital twin factories. However, the specific data standards required for a particular industry or application should be determined based on industry standards, regulations, and best practices.

3.2. Terminology for Materials Characterisation and Modelling

3.2.1. MODA - Materials modelling Terminology

It has been demonstrated in many individual cases that materials modelling is a key enabler of research & development efficiency and innovation and that the use of this technology can generate a huge economic impact [1].

Due to the huge variety and complexity of materials and the wide range of applications the materials modelling field consists of several communities. These communities have established different terminologies which typically focus on specific application domains and on particular types of models. As a result, a wide range of domain specific software codes have evolved. However, applications to industrial problems in advanced materials and nanotechnology require a strong interdisciplinary approach among these fields and communities. There is therefore a need to establish a common terminology (definition of concepts and vocabulary) in materials modelling.

A standardized terminology will improve future exchanges among experts in the entire area of materials modelling, facilitate the exchange with industrial end-users and experimentalists and reduce the barrier utilizing materials modelling. The common language is expected to foster dialogue and mutual understanding between industrial end-users, software developers, scientists and theoreticians. Standardization of terminology and classification has been identified as critical to collaboration in and dissemination of European research projects. In particular, standards will facilitate interoperability between models and databases. The standardization is relevant for an integrated technological development and brings benefits for industrial end-users due to simplified and much more efficient communication in the field of materials simulation.

The classification helps translators by translating industrial problems into problems that can be simulated with materials models. It assists workflow development where several models can interoperate in addressing a specific end-user question.

In the future, these standardized terminology and classification can be formalized into a taxonomy and an ontology of materials modelling. Such an ontology will form the basis for formal metadata development with which models and databases can be linked. These developments will further support efficient solutions for materials modelling and the communication, dissemination, storage, retrieval and mining of data about materials modelling.

[1] CEN Workshop Agreement CWA 17284 https://www.cencenelec.eu/media/CEN-CENELEC/CWAs/RI/cwa17284_2018.pdf

3.2.2. CHADA - Materials characterisation Terminology

Materials characterisation involves the identification and measurement of properties that are either intrinsic or manifest in a material. These properties identify the type, manufacturing/process history, and the state of the material. Characterisation allows us to handle, transport, process, engineer and use the material in the intended application. Characterisation methods can be divided into two broad categories; a) those used to identify the nature (structure, chemistry, microstructure, etc) of the material and b) those evaluating material behaviour and/or performance.

The production of materials presents many challenges, particularly in the industrial environment. Real time monitoring of materials during synthesis and processing is desired for obtaining e.g., high precision, small specialty batches, processing monitoring of nucleation/growth at different scales (lab, pilot, production) and potential for feedback loops (adapt T, pH, etcetera). Such latest developments have widened the application domains for materials characterisation and increased its potential impacts to economy and society. [2]

CEN Workshop Agreement CWA 17815 [2] https://www.cencenelec.eu/media/CEN-CENELEC/CWAs/ICT/cwa17815.pdf

3.3. Digital Twin System Standardization

Several standards and factors need to be considered when creating a digital twin factory system. These are grouped under eight main headings as follows.

- **Data Standards**: Data standardization is essential for feeding the digital twin factory with accurate and reliable data. Data needs to be defined and shared in a consistent, compatible, and understandable manner. This includes using standard data formats, tagging protocols, and standards for data integration.
- **Integrated Systems**: Seamless integration between different systems must be ensured within the digital twin factory. Standard protocols and interfaces should be used to enable compatible operation between production equipment, automation systems, sensors, software, and other components.
- **Data Security and Privacy**: Data security and privacy are crucial as the digital twin factory houses sensitive production data. Standards and best practices such as encryption, access controls, and security measures should be employed to protect data from unauthorized access.
- **Configuration Management**: The digital twin factory requires proper configuration and management of production systems and equipment. Standardized configuration management processes and tools should be used to ensure systems are configured correctly and kept up-to-date.
- **Data Analytics**: Data analytics in the digital twin factory encompasses aspects such as production performance monitoring, fault detection, and identification of optimization opportunities. Standard data analysis methods, algorithms, and reporting tools should be used for data analytics.
- **Industrial Internet of Things Connectivity**: In the digital twin factory, the Industrial Internet of Things (IIoT) enables continuous communication between devices and systems. Standard protocols and communication standards should be used for IIoT connectivity.
- **Human-Machine Interaction**: Human-machine interaction is important in the digital twin factory. User interfaces, interactive systems, and collaborative platforms should be designed in a user-friendly and standardized manner.
- **Model Validation and Verification**: The digital twin model needs to be validated and verified against the real-world system to ensure accuracy and reliability. This includes using simulation and other techniques to test the model under different operating conditions.

Paying attention to these standards and factors when creating a digital twin factory will ensure that the system operates efficiently, securely, and sustainably. Industry standards and regulations should also be taken into account and complied with.

3.4. Model-based Engineering related Standards

3.4.1. FMI Functional Mockup Interface

The FMI (Functional Mock-up Interface) standard is an open, vendor-neutral standard designed to facilitate the exchange and integration of dynamic system models between different simulation tools. Developed by the Modelica Association, it is widely used in multi-domain simulations and co-simulation scenarios.

- Key Features of FMI
  - Model Exchange and Simulation: Models developed in Unity can be packaged according to FMI standards and appropriate converter libraries or applications can be written to ensure seamless data exchange with other simulation tools.
  - Co-Simulation: Unity's simulation capabilities can be combined with other tools, for example, running a physics engine (Unity PhysX) synchronously with a mechanical system.
  - Cross-Platform Compatibility: Models are packaged in a standardized format as FMUs (Functional Mock-up Units), which include model equations, parameters, and metadata.
  - Wide Application Range: Used in industries like automotive, aerospace, energy, and robotics for system-level modeling, control system development, and virtual testing.
- Benefits of FMI
  - Interoperability: Allows seamless collaboration between tools and teams.
  - Reusability: FMUs can be used across different projects and software environments.
  - Scalability: Suitable for both simple and complex system simulations.

3.4.2. SmartSE – Semantic annotation for system models

The SmartSE (Smart Systems Engineering) approach is a methodology designed to support the development of smart, connected, and multi-disciplinary systems. It integrates model-based systems engineering (MBSE), data management, and simulation techniques to address the increasing complexity of modern systems, such as IoT devices, cyber-physical systems, and autonomous systems. The SmartSE approach combines traditional systems engineering with advanced tools and methods to handle the challenges of smart system development efficiently.

Key Characteristics of SmartSE

- System Lifecycle Integration: Covers the entire lifecycle from requirements engineering to design, implementation, testing, and maintenance.
- Model-Based Approach: 3D models used in Unity can be enriched with semantic annotations, specifying the functions of individual model components. Objects in Unity scenes can be semantically tagged with ScriptableObject based data structures or an external metadata management system (e.g. JSON/XML based) can be used .
- Interoperability and Data Exchange: Ensures seamless communication between tools, models, and stakeholders through standardized interfaces and data formats.
- Collaboration Support: Facilitates collaboration among multi-disciplinary teams by integrating knowledge and workflows across engineering domains.
- Real-Time Decision Making: By combining Unity's real-time simulation capabilities with semantic richness, decision-support systems can be developed.
- Focus on Smart Systems: Emphasizes the inclusion of intelligent features such as adaptive behavior, connectivity, and real-time decision-making.

3.4.3. SSP - System Structure and Parameterization

The SSP (System Structure and Parameterization) standard is an open, vendor-neutral standard developed to facilitate the exchange of structured system models between simulation tools. It is primarily designed to address the challenges of managing modular, hierarchical, and multi-domain system architectures in simulation environments.

- Key Features of SSP
  - System Architecture Exchange: Describes the hierarchical structure of a system composed of interconnected simulation models or components.
  - Hierarchical Structure Management: Unity scenes and objects can be organized hierarchically using the SSP's system description file (SSD), with the development of appropriate converter libraries
  - Integration with FMI: Built to work seamlessly with the Functional Mock-up Interface (FMI) standard, allowing the integration and orchestration of FMUs (Functional Mock-up Units) within a system.
  - Parameter Management: SSP's parameter sets can offer a centralized management system for variables used in Unity projects. Parameters such as physical properties, production parameters or scenario variables can be transferred to objects in the Unity scene via SSP parameter files. Thus, model changes are managed from a single center.
  - Tool Interoperability: Facilitates the transfer of system structures and configurations between different simulation and modeling tools.
  - XML-Based Format: Utilizes XML for representing system structure and configuration, ensuring human-readable and machine-readable descriptions.
- Core Components
  - System Description (SSD): Defines the topology of the system, including components, connections, and hierarchical structure.
  - Parameter Set (SSP): Captures and manages parameters for components and subsystems.
  - Mapping and Variants (SSV and SSD-V): Supports mapping parameters to different configurations and defining system variants.
- Applications
  - Multi-Domain Simulations: Managing complex systems involving mechanical, electrical, thermal, and software components.
  - Collaborative Engineering: Facilitating teamwork by enabling clear, modular system descriptions.
  - Reusability: Simplifies reusing and sharing system architectures and parameterizations across projects.

3.4.4. Further Standards

Several standards support Model-Based Systems Engineering (MBSE) by providing frameworks, languages, and guidelines for creating, managing, and exchanging system models across various domains. These standards promote interoperability, collaboration, and consistency in MBSE practices. Here are some key standards for MBSE:

- SysML (Systems Modeling Language)
  - Overview: SysML is a modeling language specifically designed for systems engineering. It is a subset and extension of UML (Unified Modeling Language), tailored to describe complex systems' structure, behavior, requirements, and constraints. Functional descriptions of Unity models can be associated with standards such as SysML, adding semantic meaning. Structural and behavioral models in SysML diagrams can be used by Unity as automatic scene generation or behavior definitions of objects using formats such as XMI (XML Metadata Interchange).
  - Key Features: Supports modeling of requirements, behavior, structure, and parametrics. Provides diagrams such as Use Case, Activity, Block Definition, and Requirement diagrams.
  - Applications: Widely used in industries like aerospace, defense, automotive, and healthcare for systems engineering projects.
- ISO/IEC 15288
  - Title: *Systems and Software Engineering – System Life Cycle Processes.*
  - Standards like ISO/IEC 15288 can help manage the lifecycle processes of Digital Twin Factory projects in Unity more systematically:
    - Planning and Development: Unity projects can be developed in alignment with lifecycle standards.
    - Testing and Maintenance: Unity's real-time simulation features can be aligned with these standards to create more effective testing and maintenance workflows. During the design and verification phases, the use of Unity-based simulation can be integrated with the 'Verification & Validation' processes of ISO/IEC 15288
  - Key Features: Provides a high-level framework for systems engineering processes. Focuses on process integration and lifecycle management rather than specific modelling languages.
  - Applications: Used to guide the implementation of MBSE methodologies in a structured lifecycle context.
- ISO 10303 (STEP – Standard for the Exchange of Product Model Data)
  - Overview: A standard for representing and exchanging product data models, including system models.
  - Key Features: Supports data exchange and interoperability for system models between tools. AP233 is specifically focused on systems engineering data.

  - Applications: Applied in domains like automotive and aerospace for integrating MBSE data across teams and tools.
- OMG UML (Unified Modeling Language)
  - Overview: A general-purpose modeling language, used as the foundation for SysML. Though not specific to systems engineering, it supports MBSE when extended with domain-specific profiles.
  - Applications: Used for early system architecture development and tool integration.
- ISO/IEC/IEEE 42010
  - Title: *Architecture Description of Systems and Software.*
  - Overview: Specifies how to describe and communicate system architectures effectively.
  - Key Features: Defines concepts like viewpoints, views, and stakeholder concerns. Helps in creating consistent architectural models across different disciplines.
  - Applications: Used for managing architectural complexity in large, multi-disciplinary projects.
- OMG DDS (Data Distribution Service)
  - Overview: A standard for real-time data exchange in distributed systems, supporting MBSE when modeling communication between system components.
  - Applications: Used in real-time systems, such as autonomous vehicles or IoT ecosystems.
- AP242
  - Title: *Managed Model-Based 3D Engineering.*
  - Overview: A STEP standard supporting the integration of 3D models with system-level data, bridging the gap between MBSE and CAD/PLM systems.
  - Applications: Used in aerospace, automotive, and manufacturing industries for connecting physical product data with system models.
- DoDAF/MoDAF/NATO AF
  - Overview: Architectural frameworks for modeling and analyzing systems-of-systems, used by the U.S. Department of Defense (DoDAF), UK Ministry of Defence (MoDAF), and NATO.
  - Applications: Support MBSE by providing structure and guidelines for developing models of complex system architectures.
- OSLC (Open Services for Lifecycle Collaboration):

  - Facilitates integration and interoperability of lifecycle management tools, including those for MBSE.
- ISO 20944:
  - Meta-data structures in Unity can be organized according to this standard, enabling cross-system data exchange. The ISO 20944 framework can be a guide for organizing metadata in Unity project files and sharing data compatible with different platforms. For example, asset inventory (prefabs, props, materials, etc.) can be tagged according to ISO 20944 principles and mapped to corporate databases
- Recommended Contributions for Digital Twin Factory in Unity:
  - Standards-Based Modeling Guidelines: Develop documentation based on SSP, FMI, and SmartSE standards for models used in Unity.
  - Semantic Integration: Create a standard defining how semantic annotations should be applied to Unity scenes and models.
  - Simulation Workflows: Establish workflows adhering to FMI and SSP standards to enhance the multi-domain functionalities of the Digital Twin Factory.
  - API and Tool Development: Extend Unity's API to ensure compliance with these standards.

3.4.5. Standards for meta-parameters in MBSE

Standards addressing meta-parameters in Model-Based Engineering (MBE) focus on defining, managing, and exchanging metadata or meta-parameters that describe the structure, context, and configuration of system models. These meta-parameters are essential for ensuring model reusability, interoperability, and traceability across different tools and disciplines.

- ISO/IEC 19510 (BPMN) and SysML Metadata
    - SysML Metadata:
        - SysML models often include meta-parameters such as requirements IDs, constraint definitions, and parametric relationships.
        - Metadata is typically used to manage dependencies between different model elements.
    - BPMN Metadata:
        - Focuses on metadata for processes, which can include operational parameters and contextual information for decision-making.
- Open Services for Lifecycle Collaboration (OSLC)
    - Focus: Defines metadata standards for lifecycle artifacts, enabling interoperability between engineering tools.
    - Meta-Parameter Scope: Supports linking, traceability, and querying of lifecycle artifacts, including requirements, designs, test cases, and models. Common Resource Shapes (e.g., Requirements, Change Requests) describe meta-parameters.
    - Applications: Used in MBSE for managing traceability and interconnections between models and their context.
- ISO/IEC 20944 (Metadata Registry Standards)
    - Focus: Managing and exchanging metadata in data models.
    - Meta-Parameter Scope: Defines structures for registering and describing metadata, including their semantics and relationships. Facilitates interoperability by standardizing descriptions of meta-parameters in complex systems.
- Metadata in Architecture Frameworks (e.g., DoDAF, MoDAF, TOGAF)
    - Focus: Architectural modeling and analysis for systems-of-systems.
    - Meta-Parameter Scope: Metadata includes architectural elements like stakeholders, views, constraints, and traceability links. Parameters define the context, purpose, and relationships of different model components.
    - Applications: Managing metadata for complex systems in defense, aerospace, and enterprise architecture.
- OASIS Universal Data Element Framework (UDEF)

  - Focus: Standardized metadata descriptions for data elements in engineering and beyond.
  - Meta-Parameter Scope: Provides a controlled vocabulary and hierarchical classification for defining meta-parameters.
  - Applications: Used to standardize metadata across organizations and tools.
- IEEE 1685 (IP-XACT)
  - Focus: Metadata for electronic and embedded systems models.
  - Meta-Parameter Scope: Describes meta-parameters for hardware and software models, such as interface definitions, constraints, and configurations.
  - Applications: Used in embedded systems engineering and electronic design automation (EDA).

These standards provide frameworks for defining and managing meta-parameters that describe the context, assumptions, and configuration of models in MBE workflows. They are essential for ensuring consistent, interoperable, and traceable model management across tools and domains.

3.5. Semantic data for Digital Product Passport

The foundation for the Digital Product Passport (DPP) data model is the Semantic Aspect Meta Model (SAMM) developed by the Catena-X initiative. The SAMM provides a standardised ontology for representing various aspects of a product's lifecycle, supply chain, and associated data. Modelling product-related data using this SAMM ensures a consistent and standardized representation of the information required for a Digital Product Passport.

Based on the provided SAMM models, a JSON Schema can be generated to define the structure and validation rules for the DPP data. This JSON Schema will be the foundation for the DPP's data model, ensuring consistency and interoperability across all DPP instances. The JSON Schema should include all the required data.

With the JSON Schema, the DPP solution can be developed to consume and validate the data according to the Catena-X standards. The JSON Schema will serve as the blueprint for the DPP's data model, guiding the development of data storage, retrieval, and exchange mechanisms. To align with the global standard of DPP, we are trying to follow the standard provided by Catena-X. By aligning the DPP's data model with the SAMM models and the corresponding JSON Schema, it can ensure that the DPP solution is fully compatible with the Catena-X standards, enabling seamless integration and data sharing across the product's ecosystem.

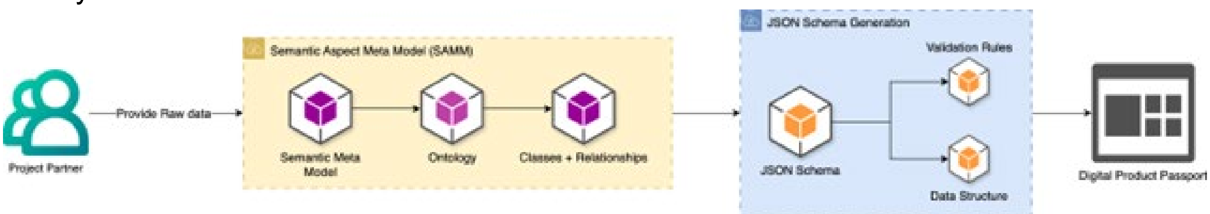


Figure 3 Semantic Data Integration System for Digital Product Passport for RESTORE use cases.

3.5.1. Catena-X standards

Catena-X standards[3] are a set of rules and requirements that govern how data and information are exchanged in the Catena-X data ecosystem. These standards ensure that technologies, components, and processes are developed and operated according to uniform rules. They enable interoperability between independent implementations. Catena-X standards enable a high level of transparency and comparability for all providers of services and applications. They enable data sovereignty and security for all participants in the data space.

3.5.2. Digital Product Passports (DPPs) and Catena-X standards

Catena-X plays a crucial role in enabling Digital Product Passports (DPPs) by providing a standardized framework and data-sharing ecosystem. The DPP concept is implemented through interoperable platforms like Catena-X, which ensure that detailed product-related information—such as (i) manufacturing data, (ii) materials and their provenance, (iii) lifecycle impacts (iv) disassembly & recycling guidelines etc. can be securely shared among stakeholders like manufacturers, suppliers, consumers, recyclers, and regulators.

For example, Catena-X supports the "Battery Passport," a specific type of DPP focusing on sustainability and traceability within the battery lifecycle. This initiative utilises Catena-X's

[3] https://catenax-ev.github.io/docs/standards/overview

shared services, such as its standardized data models and secure data exchange protocols, to provide end-users with accurate and auditable product data. These passports enhance transparency and facilitate informed decisions across the product lifecycle, from design and production to recycling and reuse.

3.5.3. EClass – Classification of Products and Services

ECLASS (formerly styled as eCl@ss) is a data standard for the classification of products and services using standardized ISO-compliant properties. The ECLASS Standard enables the digital exchange of product master data across industries, countries, languages or organizations. Its use as a standardized basis for a product group structure or with product-describing properties of master data is particularly widespread in ERP systems.

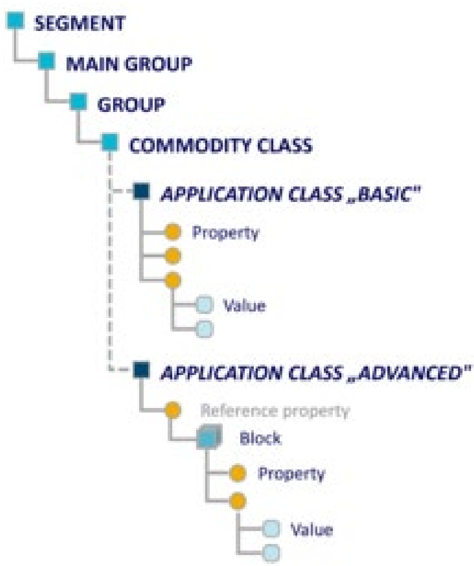
As an ISO-compliant and the world's only property-based classification standard, ECLASS also serves as a "language" for Industry 4.0 (IOTS).

The ECLASS standard is a hierarchical system, similar to the UNSPSC classification system, for grouping products and services. It consists of four levels of hierarchy (classes): Segment (Level 1), Main Group (Level 2), Group (Level 3) and Subgroup (Level 4). The hierarchy shows that a superordinate class comprises its subordinate classes.

The tree structure illustrates the standardized and comparable structure of data. The nodes of the tree structure are collectively referred to as material classes. On the 4th level (subgroup), ECLASS provides so-called property lists. Properties enable the detailed description of products and services in the associated master data and thus enable searching in the various catalogs. The properties are defined by values. Attached keywords and synonyms are used to quickly find the product classes and their property lists.

In summary, the system consists of the following elements:

- Classes - the classes or product groups allow products to be grouped and organized in this way.
- Keywords - keywords assigned to the individual classes simplify and standardize the search for products (e.g. product group "chairs" is also found with search terms such as "seat" or "office chair").
- Properties - Properties are additional product attributes that can only be used meaningfully for products in a specific class, for example the power of light bulbs or the diameter of tubes. The aim is to incorporate these properties into standardization, i.e. DIN, EN, ISO, DKE/IEC.
- Values - values specify the value range for the properties.



- Units - based on DIN and ECE units to specify the unit of the properties.

3.6. Standards in AI and ML

Artificial Intelligence (AI) and Machine Learning (ML) are pivotal technologies driving advancements across industries. Their integration into various domains requires standardized frameworks to ensure transparency, reliability, ethical compliance, and interoperability. This section summarizes the key standards categorized below.

3.6.1. ONNX – for data models

ONNX is an open format built to represent machine learning models. ONNX defines a common set of operators - the building blocks of machine learning and deep learning models - and a common file format to enable AI developers to use models with a variety of frameworks, tools, runtimes, and compilers. ONNX is a community project. The active community thrives under an open governance structure, which provides transparency and inclusion.

- Interoperability: Develop in your preferred framework without worrying about downstream inferencing implications. ONNX enables you to use your preferred framework with your chosen inference engine.
- Hardware Access: ONNX makes it easier to access hardware optimizations. Use ONNX-compatible runtimes and libraries designed to maximize performance across hardware.

3.6.2. General Frameworks and Terminology

AI and ML frameworks establish foundational principles, shared terminology, and system architectures. These standards ensure a consistent understanding of AI systems across stakeholders, reducing ambiguity and facilitating collaboration.

Standard Code	Title	Description	Focus Area	Organization
ISO/IEC 22989:2022	Information technology-Artificial intelligence- Artificial intelligence concepts and terminology	Defines foundational concepts and terminology for AI.	Terminology	ISO/IEC
ISO/IEC 23053:2022	Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)	Provides a structured framework for ML systems.	ML Framework	ISO/IEC
ISO/IEC 38507	Information technology-Governance of IT-Governance implications of the use of artificial intelligence by organizations	Governance guidelines integrated with IT frameworks.	AI Governance	ISO/IEC
IEEE P3123	Standard for Artificial Intelligence and Machine Learning (AI/ML) Terminology and Data Formats	Defines terminology and data formats in AI systems.	Terminology	IEEE
ISO/IEC 25059:2023	Software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality model for AI systems	Outlines quality model for AI systems, providing consistent terminology for specifying, measuring, and evaluating AI system quality.	Quality Model	ISO/IEC
ISO/IEC TR 24030:2024	Information technology - Artificial intelligence (AI) - Use cases	Provides a collection of representative use cases of AI applications across various domains, illustrating the applicability and potential of AI in different sectors.	AI Applications	ISO/IEC

3.6.3. Data Quality and Data Readiness

AI systems rely on high-quality, well-structured, and unbiased data for effective performance. These standards focus on ensuring data integrity, quality, and readiness is vital for building AI systems that are reliable, transparent, and free from systemic biases.

Standard Code	Title	Description	Focus Area	Organization
ISO/IEC 5259-1:2024	Artificial intelligence-Data quality for analytics and machine learning (ML)	Data quality metrics for analytics and ML systems.	Data Quality	ISO/IEC
ISO/IEC TR 24027:2021	Information technology-Artificial intelligence (AI) - Bias in AI systems and AI aided decision making	Framework for bias identification and mitigation.	Bias Mitigation	ISO/IEC
FAIR Data Principles	FAIR Data Framework	Ensures datasets are Findable, Accessible, Interoperable, and Reusable.	Data Principles	Global Initiative
ISO/IEC 8183:2023	Information technology - Artificial intelligence - Data life cycle framework	Guidelines for managing the AI data lifecycle.	Data Governance	ISO/IEC

3.6.4. Trustworthiness, Security, and Privacy

AI systems must be trustworthy, secure, and privacy-compliant to earn user confidence and ensure ethical deployment. Trust and security frameworks minimize risks, ensure data privacy, and foster confidence in AI-driven decision-making.

Standard Code	Title	Description	Focus Area	Organization
ISO/IEC 23894:2023	Information technology - Artificial intelligence - Guidance on risk management	Framework for AI-specific risk management.	Risk Management	ISO/IEC
ISO/IEC TR 24028:2020	Information technology - Artificial intelligence - Overview of trustworthiness in artificial intelligence	Guidelines for trustworthiness in AI systems.	Trustworthiness	ISO/IEC
IEEE P7002-2022	IEEE Standard for Data Privacy Process	Focuses on privacy considerations in AI systems.	Privacy	IEEE
NIST AI RMF 1.0	AI Risk Management Framework	Comprehensive risk management framework for AI systems.	Risk Management	NIST

3.6.5. Ethical and Societal Concerns

Ethical standards ensure AI deployment aligns with societal values, human rights, and legal frameworks. These standards safeguard human rights, mitigate algorithmic biases, and establish ethical benchmarks for AI.

Standard Code	Title	Description	Focus Area	Organization
ISO/IEC TR 24368:2022	Information technology-Artificial intelligence - Overview of ethical and societal concerns	Ethical and societal principles for AI deployment.	Ethical AI	ISO/IEC
IEEE 7003-2024	IEEE Approved Draft Standard for Algorithmic Bias Considerations	Guidelines for mitigating algorithmic biases.	Bias Mitigation	IEEE
EU AI Act	EU AI Regulatory Framework	Legislative AI ethics and accountability framework.	Regulation	EU

3.6.6. Performance Evaluation and Testing

Evaluation standards ensure AI systems perform reliably across diverse operational environments and use cases. These standards validate AI robustness, ensure transparency in results, and improve reliability in diverse applications.

Standard Code	Title	Description	Focus Area	Organization
ISO/IEC TR 24029-1:2021	Artificial Intelligence (AI) - Assessment of the robustness of neural networks	Robustness evaluation for AI neural networks.	Robustness	ISO/IEC
ISO/IEC TS 4213:2022	Information technology - Artificial intelligence - Assessment of machine learning classification performance	Guidelines for ML classification performance.	Testing	ISO/IEC

3.6.7. AI System Lifecycle and Governance

Governance frameworks guide AI deployment throughout its lifecycle, from conception to decommissioning. Lifecycle governance standards ensure transparency, accountability, and alignment with long-term organizational goals.

Standard Code	Title	Description	Focus Area	Organization
ISO/IEC 5338:2023	Information technology - Artificial intelligence - AI system life cycle processes	Standards for AI lifecycle governance.	Lifecycle	ISO/IEC
ISO/IEC 42001:2023	Information technology - Artificial intelligence - Management system	AI governance and risk management system standard.	Governance	ISO/IEC

3.7. Process Control in Production Environments

3.7.1. AAS Asset Administration Shell

The Asset Administration Shell is the pilar component of Industry 4.0 and its scope is to create a digital representation of the assets within the industrial ecosystem. The administration shell technology however as the name may suggest is responsible for managing the asset's behaviours and regulate any interaction among itself and external components. As such, engineers exploit the benefits of interoperability as it can act as a “shell” for interacting with the asset in a protocol-unknown manner. The AAS itself is consisted of a standardized meta-model to describe the asset information (either dynamic or static) in a standardized format usually serialized in XML, JSON, or even RDF. However, what makes the AAS functional in terms of interacting with dynamic information and controlling several asset behaviours is the industry 4.0 platform serving the functionalities defined by the Asset Administration Shell standards. In this way, connection between the asset and its administration shell are established and customized based on the underlying technology of the asset. OPC UA connection is one of the primary ways to establish a stable and active bidirectional communication between industrial assets and the hosting platform of the administration shell. As such, although an OPC UA connector can facilitate the information exchange, the administration shell acts as a proxy for redirecting that information to the requested external component following a specified information format defined by its standard. Similar connection can be established with technologies such as MQTT, REST, or even IDS connectors, depending on the capabilities of the platform and the associated industrial equipment.

An open challenge however in the Industry 4.0 research and particularly in the AAS developments, is the definition of a standardized model for the different types of equipment involved into the process. Although AAS is supported by multiple standards, and despite the AAS meta-model is already standardized, the model itself provides an extensive level of flexibility while designing the administration shell of a specific process or equipment. This is one of the main reasons that multiple administration shell models can be found describing the same type of asset yet providing slightly different characteristics and capabilities. AutomationML can potentially be used as data representation for engineering information, thus defining the template for the set of entities/ ontologies to include within the Asset Administration Shell. This can enable compatibility among AutomationML and AAS technologies and the creation of more mature AAS models.

3.7.2. Automation ML

AutomationML (Automation Markup Language, sometimes referred to as AML) is a neutral data format based on an XML schema for the storage and exchange of plant engineering information, which is provided as an open, vendor independent standard. In the context of PIONEER, the aim of AutomationML is to link modern engineering tools from different disciplines in the AM process, e.g., CAD design, simulation, mechanical plant engineering, HMI development, robot control etc.

Documents providing an extensive overview of the AutomationML concept are available from the AutomationML website[4]. For an informative introduction to the format, the reader is referred to the related webpages [5] .

AML stores information in an object-oriented manner to facilitate the representation of both physical and logical components in the data chain as data objects. Each object may itself be a container of other objects and therefore form part of a larger composition. Typical objects in plant automation describe topology, geometry, kinematics, and logic (logic comprising elements of sequencing, behaviour, and control). AML utilises the top-level data format CAEX, which interconnects the different data formats. Therefore, AML has inherent distributed document architecture.

CAEX enables an object-oriented approach where the semantics of system objects (interpretation of data objects and their meaning within the overall process chain) can be specified using roles defined and collected in role class libraries.

Interfaces between system objects can be specified using interfaces classes defined and collected in interface class libraries.

Classes of system objects can be specified using system unit classes (SUC) defined and collected in system unit class libraries.

Finally, the individual project objects are modelled in an instance hierarchy (IH) as a hierarchy of internal elements (IE) referencing both system unit classes they are derived from and role classes defining their semantics and interface objects used to interlink objects among each other or with externally stored information (e.g. COLLADA or PLCopen XML files).

By promoting a system where syntax (technical representation of data objects and the vocabulary of data exchange) and semantics are de-coupled, the AML format provides far more flexibility for adaptation and augmentation than either of the other XML formats described herein.

[4] AutomationML website, https://www.automationml.org

[5] <AutomationML/>: standardized data exchange in the engineering process of production systems, vol. 2018, http://www.unserebroschuere.de/automationml/WebView,

Appendix A List of abbreviations

Abbreviation	Explanation
ONNX	ONNX is an open format built to represent machine learning models.
SMILE	Smile is a simulation modelling language, that is based on a physical description of simulation models
FMI	The Functional Mock-up Interface is a free standard that defines a container and an interface to exchange dynamic simulation models using a combination of XML
SSP	System Structure and Parameterization is a Modelica Association Project
VMAP	*VMAP* is a vendor-neutral *standard* for CAE data storage to enhance interoperability in virtual engineering workflows
STEP	"*STEP*" stands for "*Standard* for the Exchange of Product model data".
BIM	Standard for Building information modeling
CAE	Computer Aided Engineering
MDO	Multidisciplinary Design Optimisation
CFD	Computational Fluid Dynamics
CHT	Conjugate Heat Transfer
LLM	Large Language Models
MBSE	Model-based Systems Engineering
UC	Use Case
PLM	Product Lifecycle Management
ML	Machine Learing